» Data Source: aws_arn

Parses an Amazon Resource Name (ARN) into its constituent parts.

» Example Usage

```
data "aws_arn" "db_instance" {
  arn = "arn:aws:rds:eu-west-1:123456789012:db:mysql-db"
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Required) The ARN to parse.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `partition` - The partition that the resource is in.
- `service` - The service namespace that identifies the AWS product.
- `region` - The region the resource resides in. Note that the ARNs for some resources do not require a region, so this component might be omitted.
- `account` - The ID of the AWS account that owns the resource, without the hyphens.
- `resource` - The content of this part of the ARN varies by service. It often includes an indicator of the type of resource—for example, an IAM user or Amazon RDS database —followed by a slash (/) or a colon (:), followed by the resource name itself.

» Data Source: aws_availability_zone

`aws_availability_zone` provides details about a specific availability zone (AZ) in the current region.

This can be used both to validate an availability zone given in a variable and to split the AZ name into its component parts of an AWS region and an AZ identifier letter. The latter may be useful e.g. for implementing a consistent subnet numbering scheme across several regions by mapping both the region and the subnet letter to network numbers.

This is different from the `aws_availability_zones` (plural) data source, which provides a list of the available zones.

» Example Usage

The following example shows how this data source might be used to derive VPC and subnet CIDR prefixes systematically for an availability zone.

```
variable "region_number" {
  # Arbitrary mapping of region name to number to use in
  # a VPC's CIDR prefix.
  default = {
    us-east-1      = 1
    us-west-1      = 2
    us-west-2      = 3
    eu-central-1   = 4
    ap-northeast-1 = 5
  }
}

variable "az_number" {
  # Assign a number to each AZ letter used in our configuration
  default = {
    a = 1
    b = 2
    c = 3
    d = 4
    e = 5
    f = 6
  }
}

# Retrieve the AZ where we want to create network resources
# This must be in the region selected on the AWS provider.
data "aws_availability_zone" "example" {
  name = "eu-central-1a"
}

# Create a VPC for the region associated with the AZ
resource "aws_vpc" "example" {
  cidr_block = "${cidrsubnet("10.0.0.0/8", 4, var.region_number[data.aws_availability_zone.e
}

# Create a subnet for the AZ within the regional VPC
resource "aws_subnet" "example" {
  vpc_id     = "${aws_vpc.example.id}"
  cidr_block = "${cidrsubnet(aws_vpc.example.cidr_block, 4, var.az_number[data.aws_availabil
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available availability zones. The given filters must match exactly one availability zone whose data will be exported as attributes.

- `name` - (Optional) The full name of the availability zone to select.
- `state` - (Optional) A specific availability zone state to require. May be any of `"available"`, `"information"` or `"impaired"`.
- `zone_id` - (Optional) The zone ID of the availability zone to select.

All reasonable uses of this data source will specify `name`, since `state` alone would match a single AZ only in a region that itself has only one AZ.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `name` - The name of the selected availability zone.
- `region` - The region where the selected availability zone resides. This is always the region selected on the provider, since this data source searches only within that region.
- `name_suffix` - The part of the AZ name that appears after the region name, uniquely identifying the AZ within its region.
- `state` - The current state of the AZ.
- `zone_id` - (Optional) The zone ID of the selected availability zone.

» Data Source: aws_availability_zones

The Availability Zones data source allows access to the list of AWS Availability Zones which can be accessed by an AWS account within the region configured in the provider.

This is different from the `aws_availability_zone` (singular) data source, which provides some details about a specific availability zone.

» Example Usage

```
# Declare the data source
data "aws_availability_zones" "available" {
  state = "available"
}

# e.g. Create subnets in the first two available availability zones

resource "aws_subnet" "primary" {
  availability_zone = "${data.aws_availability_zones.available.names[0]}"

  # ...
}

resource "aws_subnet" "secondary" {
  availability_zone = "${data.aws_availability_zones.available.names[1]}"

  # ...
}
```

» Argument Reference

The following arguments are supported:

- `blacklisted_names` - (Optional) List of blacklisted Availability Zone names.
- `blacklisted_zone_ids` - (Optional) List of blacklisted Availability Zone IDs.
- `state` - (Optional) Allows to filter list of Availability Zones based on their current state. Can be either `"available"`, `"information"`, `"impaired"` or `"unavailable"`. By default the list includes a complete set of Availability Zones to which the underlying AWS account has access, regardless of their state.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `names` - A list of the Availability Zone names available to the account.
- `zone_ids` - A list of the Availability Zone IDs available to the account.

Note that the indexes of Availability Zone names and IDs correspond.

» Data Source: aws_billing_service_account

Use this data source to get the Account ID of the AWS Billing and Cost Management Service Account for the purpose of whitelisting in S3 bucket policy.

» Example Usage

```
data "aws_billing_service_account" "main" {}

resource "aws_s3_bucket" "billing_logs" {
  bucket = "my-billing-tf-test-bucket"
  acl    = "private"

  policy = <<POLICY
{
  "Id": "Policy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetBucketAcl", "s3:GetBucketPolicy"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::my-billing-tf-test-bucket",
      "Principal": {
        "AWS": [
          "${data.aws_billing_service_account.main.arn}"
        ]
      }
    },
    {
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::my-billing-tf-test-bucket/*",
      "Principal": {
        "AWS": [
          "${data.aws_billing_service_account.main.arn}"
        ]
      }
    }
  ]
}
POLICY
}
```

» Attributes Reference

- `id` - The ID of the AWS billing service account.
- `arn` - The ARN of the AWS billing service account.

» Data Source: aws_caller_identity

Use this data source to get the access to the effective Account ID, User ID, and ARN in which Terraform is authorized.

» Example Usage

```
data "aws_caller_identity" "current" {}

output "account_id" {
  value = "${data.aws_caller_identity.current.account_id}"
}

output "caller_arn" {
  value = "${data.aws_caller_identity.current.arn}"
}

output "caller_user" {
  value = "${data.aws_caller_identity.current.user_id}"
}
```

» Argument Reference

There are no arguments available for this data source.

» Attributes Reference

- `account_id` - The AWS Account ID number of the account that owns or contains the calling entity.
- `arn` - The AWS ARN associated with the calling entity.
- `user_id` - The unique identifier of the calling entity.

» Data Source: aws_ip_ranges

Use this data source to get the IP ranges of various AWS products and services. For more information about the contents of this data source and required JSON syntax if referencing a custom URL, see the AWS IP Address Ranges documention.

» Example Usage

```
data "aws_ip_ranges" "european_ec2" {
  regions  = ["eu-west-1", "eu-central-1"]
  services = ["ec2"]
}

resource "aws_security_group" "from_europe" {
  name = "from_europe"

  ingress {
    from_port        = "443"
    to_port          = "443"
    protocol         = "tcp"
    cidr_blocks      = ["${data.aws_ip_ranges.european_ec2.cidr_blocks}"]
    ipv6_cidr_blocks = ["${data.aws_ip_ranges.european_ec2.ipv6_cidr_blocks}"]
  }

  tags = {
    CreateDate = "${data.aws_ip_ranges.european_ec2.create_date}"
    SyncToken  = "${data.aws_ip_ranges.european_ec2.sync_token}"
  }
}
```

» Argument Reference

- `regions` - (Optional) Filter IP ranges by regions (or include all regions, if omitted). Valid items are `global` (for `cloudfront`) as well as all AWS regions (e.g. `eu-central-1`)
- `services` - (Required) Filter IP ranges by services. Valid items are `amazon` (for amazon.com), `cloudfront`, `codebuild`, `ec2`, `route53`, `route53_healthchecks` and `S3`.

NOTE: If the specified combination of regions and services does not yield any CIDR blocks, Terraform will fail.

- url - (Optional) Custom URL for source JSON file. Syntax must match AWS IP Address Ranges documention. Defaults to https://ip-ranges.amazonaws.com/ip-ranges.json.

» Attributes Reference

- cidr_blocks - The lexically ordered list of CIDR blocks.
- ipv6_cidr_blocks - The lexically ordered list of IPv6 CIDR blocks.
- create_date - The publication time of the IP ranges (e.g. 2016-08-03-23-46-05).
- sync_token - The publication time of the IP ranges, in Unix epoch time format (e.g. 1470267965).

» Data Source: aws_partition

Use this data source to lookup information about the current AWS partition in which Terraform is working.

» Example Usage

```
data "aws_partition" "current" {}

data "aws_iam_policy_document" "s3_policy" {
  statement {
    sid = "1"

    actions = [
      "s3:ListBucket",
    ]

    resources = [
      "arn:${data.aws_partition.current.partition}:s3:::my-bucket",
    ]
  }
}
```

» Argument Reference

There are no arguments available for this data source.

» Attributes Reference

- `partition` is set to the identifier of the current partition.
- `dns_suffix` is set to the base DNS domain name for the current partition (e.g. `amazonaws.com` in AWS Commercial, `amazonaws.com.cn` in AWS China).

» Data Source: aws_region

`aws_region` provides details about a specific AWS region.

As well as validating a given region name this resource can be used to discover the name of the region configured within the provider. The latter can be useful in a child module which is inheriting an AWS provider configuration from its parent module.

» Example Usage

The following example shows how the resource might be used to obtain the name of the AWS region configured on the provider.

```
data "aws_region" "current" {}
```

» Argument Reference

The arguments of this data source act as filters for querying the available regions. The given filters must match exactly one region whose data will be exported as attributes.

- `name` - (Optional) The full name of the region to select.
- `endpoint` - (Optional) The EC2 endpoint of the region to select.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `name` - The name of the selected region.
- `endpoint` - The EC2 endpoint for the selected region.
- `description` - The region's description in this format: ”Location (Region name)”.

» Resource: aws_accessanalyzer_analyzer

Manages an Access Analyzer Analyzer. More information can be found in the Access Analyzer User Guide.

» Example Usage

```
resource "aws_accessanalyzer_analyzer" "example" {
  analyzer_name = "example"
}
```

» Argument Reference

The following arguments are required:

- `analyzer_name` - (Required) Name of the Analyzer.

The following arguments are optional:

- `tags` - (Optional) Key-value mapping of resource tags.
- `type` - (Optional) Type of Analyzer. Valid value is currently only `ACCOUNT`. Defaults to `ACCOUNT`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Analyzer name.

» Import

Access Analyzer Analyzers can be imported using the `analyzer_name`, e.g.

```
$ terraform import aws_accessanalyzer_analyzer.example example
```

» Data Source: aws_acm_certificate

Use this data source to get the ARN of a certificate in AWS Certificate Manager (ACM), you can reference it by domain without having to hard code the ARNs as input.

» Example Usage

```
# Find a certificate that is issued
data "aws_acm_certificate" "example" {
  domain   = "tf.example.com"
  statuses = ["ISSUED"]
}

# Find a certificate issued by (not imported into) ACM
data "aws_acm_certificate" "example" {
  domain      = "tf.example.com"
  types       = ["AMAZON_ISSUED"]
  most_recent = true
}

# Find a RSA 4096 bit certificate
data "aws_acm_certificate" "example" {
  domain    = "tf.example.com"
  key_types = ["RSA_4096"]
}
```

» Argument Reference

- `domain` - (Required) The domain of the certificate to look up. If no certificate is found with this name, an error will be returned.
- `key_types` - (Optional) A list of key algorithms to filter certificates. By default, ACM does not return all certificate types when searching. Valid values are `RSA_1024`, `RSA_2048`, `RSA_4096`, `EC_prime256v1`, `EC_secp384r1`, and `EC_secp521r1`.
- `statuses` - (Optional) A list of statuses on which to filter the returned list. Valid values are `PENDING_VALIDATION`, `ISSUED`, `INACTIVE`, `EXPIRED`, `VALIDATION_TIMED_OUT`, `REVOKED` and `FAILED`. If no value is specified, only certificates in the `ISSUED` state are returned.
- `types` - (Optional) A list of types on which to filter the returned list. Valid values are `AMAZON_ISSUED` and `IMPORTED`.
- `most_recent` - (Optional) If set to true, it sorts the certificates matched by previous criteria by the NotBefore field, returning only the most recent one. If set to false, it returns an error if more than one certificate is found. Defaults to false.

» Attributes Reference

- `arn` - Set to the ARN of the found certificate, suitable for referencing in other resources that support ACM certificates.

» Resource: aws_acm_certificate

The ACM certificate resource allows requesting and management of certificates from the Amazon Certificate Manager.

It deals with requesting certificates and managing their attributes and life-cycle. This resource does not deal with validation of a certificate but can provide inputs for other resources implementing the validation. It does not wait for a certificate to be issued. Use a `aws_acm_certificate_validation` resource for this.

Most commonly, this resource is used to together with `aws_route53_record` and `aws_acm_certificate_validation` to request a DNS validated certificate, deploy the required validation records and wait for validation to complete.

Domain validation through email is also supported but should be avoided as it requires a manual step outside of Terraform.

It's recommended to specify `create_before_destroy = true` in a lifecycle block to replace a certificate which is currently in use (eg, by `aws_lb_listener`).

» Example Usage

» Certificate creation

```
resource "aws_acm_certificate" "cert" {
  domain_name       = "example.com"
  validation_method = "DNS"

  tags = {
    Environment = "test"
  }

  lifecycle {
    create_before_destroy = true
  }
}
```

» Importing an existing certificate

```
resource "tls_private_key" "example" {
  algorithm = "RSA"
}

resource "tls_self_signed_cert" "example" {
  key_algorithm   = "RSA"
  private_key_pem = "${tls_private_key.example.private_key_pem}"

  subject {
    common_name  = "example.com"
    organization = "ACME Examples, Inc"
  }

  validity_period_hours = 12

  allowed_uses = [
    "key_encipherment",
    "digital_signature",
    "server_auth",
  ]
}

resource "aws_acm_certificate" "cert" {
  private_key      = "${tls_private_key.example.private_key_pem}"
  certificate_body = "${tls_self_signed_cert.example.cert_pem}"
}
```

» Argument Reference

The following arguments are supported:

- Creating an amazon issued certificate
  - `domain_name` - (Required) A domain name for which the certificate should be issued
  - `subject_alternative_names` - (Optional) A list of domains that should be SANs in the issued certificate
  - `validation_method` - (Required) Which method to use for validation. `DNS` or `EMAIL` are valid, `NONE` can be used for certificates that were imported into ACM and then into Terraform.
  - `options` - (Optional) Configuration block used to set certificate options. Detailed below.
- Importing an existing certificate
  - `private_key` - (Required) The certificate's PEM-formatted private key
  - `certificate_body` - (Required) The certificate's PEM-formatted public key
  - `certificate_chain` - (Optional) The certificate's PEM-formatted chain
- Creating a private CA issued certificate
  - `domain_name` - (Required) A domain name for which the certificate should be issued

  - `certificate_authority_arn` - (Required) ARN of an ACMPCA
  - `subject_alternative_names` - (Optional) A list of domains that should be SANs in the issued certificate
- `tags` - (Optional) A mapping of tags to assign to the resource.

» options Configuration Block

Supported nested arguments for the `options` configuration block:

- `certificate_transparency_logging_preference` - (Optional) Specifies whether certificate details should be added to a certificate transparency log. Valid values are `ENABLED` or `DISABLED`. See https://docs.aws.amazon.com/acm/latest/userguide/acm-concepts.html#concept-transparency for more details.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ARN of the certificate
- `arn` - The ARN of the certificate
- `domain_name` - The domain name for which the certificate is issued
- `domain_validation_options` - A list of attributes to feed into other resources to complete certificate validation. Can have more than one element, e.g. if SANs are defined. Only set if `DNS`-validation was used.
- `validation_emails` - A list of addresses that received a validation E-Mail. Only set if `EMAIL`-validation was used.

Domain validation objects export the following attributes:

- `domain_name` - The domain to be validated
- `resource_record_name` - The name of the DNS record to create to validate the certificate
- `resource_record_type` - The type of DNS record to create
- `resource_record_value` - The value the DNS record needs to have

» Import

Certificates can be imported using their ARN, e.g.

```
$ terraform import aws_acm_certificate.cert arn:aws:acm:eu-central-1:123456789012:certificat
```

» Resource: aws_acm_certificate_validation

This resource represents a successful validation of an ACM certificate in concert with other resources.

Most commonly, this resource is used together with `aws_route53_record` and `aws_acm_certificate` to request a DNS validated certificate, deploy the required validation records and wait for validation to complete.

WARNING: This resource implements a part of the validation workflow. It does not represent a real-world entity in AWS, therefore changing or deleting this resource on its own has no immediate effect.

» Example Usage

» DNS Validation with Route 53

```
resource "aws_acm_certificate" "cert" {
  domain_name       = "example.com"
  validation_method = "DNS"
}

data "aws_route53_zone" "zone" {
  name         = "example.com."
  private_zone = false
}

resource "aws_route53_record" "cert_validation" {
  name    = "${aws_acm_certificate.cert.domain_validation_options.0.resource_record_name}"
  type    = "${aws_acm_certificate.cert.domain_validation_options.0.resource_record_type}"
  zone_id = "${data.aws_route53_zone.zone.id}"
  records = ["${aws_acm_certificate.cert.domain_validation_options.0.resource_record_value}'
  ttl     = 60
}

resource "aws_acm_certificate_validation" "cert" {
  certificate_arn         = "${aws_acm_certificate.cert.arn}"
  validation_record_fqdns = ["${aws_route53_record.cert_validation.fqdn}"]
}

resource "aws_lb_listener" "front_end" {
  # [...]
  certificate_arn = "${aws_acm_certificate_validation.cert.certificate_arn}"
}
```

» Alternative Domains DNS Validation with Route 53

```
resource "aws_acm_certificate" "cert" {
  domain_name               = "example.com"
  subject_alternative_names = ["www.example.com", "example.org"]
  validation_method         = "DNS"
}

data "aws_route53_zone" "zone" {
  name         = "example.com."
  private_zone = false
}

data "aws_route53_zone" "zone_alt" {
  name         = "example.org."
  private_zone = false
}

resource "aws_route53_record" "cert_validation" {
  name    = "${aws_acm_certificate.cert.domain_validation_options.0.resource_record_name}"
  type    = "${aws_acm_certificate.cert.domain_validation_options.0.resource_record_type}"
  zone_id = "${data.aws_route53_zone.zone.id}"
  records = ["${aws_acm_certificate.cert.domain_validation_options.0.resource_record_value}"
  ttl     = 60
}

resource "aws_route53_record" "cert_validation_alt1" {
  name    = "${aws_acm_certificate.cert.domain_validation_options.1.resource_record_name}"
  type    = "${aws_acm_certificate.cert.domain_validation_options.1.resource_record_type}"
  zone_id = "${data.aws_route53_zone.zone.id}"
  records = ["${aws_acm_certificate.cert.domain_validation_options.1.resource_record_value}"
  ttl     = 60
}

resource "aws_route53_record" "cert_validation_alt2" {
  name    = "${aws_acm_certificate.cert.domain_validation_options.2.resource_record_name}"
  type    = "${aws_acm_certificate.cert.domain_validation_options.2.resource_record_type}"
  zone_id = "${data.aws_route53_zone.zone_alt.id}"
  records = ["${aws_acm_certificate.cert.domain_validation_options.2.resource_record_value}"
  ttl     = 60
}

resource "aws_acm_certificate_validation" "cert" {
  certificate_arn = "${aws_acm_certificate.cert.arn}"
  validation_record_fqdns = [
    "${aws_route53_record.cert_validation.fqdn}",
    "${aws_route53_record.cert_validation_alt1.fqdn}",
    "${aws_route53_record.cert_validation_alt2.fqdn}",
  ]
}

resource "aws_lb_listener" "front_end" {
  # [...]
  certificate_arn = "${aws_acm_certificate_validation.cert.certificate_arn}"
}
```

» Email Validation

In this situation, the resource is simply a waiter for manual email approval of ACM certificates.

```
resource "aws_acm_certificate" "cert" {
  domain_name       = "example.com"
  validation_method = "EMAIL"
}

resource "aws_acm_certificate_validation" "cert" {
  certificate_arn = "${aws_acm_certificate.cert.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `certificate_arn` - (Required) The ARN of the certificate that is being validated.
- `validation_record_fqdns` - (Optional) List of FQDNs that implement the validation. Only valid for DNS validation method ACM certificates. If this is set, the resource can implement additional sanity checks and has an explicit dependency on the resource that is implementing the validation

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The time at which the certificate was issued

» Timeouts

acm_certificate_validation provides the following Timeouts configuration options:

- create - (Default 45m) How long to wait for a certificate to be issued.

» Data Source: aws_acmpca_certificate_authority

Get information on a AWS Certificate Manager Private Certificate Authority (ACM PCA Certificate Authority).

» Example Usage

```
data "aws_acmpca_certificate_authority" "example" {
  arn = "arn:aws:acm-pca:us-east-1:123456789012:certificate-authority/12345678-1234-1234-123
}
```

» Argument Reference

The following arguments are supported:

- arn - (Required) Amazon Resource Name (ARN) of the certificate authority.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- id - Amazon Resource Name (ARN) of the certificate authority.
- certificate - Base64-encoded certificate authority (CA) certificate. Only available after the certificate authority certificate has been imported.
- certificate_chain - Base64-encoded certificate chain that includes any intermediate certificates and chains up to root on-premises certificate that you used to sign your private CA certificate. The chain does not include your private CA certificate. Only available after the certificate authority certificate has been imported.
- certificate_signing_request - The base64 PEM-encoded certificate signing request (CSR) for your private CA certificate.
- not_after - Date and time after which the certificate authority is not valid. Only available after the certificate authority certificate has been imported.

- `not_before` - Date and time before which the certificate authority is not valid. Only available after the certificate authority certificate has been imported.
- `revocation_configuration` - Nested attribute containing revocation configuration.
  - `revocation_configuration.0.crl_configuration` - Nested attribute containing configuration of the certificate revocation list (CRL), if any, maintained by the certificate authority.
  - `revocation_configuration.0.crl_configuration.0.custom_cname` - Name inserted into the certificate CRL Distribution Points extension that enables the use of an alias for the CRL distribution point.
  - `revocation_configuration.0.crl_configuration.0.enabled` - Boolean value that specifies whether certificate revocation lists (CRLs) are enabled.
  - `revocation_configuration.0.crl_configuration.0.expiration_in_days` - Number of days until a certificate expires.
  - `revocation_configuration.0.crl_configuration.0.s3_bucket_name` - Name of the S3 bucket that contains the CRL.
- `serial` - Serial number of the certificate authority. Only available after the certificate authority certificate has been imported.
- `status` - Status of the certificate authority.
- `tags` - Specifies a key-value map of user-defined tags that are attached to the certificate authority.
- `type` - The type of the certificate authority.

» Resource: aws_acmpca_certificate_authority

Provides a resource to manage AWS Certificate Manager Private Certificate Authorities (ACM PCA Certificate Authorities).

NOTE: Creating this resource will leave the certificate authority in a `PENDING_CERTIFICATE` status, which means it cannot yet issue certificates. To complete this setup, you must fully sign the certificate authority CSR available in the `certificate_signing_request` attribute and import the signed certificate outside of Terraform. Terraform can support another resource to manage that workflow automatically in the future.

» Example Usage

» Basic

```
resource "aws_acmpca_certificate_authority" "example" {
  certificate_authority_configuration {
    key_algorithm     = "RSA_4096"
    signing_algorithm = "SHA512WITHRSA"

    subject {
      common_name = "example.com"
    }
  }

  permanent_deletion_time_in_days = 7
}
```

» Enable Certificate Revocation List

```
resource "aws_s3_bucket" "example" {
  bucket = "example"
}

data "aws_iam_policy_document" "acmpca_bucket_access" {
  statement {
    actions = [
      "s3:GetBucketAcl",
      "s3:GetBucketLocation",
      "s3:PutObject",
      "s3:PutObjectAcl",
    ]

    resources = [
      "${aws_s3_bucket.example.arn}",
      "${aws_s3_bucket.example.arn}/*",
    ]

    principals {
      identifiers = ["acm-pca.amazonaws.com"]
      type        = "Service"
    }
  }
}

resource "aws_s3_bucket_policy" "example" {
  bucket = "${aws_s3_bucket.example.id}"
  policy = "${data.aws_iam_policy_document.acmpca_bucket_access.json}"
}

resource "aws_acmpca_certificate_authority" "example" {
  certificate_authority_configuration {
    key_algorithm     = "RSA_4096"
    signing_algorithm = "SHA512WITHRSA"

    subject {
      common_name = "example.com"
    }
  }

  revocation_configuration {
    crl_configuration {
      custom_cname       = "crl.example.com"
      enabled            = true
      expiration_in_days = 7
      s3_bucket_name     = "${aws_s3_bucket.example.id}"
    }
  }

  depends_on = ["aws_s3_bucket_policy.example"]
}
```

» Argument Reference

The following arguments are supported:

- `certificate_authority_configuration` - (Required) Nested argument containing algorithms and certificate subject information. Defined below.
- `enabled` - (Optional) Whether the certificate authority is enabled or disabled. Defaults to `true`.
- `revocation_configuration` - (Optional) Nested argument containing revocation configuration. Defined below.
- `tags` - (Optional) Specifies a key-value map of user-defined tags that are attached to the certificate authority.
- `type` - (Optional) The type of the certificate authority. Defaults to `SUBORDINATE`. Valid values: `ROOT` and `SUBORDINATE`.
- `permanent_deletion_time_in_days` - (Optional) The number of days to make a CA restorable after it has been deleted, must be between 7 to 30 days, with default to 30 days.

» certificate_authority_configuration

- `key_algorithm` - (Required) Type of the public key algorithm and size, in bits, of the key pair that your key pair creates when it issues a certificate. Valid values can be found in the ACM PCA Documentation.

- `signing_algorithm` - (Required) Name of the algorithm your private CA uses to sign certificate requests. Valid values can be found in the ACM PCA Documentation.
- `subject` - (Required) Nested argument that contains X.500 distinguished name information. At least one nested attribute must be specified.

» subject

Contains information about the certificate subject. Identifies the entity that owns or controls the public key in the certificate. The entity can be a user, computer, device, or service.

- `common_name` - (Optional) Fully qualified domain name (FQDN) associated with the certificate subject.
- `country` - (Optional) Two digit code that specifies the country in which the certificate subject located.
- `distinguished_name_qualifier` - (Optional) Disambiguating information for the certificate subject.
- `generation_qualifier` - (Optional) Typically a qualifier appended to the name of an individual. Examples include Jr. for junior, Sr. for senior, and III for third.
- `given_name` - (Optional) First name.
- `initials` - (Optional) Concatenation that typically contains the first letter of the `given_name`, the first letter of the middle name if one exists, and the first letter of the `surname`.
- `locality` - (Optional) The locality (such as a city or town) in which the certificate subject is located.
- `organization` - (Optional) Legal name of the organization with which the certificate subject is affiliated.
- `organizational_unit` - (Optional) A subdivision or unit of the organization (such as sales or finance) with which the certificate subject is affiliated.
- `pseudonym` - (Optional) Typically a shortened version of a longer `given_name`. For example, Jonathan is often shortened to John. Elizabeth is often shortened to Beth, Liz, or Eliza.
- `state` - (Optional) State in which the subject of the certificate is located.
- `surname` - (Optional) Family name. In the US and the UK for example, the surname of an individual is ordered last. In Asian cultures the surname is typically ordered first.
- `title` - (Optional) A title such as Mr. or Ms. which is pre-pended to the name to refer formally to the certificate subject.

» revocation_configuration

- `crl_configuration` - (Optional) Nested argument containing configuration of the certificate revocation list (CRL), if any, maintained by the certificate authority. Defined below.

» crl_configuration

- `custom_cname` - (Optional) Name inserted into the certificate CRL Distribution Points extension that enables the use of an alias for the CRL distribution point. Use this value if you don't want the name of your S3 bucket to be public.
- `enabled` - (Optional) Boolean value that specifies whether certificate revocation lists (CRLs) are enabled. Defaults to `false`.
- `expiration_in_days` - (Required) Number of days until a certificate expires. Must be between 1 and 5000.
- `s3_bucket_name` - (Optional) Name of the S3 bucket that contains the CRL. If you do not provide a value for the `custom_cname` argument, the name of your S3 bucket is placed into the CRL Distribution Points extension of the issued certificate. You must specify a bucket policy that allows ACM PCA to write the CRL to your bucket.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the certificate authority.
- `arn` - Amazon Resource Name (ARN) of the certificate authority.
- `certificate` - Base64-encoded certificate authority (CA) certificate. Only available after the certificate authority certificate has been imported.
- `certificate_chain` - Base64-encoded certificate chain that includes any intermediate certificates and chains up to root on-premises certificate that you used to sign your private CA certificate. The chain does not include your private CA certificate. Only available after the certificate authority certificate has been imported.
- `certificate_signing_request` - The base64 PEM-encoded certificate signing request (CSR) for your private CA certificate.
- `not_after` - Date and time after which the certificate authority is not valid. Only available after the certificate authority certificate has been imported.
- `not_before` - Date and time before which the certificate authority is not valid. Only available after the certificate authority certificate has been imported.
- `serial` - Serial number of the certificate authority. Only available after the certificate authority certificate has been imported.
- `status` - Status of the certificate authority.

» Timeouts

`aws_acmpca_certificate_authority` provides the following Timeouts configuration options:

- `create` - (Default `1m`) How long to wait for a certificate authority to be created.

» Import

`aws_acmpca_certificate_authority` can be imported by using the certificate authority Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_acmpca_certificate_authority.example arn:aws:acm-pca:us-east-1:123456
```

» Data Source: aws_api_gateway_api_key

Use this data source to get the name and value of a pre-existing API Key, for example to supply credentials for a dependency microservice.

» Example Usage

```
data "aws_api_gateway_api_key" "my_api_key" {
  id = "ru3mpjgse6"
}
```

» Argument Reference

- `id` - (Required) The ID of the API Key to look up.

» Attributes Reference

- `id` - Set to the ID of the API Key.
- `name` - Set to the name of the API Key.
- `value` - Set to the value of the API Key.

» Data Source: aws_api_gateway_resource

Use this data source to get the id of a Resource in API Gateway. To fetch the Resource, you must provide the REST API id as well as the full path.

» Example Usage

```
data "aws_api_gateway_rest_api" "my_rest_api" {
  name = "my-rest-api"
}

data "aws_api_gateway_resource" "my_resource" {
  rest_api_id = "${data.aws_api_gateway_rest_api.my_rest_api.id}"
  path        = "/endpoint/path"
}
```

» Argument Reference

- `rest_api_id` - (Required) The REST API id that owns the resource. If no REST API is found, an error will be returned.
- `path` - (Required) The full path of the resource. If no path is found, an error will be returned.

» Attributes Reference

- `id` - Set to the ID of the found Resource.
- `parent_id` - Set to the ID of the parent Resource.
- `path_part` - Set to the path relative to the parent Resource.

» Data Source: aws_api_gateway_rest_api

Use this data source to get the id and root_resource_id of a REST API in API Gateway. To fetch the REST API you must provide a name to match against. As there is no unique name constraint on REST APIs this data source will error if there is more than one match.

» Example Usage

```
data "aws_api_gateway_rest_api" "my_rest_api" {
  name = "my-rest-api"
}
```

» Argument Reference

- `name` - (Required) The name of the REST API to look up. If no REST API is found with this name, an error will be returned. If multiple REST

APIs are found with this name, an error will be returned.

» Attributes Reference

- `id` - Set to the ID of the found REST API.
- `root_resource_id` - Set to the ID of the API Gateway Resource on the found REST API where the route matches '/'.

» Data Source: aws_api_gateway_vpc_link

Use this data source to get the id of a VPC Link in API Gateway. To fetch the VPC Link you must provide a name to match against. As there is no unique name constraint on API Gateway VPC Links this data source will error if there is more than one match.

» Example Usage

```
data "aws_api_gateway_vpc_link" "my_api_gateway_vpc_link" {
  name = "my-vpc-link"
}
```

» Argument Reference

- `name` - (Required) The name of the API Gateway VPC Link to look up. If no API Gateway VPC Link is found with this name, an error will be returned. If multiple API Gateway VPC Links are found with this name, an error will be returned.

» Attributes Reference

- `id` - Set to the ID of the found API Gateway VPC Link.

» Resource: aws_api_gateway_account

Provides a settings of an API Gateway Account. Settings is applied region-wide per `provider` block.

Note: As there is no API method for deleting account settings or resetting it to defaults, destroying this resource will keep your account settings intact

» Example Usage

```
resource "aws_api_gateway_account" "demo" {
  cloudwatch_role_arn = "${aws_iam_role.cloudwatch.arn}"
}

resource "aws_iam_role" "cloudwatch" {
  name = "api_gateway_cloudwatch_global"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "apigateway.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "cloudwatch" {
  name = "default"
  role = "${aws_iam_role.cloudwatch.id}"

  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents",
                "logs:GetLogEvents",
                "logs:FilterLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
EOF
}
```

» Argument Reference

The following argument is supported:

- `cloudwatch_role_arn` - (Optional) The ARN of an IAM role for CloudWatch (to allow logging & monitoring). See more in AWS Docs. Logging & monitoring can be enabled/disabled and otherwise tuned on the API Gateway Stage level.

» Attribute Reference

The following attribute is exported:

- `throttle_settings` - Account-Level throttle settings. See exported fields below.

`throttle_settings` block exports the following:

- `burst_limit` - The absolute maximum number of times API Gateway allows the API to be called per second (RPS).
- `rate_limit` - The number of times API Gateway allows the API to be called per second on average (RPS).

» Import

API Gateway Accounts can be imported using the word `api-gateway-account`, e.g.

```
$ terraform import aws_api_gateway_account.demo api-gateway-account
```

» Resource: aws_api_gateway_api_key

Provides an API Gateway API Key.

NOTE: Since the API Gateway usage plans feature was launched on August 11, 2016, usage plans are now **required** to associate an API key with an API stage.

» Example Usage

```
resource "aws_api_gateway_api_key" "MyDemoApiKey" {
  name = "demo"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the API key
- `description` - (Optional) The API key description. Defaults to "Managed by Terraform".
- `enabled` - (Optional) Specifies whether the API key can be used by callers. Defaults to `true`.
- `value` - (Optional) The value of the API key. If not specified, it will be automatically generated by AWS on creation.
- `tags` - (Optional) Key-value mapping of resource tags

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the API key
- `created_date` - The creation date of the API key
- `last_updated_date` - The last update date of the API key
- `value` - The value of the API key
- `arn` - Amazon Resource Name (ARN)

» Import

API Gateway Keys can be imported using the `id`, e.g.

```
$ terraform import aws_api_gateway_api_key.my_demo_key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq
```

» Resource: aws_api_gateway_authorizer

Provides an API Gateway Authorizer.

» Example Usage

```
resource "aws_api_gateway_authorizer" "demo" {
  name                   = "demo"
  rest_api_id            = "${aws_api_gateway_rest_api.demo.id}"
  authorizer_uri         = "${aws_lambda_function.authorizer.invoke_arn}"
  authorizer_credentials = "${aws_iam_role.invocation_role.arn}"
}

resource "aws_api_gateway_rest_api" "demo" {
  name = "auth-demo"
}

resource "aws_iam_role" "invocation_role" {
  name = "api_gateway_auth_invocation"
  path = "/"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "apigateway.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "invocation_policy" {
  name = "default"
  role = "${aws_iam_role.invocation_role.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "lambda:InvokeFunction",
      "Effect": "Allow",
      "Resource": "${aws_lambda_function.authorizer.arn}"
    }
  ]
}
EOF
}

resource "aws_iam_role" "lambda" {
  name = "demo-lambda"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_lambda_function" "authorizer" {
  filename      = "lambda-function.zip"
  function_name = "api_gateway_authorizer"
  role          = "${aws_iam_role.lambda.arn}"
  handler       = "exports.example"

  # The filebase64sha256() function is available in Terraform 0.11.12 and later
  # For Terraform 0.11.11 and earlier, use the base64sha256() function and the file() functi
  # source_code_hash = "${base64sha256(file("lambda-function.zip"))}"
  source_code_hash = "${filebase64sha256("lambda-function.zip")}"
}
```

» Argument Reference

The following arguments are supported:

- authorizer_uri - (Optional, required for type TOKEN/REQUEST) The authorizer's Uniform Resource Identifier (URI). This must be a well-formed

Lambda function URI in the form of `arn:aws:apigateway:{region}:lambda:path/{service_api}`, e.g. `arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-wes`
- `name` - (Required) The name of the authorizer
- `rest_api_id` - (Required) The ID of the associated REST API
- `identity_source` - (Optional) The source of the identity in an incoming request. Defaults to `method.request.header.Authorization`. For `REQUEST` type, this may be a comma-separated list of values, including headers, query string parameters and stage variables - e.g. `"method.request.header.SomeHeaderName,method.request.querystring.SomeQueryStringName,st`
- `type` - (Optional) The type of the authorizer. Possible values are `TOKEN` for a Lambda function using a single authorization token submitted in a custom header, `REQUEST` for a Lambda function using incoming request parameters, or `COGNITO_USER_POOLS` for using an Amazon Cognito user pool. Defaults to `TOKEN`.
- `authorizer_credentials` - (Optional) The credentials required for the authorizer. To specify an IAM Role for API Gateway to assume, use the IAM Role ARN.
- `authorizer_result_ttl_in_seconds` - (Optional) The TTL of cached authorizer results in seconds. Defaults to `300`.
- `identity_validation_expression` - (Optional) A validation expression for the incoming identity. For `TOKEN` type, this value should be a regular expression. The incoming token from the client is matched against this expression, and will proceed if the token matches. If the token doesn't match, the client receives a 401 Unauthorized response.
- `provider_arns` - (Optional, required for type `COGNITO_USER_POOLS`) A list of the Amazon Cognito user pool ARNs. Each element is of this format: `arn:aws:cognito-idp:{region}:{account_id}:userpool/{user_pool_id}`.

» Import

AWS API Gateway Authorizer can be imported using the `REST-API-ID/AUTHORIZER-ID`, e.g.

```
$ terraform import aws_api_gateway_authorizer.authorizer 12345abcde/example
```

» Resource: aws_api_gateway_base_path_mapping

Connects a custom domain name registered via `aws_api_gateway_domain_name` with a deployed API so that its methods can be called via the custom domain name.

» Example Usage

```
resource "aws_api_gateway_deployment" "example" {
  # See aws_api_gateway_rest_api docs for how to create this
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  stage_name  = "live"
}

resource "aws_api_gateway_domain_name" "example" {
  domain_name = "example.com"

  certificate_name        = "example-api"
  certificate_body        = "${file("${path.module}/example.com/example.crt")}"
  certificate_chain       = "${file("${path.module}/example.com/ca.crt")}"
  certificate_private_key = "${file("${path.module}/example.com/example.key")}"
}

resource "aws_api_gateway_base_path_mapping" "test" {
  api_id      = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  stage_name  = "${aws_api_gateway_deployment.example.stage_name}"
  domain_name = "${aws_api_gateway_domain_name.example.domain_name}"
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Required) The already-registered domain name to connect the API to.
- `api_id` - (Required) The id of the API to connect.
- `stage_name` - (Optional) The name of a specific deployment stage to expose at the given path. If omitted, callers may select any stage by including its name as a path element after the base path.
- `base_path` - (Optional) Path segment that must be prepended to the path when accessing the API via this mapping. If omitted, the API is exposed at the root of the given domain.

» Import

`aws_api_gateway_base_path_mapping` can be imported by using the domain name and base path, e.g.

For empty `base_path` (e.g. root path (`/`)):

```
$ terraform import aws_api_gateway_base_path_mapping.example example.com/
```

Otherwise:

```
$ terraform import aws_api_gateway_base_path_mapping.example example.com/base-path
```

» Resource: aws_api_gateway_client_certificate

Provides an API Gateway Client Certificate.

» Example Usage

```
resource "aws_api_gateway_client_certificate" "demo" {
  description = "My client certificate"
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Optional) The description of the client certificate.
- `tags` - (Optional) Key-value mapping of resource tags

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The identifier of the client certificate.
- `created_date` - The date when the client certificate was created.
- `expiration_date` - The date when the client certificate will expire.
- `pem_encoded_certificate` - The PEM-encoded public key of the client certificate.
- `arn` - Amazon Resource Name (ARN)

» Import

API Gateway Client Certificates can be imported using the id, e.g.

```
$ terraform import aws_api_gateway_client_certificate.demo ab1cqe
```

» Resource: aws_api_gateway_deployment

Provides an API Gateway Deployment.

Note: Depends on having `aws_api_gateway_integration` inside your rest api (which in turn depends on `aws_api_gateway_method`). To avoid race conditions you might need to add an explicit `depends_on = ["aws_api_gateway_integration.name"]`.

» Example Usage

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_resource" "MyDemoResource" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  parent_id   = "${aws_api_gateway_rest_api.MyDemoAPI.root_resource_id}"
  path_part   = "test"
}

resource "aws_api_gateway_method" "MyDemoMethod" {
  rest_api_id   = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id   = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method   = "GET"
  authorization = "NONE"
}

resource "aws_api_gateway_integration" "MyDemoIntegration" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method = "${aws_api_gateway_method.MyDemoMethod.http_method}"
  type        = "MOCK"
}

resource "aws_api_gateway_deployment" "MyDemoDeployment" {
  depends_on = ["aws_api_gateway_integration.MyDemoIntegration"]

  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  stage_name  = "test"

  variables = {
    "answer" = "42"
  }
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API
- `stage_name` - (Optional) The name of the stage. If the specified stage already exists, it will be updated to point to the new deployment. If the stage does not exist, a new one will be created and point to this deployment.
- `description` - (Optional) The description of the deployment
- `stage_description` - (Optional) The description of the stage
- `variables` - (Optional) A map that defines variables for the stage

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the deployment
- `invoke_url` - The URL to invoke the API pointing to the stage, e.g. `https://z4675bid1j.execute-api.eu-west-2.amazonaws.com/prod`
- `execution_arn` - The execution ARN to be used in `lambda_permission`'s `source_arn` when allowing API Gateway to invoke a Lambda function, e.g. `arn:aws:execute-api:eu-west-2:123456789012:z4675bid1j/prod`
- `created_date` - The creation date of the deployment

» Resource: aws_api_gateway_documentation_part

Provides a settings of an API Gateway Documentation Part.

» Example Usage

```
resource "aws_api_gateway_documentation_part" "example" {
  location {
    type   = "METHOD"
    method = "GET"
    path   = "/example"
  }

  properties  = "{\"description\":\"Example description\"}"
  rest_api_id = "${aws_api_gateway_rest_api.example.id}"
}

resource "aws_api_gateway_rest_api" "example" {
  name = "example_api"
}
```

» Argument Reference

The following argument is supported:

- `location` - (Required) The location of the targeted API entity of the to-be-created documentation part. See below.
- `properties` - (Required) A content map of API-specific key-value pairs describing the targeted API entity. The map must be encoded as a JSON string, e.g., "{ \"description\": \"The API does ...\" }". Only Swagger-compliant key-value pairs can be exported and, hence, published.
- `rest_api_id` - (Required) The ID of the associated Rest API

» Nested fields

» `location`

See supported entity types for each field in the official docs.

- `method` - (Optional) The HTTP verb of a method. The default value is `*` for any method.
- `name` - (Optional) The name of the targeted API entity.
- `path` - (Optional) The URL path of the target. The default value is `/` for the root resource.
- `status_code` - (Optional) The HTTP status code of a response. The default value is `*` for any status code.
- `type` - (Required) The type of API entity to which the documentation content applies. e.g. `API`, `METHOD` or `REQUEST_BODY`

» Attribute Reference

The following attribute is exported in addition to the arguments listed above:

- `id` - The unique ID of the Documentation Part

» Import

API Gateway documentation_parts can be imported using `REST-API-ID/DOC-PART-ID`, e.g.

```
$ terraform import aws_api_gateway_documentation_part.example 5i4e1ko720/3oyy3t
```

» Resource: aws_api_gateway_documentation_version

Provides a resource to manage an API Gateway Documentation Version.

» Example Usage

```
resource "aws_api_gateway_documentation_version" "example" {
  version     = "example_version"
  rest_api_id = "${aws_api_gateway_rest_api.example.id}"
  description = "Example description"
  depends_on  = ["aws_api_gateway_documentation_part.example"]
}

resource "aws_api_gateway_rest_api" "example" {
  name = "example_api"
}

resource "aws_api_gateway_documentation_part" "example" {
  location {
    type = "API"
  }

  properties  = "{\"description\":\"Example\"}"
  rest_api_id = "${aws_api_gateway_rest_api.example.id}"
}
```

» Argument Reference

The following argument is supported:

- `version` - (Required) The version identifier of the API documentation snapshot.
- `rest_api_id` - (Required) The ID of the associated Rest API
- `description` - (Optional) The description of the API documentation version.

» Attribute Reference

The arguments listed above are all exported as attributes.

» Import

API Gateway documentation versions can be imported using `REST-API-ID/VERSION`, e.g.

```
$ terraform import aws_api_gateway_documentation_version.example 5i4e1ko720/example-version
```

» Resource: aws_api_gateway_domain_name

Registers a custom domain name for use with AWS API Gateway. Additional information about this functionality can be found in the API Gateway Developer Guide.

This resource just establishes ownership of and the TLS settings for a particular domain name. An API can be attached to a particular path under the registered domain name using the `aws_api_gateway_base_path_mapping` resource.

API Gateway domains can be defined as either 'edge-optimized' or 'regional'. In an edge-optimized configuration, API Gateway internally creates and manages a CloudFront distribution to route requests on the given hostname. In addition to this resource it's necessary to create a DNS record corresponding to the given domain name which is an alias (either Route53 alias or traditional CNAME) to the Cloudfront domain name exported in the `cloudfront_domain_name` attribute.

In a regional configuration, API Gateway does not create a CloudFront distribution to route requests to the API, though a distribution can be created if needed. In either case, it is necessary to create a DNS record corresponding to the given domain name which is an alias (either Route53 alias or traditional CNAME) to the regional domain name exported in the `regional_domain_name` attribute.

Note: API Gateway requires the use of AWS Certificate Manager (ACM) certificates instead of Identity and Access Management (IAM) certificates in regions that support ACM. Regions that support ACM can be found in the Regions and Endpoints Documentation. To import an existing private key and certificate into ACM or request an ACM certificate, see the `aws_acm_certificate` resource.

Note: The `aws_api_gateway_domain_name` resource expects dependency on the `aws_acm_certificate_validation` as only verified certificates can be used. This can be made either explicitly by adding the `depends_on = [aws_acm_certificate_validation.cert]` attribute. Or implicitly by referring certificate ARN from the validation resource where it will be available after the resource creation: `regional_certificate_arn = aws_acm_certificate_validation.cert.certificate_arn`.

Note: All arguments including the private key will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

» Edge Optimized (ACM Certificate)

```
resource "aws_api_gateway_domain_name" "example" {
  certificate_arn = "${aws_acm_certificate_validation.example.certificate_arn}"
  domain_name     = "api.example.com"
}

# Example DNS record using Route53.
# Route53 is not specifically required; any DNS host can be used.
resource "aws_route53_record" "example" {
  name    = "${aws_api_gateway_domain_name.example.domain_name}"
  type    = "A"
  zone_id = "${aws_route53_zone.example.id}"

  alias {
    evaluate_target_health = true
    name                   = "${aws_api_gateway_domain_name.example.cloudfront_domain_name}"
    zone_id                = "${aws_api_gateway_domain_name.example.cloudfront_zone_id}"
  }
}
```

» Edge Optimized (IAM Certificate)

```
resource "aws_api_gateway_domain_name" "example" {
  domain_name = "api.example.com"

  certificate_name        = "example-api"
  certificate_body        = "${file("${path.module}/example.com/example.crt")}"
  certificate_chain       = "${file("${path.module}/example.com/ca.crt")}"
  certificate_private_key = "${file("${path.module}/example.com/example.key")}"
}

# Example DNS record using Route53.
# Route53 is not specifically required; any DNS host can be used.
resource "aws_route53_record" "example" {
  zone_id = "${aws_route53_zone.example.id}" # See aws_route53_zone for how to create this

  name = "${aws_api_gateway_domain_name.example.domain_name}"
  type = "A"

  alias {
    name                   = "${aws_api_gateway_domain_name.example.cloudfront_domain_name}"
    zone_id                = "${aws_api_gateway_domain_name.example.cloudfront_zone_id}"
    evaluate_target_health = true
  }
}
```

» Regional (ACM Certificate)

```
resource "aws_api_gateway_domain_name" "example" {
  domain_name              = "api.example.com"
  regional_certificate_arn = "${aws_acm_certificate_validation.example.certificate_arn}"

  endpoint_configuration {
    types = ["REGIONAL"]
  }
}

# Example DNS record using Route53.
# Route53 is not specifically required; any DNS host can be used.
resource "aws_route53_record" "example" {
  name    = "${aws_api_gateway_domain_name.example.domain_name}"
  type    = "A"
  zone_id = "${aws_route53_zone.example.id}"

  alias {
    evaluate_target_health = true
    name                   = "${aws_api_gateway_domain_name.example.regional_domain_name}"
    zone_id                = "${aws_api_gateway_domain_name.example.regional_zone_id}"
  }
}
```

» Regional (IAM Certificate)

```
resource "aws_api_gateway_domain_name" "example" {
  certificate_body          = "${file("${path.module}/example.com/example.crt")}"
  certificate_chain         = "${file("${path.module}/example.com/ca.crt")}"
  certificate_private_key   = "${file("${path.module}/example.com/example.key")}"
  domain_name               = "api.example.com"
  regional_certificate_name = "example-api"

  endpoint_configuration {
    types = ["REGIONAL"]
  }
}

# Example DNS record using Route53.
# Route53 is not specifically required; any DNS host can be used.
resource "aws_route53_record" "example" {
  name    = "${aws_api_gateway_domain_name.example.domain_name}"
  type    = "A"
  zone_id = "${aws_route53_zone.example.id}"

  alias {
    evaluate_target_health = true
    name                   = "${aws_api_gateway_domain_name.example.regional_domain_name}"
    zone_id                = "${aws_api_gateway_domain_name.example.regional_zone_id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Required) The fully-qualified domain name to register
- `endpoint_configuration` - (Optional) Configuration block defining API endpoint information including type. Defined below.
- `security_policy` - (Optional) The Transport Layer Security (TLS) version + cipher suite for this DomainName. The valid values are `TLS_1_0` and `TLS_1_2`. Must be configured to perform drift detection.
- `tags` - (Optional) Key-value mapping of resource tags

When referencing an AWS-managed certificate, the following arguments are supported:

- `certificate_arn` - (Optional) The ARN for an AWS-managed certificate. AWS Certificate Manager is the only supported source. Used when an edge-optimized domain name is desired. Conflicts with `certificate_name`, `certificate_body`, `certificate_chain`, `certificate_private_key`, `regional_certificate_arn`, and `regional_certificate_name`.
- `regional_certificate_arn` - (Optional) The ARN for an AWS-managed certificate. AWS Certificate Manager is the only supported source. Used when a regional domain name is desired. Conflicts with `certificate_arn`, `certificate_name`, `certificate_body`, `certificate_chain`, and `certificate_private_key`.

When uploading a certificate, the following arguments are supported:

- `certificate_name` - (Optional) The unique name to use when registering this certificate as an IAM server certificate. Conflicts with `certificate_arn`, `regional_certificate_arn`, and `regional_certificate_name`. Required if `certificate_arn` is not set.

- `certificate_body` - (Optional) The certificate issued for the domain name being registered, in PEM format. Only valid for `EDGE` endpoint configuration type. Conflicts with `certificate_arn`, `regional_certificate_arn`, and `regional_certificate_name`.
- `certificate_chain` - (Optional) The certificate for the CA that issued the certificate, along with any intermediate CA certificates required to create an unbroken chain to a certificate trusted by the intended API clients. Only valid for `EDGE` endpoint configuration type. Conflicts with `certificate_arn`, `regional_certificate_arn`, and `regional_certificate_name`.
- `certificate_private_key` - (Optional) The private key associated with the domain certificate given in `certificate_body`. Only valid for `EDGE` endpoint configuration type. Conflicts with `certificate_arn`, `regional_certificate_arn`, and `regional_certificate_name`.
- `regional_certificate_name` - (Optional) The user-friendly name of the certificate that will be used by regional endpoint for this domain name. Conflicts with `certificate_arn`, `certificate_name`, `certificate_body`, `certificate_chain`, and `certificate_private_key`.

» endpoint_configuration

- `types` - (Required) A list of endpoint types. This resource currently only supports managing a single value. Valid values: `EDGE` or `REGIONAL`. If unspecified, defaults to `EDGE`. Must be declared as `REGIONAL` in non-Commercial partitions. Refer to the documentation for more information on the difference between edge-optimized and regional APIs.

» Attributes Reference

In addition to the arguments, the following attributes are exported:

- `id` - The internal id assigned to this domain name by API Gateway.
- `certificate_upload_date` - The upload date associated with the domain certificate.
- `cloudfront_domain_name` - The hostname created by Cloudfront to represent the distribution that implements this domain name mapping.
- `cloudfront_zone_id` - For convenience, the hosted zone ID (`Z2FDTNDATAQYW2`) that can be used to create a Route53 alias record for the distribution.
- `regional_domain_name` - The hostname for the custom domain's regional endpoint.
- `regional_zone_id` - The hosted zone ID that can be used to create a Route53 alias record for the regional endpoint.
- `arn` - Amazon Resource Name (ARN)

» Import

API Gateway domain names can be imported using their `name`, e.g.

```
$ terraform import aws_api_gateway_domain_name.example dev.example.com
```

» Resource: aws_api_gateway_gateway_response

Provides an API Gateway Gateway Response for a REST API Gateway.

» Example Usage

```
resource "aws_api_gateway_rest_api" "main" {
  name = "MyDemoAPI"
}

resource "aws_api_gateway_gateway_response" "test" {
  rest_api_id   = "${aws_api_gateway_rest_api.main.id}"
  status_code   = "401"
  response_type = "UNAUTHORIZED"

  response_templates = {
    "application/json" = "{'message':$context.error.messageString}"
  }

  response_parameters = {
    "gatewayresponse.header.Authorization" = "'Basic'"
  }
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The string identifier of the associated REST API.
- `response_type` - (Required) The response type of the associated GatewayResponse.
- `status_code` - (Optional) The HTTP status code of the Gateway Response.
- `response_templates` - (Optional) A map specifying the templates used to transform the response body.

- `response_parameters` - (Optional) A map specifying the parameters (paths, query strings and headers) of the Gateway Response.

» Import

`aws_api_gateway_gateway_response` can be imported using `REST-API-ID/RESPONSE-TYPE`, e.g.

```
$ terraform import aws_api_gateway_gateway_response.example 12345abcde/UNAUTHORIZED
```

» Resource: aws_api_gateway_integration

Provides an HTTP Method Integration for an API Gateway Integration.

» Example Usage

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_resource" "MyDemoResource" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  parent_id   = "${aws_api_gateway_rest_api.MyDemoAPI.root_resource_id}"
  path_part   = "mydemoresource"
}

resource "aws_api_gateway_method" "MyDemoMethod" {
  rest_api_id   = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id   = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method   = "GET"
  authorization = "NONE"
}

resource "aws_api_gateway_integration" "MyDemoIntegration" {
  rest_api_id          = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id          = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method          = "${aws_api_gateway_method.MyDemoMethod.http_method}"
  type                 = "MOCK"
  cache_key_parameters = ["method.request.path.param"]
  cache_namespace      = "foobar"
  timeout_milliseconds = 29000
  request_parameters = {
    "integration.request.header.X-Authorization" = "'static'"
  }

  # Transforms the incoming XML request to JSON
  request_templates = {
    "application/xml" = <<EOF
{
   "body" : $input.json('$')
}
EOF
  }
}
```

» Lambda integration

```
# Variables
variable "myregion" {}

variable "accountId" {}

# API Gateway
resource "aws_api_gateway_rest_api" "api" {
  name = "myapi"
}

resource "aws_api_gateway_resource" "resource" {
  path_part   = "resource"
  parent_id   = "${aws_api_gateway_rest_api.api.root_resource_id}"
  rest_api_id = "${aws_api_gateway_rest_api.api.id}"
}

resource "aws_api_gateway_method" "method" {
  rest_api_id   = "${aws_api_gateway_rest_api.api.id}"
  resource_id   = "${aws_api_gateway_resource.resource.id}"
  http_method   = "GET"
  authorization = "NONE"
}

resource "aws_api_gateway_integration" "integration" {
  rest_api_id             = "${aws_api_gateway_rest_api.api.id}"
  resource_id             = "${aws_api_gateway_resource.resource.id}"
  http_method             = "${aws_api_gateway_method.method.http_method}"
  integration_http_method = "POST"
  type                    = "AWS_PROXY"
  uri                     = "${aws_lambda_function.lambda.invoke_arn}"
}

# Lambda
resource "aws_lambda_permission" "apigw_lambda" {
  statement_id  = "AllowExecutionFromAPIGateway"
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.lambda.function_name}"
  principal     = "apigateway.amazonaws.com"

  # More: http://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-control-ac
  source_arn = "arn:aws:execute-api:${var.myregion}:${var.accountId}:${aws_api_gateway_rest_
}

resource "aws_lambda_function" "lambda" {
  filename      = "lambda.zip"
  function_name = "mylambda"
  role          = "${aws_iam_role.role.arn}"
  handler       = "lambda.lambda_handler"
  runtime       = "python2.7"

  # The filebase64sha256() function is available in Terraform 0.11.12 and later
  # For Terraform 0.11.11 and earlier, use the base64sha256() function and the file() functi
  # source_code_hash = "${base64sha256(file("lambda.zip"))}"
  source_code_hash = "${filebase64sha256("lambda.zip")}"
}

# IAM
resource "aws_iam_role" "role" {
  name = "myrole"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
POLICY
}
```

» VPC Link

```
variable "name" {}
variable "subnet_id" {}

resource "aws_lb" "test" {
  name               = "${var.name}"
  internal           = true
  load_balancer_type = "network"
  subnets            = ["${var.subnet_id}"]
}

resource "aws_api_gateway_vpc_link" "test" {
  name        = "${var.name}"
  target_arns = ["${aws_lb.test.arn}"]
}

resource "aws_api_gateway_rest_api" "test" {
  name = "${var.name}"
}

resource "aws_api_gateway_resource" "test" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  parent_id   = "${aws_api_gateway_rest_api.test.root_resource_id}"
  path_part   = "test"
}

resource "aws_api_gateway_method" "test" {
  rest_api_id   = "${aws_api_gateway_rest_api.test.id}"
  resource_id   = "${aws_api_gateway_resource.test.id}"
  http_method   = "GET"
  authorization = "NONE"

  request_models = {
    "application/json" = "Error"
  }
}

resource "aws_api_gateway_integration" "test" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  resource_id = "${aws_api_gateway_resource.test.id}"
  http_method = "${aws_api_gateway_method.test.http_method}"
  request_templates = {
    "application/json" = ""
    "application/xml"  = "#set($inputRoot = $input.path('$'))\n{ }"
  }

  request_parameters = {
    "integration.request.header.X-Authorization" = "'static'"
    "integration.request.header.X-Foo"           = "'Bar'"
  }

  type                    = "HTTP"
  uri                     = "https://www.google.de"
  integration_http_method = "GET"
  passthrough_behavior    = "WHEN_NO_MATCH"
  content_handling        = "CONVERT_TO_TEXT"

  connection_type = "VPC_LINK"
  connection_id   = "${aws_api_gateway_vpc_link.test.id}"
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API.
- `resource_id` - (Required) The API resource ID.
- `http_method` - (Required) The HTTP method (`GET`, `POST`, `PUT`, `DELETE`, `HEAD`, `OPTION`, `ANY`) when calling the associated resource.
- `integration_http_method` - (Optional) The integration HTTP method (`GET`, `POST`, `PUT`, `DELETE`, `HEAD`, `OPTIONs`, `ANY`, `PATCH`) specifying how API Gateway will interact with the back end. **Required** if `type` is `AWS`, `AWS_PROXY`, `HTTP` or `HTTP_PROXY`. Not all methods are compatible with all `AWS` integrations. e.g. Lambda function can only be invoked via `POST`.
- `type` - (Required) The integration input's type. Valid values are `HTTP` (for HTTP backends), `MOCK` (not calling any real backend), `AWS` (for AWS services), `AWS_PROXY` (for Lambda proxy integration) and `HTTP_PROXY` (for HTTP proxy integration). An `HTTP` or `HTTP_PROXY` integration with a `connection_type` of `VPC_LINK` is referred to as a private integration and uses a VpcLink to connect API Gateway to a network load balancer of a VPC.
- `connection_type` - (Optional) The integration input's connectionType. Valid values are `INTERNET` (default for connections through the public routable internet), and `VPC_LINK` (for private connections between API Gateway and a network load balancer in a VPC).
- `connection_id` - (Optional) The id of the VpcLink used for the integra-

tion. **Required** if `connection_type` is `VPC_LINK`

- `uri` - (Optional) The input's URI. **Required** if `type` is `AWS`, `AWS_PROXY`, `HTTP` or `HTTP_PROXY`. For HTTP integrations, the URI must be a fully formed, encoded HTTP(S) URL according to the RFC-3986 specification . For AWS integrations, the URI should be of the form `arn:aws:apigateway:{region}:{subdomain.service|service}:{path|action}/{service_api}`. `region`, `subdomain` and `service` are used to determine the right endpoint. e.g. `arn:aws:apigateway:eu-west-1:lambda:path/2015-03-31/functions/arn:aws:lambda:eu-wes`
- `credentials` - (Optional) The credentials required for the integration. For `AWS` integrations, 2 options are available. To specify an IAM Role for Amazon API Gateway to assume, use the role's ARN. To require that the caller's identity be passed through from the request, specify the string `arn:aws:iam::\*:user/\*`.
- `request_templates` - (Optional) A map of the integration's request templates.
- `request_parameters` - (Optional) A map of request query string parameters and headers that should be passed to the backend responder. For example: `request_parameters = { "integration.request.header.X-Some-Other-Header" = "method.request.header.X-Some-Header" }`
- `passthrough_behavior` - (Optional) The integration passthrough behavior (`WHEN_NO_MATCH`, `WHEN_NO_TEMPLATES`, `NEVER`). **Required** if `request_templates` is used.
- `cache_key_parameters` - (Optional) A list of cache key parameters for the integration.
- `cache_namespace` - (Optional) The integration's cache namespace.
- `content_handling` - (Optional) Specifies how to handle request payload content type conversions. Supported values are `CONVERT_TO_BINARY` and `CONVERT_TO_TEXT`. If this property is not defined, the request payload will be passed through from the method request to integration request without modification, provided that the passthroughBehaviors is configured to support payload pass-through.
- `timeout_milliseconds` - (Optional) Custom timeout between 50 and 29,000 milliseconds. The default value is 29,000 milliseconds.

» Import

`aws_api_gateway_integration` can be imported using `REST-API-ID/RESOURCE-ID/HTTP-METHOD`, e.g.

```
$ terraform import aws_api_gateway_integration.example 12345abcde/67890fghij/GET
```

» Resource: aws_api_gateway_integration_response

Provides an HTTP Method Integration Response for an API Gateway Resource.

Note: Depends on having `aws_api_gateway_integration` inside your rest api. To ensure this you might need to add an explicit `depends_on` for clean runs.

» Example Usage

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_resource" "MyDemoResource" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  parent_id   = "${aws_api_gateway_rest_api.MyDemoAPI.root_resource_id}"
  path_part   = "mydemoresource"
}

resource "aws_api_gateway_method" "MyDemoMethod" {
  rest_api_id   = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id   = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method   = "GET"
  authorization = "NONE"
}

resource "aws_api_gateway_integration" "MyDemoIntegration" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method = "${aws_api_gateway_method.MyDemoMethod.http_method}"
  type        = "MOCK"
}

resource "aws_api_gateway_method_response" "response_200" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method = "${aws_api_gateway_method.MyDemoMethod.http_method}"
  status_code = "200"
}

resource "aws_api_gateway_integration_response" "MyDemoIntegrationResponse" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method = "${aws_api_gateway_method.MyDemoMethod.http_method}"
  status_code = "${aws_api_gateway_method_response.response_200.status_code}"

  # Transforms the backend JSON response to XML
  response_templates = {
    "application/xml" = <<EOF
#set($inputRoot = $input.path('$'))
<?xml version="1.0" encoding="UTF-8"?>
<message>
    $inputRoot.body
</message>
EOF
  }
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API
- `resource_id` - (Required) The API resource ID
- `http_method` - (Required) The HTTP method (`GET`, `POST`, `PUT`, `DELETE`, `HEAD`, `OPTIONS`, `ANY`)
- `status_code` - (Required) The HTTP status code
- `selection_pattern` - (Optional) Specifies the regular expression pattern used to choose an integration response based on the response from the backend. Setting this to `-` makes the integration the default one. If the backend is an `AWS` Lambda function, the AWS Lambda function error header is matched. For all other `HTTP` and `AWS` backends, the HTTP status code is matched.
- `response_templates` - (Optional) A map specifying the templates used to transform the integration response body
- `response_parameters` - (Optional) A map of response parameters that can be read from the backend response. For example: `response_parameters = { "method.response.header.X-Some-Header" = "integration.response.header.X-Some-Other-Header" }`
- `content_handling` - (Optional) Specifies how to handle request payload content type conversions. Supported values are `CONVERT_TO_BINARY` and `CONVERT_TO_TEXT`. If this property is not defined, the response payload will be passed through from the integration response to the method response without modification.

» Import

`aws_api_gateway_integration_response` can be imported using `REST-API-ID/RESOURCE-ID/HTTP-METHOD/S` e.g.

```
$ terraform import aws_api_gateway_integration_response.example 12345abcde/67890fghij/GET/20
```

» Resource: aws_api_gateway_method

Provides a HTTP Method for an API Gateway Resource.

» Example Usage

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_resource" "MyDemoResource" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  parent_id   = "${aws_api_gateway_rest_api.MyDemoAPI.root_resource_id}"
  path_part   = "mydemoresource"
}

resource "aws_api_gateway_method" "MyDemoMethod" {
  rest_api_id   = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id   = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method   = "GET"
  authorization = "NONE"
}
```

» Usage with Cognito User Pool Authorizer

```
variable "cognito_user_pool_name" {}

data "aws_cognito_user_pools" "this" {
  name = "${var.cognito_user_pool_name}"
}

resource "aws_api_gateway_rest_api" "this" {
  name = "with-authorizer"
}

resource "aws_api_gateway_resource" "this" {
  rest_api_id = "${aws_api_gateway_rest_api.this.id}"
  parent_id   = "${aws_api_gateway_rest_api.this.root_resource_id}"
  path_part   = "{proxy+}"
}

resource "aws_api_gateway_authorizer" "this" {
  name          = "CognitoUserPoolAuthorizer"
  type          = "COGNITO_USER_POOLS"
  rest_api_id   = "${aws_api_gateway_rest_api.this.id}"
  provider_arns = ["${data.aws_cognito_user_pools.this.arns}"]
}

resource "aws_api_gateway_method" "any" {
  rest_api_id   = "${aws_api_gateway_rest_api.this.id}"
  resource_id   = "${aws_api_gateway_resource.this.id}"
  http_method   = "ANY"
  authorization = "COGNITO_USER_POOLS"
  authorizer_id = "${aws_api_gateway_authorizer.this.id}"

  request_parameters = {
    "method.request.path.proxy" = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API
- `resource_id` - (Required) The API resource ID
- `http_method` - (Required) The HTTP Method (`GET`, `POST`, `PUT`, `DELETE`, `HEAD`, `OPTIONS`, `ANY`)
- `authorization` - (Required) The type of authorization used for the method (`NONE`, `CUSTOM`, `AWS_IAM`, `COGNITO_USER_POOLS`)
- `authorizer_id` - (Optional) The authorizer id to be used when the authorization is `CUSTOM` or `COGNITO_USER_POOLS`
- `authorization_scopes` - (Optional) The authorization scopes used when the authorization is `COGNITO_USER_POOLS`
- `api_key_required` - (Optional) Specify if the method requires an API key
- `request_models` - (Optional) A map of the API models used for the request's content type where key is the content type (e.g. `application/json`) and value is either `Error`, `Empty` (built-in models) or `aws_api_gateway_model`'s `name`.
- `request_validator_id` - (Optional) The ID of a `aws_api_gateway_request_validator`
- `request_parameters` - (Optional) A map of request parameters (from the path, query string and headers) that should be passed to the integration. The boolean value indicates whether the parameter is required (`true`) or optional (`false`). For example: `request_parameters = {"method.request.header.X-Some-Header" = true "method.request.querystring.some-query-param" = true}` would define that the header `X-Some-Header` and the query string

`some-query-param` must be provided in the request.

» Import

`aws_api_gateway_method` can be imported using `REST-API-ID/RESOURCE-ID/HTTP-METHOD`, e.g.

```
$ terraform import aws_api_gateway_method.example 12345abcde/67890fghij/GET
```

» Resource: aws_api_gateway_method_response

Provides an HTTP Method Response for an API Gateway Resource.

» Example Usage

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_resource" "MyDemoResource" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  parent_id   = "${aws_api_gateway_rest_api.MyDemoAPI.root_resource_id}"
  path_part   = "mydemoresource"
}

resource "aws_api_gateway_method" "MyDemoMethod" {
  rest_api_id   = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id   = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method   = "GET"
  authorization = "NONE"
}

resource "aws_api_gateway_integration" "MyDemoIntegration" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method = "${aws_api_gateway_method.MyDemoMethod.http_method}"
  type        = "MOCK"
}

resource "aws_api_gateway_method_response" "response_200" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  resource_id = "${aws_api_gateway_resource.MyDemoResource.id}"
  http_method = "${aws_api_gateway_method.MyDemoMethod.http_method}"
  status_code = "200"
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API
- `resource_id` - (Required) The API resource ID
- `http_method` - (Required) The HTTP Method (`GET`, `POST`, `PUT`, `DELETE`, `HEAD`, `OPTIONS`, `ANY`)
- `status_code` - (Required) The HTTP status code
- `response_models` - (Optional) A map of the API models used for the response's content type
- `response_parameters` - (Optional) A map of response parameters that can be sent to the caller. For example: `response_parameters = { "method.response.header.X-Some-Header" = true }` would define that the header `X-Some-Header` can be provided on the response.

» Import

`aws_api_gateway_method_response` can be imported using `REST-API-ID/RESOURCE-ID/HTTP-METHOD/STATUS` e.g.

```
$ terraform import aws_api_gateway_method_response.example 12345abcde/67890fghij/GET/200
```

» Resource: aws_api_gateway_method_settings

Provides an API Gateway Method Settings, e.g. logging or monitoring.

» Example Usage

```
resource "aws_api_gateway_method_settings" "s" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  stage_name  = "${aws_api_gateway_stage.test.stage_name}"
  method_path = "${aws_api_gateway_resource.test.path_part}/${aws_api_gateway_method.test.ht

  settings {
    metrics_enabled = true
    logging_level   = "INFO"
  }
}

resource "aws_api_gateway_rest_api" "test" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_deployment" "test" {
  depends_on  = ["aws_api_gateway_integration.test"]
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  stage_name  = "dev"
}

resource "aws_api_gateway_stage" "test" {
  stage_name    = "prod"
  rest_api_id   = "${aws_api_gateway_rest_api.test.id}"
  deployment_id = "${aws_api_gateway_deployment.test.id}"
}

resource "aws_api_gateway_resource" "test" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  parent_id   = "${aws_api_gateway_rest_api.test.root_resource_id}"
  path_part   = "mytestresource"
}

resource "aws_api_gateway_method" "test" {
  rest_api_id   = "${aws_api_gateway_rest_api.test.id}"
  resource_id   = "${aws_api_gateway_resource.test.id}"
  http_method   = "GET"
  authorization = "NONE"
}

resource "aws_api_gateway_integration" "test" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  resource_id = "${aws_api_gateway_resource.test.id}"
  http_method = "${aws_api_gateway_method.test.http_method}"
  type        = "MOCK"

  request_templates = {
    "application/xml" = <<EOF
{
   "body" : $input.json('$')
}
EOF
  }
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the REST API
- `stage_name` - (Required) The name of the stage
- `method_path` - (Required) Method path defined as `{resource_path}/{http_method}` for an individual method override, or `*/*` for overriding all methods in the stage.
- `settings` - (Required) The settings block, see below.

» `settings`

- `metrics_enabled` - (Optional) Specifies whether Amazon CloudWatch metrics are enabled for this method.
- `logging_level` - (Optional) Specifies the logging level for this method, which effects the log entries pushed to Amazon CloudWatch Logs. The available levels are `OFF`, `ERROR`, and `INFO`.
- `data_trace_enabled` - (Optional) Specifies whether data trace logging is enabled for this method, which effects the log entries pushed to Amazon CloudWatch Logs.
- `throttling_burst_limit` - (Optional) Specifies the throttling burst limit.
- `throttling_rate_limit` - (Optional) Specifies the throttling rate limit.
- `caching_enabled` - (Optional) Specifies whether responses should be cached and returned for requests. A cache cluster must be enabled on the stage for responses to be cached.
- `cache_ttl_in_seconds` - (Optional) Specifies the time to live (TTL), in seconds, for cached responses. The higher the TTL, the longer the response will be cached.
- `cache_data_encrypted` - (Optional) Specifies whether the cached responses are encrypted.
- `require_authorization_for_cache_control` - (Optional) Specifies whether authorization is required for a cache invalidation request.
- `unauthorized_cache_control_header_strategy` - (Optional) Specifies how to handle unauthorized requests for cache invalidation. The available values are `FAIL_WITH_403`, `SUCCEED_WITH_RESPONSE_HEADER`, `SUCCEED_WITHOUT_RESPONSE_HEADER`.

» Resource: aws_api_gateway_model

Provides a Model for a API Gateway.

» Example Usage

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_model" "MyDemoModel" {
  rest_api_id  = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  name         = "user"
  description  = "a JSON schema"
  content_type = "application/json"

  schema = <<EOF
{
  "type": "object"
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API
- `name` - (Required) The name of the model
- `description` - (Optional) The description of the model
- `content_type` - (Required) The content type of the model
- `schema` - (Required) The schema of the model in a JSON form

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the model

» Import

`aws_api_gateway_model` can be imported using `REST-API-ID/NAME`, e.g.

```
$ terraform import aws_api_gateway_model.example 12345abcde/example
```

» Resource: aws_api_gateway_request_validator

Manages an API Gateway Request Validator.

» Example Usage

```
resource "aws_api_gateway_request_validator" "example" {
  name                        = "example"
  rest_api_id                 = "${aws_api_gateway_rest_api.example.id}"
  validate_request_body       = true
  validate_request_parameters = true
}
```

» Argument Reference

The following argument is supported:

- `name` - (Required) The name of the request validator
- `rest_api_id` - (Required) The ID of the associated Rest API
- `validate_request_body` - (Optional) Boolean whether to validate request body. Defaults to `false`.
- `validate_request_parameters` - (Optional) Boolean whether to validate request parameters. Defaults to `false`.

» Attribute Reference

The following attribute is exported in addition to the arguments listed above:

- `id` - The unique ID of the request validator

» Import

`aws_api_gateway_request_validator` can be imported using `REST-API-ID/REQUEST-VALIDATOR-ID`, e.g.

```
$ terraform import aws_api_gateway_request_validator.example 12345abcde/67890fghij
```

» Resource: aws_api_gateway_resource

Provides an API Gateway Resource.

» Example Usage

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_resource" "MyDemoResource" {
  rest_api_id = "${aws_api_gateway_rest_api.MyDemoAPI.id}"
  parent_id   = "${aws_api_gateway_rest_api.MyDemoAPI.root_resource_id}"
  path_part   = "mydemoresource"
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API
- `parent_id` - (Required) The ID of the parent API resource
- `path_part` - (Required) The last path segment of this API resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The resource's identifier.
- `path` - The complete path for this API resource, including all parent paths.

» Import

`aws_api_gateway_resource` can be imported using `REST-API-ID/RESOURCE-ID`, e.g.

```
$ terraform import aws_api_gateway_resource.example 12345abcde/67890fghij
```

» Resource: aws_api_gateway_rest_api

Provides an API Gateway REST API.

» Example Usage

» Basic

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}
```

» Regional Endpoint Type

```
resource "aws_api_gateway_rest_api" "example" {
  name = "regional-example"

  endpoint_configuration {
    types = ["REGIONAL"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the REST API
- `description` - (Optional) The description of the REST API
- `endpoint_configuration` - (Optional) Nested argument defining API endpoint configuration including endpoint type. Defined below.
- `binary_media_types` - (Optional) The list of binary media types supported by the RestApi. By default, the RestApi supports only UTF-8-encoded text payloads.
- `minimum_compression_size` - (Optional) Minimum response size to compress for the REST API. Integer between -1 and 10485760 (10MB). Setting a value greater than -1 will enable compression, -1 disables compression (default).
- `body` - (Optional) An OpenAPI specification that defines the set of routes and integrations to create as part of the REST API.
- `policy` - (Optional) JSON formatted policy document that controls access to the API Gateway. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide
- `api_key_source` - (Optional) The source of the API key for requests. Valid values are HEADER (default) and AUTHORIZER.
- `tags` - (Optional) Key-value mapping of resource tags

Note: If the `body` argument is provided, the OpenAPI specification will be used to configure the resources, methods and integrations for the Rest API. If this argument is provided, the following resources should not be managed as separate ones, as updates may cause manual resource updates to be overwritten:

- `aws_api_gateway_resource`
- `aws_api_gateway_method`
- `aws_api_gateway_method_response`
- `aws_api_gateway_method_settings`
- `aws_api_gateway_integration`
- `aws_api_gateway_integration_response`
- `aws_api_gateway_gateway_response`
- `aws_api_gateway_model`

» endpoint_configuration

- `types` - (Required) A list of endpoint types. This resource currently only supports managing a single value. Valid values: `EDGE`, `REGIONAL` or `PRIVATE`. If unspecified, defaults to `EDGE`. Must be declared as `REGIONAL` in non-Commercial partitions. Refer to the documentation for more information on the difference between edge-optimized and regional APIs.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the REST API
- `root_resource_id` - The resource ID of the REST API's root
- `created_date` - The creation date of the REST API
- `execution_arn` - The execution ARN part to be used in `lambda_permission`'s `source_arn` when allowing API Gateway to invoke a Lambda function, e.g. `arn:aws:execute-api:eu-west-2:123456789012:z4675bid1j`, which can be concatenated with allowed stage, method and resource path.
- `arn` - Amazon Resource Name (ARN)

» Import

`aws_api_gateway_rest_api` can be imported by using the REST API ID, e.g.

```
$ terraform import aws_api_gateway_rest_api.example 12345abcde
```

NOTE: Resource import does not currently support the `body` attribute.

» Resource: aws_api_gateway_stage

Provides an API Gateway Stage.

» Example Usage

```
resource "aws_api_gateway_stage" "test" {
  stage_name    = "prod"
  rest_api_id   = "${aws_api_gateway_rest_api.test.id}"
  deployment_id = "${aws_api_gateway_deployment.test.id}"
}

resource "aws_api_gateway_rest_api" "test" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_api_gateway_deployment" "test" {
  depends_on  = ["aws_api_gateway_integration.test"]
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  stage_name  = "dev"
}

resource "aws_api_gateway_resource" "test" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  parent_id   = "${aws_api_gateway_rest_api.test.root_resource_id}"
  path_part   = "mytestresource"
}

resource "aws_api_gateway_method" "test" {
  rest_api_id   = "${aws_api_gateway_rest_api.test.id}"
  resource_id   = "${aws_api_gateway_resource.test.id}"
  http_method   = "GET"
  authorization = "NONE"
}

resource "aws_api_gateway_method_settings" "s" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  stage_name  = "${aws_api_gateway_stage.test.stage_name}"
  method_path = "${aws_api_gateway_resource.test.path_part}/${aws_api_gateway_method.test.ht

  settings {
    metrics_enabled = true
    logging_level   = "INFO"
  }
}

resource "aws_api_gateway_integration" "test" {
  rest_api_id = "${aws_api_gateway_rest_api.test.id}"
  resource_id = "${aws_api_gateway_resource.test.id}"
  http_method = "${aws_api_gateway_method.test.http_method}"
  type        = "MOCK"
}
```

» Managing the API Logging CloudWatch Log Group

API Gateway provides the ability to enable CloudWatch API logging. To manage the CloudWatch Log Group when this feature is enabled, the `aws_cloudwatch_log_group` resource can be used where the name matches the API Gateway naming convention. If the CloudWatch Log Group previously exists, the `aws_cloudwatch_log_group` resource can be imported into Terraform as a one time operation and recreation of the environment can occur without import.

The below configuration uses `depends_on` to prevent ordering issues with API Gateway automatically creating the log group first and a variable for naming consistency. Other ordering and naming methodologies may be more appropriate for your environment.

```
variable "stage_name" {
  default = "example"
  type    = "string"
}

resource "aws_api_gateway_rest_api" "example" {
  # ... other configuration ...
}

resource "aws_api_gateway_stage" "example" {
  depends_on = ["aws_cloudwatch_log_group.example"]

  name = "${var.stage_name}"

  # ... other configuration ...
}

resource "aws_cloudwatch_log_group" "example" {
  name              = "API-Gateway-Execution-Logs_${aws_api_gateway_rest_api.example.id}/${v
  retention_in_days = 7
  # ... potentially other configuration ...
}
```

» Argument Reference

The following arguments are supported:

- `rest_api_id` - (Required) The ID of the associated REST API
- `stage_name` - (Required) The name of the stage
- `deployment_id` - (Required) The ID of the deployment that the stage points to
- `access_log_settings` - (Optional) Enables access logs for the API stage. Detailed below.
- `cache_cluster_enabled` - (Optional) Specifies whether a cache cluster is enabled for the stage
- `cache_cluster_size` - (Optional) The size of the cache cluster for the stage, if enabled. Allowed values include `0.5`, `1.6`, `6.1`, `13.5`, `28.4`, `58.2`, `118` and `237`.
- `client_certificate_id` - (Optional) The identifier of a client certificate for the stage.
- `description` - (Optional) The description of the stage
- `documentation_version` - (Optional) The version of the associated API documentation
- `variables` - (Optional) A map that defines the stage variables
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `xray_tracing_enabled` - (Optional) Whether active tracing with X-ray is enabled. Defaults to `false`.

» Nested Blocks

» `access_log_settings`

- `destination_arn` - (Required) The Amazon Resource Name (ARN) of the CloudWatch Logs log group or Kinesis Data Firehose delivery stream to receive access logs. If you specify a Kinesis Data Firehose delivery stream, the stream name must begin with `amazon-apigateway-`. Automatically removes trailing `:*` if present.
- `format` - (Required) The formatting and values recorded in the logs. For more information on configuring the log format rules visit the AWS documentation

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN)
- `id` - The ID of the stage
- `invoke_url` - The URL to invoke the API pointing to the stage, e.g. `https://z4675bid1j.execute-api.eu-west-2.amazonaws.com/prod`
- `execution_arn` - The execution ARN to be used in `lambda_permission`'s `source_arn` when allowing API Gateway to invoke a Lambda function, e.g. `arn:aws:execute-api:eu-west-2:123456789012:z4675bid1j/prod`

» Import

`aws_api_gateway_stage` can be imported using `REST-API-ID/STAGE-NAME`, e.g.

```
$ terraform import aws_api_gateway_stage.example 12345abcde/example
```

» Resource: aws_api_gateway_usage_plan

Provides an API Gateway Usage Plan.

» Example Usage

```
resource "aws_api_gateway_rest_api" "myapi" {
  name = "MyDemoAPI"
}

# ...

resource "aws_api_gateway_deployment" "dev" {
  rest_api_id = "${aws_api_gateway_rest_api.myapi.id}"
  stage_name  = "dev"
}

resource "aws_api_gateway_deployment" "prod" {
  rest_api_id = "${aws_api_gateway_rest_api.myapi.id}"
  stage_name  = "prod"
}

resource "aws_api_gateway_usage_plan" "MyUsagePlan" {
  name         = "my-usage-plan"
  description  = "my description"
  product_code = "MYCODE"

  api_stages {
    api_id = "${aws_api_gateway_rest_api.myapi.id}"
    stage  = "${aws_api_gateway_deployment.dev.stage_name}"
  }

  api_stages {
    api_id = "${aws_api_gateway_rest_api.myapi.id}"
    stage  = "${aws_api_gateway_deployment.prod.stage_name}"
  }

  quota_settings {
    limit  = 20
    offset = 2
    period = "WEEK"
  }

  throttle_settings {
    burst_limit = 5
    rate_limit  = 10
  }
}
```

» Argument Reference

The API Gateway Usage Plan argument layout is a structure composed of several sub-resources - these resources are laid out below.

» Top-Level Arguments

- `name` - (Required) The name of the usage plan.
- `description` - (Optional) The description of a usage plan.
- `api_stages` - (Optional) The associated API stages of the usage plan.
- `quota_settings` - (Optional) The quota settings of the usage plan.
- `throttle_settings` - (Optional) The throttling limits of the usage plan.
- `product_code` - (Optional) The AWS Markeplace product identifier to associate with the usage plan as a SaaS product on AWS Marketplace.
- `tags` - (Optional) Key-value mapping of resource tags

» Api Stages arguments

- `api_id` (Required) - API Id of the associated API stage in a usage plan.
- `stage` (Required) - API stage name of the associated API stage in a usage plan.

» Quota Settings Arguments

- `limit` (Optional) - The maximum number of requests that can be made in a given time period.
- `offset` (Optional) - The number of requests subtracted from the given limit in the initial time period.
- `period` (Optional) - The time period in which the limit applies. Valid values are ”DAY”, ”WEEK” or ”MONTH”.

» Throttling Settings Arguments

- `burst_limit` (Optional) - The API request burst limit, the maximum rate limit over a time ranging from one to a few seconds, depending upon whether the underlying token bucket is at its full capacity.
- `rate_limit` (Optional) - The API request steady-state rate limit.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the API resource
- `name` - The name of the usage plan.
- `description` - The description of a usage plan.
- `api_stages` - The associated API stages of the usage plan.
- `quota_settings` - The quota of the usage plan.
- `throttle_settings` - The throttling limits of the usage plan.
- `product_code` - The AWS Markeplace product identifier to associate with the usage plan as a SaaS product on AWS Marketplace.
- `arn` - Amazon Resource Name (ARN)

» Import

AWS API Gateway Usage Plan can be imported using the `id`, e.g.

```
$ terraform import aws_api_gateway_usage_plan.myusageplan <usage_plan_id>
```

» Resource: aws_api_gateway_usage_plan_key

Provides an API Gateway Usage Plan Key.

» Example Usage

```
resource "aws_api_gateway_rest_api" "test" {
  name = "MyDemoAPI"
}

# ...

resource "aws_api_gateway_usage_plan" "myusageplan" {
  name = "my_usage_plan"

  api_stages {
    api_id = "${aws_api_gateway_rest_api.test.id}"
    stage  = "${aws_api_gateway_deployment.foo.stage_name}"
  }
}

resource "aws_api_gateway_api_key" "mykey" {
  name = "my_key"
}

resource "aws_api_gateway_usage_plan_key" "main" {
  key_id        = "${aws_api_gateway_api_key.mykey.id}"
  key_type      = "API_KEY"
  usage_plan_id = "${aws_api_gateway_usage_plan.myusageplan.id}"
}
```

» Argument Reference

The following arguments are supported:

- `key_id` - (Required) The identifier of the API key resource.
- `key_type` - (Required) The type of the API key resource. Currently, the valid key type is API_KEY.
- `usage_plan_id` - (Required) The Id of the usage plan resource representing to associate the key to.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Id of a usage plan key.
- `key_id` - The identifier of the API gateway key resource.
- `key_type` - The type of a usage plan key. Currently, the valid key type is API_KEY.
- `usage_plan_id` - The ID of the API resource
- `name` - The name of a usage plan key.
- `value` - The value of a usage plan key.

» Import

AWS API Gateway Usage Plan Key can be imported using the `USAGE-PLAN-ID/USAGE-PLAN-KEY-ID`, e.g.

```
$ terraform import aws_api_gateway_usage_plan_key.key 12345abcde/zzz
```

» Resource: aws_api_gateway_vpc_link

Provides an API Gateway VPC Link.

» Example Usage

```
resource "aws_lb" "example" {
  name               = "example"
  internal           = true
  load_balancer_type = "network"

  subnet_mapping {
    subnet_id = "12345"
  }
}

resource "aws_api_gateway_vpc_link" "example" {
  name        = "example"
  description = "example description"
  target_arns = ["${aws_lb.example.arn}"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name used to label and identify the VPC link.
- `description` - (Optional) The description of the VPC link.
- `target_arns` - (Required, ForceNew) The list of network load balancer arns in the VPC targeted by the VPC link. Currently AWS only supports 1 target.
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The identifier of the VpcLink.

» Import

API Gateway VPC Link can be imported using the `id`, e.g.

```
$ terraform import aws_api_gateway_vpc_link.example <vpc_link_id>
```

» Resource: aws_appautoscaling_policy

Provides an Application AutoScaling Policy resource.

» Example Usage

» DynamoDB Table Autoscaling

```
resource "aws_appautoscaling_target" "dynamodb_table_read_target" {
  max_capacity       = 100
  min_capacity       = 5
  resource_id        = "table/tableName"
  role_arn           = "${data.aws_iam_role.DynamoDBAutoscaleRole.arn}"
  scalable_dimension = "dynamodb:table:ReadCapacityUnits"
  service_namespace  = "dynamodb"
}

resource "aws_appautoscaling_policy" "dynamodb_table_read_policy" {
  name               = "DynamoDBReadCapacityUtilization:${aws_appautoscaling_target.dynamodb
  policy_type        = "TargetTrackingScaling"
  resource_id        = "${aws_appautoscaling_target.dynamodb_table_read_target.resource_id}"
  scalable_dimension = "${aws_appautoscaling_target.dynamodb_table_read_target.scalable_dime
  service_namespace  = "${aws_appautoscaling_target.dynamodb_table_read_target.service_names

  target_tracking_scaling_policy_configuration {
    predefined_metric_specification {
      predefined_metric_type = "DynamoDBReadCapacityUtilization"
    }

    target_value = 70
  }
}
```

» ECS Service Autoscaling

```
resource "aws_appautoscaling_target" "ecs_target" {
  max_capacity       = 4
  min_capacity       = 1
  resource_id        = "service/clusterName/serviceName"
  role_arn           = "${var.ecs_iam_role}"
  scalable_dimension = "ecs:service:DesiredCount"
  service_namespace  = "ecs"
}

resource "aws_appautoscaling_policy" "ecs_policy" {
  name               = "scale-down"
  policy_type        = "StepScaling"
  resource_id        = "${aws_appautoscaling_target.ecs_target.resource_id}"
  scalable_dimension = "${aws_appautoscaling_target.ecs_target.scalable_dimension}"
  service_namespace  = "${aws_appautoscaling_target.ecs_target.service_namespace}"

  step_scaling_policy_configuration {
    adjustment_type         = "ChangeInCapacity"
    cooldown                = 60
    metric_aggregation_type = "Maximum"

    step_adjustment {
      metric_interval_upper_bound = 0
      scaling_adjustment          = -1
    }
  }
}
```

» Preserve desired count when updating an autoscaled ECS Service

```
resource "aws_ecs_service" "ecs_service" {
  name            = "serviceName"
  cluster         = "clusterName"
  task_definition = "taskDefinitionFamily:1"
  desired_count   = 2

  lifecycle {
    ignore_changes = ["desired_count"]
  }
}
```

» Aurora Read Replica Autoscaling

```
resource "aws_appautoscaling_target" "replicas" {
  service_namespace  = "rds"
  scalable_dimension = "rds:cluster:ReadReplicaCount"
  resource_id        = "cluster:${aws_rds_cluster.example.id}"
  min_capacity       = 1
  max_capacity       = 15
}

resource "aws_appautoscaling_policy" "replicas" {
  name               = "cpu-auto-scaling"
  service_namespace  = "${aws_appautoscaling_target.replicas.service_namespace}"
  scalable_dimension = "${aws_appautoscaling_target.replicas.scalable_dimension}"
  resource_id        = "${aws_appautoscaling_target.replicas.resource_id}"
  policy_type        = "TargetTrackingScaling"

  target_tracking_scaling_policy_configuration {
    predefined_metric_specification {
      predefined_metric_type = "RDSReaderAverageCPUUtilization"
    }

    target_value       = 75
    scale_in_cooldown  = 300
    scale_out_cooldown = 300
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the policy.
- `policy_type` - (Optional) For DynamoDB, only `TargetTrackingScaling` is supported. For Amazon ECS, Spot Fleet, and Amazon RDS, both `StepScaling` and `TargetTrackingScaling` are supported. For any other service, only `StepScaling` is supported. Defaults to `StepScaling`.
- `resource_id` - (Required) The resource type and unique identifier string for the resource associated with the scaling policy. Documentation can be found in the `ResourceId` parameter at: AWS Application Auto Scaling API Reference
- `scalable_dimension` - (Required) The scalable dimension of the scalable target. Documentation can be found in the `ScalableDimension` parameter at: AWS Application Auto Scaling API Reference

- `service_namespace` - (Required) The AWS service namespace of the scalable target. Documentation can be found in the `ServiceNamespace` parameter at: AWS Application Auto Scaling API Reference
- `step_scaling_policy_configuration` - (Optional) Step scaling policy configuration, requires `policy_type = "StepScaling"` (default). See supported fields below.
- `target_tracking_scaling_policy_configuration` - (Optional) A target tracking policy, requires `policy_type = "TargetTrackingScaling"`. See supported fields below.

» Nested fields

» `step_scaling_policy_configuration`

- `adjustment_type` - (Required) Specifies whether the adjustment is an absolute number or a percentage of the current capacity. Valid values are `ChangeInCapacity`, `ExactCapacity`, and `PercentChangeInCapacity`.
- `cooldown` - (Required) The amount of time, in seconds, after a scaling activity completes and before the next scaling activity can start.
- `metric_aggregation_type` - (Optional) The aggregation type for the policy's metrics. Valid values are ”Minimum”, ”Maximum”, and ”Average”. Without a value, AWS will treat the aggregation type as ”Average”.
- `min_adjustment_magnitude` - (Optional) The minimum number to adjust your scalable dimension as a result of a scaling activity. If the adjustment type is PercentChangeInCapacity, the scaling policy changes the scalable dimension of the scalable target by this amount.
- `step_adjustment` - (Optional) A set of adjustments that manage scaling. These have the following structure:

```
resource "aws_appautoscaling_policy" "ecs_policy" {
  # ...

  step_scaling_policy_configuration {
    # insert config here

    step_adjustment {
      metric_interval_lower_bound = 1.0
      metric_interval_upper_bound = 2.0
      scaling_adjustment          = -1
    }

    step_adjustment {
      metric_interval_lower_bound = 2.0
      metric_interval_upper_bound = 3.0
      scaling_adjustment          = 1
    }
  }
}
```

- `metric_interval_lower_bound` - (Optional) The lower bound for the difference between the alarm threshold and the CloudWatch metric. Without a value, AWS will treat this bound as negative infinity.
- `metric_interval_upper_bound` - (Optional) The upper bound for the difference between the alarm threshold and the CloudWatch metric. Without a value, AWS will treat this bound as infinity. The upper bound must be greater than the lower bound.
- `scaling_adjustment` - (Required) The number of members by which to scale, when the adjustment bounds are breached. A positive value scales up. A negative value scales down.

» `target_tracking_scaling_policy_configuration`

- `target_value` - (Required) The target value for the metric.
- `disable_scale_in` - (Optional) Indicates whether scale in by the target tracking policy is disabled. If the value is true, scale in is disabled and the target tracking policy won't remove capacity from the scalable resource. Otherwise, scale in is enabled and the target tracking policy can remove capacity from the scalable resource. The default value is `false`.
- `scale_in_cooldown` - (Optional) The amount of time, in seconds, after a scale in activity completes before another scale in activity can start.
- `scale_out_cooldown` - (Optional) The amount of time, in seconds, after a scale out activity completes before another scale out activity can start.
- `customized_metric_specification` - (Optional) A custom CloudWatch metric. Documentation can be found at: AWS Customized Metric Specification. See supported fields below.
- `predefined_metric_specification` - (Optional) A predefined metric. See supported fields below.

» `customized_metric_specification`

- `dimensions` - (Optional) The dimensions of the metric.
- `metric_name` - (Required) The name of the metric.
- `namespace` - (Required) The namespace of the metric.
- `statistic` - (Required) The statistic of the metric.
- `unit` - (Optional) The unit of the metric.

» `predefined_metric_specification`

- `predefined_metric_type` - (Required) The metric type.

- `resource_label` - (Optional) Reserved for future use.

» Attribute Reference

- `arn` - The ARN assigned by AWS to the scaling policy.
- `name` - The scaling policy's name.
- `policy_type` - The scaling policy's type.

» Import

Application AutoScaling Policy can be imported using the `service-namespace`
, `resource-id`, `scalable-dimension` and `policy-name` separated by `/`.

```
$ terraform import aws_appautoscaling_policy.test-policy service-namespace/resource-id/scala
```

» Resource: aws_appautoscaling_scheduled_action

Provides an Application AutoScaling ScheduledAction resource.

» Example Usage

» DynamoDB Table Autoscaling

```
resource "aws_appautoscaling_target" "dynamodb" {
  max_capacity       = 100
  min_capacity       = 5
  resource_id        = "table/tableName"
  role_arn           = "${data.aws_iam_role.DynamoDBAutoscaleRole.arn}"
  scalable_dimension = "dynamodb:table:ReadCapacityUnits"
  service_namespace  = "dynamodb"
}

resource "aws_appautoscaling_scheduled_action" "dynamodb" {
  name               = "dynamodb"
  service_namespace  = "${aws_appautoscaling_target.dynamodb.service_namespace}"
  resource_id        = "${aws_appautoscaling_target.dynamodb.resource_id}"
  scalable_dimension = "${aws_appautoscaling_target.dynamodb.scalable_dimension}"
  schedule           = "at(2006-01-02T15:04:05)"

  scalable_target_action {
    min_capacity = 1
    max_capacity = 200
  }
}
```

» ECS Service Autoscaling

```
resource "aws_appautoscaling_target" "ecs" {
  max_capacity       = 4
  min_capacity       = 1
  resource_id        = "service/clusterName/serviceName"
  role_arn           = "${var.ecs_iam_role}"
  scalable_dimension = "ecs:service:DesiredCount"
  service_namespace  = "ecs"
}

resource "aws_appautoscaling_scheduled_action" "ecs" {
  name               = "ecs"
  service_namespace  = "${aws_appautoscaling_target.ecs.service_namespace}"
  resource_id        = "${aws_appautoscaling_target.ecs.resource_id}"
  scalable_dimension = "${aws_appautoscaling_target.ecs.scalable_dimension}"
  schedule           = "at(2006-01-02T15:04:05)"

  scalable_target_action {
    min_capacity = 1
    max_capacity = 10
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the scheduled action.
- `service_namespace` - (Required) The namespace of the AWS service. Documentation can be found in the parameter at: AWS Application Auto Scaling API Reference Example: ecs
- `resource_id` - (Required) The identifier of the resource associated with the scheduled action. Documentation can be found in the parameter at: AWS Application Auto Scaling API Reference
- `scalable_dimension` - (Optional) The scalable dimension. Documentation can be found in the parameter at: AWS Application Auto Scaling API Reference Example: ecs:service:DesiredCount
- `scalable_target_action` - (Optional) The new minimum and maximum capacity. You can set both values or just one. See below
- `schedule` - (Optional) The schedule for this action. The following formats are supported: At expressions - at(yyyy-mm-ddThh:mm:ss), Rate

expressions - rate(valueunit), Cron expressions - cron(fields). In UTC. Documentation can be found in the parameter at: AWS Application Auto Scaling API Reference
- `start_time` - (Optional) The date and time for the scheduled action to start. Specify the following format: 2006-01-02T15:04:05Z
- `end_time` - (Optional) The date and time for the scheduled action to end. Specify the following format: 2006-01-02T15:04:05Z

» Scalable Target Action Arguments

- `max_capacity` - (Optional) The maximum capacity.
- `min_capacity` - (Optional) The minimum capacity.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the scheduled action.

» Resource: aws_appautoscaling_target

Provides an Application AutoScaling ScalableTarget resource. To manage policies which get attached to the target, see the `aws_appautoscaling_policy` resource.

» Example Usage

» DynamoDB Table Autoscaling

```
resource "aws_appautoscaling_target" "dynamodb_table_read_target" {
  max_capacity       = 100
  min_capacity       = 5
  resource_id        = "table/${aws_dynamodb_table.example.name}"
  role_arn           = "${data.aws_iam_role.DynamoDBAutoscaleRole.arn}"
  scalable_dimension = "dynamodb:table:ReadCapacityUnits"
  service_namespace  = "dynamodb"
}
```

» DynamoDB Index Autoscaling

```
resource "aws_appautoscaling_target" "dynamodb_index_read_target" {
  max_capacity       = 100
  min_capacity       = 5
  resource_id        = "table/${aws_dynamodb_table.example.name}/index/${var.index_name}"
  role_arn           = "${data.aws_iam_role.DynamoDBAutoscaleRole.arn}"
  scalable_dimension = "dynamodb:index:ReadCapacityUnits"
  service_namespace  = "dynamodb"
}
```

» ECS Service Autoscaling

```
resource "aws_appautoscaling_target" "ecs_target" {
  max_capacity       = 4
  min_capacity       = 1
  resource_id        = "service/${aws_ecs_cluster.example.name}/${aws_ecs_service.example.na
  role_arn           = "${var.ecs_iam_role}"
  scalable_dimension = "ecs:service:DesiredCount"
  service_namespace  = "ecs"
}
```

» Aurora Read Replica Autoscaling

```
resource "aws_appautoscaling_target" "replicas" {
  service_namespace  = "rds"
  scalable_dimension = "rds:cluster:ReadReplicaCount"
  resource_id        = "cluster:${aws_rds_cluster.example.id}"
  min_capacity       = 1
  max_capacity       = 15
}
```

» Argument Reference

The following arguments are supported:

- `max_capacity` - (Required) The max capacity of the scalable target.
- `min_capacity` - (Required) The min capacity of the scalable target.
- `resource_id` - (Required) The resource type and unique identifier string for the resource associated with the scaling policy. Documentation can be found in the `ResourceId` parameter at: AWS Application Auto Scaling API Reference
- `role_arn` - (Optional) The ARN of the IAM role that allows Application AutoScaling to modify your scalable target on your behalf.
- `scalable_dimension` - (Required) The scalable dimension of the scalable target. Documentation can be found in the `ScalableDimension` parameter at: AWS Application Auto Scaling API Reference

- service_namespace - (Required) The AWS service namespace of the scalable target. Documentation can be found in the ServiceNamespace parameter at: AWS Application Auto Scaling API Reference

» Import

Application AutoScaling Target can be imported using the service-namespace , resource-id and scalable-dimension separated by /.

```
$ terraform import aws_appautoscaling_target.test-target service-namespace/resource-id/scala
```

» Resource: aws_appmesh_mesh

Provides an AWS App Mesh service mesh resource.

» Example Usage

» Basic

```
resource "aws_appmesh_mesh" "simple" {
  name = "simpleapp"
}
```

» Egress Filter

```
resource "aws_appmesh_mesh" "simple" {
  name = "simpleapp"

  spec {
    egress_filter {
      type = "ALLOW_ALL"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- name - (Required) The name to use for the service mesh.
- spec - (Optional) The service mesh specification to apply.
- tags - (Optional) A mapping of tags to assign to the resource.

The `spec` object supports the following:

- `egress_filter`- (Optional) The egress filter rules for the service mesh.

The `egress_filter` object supports the following:

- `type` - (Optional) The egress filter type. By default, the type is `DROP_ALL`. Valid values are `ALLOW_ALL` and `DROP_ALL`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the service mesh.
- `arn` - The ARN of the service mesh.
- `created_date` - The creation date of the service mesh.
- `last_updated_date` - The last update date of the service mesh.

» Import

App Mesh service meshes can be imported using the `name`, e.g.

```
$ terraform import aws_appmesh_mesh.simple simpleapp
```

» Resource: aws_appmesh_route

Provides an AWS App Mesh route resource.

» Example Usage

» HTTP Routing

```
resource "aws_appmesh_route" "serviceb" {
  name                = "serviceB-route"
  mesh_name           = "${aws_appmesh_mesh.simple.id}"
  virtual_router_name = "${aws_appmesh_virtual_router.serviceb.name}"

  spec {
    http_route {
      match {
        prefix = "/"
      }

      action {
        weighted_target {
          virtual_node = "${aws_appmesh_virtual_node.serviceb1.name}"
          weight       = 90
        }

        weighted_target {
          virtual_node = "${aws_appmesh_virtual_node.serviceb2.name}"
          weight       = 10
        }
      }
    }
  }
}
```

» TCP Routing

```
resource "aws_appmesh_route" "serviceb" {
  name                = "serviceB-route"
  mesh_name           = "${aws_appmesh_mesh.simple.id}"
  virtual_router_name = "${aws_appmesh_virtual_router.serviceb.name}"

  spec {
    tcp_route {
      action {
        weighted_target {
          virtual_node = "${aws_appmesh_virtual_node.serviceb1.name}"
          weight       = 100
        }
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name to use for the route.
- `mesh_name` - (Required) The name of the service mesh in which to create the route.
- `virtual_router_name` - (Required) The name of the virtual router in which to create the route.
- `spec` - (Required) The route specification to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `spec` object supports the following:

- `http_route` - (Optional) The HTTP routing information for the route.
- `tcp_route` - (Optional) The TCP routing information for the route.

The `http_route` object supports the following:

- `action` - (Required) The action to take if a match is determined.
- `match` - (Required) The criteria for determining an HTTP request match.

The `tcp_route` object supports the following:

- `action` - (Required) The action to take if a match is determined.

The `action` object supports the following:

- `weighted_target` - (Required) The targets that traffic is routed to when a request matches the route. You can specify one or more targets and their relative weights with which to distribute traffic.

The `match` object supports the following:

- `prefix` - (Required) Specifies the path with which to match requests. This parameter must always start with /, which by itself matches all requests to the virtual router service name.

The `weighted_target` object supports the following:

- `virtual_node` - (Required) The virtual node to associate with the weighted target.
- `weight` - (Required) The relative weight of the weighted target. An integer between 0 and 100.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the route.
- `arn` - The ARN of the route.
- `created_date` - The creation date of the route.
- `last_updated_date` - The last update date of the route.

» Import

App Mesh virtual routes can be imported using `mesh_name` and `virtual_router_name` together with the route's `name`, e.g.

```
$ terraform import aws_appmesh_virtual_route.serviceb simpleapp/serviceB/serviceB-route
```

» Resource: aws_appmesh_virtual_node

Provides an AWS App Mesh virtual node resource.

» Breaking Changes

Because of backward incompatible API changes (read here), `aws_appmesh_virtual_node` resource definitions created with provider versions earlier than v2.3.0 will need to be modified:

- Rename the `service_name` attribute of the `dns` object to `hostname`.
- Replace the `backends` attribute of the `spec` object with one or more `backend` configuration blocks, setting `virtual_service_name` to the name of the service.

The Terraform state associated with existing resources will automatically be migrated.

» Example Usage

» Basic

```
resource "aws_appmesh_virtual_node" "serviceb1" {
  name      = "serviceBv1"
  mesh_name = "${aws_appmesh_mesh.simple.id}"

  spec {
    backend {
      virtual_service {
        virtual_service_name = "servicea.simpleapp.local"
      }
    }

    listener {
      port_mapping {
        port     = 8080
        protocol = "http"
      }
    }

    service_discovery {
      dns {
        hostname = "serviceb.simpleapp.local"
      }
    }
  }
}
```

» AWS Cloud Map Service Discovery

```
resource "aws_service_discovery_http_namespace" "example" {
  name = "example-ns"
}

resource "aws_appmesh_virtual_node" "serviceb1" {
  name      = "serviceBv1"
  mesh_name = "${aws_appmesh_mesh.simple.id}"

  spec {
    backend {
      virtual_service {
        virtual_service_name = "servicea.simpleapp.local"
      }
    }

    listener {
      port_mapping {
        port     = 8080
        protocol = "http"
      }
    }

    service_discovery {
      aws_cloud_map {
        attributes = {
          stack = "blue"
        }

        service_name   = "serviceb1"
        namespace_name = "${aws_service_discovery_http_namespace.example.name}"
      }
    }
  }
}
```

» Listener Health Check

```
resource "aws_appmesh_virtual_node" "serviceb1" {
  name      = "serviceBv1"
  mesh_name = "${aws_appmesh_mesh.simple.id}"

  spec {
    backend {
      virtual_service {
        virtual_service_name = "servicea.simpleapp.local"
      }
    }

    listener {
      port_mapping {
        port     = 8080
        protocol = "http"
      }

      health_check {
        protocol            = "http"
        path                = "/ping"
        healthy_threshold   = 2
        unhealthy_threshold = 2
        timeout_millis      = 2000
        interval_millis     = 5000
      }
    }

    service_discovery {
      dns {
        hostname = "serviceb.simpleapp.local"
      }
    }
  }
}
```

» Logging

```
resource "aws_appmesh_virtual_node" "serviceb1" {
  name      = "serviceBv1"
  mesh_name = "${aws_appmesh_mesh.simple.id}"

  spec {
    backend {
      virtual_service {
        virtual_service_name = "servicea.simpleapp.local"
      }
    }

    listener {
      port_mapping {
        port     = 8080
        protocol = "http"
      }
    }

    service_discovery {
      dns {
        hostname = "serviceb.simpleapp.local"
      }
    }

    logging {
      access_log {
        file {
          path = "/dev/stdout"
        }
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name to use for the virtual node.
- `mesh_name` - (Required) The name of the service mesh in which to create the virtual node.
- `spec` - (Required) The virtual node specification to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `spec` object supports the following:

- `backend` - (Optional) The backends to which the virtual node is expected to send outbound traffic.
- `listener` - (Optional) The listeners from which the virtual node is expected to receive inbound traffic.
- `logging` - (Optional) The inbound and outbound access logging information for the virtual node.
- `service_discovery` - (Optional) The service discovery information for the virtual node.

The `backend` object supports the following:

- `virtual_service` - (Optional) Specifies a virtual service to use as a backend for a virtual node.

The `virtual_service` object supports the following:

- `virtual_service_name` - (Required) The name of the virtual service that is acting as a virtual node backend.

The `listener` object supports the following:

- `port_mapping` - (Required) The port mapping information for the listener.
- `health_check` - (Optional) The health check information for the listener.

The `logging` object supports the following:

- `access_log` - (Optional) The access log configuration for a virtual node.

The `access_log` object supports the following:

- `file` - (Optional) The file object to send virtual node access logs to.

The `file` object supports the following:

- `path` - (Required) The file path to write access logs to. You can use `/dev/stdout` to send access logs to standard out.

The `service_discovery` object supports the following:

- `aws_cloud_map` - (Optional) Specifies any AWS Cloud Map information for the virtual node.
- `dns` - (Optional) Specifies the DNS service name for the virtual node.

The `aws_cloud_map` object supports the following:

- `attributes` - (Optional) A string map that contains attributes with values that you can use to filter instances by any custom attribute that you specified when you registered the instance. Only instances that match all of the specified key/value pairs will be returned.
- `namespace_name` - (Required) The name of the AWS Cloud Map namespace to use. Use the `aws_service_discovery_http_namespace` resource to configure a Cloud Map namespace.
- `service_name` - (Required) The name of the AWS Cloud Map service to use. Use the `aws_service_discovery_service` resource to configure a Cloud Map service.

The `dns` object supports the following:

- `hostname` - (Required) The DNS host name for your virtual node.

The `port_mapping` object supports the following:

- `port` - (Required) The port used for the port mapping.
- `protocol` - (Required) The protocol used for the port mapping. Valid values are `http` and `tcp`.

The `health_check` object supports the following:

- `healthy_threshold` - (Required) The number of consecutive successful health checks that must occur before declaring listener healthy.
- `interval_millis`- (Required) The time period in milliseconds between each health check execution.
- `protocol` - (Required) The protocol for the health check request. Valid values are `http` and `tcp`.
- `timeout_millis` - (Required) The amount of time to wait when receiving a response from the health check, in milliseconds.
- `unhealthy_threshold` - (Required) The number of consecutive failed health checks that must occur before declaring a virtual node unhealthy.
- `path` - (Optional) The destination path for the health check request. This is only required if the specified protocol is `http`.
- `port` - (Optional) The destination port for the health check request. This port must match the port defined in the `port_mapping` for the listener.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual node.
- `arn` - The ARN of the virtual node.
- `created_date` - The creation date of the virtual node.
- `last_updated_date` - The last update date of the virtual node.

» Import

App Mesh virtual nodes can be imported using `mesh_name` together with the virtual node's `name`, e.g.

```
$ terraform import aws_appmesh_virtual_node.serviceb1 simpleapp/serviceBv1
```

» Resource: aws_appmesh_virtual_router

Provides an AWS App Mesh virtual router resource.

» Breaking Changes

Because of backward incompatible API changes (read here and here), `aws_appmesh_virtual_router` resource definitions created with provider versions earlier than v2.3.0 will need to be modified:

- Remove service `service_names` from the `spec` argument. AWS has created a `aws_appmesh_virtual_service` resource for each of service names. These resource can be imported using `terraform import`.
- Add a `listener` configuration block to the `spec` argument.

The Terraform state associated with existing resources will automatically be migrated.

» Example Usage

```
resource "aws_appmesh_virtual_router" "serviceb" {
  name      = "serviceB"
  mesh_name = "${aws_appmesh_mesh.simple.id}"

  spec {
    listener {
      port_mapping {
        port     = 8080
        protocol = "http"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name to use for the virtual router.
- `mesh_name` - (Required) The name of the service mesh in which to create the virtual router.
- `spec` - (Required) The virtual router specification to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `spec` object supports the following:

- `listener` - (Required) The listeners that the virtual router is expected to receive inbound traffic from. Currently only one listener is supported per virtual router.

The `listener` object supports the following:

- `port_mapping` - (Required) The port mapping information for the listener.

The `port_mapping` object supports the following:

- `port` - (Required) The port used for the port mapping.

- `protocol` - (Required) The protocol used for the port mapping. Valid values are `http` and `tcp`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual router.
- `arn` - The ARN of the virtual router.
- `created_date` - The creation date of the virtual router.
- `last_updated_date` - The last update date of the virtual router.

» Import

App Mesh virtual routers can be imported using `mesh_name` together with the virtual router's `name`, e.g.

```
$ terraform import aws_appmesh_virtual_router.serviceb simpleapp/serviceB
```

» Resource: aws_appmesh_virtual_service

Provides an AWS App Mesh virtual service resource.

» Example Usage

» Virtual Node Provider

```
resource "aws_appmesh_virtual_service" "servicea" {
  name      = "servicea.simpleapp.local"
  mesh_name = "${aws_appmesh_mesh.simple.id}"

  spec {
    provider {
      virtual_node {
        virtual_node_name = "${aws_appmesh_virtual_node.serviceb1.name}"
      }
    }
  }
}
```

» Virtual Router Provider

```
resource "aws_appmesh_virtual_service" "servicea" {
  name      = "servicea.simpleapp.local"
  mesh_name = "${aws_appmesh_mesh.simple.id}"

  spec {
    provider {
      virtual_router {
        virtual_router_name = "${aws_appmesh_virtual_router.serviceb.name}"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name to use for the virtual service.
- `mesh_name` - (Required) The name of the service mesh in which to create the virtual service.
- `spec` - (Required) The virtual service specification to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `spec` object supports the following:

- `provider`- (Optional) The App Mesh object that is acting as the provider for a virtual service. You can specify a single virtual node or virtual router.

The `provider` object supports the following:

- `virtual_node` - (Optional) The virtual node associated with a virtual service.
- `virtual_router` - (Optional) The virtual router associated with a virtual service.

The `virtual_node` object supports the following:

- `virtual_node_name` - (Required) The name of the virtual node that is acting as a service provider.

The `virtual_router` object supports the following:

- `virtual_router_name` - (Required) The name of the virtual router that is acting as a service provider.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual service.
- `arn` - The ARN of the virtual service.
- `created_date` - The creation date of the virtual service.
- `last_updated_date` - The last update date of the virtual service.

» Import

App Mesh virtual services can be imported using `mesh_name` together with the virtual service's `name`, e.g.

```
$ terraform import aws_appmesh_virtual_service.servicea simpleapp/servicea.simpleapp.local
```

» Resource: aws_appsync_datasource

Provides an AppSync DataSource.

» Example Usage

```
resource "aws_dynamodb_table" "example" {
  name           = "example"
  read_capacity  = 1
  write_capacity = 1
  hash_key       = "UserId"

  attribute {
    name = "UserId"
    type = "S"
  }
}

resource "aws_iam_role" "example" {
  name = "example"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "appsync.amazonaws.com"
      },
      "Effect": "Allow"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "example" {
  name = "example"
  role = "${aws_iam_role.example.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "${aws_dynamodb_table.example.arn}"
      ]
    }
  ]
}
EOF
}

resource "aws_appsync_graphql_api" "example" {
  authentication_type = "API_KEY"
  name                = "tf_appsync_example"
}

resource "aws_appsync_datasource" "example" {
  api_id           = "${aws_appsync_graphql_api.example.id}"
  name             = "tf_appsync_example"
  service_role_arn = "${aws_iam_role.example.arn}"
  type             = "AMAZON_DYNAMODB"

  dynamodb_config {
    table_name = "${aws_dynamodb_table.example.name}"
  }
```

```
}
```

» Argument Reference

The following arguments are supported:

- `api_id` - (Required) The API ID for the GraphQL API for the DataSource.
- `name` - (Required) A user-supplied name for the DataSource.
- `type` - (Required) The type of the DataSource. Valid values: `AWS_LAMBDA`, `AMAZON_DYNAMODB`, `AMAZON_ELASTICSEARCH`, `HTTP`, `NONE`.
- `description` - (Optional) A description of the DataSource.
- `service_role_arn` - (Optional) The IAM service role ARN for the data source.
- `dynamodb_config` - (Optional) DynamoDB settings. See below
- `elasticsearch_config` - (Optional) Amazon Elasticsearch settings. See below
- `http_config` - (Optional) HTTP settings. See below
- `lambda_config` - (Optional) AWS Lambda settings. See below

» dynamodb_config

The following arguments are supported:

- `table_name` - (Required) Name of the DynamoDB table.
- `region` - (Optional) AWS region of the DynamoDB table. Defaults to current region.
- `use_caller_credentials` - (Optional) Set to `true` to use Amazon Cognito credentials with this data source.

» elasticsearch_config

The following arguments are supported:

- `endpoint` - (Required) HTTP endpoint of the Elasticsearch domain.
- `region` - (Optional) AWS region of Elasticsearch domain. Defaults to current region.

» http_config

The following arguments are supported:

- `endpoint` - (Required) HTTP URL.

» lambda_config

The following arguments are supported:

- `function_arn` - (Required) The ARN for the Lambda function.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN

» Import

`aws_appsync_datasource` can be imported with their `api_id`, a hyphen, and `name`, e.g.

```
$ terraform import aws_appsync_datasource.example abcdef123456-example
```

» Resource: aws_appsync_function

Provides an AppSync Function.

» Example Usage

```
resource "aws_appsync_graphql_api" "test" {
  authentication_type = "API_KEY"
  name                = "tf-example"
  schema              = <<EOF
type Mutation {
    putPost(id: ID!, title: String!): Post
}

type Post {
    id: ID!
    title: String!
}

type Query {
    singlePost(id: ID!): Post
}

schema {
    query: Query
    mutation: Mutation
}
EOF
}

resource "aws_appsync_datasource" "test" {
  api_id      = "${aws_appsync_graphql_api.test.id}"
  name        = "tf-example"
  type        = "HTTP"

  http_config {
    endpoint = "http://example.com"
  }
}

resource "aws_appsync_function" "test" {
  api_id      = "${aws_appsync_graphql_api.test.id}"
  data_source = "${aws_appsync_datasource.test.name}"
  name        = "tf_example"
  request_mapping_template = <<EOF
{
    "version": "2018-05-29",
    "method": "GET",
    "resourcePath": "/",
    "params":{
        "headers": $utils.http.copyheaders($ctx.request.headers)
    }
}
EOF
  response_mapping_template = <<EOF
#if($ctx.result.statusCode == 200)
    $ctx.result.body
#else
    $utils.appendError($ctx.result.body, $ctx.result.statusCode)
#end
EOF
}
```

» Argument Reference

The following arguments are supported:

- `api_id` - (Required) The ID of the associated AppSync API.
- `data_source` - (Required) The Function DataSource name.

- `name` - (Required) The Function name. The function name does not have to be unique.
- `request_mapping_template` - (Required) The Function request mapping template. Functions support only the 2018-05-29 version of the request mapping template.
- `response_mapping_template` - (Required) The Function response mapping template.
- `description` - (Optional) The Function description.
- `function_version` - (Optional) The version of the request mapping template. Currently the supported value is `2018-05-29`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - API Function ID (Formatted as ApiId-FunctionId)
- `arn` - The ARN of the Function object.
- `function_id` - A unique ID representing the Function object.

» Import

`aws_appsync_function` can be imported using the AppSync API ID and Function ID separated by `-`, e.g.

```
$ terraform import aws_appsync_function.example xxxxx-yyyyy
```

» Resource: aws_appsync_graphql_api

Provides an AppSync GraphQL API.

» Example Usage

» API Key Authentication

```
resource "aws_appsync_graphql_api" "example" {
  authentication_type = "API_KEY"
  name                = "example"
}
```

» AWS Cognito User Pool Authentication

```
resource "aws_appsync_graphql_api" "example" {
  authentication_type = "AMAZON_COGNITO_USER_POOLS"
  name                = "example"

  user_pool_config {
    aws_region     = "${data.aws_region.current.name}"
    default_action = "DENY"
    user_pool_id   = "${aws_cognito_user_pool.example.id}"
  }
}
```

» AWS IAM Authentication

```
resource "aws_appsync_graphql_api" "example" {
  authentication_type = "AWS_IAM"
  name                = "example"
}
```

» With Schema

```
resource "aws_appsync_graphql_api" "example" {
  authentication_type = "AWS_IAM"
  name                = "example"

  schema = <<EOF
schema {
    query: Query
}
type Query {
  test: Int
}
EOF
}
```

» OpenID Connect Authentication

```
resource "aws_appsync_graphql_api" "example" {
  authentication_type = "OPENID_CONNECT"
  name                = "example"

  openid_connect_config {
    issuer = "https://example.com"
  }
}
```

» With Multiple Authentication Providers

```
resource "aws_appsync_graphql_api" "example" {
  authentication_type = "API_KEY"
  name = "example"

  additional_authentication_provider {
    authentication_type = "AWS_IAM"
  }
}
```

» Enabling Logging

```
resource "aws_iam_role" "example" {
  name = "example"

  assume_role_policy = <<POLICY
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal": {
            "Service": "appsync.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}
POLICY
}

resource "aws_iam_role_policy_attachment" "example" {
  policy_arn = "arn:aws:iam::aws:policy/service-role/AWSAppSyncPushToCloudWatchLogs"
  role       = "${aws_iam_role.example.name}"
}

resource "aws_appsync_graphql_api" "example" {
  # ... other configuration ...
  log_config {
    cloudwatch_logs_role_arn = "${aws_iam_role.example.arn}"
    field_log_level          = "ERROR"
  }
}
```

» Argument Reference

The following arguments are supported:

- `authentication_type` - (Required) The authentication type. Valid values: `API_KEY`, `AWS_IAM`, `AMAZON_COGNITO_USER_POOLS`, `OPENID_CONNECT`
- `name` - (Required) A user-supplied name for the GraphqlApi.
- `log_config` - (Optional) Nested argument containing logging configuration. Defined below.
- `openid_connect_config` - (Optional) Nested argument containing OpenID Connect configuration. Defined below.
- `user_pool_config` - (Optional) The Amazon Cognito User Pool configuration. Defined below.
- `schema` - (Optional) The schema definition, in GraphQL schema language format. Terraform cannot perform drift detection of this configuration.
- `additional_authentication_provider` - (Optional) One or more additional authentication providers for the GraphqlApi. Defined below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» log__config

The following arguments are supported:

- `cloudwatch_logs_role_arn` - (Required) Amazon Resource Name of the service role that AWS AppSync will assume to publish to Amazon CloudWatch logs in your account.
- `field_log_level` - (Required) Field logging level. Valid values: `ALL`, `ERROR`, `NONE`.

» additional__authentication__provider

The following arguments are supported:

- `authentication_type` - (Required) The authentication type. Valid values: `API_KEY`, `AWS_IAM`, `AMAZON_COGNITO_USER_POOLS`, `OPENID_CONNECT`
- `openid_connect_config` - (Optional) Nested argument containing OpenID Connect configuration. Defined below.
- `user_pool_config` - (Optional) The Amazon Cognito User Pool configuration. Defined below.

» openid_connect_config

The following arguments are supported:

- `issuer` - (Required) Issuer for the OpenID Connect configuration. The issuer returned by discovery MUST exactly match the value of iss in the ID Token.
- `auth_ttl` - (Optional) Number of milliseconds a token is valid after being authenticated.
- `client_id` - (Optional) Client identifier of the Relying party at the OpenID identity provider. This identifier is typically obtained when the Relying party is registered with the OpenID identity provider. You can specify a regular expression so the AWS AppSync can validate against multiple client identifiers at a time.
- `iat_ttl` - (Optional) Number of milliseconds a token is valid after being issued to a user.

» user_pool_config

The following arguments are supported:

- `default_action` - (Required only if Cognito is used as the default auth provider) The action that you want your GraphQL API to take when a request that uses Amazon Cognito User Pool authentication doesn't match the Amazon Cognito User Pool configuration. Valid: `ALLOW` and `DENY`
- `user_pool_id` - (Required) The user pool ID.
- `app_id_client_regex` - (Optional) A regular expression for validating the incoming Amazon Cognito User Pool app client ID.
- `aws_region` - (Optional) The AWS region in which the user pool was created.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - API ID
- `arn` - The ARN
- `uris` - Map of URIs associated with the API. e.g. `uris["GRAPHQL"] = https://ID.appsync-api.REGION.amazonaws.com/graphql`

» Import

AppSync GraphQL API can be imported using the GraphQL API ID, e.g.

```
$ terraform import aws_appsync_graphql_api.example 0123456789
```

» Resource: aws_appsync_api_key

Provides an AppSync API Key.

» Example Usage

```
resource "aws_appsync_graphql_api" "example" {
  authentication_type = "API_KEY"
  name                = "example"
}

resource "aws_appsync_api_key" "example" {
  api_id  = "${aws_appsync_graphql_api.example.id}"
  expires = "2018-05-03T04:00:00Z"
}
```

» Argument Reference

The following arguments are supported:

- `api_id` - (Required) The ID of the associated AppSync API
- `description` - (Optional) The API key description. Defaults to ”Managed by Terraform”.
- `expires` - (Optional) RFC3339 string representation of the expiry date. Rounded down to nearest hour. By default, it is 7 days from the date of creation.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - API Key ID (Formatted as ApiId:Key)
- `key` - The API key

» Import

`aws_appsync_api_key` can be imported using the AppSync API ID and key separated by `:`, e.g.

```
$ terraform import aws_appsync_api_key.example xxxxx:yyyyy
```

» Resource: aws_appsync_resolver

Provides an AppSync Resolver.

» Example Usage

```
resource "aws_appsync_graphql_api" "test" {
  authentication_type = "API_KEY"
  name                = "tf-example"

  schema = <<EOF
type Mutation {
    putPost(id: ID!, title: String!): Post
}

type Post {
    id: ID!
    title: String!
}

type Query {
    singlePost(id: ID!): Post
}

schema {
    query: Query
    mutation: Mutation
}
EOF
}

resource "aws_appsync_datasource" "test" {
  api_id = "${aws_appsync_graphql_api.test.id}"
  name   = "tf_example"
  type   = "HTTP"

  http_config {
    endpoint = "http://example.com"
  }
}

# UNIT type resolver (default)
resource "aws_appsync_resolver" "test" {
  api_id      = "${aws_appsync_graphql_api.test.id}"
  field       = "singlePost"
  type        = "Query"
  data_source = "${aws_appsync_datasource.test.name}"

  request_template = <<EOF
{
    "version": "2018-05-29",
    "method": "GET",
    "resourcePath": "/",
    "params":{
        "headers": $utils.http.copyheaders($ctx.request.headers)
    }
}
EOF

  response_template = <<EOF
#if($ctx.result.statusCode == 200)
    $ctx.result.body
#else
    $utils.appendError($ctx.result.body, $ctx.result.statusCode)
#end
EOF
}

# PIPELINE type resolver
resource "aws_appsync_resolver" "Mutation_pipelineTest" {
  type = "Mutation"
  api_id = "${aws_appsync_graphql_api.test.id}"
  field = "pipelineTest"
  request_template = "{}"
  response_template = "$util.toJson($ctx.result)"
  kind = "PIPELINE"
  pipeline_config {
    functions = [
      "${aws_appsync_function.test1.function_id}",
      "${aws_appsync_function.test2.function_id}",
      "${aws_appsync_function.test3.function_id}"
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `api_id` - (Required) The API ID for the GraphQL API.
- `type` - (Required) The type name from the schema defined in the GraphQL API.
- `field` - (Required) The field name from the schema defined in the GraphQL API.
- `request_template` - (Required) The request mapping template for UNIT resolver or 'before mapping template' for PIPELINE resolver.
- `response_template` - (Required) The response mapping template for UNIT resolver or 'after mapping template' for PIPELINE resolver.
- `data_source` - (Optional) The DataSource name.
- `kind` - (Optional) The resolver type. Valid values are `UNIT` and `PIPELINE`.
- `pipeline_config` - (Optional) The PipelineConfig. A `pipeline_config` block is documented below.

An `pipeline_config` block supports the following arguments:

- `functions` - (Required) The list of Function ID.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN

» Import

`aws_appsync_resolver` can be imported with their `api_id`, a hyphen, `type`, a hypen and `field` e.g.

```
$ terraform import aws_appsync_resolver.example abcdef123456-exampleType-exampleField
```

» Resource: aws_athena_database

Provides an Athena database.

» Example Usage

```
resource "aws_s3_bucket" "hoge" {
  bucket = "hoge"
}

resource "aws_athena_database" "hoge" {
  name   = "database_name"
  bucket = "${aws_s3_bucket.hoge.bucket}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the database to create.
- `bucket` - (Required) Name of s3 bucket to save the results of the query execution.
- `encryption_configuration` - (Optional) The encryption key block AWS Athena uses to decrypt the data in S3, such as an AWS Key Management Service (AWS KMS) key. An `encryption_configuration` block is documented below.
- `force_destroy` - (Optional, Default: false) A boolean that indicates all tables should be deleted from the database so that the database can be destroyed without error. The tables are *not* recoverable.

An `encryption_configuration` block supports the following arguments:

- `encryption_option` - (Required) The type of key; one of `SSE_S3`, `SSE_KMS`, `CSE_KMS`
- `kms_key` - (Optional) The KMS key ARN or ID; required for key types `SSE_KMS` and `CSE_KMS`.

NOTE: When Athena queries are executed, result files may be created in the specified bucket. Consider using `force_destroy` on the bucket too in order to avoid any problems when destroying the bucket.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The database name

» Resource: aws_athena_named_query

Provides an Athena Named Query resource.

» Example Usage

```
resource "aws_s3_bucket" "hoge" {
  bucket = "tf-test"
}
```

```
resource "aws_kms_key" "test" {
  deletion_window_in_days = 7
  description             = "Athena KMS Key"
}

resource "aws_athena_workgroup" "test" {
  name = "example"

  configuration {
    result_configuration {
      encryption_configuration {
        encryption_option = "SSE_KMS"
        kms_key_arn       = "${aws_kms_key.test.arn}"
      }
    }
  }
}

resource "aws_athena_database" "hoge" {
  name   = "users"
  bucket = "${aws_s3_bucket.hoge.id}"
}

resource "aws_athena_named_query" "foo" {
  name      = "bar"
  workgroup = "${aws_athena_workgroup.test.id}"
  database  = "${aws_athena_database.hoge.name}"
  query     = "SELECT * FROM ${aws_athena_database.hoge.name} limit 10;"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The plain language name for the query. Maximum length of 128.
- `workgroup` - (Optional) The workgroup to which the query belongs. Defaults to `primary`
- `database` - (Required) The database to which the query belongs.
- `query` - (Required) The text of the query itself. In other words, all query statements. Maximum length of 262144.
- `description` - (Optional) A brief explanation of the query. Maximum length of 1024.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The unique ID of the query.

» Import

Athena Named Query can be imported using the query ID, e.g.

```
$ terraform import aws_athena_named_query.example 0123456789
```

» Resource: aws_athena_workgroup

Provides an Athena Workgroup.

» Example Usage

```
resource "aws_athena_workgroup" "example" {
  name = "example"

  configuration {
    enforce_workgroup_configuration    = true
    publish_cloudwatch_metrics_enabled = true

    result_configuration {
      output_location = "s3://{aws_s3_bucket.example.bucket}/output/"

      encryption_configuration {
        encryption_option = "SSE_KMS"
        kms_key_arn       = "${aws_kms_key.example.arn}"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the workgroup.
- `configuration` - (Optional) Configuration block with various settings for the workgroup. Documented below.

- `description` - (Optional) Description of the workgroup.
- `state` - (Optional) State of the workgroup. Valid values are `DISABLED` or `ENABLED`. Defaults to `ENABLED`.
- `tags` - (Optional) Key-value mapping of resource tags for the workgroup.

» configuration Argument Reference

The `configuration` configuration block supports the following arguments:

- `bytes_scanned_cutoff_per_query` - (Optional) Integer for the upper data usage limit (cutoff) for the amount of bytes a single query in a workgroup is allowed to scan. Must be at least `10485760`.
- `enforce_workgroup_configuration` - (Optional) Boolean whether the settings for the workgroup override client-side settings. For more information, see Workgroup Settings Override Client-Side Settings. Defaults to `true`.
- `publish_cloudwatch_metrics_enabled` - (Optional) Boolean whether Amazon CloudWatch metrics are enabled for the workgroup. Defaults to `true`.
- `result_configuration` - (Optional) Configuration block with result settings. Documented below.

» result_configuration Argument Reference

The `result_configuration` configuration block within the `configuration` supports the following arguments:

- `encryption_configuration` - (Optional) Configuration block with encryption settings. Documented below.
- `output_location` - (Optional) The location in Amazon S3 where your query results are stored, such as `s3://path/to/query/bucket/`. For more information, see Queries and Query Result Files.

» encryption_configuration Argument Reference

The `encryption_configuration` configuration block within the `result_configuration` of the `configuration` supports the following arguments:

- `encryption_option` - (Required) Indicates whether Amazon S3 server-side encryption with Amazon S3-managed keys (SSE-S3), server-side encryption with KMS-managed keys (SSE-KMS), or client-side encryption with KMS-managed keys (CSE-KMS) is used. If a query runs in a workgroup and the workgroup overrides client-side settings, then the workgroup's setting for encryption is used. It specifies whether query results must be encrypted, for all queries that run in this workgroup.

- `kms_key_arn` - (Optional) For SSE-KMS and CSE-KMS, this is the KMS key Amazon Resource Name (ARN).

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the workgroup
- `id` - The workgroup name

» Import

Athena Workgroups can be imported using their name, e.g.

```
$ terraform import aws_athena_workgroup.example example
```

» Data Source: aws_autoscaling_group

Use this data source to get information on an existing autoscaling group.

» Example Usage

```
data "aws_autoscaling_group" "foo" {
  name = "foo"
}
```

» Argument Reference

- `name` - Specify the exact name of the desired autoscaling group.

» Attributes Reference

NOTE: Some values are not always set and may not be available for interpolation.

- `arn` - The Amazon Resource Name (ARN) of the Auto Scaling group.
- `name` - The name of the Auto Scaling group.
- `availability_zones` - One or more Availability Zones for the group.
- `default_cool_down` - The amount of time, in seconds, after a scaling activity completes before another scaling activity can start.
- `desired_capacity` - The desired size of the group.

- `health_check_grace_period` - The amount of time, in seconds, that Amazon EC2 Auto Scaling waits before checking the health status of an EC2 instance that has come into service.
- `health_check_type` - The service to use for the health checks. The valid values are EC2 and ELB.
- `launch_configuration` - The name of the associated launch configuration.
- `load_balancers` - One or more load balancers associated with the group.
- `max_size` - The maximum size of the group.
- `min_size` - The minimum size of the group.
- `placement_group` - The name of the placement group into which to launch your instances, if any. For more information, see Placement Groups (http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html) in the Amazon Elastic Compute Cloud User Guide.
- `service_linked_role_arn` - The Amazon Resource Name (ARN) of the service-linked role that the Auto Scaling group uses to call other AWS services on your behalf.
- `status` - The current state of the group when DeleteAutoScalingGroup is in progress.
- `target_group_arns` - The Amazon Resource Names (ARN) of the target groups for your load balancer.
- `termination_policies` - The termination policies for the group.
- `vpc_zone_identifier` - VPC ID for the group.

» Data Source: aws_autoscaling_groups

The Autoscaling Groups data source allows access to the list of AWS ASGs within a specific region. This will allow you to pass a list of AutoScaling Groups to other resources.

» Example Usage

```
data "aws_autoscaling_groups" "groups" {
  filter {
    name   = "key"
    values = ["Team"]
  }

  filter {
    name   = "value"
    values = ["Pets"]
  }
}
```

```
resource "aws_autoscaling_notification" "slack_notifications" {
  group_names = ["${data.aws_autoscaling_groups.groups.names}"]

  notifications = [
    "autoscaling:EC2_INSTANCE_LAUNCH",
    "autoscaling:EC2_INSTANCE_TERMINATE",
    "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
    "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
  ]

  topic_arn = "TOPIC ARN"
}
```

» Argument Reference

- `filter` - (Optional) A filter used to scope the list e.g. by tags. See related docs.
  - `name` - (Required) The name of the filter. The valid values are: `auto-scaling-group`, `key`, `value`, and `propagate-at-launch`.
  - `values` - (Required) The value of the filter.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `names` - A list of the Autoscaling Groups in the current region.
- `arns` - A list of the Autoscaling Groups Arns in the current region.

» Data Source: aws_launch_configuration

Provides information about a Launch Configuration.

» Example Usage

```
data "aws_launch_configuration" "ubuntu" {
  name = "test-launch-config"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the launch configuration.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the launch configuration.
- `arn` - The Amazon Resource Name of the launch configuration.
- `name` - The Name of the launch configuration.
- `image_id` - The EC2 Image ID of the instance.
- `instance_type` - The Instance Type of the instance to launch.
- `iam_instance_profile` - The IAM Instance Profile to associate with launched instances.
- `key_name` - The Key Name that should be used for the instance.
- `security_groups` - A list of associated Security Group IDS.
- `associate_public_ip_address` - Whether a Public IP address is associated with the instance.
- `vpc_classic_link_id` - The ID of a ClassicLink-enabled VPC.
- `vpc_classic_link_security_groups` - The IDs of one or more Security Groups for the specified ClassicLink-enabled VPC.
- `user_data` - The User Data of the instance.
- `enable_monitoring` - Whether Detailed Monitoring is Enabled.
- `ebs_optimized` - Whether the launched EC2 instance will be EBS-optimized.
- `root_block_device` - The Root Block Device of the instance.
- `ebs_block_device` - The EBS Block Devices attached to the instance.
- `ephemeral_block_device` - The Ephemeral volumes on the instance.
- `spot_price` - The Price to use for reserving Spot instances.
- `placement_tenancy` - The Tenancy of the instance.

`root_block_device` is exported with the following attributes:

- `delete_on_termination` - Whether the EBS Volume will be deleted on instance termination.
- `encrypted` - Whether the volume is Encrypted.
- `iops` - The provisioned IOPs of the volume.
- `volume_size` - The Size of the volume.
- `volume_type` - The Type of the volume.

`ebs_block_device` is exported with the following attributes:

- `delete_on_termination` - Whether the EBS Volume will be deleted on instance termination.
- `device_name` - The Name of the device.

- `iops` - The provisioned IOPs of the volume.
- `snapshot_id` - The Snapshot ID of the mount.
- `volume_size` - The Size of the volume.
- `volume_type` - The Type of the volume.
- `encrypted` - Whether the volume is Encrypted.

`ephemeral_block_device` is exported with the following attributes:

- `device_name` - The Name of the device.
- `virtual_name` - The Virtual Name of the device.

» Resource: aws_autoscaling_attachment

Provides an AutoScaling Attachment resource.

NOTE on AutoScaling Groups and ASG Attachments: Terraform currently provides both a standalone ASG Attachment resource (describing an ASG attached to an ELB), and an AutoScaling Group resource with `load_balancers` defined in-line. At this time you cannot use an ASG with in-line load balancers in conjunction with an ASG Attachment resource. Doing so will cause a conflict and will overwrite attachments.

» Example Usage

```
# Create a new load balancer attachment
resource "aws_autoscaling_attachment" "asg_attachment_bar" {
  autoscaling_group_name = "${aws_autoscaling_group.asg.id}"
  elb                    = "${aws_elb.bar.id}"
}
```

```
# Create a new ALB Target Group attachment
resource "aws_autoscaling_attachment" "asg_attachment_bar" {
  autoscaling_group_name = "${aws_autoscaling_group.asg.id}"
  alb_target_group_arn   = "${aws_alb_target_group.test.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `autoscaling_group_name` - (Required) Name of ASG to associate with the ELB.
- `elb` - (Optional) The name of the ELB.
- `alb_target_group_arn` - (Optional) The ARN of an ALB Target Group.

» Resource: aws_autoscaling_group

Provides an AutoScaling Group resource.

Note: You must specify either `launch_configuration`, `launch_template`, or `mixed_instances_policy`.

» Example Usage

```
resource "aws_placement_group" "test" {
  name     = "test"
  strategy = "cluster"
}

resource "aws_autoscaling_group" "bar" {
  name                      = "foobar3-terraform-test"
  max_size                  = 5
  min_size                  = 2
  health_check_grace_period = 300
  health_check_type         = "ELB"
  desired_capacity          = 4
  force_delete              = true
  placement_group           = "${aws_placement_group.test.id}"
  launch_configuration      = "${aws_launch_configuration.foobar.name}"
  vpc_zone_identifier       = ["${aws_subnet.example1.id}", "${aws_subnet.example2.id}"]

  initial_lifecycle_hook {
    name                 = "foobar"
    default_result       = "CONTINUE"
    heartbeat_timeout    = 2000
    lifecycle_transition = "autoscaling:EC2_INSTANCE_LAUNCHING"

    notification_metadata = <<EOF
{
  "foo": "bar"
}
EOF

    notification_target_arn = "arn:aws:sqs:us-east-1:444455556666:queue1*"
    role_arn                = "arn:aws:iam::123456789012:role/S3Access"
  }

  tag {
    key                 = "foo"
    value               = "bar"
    propagate_at_launch = true
  }

  timeouts {
    delete = "15m"
  }

  tag {
    key                 = "lorem"
    value               = "ipsum"
    propagate_at_launch = false
  }
}
```

» With Latest Version Of Launch Template

```
resource "aws_launch_template" "foobar" {
  name_prefix   = "foobar"
  image_id      = "ami-1a2b3c"
  instance_type = "t2.micro"
}

resource "aws_autoscaling_group" "bar" {
  availability_zones = ["us-east-1a"]
  desired_capacity   = 1
  max_size           = 1
  min_size           = 1

  launch_template {
    id      = "${aws_launch_template.foobar.id}"
    version = "$Latest"
  }
}
```

» Mixed Instances Policy

```
resource "aws_launch_template" "example" {
  name_prefix   = "example"
  image_id      = "${data.aws_ami.example.id}"
  instance_type = "c5.large"
}

resource "aws_autoscaling_group" "example" {
  availability_zones = ["us-east-1a"]
  desired_capacity   = 1
  max_size           = 1
  min_size           = 1

  mixed_instances_policy {
    launch_template {
      launch_template_specification {
        launch_template_id = "${aws_launch_template.example.id}"
      }

      override {
        instance_type = "c4.large"
        weighted_capacity = "3"
      }

      override {
        instance_type = "c3.large"
        weighted_capacity = "2"
      }
    }
  }
}
```

» Interpolated tags

```
variable "extra_tags" {
  default = [
    {
      key                 = "Foo"
      value               = "Bar"
      propagate_at_launch = true
    },
    {
      key                 = "Baz"
      value               = "Bam"
      propagate_at_launch = true
    },
  ]
}

resource "aws_autoscaling_group" "bar" {
  name                 = "foobar3-terraform-test"
  max_size             = 5
  min_size             = 2
  launch_configuration = "${aws_launch_configuration.foobar.name}"
  vpc_zone_identifier  = ["${aws_subnet.example1.id}", "${aws_subnet.example2.id}"]

  tags = [
    {
      key                 = "explicit1"
      value               = "value1"
      propagate_at_launch = true
    },
    {
      key                 = "explicit2"
      value               = "value2"
      propagate_at_launch = true
    },
  ]

  tags = ["${concat(
    list(
      map("key", "interpolation1", "value", "value3", "propagate_at_launch", true),
      map("key", "interpolation2", "value", "value4", "propagate_at_launch", true)
    ),
    var.extra_tags)
  }"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the auto scaling group. By default generated by Terraform.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `max_size` - (Required) The maximum size of the auto scale group.
- `min_size` - (Required) The minimum size of the auto scale group. (See also Waiting for Capacity below.)
- `availability_zones` - (Required only for EC2-Classic) A list of one or more availability zones for the group. This parameter should not be specified when using `vpc_zone_identifier`.
- `default_cooldown` - (Optional) The amount of time, in seconds, after a scaling activity completes before another scaling activity can start.
- `launch_configuration` - (Optional) The name of the launch configuration to use.
- `launch_template` - (Optional) Nested argument with Launch template specification to use to launch instances. Defined below.
- `mixed_instances_policy` (Optional) Configuration block containing set-

tings to define launch targets for Auto Scaling groups. Defined below.

- `initial_lifecycle_hook` - (Optional) One or more Lifecycle Hooks to attach to the autoscaling group **before** instances are launched. The syntax is exactly the same as the separate `aws_autoscaling_lifecycle_hook` resource, without the `autoscaling_group_name` attribute. Please note that this will only work when creating a new autoscaling group. For all other use-cases, please use `aws_autoscaling_lifecycle_hook` resource.
- `health_check_grace_period` - (Optional, Default: 300) Time (in seconds) after instance comes into service before checking health.
- `health_check_type` - (Optional) "EC2" or "ELB". Controls how health checking is done.
- `desired_capacity` - (Optional) The number of Amazon EC2 instances that should be running in the group. (See also Waiting for Capacity below.)
- `force_delete` - (Optional) Allows deleting the autoscaling group without waiting for all instances in the pool to terminate. You can force an autoscaling group to delete even if it's in the process of scaling a resource. Normally, Terraform drains all the instances before deleting the group. This bypasses that behavior and potentially leaves resources dangling.
- `load_balancers` (Optional) A list of elastic load balancer names to add to the autoscaling group names. Only valid for classic load balancers. For ALBs, use `target_group_arns` instead.
- `vpc_zone_identifier` (Optional) A list of subnet IDs to launch resources in.
- `target_group_arns` (Optional) A list of `aws_alb_target_group` ARNs, for use with Application or Network Load Balancing.
- `termination_policies` (Optional) A list of policies to decide how the instances in the auto scale group should be terminated. The allowed values are `OldestInstance`, `NewestInstance`, `OldestLaunchConfiguration`, `ClosestToNextInstanceHour`, `OldestLaunchTemplate`, `AllocationStrategy`, `Default`.
- `suspended_processes` - (Optional) A list of processes to suspend for the AutoScaling Group. The allowed values are `Launch`, `Terminate`, `HealthCheck`, `ReplaceUnhealthy`, `AZRebalance`, `AlarmNotification`, `ScheduledActions`, `AddToLoadBalancer`. Note that if you suspend either the `Launch` or `Terminate` process types, it can prevent your autoscaling group from functioning properly.
- `tag` (Optional) A list of tag blocks. Tags documented below.
- `tags` (Optional) A list of tag blocks (maps). Tags documented below.
- `placement_group` (Optional) The name of the placement group into which you'll launch your instances, if any.
- `metrics_granularity` - (Optional) The granularity to associate with the metrics to collect. The only valid value is `1Minute`. Default is `1Minute`.
- `enabled_metrics` - (Optional) A list of metrics to collect. The allowed values are `GroupDesiredCapacity`, `GroupInServiceCapacity`, `GroupPendingCapacity`, `GroupMinSize`, `GroupMaxSize`, `GroupInServiceInstances`,

GroupPendingInstances, GroupStandbyInstances, GroupStandbyCapacity, , GroupTerminatingCapacity, GroupTerminatingInstances, GroupTotalCapacity, GroupTotalInstances.

- wait_for_capacity_timeout (Default: ”10m”) A maximum duration that Terraform should wait for ASG instances to be healthy before timing out. (See also Waiting for Capacity below.) Setting this to ”0” causes Terraform to skip all Capacity Waiting behavior.
- min_elb_capacity - (Optional) Setting this causes Terraform to wait for this number of instances from this autoscaling group to show up healthy in the ELB only on creation. Updates will not wait on ELB instance number changes. (See also Waiting for Capacity below.)
- wait_for_elb_capacity - (Optional) Setting this will cause Terraform to wait for exactly this number of healthy instances from this autoscaling group in all attached load balancers on both create and update operations. (Takes precedence over min_elb_capacity behavior.) (See also Waiting for Capacity below.)
- protect_from_scale_in (Optional) Allows setting instance protection. The autoscaling group will not select instances with this setting for terminination during scale in events.
- service_linked_role_arn (Optional) The ARN of the service-linked role that the ASG will use to call other AWS services
- max_instance_lifetime (Optional) The maximum amount of time, in seconds, that an instance can be in service, values must be either equal to 0 or between 604800 and 31536000 seconds.

» launch_template

NOTE: Either id or name must be specified.

The top-level launch_template block supports the following:

- id - (Optional) The ID of the launch template. Conflicts with name.
- name - (Optional) The name of the launch template. Conflicts with id.
- version - (Optional) Template version. Can be version number, $Latest, or $Default. (Default: $Default).

» mixed_instances_policy

- instances_distribution - (Optional) Nested argument containing settings on how to mix on-demand and Spot instances in the Auto Scaling group. Defined below.
- launch_template - (Required) Nested argument containing launch template settings along with the overrides to specify multiple instance types and weights. Defined below.

» mixed_instances_policy instances_distribution

This configuration block supports the following:

- `on_demand_allocation_strategy` - (Optional) Strategy to use when launching on-demand instances. Valid values: `prioritized`. Default: `prioritized`.
- `on_demand_base_capacity` - (Optional) Absolute minimum amount of desired capacity that must be fulfilled by on-demand instances. Default: `0`.
- `on_demand_percentage_above_base_capacity` - (Optional) Percentage split between on-demand and Spot instances above the base on-demand capacity. Default: `100`.
- `spot_allocation_strategy` - (Optional) How to allocate capacity across the Spot pools. Valid values: `lowest-price`, `capacity-optimized`. Default: `lowest-price`.
- `spot_instance_pools` - (Optional) Number of Spot pools per availability zone to allocate capacity. EC2 Auto Scaling selects the cheapest Spot pools and evenly allocates Spot capacity across the number of Spot pools that you specify. Default: `2`.
- `spot_max_price` - (Optional) Maximum price per unit hour that the user is willing to pay for the Spot instances. Default: an empty string which means the on-demand price.

» mixed_instances_policy launch_template

This configuration block supports the following:

- `launch_template_specification` - (Required) Nested argument defines the Launch Template. Defined below.
- `override` - (Optional) List of nested arguments provides the ability to specify multiple instance types. This will override the same parameter in the launch template. For on-demand instances, Auto Scaling considers the order of preference of instance types to launch based on the order specified in the overrides list. Defined below.

» mixed_instances_policy launch_template launch_template_specification

NOTE: Either `launch_template_id` or `launch_template_name` must be specified.

This configuration block supports the following:

- `launch_template_id` - (Optional) The ID of the launch template. Conflicts with `launch_template_name`.
- `launch_template_name` - (Optional) The name of the launch template. Conflicts with `launch_template_id`.

- `version` - (Optional) Template version. Can be version number, `$Latest`, or `$Default`. (Default: `$Default`).

» mixed_instances_policy launch_template override

This configuration block supports the following:

- `instance_type` - (Optional) Override the instance type in the Launch Template.
- `weighted_capacity` - (Optional) The number of capacity units, which gives the instance type a proportional weight to other instance types.

» tag and tags

The `tag` attribute accepts exactly one tag declaration with the following fields:

- `key` - (Required) Key
- `value` - (Required) Value
- `propagate_at_launch` - (Required) Enables propagation of the tag to Amazon EC2 instances launched via this ASG

To declare multiple tags additional `tag` blocks can be specified. Alternatively the `tags` attributes can be used, which accepts a list of maps containing the above field names as keys and their respective values. This allows the construction of dynamic lists of tags which is not possible using the single `tag` attribute. `tag` and `tags` are mutually exclusive, only one of them can be specified.

NOTE: Other AWS APIs may automatically add special tags to their associated Auto Scaling Group for management purposes, such as ECS Capacity Providers adding the `AmazonECSManaged` tag. To ignore the removal of these automatic tags, see the `ignore_tags` provider configuration or the `ignore_changes` lifecycle argument for Terraform resources.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The autoscaling group id.
- `arn` - The ARN for this AutoScaling Group
- `availability_zones` - The availability zones of the autoscale group.
- `min_size` - The minimum size of the autoscale group
- `max_size` - The maximum size of the autoscale group
- `default_cooldown` - Time between a scaling activity and the succeeding scaling activity.
- `name` - The name of the autoscale group

- `health_check_grace_period` - Time after instance comes into service before checking health.
- `health_check_type` - "EC2" or "ELB". Controls how health checking is done.
- `desired_capacity` -The number of Amazon EC2 instances that should be running in the group.
- `launch_configuration` - The launch configuration of the autoscale group
- `vpc_zone_identifier` (Optional) - The VPC zone identifier
- `load_balancers` (Optional) The load balancer names associated with the autoscaling group.
- `target_group_arns` (Optional) list of Target Group ARNs that apply to this AutoScaling Group

NOTE: When using `ELB` as the `health_check_type`, `health_check_grace_period` is required.

NOTE: Terraform has two types of ways you can add lifecycle hooks - via the `initial_lifecycle_hook` attribute from this resource, or via the separate `aws_autoscaling_lifecycle_hook` resource. `initial_lifecycle_hook` exists here because any lifecycle hooks added with `aws_autoscaling_lifecycle_hook` will not be added until the autoscaling group has been created, and depending on your capacity settings, after the initial instances have been launched, creating unintended behavior. If you need hooks to run on all instances, add them with `initial_lifecycle_hook` here, but take care to not duplicate these hooks in `aws_autoscaling_lifecycle_hook`.

» Timeouts

`autoscaling_group` provides the following Timeouts configuration options:

- `delete` - (Default `10 minutes`) Used for destroying ASG.

» Waiting for Capacity

A newly-created ASG is initially empty and begins to scale to `min_size` (or `desired_capacity`, if specified) by launching instances using the provided Launch Configuration. These instances take time to launch and boot.

On ASG Update, changes to these values also take time to result in the target number of instances providing service.

Terraform provides two mechanisms to help consistently manage ASG scale up time across dependent resources.

» Waiting for ASG Capacity

The first is default behavior. Terraform waits after ASG creation for `min_size` (or `desired_capacity`, if specified) healthy instances to show up in the ASG before continuing.

If `min_size` or `desired_capacity` are changed in a subsequent update, Terraform will also wait for the correct number of healthy instances before continuing.

Terraform considers an instance ”healthy” when the ASG reports `HealthStatus: "Healthy"` and `LifecycleState: "InService"`. See the AWS AutoScaling Docs for more information on an ASG's lifecycle.

Terraform will wait for healthy instances for up to `wait_for_capacity_timeout`. If ASG creation is taking more than a few minutes, it's worth investigating for scaling activity errors, which can be caused by problems with the selected Launch Configuration.

Setting `wait_for_capacity_timeout` to `"0"` disables ASG Capacity waiting.

» Waiting for ELB Capacity

The second mechanism is optional, and affects ASGs with attached ELBs specified via the `load_balancers` attribute or with ALBs specified with `target_group_arns`.

The `min_elb_capacity` parameter causes Terraform to wait for at least the requested number of instances to show up `"InService"` in all attached ELBs during ASG creation. It has no effect on ASG updates.

If `wait_for_elb_capacity` is set, Terraform will wait for exactly that number of Instances to be `"InService"` in all attached ELBs on both creation and updates.

These parameters can be used to ensure that service is being provided before Terraform moves on. If new instances don't pass the ELB's health checks for any reason, the Terraform apply will time out, and the ASG will be marked as tainted (i.e. marked to be destroyed in a follow up run).

As with ASG Capacity, Terraform will wait for up to `wait_for_capacity_timeout` for the proper number of instances to be healthy.

» Troubleshooting Capacity Waiting Timeouts

If ASG creation takes more than a few minutes, this could indicate one of a number of configuration problems. See the AWS Docs on Load Balancer Troubleshooting for more information.

» Import

AutoScaling Groups can be imported using the `name`, e.g.

```
$ terraform import aws_autoscaling_group.web web-asg
```

» Resource: aws_autoscaling_lifecycle_hook

Provides an AutoScaling Lifecycle Hook resource.

NOTE: Terraform has two types of ways you can add lifecycle hooks - via the `initial_lifecycle_hook` attribute from the `aws_autoscaling_group` resource, or via this one. Hooks added via this resource will not be added until the autoscaling group has been created, and depending on your capacity settings, after the initial instances have been launched, creating unintended behavior. If you need hooks to run on all instances, add them with `initial_lifecycle_hook` in `aws_autoscaling_group`, but take care to not duplicate those hooks with this resource.

» Example Usage

```
resource "aws_autoscaling_group" "foobar" {
  availability_zones   = ["us-west-2a"]
  name                 = "terraform-test-foobar5"
  health_check_type    = "EC2"
  termination_policies = ["OldestInstance"]

  tag {
    key                 = "Foo"
    value               = "foo-bar"
    propagate_at_launch = true
  }
}

resource "aws_autoscaling_lifecycle_hook" "foobar" {
  name                   = "foobar"
  autoscaling_group_name = "${aws_autoscaling_group.foobar.name}"
  default_result         = "CONTINUE"
  heartbeat_timeout      = 2000
  lifecycle_transition   = "autoscaling:EC2_INSTANCE_LAUNCHING"

  notification_metadata = <<EOF
{
  "foo": "bar"
}
EOF

  notification_target_arn = "arn:aws:sqs:us-east-1:444455556666:queue1*"
  role_arn                = "arn:aws:iam::123456789012:role/S3Access"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the lifecycle hook.
- `autoscaling_group_name` - (Required) The name of the Auto Scaling group to which you want to assign the lifecycle hook
- `default_result` - (Optional) Defines the action the Auto Scaling group should take when the lifecycle hook timeout elapses or if an unexpected failure occurs. The value for this parameter can be either CONTINUE or ABANDON. The default value for this parameter is ABANDON.
- `heartbeat_timeout` - (Optional) Defines the amount of time, in seconds, that can elapse before the lifecycle hook times out. When the lifecycle hook times out, Auto Scaling performs the action defined in the DefaultResult parameter
- `lifecycle_transition` - (Required) The instance state to which you want to attach the lifecycle hook. For a list of lifecycle hook types, see describe-lifecycle-hook-types
- `notification_metadata` - (Optional) Contains additional information that you want to include any time Auto Scaling sends a message to the notification target.
- `notification_target_arn` - (Optional) The ARN of the notification target that Auto Scaling will use to notify you when an instance is in the transition state for the lifecycle hook. This ARN target can be either an SQS queue or an SNS topic.
- `role_arn` - (Optional) The ARN of the IAM role that allows the Auto Scaling group to publish to the specified notification target.

» Import

AutoScaling Lifecycle Hooks can be imported using the role autoscaling_group_name and name separated by `/`.

```
$ terraform import aws_aws_autoscaling_lifecycle_hook.test-lifecycle-hook asg-name/lifecycle
```

» Resource: aws_autoscaling_notification

Provides an AutoScaling Group with Notification support, via SNS Topics. Each of the `notifications` map to a Notification Configuration inside Amazon Web Services, and are applied to each AutoScaling Group you supply.

» Example Usage

Basic usage:

```
resource "aws_autoscaling_notification" "example_notifications" {
  group_names = [
    "${aws_autoscaling_group.bar.name}",
    "${aws_autoscaling_group.foo.name}",
  ]

  notifications = [
    "autoscaling:EC2_INSTANCE_LAUNCH",
    "autoscaling:EC2_INSTANCE_TERMINATE",
    "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
    "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
  ]

  topic_arn = "${aws_sns_topic.example.arn}"
}

resource "aws_sns_topic" "example" {
  name = "example-topic"

  # arn is an exported attribute
}

resource "aws_autoscaling_group" "bar" {
  name = "foobar1-terraform-test"

  # ...
}

resource "aws_autoscaling_group" "foo" {
  name = "barfoo-terraform-test"

  # ...
}
```

» Argument Reference

The following arguments are supported:

- `group_names` - (Required) A list of AutoScaling Group Names
- `notifications` - (Required) A list of Notification Types that trigger notifications. Acceptable values are documented in the AWS documentation here
- `topic_arn` - (Required) The Topic ARN for notifications to be sent through

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `group_names`
- `notifications`
- `topic_arn`

» Resource: aws_autoscaling_policy

Provides an AutoScaling Scaling Policy resource.

NOTE: You may want to omit `desired_capacity` attribute from attached `aws_autoscaling_group` when using autoscaling policies. It's good practice to pick either manual or dynamic (policy-based) scaling.

» Example Usage

```
resource "aws_autoscaling_policy" "bat" {
  name                   = "foobar3-terraform-test"
  scaling_adjustment     = 4
  adjustment_type        = "ChangeInCapacity"
  cooldown               = 300
  autoscaling_group_name = "${aws_autoscaling_group.bar.name}"
}

resource "aws_autoscaling_group" "bar" {
  availability_zones        = ["us-east-1a"]
  name                      = "foobar3-terraform-test"
  max_size                  = 5
  min_size                  = 2
  health_check_grace_period = 300
  health_check_type         = "ELB"
  force_delete              = true
  launch_configuration      = "${aws_launch_configuration.foo.name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the policy.
- `autoscaling_group_name` - (Required) The name of the autoscaling group.
- `adjustment_type` - (Optional) Specifies whether the adjustment is an absolute number or a percentage of the current capacity. Valid values are `ChangeInCapacity`, `ExactCapacity`, and `PercentChangeInCapacity`.
- `policy_type` - (Optional) The policy type, either "SimpleScaling", "StepScaling" or "TargetTrackingScaling". If this value isn't provided, AWS will default to "SimpleScaling."
- `estimated_instance_warmup` - (Optional) The estimated time, in seconds, until a newly launched instance will contribute CloudWatch metrics. Without a value, AWS will default to the group's specified cooldown period.

The following arguments are only available to "SimpleScaling" type policies:

- `cooldown` - (Optional) The amount of time, in seconds, after a scaling activity completes and before the next scaling activity can start.
- `scaling_adjustment` - (Optional) The number of instances by which to scale. `adjustment_type` determines the interpretation of this number (e.g., as an absolute number or as a percentage of the existing Auto Scaling group size). A positive increment adds to the current capacity and a negative value removes from the current capacity.

The following arguments are only available to "StepScaling" type policies:

- `metric_aggregation_type` - (Optional) The aggregation type for the policy's metrics. Valid values are "Minimum", "Maximum", and "Average". Without a value, AWS will treat the aggregation type as "Average".
- `step_adjustment` - (Optional) A set of adjustments that manage group scaling. These have the following structure:

```
step_adjustment {
  scaling_adjustment          = -1
  metric_interval_lower_bound = 1.0
  metric_interval_upper_bound = 2.0
}

step_adjustment {
  scaling_adjustment          = 1
  metric_interval_lower_bound = 2.0
  metric_interval_upper_bound = 3.0
}
```

The following fields are available in step adjustments:

- `scaling_adjustment` - (Required) The number of members by which to scale, when the adjustment bounds are breached. A positive value scales up. A negative value scales down.
- `metric_interval_lower_bound` - (Optional) The lower bound for the difference between the alarm threshold and the CloudWatch metric. Without a value, AWS will treat this bound as infinity.
- `metric_interval_upper_bound` - (Optional) The upper bound for the difference between the alarm threshold and the CloudWatch metric. Without a value, AWS will treat this bound as infinity. The upper bound must be greater than the lower bound.

The following arguments are only available to ”TargetTrackingScaling” type policies:

- `target_tracking_configuration` - (Optional) A target tracking policy. These have the following structure:

```
target_tracking_configuration {
  predefined_metric_specification {
    predefined_metric_type = "ASGAverageCPUUtilization"
  }

  target_value = 40.0
}

target_tracking_configuration {
  customized_metric_specification {
    metric_dimension {
      name  = "fuga"
      value = "fuga"
    }

    metric_name = "hoge"
    namespace   = "hoge"
    statistic   = "Average"
  }

  target_value = 40.0
}
```

The following fields are available in target tracking configuration:

- `predefined_metric_specification` - (Optional) A predefined metric.

Conflicts with `customized_metric_specification`.
- `customized_metric_specification` - (Optional) A customized metric. Conflicts with `predefined_metric_specification`.
- `target_value` - (Required) The target value for the metric.
- `disable_scale_in` - (Optional, Default: false) Indicates whether scale in by the target tracking policy is disabled.

» predefined_metric_specification

The following arguments are supported:

- `predefined_metric_type` - (Required) The metric type.
- `resource_label` - (Optional) Identifies the resource associated with the metric type.

» customized_metric_specification

The following arguments are supported:

- `metric_dimension` - (Optional) The dimensions of the metric.
- `metric_name` - (Required) The name of the metric.
- `namespace` - (Required) The namespace of the metric.
- `statistic` - (Required) The statistic of the metric.
- `unit` - (Optional) The unit of the metric.

» metric_dimension

The following arguments are supported:

- `name` - (Required) The name of the dimension.
- `value` - (Required) The value of the dimension.

» Attribute Reference

- `arn` - The ARN assigned by AWS to the scaling policy.
- `name` - The scaling policy's name.
- `autoscaling_group_name` - The scaling policy's assigned autoscaling group.
- `adjustment_type` - The scaling policy's adjustment type.
- `policy_type` - The scaling policy's type.

» Import

AutoScaling scaling policy can be imported using the role autoscaling_group_name and name separated by /.

```
$ terraform import aws_autoscaling_policy.test-policy asg-name/policy-name
```

» Resource: aws_autoscaling_schedule

Provides an AutoScaling Schedule resource.

» Example Usage

```
resource "aws_autoscaling_group" "foobar" {
  availability_zones        = ["us-west-2a"]
  name                      = "terraform-test-foobar5"
  max_size                  = 1
  min_size                  = 1
  health_check_grace_period = 300
  health_check_type         = "ELB"
  force_delete              = true
  termination_policies      = ["OldestInstance"]
}

resource "aws_autoscaling_schedule" "foobar" {
  scheduled_action_name  = "foobar"
  min_size               = 0
  max_size               = 1
  desired_capacity       = 0
  start_time             = "2016-12-11T18:00:00Z"
  end_time               = "2016-12-12T06:00:00Z"
  autoscaling_group_name = "${aws_autoscaling_group.foobar.name}"
}
```

» Argument Reference

The following arguments are supported:

- `autoscaling_group_name` - (Required) The name or Amazon Resource Name (ARN) of the Auto Scaling group.
- `scheduled_action_name` - (Required) The name of this scaling action.

- `start_time` - (Optional) The time for this action to start, in "YYYY-MM-DDThh:mm:ssZ" format in UTC/GMT only (for example, 2014-06-01T00:00:00Z). If you try to schedule your action in the past, Auto Scaling returns an error message.
- `end_time` - (Optional) The time for this action to end, in "YYYY-MM-DDThh:mm:ssZ" format in UTC/GMT only (for example, 2014-06-01T00:00:00Z). If you try to schedule your action in the past, Auto Scaling returns an error message.
- `recurrence` - (Optional) The time when recurring future actions will start. Start time is specified by the user following the Unix cron syntax format.
- `min_size` - (Optional) The minimum size for the Auto Scaling group. Default 0. Set to -1 if you don't want to change the minimum size at the scheduled time.
- `max_size` - (Optional) The maximum size for the Auto Scaling group. Default 0. Set to -1 if you don't want to change the maximum size at the scheduled time.
- `desired_capacity` - (Optional) The number of EC2 instances that should be running in the group. Default 0. Set to -1 if you don't want to change the desired capacity at the scheduled time.

NOTE: When `start_time` and `end_time` are specified with `recurrence` , they form the boundaries of when the recurring action will start and stop.

» Attribute Reference

- `arn` - The ARN assigned by AWS to the autoscaling schedule.

» Import

AutoScaling ScheduledAction can be imported using the `auto-scaling-group-name` and `scheduled-action-name`, e.g.

```
$ terraform import aws_autoscaling_schedule.resource-name auto-scaling-group-name/scheduled-
```

» Resource: aws_backup_plan

Provides an AWS Backup plan resource.

» Example Usage

```
resource "aws_backup_plan" "example" {
  name = "tf_example_backup_plan"
  rule {
    rule_name         = "tf_example_backup_rule"
    target_vault_name = "${aws_backup_vault.test.name}"
    schedule          = "cron(0 12 * * ? *)"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The display name of a backup plan.
- `rule` - (Required) A rule object that specifies a scheduled task that is used to back up a selection of resources.
- `tags` - (Optional) Metadata that you can assign to help organize the plans you create.

» Rule Arguments

For **rule** the following attributes are supported:

- `rule_name` - (Required) An display name for a backup rule.
- `target_vault_name` (Required) - The name of a logical container where backups are stored.
- `schedule` (Optional) - A CRON expression specifying when AWS Backup initiates a backup job.
- `start_window` (Optional) - The amount of time in minutes before beginning a backup.
- `completion_window` (Optional) - The amount of time AWS Backup attempts a backup before canceling the job and returning an error.
- `lifecycle` (Optional) - The lifecycle defines when a protected resource is transitioned to cold storage and when it expires. Fields documented below.
- `recovery_point_tags` (Optional) - Metadata that you can assign to help organize the resources that you create.

» Lifecycle Arguments

For **lifecycle** the following attributes are supported:

- `cold_storage_after` - (Optional) Specifies the number of days after creation that a recovery point is moved to cold storage.

- `delete_after` (Optional) - Specifies the number of days after creation that a recovery point is deleted. Must be 90 days greater than `cold_storage_after`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the backup plan.
- `arn` - The ARN of the backup plan.
- `version` - Unique, randomly generated, Unicode, UTF-8 encoded string that serves as the version ID of the backup plan.

» Resource: aws_backup_selection

Manages selection conditions for AWS Backup plan resources.

» Example Usage

» IAM Role

For more information about creating and managing IAM Roles for backups and restores, see the AWS Backup Developer Guide.

The below example creates an IAM role with the default managed IAM Policy for allowing AWS Backup to create backups.

```
resource "aws_iam_role" "example" {
  name               = "example"
  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["sts:AssumeRole"],
      "Effect": "allow",
      "Principal": {
        "Service": ["backup.amazonaws.com"]
      }
    }
  ]
}
POLICY
}
```

```
resource "aws_iam_role_policy_attachment" "example" {
  policy_arn = "arn:aws:iam::aws:policy/service-role/AWSBackupServiceRolePolicyForBackup"
  role       = "${aws_iam_role.example.name}"
}

resource "aws_backup_selection" "example" {
  # ... other configuration ...

  iam_role_arn = "${aws_iam_role.example.arn}"
}
```

» Selecting Backups By Tag

```
resource "aws_backup_selection" "example" {
  iam_role_arn = "${aws_iam_role.example.arn}"
  name         = "tf_example_backup_selection"
  plan_id      = "${aws_backup_plan.example.id}"

  selection_tag {
    type  = "STRINGEQUALS"
    key   = "foo"
    value = "bar"
  }
}
```

» Selecting Backups By Resource

```
resource "aws_backup_selection" "example" {
  iam_role_arn = "${aws_iam_role.example.arn}"
  name         = "tf_example_backup_selection"
  plan_id      = "${aws_backup_plan.example.id}"

  resources = [
    "${aws_db_instance.example.arn}",
    "${aws_ebs_volume.example.arn}",
    "${aws_efs_file_system.example.arn}",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The display name of a resource selection document.
- `plan_id` - (Required) The backup plan ID to be associated with the selection of resources.
- `iam_role_arn` - (Required) The ARN of the IAM role that AWS Backup uses to authenticate when restoring and backing up the target resource. See the AWS Backup Developer Guide for additional information about using AWS managed policies or creating custom policies attached to the IAM role.
- `selection_tag` - (Optional) Tag-based conditions used to specify a set of resources to assign to a backup plan.
- `resources` - (Optional) An array of strings that either contain Amazon Resource Names (ARNs) or match patterns of resources to assign to a backup plan..

Tag conditions (`selection_tag`) support the following:

- `type` - (Required) An operation, such as `StringEquals`, that is applied to a key-value pair used to filter resources in a selection.
- `key` - (Required) The key in a key-value pair.
- `value` - (Required) The value in a key-value pair.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Backup Selection identifier

» Import

Backup selection can be imported using the role plan_id and id separated by `|`.

```
$ terraform import aws_backup_selection.example plan-id|selection-id
```

» Resource: aws_backup_vault

Provides an AWS Backup vault resource.

» Example Usage

```
resource "aws_backup_vault" "example" {
  name        = "example_backup_vault"
  kms_key_arn = "${aws_kms_key.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the backup vault to create.
- `tags` - (Optional) Metadata that you can assign to help organize the resources that you create.
- `kms_key_arn` - (Optional) The server-side encryption key that is used to protect your backups.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the vault.
- `arn` - The ARN of the vault.
- `recovery_points` - The number of recovery points that are stored in a backup vault.

» Import

Backup vault can be imported using the `name`, e.g.

```
$ terraform import aws_backup_vault.test-vault TestVault
```

» Data Source: aws_batch_compute_environment

The Batch Compute Environment data source allows access to details of a specific compute environment within AWS Batch.

» Example Usage

```
data "aws_batch_compute_environment" "batch-mongo" {
  compute_environment_name = "batch-mongo-production"
}
```

» Argument Reference

The following arguments are supported:

- `compute_environment_name` - (Required) The name of the Batch Compute Environment

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the compute environment.
- `ecs_cluster_arn` - The ARN of the underlying Amazon ECS cluster used by the compute environment.
- `service_role` - The ARN of the IAM role that allows AWS Batch to make calls to other AWS services on your behalf.
- `type` - The type of the compute environment (for example, `MANAGED` or `UNMANAGED`).
- `status` - The current status of the compute environment (for example, `CREATING` or `VALID`).
- `status_reason` - A short, human-readable string to provide additional details about the current status of the compute environment.
- `state` - The state of the compute environment (for example, `ENABLED` or `DISABLED`). If the state is `ENABLED`, then the compute environment accepts jobs from a queue and can scale out automatically based on queues.

» Data Source: aws_batch_job_queue

The Batch Job Queue data source allows access to details of a specific job queue within AWS Batch.

» Example Usage

```
data "aws_batch_job_queue" "test-queue" {
  name = "tf-test-batch-job-queue"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the job queue.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the job queue.
- `status` - The current status of the job queue (for example, `CREATING` or `VALID`).

- status_reason - A short, human-readable string to provide additional details about the current status of the job queue.
- state - Describes the ability of the queue to accept new jobs (for example, ENABLED or DISABLED).
- priority - The priority of the job queue. Job queues with a higher priority are evaluated first when associated with the same compute environment.
- compute_environment_order - The compute environments that are attached to the job queue and the order in which job placement is preferred. Compute environments are selected for job placement in ascending order.
  - compute_environment_order.#.order - The order of the compute environment.
  - compute_environment_order.#.compute_environment - The ARN of the compute environment.

» Resource: aws_batch_compute_environment

Creates a AWS Batch compute environment. Compute environments contain the Amazon ECS container instances that are used to run containerized batch jobs.

For information about AWS Batch, see What is AWS Batch? . For information about compute environment, see Compute Environments .

Note: To prevent a race condition during environment deletion, make sure to set depends_on to the related aws_iam_role_policy_attachment; otherwise, the policy may be destroyed too soon and the compute environment will then get stuck in the DELETING state, see Troubleshooting AWS Batch .

» Example Usage

```
resource "aws_iam_role" "ecs_instance_role" {
  name = "ecs_instance_role"

  assume_role_policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Action": "sts:AssumeRole",
        "Effect": "Allow",
        "Principal": {
        "Service": "ec2.amazonaws.com"
        }
    }
    ]
}
EOF
}

resource "aws_iam_role_policy_attachment" "ecs_instance_role" {
  role       = "${aws_iam_role.ecs_instance_role.name}"
  policy_arn = "arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role"
}

resource "aws_iam_instance_profile" "ecs_instance_role" {
  name = "ecs_instance_role"
  role = "${aws_iam_role.ecs_instance_role.name}"
}

resource "aws_iam_role" "aws_batch_service_role" {
  name = "aws_batch_service_role"

  assume_role_policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Action": "sts:AssumeRole",
        "Effect": "Allow",
        "Principal": {
        "Service": "batch.amazonaws.com"
        }
    }
    ]
}
EOF
}

resource "aws_iam_role_policy_attachment" "aws_batch_service_role" {
  role       = "${aws_iam_role.aws_batch_service_role.name}"
  policy_arn = "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole"
}

resource "aws_security_group" "sample" {
  name = "aws_batch_compute_environment_security_group"

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_vpc" "sample" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_subnet" "sample" {
  vpc_id     = "${aws_vpc.sample.id}"
  cidr_block = "10.1.1.0/24"
}

resource "aws_batch_compute_environment" "sample" {
  compute_environment_name = "sample"

  compute_resources {
    instance_role = "${aws_iam_instance_profile.ecs_instance_role.arn}"

    instance_type = [
      "c4.large",
    ]

    max_vcpus = 16
    min_vcpus = 0

    security_group_ids = [
      "${aws_security_group.sample.id}",
    ]

    subnets = [
      "${aws_subnet.sample.id}",
    ]

    type = "EC2"
  }

  service_role = "${aws_iam_role.aws_batch_service_role.arn}"
  type         = "MANAGED"
  depends_on   = ["aws_iam_role_policy_attachment.aws_batch_service_role"]
}
```

» Argument Reference

- `compute_environment_name` - (Optional, Forces new resource) The name for your compute environment. Up to 128 letters (uppercase and lowercase), numbers, and underscores are allowed. If omitted, Terraform will assign a random, unique name.
- `compute_environment_name_prefix` - (Optional, Forces new resource) Creates a unique compute environment name beginning with the specified prefix. Conflicts with `compute_environment_name`.
- `compute_resources` - (Optional) Details of the compute resources managed by the compute environment. This parameter is required for managed compute environments. See details below.
- `service_role` - (Required) The full Amazon Resource Name (ARN) of the IAM role that allows AWS Batch to make calls to other AWS services on your behalf.
- `state` - (Optional) The state of the compute environment. If the state is `ENABLED`, then the compute environment accepts jobs from a queue and can scale out automatically based on queues. Valid items are `ENABLED` or `DISABLED`. Defaults to `ENABLED`.
- `type` - (Required) The type of the compute environment. Valid items are `MANAGED` or `UNMANAGED`.

compute__resources is a child block with a single argument:

- `allocation_strategy` - (Optional) The allocation strategy to use for the compute resource in case not enough instances of the best fitting instance type can be allocated. Valid items are `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` or `BEST_FIT`. Defaults to `BEST_FIT`. See AWS docs for details.
- `bid_percentage` - (Optional) Integer of minimum percentage that a Spot Instance price must be when compared with the On-Demand price for that instance type before instances are launched. For example, if your bid percentage is 20% (`20`), then the Spot price must be below 20% of the current On-Demand price for that EC2 instance. This parameter is required for SPOT compute environments.
- `desired_vcpus` - (Optional) The desired number of EC2 vCPUS in the compute environment.
- `ec2_key_pair` - (Optional) The EC2 key pair that is used for instances launched in the compute environment.
- `image_id` - (Optional) The Amazon Machine Image (AMI) ID used for instances launched in the compute environment.
- `instance_role` - (Required) The Amazon ECS instance role applied to Amazon EC2 instances in a compute environment.
- `instance_type` - (Required) A list of instance types that may be launched.
- `launch_template` - (Optional) The launch template to use for your compute resources. See details below.

- `max_vcpus` - (Required) The maximum number of EC2 vCPUs that an environment can reach.
- `min_vcpus` - (Required) The minimum number of EC2 vCPUs that an environment should maintain.
- `security_group_ids` - (Required) A list of EC2 security group that are associated with instances launched in the compute environment.
- `spot_iam_fleet_role` - (Optional) The Amazon Resource Name (ARN) of the Amazon EC2 Spot Fleet IAM role applied to a SPOT compute environment. This parameter is required for SPOT compute environments.
- `subnets` - (Required) A list of VPC subnets into which the compute resources are launched.
- `tags` - (Optional) Key-value pair tags to be applied to resources that are launched in the compute environment.
- `type` - (Required) The type of compute environment. Valid items are `EC2` or `SPOT`.

» launch_template

`launch_template` supports the following:

- `launch_template_id` - (Optional) ID of the launch template. You must specify either the launch template ID or launch template name in the request, but not both.
- `launch_template_name` - (Optional) Name of the launch template.
- `version` - (Optional) The version number of the launch template. Default: The default version of the launch template.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) of the compute environment.
- `ecs_cluster_arn` - The Amazon Resource Name (ARN) of the underlying Amazon ECS cluster used by the compute environment.
- `status` - The current status of the compute environment (for example, CREATING or VALID).
- `status_reason` - A short, human-readable string to provide additional details about the current status of the compute environment.

» Import

AWS Batch compute can be imported using the `compute_environment_name`, e.g.

```
$ terraform import aws_batch_compute_environment.sample sample
```

» Resource: aws_batch_job_definition

Provides a Batch Job Definition resource.

» Example Usage

```
resource "aws_batch_job_definition" "test" {
  name = "tf_test_batch_job_definition"
  type = "container"

  container_properties = <<CONTAINER_PROPERTIES
{
    "command": ["ls", "-la"],
    "image": "busybox",
    "memory": 1024,
    "vcpus": 1,
    "volumes": [
      {
        "host": {
          "sourcePath": "/tmp"
        },
        "name": "tmp"
      }
    ],
    "environment": [
        {"name": "VARNAME", "value": "VARVAL"}
    ],
    "mountPoints": [
        {
          "sourceVolume": "tmp",
          "containerPath": "/tmp",
          "readOnly": false
        }
    ],
    "ulimits": [
      {
        "hardLimit": 1024,
        "name": "nofile",
        "softLimit": 1024
      }
    ]
}
CONTAINER_PROPERTIES
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the job definition.
- `container_properties` - (Optional) A valid container properties provided as a single valid JSON document. This parameter is required if the `type` parameter is `container`.
- `parameters` - (Optional) Specifies the parameter substitution placeholders to set in the job definition.
- `retry_strategy` - (Optional) Specifies the retry strategy to use for failed jobs that are submitted with this job definition. Maximum number of `retry_strategy` is `1`. Defined below.
- `timeout` - (Optional) Specifies the timeout for jobs so that if a job runs longer, AWS Batch terminates the job. Maximum number of `timeout` is `1`. Defined below.
- `type` - (Required) The type of job definition. Must be `container`

» retry_strategy

`retry_strategy` supports the following:

- `attempts` - (Optional) The number of times to move a job to the `RUNNABLE` status. You may specify between `1` and `10` attempts.

» timeout

`timeout` supports the following:

- `attempt_duration_seconds` - (Optional) The time duration in seconds after which AWS Batch terminates your jobs if they have not finished. The minimum value for the timeout is `60` seconds.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name of the job definition.
- `revision` - The revision of the job definition.

» Resource: aws_batch_job_queue

Provides a Batch Job Queue resource.

» Example Usage

```
resource "aws_batch_job_queue" "test_queue" {
  name                 = "tf-test-batch-job-queue"
  state                = "ENABLED"
  priority             = 1
  compute_environments = ["${aws_batch_compute_environment.test_environment_1.arn}", "${aws_
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the job queue.
- `compute_environments` - (Required) Specifies the set of compute environments mapped to a job queue and their order. The position of the compute environments in the list will dictate the order. You can associate up to 3 compute environments with a job queue.
- `priority` - (Required) The priority of the job queue. Job queues with a higher priority are evaluated first when associated with the same compute environment.
- `state` - (Required) The state of the job queue. Must be one of: `ENABLED` or `DISABLED`

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name of the job queue.

» Import

Batch Job Queue can be imported using the `arn`, e.g.

```
$ terraform import aws_batch_job_queue.test_queue arn:aws:batch:us-east-1:123456789012:job-
```

» Resource: aws_budgets_budget

Provides a budgets budget resource. Budgets use the cost visualisation provided by Cost Explorer to show you the status of your budgets, to provide forecasts of your estimated costs, and to track your AWS usage, including your free tier usage.

» Example Usage

```
resource "aws_budgets_budget" "ec2" {
  name              = "budget-ec2-monthly"
  budget_type       = "COST"
  limit_amount      = "1200"
  limit_unit        = "USD"
  time_period_end   = "2087-06-15_00:00"
  time_period_start = "2017-07-01_00:00"
  time_unit         = "MONTHLY"

  cost_filters = {
    Service = "Amazon Elastic Compute Cloud - Compute"
  }

  notification {
    comparison_operator        = "GREATER_THAN"
    threshold                  = 100
    threshold_type             = "PERCENTAGE"
    notification_type          = "FORECASTED"
    subscriber_email_addresses = ["test@example.com"]
  }
}
```

Create a budget for *$100*.

```
resource "aws_budgets_budget" "cost" {
  # ...
  budget_type  = "COST"
  limit_amount = "100"
  limit_unit   = "USD"
}
```

Create a budget for s3 with a limit of *3 GB* of storage.

```
resource "aws_budgets_budget" "s3" {
  # ...
  budget_type  = "USAGE"
  limit_amount = "3"
  limit_unit   = "GB"
}
```

Create a Savings Plan Utilization Budget

```
resource "aws_budgets_budget" "savings_plan_utilization" {
  # ...
  budget_type  = "SAVINGS_PLANS_UTILIZATION"
  limit_amount = "100.0"
  limit_unit   = "PERCENTAGE"

  cost_types {
    include_credit             = false
    include_discount           = false
    include_other_subscription = false
    include_recurring          = false
    include_refund             = false
    include_subscription       = true
    include_support            = false
    include_tax                = false
    include_upfront            = false
    use_blended                = false
  }
}
```

Create a RI Utilization Budget

```
resource "aws_budgets_budget" "ri_utilization" {
  # ...
  budget_type  = "RI_UTILIZATION"
  limit_amount = "100.0" # RI utilization must be 100
  limit_unit   = "PERCENTAGE"

  #Cost types must be defined for RI budgets because the settings conflict with the defaults
  cost_types {
    include_credit             = false
    include_discount           = false
    include_other_subscription = false
    include_recurring          = false
    include_refund             = false
    include_subscription       = true
    include_support            = false
    include_tax                = false
    include_upfront            = false
    use_blended                = false
  }

  # RI Utilization plans require a service cost filter to be set
  cost_filters = {
    Service = "Amazon Relational Database Service"
  }
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- `account_id` - (Optional) The ID of the target account for budget. Will use current user's account_id by default if omitted.
- `name` - (Optional) The name of a budget. Unique within accounts.
- `name_prefix` - (Optional) The prefix of the name of a budget. Unique within accounts.
- `budget_type` - (Required) Whether this budget tracks monetary cost or usage.
- `cost_filters` - (Optional) Map of CostFilters key/value pairs to apply to the budget.
- `cost_types` - (Optional) Object containing CostTypes The types of cost included in a budget, such as tax and subscriptions..
- `limit_amount` - (Required) The amount of cost or usage being measured for a budget.
- `limit_unit` - (Required) The unit of measurement used for the budget forecast, actual spend, or budget threshold, such as dollars or GB. See Spend documentation.
- `time_period_end` - (Optional) The end of the time period covered by the budget. There are no restrictions on the end date. Format: `2017-01-01_12:00`.
- `time_period_start` - (Required) The start of the time period covered by the budget. The start date must come before the end date. Format: `2017-01-01_12:00`.
- `time_unit` - (Required) The length of time until a budget resets the actual and forecasted spend. Valid values: `MONTHLY`, `QUARTERLY`, `ANNUALLY`.
- `notification` - (Optional) Object containing Budget Notifications. Can be used multiple times to define more than one budget notification

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - id of resource.

» CostTypes

Valid keys for `cost_types` parameter.

- `include_credit` - A boolean value whether to include credits in the cost budget. Defaults to `true`

- `include_discount` - Specifies whether a budget includes discounts. Defaults to `true`
- `include_other_subscription` - A boolean value whether to include other subscription costs in the cost budget. Defaults to `true`
- `include_recurring` - A boolean value whether to include recurring costs in the cost budget. Defaults to `true`
- `include_refund` - A boolean value whether to include refunds in the cost budget. Defaults to `true`
- `include_subscription` - A boolean value whether to include subscriptions in the cost budget. Defaults to `true`
- `include_support` - A boolean value whether to include support costs in the cost budget. Defaults to `true`
- `include_tax` - A boolean value whether to include tax in the cost budget. Defaults to `true`
- `include_upfront` - A boolean value whether to include upfront costs in the cost budget. Defaults to `true`
- `use_amortized` - Specifies whether a budget uses the amortized rate. Defaults to `false`
- `use_blended` - A boolean value whether to use blended costs in the cost budget. Defaults to `false`

Refer to AWS CostTypes documentation for further detail.

» CostFilters

Valid keys for `cost_filters` parameter vary depending on the `budget_type` value.

- `cost`
  - `AZ`
  - `LinkedAccount`
  - `Operation`
  - `PurchaseType`
  - `Service`
  - `TagKeyValue`
- `usage`
  - `AZ`
  - `LinkedAccount`
  - `Operation`
  - `PurchaseType`
  - `UsageType:<service name>`
  - `TagKeyValue`

Refer to AWS CostFilter documentation for further detail.

» BudgetNotification

Valid keys for `notification` parameter.

- `comparison_operator` - (Required) Comparison operator to use to evaluate the condition. Can be `LESS_THAN`, `EQUAL_TO` or `GREATER_THAN`.
- `threshold` - (Required) Threshold when the notification should be sent.
- `threshold_type` - (Required) What kind of threshold is defined. Can be `PERCENTAGE` OR `ABSOLUTE_VALUE`.
- `notification_type` - (Required) What kind of budget value to notify on. Can be `ACTUAL` or `FORECASTED`
- `subscriber_email_addresses` - (Optional) E-Mail addresses to notify. Either this or `subscriber_sns_topic_arns` is required.
- `subscriber_sns_topic_arns` - (Optional) SNS topics to notify. Either this or `subscriber_email_addresses` is required.

» Import

Budgets can be imported using `AccountID:BudgetName`, e.g.

```
$ terraform import aws_budgets_budget.myBudget 123456789012:myBudget
```

» Resource: aws_cloud9_environment_ec2

Provides a Cloud9 EC2 Development Environment.

» Example Usage

```
resource "aws_cloud9_environment_ec2" "example" {
  instance_type = "t2.micro"
  name          = "example-env"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the environment.
- `instance_type` - (Required) The type of instance to connect to the environment, e.g. `t2.micro`.
- `automatic_stop_time_minutes` - (Optional) The number of minutes until the running instance is shut down after the environment has last been used.
- `description` - (Optional) The description of the environment.

- `owner_arn` - (Optional) The ARN of the environment owner. This can be ARN of any AWS IAM principal. Defaults to the environment's creator.
- `subnet_id` - (Optional) The ID of the subnet in Amazon VPC that AWS Cloud9 will use to communicate with the Amazon EC2 instance.

» Attributes Reference

In addition the the arguments listed above the following attributes are exported:

- `id` - The ID of the environment.
- `arn` - The ARN of the environment.
- `type` - The type of the environment (e.g. `ssh` or `ec2`)

» Data Source: aws_cloudformation_export

The CloudFormation Export data source allows access to stack exports specified in the Output section of the Cloudformation Template using the optional Export Property.

Note: If you are trying to use a value from a Cloudformation Stack in the same Terraform run please use normal interpolation or Cloudformation Outputs.

» Example Usage

```
data "aws_cloudformation_export" "subnet_id" {
  name = "mySubnetIdExportName"
}

resource "aws_instance" "web" {
  ami           = "ami-abb07bcb"
  instance_type = "t1.micro"
  subnet_id     = "${data.aws_cloudformation_export.subnet_id.value}"
}
```

» Argument Reference

- `name` - (Required) The name of the export as it appears in the console or from list-exports

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `value` - The value from Cloudformation export identified by the export name found from list-exports
- `exporting_stack_id` - The exporting_stack_id (AWS ARNs) equivalent `ExportingStackId` from list-exports

» Data Source: aws_cloudformation_stack

The CloudFormation Stack data source allows access to stack outputs and other useful data including the template body.

» Example Usage

```
data "aws_cloudformation_stack" "network" {
  name = "my-network-stack"
}

resource "aws_instance" "web" {
  ami           = "ami-abb07bcb"
  instance_type = "t1.micro"
  subnet_id     = "${data.aws_cloudformation_stack.network.outputs["SubnetId"]}"

  tags = {
    Name = "HelloWorld"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the stack

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `capabilities` - A list of capabilities
- `description` - Description of the stack
- `disable_rollback` - Whether the rollback of the stack is disabled when stack creation fails
- `notification_arns` - A list of SNS topic ARNs to publish stack related events

- `outputs` - A map of outputs from the stack.
- `parameters` - A map of parameters that specify input parameters for the stack.
- `tags` - A map of tags associated with this stack.
- `template_body` - Structure containing the template body.
- `iam_role_arn` - The ARN of the IAM role used to create the stack.
- `timeout_in_minutes` - The amount of time that can pass before the stack status becomes `CREATE_FAILED`

» Resource: aws_cloudformation_stack

Provides a CloudFormation Stack resource.

» Example Usage

```
resource "aws_cloudformation_stack" "network" {
  name = "networking-stack"

  parameters = {
    VPCCidr = "10.0.0.0/16"
  }

  template_body = <<STACK
{
  "Parameters" : {
    "VPCCidr" : {
      "Type" : "String",
      "Default" : "10.0.0.0/16",
      "Description" : "Enter the CIDR block for the VPC. Default is 10.0.0.0/16."
    }
  },
  "Resources" : {
    "myVpc": {
      "Type" : "AWS::EC2::VPC",
      "Properties" : {
        "CidrBlock" : { "Ref" : "VPCCidr" },
        "Tags" : [
          {"Key": "Name", "Value": "Primary_CF_VPC"}
        ]
      }
    }
  }
}
STACK
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Stack name.
- `template_body` - (Optional) Structure containing the template body (max size: 51,200 bytes).
- `template_url` - (Optional) Location of a file containing the template body (max size: 460,800 bytes).
- `capabilities` - (Optional) A list of capabilities. Valid values: `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, or `CAPABILITY_AUTO_EXPAND`
- `disable_rollback` - (Optional) Set to true to disable rollback of the stack if stack creation failed. Conflicts with `on_failure`.
- `notification_arns` - (Optional) A list of SNS topic ARNs to publish stack related events.
- `on_failure` - (Optional) Action to be taken if stack creation fails. This must be one of: `DO_NOTHING`, `ROLLBACK`, or `DELETE`. Conflicts with `disable_rollback`.
- `parameters` - (Optional) A map of Parameter structures that specify input parameters for the stack.
- `policy_body` - (Optional) Structure containing the stack policy body. Conflicts w/ `policy_url`.
- `policy_url` - (Optional) Location of a file containing the stack policy. Conflicts w/ `policy_body`.
- `tags` - (Optional) A list of tags to associate with this stack.
- `iam_role_arn` - (Optional) The ARN of an IAM role that AWS CloudFormation assumes to create the stack. If you don't specify a value, AWS CloudFormation uses the role that was previously associated with the stack. If no role is available, AWS CloudFormation uses a temporary session that is generated from your user credentials.
- `timeout_in_minutes` - (Optional) The amount of time that can pass before the stack status becomes `CREATE_FAILED`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - A unique identifier of the stack.
- `outputs` - A map of outputs from the stack.

» Import

Cloudformation Stacks can be imported using the `name`, e.g.

```
$ terraform import aws_cloudformation_stack.stack networking-stack
```

» Timeouts

`aws_cloudformation_stack` provides the following Timeouts configuration options:

- `create` - (Default `30 minutes`) Used for Creating Stacks
- `update` - (Default `30 minutes`) Used for Stack modifications
- `delete` - (Default `30 minutes`) Used for destroying stacks.

» Resource: aws_cloudformation_stack_set

Manages a CloudFormation Stack Set. Stack Sets allow CloudFormation templates to be easily deployed across multiple accounts and regions via Stack Set Instances (`aws_cloudformation_stack_set_instance` resource). Additional information about Stack Sets can be found in the AWS CloudFormation User Guide.

NOTE: All template parameters, including those with a `Default`, must be configured or ignored with the `lifecycle` configuration block `ignore_changes` argument.

NOTE: All `NoEcho` template parameters must be ignored with the `lifecycle` configuration block `ignore_changes` argument.

» Example Usage

```
data "aws_iam_policy_document" "AWSCloudFormationStackSetAdministrationRole_assume_role_poli
  statement {
    actions = ["sts:AssumeRole"]
    effect  = "Allow"

    principals {
      identifiers = ["cloudformation.amazonaws.com"]
      type        = "Service"
    }
  }
}
```

```
resource "aws_iam_role" "AWSCloudFormationStackSetAdministrationRole" {
  assume_role_policy = "${data.aws_iam_policy_document.AWSCloudFormationStackSetAdministrati
  name               = "AWSCloudFormationStackSetAdministrationRole"
}

resource "aws_cloudformation_stack_set" "example" {
  administration_role_arn = "${aws_iam_role.AWSCloudFormationStackSetAdministrationRole.arn
  name                    = "example"

  parameters = {
    VPCCidr = "10.0.0.0/16"
  }

  template_body = <<TEMPLATE
{
  "Parameters" : {
    "VPCCidr" : {
      "Type" : "String",
      "Default" : "10.0.0.0/16",
      "Description" : "Enter the CIDR block for the VPC. Default is 10.0.0.0/16."
    }
  },
  "Resources" : {
    "myVpc": {
      "Type" : "AWS::EC2::VPC",
      "Properties" : {
        "CidrBlock" : { "Ref" : "VPCCidr" },
        "Tags" : [
          {"Key": "Name", "Value": "Primary_CF_VPC"}
        ]
      }
    }
  }
}
TEMPLATE
}

data "aws_iam_policy_document" "AWSCloudFormationStackSetAdministrationRole_ExecutionPolicy"
  statement {
    actions   = ["sts:AssumeRole"]
    effect    = "Allow"
    resources = ["arn:aws:iam::*:role/${aws_cloudformation_stack_set.example.execution_role_
  }
}

resource "aws_iam_role_policy" "AWSCloudFormationStackSetAdministrationRole_ExecutionPolicy"
```

```
  name   = "ExecutionPolicy"
  policy = "${data.aws_iam_policy_document.AWSCloudFormationStackSetAdministrationRole_Execu
  role   = "${aws_iam_role.AWSCloudFormationStackSetAdministrationRole.name}"
}
```

» Argument Reference

The following arguments are supported:

- `administration_role_arn` - (Required) Amazon Resource Number (ARN) of the IAM Role in the administrator account.
- `name` - (Required) Name of the Stack Set. The name must be unique in the region where you create your Stack Set. The name can contain only alphanumeric characters (case-sensitive) and hyphens. It must start with an alphabetic character and cannot be longer than 128 characters.
- `capabilities` - (Optional) A list of capabilities. Valid values: `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_AUTO_EXPAND`.
- `description` - (Optional) Description of the Stack Set.
- `execution_role_name` - (Optional) Name of the IAM Role in all target accounts for Stack Set operations. Defaults to `AWSCloudFormationStackSetExecutionRole`.
- `parameters` - (Optional) Key-value map of input parameters for the Stack Set template. All template parameters, including those with a `Default`, must be configured or ignored with `lifecycle` configuration block `ignore_changes` argument. All `NoEcho` template parameters must be ignored with the `lifecycle` configuration block `ignore_changes` argument.
- `tags` - (Optional) Key-value map of tags to associate with this Stack Set and the Stacks created from it. AWS CloudFormation also propagates these tags to supported resources that are created in the Stacks. A maximum number of 50 tags can be specified.
- `template_body` - (Optional) String containing the CloudFormation template body. Maximum size: 51,200 bytes. Conflicts with `template_url`.
- `template_url` - (Optional) String containing the location of a file containing the CloudFormation template body. The URL must point to a template that is located in an Amazon S3 bucket. Maximum location file size: 460,800 bytes. Conflicts with `template_body`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the Stack Set.
- `id` - Name of the Stack Set.
- `stack_set_id` - Unique identifier of the Stack Set.

» Import

CloudFormation Stack Sets can be imported using the `name`, e.g.

```
$ terraform import aws_cloudformation_stack.example example
```

» Resource: aws_cloudformation_stack_set_instance

Manages a CloudFormation Stack Set Instance. Instances are managed in the account and region of the Stack Set after the target account permissions have been configured. Additional information about Stack Sets can be found in the AWS CloudFormation User Guide.

NOTE: All target accounts must have an IAM Role created that matches the name of the execution role configured in the Stack Set (the `execution_role_name` argument in the `aws_cloudformation_stack_set` resource) in a trust relationship with the administrative account or administration IAM Role. The execution role must have appropriate permissions to manage resources defined in the template along with those required for Stack Sets to operate. See the AWS CloudFormation User Guide for more details.

NOTE: To retain the Stack during Terraform resource destroy, ensure `retain_stack = true` has been successfully applied into the Terraform state first. This must be completed *before* an apply that would destroy the resource.

» Example Usage

```
resource "aws_cloudformation_stack_set_instance" "example" {
  account_id     = "123456789012"
  region         = "us-east-1"
  stack_set_name = "${aws_cloudformation_stack_set.example.name}"
}
```

» Example IAM Setup in Target Account

```
data "aws_iam_policy_document" "AWSCloudFormationStackSetExecutionRole_assume_role_policy" 
  statement {
    actions = ["sts:AssumeRole"]
    effect  = "Allow"

    principals {
      identifiers = ["${aws_iam_role.AWSCloudFormationStackSetAdministrationRole.arn}"]
      type        = "AWS"
    }
  }
}

resource "aws_iam_role" "AWSCloudFormationStackSetExecutionRole" {
  assume_role_policy = "${data.aws_iam_policy_document.AWSCloudFormationStackSetExecutionRol
  name               = "AWSCloudFormationStackSetExecutionRole"
}

# Documentation: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacksets-pr
# Additional IAM permissions necessary depend on the resources defined in the Stack Set temp
data "aws_iam_policy_document" "AWSCloudFormationStackSetExecutionRole_MinimumExecutionPolic
  statement {
    actions = [
      "cloudformation:*",
      "s3:*",
      "sns:*",
    ]

    effect    = "Allow"
    resources = ["*"]
  }
}

resource "aws_iam_role_policy" "AWSCloudFormationStackSetExecutionRole_MinimumExecutionPolic
  name   = "MinimumExecutionPolicy"
  policy = "${data.aws_iam_policy_document.AWSCloudFormationStackSetExecutionRole_MinimumExe
  role   = "${aws_iam_role.AWSCloudFormationStackSetExecutionRole.name}"
}
```

» Argument Reference

The following arguments are supported:

- stack_set_name - (Required) Name of the Stack Set.
- account_id - (Optional) Target AWS Account ID to create a Stack based on the Stack Set. Defaults to current account.
- parameter_overrides - (Optional) Key-value map of input parameters to override from the Stack Set for this Instance.
- region - (Optional) Target AWS Region to create a Stack based on the Stack Set. Defaults to current region.
- retain_stack - (Optional) During Terraform resource destroy, remove Instance from Stack Set while keeping the Stack and its associated resources. Must be enabled in Terraform state *before* destroy operation to take effect. You cannot reassociate a retained Stack or add an existing, saved Stack to a new Stack Set. Defaults to false.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Stack Set name, target AWS account ID, and target AWS region separated by commas (`,`)
- `stack_id` - Stack identifier

» Timeouts

`aws_cloudformation_stack_set_instance` provides the following Timeouts configuration options:

- `create` - (Default `30m`) How long to wait for a Stack to be created.
- `update` - (Default `30m`) How long to wait for a Stack to be updated.
- `delete` - (Default `30m`) How long to wait for a Stack to be deleted.

» Import

CloudFormation Stack Set Instances can be imported using the Stack Set name, target AWS account ID, and target AWS region separated by commas (`,`) e.g.

```
$ terraform import aws_cloudformation_stack_set_instance.example example,123456789012,us-eas
```

» Resource: aws_cloudfront_distribution

Creates an Amazon CloudFront web distribution.

For information about CloudFront distributions, see the Amazon CloudFront Developer Guide. For specific information about creating CloudFront web distributions, see the POST Distribution page in the Amazon CloudFront API Reference.

NOTE: CloudFront distributions take about 15 minutes to a deployed state after creation or modification. During this time, deletes to resources will be blocked. If you need to delete a distribution that is enabled and you do not want to wait, you need to use the `retain_on_delete` flag.

» Example Usage

The following example below creates a CloudFront distribution with an S3 origin.

```
resource "aws_s3_bucket" "b" {
  bucket = "mybucket"
  acl    = "private"

  tags = {
    Name = "My bucket"
  }
}

locals {
  s3_origin_id = "myS3Origin"
}

resource "aws_cloudfront_distribution" "s3_distribution" {
  origin {
    domain_name = "${aws_s3_bucket.b.bucket_regional_domain_name}"
    origin_id   = "${local.s3_origin_id}"

    s3_origin_config {
      origin_access_identity = "origin-access-identity/cloudfront/ABCDEFG1234567"
    }
  }

  enabled             = true
  is_ipv6_enabled     = true
  comment             = "Some comment"
  default_root_object = "index.html"

  logging_config {
    include_cookies = false
    bucket          = "mylogs.s3.amazonaws.com"
    prefix          = "myprefix"
  }

  aliases = ["mysite.example.com", "yoursite.example.com"]

  default_cache_behavior {
    allowed_methods  = ["DELETE", "GET", "HEAD", "OPTIONS", "PATCH", "POST", "PUT"]
    cached_methods   = ["GET", "HEAD"]
    target_origin_id = "${local.s3_origin_id}"

    forwarded_values {
      query_string = false

      cookies {
        forward = "none"
      }
    }

    viewer_protocol_policy = "allow-all"
    min_ttl                = 0
    default_ttl            = 3600
    max_ttl                = 86400
  }

  # Cache behavior with precedence 0
  ordered_cache_behavior {
    path_pattern     = "/content/immutable/*"
    allowed_methods  = ["GET", "HEAD", "OPTIONS"]
    cached_methods   = ["GET", "HEAD", "OPTIONS"]
    target_origin_id = "${local.s3_origin_id}"

    forwarded_values {
      query_string = false
      headers      = ["Origin"]

      cookies {
        forward = "none"
      }
    }

    min_ttl                = 0
    default_ttl            = 86400
    max_ttl                = 31536000
    compress               = true
    viewer_protocol_policy = "redirect-to-https"
  }

  # Cache behavior with precedence 1
  ordered_cache_behavior {
    path_pattern     = "/content/*"
    allowed_methods  = ["GET", "HEAD", "OPTIONS"]
    cached_methods   = ["GET", "HEAD"]
    target_origin_id = "${local.s3_origin_id}"

    forwarded_values {
      query_string = false

      cookies {
        forward = "none"
      }
    }
```

```
    min_ttl                = 0
    default_ttl            = 3600
    max_ttl                = 86400
    compress               = true
    viewer_protocol_policy = "redirect-to-https"
  }

  price_class = "PriceClass_200"

  restrictions {
    geo_restriction {
      restriction_type = "whitelist"
      locations        = ["US", "CA", "GB", "DE"]
    }
  }

  tags = {
    Environment = "production"
  }

  viewer_certificate {
    cloudfront_default_certificate = true
  }
}
```

The following example below creates a Cloudfront distribution with an origin group for failover routing:

```
resource "aws_cloudfront_distribution" "s3_distribution" {
  origin_group {
    origin_id = "groupS3"

    failover_criteria {
      status_codes = [403, 404, 500, 502]
    }

    member {
      origin_id = "primaryS3"
    }

    member {
      origin_id = "failoverS3"
    }
  }

  origin {
    domain_name = "${aws_s3_bucket.primary.bucket_regional_domain_name}"
    origin_id   = "primaryS3"

    s3_origin_config {
      origin_access_identity = "${aws_cloudfront_origin_access_identity.default.cloudfront_a
    }
  }

  origin {
    domain_name = "${aws_s3_bucket.failover.bucket_regional_domain_name}"
    origin_id   = "failoverS3"

    s3_origin_config {
      origin_access_identity = "${aws_cloudfront_origin_access_identity.default.cloudfront_a
    }
  }

  default_cache_behavior {
    # ... other configuration ...
    target_origin_id = "groupS3"
  }

  # ... other configuration ...
}
```

» Argument Reference

The CloudFront distribution argument layout is a complex structure composed of several sub-resources - these resources are laid out below.

» Top-Level Arguments

- `aliases` (Optional) - Extra CNAMEs (alternate domain names), if any, for this distribution.
- `comment` (Optional) - Any comments you want to include about the distribution.
- `custom_error_response` (Optional) - One or more custom error response elements (multiples allowed).
- `default_cache_behavior` (Required) - The default cache behavior for this distribution (maximum one).
- `default_root_object` (Optional) - The object that you want CloudFront to return (for example, index.html) when an end user requests the root

URL.

- enabled (Required) - Whether the distribution is enabled to accept end user requests for content.
- is_ipv6_enabled (Optional) - Whether the IPv6 is enabled for the distribution.
- http_version (Optional) - The maximum HTTP version to support on the distribution. Allowed values are http1.1 and http2. The default is http2.
- logging_config (Optional) - The logging configuration that controls how logs are written to your distribution (maximum one).
- ordered_cache_behavior (Optional) - An ordered list of cache behaviors resource for this distribution. List from top to bottom in order of precedence. The topmost cache behavior will have precedence 0.
- origin (Required) - One or more origins for this distribution (multiples allowed).
- origin_group (Optional) - One or more origin_group for this distribution (multiples allowed).
- price_class (Optional) - The price class for this distribution. One of PriceClass_All, PriceClass_200, PriceClass_100
- restrictions (Required) - The restriction configuration for this distribution (maximum one).
- tags - (Optional) A mapping of tags to assign to the resource.
- viewer_certificate (Required) - The SSL configuration for this distribution (maximum one).
- web_acl_id (Optional) - If you're using AWS WAF to filter CloudFront requests, the Id of the AWS WAF web ACL that is associated with the distribution. The WAF Web ACL must exist in the WAF Global (CloudFront) region and the credentials configuring this argument must have waf:GetWebACL permissions assigned.
- retain_on_delete (Optional) - Disables the distribution instead of deleting it when destroying the resource through Terraform. If this is set, the distribution needs to be deleted manually afterwards. Default: false.
- wait_for_deployment (Optional) - If enabled, the resource will wait for the distribution status to change from InProgress to Deployed. Setting this tofalse will skip the process. Default: true.

» Cache Behavior Arguments

- `allowed_methods` (Required) - Controls which HTTP methods CloudFront processes and forwards to your Amazon S3 bucket or your custom origin.
- `cached_methods` (Required) - Controls whether CloudFront caches the response to requests using the specified HTTP methods.
- `compress` (Optional) - Whether you want CloudFront to automatically compress content for web requests that include `Accept-Encoding: gzip` in the request header (default: `false`).
- `default_ttl` (Optional) - The default amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request in the absence of an `Cache-Control max-age` or `Expires` header. Defaults to 1 day.
- `field_level_encryption_id` (Optional) - Field level encryption configuration ID
- `forwarded_values` (Required) - The forwarded values configuration that specifies how CloudFront handles query strings, cookies and headers (maximum one).
- `lambda_function_association` (Optional) - A config block that triggers a lambda function with specific actions. Defined below, maximum 4.
- `max_ttl` (Optional) - The maximum amount of time (in seconds) that an object is in a CloudFront cache before CloudFront forwards another request to your origin to determine whether the object has been updated. Only effective in the presence of `Cache-Control max-age`, `Cache-Control s-maxage`, and `Expires` headers. Defaults to 365 days.
- `min_ttl` (Optional) - The minimum amount of time that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated. Defaults to 0 seconds.
- `path_pattern` (Required) - The pattern (for example, `images/*.jpg)` that specifies which requests you want this cache behavior to apply to.
- `smooth_streaming` (Optional) - Indicates whether you want to distribute media files in Microsoft Smooth Streaming format using the origin that is associated with this cache behavior.
- `target_origin_id` (Required) - The value of ID for the origin that you want CloudFront to route requests to when a request matches the path pattern either for a cache behavior or for the default cache behavior.
- `trusted_signers` (Optional) - The AWS accounts, if any, that you want to allow to create signed URLs for private content.
- `viewer_protocol_policy` (Required) - Use this element to specify the protocol that users can use to access the files in the origin specified by

TargetOriginId when a request matches the path pattern in PathPattern. One of `allow-all`, `https-only`, or `redirect-to-https`.

» Forwarded Values Arguments

- `cookies` (Required) - The forwarded values cookies that specifies how CloudFront handles cookies (maximum one).
- `headers` (Optional) - Specifies the Headers, if any, that you want CloudFront to vary upon for this cache behavior. Specify `*` to include all headers.
- `query_string` (Required) - Indicates whether you want CloudFront to forward query strings to the origin that is associated with this cache behavior.
- `query_string_cache_keys` (Optional) - When specified, along with a value of `true` for `query_string`, all query strings are forwarded, however only the query string keys listed in this argument are cached. When omitted with a value of `true` for `query_string`, all query string keys are cached.

» Lambda Function Association

Lambda@Edge allows you to associate an AWS Lambda Function with a predefined event. You can associate a single function per event type. See What is Lambda@Edge for more information.

Example configuration:

```
resource "aws_cloudfront_distribution" "example" {
  # ... other configuration ...

  # lambda_function_association is also supported by default_cache_behavior
  ordered_cache_behavior {
    # ... other configuration ...

    lambda_function_association {
      event_type   = "viewer-request"
      lambda_arn   = "${aws_lambda_function.example.qualified_arn}"
      include_body = false
    }
  }
}
```

- `event_type` (Required) - The specific event to trigger this function. Valid values: `viewer-request`, `origin-request`, `viewer-response`, `origin-response`
- `lambda_arn` (Required) - ARN of the Lambda function.

- include_body (Optional) - When set to true it exposes the request body to the lambda function. Defaults to false. Valid values: true, false.

» Cookies Arguments

- forward (Required) - Specifies whether you want CloudFront to forward cookies to the origin that is associated with this cache behavior. You can specify all, none or whitelist. If whitelist, you must include the subsequent whitelisted_names
- whitelisted_names (Optional) - If you have specified whitelist to forward, the whitelisted cookies that you want CloudFront to forward to your origin.

» Custom Error Response Arguments

- error_caching_min_ttl (Optional) - The minimum amount of time you want HTTP error codes to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated.
- error_code (Required) - The 4xx or 5xx HTTP status code that you want to customize.
- response_code (Optional) - The HTTP status code that you want CloudFront to return with the custom error page to the viewer.
- response_page_path (Optional) - The path of the custom error page (for example, /custom_404.html).

» Default Cache Behavior Arguments

The arguments for default_cache_behavior are the same as for ordered_cache_behavior, except for the path_pattern argument is not required.

» Logging Config Arguments

- bucket (Required) - The Amazon S3 bucket to store the access logs in, for example, myawslogbucket.s3.amazonaws.com.
- include_cookies (Optional) - Specifies whether you want CloudFront to include cookies in access logs (default: false).
- prefix (Optional) - An optional string that you want CloudFront to prefix to the access log filenames for this distribution, for example, myprefix/.

» Origin Arguments

- `custom_origin_config` - The CloudFront custom origin configuration information. If an S3 origin is required, use `s3_origin_config` instead.
- `domain_name` (Required) - The DNS domain name of either the S3 bucket, or web site of your custom origin.
- `custom_header` (Optional) - One or more sub-resources with `name` and `value` parameters that specify header data that will be sent to the origin (multiples allowed).
- `origin_id` (Required) - A unique identifier for the origin.
- `origin_path` (Optional) - An optional element that causes CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin.
- `s3_origin_config` - The CloudFront S3 origin configuration information. If a custom origin is required, use `custom_origin_config` instead.

» Custom Origin Config Arguments

- `http_port` (Required) - The HTTP port the custom origin listens on.
- `https_port` (Required) - The HTTPS port the custom origin listens on.
- `origin_protocol_policy` (Required) - The origin protocol policy to apply to your origin. One of `http-only`, `https-only`, or `match-viewer`.
- `origin_ssl_protocols` (Required) - The SSL/TLS protocols that you want CloudFront to use when communicating with your origin over HTTPS. A list of one or more of `SSLv3`, `TLSv1`, `TLSv1.1`, and `TLSv1.2`.
- `origin_keepalive_timeout` - (Optional) The Custom KeepAlive timeout, in seconds. By default, AWS enforces a limit of `60`. But you can request an increase.
- `origin_read_timeout` - (Optional) The Custom Read timeout, in seconds. By default, AWS enforces a limit of `60`. But you can request an increase.

» S3 Origin Config Arguments

- `origin_access_identity` (Optional) - The CloudFront origin access identity to associate with the origin.

» Origin Group Arguments

- `origin_id` (Required) - A unique identifier for the origin group.

- failover_criteria (Required) - The failover criteria for when to failover to the secondary origin
- member (Required) - Ordered member configuration blocks assigned to the origin group, where the first member is the primary origin. Minimum 2.

» Failover Criteria Arguments

- status_codes (Required) - A list of HTTP status codes for the origin group

» Member Arguments

- origin_id (Required) - The unique identifier of the member origin

» Restrictions Arguments

The restrictions sub-resource takes another single sub-resource named geo_restriction (see the example for usage).

The arguments of geo_restriction are:

- locations (Optional) - The ISO 3166-1-alpha-2 codes for which you want CloudFront either to distribute your content (whitelist) or not distribute your content (blacklist).
- restriction_type (Required) - The method that you want to use to restrict distribution of your content by country: none, whitelist, or blacklist.

» Viewer Certificate Arguments

- acm_certificate_arn - The ARN of the AWS Certificate Manager certificate that you wish to use with this distribution. Specify this, cloudfront_default_certificate, or iam_certificate_id. The ACM certificate must be in US-EAST-1.
- cloudfront_default_certificate - true if you want viewers to use HTTPS to request your objects and you're using the CloudFront domain name for your distribution. Specify this, acm_certificate_arn, or iam_certificate_id.
- iam_certificate_id - The IAM certificate identifier of the custom viewer certificate for this distribution if you are using a custom domain. Specify this, acm_certificate_arn, or cloudfront_default_certificate.
- minimum_protocol_version - The minimum version of the SSL protocol that you want CloudFront to use for HTTPS connections. One of

SSLv3, TLSv1, TLSv1_2016, TLSv1.1_2016 or TLSv1.2_2018. Default: TLSv1. **NOTE**: If you are using a custom certificate (specified with acm_certificate_arn or iam_certificate_id), and have specified sni-only in ssl_support_method, TLSv1 or later must be specified. If you have specified vip in ssl_support_method, only SSLv3 or TLSv1 can be specified. If you have specified cloudfront_default_certificate, TLSv1 must be specified.

- ssl_support_method: Specifies how you want CloudFront to serve HTTPS requests. One of vip or sni-only. Required if you specify acm_certificate_arn or iam_certificate_id. **NOTE:** vip causes CloudFront to use a dedicated IP address and may incur extra charges.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- id - The identifier for the distribution. For example: EDFDVBD632BHDS5.
- arn - The ARN (Amazon Resource Name) for the distribution. For example: arn:aws:cloudfront::123456789012:distribution/EDFDVBD632BHDS5, where 123456789012 is your AWS account ID.
- caller_reference - Internal value used by CloudFront to allow future updates to the distribution configuration.
- status - The current status of the distribution. Deployed if the distribution's information is fully propagated throughout the Amazon CloudFront system.
- active_trusted_signers - The key pair IDs that CloudFront is aware of for each trusted signer, if the distribution is set up to serve private content with signed URLs.
- domain_name - The domain name corresponding to the distribution. For example: d604721fxaaqy9.cloudfront.net.
- last_modified_time - The date and time the distribution was last modified.
- in_progress_validation_batches - The number of invalidation batches currently in progress.
- etag - The current version of the distribution's information. For example: E2QWRUHAPOMQZL.
- hosted_zone_id - The CloudFront Route 53 zone ID that can be used to route an Alias Resource Record Set to. This attribute is simply an alias for the zone ID Z2FDTNDATAQYW2.

» Import

Cloudfront Distributions can be imported using the `id`, e.g.

```
$ terraform import aws_cloudfront_distribution.distribution E74FTE3EXAMPLE
```

» Resource: aws_cloudfront_origin_access_identity

Creates an Amazon CloudFront origin access identity.

For information about CloudFront distributions, see the Amazon CloudFront Developer Guide. For more information on generating origin access identities, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

» Example Usage

The following example below creates a CloudFront origin access identity.

```
resource "aws_cloudfront_origin_access_identity" "origin_access_identity" {
  comment = "Some comment"
}
```

» Argument Reference

- `comment` (Optional) - An optional comment for the origin access identity.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The identifier for the distribution. For example: `EDFDVBD632BHDS5`.
- `caller_reference` - Internal value used by CloudFront to allow future updates to the origin access identity.
- `cloudfront_access_identity_path` - A shortcut to the full path for the origin access identity to use in CloudFront, see below.
- `etag` - The current version of the origin access identity's information. For example: `E2QWRUHAPOMQZL`.
- `iam_arn` - A pre-generated ARN for use in S3 bucket policies (see below). Example: `arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity E2QWRUHAPOMQZL`.

- s3_canonical_user_id - The Amazon S3 canonical user ID for the origin access identity, which you use when giving the origin access identity read permission to an object in Amazon S3.

» Using With CloudFront

Normally, when referencing an origin access identity in CloudFront, you need to prefix the ID with the origin-access-identity/cloudfront/ special path. The cloudfront_access_identity_path allows this to be circumvented. The below snippet demonstrates use with the s3_origin_config structure for the aws_cloudfront_distribution resource:

```
s3_origin_config {
  origin_access_identity = "${aws_cloudfront_origin_access_identity.origin_access_identity.c
}
```

» Updating your bucket policy

Note that the AWS API may translate the s3_canonical_user_id CanonicalUser principal into an AWS IAM ARN principal when supplied in an aws_s3_bucket bucket policy, causing spurious diffs in Terraform. If you see this behaviour, use the iam_arn instead:

```
data "aws_iam_policy_document" "s3_policy" {
  statement {
    actions   = ["s3:GetObject"]
    resources = ["${aws_s3_bucket.example.arn}/*"]

    principals {
      type        = "AWS"
      identifiers = ["${aws_cloudfront_origin_access_identity.origin_access_identity.iam_arn
    }
  }

  statement {
    actions   = ["s3:ListBucket"]
    resources = ["${aws_s3_bucket.example.arn}"]

    principals {
      type        = "AWS"
      identifiers = ["${aws_cloudfront_origin_access_identity.origin_access_identity.iam_arn
    }
  }
}
```

```
resource "aws_s3_bucket_policy" "example" {
  bucket = "${aws_s3_bucket.example.id}"
  policy = "${data.aws_iam_policy_document.s3_policy.json}"
}
```

» Import

Cloudfront Origin Access Identities can be imported using the `id`, e.g.

```
$ terraform import aws_cloudfront_origin_access_identity.origin_access E74FTE3AEXAMPLE
```

» Resource: aws_cloudfront_public_key

» Example Usage

The following example below creates a CloudFront public key.

```
resource "aws_cloudfront_public_key" "example" {
  comment     = "test public key"
  encoded_key = "${file("public_key.pem")}"
  name        = "test_key"
}
```

» Argument Reference

The following arguments are supported:

- `comment` - (Optional) An optional comment about the public key.
- `encoded_key` - (Required) The encoded public key that you want to add to CloudFront to use with features like field-level encryption.
- `name` - (Optional) The name for the public key. By default generated by Terraform.
- `name_prefix` - (Optional) The name for the public key. Conflicts with `name`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `caller_reference` - Internal value used by CloudFront to allow future updates to the public key configuration.
- `etag` - The current version of the public key. For example: `E2QWRUHAPOMQZL`.
- `id` - The identifier for the public key. For example: `K3D5EWEUDCCXON`.

» Data Source: aws_cloudhsm_v2_cluster

Use this data source to get information about a CloudHSM v2 cluster

» Example Usage

```
data "aws_cloudhsm_v2_cluster" "cluster" {
  cluster_id = "cluster-testclusterid"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_id` - (Required) The id of Cloud HSM v2 cluster.
- `cluster_state` - (Optional) The state of the cluster to be found.

» Attributes Reference

The following attributes are exported:

- `vpc_id` - The id of the VPC that the CloudHSM cluster resides in.
- `security_group_id` - The ID of the security group associated with the CloudHSM cluster.
- `subnet_ids` - The IDs of subnets in which cluster operates.
- `cluster_certificates` - The list of cluster certificates.
  - `cluster_certificates.0.cluster_certificate` - The cluster certificate issued (signed) by the issuing certificate authority (CA) of the cluster's owner.
  - `cluster_certificates.0.cluster_csr` - The certificate signing request (CSR). Available only in UNINITIALIZED state.
  - `cluster_certificates.0.aws_hardware_certificate` - The HSM hardware certificate issued (signed) by AWS CloudHSM.
  - `cluster_certificates.0.hsm_certificate` - The HSM certificate issued (signed) by the HSM hardware.
  - `cluster_certificates.0.manufacturer_hardware_certificate` - The HSM hardware certificate issued (signed) by the hardware manufacturer. The number of available cluster certificates may vary depending on state of the cluster.

» Resource: aws_cloudhsm_v2_cluster

Creates an Amazon CloudHSM v2 cluster.

For information about CloudHSM v2, see the AWS CloudHSM User Guide and the Amazon CloudHSM API Reference.

NOTE: CloudHSM can take up to several minutes to be set up. Practically no single attribute can be updated except TAGS. If you need to delete a cluster, you have to remove its HSM modules first. To initialize cluster, you have to add an hsm instance to the cluster then sign CSR and upload it.

» Example Usage

The following example below creates a CloudHSM cluster.

```
provider "aws" {
  region = "${var.aws_region}"
}

data "aws_availability_zones" "available" {}

resource "aws_vpc" "cloudhsm2_vpc" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "example-aws_cloudhsm_v2_cluster"
  }
}

resource "aws_subnet" "cloudhsm2_subnets" {
  count                   = 2
  vpc_id                  = "${aws_vpc.cloudhsm2_vpc.id}"
  cidr_block              = "${element(var.subnets, count.index)}"
  map_public_ip_on_launch = false
  availability_zone       = "${element(data.aws_availability_zones.available.names, count.in

  tags = {
    Name = "example-aws_cloudhsm_v2_cluster"
  }
}

resource "aws_cloudhsm_v2_cluster" "cloudhsm_v2_cluster" {
  hsm_type   = "hsm1.medium"
  subnet_ids = ["${aws_subnet.cloudhsm2_subnets.*.id}"]
  tags = {
    Name = "example-aws_cloudhsm_v2_cluster"
  }
}
```

» Argument Reference

The following arguments are supported:

- `source_backup_identifier` - (Optional) The id of Cloud HSM v2 cluster backup to be restored.
- `hsm_type` - (Required) The type of HSM module in the cluster. Currently, only hsm1.medium is supported.
- `subnet_ids` - (Required) The IDs of subnets in which cluster will operate.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `cluster_id` - The id of the CloudHSM cluster.
- `cluster_state` - The state of the cluster.
- `vpc_id` - The id of the VPC that the CloudHSM cluster resides in.
- `security_group_id` - The ID of the security group associated with the CloudHSM cluster.
- `cluster_certificates` - The list of cluster certificates.
  - `cluster_certificates.0.cluster_certificate` - The cluster certificate issued (signed) by the issuing certificate authority (CA) of the cluster's owner.
  - `cluster_certificates.0.cluster_csr` - The certificate signing request (CSR). Available only in UNINITIALIZED state after an hsm instance is added to the cluster.
  - `cluster_certificates.0.aws_hardware_certificate` - The HSM hardware certificate issued (signed) by AWS CloudHSM.
  - `cluster_certificates.0.hsm_certificate` - The HSM certificate issued (signed) by the HSM hardware.
  - `cluster_certificates.0.manufacturer_hardware_certificate` - The HSM hardware certificate issued (signed) by the hardware manufacturer.

» Resource: aws_cloudhsm_v2_hsm

Creates an HSM module in Amazon CloudHSM v2 cluster.

» Example Usage

The following example below creates an HSM module in CloudHSM cluster.

```
data "aws_cloudhsm_v2_cluster" "cluster" {
  cluster_id = "${var.cloudhsm_cluster_id}"
}

resource "aws_cloudhsm_v2_hsm" "cloudhsm_v2_hsm" {
  subnet_id  = "${data.aws_cloudhsm_v2_cluster.cluster.subnet_ids[0]}"
  cluster_id = "${data.aws_cloudhsm_v2_cluster.cluster.cluster_id}"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_id` - (Required) The ID of Cloud HSM v2 cluster to which HSM will be added.
- `subnet_id` - (Optional) The ID of subnet in which HSM module will be located.
- `availability_zone` - (Optional) The IDs of AZ in which HSM module will be located. Do not use together with subnet_id.
- `ip_address` - (Optional) The IP address of HSM module. Must be within the CIDR of selected subnet.

» Attributes Reference

The following attributes are exported:

- `hsm_id` - The id of the HSM module.
- `hsm_state` - The state of the HSM module.
- `hsm_eni_id` - The id of the ENI interface allocated for HSM module.

» Data Source: aws_cloudtrail_service_account

Use this data source to get the Account ID of the AWS CloudTrail Service Account in a given region for the purpose of allowing CloudTrail to store trail data in S3.

» Example Usage

```
data "aws_cloudtrail_service_account" "main" {}
```

```
resource "aws_s3_bucket" "bucket" {
  bucket        = "tf-cloudtrail-logging-test-bucket"
  force_destroy = true

  policy = <<EOF
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "Put bucket policy needed for trails",
      "Effect": "Allow",
      "Principal": {
        "AWS": "${data.aws_cloudtrail_service_account.main.arn}"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::tf-cloudtrail-logging-test-bucket/*"
    },
    {
      "Sid": "Get bucket policy needed for trails",
      "Effect": "Allow",
      "Principal": {
        "AWS": "${data.aws_cloudtrail_service_account.main.arn}"
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::tf-cloudtrail-logging-test-bucket"
    }
  ]
}
EOF
}
```

» Argument Reference

- `region` - (Optional) Name of the region whose AWS CloudTrail account ID is desired. Defaults to the region from the AWS provider configuration.

» Attributes Reference

- `id` - The ID of the AWS CloudTrail service account in the selected region.
- `arn` - The ARN of the AWS CloudTrail service account in the selected region.

» Resource: aws_cloudtrail

Provides a CloudTrail resource.

NOTE: For a multi-region trail, this resource must be in the home region of the trail.

NOTE: For an organization trail, this resource must be in the master account of the organization.

» Example Usage

» Basic

Enable CloudTrail to capture all compatible management events in region. For capturing events from services like IAM, `include_global_service_events` must be enabled.

```
data "aws_caller_identity" "current" {}

resource "aws_cloudtrail" "foobar" {
  name                          = "tf-trail-foobar"
  s3_bucket_name                = "${aws_s3_bucket.foo.id}"
  s3_key_prefix                 = "prefix"
  include_global_service_events = false
}

resource "aws_s3_bucket" "foo" {
  bucket        = "tf-test-trail"
  force_destroy = true

  policy = <<POLICY
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailAclCheck",
            "Effect": "Allow",
            "Principal": {
              "Service": "cloudtrail.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::tf-test-trail"
        },
        {
            "Sid": "AWSCloudTrailWrite",
            "Effect": "Allow",
            "Principal": {
              "Service": "cloudtrail.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::tf-test-trail/prefix/AWSLogs/${data.aws_caller_identit
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
POLICY
}
```

» Data Event Logging

CloudTrail can log Data Events for certain services such as S3 bucket objects and Lambda function invocations. Additional information about data event configuration can be found in the CloudTrail API DataResource documentation.

» Logging All Lambda Function Invocations

```
resource "aws_cloudtrail" "example" {
  # ... other configuration ...

  event_selector {
    read_write_type           = "All"
    include_management_events = true

    data_resource {
      type   = "AWS::Lambda::Function"
      values = ["arn:aws:lambda"]
    }
  }
}
```

» Logging All S3 Bucket Object Events

```
resource "aws_cloudtrail" "example" {
  # ... other configuration ...

  event_selector {
    read_write_type           = "All"
    include_management_events = true

    data_resource {
      type   = "AWS::S3::Object"
      values = ["arn:aws:s3:::"]
    }
  }
}
```

» Logging Individual S3 Bucket Events

```
data "aws_s3_bucket" "important-bucket" {
  bucket = "important-bucket"
}

resource "aws_cloudtrail" "example" {
  # ... other configuration ...

  event_selector {
    read_write_type           = "All"
    include_management_events = true

    data_resource {
      type = "AWS::S3::Object"

      # Make sure to append a trailing '/' to your ARN if you want
      # to monitor all objects in a bucket.
      values = ["${data.aws_s3_bucket.important-bucket.arn}/"]
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the trail.
- `s3_bucket_name` - (Required) Specifies the name of the S3 bucket designated for publishing log files.
- `s3_key_prefix` - (Optional) Specifies the S3 key prefix that follows the name of the bucket you have designated for log file delivery.
- `cloud_watch_logs_role_arn` - (Optional) Specifies the role for the CloudWatch Logs endpoint to assume to write to a user's log group.

- `cloud_watch_logs_group_arn` - (Optional) Specifies a log group name using an Amazon Resource Name (ARN), that represents the log group to which CloudTrail logs will be delivered.
- `enable_logging` - (Optional) Enables logging for the trail. Defaults to `true`. Setting this to `false` will pause logging.
- `include_global_service_events` - (Optional) Specifies whether the trail is publishing events from global services such as IAM to the log files. Defaults to `true`.
- `is_multi_region_trail` - (Optional) Specifies whether the trail is created in the current region or in all regions. Defaults to `false`.
- `is_organization_trail` - (Optional) Specifies whether the trail is an AWS Organizations trail. Organization trails log events for the master account and all member accounts. Can only be created in the organization master account. Defaults to `false`.
- `sns_topic_name` - (Optional) Specifies the name of the Amazon SNS topic defined for notification of log file delivery.
- `enable_log_file_validation` - (Optional) Specifies whether log file integrity validation is enabled. Defaults to `false`.
- `kms_key_id` - (Optional) Specifies the KMS key ARN to use to encrypt the logs delivered by CloudTrail.
- `event_selector` - (Optional) Specifies an event selector for enabling data event logging. Fields documented below. Please note the CloudTrail limits when configuring these.
- `tags` - (Optional) A mapping of tags to assign to the trail

» Event Selector Arguments

For **event_selector** the following attributes are supported.

- `read_write_type` (Optional) - Specify if you want your trail to log read-only events, write-only events, or all. By default, the value is All. You can specify only the following value: ”ReadOnly”, ”WriteOnly”, ”All”. Defaults to `All`.
- `include_management_events` (Optional) - Specify if you want your event selector to include management events for your trail.
- `data_resource` (Optional) - Specifies logging data events. Fields documented below.

» Data Resource Arguments

For **data_resource** the following attributes are supported.

- `type` (Required) - The resource type in which you want to log data events. You can specify only the follwing value: ”AWS::S3::Object”, ”AWS::Lambda::Function”

- `values` (Required) - A list of ARN for the specified S3 buckets and object prefixes..

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the trail.
- `home_region` - The region in which the trail was created.
- `arn` - The Amazon Resource Name of the trail.

» Import

Cloudtrails can be imported using the `name`, e.g.

```
$ terraform import aws_cloudtrail.sample my-sample-trail
```

» Data Source: aws_cloudwatch_log_group

Use this data source to get information about an AWS Cloudwatch Log Group

» Example Usage

```
data "aws_cloudwatch_log_group" "example" {
  name = "MyImportantLogs"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Cloudwatch log group

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the Cloudwatch log group
- `creation_time` - The creation time of the log group, expressed as the number of milliseconds after Jan 1, 1970 00:00:00 UTC.

» Resource: aws_cloudwatch_dashboard

Provides a CloudWatch Dashboard resource.

» Example Usage

```
resource "aws_cloudwatch_dashboard" "main" {
  dashboard_name = "my-dashboard"

  dashboard_body = <<EOF
 {
   "widgets": [
       {
          "type":"metric",
          "x":0,
          "y":0,
          "width":12,
          "height":6,
          "properties":{
             "metrics":[
                [
                   "AWS/EC2",
                   "CPUUtilization",
                   "InstanceId",
                   "i-012345"
                ]
             ],
             "period":300,
             "stat":"Average",
             "region":"us-east-1",
             "title":"EC2 Instance CPU"
          }
       },
       {
          "type":"text",
          "x":0,
          "y":7,
          "width":3,
          "height":3,
          "properties":{
             "markdown":"Hello world"
          }
       }
   ]
 }
 EOF
}
```

» Argument Reference

The following arguments are supported:

- `dashboard_name` - (Required) The name of the dashboard.
- `dashboard_body` - (Required) The detailed information about the dashboard, including what widgets are included and their location on the dashboard. You can read more about the body structure in the documentation.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `dashboard_arn` - The Amazon Resource Name (ARN) of the dashboard.

» Import

CloudWatch dashboards can be imported using the `dashboard_name`, e.g.

```
$ terraform import aws_cloudwatch_dashboard.sample <dashboard_name>
```

» Resource: aws_cloudwatch_event_permission

Provides a resource to create a CloudWatch Events permission to support cross-account events in the current account default event bus.

» Example Usage

» Account Access

```
resource "aws_cloudwatch_event_permission" "DevAccountAccess" {
  principal    = "123456789012"
  statement_id = "DevAccountAccess"
}
```

» Organization Access

```
resource "aws_cloudwatch_event_permission" "OrganizationAccess" {
  principal    = "*"
  statement_id = "OrganizationAccess"

  condition {
    key   = "aws:PrincipalOrgID"
    type  = "StringEquals"
    value = "${aws_organizations_organization.example.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `principal` - (Required) The 12-digit AWS account ID that you are permitting to put events to your default event bus. Specify `*` to permit any account to put events to your default event bus, optionally limited by `condition`.
- `statement_id` - (Required) An identifier string for the external account that you are granting permissions to.
- `action` - (Optional) The action that you are enabling the other account to perform. Defaults to `events:PutEvents`.
- `condition` - (Optional) Configuration block to limit the event bus permissions you are granting to only accounts that fulfill the condition. Specified below.

» condition

- `key` - (Required) Key for the condition. Valid values: `aws:PrincipalOrgID`.
- `type` - (Required) Type of condition. Value values: `StringEquals`.
- `value` - (Required) Value for the key.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The statement ID of the CloudWatch Events permission.

» Import

CloudWatch Events permissions can be imported using the statement ID, e.g.

```
$ terraform import aws_cloudwatch_event_permission.DevAccountAccess DevAccountAccess
```

» Resource: aws_cloudwatch_event_rule

Provides a CloudWatch Event Rule resource.

» Example Usage

```
resource "aws_cloudwatch_event_rule" "console" {
  name        = "capture-aws-sign-in"
  description = "Capture each AWS Console Sign In"

  event_pattern = <<PATTERN
{
  "detail-type": [
    "AWS Console Sign In via CloudTrail"
  ]
}
PATTERN
}

resource "aws_cloudwatch_event_target" "sns" {
  rule      = "${aws_cloudwatch_event_rule.console.name}"
  target_id = "SendToSNS"
  arn       = "${aws_sns_topic.aws_logins.arn}"
}

resource "aws_sns_topic" "aws_logins" {
  name = "aws-console-logins"
}

resource "aws_sns_topic_policy" "default" {
  arn    = "${aws_sns_topic.aws_logins.arn}"
  policy = "${data.aws_iam_policy_document.sns_topic_policy.json}"
}

data "aws_iam_policy_document" "sns_topic_policy" {
  statement {
    effect  = "Allow"
    actions = ["SNS:Publish"]
    principals {
      type        = "Service"
      identifiers = ["events.amazonaws.com"]
    }

    resources = ["${aws_sns_topic.aws_logins.arn}"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The rule's name. By default generated by Terraform.
- `name_prefix` - (Optional) The rule's name. Conflicts with `name`.
- `schedule_expression` - (Required, if `event_pattern` isn't specified) The scheduling expression. For example, `cron(0 20 * * ? *)` or `rate(5 minutes)`.
- `event_pattern` - (Required, if `schedule_expression` isn't specified) Event pattern described a JSON object. See full documentation of CloudWatch Events and Event Patterns for details.
- `description` - (Optional) The description of the rule.
- `role_arn` - (Optional) The Amazon Resource Name (ARN) associated with the role that is used for target invocation.
- `is_enabled` - (Optional) Whether the rule should be enabled (defaults to `true`).
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the rule.
- `arn` - The Amazon Resource Name (ARN) of the rule.

» Import

Cloudwatch Event Rules can be imported using the `name`, e.g.

```
$ terraform import aws_cloudwatch_event_rule.console capture-console-sign-in
```

» Resource: aws_cloudwatch_event_target

Provides a CloudWatch Event Target resource.

» Example Usage

```
resource "aws_cloudwatch_event_target" "yada" {
  target_id = "Yada"
  rule      = "${aws_cloudwatch_event_rule.console.name}"
  arn       = "${aws_kinesis_stream.test_stream.arn}"

  run_command_targets {
    key    = "tag:Name"
    values = ["FooBar"]
  }

  run_command_targets {
    key    = "InstanceIds"
    values = ["i-162058cd308bffec2"]
  }
}

resource "aws_cloudwatch_event_rule" "console" {
  name        = "capture-ec2-scaling-events"
  description = "Capture all EC2 scaling events"

  event_pattern = <<PATTERN
{
  "source": [
    "aws.autoscaling"
  ],
  "detail-type": [
    "EC2 Instance Launch Successful",
    "EC2 Instance Terminate Successful",
    "EC2 Instance Launch Unsuccessful",
    "EC2 Instance Terminate Unsuccessful"
  ]
}
PATTERN
}

resource "aws_kinesis_stream" "test_stream" {
  name        = "terraform-kinesis-test"
  shard_count = 1
}
```

» Example SSM Document Usage

```
data "aws_iam_policy_document" "ssm_lifecycle_trust" {
  statement {
    actions = ["sts:AssumeRole"]

    principals {
      type        = "Service"
      identifiers = ["events.amazonaws.com"]
    }
  }
}

data "aws_iam_policy_document" "ssm_lifecycle" {
  statement {
    effect    = "Allow"
    actions   = ["ssm:SendCommand"]
    resources = ["arn:aws:ec2:eu-west-1:1234567890:instance/*"]

    condition {
      test     = "StringEquals"
      variable = "ec2:ResourceTag/Terminate"
      values   = ["*"]
    }
  }

  statement {
    effect    = "Allow"
    actions   = ["ssm:SendCommand"]
    resources = ["${aws_ssm_document.stop_instance.arn}"]
  }
}

resource "aws_iam_role" "ssm_lifecycle" {
  name               = "SSMLifecycle"
  assume_role_policy = "${data.aws_iam_policy_document.ssm_lifecycle_trust.json}"
}

resource "aws_iam_policy" "ssm_lifecycle" {
  name   = "SSMLifecycle"
  policy = "${data.aws_iam_policy_document.ssm_lifecycle.json}"
}
```

```
resource "aws_ssm_document" "stop_instance" {
  name          = "stop_instance"
  document_type = "Command"

  content = <<DOC
  {
    "schemaVersion": "1.2",
    "description": "Stop an instance",
    "parameters": {

    },
    "runtimeConfig": {
      "aws:runShellScript": {
        "properties": [
          {
            "id": "0.aws:runShellScript",
            "runCommand": ["halt"]
          }
        ]
      }
    }
  }
DOC
}

resource "aws_cloudwatch_event_rule" "stop_instances" {
  name                = "StopInstance"
  description         = "Stop instances nightly"
  schedule_expression = "cron(0 0 * * ? *)"
}

resource "aws_cloudwatch_event_target" "stop_instances" {
  target_id = "StopInstance"
  arn       = "${aws_ssm_document.stop_instance.arn}"
  rule      = "${aws_cloudwatch_event_rule.stop_instances.name}"
  role_arn  = "${aws_iam_role.ssm_lifecycle.arn}"

  run_command_targets {
    key    = "tag:Terminate"
    values = ["midnight"]
  }
}
```

» Example RunCommand Usage

```
resource "aws_cloudwatch_event_rule" "stop_instances" {
  name                = "StopInstance"
  description         = "Stop instances nightly"
  schedule_expression = "cron(0 0 * * ? *)"
}

resource "aws_cloudwatch_event_target" "stop_instances" {
  target_id = "StopInstance"
  arn       = "arn:aws:ssm:${var.aws_region}::document/AWS-RunShellScript"
  input     = "{\"commands\":[\"halt\"]}"
  rule      = "${aws_cloudwatch_event_rule.stop_instances.name}"
  role_arn  = "${aws_iam_role.ssm_lifecycle.arn}"

  run_command_targets {
    key    = "tag:Terminate"
    values = ["midnight"]
  }
}
```

» Example ECS Run Task with Role and Task Override Usage

```
resource "aws_iam_role" "ecs_events" {
  name = "ecs_events"

  assume_role_policy = <<DOC
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
DOC
}
```

```
resource "aws_iam_role_policy" "ecs_events_run_task_with_any_role" {
  name = "ecs_events_run_task_with_any_role"
  role = "${aws_iam_role.ecs_events.id}"

  policy = <<DOC
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ecs:RunTask",
            "Resource": "${replace(aws_ecs_task_definition.task_name.arn, "/:\\d+$/", ":*")}
        }
    ]
}
DOC
}

resource "aws_cloudwatch_event_target" "ecs_scheduled_task" {
  target_id = "run-scheduled-task-every-hour"
  arn       = "${aws_ecs_cluster.cluster_name.arn}"
  rule      = "${aws_cloudwatch_event_rule.every_hour.name}"
  role_arn  = "${aws_iam_role.ecs_events.arn}"

  ecs_target {
    task_count          = 1
    task_definition_arn = "${aws_ecs_task_definition.task_name.arn}"
  }

  input = <<DOC
{
  "containerOverrides": [
    {
      "name": "name-of-container-to-override",
      "command": ["bin/console", "scheduled-task"]
    }
  ]
}
DOC
}
```

» Argument Reference

Note: `input` and `input_path` are mutually exclusive options.

Note: In order to be able to have your AWS Lambda function or SNS topic invoked by a CloudWatch Events rule, you must setup the right permissions using `aws_lambda_permission` or `aws_sns_topic.policy`. More info here.

The following arguments are supported:

- `rule` - (Required) The name of the rule you want to add targets to.
- `target_id` - (Optional) The unique target assignment ID. If missing, will generate a random, unique id.
- `arn` - (Required) The Amazon Resource Name (ARN) associated of the target.
- `input` - (Optional) Valid JSON text passed to the target.
- `input_path` - (Optional) The value of the JSONPath that is used for extracting part of the matched event when passing it to the target.
- `role_arn` - (Optional) The Amazon Resource Name (ARN) of the IAM role to be used for this target when the rule is triggered. Required if `ecs_target` is used.
- `run_command_targets` - (Optional) Parameters used when you are using the rule to invoke Amazon EC2 Run Command. Documented below. A maximum of 5 are allowed.
- `ecs_target` - (Optional) Parameters used when you are using the rule to invoke Amazon ECS Task. Documented below. A maximum of 1 are allowed.
- `batch_target` - (Optional) Parameters used when you are using the rule to invoke an Amazon Batch Job. Documented below. A maximum of 1 are allowed.
- `kinesis_target` - (Optional) Parameters used when you are using the rule to invoke an Amazon Kinesis Stream. Documented below. A maximum of 1 are allowed.
- `sqs_target` - (Optional) Parameters used when you are using the rule to invoke an Amazon SQS Queue. Documented below. A maximum of 1 are allowed.
- `input_transformer` - (Optional) Parameters used when you are providing a custom input to a target based on certain event data.

`run_command_targets` support the following:

- `key` - (Required) Can be either `tag:tag-key` or `InstanceIds`.
- `values` - (Required) If Key is `tag:tag-key`, Values is a list of tag values. If Key is `InstanceIds`, Values is a list of Amazon EC2 instance IDs.

`ecs_target` support the following:

- `group` - (Optional) Specifies an ECS task group for the task. The maximum length is 255 characters.

- `launch_type` - (Optional) Specifies the launch type on which your task is running. The launch type that you specify here must match one of the launch type (compatibilities) of the target task. Valid values are EC2 or FARGATE.
- `network_configuration` - (Optional) Use this if the ECS task uses the awsvpc network mode. This specifies the VPC subnets and security groups associated with the task, and whether a public IP address is to be used. Required if launch_type is FARGATE because the awsvpc mode is required for Fargate tasks.
- `platform_version` - (Optional) Specifies the platform version for the task. Specify only the numeric portion of the platform version, such as 1.1.0. This is used only if LaunchType is FARGATE. For more information about valid platform versions, see AWS Fargate Platform Versions.
- `task_count` - (Optional) The number of tasks to create based on the TaskDefinition. The default is 1.
- `task_definition_arn` - (Required) The ARN of the task definition to use if the event target is an Amazon ECS cluster.

`network_configuration` support the following:

- `subnets` - (Required) The subnets associated with the task or service.
- `security_groups` - (Optional) The security groups associated with the task or service. If you do not specify a security group, the default security group for the VPC is used.
- `assign_public_ip` - (Optional) Assign a public IP address to the ENI (Fargate launch type only). Valid values are `true` or `false`. Default `false`.

For more information, see Task Networking

`batch_target` support the following:

- `job_definition` - (Required) The ARN or name of the job definition to use if the event target is an AWS Batch job. This job definition must already exist.
- `job_name` - (Required) The name to use for this execution of the job, if the target is an AWS Batch job.
- `array_size` - (Optional) The size of the array, if this is an array batch job. Valid values are integers between 2 and 10,000.
- `job_attempts` - (Optional) The number of times to attempt to retry, if the job fails. Valid values are 1 to 10.

`kinesis_target` support the following:

- `partition_key_path` - (Optional) The JSON path to be extracted from the event and used as the partition key.

`sqs_target` support the following:

- `message_group_id` - (Optional) The FIFO message group ID to use as the target.

`input_transformer` support the following:

- `input_paths` - (Optional) Key value pairs specified in the form of JSONPath (for example, time = $.time)
- `input_template` - (Required) Structure containing the template body.

» Import

Cloud Watch Event Target can be imported using the role event_rule and target_id separated by `/`.

```
$ terraform import aws_cloudwatch_event_target.test-event-target rule-name/target-id
```

» Resource: aws_cloudwatch_log_destination

Provides a CloudWatch Logs destination resource.

» Example Usage

```
resource "aws_cloudwatch_log_destination" "test_destination" {
  name       = "test_destination"
  role_arn   = "${aws_iam_role.iam_for_cloudwatch.arn}"
  target_arn = "${aws_kinesis_stream.kinesis_for_cloudwatch.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A name for the log destination
- `role_arn` - (Required) The ARN of an IAM role that grants Amazon CloudWatch Logs permissions to put data into the target
- `target_arn` - (Required) The ARN of the target Amazon Kinesis stream resource for the destination

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) specifying the log destination.

» Import

CloudWatch Logs destinations can be imported using the `name`, e.g.

```
$ terraform import aws_cloudwatch_log_destination.test_destination test_destination
```

» Resource: aws_cloudwatch_log_destination_policy

Provides a CloudWatch Logs destination policy resource.

» Example Usage

```
resource "aws_cloudwatch_log_destination" "test_destination" {
  name       = "test_destination"
  role_arn   = "${aws_iam_role.iam_for_cloudwatch.arn}"
  target_arn = "${aws_kinesis_stream.kinesis_for_cloudwatch.arn}"
}

data "aws_iam_policy_document" "test_destination_policy" {
  statement {
    effect = "Allow"

    principals {
      type = "AWS"

      identifiers = [
        "123456789012",
      ]
    }

    actions = [
      "logs:PutSubscriptionFilter",
    ]

    resources = [
      "${aws_cloudwatch_log_destination.test_destination.arn}",
    ]
  }
}

resource "aws_cloudwatch_log_destination_policy" "test_destination_policy" {
  destination_name = "${aws_cloudwatch_log_destination.test_destination.name}"
  access_policy    = "${data.aws_iam_policy_document.test_destination_policy.json}"
}
```

» Argument Reference

The following arguments are supported:

- `destination_name` - (Required) A name for the subscription filter
- `access_policy` - (Required) The policy document. This is a JSON formatted string.

» Import

CloudWatch Logs destination policies can be imported using the `destination_name`, e.g.

```
$ terraform import aws_cloudwatch_log_destination_policy.test_destination_policy test_destin
```

» Resource: aws_cloudwatch_log_group

Provides a CloudWatch Log Group resource.

» Example Usage

```
resource "aws_cloudwatch_log_group" "yada" {
  name = "Yada"

  tags = {
    Environment = "production"
    Application = "serviceA"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the log group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `retention_in_days` - (Optional) Specifies the number of days you want to retain log events in the specified log group.
- `kms_key_id` - (Optional) The ARN of the KMS Key to use when encrypting log data. Please note, after the AWS KMS CMK is disassociated from the log group, AWS CloudWatch Logs stops encrypting newly ingested

data for the log group. All previously ingested data remains encrypted, and AWS CloudWatch Logs requires permissions for the CMK whenever the encrypted data is requested.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) specifying the log group.

» Import

Cloudwatch Log Groups can be imported using the `name`, e.g.

```
$ terraform import aws_cloudwatch_log_group.test_group yada
```

» Resource: aws_cloudwatch_log_metric_filter

Provides a CloudWatch Log Metric Filter resource.

» Example Usage

```
resource "aws_cloudwatch_log_metric_filter" "yada" {
  name           = "MyAppAccessCount"
  pattern        = ""
  log_group_name = "${aws_cloudwatch_log_group.dada.name}"

  metric_transformation {
    name      = "EventCount"
    namespace = "YourNamespace"
    value     = "1"
  }
}

resource "aws_cloudwatch_log_group" "dada" {
  name = "MyApp/access.log"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A name for the metric filter.
- `pattern` - (Required) A valid CloudWatch Logs filter pattern for extracting metric data out of ingested log events.
- `log_group_name` - (Required) The name of the log group to associate the metric filter with.
- `metric_transformation` - (Required) A block defining collection of information needed to define how metric data gets emitted. See below.

The `metric_transformation` block supports the following arguments:

- `name` - (Required) The name of the CloudWatch metric to which the monitored log information should be published (e.g. `ErrorCount`)
- `namespace` - (Required) The destination namespace of the CloudWatch metric.
- `value` - (Required) What to publish to the metric. For example, if you're counting the occurrences of a particular term like "Error", the value will be "1" for each occurrence. If you're counting the bytes transferred the published value will be the value in the log event.
- `default_value` - (Optional) The value to emit when a filter pattern does not match a log event.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the metric filter.

» Resource: aws_cloudwatch_log_resource_policy

Provides a resource to manage a CloudWatch log resource policy.

» Example Usage

» Elasticsearch Log Publishing

```
data "aws_iam_policy_document" "elasticsearch-log-publishing-policy" {
  statement {
    actions = [
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:PutLogEventsBatch",
    ]

    resources = ["arn:aws:logs:*"]
    principals {
      identifiers = ["es.amazonaws.com"]
      type        = "Service"
    }
  }
}

resource "aws_cloudwatch_log_resource_policy" "elasticsearch-log-publishing-policy" {
  policy_document = "${data.aws_iam_policy_document.elasticsearch-log-publishing-policy.json
  policy_name     = "elasticsearch-log-publishing-policy"
}
```

» Route53 Query Logging

```
data "aws_iam_policy_document" "route53-query-logging-policy" {
  statement {
    actions = [
      "logs:CreateLogStream",
      "logs:PutLogEvents",
    ]

    resources = ["arn:aws:logs:*:*:log-group:/aws/route53/*"]

    principals {
      identifiers = ["route53.amazonaws.com"]
      type        = "Service"
    }
  }
}

resource "aws_cloudwatch_log_resource_policy" "route53-query-logging-policy" {
  policy_document = "${data.aws_iam_policy_document.route53-query-logging-policy.json}"
  policy_name     = "route53-query-logging-policy"
}
```

» Argument Reference

The following arguments are supported:

- policy_document - (Required) Details of the resource policy, including the identity of the principal that is enabled to put logs to this account. This is formatted as a JSON string. Maximum length of 5120 characters.
- policy_name - (Required) Name of the resource policy.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the CloudWatch log resource policy

» Import

CloudWatch log resource policies can be imported using the policy name, e.g.

```
$ terraform import aws_cloudwatch_log_resource_policy.MyPolicy MyPolicy
```

» Resource: aws_cloudwatch_log_stream

Provides a CloudWatch Log Stream resource.

» Example Usage

```
resource "aws_cloudwatch_log_group" "yada" {
  name = "Yada"
}

resource "aws_cloudwatch_log_stream" "foo" {
  name           = "SampleLogStream1234"
  log_group_name = "${aws_cloudwatch_log_group.yada.name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the log stream. Must not be longer than 512 characters and must not contain `:`
- `log_group_name` - (Required) The name of the log group under which the log stream is to be created.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) specifying the log stream.

» Resource: aws_cloudwatch_log_subscription_filter

Provides a CloudWatch Logs subscription filter resource.

» Example Usage

```
resource "aws_cloudwatch_log_subscription_filter" "test_lambdafunction_logfilter" {
  name            = "test_lambdafunction_logfilter"
  role_arn        = "${aws_iam_role.iam_for_lambda.arn}"
  log_group_name  = "/aws/lambda/example_lambda_name"
  filter_pattern  = "logtype test"
  destination_arn = "${aws_kinesis_stream.test_logstream.arn}"
  distribution    = "Random"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A name for the subscription filter
- `destination_arn` - (Required) The ARN of the destination to deliver matching log events to. Kinesis stream or Lambda function ARN.
- `filter_pattern` - (Required) A valid CloudWatch Logs filter pattern for subscribing to a filtered stream of log events.
- `log_group_name` - (Required) The name of the log group to associate the subscription filter with
- `role_arn` - (Optional) The ARN of an IAM role that grants Amazon CloudWatch Logs permissions to deliver ingested log events to the destination. If you use Lambda as a destination, you should skip this argument and use `aws_lambda_permission` resource for granting access from CloudWatch logs to the destination Lambda function.
- `distribution` - (Optional) The method used to distribute log data to the destination. By default log data is grouped by log stream, but the grouping can be set to random for a more even distribution. This property is only applicable when the destination is an Amazon Kinesis stream. Valid values are ”Random” and ”ByLogStream”.

» Attributes Reference

No extra attributes are exported.

» Import

CloudWatch Logs subscription filter can be imported using the log group name and subscription filter name separated by `|`.

```
$ terraform import aws_cloudwatch_log_subscription_filter.test_lambdafunction_logfilter /aws
```

» Resource: aws_cloudwatch_metric_alarm

Provides a CloudWatch Metric Alarm resource.

» Example Usage

```
resource "aws_cloudwatch_metric_alarm" "foobar" {
  alarm_name                = "terraform-test-foobar5"
  comparison_operator       = "GreaterThanOrEqualToThreshold"
  evaluation_periods        = "2"
  metric_name               = "CPUUtilization"
  namespace                 = "AWS/EC2"
  period                    = "120"
  statistic                 = "Average"
  threshold                 = "80"
  alarm_description         = "This metric monitors ec2 cpu utilization"
  insufficient_data_actions = []
}
```

» Example in Conjunction with Scaling Policies

```
resource "aws_autoscaling_policy" "bat" {
  name                   = "foobar3-terraform-test"
  scaling_adjustment     = 4
  adjustment_type        = "ChangeInCapacity"
  cooldown               = 300
  autoscaling_group_name = "${aws_autoscaling_group.bar.name}"
}

resource "aws_cloudwatch_metric_alarm" "bat" {
  alarm_name          = "terraform-test-foobar5"
  comparison_operator = "GreaterThanOrEqualToThreshold"
  evaluation_periods  = "2"
  metric_name         = "CPUUtilization"
  namespace           = "AWS/EC2"
  period              = "120"
  statistic           = "Average"
  threshold           = "80"

  dimensions = {
    AutoScalingGroupName = "${aws_autoscaling_group.bar.name}"
  }

  alarm_description = "This metric monitors ec2 cpu utilization"
  alarm_actions     = ["${aws_autoscaling_policy.bat.arn}"]
}
```

» Example with an Expression

```
resource "aws_cloudwatch_metric_alarm" "foobar" {
  alarm_name                = "terraform-test-foobar"
  comparison_operator       = "GreaterThanOrEqualToThreshold"
  evaluation_periods        = "2"
  threshold                 = "10"
  alarm_description         = "Request error rate has exceeded 10%"
  insufficient_data_actions = []

  metric_query {
    id          = "e1"
    expression  = "m2/m1*100"
    label       = "Error Rate"
    return_data = "true"
  }

  metric_query {
    id = "m1"

    metric {
      metric_name = "RequestCount"
      namespace   = "AWS/ApplicationELB"
      period      = "120"
      stat        = "Sum"
      unit        = "Count"

      dimensions = {
        LoadBalancer = "app/web"
      }
    }
  }

  metric_query {
    id = "m2"

    metric {
      metric_name = "HTTPCode_ELB_5XX_Count"
      namespace   = "AWS/ApplicationELB"
      period      = "120"
      stat        = "Sum"
      unit        = "Count"

      dimensions = {
        LoadBalancer = "app/web"
      }
    }
  }
}

resource "aws_cloudwatch_metric_alarm" "xx_anomaly_detection" {
  alarm_name                = "terraform-test-foobar"
  comparison_operator       = "GreaterThanUpperThreshold"
  evaluation_periods        = "2"
  threshold_metric_id       = "e1"
  alarm_description         = "This metric monitors ec2 cpu utilization"
  insufficient_data_actions = []

  metric_query {
    id          = "e1"
    expression  = "ANOMALY_DETECTION_BAND(m1)"
    label       = "CPUUtilization (Expected)"
    return_data = "true"
  }

  metric_query {
    id          = "m1"
    return_data = "true"
    metric {
      metric_name = "CPUUtilization"
      namespace   = "AWS/EC2"
      period      = "120"
      stat        = "Average"
      unit        = "Count"

      dimensions = {
        InstanceId = "i-abc123"
      }
    }
  }
```

```
}
```

» Example of monitoring Healthy Hosts on NLB using Target Group and NLB

```
resource "aws_cloudwatch_metric_alarm" "xxx_nlb_healthyhosts" {
  alarm_name          = "alarmname"
  comparison_operator = "LessThanThreshold"
  evaluation_periods  = "1"
  metric_name         = "HealthyHostCount"
  namespace           = "AWS/NetworkELB"
  period              = "60"
  statistic           = "Average"
  threshold           = var.logstash_servers_count
  alarm_description   = "Number of XXXX nodes healthy in Target Group"
  actions_enabled     = "true"
  alarm_actions       = [aws_sns_topic.sns.arn]
  ok_actions          = [aws_sns_topic.sns.arn]
  dimensions = {
    TargetGroup  = aws_lb_target_group.lb-tg.arn_suffix
    LoadBalancer = aws_lb.lb.arn_suffix
  }
}
```

NOTE: You cannot create a metric alarm consisting of both `statistic` and `extended_statistic` parameters. You must choose one or the other

» Argument Reference

See related part of AWS Docs for details about valid values.

The following arguments are supported:

- `alarm_name` - (Required) The descriptive name for the alarm. This name must be unique within the user's AWS account
- `comparison_operator` - (Required) The arithmetic operation to use when comparing the specified Statistic and Threshold. The specified Statistic value is used as the first operand. Either of the following is supported: `GreaterThanOrEqualToThreshold`, `GreaterThanThreshold`, `LessThanThreshold`, `LessThanOrEqualToThreshold`. Additionally, the values `LessThanLowerOrGreaterThanUpperThreshold`, `LessThanLowerThreshold`, and `GreaterThanUpperThreshold` are used only for alarms based on anomaly detection models.
- `evaluation_periods` - (Required) The number of periods over which data is compared to the specified threshold.

- `metric_name` - (Optional) The name for the alarm's associated metric. See docs for supported metrics.
- `namespace` - (Optional) The namespace for the alarm's associated metric. See docs for the list of namespaces. See docs for supported metrics.
- `period` - (Optional) The period in seconds over which the specified `statistic` is applied.
- `statistic` - (Optional) The statistic to apply to the alarm's associated metric. Either of the following is supported: `SampleCount`, `Average`, `Sum`, `Minimum`, `Maximum`
- `threshold` - (Optional) The value against which the specified statistic is compared. This parameter is required for alarms based on static thresholds, but should not be used for alarms based on anomaly detection models.
- `threshold_metric_id` - (Optional) If this is an alarm based on an anomaly detection model, make this value match the ID of the ANOMALY_DETECTION_BAND function.
- `actions_enabled` - (Optional) Indicates whether or not actions should be executed during any changes to the alarm's state. Defaults to `true`.
- `alarm_actions` - (Optional) The list of actions to execute when this alarm transitions into an ALARM state from any other state. Each action is specified as an Amazon Resource Name (ARN).
- `alarm_description` - (Optional) The description for the alarm.
- `datapoints_to_alarm` - (Optional) The number of datapoints that must be breaching to trigger the alarm.
- `dimensions` - (Optional) The dimensions for the alarm's associated metric. For the list of available dimensions see the AWS documentation here.
- `insufficient_data_actions` - (Optional) The list of actions to execute when this alarm transitions into an INSUFFICIENT_DATA state from any other state. Each action is specified as an Amazon Resource Name (ARN).
- `ok_actions` - (Optional) The list of actions to execute when this alarm transitions into an OK state from any other state. Each action is specified as an Amazon Resource Name (ARN).
- `unit` - (Optional) The unit for the alarm's associated metric.
- `extended_statistic` - (Optional) The percentile statistic for the metric associated with the alarm. Specify a value between p0.0 and p100.
- `treat_missing_data` - (Optional) Sets how this alarm is to handle missing data points. The following values are supported: `missing`, `ignore`, `breaching` and `notBreaching`. Defaults to `missing`.
- `evaluate_low_sample_count_percentiles` - (Optional) Used only for alarms based on percentiles. If you specify `ignore`, the alarm state will not change during periods with too few data points to be statistically significant. If you specify `evaluate` or omit this parameter, the alarm will always be evaluated and possibly change state no matter how many data points are available. The following values are supported: `ignore`, and `evaluate`.

- `metric_query` (Optional) Enables you to create an alarm based on a metric math expression. You may specify at most 20.
- `tags` - (Optional) A mapping of tags to assign to the resource.

NOTE: If you specify at least one `metric_query`, you may not specify a `metric_name`, `namespace`, `period` or `statistic`. If you do not specify a `metric_query`, you must specify each of these (although you may use `extended_statistic` instead of `statistic`).

» Nested fields

» `metric_query`

- `id` - (Required) A short name used to tie this object to the results in the response. If you are performing math expressions on this set of data, this name represents that data and can serve as a variable in the mathematical expression. The valid characters are letters, numbers, and underscore. The first character must be a lowercase letter.
- `expression` - (Optional) The math expression to be performed on the returned data, if this object is performing a math expression. This expression can use the id of the other metrics to refer to those metrics, and can also use the id of other expressions to use the result of those expressions. For more information about metric math expressions, see Metric Math Syntax and Functions in the Amazon CloudWatch User Guide.
- `label` - (Optional) A human-readable label for this metric or expression. This is especially useful if this is an expression, so that you know what the value represents.
- `return_data` (Optional) Specify exactly one `metric_query` to be `true` to use that `metric_query` result as the alarm.
- `metric` (Optional) The metric to be returned, along with statistics, period, and units. Use this parameter only if this object is retrieving a metric and not performing a math expression on returned data.

NOTE: You must specify either `metric` or `expression`. Not both.

» `metric`

- `dimensions` - (Optional) The dimensions for this metric. For the list of available dimensions see the AWS documentation here.
- `metric_name` - (Required) The name for this metric. See docs for supported metrics.
- `namespace` - (Required) The namespace for this metric. See docs for the list of namespaces. See docs for supported metrics.
- `period` - (Required) The period in seconds over which the specified `stat` is applied.

- `stat` - (Required) The statistic to apply to this metric. Either of the following is supported: `SampleCount`, `Average`, `Sum`, `Minimum`, `Maximum`
- `unit` - (Optional) The unit for this metric.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the cloudwatch metric alarm.
- `id` - The ID of the health check

» Import

Cloud Metric Alarms can be imported using the `alarm_name`, e.g.

```
$ terraform import aws_cloudwatch_metric_alarm.test alarm-12345
```

» Resource: aws_codebuild_project

Provides a CodeBuild Project resource. See also the `aws_codebuild_webhook` resource, which manages the webhook to the source (e.g. the "rebuild every time a code change is pushed" option in the CodeBuild web console).

» Example Usage

```
resource "aws_s3_bucket" "example" {
  bucket = "example"
  acl    = "private"
}

resource "aws_iam_role" "example" {
  name = "example"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "example" {
  role = "${aws_iam_role.example.name}"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": [
        "*"
      ],
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterfacePermission"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:network-interface/*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:Subnet": [
            "${aws_subnet.example1.arn}",
            "${aws_subnet.example2.arn}"
          ],
          "ec2:AuthorizedService": "codebuild.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "${aws_s3_bucket.example.arn}",
        "${aws_s3_bucket.example.arn}/*"
      ]
    }
  ]
}
POLICY
}

resource "aws_codebuild_project" "example" {
  name          = "test-project"
  description   = "test_codebuild_project"
  build_timeout = "5"
  service_role  = "${aws_iam_role.example.arn}"

  artifacts {
    type = "NO_ARTIFACTS"
  }

  cache {
    type     = "S3"
    location = "${aws_s3_bucket.example.bucket}"
  }

  environment {
    compute_type                = "BUILD_GENERAL1_SMALL"
    image                       = "aws/codebuild/standard:1.0"
    type                        = "LINUX_CONTAINER"
    image_pull_credentials_type = "CODEBUILD"
    environment_variable {
      name  = "SOME_KEY1"
      value = "SOME_VALUE1"
    }

    environment_variable {
      name  = "SOME_KEY2"
      value = "SOME_VALUE2"
      type  = "PARAMETER_STORE"
    }
  }

  logs_config {
    cloudwatch_logs {
      group_name = "log-group"
      stream_name = "log-stream"
    }

    s3_logs {
      status = "ENABLED"
      location = "${aws_s3_bucket.example.id}/build-log"
    }
  }

  source {
    type            = "GITHUB"
    location        = "https://github.com/mitchellh/packer.git"
    git_clone_depth = 1
  }

  vpc_config {
    vpc_id = "${aws_vpc.example.id}"

    subnets = [
      "${aws_subnet.example1.id}",
      "${aws_subnet.example2.id}",
    ]

    security_group_ids = [
      "${aws_security_group.example1.id}",
      "${aws_security_group.example2.id}",
    ]
  }

  tags = {
    Environment = "Test"
  }
}

resource "aws_codebuild_project" "project-with-cache" {
  name           = "test-project-cache"
  description    = "test_codebuild_project_cache"
  build_timeout  = "5"
  queued_timeout = "5"

  service_role  = "${aws_iam_role.example.arn}"

  artifacts {
    type = "NO_ARTIFACTS"
  }

  cache {
    type  = "LOCAL"
    modes = ["LOCAL_DOCKER_LAYER_CACHE", "LOCAL_SOURCE_CACHE"]
  }

  environment {
    compute_type                = "BUILD_GENERAL1_SMALL"
    image                       = "aws/codebuild/standard:1.0"
    type                        = "LINUX_CONTAINER"
    image_pull_credentials_type = "CODEBUILD"

    environment_variable {
      name  = "SOME_KEY1"
      value = "SOME_VALUE1"
    }
  }

  source {
    type            = "GITHUB"
    location        = "https://github.com/mitchellh/packer.git"
    git_clone_depth = 1
  }

  tags = {
    Environment = "Test"
  }
}
```

» Argument Reference

The following arguments are supported:

- `artifacts` - (Required) Information about the project's build output artifacts. Artifact blocks are documented below.
- `environment` - (Required) Information about the project's build environment. Environment blocks are documented below.
- `name` - (Required) The projects name.
- `source` - (Required) Information about the project's input source code. Source blocks are documented below.
- `badge_enabled` - (Optional) Generates a publicly-accessible URL for the projects build badge. Available as `badge_url` attribute when enabled.
- `build_timeout` - (Optional) How long in minutes, from 5 to 480 (8 hours), for AWS CodeBuild to wait until timing out any related build that does not get marked as completed. The default is 60 minutes.
- `queued_timeout` - (Optional) How long in minutes, from 5 to 480 (8 hours), a build is allowed to be queued before it times out. The default is 8 hours.
- `cache` - (Optional) Information about the cache storage for the project. Cache blocks are documented below.
- `description` - (Optional) A short description of the project.
- `encryption_key` - (Optional) The AWS Key Management Service (AWS KMS) customer master key (CMK) to be used for encrypting the build project's build output artifacts.
- `logs_config` - (Optional) Configuration for the builds to store log data to CloudWatch or S3.
- `service_role` - (Required) The Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role that enables AWS CodeBuild to interact with dependent AWS services on behalf of the AWS account.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `vpc_config` - (Optional) Configuration for the builds to run inside a VPC. VPC config blocks are documented below.
- `secondary_artifacts` - (Optional) A set of secondary artifacts to be used inside the build. Secondary artifacts blocks are documented below.
- `secondary_sources` - (Optional) A set of secondary sources to be used inside the build. Secondary sources blocks are documented below.

`artifacts` supports the following:

- `type` - (Required) The build output artifact's type. Valid values for this parameter are: `CODEPIPELINE`, `NO_ARTIFACTS` or `S3`.
- `artifact_identifier` - (Optional) The artifact identifier. Must be the same specified inside AWS CodeBuild buildspec.
- `encryption_disabled` - (Optional) If set to true, output artifacts will not be encrypted. If `type` is set to `NO_ARTIFACTS` then this value will be

ignored. Defaults to `false`.

- `override_artifact_name` (Optional) If set to true, a name specified in the build spec file overrides the artifact name.
- `location` - (Optional) Information about the build output artifact location. If `type` is set to `CODEPIPELINE` or `NO_ARTIFACTS` then this value will be ignored. If `type` is set to `S3`, this is the name of the output bucket.
- `name` - (Optional) The name of the project. If `type` is set to `S3`, this is the name of the output artifact object
- `namespace_type` - (Optional) The namespace to use in storing build artifacts. If `type` is set to `S3`, then valid values for this parameter are: `BUILD_ID` or `NONE`.
- `packaging` - (Optional) The type of build output artifact to create. If `type` is set to `S3`, valid values for this parameter are: `NONE` or `ZIP`
- `path` - (Optional) If `type` is set to `S3`, this is the path to the output artifact

`cache` supports the following:

- `type` - (Optional) The type of storage that will be used for the AWS CodeBuild project cache. Valid values: `NO_CACHE`, `LOCAL`, and `S3`. Defaults to `NO_CACHE`.
- `location` - (Required when cache type is `S3`) The location where the AWS CodeBuild project stores cached resources. For type `S3` the value must be a valid S3 bucket name/prefix.
- `modes` - (Required when cache type is `LOCAL`) Specifies settings that AWS CodeBuild uses to store and reuse build dependencies. Valid values: `LOCAL_SOURCE_CACHE`, `LOCAL_DOCKER_LAYER_CACHE`, and `LOCAL_CUSTOM_CACHE`

`environment` supports the following:

- `compute_type` - (Required) Information about the compute resources the build project will use. Available values for this parameter are: `BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE` or `BUILD_GENERAL1_2XLARGE`. `BUILD_GENERAL1_SMALL` is only valid if `type` is set to `LINUX_CONTAINER`. When `type` is set to `LINUX_GPU_CONTAINER`, `compute_type` need to be `BUILD_GENERAL1_LARGE`.
- `image` - (Required) The Docker image to use for this build project. Valid values include Docker images provided by CodeBuild (e.g `aws/codebuild/standard:2.0`), Docker Hub images (e.g. `hashicorp/terraform:latest`), and full Docker repository URIs such as those for ECR (e.g. `137112412989.dkr.ecr.us-west-2.amazonaws.com/amazonlinux:latest`).
- `type` - (Required) The type of build environment to use for related builds. Available values are: `LINUX_CONTAINER`, `LINUX_GPU_CONTAINER`, `WINDOWS_CONTAINER` or `ARM_CONTAINER`.
- `image_pull_credentials_type` - (Optional) The type of credentials AWS CodeBuild uses to pull images in your build. Available values for this parameter are `CODEBUILD` or `SERVICE_ROLE`. When you use a cross-account or private registry image, you must use SERVICE_ROLE

credentials. When you use an AWS CodeBuild curated image, you must use CODEBUILD credentials. Default to `CODEBUILD`

- `environment_variable` - (Optional) A set of environment variables to make available to builds for this build project.
- `privileged_mode` - (Optional) If set to true, enables running the Docker daemon inside a Docker container. Defaults to `false`.
- `certificate` - (Optional) The ARN of the S3 bucket, path prefix and object key that contains the PEM-encoded certificate.
- `registry_credential` - (Optional) Information about credentials for access to a private Docker registry. Registry Credential config blocks are documented below.

`environment_variable` supports the following:

- `name` - (Required) The environment variable's name or key.
- `value` - (Required) The environment variable's value.
- `type` - (Optional) The type of environment variable. Valid values: `PARAMETER_STORE`, `PLAINTEXT`.

`logs_config` supports the following:

- `cloudwatch_logs` - (Optional) Configuration for the builds to store logs to CloudWatch
- `s3_logs` - (Optional) Configuration for the builds to store logs to S3.

`cloudwatch_logs` supports the following:

- `status` - (Optional) Current status of logs in CloudWatch Logs for a build project. Valid values: `ENABLED`, `DISABLED`. Defaults to `ENABLED`.
- `group_name` - (Optional) The group name of the logs in CloudWatch Logs.
- `stream_name` - (Optional) The stream name of the logs in CloudWatch Logs.

`s3_logs` supports the following:

- `status` - (Optional) Current status of logs in S3 for a build project. Valid values: `ENABLED`, `DISABLED`. Defaults to `DISABLED`.
- `location` - (Optional) The name of the S3 bucket and the path prefix for S3 logs. Must be set if status is `ENABLED`, otherwise it must be empty.
- `encryption_disabled` - (Optional) Set to `true` if you do not want S3 logs encrypted. Defaults to `false`.

`source` supports the following:

- `type` - (Required) The type of repository that contains the source code to be built. Valid values for this parameter are: `CODECOMMIT`, `CODEPIPELINE`, `GITHUB`, `GITHUB_ENTERPRISE`, `BITBUCKET`, `S3` or `NO_SOURCE`.
- `auth` - (Optional) Information about the authorization settings for AWS CodeBuild to access the source code to be built. Auth blocks are documented below.

- `buildspec` - (Optional) The build spec declaration to use for this build project's related builds. This must be set when `type` is `NO_SOURCE`.
- `git_clone_depth` - (Optional) Truncate git history to this many commits.
- `insecure_ssl` - (Optional) Ignore SSL warnings when connecting to source control.
- `location` - (Optional) The location of the source code from git or s3.
- `report_build_status` - (Optional) Set to `true` to report the status of a build's start and finish to your source provider. This option is only valid when the `type` is `BITBUCKET` or `GITHUB`.

`auth` supports the following:

- `type` - (Required) The authorization type to use. The only valid value is `OAUTH`
- `resource` - (Optional) The resource value that applies to the specified authorization type.

`vpc_config` supports the following:

- `security_group_ids` - (Required) The security group IDs to assign to running builds.
- `subnets` - (Required) The subnet IDs within which to run builds.
- `vpc_id` - (Required) The ID of the VPC within which to run builds.

`registry_credential` supports the following:

- `credential` - (Required) The Amazon Resource Name (ARN) or name of credentials created using AWS Secrets Manager.
- `credential_provider` - (Required) The service that created the credentials to access a private Docker registry. The valid value, SECRETS_MANAGER, is for AWS Secrets Manager.

`secondary_artifacts` supports the following:

- `type` - (Required) The build output artifact's type. Valid values for this parameter are: `CODEPIPELINE`, `NO_ARTIFACTS` or `S3`.
- `artifact_identifier` - (Required) The artifact identifier. Must be the same specified inside AWS CodeBuild buildspec.
- `encryption_disabled` - (Optional) If set to true, output artifacts will not be encrypted. If `type` is set to `NO_ARTIFACTS` then this value will be ignored. Defaults to `false`.
- `override_artifact_name` (Optional) If set to true, a name specified in the build spec file overrides the artifact name.
- `location` - (Optional) Information about the build output artifact location. If `type` is set to `CODEPIPELINE` or `NO_ARTIFACTS` then this value will be ignored. If `type` is set to `S3`, this is the name of the output bucket. If `path` is not also specified, then `location` can also specify the path of the output artifact in the output bucket.
- `name` - (Optional) The name of the project. If `type` is set to `S3`, this is the name of the output artifact object

- namespace_type - (Optional) The namespace to use in storing build artifacts. If type is set to S3, then valid values for this parameter are: BUILD_ID or NONE.
- packaging - (Optional) The type of build output artifact to create. If type is set to S3, valid values for this parameter are: NONE or ZIP
- path - (Optional) If type is set to S3, this is the path to the output artifact

secondary_sources supports the following:

- type - (Required) The type of repository that contains the source code to be built. Valid values for this parameter are: CODECOMMIT, CODEPIPELINE, GITHUB, GITHUB_ENTERPRISE, BITBUCKET or S3.
- source_identifier - (Required) The source identifier. Source data will be put inside a folder named as this parameter inside AWS CodeBuild source directory
- auth - (Optional) Information about the authorization settings for AWS CodeBuild to access the source code to be built. Auth blocks are documented below.
- buildspec - (Optional) The build spec declaration to use for this build project's related builds.
- git_clone_depth - (Optional) Truncate git history to this many commits.
- insecure_ssl - (Optional) Ignore SSL warnings when connecting to source control.
- location - (Optional) The location of the source code from git or s3.
- report_build_status - (Optional) Set to true to report the status of a build's start and finish to your source provider. This option is only valid when your source provider is GITHUB, BITBUCKET, or GITHUB_ENTERPRISE.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The name (if imported via name) or ARN (if created via Terraform or imported via ARN) of the CodeBuild project.
- arn - The ARN of the CodeBuild project.
- badge_url - The URL of the build badge when badge_enabled is enabled.

» Import

CodeBuild Project can be imported using the name, e.g.

```
$ terraform import aws_codebuild_project.name project-name
```

» Resource: aws_codebuild_source_credential

Provides a CodeBuild Source Credentials Resource.

» Example Usage

```
resource "aws_codebuild_source_credential" "example" {
  auth_type = "PERSONAL_ACCESS_TOKEN"
  server_type = "GITHUB"
  token = "example"
}
```

» Bitbucket Server Usage

```
resource "aws_codebuild_source_credential" "example" {
  auth_type   = "BASIC_AUTH"
  server_type = "BITBUCKET"
  token       = "example"
  user_name   = "test-user"
}
```

» Argument Reference

The following arguments are supported:

- `auth_type` - (Required) The type of authentication used to connect to a GitHub, GitHub Enterprise, or Bitbucket repository. An OAUTH connection is not supported by the API.
- `server_type` - (Required) The source provider used for this project.
- `token` - (Required) For `GitHub` or `GitHub Enterprise`, this is the personal access token. For `Bitbucket`, this is the app password.
- `user_name` - (Optional) The Bitbucket username when the authType is `BASIC_AUTH`. This parameter is not valid for other types of source providers or connections.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ARN of Source Credential.
- `arn` - The ARN of Source Credential.

» Import

CodeBuild Source Credential can be imported using the CodeBuild Source Credential arn, e.g.

```
$ terraform import aws_codebuild_source_credential.example arn:aws:codebuild:us-west-2:12345
```

» Resource: aws_codebuild_webhook

Manages a CodeBuild webhook, which is an endpoint accepted by the CodeBuild service to trigger builds from source code repositories. Depending on the source type of the CodeBuild project, the CodeBuild service may also automatically create and delete the actual repository webhook as well.

» Example Usage

» Bitbucket and GitHub

When working with Bitbucket and GitHub source CodeBuild webhooks, the CodeBuild service will automatically create (on `aws_codebuild_webhook` resource creation) and delete (on `aws_codebuild_webhook` resource deletion) the Bitbucket/GitHub repository webhook using its granted OAuth permissions. This behavior cannot be controlled by Terraform.

Note: The AWS account that Terraform uses to create this resource *must* have authorized CodeBuild to access Bitbucket/GitHub's OAuth API in each applicable region. This is a manual step that must be done *before* creating webhooks with this resource. If OAuth is not configured, AWS will return an error similar to `ResourceNotFoundException: Could not find access token for server type github`. More information can be found in the CodeBuild User Guide for Bitbucket and GitHub.

Note: Further managing the automatically created Bitbucket/GitHub webhook with the `bitbucket_hook`/`github_repository_webhook` resource is only possible with importing that resource after creation of the `aws_codebuild_webhook` resource. The CodeBuild API does not ever provide the `secret` attribute for the `aws_codebuild_webhook` resource in this scenario.

```
resource "aws_codebuild_webhook" "example" {
  project_name = "${aws_codebuild_project.example.name}"

  filter_group {
    filter {
      type = "EVENT"
      pattern = "PUSH"
    }

    filter {
      type = "HEAD_REF"
      pattern = "master"
    }
  }
}
```

» GitHub Enterprise

When working with GitHub Enterprise source CodeBuild webhooks, the GHE repository webhook must be separately managed (e.g. manually or with the `github_repository_webhook` resource).

More information creating webhooks with GitHub Enterprise can be found in the CodeBuild User Guide.

```
resource "aws_codebuild_webhook" "example" {
  project_name = "${aws_codebuild_project.example.name}"
}

resource "github_repository_webhook" "example" {
  active     = true
  events     = ["push"]
  name       = "example"
  repository = "${github_repository.example.name}"

  configuration {
    url          = "${aws_codebuild_webhook.example.payload_url}"
    secret       = "${aws_codebuild_webhook.example.secret}"
    content_type = "json"
    insecure_ssl = false
  }
}
```

» Argument Reference

The following arguments are supported:

- `project_name` - (Required) The name of the build project.
- `branch_filter` - (Optional) A regular expression used to determine which branches get built. Default is all branches are built. It is recommended to use `filter_group` over `branch_filter`.
- `filter_group` - (Optional) Information about the webhook's trigger. Filter group blocks are documented below.

`filter_group` supports the following:

- `filter` - (Required) A webhook filter for the group. Filter blocks are documented below.

`filter` supports the following:

- `type` - (Required) The webhook filter group's type. Valid values for this parameter are: `EVENT`, `BASE_REF`, `HEAD_REF`, `ACTOR_ACCOUNT_ID`, `FILE_PATH`. At least one filter group must specify `EVENT` as its type.
- `pattern` - (Required) For a filter that uses `EVENT` type, a comma-separated string that specifies one event: `PUSH`, `PULL_REQUEST_CREATED`, `PULL_REQUEST_UPDATED`, `PULL_REQUEST_REOPENED`. `PULL_REQUEST_MERGED` works with GitHub & GitHub Enterprise only. For a filter that uses any of the other filter types, a regular expression.
- `exclude_matched_pattern` - (Optional) If set to `true`, the specified filter does *not* trigger a build. Defaults to `false`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the build project.
- `payload_url` - The CodeBuild endpoint where webhook events are sent.
- `secret` - The secret token of the associated repository. Not returned by the CodeBuild API for all source types.
- `url` - The URL to the webhook.

Note: The `secret` attribute is only set on resource creation, so if the secret is manually rotated, terraform will not pick up the change on subsequent runs. In that case, the webhook resource should be tainted and re-created to get the secret back in sync.

» Import

CodeBuild Webhooks can be imported using the CodeBuild Project name, e.g.

```
$ terraform import aws_codebuild_webhook.example MyProjectName
```

» Data Source: aws_codecommit_repository

The CodeCommit Repository data source allows the ARN, Repository ID, Repository URL for HTTP and Repository URL for SSH to be retrieved for an CodeCommit repository.

» Example Usage

```
data "aws_codecommit_repository" "test" {
  repository_name = "MyTestRepository"
}
```

» Argument Reference

The following arguments are supported:

- `repository_name` - (Required) The name for the repository. This needs to be less than 100 characters.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `repository_id` - The ID of the repository
- `arn` - The ARN of the repository
- `clone_url_http` - The URL to use for cloning the repository over HTTPS.
- `clone_url_ssh` - The URL to use for cloning the repository over SSH.

» Resource: aws_codecommit_repository

Provides a CodeCommit Repository Resource.

NOTE on CodeCommit Availability: The CodeCommit is not yet rolled out in all regions - available regions are listed the AWS Docs.

» Example Usage

```
resource "aws_codecommit_repository" "test" {
  repository_name = "MyTestRepository"
  description     = "This is the Sample App Repository"
}
```

» Argument Reference

The following arguments are supported:

- `repository_name` - (Required) The name for the repository. This needs to be less than 100 characters.

- `description` - (Optional) The description of the repository. This needs to be less than 1000 characters
- `default_branch` - (Optional) The default branch of the repository. The branch specified here needs to exist.
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `repository_id` - The ID of the repository
- `arn` - The ARN of the repository
- `clone_url_http` - The URL to use for cloning the repository over HTTPS.
- `clone_url_ssh` - The URL to use for cloning the repository over SSH.

» Import

Codecommit repository can be imported using repository name, e.g.

```
$ terraform import aws_codecommit_repository.imported ExistingRepo
```

» Resource: aws_codecommit_trigger

Provides a CodeCommit Trigger Resource.

NOTE on CodeCommit: The CodeCommit is not yet rolled out in all regions - available regions are listed the AWS Docs.

» Example Usage

```
resource "aws_codecommit_repository" "test" {
  repository_name = "test"
}

resource "aws_codecommit_trigger" "test" {
  repository_name = "${aws_codecommit_repository.test.repository_name}"

  trigger {
    name            = "all"
    events          = ["all"]
    destination_arn = "${aws_sns_topic.test.arn}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `repository_name` - (Required) The name for the repository. This needs to be less than 100 characters.
- `name` - (Required) The name of the trigger.
- `destination_arn` - (Required) The ARN of the resource that is the target for a trigger. For example, the ARN of a topic in Amazon Simple Notification Service (SNS).
- `custom_data` - (Optional) Any custom data associated with the trigger that will be included in the information sent to the target of the trigger.
- `branches` - (Optional) The branches that will be included in the trigger configuration. If no branches are specified, the trigger will apply to all branches.
- `events` - (Required) The repository events that will cause the trigger to run actions in another service, such as sending a notification through Amazon Simple Notification Service (SNS). If no events are specified, the trigger will run for all repository events. Event types include: `all`, `updateReference`, `createReference`, `deleteReference`.

» Resource: aws_codedeploy_app

Provides a CodeDeploy application to be used as a basis for deployments

» Example Usage

» ECS Application

```
resource "aws_codedeploy_app" "example" {
  compute_platform = "ECS"
  name             = "example"
}
```

» Lambda Application

```
resource "aws_codedeploy_app" "example" {
  compute_platform = "Lambda"
  name             = "example"
}
```

» Server Application

```
resource "aws_codedeploy_app" "example" {
  compute_platform = "Server"
  name             = "example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the application.
- `compute_platform` - (Optional) The compute platform can either be `ECS`, `Lambda`, or `Server`. Default is `Server`.

» Attribute Reference

The following arguments are exported:

- `id` - Amazon's assigned ID for the application.
- `name` - The application's name.

» Import

CodeDeploy Applications can be imported using the `name`, e.g.

```
$ terraform import aws_codedeploy_app.example my-application
```

» Resource: aws_codedeploy_deployment_config

Provides a CodeDeploy deployment config for an application

» Example Usage

» Server Usage

```
resource "aws_codedeploy_deployment_config" "foo" {
  deployment_config_name = "test-deployment-config"

  minimum_healthy_hosts {
    type  = "HOST_COUNT"
    value = 2
  }
}

resource "aws_codedeploy_deployment_group" "foo" {
  app_name               = "${aws_codedeploy_app.foo_app.name}"
  deployment_group_name  = "bar"
  service_role_arn       = "${aws_iam_role.foo_role.arn}"
  deployment_config_name = "${aws_codedeploy_deployment_config.foo.id}"

  ec2_tag_filter {
    key   = "filterkey"
    type  = "KEY_AND_VALUE"
    value = "filtervalue"
  }

  trigger_configuration {
    trigger_events     = ["DeploymentFailure"]
    trigger_name       = "foo-trigger"
    trigger_target_arn = "foo-topic-arn"
  }

  auto_rollback_configuration {
    enabled = true
    events  = ["DEPLOYMENT_FAILURE"]
  }

  alarm_configuration {
    alarms  = ["my-alarm-name"]
    enabled = true
  }
}
```

» Lambda Usage

```
resource "aws_codedeploy_deployment_config" "foo" {
  deployment_config_name = "test-deployment-config"
  compute_platform       = "Lambda"

  traffic_routing_config {
    type = "TimeBasedLinear"

    time_based_linear {
      interval   = 10
      percentage = 10
    }
  }
}

resource "aws_codedeploy_deployment_group" "foo" {
  app_name               = "${aws_codedeploy_app.foo_app.name}"
  deployment_group_name  = "bar"
  service_role_arn       = "${aws_iam_role.foo_role.arn}"
  deployment_config_name = "${aws_codedeploy_deployment_config.foo.id}"

  auto_rollback_configuration {
    enabled = true
    events  = ["DEPLOYMENT_STOP_ON_ALARM"]
  }

  alarm_configuration {
    alarms  = ["my-alarm-name"]
    enabled = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `deployment_config_name` - (Required) The name of the deployment config.
- `compute_platform` - (Optional) The compute platform can be `Server`, `Lambda`, or `ECS`. Default is `Server`.
- `minimum_healthy_hosts` - (Optional) A minimum_healthy_hosts block. Required for `Server` compute platform. Minimum Healthy Hosts are documented below.
- `traffic_routing_config` - (Optional) A traffic_routing_config block. Traffic Routing Config is documented below.

The `minimum_healthy_hosts` block supports the following:

- `type` - (Required) The type can either be `FLEET_PERCENT` or `HOST_COUNT`.
- `value` - (Required) The value when the type is `FLEET_PERCENT` represents the minimum number of healthy instances as a percentage of the total number of instances in the deployment. If you specify FLEET_PERCENT, at the start of the deployment, AWS CodeDeploy converts the percentage to the equivalent number of instance and rounds up fractional instances. When the type is `HOST_COUNT`, the value represents the minimum number of healthy instances as an absolute value.

The `traffic_routing_config` block supports the following:

- `type` - (Optional) Type of traffic routing config. One of `TimeBasedCanary`, `TimeBasedLinear`, `AllAtOnce`.
- `time_based_canary` - (Optional) The time based canary configuration information. If `type` is `TimeBasedLinear`, use `time_based_linear` instead.
- `time_based_linear` - (Optional) The time based linear configuration information. If `type` is `TimeBasedCanary`, use `time_based_canary` instead.

The `time_based_canary` block supports the following:

- `interval` - (Optional) The number of minutes between the first and second traffic shifts of a `TimeBasedCanary` deployment.
- `percentage` - (Optional) The percentage of traffic to shift in the first increment of a `TimeBasedCanary` deployment.

The `time_based_linear` block supports the following:

- `interval` - (Optional) The number of minutes between each incremental traffic shift of a `TimeBasedLinear` deployment.
- `percentage` - (Optional) The percentage of traffic that is shifted at the start of each increment of a `TimeBasedLinear` deployment.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The deployment group's config name.
- `deployment_config_id` - The AWS Assigned deployment config id

» Import

CodeDeploy Deployment Configurations can be imported using the `deployment_config_name`, e.g.

```
$ terraform import aws_codedeploy_app.example my-deployment-config
```

» Resource: aws_codedeploy_deployment_group

Provides a CodeDeploy Deployment Group for a CodeDeploy Application

NOTE on blue/green deployments: When using `green_fleet_provisioning_option` with the `COPY_AUTO_SCALING_GROUP` action, CodeDeploy will create a new ASG with a different name. This ASG is *not* managed by terraform and will conflict with existing configuration and state. You may want to use a different approach to managing deployments that involve multiple ASG, such as `DISCOVER_EXISTING` with separate blue and green ASG.

» Example Usage

```
resource "aws_iam_role" "example" {
  name = "example-role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "codedeploy.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy_attachment" "AWSCodeDeployRole" {
  policy_arn = "arn:aws:iam::aws:policy/service-role/AWSCodeDeployRole"
  role       = "${aws_iam_role.example.name}"
}

resource "aws_codedeploy_app" "example" {
  name = "example-app"
}

resource "aws_sns_topic" "example" {
  name = "example-topic"
}

resource "aws_codedeploy_deployment_group" "example" {
  app_name              = "${aws_codedeploy_app.example.name}"
  deployment_group_name = "example-group"
  service_role_arn      = "${aws_iam_role.example.arn}"

  ec2_tag_set {
    ec2_tag_filter {
      key   = "filterkey1"
      type  = "KEY_AND_VALUE"
      value = "filtervalue"
    }

    ec2_tag_filter {
      key   = "filterkey2"
      type  = "KEY_AND_VALUE"
      value = "filtervalue"
    }
  }

  trigger_configuration {
    trigger_events     = ["DeploymentFailure"]
    trigger_name       = "example-trigger"
    trigger_target_arn = "${aws_sns_topic.example.arn}"
  }

  auto_rollback_configuration {
    enabled = true
    events  = ["DEPLOYMENT_FAILURE"]
  }

  alarm_configuration {
    alarms  = ["my-alarm-name"]
    enabled = true
  }
}
```

» Blue Green Deployments with ECS

```
resource "aws_codedeploy_app" "example" {
  compute_platform = "ECS"
  name             = "example"
}

resource "aws_codedeploy_deployment_group" "example" {
  app_name               = "${aws_codedeploy_app.example.name}"
  deployment_config_name = "CodeDeployDefault.ECSAllAtOnce"
  deployment_group_name  = "example"
  service_role_arn       = "${aws_iam_role.example.arn}"

  auto_rollback_configuration {
    enabled = true
    events  = ["DEPLOYMENT_FAILURE"]
  }

  blue_green_deployment_config {
    deployment_ready_option {
      action_on_timeout = "CONTINUE_DEPLOYMENT"
    }

    terminate_blue_instances_on_deployment_success {
      action                           = "TERMINATE"
      termination_wait_time_in_minutes = 5
    }
  }

  deployment_style {
    deployment_option = "WITH_TRAFFIC_CONTROL"
    deployment_type   = "BLUE_GREEN"
  }

  ecs_service {
    cluster_name = "${aws_ecs_cluster.example.name}"
    service_name = "${aws_ecs_service.example.name}"
  }

  load_balancer_info {
    target_group_pair_info {
      prod_traffic_route {
        listener_arns = ["${aws_lb_listener.example.arn}"]
      }

      target_group {
        name = "${aws_lb_target_group.blue.name}"
      }

      target_group {
        name = "${aws_lb_target_group.green.name}"
      }
    }
  }
}
```

» Blue Green Deployments with Servers and Classic ELB

```
resource "aws_codedeploy_app" "example" {
  name = "example-app"
}

resource "aws_codedeploy_deployment_group" "example" {
  app_name              = "${aws_codedeploy_app.example.name}"
  deployment_group_name = "example-group"
  service_role_arn      = "${aws_iam_role.example.arn}"

  deployment_style {
    deployment_option = "WITH_TRAFFIC_CONTROL"
    deployment_type   = "BLUE_GREEN"
  }

  load_balancer_info {
    elb_info {
      name = "${aws_elb.example.name}"
    }
  }

  blue_green_deployment_config {
    deployment_ready_option {
      action_on_timeout    = "STOP_DEPLOYMENT"
      wait_time_in_minutes = 60
    }

    green_fleet_provisioning_option {
      action = "DISCOVER_EXISTING"
    }

    terminate_blue_instances_on_deployment_success {
      action = "KEEP_ALIVE"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_name` - (Required) The name of the application.
- `deployment_group_name` - (Required) The name of the deployment group.
- `service_role_arn` - (Required) The service role ARN that allows deployments.
- `alarm_configuration` - (Optional) Configuration block of alarms associated with the deployment group (documented below).
- `auto_rollback_configuration` - (Optional) Configuration block of the automatic rollback configuration associated with the deployment group (documented below).
- `autoscaling_groups` - (Optional) Autoscaling groups associated with the deployment group.

- `blue_green_deployment_config` - (Optional) Configuration block of the blue/green deployment options for a deployment group (documented below).
- `deployment_config_name` - (Optional) The name of the group's deployment config. The default is "CodeDeployDefault.OneAtATime".
- `deployment_style` - (Optional) Configuration block of the type of deployment, either in-place or blue/green, you want to run and whether to route deployment traffic behind a load balancer (documented below).
- `ec2_tag_filter` - (Optional) Tag filters associated with the deployment group. See the AWS docs for details.
- `ec2_tag_set` - (Optional) Configuration block(s) of Tag filters associated with the deployment group, which are also referred to as tag groups (documented below). See the AWS docs for details.
- `ecs_service` - (Optional) Configuration block(s) of the ECS services for a deployment group (documented below).
- `load_balancer_info` - (Optional) Single configuration block of the load balancer to use in a blue/green deployment (documented below).
- `on_premises_instance_tag_filter` - (Optional) On premise tag filters associated with the group. See the AWS docs for details.
- `trigger_configuration` - (Optional) Configuration block(s) of the triggers for the deployment group (documented below).

» alarm_configuration Argument Reference

You can configure a deployment to stop when a **CloudWatch** alarm detects that a metric has fallen below or exceeded a defined threshold. `alarm_configuration` supports the following:

- `alarms` - (Optional) A list of alarms configured for the deployment group. *A maximum of 10 alarms can be added to a deployment group.*
- `enabled` - (Optional) Indicates whether the alarm configuration is enabled. This option is useful when you want to temporarily deactivate alarm monitoring for a deployment group without having to add the same alarms again later.
- `ignore_poll_alarm_failure` - (Optional) Indicates whether a deployment should continue if information about the current state of alarms cannot be retrieved from CloudWatch. The default value is `false`.
  - `true`: The deployment will proceed even if alarm status information can't be retrieved.
  - `false`: The deployment will stop if alarm status information can't be retrieved.

Only one `alarm_configuration` is allowed.

» auto_rollback_configuration Argument Reference

You can configure a deployment group to automatically rollback when a deployment fails or when a monitoring threshold you specify is met. In this case, the last known good version of an application revision is deployed. `auto_rollback_configuration` supports the following:

- `enabled` - (Optional) Indicates whether a defined automatic rollback configuration is currently enabled for this Deployment Group. If you enable automatic rollback, you must specify at least one event type.
- `events` - (Optional) The event type or types that trigger a rollback. Supported types are `DEPLOYMENT_FAILURE` and `DEPLOYMENT_STOP_ON_ALARM`.

Only one `auto_rollback_configuration` is allowed.

» blue_green_deployment_config Argument Reference

You can configure options for a blue/green deployment. `blue_green_deployment_config` supports the following:

- `deployment_ready_option` - (Optional) Information about the action to take when newly provisioned instances are ready to receive traffic in a blue/green deployment (documented below).
- `green_fleet_provisioning_option` - (Optional) Information about how instances are provisioned for a replacement environment in a blue/green deployment (documented below).
- `terminate_blue_instances_on_deployment_success` - (Optional) Information about whether to terminate instances in the original fleet during a blue/green deployment (documented below).

Only one `blue_green_deployment_config` is allowed.

You can configure how traffic is rerouted to instances in a replacement environment in a blue/green deployment. `deployment_ready_option` supports the following:

- `action_on_timeout` - (Optional) When to reroute traffic from an original environment to a replacement environment in a blue/green deployment.
  - `CONTINUE_DEPLOYMENT`: Register new instances with the load balancer immediately after the new application revision is installed on the instances in the replacement environment.
  - `STOP_DEPLOYMENT`: Do not register new instances with load balancer unless traffic is rerouted manually. If traffic is not rerouted manually before the end of the specified wait period, the deployment status is changed to Stopped.
- `wait_time_in_minutes` - (Optional) The number of minutes to wait before the status of a blue/green deployment changed to Stopped if rerouting

is not started manually. Applies only to the `STOP_DEPLOYMENT` option for `action_on_timeout`.

You can configure how instances will be added to the replacement environment in a blue/green deployment. `green_fleet_provisioning_option` supports the following:

- `action` - (Optional) The method used to add instances to a replacement environment.
  - `DISCOVER_EXISTING`: Use instances that already exist or will be created manually.
  - `COPY_AUTO_SCALING_GROUP`: Use settings from a specified **Auto Scaling** group to define and create instances in a new Auto Scaling group. *Exactly one Auto Scaling group must be specified* when selecting `COPY_AUTO_SCALING_GROUP`. Use `autoscaling_groups` to specify the Auto Scaling group.

You can configure how instances in the original environment are terminated when a blue/green deployment is successful. `terminate_blue_instances_on_deployment_success` supports the following:

- `action` - (Optional) The action to take on instances in the original environment after a successful blue/green deployment.
  - `TERMINATE`: Instances are terminated after a specified wait time.
  - `KEEP_ALIVE`: Instances are left running after they are deregistered from the load balancer and removed from the deployment group.
- `termination_wait_time_in_minutes` - (Optional) The number of minutes to wait after a successful blue/green deployment before terminating instances from the original environment.

» deployment_style Argument Reference

You can configure the type of deployment, either in-place or blue/green, you want to run and whether to route deployment traffic behind a load balancer. `deployment_style` supports the following:

- `deployment_option` - (Optional) Indicates whether to route deployment traffic behind a load balancer. Valid Values are `WITH_TRAFFIC_CONTROL` or `WITHOUT_TRAFFIC_CONTROL`.
- `deployment_type` - (Optional) Indicates whether to run an in-place deployment or a blue/green deployment. Valid Values are `IN_PLACE` or `BLUE_GREEN`.

Only one `deployment_style` is allowed.

» ec2_tag_filter Argument Reference

The `ec2_tag_filter` configuration block supports the following:

- `key` - (Optional) The key of the tag filter.
- `type` - (Optional) The type of the tag filter, either `KEY_ONLY`, `VALUE_ONLY`, or `KEY_AND_VALUE`.
- `value` - (Optional) The value of the tag filter.

Multiple occurrences of `ec2_tag_filter` are allowed, where any instance that matches to at least one of the tag filters is selected.

» ec2_tag_set Argument Reference

You can form a tag group by putting a set of tag filters into `ec2_tag_set`. If multiple tag groups are specified, any instance that matches to at least one tag filter of every tag group is selected.

» ecs_service Argument Reference

Each `ecs_service` configuration block supports the following:

- `cluster_name` - (Required) The name of the ECS cluster.
- `service_name` - (Required) The name of the ECS service.

» load_balancer_info Argument Reference

You can configure the **Load Balancer** to use in a deployment. `load_balancer_info` supports the following:

- `elb_info` - (Optional) The Classic Elastic Load Balancer to use in a deployment. Conflicts with `target_group_info` and `target_group_pair_info`.
- `target_group_info` - (Optional) The (Application/Network Load Balancer) target group to use in a deployment. Conflicts with `elb_info` and `target_group_pair_info`.
- `target_group_pair_info` - (Optional) The (Application/Network Load Balancer) target group pair to use in a deployment. Conflicts with `elb_info` and `target_group_info`.

» load_balancer_info elb_info Argument Reference

The `elb_info` configuration block supports the following:

- `name` - (Optional) The name of the load balancer that will be used to route traffic from original instances to replacement instances in a blue/green deployment. For in-place deployments, the name of the load balancer that instances are deregistered from so they are not serving traffic during a deployment, and then re-registered with after the deployment completes.

» load_balancer_info target_group_info Argument Reference

The `target_group_info` configuration block supports the following:

- `name` - (Optional) The name of the target group that instances in the original environment are deregistered from, and instances in the replacement environment registered with. For in-place deployments, the name of the target group that instances are deregistered from, so they are not serving traffic during a deployment, and then re-registered with after the deployment completes.

» load_balancer_info target_group_pair_info Argument Reference

The `target_group_pair_info` configuration block supports the following:

- `prod_traffic_route` - (Required) Configuration block for the production traffic route (documented below).
- `target_group` - (Required) Configuration blocks for a target group within a target group pair (documented below).
- `test_traffic_route` - (Optional) Configuration block for the test traffic route (documented below).

» load_balancer_info target_group_pair_info prod_traffic_route Argument Reference

The `prod_traffic_route` configuration block supports the following:

- `listener_arns` - (Required) List of Amazon Resource Names (ARNs) of the load balancer listeners.

» load_balancer_info target_group_pair_info target_group Argument Reference

The `target_group` configuration block supports the following:

- `name` - (Required) Name of the target group.

» load_balancer_info target_group_pair_info test_traffic_route Argument Reference

The `test_traffic_route` configuration block supports the following:

- `listener_arns` - (Required) List of Amazon Resource Names (ARNs) of the load balancer listeners.

» on_premises_tag_filter Argument Reference

The `on_premises_tag_filter` configuration block supports the following:

- `key` - (Optional) The key of the tag filter.
- `type` - (Optional) The type of the tag filter, either `KEY_ONLY`, `VALUE_ONLY`, or `KEY_AND_VALUE`.
- `value` - (Optional) The value of the tag filter.

» trigger_configuration Argument Reference

Add triggers to a Deployment Group to receive notifications about events related to deployments or instances in the group. Notifications are sent to subscribers of the **SNS** topic associated with the trigger. *CodeDeploy must have permission to publish to the topic from this deployment group.* `trigger_configuration` supports the following:

- `trigger_events` - (Required) The event type or types for which notifications are triggered. Some values that are supported: `DeploymentStart`, `DeploymentSuccess`, `DeploymentFailure`, `DeploymentStop`, `DeploymentRollback`, `InstanceStart`, `InstanceSuccess`, `InstanceFailure`. See the CodeDeploy documentation for all possible values.
- `trigger_name` - (Required) The name of the notification trigger.
- `trigger_target_arn` - (Required) The ARN of the SNS topic through which notifications are sent.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Application name and deployment group name.

» Import

CodeDeploy Deployment Groups can be imported by their `app_name`, a colon, and `deployment_group_name`, e.g.

```
$ terraform import aws_codedeploy_deployment_group.example my-application:my-deployment-grou
```

» Resource: aws_codepipeline

Provides a CodePipeline.

NOTE on `aws_codepipeline`: - the `GITHUB_TOKEN` environment variable must be set if the GitHub provider is specified.

» Example Usage

```
resource "aws_s3_bucket" "codepipeline_bucket" {
  bucket = "test-bucket"
  acl    = "private"
}

resource "aws_iam_role" "codepipeline_role" {
  name = "test-role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codepipeline.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "codepipeline_policy" {
  name = "codepipeline_policy"
  role = "${aws_iam_role.codepipeline_role.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketVersioning",
        "s3:PutObject"
      ],
      "Resource": [
        "${aws_s3_bucket.codepipeline_bucket.arn}",
        "${aws_s3_bucket.codepipeline_bucket.arn}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:BatchGetBuilds",
        "codebuild:StartBuild"
      ],
      "Resource": "*"
    }
  ]
}
EOF
}

data "aws_kms_alias" "s3kmskey" {
  name = "alias/myKmsKey"
}

resource "aws_codepipeline" "codepipeline" {
  name     = "tf-test-pipeline"
  role_arn = "${aws_iam_role.codepipeline_role.arn}"

  artifact_store {
    location = "${aws_s3_bucket.codepipeline_bucket.bucket}"
    type     = "S3"

    encryption_key {
      id   = "${data.aws_kms_alias.s3kmskey.arn}"
      type = "KMS"
    }
  }

  stage {
    name = "Source"

    action {
      name             = "Source"
      category         = "Source"
      owner            = "ThirdParty"
      provider         = "GitHub"
      version          = "1"
      output_artifacts = ["source_output"]
      configuration = {
        Owner  = "my-organization"
        Repo   = "test"
        Branch = "master"
      }
    }
  }

  stage {
    name = "Build"

    action {
      name             = "Build"
      category         = "Build"
      owner            = "AWS"
      provider         = "CodeBuild"
      input_artifacts  = ["source_output"]
      output_artifacts = ["build_output"]
      version          = "1"

      configuration = {
        ProjectName = "test"
      }
    }
  }

  stage {
    name = "Deploy"

    action {
      name            = "Deploy"
      category        = "Deploy"
      owner           = "AWS"
      provider        = "CloudFormation"
      input_artifacts = ["build_output"]
      version         = "1"

      configuration = {
        ActionMode     = "REPLACE_ON_FAILURE"
        Capabilities   = "CAPABILITY_AUTO_EXPAND,CAPABILITY_IAM"
        OutputFileName = "CreateStackOutput.json"
        StackName      = "MyStack"
        TemplatePath   = "build_output::sam-templated.yaml"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the pipeline.
- `role_arn` - (Required) A service role Amazon Resource Name (ARN) that grants AWS CodePipeline permission to make calls to AWS services on your behalf.
- `artifact_store` (Required) An artifact_store block. Artifact stores are documented below.
- `stage` (Minimum of at least two `stage` blocks is required) A stage block. Stages are documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

An `artifact_store` block supports the following arguments:

- `location` - (Required) The location where AWS CodePipeline stores artifacts for a pipeline, such as an S3 bucket.
- `type` - (Required) The type of the artifact store, such as Amazon S3
- `encryption_key` - (Optional) The encryption key block AWS CodePipeline uses to encrypt the data in the artifact store, such as an AWS Key Management Service (AWS KMS) key. If you don't specify a key, AWS CodePipeline uses the default key for Amazon Simple Storage Service (Amazon S3). An `encryption_key` block is documented below.

An `encryption_key` block supports the following arguments:

- `id` - (Required) The KMS key ARN or ID
- `type` - (Required) The type of key; currently only `KMS` is supported

A `stage` block supports the following arguments:

- `name` - (Required) The name of the stage.
- `action` - (Required) The action(s) to include in the stage. Defined as an `action` block below

A `action` block supports the following arguments:

- `category` - (Required) A category defines what kind of action can be taken in the stage, and constrains the provider type for the action. Possible values are `Approval`, `Build`, `Deploy`, `Invoke`, `Source` and `Test`.
- `owner` - (Required) The creator of the action being called. Possible values are `AWS`, `Custom` and `ThirdParty`.
- `name` - (Required) The action declaration's name.
- `provider` - (Required) The provider of the service being called by the action. Valid providers are determined by the action category. For exam-

ple, an action in the Deploy category type might have a provider of AWS CodeDeploy, which would be specified as CodeDeploy.

- `version` - (Required) A string that identifies the action type.
- `configuration` - (Optional) A Map of the action declaration's configuration. Find out more about configuring action configurations in the Reference Pipeline Structure documentation.
- `input_artifacts` - (Optional) A list of artifact names to be worked on.
- `output_artifacts` - (Optional) A list of artifact names to output. Output artifact names must be unique within a pipeline.
- `role_arn` - (Optional) The ARN of the IAM service role that will perform the declared action. This is assumed through the roleArn for the pipeline.
- `run_order` - (Optional) The order in which actions are run.

Note: The input artifact of an action must exactly match the output artifact declared in a preceding action, but the input artifact does not have to be the next action in strict sequence from the action that provided the output artifact. Actions in parallel can declare different output artifacts, which are in turn consumed by different following actions.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The codepipeline ID.
- `arn` - The codepipeline ARN.

» Import

CodePipelines can be imported using the name, e.g.

```
$ terraform import aws_codepipeline.foo example
```

» Resource: aws_codepipeline_webhook

Provides a CodePipeline Webhook.

» Example Usage

```
resource "aws_codepipeline" "bar" {
  name     = "tf-test-pipeline"
  role_arn = "${aws_iam_role.bar.arn}"

  artifact_store {
    location = "${aws_s3_bucket.bar.bucket}"
    type     = "S3"

    encryption_key {
      id   = "${data.aws_kms_alias.s3kmskey.arn}"
      type = "KMS"
    }
  }

  stage {
    name = "Source"

    action {
      name             = "Source"
      category         = "Source"
      owner            = "ThirdParty"
      provider         = "GitHub"
      version          = "1"
      output_artifacts = ["test"]

      configuration = {
        Owner  = "my-organization"
        Repo   = "test"
        Branch = "master"
      }
    }
  }

  stage {
    name = "Build"

    action {
      name            = "Build"
      category        = "Build"
      owner           = "AWS"
      provider        = "CodeBuild"
      input_artifacts = ["test"]
      version         = "1"

      configuration = {
        ProjectName = "test"
      }
    }
  }
}
```

```
# A shared secret between GitHub and AWS that allows AWS
# CodePipeline to authenticate the request came from GitHub.
# Would probably be better to pull this from the environment
# or something like SSM Parameter Store.
locals {
  webhook_secret = "super-secret"
}

resource "aws_codepipeline_webhook" "bar" {
  name            = "test-webhook-github-bar"
  authentication  = "GITHUB_HMAC"
  target_action   = "Source"
  target_pipeline = "${aws_codepipeline.bar.name}"

  authentication_configuration {
    secret_token = "${local.webhook_secret}"
  }

  filter {
    json_path    = "$.ref"
    match_equals = "refs/heads/{Branch}"
  }
}

# Wire the CodePipeline webhook into a GitHub repository.
resource "github_repository_webhook" "bar" {
  repository = "${github_repository.repo.name}"

  name = "web"

  configuration {
    url          = "${aws_codepipeline_webhook.bar.url}"
    content_type = "json"
    insecure_ssl = true
    secret       = "${local.webhook_secret}"
  }

  events = ["push"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the webhook.

- `authentication` - (Required) The type of authentication to use. One of `IP`, `GITHUB_HMAC`, or `UNAUTHENTICATED`.
- `authentication_configuration` - (Optional) An `auth` block. Required for `IP` and `GITHUB_HMAC`. Auth blocks are documented below.
- `filter` (Required) One or more `filter` blocks. Filter blocks are documented below.
- `target_action` - (Required) The name of the action in a pipeline you want to connect to the webhook. The action must be from the source (first) stage of the pipeline.
- `target_pipeline` - (Required) The name of the pipeline.
- `tags` - (Optional) A mapping of tags to assign to the resource.

An `authentication_configuration` block supports the following arguments:

- `secret_token` - (Optional) The shared secret for the GitHub repository webhook. Set this as `secret` in your `github_repository_webhook`'s `configuration` block. Required for `GITHUB_HMAC`.
- `allowed_ip_range` - (Optional) A valid CIDR block for `IP` filtering. Required for `IP`.

A `filter` block supports the following arguments:

- `json_path` - (Required) The JSON path to filter on.
- `match_equals` - (Required) The value to match on (e.g. `refs/heads/{Branch}`). See AWS docs for details.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The CodePipeline webhook's ARN.
- `url` - The CodePipeline webhook's URL. POST events to this endpoint to trigger the target.

» Import

CodePipeline Webhooks can be imported by their ARN, e.g.

```
$ terraform import aws_codepipeline_webhook.example arn:aws:codepipeline:us-west-2:123456789
```

» Data Source: aws_cognito_user_pools

Use this data source to get a list of cognito user pools.

» Example Usage

```
data "aws_api_gateway_rest_api" "selected" {
  name = "${var.api_gateway_name}"
}

data "aws_cognito_user_pools" "selected" {
  name = "${var.cognito_user_pool_name}"
}

resource "aws_api_gateway_authorizer" "cognito" {
  name          = "cognito"
  type          = "COGNITO_USER_POOLS"
  rest_api_id   = "${data.aws_api_gateway_rest_api.selected.id}"
  provider_arns = ["${data.aws_cognito_user_pools.selected.arns}"]
}
```

» Argument Reference

- `name` - (required) Name of the cognito user pools. Name is not a unique attribute for cognito user pool, so multiple pools might be returned with given name.

» Attributes Reference

- `ids` - The list of cognito user pool ids.

» Resource: aws_cognito_identity_pool

Provides an AWS Cognito Identity Pool.

» Example Usage

```
resource "aws_iam_saml_provider" "default" {
  name                   = "my-saml-provider"
  saml_metadata_document = "${file("saml-metadata.xml")}"
}

resource "aws_cognito_identity_pool" "main" {
  identity_pool_name               = "identity pool"
  allow_unauthenticated_identities = false
  cognito_identity_providers {
    client_id               = "6lhlkkfbfb4q5kpp90urffae"
    provider_name           = "cognito-idp.us-east-1.amazonaws.com/us-east-1_Tv0493apJ"
    server_side_token_check = false
  }

  cognito_identity_providers {
    client_id               = "7kodkvfqfb4qfkp39eurffae"
    provider_name           = "cognito-idp.us-east-1.amazonaws.com/eu-west-1_Zr231apJu"
    server_side_token_check = false
  }

  supported_login_providers = {
    "graph.facebook.com"  = "7346241598935552"
    "accounts.google.com" = "123456789012.apps.googleusercontent.com"
  }

  saml_provider_arns           = ["${aws_iam_saml_provider.default.arn}"]
  openid_connect_provider_arns = ["arn:aws:iam::123456789012:oidc-provider/foo.example.com"]
}
```

» Argument Reference

The Cognito Identity Pool argument layout is a structure composed of several sub-resources - these resources are laid out below.

- `identity_pool_name` (Required) - The Cognito Identity Pool name.
- `allow_unauthenticated_identities` (Required) - Whether the identity pool supports unauthenticated logins or not.
- `developer_provider_name` (Optional) - The ”domain” by which Cognito will refer to your users. This name acts as a placeholder that allows your backend and the Cognito service to communicate about the developer provider.
- `cognito_identity_providers` (Optional) - An array of Amazon Cognito Identity user pools and their client IDs.
- `openid_connect_provider_arns` (Optional) - A list of OpendID Connect provider ARNs.
- `saml_provider_arns` (Optional) - An array of Amazon Resource Names (ARNs) of the SAML provider for your identity.
- `supported_login_providers` (Optional) - Key-Value pairs mapping provider names to provider app IDs.
- `tags` - (Optional) A mapping of tags to assign to the Identity Pool.

» Cognito Identity Providers

- `client_id` (Optional) - The client ID for the Amazon Cognito Identity User Pool.
- `provider_name` (Optional) - The provider name for an Amazon Cognito Identity User Pool.
- `server_side_token_check` (Optional) - Whether server-side token validation is enabled for the identity provider's token or not.

» Attributes Reference

In addition to the arguments, which are exported, the following attributes are exported:

- `id` - An identity pool ID in the format REGION:GUID.
- `arn` - The ARN of the identity pool.

» Import

Cognito Identity Pool can be imported using the name, e.g.

```
$ terraform import aws_cognito_identity_pool.mypool <identity-pool-id>
```

» Resource: aws_cognito_identity_pool_roles_attachment

Provides an AWS Cognito Identity Pool Roles Attachment.

» Example Usage

```
resource "aws_cognito_identity_pool" "main" {
  identity_pool_name               = "identity pool"
  allow_unauthenticated_identities = false

  supported_login_providers = {
    "graph.facebook.com" = "7346241598935555"
  }
}

resource "aws_iam_role" "authenticated" {
  name = "cognito_authenticated"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "${aws_cognito_identity_pool.main.id}"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "authenticated" {
  name = "authenticated_policy"
  role = "${aws_iam_role.authenticated.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "mobileanalytics:PutEvents",
        "cognito-sync:*",
        "cognito-identity:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
EOF
}

resource "aws_cognito_identity_pool_roles_attachment" "main" {
  identity_pool_id = "${aws_cognito_identity_pool.main.id}"

  role_mapping {
    identity_provider         = "graph.facebook.com"
    ambiguous_role_resolution = "AuthenticatedRole"
    type                      = "Rules"

    mapping_rule {
      claim      = "isAdmin"
      match_type = "Equals"
      role_arn   = "${aws_iam_role.authenticated.arn}"
      value      = "paid"
    }
  }

  roles = {
    "authenticated" = "${aws_iam_role.authenticated.arn}"
  }
}
```

» Argument Reference

The Cognito Identity Pool Roles Attachment argument layout is a structure composed of several sub-resources - these resources are laid out below.

- `identity_pool_id` (Required) - An identity pool ID in the format REGION:GUID.
- `role_mapping` (Optional) - A List of Role Mapping.
- `roles` (Required) - The map of roles associated with this pool. For a given role, the key will be either "authenticated" or "unauthenticated" and the value will be the Role ARN.

» Role Mappings

- `identity_provider` (Required) - A string identifying the identity provider, for example, "graph.facebook.com" or "cognito-idp.us-east-1.amazonaws.com/us-east-1_abcdefghi:app_client_id".
- `ambiguous_role_resolution` (Optional) - Specifies the action to be taken if either no rules match the claim value for the Rules type, or there is no cognito:preferred_role claim and there are multiple cognito:roles matches for the Token type. `Required` if you specify Token or Rules as the Type.
- `mapping_rule` (Optional) - The Rules Configuration to be used for mapping users to roles. You can specify up to 25 rules per identity provider. Rules are evaluated in order. The first one to match specifies the role.
- `type` (Required) - The role mapping type.

» Rules Configuration

- `claim` (Required) - The claim name that must be present in the token, for example, ”isAdmin” or ”paid”.
- `match_type` (Required) - The match condition that specifies how closely the claim value in the IdP token must match Value.
- `role_arn` (Required) - The role ARN.
- `value` (Required) - A brief string that the claim must match, for example, ”paid” or ”yes”.

» Attributes Reference

In addition to the arguments, which are exported, the following attributes are exported:

- `id` - The identity pool ID.
- `identity_pool_id` (Required) - An identity pool ID in the format REGION:GUID.
- `role_mapping` (Optional) - The List of Role Mapping.
- `roles` (Required) - The map of roles associated with this pool. For a given role, the key will be either ”authenticated” or ”unauthenticated” and the value will be the Role ARN.

» Resource: aws_cognito_identity_provider

Provides a Cognito User Identity Provider resource.

» Example Usage

```
resource "aws_cognito_user_pool" "example" {
  name                     = "example-pool"
  auto_verified_attributes = ["email"]
}

resource "aws_cognito_identity_provider" "example_provider" {
  user_pool_id  = "${aws_cognito_user_pool.example.id}"
  provider_name = "Google"
  provider_type = "Google"

  provider_details = {
    authorize_scopes = "email"
    client_id        = "your client_id"
    client_secret    = "your client_secret"
  }

  attribute_mapping = {
    email    = "email"
    username = "sub"
  }
}
```

» Argument Reference

The following arguments are supported:

- `user_pool_id` (Required) - The user pool id
- `provider_name` (Required) - The provider name
- `provider_type` (Required) - The provider type. See AWS API for valid values
- `attribute_mapping` (Optional) - The map of attribute mapping of user pool attributes. AttributeMapping in AWS API documentation
- `idp_identifiers` (Optional) - The list of identity providers.
- `provider_details` (Optional) - The map of identity details, such as access token

» Import

`aws_cognito_identity_provider` resources can be imported using their User Pool ID and Provider Name, e.g.

```
$ terraform import aws_cognito_identity_provider.example xxx_yyyyy:example
```

» Resource: aws_cognito_resource_server

Provides a Cognito Resource Server.

» Example Usage

» Create a basic resource server

```
resource "aws_cognito_user_pool" "pool" {
  name = "pool"
}

resource "aws_cognito_resource_server" "resource" {
  identifier = "https://example.com"
  name       = "example"

  user_pool_id = "${aws_cognito_user_pool.pool.id}"
}
```

» Create a resource server with sample-scope

```
resource "aws_cognito_user_pool" "pool" {
  name = "pool"
}

resource "aws_cognito_resource_server" "resource" {
  identifier = "https://example.com"
  name       = "example"

  scope {
    scope_name        = "sample-scope"
    scope_description = "a Sample Scope Description"
  }

  user_pool_id = "${aws_cognito_user_pool.pool.id}"
}
```

» Argument Reference

The following arguments are supported:

- `identifier` - (Required) An identifier for the resource server.
- `name` - (Required) A name for the resource server.
- `scope` - (Optional) A list of Authorization Scope.

» Authorization Scope

- `scope_name` - (Required) The scope name.
- `scope_description` - (Required) The scope description.

» Attribute Reference

In addition to the arguments, which are exported, the following attributes are exported:

- `scope_identifiers` - A list of all scopes configured for this resource server in the format identifier/scope_name.

» Import

`aws_cognito_resource_server` can be imported using their User Pool ID and Identifier, e.g.

```
$ terraform import aws_cognito_resource_server.example xxx_yyyyy|https://example.com
```

» Resource: aws_cognito_user_group

Provides a Cognito User Group resource.

» Example Usage

```
resource "aws_cognito_user_pool" "main" {
  name = "identity pool"
}

resource "aws_iam_role" "group_role" {
  name = "user-group-role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-dead-beef-cafe-123456790
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
EOF
}
```

```
resource "aws_cognito_user_group" "main" {
  name         = "user-group"
  user_pool_id = "${aws_cognito_user_pool.main.id}"
  description  = "Managed by Terraform"
  precedence   = 42
  role_arn     = "${aws_iam_role.group_role.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the user group.
- `user_pool_id` - (Required) The user pool ID.
- `description` - (Optional) The description of the user group.
- `precedence` - (Optional) The precedence of the user group.
- `role_arn` - (Optional) The ARN of the IAM role to be associated with the user group.

» Import

Cognito User Groups can be imported using the `user_pool_id/name` attributes concatenated, e.g.

```
$ terraform import aws_cognito_user_group.group us-east-1_vG78M4goG/user-group
```

» Resource: aws_cognito_user_pool

Provides a Cognito User Pool resource.

» Example Usage

» Basic configuration

```
resource "aws_cognito_user_pool" "pool" {
  name = "mypool"
}
```

» Argument Reference

The following arguments are supported:

- admin_create_user_config (Optional) - The configuration for AdminCreateUser requests.
- alias_attributes - (Optional) Attributes supported as an alias for this user pool. Possible values: phone_number, email, or preferred_username. Conflicts with username_attributes.
- auto_verified_attributes - (Optional) The attributes to be auto-verified. Possible values: email, phone_number.
- device_configuration (Optional) - The configuration for the user pool's device tracking.
- email_configuration (Optional) - The Email Configuration.
- name - (Required) The name of the user pool.
- email_verification_subject - (Optional) A string representing the email verification subject. Conflicts with verification_message_template configuration block email_subject argument.
- email_verification_message - (Optional) A string representing the email verification message. Conflicts with verification_message_template configuration block email_message argument.
- lambda_config (Optional) - A container for the AWS Lambda triggers associated with the user pool.
- mfa_configuration - (Optional, Default: OFF) Set to enable multi-factor authentication. Must be one of the following values (ON, OFF, OPTIONAL)
- password_policy (Optional) - A container for information about the user pool password policy.
- schema (Optional) - A container with the schema attributes of a user pool. Schema attributes from the standard attribute set only need to be specified if they are different from the default configuration. Maximum of 50 attributes.
- sms_authentication_message - (Optional) A string representing the SMS authentication message.
- sms_configuration (Optional) - The SMS Configuration.
- sms_verification_message - (Optional) A string representing the SMS verification message. Conflicts with verification_message_template configuration block sms_message argument.
- tags - (Optional) A mapping of tags to assign to the User Pool.
- username_attributes - (Optional) Specifies whether email addresses or phone numbers can be specified as usernames when a user signs up. Conflicts with alias_attributes.
- user_pool_add_ons - (Optional) Configuration block for user pool add-ons to enable user pool advanced security mode features.
- verification_message_template (Optional) - The verification message templates configuration.

» Admin Create User Config

- `allow_admin_create_user_only` (Optional) - Set to True if only the administrator is allowed to create user profiles. Set to False if users can sign themselves up via an app.
- `invite_message_template` (Optional) - The invite message template structure.
- `unused_account_validity_days` (Optional) - The user account expiration limit, in days, after which the account is no longer usable.

» Invite Message template

- `email_message` (Optional) - The message template for email messages. Must contain `{username}` and `{####}` placeholders, for username and temporary password, respectively.
- `email_subject` (Optional) - The subject line for email messages.
- `sms_message` (Optional) - The message template for SMS messages. Must contain `{username}` and `{####}` placeholders, for username and temporary password, respectively.

» Device Configuration

- `challenge_required_on_new_device` (Optional) - Indicates whether a challenge is required on a new device. Only applicable to a new device.
- `device_only_remembered_on_user_prompt` (Optional) - If true, a device is only remembered on user prompt.

» Email Configuration

- `reply_to_email_address` (Optional) - The REPLY-TO email address.
- `source_arn` (Optional) - The ARN of the email source.
- `email_sending_account` (Optional) - Instruct Cognito to either use its built-in functional or Amazon SES to send out emails.

» Lambda Configuration

- `create_auth_challenge` (Optional) - The ARN of the lambda creating an authentication challenge.
- `custom_message` (Optional) - A custom Message AWS Lambda trigger.
- `define_auth_challenge` (Optional) - Defines the authentication challenge.
- `post_authentication` (Optional) - A post-authentication AWS Lambda trigger.
- `post_confirmation` (Optional) - A post-confirmation AWS Lambda trigger.

- pre_authentication (Optional) - A pre-authentication AWS Lambda trigger.
- pre_sign_up (Optional) - A pre-registration AWS Lambda trigger.
- pre_token_generation (Optional) - Allow to customize identity token claims before token generation.
- user_migration (Optional) - The user migration Lambda config type.
- verify_auth_challenge_response (Optional) - Verifies the authentication challenge response.

» Password Policy

- minimum_length (Optional) - The minimum length of the password policy that you have set.
- require_lowercase (Optional) - Whether you have required users to use at least one lowercase letter in their password.
- require_numbers (Optional) - Whether you have required users to use at least one number in their password.
- require_symbols (Optional) - Whether you have required users to use at least one symbol in their password.
- require_uppercase (Optional) - Whether you have required users to use at least one uppercase letter in their password.

» Schema Attributes

NOTE: When defining an attribute_data_type of String or Number, the respective attribute constraints configuration block (e.g string_attribute_constraints or number_attribute_contraints) is required to prevent recreation of the Terraform resource. This requirement is true for both standard (e.g. name, email) and custom schema attributes.

- attribute_data_type (Required) - The attribute data type. Must be one of Boolean, Number, String, DateTime.
- developer_only_attribute (Optional) - Specifies whether the attribute type is developer only.
- mutable (Optional) - Specifies whether the attribute can be changed once it has been created.
- name (Required) - The name of the attribute.
- number_attribute_constraints (Optional) - Specifies the constraints for an attribute of the number type.
- required (Optional) - Specifies whether a user pool attribute is required. If the attribute is required and the user does not provide a value, registration or sign-in will fail.
- string_attribute_constraints (Optional) -Specifies the constraints for an attribute of the string type.

» Defaults for Standard Attributes

The standard attributes have the following defaults. Note that attributes which match the default values are not stored in Terraform state when importing.

```
schema {
  name                     = <name>
  attribute                = <appropriate type>
  developer_only_attribute = false
  mutable                  = true  // false for "sub"
  required                 = false // true for "sub"
  string_attribute_constraints { // if it's a string
    min_length = 0    // 10 for "birthdate"
    max_length = 2048 // 10 for "birthdate"
  }
}
```

» Number Attribute Constraints

- `max_value` (Optional) - The maximum value of an attribute that is of the number data type.
- `min_value` (Optional) - The minimum value of an attribute that is of the number data type.

» String Attribute Constraints

- `max_length` (Optional) - The maximum length of an attribute value of the string type.
- `min_length` (Optional) - The minimum length of an attribute value of the string type.

» SMS Configuration

- `external_id` (Required) - The external ID used in IAM role trust relationships. For more information about using external IDs, see How to Use an External ID When Granting Access to Your AWS Resources to a Third Party.
- `sns_caller_arn` (Required) - The ARN of the Amazon SNS caller. This is usually the IAM role that you've given Cognito permission to assume.

» User Pool Add-ons

- `advanced_security_mode` (Required) - The mode for advanced security, must be one of `OFF`, `AUDIT` or `ENFORCED`.

» Verification Message Template

- `default_email_option` (Optional) - The default email option. Must be either `CONFIRM_WITH_CODE` or `CONFIRM_WITH_LINK`. Defaults to `CONFIRM_WITH_CODE`.
- `email_message` (Optional) - The email message template. Must contain the `{####}` placeholder. Conflicts with `email_verification_message` argument.
- `email_message_by_link` (Optional) - The email message template for sending a confirmation link to the user, it must contain the `{##Click Here##}` placeholder.
- `email_subject` (Optional) - The subject line for the email message template. Conflicts with `email_verification_subject` argument.
- `email_subject_by_link` (Optional) - The subject line for the email message template for sending a confirmation link to the user.
- `sms_message` (Optional) - The SMS message template. Must contain the `{####}` placeholder. Conflicts with `sms_verification_message` argument.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the user pool.
- `arn` - The ARN of the user pool.
- `endpoint` - The endpoint name of the user pool. Example format: cognito-idp.REGION.amazonaws.com/xxxx_yyyyy
- `creation_date` - The date the user pool was created.
- `last_modified_date` - The date the user pool was last modified.

» Import

Cognito User Pools can be imported using the `id`, e.g.

```
$ terraform import aws_cognito_user_pool.pool <id>
```

» Resource: aws_cognito_user_pool_client

Provides a Cognito User Pool Client resource.

» Example Usage

» Create a basic user pool client

```
resource "aws_cognito_user_pool" "pool" {
  name = "pool"
}

resource "aws_cognito_user_pool_client" "client" {
  name = "client"

  user_pool_id = "${aws_cognito_user_pool.pool.id}"
}
```

» Create a user pool client with no SRP authentication

```
resource "aws_cognito_user_pool" "pool" {
  name = "pool"
}

resource "aws_cognito_user_pool_client" "client" {
  name = "client"

  user_pool_id = "${aws_cognito_user_pool.pool.id}"

  generate_secret     = true
  explicit_auth_flows = ["ADMIN_NO_SRP_AUTH"]
}
```

» Argument Reference

The following arguments are supported:

- `allowed_oauth_flows` - (Optional) List of allowed OAuth flows (code, implicit, client_credentials).
- `allowed_oauth_flows_user_pool_client` - (Optional) Whether the client is allowed to follow the OAuth protocol when interacting with Cognito user pools.
- `allowed_oauth_scopes` - (Optional) List of allowed OAuth scopes (phone, email, openid, profile, and aws.cognito.signin.user.admin).
- `callback_urls` - (Optional) List of allowed callback URLs for the identity providers.
- `default_redirect_uri` - (Optional) The default redirect URI. Must be in the list of callback URLs.

- `explicit_auth_flows` - (Optional) List of authentication flows (ADMIN_NO_SRP_AUTH, CUSTOM_AUTH_FLOW_ONLY, USER_PASSWORD_AUTH).
- `generate_secret` - (Optional) Should an application secret be generated.
- `logout_urls` - (Optional) List of allowed logout URLs for the identity providers.
- `name` - (Required) The name of the application client.
- `read_attributes` - (Optional) List of user pool attributes the application client can read from.
- `refresh_token_validity` - (Optional) The time limit in days refresh tokens are valid for.
- `supported_identity_providers` - (Optional) List of provider names for the identity providers that are supported on this client.
- `user_pool_id` - (Required) The user pool the client belongs to.
- `write_attributes` - (Optional) List of user pool attributes the application client can write to.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the user pool client.
- `client_secret` - The client secret of the user pool client.

» Import

Cognito User Pool Clients can be imported using the `id` of the Cognito User Pool, and the `id` of the Cognito User Pool Client, e.g.

```
$ terraform import aws_cognito_user_pool_client.client <user_pool_id>/<user_pool_client_id>
```

» Resource: aws_cognito_user_pool_domain

Provides a Cognito User Pool Domain resource.

» Example Usage

» Amazon Cognito domain

```
resource "aws_cognito_user_pool_domain" "main" {
  domain       = "example-domain"
  user_pool_id = "${aws_cognito_user_pool.example.id}"
}

resource "aws_cognito_user_pool" "example" {
  name = "example-pool"
}
```

» Custom Cognito domain

```
resource "aws_cognito_user_pool_domain" "main" {
  domain          = "example-domain.example.com"
  certificate_arn = "${aws_acm_certificate.cert.arn}"
  user_pool_id    = "${aws_cognito_user_pool.example.id}"
}

resource "aws_cognito_user_pool" "example" {
  name = "example-pool"
}
```

» Argument Reference

The following arguments are supported:

- `domain` - (Required) The domain string.
- `user_pool_id` - (Required) The user pool ID.
- `certificate_arn` - (Optional) The ARN of an ISSUED ACM certificate in us-east-1 for a custom domain.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `aws_account_id` - The AWS account ID for the user pool owner.
- `cloudfront_distribution_arn` - The ARN of the CloudFront distribution.
- `s3_bucket` - The S3 bucket where the static files for this domain are stored.
- `version` - The app version.

» Import

Cognito User Pool Domains can be imported using the `domain`, e.g.

```
$ terraform import aws_cognito_user_pool_domain.main <domain>
```

» Resource: aws_config_aggregate_authorization

Manages an AWS Config Aggregate Authorization

» Example Usage

```
resource "aws_config_aggregate_authorization" "example" {
  account_id = "123456789012"
  region     = "eu-west-2"
}
```

» Argument Reference

The following arguments are supported:

- `account_id` - (Required) Account ID
- `region` - (Required) Region
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the authorization

» Import

Config aggregate authorizations can be imported using `account_id:region`, e.g.

```
$ terraform import aws_config_authorization.example 123456789012:us-east-1
```

» Resource: aws_config_configuration_aggregator

Manages an AWS Config Configuration Aggregator

» Example Usage

» Account Based Aggregation

```
resource "aws_config_configuration_aggregator" "account" {
  name = "example"

  account_aggregation_source {
    account_ids = ["123456789012"]
    regions     = ["us-west-2"]
  }
}
```

» Organization Based Aggregation

```
resource "aws_config_configuration_aggregator" "organization" {
  depends_on = ["aws_iam_role_policy_attachment.organization"]

  name = "example" # Required

  organization_aggregation_source {
    all_regions = true
    role_arn    = "${aws_iam_role.organization.arn}"
  }
}

resource "aws_iam_role" "organization" {
  name = "example"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy_attachment" "organization" {
  role       = "${aws_iam_role.organization.name}"
  policy_arn = "arn:aws:iam::aws:policy/service-role/AWSConfigRoleForOrganizations"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the configuration aggregator.
- `account_aggregation_source` - (Optional) The account(s) to aggregate config data from as documented below.
- `organization_aggregation_source` - (Optional) The organization to aggregate config data from as documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Either `account_aggregation_source` or `organization_aggregation_source` must be specified.

» `account_aggregation_source`

- `account_ids` - (Required) List of 12-digit account IDs of the account(s) being aggregated.
- `all_regions` - (Optional) If true, aggregate existing AWS Config regions and future regions.
- `regions` - (Optional) List of source regions being aggregated.

Either `regions` or `all_regions` (as true) must be specified.

» `organization_aggregation_source`

Note: If your source type is an organization, you must be signed in to the master account and all features must be enabled in your organization. AWS Config calls EnableAwsServiceAccess API to enable integration between AWS Config and AWS Organizations.

- `all_regions` - (Optional) If true, aggregate existing AWS Config regions and future regions.
- `regions` - (Optional) List of source regions being aggregated.
- `role_arn` - (Required) ARN of the IAM role used to retrieve AWS Organization details associated with the aggregator account.

Either `regions` or `all_regions` (as true) must be specified.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the aggregator

» Import

Configuration Aggregators can be imported using the name, e.g.

```
$ terraform import aws_config_configuration_aggregator.example foo
```

» Resource: aws_config_config_rule

Provides an AWS Config Rule.

Note: Config Rule requires an existing Configuration Recorder to be present. Use of `depends_on` is recommended (as shown below) to avoid race conditions.

» Example Usage

» AWS Managed Rules

AWS managed rules can be used by setting the source owner to `AWS` and the source identifier to the name of the managed rule. More information about AWS managed rules can be found in the AWS Config Developer Guide.

```
resource "aws_config_config_rule" "r" {
  name = "example"

  source {
    owner             = "AWS"
    source_identifier = "S3_BUCKET_VERSIONING_ENABLED"
  }

  depends_on = ["aws_config_configuration_recorder.foo"]
}

resource "aws_config_configuration_recorder" "foo" {
  name     = "example"
  role_arn = "${aws_iam_role.r.arn}"
}

resource "aws_iam_role" "r" {
  name = "my-awsconfig-role"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy" "p" {
  name = "my-awsconfig-policy"
  role = "${aws_iam_role.r.id}"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Action": "config:Put*",
        "Effect": "Allow",
        "Resource": "*"

    }
  ]
}
POLICY
}
```

» Custom Rules

Custom rules can be used by setting the source owner to `CUSTOM_LAMBDA` and the source identifier to the Amazon Resource Name (ARN) of the Lambda Function. The AWS Config service must have permissions to invoke the Lambda Function, e.g. via the `aws_lambda_permission` resource. More information about custom rules can be found in the AWS Config Developer Guide.

```
resource "aws_config_configuration_recorder" "example" {
  # ... other configuration ...
}

resource "aws_lambda_function" "example" {
  # ... other configuration ...
```

```
}

resource "aws_lambda_permission" "example" {
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.example.arn}"
  principal     = "config.amazonaws.com"
  statement_id  = "AllowExecutionFromConfig"
}

resource "aws_config_config_rule" "example" {
  # ... other configuration ...

  source {
    owner             = "CUSTOM_LAMBDA"
    source_identifier = "${aws_lambda_function.example.arn}"
  }

  depends_on = ["aws_config_configuration_recorder.example", "aws_lambda_permission.example'
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the rule
- `description` - (Optional) Description of the rule
- `input_parameters` - (Optional) A string in JSON format that is passed to the AWS Config rule Lambda function.
- `maximum_execution_frequency` - (Optional) The maximum frequency with which AWS Config runs evaluations for a rule.
- `scope` - (Optional) Scope defines which resources can trigger an evaluation for the rule as documented below.
- `source` - (Required) Source specifies the rule owner, the rule identifier, and the notifications that cause the function to evaluate your AWS resources as documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» `scope`

Defines which resources can trigger an evaluation for the rule. If you do not specify a scope, evaluations are triggered when any resource in the recording group changes.

- `compliance_resource_id` - (Optional) The IDs of the only AWS resource that you want to trigger an evaluation for the rule. If you specify a resource

ID, you must specify one resource type for `compliance_resource_types`.

- `compliance_resource_types` - (Optional) A list of resource types of only those AWS resources that you want to trigger an evaluation for the rule. e.g. `AWS::EC2::Instance`. You can only specify one type if you also specify a resource ID for `compliance_resource_id`. See relevant part of AWS Docs for available types.
- `tag_key` - (Optional, Required if `tag_value` is specified) The tag key that is applied to only those AWS resources that you want you want to trigger an evaluation for the rule.
- `tag_value` - (Optional) The tag value applied to only those AWS resources that you want to trigger an evaluation for the rule.

» `source`

Provides the rule owner (AWS or customer), the rule identifier, and the notifications that cause the function to evaluate your AWS resources.

- `owner` - (Required) Indicates whether AWS or the customer owns and manages the AWS Config rule. Valid values are `AWS` or `CUSTOM_LAMBDA`. For more information about managed rules, see the AWS Config Managed Rules documentation. For more information about custom rules, see the AWS Config Custom Rules documentation. Custom Lambda Functions require permissions to allow the AWS Config service to invoke them, e.g. via the `aws_lambda_permission` resource.
- `source_identifier` - (Required) For AWS Config managed rules, a predefined identifier, e.g `IAM_PASSWORD_POLICY`. For custom Lambda rules, the identifier is the ARN of the Lambda Function, such as `arn:aws:lambda:us-east-1:123456789012:function:custom_rule_name` or the `arn` attribute of the `aws_lambda_function` resource.
- `source_detail` - (Optional) Provides the source and type of the event that causes AWS Config to evaluate your AWS resources. Only valid if `owner` is `CUSTOM_LAMBDA`.
  - `event_source` - (Optional) The source of the event, such as an AWS service, that triggers AWS Config to evaluate your AWS resources. This defaults to `aws.config` and is the only valid value.
  - `maximum_execution_frequency` - (Optional) The frequency that you want AWS Config to run evaluations for a rule that is triggered periodically. If specified, requires `message_type` to be `ScheduledNotification`.
  - `message_type` - (Optional) The type of notification that triggers AWS Config to run an evaluation for a rule. You can specify the following notification types:
    - `ConfigurationItemChangeNotification` - Triggers an evaluation when AWS Config delivers a configuration item as a result of a resource change.

    * `OversizedConfigurationItemChangeNotification` - Triggers an evaluation when AWS Config delivers an oversized configuration item. AWS Config may generate this notification type when a resource changes and the notification exceeds the maximum size allowed by Amazon SNS.
    * `ScheduledNotification` - Triggers a periodic evaluation at the frequency specified for `maximum_execution_frequency`.
    * `ConfigurationSnapshotDeliveryCompleted` - Triggers a periodic evaluation when AWS Config delivers a configuration snapshot.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the config rule
- `rule_id` - The ID of the config rule

» Import

Config Rule can be imported using the name, e.g.

```
$ terraform import aws_config_config_rule.foo example
```

» Resource: aws_config_configuration_recorder

Provides an AWS Config Configuration Recorder. Please note that this resource **does not start** the created recorder automatically.

Note: *Starting* the Configuration Recorder requires a delivery channel (while delivery channel creation requires Configuration Recorder). This is why `aws_config_configuration_recorder_status` is a separate resource.

» Example Usage

```
resource "aws_config_configuration_recorder" "foo" {
  name     = "example"
  role_arn = "${aws_iam_role.r.arn}"
}

resource "aws_iam_role" "r" {
  name = "awsconfig-example"
  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
POLICY
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the recorder. Defaults to `default`. Changing it recreates the resource.
- `role_arn` - (Required) Amazon Resource Name (ARN) of the IAM role. used to make read or write requests to the delivery channel and to describe the AWS resources associated with the account. See AWS Docs for more details.
- `recording_group` - (Optional) Recording group - see below.

» `recording_group`

- `all_supported` - (Optional) Specifies whether AWS Config records configuration changes for every supported type of regional resource (which includes any new type that will become supported in the future). Conflicts with `resource_types`. Defaults to `true`.
- `include_global_resource_types` - (Optional) Specifies whether AWS Config includes all supported types of *global resources* with the resources that it records. Requires `all_supported = true`. Conflicts with `resource_types`.
- `resource_types` - (Optional) A list that specifies the types of AWS resources for which AWS Config records configuration changes (for example, `AWS::EC2::Instance` or `AWS::CloudTrail::Trail`). See relevant part of AWS Docs for available types.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Name of the recorder

» Import

Configuration Recorder can be imported using the name, e.g.

```
$ terraform import aws_config_configuration_recorder.foo example
```

» Resource: aws_config_configuration_recorder_status

Manages status (recording / stopped) of an AWS Config Configuration Recorder.

Note: Starting Configuration Recorder requires a Delivery Channel to be present. Use of `depends_on` (as shown below) is recommended to avoid race conditions.

» Example Usage

```
resource "aws_config_configuration_recorder_status" "foo" {
  name       = "${aws_config_configuration_recorder.foo.name}"
  is_enabled = true
  depends_on = ["aws_config_delivery_channel.foo"]
}

resource "aws_iam_role_policy_attachment" "a" {
  role       = "${aws_iam_role.r.name}"
  policy_arn = "arn:aws:iam::aws:policy/service-role/AWSConfigRole"
}

resource "aws_s3_bucket" "b" {
  bucket = "awsconfig-example"
}

resource "aws_config_delivery_channel" "foo" {
  name           = "example"
  s3_bucket_name = "${aws_s3_bucket.b.bucket}"
}

resource "aws_config_configuration_recorder" "foo" {
  name     = "example"
  role_arn = "${aws_iam_role.r.arn}"
}

resource "aws_iam_role" "r" {
  name = "example-awsconfig"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy" "p" {
  name = "awsconfig-example"
  role = "${aws_iam_role.r.id}"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "${aws_s3_bucket.b.arn}",
        "${aws_s3_bucket.b.arn}/*"
      ]
    }
  ]
}
POLICY
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the recorder
- `is_enabled` - (Required) Whether the configuration recorder should be enabled or disabled.

» Import

Configuration Recorder Status can be imported using the name of the Configuration Recorder, e.g.

```
$ terraform import aws_config_configuration_recorder_status.foo example
```

» Resource: aws_config_delivery_channel

Provides an AWS Config Delivery Channel.

Note: Delivery Channel requires a Configuration Recorder to be present. Use of `depends_on` (as shown below) is recommended to avoid race conditions.

» Example Usage

```
resource "aws_config_delivery_channel" "foo" {
  name           = "example"
  s3_bucket_name = "${aws_s3_bucket.b.bucket}"
  depends_on     = ["aws_config_configuration_recorder.foo"]
}

resource "aws_s3_bucket" "b" {
  bucket        = "example-awsconfig"
  force_destroy = true
}

resource "aws_config_configuration_recorder" "foo" {
  name     = "example"
  role_arn = "${aws_iam_role.r.arn}"
}

resource "aws_iam_role" "r" {
  name = "awsconfig-example"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy" "p" {
  name = "awsconfig-example"
  role = "${aws_iam_role.r.id}"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "${aws_s3_bucket.b.arn}",
        "${aws_s3_bucket.b.arn}/*"
      ]
    }
  ]
}
POLICY
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the delivery channel. Defaults to `default`. Changing it recreates the resource.

- s3_bucket_name - (Required) The name of the S3 bucket used to store the configuration history.
- s3_key_prefix - (Optional) The prefix for the specified S3 bucket.
- sns_topic_arn - (Optional) The ARN of the SNS topic that AWS Config delivers notifications to.
- snapshot_delivery_properties - (Optional) Options for how AWS Config delivers configuration snapshots. See below

» snapshot_delivery_properties

- delivery_frequency - (Optional) - The frequency with which AWS Config recurringly delivers configuration snapshots. e.g. One_Hour or Three_Hours. Valid values are listed here.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The name of the delivery channel.

» Import

Delivery Channel can be imported using the name, e.g.

```
$ terraform import aws_config_delivery_channel.foo example
```

» Resource: aws_config_organization_custom_rule

Manages a Config Organization Custom Rule. More information about these rules can be found in the Enabling AWS Config Rules Across all Accounts in Your Organization and AWS Config Managed Rules documentation. For working with Organization Managed Rules (those invoking an AWS managed rule), see the aws_config_organization_managed__rule resource.

NOTE: This resource must be created in the Organization master account and rules will include the master account unless its ID is added to the excluded_accounts argument.

NOTE: The proper Lambda permission to allow the AWS Config service invoke the Lambda Function must be in place before the rule will successfully create or update. See also the aws_lambda_permission resource.

» Example Usage

```
resource "aws_lambda_permission" "example" {
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.example.arn}"
  principal     = "config.amazonaws.com"
  statement_id  = "AllowExecutionFromConfig"
}

resource "aws_organizations_organization" "example" {
  aws_service_access_principals = ["config-multiaccountsetup.amazonaws.com"]
  feature_set                   = "ALL"
}

resource "aws_config_organization_custom_rule" "example" {
  depends_on = ["aws_lambda_permission.example", "aws_organizations_organization.example"]

  lambda_function_arn = "${aws_lambda_function.example.arn}"
  name                = "example"
  trigger_types       = ["ConfigurationItemChangeNotification"]
}
```

» Argument Reference

The following arguments are supported:

- `lambda_function_arn` - (Required) Amazon Resource Name (ARN) of the rule Lambda Function
- `name` - (Required) The name of the rule
- `trigger_types` - (Required) List of notification types that trigger AWS Config to run an evaluation for the rule. Valid values: `ConfigurationItemChangeNotification`, `OversizedConfigurationItemChangeNotification`, and `ScheduledNotification`
- `description` - (Optional) Description of the rule
- `excluded_accounts` - (Optional) List of AWS account identifiers to exclude from the rule
- `input_parameters` - (Optional) A string in JSON format that is passed to the AWS Config Rule Lambda Function
- `maximum_execution_frequency` - (Optional) The maximum frequency with which AWS Config runs evaluations for a rule, if the rule is triggered at a periodic frequency. Defaults to `TwentyFour_Hours` for periodic frequency triggered rules. Valid values: `One_Hour`, `Three_Hours`, `Six_Hours`, `Twelve_Hours`, or `TwentyFour_Hours`.
- `resource_id_scope` - (Optional) Identifier of the AWS resource to evaluate

- `resource_types_scope` - (Optional) List of types of AWS resources to evaluate
- `tag_key_scope` - (Optional, Required if `tag_value_scope` is configured) Tag key of AWS resources to evaluate
- `tag_value_scope` - (Optional) Tag value of AWS resources to evaluate

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the rule

» Timeouts

`aws_config_organization_custom_rule` provides the following Timeouts configuration options:

- `create` - (Default `5m`) How long to wait for the rule to be created.
- `delete` - (Default `5m`) How long to wait for the rule to be deleted.
- `update` - (Default `5m`) How long to wait for the rule to be updated.

» Import

Config Organization Custom Rules can be imported using the name, e.g.

```
$ terraform import aws_config_organization_custom_rule.example example
```

» Resource: aws_config_organization_managed_rule

Manages a Config Organization Managed Rule. More information about these rules can be found in the Enabling AWS Config Rules Across all Accounts in Your Organization and AWS Config Managed Rules documentation. For working with Organization Custom Rules (those invoking a custom Lambda Function), see the `aws_config_organization_custom_rule` resource.

NOTE: This resource must be created in the Organization master account and rules will include the master account unless its ID is added to the `excluded_accounts` argument.

NOTE: Every Organization account except those configured in the `excluded_accounts` argument must have a Configuration Recorder with proper IAM permissions before the rule will successfully create or update. See also the `aws_config_configuration_recorder` resource.

» Example Usage

```
resource "aws_organizations_organization" "example" {
  aws_service_access_principals = ["config-multiaccountsetup.amazonaws.com"]
  feature_set                   = "ALL"
}

resource "aws_config_organization_managed_rule" "example" {
  depends_on = ["aws_organizations_organization.example"]

  name            = "example"
  rule_identifier = "IAM_PASSWORD_POLICY"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the rule
- `rule_identifier` - (Required) Identifier of an available AWS Config Managed Rule to call. For available values, see the List of AWS Config Managed Rules documentation
- `description` - (Optional) Description of the rule
- `excluded_accounts` - (Optional) List of AWS account identifiers to exclude from the rule
- `input_parameters` - (Optional) A string in JSON format that is passed to the AWS Config Rule Lambda Function
- `maximum_execution_frequency` - (Optional) The maximum frequency with which AWS Config runs evaluations for a rule, if the rule is triggered at a periodic frequency. Defaults to `TwentyFour_Hours` for periodic frequency triggered rules. Valid values: `One_Hour`, `Three_Hours`, `Six_Hours`, `Twelve_Hours`, or `TwentyFour_Hours`.
- `resource_id_scope` - (Optional) Identifier of the AWS resource to evaluate
- `resource_types_scope` - (Optional) List of types of AWS resources to evaluate
- `tag_key_scope` - (Optional, Required if `tag_value_scope` is configured) Tag key of AWS resources to evaluate
- `tag_value_scope` - (Optional) Tag value of AWS resources to evaluate

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the rule

» Timeouts

`aws_config_organization_managed_rule` provides the following Timeouts configuration options:

- `create` - (Default `5m`) How long to wait for the rule to be created.
- `delete` - (Default `5m`) How long to wait for the rule to be deleted.
- `update` - (Default `5m`) How long to wait for the rule to be updated.

» Import

Config Organization Managed Rules can be imported using the name, e.g.

```
$ terraform import aws_config_organization_managed_rule.example example
```

» Data Source: aws_cur_report_definition

Use this data source to get information on an AWS Cost and Usage Report Definition.

NOTE: The AWS Cost and Usage Report service is only available in `us-east-1` currently.

NOTE: If AWS Organizations is enabled, only the master account can use this resource.

» Example Usage

```
data "aws_cur_report_definition" "report_definition" {
  report_name = "example"
}
```

» Argument Reference

The following arguments are supported:

- `report_name` - (Required) The name of the report definition to match.

» Attributes Reference

- `time_unit` - The frequency on which report data are measured and displayed.
- `format` - Preferred compression format for report.

- `compression` - Preferred format for report.
- `additional_schema_elements` - A list of schema elements.
- `s3_bucket` - Name of customer S3 bucket.
- `s3_prefix` - Preferred report path prefix.
- `s3_region` - Region of customer S3 bucket.
- `additional_artifacts` - A list of additional artifacts.

» Resource: aws_cur_report_definition

Manages Cost and Usage Report Definitions.

NOTE: The AWS Cost and Usage Report service is only available in `us-east-1` currently.

NOTE: If AWS Organizations is enabled, only the master account can use this resource.

» Example Usage

```
resource "aws_cur_report_definition" "example_cur_report_definition" {
  report_name                = "example-cur-report-definition"
  time_unit                  = "HOURLY"
  format                     = "textORcsv"
  compression                = "GZIP"
  additional_schema_elements = ["RESOURCES"]
  s3_bucket                  = "example-bucket-name"
  s3_region                  = "us-east-1"
  additional_artifacts       = ["REDSHIFT", "QUICKSIGHT"]
}
```

» Argument Reference

The following arguments are supported:

- `report_name` - (Required) Unique name for the report. Must start with a number/letter and is case sensitive. Limited to 256 characters.
- `time_unit` - (Required) The frequency on which report data are measured and displayed. Valid values are: HOURLY, DAILY.
- `format` - (Required) Format for report. Valid values are: textORcsv.
- `compression` - (Required) Compression format for report. Valid values are: GZIP, ZIP.
- `additional_schema_elements` - (Required) A list of schema elements. Valid values are: RESOURCES.

- `s3_bucket` - (Required) Name of the existing S3 bucket to hold generated reports.
- `s3_prefix` - (Optional) Report path prefix. Limited to 256 characters.
- `s3_region` - (Required) Region of the existing S3 bucket to hold generated reports.
- `additional_artifacts` - (Required) A list of additional artifacts. Valid values are: REDSHIFT, QUICKSIGHT.

» Import

Report Definitions can be imported using the `report_name`, e.g.

```
$ terraform import aws_cur_report_definition.example_cur_report_definition example-cur-repor
```

» Resource: aws_dlm_lifecycle_policy

Provides a Data Lifecycle Manager (DLM) lifecycle policy for managing snapshots.

» Example Usage

```
resource "aws_iam_role" "dlm_lifecycle_role" {
  name = "dlm-lifecycle-role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "dlm.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "dlm_lifecycle" {
  name = "dlm-lifecycle-policy"
  role = "${aws_iam_role.dlm_lifecycle_role.id}"

  policy = <<EOF
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:CreateSnapshot",
            "ec2:DeleteSnapshot",
            "ec2:DescribeVolumes",
            "ec2:DescribeSnapshots"
         ],
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Action": [
            "ec2:CreateTags"
         ],
         "Resource": "arn:aws:ec2:*::snapshot/*"
      }
   ]
}
EOF
}

resource "aws_dlm_lifecycle_policy" "example" {
  description        = "example DLM lifecycle policy"
  execution_role_arn = "${aws_iam_role.dlm_lifecycle_role.arn}"
  state              = "ENABLED"

  policy_details {
    resource_types = ["VOLUME"]

    schedule {
      name = "2 weeks of daily snapshots"

      create_rule {
        interval      = 24
        interval_unit = "HOURS"
        times         = ["23:45"]
      }

      retain_rule {
        count = 14
      }

      tags_to_add = {
        SnapshotCreator = "DLM"
      }

      copy_tags = false
    }

    target_tags = {
      Snapshot = "true"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Required) A description for the DLM lifecycle policy.
- `execution_role_arn` - (Required) The ARN of an IAM role that is able to be assumed by the DLM service.
- `policy_details` - (Required) See the `policy_details` configuration block. Max of 1.
- `state` - (Optional) Whether the lifecycle policy should be enabled or disabled. `ENABLED` or `DISABLED` are valid values. Defaults to `ENABLED`.
- `tags` - (Optional) Key-value mapping of resource tags.

» Policy Details arguments

- `resource_types` - (Required) A list of resource types that should be targeted by the lifecycle policy. `VOLUME` is currently the only allowed value.
- `schedule` - (Required) See the `schedule` configuration block.
- `target_tags` (Required) A mapping of tag keys and their values. Any resources that match the `resource_types` and are tagged with *any* of these tags will be targeted.

Note: You cannot have overlapping lifecycle policies that share the same `target_tags`. Terraform is unable to detect this at plan time but it will fail during apply.

» Schedule arguments

- `copy_tags` - (Optional) Copy all user-defined tags on a source volume to snapshots of the volume created by this policy.

- `create_rule` - (Required) See the `create_rule` block. Max of 1 per schedule.
- `name` - (Required) A name for the schedule.
- `retain_rule` - (Required) See the `retain_rule` block. Max of 1 per schedule.
- `tags_to_add` - (Optional) A mapping of tag keys and their values. DLM lifecycle policies will already tag the snapshot with the tags on the volume. This configuration adds extra tags on top of these.

» Create Rule arguments

- `interval` - (Required) How often this lifecycle policy should be evaluated. `2`,`3`,`4`,`6`,`8`,`12` or `24` are valid values.
- `interval_unit` - (Optional) The unit for how often the lifecycle policy should be evaluated. `HOURS` is currently the only allowed value and also the default value.
- `times` - (Optional) A list of times in 24 hour clock format that sets when the lifecycle policy should be evaluated. Max of 1.

» Retain Rule arguments

- `count` - (Required) How many snapshots to keep. Must be an integer between 1 and 1000.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the DLM Lifecycle Policy.
- `id` - Identifier of the DLM Lifecycle Policy.

» Import

DLM lifecyle policies can be imported by their policy ID:

```
$ terraform import aws_dlm_lifecycle_policy.example policy-abcdef12345678901
```

» Resource: aws_dms_certificate

Provides a DMS (Data Migration Service) certificate resource. DMS certificates can be created, deleted, and imported.

Note: All arguments including the PEM encoded certificate will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
# Create a new certificate
resource "aws_dms_certificate" "test" {
  certificate_id  = "test-dms-certificate-tf"
  certificate_pem = "..."
}
```

» Argument Reference

The following arguments are supported:

- `certificate_id` - (Required) The certificate identifier.
  - Must contain from 1 to 255 alphanumeric characters and hyphens.
- `certificate_pem` - (Optional) The contents of the .pem X.509 certificate file for the certificate. Either `certificate_pem` or `certificate_wallet` must be set.
- `certificate_wallet` - (Optional) The contents of the Oracle Wallet certificate for use with SSL. Either `certificate_pem` or `certificate_wallet` must be set.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `certificate_arn` - The Amazon Resource Name (ARN) for the certificate.

» Import

Certificates can be imported using the `certificate_arn`, e.g.

```
$ terraform import aws_dms_certificate.test arn:aws:dms:us-west-2:123456789:cert:xxxxxxxxxx
```

» Resource: aws_dms_endpoint

Provides a DMS (Data Migration Service) endpoint resource. DMS endpoints can be created, updated, deleted, and imported.

Note: All arguments including the password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
# Create a new endpoint
resource "aws_dms_endpoint" "test" {
  certificate_arn             = "arn:aws:acm:us-east-1:123456789012:certificate/12345678-123
  database_name               = "test"
  endpoint_id                 = "test-dms-endpoint-tf"
  endpoint_type               = "source"
  engine_name                 = "aurora"
  extra_connection_attributes = ""
  kms_key_arn                 = "arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1
  password                    = "test"
  port                        = 3306
  server_name                 = "test"
  ssl_mode                    = "none"

  tags = {
    Name = "test"
  }

  username = "test"
}
```

» Argument Reference

The following arguments are supported:

- `certificate_arn` - (Optional, Default: empty string) The Amazon Resource Name (ARN) for the certificate.
- `database_name` - (Optional) The name of the endpoint database.
- `endpoint_id` - (Required) The database endpoint identifier.
  - Must contain from 1 to 255 alphanumeric characters or hyphens.
  - Must begin with a letter
  - Must contain only ASCII letters, digits, and hyphens
  - Must not end with a hyphen
  - Must not contain two consecutive hyphens
- `endpoint_type` - (Required) The type of endpoint. Can be one of `source | target`.
- `engine_name` - (Required) The type of engine for the endpoint. Can be one of `aurora | azuredb | db2 | docdb | dynamodb | mariadb | mongodb | mysql | oracle | postgres | redshift | s3 | sqlserver | sybase`.

- `extra_connection_attributes` - (Optional) Additional attributes associated with the connection. For available attributes see Using Extra Connection Attributes with AWS Database Migration Service.
- `kms_key_arn` - (Required when `engine_name` is `mongodb`, optional otherwise) The Amazon Resource Name (ARN) for the KMS key that will be used to encrypt the connection parameters. If you do not specify a value for `kms_key_arn`, then AWS DMS will use your default encryption key. AWS KMS creates the default encryption key for your AWS account. Your AWS account has a different default encryption key for each AWS region.
- `password` - (Optional) The password to be used to login to the endpoint database.
- `port` - (Optional) The port used by the endpoint database.
- `server_name` - (Optional) The host name of the server.
- `ssl_mode` - (Optional, Default: none) The SSL mode to use for the connection. Can be one of `none | require | verify-ca | verify-full`
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `username` - (Optional) The user name to be used to login to the endpoint database.
- `service_access_role` - (Optional) The Amazon Resource Name (ARN) used by the service access IAM role for dynamodb endpoints.
- `mongodb_settings` - (Optional) Settings for the source MongoDB endpoint. Available settings are `auth_type` (default: `password`), `auth_mechanism` (default: `default`), `nesting_level` (default: `none`), `extract_doc_id` (default: `false`), `docs_to_investigate` (default: `1000`) and `auth_source` (default: `admin`). For more details, see Using MongoDB as a Source for AWS DMS.
- `s3_settings` - (Optional) Settings for the target S3 endpoint. Available settings are `service_access_role_arn`, `external_table_definition`, `csv_row_delimiter` (default: `\\n`), `csv_delimiter` (default: `,`), `bucket_folder`, `bucket_name` and `compression_type` (default: `NONE`). For more details, see Using Amazon S3 as a Target for AWS Database Migration Service.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `endpoint_arn` - The Amazon Resource Name (ARN) for the endpoint.

» Import

Endpoints can be imported using the `endpoint_id`, e.g.

```
$ terraform import aws_dms_endpoint.test test-dms-endpoint-tf
```

» Resource: aws_dms_replication_instance

Provides a DMS (Data Migration Service) replication instance resource. DMS replication instances can be created, updated, deleted, and imported.

» Example Usage

```
# Database Migration Service requires the below IAM Roles to be created before
# replication instances can be created. See the DMS Documentation for
# additional information: https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Security.API
#  * dms-vpc-role
#  * dms-cloudwatch-logs-role
#  * dms-access-for-endpoint

data "aws_iam_policy_document" "dms_assume_role" {
  statement {
    actions = ["sts:AssumeRole"]

    principals {
      identifiers = ["dms.amazonaws.com"]
      type        = "Service"
    }
  }
}

resource "aws_iam_role" "dms-access-for-endpoint" {
  assume_role_policy = "${data.aws_iam_policy_document.dms_assume_role.json}"
  name               = "dms-access-for-endpoint"
}

resource "aws_iam_role_policy_attachment" "dms-access-for-endpoint-AmazonDMSRedshiftS3Role"
  policy_arn = "arn:aws:iam::aws:policy/service-role/AmazonDMSRedshiftS3Role"
  role       = "${aws_iam_role.dms-access-for-endpoint.name}"
}

resource "aws_iam_role" "dms-cloudwatch-logs-role" {
  assume_role_policy = "${data.aws_iam_policy_document.dms_assume_role.json}"
  name               = "dms-cloudwatch-logs-role"
```

```
}

resource "aws_iam_role_policy_attachment" "dms-cloudwatch-logs-role-AmazonDMSCloudWatchLogsR
  policy_arn = "arn:aws:iam::aws:policy/service-role/AmazonDMSCloudWatchLogsRole"
  role       = "${aws_iam_role.dms-cloudwatch-logs-role.name}"
}

resource "aws_iam_role" "dms-vpc-role" {
  assume_role_policy = "${data.aws_iam_policy_document.dms_assume_role.json}"
  name               = "dms-vpc-role"
}

resource "aws_iam_role_policy_attachment" "dms-vpc-role-AmazonDMSVPCManagementRole" {
  policy_arn = "arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole"
  role       = "${aws_iam_role.dms-vpc-role.name}"
}

# Create a new replication instance
resource "aws_dms_replication_instance" "test" {
  allocated_storage            = 20
  apply_immediately            = true
  auto_minor_version_upgrade   = true
  availability_zone            = "us-west-2c"
  engine_version               = "3.1.4"
  kms_key_arn                  = "arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-
  multi_az                     = false
  preferred_maintenance_window = "sun:10:30-sun:14:30"
  publicly_accessible          = true
  replication_instance_class   = "dms.t2.micro"
  replication_instance_id      = "test-dms-replication-instance-tf"
  replication_subnet_group_id  = "${aws_dms_replication_subnet_group.test-dms-replication-su

  tags = {
    Name = "test"
  }

  vpc_security_group_ids = [
    "sg-12345678",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `allocated_storage` - (Optional, Default: 50, Min: 5, Max: 6144) The amount of storage (in gigabytes) to be initially allocated for the replication instance.
- `apply_immediately` - (Optional, Default: false) Indicates whether the changes should be applied immediately or during the next maintenance window. Only used when updating an existing resource.
- `auto_minor_version_upgrade` - (Optional, Default: false) Indicates that minor engine upgrades will be applied automatically to the replication instance during the maintenance window.
- `availability_zone` - (Optional) The EC2 Availability Zone that the replication instance will be created in.
- `engine_version` - (Optional) The engine version number of the replication instance.
- `kms_key_arn` - (Optional) The Amazon Resource Name (ARN) for the KMS key that will be used to encrypt the connection parameters. If you do not specify a value for `kms_key_arn`, then AWS DMS will use your default encryption key. AWS KMS creates the default encryption key for your AWS account. Your AWS account has a different default encryption key for each AWS region.
- `multi_az` - (Optional) Specifies if the replication instance is a multi-az deployment. You cannot set the `availability_zone` parameter if the `multi_az` parameter is set to `true`.
- `preferred_maintenance_window` - (Optional) The weekly time range during which system maintenance can occur, in Universal Coordinated Time (UTC).
  - Default: A 30-minute window selected at random from an 8-hour block of time per region, occurring on a random day of the week.
  - Format: `ddd:hh24:mi-ddd:hh24:mi`
  - Valid Days: `mon, tue, wed, thu, fri, sat, sun`
  - Constraints: Minimum 30-minute window.
- `publicly_accessible` - (Optional, Default: false) Specifies the accessibility options for the replication instance. A value of true represents an instance with a public IP address. A value of false represents an instance with a private IP address.
- `replication_instance_class` - (Required) The compute and memory capacity of the replication instance as specified by the replication instance class. Can be one of `dms.t2.micro | dms.t2.small | dms.t2.medium | dms.t2.large | dms.c4.large | dms.c4.xlarge | dms.c4.2xlarge | dms.c4.4xlarge`

- `replication_instance_id` - (Required) The replication instance identifier. This parameter is stored as a lowercase string.
  - Must contain from 1 to 63 alphanumeric characters or hyphens.
  - First character must be a letter.
  - Cannot end with a hyphen
  - Cannot contain two consecutive hyphens.
- `replication_subnet_group_id` - (Optional) A subnet group to associate with the replication instance.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `vpc_security_group_ids` - (Optional) A list of VPC security group IDs to be used with the replication instance. The VPC security groups must work with the VPC containing the replication instance.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `replication_instance_arn` - The Amazon Resource Name (ARN) of the replication instance.
- `replication_instance_private_ips` - A list of the private IP addresses of the replication instance.
- `replication_instance_public_ips` - A list of the public IP addresses of the replication instance.

» Timeouts

`aws_dms_replication_instance` provides the following Timeouts configuration options:

- `create` - (Default `30 minutes`) Used for Creating Instances
- `update` - (Default `30 minutes`) Used for Database modifications
- `delete` - (Default `30 minutes`) Used for destroying databases.

» Import

Replication instances can be imported using the `replication_instance_id`, e.g.

```
$ terraform import aws_dms_replication_instance.test test-dms-replication-instance-tf
```

» Resource: aws_dms_replication_subnet_group

Provides a DMS (Data Migration Service) replication subnet group resource. DMS replication subnet groups can be created, updated, deleted, and imported.

» Example Usage

```
# Create a new replication subnet group
resource "aws_dms_replication_subnet_group" "test" {
  replication_subnet_group_description = "Test replication subnet group"
  replication_subnet_group_id          = "test-dms-replication-subnet-group-tf"

  subnet_ids = [
    "subnet-12345678",
  ]

  tags = {
    Name = "test"
  }
}
```

» Argument Reference

The following arguments are supported:

- `replication_subnet_group_description` - (Required) The description for the subnet group.
- `replication_subnet_group_id` - (Required) The name for the replication subnet group. This value is stored as a lowercase string.
  - Must contain no more than 255 alphanumeric characters, periods, spaces, underscores, or hyphens.
  - Must not be ”default”.
- `subnet_ids` - (Required) A list of the EC2 subnet IDs for the subnet group.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `vpc_id` - The ID of the VPC the subnet group is in.

» Import

Replication subnet groups can be imported using the `replication_subnet_group_id`, e.g.

```
$ terraform import aws_dms_replication_subnet_group.test test-dms-replication-subnet-group-t
```

» Resource: aws_dms_replication_task

Provides a DMS (Data Migration Service) replication task resource. DMS replication tasks can be created, updated, deleted, and imported.

» Example Usage

```
# Create a new replication task
resource "aws_dms_replication_task" "test" {
  cdc_start_time            = 1484346880
  migration_type            = "full-load"
  replication_instance_arn  = "${aws_dms_replication_instance.test-dms-replication-instance-
  replication_task_id       = "test-dms-replication-task-tf"
  replication_task_settings = "..."
  source_endpoint_arn       = "${aws_dms_endpoint.test-dms-source-endpoint-tf.endpoint_arn}"
  table_mappings            = "{\"rules\":[{\"rule-type\":\"selection\",\"rule-id\":\"1\",\"

  tags = {
    Name = "test"
  }

  target_endpoint_arn = "${aws_dms_endpoint.test-dms-target-endpoint-tf.endpoint_arn}"
}
```

» Argument Reference

The following arguments are supported:

- `cdc_start_time` - (Optional) The Unix timestamp integer for the start of the Change Data Capture (CDC) operation.
- `migration_type` - (Required) The migration type. Can be one of `full-load | cdc | full-load-and-cdc`.
- `replication_instance_arn` - (Required) The Amazon Resource Name (ARN) of the replication instance.
- `replication_task_id` - (Required) The replication task identifier.

  - Must contain from 1 to 255 alphanumeric characters or hyphens.
  - First character must be a letter.
  - Cannot end with a hyphen.
  - Cannot contain two consecutive hyphens.

- `replication_task_settings` - (Optional) An escaped JSON string that contains the task settings. For a complete list of task settings, see Task Settings for AWS Database Migration Service Tasks.
- `source_endpoint_arn` - (Required) The Amazon Resource Name (ARN) string that uniquely identifies the source endpoint.
- `table_mappings` - (Required) An escaped JSON string that contains the table mappings. For information on table mapping see Using Table Mapping with an AWS Database Migration Service Task to Select and Filter Data
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `target_endpoint_arn` - (Required) The Amazon Resource Name (ARN) string that uniquely identifies the target endpoint.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `replication_task_arn` - The Amazon Resource Name (ARN) for the replication task.

» Import

Replication tasks can be imported using the `replication_task_id`, e.g.

```
$ terraform import aws_dms_replication_task.test test-dms-replication-task-tf
```

» Resource: aws_datapipeline_pipeline

Provides a Data Pipeline resource.

» Example Usage

```
resource "aws_datapipeline_pipeline" "default" {
    name        = "tf-pipeline-default"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of Pipeline.
- `description` - (Optional) The description of Pipeline.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The identifier of the client certificate.

» Import

`aws_datapipeline_pipeline` can be imported by using the id (Pipeline ID), e.g.

```
$ terraform import aws_datapipeline_pipeline.default df-1234567890
```

» Resource: aws_datasync_agent

Manages an AWS DataSync Agent deployed on premises.

NOTE: One of `activation_key` or `ip_address` must be provided for resource creation (agent activation). Neither is required for resource import. If using `ip_address`, Terraform must be able to make an HTTP (port 80) GET request to the specified IP address from where it is running. The agent will turn off that HTTP server after activation.

» Example Usage

```
resource "aws_datasync_agent" "example" {
  ip_address = "1.2.3.4"
  name       = "example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the DataSync Agent.

- `activation_key` - (Optional) DataSync Agent activation key during resource creation. Conflicts with `ip_address`. If an `ip_address` is provided instead, Terraform will retrieve the `activation_key` as part of the resource creation.
- `ip_address` - (Optional) DataSync Agent IP address to retrieve activation key during resource creation. Conflicts with `activation_key`. DataSync Agent must be accessible on port 80 from where Terraform is running.
- `tags` - (Optional) Key-value pairs of resource tags to assign to the DataSync Agent.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the DataSync Agent.
- `arn` - Amazon Resource Name (ARN) of the DataSync Agent.

» Timeouts

`aws_datasync_agent` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for agent activation and connection to DataSync.

» Import

`aws_datasync_agent` can be imported by using the DataSync Agent Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_datasync_agent.example arn:aws:datasync:us-east-1:123456789012:agent/
```

» Resource: aws_datasync_location_efs

Manages an AWS DataSync EFS Location.

NOTE: The EFS File System must have a mounted EFS Mount Target before creating this resource.

» Example Usage

```
resource "aws_datasync_location_efs" "example" {
  # The below example uses aws_efs_mount_target as a reference to ensure a mount target alre
  # You can accomplish the same behavior with depends_on or an aws_efs_mount_target data sou
  efs_file_system_arn = "${aws_efs_mount_target.example.file_system_arn}"

  ec2_config {
    security_group_arns = ["${aws_security_group.example.arn}"]
    subnet_arn          = "${aws_subnet.example.arn}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `ec2_config` - (Required) Configuration block containing EC2 configurations for connecting to the EFS File System.
- `efs_file_system_arn` - (Required) Amazon Resource Name (ARN) of EFS File System.
- `subdirectory` - (Optional) Subdirectory to perform actions as source or destination. Default `/`.
- `tags` - (Optional) Key-value pairs of resource tags to assign to the DataSync Location.

» ec2_config Argument Reference

The following arguments are supported inside the `ec2_config` configuration block:

- `security_group_arns` - (Required) List of Amazon Resource Names (ARNs) of the EC2 Security Groups that are associated with the EFS Mount Target.
- `subnet_arn` - (Required) Amazon Resource Name (ARN) of the EC2 Subnet that is associated with the EFS Mount Target.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the DataSync Location.
- `arn` - Amazon Resource Name (ARN) of the DataSync Location.

» Import

`aws_datasync_location_efs` can be imported by using the DataSync Task Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_datasync_location_efs.example arn:aws:datasync:us-east-1:12345678901
```

» Resource: aws_datasync_location_nfs

Manages an NFS Location within AWS DataSync.

NOTE: The DataSync Agents must be available before creating this resource.

» Example Usage

```
resource "aws_datasync_location_nfs" "example" {
  server_hostname = "nfs.example.com"
  subdirectory    = "/exported/path"

  on_prem_config {
    agent_arns = ["${aws_datasync_agent.example.arn}"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `on_prem_config` - (Required) Configuration block containing information for connecting to the NFS File System.
- `server_hostname` - (Required) Specifies the IP address or DNS name of the NFS server. The DataSync Agent(s) use this to mount the NFS server.
- `subdirectory` - (Required) Subdirectory to perform actions as source or destination. Should be exported by the NFS server.
- `tags` - (Optional) Key-value pairs of resource tags to assign to the DataSync Location.

» on_prem_config Argument Reference

The following arguments are supported inside the `on_prem_config` configuration block:

- `agent_arns` - (Required) List of Amazon Resource Names (ARNs) of the DataSync Agents used to connect to the NFS server.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the DataSync Location.
- `arn` - Amazon Resource Name (ARN) of the DataSync Location.

» Import

`aws_datasync_location_nfs` can be imported by using the DataSync Task Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_datasync_location_nfs.example arn:aws:datasync:us-east-1:123456789012
```

» Resource: aws_datasync_location_s3

Manages an S3 Location within AWS DataSync.

» Example Usage

```
resource "aws_datasync_location_s3" "example" {
  s3_bucket_arn = "${aws_s3_bucket.example.arn}"
  subdirectory  = "/example/prefix"

  s3_config {
    bucket_access_role_arn = "${aws_iam_role.example.arn}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `s3_bucket_arn` - (Required) Amazon Resource Name (ARN) of the S3 Bucket.
- `s3_config` - (Required) Configuration block containing information for connecting to S3.
- `subdirectory` - (Required) Prefix to perform actions as source or destination.
- `tags` - (Optional) Key-value pairs of resource tags to assign to the DataSync Location.

» s3_config Argument Reference

The following arguments are supported inside the `s3_config` configuration block:

- `bucket_access_role_arn` - (Required) Amazon Resource Names (ARN) of the IAM Role used to connect to the S3 Bucket.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the DataSync Location.
- `arn` - Amazon Resource Name (ARN) of the DataSync Location.

» Import

`aws_datasync_location_s3` can be imported by using the DataSync Task Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_datasync_location_s3.example arn:aws:datasync:us-east-1:123456789012:
```

» Resource: aws_datasync_task

Manages an AWS DataSync Task, which represents a configuration for synchronization. Starting an execution of these DataSync Tasks (actually synchronizing files) is performed outside of this Terraform resource.

» Example Usage

```
resource "aws_datasync_task" "example" {
  destination_location_arn = "${aws_datasync_location_s3.destination.arn}"
  name                     = "example"
  source_location_arn      = "${aws_datasync_location_nfs.source.arn}"

  options {
    bytes_per_second = -1
  }
}
```

» Argument Reference

The following arguments are supported:

- `destination_location_arn` - (Required) Amazon Resource Name (ARN) of destination DataSync Location.
- `source_location_arn` - (Required) Amazon Resource Name (ARN) of source DataSync Location.
- `cloudwatch_log_group_arn` - (Optional) Amazon Resource Name (ARN) of the CloudWatch Log Group that is used to monitor and log events in the sync task.
- `name` - (Optional) Name of the DataSync Task.
- `options` - (Optional) Configuration block containing option that controls the default behavior when you start an execution of this DataSync Task. For each individual task execution, you can override these options by specifying an overriding configuration in those executions.
- `tags` - (Optional) Key-value pairs of resource tags to assign to the DataSync Task.

» options Argument Reference

NOTE: If `atime` is set to `BEST_EFFORT`, `mtime` must be set to `PRESERVE`. If `atime` is set to `NONE`, `mtime` must be set to `NONE`.

The following arguments are supported inside the `options` configuration block:

- `atime` - (Optional) A file metadata that shows the last time a file was accessed (that is when the file was read or written to). If set to `BEST_EFFORT`, the DataSync Task attempts to preserve the original (that is, the version before sync `PREPARING` phase) `atime` attribute on all source files. Valid values: `BEST_EFFORT`, `NONE`. Default: `BEST_EFFORT`.
- `bytes_per_second` - (Optional) Limits the bandwidth utilized. For example, to set a maximum of 1 MB, set this value to `1048576`. Value values: `-1` or greater. Default: `-1` (unlimited).
- `gid` - (Optional) Group identifier of the file's owners. Valid values: `BOTH`, `INT_VALUE`, `NAME`, `NONE`. Default: `INT_VALUE` (preserve integer value of the ID).
- `mtime` - (Optional) A file metadata that indicates the last time a file was modified (written to) before the sync `PREPARING` phase. Value values: `NONE`, `PRESERVE`. Default: `PRESERVE`.
- `posix_permissions` - (Optional) Determines which users or groups can access a file for a specific purpose such as reading, writing, or execution of the file. Valid values: `NONE`, `PRESERVE`. Default: `PRESERVE`.
- `preserve_deleted_files` - (Optional) Whether files deleted in the source should be removed or preserved in the destination file system. Valid values: `PRESERVE`, `REMOVE`. Default: `PRESERVE`.

- `preserve_devices` - (Optional) Whether the DataSync Task should preserve the metadata of block and character devices in the source files system, and recreate the files with that device name and metadata on the destination. The DataSync Task can't sync the actual contents of such devices, because many of the devices are non-terminal and don't return an end of file (EOF) marker. Valid values: `NONE`, `PRESERVE`. Default: `NONE` (ignore special devices).
- `uid` - (Optional) User identifier of the file's owners. Valid values: `BOTH`, `INT_VALUE`, `NAME`, `NONE`. Default: `INT_VALUE` (preserve integer value of the ID).
- `verify_mode` - (Optional) Whether a data integrity verification should be performed at the end of a task execution after all data and metadata have been transferred. Valid values: `NONE`, `POINT_IN_TIME_CONSISTENT`. Default: `POINT_IN_TIME_CONSISTENT`.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the DataSync Task.
- `arn` - Amazon Resource Name (ARN) of the DataSync Task.

» Timeouts

`aws_datasync_task` provides the following Timeouts configuration options:

- `create` - (Default `5m`) How long to wait for DataSync Task availability.

» Import

`aws_datasync_task` can be imported by using the DataSync Task Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_datasync_task.example arn:aws:datasync:us-east-1:123456789012:task/ta
```

» Resource: aws_devicefarm_project

Provides a resource to manage AWS Device Farm Projects. Please keep in mind that this feature is only supported on the "us-west-2" region. This resource will error if you try to create a project in another region.

For more information about Device Farm Projects, see the AWS Documentation on Device Farm Projects.

» Basic Example Usage

```
resource "aws_devicefarm_project" "awesome_devices" {
  name = "my-device-farm"
}
```

» Argument Reference

- `name` - (Required) The name of the project

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name of this project

» Data Source: aws_directory_service_directory

Get attributes of AWS Directory Service directory (SimpleAD, Managed AD, AD Connector). It's especially useful to refer AWS Managed AD or on-premise AD in AD Connector configuration.

» Example Usage

```
data "aws_directory_service_directory" "example" {
  directory_id = "${aws_directory_service_directory.main.id}"
}
```

» Argument Reference

- `directory_id` - (Required) The ID of the directory.

» Attributes Reference

- `type` - The directory type (`SimpleAD`, `ADConnector` or `MicrosoftAD`).
- `edition` - (for `MicrosoftAD`) The Microsoft AD edition (`Standard` or `Enterprise`).
- `name` - The fully qualified name for the directory/connector.
- `password` - The password for the directory administrator or connector user.

- `size` - (for `SimpleAD` and `ADConnector`) The size of the directory/connector (`Small` or `Large`).
- `alias` - The alias for the directory/connector, such as `d-991708b282.awsapps.com`.
- `description` - A textual description for the directory/connector.
- `short_name` - The short name of the directory/connector, such as `CORP`.
- `enable_sso` - The directory/connector single-sign on status.
- `access_url` - The access URL for the directory/connector, such as http://alias.awsapps.com.
- `dns_ip_addresses` - A list of IP addresses of the DNS servers for the directory/connector.
- `security_group_id` - The ID of the security group created by the directory/connector.
- `tags` – A mapping of tags assigned to the directory/connector.

`vpc_settings` (for `SimpleAD` and `MicrosoftAD`) is also exported with the following attributes:

- `subnet_ids` - The identifiers of the subnets for the directory servers (2 subnets in 2 different AZs).
- `vpc_id` - The ID of the VPC that the directory is in.

`connect_settings` (for `ADConnector`) is also exported with the following attributes:

- `customer_username` - The username corresponding to the password provided.
- `customer_dns_ips` - The DNS IP addresses of the domain to connect to.
- `subnet_ids` - The identifiers of the subnets for the connector servers (2 subnets in 2 different AZs).
- `vpc_id` - The ID of the VPC that the connector is in.

» Resource: aws_directory_service_directory

Provides a Simple or Managed Microsoft directory in AWS Directory Service.

Note: All arguments including the password and customer username will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

» SimpleAD

```
resource "aws_directory_service_directory" "bar" {
  name     = "corp.notexample.com"
  password = "SuperSecretPassw0rd"
  size     = "Small"
  vpc_settings {
    vpc_id     = "${aws_vpc.main.id}"
    subnet_ids = ["${aws_subnet.foo.id}", "${aws_subnet.bar.id}"]
  }

  tags = {
    Project = "foo"
  }
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "foo" {
  vpc_id            = "${aws_vpc.main.id}"
  availability_zone = "us-west-2a"
  cidr_block        = "10.0.1.0/24"
}

resource "aws_subnet" "bar" {
  vpc_id            = "${aws_vpc.main.id}"
  availability_zone = "us-west-2b"
  cidr_block        = "10.0.2.0/24"
}
```

» Microsoft Active Directory (MicrosoftAD)

```
resource "aws_directory_service_directory" "bar" {
  name     = "corp.notexample.com"
  password = "SuperSecretPassw0rd"
  edition  = "Standard"
  type     = "MicrosoftAD"

  vpc_settings {
    vpc_id     = "${aws_vpc.main.id}"
    subnet_ids = ["${aws_subnet.foo.id}", "${aws_subnet.bar.id}"]
  }

  tags = {
    Project = "foo"
  }
}
```

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "foo" {
  vpc_id            = "${aws_vpc.main.id}"
  availability_zone = "us-west-2a"
  cidr_block        = "10.0.1.0/24"
}

resource "aws_subnet" "bar" {
  vpc_id            = "${aws_vpc.main.id}"
  availability_zone = "us-west-2b"
  cidr_block        = "10.0.2.0/24"
}
```

» Microsoft Active Directory Connector (ADConnector)

```
resource "aws_directory_service_directory" "connector" {
  name     = "corp.notexample.com"
  password = "SuperSecretPassw0rd"
  size     = "Small"
  type     = "ADConnector"

  connect_settings {
    customer_dns_ips  = ["A.B.C.D"]
    customer_username = "Admin"
    subnet_ids        = ["${aws_subnet.foo.id}", "${aws_subnet.bar.id}"]
    vpc_id            = "${aws_vpc.main.id}"
  }
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "foo" {
  vpc_id            = "${aws_vpc.main.id}"
  availability_zone = "us-west-2a"
  cidr_block        = "10.0.1.0/24"
}

resource "aws_subnet" "bar" {
  vpc_id            = "${aws_vpc.main.id}"
  availability_zone = "us-west-2b"
  cidr_block        = "10.0.2.0/24"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The fully qualified name for the directory, such as `corp.example.com`
- `password` - (Required) The password for the directory administrator or connector user.
- `size` - (Required for `SimpleAD` and `ADConnector`) The size of the directory (`Small` or `Large` are accepted values).
- `vpc_settings` - (Required for `SimpleAD` and `MicrosoftAD`) VPC related information about the directory. Fields documented below.
- `connect_settings` - (Required for `ADConnector`) Connector related information about the directory. Fields documented below.
- `alias` - (Optional) The alias for the directory (must be unique amongst all aliases in AWS). Required for `enable_sso`.
- `description` - (Optional) A textual description for the directory.
- `short_name` - (Optional) The short name of the directory, such as `CORP`.
- `enable_sso` - (Optional) Whether to enable single-sign on for the directory. Requires `alias`. Defaults to `false`.
- `type` (Optional) - The directory type (`SimpleAD`, `ADConnector` or `MicrosoftAD` are accepted values). Defaults to `SimpleAD`.
- `edition` - (Optional) The MicrosoftAD edition (`Standard` or `Enterprise`). Defaults to `Enterprise` (applies to MicrosoftAD type only).
- `tags` - (Optional) A mapping of tags to assign to the resource.

vpc_settings supports the following:

- `subnet_ids` - (Required) The identifiers of the subnets for the directory servers (2 subnets in 2 different AZs).
- `vpc_id` - (Required) The identifier of the VPC that the directory is in.

connect_settings supports the following:

- `customer_username` - (Required) The username corresponding to the password provided.
- `customer_dns_ips` - (Required) The DNS IP addresses of the domain to connect to.
- `subnet_ids` - (Required) The identifiers of the subnets for the directory servers (2 subnets in 2 different AZs).
- `vpc_id` - (Required) The identifier of the VPC that the directory is in.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The directory identifier.
- `access_url` - The access URL for the directory, such as `http://alias.awsapps.com`.
- `dns_ip_addresses` - A list of IP addresses of the DNS servers for the directory or connector.
- `security_group_id` - The ID of the security group created by the directory.

» Import

DirectoryService directories can be imported using the directory `id`, e.g.

```
$ terraform import aws_directory_service_directory.sample d-926724cf57
```

» Resource: aws_directory_service_conditional_forwarder

Provides a conditional forwarder for managed Microsoft AD in AWS Directory Service.

» Example Usage

```
resource "aws_directory_service_conditional_forwarder" "example" {
  directory_id       = "${aws_directory_service_directory.ad.id}"
  remote_domain_name = "example.com"

  dns_ips = [
    "8.8.8.8",
    "8.8.4.4",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `directory_id` - (Required) The id of directory.
- `dns_ips` - (Required) A list of forwarder IP addresses.
- `remote_domain_name` - (Required) The fully qualified domain name of the remote domain for which forwarders will be used.

» Import

Conditional forwarders can be imported using the directory id and remote_domain_name, e.g.

```
$ terraform import aws_directory_service_conditional_forwarder.example d-1234567890:example.
```

» Resource: aws_directory_service_log_subscription

Provides a Log subscription for AWS Directory Service that pushes logs to cloudwatch.

» Example Usage

```
resource "aws_cloudwatch_log_group" "example" {
  name              = "/aws/directoryservice/${aws_directory_service_directory.example.id}"
  retention_in_days = 14
}

data "aws_iam_policy_document" "ad-log-policy" {
  statement {
    actions = [
      "logs:CreateLogStream",
      "logs:PutLogEvents"
    ]

    principals {
      identifiers = ["ds.amazonaws.com"]
      type = "Service"
    }

    resources = ["${aws_cloudwatch_log_group.example.arn}"]

    effect = "Allow"
  }
}

resource "aws_cloudwatch_log_resource_policy" "ad-log-policy" {
  policy_document = "${data.aws_iam_policy_document.ad-log-policy.json}"
  policy_name     = "ad-log-policy"
}

resource "aws_directory_service_log_subscription" "example" {
  directory_id   = "${aws_directory_service_directory.example.id}"
  log_group_name = "${aws_cloudwatch_log_group.example.name}"
}
```

» Argument Reference

The following arguments are supported:

- `directory_id` - (Required) The id of directory.
- `log_group_name` - (Required) Name of the cloudwatch log group to which the logs should be published. The log group should be already created and the directory service principal should be provided with required permission to create stream and publish logs. Changing this value would delete the current subscription and create a new one. A directory can only have one log subscription at a time.

» Import

Directory Service Log Subscriptions can be imported using the directory id, e.g.

```
$ terraform import aws_directory_service_log_subscription.msad d-1234567890
```

» Data Source: aws_dx_gateway

Retrieve information about a Direct Connect Gateway.

» Example Usage

```
data "aws_dx_gateway" "example" {
  name = "example"
}
```

» Argument Reference

- `name` - (Required) The name of the gateway to retrieve.

» Attributes Reference

- `amazon_side_asn` - The ASN on the Amazon side of the connection.
- `id` - The ID of the gateway.
- `owner_account_id` - AWS Account ID of the gateway.

» Resource: aws_dx_bgp_peer

Provides a Direct Connect BGP peer resource.

» Example Usage

```
resource "aws_dx_bgp_peer" "peer" {
  virtual_interface_id = "${aws_dx_private_virtual_interface.foo.id}"
  address_family       = "ipv6"
  bgp_asn              = 65351
}
```

» Argument Reference

The following arguments are supported:

- address_family - (Required) The address family for the BGP peer. ipv4 or ipv6.
- bgp_asn - (Required) The autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
- virtual_interface_id - (Required) The ID of the Direct Connect virtual interface on which to create the BGP peer.
- amazon_address - (Optional) The IPv4 CIDR address to use to send traffic to Amazon. Required for IPv4 BGP peers on public virtual interfaces.
- bgp_auth_key - (Optional) The authentication key for BGP configuration.
- customer_address - (Optional) The IPv4 CIDR destination address to which Amazon should send traffic. Required for IPv4 BGP peers on public virtual interfaces.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The ID of the BGP peer resource.
- bgp_status - The Up/Down state of the BGP peer.
- bgp_peer_id - The ID of the BGP peer.
- aws_device - The Direct Connect endpoint on which the BGP peer terminates.

» Timeouts

aws_dx_bgp_peer provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating BGP peer
- `delete` - (Default `10 minutes`) Used for destroying BGP peer

» Resource: aws_dx_connection

Provides a Connection of Direct Connect.

» Example Usage

```
resource "aws_dx_connection" "hoge" {
  name      = "tf-dx-connection"
  bandwidth = "1Gbps"
  location  = "EqDC2"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the connection.
- `bandwidth` - (Required) The bandwidth of the connection. Available values: 1Gbps, 10Gbps. Case sensitive.
- `location` - (Required) The AWS Direct Connect location where the connection is located. See DescribeLocations for the list of AWS Direct Connect locations. Use `locationCode`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the connection.
- `arn` - The ARN of the connection.
- `jumbo_frame_capable` - Boolean value representing if jumbo frames have been enabled for this connection.
- `has_logical_redundancy` - Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).
- `aws_device` - The Direct Connect endpoint on which the physical connection terminates.

» Import

Direct Connect connections can be imported using the `connection id`, e.g.

```
$ terraform import aws_dx_connection.test_connection dxcon-ffre0ec3
```

» Resource: aws_dx_connection_association

Associates a Direct Connect Connection with a LAG.

» Example Usage

```
resource "aws_dx_connection" "example" {
  name      = "example"
  bandwidth = "1Gbps"
  location  = "EqSe2"
}

resource "aws_dx_lag" "example" {
  name                  = "example"
  connections_bandwidth = "1Gbps"
  location              = "EqSe2"
}

resource "aws_dx_connection_association" "example" {
  connection_id = "${aws_dx_connection.example.id}"
  lag_id        = "${aws_dx_lag.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `connection_id` - (Required) The ID of the connection.
- `lag_id` - (Required) The ID of the LAG with which to associate the connection.

» Resource: aws_dx_gateway

Provides a Direct Connect Gateway.

» Example Usage

```
resource "aws_dx_gateway" "example" {
  name            = "tf-dxg-example"
  amazon_side_asn = "64512"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the connection.
- `amazon_side_asn` - (Required) The ASN to be configured on the Amazon side of the connection. The ASN must be in the private range of 64,512 to 65,534 or 4,200,000,000 to 4,294,967,294.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the gateway.
- `owner_account_id` - AWS Account ID of the gateway.

» Timeouts

`aws_dx_gateway` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating the gateway
- `delete` - (Default `10 minutes`) Used for destroying the gateway

» Import

Direct Connect Gateways can be imported using the `gateway id`, e.g.

```
$ terraform import aws_dx_gateway.test abcd1234-dcba-5678-be23-cdef9876ab45
```

» Resource: aws_dx_gateway_association

Associates a Direct Connect Gateway with a VGW or transit gateway.

To create a cross-account association, create an `aws_dx_gateway_association_proposal` resource in the AWS account that owns the VGW or transit gateway and then accept the proposal in the AWS account that owns the Direct Connect Gateway by creating an `aws_dx_gateway_association` resource with the `proposal_id` and `associated_gateway_owner_account_id` attributes set.

» Example Usage

» VPN Gateway Association

```
resource "aws_dx_gateway" "example" {
  name            = "example"
  amazon_side_asn = "64512"
}

resource "aws_vpc" "example" {
  cidr_block = "10.255.255.0/28"
}

resource "aws_vpn_gateway" "example" {
  vpc_id = "${aws_vpc.example.id}"
}

resource "aws_dx_gateway_association" "example" {
  dx_gateway_id         = "${aws_dx_gateway.example.id}"
  associated_gateway_id = "${aws_vpn_gateway.example.id}"
}
```

» Transit Gateway Association

```
resource "aws_dx_gateway" "example" {
  name            = "example"
  amazon_side_asn = "64512"
}

resource "aws_ec2_transit_gateway" "example" {}

resource "aws_dx_gateway_association" "example" {
  dx_gateway_id         = "${aws_dx_gateway.example.id}"
  associated_gateway_id = "${aws_ec2_transit_gateway.example.id}"

  allowed_prefixes = [
    "10.255.255.0/30",
    "10.255.255.8/30",
  ]
}
```

» Allowed Prefixes

```
resource "aws_dx_gateway" "example" {
  name            = "example"
  amazon_side_asn = "64512"
}

resource "aws_vpc" "example" {
  cidr_block = "10.255.255.0/28"
}

resource "aws_vpn_gateway" "example" {
  vpc_id = "${aws_vpc.example.id}"
}

resource "aws_dx_gateway_association" "example" {
  dx_gateway_id         = "${aws_dx_gateway.example.id}"
  associated_gateway_id = "${aws_vpn_gateway.example.id}"

  allowed_prefixes = [
    "210.52.109.0/24",
    "175.45.176.0/22",
  ]
}
```

A full example of how to create a VPN Gateway in one AWS account, create a Direct Connect Gateway in a second AWS account, and associate the VPN Gateway with the Direct Connect Gateway via the `aws_dx_gateway_association_proposal` and `aws_dx_gateway_association` resources can be found in the `./examples/dx-gateway-cross-account-vgw-association` directory within the Github Repository.

» Argument Reference

NOTE: `dx_gateway_id` plus one of `associated_gateway_id`, or `vpn_gateway_id` must be specified for single account Direct Connect gateway associations.

The following arguments are supported:

- `dx_gateway_id` - (Required) The ID of the Direct Connect gateway.
- `associated_gateway_id` - (Optional) The ID of the VGW or transit gateway with which to associate the Direct Connect gateway. Used for single account Direct Connect gateway associations.
- `vpn_gateway_id` - (Optional) *Deprecated:* Use `associated_gateway_id` instead. The ID of the VGW with which to associate the gateway. Used for single account Direct Connect gateway associations.

- associated_gateway_owner_account_id - (Optional) The ID of the AWS account that owns the VGW or transit gateway with which to associate the Direct Connect gateway. Used for cross-account Direct Connect gateway associations.
- proposal_id - (Optional) The ID of the Direct Connect gateway association proposal. Used for cross-account Direct Connect gateway associations.
- allowed_prefixes - (Optional) VPC prefixes (CIDRs) to advertise to the Direct Connect gateway. Defaults to the CIDR block of the VPC associated with the Virtual Gateway. To enable drift detection, must be configured.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The ID of the Direct Connect gateway association resource.
- associated_gateway_type - The type of the associated gateway, transitGateway or virtualPrivateGateway.
- dx_gateway_association_id - The ID of the Direct Connect gateway association.
- dx_gateway_owner_account_id - The ID of the AWS account that owns the Direct Connect gateway.

» Timeouts

aws_dx_gateway_association provides the following Timeouts configuration options:

- create - (Default 15 minutes) Used for creating the association
- update - (Default 10 minutes) Used for updating the association
- delete - (Default 15 minutes) Used for destroying the association

» Import

Direct Connect gateway associations can be imported using dx_gateway_id together with associated_gateway_id, e.g.

```
$ terraform import aws_dx_gateway_association.example dxgw-12345678/vgw-98765432
```

» Resource: aws_dx_gateway_association_proposal

Manages a Direct Connect Gateway Association Proposal, typically for enabling cross-account associations. For single account associations, see the `aws_dx_gateway_association` resource.

» Example Usage

```
resource "aws_dx_gateway_association_proposal" "example" {
  dx_gateway_id               = "${aws_dx_gateway.example.id}"
  dx_gateway_owner_account_id = "${aws_dx_gateway.example.owner_account_id}"
  associated_gateway_id       = "${aws_vpn_gateway.example.id}"
}
```

A full example of how to create a VPN Gateway in one AWS account, create a Direct Connect Gateway in a second AWS account, and associate the VPN Gateway with the Direct Connect Gateway via the `aws_dx_gateway_association_proposal` and `aws_dx_gateway_association` resources can be found in the `./examples/dx-gateway-cross-account-vgw-association` directory within the Github Repository.

» Argument Reference

NOTE: One of `associated_gateway_id`, or `vpn_gateway_id` must be specified.

The following arguments are supported:

- `dx_gateway_id` - (Required) Direct Connect Gateway identifier.
- `dx_gateway_owner_account_id` - (Required) AWS Account identifier of the Direct Connect Gateway's owner.
- `associated_gateway_id` - (Optional) The ID of the VGW or transit gateway with which to associate the Direct Connect gateway.
- `vpn_gateway_id` - (Optional) *Deprecated:* Use `associated_gateway_id` instead. Virtual Gateway identifier to associate with the Direct Connect Gateway.
- `allowed_prefixes` - (Optional) VPC prefixes (CIDRs) to advertise to the Direct Connect gateway. Defaults to the CIDR block of the VPC associated with the Virtual Gateway. To enable drift detection, must be configured.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Direct Connect Gateway Association Proposal identifier.
- `associated_gateway_owner_account_id` - The ID of the AWS account that owns the VGW or transit gateway with which to associate the Direct Connect gateway.
- `associated_gateway_type` - The type of the associated gateway, `transitGateway` or `virtualPrivateGateway`.

» Import

Direct Connect Gateway Association Proposals can be imported using the proposal ID, e.g.

```
$ terraform import aws_dx_gateway_association_proposal.example ac90e981-b718-4364-872d-65478
```

» Resource: aws_dx_hosted_private_virtual_interface

Provides a Direct Connect hosted private virtual interface resource. This resource represents the allocator's side of the hosted virtual interface. A hosted virtual interface is a virtual interface that is owned by another AWS account.

» Example Usage

```
resource "aws_dx_hosted_private_virtual_interface" "foo" {
  connection_id = "dxcon-zzzzzzzz"

  name           = "vif-foo"
  vlan           = 4094
  address_family = "ipv4"
  bgp_asn        = 65352
}
```

» Argument Reference

The following arguments are supported:

- `address_family` - (Required) The address family for the BGP peer. `ipv4` or `ipv6`.
- `bgp_asn` - (Required) The autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
- `connection_id` - (Required) The ID of the Direct Connect connection (or LAG) on which to create the virtual interface.
- `name` - (Required) The name for the virtual interface.

- `owner_account_id` - (Required) The AWS account that will own the new virtual interface.
- `vlan` - (Required) The VLAN ID.
- `amazon_address` - (Optional) The IPv4 CIDR address to use to send traffic to Amazon. Required for IPv4 BGP peers.
- `mtu` - (Optional) The maximum transmission unit (MTU) is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual private interface can be either `1500` or `9001` (jumbo frames). Default is `1500`.
- `bgp_auth_key` - (Optional) The authentication key for BGP configuration.
- `customer_address` - (Optional) The IPv4 CIDR destination address to which Amazon should send traffic. Required for IPv4 BGP peers.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual interface.
- `arn` - The ARN of the virtual interface.
- `jumbo_frame_capable` - Indicates whether jumbo frames (9001 MTU) are supported.
- `aws_device` - The Direct Connect endpoint on which the virtual interface terminates.

» Timeouts

`aws_dx_hosted_private_virtual_interface` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating virtual interface
- `update` - (Default `10 minutes`) Used for virtual interface modifications
- `delete` - (Default `10 minutes`) Used for destroying virtual interface

» Import

Direct Connect hosted private virtual interfaces can be imported using the `vif id`, e.g.

```
$ terraform import aws_dx_hosted_private_virtual_interface.test dxvif-33cc44dd
```

» Resource: aws_dx_hosted_private_virtual_interface_accepter

Provides a resource to manage the accepter's side of a Direct Connect hosted private virtual interface. This resource accepts ownership of a private virtual interface created by another AWS account.

» Example Usage

```
provider "aws" {
  # Creator's credentials.
}

provider "aws" {
  alias = "accepter"

  # Accepter's credentials.
}

data "aws_caller_identity" "accepter" {
  provider = "aws.accepter"
}

# Creator's side of the VIF
resource "aws_dx_hosted_private_virtual_interface" "creator" {
  connection_id    = "dxcon-zzzzzzzz"
  owner_account_id = "${data.aws_caller_identity.accepter.account_id}"

  name           = "vif-foo"
  vlan           = 4094
  address_family = "ipv4"
  bgp_asn        = 65352

  # The aws_dx_hosted_private_virtual_interface
  # must be destroyed before the aws_vpn_gateway.
  depends_on = ["aws_vpn_gateway.vpn_gw"]
}

# Accepter's side of the VIF.
resource "aws_vpn_gateway" "vpn_gw" {
  provider = "aws.accepter"
}

resource "aws_dx_hosted_private_virtual_interface_accepter" "accepter" {
  provider             = "aws.accepter"
  virtual_interface_id = "${aws_dx_hosted_private_virtual_interface.creator.id}"
  vpn_gateway_id       = "${aws_vpn_gateway.vpn_gw.id}"

  tags = {
    Side = "Accepter"
  }
}
```

» Argument Reference

The following arguments are supported:

- `virtual_interface_id` - (Required) The ID of the Direct Connect virtual interface to accept.
- `dx_gateway_id` - (Optional) The ID of the Direct Connect gateway to which to connect the virtual interface.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `vpn_gateway_id` - (Optional) The ID of the virtual private gateway to which to connect the virtual interface.

» Removing `aws_dx_hosted_private_virtual_interface_accepter` from your configuration

AWS allows a Direct Connect hosted private virtual interface to be deleted from either the allocator's or accepter's side. However, Terraform only allows the Direct Connect hosted private virtual interface to be deleted from the allocator's side by removing the corresponding `aws_dx_hosted_private_virtual_interface` resource from your configuration. Removing a `aws_dx_hosted_private_virtual_interface_accepter` resource from your configuration will remove it from your statefile and management, **but will not delete the Direct Connect virtual interface.**

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual interface.
- `arn` - The ARN of the virtual interface.

» Timeouts

`aws_dx_hosted_private_virtual_interface_accepter` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating virtual interface
- `delete` - (Default `10 minutes`) Used for destroying virtual interface

» Import

Direct Connect hosted private virtual interfaces can be imported using the `vif id`, e.g.

```
$ terraform import aws_dx_hosted_private_virtual_interface_accepter.test dxvif-33cc44dd
```

» Resource: aws_dx_hosted_public_virtual_interface

Provides a Direct Connect hosted public virtual interface resource. This resource represents the allocator's side of the hosted virtual interface. A hosted virtual interface is a virtual interface that is owned by another AWS account.

» Example Usage

```
resource "aws_dx_hosted_public_virtual_interface" "foo" {
  connection_id = "dxcon-zzzzzzzz"

  name           = "vif-foo"
  vlan           = 4094
  address_family = "ipv4"
  bgp_asn        = 65352

  customer_address = "175.45.176.1/30"
  amazon_address   = "175.45.176.2/30"

  route_filter_prefixes = [
    "210.52.109.0/24",
    "175.45.176.0/22",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `address_family` - (Required) The address family for the BGP peer. `ipv4` or `ipv6`.

- `bgp_asn` - (Required) The autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
- `connection_id` - (Required) The ID of the Direct Connect connection (or LAG) on which to create the virtual interface.
- `name` - (Required) The name for the virtual interface.
- `owner_account_id` - (Required) The AWS account that will own the new virtual interface.
- `route_filter_prefixes` - (Required) A list of routes to be advertised to the AWS network in this region.
- `vlan` - (Required) The VLAN ID.
- `amazon_address` - (Optional) The IPv4 CIDR address to use to send traffic to Amazon. Required for IPv4 BGP peers.
- `bgp_auth_key` - (Optional) The authentication key for BGP configuration.
- `customer_address` - (Optional) The IPv4 CIDR destination address to which Amazon should send traffic. Required for IPv4 BGP peers.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual interface.
- `arn` - The ARN of the virtual interface.
- `aws_device` - The Direct Connect endpoint on which the virtual interface terminates.

» Timeouts

`aws_dx_hosted_public_virtual_interface` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating virtual interface
- `delete` - (Default `10 minutes`) Used for destroying virtual interface

» Import

Direct Connect hosted public virtual interfaces can be imported using the `vif id`, e.g.

```
$ terraform import aws_dx_hosted_public_virtual_interface.test dxvif-33cc44dd
```

» Resource: aws_dx_hosted_public_virtual_interface_accepter

Provides a resource to manage the accepter's side of a Direct Connect hosted public virtual interface. This resource accepts ownership of a public virtual interface created by another AWS account.

» Example Usage

```
provider "aws" {
  # Creator's credentials.
}

provider "aws" {
  alias = "accepter"

  # Accepter's credentials.
}

data "aws_caller_identity" "accepter" {
  provider = "aws.accepter"
}

# Creator's side of the VIF
resource "aws_dx_hosted_public_virtual_interface" "creator" {
  connection_id    = "dxcon-zzzzzzzz"
  owner_account_id = "${data.aws_caller_identity.accepter.account_id}"

  name           = "vif-foo"
  vlan           = 4094
  address_family = "ipv4"
  bgp_asn        = 65352

  customer_address = "175.45.176.1/30"
  amazon_address   = "175.45.176.2/30"

  route_filter_prefixes = [
    "210.52.109.0/24",
    "175.45.176.0/22",
  ]
}

# Accepter's side of the VIF.
resource "aws_dx_hosted_public_virtual_interface_accepter" "accepter" {
  provider             = "aws.accepter"
  virtual_interface_id = "${aws_dx_hosted_public_virtual_interface.creator.id}"

  tags = {
    Side = "Accepter"
  }
}
```

» Argument Reference

The following arguments are supported:

- `virtual_interface_id` - (Required) The ID of the Direct Connect virtual interface to accept.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Removing `aws_dx_hosted_public_virtual_interface_accepter` from your configuration

AWS allows a Direct Connect hosted public virtual interface to be deleted from either the allocator's or accepter's side. However, Terraform only allows the Direct Connect hosted public virtual interface to be deleted from the allocator's side by removing the corresponding `aws_dx_hosted_public_virtual_interface` resource from your configuration. Removing a `aws_dx_hosted_public_virtual_interface_accepter` resource from your configuration will remove it from your statefile and management, **but will not delete the Direct Connect virtual interface.**

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual interface.
- `arn` - The ARN of the virtual interface.

» Timeouts

`aws_dx_hosted_public_virtual_interface_accepter` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating virtual interface
- `delete` - (Default `10 minutes`) Used for destroying virtual interface

» Import

Direct Connect hosted public virtual interfaces can be imported using the `vif id`, e.g.

```
$ terraform import aws_dx_hosted_public_virtual_interface_accepter.test dxvif-33cc44dd
```

» Resource: aws_dx_hosted_transit_virtual_interface

Provides a Direct Connect hosted transit virtual interface resource. This resource represents the allocator's side of the hosted virtual interface. A hosted virtual interface is a virtual interface that is owned by another AWS account.

» Example Usage

```
resource "aws_dx_hosted_transit_virtual_interface" "example" {
  connection_id = "${aws_dx_connection.example.id}"

  name           = "tf-transit-vif-example"
  vlan           = 4094
  address_family = "ipv4"
  bgp_asn        = 65352
}
```

» Argument Reference

The following arguments are supported:

- `address_family` - (Required) The address family for the BGP peer. `ipv4` or `ipv6`.
- `bgp_asn` - (Required) The autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
- `connection_id` - (Required) The ID of the Direct Connect connection (or LAG) on which to create the virtual interface.
- `name` - (Required) The name for the virtual interface.
- `owner_account_id` - (Required) The AWS account that will own the new virtual interface.
- `vlan` - (Required) The VLAN ID.
- `amazon_address` - (Optional) The IPv4 CIDR address to use to send traffic to Amazon. Required for IPv4 BGP peers.
- `bgp_auth_key` - (Optional) The authentication key for BGP configuration.
- `customer_address` - (Optional) The IPv4 CIDR destination address to which Amazon should send traffic. Required for IPv4 BGP peers.

- `mtu` - (Optional) The maximum transmission unit (MTU) is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual transit interface can be either `1500` or `8500` (jumbo frames). Default is `1500`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual interface.
- `arn` - The ARN of the virtual interface.
- `aws_device` - The Direct Connect endpoint on which the virtual interface terminates.
- `jumbo_frame_capable` - Indicates whether jumbo frames (8500 MTU) are supported.

» Timeouts

`aws_dx_hosted_transit_virtual_interface` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating virtual interface
- `update` - (Default `10 minutes`) Used for virtual interface modifications
- `delete` - (Default `10 minutes`) Used for destroying virtual interface

» Import

Direct Connect hosted transit virtual interfaces can be imported using the `vif id`, e.g.

```
$ terraform import aws_dx_hosted_transit_virtual_interface.test dxvif-33cc44dd
```

» Resource: aws_dx_hosted_transit_virtual_interface_accepter

Provides a resource to manage the accepter's side of a Direct Connect hosted transit virtual interface. This resource accepts ownership of a transit virtual interface created by another AWS account.

NOTE: AWS allows a Direct Connect hosted transit virtual interface to be deleted from either the allocator's or accepter's side. However, Terraform only allows the Direct Connect hosted transit virtual interface to be deleted from the allocator's side by removing the corresponding aws_dx_hosted_transit_virtual_interface resource from your configuration. Removing a aws_dx_hosted_transit_virtual_interface_accepter resource from your configuration will remove it from your statefile and management, **but will not delete the Direct Connect virtual interface.**

» Example Usage

```
provider "aws" {
  # Creator's credentials.
}

provider "aws" {
  alias = "accepter"

  # Accepter's credentials.
}

data "aws_caller_identity" "accepter" {
  provider = "aws.accepter"
}

# Creator's side of the VIF
resource "aws_dx_hosted_transit_virtual_interface" "creator" {
  connection_id    = "dxcon-zzzzzzzz"
  owner_account_id = "${data.aws_caller_identity.accepter.account_id}"

  name           = "tf-transit-vif-example"
  vlan           = 4094
  address_family = "ipv4"
  bgp_asn        = 65352

  # The aws_dx_hosted_transit_virtual_interface
  # must be destroyed before the aws_dx_gateway.
  depends_on = ["aws_dx_gateway.example"]
}

# Accepter's side of the VIF.
resource "aws_dx_gateway" "example" {
  provider = "aws.accepter"

  name            = "tf-dxg-example"
  amazon_side_asn = 64512
}

resource "aws_dx_hosted_transit_virtual_interface_accepter" "accepter" {
  provider             = "aws.accepter"
  virtual_interface_id = "${aws_dx_hosted_transit_virtual_interface.creator.id}"
  dx_gateway_id        = "${aws_dx_gateway.example.id}"

  tags = {
    Side = "Accepter"
  }
}
```

» Argument Reference

The following arguments are supported:

- `dx_gateway_id` - (Required) The ID of the Direct Connect gateway to which to connect the virtual interface.
- `virtual_interface_id` - (Required) The ID of the Direct Connect virtual interface to accept.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual interface.
- `arn` - The ARN of the virtual interface.

» Timeouts

`aws_dx_hosted_transit_virtual_interface_accepter` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating virtual interface
- `delete` - (Default `10 minutes`) Used for destroying virtual interface

» Import

Direct Connect hosted transit virtual interfaces can be imported using the `vif id`, e.g.

```
$ terraform import aws_dx_hosted_transit_virtual_interface_accepter.test dxvif-33cc44dd
```

» Resource: aws_dx_lag

Provides a Direct Connect LAG. Connections can be added to the LAG via the `aws_dx_connection` and `aws_dx_connection_association` resources.

NOTE: When creating a LAG, Direct Connect requires creating a Connection. Terraform will remove this unmanaged connection during resource creation.

» Example Usage

```
resource "aws_dx_lag" "hoge" {
  name                  = "tf-dx-lag"
  connections_bandwidth = "1Gbps"
  location              = "EqDC2"
  force_destroy         = true
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the LAG.
- `connections_bandwidth` - (Required) The bandwidth of the individual physical connections bundled by the LAG. Available values: 1Gbps, 10Gbps. Case sensitive.
- `location` - (Required) The AWS Direct Connect location in which the LAG should be allocated. See DescribeLocations for the list of AWS Direct Connect locations. Use `locationCode`.
- `force_destroy` - (Optional, Default:false) A boolean that indicates all connections associated with the LAG should be deleted so that the LAG can be destroyed without error. These objects are *not* recoverable.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the LAG.
- `arn` - The ARN of the LAG.
- `jumbo_frame_capable` -Indicates whether jumbo frames (9001 MTU) are supported.
- `has_logical_redundancy` - Indicates whether the LAG supports a secondary BGP peer in the same address family (IPv4/IPv6).

» Import

Direct Connect LAGs can be imported using the `lag id`, e.g.

```
$ terraform import aws_dx_lag.test_lag dxlag-fgnsp5rq
```

» Resource: aws_dx_transit_virtual_interface

Provides a Direct Connect transit virtual interface resource. A transit virtual interface is a VLAN that transports traffic from a Direct Connect gateway to one or more transit gateways.

» Example Usage

```
resource "aws_dx_gateway" "example" {
  name            = "tf-dxg-example"
  amazon_side_asn = 64512
}

resource "aws_dx_transit_virtual_interface" "example" {
  connection_id = "${aws_dx_connection.example.id}"

  dx_gateway_id  = "${aws_dx_gateway.example.id}"
  name           = "tf-transit-vif-example"
  vlan           = 4094
  address_family = "ipv4"
  bgp_asn        = 65352
}
```

» Argument Reference

The following arguments are supported:

- `address_family` - (Required) The address family for the BGP peer. `ipv4` or `ipv6`.
- `bgp_asn` - (Required) The autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
- `connection_id` - (Required) The ID of the Direct Connect connection (or LAG) on which to create the virtual interface.
- `dx_gateway_id` - (Required) The ID of the Direct Connect gateway to which to connect the virtual interface.
- `name` - (Required) The name for the virtual interface.
- `vlan` - (Required) The VLAN ID.

- `amazon_address` - (Optional) The IPv4 CIDR address to use to send traffic to Amazon. Required for IPv4 BGP peers.
- `bgp_auth_key` - (Optional) The authentication key for BGP configuration.
- `customer_address` - (Optional) The IPv4 CIDR destination address to which Amazon should send traffic. Required for IPv4 BGP peers.
- `mtu` - (Optional) The maximum transmission unit (MTU) is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual transit interface can be either `1500` or `8500` (jumbo frames). Default is `1500`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the virtual interface.
- `arn` - The ARN of the virtual interface.
- `aws_device` - The Direct Connect endpoint on which the virtual interface terminates.
- `jumbo_frame_capable` - Indicates whether jumbo frames (8500 MTU) are supported.

» Timeouts

`aws_dx_transit_virtual_interface` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating virtual interface
- `update` - (Default `10 minutes`) Used for virtual interface modifications
- `delete` - (Default `10 minutes`) Used for destroying virtual interface

» Import

Direct Connect transit virtual interfaces can be imported using the `vif id`, e.g.

```
$ terraform import aws_dx_transit_virtual_interface.test dxvif-33cc44dd
```

» Data Source: aws_dynamodb_table

Provides information about a DynamoDB table.

» Example Usage

```
data "aws_dynamodb_table" "tableName" {
  name = "tableName"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the DynamoDB table.

» Attributes Reference

See the DynamoDB Table Resource for details on the returned attributes - they are identical.

» Resource: aws_dynamodb_global_table

Provides a resource to manage a DynamoDB Global Table. These are layered on top of existing DynamoDB Tables.

Note: There are many restrictions before you can properly create DynamoDB Global Tables in multiple regions. See the AWS DynamoDB Global Table Requirements for more information.

» Example Usage

```
provider "aws" {
  alias  = "us-east-1"
  region = "us-east-1"
}

provider "aws" {
  alias  = "us-west-2"
  region = "us-west-2"
}

resource "aws_dynamodb_table" "us-east-1" {
  provider = "aws.us-east-1"

  hash_key         = "myAttribute"
  name             = "myTable"
  stream_enabled   = true
  stream_view_type = "NEW_AND_OLD_IMAGES"
  read_capacity    = 1
  write_capacity   = 1

  attribute {
    name = "myAttribute"
    type = "S"
  }
}

resource "aws_dynamodb_table" "us-west-2" {
  provider = "aws.us-west-2"

  hash_key         = "myAttribute"
  name             = "myTable"
  stream_enabled   = true
  stream_view_type = "NEW_AND_OLD_IMAGES"
  read_capacity    = 1
  write_capacity   = 1

  attribute {
    name = "myAttribute"
    type = "S"
  }
}

resource "aws_dynamodb_global_table" "myTable" {
  depends_on = ["aws_dynamodb_table.us-east-1", "aws_dynamodb_table.us-west-2"]
  provider   = "aws.us-east-1"

  name = "myTable"

  replica {
    region_name = "us-east-1"
  }

  replica {
    region_name = "us-west-2"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the global table. Must match underlying DynamoDB Table names in all regions.
- `replica` - (Required) Underlying DynamoDB Table. At least 1 replica must be defined. See below.

» Nested Fields

» `replica`

- `region_name` - (Required) AWS region name of replica DynamoDB Table. e.g. `us-east-1`

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the DynamoDB Global Table
- `arn` - The ARN of the DynamoDB Global Table

» Import

DynamoDB Global Tables can be imported using the global table name, e.g.

```
$ terraform import aws_dynamodb_global_table.MyTable MyTable
```

» Resource: aws_dynamodb_table

Provides a DynamoDB table resource

Note: It is recommended to use `lifecycle` `ignore_changes` for `read_capacity` and/or `write_capacity` if there's autoscaling policy attached to the table.

» Example Usage

The following dynamodb table description models the table and GSI shown in the AWS SDK example documentation

```
resource "aws_dynamodb_table" "basic-dynamodb-table" {
  name           = "GameScores"
  billing_mode   = "PROVISIONED"
  read_capacity  = 20
  write_capacity = 20
  hash_key       = "UserId"
  range_key      = "GameTitle"

  attribute {
    name = "UserId"
    type = "S"
  }

  attribute {
    name = "GameTitle"
    type = "S"
  }

  attribute {
    name = "TopScore"
    type = "N"
  }

  ttl {
    attribute_name = "TimeToExist"
    enabled        = false
  }

  global_secondary_index {
    name               = "GameTitleIndex"
    hash_key           = "GameTitle"
    range_key          = "TopScore"
    write_capacity     = 10
    read_capacity      = 10
    projection_type    = "INCLUDE"
    non_key_attributes = ["UserId"]
  }

  tags = {
    Name        = "dynamodb-table-1"
    Environment = "production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the table, this needs to be unique within a region.
- `billing_mode` - (Optional) Controls how you are charged for read and write throughput and how you manage capacity. The valid values are `PROVISIONED` and `PAY_PER_REQUEST`. Defaults to `PROVISIONED`.
- `hash_key` - (Required, Forces new resource) The attribute to use as the hash (partition) key. Must also be defined as an `attribute`, see below.
- `range_key` - (Optional, Forces new resource) The attribute to use as the range (sort) key. Must also be defined as an `attribute`, see below.
- `write_capacity` - (Optional) The number of write units for this table. If the `billing_mode` is `PROVISIONED`, this field is required.
- `read_capacity` - (Optional) The number of read units for this table. If the `billing_mode` is `PROVISIONED`, this field is required.
- `attribute` - (Required) List of nested attribute definitions. Only required for `hash_key` and `range_key` attributes. Each attribute has two properties:
  - `name` - (Required) The name of the attribute
  - `type` - (Required) Attribute type, which must be a scalar type: `S`, `N`, or `B` for (S)tring, (N)umber or (B)inary data
- `ttl` - (Optional) Defines ttl, has two properties, and can only be specified once:
  - `enabled` - (Required) Indicates whether ttl is enabled (true) or disabled (false).
  - `attribute_name` - (Required) The name of the table attribute to store the TTL timestamp in.
- `local_secondary_index` - (Optional, Forces new resource) Describe an LSI on the table; these can only be allocated *at creation* so you cannot change this definition after you have created the resource.
- `global_secondary_index` - (Optional) Describe a GSI for the table; subject to the normal limits on the number of GSIs, projected attributes, etc.
- `stream_enabled` - (Optional) Indicates whether Streams are to be enabled (true) or disabled (false).
- `stream_view_type` - (Optional) When an item in the table is modified, StreamViewType determines what information is written to the table's stream. Valid values are `KEYS_ONLY`, `NEW_IMAGE`, `OLD_IMAGE`, `NEW_AND_OLD_IMAGES`.
- `server_side_encryption` - (Optional) Encryption at rest options. AWS DynamoDB tables are automatically encrypted at rest with an AWS owned Customer Master Key if this argument isn't specified.
- `tags` - (Optional) A map of tags to populate on the created table.
- `point_in_time_recovery` - (Optional) Point-in-time recovery options.

» Timeouts

The `timeouts` block allows you to specify timeouts for certain actions:

- `create` - (Defaults to 10 mins) Used when creating the table
- `update` - (Defaults to 60 mins) Used when updating the table configuration and reset for each individual Global Secondary Index update
- `delete` - (Defaults to 10 mins) Used when deleting the table

» Nested fields

» `local_secondary_index`

- `name` - (Required) The name of the index
- `range_key` - (Required) The name of the range key; must be defined
- `projection_type` - (Required) One of `ALL`, `INCLUDE` or `KEYS_ONLY` where `ALL` projects every attribute into the index, `KEYS_ONLY` projects just the hash and range key into the index, and `INCLUDE` projects only the keys specified in the *non_key_attributes* parameter.
- `non_key_attributes` - (Optional) Only required with `INCLUDE` as a projection type; a list of attributes to project into the index. These do not need to be defined as attributes on the table.

» `global_secondary_index`

- `name` - (Required) The name of the index
- `write_capacity` - (Optional) The number of write units for this index. Must be set if billing_mode is set to PROVISIONED.
- `read_capacity` - (Optional) The number of read units for this index. Must be set if billing_mode is set to PROVISIONED.
- `hash_key` - (Required) The name of the hash key in the index; must be defined as an attribute in the resource.
- `range_key` - (Optional) The name of the range key; must be defined
- `projection_type` - (Required) One of `ALL`, `INCLUDE` or `KEYS_ONLY` where `ALL` projects every attribute into the index, `KEYS_ONLY` projects just the hash and range key into the index, and `INCLUDE` projects only the keys specified in the *non_key_attributes* parameter.
- `non_key_attributes` - (Optional) Only required with `INCLUDE` as a projection type; a list of attributes to project into the index. These do not need to be defined as attributes on the table.

» `server_side_encryption`

- `enabled` - (Required) Whether or not to enable encryption at rest using an AWS managed Customer Master Key. If `enabled` is `false` then server-side encryption is set to AWS owned CMK (shown as `DEFAULT` in the AWS console). If `enabled` is `true` then server-side encryption is set to AWS managed CMK (shown as `KMS` in the AWS console). The AWS KMS documentation explains the difference between AWS owned and AWS managed CMKs.

» `point_in_time_recovery`

- `enabled` - (Required) Whether to enable point-in-time recovery - note that it can take up to 10 minutes to enable for new tables. If the `point_in_time_recovery` block is not provided then this defaults to `false`.

» A note about attributes

Only define attributes on the table object that are going to be used as:

- Table hash key or range key
- LSI or GSI hash key or range key

The DynamoDB API expects attribute structure (name and type) to be passed along when creating or updating GSI/LSIs or creating the initial table. In these cases it expects the Hash / Range keys to be provided; because these get re-used in numerous places (i.e the table's range key could be a part of one or more GSIs), they are stored on the table object to prevent duplication and increase consistency. If you add attributes here that are not used in these scenarios it can cause an infinite loop in planning.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The arn of the table
- `id` - The name of the table
- `stream_arn` - The ARN of the Table Stream. Only available when `stream_enabled = true`
- `stream_label` - A timestamp, in ISO 8601 format, for this stream. Note that this timestamp is not a unique identifier for the stream on its own. However, the combination of AWS customer ID, table name and this field is guaranteed to be unique. It can be used for creating CloudWatch Alarms. Only available when `stream_enabled = true`

» Import

DynamoDB tables can be imported using the `name`, e.g.

```
$ terraform import aws_dynamodb_table.basic-dynamodb-table GameScores
```

» Resource: aws_dynamodb_table_item

Provides a DynamoDB table item resource

Note: This resource is not meant to be used for managing large amounts of data in your table, it is not designed to scale. You should perform **regular backups** of all data in the table, see AWS docs for more.

» Example Usage

```
resource "aws_dynamodb_table_item" "example" {
  table_name = "${aws_dynamodb_table.example.name}"
  hash_key   = "${aws_dynamodb_table.example.hash_key}"

  item = <<ITEM
{
  "exampleHashKey": {"S": "something"},
  "one": {"N": "11111"},
  "two": {"N": "22222"},
  "three": {"N": "33333"},
  "four": {"N": "44444"}
}
ITEM
}

resource "aws_dynamodb_table" "example" {
  name           = "example-name"
  read_capacity  = 10
  write_capacity = 10
  hash_key       = "exampleHashKey"

  attribute {
    name = "exampleHashKey"
    type = "S"
  }
}
```

» Argument Reference

The following arguments are supported:

- `table_name` - (Required) The name of the table to contain the item.
- `hash_key` - (Required) Hash key to use for lookups and identification of the item
- `range_key` - (Optional) Range key to use for lookups and identification of the item. Required if there is range key defined in the table.
- `item` - (Required) JSON representation of a map of attribute name/value pairs, one for each attribute. Only the primary key attributes are required; you can optionally provide other attribute name-value pairs for the item.

» Attributes Reference

All of the arguments above are exported as attributes.

» Import

DynamoDB table items cannot be imported.

» Resource: aws_dax_cluster

Provides a DAX Cluster resource.

» Example Usage

```
resource "aws_dax_cluster" "bar" {
  cluster_name       = "cluster-example"
  iam_role_arn       = "${data.aws_iam_role.example.arn}"
  node_type          = "dax.r4.large"
  replication_factor = 1
}
```

» Argument Reference

The following arguments are supported:

- `cluster_name` – (Required) Group identifier. DAX converts this name to lowercase

- `iam_role_arn` - (Required) A valid Amazon Resource Name (ARN) that identifies an IAM role. At runtime, DAX will assume this role and use the role's permissions to access DynamoDB on your behalf

- `node_type` – (Required) The compute and memory capacity of the nodes. See Nodes for supported node types
- `replication_factor` – (Required) The number of nodes in the DAX cluster. A replication factor of 1 will create a single-node cluster, without any read replicas
- `availability_zones` - (Optional) List of Availability Zones in which the nodes will be created
- `description` – (Optional) Description for the cluster
- `notification_topic_arn` – (Optional) An Amazon Resource Name (ARN) of an SNS topic to send DAX notifications to. Example: `arn:aws:sns:us-east-1:012345678999:my_sns_topic`
- `parameter_group_name` – (Optional) Name of the parameter group to associate with this DAX cluster
- `maintenance_window` – (Optional) Specifies the weekly time range for when maintenance on the cluster is performed. The format is `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Example: `sun:05:00-sun:09:00`
- `security_group_ids` – (Optional) One or more VPC security groups associated with the cluster
- `server_side_encryption` - (Optional) Encrypt at rest options
- `subnet_group_name` – (Optional) Name of the subnet group to be used for the cluster
- `tags` - (Optional) A mapping of tags to assign to the resource

The `server_side_encryption` object supports the following:

- `enabled` - (Optional) Whether to enable encryption at rest. Defaults to `false`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the DAX cluster
- `nodes` - List of node objects including `id`, `address`, `port` and `availability_zone`. Referenceable e.g. as `${aws_dax_cluster.test.nodes.0.address}`
- `configuration_endpoint` - The configuration endpoint for this DAX cluster, consisting of a DNS name and a port number
- `cluster_address` - The DNS name of the DAX cluster without the port appended

- `port` - The port used by the configuration endpoint

» Timeouts

`aws_dax_cluster` provides the following Timeouts configuration options:

- `create` - (Default `45 minutes`) Used for creating a DAX cluster
- `update` - (Default `45 minutes`) Used for cluster modifications
- `delete` - (Default `90 minutes`) Used for destroying a DAX cluster

» Import

DAX Clusters can be imported using the `cluster_id`, e.g.

```
$ terraform import aws_dax_cluster.my_cluster my_cluster
```

» Resource: aws_dax_parameter_group

Provides a DAX Parameter Group resource.

» Example Usage

```
resource "aws_dax_parameter_group" "example" {
  name = "example"

  parameters {
    name  = "query-ttl-millis"
    value = "100000"
  }

  parameters {
    name  = "record-ttl-millis"
    value = "100000"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` – (Required) The name of the parameter group.

- `description` - (Optional, ForceNew) A description of the parameter group.
- `parameters` – (Optional) The parameters of the parameter group.

» parameters

`parameters` supports the following:

- `name` - (Required) The name of the parameter.
- `value` - (Required) The value for the parameter.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the parameter group.

» Import

DAX Parameter Group can be imported using the `name`, e.g.

```
$ terraform import aws_dax_parameter_group.example my_dax_pg
```

» Resource: aws_dax_subnet_group

Provides a DAX Subnet Group resource.

» Example Usage

```
resource "aws_dax_subnet_group" "example" {
  name       = "example"
  subnet_ids = ["${aws_subnet.example1.id}", "${aws_subnet.example2.id}"]
}
```

» Argument Reference

The following arguments are supported:

- `name` – (Required) The name of the subnet group.
- `description` - (Optional) A description of the subnet group.
- `subnet_ids` – (Required) A list of VPC subnet IDs for the subnet group.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the subnet group.
- `vpc_id` – VPC ID of the subnet group.

» Import

DAX Subnet Group can be imported using the `name`, e.g.

```
$ terraform import aws_dax_subnet_group.example my_dax_sg
```

» Resource: aws_docdb_cluster

Manages a DocDB Cluster.

Changes to a DocDB Cluster can occur when you manually change a parameter, such as `port`, and are reflected in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because a modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately (see documentation below).

Note: using `apply_immediately` can result in a brief downtime as the server reboots. **Note:** All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_docdb_cluster" "docdb" {
  cluster_identifier      = "my-docdb-cluster"
  engine                  = "docdb"
  master_username         = "foo"
  master_password         = "mustbeeightchars"
  backup_retention_period = 5
  preferred_backup_window = "07:00-09:00"
  skip_final_snapshot     = true
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- `apply_immediately` - (Optional) Specifies whether any cluster modifications are applied immediately, or during the next maintenance window. Default is `false`.
- `availability_zones` - (Optional) A list of EC2 Availability Zones that instances in the DB cluster can be created in.
- `backup_retention_period` - (Optional) The days to retain backups for. Default `1`
- `cluster_identifier_prefix` - (Optional, Forces new resource) Creates a unique cluster identifier beginning with the specified prefix. Conflicts with `cluster_identifer`.
- `cluster_identifier` - (Optional, Forces new resources) The cluster identifier. If omitted, Terraform will assign a random, unique identifier.
- `db_subnet_group_name` - (Optional) A DB subnet group to associate with this DB instance.
- `db_cluster_parameter_group_name` - (Optional) A cluster parameter group to associate with the cluster.
- `enabled_cloudwatch_logs_exports` - (Optional) List of log types to export to cloudwatch. If omitted, no logs will be exported. The following log types are supported: `audit`, `profiler`.
- `engine_version` - (Optional) The database engine version. Updating this argument results in an outage.
- `engine` - (Optional) The name of the database engine to be used for this DB cluster. Defaults to `docdb`. Valid Values: `docdb`
- `final_snapshot_identifier` - (Optional) The name of your final DB snapshot when this DB cluster is deleted. If omitted, no final snapshot will be made.
- `kms_key_id` - (Optional) The ARN for the KMS encryption key. When specifying `kms_key_id`, `storage_encrypted` needs to be set to true.
- `master_password` - (Required unless a `snapshot_identifier` is provided) Password for the master DB user. Note that this may show up in logs, and it will be stored in the state file. Please refer to the DocDB Naming Constraints.
- `master_username` - (Required unless a `snapshot_identifier` is provided) Username for the master DB user.
- `port` - (Optional) The port on which the DB accepts connections
- `preferred_backup_window` - (Optional) The daily time range during which automated backups are created if automated backups are enabled using the BackupRetentionPeriod parameter.Time in UTC Default: A 30-minute window selected at random from an 8-hour block of time per region. e.g. 04:00-09:00
- `skip_final_snapshot` - (Optional) Determines whether a final DB snapshot is created before the DB cluster is deleted. If true is specified, no DB snapshot is created. If false is specified, a DB snapshot is created before the DB cluster is deleted, using the value from

final_snapshot_identifier. Default is false.

- snapshot_identifier - (Optional) Specifies whether or not to create this cluster from a snapshot. You can use either the name or ARN when specifying a DB cluster snapshot, or the ARN when specifying a DB snapshot.
- storage_encrypted - (Optional) Specifies whether the DB cluster is encrypted. The default is false.
- tags - (Optional) A mapping of tags to assign to the DB cluster.
- vpc_security_group_ids - (Optional) List of VPC security groups to associate with the Cluster

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- arn - Amazon Resource Name (ARN) of cluster
- cluster_members – List of DocDB Instances that are a part of this cluster
- cluster_resource_id - The DocDB Cluster Resource ID
- endpoint - The DNS address of the DocDB instance
- hosted_zone_id - The Route53 Hosted Zone ID of the endpoint
- id - The DocDB Cluster Identifier
- maintenance_window - The instance maintenance window
- reader_endpoint - A read-only endpoint for the DocDB cluster, automatically load-balanced across replicas

» Timeouts

aws_docdb_cluster provides the following Timeouts configuration options:

- create - (Default 120 minutes) Used for Cluster creation
- update - (Default 120 minutes) Used for Cluster modifications
- delete - (Default 120 minutes) Used for destroying cluster. This includes any cleanup task during the destroying process.

» Import

DocDB Clusters can be imported using the cluster_identifier, e.g.

```
$ terraform import aws_docdb_cluster.docdb_cluster docdb-prod-cluster
```

» Resource: aws_docdb_cluster_instance

Provides an DocDB Cluster Resource Instance. A Cluster Instance Resource defines attributes that are specific to a single instance in a DocDB Cluster.

You do not designate a primary and subsequent replicas. Instead, you simply add DocDB Instances and DocDB manages the replication. You can use the count meta-parameter to make multiple instances and join them all to the same DocDB Cluster, or you may specify different Cluster Instance resources with various `instance_class` sizes.

» Example Usage

```
resource "aws_docdb_cluster_instance" "cluster_instances" {
  count              = 2
  identifier         = "docdb-cluster-demo-${count.index}"
  cluster_identifier = "${aws_docdb_cluster.default.id}"
  instance_class     = "db.r5.large"
}

resource "aws_docdb_cluster" "default" {
  cluster_identifier = "docdb-cluster-demo"
  availability_zones = ["us-west-2a", "us-west-2b", "us-west-2c"]
  master_username    = "foo"
  master_password    = "barbut8chars"
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- `apply_immediately` - (Optional) Specifies whether any database modifications are applied immediately, or during the next maintenance window. Default is`false`.
- `auto_minor_version_upgrade` - (Optional) Indicates that minor engine upgrades will be applied automatically to the DB instance during the maintenance window. Default `true`.
- `availability_zone` - (Optional, Computed) The EC2 Availability Zone that the DB instance is created in. See docs about the details.
- `cluster_identifier` - (Required) The identifier of the `aws_docdb_cluster` in which to launch this instance.
- `engine` - (Optional) The name of the database engine to be used for the DocDB instance. Defaults to `docdb`. Valid Values: `docdb`.
- `identifier` - (Optional, Forces new resource) The indentifier for the DocDB instance, if omitted, Terraform will assign a random, unique identifier.

- `identifier_prefix` - (Optional, Forces new resource) Creates a unique identifier beginning with the specified prefix. Conflicts with `identifer`.
- `instance_class` - (Required) The instance class to use. For details on CPU and memory, see Scaling for DocDB Instances. DocDB currently supports the below instance classes. Please see AWS Documentation for complete details.
  - db.r4.large
  - db.r4.xlarge
  - db.r4.2xlarge
  - db.r4.4xlarge
  - db.r4.8xlarge
  - db.r4.16xlarge
- `preferred_maintenance_window` - (Optional) The window to perform maintenance in. Syntax: "ddd:hh24:mi-ddd:hh24:mi". Eg: "Mon:00:00-Mon:03:00".
- `promotion_tier` - (Optional) Default 0. Failover Priority setting on instance level. The reader who has lower tier has higher priority to get promoter to writer.
- `tags` - (Optional) A mapping of tags to assign to the instance.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of cluster instance
- `db_subnet_group_name` - The DB subnet group to associate with this DB instance.
- `dbi_resource_id` - The region-unique, immutable identifier for the DB instance.
- `endpoint` - The DNS address for this instance. May not be writable
- `engine_version` - The database engine version
- `kms_key_id` - The ARN for the KMS encryption key if one is set to the cluster.
- `port` - The database port
- `preferred_backup_window` - The daily time range during which automated backups are created if automated backups are enabled.
- `storage_encrypted` - Specifies whether the DB cluster is encrypted.
- `writer` – Boolean indicating if this instance is writable. `False` indicates this instance is a read replica.
- `ca_cert_identifier` - (Optional) The identifier of the CA certificate for the DB instance.

» Timeouts

`aws_docdb_cluster_instance` provides the following Timeouts configuration options:

- `create` - (Default `90 minutes`) Used for Creating Instances, Replicas, and restoring from Snapshots
- `update` - (Default `90 minutes`) Used for Database modifications
- `delete` - (Default `90 minutes`) Used for destroying databases. This includes the time required to take snapshots

» Import

DocDB Cluster Instances can be imported using the `identifier`, e.g.

```
$ terraform import aws_docdb_cluster_instance.prod_instance_1 aurora-cluster-instance-1
```

» Resource: aws_docdb_cluster_parameter_group

Manages a DocumentDB Cluster Parameter Group

» Example Usage

```
resource "aws_docdb_cluster_parameter_group" "example" {
  family      = "docdb3.6"
  name        = "example"
  description = "docdb cluster parameter group"

  parameter {
    name  = "tls"
    value = "enabled"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the documentDB cluster parameter group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.

- `family` - (Required, Forces new resource) The family of the documentDB cluster parameter group.
- `description` - (Optional, Forces new resource) The description of the documentDB cluster parameter group. Defaults to ”Managed by Terraform”.
- `parameter` - (Optional) A list of documentDB parameters to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Parameter blocks support the following:

- `name` - (Required) The name of the documentDB parameter.
- `value` - (Required) The value of the documentDB parameter.
- `apply_method` - (Optional) Valid values are `immediate` and `pending-reboot`. Defaults to `pending-reboot`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The documentDB cluster parameter group name.
- `arn` - The ARN of the documentDB cluster parameter group.

» Import

DocumentDB Cluster Parameter Groups can be imported using the `name`, e.g.

```
$ terraform import aws_docdb_cluster_parameter_group.cluster_pg production-pg-1
```

» Resource: aws_docdb_cluster_snapshot

Manages a DocDB database cluster snapshot for DocDB clusters.

» Example Usage

```
resource "aws_docdb_cluster_snapshot" "example" {
  db_cluster_identifier          = "${aws_docdb_cluster.example.id}"
  db_cluster_snapshot_identifier = "resourcetestsnapshot1234"
}
```

» Argument Reference

The following arguments are supported:

- `db_cluster_identifier` - (Required) The DocDB Cluster Identifier from which to take the snapshot.
- `db_cluster_snapshot_identifier` - (Required) The Identifier for the snapshot.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `availability_zones` - List of EC2 Availability Zones that instances in the DocDB cluster snapshot can be restored in.
- `db_cluster_snapshot_arn` - The Amazon Resource Name (ARN) for the DocDB Cluster Snapshot.
- `engine` - Specifies the name of the database engine.
- `engine_version` - Version of the database engine for this DocDB cluster snapshot.
- `kms_key_id` - If storage_encrypted is true, the AWS KMS key identifier for the encrypted DocDB cluster snapshot.
- `port` - Port that the DocDB cluster was listening on at the time of the snapshot.
- `source_db_cluster_snapshot_identifier` - The DocDB Cluster Snapshot Arn that the DocDB Cluster Snapshot was copied from. It only has value in case of cross customer or cross region copy.
- `storage_encrypted` - Specifies whether the DocDB cluster snapshot is encrypted.
- `status` - The status of this DocDB Cluster Snapshot.
- `vpc_id` - The VPC ID associated with the DocDB cluster snapshot.

» Timeouts

`aws_docdb_cluster_snapshot` provides the following Timeouts configuration options:

- `create` - (Default `20m`) How long to wait for the snapshot to be available.

» Import

`aws_docdb_cluster_snapshot` can be imported by using the cluster snapshot identifier, e.g.

```
$ terraform import aws_docdb_cluster_snapshot.example my-cluster-snapshot
```

» Resource: aws_docdb_subnet_group

Provides an DocumentDB subnet group resource.

» Example Usage

```
resource "aws_docdb_subnet_group" "default" {
  name       = "main"
  subnet_ids = ["${aws_subnet.frontend.id}", "${aws_subnet.backend.id}"]

  tags = {
    Name = "My docdb subnet group"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the docDB subnet group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `description` - (Optional) The description of the docDB subnet group. Defaults to "Managed by Terraform".
- `subnet_ids` - (Required) A list of VPC subnet IDs.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The docDB subnet group name.
- `arn` - The ARN of the docDB subnet group.

» Import

DocumentDB Subnet groups can be imported using the `name`, e.g.

```
$ terraform import aws_docdb_subnet_group.default production-subnet-group
```

» Data Source: aws_ami

Use this data source to get the ID of a registered AMI for use in other resources.

» Example Usage

```
data "aws_ami" "example" {
  executable_users = ["self"]
  most_recent      = true
  name_regex       = "^myami-\\d{3}"
  owners           = ["self"]

  filter {
    name   = "name"
    values = ["myami-*"]
  }

  filter {
    name   = "root-device-type"
    values = ["ebs"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }
}
```

» Argument Reference

- `owners` - (Required) List of AMI owners to limit search. At least 1 value must be specified. Valid values: an AWS account ID, `self` (the current account), or an AWS owner alias (e.g. `amazon`, `aws-marketplace`, `microsoft`).
- `most_recent` - (Optional) If more than one result is returned, use the most recent AMI.
- `executable_users` - (Optional) Limit search to users with *explicit* launch permission on the image. Valid items are the numeric account ID or `self`.
- `filter` - (Optional) One or more name/value pairs to filter off of. There are several valid keys, for a full reference, check out describe-images in the AWS CLI reference.

- `name_regex` - (Optional) A regex string to apply to the AMI list returned by AWS. This allows more advanced filtering not supported from the AWS API. This filtering is done locally on what AWS returns, and could have a performance impact if the result is large. It is recommended to combine this with other options to narrow down the list AWS returns.

NOTE: If more or less than a single match is returned by the search, Terraform will fail. Ensure that your search is specific enough to return a single AMI ID only, or use `most_recent` to choose the most recent one. If you want to match multiple AMIs, use the `aws_ami_ids` data source instead.

» Attributes Reference

`id` is set to the ID of the found AMI. In addition, the following attributes are exported:

NOTE: Some values are not always set and may not be available for interpolation.

- `architecture` - The OS architecture of the AMI (ie: `i386` or `x86_64`).
- `block_device_mappings` - The block device mappings of the AMI.
  - `block_device_mappings.#.device_name` - The physical name of the device.
  - `block_device_mappings.#.ebs.delete_on_termination` - `true` if the EBS volume will be deleted on termination.
  - `block_device_mappings.#.ebs.encrypted` - `true` if the EBS volume is encrypted.
  - `block_device_mappings.#.ebs.iops` - `0` if the EBS volume is not a provisioned IOPS image, otherwise the supported IOPS count.
  - `block_device_mappings.#.ebs.snapshot_id` - The ID of the snapshot.
  - `block_device_mappings.#.ebs.volume_size` - The size of the volume, in GiB.
  - `block_device_mappings.#.ebs.volume_type` - The volume type.
  - `block_device_mappings.#.no_device` - Suppresses the specified device included in the block device mapping of the AMI.
  - `block_device_mappings.#.virtual_name` - The virtual device name (for instance stores).
- `creation_date` - The date and time the image was created.
- `description` - The description of the AMI that was provided during image creation.
- `hypervisor` - The hypervisor type of the image.
- `image_id` - The ID of the AMI. Should be the same as the resource `id`.
- `image_location` - The location of the AMI.
- `image_owner_alias` - The AWS account alias (for example, `amazon`, `self`) or the AWS account ID of the AMI owner.

- image_type - The type of image.
- kernel_id - The kernel associated with the image, if any. Only applicable for machine images.
- name - The name of the AMI that was provided during image creation.
- owner_id - The AWS account ID of the image owner.
- platform - The value is Windows for Windows AMIs; otherwise blank.
- product_codes - Any product codes associated with the AMI.
  - product_codes.#.product_code_id - The product code.
  - product_codes.#.product_code_type - The type of product code.
- public - true if the image has public launch permissions.
- ramdisk_id - The RAM disk associated with the image, if any. Only applicable for machine images.
- root_device_name - The device name of the root device.
- root_device_type - The type of root device (ie: ebs or instance-store).
- root_snapshot_id - The snapshot id associated with the root device, if any (only applies to ebs root devices).
- sriov_net_support - Specifies whether enhanced networking is enabled.
- state - The current state of the AMI. If the state is available, the image is successfully registered and can be used to launch an instance.
- state_reason - Describes a state change. Fields are UNSET if not available.
  - state_reason.code - The reason code for the state change.
  - state_reason.message - The message for the state change.
- tags - Any tags assigned to the image.
  - tags.#.key - The key name of the tag.
  - tags.#.value - The value of the tag.
- virtualization_type - The type of virtualization of the AMI (ie: hvm or paravirtual).

» Data Source: aws_ami_ids

Use this data source to get a list of AMI IDs matching the specified criteria.

» Example Usage

```
data "aws_ami_ids" "ubuntu" {
  owners = ["099720109477"]

  filter {
    name   = "name"
    values = ["ubuntu/images/ubuntu-*-*-amd64-server-*"]
  }
}
```

» Argument Reference

- `owners` - (Required) List of AMI owners to limit search. At least 1 value must be specified. Valid values: an AWS account ID, `self` (the current account), or an AWS owner alias (e.g. `amazon`, `aws-marketplace`, `microsoft`).
- `executable_users` - (Optional) Limit search to users with *explicit* launch permission on the image. Valid items are the numeric account ID or `self`.
- `filter` - (Optional) One or more name/value pairs to filter off of. There are several valid keys, for a full reference, check out describe-images in the AWS CLI reference.
- `name_regex` - (Optional) A regex string to apply to the AMI list returned by AWS. This allows more advanced filtering not supported from the AWS API. This filtering is done locally on what AWS returns, and could have a performance impact if the result is large. It is recommended to combine this with other options to narrow down the list AWS returns.
- `sort_ascending` - (Defaults to `false`) Used to sort AMIs by creation time.

» Attributes Reference

`ids` is set to the list of AMI IDs, sorted by creation time according to `sort_ascending`.

» Data Source: aws_ebs_default_kms_key

Use this data source to get the default EBS encryption KMS key in the current region.

» Example Usage

```
data "aws_ebs_default_kms_key" "current" { }

resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"

  encrypted         = true
  kms_key_id        = "${data.aws_ebs_default_kms_key.current.key_arn}"

}
```

» Attributes Reference

The following attributes are exported: * `key_arn` - Amazon Resource Name (ARN) of the default KMS key uses to encrypt an EBS volume in this region when no key is specified in an API call that creates the volume and encryption by default is enabled.

» Data Source: aws_ebs_encryption_by_default

Provides a way to check whether default EBS encryption is enabled for your AWS account in the current AWS region.

» Example Usage

```
data "aws_ebs_encryption_by_default" "current" { }
```

» Attributes Reference

The following attributes are exported:

- `enabled` - Whether or not default EBS encryption is enabled. Returns as `true` or `false`.

» Data Source: aws_ebs_snapshot

Use this data source to get information about an EBS Snapshot for use when provisioning EBS Volumes

» Example Usage

```
data "aws_ebs_snapshot" "ebs_volume" {
  most_recent = true
  owners      = ["self"]

  filter {
    name   = "volume-size"
    values = ["40"]
  }

  filter {
    name   = "tag:Name"
    values = ["Example"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `most_recent` - (Optional) If more than one result is returned, use the most recent snapshot.
- `owners` - (Optional) Returns the snapshots owned by the specified owner id. Multiple owners can be specified.
- `snapshot_ids` - (Optional) Returns information on a specific snapshot_id.
- `restorable_by_user_ids` - (Optional) One or more AWS accounts IDs that can create volumes from the snapshot.
- `filter` - (Optional) One or more name/value pairs to filter off of. There are several valid keys, for a full reference, check out describe-snapshots in the AWS CLI reference.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The snapshot ID (e.g. snap-59fcb34e).
- `snapshot_id` - The snapshot ID (e.g. snap-59fcb34e).
- `description` - A description for the snapshot
- `owner_id` - The AWS account ID of the EBS snapshot owner.
- `owner_alias` - Value from an Amazon-maintained list (`amazon`, `aws-marketplace`, `microsoft`) of snapshot owners.
- `volume_id` - The volume ID (e.g. vol-59fcb34e).
- `encrypted` - Whether the snapshot is encrypted.
- `volume_size` - The size of the drive in GiBs.
- `kms_key_id` - The ARN for the KMS encryption key.
- `data_encryption_key_id` - The data encryption key identifier for the snapshot.
- `state` - The snapshot state.
- `tags` - A mapping of tags for the resource.

» Data Source: aws_ebs_snapshot_ids

Use this data source to get a list of EBS Snapshot IDs matching the specified criteria.

» Example Usage

```
data "aws_ebs_snapshot_ids" "ebs_volumes" {
  owners = ["self"]

  filter {
    name   = "volume-size"
    values = ["40"]
  }

  filter {
    name   = "tag:Name"
    values = ["Example"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `owners` - (Optional) Returns the snapshots owned by the specified owner id. Multiple owners can be specified.
- `restorable_by_user_ids` - (Optional) One or more AWS accounts IDs that can create volumes from the snapshot.
- `filter` - (Optional) One or more name/value pairs to filter off of. There are several valid keys, for a full reference, check out describe-volumes in the AWS CLI reference.

» Attributes Reference

`ids` is set to the list of EBS snapshot IDs, sorted by creation time in descending order.

» Data Source: aws_ebs_volume

Use this data source to get information about an EBS volume for use in other resources.

» Example Usage

```
data "aws_ebs_volume" "ebs_volume" {
  most_recent = true

  filter {
    name   = "volume-type"
    values = ["gp2"]
  }

  filter {
    name   = "tag:Name"
    values = ["Example"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `most_recent` - (Optional) If more than one result is returned, use the most recent Volume.
- `filter` - (Optional) One or more name/value pairs to filter off of. There are several valid keys, for a full reference, check out describe-volumes in the AWS CLI reference.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The volume ID (e.g. vol-59fcb34e).
- `volume_id` - The volume ID (e.g. vol-59fcb34e).
- `arn` - The volume ARN (e.g. arn:aws:ec2:us-east-1:0123456789012:volume/vol-59fcb34e).
- `availability_zone` - The AZ where the EBS volume exists.
- `encrypted` - Whether the disk is encrypted.
- `iops` - The amount of IOPS for the disk.
- `size` - The size of the drive in GiBs.
- `snapshot_id` - The snapshot_id the EBS volume is based off.

- `volume_type` - The type of EBS volume.
- `kms_key_id` - The ARN for the KMS encryption key.
- `tags` - A mapping of tags for the resource.

» Data Source: aws_ec2_transit_gateway

Get information on an EC2 Transit Gateway.

» Example Usage

» By Filter

```
data "aws_ec2_transit_gateway" "example" {
  filter {
    name   = "options.amazon-side-asn"
    values = ["64512"]
  }
}
```

» By Identifier

```
data "aws_ec2_transit_gateway" "example" {
  id = "tgw-12345678"
}
```

» Argument Reference

The following arguments are supported:

- `filter` - (Optional) One or more configuration blocks containing name-values filters. Detailed below.
- `id` - (Optional) Identifier of the EC2 Transit Gateway.

» filter Argument Reference

- `name` - (Required) Name of the filter.
- `values` - (Required) List of one or more values for the filter.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `amazon_side_asn` - Private Autonomous System Number (ASN) for the Amazon side of a BGP session
- `arn` - EC2 Transit Gateway Amazon Resource Name (ARN)
- `association_default_route_table_id` - Identifier of the default association route table
- `auto_accept_shared_attachments` - Whether resource attachment requests are automatically accepted.
- `default_route_table_association` - Whether resource attachments are automatically associated with the default association route table.
- `default_route_table_propagation` - Whether resource attachments automatically propagate routes to the default propagation route table.
- `description` - Description of the EC2 Transit Gateway
- `dns_support` - Whether DNS support is enabled.
- `id` - EC2 Transit Gateway identifier
- `owner_id` - Identifier of the AWS account that owns the EC2 Transit Gateway
- `propagation_default_route_table_id` - Identifier of the default propagation route table.
- `tags` - Key-value tags for the EC2 Transit Gateway
- `vpn_ecmp_support` - Whether VPN Equal Cost Multipath Protocol support is enabled.

» Data Source: aws_ec2_transit_gateway_dx_gateway_attachmen

Get information on an EC2 Transit Gateway's attachment to a Direct Connect Gateway.

» Example Usage

» By Transit Gateway and Direct Connect Gateway Identifiers

```
data "aws_ec2_transit_gateway_dx_gateway_attachment" "example" {
  transit_gateway_id = "${aws_ec2_transit_gateway.example.id}"
  dx_gateway_id      = "${aws_dx_gateway.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `transit_gateway_id` - (Required) Identifier of the EC2 Transit Gateway.
- `dx_gateway_id` - (Required) Identifier of the Direct Connect Gateway.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - EC2 Transit Gateway Attachment identifier
- `tags` - Key-value tags for the EC2 Transit Gateway Attachment

» Data Source: aws_ec2_transit_gateway_route_table

Get information on an EC2 Transit Gateway Route Table.

» Example Usage

» By Filter

```
data "aws_ec2_transit_gateway_route_table" "example" {
  filter {
    name   = "default-association-route-table"
    values = ["true"]
  }

  filter {
    name   = "transit-gateway-id"
    values = ["tgw-12345678"]
  }
}
```

» By Identifier

```
data "aws_ec2_transit_gateway_route_table" "example" {
  id = "tgw-rtb-12345678"
}
```

» Argument Reference

The following arguments are supported:

- `filter` - (Optional) One or more configuration blocks containing name-values filters. Detailed below.

- `id` - (Optional) Identifier of the EC2 Transit Gateway Route Table.

» filter Argument Reference

- `name` - (Required) Name of the filter.
- `values` - (Required) List of one or more values for the filter.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `default_association_route_table` - Boolean whether this is the default association route table for the EC2 Transit Gateway
- `default_propagation_route_table` - Boolean whether this is the default propagation route table for the EC2 Transit Gateway
- `id` - EC2 Transit Gateway Route Table identifier
- `transit_gateway_id` - EC2 Transit Gateway identifier
- `tags` - Key-value tags for the EC2 Transit Gateway Route Table

» Data Source: aws_ec2_transit_gateway_vpc_attachment

Get information on an EC2 Transit Gateway VPC Attachment.

» Example Usage

» By Filter

```
data "aws_ec2_transit_gateway_vpc_attachment" "example" {
  filter {
    name   = "vpc-id"
    values = ["vpc-12345678"]
  }
}
```

» By Identifier

```
data "aws_ec2_transit_gateway_vpc_attachment" "example" {
  id = "tgw-attach-12345678"
}
```

» Argument Reference

The following arguments are supported:

- `filter` - (Optional) One or more configuration blocks containing name-values filters. Detailed below.
- `id` - (Optional) Identifier of the EC2 Transit Gateway VPC Attachment.

» filter Argument Reference

- `name` - (Required) Name of the filter.
- `values` - (Required) List of one or more values for the filter.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `dns_support` - Whether DNS support is enabled.
- `id` - EC2 Transit Gateway VPC Attachment identifier
- `ipv6_support` - Whether IPv6 support is enabled.
- `subnet_ids` - Identifiers of EC2 Subnets.
- `transit_gateway_id` - EC2 Transit Gateway identifier
- `tags` - Key-value tags for the EC2 Transit Gateway VPC Attachment
- `vpc_id` - Identifier of EC2 VPC.
- `vpc_owner_id` - Identifier of the AWS account that owns the EC2 VPC.

» Data Source: aws_ec2_transit_gateway_vpn_attachment

Get information on an EC2 Transit Gateway VPN Attachment.

» Example Usage

» By Transit Gateway and VPN Connection Identifiers

```
data "aws_ec2_transit_gateway_vpn_attachment" "example" {
  transit_gateway_id = "${aws_ec2_transit_gateway.example.id}"
  vpn_connection_id  = "${aws_vpn_connection.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `transit_gateway_id` - (Required) Identifier of the EC2 Transit Gateway.
- `vpn_connection_id` - (Required) Identifier of the EC2 VPN Connection.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - EC2 Transit Gateway VPN Attachment identifier
- `tags` - Key-value tags for the EC2 Transit Gateway VPN Attachment

» Data Source: aws_eip

`aws_eip` provides details about a specific Elastic IP.

» Example Usage

» Search By Allocation ID (VPC only)

```
data "aws_eip" "by_allocation_id" {
  id = "eipalloc-12345678"
}
```

» Search By Filters (EC2-Classic or VPC)

```
data "aws_eip" "by_filter" {
  filter {
    name   = "tag:Name"
    values = ["exampleNameTagValue"]
  }
}
```

» Search By Public IP (EC2-Classic or VPC)

```
data "aws_eip" "by_public_ip" {
  public_ip = "1.2.3.4"
}
```

» Search By Tags (EC2-Classic or VPC)

```
data "aws_eip" "by_tags" {
  tags = {
    Name = "exampleNameTagValue"
  }
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available Elastic IPs in the current region. The given filters must match exactly one Elastic IP whose data will be exported as attributes.

- `filter` - (Optional) One or more name/value pairs to use as filters. There are several valid keys, for a full reference, check out the EC2 API Reference.
- `id` - (Optional) The allocation id of the specific VPC EIP to retrieve. If a classic EIP is required, do NOT set `id`, only set `public_ip`
- `public_ip` - (Optional) The public IP of the specific EIP to retrieve.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired Elastic IP

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `association_id` - The ID representing the association of the address with an instance in a VPC.
- `domain` - Indicates whether the address is for use in EC2-Classic (standard) or in a VPC (vpc).
- `id` - If VPC Elastic IP, the allocation identifier. If EC2-Classic Elastic IP, the public IP address.
- `instance_id` - The ID of the instance that the address is associated with (if any).
- `network_interface_id` - The ID of the network interface.
- `network_interface_owner_id` - The ID of the AWS account that owns the network interface.
- `private_ip` - The private IP address associated with the Elastic IP address.
- `private_dns` - The Private DNS associated with the Elastic IP address.
- `public_ip` - Public IP address of Elastic IP.
- `public_dns` - Public DNS associated with the Elastic IP address.
- `public_ipv4_pool` - The ID of an address pool.
- `tags` - Key-value map of tags associated with Elastic IP.

Note: The data source computes the `public_dns` and `private_dns` attributes according to the VPC DNS Guide as they are not available with the EC2 API.

» Data Source: aws_instance

Use this data source to get the ID of an Amazon EC2 Instance for use in other resources.

» Example Usage

```
data "aws_instance" "foo" {
  instance_id = "i-instanceid"

  filter {
    name   = "image-id"
    values = ["ami-xxxxxxxx"]
  }

  filter {
    name   = "tag:Name"
    values = ["instance-name-tag"]
  }
}
```

» Argument Reference

- `instance_id` - (Optional) Specify the exact Instance ID with which to populate the data source.
- `instance_tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired Instance.
- `filter` - (Optional) One or more name/value pairs to use as filters. There are several valid keys, for a full reference, check out describe-instances in the AWS CLI reference.
- `get_password_data` - (Optional) If true, wait for password data to become available and retrieve it. Useful for getting the administrator password for instances running Microsoft Windows. The password data is exported to the `password_data` attribute. See GetPasswordData for more information.
- `get_user_data` - (Optional) Retrieve Base64 encoded User Data contents into the `user_data_base64` attribute. A SHA-1 hash of the User Data

contents will always be present in the `user_data` attribute. Defaults to `false`.

NOTE: At least one of `filter`, `instance_tags`, or `instance_id` must be specified.

NOTE: If anything other than a single match is returned by the search, Terraform will fail. Ensure that your search is specific enough to return a single Instance ID only.

» Attributes Reference

`id` is set to the ID of the found Instance. In addition, the following attributes are exported:

NOTE: Some values are not always set and may not be available for interpolation.

- `ami` - The ID of the AMI used to launch the instance.
- `arn` - The ARN of the instance.
- `associate_public_ip_address` - Whether or not the Instance is associated with a public IP address or not (Boolean).
- `availability_zone` - The availability zone of the Instance.
- `ebs_block_device` - The EBS block device mappings of the Instance.
  - `delete_on_termination` - If the EBS volume will be deleted on termination.
  - `device_name` - The physical name of the device.
  - `encrypted` - If the EBS volume is encrypted.
  - `iops` - `0` If the EBS volume is not a provisioned IOPS image, otherwise the supported IOPS count.
  - `kms_key_arn` - Amazon Resource Name (ARN) of KMS Key, if EBS volume is encrypted.
  - `snapshot_id` - The ID of the snapshot.
  - `volume_size` - The size of the volume, in GiB.
  - `volume_type` - The volume type.
- `ebs_optimized` - Whether the Instance is EBS optimized or not (Boolean).
- `ephemeral_block_device` - The ephemeral block device mappings of the Instance.
  - `device_name` - The physical name of the device.
  - `no_device` - Whether the specified device included in the device mapping was suppressed or not (Boolean).
  - `virtual_name` - The virtual device name.
- `iam_instance_profile` - The name of the instance profile associated with the Instance.
- `ipv6_addresses` - The IPv6 addresses associated to the Instance, if applicable. **NOTE**: Unlike the IPv4 address, this doesn't change if you attach

an EIP to the instance.

- `instance_state` - The state of the instance. One of: `pending`, `running`, `shutting-down`, `terminated`, `stopping`, `stopped`. See Instance Lifecycle for more information.
- `instance_type` - The type of the Instance.
- `key_name` - The key name of the Instance.
- `monitoring` - Whether detailed monitoring is enabled or disabled for the Instance (Boolean).
- `network_interface_id` - The ID of the network interface that was created with the Instance.
- `password_data` - Base-64 encoded encrypted password data for the instance. Useful for getting the administrator password for instances running Microsoft Windows. This attribute is only exported if `get_password_data` is true. See GetPasswordData for more information.
- `placement_group` - The placement group of the Instance.
- `private_dns` - The private DNS name assigned to the Instance. Can only be used inside the Amazon EC2, and only available if you've enabled DNS hostnames for your VPC.
- `private_ip` - The private IP address assigned to the Instance.
- `public_dns` - The public DNS name assigned to the Instance. For EC2-VPC, this is only available if you've enabled DNS hostnames for your VPC.
- `public_ip` - The public IP address assigned to the Instance, if applicable. **NOTE**: If you are using an `aws_eip` with your instance, you should refer to the EIP's address directly and not use `public_ip`, as this field will change after the EIP is attached.
- `root_block_device` - The root block device mappings of the Instance
  - `delete_on_termination` - If the root block device will be deleted on termination.
  - `encrypted` - If the EBS volume is encrypted.
  - `iops` - `0` If the volume is not a provisioned IOPS image, otherwise the supported IOPS count.
  - `kms_key_arn` - Amazon Resource Name (ARN) of KMS Key, if EBS volume is encrypted.
  - `volume_size` - The size of the volume, in GiB.
  - `volume_type` - The type of the volume.
- `security_groups` - The associated security groups.
- `source_dest_check` - Whether the network interface performs source/destination checking (Boolean).
- `subnet_id` - The VPC subnet ID.
- `user_data` - SHA-1 hash of User Data supplied to the Instance.
- `user_data_base64` - Base64 encoded contents of User Data supplied to the Instance. Valid UTF-8 contents can be decoded with the `base64decode` function. This attribute is only exported if `get_user_data` is true.
- `tags` - A mapping of tags assigned to the Instance.
- `tenancy` - The tenancy of the instance: `dedicated`, `default`, `host`.

- `host_id` - The Id of the dedicated host the instance will be assigned to.
- `vpc_security_group_ids` - The associated security groups in a non-default VPC.
- `credit_specification` - The credit specification of the Instance.

» Data Source: aws_instances

Use this data source to get IDs or IPs of Amazon EC2 instances to be referenced elsewhere, e.g. to allow easier migration from another management solution or to make it easier for an operator to connect through bastion host(s).

Note: It's a best practice to expose instance details via outputs and remote state and **use `terraform_remote_state` data source instead** if you manage referenced instances via Terraform.

Note: It's strongly discouraged to use this data source for querying ephemeral instances (e.g. managed via autoscaling group), as the output may change at any time and you'd need to re-run `apply` every time an instance comes up or dies.

» Example Usage

```
data "aws_instances" "test" {
  instance_tags = {
    Role = "HardWorker"
  }

  filter {
    name   = "instance.group-id"
    values = ["sg-12345678"]
  }

  instance_state_names = ["running", "stopped"]
}

resource "aws_eip" "test" {
  count    = "${length(data.aws_instances.test.ids)}"
  instance = "${data.aws_instances.test.ids[count.index]}"
}
```

» Argument Reference

- `instance_tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on desired instances.
- `instance_state_names` - (Optional) A list of instance states that should be applicable to the desired instances. The permitted values are: `pending, running, shutting-down, stopped, stopping, terminated`. The default value is `running`.
- `filter` - (Optional) One or more name/value pairs to use as filters. There are several valid keys, for a full reference, check out describe-instances in the AWS CLI reference.

» Attributes Reference

- `ids` - IDs of instances found through the filter
- `private_ips` - Private IP addresses of instances found through the filter
- `public_ips` - Public IP addresses of instances found through the filter

» Data Source: aws_launch_template

Provides information about a Launch Template.

» Example Usage

```
data "aws_launch_template" "default" {
  name = "my-launch-template"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the launch template.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the launch template.
- `id` - The ID of the launch template.

- `default_version` - The default version of the launch template.
- `latest_version` - The latest version of the launch template.
- `description` - Description of the launch template.
- `block_device_mappings` - Specify volumes to attach to the instance besides the volumes specified by the AMI.
- `credit_specification` - Customize the credit specification of the instance. See Credit Specification below for more details.
- `disable_api_termination` - If `true`, enables EC2 Instance Termination Protection
- `ebs_optimized` - If `true`, the launched EC2 instance will be EBS-optimized.
- `elastic_gpu_specifications` - The elastic GPU to attach to the instance. See Elastic GPU below for more details.
- `iam_instance_profile` - The IAM Instance Profile to launch the instance with. See Instance Profile below for more details.
- `image_id` - The AMI from which to launch the instance.
- `instance_initiated_shutdown_behavior` - Shutdown behavior for the instance. Can be `stop` or `terminate`. (Default: `stop`).
- `instance_market_options` - The market (purchasing) option for the instance. below for details.
- `instance_type` - The type of the instance.
- `kernel_id` - The kernel ID.
- `key_name` - The key name to use for the instance.
- `monitoring` - The monitoring option for the instance.
- `network_interfaces` - Customize network interfaces to be attached at instance boot time. See Network Interfaces below for more details.
- `placement` - The placement of the instance.
- `ram_disk_id` - The ID of the RAM disk.
- `security_group_names` - A list of security group names to associate with. If you are creating Instances in a VPC, use `vpc_security_group_ids` instead.
- `vpc_security_group_ids` - A list of security group IDs to associate with.
- `tag_specifications` - The tags to apply to the resources during launch.
- `tags` - (Optional) A mapping of tags to assign to the launch template.
- `user_data` - The Base64-encoded user data to provide when launching the instance.

» Resource: aws_ami

The AMI resource allows the creation and management of a completely-custom *Amazon Machine Image* (AMI).

If you just want to duplicate an existing AMI, possibly copying it to another region, it's better to use `aws_ami_copy` instead.

If you just want to share an existing AMI with another AWS account, it's better to use `aws_ami_launch_permission` instead.

» Example Usage

```
# Create an AMI that will start a machine whose root device is backed by
# an EBS volume populated from a snapshot. It is assumed that such a snapshot
# already exists with the id "snap-xxxxxxxx".
resource "aws_ami" "example" {
  name                = "terraform-example"
  virtualization_type = "hvm"
  root_device_name    = "/dev/xvda"

  ebs_block_device {
    device_name = "/dev/xvda"
    snapshot_id = "snap-xxxxxxxx"
    volume_size = 8
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A region-unique name for the AMI.
- `description` - (Optional) A longer, human-readable description for the AMI.
- `ena_support` - (Optional) Specifies whether enhanced networking with ENA is enabled. Defaults to `false`.
- `root_device_name` - (Optional) The name of the root device (for example, `/dev/sda1`, or `/dev/xvda`).
- `virtualization_type` - (Optional) Keyword to choose what virtualization mode created instances will use. Can be either "paravirtual" (the default) or "hvm". The choice of virtualization type changes the set of further arguments that are required, as described below.
- `architecture` - (Optional) Machine architecture for created instances. Defaults to "x86_64".
- `ebs_block_device` - (Optional) Nested block describing an EBS block device that should be attached to created instances. The structure of this block is described below.
- `ephemeral_block_device` - (Optional) Nested block describing an ephemeral block device that should be attached to created instances. The structure of this block is described below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

When virtualization_type is ”paravirtual” the following additional arguments apply:

- image_location - (Required) Path to an S3 object containing an image manifest, e.g. created by the ec2-upload-bundle command in the EC2 command line tools.
- kernel_id - (Required) The id of the kernel image (AKI) that will be used as the paravirtual kernel in created instances.
- ramdisk_id - (Optional) The id of an initrd image (ARI) that will be used when booting the created instances.

When virtualization_type is ”hvm” the following additional arguments apply:

- sriov_net_support - (Optional) When set to ”simple” (the default), enables enhanced networking for created instances. No other value is supported at this time.

Nested ebs_block_device blocks have the following structure:

- device_name - (Required) The path at which the device is exposed to created instances.
- delete_on_termination - (Optional) Boolean controlling whether the EBS volumes created to support each created instance will be deleted once that instance is terminated.
- encrypted - (Optional) Boolean controlling whether the created EBS volumes will be encrypted. Can't be used with snapshot_id.
- iops - (Required only when volume_type is ”io1”) Number of I/O operations per second the created volumes will support.
- snapshot_id - (Optional) The id of an EBS snapshot that will be used to initialize the created EBS volumes. If set, the volume_size attribute must be at least as large as the referenced snapshot.
- volume_size - (Required unless snapshot_id is set) The size of created volumes in GiB. If snapshot_id is set and volume_size is omitted then the volume will have the same size as the selected snapshot.
- volume_type - (Optional) The type of EBS volume to create. Can be one of ”standard” (the default), ”io1” or ”gp2”.
- kms_key_id - (Optional) The full ARN of the AWS Key Management Service (AWS KMS) CMK to use when encrypting the snapshots of an image during a copy operation. This parameter is only required if you want to use a non-default CMK; if this parameter is not specified, the default CMK for EBS is used

Note: You can specify encrypted or snapshot_id but not both.

Nested ephemeral_block_device blocks have the following structure:

- device_name - (Required) The path at which the device is exposed to created instances.

- `virtual_name` - (Required) A name for the ephemeral device, of the form ”ephemeralN” where *N* is a volume number starting from zero.

» Timeouts

The `timeouts` block allows you to specify timeouts for certain actions:

- `create` - (Defaults to 40 mins) Used when creating the AMI
- `update` - (Defaults to 40 mins) Used when updating the AMI
- `delete` - (Defaults to 90 mins) Used when deregistering the AMI

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the created AMI.
- `root_snapshot_id` - The Snapshot ID for the root volume (for EBS-backed AMIs)

» Import

`aws_ami` can be imported using the ID of the AMI, e.g.

```
$ terraform import aws_ami.example ami-12345678
```

» Resource: aws_ami_copy

The ”AMI copy” resource allows duplication of an Amazon Machine Image (AMI), including cross-region copies.

If the source AMI has associated EBS snapshots, those will also be duplicated along with the AMI.

This is useful for taking a single AMI provisioned in one region and making it available in another for a multi-region deployment.

Copying an AMI can take several minutes. The creation of this resource will block until the new AMI is available for use on new instances.

» Example Usage

```
resource "aws_ami_copy" "example" {
  name              = "terraform-example"
  description       = "A copy of ami-xxxxxxxx"
  source_ami_id     = "ami-xxxxxxxx"
  source_ami_region = "us-west-1"

  tags = {
    Name = "HelloWorld"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A region-unique name for the AMI.
- `source_ami_id` - (Required) The id of the AMI to copy. This id must be valid in the region given by `source_ami_region`.
- `source_ami_region` - (Required) The region from which the AMI will be copied. This may be the same as the AWS provider region in order to create a copy within the same region.
- `encrypted` - (Optional) Specifies whether the destination snapshots of the copied image should be encrypted. Defaults to `false`
- `kms_key_id` - (Optional) The full ARN of the KMS Key to use when encrypting the snapshots of an image during a copy operation. If not specified, then the default AWS KMS Key will be used
- `tags` - (Optional) A mapping of tags to assign to the resource.

This resource also exposes the full set of arguments from the `aws_ami` resource.

» Timeouts

The `timeouts` block allows you to specify timeouts for certain actions:

- `create` - (Defaults to 40 mins) Used when creating the AMI
- `update` - (Defaults to 40 mins) Used when updating the AMI
- `delete` - (Defaults to 90 mins) Used when deregistering the AMI

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the created AMI.

This resource also exports a full set of attributes corresponding to the arguments of the `aws_ami` resource, allowing the properties of the created AMI to be used elsewhere in the configuration.

» Resource: aws_ami_from_instance

The ”AMI from instance” resource allows the creation of an Amazon Machine Image (AMI) modelled after an existing EBS-backed EC2 instance.

The created AMI will refer to implicitly-created snapshots of the instance’s EBS volumes and mimick its assigned block device configuration at the time the resource is created.

This resource is best applied to an instance that is stopped when this instance is created, so that the contents of the created image are predictable. When applied to an instance that is running, *the instance will be stopped before taking the snapshots and then started back up again*, resulting in a period of downtime.

Note that the source instance is inspected only at the initial creation of this resource. Ongoing updates to the referenced instance will not be propagated into the generated AMI. Users may taint or otherwise recreate the resource in order to produce a fresh snapshot.

» Example Usage

```
resource "aws_ami_from_instance" "example" {
  name               = "terraform-example"
  source_instance_id = "i-xxxxxxxx"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A region-unique name for the AMI.
- `source_instance_id` - (Required) The id of the instance to use as the basis of the AMI.
- `snapshot_without_reboot` - (Optional) Boolean that overrides the behavior of stopping the instance before snapshotting. This is risky since it may cause a snapshot of an inconsistent filesystem state, but can be used to avoid downtime if the user otherwise guarantees that no filesystem writes will be underway at the time of snapshot.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Timeouts

The `timeouts` block allows you to specify timeouts for certain actions:

- `create` - (Defaults to 40 mins) Used when creating the AMI

- `update` - (Defaults to 40 mins) Used when updating the AMI
- `delete` - (Defaults to 90 mins) Used when deregistering the AMI

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the created AMI.

This resource also exports a full set of attributes corresponding to the arguments of the `aws_ami` resource, allowing the properties of the created AMI to be used elsewhere in the configuration.

» Resource: aws_ami_launch_permission

Adds launch permission to Amazon Machine Image (AMI) from another AWS account.

» Example Usage

```
resource "aws_ami_launch_permission" "example" {
  image_id   = "ami-12345678"
  account_id = "123456789012"
}
```

» Argument Reference

The following arguments are supported:

- `image_id` - (required) A region-unique name for the AMI.
- `account_id` - (required) An AWS Account ID to add launch permissions.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - A combination of "`image_id-account_id`".

» Import

AWS AMI Launch Permission can be imported using the `ACCOUNT-ID/IMAGE-ID`, e.g.

```
$ terraform import aws_ami_launch_permission.example 123456789012/ami-12345678
```

» Resource: aws_ebs_default_kms_key

Provides a resource to manage the default customer master key (CMK) that your AWS account uses to encrypt EBS volumes.

Your AWS account has an AWS-managed default CMK that is used for encrypting an EBS volume when no CMK is specified in the API call that creates the volume. By using the `aws_ebs_default_kms_key` resource, you can specify a customer-managed CMK to use in place of the AWS-managed default CMK.

NOTE: Creating an `aws_ebs_default_kms_key` resource does not enable default EBS encryption. Use the `aws_ebs_encryption_by_default` to enable default EBS encryption.

NOTE: Destroying this resource will reset the default CMK to the account's AWS-managed default CMK for EBS.

» Example Usage

```
resource "aws_ebs_default_kms_key" "example" {
  key_arn = "${aws_kms_key.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `key_arn` - (Required, ForceNew) The ARN of the AWS Key Management Service (AWS KMS) customer master key (CMK) to use to encrypt the EBS volume.

» Import

The EBS default KMS CMK can be imported with the KMS key ARN, e.g.

```
$ terraform import aws_ebs_default_kms_key.example arn:aws:kms:us-east-1:123456789012:key/ab
```

» Resource: aws_ebs_encryption_by_default

Provides a resource to manage whether default EBS encryption is enabled for your AWS account in the current AWS region. To manage the default KMS key for the region, see the `aws_ebs_default_kms_key` resource.

NOTE: Removing this Terraform resource disables default EBS encryption.

» Example Usage

```
resource "aws_ebs_encryption_by_default" "example" {
  enabled = true
}
```

» Argument Reference

The following arguments are supported:

- `enabled` - (Optional) Whether or not default EBS encryption is enabled. Valid values are `true` or `false`. Defaults to `true`.

» Resource: aws_ebs_snapshot

Creates a Snapshot of an EBS Volume.

» Example Usage

```
resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"
  size              = 40

  tags = {
    Name = "HelloWorld"
  }
}

resource "aws_ebs_snapshot" "example_snapshot" {
  volume_id = "${aws_ebs_volume.example.id}"

  tags = {
    Name = "HelloWorld_snap"
  }
}
```

» Argument Reference

The following arguments are supported:

- `volume_id` - (Required) The Volume ID of which to make a snapshot.
- `description` - (Optional) A description of what the snapshot is.
- `tags` - (Optional) A mapping of tags to assign to the snapshot

» Timeouts

`aws_ebs_snapshot` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating the ebs snapshot
- `delete` - (Default `10 minutes`) Used for deleting the ebs snapshot

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The snapshot ID (e.g. snap-59fcb34e).
- `owner_id` - The AWS account ID of the EBS snapshot owner.
- `owner_alias` - Value from an Amazon-maintained list (`amazon`, `aws-marketplace`, `microsoft`) of snapshot owners.
- `encrypted` - Whether the snapshot is encrypted.
- `volume_size` - The size of the drive in GiBs.
- `kms_key_id` - The ARN for the KMS encryption key.
- `data_encryption_key_id` - The data encryption key identifier for the snapshot.
- `tags` - A mapping of tags for the snapshot.

» Resource: aws_ebs_snapshot_copy

Creates a Snapshot of a snapshot.

» Example Usage

```
resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"
  size              = 40
  tags = {
    Name = "HelloWorld"
  }
}

resource "aws_ebs_snapshot" "example_snapshot" {
  volume_id = "${aws_ebs_volume.example.id}"

  tags = {
    Name = "HelloWorld_snap"
  }
}

resource "aws_ebs_snapshot_copy" "example_copy" {
  source_snapshot_id = "${aws_ebs_snapshot.example_snapshot.id}"
  source_region      = "us-west-2"

  tags = {
    Name = "HelloWorld_copy_snap"
  }
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Optional) A description of what the snapshot is.
- `encrypted` - Whether the snapshot is encrypted.
- `kms_key_id` - The ARN for the KMS encryption key.
- `source_snapshot_id` The ARN for the snapshot to be copied.
- `source_region` The region of the source snapshot.
- `tags` - A mapping of tags for the snapshot.

» Attributes Reference

The following attributes are exported:

- `id` - The snapshot ID (e.g. snap-59fcb34e).
- `owner_id` - The AWS account ID of the snapshot owner.
- `owner_alias` - Value from an Amazon-maintained list (`amazon`, `aws-marketplace`, `microsoft`) of snapshot owners.
- `encrypted` - Whether the snapshot is encrypted.
- `volume_size` - The size of the drive in GiBs.
- `kms_key_id` - The ARN for the KMS encryption key.

- data_encryption_key_id - The data encryption key identifier for the snapshot.
- source_snapshot_id The ARN of the copied snapshot.
- source_region The region of the source snapshot.
- tags - A mapping of tags for the snapshot.

» Resource: aws_ebs_volume

Manages a single EBS volume.

» Example Usage

```
resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"
  size              = 40

  tags = {
    Name = "HelloWorld"
  }
}
```

NOTE: One of size or snapshot_id is required when specifying an EBS volume

» Argument Reference

The following arguments are supported:

- availability_zone - (Required) The AZ where the EBS volume will exist.
- encrypted - (Optional) If true, the disk will be encrypted.
- iops - (Optional) The amount of IOPS to provision for the disk.
- size - (Optional) The size of the drive in GiBs.
- snapshot_id (Optional) A snapshot to base the EBS volume off of.
- type - (Optional) The type of EBS volume. Can be ”standard”, ”gp2”, ”io1”, ”sc1” or ”st1” (Default: ”standard”).
- kms_key_id - (Optional) The ARN for the KMS encryption key. When specifying kms_key_id, encrypted needs to be set to true.
- tags - (Optional) A mapping of tags to assign to the resource.

NOTE: When changing the size, iops or type of an instance, there are considerations to be aware of that Amazon have written about this.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The volume ID (e.g. vol-59fcb34e).
- `arn` - The volume ARN (e.g. arn:aws:ec2:us-east-1:0123456789012:volume/vol-59fcb34e).

» Import

EBS Volumes can be imported using the `id`, e.g.

```
$ terraform import aws_ebs_volume.id vol-049df61146c4d7901
```

» Resource: aws_ec2_capacity_reservation

Provides an EC2 Capacity Reservation. This allows you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration.

» Example Usage

```
resource "aws_ec2_capacity_reservation" "default" {
  instance_type     = "t2.micro"
  instance_platform = "Linux/UNIX"
  availability_zone = "eu-west-1a"
  instance_count    = 1
}
```

» Argument Reference

The following arguments are supported:

- `availability_zone` - (Required) The Availability Zone in which to create the Capacity Reservation.
- `ebs_optimized` - (Optional) Indicates whether the Capacity Reservation supports EBS-optimized instances.
- `end_date` - (Optional) The date and time at which the Capacity Reservation expires. When a Capacity Reservation expires, the reserved capacity is released and you can no longer launch instances into it. Valid values: RFC3339 time string (`YYYY-MM-DDTHH:MM:SSZ`)
- `end_date_type` - (Optional) Indicates the way in which the Capacity Reservation ends. Specify either `unlimited` or `limited`.

- `ephemeral_storage` - (Optional) Indicates whether the Capacity Reservation supports instances with temporary, block-level storage.
- `instance_count` - (Required) The number of instances for which to reserve capacity.
- `instance_match_criteria` - (Optional) Indicates the type of instance launches that the Capacity Reservation accepts. Specify either `open` or `targeted`.
- `instance_platform` - (Required) The type of operating system for which to reserve capacity. Valid options are `Linux/UNIX`, `Red Hat Enterprise Linux`, `SUSE Linux`, `Windows`, `Windows with SQL Server`, `Windows with SQL Server Enterprise`, `Windows with SQL Server Standard` or `Windows with SQL Server Web`.
- `instance_type` - (Required) The instance type for which to reserve capacity.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `tenancy` - (Optional) Indicates the tenancy of the Capacity Reservation. Specify either `default` or `dedicated`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Capacity Reservation ID.

» Import

Capacity Reservations can be imported using the `id`, e.g.

```
$ terraform import aws_ec2_capacity_reservation.web cr-0123456789abcdef0
```

» Resource: aws_ec2_client_vpn_endpoint

Provides an AWS Client VPN endpoint for OpenVPN clients. For more information on usage, please see the AWS Client VPN Administrator's Guide.

» Example Usage

```
resource "aws_ec2_client_vpn_endpoint" "example" {
  description            = "terraform-clientvpn-example"
  server_certificate_arn = "${aws_acm_certificate.cert.arn}"
  client_cidr_block      = "10.0.0.0/16"
  authentication_options {
    type                       = "certificate-authentication"
    root_certificate_chain_arn = "${aws_acm_certificate.root_cert.arn}"
  }

  connection_log_options {
    enabled               = true
    cloudwatch_log_group  = "${aws_cloudwatch_log_group.lg.name}"
    cloudwatch_log_stream = "${aws_cloudwatch_log_stream.ls.name}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `authentication_options` - (Required) Information about the authentication method to be used to authenticate clients.
- `client_cidr_block` - (Required) The IPv4 address range, in CIDR notation, from which to assign client IP addresses. The address range cannot overlap with the local CIDR of the VPC in which the associated subnet is located, or the routes that you add manually. The address range cannot be changed after the Client VPN endpoint has been created. The CIDR block should be /22 or greater.
- `connection_log_options` - (Required) Information about the client connection logging options.
- `description` - (Optional) Name of the repository.
- `dns_servers` - (Optional) Information about the DNS servers to be used for DNS resolution. A Client VPN endpoint can have up to two DNS servers. If no DNS server is specified, the DNS address of the VPC that is to be associated with Client VPN endpoint is used as the DNS server.
- `server_certificate_arn` - (Required) The ARN of the ACM server certificate.
- `split_tunnel` - (Optional) Indicates whether split-tunnel is enabled on VPN endpoint. Default value is `false`.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `transport_protocol` - (Optional) The transport protocol to be used by the VPN session. Default value is `udp`.

» `authentication_options` Argument Reference

One of the following arguments must be supplied:

- `type` - (Required) The type of client authentication to be used. Specify `certificate-authentication` to use certificate-based authentication, or directory-service-authentication to use Active Directory authentication.

- active_directory_id - (Optional) The ID of the Active Directory to be used for authentication if type is directory-service-authentication.
- root_certificate_chain_arn - (Optional) The ARN of the client certificate. The certificate must be signed by a certificate authority (CA) and it must be provisioned in AWS Certificate Manager (ACM). Only necessary when type is set to certificate-authentication.

» connection_log_options Argument Reference

One of the following arguments must be supplied:

- enabled - (Required) Indicates whether connection logging is enabled.
- cloudwatch_log_group - (Optional) The name of the CloudWatch Logs log group.
- cloudwatch_log_stream - (Optional) The name of the CloudWatch Logs log stream to which the connection data is published.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The ID of the Client VPN endpoint.
- dns_name - The DNS name to be used by clients when establishing their VPN session.
- status - The current state of the Client VPN endpoint.

» Import

AWS Client VPN endpoints can be imported using the id value found via aws ec2 describe-client-vpn-endpoints, e.g.

```
$ terraform import aws_ec2_client_vpn_endpoint.example cvpn-endpoint-0ac3a1abbccddd666
```

» Resource: aws_ec2_client_vpn_network_association

Provides network associations for AWS Client VPN endpoints. For more information on usage, please see the AWS Client VPN Administrator's Guide.

» Example Usage

```
resource "aws_ec2_client_vpn_network_association" "example" {
  client_vpn_endpoint_id = "${aws_ec2_client_vpn_endpoint.example.id}"
  subnet_id              = "${aws_subnet.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `client_vpn_endpoint_id` - (Required) The ID of the Client VPN endpoint.
- `subnet_id` - (Required) The ID of the subnet to associate with the Client VPN endpoint.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The unique ID of the target network association.
- `security_groups` - The IDs of the security groups applied to the target network association.
- `status` - The current state of the target network association.
- `vpc_id` - The ID of the VPC in which the target network (subnet) is located.

» Resource: aws_ec2_fleet

Provides a resource to manage EC2 Fleets.

» Example Usage

```
resource "aws_ec2_fleet" "example" {
  launch_template_config {
    launch_template_specification {
      launch_template_id = "${aws_launch_template.example.id}"
      version            = "${aws_launch_template.example.latest_version}"
    }
  }

  target_capacity_specification {
    default_target_capacity_type = "spot"
    total_target_capacity        = 5
  }
}
```

» Argument Reference

The following arguments are supported:

- `launch_template_config` - (Required) Nested argument containing EC2 Launch Template configurations. Defined below.
- `target_capacity_specification` - (Required) Nested argument containing target capacity configurations. Defined below.
- `excess_capacity_termination_policy` - (Optional) Whether running instances should be terminated if the total target capacity of the EC2 Fleet is decreased below the current size of the EC2. Valid values: `no-termination`, `termination`. Defaults to `termination`.
- `on_demand_options` - (Optional) Nested argument containing On-Demand configurations. Defined below.
- `replace_unhealthy_instances` - (Optional) Whether EC2 Fleet should replace unhealthy instances. Defaults to `false`.
- `spot_options` - (Optional) Nested argument containing Spot configurations. Defined below.
- `tags` - (Optional) Map of Fleet tags. To tag instances at launch, specify the tags in the Launch Template.
- `terminate_instances` - (Optional) Whether to terminate instances for an EC2 Fleet if it is deleted successfully. Defaults to `false`.
- `terminate_instances_with_expiration` - (Optional) Whether running instances should be terminated when the EC2 Fleet expires. Defaults to `false`.
- `type` - (Optional) The type of request. Indicates whether the EC2 Fleet only requests the target capacity, or also attempts to maintain it. Valid values: `maintain`, `request`. Defaults to `maintain`.

» launch_template_config

- `launch_template_specification` - (Required) Nested argument containing EC2 Launch Template to use. Defined below.
- `override` - (Optional) Nested argument(s) containing parameters to override the same parameters in the Launch Template. Defined below.

» launch_template_specification

NOTE: Either `launch_template_id` or `launch_template_name` must be specified.

- `version` - (Required) Version number of the launch template.
- `launch_template_id` - (Optional) ID of the launch template.
- `launch_template_name` - (Optional) Name of the launch template.

» override

Example:

```
resource "aws_ec2_fleet" "example" {
  # ... other configuration ...

  launch_template_config {
    # ... other configuration ...

    override {
      instance_type     = "m4.xlarge"
      weighted_capacity = 1
    }

    override {
      instance_type     = "m4.2xlarge"
      weighted_capacity = 2
    }
  }
}
```

- `availability_zone` - (Optional) Availability Zone in which to launch the instances.
- `instance_type` - (Optional) Instance type.
- `max_price` - (Optional) Maximum price per unit hour that you are willing to pay for a Spot Instance.
- `priority` - (Optional) Priority for the launch template override. If `on_demand_options allocation_strategy` is set to `prioritized`, EC2 Fleet uses priority to determine which launch template override to use first in fulfilling On-Demand capacity. The highest priority is launched first. The lower the number, the higher the priority. If no number is set, the launch template override has the lowest priority. Valid values are whole numbers starting at 0.
- `subnet_id` - (Optional) ID of the subnet in which to launch the instances.
- `weighted_capacity` - (Optional) Number of units provided by the specified instance type.

» on_demand_options

- `allocation_strategy` - (Optional) The order of the launch template overrides to use in fulfilling On-Demand capacity. Valid values: `lowestPrice`, `prioritized`. Default: `lowestPrice`.

» spot_options

- `allocation_strategy` - (Optional) How to allocate the target capacity across the Spot pools. Valid values: `diversified`, `lowestPrice`. Default: `lowestPrice`.
- `instance_interruption_behavior` - (Optional) Behavior when a Spot Instance is interrupted. Valid values: `hibernate`, `stop`, `terminate`. Default: `terminate`.
- `instance_pools_to_use_count` - (Optional) Number of Spot pools across which to allocate your target Spot capacity. Valid only when Spot `allocation_strategy` is set to `lowestPrice`. Default: `1`.

» target_capacity_specification

- `default_target_capacity_type` - (Required) Default target capacity type. Valid values: `on-demand`, `spot`.
- `total_target_capacity` - (Required) The number of units to request, filled using `default_target_capacity_type`.
- `on_demand_target_capacity` - (Optional) The number of On-Demand units to request.
- `spot_target_capacity` - (Optional) The number of Spot units to request.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Fleet identifier

» Timeouts

`aws_ec2_fleet` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for a fleet to be active.
- `update` - (Default `10m`) How long to wait for a fleet to be modified.
- `delete` - (Default `10m`) How long to wait for a fleet to be deleted. If `terminate_instances` is `true`, how long to wait for instances to terminate.

» Import

`aws_ec2_fleet` can be imported by using the Fleet identifier, e.g.

```
$ terraform import aws_ec2_fleet.example fleet-b9b55d27-c5fc-41ac-a6f3-48fcc91f080c
```

» Resource: aws_ec2_transit_gateway

Manages an EC2 Transit Gateway.

» Example Usage

```
resource "aws_ec2_transit_gateway" "example" {
  description = "example"
}
```

» Argument Reference

The following arguments are supported:

- `amazon_side_asn` - (Optional) Private Autonomous System Number (ASN) for the Amazon side of a BGP session. The range is `64512` to `65534` for 16-bit ASNs and `4200000000` to `4294967294` for 32-bit ASNs. Default value: `64512`.
- `auto_accept_shared_attachments` - (Optional) Whether resource attachment requests are automatically accepted. Valid values: `disable`, `enable`. Default value: `disable`.
- `default_route_table_association` - (Optional) Whether resource attachments are automatically associated with the default association route table. Valid values: `disable`, `enable`. Default value: `enable`.
- `default_route_table_propagation` - (Optional) Whether resource attachments automatically propagate routes to the default propagation route table. Valid values: `disable`, `enable`. Default value: `enable`.
- `description` - (Optional) Description of the EC2 Transit Gateway.
- `dns_support` - (Optional) Whether DNS support is enabled. Valid values: `disable`, `enable`. Default value: `enable`.
- `tags` - (Optional) Key-value tags for the EC2 Transit Gateway.
- `vpn_ecmp_support` - (Optional) Whether VPN Equal Cost Multipath Protocol support is enabled. Valid values: `disable`, `enable`. Default value: `enable`.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - EC2 Transit Gateway Amazon Resource Name (ARN)
- `association_default_route_table_id` - Identifier of the default association route table
- `id` - EC2 Transit Gateway identifier
- `owner_id` - Identifier of the AWS account that owns the EC2 Transit Gateway
- `propagation_default_route_table_id` - Identifier of the default propagation route table

» Import

`aws_ec2_transit_gateway` can be imported by using the EC2 Transit Gateway identifier, e.g.

```
$ terraform import aws_ec2_transit_gateway.example tgw-12345678
```

» Resource: aws_ec2_transit_gateway_route

Manages an EC2 Transit Gateway Route.

» Example Usage

» Standard usage

```
resource "aws_ec2_transit_gateway_route" "example" {
  destination_cidr_block         = "0.0.0.0/0"
  transit_gateway_attachment_id  = "${aws_ec2_transit_gateway_vpc_attachment.example.id}"
  transit_gateway_route_table_id = "${aws_ec2_transit_gateway.example.association_default_ro
}
```

» Blackhole route

```
resource "aws_ec2_transit_gateway_route" "example" {
  destination_cidr_block         = "0.0.0.0/0"
  blackhole                      = true
  transit_gateway_route_table_id = "${aws_ec2_transit_gateway.example.association_default_ro
}
```

» Argument Reference

The following arguments are supported:

- `destination_cidr_block` - (Required) IPv4 CIDR range used for destination matches. Routing decisions are based on the most specific match.
- `transit_gateway_attachment_id` - (Optional) Identifier of EC2 Transit Gateway Attachment (required if `blackhole` is set to false).
- `blackhole` - (Optional) Indicates whether to drop traffic that matches this route (default to `false`).
- `transit_gateway_route_table_id` - (Required) Identifier of EC2 Transit Gateway Route Table.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - EC2 Transit Gateway Route Table identifier combined with destination

» Import

`aws_ec2_transit_gateway_route` can be imported by using the EC2 Transit Gateway Route Table, an underscore, and the destination, e.g.

```
$ terraform import aws_ec2_transit_gateway_route.example tgw-rtb-12345678_0.0.0.0/0
```

» Resource: aws_ec2_transit_gateway_route_table

Manages an EC2 Transit Gateway Route Table.

» Example Usage

```
resource "aws_ec2_transit_gateway_route_table" "example" {
  transit_gateway_id = "${aws_ec2_transit_gateway.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `transit_gateway_id` - (Required) Identifier of EC2 Transit Gateway.

- `tags` - (Optional) Key-value tags for the EC2 Transit Gateway Route Table.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `default_association_route_table` - Boolean whether this is the default association route table for the EC2 Transit Gateway.
- `default_propagation_route_table` - Boolean whether this is the default propagation route table for the EC2 Transit Gateway.
- `id` - EC2 Transit Gateway Route Table identifier

» Import

`aws_ec2_transit_gateway_route_table` can be imported by using the EC2 Transit Gateway Route Table identifier, e.g.

```
$ terraform import aws_ec2_transit_gateway_route_table.example tgw-rtb-12345678
```

» Resource: aws_ec2_transit_gateway_route_table_association

Manages an EC2 Transit Gateway Route Table association.

» Example Usage

```
resource "aws_ec2_transit_gateway_route_table_association" "example" {
  transit_gateway_attachment_id  = "${aws_ec2_transit_gateway_vpc_attachment.example.id}"
  transit_gateway_route_table_id = "${aws_ec2_transit_gateway_route_table.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `transit_gateway_attachment_id` - (Required) Identifier of EC2 Transit Gateway Attachment.
- `transit_gateway_route_table_id` - (Required) Identifier of EC2 Transit Gateway Route Table.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - EC2 Transit Gateway Route Table identifier combined with EC2 Transit Gateway Attachment identifier
- `resource_id` - Identifier of the resource
- `resource_type` - Type of the resource

» Import

`aws_ec2_transit_gateway_route_table_association` can be imported by using the EC2 Transit Gateway Route Table identifier, an underscore, and the EC2 Transit Gateway Attachment identifier, e.g.

```
$ terraform import aws_ec2_transit_gateway_route_table_association.example tgw-rtb-12345678_
```

» Resource: aws_ec2_transit_gateway_route_table_propagation

Manages an EC2 Transit Gateway Route Table propagation.

» Example Usage

```
resource "aws_ec2_transit_gateway_route_table_propagation" "example" {
  transit_gateway_attachment_id  = "${aws_ec2_transit_gateway_vpc_attachment.example.id}"
  transit_gateway_route_table_id = "${aws_ec2_transit_gateway_route_table.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `transit_gateway_attachment_id` - (Required) Identifier of EC2 Transit Gateway Attachment.
- `transit_gateway_route_table_id` - (Required) Identifier of EC2 Transit Gateway Route Table.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - EC2 Transit Gateway Route Table identifier combined with EC2 Transit Gateway Attachment identifier

- `resource_id` - Identifier of the resource
- `resource_type` - Type of the resource

» Import

`aws_ec2_transit_gateway_route_table_propagation` can be imported by using the EC2 Transit Gateway Route Table identifier, an underscore, and the EC2 Transit Gateway Attachment identifier, e.g.

```
$ terraform import aws_ec2_transit_gateway_route_table_propagation.example tgw-rtb-12345678_
```

» Resource: aws_ec2_transit_gateway_vpc_attachment

Manages an EC2 Transit Gateway VPC Attachment. For examples of custom route table association and propagation, see the EC2 Transit Gateway Networking Examples Guide.

» Example Usage

```
resource "aws_ec2_transit_gateway_vpc_attachment" "example" {
  subnet_ids         = ["${aws_subnet.example.id}"]
  transit_gateway_id = "${aws_ec2_transit_gateway.example.id}"
  vpc_id             = "${aws_vpc.example.id}"
}
```

A full example of how to create a Transit Gateway in one AWS account, share it with a second AWS account, and attach a VPC in the second account to the Transit Gateway via the `aws_ec2_transit_gateway_vpc_attachment` and `aws_ec2_transit_gateway_vpc_attachment_accepter` resources can be found in the `./examples/transit-gateway-cross-account-vpc-attachment` directory within the Github Repository.

» Argument Reference

The following arguments are supported:

- `subnet_ids` - (Required) Identifiers of EC2 Subnets.
- `transit_gateway_id` - (Required) Identifier of EC2 Transit Gateway.
- `vpc_id` - (Required) Identifier of EC2 VPC.
- `dns_support` - (Optional) Whether DNS support is enabled. Valid values: `disable`, `enable`. Default value: `enable`.
- `ipv6_support` - (Optional) Whether IPv6 support is enabled. Valid values: `disable`, `enable`. Default value: `disable`.

- tags - (Optional) Key-value tags for the EC2 Transit Gateway VPC Attachment.
- transit_gateway_default_route_table_association - (Optional) Boolean whether the VPC Attachment should be associated with the EC2 Transit Gateway association default route table. This cannot be configured or perform drift detection with Resource Access Manager shared EC2 Transit Gateways. Default value: true.
- transit_gateway_default_route_table_propagation - (Optional) Boolean whether the VPC Attachment should propagate routes with the EC2 Transit Gateway propagation default route table. This cannot be configured or perform drift detection with Resource Access Manager shared EC2 Transit Gateways. Default value: true.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- id - EC2 Transit Gateway Attachment identifier
- vpc_owner_id - Identifier of the AWS account that owns the EC2 VPC.

» Import

aws_ec2_transit_gateway_vpc_attachment can be imported by using the EC2 Transit Gateway Attachment identifier, e.g.

```
$ terraform import aws_ec2_transit_gateway_vpc_attachment.example tgw-attach-12345678
```

» Resource: aws_ec2_transit_gateway_vpc_attachment_accepter

Manages the accepter's side of an EC2 Transit Gateway VPC Attachment.

When a cross-account (requester's AWS account differs from the accepter's AWS account) EC2 Transit Gateway VPC Attachment is created, an EC2 Transit Gateway VPC Attachment resource is automatically created in the accepter's account. The requester can use the aws_ec2_transit_gateway_vpc_attachment resource to manage its side of the connection and the accepter can use the aws_ec2_transit_gateway_vpc_attachment_accepter resource to "adopt" its side of the connection into management.

» Example Usage

```
resource "aws_ec2_transit_gateway_vpc_attachment_accepter" "example" {
  transit_gateway_attachment_id = "${aws_ec2_transit_gateway_vpc_attachment.example.id}"
  tags = {
    Name = "Example cross-account attachment"
  }
}
```

A full example of how to how to create a Transit Gateway in one AWS account, share it with a second AWS account, and attach a VPC in the second account to the Transit Gateway via the `aws_ec2_transit_gateway_vpc_attachment` and `aws_ec2_transit_gateway_vpc_attachment_accepter` resources can be found in the `./examples/transit-gateway-cross-account-vpc-attachment` directory within the Github Repository.

» Argument Reference

The following arguments are supported:

- `transit_gateway_attachment_id` - (Required) The ID of the EC2 Transit Gateway Attachment to manage.
- `transit_gateway_default_route_table_association` - (Optional) Boolean whether the VPC Attachment should be associated with the EC2 Transit Gateway association default route table. Default value: `true`.
- `transit_gateway_default_route_table_propagation` - (Optional) Boolean whether the VPC Attachment should propagate routes with the EC2 Transit Gateway propagation default route table. Default value: `true`.
- `tags` - (Optional) Key-value tags for the EC2 Transit Gateway VPC Attachment.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - EC2 Transit Gateway Attachment identifier
- `dns_support` - Whether DNS support is enabled. Valid values: `disable`, `enable`.
- `ipv6_support` - Whether IPv6 support is enabled. Valid values: `disable`, `enable`.
- `subnet_ids` - Identifiers of EC2 Subnets.
- `transit_gateway_id` - Identifier of EC2 Transit Gateway.
- `vpc_id` - Identifier of EC2 VPC.
- `vpc_owner_id` - Identifier of the AWS account that owns the EC2 VPC.

» Import

`aws_ec2_transit_gateway_vpc_attachment_accepter` can be imported by using the EC2 Transit Gateway Attachment identifier, e.g.

```
$ terraform import aws_ec2_transit_gateway_vpc_attachment_accepter.example tgw-attach-123456
```

» Resource: aws_eip

Provides an Elastic IP resource.

Note: EIP may require IGW to exist prior to association. Use `depends_on` to set an explicit dependency on the IGW.

Note: Do not use `network_interface` to associate the EIP to `aws_lb` or `aws_nat_gateway` resources. Instead use the `allocation_id` available in those resources to allow AWS to manage the association, otherwise you will see `AuthFailure` errors.

» Example Usage

Single EIP associated with an instance:

```
resource "aws_eip" "lb" {
  instance = "${aws_instance.web.id}"
  vpc      = true
}
```

Multiple EIPs associated with a single network interface:

```
resource "aws_network_interface" "multi-ip" {
  subnet_id   = "${aws_subnet.main.id}"
  private_ips = ["10.0.0.10", "10.0.0.11"]
}

resource "aws_eip" "one" {
  vpc                       = true
  network_interface         = "${aws_network_interface.multi-ip.id}"
  associate_with_private_ip = "10.0.0.10"
}

resource "aws_eip" "two" {
  vpc                       = true
  network_interface         = "${aws_network_interface.multi-ip.id}"
  associate_with_private_ip = "10.0.0.11"
}
```

Attaching an EIP to an Instance with a pre-assigned private ip (VPC Only):

```
resource "aws_vpc" "default" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_hostnames = true
}

resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.default.id}"
}

resource "aws_subnet" "tf_test_subnet" {
  vpc_id                  = "${aws_vpc.default.id}"
  cidr_block              = "10.0.0.0/24"
  map_public_ip_on_launch = true

  depends_on = ["aws_internet_gateway.gw"]
}

resource "aws_instance" "foo" {
  # us-west-2
  ami           = "ami-5189a661"
  instance_type = "t2.micro"

  private_ip = "10.0.0.12"
  subnet_id  = "${aws_subnet.tf_test_subnet.id}"
}

resource "aws_eip" "bar" {
  vpc = true

  instance                  = "${aws_instance.foo.id}"
  associate_with_private_ip = "10.0.0.12"
  depends_on                = ["aws_internet_gateway.gw"]
}
```

Allocating EIP from the BYOIP pool:

```
resource "aws_eip" "byoip-ip" {
  vpc              = true
  public_ipv4_pool = "ipv4pool-ec2-012345"
}
```

» Argument Reference

The following arguments are supported:

- `vpc` - (Optional) Boolean if the EIP is in a VPC or not.
- `instance` - (Optional) EC2 instance ID.
- `network_interface` - (Optional) Network interface ID to associate with.
- `associate_with_private_ip` - (Optional) A user specified primary or secondary private IP address to associate with the Elastic IP address. If no private IP address is specified, the Elastic IP address is associated with the primary private IP address.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `public_ipv4_pool` - (Optional) EC2 IPv4 address pool identifier or `amazon`. This option is only available for VPC EIPs.

NOTE: You can specify either the `instance` ID or the `network_interface` ID, but not both. Including both will **not** return an error from the AWS API, but will have undefined behavior. See the relevant AssociateAddress API Call for more information.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Contains the EIP allocation ID.
- `private_ip` - Contains the private IP address (if in VPC).
- `private_dns` - The Private DNS associated with the Elastic IP address (if in VPC).
- `associate_with_private_ip` - Contains the user specified private IP address (if in VPC).
- `public_ip` - Contains the public IP address.
- `public_dns` - Public DNS associated with the Elastic IP address.
- `instance` - Contains the ID of the attached instance.
- `network_interface` - Contains the ID of the attached network interface.
- `public_ipv4_pool` - EC2 IPv4 address pool identifier (if in VPC).

Note: The resource computes the `public_dns` and `private_dns` attributes according to the VPC DNS Guide as they are not available with the EC2 API.

» Timeouts

`aws_eip` provides the following Timeouts configuration options:

- `read` - (Default `15 minutes`) How long to wait querying for information about EIPs.
- `update` - (Default `5 minutes`) How long to wait for an EIP to be updated.
- `delete` - (Default `3 minutes`) How long to wait for an EIP to be deleted.

» Import

EIPs in a VPC can be imported using their Allocation ID, e.g.

```
$ terraform import aws_eip.bar eipalloc-00a10e96
```

EIPs in EC2 Classic can be imported using their Public IP, e.g.

```
$ terraform import aws_eip.bar 52.0.0.0
```

» Resource: aws_eip_association

Provides an AWS EIP Association as a top level resource, to associate and disassociate Elastic IPs from AWS Instances and Network Interfaces.

NOTE: Do not use this resource to associate an EIP to `aws_lb` or `aws_nat_gateway` resources. Instead use the `allocation_id` available in those resources to allow AWS to manage the association, otherwise you will see `AuthFailure` errors.

NOTE: `aws_eip_association` is useful in scenarios where EIPs are either preexisting or distributed to customers or users and therefore cannot be changed.

» Example Usage

```
resource "aws_eip_association" "eip_assoc" {
  instance_id   = "${aws_instance.web.id}"
  allocation_id = "${aws_eip.example.id}"
}

resource "aws_instance" "web" {
  ami               = "ami-21f78e11"
  availability_zone = "us-west-2a"
  instance_type     = "t1.micro"

  tags = {
    Name = "HelloWorld"
  }
}

resource "aws_eip" "example" {
  vpc = true
}
```

» Argument Reference

The following arguments are supported:

- `allocation_id` - (Optional) The allocation ID. This is required for EC2-VPC.
- `allow_reassociation` - (Optional, Boolean) Whether to allow an Elastic IP to be re-associated. Defaults to `true` in VPC.
- `instance_id` - (Optional) The ID of the instance. This is required for EC2-Classic. For EC2-VPC, you can specify either the instance ID or the network interface ID, but not both. The operation fails if you specify an instance ID unless exactly one network interface is attached.
- `network_interface_id` - (Optional) The ID of the network interface. If the instance has more than one network interface, you must specify a network interface ID.
- `private_ip_address` - (Optional) The primary or secondary private IP address to associate with the Elastic IP address. If no private IP address is specified, the Elastic IP address is associated with the primary private IP address.
- `public_ip` - (Optional) The Elastic IP address. This is required for EC2-Classic.

» Attributes Reference

- `association_id` - The ID that represents the association of the Elastic IP address with an instance.
- `allocation_id` - As above
- `instance_id` - As above
- `network_interface_id` - As above
- `private_ip_address` - As above
- `public_ip` - As above

» Import

EIP Assocations can be imported using their association ID.

```
$ terraform import aws_eip_association.test eipassoc-ab12c345
```

» Resource: aws_instance

Provides an EC2 instance resource. This allows instances to be created, updated, and deleted. Instances also support provisioning.

» Example Usage

```
# Create a new instance of the latest Ubuntu 14.04 on an
# t2.micro node with an AWS Tag naming it "HelloWorld"
provider "aws" {
  region = "us-west-2"
}

data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_instance" "web" {
  ami           = "${data.aws_ami.ubuntu.id}"
  instance_type = "t2.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

» Argument Reference

The following arguments are supported:

- `ami` - (Required) The AMI to use for the instance.
- `availability_zone` - (Optional) The AZ to start the instance in.
- `placement_group` - (Optional) The Placement Group to start the instance in.
- `tenancy` - (Optional) The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of dedicated runs on single-tenant hardware. The host tenancy is not supported for the import-instance command.

- `host_id` - (optional) The Id of a dedicated host that the instance will be assigned to. Use when an instance is to be launched on a specific dedicated host.
- `cpu_core_count` - (Optional) Sets the number of CPU cores for an instance. This option is only supported on creation of instance type that support CPU Options CPU Cores and Threads Per CPU Core Per Instance Type - specifying this option for unsupported instance types will return an error from the EC2 API.
- `cpu_threads_per_core` - (Optional - has no effect unless `cpu_core_count` is also set) If set to to 1, hyperthreading is disabled on the launched instance. Defaults to 2 if not set. See Optimizing CPU Options for more information.

NOTE: Changing `cpu_core_count` and/or `cpu_threads_per_core` will cause the resource to be destroyed and re-created.

- `ebs_optimized` - (Optional) If true, the launched EC2 instance will be EBS-optimized. Note that if this is not set on an instance type that is optimized by default then this will show as disabled but if the instance type is optimized by default then there is no need to set this and there is no effect to disabling it. See the EBS Optimized section of the AWS User Guide for more information.

- `disable_api_termination` - (Optional) If true, enables EC2 Instance Termination Protection

- `instance_initiated_shutdown_behavior` - (Optional) Shutdown behavior for the instance. Amazon defaults this to `stop` for EBS-backed instances and `terminate` for instance-store instances. Cannot be set on instance-store instances. See Shutdown Behavior for more information.

- `instance_type` - (Required) The type of instance to start. Updates to this field will trigger a stop/start of the EC2 instance.

- `key_name` - (Optional) The key name of the Key Pair to use for the instance; which can be managed using the `aws_key_pair` resource.

- `get_password_data` - (Optional) If true, wait for password data to become available and retrieve it. Useful for getting the administrator password for instances running Microsoft Windows. The password data is exported to the `password_data` attribute. See GetPasswordData for more information.

- `monitoring` - (Optional) If true, the launched EC2 instance will have detailed monitoring enabled. (Available since v0.6.0)

- `security_groups` - (Optional, EC2-Classic and default VPC only) A list of security group names (EC2-Classic) or IDs (default VPC) to associate with.

NOTE: If you are creating Instances in a VPC, use `vpc_security_group_ids` instead.

- `vpc_security_group_ids` - (Optional, VPC only) A list of security group IDs to associate with.
- `subnet_id` - (Optional) The VPC Subnet ID to launch in.
- `associate_public_ip_address` - (Optional) Associate a public ip address with an instance in a VPC. Boolean value.
- `private_ip` - (Optional) Private IP address to associate with the instance in a VPC.
- `source_dest_check` - (Optional) Controls if traffic is routed to the instance when the destination address does not match the instance. Used for NAT or VPNs. Defaults true.
- `user_data` - (Optional) The user data to provide when launching the instance. Do not pass gzip-compressed data via this argument; see `user_data_base64` instead.
- `user_data_base64` - (Optional) Can be used instead of `user_data` to pass base64-encoded binary data directly. Use this instead of `user_data` whenever the value is not a valid UTF-8 string. For example, gzip-encoded user data must be base64-encoded and passed via this argument to avoid corruption.
- `iam_instance_profile` - (Optional) The IAM Instance Profile to launch the instance with. Specified as the name of the Instance Profile. Ensure your credentials have the correct permission to assign the instance profile according to the EC2 documentation, notably `iam:PassRole`.
- `ipv6_address_count`- (Optional) A number of IPv6 addresses to associate with the primary network interface. Amazon EC2 chooses the IPv6 addresses from the range of your subnet.
- `ipv6_addresses` - (Optional) Specify one or more IPv6 addresses from the range of the subnet to associate with the primary network interface
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `volume_tags` - (Optional) A mapping of tags to assign to the devices created by the instance at launch time.
- `root_block_device` - (Optional) Customize details about the root block device of the instance. See Block Devices below for details.
- `ebs_block_device` - (Optional) Additional EBS block devices to attach to the instance. Block device configurations only apply on resource creation. See Block Devices below for details on attributes and drift detection.
- `ephemeral_block_device` - (Optional) Customize Ephemeral (also known as "Instance Store") volumes on the instance. See Block Devices below for details.
- `network_interface` - (Optional) Customize network interfaces to be attached at instance boot time. See Network Interfaces below for more details.
- `credit_specification` - (Optional) Customize the credit specification of the instance. See Credit Specification below for more details.

» Timeouts

The `timeouts` block allows you to specify timeouts for certain actions:

- `create` - (Defaults to 10 mins) Used when launching the instance (until it reaches the initial `running` state)
- `update` - (Defaults to 10 mins) Used when stopping and starting the instance when necessary during update - e.g. when changing instance type
- `delete` - (Defaults to 20 mins) Used when terminating the instance

» Block devices

Each of the `*_block_device` attributes control a portion of the AWS Instance's "Block Device Mapping". It's a good idea to familiarize yourself with AWS's Block Device Mapping docs to understand the implications of using these attributes.

The `root_block_device` mapping supports the following:

- `volume_type` - (Optional) The type of volume. Can be `"standard"`, `"gp2"`, `"io1"`, `"sc1"`, or `"st1"`. (Default: `"standard"`).
- `volume_size` - (Optional) The size of the volume in gibibytes (GiB).
- `iops` - (Optional) The amount of provisioned IOPS. This is only valid for `volume_type` of `"io1"`, and must be specified if using that type
- `delete_on_termination` - (Optional) Whether the volume should be destroyed on instance termination (Default: `true`).
- `encrypted` - (Optional) Enable volume encryption. (Default: `false`). Must be configured to perform drift detection.
- `kms_key_id` - (Optional) Amazon Resource Name (ARN) of the KMS Key to use when encrypting the volume. Must be configured to perform drift detection.

Modifying any of the `root_block_device` settings requires resource replacement.

Each `ebs_block_device` supports the following:

- `device_name` - The name of the device to mount.
- `snapshot_id` - (Optional) The Snapshot ID to mount.
- `volume_type` - (Optional) The type of volume. Can be `"standard"`, `"gp2"`, or `"io1"`. (Default: `"standard"`).
- `volume_size` - (Optional) The size of the volume in gibibytes (GiB).
- `iops` - (Optional) The amount of provisioned IOPS. This must be set with a `volume_type` of `"io1"`.
- `delete_on_termination` - (Optional) Whether the volume should be destroyed on instance termination (Default: `true`).

- `encrypted` - (Optional) Enables EBS encryption on the volume (Default: `false`). Cannot be used with `snapshot_id`. Must be configured to perform drift detection.
- `kms_key_id` - (Optional) Amazon Resource Name (ARN) of the KMS Key to use when encrypting the volume. Must be configured to perform drift detection.

NOTE: Currently, changes to the `ebs_block_device` configuration of *existing* resources cannot be automatically detected by Terraform. To manage changes and attachments of an EBS block to an instance, use the `aws_ebs_volume` and `aws_volume_attachment` resources instead. If you use `ebs_block_device` on an `aws_instance`, Terraform will assume management over the full set of non-root EBS block devices for the instance, treating additional block devices as drift. For this reason, `ebs_block_device` cannot be mixed with external `aws_ebs_volume` and `aws_volume_attachment` resources for a given instance.

Each `ephemeral_block_device` supports the following:

- `device_name` - The name of the block device to mount on the instance.
- `virtual_name` - (Optional) The Instance Store Device Name (e.g. `"ephemeral0"`).
- `no_device` - (Optional) Suppresses the specified device included in the AMI's block device mapping.

Each AWS Instance type has a different set of Instance Store block devices available for attachment. AWS publishes a list of which ephemeral devices are available on each type. The devices are always identified by the `virtual_name` in the format `"ephemeral{0..N}"`.

» Network Interfaces

Each of the `network_interface` blocks attach a network interface to an EC2 Instance during boot time. However, because the network interface is attached at boot-time, replacing/modifying the network interface **WILL** trigger a recreation of the EC2 Instance. If you should need at any point to detach/modify/re-attach a network interface to the instance, use the `aws_network_interface` or `aws_network_interface_attachment` resources instead.

The `network_interface` configuration block *does*, however, allow users to supply their own network interface to be used as the default network interface on an EC2 Instance, attached at `eth0`.

Each `network_interface` block supports the following:

- `device_index` - (Required) The integer index of the network interface attachment. Limited by instance type.
- `network_interface_id` - (Required) The ID of the network interface to attach.

- `delete_on_termination` - (Optional) Whether or not to delete the network interface on instance termination. Defaults to `false`. Currently, the only valid value is `false`, as this is only supported when creating new network interfaces when launching an instance.

» Credit Specification

NOTE: Removing this configuration on existing instances will only stop managing it. It will not change the configuration back to the default for the instance type.

Credit specification can be applied/modified to the EC2 Instance at any time.

The `credit_specification` block supports the following:

- `cpu_credits` - (Optional) The credit option for CPU usage. Can be `"standard"` or `"unlimited"`. T3 instances are launched as unlimited by default. T2 instances are launched as standard by default.

» Example

```
resource "aws_vpc" "my_vpc" {
  cidr_block = "172.16.0.0/16"

  tags = {
    Name = "tf-example"
  }
}

resource "aws_subnet" "my_subnet" {
  vpc_id            = "${aws_vpc.my_vpc.id}"
  cidr_block        = "172.16.10.0/24"
  availability_zone = "us-west-2a"

  tags = {
    Name = "tf-example"
  }
}

resource "aws_network_interface" "foo" {
  subnet_id   = "${aws_subnet.my_subnet.id}"
  private_ips = ["172.16.10.100"]

  tags = {
    Name = "primary_network_interface"
  }
}

resource "aws_instance" "foo" {
  ami           = "ami-22b9a343" # us-west-2
  instance_type = "t2.micro"

  network_interface {
    network_interface_id = "${aws_network_interface.foo.id}"
    device_index         = 0
  }

  credit_specification {
    cpu_credits = "unlimited"
  }
}
```

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The instance ID.
- `arn` - The ARN of the instance.
- `availability_zone` - The availability zone of the instance.
- `placement_group` - The placement group of the instance.
- `key_name` - The key name of the instance
- `password_data` - Base-64 encoded encrypted password data for the instance. Useful for getting the administrator password for instances running Microsoft Windows. This attribute is only exported if `get_password_data` is true. Note that this encrypted value will be stored in the state file, as with all exported attributes. See GetPasswordData for more information.
- `public_dns` - The public DNS name assigned to the instance. For EC2-VPC, this is only available if you've enabled DNS hostnames for your VPC
- `public_ip` - The public IP address assigned to the instance, if applicable. **NOTE**: If you are using an `aws_eip` with your instance, you should refer to the EIP's address directly and not use `public_ip`, as this field will change after the EIP is attached.
- `ipv6_addresses` - A list of assigned IPv6 addresses, if any
- `primary_network_interface_id` - The ID of the instance's primary network interface.
- `private_dns` - The private DNS name assigned to the instance. Can only be used inside the Amazon EC2, and only available if you've enabled DNS hostnames for your VPC
- `private_ip` - The private IP address assigned to the instance

- `security_groups` - The associated security groups.
- `vpc_security_group_ids` - The associated security groups in non-default VPC
- `subnet_id` - The VPC subnet ID.
- `credit_specification` - Credit specification of instance.
- `instance_state` - The state of the instance. One of: `pending`, `running`, `shutting-down`, `terminated`, `stopping`, `stopped`. See Instance Lifecycle for more information.

For any `root_block_device` and `ebs_block_device` the `volume_id` is exported. e.g. `aws_instance.web.root_block_device.0.volume_id`

» Import

Instances can be imported using the `id`, e.g.

```
$ terraform import aws_instance.web i-12345678
```

» Resource: aws_key_pair

Provides an EC2 key pair resource. A key pair is used to control login access to EC2 instances.

Currently this resource requires an existing user-supplied key pair. This key pair's public key will be registered with AWS to allow logging-in to EC2 instances.

When importing an existing key pair the public key material may be in any format supported by AWS. Supported formats (per the AWS documentation) are:

- OpenSSH public key format (the format in ~/.ssh/authorized_keys)
- Base64 encoded DER format
- SSH public key file format as specified in RFC4716

» Example Usage

```
resource "aws_key_pair" "deployer" {
  key_name   = "deployer-key"
  public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD3F6tyPEFEzV0LX3X8BsXdMsQz1x2cEikKDEY0
}
```

» Argument Reference

The following arguments are supported:

- `key_name` - (Optional) The name for the key pair.
- `key_name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `key_name`.
- `public_key` - (Required) The public key material.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `key_name` - The key pair name.
- `fingerprint` - The MD5 public key fingerprint as specified in section 4 of RFC 4716.

» Import

Key Pairs can be imported using the `key_name`, e.g.

```
$ terraform import aws_key_pair.deployer deployer-key
```

» Resource: aws_launch_configuration

Provides a resource to create a new launch configuration, used for autoscaling groups.

» Example Usage

```
data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }
  owners = ["099720109477"] # Canonical
}

resource "aws_launch_configuration" "as_conf" {
  name          = "web_config"
  image_id      = "${data.aws_ami.ubuntu.id}"
  instance_type = "t2.micro"
}
```

» Using with AutoScaling Groups

Launch Configurations cannot be updated after creation with the Amazon Web Service API. In order to update a Launch Configuration, Terraform will destroy the existing resource and create a replacement. In order to effectively use a Launch Configuration resource with an AutoScaling Group resource, it's recommended to specify `create_before_destroy` in a lifecycle block. Either omit the Launch Configuration `name` attribute, or specify a partial name with `name_prefix`. Example:

```
data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_launch_configuration" "as_conf" {
  name_prefix   = "terraform-lc-example-"
  image_id      = "${data.aws_ami.ubuntu.id}"
  instance_type = "t2.micro"

  lifecycle {
    create_before_destroy = true
  }
}
```

```
resource "aws_autoscaling_group" "bar" {
  name                 = "terraform-asg-example"
  launch_configuration = "${aws_launch_configuration.as_conf.name}"
  min_size             = 1
  max_size             = 2

  lifecycle {
    create_before_destroy = true
  }
}
```

With this setup Terraform generates a unique name for your Launch Configuration and can then update the AutoScaling Group without conflict before destroying the previous Launch Configuration.

» Using with Spot Instances

Launch configurations can set the spot instance pricing to be used for the Auto Scaling Group to reserve instances. Simply specifying the `spot_price` parameter will set the price on the Launch Configuration which will attempt to reserve your instances at this price. See the AWS Spot Instance documentation for more information or how to launch Spot Instances with Terraform.

```
data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_launch_configuration" "as_conf" {
  image_id      = "${data.aws_ami.ubuntu.id}"
  instance_type = "m4.large"
  spot_price    = "0.001"

  lifecycle {
    create_before_destroy = true
  }
}

resource "aws_autoscaling_group" "bar" {
  name                 = "terraform-asg-example"
  launch_configuration = "${aws_launch_configuration.as_conf.name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the launch configuration. If you leave this blank, Terraform will auto-generate a unique name.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `image_id` - (Required) The EC2 image ID to launch.
- `instance_type` - (Required) The size of instance to launch.
- `iam_instance_profile` - (Optional) The name attribute of the IAM instance profile to associate with launched instances.
- `key_name` - (Optional) The key name that should be used for the instance.
- `security_groups` - (Optional) A list of associated security group IDS.
- `associate_public_ip_address` - (Optional) Associate a public ip address with an instance in a VPC.
- `vpc_classic_link_id` - (Optional) The ID of a ClassicLink-enabled VPC. Only applies to EC2-Classic instances. (eg. `vpc-2730681a`)
- `vpc_classic_link_security_groups` - (Optional) The IDs of one or more security groups for the specified ClassicLink-enabled VPC (eg. `sg-46ae3d11`).
- `user_data` - (Optional) The user data to provide when launching the instance. Do not pass gzip-compressed data via this argument; see `user_data_base64` instead.
- `user_data_base64` - (Optional) Can be used instead of `user_data` to pass base64-encoded binary data directly. Use this instead of `user_data` whenever the value is not a valid UTF-8 string. For example, gzip-encoded user data must be base64-encoded and passed via this argument to avoid corruption.
- `enable_monitoring` - (Optional) Enables/disables detailed monitoring. This is enabled by default.
- `ebs_optimized` - (Optional) If true, the launched EC2 instance will be EBS-optimized.
- `root_block_device` - (Optional) Customize details about the root block device of the instance. See Block Devices below for details.
- `ebs_block_device` - (Optional) Additional EBS block devices to attach to the instance. See Block Devices below for details.

- `ephemeral_block_device` - (Optional) Customize Ephemeral (also known as "Instance Store") volumes on the instance. See Block Devices below for details.
- `spot_price` - (Optional; Default: On-demand price) The maximum price to use for reserving spot instances.
- `placement_tenancy` - (Optional) The tenancy of the instance. Valid values are `"default"` or `"dedicated"`, see AWS's Create Launch Configuration for more details

» Block devices

Each of the `*_block_device` attributes controls a portion of the AWS Launch Configuration's "Block Device Mapping". It's a good idea to familiarize yourself with AWS's Block Device Mapping docs to understand the implications of using these attributes.

The `root_block_device` mapping supports the following:

- `volume_type` - (Optional) The type of volume. Can be `"standard"`, `"gp2"`, or `"io1"`. (Default: `"standard"`).
- `volume_size` - (Optional) The size of the volume in gigabytes.
- `iops` - (Optional) The amount of provisioned IOPS. This must be set with a `volume_type` of `"io1"`.
- `delete_on_termination` - (Optional) Whether the volume should be destroyed on instance termination (Default: `true`).
- `encrypted` - (Optional) Whether the volume should be encrypted or not. (Default: `false`).

Modifying any of the `root_block_device` settings requires resource replacement.

Each `ebs_block_device` supports the following:

- `device_name` - (Required) The name of the device to mount.
- `snapshot_id` - (Optional) The Snapshot ID to mount.
- `volume_type` - (Optional) The type of volume. Can be `"standard"`, `"gp2"`, or `"io1"`. (Default: `"standard"`).
- `volume_size` - (Optional) The size of the volume in gigabytes.
- `iops` - (Optional) The amount of provisioned IOPS. This must be set with a `volume_type` of `"io1"`.
- `delete_on_termination` - (Optional) Whether the volume should be destroyed on instance termination (Default: `true`).
- `encrypted` - (Optional) Whether the volume should be encrypted or not. Do not use this option if you are using `snapshot_id` as the encrypted flag will be determined by the snapshot. (Default: `false`).

Modifying any `ebs_block_device` currently requires resource replacement.

Each `ephemeral_block_device` supports the following:

- `device_name` - The name of the block device to mount on the instance.
- `virtual_name` - The Instance Store Device Name (e.g. `"ephemeral0"`)

Each AWS Instance type has a different set of Instance Store block devices available for attachment. AWS publishes a list of which ephemeral devices are available on each type. The devices are always identified by the `virtual_name` in the format `"ephemeral{0..N}"`.

NOTE: Changes to `*_block_device` configuration of *existing* resources cannot currently be detected by Terraform. After updating to block device configuration, resource recreation can be manually triggered by using the `taint` command.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the launch configuration.
- `arn` - The Amazon Resource Name of the launch configuration.
- `name` - The name of the launch configuration.

» Import

Launch configurations can be imported using the `name`, e.g.

```
$ terraform import aws_launch_configuration.as_conf terraform-lg-123456
```

» Resource: aws_launch_template

Provides an EC2 launch template resource. Can be used to create instances or auto scaling groups.

» Example Usage

```
resource "aws_launch_template" "foo" {
  name = "foo"

  block_device_mappings {
    device_name = "/dev/sda1"

    ebs {
      volume_size = 20
    }
  }

  capacity_reservation_specification {
    capacity_reservation_preference = "open"
  }

  credit_specification {
    cpu_credits = "standard"
  }

  disable_api_termination = true

  ebs_optimized = true

  elastic_gpu_specifications {
    type = "test"
  }

  elastic_inference_accelerator {
    type = "eia1.medium"
  }

  iam_instance_profile {
    name = "test"
  }

  image_id = "ami-test"

  instance_initiated_shutdown_behavior = "terminate"

  instance_market_options {
    market_type = "spot"
  }

  instance_type = "t2.micro"

  kernel_id = "test"

  key_name = "test"

  license_specification {
    license_configuration_arn = "arn:aws:license-manager:eu-west-1:123456789012:license-conf
  }

  monitoring {
    enabled = true
  }

  network_interfaces {
    associate_public_ip_address = true
  }

  placement {
    availability_zone = "us-west-2a"
  }

  ram_disk_id = "test"

  vpc_security_group_ids = ["sg-12345678"]

  tag_specifications {
    resource_type = "instance"

    tags = {
      Name = "test"
    }
  }

  user_data = "${base64encode(...)}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - The name of the launch template. If you leave this blank, Terraform will auto-generate a unique name.
- `name_prefix` - Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `description` - Description of the launch template.
- `block_device_mappings` - Specify volumes to attach to the instance besides the volumes specified by the AMI. See Block Devices below for details.
- `capacity_reservation_specification` - Targeting for EC2 capacity reservations. See Capacity Reservation Specification below for more details.
- `credit_specification` - Customize the credit specification of the instance. See Credit Specification below for more details.
- `disable_api_termination` - If `true`, enables EC2 Instance Termination Protection

- `ebs_optimized` - If `true`, the launched EC2 instance will be EBS-optimized.
- `elastic_gpu_specifications` - The elastic GPU to attach to the instance. See Elastic GPU below for more details.
- `elastic_inference_accelerator` - (Optional) Configuration block containing an Elastic Inference Accelerator to attach to the instance. See Elastic Inference Accelerator below for more details.
- `iam_instance_profile` - The IAM Instance Profile to launch the instance with. See Instance Profile below for more details.
- `image_id` - The AMI from which to launch the instance.
- `instance_initiated_shutdown_behavior` - Shutdown behavior for the instance. Can be `stop` or `terminate`. (Default: `stop`).
- `instance_market_options` - The market (purchasing) option for the instance. See Market Options below for details.
- `instance_type` - The type of the instance.
- `kernel_id` - The kernel ID.
- `key_name` - The key name to use for the instance.
- `license_specification` - A list of license specifications to associate with. See License Specification below for more details.
- `monitoring` - The monitoring option for the instance. See Monitoring below for more details.
- `network_interfaces` - Customize network interfaces to be attached at instance boot time. See Network Interfaces below for more details.
- `placement` - The placement of the instance. See Placement below for more details.
- `ram_disk_id` - The ID of the RAM disk.
- `security_group_names` - A list of security group names to associate with. If you are creating Instances in a VPC, use `vpc_security_group_ids` instead.
- `vpc_security_group_ids` - A list of security group IDs to associate with.
- `tag_specifications` - The tags to apply to the resources during launch. See Tag Specifications below for more details.
- `tags` - (Optional) A mapping of tags to assign to the launch template.
- `user_data` - The Base64-encoded user data to provide when launching the instance.

» Block devices

Configure additional volumes of the instance besides specified by the AMI. It's a good idea to familiarize yourself with AWS's Block Device Mapping docs to understand the implications of using these attributes.

To find out more information for an existing AMI to override the configuration, such as `device_name`, you can use the AWS CLI ec2 describe-images command.

Each `block_device_mappings` supports the following:

- `device_name` - The name of the device to mount.
- `ebs` - Configure EBS volume properties.
- `no_device` - Suppresses the specified device included in the AMI's block device mapping.
- `virtual_name` - The Instance Store Device Name (e.g. `"ephemeral0"`).

The `ebs` block supports the following:

- `delete_on_termination` - Whether the volume should be destroyed on instance termination (Default: `false`). See Preserving Amazon EBS Volumes on Instance Termination for more information.
- `encrypted` - Enables EBS encryption on the volume (Default: `false`). Cannot be used with `snapshot_id`.
- `iops` - The amount of provisioned IOPS. This must be set with a `volume_type` of `"io1"`.
- `kms_key_id` - AWS Key Management Service (AWS KMS) customer master key (CMK) to use when creating the encrypted volume. `encrypted` must be set to `true` when this is set.
- `snapshot_id` - The Snapshot ID to mount.
- `volume_size` - The size of the volume in gigabytes.
- `volume_type` - The type of volume. Can be `"standard"`, `"gp2"`, or `"io1"`. (Default: `"standard"`).

» Capacity Reservation Specification

The `capacity_reservation_specification` block supports the following:

- `capacity_reservation_preference` - Indicates the instance's Capacity Reservation preferences. Can be `open` or `none`. (Default `none`).
- `capacity_reservation_target` - Used to target a specific Capacity Reservation:

The `capacity_reservation_target` block supports the following:

- `capacity_reservation_id` - The ID of the Capacity Reservation to target.

» Credit Specification

Credit specification can be applied/modified to the EC2 Instance at any time.

The `credit_specification` block supports the following:

- `cpu_credits` - The credit option for CPU usage. Can be `"standard"` or `"unlimited"`. T3 instances are launched as unlimited by default. T2 instances are launched as standard by default.

» Elastic GPU

Attach an elastic GPU the instance.

The `elastic_gpu_specifications` block supports the following:

- `type` - The Elastic GPU Type

» Elastic Inference Accelerator

Attach an Elastic Inference Accelerator to the instance. Additional information about Elastic Inference in EC2 can be found in the EC2 User Guide.

The `elastic_inference_accelerator` configuration block supports the following:

- `type` - (Required) Accelerator type.

» Instance Profile

The IAM Instance Profile to attach.

The `iam_instance_profile` block supports the following:

- `arn` - The Amazon Resource Name (ARN) of the instance profile.
- `name` - The name of the instance profile.

» License Specification

Associate one of more license configurations.

The `license_specification` block supports the following:

- `license_configuration_arn` - (Required) ARN of the license configuration.

» Market Options

The market (purchasing) option for the instances.

The `instance_market_options` block supports the following:

- `market_type` - The market type. Can be `spot`.
- `spot_options` - The options for Spot Instance

The `spot_options` block supports the following:

- `block_duration_minutes` - The required duration in minutes. This value must be a multiple of 60.

- `instance_interruption_behavior` - The behavior when a Spot Instance is interrupted. Can be `hibernate`, `stop`, or `terminate`. (Default: `terminate`).
- `max_price` - The maximum hourly price you're willing to pay for the Spot Instances.
- `spot_instance_type` - The Spot Instance request type. Can be `one-time`, or `persistent`.
- `valid_until` - The end date of the request.

» Monitoring

The `monitoring` block supports the following:

- `enabled` - If `true`, the launched EC2 instance will have detailed monitoring enabled.

» Network Interfaces

Attaches one or more Network Interfaces to the instance.

Check limitations for autoscaling group in Creating an Auto Scaling Group Using a Launch Template Guide

Each `network_interfaces` block supports the following:

- `associate_public_ip_address` - Associate a public ip address with the network interface. Boolean value.
- `delete_on_termination` - Whether the network interface should be destroyed on instance termination.
- `description` - Description of the network interface.
- `device_index` - The integer index of the network interface attachment.
- `ipv6_addresses` - One or more specific IPv6 addresses from the IPv6 CIDR block range of your subnet. Conflicts with `ipv6_address_count`
- `ipv6_address_count` - The number of IPv6 addresses to assign to a network interface. Conflicts with `ipv6_addresses`
- `network_interface_id` - The ID of the network interface to attach.
- `private_ip_address` - The primary private IPv4 address.
- `ipv4_address_count` - The number of secondary private IPv4 addresses to assign to a network interface. Conflicts with `ipv4_address_count`
- `ipv4_addresses` - One or more private IPv4 addresses to associate. Conflicts with `ipv4_addresses`
- `security_groups` - A list of security group IDs to associate.
- `subnet_id` - The VPC Subnet ID to associate.

» Placement

The Placement Group of the instance.

The `placement` block supports the following:

- `affinity` - The affinity setting for an instance on a Dedicated Host.
- `availability_zone` - The Availability Zone for the instance.
- `group_name` - The name of the placement group for the instance.
- `host_id` - The ID of the Dedicated Host for the instance.
- `spread_domain` - Reserved for future use.
- `tenancy` - The tenancy of the instance (if the instance is running in a VPC). Can be `default`, `dedicated`, or `host`.

» Tag Specifications

The tags to apply to the resources during launch. You can tag instances and volumes. More information can be found in the EC2 API documentation.

Each `tag_specifications` block supports the following:

- `resource_type` - The type of resource to tag. Valid values are `instance` and `volume`.
- `tags` - A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported along with all argument references:

- `arn` - Amazon Resource Name (ARN) of the launch template.
- `id` - The ID of the launch template.
- `default_version` - The default version of the launch template.
- `latest_version` - The latest version of the launch template.

» Import

Launch Templates can be imported using the `id`, e.g.

```
$ terraform import aws_launch_template.web lt-12345678
```

» Resource: aws_placement_group

Provides an EC2 placement group. Read more about placement groups in AWS Docs.

» Example Usage

```
resource "aws_placement_group" "web" {
  name     = "hunky-dory-pg"
  strategy = "cluster"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the placement group.
- `strategy` - (Required) The placement strategy.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the placement group.

» Import

Placement groups can be imported using the `name`, e.g.

```
$ terraform import aws_placement_group.prod_pg production-placement-group
```

» Resource: aws_snapshot_create_volume_permission

Adds permission to create volumes off of a given EBS Snapshot.

» Example Usage

```
resource "aws_snapshot_create_volume_permission" "example_perm" {
  snapshot_id = "${aws_ebs_snapshot.example_snapshot.id}"
  account_id  = "12345678"
}

resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"
  size              = 40
}
```

```
resource "aws_ebs_snapshot" "example_snapshot" {
  volume_id = "${aws_ebs_volume.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `snapshot_id` - (required) A snapshot ID
- `account_id` - (required) An AWS Account ID to add create volume permissions

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - A combination of "`snapshot_id-account_id`".

» Resource: aws_spot_datafeed_subscription

Note: There is only a single subscription allowed per account.

To help you understand the charges for your Spot instances, Amazon EC2 provides a data feed that describes your Spot instance usage and pricing. This data feed is sent to an Amazon S3 bucket that you specify when you subscribe to the data feed.

» Example Usage

```
resource "aws_s3_bucket" "default" {
  bucket = "tf-spot-datafeed"
}

resource "aws_spot_datafeed_subscription" "default" {
  bucket = "${aws_s3_bucket.default.bucket}"
  prefix = "my_subdirectory"
}
```

» Argument Reference

- `bucket` - (Required) The Amazon S3 bucket in which to store the Spot instance data feed.

- `prefix` - (Optional) Path of folder inside bucket to place spot pricing data.

» Import

A Spot Datafeed Subscription can be imported using the word `spot-datafeed-subscription`, e.g.

```
$ terraform import aws_spot_datafeed_subscription.mysubscription spot-datafeed-subscription
```

» Resource: aws_spot_fleet_request

Provides an EC2 Spot Fleet Request resource. This allows a fleet of Spot instances to be requested on the Spot market.

» Example Usage

```
# Request a Spot fleet
resource "aws_spot_fleet_request" "cheap_compute" {
  iam_fleet_role      = "arn:aws:iam::12345678:role/spot-fleet"
  spot_price          = "0.03"
  allocation_strategy = "diversified"
  target_capacity     = 6
  valid_until         = "2019-11-04T20:44:20Z"

  launch_specification {
    instance_type            = "m4.10xlarge"
    ami                      = "ami-1234"
    spot_price               = "2.793"
    placement_tenancy        = "dedicated"
    iam_instance_profile_arn = "${aws_iam_instance_profile.example.arn}"
  }

  launch_specification {
    instance_type            = "m4.4xlarge"
    ami                      = "ami-5678"
    key_name                 = "my-key"
    spot_price               = "1.117"
    iam_instance_profile_arn = "${aws_iam_instance_profile.example.arn}"
    availability_zone        = "us-west-1a"
    subnet_id                = "subnet-1234"
    weighted_capacity        = 35

    root_block_device {
      volume_size = "300"
      volume_type = "gp2"
    }

    tags = {
      Name = "spot-fleet-example"
    }
  }
}
```

NOTE: Terraform does not support the functionality where multiple `subnet_id` or `availability_zone` parameters can be specified in the same launch configuration block. If you want to specify multiple values, then separate launch configuration blocks should be used:

```
resource "aws_spot_fleet_request" "foo" {
  iam_fleet_role  = "arn:aws:iam::12345678:role/spot-fleet"
  spot_price      = "0.005"
  target_capacity = 2
  valid_until     = "2019-11-04T20:44:20Z"

  launch_specification {
    instance_type     = "m1.small"
    ami               = "ami-d06a90b0"
    key_name          = "my-key"
    availability_zone = "us-west-2a"
  }

  launch_specification {
    instance_type     = "m5.large"
    ami               = "ami-d06a90b0"
    key_name          = "my-key"
    availability_zone = "us-west-2a"
  }
}
```

» Argument Reference

Most of these arguments directly correspond to the official API.

- `iam_fleet_role` - (Required) Grants the Spot fleet permission to terminate Spot instances on your behalf when you cancel its Spot fleet request using CancelSpotFleetRequests or when the Spot fleet request expires, if you set terminateInstancesWithExpiration.
- `replace_unhealthy_instances` - (Optional) Indicates whether Spot fleet should replace unhealthy instances. Default `false`.

- `launch_specification` - Used to define the launch configuration of the spot-fleet request. Can be specified multiple times to define different bids across different markets and instance types.

  Note: This takes in similar but not identical inputs as `aws_instance`. There are limitations on what you can specify. See the list of officially supported inputs in the reference documentation. Any normal `aws_instance` parameter that corresponds to those inputs may be used and it have a additional parameter `iam_instance_profile_arn` takes `aws_iam_instance_profile` attribute `arn` as input.

- `spot_price` - (Optional; Default: On-demand price) The maximum bid price per unit hour.
- `wait_for_fulfillment` - (Optional; Default: false) If set, Terraform will wait for the Spot Request to be fulfilled, and will throw an error if the timeout of 10m is reached.
- `target_capacity` - The number of units to request. You can choose to set the target capacity in terms of instances or a performance characteristic that is important to your application workload, such as vCPUs, memory, or I/O.
- `allocation_strategy` - Indicates how to allocate the target capacity across the Spot pools specified by the Spot fleet request. The default is `lowestPrice`.
- `instance_pools_to_use_count` - (Optional; Default: 1) The number of Spot pools across which to allocate your target Spot capacity. Valid only when `allocation_strategy` is set to `lowestPrice`. Spot Fleet selects the cheapest Spot pools and evenly allocates your target Spot capacity across the number of Spot pools that you specify.
- `excess_capacity_termination_policy` - Indicates whether running Spot instances should be terminated if the target capacity of the Spot fleet request is decreased below the current size of the Spot fleet.
- `terminate_instances_with_expiration` - Indicates whether running Spot instances should be terminated when the Spot fleet request expires.
- `instance_interruption_behaviour` - (Optional) Indicates whether a Spot instance stops or terminates when it is interrupted. Default is `terminate`.
- `fleet_type` - (Optional) The type of fleet request. Indicates whether the Spot Fleet only requests the target capacity or also attempts to maintain it. Default is `maintain`.
- `valid_until` - (Optional) The end date and time of the request, in UTC RFC3339 format(for example, YYYY-MM-DDTHH:MM:SSZ). At this

point, no new Spot instance requests are placed or enabled to fulfill the request. Defaults to 24 hours.

- `valid_from` - (Optional) The start date and time of the request, in UTC RFC3339 format(for example, YYYY-MM-DDTHH:MM:SSZ). The default is to start fulfilling the request immediately.
- `load_balancers` (Optional) A list of elastic load balancer names to add to the Spot fleet.
- `target_group_arns` (Optional) A list of `aws_alb_target_group` ARNs, for use with Application Load Balancing.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Timeouts

The `timeouts` block allows you to specify timeouts for certain actions:

- `create` - (Defaults to 10 mins) Used when requesting the spot instance (only valid if `wait_for_fulfillment = true`)
- `delete` - (Defaults to 5 mins) Used when destroying the spot instance

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Spot fleet request ID
- `spot_request_state` - The state of the Spot fleet request.

» Resource: aws_spot_instance_request

Provides an EC2 Spot Instance Request resource. This allows instances to be requested on the spot market.

By default Terraform creates Spot Instance Requests with a `persistent` type, which means that for the duration of their lifetime, AWS will launch an instance with the configured details if and when the spot market will accept the requested price.

On destruction, Terraform will make an attempt to terminate the associated Spot Instance if there is one present.

Spot Instances requests with a `one-time` type will close the spot request when the instance is terminated either by the request being below the current spot price availability or by a user.

NOTE: Because their behavior depends on the live status of the spot market, Spot Instance Requests have a unique lifecycle that makes them behave differently than other Terraform resources. Most importantly: there is **no guarantee** that a Spot Instance exists to fulfill the request at any given point in time. See the AWS Spot Instance documentation for more information.

» Example Usage

```
# Request a spot instance at $0.03
resource "aws_spot_instance_request" "cheap_worker" {
  ami           = "ami-1234"
  spot_price    = "0.03"
  instance_type = "c4.xlarge"

  tags = {
    Name = "CheapWorker"
  }
}
```

» Argument Reference

Spot Instance Requests support all the same arguments as `aws_instance`, with the addition of:

- `spot_price` - (Optional; Default: On-demand price) The maximum price to request on the spot market.
- `wait_for_fulfillment` - (Optional; Default: false) If set, Terraform will wait for the Spot Request to be fulfilled, and will throw an error if the timeout of 10m is reached.
- `spot_type` - (Optional; Default: `persistent`) If set to `one-time`, after the instance is terminated, the spot request will be closed.
- `launch_group` - (Optional) A launch group is a group of spot instances that launch together and terminate together. If left empty instances are launched and terminated individually.
- `block_duration_minutes` - (Optional) The required duration for the Spot instances, in minutes. This value must be a multiple of 60 (60, 120, 180, 240, 300, or 360). The duration period starts as soon as your Spot instance receives its instance ID. At the end of the duration period, Amazon EC2 marks the Spot instance for termination and provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates. Note that you can't specify an Availability Zone group or a launch group if you specify a duration.
- `instance_interruption_behaviour` - (Optional) Indicates whether a Spot instance stops or terminates when it is interrupted. Default is

terminate as this is the current AWS behaviour.

- valid_until - (Optional) The end date and time of the request, in UTC RFC3339 format(for example, YYYY-MM-DDTHH:MM:SSZ). At this point, no new Spot instance requests are placed or enabled to fulfill the request. The default end date is 7 days from the current date.
- valid_from - (Optional) The start date and time of the request, in UTC RFC3339 format(for example, YYYY-MM-DDTHH:MM:SSZ). The default is to start fulfilling the request immediately.
- tags - (Optional) A mapping of tags to assign to the resource.

» Timeouts

The timeouts block allows you to specify timeouts for certain actions:

- create - (Defaults to 10 mins) Used when requesting the spot instance (only valid if wait_for_fulfillment = true)
- delete - (Defaults to 20 mins) Used when terminating all instances launched via the given spot instance request

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The Spot Instance Request ID.

These attributes are exported, but they are expected to change over time and so should only be used for informational purposes, not for resource dependencies:

- spot_bid_status - The current bid status of the Spot Instance Request.
- spot_request_state The current request state of the Spot Instance Request.
- spot_instance_id - The Instance ID (if any) that is currently fulfilling the Spot Instance request.
- public_dns - The public DNS name assigned to the instance. For EC2-VPC, this is only available if you've enabled DNS hostnames for your VPC
- public_ip - The public IP address assigned to the instance, if applicable.
- private_dns - The private DNS name assigned to the instance. Can only be used inside the Amazon EC2, and only available if you've enabled DNS hostnames for your VPC
- private_ip - The private IP address assigned to the instance

» Resource: aws_volume_attachment

Provides an AWS EBS Volume Attachment as a top level resource, to attach and detach volumes from AWS Instances.

NOTE on EBS block devices: If you use `ebs_block_device` on an `aws_instance`, Terraform will assume management over the full set of non-root EBS block devices for the instance, and treats additional block devices as drift. For this reason, `ebs_block_device` cannot be mixed with external `aws_ebs_volume` + `aws_ebs_volume_attachment` resources for a given instance.

» Example Usage

```
resource "aws_volume_attachment" "ebs_att" {
  device_name = "/dev/sdh"
  volume_id   = "${aws_ebs_volume.example.id}"
  instance_id = "${aws_instance.web.id}"
}

resource "aws_instance" "web" {
  ami               = "ami-21f78e11"
  availability_zone = "us-west-2a"
  instance_type     = "t1.micro"

  tags = {
    Name = "HelloWorld"
  }
}

resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"
  size              = 1
}
```

» Argument Reference

The following arguments are supported:

- `device_name` - (Required) The device name to expose to the instance (for example, `/dev/sdh` or `xvdh`). See Device Naming on Linux Instances and Device Naming on Windows Instances for more information.
- `instance_id` - (Required) ID of the Instance to attach to
- `volume_id` - (Required) ID of the Volume to be attached

- `force_detach` - (Optional, Boolean) Set to `true` if you want to force the volume to detach. Useful if previous attempts failed, but use this option only as a last resort, as this can result in **data loss**. See Detaching an Amazon EBS Volume from an Instance for more information.
- `skip_destroy` - (Optional, Boolean) Set this to true if you do not wish to detach the volume from the instance to which it is attached at destroy time, and instead just remove the attachment from Terraform state. This is useful when destroying an instance which has volumes created by some other means attached.

» Attributes Reference

- `device_name` - The device name exposed to the instance
- `instance_id` - ID of the Instance
- `volume_id` - ID of the Volume

» Data Source: aws_ecr_image

The ECR Image data source allows the details of an image with a particular tag or digest to be retrieved.

» Example Usage

```
data "aws_ecr_image" "service_image" {
  repository_name = "my/service"
  image_tag       = "latest"
}
```

» Argument Reference

The following arguments are supported:

- `registry_id` - (Optional) The ID of the Registry where the repository resides.
- `repository_name` - (Required) The name of the ECR Repository.
- `image_digest` - (Optional) The sha256 digest of the image manifest. At least one of `image_digest` or `image_tag` must be specified.
- `image_tag` - (Optional) The tag associated with this image. At least one of `image_digest` or `image_tag` must be specified.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `image_pushed_at` - The date and time, expressed as a unix timestamp, at which the current image was pushed to the repository.
- `image_size_in_bytes` - The size, in bytes, of the image in the repository.
- `image_tags` - The list of tags associated with this image.

» Data Source: aws_ecr_repository

The ECR Repository data source allows the ARN, Repository URI and Registry ID to be retrieved for an ECR repository.

» Example Usage

```
data "aws_ecr_repository" "service" {
  name = "ecr-repository"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ECR Repository.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Full ARN of the repository.
- `registry_id` - The registry ID where the repository was created.
- `repository_url` - The URL of the repository (in the form `aws_account_id.dkr.ecr.region.amazonaws`
- `tags` - A mapping of tags assigned to the resource.

» Resource: aws_ecr_lifecycle_policy

Manages an ECR repository lifecycle policy.

NOTE: Only one `aws_ecr_lifecycle_policy` resource can be used with the same ECR repository. To apply multiple rules, they must be combined in the `policy` JSON.

NOTE: The AWS ECR API seems to reorder rules based on `rulePriority`. If you define multiple rules that are not sorted in ascending `rulePriority` order in the Terraform code, the resource will be flagged for recreation every `terraform plan`.

» Example Usage

» Policy on untagged image

```
resource "aws_ecr_repository" "foo" {
  name = "bar"
}

resource "aws_ecr_lifecycle_policy" "foopolicy" {
  repository = "${aws_ecr_repository.foo.name}"

  policy = <<EOF
{
    "rules": [
        {
            "rulePriority": 1,
            "description": "Expire images older than 14 days",
            "selection": {
                "tagStatus": "untagged",
                "countType": "sinceImagePushed",
                "countUnit": "days",
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
EOF
}
```

» Policy on tagged image

```
resource "aws_ecr_repository" "foo" {
  name = "bar"
}

resource "aws_ecr_lifecycle_policy" "foopolicy" {
  repository = "${aws_ecr_repository.foo.name}"

  policy = <<EOF
{
    "rules": [
        {
            "rulePriority": 1,
            "description": "Keep last 30 images",
            "selection": {
                "tagStatus": "tagged",
                "tagPrefixList": ["v"],
                "countType": "imageCountMoreThan",
                "countNumber": 30
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `repository` - (Required) Name of the repository to apply the policy.
- `policy` - (Required) The policy document. This is a JSON formatted string. See more details about Policy Parameters in the official AWS docs. For more information about building IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `repository` - The name of the repository.
- `registry_id` - The registry ID where the repository was created.

» Import

ECR Lifecycle Policy can be imported using the name of the repository, e.g.

```
$ terraform import aws_ecr_lifecycle_policy.example tf-example
```

» Resource: aws_ecr_repository

Provides an Elastic Container Registry Repository.

» Example Usage

```
resource "aws_ecr_repository" "foo" {
  name                 = "bar"
  image_tag_mutability = "MUTABLE"

  image_scanning_configuration {
    scan_on_push = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the repository.
- `image_tag_mutability` - (Optional) The tag mutability setting for the repository. Must be one of: `MUTABLE` or `IMMUTABLE`. Defaults to `MUTABLE`.
- `image_scanning_configuration` - (Optional) Configuration block that defines image scanning configuration for the repository. By default, image scanning must be manually triggered. See the ECR User Guide for more information about image scanning.
  - `scan_on_push` - (Required) Indicates whether images are scanned after being pushed to the repository (true) or not scanned (false).
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Full ARN of the repository.
- `name` - The name of the repository.
- `registry_id` - The registry ID where the repository was created.
- `repository_url` - The URL of the repository (in the form `aws_account_id.dkr.ecr.region.amazonaws`

» Timeouts

`aws_ecr_repository` provides the following Timeouts configuration options:

- `delete` - (Default `20 minutes`) How long to wait for a repository to be deleted.

» Import

ECR Repositories can be imported using the `name`, e.g.

```
$ terraform import aws_ecr_repository.service test-service
```

» Resource: aws_ecr_repository_policy

Provides an Elastic Container Registry Repository Policy.

Note that currently only one policy may be applied to a repository.

» Example Usage

```
resource "aws_ecr_repository" "foo" {
  name = "bar"
}

resource "aws_ecr_repository_policy" "foopolicy" {
  repository = "${aws_ecr_repository.foo.name}"

  policy = <<EOF
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "new policy",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "ecr:PutImage",
                "ecr:InitiateLayerUpload",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload",
                "ecr:DescribeRepositories",
                "ecr:GetRepositoryPolicy",
                "ecr:ListImages",
                "ecr:DeleteRepository",
                "ecr:BatchDeleteImage",
                "ecr:SetRepositoryPolicy",
                "ecr:DeleteRepositoryPolicy"
            ]
        }
    ]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `repository` - (Required) Name of the repository to apply the policy.
- `policy` - (Required) The policy document. This is a JSON formatted string. For more information about building IAM policy documents with Terraform, see the AWS IAM Policy Document Guide

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `repository` - The name of the repository.
- `registry_id` - The registry ID where the repository was created.

» Import

ECR Repository Policy can be imported using the repository name, e.g.

```
$ terraform import aws_ecr_repository_policy.example example
```

» Data Source: aws_ecs_cluster

The ECS Cluster data source allows access to details of a specific cluster within an AWS ECS service.

» Example Usage

```
data "aws_ecs_cluster" "ecs-mongo" {
  cluster_name = "ecs-mongo-production"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_name` - (Required) The name of the ECS Cluster

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the ECS Cluster
- `status` - The status of the ECS Cluster
- `pending_tasks_count` - The number of pending tasks for the ECS Cluster
- `running_tasks_count` - The number of running tasks for the ECS Cluster
- `registered_container_instances_count` - The number of registered container instances for the ECS Cluster
- `setting` - The settings associated with the ECS Cluster.

» Data Source: aws_ecs_container_definition

The ECS container definition data source allows access to details of a specific container within an AWS ECS service.

» Example Usage

```
data "aws_ecs_container_definition" "ecs-mongo" {
  task_definition = "${aws_ecs_task_definition.mongo.id}"
  container_name  = "mongodb"
}
```

» Argument Reference

The following arguments are supported:

- `task_definition` - (Required) The ARN of the task definition which contains the container

- `container_name` - (Required) The name of the container definition

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `image` - The docker image in use, including the digest
- `image_digest` - The digest of the docker image in use
- `cpu` - The CPU limit for this container definition
- `memory` - The memory limit for this container definition
- `memory_reservation` - The soft limit (in MiB) of memory to reserve for the container. When system memory is under contention, Docker attempts to keep the container memory to this soft limit
- `environment` - The environment in use
- `disable_networking` - Indicator if networking is disabled
- `docker_labels` - Set docker labels

» Data Source: aws_ecs_service

The ECS Service data source allows access to details of a specific Service within a AWS ECS Cluster.

» Example Usage

```
data "aws_ecs_service" "example" {
  service_name = "example"
  cluster_arn  = "${data.aws_ecs_cluster.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The name of the ECS Service
- `cluster_arn` - (Required) The arn of the ECS Cluster

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the ECS Service
- `desired_count` - The number of tasks for the ECS Service

- `launch_type` - The launch type for the ECS Service
- `scheduling_strategy` - The scheduling strategy for the ECS Service
- `task_definition` - The family for the latest ACTIVE revision

» Data Source: aws_ecs_task_definition

The ECS task definition data source allows access to details of a specific AWS ECS task definition.

» Example Usage

```
# Simply specify the family to find the latest ACTIVE revision in that family.
data "aws_ecs_task_definition" "mongo" {
  task_definition = "${aws_ecs_task_definition.mongo.family}"
}

resource "aws_ecs_cluster" "foo" {
  name = "foo"
}

resource "aws_ecs_task_definition" "mongo" {
  family = "mongodb"

  container_definitions = <<DEFINITION
[
  {
    "cpu": 128,
    "environment": [{
      "name": "SECRET",
      "value": "KEY"
    }],
    "essential": true,
    "image": "mongo:latest",
    "memory": 128,
    "memoryReservation": 64,
    "name": "mongodb"
  }
]
DEFINITION
}

resource "aws_ecs_service" "mongo" {
  name          = "mongo"
  cluster       = "${aws_ecs_cluster.foo.id}"
  desired_count = 2

  # Track the latest ACTIVE revision
  task_definition = "${aws_ecs_task_definition.mongo.family}:${max("${aws_ecs_task_definitio
}
```

» Argument Reference

The following arguments are supported:

- `task_definition` - (Required) The family for the latest ACTIVE revision, family and revision (family:revision) for a specific revision in the family, the ARN of the task definition to access to.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `family` - The family of this task definition
- `network_mode` - The Docker networking mode to use for the containers in this task.
- `revision` - The revision of this task definition
- `status` - The status of this task definition
- `task_role_arn` - The ARN of the IAM role that containers in this task can assume

» Resource: aws_ecs_capacity_provider

Provides an ECS cluster capacity provider. More information can be found on the ECS Developer Guide.

NOTE: The AWS API does not currently support deleting ECS cluster capacity providers. Removing this Terraform resource will only remove the Terraform state for it.

» Example Usage

```
resource "aws_ecs_capacity_provider" "test" {
  name = "test"

  auto_scaling_group_provider {
    auto_scaling_group_arn         = aws_autoscaling_group.test.arn
    managed_termination_protection = "ENABLED"

    managed_scaling {
      maximum_scaling_step_size = 1000
      minimum_scaling_step_size = 1
      status                    = "ENABLED"
      target_capacity           = 10
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the capacity provider.
- `auto_scaling_group_provider` - (Required) Nested argument defining the provider for the ECS auto scaling group. Defined below.
- `tags` - (Optional) Key-value mapping of resource tags.

» auto_scaling_group_provider

The `auto_scaling_group_provider` block supports the following:

- `auto_scaling_group_arn` - (Required) - The Amazon Resource Name (ARN) of the associated auto scaling group.
- `managed_scaling` - (Optional) - Nested argument defining the parameters of the auto scaling. Defined below.
- `managed_termination_protection` - (Optional) - Enables or disables container-aware termination of instances in the auto scaling group when scale-in happens. Valid values are `ENABLED` and `DISABLED`.

» managed_scaling

The `managed_scaling` block supports the following:

- `maximum_scaling_step_size` - (Optional) The maximum step adjustment size. A number between 1 and 10,000.
- `minimum_scaling_step_size` - (Optional) The minimum step adjustment size. A number between 1 and 10,000.
- `status` - (Optional) Whether auto scaling is managed by ECS. Valid values are `ENABLED` and `DISABLED`.
- `target_capacity` - (Optional) The target utilization for the capacity provider. A number between 1 and 100.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) that identifies the capacity provider.
- `arn` - The Amazon Resource Name (ARN) that identifies the capacity provider.

» Import

ECS Capacity Providers can be imported using the `name`, e.g.

```
$ terraform import aws_ecs_capacity_provider.example example
```

» Resource: aws_ecs_cluster

Provides an ECS cluster.

» Example Usage

```
resource "aws_ecs_cluster" "foo" {
  name = "white-hart"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the cluster (up to 255 letters, numbers, hyphens, and underscores)
- `capacity_providers` - (Optional) List of short names of one or more capacity providers to associate with the cluster. Valid values also include `FARGATE` and `FARGATE_SPOT`.
- `default_capacity_provider_strategy` - (Optional) The capacity provider strategy to use by default for the cluster. Can be one or more. Defined below.
- `tags` - (Optional) Key-value mapping of resource tags
- `setting` - (Optional) Configuration block(s) with cluster settings. For example, this can be used to enable CloudWatch Container Insights for a cluster. Defined below.

» setting

The `setting` configuration block supports the following:

- `name` - (Required) Name of the setting to manage. Valid values: `containerInsights`.
- `value` - (Required) The value to assign to the setting. Value values are `enabled` and `disabled`.

» default_capacity_provider_strategy

The `default_capacity_provider_strategy` configuration block supports the following:

- `capacity_provider` - (Required) The short name of the capacity provider.
- `weight` - (Required) The relative percentage of the total number of launched tasks that should use the specified capacity provider.
- `base` - (Optional) The number of tasks, at a minimum, to run on the specified capacity provider. Only one capacity provider in a capacity provider strategy can have a base defined.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) that identifies the cluster
- `arn` - The Amazon Resource Name (ARN) that identifies the cluster

» Import

ECS clusters can be imported using the `name`, e.g.

```
$ terraform import aws_ecs_cluster.stateless stateless-app
```

» Resource: aws_ecs_service

Note: To prevent a race condition during service deletion, make sure to set `depends_on` to the related `aws_iam_role_policy`; otherwise, the policy may be destroyed too soon and the ECS service will then get stuck in the `DRAINING` state.

Provides an ECS service - effectively a task that is expected to run until an error occurs or a user terminates it (typically a webserver or a database).

See ECS Services section in AWS developer guide.

» Example Usage

```
resource "aws_ecs_service" "mongo" {
  name            = "mongodb"
  cluster         = "${aws_ecs_cluster.foo.id}"
  task_definition = "${aws_ecs_task_definition.mongo.arn}"
  desired_count   = 3
  iam_role        = "${aws_iam_role.foo.arn}"
  depends_on      = ["aws_iam_role_policy.foo"]

  ordered_placement_strategy {
    type  = "binpack"
    field = "cpu"
  }

  load_balancer {
    target_group_arn = "${aws_lb_target_group.foo.arn}"
    container_name   = "mongo"
    container_port   = 8080
  }

  placement_constraints {
    type       = "memberOf"
    expression = "attribute:ecs.availability-zone in [us-west-2a, us-west-2b]"
  }
}
```

» Ignoring Changes to Desired Count

You can utilize the generic Terraform resource lifecycle configuration block with `ignore_changes` to create an ECS service with an initial count of running instances, then ignore any changes to that count caused externally (e.g. Application Autoscaling).

```
resource "aws_ecs_service" "example" {
  # ... other configurations ...

  # Example: Create service with 2 instances to start
  desired_count = 2

  # Optional: Allow external changes without Terraform plan difference
  lifecycle {
    ignore_changes = ["desired_count"]
  }
}
```

» Daemon Scheduling Strategy

```
resource "aws_ecs_service" "bar" {
  name                = "bar"
  cluster             = "${aws_ecs_cluster.foo.id}"
  task_definition     = "${aws_ecs_task_definition.bar.arn}"
  scheduling_strategy = "DAEMON"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the service (up to 255 letters, numbers, hyphens, and underscores)
- `capacity_provider_strategy` - (Optional) The capacity provider strategy to use for the service. Can be one or more. Defined below.
- `cluster` - (Optional) ARN of an ECS cluster
- `deployment_controller` - (Optional) Configuration block containing deployment controller configuration. Defined below.
- `deployment_maximum_percent` - (Optional) The upper limit (as a percentage of the service's desiredCount) of the number of running tasks that can be running in a service during a deployment. Not valid when using the `DAEMON` scheduling strategy.
- `deployment_minimum_healthy_percent` - (Optional) The lower limit (as a percentage of the service's desiredCount) of the number of running tasks that must remain running and healthy in a service during a deployment.
- `desired_count` - (Optional) The number of instances of the task definition to place and keep running. Defaults to 0. Do not specify if using the `DAEMON` scheduling strategy.
- `enable_ecs_managed_tags` - (Optional) Specifies whether to enable Amazon ECS managed tags for the tasks within the service.
- `health_check_grace_period_seconds` - (Optional) Seconds to ignore failing load balancer health checks on newly instantiated tasks to prevent premature shutdown, up to 2147483647. Only valid for services configured to use load balancers.
- `iam_role` - (Optional) ARN of the IAM role that allows Amazon ECS to make calls to your load balancer on your behalf. This parameter is required if you are using a load balancer with your service, but only if your task definition does not use the `awsvpc` network mode. If using `awsvpc` network mode, do not specify this role. If your account has already created the Amazon ECS service-linked role, that role is used by default for your service unless you specify a role here.
- `launch_type` - (Optional) The launch type on which to run your service. The valid values are `EC2` and `FARGATE`. Defaults to `EC2`.

- `load_balancer` - (Optional) A load balancer block. Load balancers documented below.
- `network_configuration` - (Optional) The network configuration for the service. This parameter is required for task definitions that use the `awsvpc` network mode to receive their own Elastic Network Interface, and it is not supported for other network modes.
- `ordered_placement_strategy` - (Optional) Service level strategy rules that are taken into consideration during task placement. List from top to bottom in order of precedence. The maximum number of `ordered_placement_strategy` blocks is `5`. Defined below.
- `placement_constraints` - (Optional) rules that are taken into consideration during task placement. Maximum number of `placement_constraints` is `10`. Defined below.
- `platform_version` - (Optional) The platform version on which to run your service. Only applicable for `launch_type` set to `FARGATE`. Defaults to `LATEST`. More information about Fargate platform versions can be found in the AWS ECS User Guide.
- `propagate_tags` - (Optional) Specifies whether to propagate the tags from the task definition or the service to the tasks. The valid values are `SERVICE` and `TASK_DEFINITION`.
- `scheduling_strategy` - (Optional) The scheduling strategy to use for the service. The valid values are `REPLICA` and `DAEMON`. Defaults to `REPLICA`. Note that *Fargate tasks do not support the `DAEMON` scheduling strategy*.
- `service_registries` - (Optional) The service discovery registries for the service. The maximum number of `service_registries` blocks is `1`.
- `tags` - (Optional) Key-value mapping of resource tags
- `task_definition` - (Required) The family and revision (`family:revision`) or full ARN of the task definition that you want to run in your service.

» capacity_provider_strategy

The `capacity_provider_strategy` configuration block supports the following:

- `capacity_provider` - (Required) The short name or full Amazon Resource Name (ARN) of the capacity provider.
- `weight` - (Required) The relative percentage of the total number of launched tasks that should use the specified capacity provider.
- `base` - (Optional) The number of tasks, at a minimum, to run on the specified capacity provider. Only one capacity provider in a capacity provider strategy can have a base defined.

» deployment_controller

The `deployment_controller` configuration block supports the following:

- type - (Optional) Type of deployment controller. Valid values: CODE_DEPLOY, ECS. Default: ECS.

» load_balancer

load_balancer supports the following:

- elb_name - (Required for ELB Classic) The name of the ELB (Classic) to associate with the service.
- target_group_arn - (Required for ALB/NLB) The ARN of the Load Balancer target group to associate with the service.
- container_name - (Required) The name of the container to associate with the load balancer (as it appears in a container definition).
- container_port - (Required) The port on the container to associate with the load balancer.

Version note: Multiple load_balancer configuration block support was added in Terraform AWS Provider version 2.22.0. This allows configuration of ECS service support for multiple target groups.

» ordered_placement_strategy

ordered_placement_strategy supports the following:

- type - (Required) The type of placement strategy. Must be one of: binpack, random, or spread
- field - (Optional) For the spread placement strategy, valid values are instanceId (or host, which has the same effect), or any platform or custom attribute that is applied to a container instance. For the binpack type, valid values are memory and cpu. For the random type, this attribute is not needed. For more information, see Placement Strategy.

Note: for spread, host and instanceId will be normalized, by AWS, to be instanceId. This means the statefile will show instanceId but your config will differ if you use host.

» placement_constraints

placement_constraints support the following:

- type - (Required) The type of constraint. The only valid values at this time are memberOf and distinctInstance.
- expression - (Optional) Cluster Query Language expression to apply to the constraint. Does not need to be specified for the distinctInstance type. For more information, see Cluster Query Language in the Amazon EC2 Container Service Developer Guide.

» network_configuration

`network_configuration` support the following:

- `subnets` - (Required) The subnets associated with the task or service.
- `security_groups` - (Optional) The security groups associated with the task or service. If you do not specify a security group, the default security group for the VPC is used.
- `assign_public_ip` - (Optional) Assign a public IP address to the ENI (Fargate launch type only). Valid values are `true` or `false`. Default `false`.

For more information, see Task Networking

» service_registries

`service_registries` support the following:

- `registry_arn` - (Required) The ARN of the Service Registry. The currently supported service registry is Amazon Route 53 Auto Naming Service(`aws_service_discovery_service`). For more information, see Service
- `port` - (Optional) The port value used if your Service Discovery service specified an SRV record.
- `container_port` - (Optional) The port value, already specified in the task definition, to be used for your service discovery service.
- `container_name` - (Optional) The container name value, already specified in the task definition, to be used for your service discovery service.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) that identifies the service
- `name` - The name of the service
- `cluster` - The Amazon Resource Name (ARN) of cluster which the service runs on
- `iam_role` - The ARN of IAM role used for ELB
- `desired_count` - The number of instances of the task definition

» Import

ECS services can be imported using the `name` together with ecs cluster `name`, e.g.

```
$ terraform import aws_ecs_service.imported cluster-name/service-name
```

» Resource: aws_ecs_task_definition

Manages a revision of an ECS task definition to be used in `aws_ecs_service`.

» Example Usage

```
resource "aws_ecs_task_definition" "service" {
  family                = "service"
  container_definitions = "${file("task-definitions/service.json")}"

  volume {
    name      = "service-storage"
    host_path = "/ecs/service-storage"
  }

  placement_constraints {
    type       = "memberOf"
    expression = "attribute:ecs.availability-zone in [us-west-2a, us-west-2b]"
  }
}
```

The referenced `task-definitions/service.json` file contains a valid JSON document, which is shown below, and its content is going to be passed directly into the `container_definitions` attribute as a string. Please note that this example contains only a small subset of the available parameters.

```
[
  {
    "name": "first",
    "image": "service-first",
    "cpu": 10,
    "memory": 512,
    "essential": true,
    "portMappings": [
      {
        "containerPort": 80,
        "hostPort": 80
      }
    ]
  },
  {
    "name": "second",
    "image": "service-second",
    "cpu": 10,
    "memory": 256,
    "essential": true,
    "portMappings": [
      {
        "containerPort": 443,
        "hostPort": 443
      }
    ]
  }
]
```

» With AppMesh Proxy

```
resource "aws_ecs_task_definition" "service" {
  family                = "service"
  container_definitions = "${file("task-definitions/service.json")}"

  proxy_configuration {
    type           = "APPMESH"
    container_name = "applicationContainerName"
    properties = {
      AppPorts         = "8080"
      EgressIgnoredIPs = "169.254.170.2,169.254.169.254"
      IgnoredUID       = "1337"
      ProxyEgressPort  = 15001
      ProxyIngressPort = 15000
    }
  }
}
```

» Argument Reference

» Top-Level Arguments

- `family` - (Required) A unique name for your task definition.
- `container_definitions` - (Required) A list of valid container definitions provided as a single valid JSON document. Please note that you should only provide values that are part of the container definition document. For a detailed description of what parameters are available, see the Task Definition Parameters section from the official Developer Guide.

NOTE: Proper escaping is required for JSON field values containing quotes (") such as `environment` values. If directly setting the JSON, they should be escaped as `\"` in the JSON, e.g. `"value": "I \"love\" escaped quotes"`. If using a Terraform variable value, they should be escaped as `\\\"` in the variable, e.g. `value = "I \\\"love\\\" escaped quotes"` in the variable and `"value": "${var.myvariable}"` in the JSON.

- `task_role_arn` - (Optional) The ARN of IAM role that allows your Amazon ECS container task to make calls to other AWS services.
- `execution_role_arn` - (Optional) The Amazon Resource Name (ARN) of the task execution role that the Amazon ECS container agent and the Docker daemon can assume.
- `network_mode` - (Optional) The Docker networking mode to use for the containers in the task. The valid values are `none`, `bridge`, `awsvpc`, and `host`.
- `ipc_mode` - (Optional) The IPC resource namespace to be used for the containers in the task The valid values are `host`, `task`, and `none`.
- `pid_mode` - (Optional) The process namespace to use for the containers in the task. The valid values are `host` and `task`.
- `volume` - (Optional) A set of volume blocks that containers in your task may use.
- `placement_constraints` - (Optional) A set of placement constraints rules that are taken into consideration during task placement. Maximum number of `placement_constraints` is `10`.
- `cpu` - (Optional) The number of cpu units used by the task. If the `requires_compatibilities` is `FARGATE` this field is required.
- `memory` - (Optional) The amount (in MiB) of memory used by the task. If the `requires_compatibilities` is `FARGATE` this field is required.
- `requires_compatibilities` - (Optional) A set of launch types required by the task. The valid values are `EC2` and `FARGATE`.
- `proxy_configuration` - (Optional) The proxy configuration details for the App Mesh proxy.
- `tags` - (Optional) Key-value mapping of resource tags

» Volume Block Arguments

- `name` - (Required) The name of the volume. This name is referenced in the `sourceVolume` parameter of container definition in the `mountPoints` section.
- `host_path` - (Optional) The path on the host container instance that is presented to the container. If not set, ECS will create a nonpersistent data volume that starts empty and is deleted after the task has finished.
- `docker_volume_configuration` - (Optional) Used to configure a docker volume

» Docker Volume Configuration Arguments

For more information, see Specifying a Docker volume in your Task Definition Developer Guide

- `scope` - (Optional) The scope for the Docker volume, which determines its lifecycle, either `task` or `shared`. Docker volumes that are scoped to a `task` are automatically provisioned when the task starts and destroyed when the task stops. Docker volumes that are `scoped` as shared persist after the task stops.
- `autoprovision` - (Optional) If this value is `true`, the Docker volume is created if it does not already exist. *Note*: This field is only used if the scope is `shared`.
- `driver` - (Optional) The Docker volume driver to use. The driver value must match the driver name provided by Docker because it is used for task placement.
- `driver_opts` - (Optional) A map of Docker driver specific options.
- `labels` - (Optional) A map of custom metadata to add to your Docker volume.

» Example Usage:

```
resource "aws_ecs_task_definition" "service" {
  family                = "service"
  container_definitions = "${file("task-definitions/service.json")}"

  volume {
    name = "service-storage"

    docker_volume_configuration {
      scope         = "shared"
      autoprovision = true
    }
  }
}
```

» Placement Constraints Arguments

- `type` - (Required) The type of constraint. Use `memberOf` to restrict selection to a group of valid candidates. Note that `distinctInstance` is not supported in task definitions.
- `expression` - (Optional) Cluster Query Language expression to apply to the constraint. For more information, see Cluster Query Language in the Amazon EC2 Container Service Developer Guide.

» Proxy Configuration Arguments

- `container_name` - (Required) The name of the container that will serve as the App Mesh proxy.
- `properties` - (Required) The set of network configuration parameters to provide the Container Network Interface (CNI) plugin, specified a key-value mapping.
- `type` - (Optional) The proxy type. The default value is `APPMESH`. The only supported value is `APPMESH`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Full ARN of the Task Definition (including both `family` and `revision`).
- `family` - The family of the Task Definition.
- `revision` - The revision of the task in a particular family.

» Import

ECS Task Definitions can be imported via their Amazon Resource Name (ARN):

```
$ terraform import aws_ecs_task_definition.example arn:aws:ecs:us-east-1:012345678910:task-d
```

» Data Source: aws_efs_file_system

Provides information about an Elastic File System (EFS).

» Example Usage

```
variable "file_system_id" {
  type    = "string"
  default = ""
}

data "aws_efs_file_system" "by_id" {
  file_system_id = "${var.file_system_id}"
}
```

» Argument Reference

The following arguments are supported:

- `file_system_id` - (Optional) The ID that identifies the file system (e.g. fs-ccfc0d65).
- `creation_token` - (Optional) Restricts the list to the file system with this creation token.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name of the file system.
- `performance_mode` - The PerformanceMode of the file system.
- `tags` - The list of tags assigned to the file system.
- `encrypted` - Whether EFS is encrypted.
- `kms_key_id` - The ARN for the KMS encryption key.
- `dns_name` - The DNS name for the filesystem per documented convention.

» Data Source: aws_efs_mount_target

Provides information about an Elastic File System Mount Target (EFS).

» Example Usage

```
variable "mount_target_id" {
  type    = "string"
  default = ""
}

data "aws_efs_mount_target" "by_id" {
  mount_target_id = "${var.mount_target_id}"
}
```

» Argument Reference

The following arguments are supported:

- `mount_target_id` - (Required) ID of the mount target that you want to have described

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- file_system_arn - Amazon Resource Name of the file system for which the mount target is intended.
- file_system_id - ID of the file system for which the mount target is intended.
- subnet_id - ID of the mount target's subnet.
- ip_address - Address at which the file system may be mounted via the mount target.
- security_groups - List of VPC security group IDs attached to the mount target.
- dns_name - The DNS name for the given subnet/AZ per documented convention.
- network_interface_id - The ID of the network interface that Amazon EFS created when it created the mount target.

» Resource: aws_efs_file_system

Provides an Elastic File System (EFS) resource.

» Example Usage

» EFS File System w/ tags

```
resource "aws_efs_file_system" "foo" {
  creation_token = "my-product"

  tags = {
    Name = "MyProduct"
  }
}
```

» Using lifecycle policy

```
resource "aws_efs_file_system" "foo_with_lifecyle_policy" {
  creation_token = "my-product"

  lifecycle_policy {
    transition_to_ia = "AFTER_30_DAYS"
  }
}
```

» Argument Reference

The following arguments are supported:

- `creation_token` - (Optional) A unique name (a maximum of 64 characters are allowed) used as reference when creating the Elastic File System to ensure idempotent file system creation. By default generated by Terraform. See Elastic File System user guide for more information.
- `encrypted` - (Optional) If true, the disk will be encrypted.
- `kms_key_id` - (Optional) The ARN for the KMS encryption key. When specifying kms_key_id, encrypted needs to be set to true.
- `lifecycle_policy` - (Optional) A file system lifecycle policy object (documented below).
- `performance_mode` - (Optional) The file system performance mode. Can be either `"generalPurpose"` or `"maxIO"` (Default: `"generalPurpose"`).
- `provisioned_throughput_in_mibps` - (Optional) The throughput, measured in MiB/s, that you want to provision for the file system. Only applicable with `throughput_mode` set to `provisioned`.
- `tags` - (Optional) A mapping of tags to assign to the file system.
- `throughput_mode` - (Optional) Throughput mode for the file system. Defaults to `bursting`. Valid values: `bursting`, `provisioned`. When using `provisioned`, also set `provisioned_throughput_in_mibps`.

» Lifecycle Policy Arguments

For **lifecycle_policy** the following attributes are supported:

- `transition_to_ia` - (Optional) Indicates how long it takes to transition files to the IA storage class. Valid values: `AFTER_7_DAYS`, `AFTER_14_DAYS`, `AFTER_30_DAYS`, `AFTER_60_DAYS`, or `AFTER_90_DAYS`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name of the file system.
- `id` - The ID that identifies the file system (e.g. fs-ccfc0d65).
- `dns_name` - The DNS name for the filesystem per documented convention.

» Import

The EFS file systems can be imported using the `id`, e.g.

```
$ terraform import aws_efs_file_system.foo fs-6fa144c6
```

» Resource: aws_efs_mount_target

Provides an Elastic File System (EFS) mount target.

» Example Usage

```
resource "aws_efs_mount_target" "alpha" {
  file_system_id = "${aws_efs_file_system.foo.id}"
  subnet_id      = "${aws_subnet.alpha.id}"
}

resource "aws_vpc" "foo" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "alpha" {
  vpc_id            = "${aws_vpc.foo.id}"
  availability_zone = "us-west-2a"
  cidr_block        = "10.0.1.0/24"
}
```

» Argument Reference

The following arguments are supported:

- `file_system_id` - (Required) The ID of the file system for which the mount target is intended.
- `subnet_id` - (Required) The ID of the subnet to add the mount target in.
- `ip_address` - (Optional) The address (within the address range of the specified subnet) at which the file system may be mounted via the mount target.
- `security_groups` - (Optional) A list of up to 5 VPC security group IDs (that must be for the same VPC as subnet specified) in effect for the mount target.

» Attributes Reference

Note: The `dns_name` attribute is only useful if the mount target is in a VPC that has support for DNS hostnames enabled. See Using DNS with Your VPC and VPC resource in Terraform for more information.

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the mount target.

- `dns_name` - The DNS name for the given subnet/AZ per documented convention.
- `file_system_arn` - Amazon Resource Name of the file system.
- `network_interface_id` - The ID of the network interface that Amazon EFS created when it created the mount target.

» Import

The EFS mount targets can be imported using the `id`, e.g.

```
$ terraform import aws_efs_mount_target.alpha fsmt-52a643fb
```

» Data Source: aws_eks_cluster

Retrieve information about an EKS Cluster.

» Example Usage

```
data "aws_eks_cluster" "example" {
  name = "example"
}

output "endpoint" {
  value = "${data.aws_eks_cluster.example.endpoint}"
}

output "kubeconfig-certificate-authority-data" {
  value = "${data.aws_eks_cluster.example.certificate_authority.0.data}"
}

# Only available on Kubernetes version 1.13 and 1.14 clusters created or upgraded on or afte
output "identity-oidc-issuer" {
  value = "${data.aws_eks_cluster.example.identity.0.oidc.0.issuer}"
}
```

» Argument Reference

- `name` - (Required) The name of the cluster

» Attributes Reference

- `id` - The name of the cluster
- `arn` - The Amazon Resource Name (ARN) of the cluster.
- `certificate_authority` - Nested attribute containing `certificate-authority-data` for your cluster.
  - `data` - The base64 encoded certificate data required to communicate with your cluster. Add this to the `certificate-authority-data` section of the `kubeconfig` file for your cluster.
- `created_at` - The Unix epoch time stamp in seconds for when the cluster was created.
- `enabled_cluster_log_types` - The enabled control plane logs.
- `endpoint` - The endpoint for your Kubernetes API server.
- `identity` - Nested attribute containing identity provider information for your cluster. Only available on Kubernetes version 1.13 and 1.14 clusters created or upgraded on or after September 3, 2019. For an example using this information to enable IAM Roles for Service Accounts, see the `aws_eks_cluster` resource documentation.
  - `oidc` - Nested attribute containing OpenID Connect identity provider information for the cluster.
  - `issuer` - Issuer URL for the OpenID Connect identity provider.
- `platform_version` - The platform version for the cluster.
- `role_arn` - The Amazon Resource Name (ARN) of the IAM role that provides permissions for the Kubernetes control plane to make calls to AWS API operations on your behalf.
- `status` - The status of the EKS cluster. One of `CREATING`, `ACTIVE`, `DELETING`, `FAILED`.
- `tags` - Key-value mapping of resource tags.
- `version` - The Kubernetes server version for the cluster.
- `vpc_config` - Nested list containing VPC configuration for the cluster.
  - `cluster_security_group_id` - The cluster security group that was created by Amazon EKS for the cluster.
  - `endpoint_private_access` - Indicates whether or not the Amazon EKS private API server endpoint is enabled.
  - `endpoint_public_access` - Indicates whether or not the Amazon EKS public API server endpoint is enabled.
  - `public_access_cidrs` - List of CIDR blocks. Indicates which CIDR blocks can access the Amazon EKS public API server endpoint.
  - `security_group_ids` – List of security group IDs
  - `subnet_ids` – List of subnet IDs
  - `vpc_id` – The VPC associated with your cluster.

» Data Source: aws_eks_cluster_auth

Get an authentication token to communicate with an EKS cluster.

Uses IAM credentials from the AWS provider to generate a temporary token that is compatible with AWS IAM Authenticator authentication. This can be used to authenticate to an EKS cluster or to a cluster that has the AWS IAM Authenticator server configured.

» Example Usage

```
data "aws_eks_cluster" "example" {
  name = "example"
}

data "aws_eks_cluster_auth" "example" {
  name = "example"
}

provider "kubernetes" {
  host                   = "${data.aws_eks_cluster.example.endpoint}"
  cluster_ca_certificate = "${base64decode(data.aws_eks_cluster.example.certificate_authorit
  token                  = "${data.aws_eks_cluster_auth.example.token}"
  load_config_file       = false
}
```

» Argument Reference

- `name` - (Required) The name of the cluster

» Attributes Reference

- `token` - The token to use to authenticate with the cluster.

» Resource: aws_eks_cluster

Manages an EKS Cluster.

» Example Usage

» Basic Usage

```
resource "aws_eks_cluster" "example" {
  name     = "example"
  role_arn = "${aws_iam_role.example.arn}"

  vpc_config {
    subnet_ids = ["${aws_subnet.example1.id}", "${aws_subnet.example2.id}"]
  }

  # Ensure that IAM Role permissions are created before and deleted after EKS Cluster handli
  # Otherwise, EKS will not be able to properly delete EKS managed EC2 infrastructure such a
  depends_on = [
    "aws_iam_role_policy_attachment.example-AmazonEKSClusterPolicy",
    "aws_iam_role_policy_attachment.example-AmazonEKSServicePolicy",
  ]
}

output "endpoint" {
  value = "${aws_eks_cluster.example.endpoint}"
}

output "kubeconfig-certificate-authority-data" {
  value = "${aws_eks_cluster.example.certificate_authority.0.data}"
}
```

» Example IAM Role for EKS Cluster

```
resource "aws_iam_role" "example" {
  name = "eks-cluster-example"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy_attachment" "example-AmazonEKSClusterPolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
  role       = "${aws_iam_role.example.name}"
}

resource "aws_iam_role_policy_attachment" "example-AmazonEKSServicePolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSServicePolicy"
  role       = "${aws_iam_role.example.name}"
}
```

» Enabling Control Plane Logging

EKS Control Plane Logging can be enabled via the `enabled_cluster_log_types` argument. To manage the CloudWatch Log Group retention period, the `aws_cloudwatch_log_group` resource can be used.

The below configuration uses `depends_on` to prevent ordering issues with EKS automatically creating the log group first and a variable for naming consistency. Other ordering and naming methodologies may be more appropriate for your environment.

```
variable "cluster_name" {
  default = "example"
  type    = "string"
}

resource "aws_eks_cluster" "example" {
  depends_on = ["aws_cloudwatch_log_group.example"]

  enabled_cluster_log_types = ["api", "audit"]
  name                      = "${var.cluster_name}"

  # ... other configuration ...
}

resource "aws_cloudwatch_log_group" "example" {
  name              = "/aws/eks/${var.cluster_name}/cluster"
  retention_in_days = 7

  # ... potentially other configuration ...
}
```

» Enabling IAM Roles for Service Accounts

Only available on Kubernetes version 1.13 and 1.14 clusters created or upgraded on or after September 3, 2019. For more information about this feature, see the EKS User Guide.

```
resource "aws_eks_cluster" "example" {
  # ... other configuration ...
}

resource "aws_iam_openid_connect_provider" "example" {
  client_id_list  = ["sts.amazonaws.com"]
  thumbprint_list = []
  url             = "${aws_eks_cluster.example.identity.0.oidc.0.issuer}"
}

data "aws_caller_identity" "current" {}

data "aws_iam_policy_document" "example_assume_role_policy" {
  statement {
    actions = ["sts:AssumeRoleWithWebIdentity"]
    effect  = "Allow"

    condition {
      test     = "StringEquals"
      variable = "${replace(aws_iam_openid_connect_provider.example.url, "https://", "")}:su
      values   = ["system:serviceaccount:kube-system:aws-node"]
    }

    principals {
      identifiers = ["${aws_iam_openid_connect_provider.example.arn}"]
      type        = "Federated"
    }
  }
}

resource "aws_iam_role" "example" {
  assume_role_policy = "${data.aws_iam_policy_document.example_assume_role_policy.json}"
  name               = "example"
}
```

After adding inline IAM Policies (e.g. `aws_iam_role_policy` resource) or attaching IAM Policies (e.g. `aws_iam_policy` resource and `aws_iam_role_policy_attachment` resource) with the desired permissions to the IAM Role, annotate the Kubernetes service account (e.g. `kubernetes_service_account` resource) and recreate any pods.

» Argument Reference

The following arguments are supported:

- `name` – (Required) Name of the cluster.
- `role_arn` - (Required) The Amazon Resource Name (ARN) of the IAM role that provides permissions for the Kubernetes control plane to make calls to AWS API operations on your behalf. Ensure the resource configuration includes explicit dependencies on the IAM Role permissions by adding `depends_on` if using the `aws_iam_role_policy` resource or `aws_iam_role_policy_attachment` resource, otherwise EKS cannot delete EKS managed EC2 infrastructure such as Security Groups on EKS Cluster deletion.
- `vpc_config` - (Required) Nested argument for the VPC associated with your cluster. Amazon EKS VPC resources have specific requirements to work properly with Kubernetes. For more information, see Cluster VPC Considerations and Cluster Security Group Considerations in the Amazon EKS User Guide. Configuration detailed below.
- `enabled_cluster_log_types` - (Optional) A list of the desired control plane logging to enable. For more information, see Amazon EKS Control Plane Logging
- `tags` - (Optional) Key-value mapping of resource tags.
- `version` – (Optional) Desired Kubernetes master version. If you do not specify a value, the latest available version at resource creation is used and no upgrades will occur except those automatically triggered by EKS. The value must be configured and increased to upgrade the version when desired. Downgrades are not supported by EKS.

» vpc_config

- `endpoint_private_access` - (Optional) Indicates whether or not the Amazon EKS private API server endpoint is enabled. Default is `false`.
- `endpoint_public_access` - (Optional) Indicates whether or not the Amazon EKS public API server endpoint is enabled. Default is `true`.
- `public_access_cidrs` - (Optional) List of CIDR blocks. Indicates which CIDR blocks can access the Amazon EKS public API server endpoint when enabled. EKS defaults this to a list with `0.0.0.0/0`. Terraform will only perform drift detection of its value when present in a configuration.
- `security_group_ids` – (Optional) List of security group IDs for the cross-account elastic network interfaces that Amazon EKS creates to use to allow communication between your worker nodes and the Kubernetes control plane.
- `subnet_ids` – (Required) List of subnet IDs. Must be in at least two different availability zones. Amazon EKS creates cross-account elastic network interfaces in these subnets to allow communication between your

worker nodes and the Kubernetes control plane.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the cluster.
- `arn` - The Amazon Resource Name (ARN) of the cluster.
- `certificate_authority` - Nested attribute containing `certificate-authority-data` for your cluster.
  - `data` - The base64 encoded certificate data required to communicate with your cluster. Add this to the `certificate-authority-data` section of the `kubeconfig` file for your cluster.
- `endpoint` - The endpoint for your Kubernetes API server.
- `identity` - Nested attribute containing identity provider information for your cluster. Only available on Kubernetes version 1.13 and 1.14 clusters created or upgraded on or after September 3, 2019.
  - `oidc` - Nested attribute containing OpenID Connect identity provider information for the cluster.
  - `issuer` - Issuer URL for the OpenID Connect identity provider.
- `platform_version` - The platform version for the cluster.
- `status` - The status of the EKS cluster. One of `CREATING`, `ACTIVE`, `DELETING`, `FAILED`.
- `version` - The Kubernetes server version for the cluster.
- `vpc_config` - Additional nested attributes:
  - `cluster_security_group_id` - The cluster security group that was created by Amazon EKS for the cluster.
  - `vpc_id` - The VPC associated with your cluster.

» Timeouts

`aws_eks_cluster` provides the following Timeouts configuration options:

- `create` - (Default `30 minutes`) How long to wait for the EKS Cluster to be created.
- `update` - (Default `60 minutes`) How long to wait for the EKS Cluster to be updated. Note that the `update` timeout is used separately for both `version` and `vpc_config` update timeouts.
- `delete` - (Default `15 minutes`) How long to wait for the EKS Cluster to be deleted.

» Import

EKS Clusters can be imported using the `name`, e.g.

```
$ terraform import aws_eks_cluster.my_cluster my_cluster
```

» Resource: aws_eks_fargate_profile

Manages an EKS Fargate Profile.

» Example Usage

```
resource "aws_eks_fargate_profile" "example" {
  cluster_name           = aws_eks_cluster.example.name
  fargate_profile_name   = "example"
  pod_execution_role_arn = aws_iam_role.example.arn
  subnet_ids             = aws_subnet.example[*].id

  selector {
    namespace = "example"
  }
}
```

» Example IAM Role for EKS Fargate Profile

```
resource "aws_iam_role" "example" {
  name = "eks-fargate-profile-example"

  assume_role_policy = jsonencode({
    Statement = [{
      Action    = "sts:AssumeRole"
      Effect    = "Allow"
      Principal = {
        Service = "eks-fargate-pods.amazonaws.com"
      }
    }]
    Version = "2012-10-17"
  })
}

resource "aws_iam_role_policy_attachment" "example-AmazonEKSFargatePodExecutionRolePolicy"
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSFargatePodExecutionRolePolicy"
  role       = aws_iam_role.example.name
}
```

» Argument Reference

The following arguments are required:

- `cluster_name` – (Required) Name of the EKS Cluster.
- `fargate_profile_name` – (Required) Name of the EKS Fargate Profile.
- `pod_execution_role_arn` – (Required) Amazon Resource Name (ARN) of the IAM Role that provides permissions for the EKS Fargate Profile.
- `selector` - (Required) Configuration block(s) for selecting Kubernetes Pods to execute with this EKS Fargate Profile. Detailed below.
- `subnet_ids` – (Required) Identifiers of private EC2 Subnets to associate with the EKS Fargate Profile. These subnets must have the following resource tag: `kubernetes.io/cluster/CLUSTER_NAME` (where `CLUSTER_NAME` is replaced with the name of the EKS Cluster).

The following arguments are optional:

- `tags` - (Optional) Key-value mapping of resource tags.

» selector Configuration Block

The following arguments are required:

- `namespace` - (Required) Kubernetes namespace for selection.

The following arguments are optional:

- `labels` - (Optional) Key-value mapping of Kubernetes labels for selection.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the EKS Fargate Profile.
- `id` - EKS Cluster name and EKS Fargate Profile name separated by a colon (`:`).
- `status` - Status of the EKS Fargate Profile.

» Timeouts

`aws_eks_fargate_profile` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) How long to wait for the EKS Fargate Profile to be created.
- `delete` - (Default `10 minutes`) How long to wait for the EKS Fargate Profile to be deleted.

» Import

EKS Fargate Profiles can be imported using the `cluster_name` and `fargate_profile_name` separated by a colon (`:`), e.g.

```
$ terraform import aws_eks_fargate_profile.my_fargate_profile my_cluster:my_fargate_profile
```

» Resource: aws_eks_node_group

Manages an EKS Node Group, which can provision and optionally update an Auto Scaling Group of Kubernetes worker nodes compatible with EKS. Additional documentation about this functionality can be found in the EKS User Guide.

» Example Usage

```
resource "aws_eks_node_group" "example" {
  cluster_name    = aws_eks_cluster.example.name
  node_group_name = "example"
  node_role_arn   = aws_iam_role.example.arn
  subnet_ids      = aws_subnet.example[*].id

  scaling_config {
    desired_size = 1
    max_size     = 1
    min_size     = 1
  }

  # Ensure that IAM Role permissions are created before and deleted after EKS Node Group han
  # Otherwise, EKS will not be able to properly delete EC2 Instances and Elastic Network Int
  depends_on = [
    aws_iam_role_policy_attachment.example-AmazonEKSWorkerNodePolicy,
    aws_iam_role_policy_attachment.example-AmazonEKS_CNI_Policy,
    aws_iam_role_policy_attachment.example-AmazonEC2ContainerRegistryReadOnly,
  ]
}
```

» Example IAM Role for EKS Node Group

```
resource "aws_iam_role" "example" {
  name = "eks-node-group-example"

  assume_role_policy = jsonencode({
    Statement = [{
      Action    = "sts:AssumeRole"
      Effect    = "Allow"
      Principal = {
        Service = "ec2.amazonaws.com"
      }
    }]
    Version = "2012-10-17"
  })
}

resource "aws_iam_role_policy_attachment" "example-AmazonEKSWorkerNodePolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
  role       = aws_iam_role.example.name
}

resource "aws_iam_role_policy_attachment" "example-AmazonEKS_CNI_Policy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
  role       = aws_iam_role.example.name
}

resource "aws_iam_role_policy_attachment" "example-AmazonEC2ContainerRegistryReadOnly" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"
  role       = aws_iam_role.example.name
}
```

» Example Subnets for EKS Node Group

```
data "aws_availability_zones" "available" {
  state = "available"
}

resource "aws_subnet" "example" {
  count = 2

  availability_zone = data.aws_availability_zones.available.names[count.index]
  cidr_block        = cidrsubnet(aws_vpc.example.cidr_block, 8, count.index)
  vpc_id            = aws_vpc.example.id

  tags = {
    "kubernetes.io/cluster/${aws_eks_cluster.example.name}" = "shared"
  }
}
```

» Argument Reference

The following arguments are required:

- `cluster_name` – (Required) Name of the EKS Cluster.
- `node_group_name` – (Required) Name of the EKS Node Group.
- `node_role_arn` – (Required) Amazon Resource Name (ARN) of the IAM Role that provides permissions for the EKS Node Group.
- `scaling_config` - (Required) Configuration block with scaling settings. Detailed below.
- `subnet_ids` – (Required) Identifiers of EC2 Subnets to associate with the EKS Node Group. These subnets must have the following resource tag: `kubernetes.io/cluster/CLUSTER_NAME` (where `CLUSTER_NAME` is replaced with the name of the EKS Cluster).

The following arguments are optional:

- `ami_type` - (Optional) Type of Amazon Machine Image (AMI) associated with the EKS Node Group. Defaults to `AL2_x86_64`. Valid values: `AL2_x86_64`, `AL2_x86_64_GPU`. Terraform will only perform drift detection if a configuration value is provided.
- `disk_size` - (Optional) Disk size in GiB for worker nodes. Defaults to `20`. Terraform will only perform drift detection if a configuration value is provided.
- `instance_types` - (Optional) Set of instance types associated with the EKS Node Group. Defaults to `["t3.medium"]`. Terraform will only perform drift detection if a configuration value is provided. Currently, the EKS API only accepts a single value in the set.
- `labels` - (Optional) Key-value mapping of Kubernetes labels. Only labels that are applied with the EKS API are managed by this argument. Other Kubernetes labels applied to the EKS Node Group will not be managed.
- `release_version` – (Optional) AMI version of the EKS Node Group. Defaults to latest version for Kubernetes version.
- `remote_access` - (Optional) Configuration block with remote access settings. Detailed below.
- `tags` - (Optional) Key-value mapping of resource tags.
- `version` – (Optional) Kubernetes version. Defaults to EKS Cluster Kubernetes version. Terraform will only perform drift detection if a configuration value is provided.

» remote_access Configuration Block

- `ec2_ssh_key` - (Optional) EC2 Key Pair name that provides access for SSH communication with the worker nodes in the EKS Node Group. If you specify this configuration, but do not specify

source_security_group_ids when you create an EKS Node Group, port 22 on the worker nodes is opened to the Internet (0.0.0.0/0).
- source_security_group_ids - (Optional) Set of EC2 Security Group IDs to allow SSH access (port 22) from on the worker nodes. If you specify ec2_ssh_key, but do not specify this configuration when you create an EKS Node Group, port 22 on the worker nodes is opened to the Internet (0.0.0.0/0).

» scaling_config Configuration Block

- desired_size - (Required) Desired number of worker nodes.
- max_size - (Required) Maximum number of worker nodes.
- min_size - (Required) Minimum number of worker nodes.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- arn - Amazon Resource Name (ARN) of the EKS Node Group.
- id - EKS Cluster name and EKS Node Group name separated by a colon (:).
- resources - List of objects containing information about underlying resources.
  - autoscaling_groups - List of objects containing information about AutoScaling Groups.
  - name - Name of the AutoScaling Group.
  - remote_access_security_group_id - Identifier of the remote access EC2 Security Group.
- status - Status of the EKS Node Group.

» Timeouts

aws_eks_node_group provides the following Timeouts configuration options:

- create - (Default 60 minutes) How long to wait for the EKS Node Group to be created.
- update - (Default 60 minutes) How long to wait for the EKS Node Group to be updated. Note that the update timeout is used separately for both configuration and version update operations.
- delete - (Default 60 minutes) How long to wait for the EKS Node Group to be deleted.

» Import

EKS Node Groups can be imported using the `cluster_name` and `node_group_name` separated by a colon (`:`), e.g.

```
$ terraform import aws_eks_node_group.my_node_group my_cluster:my_node_group
```

» Data Source: aws_elasticache_cluster

Use this data source to get information about an Elasticache Cluster

» Example Usage

```
data "aws_elasticache_cluster" "my_cluster" {
  cluster_id = "my-cluster-id"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_id` – (Required) Group identifier.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `node_type` – The cluster node type.
- `num_cache_nodes` – The number of cache nodes that the cache cluster has.
- `engine` – Name of the cache engine.
- `engine_version` – Version number of the cache engine.
- `subnet_group_name` – Name of the subnet group associated to the cache cluster.
- `security_group_names` – List of security group names associated with this cache cluster.
- `security_group_ids` – List VPC security groups associated with the cache cluster.
- `parameter_group_name` – Name of the parameter group associated with this cache cluster.
- `replication_group_id` - The replication group to which this cache cluster belongs.

- `maintenance_window` – Specifies the weekly time range for when maintenance on the cache cluster is performed.
- `snapshot_window` - The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of the cache cluster.
- `snapshot_retention_limit` - The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them.
- `availability_zone` - The Availability Zone for the cache cluster.
- `notification_topic_arn` – An Amazon Resource Name (ARN) of an SNS topic that ElastiCache notifications get sent to.
- `port` – The port number on which each of the cache nodes will accept connections.
- `configuration_endpoint` - (Memcached only) The configuration endpoint to allow host discovery.
- `cluster_address` - (Memcached only) The DNS name of the cache cluster without the port appended.
- `cache_nodes` - List of node objects including `id`, `address`, `port` and `availability_zone`. Referenceable e.g. as `${data.aws_elasticache_cluster.bar.cache_nodes.0.ad`
- `tags` - The tags assigned to the resource

» Data Source: aws_elasticache_replication_group

Use this data source to get information about an Elasticache Replication Group.

» Example Usage

```
data "aws_elasticache_replication_group" "bar" {
  replication_group_id = "example"
}
```

» Argument Reference

The following arguments are supported:

- `replication_group_id` – (Required) The identifier for the replication group.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `replication_group_id` - The identifier for the replication group.

- `replication_group_description` - The description of the replication group.
- `auth_token_enabled` - A flag that enables using an AuthToken (password) when issuing Redis commands.
- `automatic_failover_enabled` - A flag whether a read-only replica will be automatically promoted to read/write primary if the existing primary fails.
- `node_type` – The cluster node type.
- `number_cache_clusters` – The number of cache clusters that the replication group has.
- `member_clusters` - The identifiers of all the nodes that are part of this replication group.
- `snapshot_window` - The daily time range (in UTC) during which ElastiCache begins taking a daily snapshot of your node group (shard).
- `snapshot_retention_limit` - The number of days for which ElastiCache retains automatic cache cluster snapshots before deleting them.
- `port` – The port number on which the configuration endpoint will accept connections.
- `configuration_endpoint_address` - The configuration endpoint address to allow host discovery.
- `primary_endpoint_address` - The endpoint of the primary node in this node group (shard).

» Resource: aws_elasticache_cluster

Provides an ElastiCache Cluster resource, which manages a Memcached cluster or Redis instance. For working with Redis (Cluster Mode Enabled) replication groups, see the `aws_elasticache_replication_group` resource.

Note: When you change an attribute, such as `node_type`, by default it is applied in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because the actual modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately. Using `apply_immediately` can result in a brief downtime as the server reboots. See the AWS Docs on Modifying an ElastiCache Cache Cluster for more information.

» Example Usage

» Memcached Cluster

```
resource "aws_elasticache_cluster" "example" {
  cluster_id           = "cluster-example"
  engine               = "memcached"
  node_type            = "cache.m4.large"
  num_cache_nodes      = 2
  parameter_group_name = "default.memcached1.4"
  port                 = 11211
}
```

» Redis Instance

```
resource "aws_elasticache_cluster" "example" {
  cluster_id           = "cluster-example"
  engine               = "redis"
  node_type            = "cache.m4.large"
  num_cache_nodes      = 1
  parameter_group_name = "default.redis3.2"
  engine_version       = "3.2.10"
  port                 = 6379
}
```

» Redis Cluster Mode Disabled Read Replica Instance

These inherit their settings from the replication group.

```
resource "aws_elasticache_cluster" "replica" {
  cluster_id           = "cluster-example"
  replication_group_id = "${aws_elasticache_replication_group.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_id` – (Required) Group identifier. ElastiCache converts this name to lowercase
- `replication_group_id` - (Optional) The ID of the replication group to which this cluster should belong. If this parameter is specified, the cluster is added to the specified replication group as a read replica; otherwise, the cluster is a standalone primary that is not part of any replication group.
- `engine` – (Required unless `replication_group_id` is provided) Name of the cache engine to be used for this cache cluster. Valid values for this parameter are `memcached` or `redis`
- `engine_version` – (Optional) Version number of the cache engine to be used. See Describe Cache Engine Versions in the AWS Documentation center for supported versions

- `maintenance_window` – (Optional) Specifies the weekly time range for when maintenance on the cache cluster is performed. The format is `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Example: `sun:05:00-sun:09:00`
- `node_type` – (Required unless `replication_group_id` is provided) The compute and memory capacity of the nodes. See Available Cache Node Types for supported node types
- `num_cache_nodes` – (Required unless `replication_group_id` is provided) The initial number of cache nodes that the cache cluster will have. For Redis, this value must be 1. For Memcache, this value must be between 1 and 20. If this number is reduced on subsequent runs, the highest numbered nodes will be removed.
- `parameter_group_name` – (Required unless `replication_group_id` is provided) Name of the parameter group to associate with this cache cluster
- `port` – (Optional) The port number on which each of the cache nodes will accept connections. For Memcache the default is 11211, and for Redis the default port is 6379. Cannot be provided with `replication_group_id`.
- `subnet_group_name` – (Optional, VPC only) Name of the subnet group to be used for the cache cluster.
- `security_group_names` – (Optional, EC2 Classic only) List of security group names to associate with this cache cluster
- `security_group_ids` – (Optional, VPC only) One or more VPC security groups associated with the cache cluster
- `apply_immediately` - (Optional) Specifies whether any database modifications are applied immediately, or during the next maintenance window. Default is `false`. See Amazon ElastiCache Documentation for more information. (Available since v0.6.0)
- `snapshot_arns` – (Optional) A single-element string list containing an Amazon Resource Name (ARN) of a Redis RDB snapshot file stored in Amazon S3. Example: `arn:aws:s3:::my_bucket/snapshot1.rdb`
- `snapshot_name` - (Optional) The name of a snapshot from which to restore data into the new node group. Changing the `snapshot_name` forces a new resource.
- `snapshot_window` - (Optional, Redis only) The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster. Example: 05:00-09:00
- `snapshot_retention_limit` - (Optional, Redis only) The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set SnapshotRetentionLimit to 5, then

a snapshot that was taken today will be retained for 5 days before being deleted. If the value of SnapshotRetentionLimit is set to zero (0), backups are turned off. Please note that setting a `snapshot_retention_limit` is not supported on cache.t1.micro or cache.t2.* cache nodes

- `notification_topic_arn` – (Optional) An Amazon Resource Name (ARN) of an SNS topic to send ElastiCache notifications to. Example: `arn:aws:sns:us-east-1:012345678999:my_sns_topic`
- `az_mode` - (Optional, Memcached only) Specifies whether the nodes in this Memcached node group are created in a single Availability Zone or created across multiple Availability Zones in the cluster's region. Valid values for this parameter are `single-az` or `cross-az`, default is `single-az`. If you want to choose `cross-az`, `num_cache_nodes` must be greater than `1`
- `availability_zone` - (Optional) The Availability Zone for the cache cluster. If you want to create cache nodes in multi-az, use `preferred_availability_zones` instead. Default: System chosen Availability Zone.
- `preferred_availability_zones` - (Optional, Memcached only) A list of the Availability Zones in which cache nodes are created. If you are creating your cluster in an Amazon VPC you can only locate nodes in Availability Zones that are associated with the subnets in the selected subnet group. The number of Availability Zones listed must equal the value of `num_cache_nodes`. If you want all the nodes in the same Availability Zone, use `availability_zone` instead, or repeat the Availability Zone multiple times in the list. Default: System chosen Availability Zones. Detecting drift of existing node availability zone is not currently supported. Updating this argument by itself to migrate existing node availability zones is not currently supported and will show a perpetual difference.
- `tags` - (Optional) A mapping of tags to assign to the resource

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `cache_nodes` - List of node objects including `id`, `address`, `port` and `availability_zone`. Referenceable e.g. as `${aws_elasticache_cluster.bar.cache_nodes.0.address`
- `configuration_endpoint` - (Memcached only) The configuration endpoint to allow host discovery.
- `cluster_address` - (Memcached only) The DNS name of the cache cluster without the port appended.

» Import

ElastiCache Clusters can be imported using the `cluster_id`, e.g.

```
$ terraform import aws_elasticache_cluster.my_cluster my_cluster
```

» Resource: aws_elasticache_parameter_group

Provides an ElastiCache parameter group resource.

NOTE: Attempting to remove the `reserved-memory` parameter when `family` is set to `redis2.6` or `redis2.8` may show a perpetual difference in Terraform due to an Elasticache API limitation. Leave that parameter configured with any value to workaround the issue.

» Example Usage

```
resource "aws_elasticache_parameter_group" "default" {
  name   = "cache-params"
  family = "redis2.8"

  parameter {
    name  = "activerehashing"
    value = "yes"
  }

  parameter {
    name  = "min-slaves-to-write"
    value = "2"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ElastiCache parameter group.
- `family` - (Required) The family of the ElastiCache parameter group.
- `description` - (Optional) The description of the ElastiCache parameter group. Defaults to "Managed by Terraform".
- `parameter` - (Optional) A list of ElastiCache parameters to apply.

Parameter blocks support the following:

- `name` - (Required) The name of the ElastiCache parameter.

- `value` - (Required) The value of the ElastiCache parameter.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ElastiCache parameter group name.

» Import

ElastiCache Parameter Groups can be imported using the `name`, e.g.

```
$ terraform import aws_elasticache_parameter_group.default redis-params
```

» Resource: aws_elasticache_replication_group

Provides an ElastiCache Replication Group resource. For working with Memcached or single primary Redis instances (Cluster Mode Disabled), see the `aws_elasticache_cluster` resource.

Note: When you change an attribute, such as `engine_version`, by default the ElastiCache API applies it in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because the actual modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately. Using `apply_immediately` can result in a brief downtime as servers reboots.

» Example Usage

» Redis Cluster Mode Disabled

To create a single shard primary with single read replica:

```
resource "aws_elasticache_replication_group" "example" {
  automatic_failover_enabled    = true
  availability_zones            = ["us-west-2a", "us-west-2b"]
  replication_group_id          = "tf-rep-group-1"
  replication_group_description = "test description"
  node_type                     = "cache.m4.large"
  number_cache_clusters         = 2
  parameter_group_name          = "default.redis3.2"
  port                          = 6379
}
```

You have two options for adjusting the number of replicas:

- Adjusting `number_cache_clusters` directly. This will attempt to automatically add or remove replicas, but provides no granular control (e.g. preferred availability zone, cache cluster ID) for the added or removed replicas. This also currently expects cache cluster IDs in the form of `replication_group_id-00#`.
- Otherwise for fine grained control of the underlying cache clusters, they can be added or removed with the `aws_elasticache_cluster` resource and its `replication_group_id` attribute. In this situation, you will need to utilize the lifecycle configuration block with `ignore_changes` to prevent perpetual differences during Terraform plan with the `number_cache_cluster` attribute.

```
resource "aws_elasticache_replication_group" "example" {
  automatic_failover_enabled    = true
  availability_zones            = ["us-west-2a", "us-west-2b"]
  replication_group_id          = "tf-rep-group-1"
  replication_group_description = "test description"
  node_type                     = "cache.m4.large"
  number_cache_clusters         = 2
  parameter_group_name          = "default.redis3.2"
  port                          = 6379

  lifecycle {
    ignore_changes = ["number_cache_clusters"]
  }
}

resource "aws_elasticache_cluster" "replica" {
  count = 1

  cluster_id           = "tf-rep-group-1-${count.index}"
  replication_group_id = "${aws_elasticache_replication_group.example.id}"
}
```

» Redis Cluster Mode Enabled

To create two shards with a primary and a single read replica each:

```
resource "aws_elasticache_replication_group" "baz" {
  replication_group_id          = "tf-redis-cluster"
  replication_group_description = "test description"
  node_type                     = "cache.t2.small"
  port                          = 6379
  parameter_group_name          = "default.redis3.2.cluster.on"
  automatic_failover_enabled    = true

  cluster_mode {
    replicas_per_node_group = 1
    num_node_groups         = 2
  }
}
```

Note: We currently do not support passing a `primary_cluster_id` in order to create the Replication Group.

Note: Automatic Failover is unavailable for Redis versions earlier than 2.8.6, and unavailable on T1 node types. For T2 node types, it is only available on Redis version 3.2.4 or later with cluster mode enabled. See the High Availability Using Replication Groups guide for full details on using Replication Groups.

» Argument Reference

The following arguments are supported:

- `replication_group_id` – (Required) The replication group identifier. This parameter is stored as a lowercase string.
- `replication_group_description` – (Required) A user-created description for the replication group.
- `number_cache_clusters` - (Required for Cluster Mode Disabled) The number of cache clusters (primary and replicas) this replication group will have. If Multi-AZ is enabled, the value of this parameter must be at least 2. Updates will occur before other modifications.
- `node_type` - (Required) The compute and memory capacity of the nodes in the node group.
- `automatic_failover_enabled` - (Optional) Specifies whether a read-only replica will be automatically promoted to read/write primary if the existing primary fails. If true, Multi-AZ is enabled for this replication group. If false, Multi-AZ is disabled for this replication group. Must be enabled for Redis (cluster mode enabled) replication groups. Defaults to `false`.
- `auto_minor_version_upgrade` - (Optional) Specifies whether a minor engine upgrades will be applied automatically to the underlying Cache Cluster instances during the maintenance window. Defaults to `true`.
- `availability_zones` - (Optional) A list of EC2 availability zones in which the replication group's cache clusters will be created. The order of the availability zones in the list is not important.
- `engine` - (Optional) The name of the cache engine to be used for the clusters in this replication group. e.g. `redis`
- `at_rest_encryption_enabled` - (Optional) Whether to enable encryption at rest.

- `transit_encryption_enabled` - (Optional) Whether to enable encryption in transit.
- `auth_token` - (Optional) The password used to access a password protected server. Can be specified only if `transit_encryption_enabled = true`.
- `kms_key_id` - (Optional) The ARN of the key that you wish to use if encrypting at rest. If not supplied, uses service managed encryption. Can be specified only if `at_rest_encryption_enabled = true`.
- `engine_version` - (Optional) The version number of the cache engine to be used for the cache clusters in this replication group.
- `parameter_group_name` - (Optional) The name of the parameter group to associate with this replication group. If this argument is omitted, the default cache parameter group for the specified engine is used.
- `port` – (Optional) The port number on which each of the cache nodes will accept connections. For Memcache the default is 11211, and for Redis the default port is 6379.
- `subnet_group_name` - (Optional) The name of the cache subnet group to be used for the replication group.
- `security_group_names` - (Optional) A list of cache security group names to associate with this replication group.
- `security_group_ids` - (Optional) One or more Amazon VPC security groups associated with this replication group. Use this parameter only when you are creating a replication group in an Amazon Virtual Private Cloud
- `snapshot_arns` – (Optional) A single-element string list containing an Amazon Resource Name (ARN) of a Redis RDB snapshot file stored in Amazon S3. Example: `arn:aws:s3:::my_bucket/snapshot1.rdb`
- `snapshot_name` - (Optional) The name of a snapshot from which to restore data into the new node group. Changing the `snapshot_name` forces a new resource.
- `maintenance_window` – (Optional) Specifies the weekly time range for when maintenance on the cache cluster is performed. The format is `ddd:hh24:mi-ddd:hh24:mi` (24H Clock UTC). The minimum maintenance window is a 60 minute period. Example: `sun:05:00-sun:09:00`
- `notification_topic_arn` – (Optional) An Amazon Resource Name (ARN) of an SNS topic to send ElastiCache notifications to. Example: `arn:aws:sns:us-east-1:012345678999:my_sns_topic`
- `snapshot_window` - (Optional, Redis only) The daily time range (in UTC) during which ElastiCache will begin taking a daily snapshot of your cache cluster. The minimum snapshot window is a 60 minute period. Example: `05:00-09:00`
- `snapshot_retention_limit` - (Optional, Redis only) The number of days for which ElastiCache will retain automatic cache cluster snapshots before deleting them. For example, if you set SnapshotRetentionLimit to 5, then a snapshot that was taken today will be retained for 5 days before being deleted. If the value of SnapshotRetentionLimit is set to zero (0), backups

are turned off. Please note that setting a `snapshot_retention_limit` is not supported on cache.t1.micro or cache.t2.* cache nodes
- `apply_immediately` - (Optional) Specifies whether any modifications are applied immediately, or during the next maintenance window. Default is `false`.
- `tags` - (Optional) A mapping of tags to assign to the resource
- `cluster_mode` - (Optional) Create a native redis cluster. `automatic_failover_enabled` must be set to true. Cluster Mode documented below. Only 1 `cluster_mode` block is allowed.

Cluster Mode (`cluster_mode`) supports the following:

- `replicas_per_node_group` - (Required) Specify the number of replica nodes in each node group. Valid values are 0 to 5. Changing this number will force a new resource.
- `num_node_groups` - (Required) Specify the number of node groups (shards) for this Redis replication group. Changing this number will trigger an online resizing operation before other settings modifications.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the ElastiCache Replication Group.
- `configuration_endpoint_address` - The address of the replication group configuration endpoint when cluster mode is enabled.
- `primary_endpoint_address` - (Redis only) The address of the endpoint for the primary node in the replication group, if the cluster mode is disabled.
- `member_clusters` - The identifiers of all the nodes that are part of this replication group.

» Timeouts

`aws_elasticache_replication_group` provides the following Timeouts configuration options:

- `create` - (Default `60m`) How long to wait for a replication group to be created.
- `delete` - (Default `40m`) How long to wait for a replication group to be deleted.
- `update` - (Default `40m`) How long to wait for replication group settings to be updated. This is also separately used for adding/removing replicas and online resize operation completion, if necessary.

» Import

ElastiCache Replication Groups can be imported using the `replication_group_id`, e.g.

```
$ terraform import aws_elasticache_replication_group.my_replication_group replication-group-
```

» Resource: aws_elasticache_security_group

Provides an ElastiCache Security Group to control access to one or more cache clusters.

NOTE: ElastiCache Security Groups are for use only when working with an ElastiCache cluster **outside** of a VPC. If you are using a VPC, see the ElastiCache Subnet Group resource.

» Example Usage

```
resource "aws_security_group" "bar" {
  name = "security-group"
}

resource "aws_elasticache_security_group" "bar" {
  name                 = "elasticache-security-group"
  security_group_names = ["${aws_security_group.bar.name}"]
}
```

» Argument Reference

The following arguments are supported:

- `name` – (Required) Name for the cache security group. This value is stored as a lowercase string.
- `description` – (Optional) description for the cache security group. Defaults to "Managed by Terraform".
- `security_group_names` – (Required) List of EC2 security group names to be authorized for ingress to the cache security group

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `description`

- name
- security_group_names

» Import

ElastiCache Security Groups can be imported by name, e.g.

```
$ terraform import aws_elasticache_security_group.my_ec_security_group ec-security-group-1
```

» Resource: aws_elasticache_subnet_group

Provides an ElastiCache Subnet Group resource.

NOTE: ElastiCache Subnet Groups are only for use when working with an ElastiCache cluster **inside** of a VPC. If you are on EC2 Classic, see the ElastiCache Security Group resource.

» Example Usage

```
resource "aws_vpc" "foo" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "tf-test"
  }
}

resource "aws_subnet" "foo" {
  vpc_id            = "${aws_vpc.foo.id}"
  cidr_block        = "10.0.0.0/24"
  availability_zone = "us-west-2a"

  tags = {
    Name = "tf-test"
  }
}

resource "aws_elasticache_subnet_group" "bar" {
  name       = "tf-test-cache-subnet"
  subnet_ids = ["${aws_subnet.foo.id}"]
}
```

» Argument Reference

The following arguments are supported:

- `name` – (Required) Name for the cache subnet group. Elasticache converts this name to lowercase.
- `description` – (Optional) Description for the cache subnet group. Defaults to "Managed by Terraform".
- `subnet_ids` – (Required) List of VPC Subnet IDs for the cache subnet group

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `description`
- `name`
- `subnet_ids`

» Import

ElastiCache Subnet Groups can be imported using the `name`, e.g.

```
$ terraform import aws_elasticache_subnet_group.bar tf-test-cache-subnet
```

» Data Source: aws_elastic_beanstalk_application

Retrieve information about an Elastic Beanstalk Application.

» Example Usage

```
data "aws_elastic_beanstalk_application" "example" {
  name = "example"
}

output "arn" {
  value = "${data.aws_elastic_beanstalk_application.example.arn}"
}

output "description" {
  value = "${data.aws_elastic_beanstalk_application.example.description}"
}
```

» Argument Reference

- `name` - (Required) The name of the application

» Attributes Reference

- `id` - The name of the application
- `arn` - The Amazon Resource Name (ARN) of the application.
- `description` - Short description of the application

Application version lifecycle (`appversion_lifecycle`) supports the nested attribute containing.

- `service_role` - The ARN of an IAM service role under which the application version is deleted. Elastic Beanstalk must have permission to assume this role.
- `max_count` - The maximum number of application versions to retain.
- `max_age_in_days` - The number of days to retain an application version.
- `delete_source_from_s3` - Specifies whether delete a version's source bundle from S3 when the application version is deleted.

» Data Source: aws_elastic_beanstalk_hosted_zone

Use this data source to get the ID of an elastic beanstalk hosted zone.

» Example Usage

```
data "aws_elastic_beanstalk_hosted_zone" "current" {}
```

» Argument Reference

- `region` - (Optional) The region you'd like the zone for. By default, fetches the current region.

» Attributes Reference

- `id` - The ID of the hosted zone.
- `region` - The region of the hosted zone.

» Data Source: aws_elastic_beanstalk_solution_stack

Use this data source to get the name of a elastic beanstalk solution stack.

» Example Usage

```
data "aws_elastic_beanstalk_solution_stack" "multi_docker" {
  most_recent = true

  name_regex = "^64bit Amazon Linux (.*) Multi-container Docker (.*)$"
}
```

» Argument Reference

- `most_recent` - (Optional) If more than one result is returned, use the most recent solution stack.
- `name_regex` - A regex string to apply to the solution stack list returned by AWS. See Elastic Beanstalk Supported Platforms from AWS documentation for reference solution stack names.

NOTE: If more or less than a single match is returned by the search, Terraform will fail. Ensure that your search is specific enough to return a single solution stack, or use `most_recent` to choose the most recent one.

» Attributes Reference

- `name` - The name of the solution stack.

» Resource: aws_elastic_beanstalk_application

Provides an Elastic Beanstalk Application Resource. Elastic Beanstalk allows you to deploy and manage applications in the AWS cloud without worrying about the infrastructure that runs those applications.

This resource creates an application that has one configuration template named `default`, and no application versions

» Example Usage

```
resource "aws_elastic_beanstalk_application" "tftest" {
  name        = "tf-test-name"
  description = "tf-test-desc"

  appversion_lifecycle {
    service_role          = "${aws_iam_role.beanstalk_service.arn}"
    max_count             = 128
    delete_source_from_s3 = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the application, must be unique within your account
- `description` - (Optional) Short description of the application
- `tags` - (Optional) Key-value mapping of tags for the Elastic Beanstalk Application.

Application version lifecycle (`appversion_lifecycle`) supports the following settings. Only one of either `max_count` or `max_age_in_days` can be provided:

- `service_role` - (Required) The ARN of an IAM service role under which the application version is deleted. Elastic Beanstalk must have permission to assume this role.
- `max_count` - (Optional) The maximum number of application versions to retain.
- `max_age_in_days` - (Optional) The number of days to retain an application version.
- `delete_source_from_s3` - (Optional) Set to `true` to delete a version's source bundle from S3 when the application version is deleted.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS for this Elastic Beanstalk Application.

» Import

Elastic Beanstalk Applications can be imported using the `name`, e.g.

```
$ terraform import aws_elastic_beanstalk_application.tf_test tf-test-name
```

» Resource: aws_elastic_beanstalk_application_version

Provides an Elastic Beanstalk Application Version Resource. Elastic Beanstalk allows you to deploy and manage applications in the AWS cloud without worrying about the infrastructure that runs those applications.

This resource creates a Beanstalk Application Version that can be deployed to a Beanstalk Environment.

NOTE on Application Version Resource: When using the Application Version resource with multiple Elastic Beanstalk Environments it is possible that an error may be returned when attempting to delete an Application Version while it is still in use by a different environment. To work around this you can:

1. Create each environment in a separate AWS account
2. Create your `aws_elastic_beanstalk_application_version` resources with a unique names in your Elastic Beanstalk Application. For example <revision>-<environment>.

» Example Usage

```
resource "aws_s3_bucket" "default" {
  bucket = "tftest.applicationversion.bucket"
}

resource "aws_s3_bucket_object" "default" {
  bucket = "${aws_s3_bucket.default.id}"
  key    = "beanstalk/go-v1.zip"
  source = "go-v1.zip"
}

resource "aws_elastic_beanstalk_application" "default" {
  name        = "tf-test-name"
  description = "tf-test-desc"
}

resource "aws_elastic_beanstalk_application_version" "default" {
  name        = "tf-test-version-label"
  application = "tf-test-name"
  description = "application version created by terraform"
  bucket      = "${aws_s3_bucket.default.id}"
  key         = "${aws_s3_bucket_object.default.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the this Application Version.
- `application` - (Required) Name of the Beanstalk Application the version is associated with.
- `description` - (Optional) Short description of the Application Version.
- `bucket` - (Required) S3 bucket that contains the Application Version source bundle.
- `key` - (Required) S3 object that is the Application Version source bundle.
- `force_delete` - (Optional) On delete, force an Application Version to be deleted when it may be in use by multiple Elastic Beanstalk Environments.
- `tags` - Key-value mapping of tags for the Elastic Beanstalk Application Version.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS for this Elastic Beanstalk Application.

» Resource: aws_elastic_beanstalk_configuration_template

Provides an Elastic Beanstalk Configuration Template, which are associated with a specific application and are used to deploy different versions of the application with the same configuration settings.

» Example Usage

```
resource "aws_elastic_beanstalk_application" "tftest" {
  name        = "tf-test-name"
  description = "tf-test-desc"
}

resource "aws_elastic_beanstalk_configuration_template" "tf_template" {
  name                = "tf-test-template-config"
  application         = "${aws_elastic_beanstalk_application.tftest.name}"
  solution_stack_name = "64bit Amazon Linux 2015.09 v2.0.8 running Go 1.4"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for this Template.
- `application` – (Required) name of the application to associate with this configuration template
- `description` - (Optional) Short description of the Template
- `environment_id` – (Optional) The ID of the environment used with this configuration template
- `setting` – (Optional) Option settings to configure the new Environment. These override specific values that are set as defaults. The format is detailed below in Option Settings
- `solution_stack_name` – (Optional) A solution stack to base your Template off of. Example stacks can be found in the Amazon API documentation

» Option Settings

The `setting` field supports the following format:

- `namespace` - unique namespace identifying the option's associated AWS resource
- `name` - name of the configuration option
- `value` - value for the configuration option
- `resource` - (Optional) resource name for scheduled action

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `name`
- `application`
- `description`
- `environment_id`
- `option_settings`
- `solution_stack_name`

» Resource: aws_elastic_beanstalk_environment

Provides an Elastic Beanstalk Environment Resource. Elastic Beanstalk allows you to deploy and manage applications in the AWS cloud without worrying about the infrastructure that runs those applications.

Environments are often things such as `development`, `integration`, or `production`.

» Example Usage

```
resource "aws_elastic_beanstalk_application" "tftest" {
  name        = "tf-test-name"
  description = "tf-test-desc"
}

resource "aws_elastic_beanstalk_environment" "tfenvtest" {
  name                = "tf-test-name"
  application         = "${aws_elastic_beanstalk_application.tftest.name}"
  solution_stack_name = "64bit Amazon Linux 2015.03 v2.0.3 running Go 1.4"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for this Environment. This name is used in the application URL
- `application` – (Required) Name of the application that contains the version to be deployed
- `cname_prefix` - (Optional) Prefix to use for the fully qualified DNS name of the Environment.
- `description` - (Optional) Short description of the Environment
- `tier` - (Optional) Elastic Beanstalk Environment tier. Valid values are `Worker` or `WebServer`. If tier is left blank `WebServer` will be used.
- `setting` – (Optional) Option settings to configure the new Environment. These override specific values that are set as defaults. The format is detailed below in Option Settings
- `solution_stack_name` – (Optional) A solution stack to base your environment off of. Example stacks can be found in the Amazon API documentation
- `template_name` – (Optional) The name of the Elastic Beanstalk Configuration template to use in deployment
- `platform_arn` – (Optional) The ARN of the Elastic Beanstalk Platform to use in deployment
- `wait_for_ready_timeout` - (Default: `20m`) The maximum duration that Terraform should wait for an Elastic Beanstalk Environment to be in a ready state before timing out.
- `poll_interval` – The time between polling the AWS API to check if changes have been applied. Use this to adjust the rate of API calls for

any `create` or `update` action. Minimum `10s`, maximum `180s`. Omit this to use the default behavior, which is an exponential backoff

- `version_label` - (Optional) The name of the Elastic Beanstalk Application Version to use in deployment.
- `tags` – (Optional) A set of tags to apply to the Environment.

» Option Settings

Some options can be stack-specific, check AWS Docs for supported options and examples.

The `setting` and `all_settings` mappings support the following format:

- `namespace` - unique namespace identifying the option's associated AWS resource
- `name` - name of the configuration option
- `value` - value for the configuration option
- `resource` - (Optional) resource name for scheduled action

» Example With Options

```
resource "aws_elastic_beanstalk_application" "tftest" {
  name        = "tf-test-name"
  description = "tf-test-desc"
}

resource "aws_elastic_beanstalk_environment" "tfenvtest" {
  name                = "tf-test-name"
  application         = "${aws_elastic_beanstalk_application.tftest.name}"
  solution_stack_name = "64bit Amazon Linux 2015.03 v2.0.3 running Go 1.4"

  setting {
    namespace = "aws:ec2:vpc"
    name      = "VPCId"
    value     = "vpc-xxxxxxxx"
  }

  setting {
    namespace = "aws:ec2:vpc"
    name      = "Subnets"
    value     = "subnet-xxxxxxxx"
  }
}
```

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - ID of the Elastic Beanstalk Environment.
- `name` - Name of the Elastic Beanstalk Environment.
- `description` - Description of the Elastic Beanstalk Environment.
- `tier` - The environment tier specified.
- `application` – The Elastic Beanstalk Application specified for this environment.
- `setting` – Settings specifically set for this Environment.
- `all_settings` – List of all option settings configured in this Environment. These are a combination of default settings and their overrides from `setting` in the configuration.
- `cname` - Fully qualified DNS name for this Environment.
- `autoscaling_groups` - The autoscaling groups used by this Environment.
- `instances` - Instances used by this Environment.
- `launch_configurations` - Launch configurations in use by this Environment.
- `load_balancers` - Elastic load balancers in use by this Environment.
- `queues` - SQS queues in use by this Environment.
- `triggers` - Autoscaling triggers in use by this Environment.
- `endpoint_url` - The URL to the Load Balancer for this Environment

» Import

Elastic Beanstalk Environments can be imported using the `id`, e.g.

```
$ terraform import aws_elastic_beanstalk_environment.prodenv e-rpqsewtp2j
```

» aws_elb

Provides information about a ”classic” Elastic Load Balancer (ELB). See LB Data Source if you are looking for ”v2” Application Load Balancer (ALB) or Network Load Balancer (NLB).

This data source can prove useful when a module accepts an LB as an input variable and needs to, for example, determine the security groups associated with it, etc.

» Example Usage

```
variable "lb_name" {
  type    = "string"
  default = ""
}

data "aws_elb" "test" {
  name = "${var.lb_name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The unique name of the load balancer.

» Attributes Reference

See the ELB Resource for details on the returned attributes - they are identical.

» Data Source: aws_elb_hosted_zone_id

Use this data source to get the HostedZoneId of the AWS Elastic Load Balancing HostedZoneId in a given region for the purpose of using in an AWS Route53 Alias.

» Example Usage

```
data "aws_elb_hosted_zone_id" "main" {}

resource "aws_route53_record" "www" {
  zone_id = "${aws_route53_zone.primary.zone_id}"
  name    = "example.com"
  type    = "A"

  alias {
    name                   = "${aws_elb.main.dns_name}"
    zone_id                = "${data.aws_elb_hosted_zone_id.main.id}"
    evaluate_target_health = true
  }
}
```

» Argument Reference

- `region` - (Optional) Name of the region whose AWS ELB HostedZoneId is desired. Defaults to the region from the AWS provider configuration.

» Attributes Reference

- `id` - The ID of the AWS ELB HostedZoneId in the selected region.

» Data Source: aws_elb_service_account

Use this data source to get the Account ID of the AWS Elastic Load Balancing Service Account in a given region for the purpose of whitelisting in S3 bucket policy.

» Example Usage

```
data "aws_elb_service_account" "main" {}

resource "aws_s3_bucket" "elb_logs" {
  bucket = "my-elb-tf-test-bucket"
  acl    = "private"

  policy = <<POLICY
{
  "Id": "Policy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::my-elb-tf-test-bucket/AWSLogs/*",
      "Principal": {
        "AWS": [
          "${data.aws_elb_service_account.main.arn}"
        ]
      }
    }
  ]
}
POLICY
}

resource "aws_elb" "bar" {
  name               = "my-foobar-terraform-elb"
  availability_zones = ["us-west-2a"]

  access_logs {
    bucket   = "${aws_s3_bucket.elb_logs.bucket}"
    interval = 5
  }

  listener {
    instance_port     = 8000
    instance_protocol = "http"
    lb_port           = 80
    lb_protocol       = "http"
  }
}
```

» Argument Reference

- `region` - (Optional) Name of the region whose AWS ELB account ID is desired. Defaults to the region from the AWS provider configuration.

» Attributes Reference

- `id` - The ID of the AWS ELB service account in the selected region.
- `arn` - The ARN of the AWS ELB service account in the selected region.

» Resource: aws_app_cookie_stickiness_policy

Provides an application cookie stickiness policy, which allows an ELB to wed its sticky cookie's expiration to a cookie generated by your application.

» Example Usage

```
resource "aws_elb" "lb" {
  name               = "test-lb"
  availability_zones = ["us-east-1a"]
  listener {
    instance_port     = 8000
    instance_protocol = "http"
    lb_port           = 80
    lb_protocol       = "http"
  }
}

resource "aws_app_cookie_stickiness_policy" "foo" {
  name          = "foo_policy"
  load_balancer = "${aws_elb.lb.name}"
  lb_port       = 80
  cookie_name   = "MyAppCookie"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the stickiness policy.
- `load_balancer` - (Required) The name of load balancer to which the policy should be attached.
- `lb_port` - (Required) The load balancer port to which the policy should be applied. This must be an active listener on the load balancer.
- `cookie_name` - (Required) The application cookie whose lifetime the ELB's cookie should follow.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the policy.
- `name` - The name of the stickiness policy.
- `load_balancer` - The name of load balancer to which the policy is attached.
- `lb_port` - The load balancer port to which the policy is applied.
- `cookie_name` - The application cookie whose lifetime the ELB's cookie should follow.

» Import

Application cookie stickiness policies can be imported using the ELB name, port, and policy name separated by colons (`:`), e.g.

```
$ terraform import aws_app_cookie_stickiness_policy.example my-elb:80:my-policy
```

» Resource: aws_elb

Provides an Elastic Load Balancer resource, also known as a ”Classic Load Balancer” after the release of Application/Network Load Balancers.

NOTE on ELB Instances and ELB Attachments: Terraform currently provides both a standalone ELB Attachment resource (describing an instance attached to an ELB), and an ELB resource with `instances` defined in-line. At this time you cannot use an ELB with in-line instances in conjunction with a ELB Attachment resources. Doing so will cause a conflict and will overwrite attachments.

» Example Usage

```
# Create a new load balancer
resource "aws_elb" "bar" {
  name               = "foobar-terraform-elb"
  availability_zones = ["us-west-2a", "us-west-2b", "us-west-2c"]

  access_logs {
    bucket        = "foo"
    bucket_prefix = "bar"
    interval      = 60
  }

  listener {
    instance_port     = 8000
    instance_protocol = "http"
    lb_port           = 80
    lb_protocol       = "http"
  }

  listener {
    instance_port      = 8000
    instance_protocol  = "http"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "arn:aws:iam::123456789012:server-certificate/certName"
  }

  health_check {
    healthy_threshold   = 2
    unhealthy_threshold = 2
    timeout             = 3
    target              = "HTTP:8000/"
    interval            = 30
  }

  instances                   = ["${aws_instance.foo.id}"]
  cross_zone_load_balancing   = true
  idle_timeout                = 400
  connection_draining         = true
  connection_draining_timeout = 400

  tags = {
    Name = "foobar-terraform-elb"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the ELB. By default generated by Terraform.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `access_logs` - (Optional) An Access Logs block. Access Logs documented below.
- `availability_zones` - (Required for an EC2-classic ELB) The AZ's to serve traffic in.
- `security_groups` - (Optional) A list of security group IDs to assign to the ELB. Only valid if creating an ELB within a VPC
- `subnets` - (Required for a VPC ELB) A list of subnet IDs to attach to the ELB.
- `instances` - (Optional) A list of instance ids to place in the ELB pool.
- `internal` - (Optional) If true, ELB will be an internal ELB.
- `listener` - (Required) A list of listener blocks. Listeners documented below.
- `health_check` - (Optional) A health_check block. Health Check documented below.
- `cross_zone_load_balancing` - (Optional) Enable cross-zone load balancing. Default: `true`
- `idle_timeout` - (Optional) The time in seconds that the connection is allowed to be idle. Default: `60`
- `connection_draining` - (Optional) Boolean to enable connection draining. Default: `false`

- `connection_draining_timeout` - (Optional) The time in seconds to allow for connections to drain. Default: `300`
- `tags` - (Optional) A mapping of tags to assign to the resource.

Exactly one of `availability_zones` or `subnets` must be specified: this determines if the ELB exists in a VPC or in EC2-classic.

Access Logs (`access_logs`) support the following:

- `bucket` - (Required) The S3 bucket name to store the logs in.
- `bucket_prefix` - (Optional) The S3 bucket prefix. Logs are stored in the root if not configured.
- `interval` - (Optional) The publishing interval in minutes. Default: 60 minutes.
- `enabled` - (Optional) Boolean to enable / disable `access_logs`. Default is `true`

Listeners (`listener`) support the following:

- `instance_port` - (Required) The port on the instance to route to
- `instance_protocol` - (Required) The protocol to use to the instance. Valid values are `HTTP`, `HTTPS`, `TCP`, or `SSL`
- `lb_port` - (Required) The port to listen on for the load balancer
- `lb_protocol` - (Required) The protocol to listen on. Valid values are `HTTP`, `HTTPS`, `TCP`, or `SSL`
- `ssl_certificate_id` - (Optional) The ARN of an SSL certificate you have uploaded to AWS IAM. **Note ECDSA-specific restrictions below. Only valid when `lb_protocol` is either HTTPS or SSL**

Health Check (`health_check`) supports the following:

- `healthy_threshold` - (Required) The number of checks before the instance is declared healthy.
- `unhealthy_threshold` - (Required) The number of checks before the instance is declared unhealthy.
- `target` - (Required) The target of the check. Valid pattern is "${PROTOCOL}:${PORT}${PATH}", where PROTOCOL values are:
  - `HTTP`, `HTTPS` - PORT and PATH are required
  - `TCP`, `SSL` - PORT is required, PATH is not supported
- `interval` - (Required) The interval between checks.
- `timeout` - (Required) The length of time before the check times out.

» Note on ECDSA Key Algorithm

If the ARN of the `ssl_certificate_id` that is pointed to references a certificate that was signed by an ECDSA key, note that ELB only supports the P256 and P384 curves. Using a certificate signed by a key using a different curve could produce the error `ERR_SSL_VERSION_OR_CIPHER_MISMATCH` in your browser.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the ELB
- `arn` - The ARN of the ELB
- `name` - The name of the ELB
- `dns_name` - The DNS name of the ELB
- `instances` - The list of instances in the ELB
- `source_security_group` - The name of the security group that you can use as part of your inbound rules for your load balancer's back-end application instances. Use this for Classic or Default VPC only.
- `source_security_group_id` - The ID of the security group that you can use as part of your inbound rules for your load balancer's back-end application instances. Only available on ELBs launched in a VPC.
- `zone_id` - The canonical hosted zone ID of the ELB (to be used in a Route 53 Alias record)

» Import

ELBs can be imported using the `name`, e.g.

```
$ terraform import aws_elb.bar elb-production-12345
```

» Resource: aws_elb_attachment

Attaches an EC2 instance to an Elastic Load Balancer (ELB). For attaching resources with Application Load Balancer (ALB) or Network Load Balancer (NLB), see the `aws_lb_target_group_attachment` resource.

NOTE on ELB Instances and ELB Attachments: Terraform currently provides both a standalone ELB Attachment resource (describing an instance attached to an ELB), and an Elastic Load Balancer resource with `instances` defined in-line. At this time you cannot use an ELB with in-line instances in conjunction with an ELB Attachment resource. Doing so will cause a conflict and will overwrite attachments.

» Example Usage

```
# Create a new load balancer attachment
resource "aws_elb_attachment" "baz" {
  elb      = "${aws_elb.bar.id}"
  instance = "${aws_instance.foo.id}"
}
```

» Argument Reference

The following arguments are supported:

- `elb` - (Required) The name of the ELB.
- `instance` - (Required) Instance ID to place in the ELB pool.

» Resource: aws_lb_cookie_stickiness_policy

Provides a load balancer cookie stickiness policy, which allows an ELB to control the sticky session lifetime of the browser.

» Example Usage

```
resource "aws_elb" "lb" {
  name               = "test-lb"
  availability_zones = ["us-east-1a"]

  listener {
    instance_port     = 8000
    instance_protocol = "http"
    lb_port           = 80
    lb_protocol       = "http"
  }
}

resource "aws_lb_cookie_stickiness_policy" "foo" {
  name                     = "foo-policy"
  load_balancer            = "${aws_elb.lb.id}"
  lb_port                  = 80
  cookie_expiration_period = 600
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the stickiness policy.
- `load_balancer` - (Required) The load balancer to which the policy should be attached.
- `lb_port` - (Required) The load balancer port to which the policy should be applied. This must be an active listener on the load balancer.

- `cookie_expiration_period` - (Optional) The time period after which the session cookie should be considered stale, expressed in seconds.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the policy.
- `name` - The name of the stickiness policy.
- `load_balancer` - The load balancer to which the policy is attached.
- `lb_port` - The load balancer port to which the policy is applied.
- `cookie_expiration_period` - The time period after which the session cookie is considered stale, expressed in seconds.

» Resource: aws_lb_ssl_negotiation_policy

Provides a load balancer SSL negotiation policy, which allows an ELB to control the ciphers and protocols that are supported during SSL negotiations between a client and a load balancer.

» Example Usage

```
resource "aws_elb" "lb" {
  name               = "test-lb"
  availability_zones = ["us-east-1a"]

  listener {
    instance_port      = 8000
    instance_protocol  = "https"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "arn:aws:iam::123456789012:server-certificate/certName"
  }
}

resource "aws_lb_ssl_negotiation_policy" "foo" {
  name          = "foo-policy"
  load_balancer = "${aws_elb.lb.id}"
  lb_port       = 443

  attribute {
    name  = "Protocol-TLSv1"
    value = "false"
  }

  attribute {
    name  = "Protocol-TLSv1.1"
    value = "false"
  }

  attribute {
    name  = "Protocol-TLSv1.2"
    value = "true"
  }

  attribute {
    name  = "Server-Defined-Cipher-Order"
    value = "true"
  }

  attribute {
    name  = "ECDHE-RSA-AES128-GCM-SHA256"
    value = "true"
  }

  attribute {
    name  = "AES128-GCM-SHA256"
    value = "true"
  }

  attribute {
    name  = "EDH-RSA-DES-CBC3-SHA"
    value = "false"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the SSL negotiation policy.
- `load_balancer` - (Required) The load balancer to which the policy should be attached.
- `lb_port` - (Required) The load balancer port to which the policy should be applied. This must be an active listener on the load balancer.
- `attribute` - (Optional) An SSL Negotiation policy attribute. Each has two properties:
    - `name` - The name of the attribute

  - `value` - The value of the attribute

To set your attributes, please see the AWS Elastic Load Balancing Developer Guide for a listing of the supported SSL protocols, SSL options, and SSL ciphers.

NOTE: The AWS documentation references Server Order Preference, which the AWS Elastic Load Balancing API refers to as `Server-Defined-Cipher-Order`. If you wish to set Server Order Preference, use this value instead.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the policy.
- `name` - The name of the stickiness policy.
- `load_balancer` - The load balancer to which the policy is attached.
- `lb_port` - The load balancer port to which the policy is applied.
- `attribute` - The SSL Negotiation policy attributes.

» Resource: aws_load_balancer_backend_server_policy

Attaches a load balancer policy to an ELB backend server.

» Example Usage

```
resource "aws_elb" "wu-tang" {
  name               = "wu-tang"
  availability_zones = ["us-east-1a"]

  listener {
    instance_port      = 443
    instance_protocol  = "http"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "arn:aws:iam::000000000000:server-certificate/wu-tang.net"
  }

  tags = {
    Name = "wu-tang"
  }
}

resource "aws_load_balancer_policy" "wu-tang-ca-pubkey-policy" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-ca-pubkey-policy"
  policy_type_name   = "PublicKeyPolicyType"

  policy_attribute {
    name  = "PublicKey"
    value = "${file("wu-tang-pubkey")}"
  }
}

resource "aws_load_balancer_policy" "wu-tang-root-ca-backend-auth-policy" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-root-ca-backend-auth-policy"
  policy_type_name   = "BackendServerAuthenticationPolicyType"

  policy_attribute {
    name  = "PublicKeyPolicyName"
    value = "${aws_load_balancer_policy.wu-tang-root-ca-pubkey-policy.policy_name}"
  }
}

resource "aws_load_balancer_backend_server_policy" "wu-tang-backend-auth-policies-443" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  instance_port      = 443

  policy_names = [
    "${aws_load_balancer_policy.wu-tang-root-ca-backend-auth-policy.policy_name}",
  ]
}
```

Where the file `pubkey` in the current directory contains only the *public key* of the certificate.

```
cat wu-tang-ca.pem | openssl x509 -pubkey -noout | grep -v '\-\-\-\-' | tr -d '\n' > wu-tang
```

This example shows how to enable backend authentication for an ELB as well as customize the TLS settings.

» Argument Reference

The following arguments are supported:

- `load_balancer_name` - (Required) The load balancer to attach the policy to.
- `policy_names` - (Required) List of Policy Names to apply to the backend server.
- `instance_port` - (Required) The instance port to apply the policy to.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the policy.
- `load_balancer_name` - The load balancer on which the policy is defined.
- `instance_port` - The backend port the policies are applied to

» Resource: aws_load_balancer_listener_policy

Attaches a load balancer policy to an ELB Listener.

» Example Usage for Custom Policy

```
resource "aws_elb" "wu-tang" {
  name               = "wu-tang"
  availability_zones = ["us-east-1a"]

  listener {
    instance_port      = 443
    instance_protocol  = "http"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "arn:aws:iam::000000000000:server-certificate/wu-tang.net"
  }

  tags = {
    Name = "wu-tang"
  }
}

resource "aws_load_balancer_policy" "wu-tang-ssl" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-ssl"
  policy_type_name   = "SSLNegotiationPolicyType"

  policy_attribute {
    name  = "ECDHE-ECDSA-AES128-GCM-SHA256"
    value = "true"
  }

  policy_attribute {
    name  = "Protocol-TLSv1.2"
    value = "true"
  }
}

resource "aws_load_balancer_listener_policy" "wu-tang-listener-policies-443" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  load_balancer_port = 443

  policy_names = [
    "${aws_load_balancer_policy.wu-tang-ssl.policy_name}",
  ]
}
```

This example shows how to customize the TLS settings of an HTTPS listener.

» Example Usage for AWS Predefined Security Policy

```
resource "aws_elb" "wu-tang" {
  name               = "wu-tang"
  availability_zones = ["us-east-1a"]

  listener {
    instance_port      = 443
    instance_protocol  = "http"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "arn:aws:iam::000000000000:server-certificate/wu-tang.net"
  }

  tags = {
    Name = "wu-tang"
  }
}

resource "aws_load_balancer_policy" "wu-tang-ssl-tls-1-1" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-ssl"
  policy_type_name   = "SSLNegotiationPolicyType"

  policy_attribute {
    name  = "Reference-Security-Policy"
    value = "ELBSecurityPolicy-TLS-1-1-2017-01"
  }
}

resource "aws_load_balancer_listener_policy" "wu-tang-listener-policies-443" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  load_balancer_port = 443

  policy_names = [
    "${aws_load_balancer_policy.wu-tang-ssl-tls-1-1.policy_name}",
  ]
}
```

This example shows how to add a Predefined Security Policy for ELBs

» Argument Reference

The following arguments are supported:

- `load_balancer_name` - (Required) The load balancer to attach the policy to.
- `load_balancer_port` - (Required) The load balancer listener port to apply the policy to.
- `policy_names` - (Required) List of Policy Names to apply to the backend server.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the policy.
- `load_balancer_name` - The load balancer on which the policy is defined.
- `load_balancer_port` - The load balancer listener port the policies are applied to

» Resource: aws_load_balancer_policy

Provides a load balancer policy, which can be attached to an ELB listener or backend server.

» Example Usage

```
resource "aws_elb" "wu-tang" {
  name               = "wu-tang"
  availability_zones = ["us-east-1a"]

  listener {
    instance_port      = 443
    instance_protocol  = "http"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "arn:aws:iam::000000000000:server-certificate/wu-tang.net"
  }

  tags = {
    Name = "wu-tang"
  }
}

resource "aws_load_balancer_policy" "wu-tang-ca-pubkey-policy" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-ca-pubkey-policy"
  policy_type_name   = "PublicKeyPolicyType"

  policy_attribute {
    name  = "PublicKey"
    value = "${file("wu-tang-pubkey")}"
  }
}

resource "aws_load_balancer_policy" "wu-tang-root-ca-backend-auth-policy" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-root-ca-backend-auth-policy"
  policy_type_name   = "BackendServerAuthenticationPolicyType"

  policy_attribute {
    name  = "PublicKeyPolicyName"
    value = "${aws_load_balancer_policy.wu-tang-root-ca-pubkey-policy.policy_name}"
  }
}

resource "aws_load_balancer_policy" "wu-tang-ssl" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-ssl"
  policy_type_name   = "SSLNegotiationPolicyType"

  policy_attribute {
    name  = "ECDHE-ECDSA-AES128-GCM-SHA256"
    value = "true"
  }

  policy_attribute {
    name  = "Protocol-TLSv1.2"
    value = "true"
  }
}

resource "aws_load_balancer_policy" "wu-tang-ssl-tls-1-1" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  policy_name        = "wu-tang-ssl"
  policy_type_name   = "SSLNegotiationPolicyType"

  policy_attribute {
    name  = "Reference-Security-Policy"
    value = "ELBSecurityPolicy-TLS-1-1-2017-01"
  }
}

resource "aws_load_balancer_backend_server_policy" "wu-tang-backend-auth-policies-443" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  instance_port      = 443

  policy_names = [
    "${aws_load_balancer_policy.wu-tang-root-ca-backend-auth-policy.policy_name}",
  ]
}

resource "aws_load_balancer_listener_policy" "wu-tang-listener-policies-443" {
  load_balancer_name = "${aws_elb.wu-tang.name}"
  load_balancer_port = 443

  policy_names = [
    "${aws_load_balancer_policy.wu-tang-ssl.policy_name}",
  ]
}
```

Where the file `pubkey` in the current directory contains only the *public key* of the certificate.

```
cat wu-tang-ca.pem | openssl x509 -pubkey -noout | grep -v '\-\-\-\-' | tr -d '\n' > wu-tang
```

This example shows how to enable backend authentication for an ELB as well as customize the TLS settings.

» Argument Reference

The following arguments are supported:

- `load_balancer_name` - (Required) The load balancer on which the policy is defined.
- `policy_name` - (Required) The name of the load balancer policy.

- policy_type_name - (Required) The policy type.
- policy_attribute - (Optional) Policy attribute to apply to the policy.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The ID of the policy.
- policy_name - The name of the stickiness policy.
- policy_type_name - The policy type of the policy.
- load_balancer_name - The load balancer on which the policy is defined.

» Resource: aws_proxy_protocol_policy

Provides a proxy protocol policy, which allows an ELB to carry a client connection information to a backend.

» Example Usage

```
resource "aws_elb" "lb" {
  name               = "test-lb"
  availability_zones = ["us-east-1a"]

  listener {
    instance_port     = 25
    instance_protocol = "tcp"
    lb_port           = 25
    lb_protocol       = "tcp"
  }

  listener {
    instance_port     = 587
    instance_protocol = "tcp"
    lb_port           = 587
    lb_protocol       = "tcp"
  }
}

resource "aws_proxy_protocol_policy" "smtp" {
  load_balancer  = "${aws_elb.lb.name}"
  instance_ports = ["25", "587"]
}
```

» Argument Reference

The following arguments are supported:

- `load_balancer` - (Required) The load balancer to which the policy should be attached.
- `instance_ports` - (Required) List of instance ports to which the policy should be applied. This can be specified if the protocol is SSL or TCP.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the policy.
- `load_balancer` - The load balancer to which the policy is attached.

» Data Source: aws_lb

Note: `aws_alb` is known as `aws_lb`. The functionality is identical.

Provides information about a Load Balancer.

This data source can prove useful when a module accepts an LB as an input variable and needs to, for example, determine the security groups associated with it, etc.

» Example Usage

```
variable "lb_arn" {
  type    = "string"
  default = ""
}

variable "lb_name" {
  type    = "string"
  default = ""
}

data "aws_lb" "test" {
  arn  = "${var.lb_arn}"
  name = "${var.lb_name}"
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Optional) The full ARN of the load balancer.
- `name` - (Optional) The unique name of the load balancer.

NOTE: When both `arn` and `name` are specified, `arn` takes precedence.

» Attributes Reference

See the LB Resource for details on the returned attributes - they are identical.

» Data Source: aws_lb_listener

Note: `aws_alb_listener` is known as `aws_lb_listener`. The functionality is identical.

Provides information about a Load Balancer Listener.

This data source can prove useful when a module accepts an LB Listener as an input variable and needs to know the LB it is attached to, or other information specific to the listener in question.

» Example Usage

```
# get listener from listener arn

variable "listener_arn" {
  type = "string"
}

data "aws_lb_listener" "listener" {
  arn = "${var.listener_arn}"
}

# get listener from load_balancer_arn and port

data "aws_lb" "selected" {
  name = "default-public"
}

data "aws_lb_listener" "selected443" {
  load_balancer_arn = "${data.aws_lb.selected.arn}"
  port              = 443
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Optional) The arn of the listener. Required if `load_balancer_arn` and `port` is not set.
- `load_balancer_arn` - (Optional) The arn of the load balancer. Required if `arn` is not set.
- `port` - (Optional) The port of the listener. Required if `arn` is not set.

» Attributes Reference

See the LB Listener Resource for details on the returned attributes - they are identical.

» Data Source: aws_lb_target_group

Note: `aws_alb_target_group` is known as `aws_lb_target_group`. The functionality is identical.

Provides information about a Load Balancer Target Group.

This data source can prove useful when a module accepts an LB Target Group as an input variable and needs to know its attributes. It can also be used to get the ARN of an LB Target Group for use in other resources, given LB Target Group name.

» Example Usage

```
variable "lb_tg_arn" {
  type    = "string"
  default = ""
}

variable "lb_tg_name" {
  type    = "string"
  default = ""
}

data "aws_lb_target_group" "test" {
  arn  = "${var.lb_tg_arn}"
  name = "${var.lb_tg_name}"
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Optional) The full ARN of the target group.
- `name` - (Optional) The unique name of the target group.

NOTE: When both `arn` and `name` are specified, `arn` takes precedence.

» Attributes Reference

See the LB Target Group Resource for details on the returned attributes - they are identical.

» Data Source: aws_lb

Note: `aws_alb` is known as `aws_lb`. The functionality is identical.

Provides information about a Load Balancer.

This data source can prove useful when a module accepts an LB as an input variable and needs to, for example, determine the security groups associated with it, etc.

» Example Usage

```
variable "lb_arn" {
  type    = "string"
  default = ""
}

variable "lb_name" {
  type    = "string"
  default = ""
}

data "aws_lb" "test" {
  arn  = "${var.lb_arn}"
  name = "${var.lb_name}"
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Optional) The full ARN of the load balancer.
- `name` - (Optional) The unique name of the load balancer.

NOTE: When both `arn` and `name` are specified, `arn` takes precedence.

» Attributes Reference

See the LB Resource for details on the returned attributes - they are identical.

» Data Source: aws_lb_listener

Note: `aws_alb_listener` is known as `aws_lb_listener`. The functionality is identical.

Provides information about a Load Balancer Listener.

This data source can prove useful when a module accepts an LB Listener as an input variable and needs to know the LB it is attached to, or other information specific to the listener in question.

» Example Usage

```
# get listener from listener arn

variable "listener_arn" {
  type = "string"
}

data "aws_lb_listener" "listener" {
  arn = "${var.listener_arn}"
}

# get listener from load_balancer_arn and port

data "aws_lb" "selected" {
  name = "default-public"
}

data "aws_lb_listener" "selected443" {
  load_balancer_arn = "${data.aws_lb.selected.arn}"
  port              = 443
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Optional) The arn of the listener. Required if `load_balancer_arn` and `port` is not set.
- `load_balancer_arn` - (Optional) The arn of the load balancer. Required if `arn` is not set.
- `port` - (Optional) The port of the listener. Required if `arn` is not set.

» Attributes Reference

See the LB Listener Resource for details on the returned attributes - they are identical.

» Data Source: aws_lb_target_group

Note: `aws_alb_target_group` is known as `aws_lb_target_group`. The functionality is identical.

Provides information about a Load Balancer Target Group.

This data source can prove useful when a module accepts an LB Target Group as an input variable and needs to know its attributes. It can also be used to get the ARN of an LB Target Group for use in other resources, given LB Target Group name.

» Example Usage

```
variable "lb_tg_arn" {
  type    = "string"
  default = ""
}

variable "lb_tg_name" {
  type    = "string"
  default = ""
}

data "aws_lb_target_group" "test" {
  arn  = "${var.lb_tg_arn}"
  name = "${var.lb_tg_name}"
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Optional) The full ARN of the target group.
- `name` - (Optional) The unique name of the target group.

NOTE: When both `arn` and `name` are specified, `arn` takes precedence.

» Attributes Reference

See the LB Target Group Resource for details on the returned attributes - they are identical.

» Resource: aws_lb

Provides a Load Balancer resource.

Note: `aws_alb` is known as `aws_lb`. The functionality is identical.

» Example Usage

» Application Load Balancer

```
resource "aws_lb" "test" {
  name               = "test-lb-tf"
  internal           = false
  load_balancer_type = "application"
  security_groups    = ["${aws_security_group.lb_sg.id}"]
  subnets            = ["${aws_subnet.public.*.id}"]

  enable_deletion_protection = true

  access_logs {
    bucket  = "${aws_s3_bucket.lb_logs.bucket}"
    prefix  = "test-lb"
    enabled = true
  }
  tags = {
    Environment = "production"
  }
}
```

» Network Load Balancer

```
resource "aws_lb" "test" {
  name               = "test-lb-tf"
  internal           = false
  load_balancer_type = "network"
  subnets            = ["${aws_subnet.public.*.id}"]

  enable_deletion_protection = true

  tags = {
    Environment = "production"
  }
}
```

» Specifying Elastic IPs

```
resource "aws_lb" "example" {
  name               = "example"
  load_balancer_type = "network"

  subnet_mapping {
    subnet_id     = "${aws_subnet.example1.id}"
    allocation_id = "${aws_eip.example1.id}"
  }

  subnet_mapping {
    subnet_id     = "${aws_subnet.example2.id}"
    allocation_id = "${aws_eip.example2.id}"
  }
}
```

» Argument Reference

NOTE: Please note that internal LBs can only use `ipv4` as the ip_address_type. You can only change to `dualstack` ip_address_type if the selected subnets are IPv6 enabled.

NOTE: Please note that one of either `subnets` or `subnet_mapping` is required.

The following arguments are supported:

- `name` - (Optional) The name of the LB. This name must be unique within your AWS account, can have a maximum of 32 characters, must contain only alphanumeric characters or hyphens, and must not begin or end with a hyphen. If not specified, Terraform will autogenerate a name beginning with `tf-lb`.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `internal` - (Optional) If true, the LB will be internal.
- `load_balancer_type` - (Optional) The type of load balancer to create. Possible values are `application` or `network`. The default value is `application`.
- `security_groups` - (Optional) A list of security group IDs to assign to the LB. Only valid for Load Balancers of type `application`.
- `access_logs` - (Optional) An Access Logs block. Access Logs documented below.
- `subnets` - (Optional) A list of subnet IDs to attach to the LB. Subnets cannot be updated for Load Balancers of type `network`. Changing this value for load balancers of type `network` will force a recreation of the resource.
- `subnet_mapping` - (Optional) A subnet mapping block as documented below.
- `idle_timeout` - (Optional) The time in seconds that the connection is allowed to be idle. Only valid for Load Balancers of type `application`. Default: 60.
- `enable_deletion_protection` - (Optional) If true, deletion of the load balancer will be disabled via the AWS API. This will prevent Terraform from deleting the load balancer. Defaults to `false`.
- `enable_cross_zone_load_balancing` - (Optional) If true, cross-zone load balancing of the load balancer will be enabled. This is a `network` load balancer feature. Defaults to `false`.
- `enable_http2` - (Optional) Indicates whether HTTP/2 is enabled in `application` load balancers. Defaults to `true`.
- `ip_address_type` - (Optional) The type of IP addresses used by the subnets for your load balancer. The possible values are `ipv4` and `dualstack`
- `tags` - (Optional) A mapping of tags to assign to the resource.

Access Logs (`access_logs`) support the following:

- `bucket` - (Required) The S3 bucket name to store the logs in.
- `prefix` - (Optional) The S3 bucket prefix. Logs are stored in the root if not configured.
- `enabled` - (Optional) Boolean to enable / disable `access_logs`. Defaults to `false`, even when `bucket` is specified.

Subnet Mapping (`subnet_mapping`) blocks support the following:

- `subnet_id` - (Required) The id of the subnet of which to attach to the load balancer. You can specify only one subnet per Availability Zone.
- `allocation_id` - (Optional) The allocation ID of the Elastic IP address.

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The ARN of the load balancer (matches `arn`).
- `arn` - The ARN of the load balancer (matches `id`).
- `arn_suffix` - The ARN suffix for use with CloudWatch Metrics.
- `dns_name` - The DNS name of the load balancer.
- `zone_id` - The canonical hosted zone ID of the load balancer (to be used in a Route 53 Alias record).

» Timeouts

`aws_lb` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for Creating LB
- `update` - (Default `10 minutes`) Used for LB modifications
- `delete` - (Default `10 minutes`) Used for destroying LB

» Import

LBs can be imported using their ARN, e.g.

```
$ terraform import aws_lb.bar arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalanc
```

» Resource: aws_lb_listener

Provides a Load Balancer Listener resource.

Note: `aws_alb_listener` is known as `aws_lb_listener`. The functionality is identical.

» Example Usage

» Forward Action

```
resource "aws_lb" "front_end" {
  # ...
}

resource "aws_lb_target_group" "front_end" {
  # ...
}

resource "aws_lb_listener" "front_end" {
  load_balancer_arn = "${aws_lb.front_end.arn}"
  port              = "443"
  protocol          = "HTTPS"
  ssl_policy        = "ELBSecurityPolicy-2016-08"
  certificate_arn   = "arn:aws:iam::187416307283:server-certificate/test_cert_rab3wuqwgja25c

  default_action {
    type             = "forward"
    target_group_arn = "${aws_lb_target_group.front_end.arn}"
  }
}
```

» Redirect Action

```
resource "aws_lb" "front_end" {
  # ...
}

resource "aws_lb_listener" "front_end" {
  load_balancer_arn = "${aws_lb.front_end.arn}"
  port              = "80"
  protocol          = "HTTP"

  default_action {
    type = "redirect"

    redirect {
      port        = "443"
      protocol    = "HTTPS"
      status_code = "HTTP_301"
    }
  }
}
```

» Fixed-response Action

```
resource "aws_lb" "front_end" {
  # ...
}

resource "aws_lb_listener" "front_end" {
  load_balancer_arn = "${aws_lb.front_end.arn}"
  port              = "80"
  protocol          = "HTTP"

  default_action {
    type = "fixed-response"

    fixed_response {
      content_type = "text/plain"
      message_body = "Fixed response content"
      status_code  = "200"
    }
  }
}
```

» Authenticate-cognito Action

```
resource "aws_lb" "front_end" {
  # ...
}

resource "aws_lb_target_group" "front_end" {
  # ...
}

resource "aws_cognito_user_pool" "pool" {
  # ...
}

resource "aws_cognito_user_pool_client" "client" {
  # ...
}

resource "aws_cognito_user_pool_domain" "domain" {
  # ...
}

resource "aws_lb_listener" "front_end" {
  load_balancer_arn = "${aws_lb.front_end.arn}"
  port              = "80"
  protocol          = "HTTP"
  default_action {
    type = "authenticate-cognito"

    authenticate_cognito {
      user_pool_arn       = "${aws_cognito_user_pool.pool.arn}"
      user_pool_client_id = "${aws_cognito_user_pool_client.client.id}"
      user_pool_domain    = "${aws_cognito_user_pool_domain.domain.domain}"
    }
  }

  default_action {
    type             = "forward"
    target_group_arn = "${aws_lb_target_group.front_end.arn}"
  }
}
```

» Authenticate-oidc Action

```
resource "aws_lb" "front_end" {
  # ...
}

resource "aws_lb_target_group" "front_end" {
  # ...
}

resource "aws_lb_listener" "front_end" {
  load_balancer_arn = "${aws_lb.front_end.arn}"
  port              = "80"
  protocol          = "HTTP"

  default_action {
    type = "authenticate-oidc"

    authenticate_oidc {
      authorization_endpoint = "https://example.com/authorization_endpoint"
      client_id              = "client_id"
      client_secret          = "client_secret"
      issuer                 = "https://example.com"
      token_endpoint         = "https://example.com/token_endpoint"
      user_info_endpoint     = "https://example.com/user_info_endpoint"
    }
  }
  default_action {
    type             = "forward"
    target_group_arn = "${aws_lb_target_group.front_end.arn}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `load_balancer_arn` - (Required, Forces New Resource) The ARN of the load balancer.
- `port` - (Required) The port on which the load balancer is listening.
- `protocol` - (Optional) The protocol for connections from clients to the load balancer. Valid values are `TCP`, `TLS`, `UDP`, `TCP_UDP`, `HTTP` and `HTTPS`. Defaults to `HTTP`.
- `ssl_policy` - (Optional) The name of the SSL Policy for the listener. Required if `protocol` is `HTTPS` or `TLS`.
- `certificate_arn` - (Optional) The ARN of the default SSL server certificate. Exactly one certificate is required if the protocol is HTTPS. For adding additional SSL certificates, see the `aws_lb_listener_certificate` resource.
- `default_action` - (Required) An Action block. Action blocks are documented below.

NOTE:: Please note that listeners that are attached to Application Load Balancers must use either `HTTP` or `HTTPS` protocols while listeners that are attached to Network Load Balancers must use the `TCP` protocol.

Action Blocks (for `default_action`) support the following:

- `type` - (Required) The type of routing action. Valid values are `forward`, `redirect`, `fixed-response`, `authenticate-cognito` and `authenticate-oidc`.
- `target_group_arn` - (Optional) The ARN of the Target Group to which to route traffic. Required if `type` is `forward`.
- `redirect` - (Optional) Information for creating a redirect action. Required if `type` is `redirect`.
- `fixed_response` - (Optional) Information for creating an action that returns a custom HTTP response. Required if `type` is `fixed-response`.

Redirect Blocks (for `redirect`) support the following:

NOTE:: You can reuse URI components using the following reserved keywords: `#{protocol}`, `#{host}`, `#{port}`, `#{path}` (the leading "/" is removed) and `#{query}`.

- host - (Optional) The hostname. This component is not percent-encoded. The hostname can contain #{host}. Defaults to #{host}.
- path - (Optional) The absolute path, starting with the leading "/". This component is not percent-encoded. The path can contain #{host}, #{path}, and #{port}. Defaults to /#{path}.
- port - (Optional) The port. Specify a value from 1 to 65535 or #{port}. Defaults to #{port}.
- protocol - (Optional) The protocol. Valid values are HTTP, HTTPS, or #{protocol}. Defaults to #{protocol}.
- query - (Optional) The query parameters, URL-encoded when necessary, but not percent-encoded. Do not include the leading "?". Defaults to #{query}.
- status_code - (Required) The HTTP redirect code. The redirect is either permanent (HTTP_301) or temporary (HTTP_302).

Fixed-response Blocks (for fixed_response) support the following:

- content_type - (Required) The content type. Valid values are text/plain, text/css, text/html, application/javascript and application/json.
- message_body - (Optional) The message body.
- status_code - (Optional) The HTTP response code. Valid values are 2XX, 4XX, or 5XX.

Authenticate Cognito Blocks (for authenticate_cognito) supports the following:

- authentication_request_extra_params - (Optional) The query parameters to include in the redirect request to the authorization endpoint. Max: 10.
- on_unauthenticated_request - (Optional) The behavior if the user is not authenticated. Valid values: deny, allow and authenticate
- scope - (Optional) The set of user claims to be requested from the IdP.
- session_cookie_name - (Optional) The name of the cookie used to maintain session information.
- session_timeout - (Optional) The maximum duration of the authentication session, in seconds.
- user_pool_arn - (Required) The ARN of the Cognito user pool.
- user_pool_client_id - (Required) The ID of the Cognito user pool client.
- user_pool_domain - (Required) The domain prefix or fully-qualified domain name of the Cognito user pool.

Authenticate OIDC Blocks (for authenticate_oidc) supports the following:

- authentication_request_extra_params - (Optional) The query parameters to include in the redirect request to the authorization endpoint. Max: 10.

- `authorization_endpoint` - (Required) The authorization endpoint of the IdP.
- `client_id` - (Required) The OAuth 2.0 client identifier.
- `client_secret` - (Required) The OAuth 2.0 client secret.
- `issuer` - (Required) The OIDC issuer identifier of the IdP.
- `on_unauthenticated_request` - (Optional) The behavior if the user is not authenticated. Valid values: `deny`, `allow` and `authenticate`
- `scope` - (Optional) The set of user claims to be requested from the IdP.
- `session_cookie_name` - (Optional) The name of the cookie used to maintain session information.
- `session_timeout` - (Optional) The maximum duration of the authentication session, in seconds.
- `token_endpoint` - (Required) The token endpoint of the IdP.
- `user_info_endpoint` - (Required) The user info endpoint of the IdP.

Authentication Request Extra Params Blocks (for `authentication_request_extra_params`) supports the following:

- `key` - (Required) The key of query parameter
- `value` - (Required) The value of query parameter

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The ARN of the listener (matches `arn`)
- `arn` - The ARN of the listener (matches `id`)

» Import

Listeners can be imported using their ARN, e.g.

```
$ terraform import aws_lb_listener.front_end arn:aws:elasticloadbalancing:us-west-2:18741630
```

» Resource: aws_lb_listener_certificate

Provides a Load Balancer Listener Certificate resource.

This resource is for additional certificates and does not replace the default certificate on the listener.

Note: `aws_alb_listener_certificate` is known as `aws_lb_listener_certificate`. The functionality is identical.

» Example Usage

```
resource "aws_acm_certificate" "example" {
  # ...
}

resource "aws_lb" "front_end" {
  # ...
}

resource "aws_lb_listener" "front_end" {
  # ...
}

resource "aws_lb_listener_certificate" "example" {
  listener_arn    = "${aws_lb_listener.front_end.arn}"
  certificate_arn = "${aws_acm_certificate.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `listener_arn` - (Required, Forces New Resource) The ARN of the listener to which to attach the certificate.
- `certificate_arn` - (Required, Forces New Resource) The ARN of the certificate to attach to the listener.

» Resource: aws_lb_listener_rule

Provides a Load Balancer Listener Rule resource.

Note: `aws_alb_listener_rule` is known as `aws_lb_listener_rule`. The functionality is identical.

» Example Usage

```
resource "aws_lb" "front_end" {
  # ...
}

resource "aws_lb_listener" "front_end" {
  # Other parameters
```

```
}

resource "aws_lb_listener_rule" "static" {
  listener_arn = "${aws_lb_listener.front_end.arn}"
  priority     = 100

  action {
    type             = "forward"
    target_group_arn = "${aws_lb_target_group.static.arn}"
  }

  condition {
    path_pattern {
      values = ["/static/*"]
    }
  }

  condition {
    host_header {
      values = ["example.com"]
    }
  }
}

# Forward action

resource "aws_lb_listener_rule" "host_based_routing" {
  listener_arn = "${aws_lb_listener.front_end.arn}"
  priority     = 99

  action {
    type             = "forward"
    target_group_arn = "${aws_lb_target_group.static.arn}"
  }

  condition {
    host_header {
      values = ["my-service.*.terraform.io"]
    }
  }
}

# Redirect action

resource "aws_lb_listener_rule" "redirect_http_to_https" {
  listener_arn = "${aws_lb_listener.front_end.arn}"
  action {
    type = "redirect"

    redirect {
      port        = "443"
      protocol    = "HTTPS"
      status_code = "HTTP_301"
    }
  }

  condition {
    http_header {
      http_header_name = "X-Forwarded-For"
      values           = ["192.168.1.*"]
    }
  }
}

# Fixed-response action

resource "aws_lb_listener_rule" "health_check" {
  listener_arn = "${aws_lb_listener.front_end.arn}"

  action {
    type = "fixed-response"

    fixed_response {
      content_type = "text/plain"
      message_body = "HEALTHY"
      status_code  = "200"
    }
  }

  condition {
    query_string {
      key   = "health"
      value = "check"
    }

    query_string {
      value = "bar"
    }
  }
}
```

```
# Authenticate-cognito Action

resource "aws_cognito_user_pool" "pool" {
  # ...
}

resource "aws_cognito_user_pool_client" "client" {
  # ...
}

resource "aws_cognito_user_pool_domain" "domain" {
  # ...
}

resource "aws_lb_listener_rule" "admin" {
  listener_arn = "${aws_lb_listener.front_end.arn}"

  action {
    type = "authenticate-cognito"

    authenticate_cognito {
      user_pool_arn       = "${aws_cognito_user_pool.pool.arn}"
      user_pool_client_id = "${aws_cognito_user_pool_client.client.id}"
      user_pool_domain    = "${aws_cognito_user_pool_domain.domain.domain}"
    }
  }

  action {
    type             = "forward"
    target_group_arn = "${aws_lb_target_group.static.arn}"
  }
}

# Authenticate-oidc Action

resource "aws_lb_listener_rule" "admin" {
  listener_arn = "${aws_lb_listener.front_end.arn}"

  action {
    type = "authenticate-oidc"

    authenticate_oidc {
      authorization_endpoint = "https://example.com/authorization_endpoint"
      client_id              = "client_id"
      client_secret          = "client_secret"
      issuer                 = "https://example.com"
      token_endpoint         = "https://example.com/token_endpoint"
      user_info_endpoint     = "https://example.com/user_info_endpoint"
    }
  }

  action {
    type             = "forward"
    target_group_arn = "${aws_lb_target_group.static.arn}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `listener_arn` - (Required, Forces New Resource) The ARN of the listener to which to attach the rule.
- `priority` - (Optional) The priority for the rule between `1` and `50000`. Leaving it unset will automatically set the rule with next available priority after currently existing highest rule. A listener can't have multiple rules with the same priority.
- `action` - (Required) An Action block. Action blocks are documented below.
- `condition` - (Required) A Condition block. Multiple condition blocks of different types can be set and all must be satisfied for the rule to match. Condition blocks are documented below.

» Action Blocks

Action Blocks (for `action`) support the following:

- `type` - (Required) The type of routing action. Valid values are `forward`, `redirect`, `fixed-response`, `authenticate-cognito` and `authenticate-oidc`.
- `target_group_arn` - (Optional) The ARN of the Target Group to which to route traffic. Required if `type` is `forward`.
- `redirect` - (Optional) Information for creating a redirect action. Required if `type` is `redirect`.
- `fixed_response` - (Optional) Information for creating an action that returns a custom HTTP response. Required if `type` is `fixed-response`.
- `authenticate_cognito` - (Optional) Information for creating an authenticate action using Cognito. Required if `type` is `authenticate-cognito`.
- `authenticate_oidc` - (Optional) Information for creating an authenticate action using OIDC. Required if `type` is `authenticate-oidc`.

Redirect Blocks (for `redirect`) support the following:

NOTE:: You can reuse URI components using the following reserved keywords: `#{protocol}`, `#{host}`, `#{port}`, `#{path}` (the leading "/" is removed) and `#{query}`.

- `host` - (Optional) The hostname. This component is not percent-encoded. The hostname can contain `#{host}`. Defaults to `#{host}`.
- `path` - (Optional) The absolute path, starting with the leading "/". This component is not percent-encoded. The path can contain #{host}, #{path}, and #{port}. Defaults to `/#{path}`.
- `port` - (Optional) The port. Specify a value from `1` to `65535` or `#{port}`. Defaults to `#{port}`.
- `protocol` - (Optional) The protocol. Valid values are `HTTP`, `HTTPS`, or `#{protocol}`. Defaults to `#{protocol}`.
- `query` - (Optional) The query parameters, URL-encoded when necessary, but not percent-encoded. Do not include the leading "?". Defaults to `#{query}`.
- `status_code` - (Required) The HTTP redirect code. The redirect is either permanent (`HTTP_301`) or temporary (`HTTP_302`).

Fixed-response Blocks (for `fixed_response`) support the following:

- `content_type` - (Required) The content type. Valid values are `text/plain`, `text/css`, `text/html`, `application/javascript` and `application/json`.
- `message_body` - (Optional) The message body.
- `status_code` - (Optional) The HTTP response code. Valid values are `2XX`, `4XX`, or `5XX`.

Authenticate Cognito Blocks (for `authenticate_cognito`) supports the following:

- `authentication_request_extra_params` - (Optional) The query parameters to include in the redirect request to the authorization endpoint. Max: 10.
- `on_unauthenticated_request` - (Optional) The behavior if the user is not authenticated. Valid values: `deny`, `allow` and `authenticate`
- `scope` - (Optional) The set of user claims to be requested from the IdP.
- `session_cookie_name` - (Optional) The name of the cookie used to maintain session information.
- `session_timeout` - (Optional) The maximum duration of the authentication session, in seconds.
- `user_pool_arn` - (Required) The ARN of the Cognito user pool.
- `user_pool_client_id` - (Required) The ID of the Cognito user pool client.
- `user_pool_domain` - (Required) The domain prefix or fully-qualified domain name of the Cognito user pool.

Authenticate OIDC Blocks (for `authenticate_oidc`) supports the following:

- `authentication_request_extra_params` - (Optional) The query parameters to include in the redirect request to the authorization endpoint. Max: 10.
- `authorization_endpoint` - (Required) The authorization endpoint of the IdP.
- `client_id` - (Required) The OAuth 2.0 client identifier.
- `client_secret` - (Required) The OAuth 2.0 client secret.
- `issuer` - (Required) The OIDC issuer identifier of the IdP.
- `on_unauthenticated_request` - (Optional) The behavior if the user is not authenticated. Valid values: `deny`, `allow` and `authenticate`
- `scope` - (Optional) The set of user claims to be requested from the IdP.
- `session_cookie_name` - (Optional) The name of the cookie used to maintain session information.
- `session_timeout` - (Optional) The maximum duration of the authentication session, in seconds.
- `token_endpoint` - (Required) The token endpoint of the IdP.
- `user_info_endpoint` - (Required) The user info endpoint of the IdP.

Authentication Request Extra Params Blocks (for `authentication_request_extra_params`) supports the following:

- `key` - (Required) The key of query parameter
- `value` - (Required) The value of query parameter

» Condition Blocks

One or more condition blocks can be set per rule. Most condition types can only be specified once per rule except for `http-header` and `query-string` which can be specified multiple times.

Condition Blocks (for `condition`) support the following:

- `field` - (Optional, **DEPRECATED**) The type of condition. Valid values are `host-header` or `path-pattern`. Must also set `values`.
- `values` - (Optional, **DEPRECATED**) List of exactly one pattern to match. Required when `field` is set.
- `host_header` - (Optional) Contains a single `value` item which is a list of host header patterns to match. The maximum size of each pattern is 128 characters. Comparison is case insensitive. Wildcard characters supported: * (matches 0 or more characters) and ? (matches exactly 1 character). Only one pattern needs to match for the condition to be satisfied.
- `http_header` - (Optional) HTTP headers to match. HTTP Header block fields documented below.

- http_request_method - (Optional) Contains a single value item which is a list of HTTP request methods or verbs to match. Maximum size is 40 characters. Only allowed characters are A-Z, hyphen (-) and underscore (_). Comparison is case sensitive. Wildcards are not supported. Only one needs to match for the condition to be satisfied. AWS recommends that GET and HEAD requests are routed in the same way because the response to a HEAD request may be cached.
- path_pattern - (Optional) Contains a single value item which is a list of path patterns to match against the request URL. Maximum size of each pattern is 128 characters. Comparison is case sensitive. Wildcard charaters supported: * (matches 0 or more characters) and ? (matches exactly 1 character). Only one pattern needs to match for the condition to be satisfied. Path pattern is compared only to the path of the URL, not to its query string. To compare against the query string, use a query-string condition.
- query_string - (Optional) Query strings to match. Query String block fields documented below.
- source_ip - (Optional) Contains a single value item which is a list of source IP CIDR notations to match. You can use both IPv4 and IPv6 addresses. Wildcards are not supported. Condition is satisfied if the source IP address of the request matches one of the CIDR blocks. Condition is not satisfied by the addresses in the X-Forwarded-For header, use http-header condition instead.

NOTE:: Exactly one of field, host_header, http_header, http_request_method, path_pattern, query_string or source_ip must be set per condition.

» HTTP Header Blocks

HTTP Header Blocks (for http_header) support the following:

- http_header_name - (Required) Name of HTTP header to search. The maximum size is 40 characters. Comparison is case insensitive. Only RFC7240 characters are supported. Wildcards are not supported. You cannot use HTTP header condition to specify the host header, use a host-header condition instead.
- values - (Required) List of header value patterns to match. Maximum size of each pattern is 128 characters. Comparison is case insensitive. Wildcard characters supported: * (matches 0 or more characters) and ? (matches exactly 1 character). If the same header appears multiple times in the request they will be searched in order until a match is found. Only one pattern needs to match for the condition to be satisfied. To require that all of the strings are a match, create one condition block per string.

» Query String Blocks

Query String Blocks (for `query_string`) support the following:

- `values` - (Required) Query string pairs or values to match. Query String Value blocks documented below. Multiple `values` blocks can be specified, see example above. Maximum size of each string is 128 characters. Comparison is case insensitive. Wildcard characters supported: * (matches 0 or more characters) and ? (matches exactly 1 character). To search for a literal '*' or '?' character in a query string, escape the character with a backslash (\). Only one pair needs to match for the condition to be satisfied.

Query String Value Blocks (for `query_string.values`) support the following:

- `key` - (Optional) Query string key pattern to match.
- `value` - (Required) Query string value pattern to match.

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The ARN of the rule (matches `arn`)
- `arn` - The ARN of the rule (matches `id`)

» Import

Rules can be imported using their ARN, e.g.

```
$ terraform import aws_lb_listener_rule.front_end arn:aws:elasticloadbalancing:us-west-2:187
```

» Resource: aws_lb_target_group

Provides a Target Group resource for use with Load Balancer resources.

Note: `aws_alb_target_group` is known as `aws_lb_target_group`. The functionality is identical.

» Example Usage

» Instance Target Group

```
resource "aws_lb_target_group" "test" {
  name     = "tf-example-lb-tg"
  port     = 80
  protocol = "HTTP"
  vpc_id   = "${aws_vpc.main.id}"
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}
```

» IP Target Group

```
resource "aws_lb_target_group" "ip-example" {
  name        = "tf-example-lb-tg"
  port        = 80
  protocol    = "HTTP"
  target_type = "ip"
  vpc_id      = "${aws_vpc.main.id}"
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}
```

» Lambda Target Group

```
resource "aws_lb_target_group" "lambda-example" {
  name        = "tf-example-lb-tg"
  target_type = "lambda"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the target group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`. Cannot be longer than 6 characters.
- `port` - (Optional, Forces new resource) The port on which targets receive traffic, unless overridden when registering a specific target. Required when `target_type` is `instance` or `ip`. Does not apply when `target_type` is `lambda`.

- `protocol` - (Optional, Forces new resource) The protocol to use for routing traffic to the targets. Should be one of "TCP", "TLS", "UDP", "TCP_UDP", "HTTP" or "HTTPS". Required when `target_type` is `instance` or `ip`. Does not apply when `target_type` is `lambda`.
- `vpc_id` - (Optional, Forces new resource) The identifier of the VPC in which to create the target group. Required when `target_type` is `instance` or `ip`. Does not apply when `target_type` is `lambda`.
- `deregistration_delay` - (Optional) The amount time for Elastic Load Balancing to wait before changing the state of a deregistering target from draining to unused. The range is 0-3600 seconds. The default value is 300 seconds.
- `slow_start` - (Optional) The amount time for targets to warm up before the load balancer sends them a full share of requests. The range is 30-900 seconds or 0 to disable. The default value is 0 seconds.
- `lambda_multi_value_headers_enabled` - (Optional) Boolean whether the request and response headers exchanged between the load balancer and the Lambda function include arrays of values or strings. Only applies when `target_type` is `lambda`.
- `proxy_protocol_v2` - (Optional) Boolean to enable / disable support for proxy protocol v2 on Network Load Balancers. See doc for more information.
- `stickiness` - (Optional) A Stickiness block. Stickiness blocks are documented below. `stickiness` is only valid if used with Load Balancers of type `Application`
- `health_check` - (Optional) A Health Check block. Health Check blocks are documented below.
- `target_type` - (Optional, Forces new resource) The type of target that you must specify when registering targets with this target group. The possible values are `instance` (targets are specified by instance ID) or `ip` (targets are specified by IP address) or `lambda` (targets are specified by lambda arn). The default is `instance`. Note that you can't specify targets for a target group using both instance IDs and IP addresses. If the target type is `ip`, specify IP addresses from the subnets of the virtual private cloud (VPC) for the target group, the RFC 1918 range (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16), and the RFC 6598 range (100.64.0.0/10). You can't specify publicly routable IP addresses.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Stickiness Blocks (`stickiness`) support the following:

- `type` - (Required) The type of sticky sessions. The only current possible value is `lb_cookie`.

- `cookie_duration` - (Optional) The time period, in seconds, during which requests from a client should be routed to the same target. After this time period expires, the load balancer-generated cookie is considered stale. The range is 1 second to 1 week (604800 seconds). The default value is 1 day (86400 seconds).
- `enabled` - (Optional) Boolean to enable / disable `stickiness`. Default is `true`

NOTE: To help facilitate the authoring of modules that support target groups of any protocol, you can define `stickiness` regardless of the protocol chosen. However, for `TCP` target groups, `enabled` must be `false`.

Health Check Blocks (`health_check`):

Note: The Health Check parameters you can set vary by the `protocol` of the Target Group. Many parameters cannot be set to custom values for `network` load balancers at this time. See http://docs.aws.amazon.com/elasticloadbalancing/latest/APIReference/API_CreateTargetGroup.html for a complete reference. Keep in mind, that health checks produce actual requests to the backend. The underlying function is invoked when `target_type` is set to `lambda`.

- `enabled` - (Optional) Indicates whether health checks are enabled. Defaults to true.
- `interval` - (Optional) The approximate amount of time, in seconds, between health checks of an individual target. Minimum value 5 seconds, Maximum value 300 seconds. For `lambda` target groups, it needs to be greater as the `timeout` of the underlying `lambda`. Default 30 seconds.
- `path` - (Required for HTTP/HTTPS ALB) The destination for the health check request. Applies to Application Load Balancers only (HTTP/HTTPS), not Network Load Balancers (TCP).
- `port` - (Optional) The port to use to connect with the target. Valid values are either ports 1-65535, or `traffic-port`. Defaults to `traffic-port`.
- `protocol` - (Optional) The protocol to use to connect with the target. Defaults to `HTTP`. Not applicable when `target_type` is `lambda`.
- `timeout` - (Optional) The amount of time, in seconds, during which no response means a failed health check. For Application Load Balancers, the range is 2 to 120 seconds, and the default is 5 seconds for the `instance` target type and 30 seconds for the `lambda` target type. For Network Load Balancers, you cannot set a custom value, and the default is 10 seconds for TCP and HTTPS health checks and 6 seconds for HTTP health checks.
- `healthy_threshold` - (Optional) The number of consecutive health checks successes required before considering an unhealthy target healthy. Defaults to 3.
- `unhealthy_threshold` - (Optional) The number of consecutive health check failures required before considering the target unhealthy . For Network Load Balancers, this value must be the same as the `healthy_threshold`. Defaults to 3.

- `matcher` (Required for HTTP/HTTPS ALB) The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, ”200,202”) or a range of values (for example, ”200-299”). Applies to Application Load Balancers only (HTTP/HTTPS), not Network Load Balancers (TCP).

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The ARN of the Target Group (matches `arn`)
- `arn` - The ARN of the Target Group (matches `id`)
- `arn_suffix` - The ARN suffix for use with CloudWatch Metrics.
- `name` - The name of the Target Group

» Import

Target Groups can be imported using their ARN, e.g.

```
$ terraform import aws_lb_target_group.app_front_end arn:aws:elasticloadbalancing:us-west-2:
```

» Resource: aws_lb_target_group_attachment

Provides the ability to register instances and containers with an Application Load Balancer (ALB) or Network Load Balancer (NLB) target group. For attaching resources with Elastic Load Balancer (ELB), see the `aws_elb_attachment` resource.

Note: `aws_alb_target_group_attachment` is known as `aws_lb_target_group_attachment`. The functionality is identical.

» Example Usage

```
resource "aws_lb_target_group_attachment" "test" {
  target_group_arn = "${aws_lb_target_group.test.arn}"
  target_id        = "${aws_instance.test.id}"
  port             = 80
}

resource "aws_lb_target_group" "test" {
  // Other arguments
}
```

```
resource "aws_instance" "test" {
  // Other arguments
}
```

» Usage with lambda

```
resource "aws_lambda_permission" "with_lb" {
  statement_id  = "AllowExecutionFromlb"
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.test.arn}"
  principal     = "elasticloadbalancing.amazonaws.com"
  source_arn    = "${aws_lb_target_group.test.arn}"
}

resource "aws_lb_target_group" "test" {
  name        = "test"
  target_type = "lambda"
}

resource "aws_lambda_function" "test" {
  // Other arguments
}

resource "aws_lb_target_group_attachment" "test" {
  target_group_arn = "${aws_lb_target_group.test.arn}"
  target_id        = "${aws_lambda_function.test.arn}"
  depends_on       = ["aws_lambda_permission.with_lb"]
}
```

» Argument Reference

The following arguments are supported:

- `target_group_arn` - (Required) The ARN of the target group with which to register targets
- `target_id` (Required) The ID of the target. This is the Instance ID for an instance, or the container ID for an ECS container. If the target type is ip, specify an IP address. If the target type is lambda, specify the arn of lambda.
- `port` - (Optional) The port on which targets receive traffic.
- `availability_zone` - (Optional) The Availability Zone where the IP address of the target is to be registered.

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - A unique identifier for the attachment

» Import

Target Group Attachments cannot be imported.

» Resource: aws_emr_cluster

Provides an Elastic MapReduce Cluster, a web service that makes it easy to process large amounts of data efficiently. See Amazon Elastic MapReduce Documentation for more information.

To configure Instance Groups for task nodes, see the `aws_emr_instance_group` resource.

Support for Instance Fleets will be made available in an upcoming release.

» Example Usage

```
resource "aws_emr_cluster" "cluster" {
  name          = "emr-test-arn"
  release_label = "emr-4.6.0"
  applications  = ["Spark"]

  additional_info = <<EOF
{
  "instanceAwsClientConfiguration": {
    "proxyPort": 8099,
    "proxyHost": "myproxy.example.com"
  }
}
EOF

  termination_protection            = false
  keep_job_flow_alive_when_no_steps = true

  ec2_attributes {
    subnet_id                         = "${aws_subnet.main.id}"
    emr_managed_master_security_group = "${aws_security_group.sg.id}"
    emr_managed_slave_security_group  = "${aws_security_group.sg.id}"
    instance_profile                  = "${aws_iam_instance_profile.emr_profile.arn}"
  }

  master_instance_group {
    instance_type = "m4.large"
  }

  core_instance_group {
    instance_type  = "c4.large"
    instance_count = 1

    ebs_config {
      size                 = "40"
      type                 = "gp2"
      volumes_per_instance = 1
    }

    bid_price = "0.30"

    autosscaling_policy = <<EOF
{
"Constraints": {
  "MinCapacity": 1,
  "MaxCapacity": 2
},
"Rules": [
  {
    "Name": "ScaleOutMemoryPercentage",
    "Description": "Scale out if YARNMemoryAvailablePercentage is less than 15",
    "Action": {
      "SimpleScalingPolicyConfiguration": {
        "AdjustmentType": "CHANGE_IN_CAPACITY",
        "ScalingAdjustment": 1,
        "CoolDown": 300
      }
    },
    "Trigger": {
      "CloudWatchAlarmDefinition": {
        "ComparisonOperator": "LESS_THAN",
        "EvaluationPeriods": 1,
        "MetricName": "YARNMemoryAvailablePercentage",
        "Namespace": "AWS/ElasticMapReduce",
        "Period": 300,
        "Statistic": "AVERAGE",
        "Threshold": 15.0,
        "Unit": "PERCENT"
      }
    }
  }
]
}
EOF
  }

  ebs_root_volume_size = 100

  tags = {
    role = "rolename"
    env  = "env"
  }

  bootstrap_action {
    path = "s3://elasticmapreduce/bootstrap-actions/run-if"
    name = "runif"
    args = ["instance.isMaster=true", "echo running on master node"]
  }

  configurations_json = <<EOF
  [
    {
      "Classification": "hadoop-env",
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "JAVA_HOME": "/usr/lib/jvm/java-1.8.0"
          }
        }
      ],
      "Properties": {}
    },
    {
      "Classification": "spark-env",
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "JAVA_HOME": "/usr/lib/jvm/java-1.8.0"
          }
        }
      ],
      "Properties": {}
    }
  ]
EOF

  service_role = "${aws_iam_role.iam_emr_service_role.arn}"
}
```

The `aws_emr_cluster` resource typically requires two IAM roles, one for the EMR Cluster to use as a service, and another to place on your Cluster Instances to interact with AWS from those instances. The suggested role policy template for the EMR service is `AmazonElasticMapReduceRole`, and `AmazonElasticMapReduceforEC2Role` for the EC2 profile. See the Getting Started guide for more information on these IAM roles. There is also a fully-bootable example Terraform configuration at the bottom of this page.

» Enable Debug Logging

Debug logging in EMR is implemented as a step. It is highly recommended to utilize the lifecycle configuration block with `ignore_changes` if other steps are being managed outside of Terraform.

```
resource "aws_emr_cluster" "example" {
  # ... other configuration ...

  step {
    action_on_failure = "TERMINATE_CLUSTER"
    name              = "Setup Hadoop Debugging"

    hadoop_jar_step {
      jar  = "command-runner.jar"
      args = ["state-pusher-script"]
    }
  }

  # Optional: ignore outside changes to running cluster steps
  lifecycle {
    ignore_changes = ["step"]
  }
}
```

» Multiple Node Master Instance Group

Available in EMR version 5.23.0 and later, an EMR Cluster can be launched with three master nodes for high availability. Additional information about this functionality and its requirements can be found in the EMR Management Guide.

```
# This configuration is for illustrative purposes and highlights
# only relevant configurations for working with this functionality.

# Map public IP on launch must be enabled for public (Internet accessible) subnets
resource "aws_subnet" "example" {
  # ... other configuration ...

  map_public_ip_on_launch = true
}

resource "aws_emr_cluster" "example" {
  # ... other configuration ...

  # EMR version must be 5.23.0 or later
  release_label = "emr-5.24.1"

  # Termination protection is automatically enabled for multiple masters
  # To destroy the cluster, this must be configured to false and applied first
  termination_protection = true

  ec2_attributes {
    # ... other configuration ...

    subnet_id = "${aws_subnet.example.id}"
  }

  master_instance_group {
    # ... other configuration ...

    # Master instance count must be set to 3
    instance_count = 3
  }

  # core_instance_group must be configured
  core_instance_group {
    # ... other configuration ...
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the job flow
- `release_label` - (Required) The release label for the Amazon EMR release
- `master_instance_group` - (Optional) Configuration block to use an Instance Group for the master node type. Cannot be specified if `master_instance_type` argument or `instance_group` configuration blocks are set. Detailed below.
- `master_instance_type` - (Optional, **DEPRECATED**) Use the `master_instance_group` configuration block `instance_type` argument instead. The EC2 instance type of the master node. Cannot be specified if `master_instance_group` or `instance_group` configuration blocks are set.
- `scale_down_behavior` - (Optional) The way that individual Amazon EC2 instances terminate when an automatic scale-in activity occurs or an `instance group` is resized.
- `additional_info` - (Optional) A JSON string for selecting additional features such as adding proxy information. Note: Currently there is no API to retrieve the value of this argument after EMR cluster creation from provider, therefore Terraform cannot detect drift from the actual EMR cluster if its value is changed outside Terraform.
- `service_role` - (Required) IAM role that will be assumed by the Amazon EMR service to access AWS resources
- `security_configuration` - (Optional) The security configuration name to attach to the EMR cluster. Only valid for EMR clusters with `release_label` 4.8.0 or greater
- `core_instance_group` - (Optional) Configuration block to use an Instance Group for the core node type. Cannot be specified if `core_instance_count` argument, `core_instance_type` argument, or `instance_group` configuration blocks are set. Detailed below.
- `core_instance_type` - (Optional, **DEPRECATED**) Use the `core_instance_group` configuration block `instance_type` argument instead. The EC2 instance type of the slave nodes. Cannot be specified if `core_instance_group` or `instance_group` configuration blocks are set.
- `core_instance_count` - (Optional, **DEPRECATED**) Use the `core_instance_group` configuration block `instance_count` argument instead. Number of Amazon EC2 instances used to execute the job flow. EMR will use one node as the cluster's master node and use the remainder of the nodes (`core_instance_count`-1) as core nodes. Cannot be specified if `core_instance_group` or `instance_group` configuration blocks are set. Default `1`
- `instance_group` - (Optional, **DEPRECATED**) Use the `master_instance_group` configuration block, `core_instance_group` configuration block and `aws_emr_instance_group` resource(s) instead. A list of `instance_group` objects for each instance group in the cluster. Exactly one of `master_instance_type` and `instance_group` must be specified. If `instance_group` is set, then it must contain a configuration block for

at least the MASTER instance group type (as well as any additional instance groups). Cannot be specified if master_instance_group or core_instance_group configuration blocks are set. Defined below

- log_uri - (Optional) S3 bucket to write the log files of the job flow. If a value is not provided, logs are not created
- applications - (Optional) A list of applications for the cluster. Valid values are: Flink, Hadoop, Hive, Mahout, Pig, Spark, and JupyterHub (as of EMR 5.14.0). Case insensitive
- termination_protection - (Optional) Switch on/off termination protection (default is false, except when using multiple master nodes). Before attempting to destroy the resource when termination protection is enabled, this configuration must be applied with its value set to false.
- keep_job_flow_alive_when_no_steps - (Optional) Switch on/off run cluster with no steps or when all steps are complete (default is on)
- ec2_attributes - (Optional) Attributes for the EC2 instances running the job flow. Defined below
- kerberos_attributes - (Optional) Kerberos configuration for the cluster. Defined below
- ebs_root_volume_size - (Optional) Size in GiB of the EBS root device volume of the Linux AMI that is used for each EC2 instance. Available in Amazon EMR version 4.x and later.
- custom_ami_id - (Optional) A custom Amazon Linux AMI for the cluster (instead of an EMR-owned AMI). Available in Amazon EMR version 5.7.0 and later.
- bootstrap_action - (Optional) List of bootstrap actions that will be run before Hadoop is started on the cluster nodes. Defined below
- configurations - (Optional) List of configurations supplied for the EMR cluster you are creating
- configurations_json - (Optional) A JSON string for supplying list of configurations for the EMR cluster.

NOTE on configurations_json: If the Configurations value is empty then you should skip the Configurations field instead of providing empty list as value "Configurations": [].

```
configurations_json = <<EOF
  [
    {
      "Classification": "hadoop-env",
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "JAVA_HOME": "/usr/lib/jvm/java-1.8.0"
          }
        }
      ],
      "Properties": {}
    }
  ]
EOF
```

- `visible_to_all_users` - (Optional) Whether the job flow is visible to all IAM users of the AWS account associated with the job flow. Default `true`
- `autoscaling_role` - (Optional) An IAM role for automatic scaling policies. The IAM role provides permissions that the automatic scaling feature requires to launch and terminate EC2 instances in an instance group.
- `step` - (Optional) List of steps to run when creating the cluster. Defined below. It is highly recommended to utilize the lifecycle configuration block with `ignore_changes` if other steps are being managed outside of Terraform. This argument is processed in attribute-as-blocks mode.
- `step_concurrency_level` - (Optional) The number of steps that can be executed concurrently. You can specify a maximum of 256 steps. Only valid for EMR clusters with `release_label` 5.28.0 or greater. (default is 1)
- `tags` - (Optional) list of tags to apply to the EMR Cluster

» core_instance_group Configuration Block

Supported arguments for the `core_instance_group` configuration block:

- `instance_type` - (Required) EC2 instance type for all instances in the instance group.
- `autoscaling_policy` - (Optional) String containing the EMR Auto Scaling Policy JSON.
- `bid_price` - (Optional) Bid price for each EC2 instance in the instance group, expressed in USD. By setting this attribute, the instance group is being declared as a Spot Instance, and will implicitly create a Spot request. Leave this blank to use On-Demand Instances.
- `ebs_config` - (Optional) Configuration block(s) for EBS volumes attached to each instance in the instance group. Detailed below.
- `instance_count` - (Optional) Target number of instances for the instance group. Must be at least 1. Defaults to 1.
- `name` - (Optional) Friendly name given to the instance group.

» ec2_attributes

Attributes for the Amazon EC2 instances running the job flow

- `key_name` - (Optional) Amazon EC2 key pair that can be used to ssh to the master node as the user called `hadoop`

- subnet_id - (Optional) VPC subnet id where you want the job flow to launch. Cannot specify the cc1.4xlarge instance type for nodes of a job flow launched in a Amazon VPC
- additional_master_security_groups - (Optional) String containing a comma separated list of additional Amazon EC2 security group IDs for the master node
- additional_slave_security_groups - (Optional) String containing a comma separated list of additional Amazon EC2 security group IDs for the slave nodes as a comma separated string
- emr_managed_master_security_group - (Optional) Identifier of the Amazon EC2 EMR-Managed security group for the master node
- emr_managed_slave_security_group - (Optional) Identifier of the Amazon EC2 EMR-Managed security group for the slave nodes
- service_access_security_group - (Optional) Identifier of the Amazon EC2 service-access security group - required when the cluster runs on a private subnet
- instance_profile - (Required) Instance Profile for EC2 instances of the cluster assume this role

NOTE on EMR-Managed security groups: These security groups will have any missing inbound or outbound access rules added and maintained by AWS, to ensure proper communication between instances in a cluster. The EMR service will maintain these rules for groups provided in emr_managed_master_security_group and emr_managed_slave_security_group; attempts to remove the required rules may succeed, only for the EMR service to re-add them in a matter of minutes. This may cause Terraform to fail to destroy an environment that contains an EMR cluster, because the EMR service does not revoke rules added on deletion, leaving a cyclic dependency between the security groups that prevents their deletion. To avoid this, use the revoke_rules_on_delete optional attribute for any Security Group used in emr_managed_master_security_group and emr_managed_slave_security_group. See Amazon EMR-Managed Security Groups for more information about the EMR-managed security group rules.

» kerberos_attributes

Attributes for Kerberos configuration

- ad_domain_join_password - (Optional) The Active Directory password for ad_domain_join_user. Terraform cannot perform drift detection of this configuration.
- ad_domain_join_user - (Optional) Required only when establishing a cross-realm trust with an Active Directory domain. A user with sufficient privileges to join resources to the domain. Terraform cannot perform drift detection of this configuration.

- `cross_realm_trust_principal_password` - (Optional) Required only when establishing a cross-realm trust with a KDC in a different realm. The cross-realm principal password, which must be identical across realms. Terraform cannot perform drift detection of this configuration.
- `kdc_admin_password` - (Required) The password used within the cluster for the kadmin service on the cluster-dedicated KDC, which maintains Kerberos principals, password policies, and keytabs for the cluster. Terraform cannot perform drift detection of this configuration.
- `realm` - (Required) The name of the Kerberos realm to which all nodes in a cluster belong. For example, `EC2.INTERNAL`

» instance_group

Attributes for each task instance group in the cluster

- `instance_role` - (Required) The role of the instance group in the cluster. Valid values are: `MASTER`, `CORE`, and `TASK`.
- `instance_type` - (Required) The EC2 instance type for all instances in the instance group
- `instance_count` - (Optional) Target number of instances for the instance group
- `name` - (Optional) Friendly name given to the instance group
- `bid_price` - (Optional) If set, the bid price for each EC2 instance in the instance group, expressed in USD. By setting this attribute, the instance group is being declared as a Spot Instance, and will implicitly create a Spot request. Leave this blank to use On-Demand Instances.
- `ebs_config` - (Optional) A list of attributes for the EBS volumes attached to each instance in the instance group. Each `ebs_config` defined will result in additional EBS volumes being attached to *each* instance in the instance group. Defined below
- `autoscaling_policy` - (Optional) The autoscaling policy document. This is a JSON formatted string. See EMR Auto Scaling

» master_instance_group Configuration Block

Supported nested arguments for the `master_instance_group` configuration block:

- `instance_type` - (Required) EC2 instance type for all instances in the instance group.
- `bid_price` - (Optional) Bid price for each EC2 instance in the instance group, expressed in USD. By setting this attribute, the instance group is being declared as a Spot Instance, and will implicitly create a Spot request. Leave this blank to use On-Demand Instances.

- `ebs_config` - (Optional) Configuration block(s) for EBS volumes attached to each instance in the instance group. Detailed below.
- `instance_count` - (Optional) Target number of instances for the instance group. Must be 1 or 3. Defaults to 1. Launching with multiple master nodes is only supported in EMR version 5.23.0+, and requires this resource's `core_instance_group` to be configured. Public (Internet accessible) instances must be created in VPC subnets that have map public IP on launch enabled. Termination protection is automatically enabled when launched with multiple master nodes and Terraform must have the `termination_protection = false` configuration applied before destroying this resource.
- `name` - (Optional) Friendly name given to the instance group.

» ebs_config

Attributes for the EBS volumes attached to each EC2 instance in the `instance_group`

- `size` - (Required) The volume size, in gibibytes (GiB).
- `type` - (Required) The volume type. Valid options are `gp2`, `io1`, `standard` and `st1`. See EBS Volume Types.
- `iops` - (Optional) The number of I/O operations per second (IOPS) that the volume supports
- `volumes_per_instance` - (Optional) The number of EBS volumes with this configuration to attach to each EC2 instance in the instance group (default is 1)

» bootstrap_action

- `name` - (Required) Name of the bootstrap action
- `path` - (Required) Location of the script to run during a bootstrap action. Can be either a location in Amazon S3 or on a local file system
- `args` - (Optional) List of command line arguments to pass to the bootstrap action script

» step

Attributes for step configuration

- `action_on_failure` - (Required) The action to take if the step fails. Valid values: `TERMINATE_JOB_FLOW`, `TERMINATE_CLUSTER`, `CANCEL_AND_WAIT`, and `CONTINUE`
- `hadoop_jar_step` - (Required) The JAR file used for the step. Defined below.

- `name` - (Required) The name of the step.

» hadoop_jar_step

Attributes for Hadoop job step configuration

- `args` - (Optional) List of command line arguments passed to the JAR file's main function when executed.
- `jar` - (Required) Path to a JAR file run during the step.
- `main_class` - (Optional) Name of the main class in the specified Java file. If not specified, the JAR file should specify a Main-Class in its manifest file.
- `properties` - (Optional) Key-Value map of Java properties that are set when the step runs. You can use these properties to pass key value pairs to your main function.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn`- The ARN of the cluster.
- `id` - The ID of the EMR Cluster
- `name` - The name of the cluster.
- `release_label` - The release label for the Amazon EMR release.
- `master_instance_group.0.id` - Master node type Instance Group ID, if using Instance Group for this node type.
- `master_public_dns` - The public DNS name of the master EC2 instance.
- `core_instance_group.0.id` - Core node type Instance Group ID, if using Instance Group for this node type.
- `log_uri` - The path to the Amazon S3 location where logs for this cluster are stored.
- `applications` - The applications installed on this cluster.
- `ec2_attributes` - Provides information about the EC2 instances in a cluster grouped by category: key name, subnet ID, IAM instance profile, and so on.
- `bootstrap_action` - A list of bootstrap actions that will be run before Hadoop is started on the cluster nodes.
- `configurations` - The list of Configurations supplied to the EMR cluster.
- `service_role` - The IAM role that will be assumed by the Amazon EMR service to access AWS resources on your behalf.
- `visible_to_all_users` - Indicates whether the job flow is visible to all IAM users of the AWS account associated with the job flow.
- `tags` - The list of tags associated with a cluster.

» Example bootable config

NOTE: This configuration demonstrates a minimal configuration needed to boot an example EMR Cluster. It is not meant to display best practices. Please use at your own risk.

```
provider "aws" {
  region = "us-west-2"
}

resource "aws_emr_cluster" "cluster" {
  name          = "emr-test-arn"
  release_label = "emr-4.6.0"
  applications  = ["Spark"]

  ec2_attributes {
    subnet_id                         = "${aws_subnet.main.id}"
    emr_managed_master_security_group = "${aws_security_group.allow_all.id}"
    emr_managed_slave_security_group  = "${aws_security_group.allow_all.id}"
    instance_profile                  = "${aws_iam_instance_profile.emr_profile.arn}"
  }

  master_instance_type = "m5.xlarge"
  core_instance_type   = "m5.xlarge"
  core_instance_count  = 1

  tags = {
    role     = "rolename"
    dns_zone = "env_zone"
    env      = "env"
    name     = "name-env"
  }

  bootstrap_action {
    path = "s3://elasticmapreduce/bootstrap-actions/run-if"
    name = "runif"
    args = ["instance.isMaster=true", "echo running on master node"]
  }

  configurations_json = <<EOF
  [
    {
      "Classification": "hadoop-env",
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "JAVA_HOME": "/usr/lib/jvm/java-1.8.0"
          }
        }
      ],
      "Properties": {}
    },
    {
      "Classification": "spark-env",
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "JAVA_HOME": "/usr/lib/jvm/java-1.8.0"
          }
        }
      ],
      "Properties": {}
    }
  ]
EOF

  service_role = "${aws_iam_role.iam_emr_service_role.arn}"
}

resource "aws_security_group" "allow_access" {
  name        = "allow_access"
  description = "Allow inbound traffic"
  vpc_id      = "${aws_vpc.main.id}"

  ingress {
    # these ports should be locked down
    from_port   = 0
    to_port     = 0
    protocol    = "-1"

    # we do not recommend opening your cluster to 0.0.0.0/0
    cidr_blocks = # add your IP address here
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
  depends_on = ["aws_subnet.main"]

  lifecycle {
    ignore_changes = ["ingress", "egress"]
  }

  tags = {
    name = "emr_test"
  }
}

resource "aws_vpc" "main" {
  cidr_block           = "168.31.0.0/16"
  enable_dns_hostnames = true

  tags = {
    name = "emr_test"
  }
}

resource "aws_subnet" "main" {
  vpc_id     = "${aws_vpc.main.id}"
  cidr_block = "168.31.0.0/20"

  tags = {
    name = "emr_test"
  }
}

resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.main.id}"
}

resource "aws_route_table" "r" {
  vpc_id = "${aws_vpc.main.id}"

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = "${aws_internet_gateway.gw.id}"
  }
}

resource "aws_main_route_table_association" "a" {
  vpc_id         = "${aws_vpc.main.id}"
  route_table_id = "${aws_route_table.r.id}"
```

```
}

###

# IAM Role setups

###

# IAM role for EMR Service
resource "aws_iam_role" "iam_emr_service_role" {
  name = "iam_emr_service_role"

  assume_role_policy = <<EOF
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticmapreduce.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "iam_emr_service_policy" {
  name = "iam_emr_service_policy"
  role = "${aws_iam_role.iam_emr_service_role.id}"

  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Resource": "*",
        "Action": [
            "ec2:AuthorizeSecurityGroupEgress",
            "ec2:AuthorizeSecurityGroupIngress",
            "ec2:CancelSpotInstanceRequests",
            "ec2:CreateNetworkInterface",
            "ec2:CreateSecurityGroup",
            "ec2:CreateTags",
            "ec2:DeleteNetworkInterface",
            "ec2:DeleteSecurityGroup",
            "ec2:DeleteTags",
            "ec2:DescribeAvailabilityZones",
            "ec2:DescribeAccountAttributes",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeInstanceStatus",
            "ec2:DescribeInstances",
            "ec2:DescribeKeyPairs",
            "ec2:DescribeNetworkAcls",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribePrefixLists",
            "ec2:DescribeRouteTables",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSpotInstanceRequests",
            "ec2:DescribeSpotPriceHistory",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcAttribute",
            "ec2:DescribeVpcEndpoints",
            "ec2:DescribeVpcEndpointServices",
            "ec2:DescribeVpcs",
            "ec2:DetachNetworkInterface",
            "ec2:ModifyImageAttribute",
            "ec2:ModifyInstanceAttribute",
            "ec2:RequestSpotInstances",
            "ec2:RevokeSecurityGroupEgress",
            "ec2:RunInstances",
            "ec2:TerminateInstances",
            "ec2:DeleteVolume",
            "ec2:DescribeVolumeStatus",
            "ec2:DescribeVolumes",
            "ec2:DetachVolume",
            "iam:GetRole",
            "iam:GetRolePolicy",
            "iam:ListInstanceProfiles",
            "iam:ListRolePolicies",
            "iam:PassRole",
            "s3:CreateBucket",
            "s3:Get*",
            "s3:List*",
            "sdb:BatchPutAttributes",
            "sdb:Select",
            "sqs:CreateQueue",
            "sqs:Delete*",
            "sqs:GetQueue*",
            "sqs:PurgeQueue",
            "sqs:ReceiveMessage"
        ]
    }]
}
EOF
}

# IAM Role for EC2 Instance Profile
resource "aws_iam_role" "iam_emr_profile_role" {
  name = "iam_emr_profile_role"

  assume_role_policy = <<EOF
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_instance_profile" "emr_profile" {
  name  = "emr_profile"
  roles = ["${aws_iam_role.iam_emr_profile_role.name}"]
}

resource "aws_iam_role_policy" "iam_emr_profile_policy" {
  name = "iam_emr_profile_policy"
  role = "${aws_iam_role.iam_emr_profile_role.id}"

  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Resource": "*",
        "Action": [
            "cloudwatch:*",
            "dynamodb:*",
            "ec2:Describe*",
            "elasticmapreduce:Describe*",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListInstances",
            "elasticmapreduce:ListSteps",
            "kinesis:CreateStream",
            "kinesis:DeleteStream",
            "kinesis:DescribeStream",
            "kinesis:GetRecords",
            "kinesis:GetShardIterator",
            "kinesis:MergeShards",
            "kinesis:PutRecord",
            "kinesis:SplitShard",
            "rds:Describe*",
            "s3:*",
            "sdb:*",
            "sns:*",
            "sqs:*"
        ]
    }]
}
EOF
}
```

» Import

EMR clusters can be imported using the `id`, e.g.

```
$ terraform import aws_emr_cluster.cluster j-123456ABCDEF
```

Since the API does not return the actual values for Kerberos configurations, environments with those Terraform configurations will need to use the `lifecycle` configuration block `ignore_changes` argument available to all Terraform resources to prevent perpetual differences, e.g.

```
resource "aws_emr_cluster" "example" {
  # ... other configuration ...

  lifecycle {
    ignore_changes = ["kerberos_attributes"]
  }
}
```

» Resource: aws_emr_instance_group

Provides an Elastic MapReduce Cluster Instance Group configuration. See Amazon Elastic MapReduce Documentation for more information.

NOTE: At this time, Instance Groups cannot be destroyed through the API nor web interface. Instance Groups are destroyed when the EMR Cluster is destroyed. Terraform will resize any Instance Group to zero when destroying the resource.

» Example Usage

```
resource "aws_emr_instance_group" "task" {
  cluster_id     = "${aws_emr_cluster.tf-test-cluster.id}"
  instance_count = 1
  instance_type  = "m5.xlarge"
  name           = "my little instance group"
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) Human friendly name given to the instance group. Changing this forces a new resource to be created.
- `cluster_id` (Required) ID of the EMR Cluster to attach to. Changing this forces a new resource to be created.
- `instance_type` (Required) The EC2 instance type for all instances in the instance group. Changing this forces a new resource to be created.
- `instance_count` (optional) target number of instances for the instance group. defaults to 0.
- `bid_price` - (Optional) If set, the bid price for each EC2 instance in the instance group, expressed in USD. By setting this attribute, the instance group is being declared as a Spot Instance, and will implicitly create a Spot request. Leave this blank to use On-Demand Instances.
- `ebs_optimized` (Optional) Indicates whether an Amazon EBS volume is EBS-optimized. Changing this forces a new resource to be created.
- `ebs_config` (Optional) One or more `ebs_config` blocks as defined below. Changing this forces a new resource to be created.
- `autoscaling_policy` - (Optional) The autoscaling policy document. This is a JSON formatted string. See EMR Auto Scaling
- `configurations_json` - (Optional) A JSON string for supplying list of configurations specific to the EMR instance group. Note that this can only be changed when using EMR release 5.21 or later.

```
configurations_json = <<EOF
  [
    {
      "Classification": "hadoop-env",
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "JAVA_HOME": "/usr/lib/jvm/java-1.8.0"
          }
        }
      ],
      "Properties": {}
    }
  ]
EOF
```

`ebs_config` supports the following:

- `iops` - (Optional) The number of I/O operations per second (IOPS) that the volume supports.
- `size` - (Optional) The volume size, in gibibytes (GiB). This can be a number from 1 - 1024. If the volume type is EBS-optimized, the minimum value is 10.
- `type` - (Optional) The volume type. Valid options are 'gp2', 'io1' and 'standard'.
- `volumes_per_instance` - (Optional) The number of EBS Volumes to attach per instance.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The EMR Instance ID
- `running_instance_count` The number of instances currently running in this instance group.
- `status` The current status of the instance group.

» Import

EMR task instance group can be imported using their EMR Cluster id and Instance Group id separated by a forward-slash `/`, e.g.

```
$ terraform import aws_emr_instance_group.task_greoup j-123456ABCDEF/ig-15EK4O09RZLNR
```

» Resource: aws_emr_security_configuration

Provides a resource to manage AWS EMR Security Configurations

» Example Usage

```
resource "aws_emr_security_configuration" "foo" {
  name = "emrsc_other"

  configuration = <<EOF
{
  "EncryptionConfiguration": {
    "AtRestEncryptionConfiguration": {
      "S3EncryptionConfiguration": {
        "EncryptionMode": "SSE-S3"
      },
      "LocalDiskEncryptionConfiguration": {
        "EncryptionKeyProviderType": "AwsKms",
        "AwsKmsKey": "arn:aws:kms:us-west-2:187416307283:alias/tf_emr_test_key"
      }
    },
    "EnableInTransitEncryption": false,
    "EnableAtRestEncryption": true
  }
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the EMR Security Configuration. By default generated by Terraform.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `configuration` - (Required) A JSON formatted Security Configuration

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the EMR Security Configuration (Same as the `name`)
- `name` - The Name of the EMR Security Configuration

- `configuration` - The JSON formatted Security Configuration
- `creation_date` - Date the Security Configuration was created

» Import

EMR Security Configurations can be imported using the `name`, e.g.

```
$ terraform import aws_emr_security_configuration.sc example-sc-name
```

» Data Source: aws_elasticsearch_domain

Use this data source to get information about an Elasticsearch Domain

» Example Usage

```
data "aws_elasticsearch_domain" "my_domain" {
  domain_name = "my-domain-name"
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` – (Required) Name of the domain.

» Attributes Reference

The following attributes are exported:

- `access_policies` – The policy document attached to the domain.
- `advanced_options` - Key-value string pairs to specify advanced configuration options.
- `arn` – The Amazon Resource Name (ARN) of the domain.
- `cluster_config` - Cluster configuration of the domain.
  - `instance_type` - Instance type of data nodes in the cluster.
  - `instance_count` - Number of instances in the cluster.
  - `dedicated_master_enabled` - Indicates whether dedicated master nodes are enabled for the cluster.
  - `dedicated_master_type` - Instance type of the dedicated master nodes in the cluster.
  - `dedicated_master_count` - Number of dedicated master nodes in the cluster.

    - `zone_awareness_enabled` - Indicates whether zone awareness is enabled.
    - `zone_awareness_config` - Configuration block containing zone awareness settings.
        - `availability_zone_count` - Number of availability zones used.
- `cognito_options` - Domain Amazon Cognito Authentication options for Kibana.
    - `enabled` - Whether Amazon Cognito Authentication is enabled.
    - `user_pool_id` - The Cognito User pool used by the domain.
    - `identity_pool_id` - The Cognito Identity pool used by the domain.
    - `role_arn` - The IAM Role with the AmazonESCognitoAccess policy attached.
- `created` – Status of the creation of the domain.
- `deleted` – Status of the deletion of the domain.
- `domain_id` – Unique identifier for the domain.
- `ebs_options` - EBS Options for the instances in the domain.
    - `ebs_enabled` - Whether EBS volumes are attached to data nodes in the domain.
    - `volume_type` - The type of EBS volumes attached to data nodes.
    - `volume_size` - The size of EBS volumes attached to data nodes (in GB).
    - `iops` - The baseline input/output (I/O) performance of EBS volumes attached to data nodes.
- `elasticsearch_version` – ElasticSearch version for the domain.
- `encryption_at_rest` - Domain encryption at rest related options.
    - `enabled` - Whether encryption at rest is enabled in the domain.
    - `kms_key_id` - The KMS key id used to encrypt data at rest.
- `endpoint` – Domain-specific endpoint used to submit index, search, and data upload requests.
- `kibana_endpoint` - Domain-specific endpoint used to access the Kibana application.
- `log_publishing_options` - Domain log publishing related options.
    - `log_type` - The type of Elasticsearch log being published.
    - `cloudwatch_log_group_arn` - The CloudWatch Log Group where the logs are published.
    - `enabled` - Whether log publishing is enabled.
- `node_to_node_encryption` - Domain in transit encryption related options.
    - `enabled` - Whether node to node encryption is enabled.
- `processing` – Status of a configuration change in the domain.
- `snapshot_options` – Domain snapshot related options.
    - `automated_snapshot_start_hour` - Hour during which the service takes an automated daily snapshot of the indices in the domain.
- `tags` - The tags assigned to the domain.
- `vpc_options` - VPC Options for private Elasticsearch domains.
    - `availability_zones` - The availability zones used by the domain.

  - `security_group_ids` - The security groups used by the domain.
  - `subnet_ids` - The subnets used by the domain.
  - `vpc_id` - The VPC used by the domain.

» Resource: aws_elasticsearch_domain

Manages an AWS Elasticsearch Domain.

» Example Usage

» Basic Usage

```
resource "aws_elasticsearch_domain" "example" {
  domain_name           = "example"
  elasticsearch_version = "1.5"

  cluster_config {
    instance_type = "r4.large.elasticsearch"
  }

  snapshot_options {
    automated_snapshot_start_hour = 23
  }

  tags = {
    Domain = "TestDomain"
  }
}
```

» Access Policy

See also: `aws_elasticsearch_domain_policy` resource

```
variable "domain" {
  default = "tf-test"
}

data "aws_region" "current" {}

data "aws_caller_identity" "current" {}

resource "aws_elasticsearch_domain" "example" {
  domain_name = "${var.domain}"
  # ... other configuration ...

  access_policies = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "es:*",
      "Principal": "*",
      "Effect": "Allow",
      "Resource": "arn:aws:es:${data.aws_region.current.name}:${data.aws_caller_identity.cur
      "Condition": {
        "IpAddress": {"aws:SourceIp": ["66.193.100.22/32"]}
      }
    }
  ]
}
POLICY
}
```

» Log Publishing to CloudWatch Logs

```
resource "aws_cloudwatch_log_group" "example" {
  name = "example"
}

resource "aws_cloudwatch_log_resource_policy" "example" {
  policy_name = "example"

  policy_document = <<CONFIG
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:PutLogEventsBatch",
        "logs:CreateLogStream"
      ],
      "Resource": "arn:aws:logs:*"
    }
  ]
}
CONFIG
}

resource "aws_elasticsearch_domain" "example" {
  # .. other configuration ...

  log_publishing_options {
    cloudwatch_log_group_arn = "${aws_cloudwatch_log_group.example.arn}"
    log_type                 = "INDEX_SLOW_LOGS"
  }
}
```

» VPC based ES

```
variable "vpc" {}

variable "domain" {
  default = "tf-test"
}

data "aws_vpc" "selected" {
  tags = {
    Name = "${var.vpc}"
  }
}

data "aws_subnet_ids" "selected" {
  vpc_id = "${data.aws_vpc.selected.id}"

  tags = {
    Tier = "private"
  }
}

data "aws_region" "current" {}

data "aws_caller_identity" "current" {}

resource "aws_security_group" "es" {
  name        = "${var.vpc}-elasticsearch-${var.domain}"
  description = "Managed by Terraform"
  vpc_id      = "${data.aws_vpc.selected.id}"
  ingress {
    from_port = 443
    to_port   = 443
    protocol  = "tcp"

    cidr_blocks = [
      "${data.aws_vpc.selected.cidr_block}",
    ]
  }
}

resource "aws_iam_service_linked_role" "es" {
  aws_service_name = "es.amazonaws.com"
}

resource "aws_elasticsearch_domain" "es" {
  domain_name           = "${var.domain}"
  elasticsearch_version = "6.3"

  cluster_config {
    instance_type = "m4.large.elasticsearch"
  }

  vpc_options {
    subnet_ids = [
      "${data.aws_subnet_ids.selected.ids[0]}",
      "${data.aws_subnet_ids.selected.ids[1]}",
    ]

    security_group_ids = ["${aws_security_group.elasticsearch.id}"]
  }

  advanced_options = {
    "rest.action.multi.allow_explicit_index" = "true"
  }

  access_policies = <<CONFIG
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "es:*",
            "Principal": "*",
            "Effect": "Allow",
            "Resource": "arn:aws:es:${data.aws_region.current.name}:${data.aws_caller_identi
        }
    ]
}
CONFIG

  snapshot_options {
    automated_snapshot_start_hour = 23
  }

  tags = {
    Domain = "TestDomain"
  }

  depends_on = [
    "aws_iam_service_linked_role.es",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Required) Name of the domain.
- `access_policies` - (Optional) IAM policy document specifying the access policies for the domain
- `advanced_options` - (Optional) Key-value string pairs to specify advanced configuration options. Note that the values for these configuration options must be strings (wrapped in quotes) or they may be wrong and cause a perpetual diff, causing Terraform to want to recreate your Elasticsearch domain on every apply.
- `ebs_options` - (Optional) EBS related options, may be required based on chosen instance size. See below.
- `encrypt_at_rest` - (Optional) Encrypt at rest options. Only available for certain instance types. See below.
- `node_to_node_encryption` - (Optional) Node-to-node encryption options. See below.
- `cluster_config` - (Optional) Cluster configuration of the domain, see below.
- `snapshot_options` - (Optional) Snapshot related options, see below.
- `vpc_options` - (Optional) VPC related options, see below. Adding or removing this configuration forces a new resource (documentation).
- `log_publishing_options` - (Optional) Options for publishing slow logs to CloudWatch Logs.
- `elasticsearch_version` - (Optional) The version of Elasticsearch to deploy. Defaults to `1.5`

- `tags` - (Optional) A mapping of tags to assign to the resource

ebs_options supports the following attributes:

- `ebs_enabled` - (Required) Whether EBS volumes are attached to data nodes in the domain.
- `volume_type` - (Optional) The type of EBS volumes attached to data nodes.
- `volume_size` - The size of EBS volumes attached to data nodes (in GB). **Required** if `ebs_enabled` is set to `true`.
- `iops` - (Optional) The baseline input/output (I/O) performance of EBS volumes attached to data nodes. Applicable only for the Provisioned IOPS EBS volume type.

encrypt_at_rest supports the following attributes:

- `enabled` - (Required) Whether to enable encryption at rest. If the `encrypt_at_rest` block is not provided then this defaults to `false`.
- `kms_key_id` - (Optional) The KMS key id to encrypt the Elasticsearch domain with. If not specified then it defaults to using the `aws/es` service KMS key.

cluster_config supports the following attributes:

- `instance_type` - (Optional) Instance type of data nodes in the cluster.
- `instance_count` - (Optional) Number of instances in the cluster.
- `dedicated_master_enabled` - (Optional) Indicates whether dedicated master nodes are enabled for the cluster.
- `dedicated_master_type` - (Optional) Instance type of the dedicated master nodes in the cluster.
- `dedicated_master_count` - (Optional) Number of dedicated master nodes in the cluster
- `zone_awareness_config` - (Optional) Configuration block containing zone awareness settings. Documented below.
- `zone_awareness_enabled` - (Optional) Indicates whether zone awareness is enabled. To enable awareness with three Availability Zones, the `availability_zone_count` within the `zone_awareness_config` must be set to `3`.

zone_awareness_config supports the following attributes:

- `availability_zone_count` - (Optional) Number of Availability Zones for the domain to use with `zone_awareness_enabled`. Defaults to `2`. Valid values: `2` or `3`.

node_to_node_encryption supports the following attributes:

- `enabled` - (Required) Whether to enable node-to-node encryption. If the `node_to_node_encryption` block is not provided then this defaults to `false`.

vpc_options supports the following attributes:

AWS documentation: VPC Support for Amazon Elasticsearch Service Domains

Note you must have created the service linked role for the Elasticsearch service to use the `vpc_options`. If you need to create the service linked role at the same time as the Elasticsearch domain then you must use `depends_on` to make sure that the role is created before the Elasticsearch domain. See the VPC based ES domain example above.

- `security_group_ids` - (Optional) List of VPC Security Group IDs to be applied to the Elasticsearch domain endpoints. If omitted, the default Security Group for the VPC will be used.
- `subnet_ids` - (Required) List of VPC Subnet IDs for the Elasticsearch domain endpoints to be created in.

Security Groups and Subnets referenced in these attributes must all be within the same VPC; this determines what VPC the endpoints are created in.

snapshot_options supports the following attribute:

- `automated_snapshot_start_hour` - (Required) Hour during which the service takes an automated daily snapshot of the indices in the domain.

log_publishing_options supports the following attribute:

- `log_type` - (Required) A type of Elasticsearch log. Valid values: INDEX_SLOW_LOGS, SEARCH_SLOW_LOGS, ES_APPLICATION_LOGS
- `cloudwatch_log_group_arn` - (Required) ARN of the Cloudwatch log group to which log needs to be published.
- `enabled` - (Optional, Default: true) Specifies whether given log publishing option is enabled or not.

cognito_options supports the following attribute:

AWS documentation: Amazon Cognito Authentication for Kibana

- `enabled` - (Optional, Default: false) Specifies whether Amazon Cognito authentication with Kibana is enabled or not
- `user_pool_id` - (Required) ID of the Cognito User Pool to use
- `identity_pool_id` - (Required) ID of the Cognito Identity Pool to use
- `role_arn` - (Required) ARN of the IAM role that has the AmazonESCognitoAccess policy attached

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the domain.
- `domain_id` - Unique identifier for the domain.
- `domain_name` - The name of the Elasticsearch domain.

- `endpoint` - Domain-specific endpoint used to submit index, search, and data upload requests.
- `kibana_endpoint` - Domain-specific endpoint for kibana without https scheme.
- `vpc_options.0.availability_zones` - If the domain was created inside a VPC, the names of the availability zones the configured `subnet_ids` were created inside.
- `vpc_options.0.vpc_id` - If the domain was created inside a VPC, the ID of the VPC.

» Import

Elasticsearch domains can be imported using the `domain_name`, e.g.

```
$ terraform import aws_elasticsearch_domain.example domain_name
```

» Resource: aws_elasticsearch_domain_policy

Allows setting policy to an Elasticsearch domain while referencing domain attributes (e.g. ARN)

» Example Usage

```
resource "aws_elasticsearch_domain" "example" {
  domain_name           = "tf-test"
  elasticsearch_version = "2.3"
}

resource "aws_elasticsearch_domain_policy" "main" {
  domain_name = "${aws_elasticsearch_domain.example.domain_name}"

  access_policies = <<POLICIES
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "es:*",
            "Principal": "*",
            "Effect": "Allow",
            "Condition": {
                "IpAddress": {"aws:SourceIp": "127.0.0.1/32"}
            },
            "Resource": "${aws_elasticsearch_domain.example.arn}/*"
        }
    ]
}
POLICIES
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Required) Name of the domain.
- `access_policies` - (Optional) IAM policy document specifying the access policies for the domain

» Resource: aws_elastictranscoder_pipeline

Provides an Elastic Transcoder pipeline resource.

» Example Usage

```
resource "aws_elastictranscoder_pipeline" "bar" {
  input_bucket = "${aws_s3_bucket.input_bucket.bucket}"
  name         = "aws_elastictranscoder_pipeline_tf_test_"
  role         = "${aws_iam_role.test_role.arn}"

  content_config {
    bucket        = "${aws_s3_bucket.content_bucket.bucket}"
    storage_class = "Standard"
  }

  thumbnail_config {
    bucket        = "${aws_s3_bucket.thumb_bucket.bucket}"
    storage_class = "Standard"
  }
}
```

» Argument Reference

See ”Create Pipeline” in the AWS docs for reference.

The following arguments are supported:

- `aws_kms_key_arn` - (Optional) The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.
- `content_config` - (Optional) The ContentConfig object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists. (documented below)
- `content_config_permissions` - (Optional) The permissions for the `content_config` object. (documented below)
- `input_bucket` - (Required) The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics that you want to use as watermarks.
- `name` - (Optional, Forces new resource) The name of the pipeline. Maximum 40 characters
- `notifications` - (Optional) The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify to report job status. (documented below)
- `output_bucket` - (Optional) The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files.
- `role` - (Required) The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.
- `thumbnail_config` - (Optional) The ThumbnailConfig object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files. (documented below)
- `thumbnail_config_permissions` - (Optional) The permissions for the `thumbnail_config` object. (documented below)

The `content_config` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, and the storage class that you want to assign to the files. If you specify values for `content_config`, you must also specify values for `thumbnail_config`. If you specify values for `content_config` and `thumbnail_config`, omit the `output_bucket` object.

The `content_config` object supports the following:

- `bucket` - The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.
- `storage_class` - The Amazon S3 storage class, Standard or ReducedRedundancy, that you want Elastic Transcoder to assign to the files and playlists that it stores in your Amazon S3 bucket.

The `content_config_permissions` object supports the following:

- `access` - The permission that you want to give to the AWS user that you specified in `content_config_permissions.grantee`
- `grantee` - The AWS user or group that you want to have access to transcoded files and playlists.
- `grantee_type` - Specify the type of value that appears in the `content_config_permissions.grantee` object. Valid values are

Canonical, Email or Group.

The notifications object supports the following:

- completed - The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job in this pipeline.
- error - The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition while processing a job in this pipeline.
- progressing - The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline.
- warning - The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline.

The thumbnail_config object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files. If you specify values for content_config, you must also specify values for thumbnail_config even if you don't want to create thumbnails. (You control whether to create thumbnails when you create a job. For more information, see ThumbnailPattern in the topic Create Job.) If you specify values for content_config and thumbnail_config, omit the OutputBucket object.

The thumbnail_config object supports the following:

- bucket - The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.
- storage_class - The Amazon S3 storage class, Standard or ReducedRedundancy, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket.

The thumbnail_config_permissions object supports the following:

- access - The permission that you want to give to the AWS user that you specified in thumbnail_config_permissions.grantee.
- grantee - The AWS user or group that you want to have access to thumbnail files.
- grantee_type - Specify the type of value that appears in the thumbnail_config_permissions.grantee object.

» Import

Elastic Transcoder pipelines can be imported using the id, e.g.

```
$ terraform import aws_elastictranscoder_pipeline.basic_pipeline 1407981661351-cttk8b
```

» Resource: aws_elastictranscoder_preset

Provides an Elastic Transcoder preset resource.

» Example Usage

```
resource "aws_elastictranscoder_preset" "bar" {
  container   = "mp4"
  description = "Sample Preset"
  name        = "sample_preset"

  audio {
    audio_packing_mode = "SingleTrack"
    bit_rate           = 96
    channels           = 2
    codec              = "AAC"
    sample_rate        = 44100
  }

  audio_codec_options {
    profile = "AAC-LC"
  }

  video {
    bit_rate             = "1600"
    codec                = "H.264"
    display_aspect_ratio = "16:9"
    fixed_gop            = "false"
    frame_rate           = "auto"
    max_frame_rate       = "60"
    keyframes_max_dist   = 240
    max_height           = "auto"
    max_width            = "auto"
    padding_policy       = "Pad"
    sizing_policy        = "Fit"
  }

  video_codec_options = {
    Profile                  = "main"
    Level                    = "2.2"
    MaxReferenceFrames       = 3
    InterlacedMode           = "Progressive"
    ColorSpaceConversionMode = "None"
  }

  video_watermarks {
    id                = "Terraform Test"
    max_width         = "20%"
    max_height        = "20%"
    sizing_policy     = "ShrinkToFit"
    horizontal_align  = "Right"
    horizontal_offset = "10px"
    vertical_align    = "Bottom"
    vertical_offset   = "10px"
    opacity           = "55.5"
    target            = "Content"
  }

  thumbnails {
    format         = "png"
    interval       = 120
    max_width      = "auto"
    max_height     = "auto"
    padding_policy = "Pad"
    sizing_policy  = "Fit"
  }
}
```

» Argument Reference

See "Create Preset" in the AWS docs for reference.

The following arguments are supported:

- `audio` - (Optional, Forces new resource) Audio parameters object (documented below).
- `audio_codec_options` - (Optional, Forces new resource) Codec options for the audio parameters (documented below)
- `container` - (Required, Forces new resource) The container type for the output file. Valid values are `flac`, `flv`, `fmp4`, `gif`, `mp3`, `mp4`, `mpg`, `mxf`, `oga`, `ogg`, `ts`, and `webm`.
- `description` - (Optional, Forces new resource) A description of the preset (maximum 255 characters)
- `name` - (Optional, Forces new resource) The name of the preset. (maximum 40 characters)
- `thumbnails` - (Optional, Forces new resource) Thumbnail parameters object (documented below)

- `video` - (Optional, Forces new resource) Video parameters object (documented below)
- `video_watermarks` - (Optional, Forces new resource) Watermark parameters for the video parameters (documented below)
- `video_codec_options` (Optional, Forces new resource) Codec options for the video parameters

The `audio` object supports the following:

- `audio_packing_mode` - The method of organizing audio channels and tracks. Use Audio:Channels to specify the number of channels in your output, and Audio:AudioPackingMode to specify the number of tracks and their relation to the channels. If you do not specify an Audio:AudioPackingMode, Elastic Transcoder uses SingleTrack.
- `bit_rate` - The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.
- `channels` - The number of audio channels in the output file
- `codec` - The audio codec for the output file. Valid values are `AAC`, `flac`, `mp2`, `mp3`, `pcm`, and `vorbis`.
- `sample_rate` - The sample rate of the audio stream in the output file, in hertz. Valid values are: `auto`, `22050`, `32000`, `44100`, `48000`, `96000`

The `audio_codec_options` object supports the following:

- `bit_depth` - The bit depth of a sample is how many bits of information are included in the audio samples. Valid values are `16` and `24`. (FLAC/PCM Only)
- `bit_order` - The order the bits of a PCM sample are stored in. The supported value is LittleEndian. (PCM Only)
- `profile` - If you specified AAC for Audio:Codec, choose the AAC profile for the output file.
- `signed` - Whether audio samples are represented with negative and positive numbers (signed) or only positive numbers (unsigned). The supported value is Signed. (PCM Only)

The `thumbnails` object supports the following:

- `aspect_ratio` - The aspect ratio of thumbnails. The following values are valid: auto, 1:1, 4:3, 3:2, 16:9
- `format` - The format of thumbnails, if any. Valid formats are jpg and png.
- `interval` - The approximate number of seconds between thumbnails. The value must be an integer. The actual interval can vary by several seconds from one thumbnail to the next.
- `max_height` - The maximum height of thumbnails, in pixels. If you specify auto, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.
- `max_width` - The maximum width of thumbnails, in pixels. If you specify auto, Elastic Transcoder uses 1920 (Full HD) as the default value. If

you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

- `padding_policy` - When you set PaddingPolicy to Pad, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail MaxWidth and MaxHeight settings.
- `resolution` - The width and height of thumbnail files in pixels, in the format WidthxHeight, where both values are even integers. The values cannot exceed the width and height that you specified in the Video:Resolution object. (To better control resolution and aspect ratio of thumbnails, we recommend that you use the thumbnail values `max_width`, `max_height`, `sizing_policy`, and `padding_policy` instead of `resolution` and `aspect_ratio`. The two groups of settings are mutually exclusive. Do not use them together)
- `sizing_policy` - A value that controls scaling of thumbnails. Valid values are: `Fit`, `Fill`, `Stretch`, `Keep`, `ShrinkToFit`, and `ShrinkToFill`.

The `video` object supports the following:

- `aspect_ratio` - The display aspect ratio of the video in the output file. Valid values are: `auto`, `1:1`, `4:3`, `3:2`, `16:9`. (Note; to better control resolution and aspect ratio of output videos, we recommend that you use the values `max_width`, `max_height`, `sizing_policy`, `padding_policy`, and `display_aspect_ratio` instead of `resolution` and `aspect_ratio`.)
- `bit_rate` - The bit rate of the video stream in the output file, in kilobits/second. You can configure variable bit rate or constant bit rate encoding.
- `codec` - The video codec for the output file. Valid values are `gif`, `H.264`, `mpeg2`, `vp8`, and `vp9`.
- `display_aspect_ratio` - The value that Elastic Transcoder adds to the metadata in the output file. If you set DisplayAspectRatio to auto, Elastic Transcoder chooses an aspect ratio that ensures square pixels. If you specify another option, Elastic Transcoder sets that value in the output file.
- `fixed_gop` - Whether to use a fixed value for Video:FixedGOP. Not applicable for containers of type gif. Valid values are true and false. Also known as, Fixed Number of Frames Between Keyframes.
- `frame_rate` - The frames per second for the video stream in the output file. The following values are valid: `auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`.
- `keyframes_max_dist` - The maximum number of frames between key frames. Not applicable for containers of type gif.
- `max_frame_rate` - If you specify auto for FrameRate, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video, up to the maximum frame rate. If you do not specify a MaxFrameRate, Elastic Transcoder will use a default of 30.

- `max_height` - The maximum height of the output video in pixels. If you specify auto, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072, inclusive.
- `max_width` - The maximum width of the output video in pixels. If you specify auto, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096, inclusive.
- `padding_policy` - When you set PaddingPolicy to Pad, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `max_width` and `max_height`.
- `resolution` - The width and height of the video in the output file, in pixels. Valid values are `auto` and `widthxheight`. (see note for `aspect_ratio`)
- `sizing_policy` - A value that controls scaling of the output video. Valid values are: `Fit`, `Fill`, `Stretch`, `Keep`, `ShrinkToFit`, `ShrinkToFill`.

The `video_watermarks` object supports the following:

- `horizontal_align` - The horizontal position of the watermark unless you specify a nonzero value for `horzontal_offset`.
- `horizontal_offset` - The amount by which you want the horizontal position of the watermark to be offset from the position specified by `horizontal_align`.
- `id` - A unique identifier for the settings for one watermark. The value of Id can be up to 40 characters long. You can specify settings for up to four watermarks.
- `max_height` - The maximum height of the watermark.
- `max_width` - The maximum width of the watermark.
- `opacity` - A percentage that indicates how much you want a watermark to obscure the video in the location where it appears.
- `sizing_policy` - A value that controls scaling of the watermark. Valid values are: `Fit`, `Stretch`, `ShrinkToFit`
- `target` - A value that determines how Elastic Transcoder interprets values that you specified for `video_watermarks.horizontal_offset`, `video_watermarks.vertical_offset`, `video_watermarks.max_width`, and `video_watermarks.max_height`. Valid values are `Content` and `Frame`.
- `vertical_align` - The vertical position of the watermark unless you specify a nonzero value for `vertical_align`. Valid values are `Top`, `Bottom`, `Center`.
- `vertical_offset` - The amount by which you want the vertical position of the watermark to be offset from the position specified by `vertical_align`

The `video_codec_options` map supports the following:

- `Profile` - The codec profile that you want to use for the output file. (H.264/VP8 Only)

- `Level` - The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels: `1`, `1b`, `1.1`, `1.2`, `1.3`, `2`, `2.1`, `2.2`, `3`, `3.1`, `3.2`, `4`, `4.1` (H.264 only)
- `MaxReferenceFrames` - The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16. (H.264 only)
- `MaxBitRate` - The maximum number of kilobits per second in the output video. Specify a value between 16 and 62,500 inclusive, or `auto`. (Optional, H.264/MPEG2/VP8/VP9 only)
- `BufferSize` - The maximum number of kilobits in any x seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using ts for the container type of the output video. Specify an integer greater than 0. If you specify MaxBitRate and omit BufferSize, Elastic Transcoder sets BufferSize to 10 times the value of MaxBitRate. (Optional, H.264/MPEG2/VP8/VP9 only)
- `InterlacedMode` - The interlace mode for the output video. (Optional, H.264/MPEG2 Only)
- `ColorSpaceConversion` - The color space conversion Elastic Transcoder applies to the output video. Valid values are `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`. (Optional, H.264/MPEG2 Only)
- `ChromaSubsampling` - The sampling pattern for the chroma (color) channels of the output video. Valid values are `yuv420p` and `yuv422p`.
- `LoopCount` - The number of times you want the output gif to loop (Gif only)

» Import

Elastic Transcoder presets can be imported using the `id`, e.g.

```
$ terraform import aws_elastictranscoder_preset.basic_preset 1407981661351-cttk8b
```

» Resource: aws_fms_admin_account

Provides a resource to associate/disassociate an AWS Firewall Manager administrator account. This operation must be performed in the `us-east-1` region.

» Example Usage

```
resource "aws_fms_admin_account" "example" {}
```

» Argument Reference

The following arguments are supported:

- `account_id` - (Optional) The AWS account ID to associate with AWS Firewall Manager as the AWS Firewall Manager administrator account. This can be an AWS Organizations master account or a member account. Defaults to the current account. Must be configured to perform drift detection.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The AWS account ID of the AWS Firewall Manager administrator account.

» Import

Firewall Manager administrator account association can be imported using the account ID, e.g.

```
$ terraform import aws_fms_admin_account.example 123456789012
```

» Resource: aws_fsx_lustre_file_system

Manages a FSx Lustre File System. See the FSx Lustre Guide for more information.

» Example Usage

```
resource "aws_fsx_lustre_file_system" "example" {
  import_path      = "s3://${aws_s3_bucket.example.bucket}"
  storage_capacity = 3600
  subnet_ids       = ["${aws_subnet.example.id}"]
}
```

» Argument Reference

The following arguments are supported:

- `storage_capacity` - (Required) The storage capacity (GiB) of the file system. Minimum of `3600`. Storage capacity is provisioned in increments of 3,600 GiB.
- `subnet_ids` - (Required) A list of IDs for the subnets that the file system will be accessible from. File systems currently support only one subnet. The file server is also launched in that subnet's Availability Zone.
- `export_path` - (Optional) S3 URI (with optional prefix) where the root of your Amazon FSx file system is exported. Can only be specified with `import_path` argument and the path must use the same Amazon S3 bucket as specified in `import_path`. Set equal to `import_path` to overwrite files on export. Defaults to `s3://{IMPORT BUCKET}/FSxLustre{CREATION TIMESTAMP}`.
- `import_path` - (Optional) S3 URI (with optional prefix) that you're using as the data repository for your FSx for Lustre file system. For example, `s3://example-bucket/optional-prefix/`.
- `imported_file_chunk_size` - (Optional) For files imported from a data repository, this value determines the stripe count and maximum amount of data per file (in MiB) stored on a single physical disk. Can only be specified with `import_path` argument. Defaults to `1024`. Minimum of `1` and maximum of `512000`.
- `security_group_ids` - (Optional) A list of IDs for the security groups that apply to the specified network interfaces created for file system access. These security groups will apply to all network interfaces.
- `tags` - (Optional) A mapping of tags to assign to the file system.
- `weekly_maintenance_start_time` - (Optional) The preferred start time (in `d:HH:MM` format) to perform weekly maintenance, in the UTC time zone.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name of the file system.
- `dns_name` - DNS name for the file system, e.g. `fs-12345678.fsx.us-west-2.amazonaws.com`
- `id` - Identifier of the file system, e.g. `fs-12345678`
- `network_interface_ids` - Set of Elastic Network Interface identifiers from which the file system is accessible.
- `owner_id` - AWS account identifier that created the file system.
- `vpc_id` - Identifier of the Virtual Private Cloud for the file system.

» Timeouts

`aws_fsx_lustre_file_system` provides the following Timeouts configuration options:

- `create` - (Default `30m`) How long to wait for the file system to be created.
- `delete` - (Default `30m`) How long to wait for the file system to be deleted.

» Import

FSx File Systems can be imported using the `id`, e.g.

```
$ terraform import aws_fsx_lustre_file_system.example fs-543ab12b1ca672f33
```

Certain resource arguments, like `security_group_ids`, do not have a FSx API method for reading the information after creation. If the argument is set in the Terraform configuration on an imported resource, Terraform will always show a difference. To workaround this behavior, either omit the argument from the Terraform configuration or use `ignore_changes` to hide the difference, e.g.

```
resource "aws_fsx_lustre_file_system" "example" {
  # ... other configuration ...
  security_group_ids = ["${aws_security_group.example.id}"]

  # There is no FSx API for reading security_group_ids
  lifecycle {
    ignore_changes = ["security_group_ids"]
  }
}
```

» Resource: aws_fsx_windows_file_system

Manages a FSx Windows File System. See the FSx Windows Guide for more information.

NOTE: Either the `active_directory_id` argument or `self_managed_active_directory` configuration block must be specified.

» Example Usage

» Using AWS Directory Service

Additional information for using AWS Directory Service with Windows File Systems can be found in the FSx Windows Guide.

```
resource "aws_fsx_windows_file_system" "example" {
  active_directory_id = "${aws_directory_service_directory.example.id}"
  kms_key_id          = "${aws_kms_key.example.arn}"
  storage_capacity    = 300
  subnet_ids          = ["${aws_subnet.example.id}"]
  throughput_capacity = 1024
}
```

» Using a Self-Managed Microsoft Active Directory

Additional information for using AWS Directory Service with Windows File Systems can be found in the FSx Windows Guide.

```
resource "aws_fsx_windows_file_system" "example" {
  kms_key_id          = "${aws_kms_key.example.arn}"
  storage_capacity    = 300
  subnet_ids          = ["${aws_subnet.example.id}"]
  throughput_capacity = 1024

  self_managed_active_directory {
    dns_ips     = ["10.0.0.111", "10.0.0.222"]
    domain_name = "corp.example.com"
    password    = "avoid-plaintext-passwords"
    username    = "Admin"
  }
}
```

» Argument Reference

The following arguments are supported:

- `storage_capacity` - (Required) Storage capacity (GiB) of the file system. Minimum of 32 and maximum of 65536.
- `subnet_ids` - (Required) A list of IDs for the subnets that the file system will be accessible from. File systems support only one subnet. The file server is also launched in that subnet's Availability Zone.
- `throughput_capacity` - (Required) Throughput (megabytes per second) of the file system in power of 2 increments. Minimum of `8` and maximum of `2048`.
- `active_directory_id` - (Optional) The ID for an existing Microsoft Active Directory instance that the file system should join when it's created. Cannot be specified with `self_managed_active_directory`.
- `automatic_backup_retention_days` - (Optional) The number of days to retain automatic backups. Minimum of `0` and maximum of `35`. Defaults to `7`. Set to `0` to disable.
- `copy_tags_to_backups` - (Optional) A boolean flag indicating whether tags on the file system should be copied to backups. Defaults to `false`.
- `daily_automatic_backup_start_time` - (Optional) The preferred time (in `HH:MM` format) to take daily automatic backups, in the UTC time zone.

- kms_key_id - (Optional) ARN for the KMS Key to encrypt the file system at rest. Defaults to an AWS managed KMS Key.
- security_group_ids - (Optional) A list of IDs for the security groups that apply to the specified network interfaces created for file system access. These security groups will apply to all network interfaces.
- self_managed_active_directory - (Optional) Configuration block that Amazon FSx uses to join the Windows File Server instance to your self-managed (including on-premises) Microsoft Active Directory (AD) directory. Cannot be specified with active_directory_id. Detailed below.
- skip_final_backup - (Optional) When enabled, will skip the default final backup taken when the file system is deleted. This configuration must be applied separately before attempting to delete the resource to have the desired behavior. Defaults to false.
- tags - (Optional) A mapping of tags to assign to the file system.
- weekly_maintenance_start_time - (Optional) The preferred start time (in d:HH:MM format) to perform weekly maintenance, in the UTC time zone.

» self_managed_active_directory

The following arguments are supported for self_managed_active_directory configuration block:

- dns_ips - (Required) A list of up to two IP addresses of DNS servers or domain controllers in the self-managed AD directory. The IP addresses need to be either in the same VPC CIDR range as the file system or in the private IP version 4 (IPv4) address ranges as specified in RFC 1918.
- domain_name - (Required) The fully qualified domain name of the self-managed AD directory. For example, corp.example.com.
- password - (Required) The password for the service account on your self-managed AD domain that Amazon FSx will use to join to your AD domain.
- username - (Required) The user name for the service account on your self-managed AD domain that Amazon FSx will use to join to your AD domain.
- file_system_administrators_group - (Optional) The name of the domain group whose members are granted administrative privileges for the file system. Administrative privileges include taking ownership of files and folders, and setting audit controls (audit ACLs) on files and folders. The group that you specify must already exist in your domain. Defaults to Domain Admins.
- organizational_unit_distinguished_name - (Optional) The fully qualified distinguished name of the organizational unit within your self-managed AD directory that the Windows File Server instance will join. For example, OU=FSx,DC=yourdomain,DC=corp,DC=com. Only accepts OU as the direct parent of the file system. If none is provided,

the FSx file system is created in the default location of your self-managed AD directory. To learn more, see RFC 2253.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name of the file system.
- `dns_name` - DNS name for the file system, e.g. `fs-12345678.corp.example.com` (domain name matching the Active Directory domain name)
- `id` - Identifier of the file system, e.g. `fs-12345678`
- `network_interface_ids` - Set of Elastic Network Interface identifiers from which the file system is accessible.
- `owner_id` - AWS account identifier that created the file system.
- `vpc_id` - Identifier of the Virtual Private Cloud for the file system.

» Timeouts

`aws_fsx_windows_file_system` provides the following Timeouts configuration options:

- `create` - (Default `30m`) How long to wait for the file system to be created.
- `delete` - (Default `30m`) How long to wait for the file system to be deleted.

» Import

FSx File Systems can be imported using the `id`, e.g.

```
$ terraform import aws_fsx_windows_file_system.example fs-543ab12b1ca672f33
```

Certain resource arguments, like `security_group_ids` and the `self_managed_active_directory` configuation block `password`, do not have a FSx API method for reading the information after creation. If these arguments are set in the Terraform configuration on an imported resource, Terraform will always show a difference. To workaround this behavior, either omit the argument from the Terraform configuration or use `ignore_changes` to hide the difference, e.g.

```
resource "aws_fsx_windows_file_system" "example" {
  # ... other configuration ...
  security_group_ids = ["${aws_security_group.example.id}"]

  # There is no FSx API for reading security_group_ids
  lifecycle {
    ignore_changes = ["security_group_ids"]
  }
}
```

» Resource: aws_gamelift_alias

Provides a Gamelift Alias resource.

» Example Usage

```
resource "aws_gamelift_alias" "example" {
  name        = "example-alias"
  description = "Example Description"

  routing_strategy {
    message = "Example Message"
    type    = "TERMINAL"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the alias.
- `description` - (Optional) Description of the alias.
- `routing_strategy` - (Required) Specifies the fleet and/or routing type to use for the alias.

» Nested Fields

» `routing_strategy`

- `fleet_id` - (Optional) ID of the Gamelift Fleet to point the alias to.
- `message` - (Optional) Message text to be used with the `TERMINAL` routing strategy.
- `type` - (Required) Type of routing strategy. e.g. `SIMPLE` or `TERMINAL`
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Alias ID.
- `arn` - Alias ARN.

» Import

Gamelift Aliases can be imported using the ID, e.g.

```
$ terraform import aws_gamelift_alias.example <alias-id>
```

» Resource: aws_gamelift_build

Provides an Gamelift Build resource.

» Example Usage

```
resource "aws_gamelift_build" "test" {
  name             = "example-build"
  operating_system = "WINDOWS_2012"

  storage_location {
    bucket   = "${aws_s3_bucket.test.bucket}"
    key      = "${aws_s3_bucket_object.test.key}"
    role_arn = "${aws_iam_role.test.arn}"
  }

  depends_on = ["aws_iam_role_policy.test"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the build
- `operating_system` - (Required) Operating system that the game server binaries are built to run on. e.g. `WINDOWS_2012` or `AMAZON_LINUX`.
- `storage_location` - (Required) Information indicating where your game build files are stored. See below.
- `version` - (Optional) Version that is associated with this build.
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Fields

» `storage_location`

- `bucket` - (Required) Name of your S3 bucket.
- `key` - (Required) Name of the zip file containing your build files.

- role_arn - (Required) ARN of the access role that allows Amazon GameLift to access your S3 bucket.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - Gamelift Build ID.
- arn - Gamelift Build ARN.

» Import

Gamelift Builds cannot be imported at this time.

» Resource: aws_gamelift_fleet

Provides a Gamelift Fleet resource.

» Example Usage

```
resource "aws_gamelift_fleet" "example" {
  build_id          = "${aws_gamelift_build.example.id}"
  ec2_instance_type = "t2.micro"
  name              = "example-fleet-name"

  runtime_configuration {
    server_process {
      concurrent_executions = 1
      launch_path           = "C:\\game\\GomokuServer.exe"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- build_id - (Required) ID of the Gamelift Build to be deployed on the fleet.
- description - (Optional) Human-readable description of the fleet.

- `ec2_inbound_permission` - (Optional) Range of IP addresses and port settings that permit inbound traffic to access server processes running on the fleet. See below.
- `ec2_instance_type` - (Required) Name of an EC2 instance type. e.g. `t2.micro`
- `instance_role_arn` - (Optional) ARN of an IAM role that instances in the fleet can assume.
- `metric_groups` - (Optional) List of names of metric groups to add this fleet to. A metric group tracks metrics across all fleets in the group. Defaults to `default`.
- `name` - (Required) The name of the fleet.
- `new_game_session_protection_policy` - (Optional) Game session protection policy to apply to all instances in this fleet. e.g. `FullProtection`. Defaults to `NoProtection`.
- `resource_creation_limit_policy` - (Optional) Policy that limits the number of game sessions an individual player can create over a span of time for this fleet. See below.
- `runtime_configuration` - (Optional) Instructions for launching server processes on each instance in the fleet. See below.

» Nested Fields

» `ec2_inbound_permission`

- `from_port` - (Required) Starting value for a range of allowed port numbers.
- `ip_range` - (Required) Range of allowed IP addresses expressed in CIDR notation. e.g. `000.000.000.000/[subnet mask]` or `0.0.0.0/[subnet mask]`.
- `protocol` - (Required) Network communication protocol used by the fleet. e.g. `TCP` or `UDP`
- `to_port` - (Required) Ending value for a range of allowed port numbers. Port numbers are end-inclusive. This value must be higher than `from_port`.

» `resource_creation_limit_policy`

- `new_game_sessions_per_creator` - (Optional) Maximum number of game sessions that an individual can create during the policy period.
- `policy_period_in_minutes` - (Optional) Time span used in evaluating the resource creation limit policy.

» `runtime_configuration`

- `game_session_activation_timeout_seconds` - (Optional) Maximum amount of time (in seconds) that a game session can remain in status

ACTIVATING.

- max_concurrent_game_session_activations - (Optional) Maximum number of game sessions with status ACTIVATING to allow on an instance simultaneously.
- server_process - (Optional) Collection of server process configurations that describe which server processes to run on each instance in a fleet. See below.

» server_process

- concurrent_executions - (Required) Number of server processes using this configuration to run concurrently on an instance.
- launch_path - (Required) Location of the server executable in a game build. All game builds are installed on instances at the root : for Windows instances C:\game, and for Linux instances /local/game.
- parameters - (Optional) Optional list of parameters to pass to the server executable on launch.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - Fleet ID.
- arn - Fleet ARN.
- operating_system - Operating system of the fleet's computing resources.

» Timeouts

aws_gamelift_fleet provides the following Timeouts configuration options:

- create - (Default 70m) How long to wait for a fleet to be created.
- delete - (Default 20m) How long to wait for a fleet to be deleted.

» Import

Gamelift Fleets cannot be imported at this time.

» Resource: aws_gamelift_game_session_queue

Provides an Gamelift Game Session Queue resource.

» Example Usage

```
resource "aws_gamelift_game_session_queue" "test" {
  name = "example-session-queue"

  destinations = [
    "${aws_gamelift_fleet.us_west_2_fleet.arn}",
    "${aws_gamelift_fleet.eu_central_1_fleet.arn}",
  ]

  player_latency_policy {
    maximum_individual_player_latency_milliseconds = 100
    policy_duration_seconds                        = 5
  }

  player_latency_policy {
    maximum_individual_player_latency_milliseconds = 200
  }

  timeout_in_seconds = 60
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the session queue.
- `timeout_in_seconds` - (Required) Maximum time a game session request can remain in the queue.
- `destinations` - (Optional) List of fleet/alias ARNs used by session queue for placing game sessions.
- `player_latency_policy` - (Optional) One or more policies used to choose fleet based on player latency. See below.
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Fields

» `player_latency_policy`

- `maximum_individual_player_latency_milliseconds` - (Required) Maximum latency value that is allowed for any player.
- `policy_duration_seconds` - (Optional) Length of time that the policy is enforced while placing a new game session. Absence of value for this attribute means that the policy is enforced until the queue times out.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Game Session Queue ARN.

» Import

Gamelift Game Session Queues can be imported by their `name`, e.g.

```
$ terraform import aws_gamelift_game_session_queue.example example
```

» Resource: aws_glacier_vault

Provides a Glacier Vault Resource. You can refer to the Glacier Developer Guide for a full explanation of the Glacier Vault functionality

NOTE: When removing a Glacier Vault, the Vault must be empty.

» Example Usage

```
resource "aws_sns_topic" "aws_sns_topic" {
  name = "glacier-sns-topic"
}

resource "aws_glacier_vault" "my_archive" {
  name = "MyArchive"

  notification {
    sns_topic = "${aws_sns_topic.aws_sns_topic.arn}"
    events    = ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
  }

  access_policy = <<EOF
{
    "Version":"2012-10-17",
    "Statement":[
       {
          "Sid": "add-read-only-perm",
          "Principal": "*",
          "Effect": "Allow",
          "Action": [
             "glacier:InitiateJob",
             "glacier:GetJobOutput"
          ],
          "Resource": "arn:aws:glacier:eu-west-1:432981146916:vaults/MyArchive"
       }
    ]
}
EOF

  tags = {
    Test = "MyArchive"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Vault. Names can be between 1 and 255 characters long and the valid characters are a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), and '.' (period).
- `access_policy` - (Optional) The policy document. This is a JSON formatted string. The heredoc syntax or `file` function is helpful here. Use the Glacier Developer Guide for more information on Glacier Vault Policy
- `notification` - (Optional) The notifications for the Vault. Fields documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

notification supports the following:

- `events` - (Required) You can configure a vault to publish a notification for `ArchiveRetrievalCompleted` and `InventoryRetrievalCompleted` events.
- `sns_topic` - (Required) The SNS Topic ARN.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `location` - The URI of the vault that was created.
- `arn` - The ARN of the vault.

» Import

Glacier Vaults can be imported using the `name`, e.g.

```
$ terraform import aws_glacier_vault.archive my_archive
```

» Resource: aws_glacier_vault_lock

Manages a Glacier Vault Lock. You can refer to the Glacier Developer Guide for a full explanation of the Glacier Vault Lock functionality.

NOTE: This resource allows you to test Glacier Vault Lock policies by setting the `complete_lock` argument to `false`. When testing policies in this manner, the Glacier Vault Lock automatically expires after 24 hours and Terraform will show this resource as needing recreation after that time. To permanently apply the policy, set the `complete_lock` argument to `true`. When changing `complete_lock` to `true`, it is expected the resource will show as recreating.

WARNING: Once a Glacier Vault Lock is completed, it is immutable. The deletion of the Glacier Vault Lock is not be possible and attempting to remove it from Terraform will return an error. Set the `ignore_deletion_error` argument to `true` and apply this configuration before attempting to delete this resource via Terraform or use `terraform state rm` to remove this resource from Terraform management.

» Example Usage

» Testing Glacier Vault Lock Policy

```
resource "aws_glacier_vault" "example" {
  name = "example"
}

data "aws_iam_policy_document" "example" {
  statement {
    actions   = ["glacier:DeleteArchive"]
    effect    = "Deny"
    resources = ["${aws_glacier_vault.example.arn}"]

    condition {
      test     = "NumericLessThanEquals"
      variable = "glacier:ArchiveAgeinDays"
      values   = ["365"]
    }
  }
}

resource "aws_glacier_vault_lock" "example" {
  complete_lock = false
  policy        = "${data.aws_iam_policy_document.example.json}"
  vault_name    = "${aws_glacier_vault.example.name}"
}
```

» Permanently Applying Glacier Vault Lock Policy

```
resource "aws_glacier_vault_lock" "example" {
  complete_lock = true
  policy        = "${data.aws_iam_policy_document.example.json}"
  vault_name    = "${aws_glacier_vault.example.name}"
}
```

» Argument Reference

The following arguments are supported:

- `complete_lock` - (Required) Boolean whether to permanently apply this Glacier Lock Policy. Once completed, this cannot be undone. If set to `false`, the Glacier Lock Policy remains in a testing mode for 24 hours. After that time, the Glacier Lock Policy is automatically removed by Glacier and the Terraform resource will show as needing recreation. Changing this from `false` to `true` will show as resource recreation, which is expected. Changing this from `true` to `false` is not possible unless the Glacier Vault is recreated at the same time.
- `policy` - (Required) JSON string containing the IAM policy to apply as the Glacier Vault Lock policy.
- `vault_name` - (Required) The name of the Glacier Vault.
- `ignore_deletion_error` - (Optional) Allow Terraform to ignore the error returned when attempting to delete the Glacier Lock Policy. This can be used to delete or recreate the Glacier Vault via Terraform, for example, if the Glacier Vault Lock policy permits that action. This should only be used in conjunction with `complete_lock` being set to `true`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Glacier Vault name.

» Import

Glacier Vault Locks can be imported using the Glacier Vault name, e.g.

```
$ terraform import aws_glacier_vault_lock.example example-vault
```

» Resource: aws_globalaccelerator_accelerator

Provides a Global Accelerator accelerator.

» Example Usage

```
resource "aws_globalaccelerator_accelerator" "example" {
  name            = "Example"
  ip_address_type = "IPV4"
  enabled         = true

  attributes {
    flow_logs_enabled   = true
    flow_logs_s3_bucket = "example-bucket"
    flow_logs_s3_prefix = "flow-logs/"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the accelerator.
- `ip_address_type` - (Optional) The value for the address type must be `IPV4`.
- `enabled` - (Optional) Indicates whether the accelerator is enabled. The value is true or false. The default value is true.
- `attributes` - (Optional) The attributes of the accelerator. Fields documented below.

attributes supports the following attributes:

- `flow_logs_enabled` - (Optional) Indicates whether flow logs are enabled.
- `flow_logs_s3_bucket` - (Optional) The name of the Amazon S3 bucket for the flow logs.
- `flow_logs_s3_prefix` - (Optional) The prefix for the location in the Amazon S3 bucket for the flow logs.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) of the accelerator.
- `ip_sets` - IP address set associated with the accelerator.

ip_sets exports the following attributes:

- `ip_addresses` - The array of IP addresses in the IP address set.
- `ip_family` - The types of IP addresses included in this IP set.

» Import

Global Accelerator accelerators can be imported using the `id`, e.g.

```
$ terraform import aws_globalaccelerator_accelerator.example arn:aws:globalaccelerator::1111
```

» Resource: aws_globalaccelerator_endpoint_group

Provides a Global Accelerator endpoint group.

» Example Usage

```
resource "aws_globalaccelerator_endpoint_group" "example" {
  listener_arn = "${aws_globalaccelerator_listener.example.id}"

  endpoint_configuration {
    endpoint_id = "${aws_lb.example.arn}"
    weight      = 100
  }
}
```

» Argument Reference

The following arguments are supported:

- `listener_arn` - (Required) The Amazon Resource Name (ARN) of the listener.
- `health_check_interval_seconds` - (Optional) The time—10 seconds or 30 seconds—between each health check for an endpoint. The default value is 30.
- `health_check_path` - (Optional) If the protocol is HTTP/S, then this specifies the path that is the destination for health check targets. The default value is slash (/).
- `health_check_port` - (Optional) The port that AWS Global Accelerator uses to check the health of endpoints that are part of this endpoint group. The default port is the listener port that this endpoint group is associated with. If listener port is a list of ports, Global Accelerator uses the first port in the list.

- `health_check_protocol` - (Optional) The protocol that AWS Global Accelerator uses to check the health of endpoints that are part of this endpoint group. The default value is TCP.
- `threshold_count` - (Optional) The number of consecutive health checks required to set the state of a healthy endpoint to unhealthy, or to set an unhealthy endpoint to healthy. The default value is 3.
- `traffic_dial_percentage` - (Optional) The percentage of traffic to send to an AWS Region. Additional traffic is distributed to other endpoint groups for this listener. The default value is 100.
- `endpoint_configuration` - (Optional) The list of endpoint objects. Fields documented below.

endpoint_configuration supports the following attributes:

- `endpoint_id` - (Optional) An ID for the endpoint. If the endpoint is a Network Load Balancer or Application Load Balancer, this is the Amazon Resource Name (ARN) of the resource. If the endpoint is an Elastic IP address, this is the Elastic IP address allocation ID.
- `weight` - (Optional) The weight associated with the endpoint. When you add weights to endpoints, you configure AWS Global Accelerator to route traffic based on proportions that you specify.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) of the endpoint group.

» Import

Global Accelerator endpoint groups can be imported using the `id`, e.g.

```
$ terraform import aws_globalaccelerator_endpoint_group.example arn:aws:globalaccelerator::1
```

» Resource: aws_globalaccelerator_listener

Provides a Global Accelerator listener.

» Example Usage

```
resource "aws_globalaccelerator_accelerator" "example" {
  name            = "Example"
  ip_address_type = "IPV4"
  enabled         = true

  attributes {
    flow_logs_enabled   = true
    flow_logs_s3_bucket = "example-bucket"
    flow_logs_s3_prefix = "flow-logs/"
  }
}

resource "aws_globalaccelerator_listener" "example" {
  accelerator_arn = "${aws_globalaccelerator_accelerator.example.id}"
  client_affinity = "SOURCE_IP"
  protocol        = "TCP"

  port_range {
    from_port = 80
    to_port   = 80
  }
}
```

» Argument Reference

The following arguments are supported:

- `accelerator_arn` - (Required) The Amazon Resource Name (ARN) of your accelerator.
- `client_affinity` - (Optional) Direct all requests from a user to the same endpoint. Valid values are `NONE`, `SOURCE_IP`. Default: `NONE`. If `NONE`, Global Accelerator uses the "five-tuple" properties of source IP address, source port, destination IP address, destination port, and protocol to select the hash value. If `SOURCE_IP`, Global Accelerator uses the "two-tuple" properties of source (client) IP address and destination IP address to select the hash value.
- `protocol` - (Optional) The protocol for the connections from clients to the accelerator. Valid values are `TCP`, `UDP`.
- `port_range` - (Optional) The list of port ranges for the connections from clients to the accelerator. Fields documented below.

port_range supports the following attributes:

- `from_port` - (Optional) The first port in the range of ports, inclusive.
- `to_port` - (Optional) The last port in the range of ports, inclusive.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) of the listener.

» Import

Global Accelerator listeners can be imported using the `id`, e.g.

```
$ terraform import aws_globalaccelerator_listener.example arn:aws:globalaccelerator::1111111
```

» Data Source: aws_glue_script

Use this data source to generate a Glue script from a Directed Acyclic Graph (DAG).

» Example Usage

» Generate Python Script

```
data "aws_glue_script" "example" {
  language = "PYTHON"

  dag_edge {
    source = "datasource0"
    target = "applymapping1"
  }

  dag_edge {
    source = "applymapping1"
    target = "selectfields2"
  }

  dag_edge {
    source = "selectfields2"
    target = "resolvechoice3"
  }

  dag_edge {
    source = "resolvechoice3"
    target = "datasink4"
  }
```

```
dag_node {
  id        = "datasource0"
  node_type = "DataSource"

  args {
    name  = "database"
    value = "\"${aws_glue_catalog_database.source.name}\""
  }

  args {
    name  = "table_name"
    value = "\"${aws_glue_catalog_table.source.name}\""
  }
}

dag_node {
  id        = "applymapping1"
  node_type = "ApplyMapping"

  args {
    name  = "mapping"
    value = "[(\"column1\", \"string\", \"column1\", \"string\")]"
  }
}

dag_node {
  id        = "selectfields2"
  node_type = "SelectFields"

  args {
    name  = "paths"
    value = "[\"column1\"]"
  }
}

dag_node {
  id        = "resolvechoice3"
  node_type = "ResolveChoice"

  args {
    name  = "choice"
    value = "\"MATCH_CATALOG\""
  }

  args {
      name  = "database"
      value = "\"${aws_glue_catalog_database.destination.name}\""
    }

    args {
      name  = "table_name"
      value = "\"${aws_glue_catalog_table.destination.name}\""
    }
  }

  dag_node {
    id        = "datasink4"
    node_type = "DataSink"

    args {
      name  = "database"
      value = "\"${aws_glue_catalog_database.destination.name}\""
    }

    args {
      name  = "table_name"
      value = "\"${aws_glue_catalog_table.destination.name}\""
    }
  }
}

output "python_script" {
  value = "${data.aws_glue_script.example.python_script}"
}
```

» Generate Scala Code

```
data "aws_glue_script" "example" {
  language = "SCALA"

  dag_edge {
    source = "datasource0"
    target = "applymapping1"
  }

  dag_edge {
    source = "applymapping1"
    target = "selectfields2"
  }
```

```
dag_edge {
  source = "selectfields2"
  target = "resolvechoice3"
}

dag_edge {
  source = "resolvechoice3"
  target = "datasink4"
}

dag_node {
  id        = "datasource0"
  node_type = "DataSource"

  args {
    name  = "database"
    value = "\"${aws_glue_catalog_database.source.name}\""
  }

  args {
    name  = "table_name"
    value = "\"${aws_glue_catalog_table.source.name}\""
  }
}

dag_node {
  id        = "applymapping1"
  node_type = "ApplyMapping"

  args {
    name  = "mappings"
    value = "[(\"column1\", \"string\", \"column1\", \"string\")]"
  }
}

dag_node {
  id        = "selectfields2"
  node_type = "SelectFields"

  args {
    name  = "paths"
    value = "[\"column1\"]"
  }
}

dag_node {
    id        = "resolvechoice3"
    node_type = "ResolveChoice"

    args {
      name  = "choice"
      value = "\"MATCH_CATALOG\""
    }

    args {
      name  = "database"
      value = "\"${aws_glue_catalog_database.destination.name}\""
    }

    args {
      name  = "table_name"
      value = "\"${aws_glue_catalog_table.destination.name}\""
    }
  }

  dag_node {
    id        = "datasink4"
    node_type = "DataSink"

    args {
      name  = "database"
      value = "\"${aws_glue_catalog_database.destination.name}\""
    }

    args {
      name  = "table_name"
      value = "\"${aws_glue_catalog_table.destination.name}\""
    }
  }
}

output "scala_code" {
  value = "${data.aws_glue_script.example.scala_code}"
}
```

» Argument Reference

- `dag_edge` - (Required) A list of the edges in the DAG. Defined below.
- `dag_node` - (Required) A list of the nodes in the DAG. Defined below.
- `language` - (Optional) The programming language of the resulting code from the DAG. Defaults to `PYTHON`. Valid values are `PYTHON` and `SCALA`.

» dag_edge Argument Reference

- `source` - (Required) The ID of the node at which the edge starts.
- `target` - (Required) The ID of the node at which the edge ends.
- `target_parameter` - (Optional) The target of the edge.

» dag_node Argument Reference

- `args` - (Required) Nested configuration an argument or property of a node. Defined below.
- `id` - (Required) A node identifier that is unique within the node's graph.
- `node_type` - (Required) The type of node this is.
- `line_number` - (Optional) The line number of the node.

» args Argument Reference

- `name` - (Required) The name of the argument or property.
- `value` - (Required) The value of the argument or property.
- `param` - (Optional) Boolean if the value is used as a parameter. Defaults to `false`.

» Attributes Reference

- `python_script` - The Python script generated from the DAG when the `language` argument is set to `PYTHON`.
- `scala_code` - The Scala code generated from the DAG when the `language` argument is set to `SCALA`.

» Resource: aws_glue_catalog_database

Provides a Glue Catalog Database Resource. You can refer to the Glue Developer Guide for a full explanation of the Glue Data Catalog functionality

» Example Usage

```
resource "aws_glue_catalog_database" "aws_glue_catalog_database" {
  name = "MyCatalogDatabase"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the database.
- `catalog_id` - (Optional) ID of the Glue Catalog to create the database in. If omitted, this defaults to the AWS Account ID.
- `description` - (Optional) Description of the database.
- `location_uri` - (Optional) The location of the database (for example, an HDFS path).
- `parameters` - (Optional) A list of key-value pairs that define parameters and properties of the database.

» Import

Glue Catalog Databases can be imported using the `catalog_id:name`. If you have not set a Catalog ID specify the AWS Account ID that the database is in, e.g.

```
$ terraform import aws_glue_catalog_database.database 123456789012:my_database
```

» Resource: aws_glue_catalog_table

Provides a Glue Catalog Table Resource. You can refer to the Glue Developer Guide for a full explanation of the Glue Data Catalog functionality.

» Example Usage

» Basic Table

```
resource "aws_glue_catalog_table" "aws_glue_catalog_table" {
  name          = "MyCatalogTable"
  database_name = "MyCatalogDatabase"
}
```

» Parquet Table for Athena

```
resource "aws_glue_catalog_table" "aws_glue_catalog_table" {
  name          = "MyCatalogTable"
  database_name = "MyCatalogDatabase"

  table_type = "EXTERNAL_TABLE"
```

```
parameters = {
  EXTERNAL              = "TRUE"
  "parquet.compression" = "SNAPPY"
}

storage_descriptor {
  location      = "s3://my-bucket/event-streams/my-stream"
  input_format  = "org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat"
  output_format = "org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat"

  ser_de_info {
    name                  = "my-stream"
    serialization_library = "org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe"

    parameters = {
      "serialization.format" = 1
    }
  }

  columns {
    name = "my_string"
    type = "string"
  }

  columns {
    name = "my_double"
    type = "double"
  }

  columns {
    name    = "my_date"
    type    = "date"
    comment = ""
  }

  columns {
    name    = "my_bigint"
    type    = "bigint"
    comment = ""
  }

  columns {
    name    = "my_struct"
    type    = "struct<my_nested_string:string>"
    comment = ""
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the table. For Hive compatibility, this must be entirely lowercase.
- `database_name` - (Required) Name of the metadata database where the table metadata resides. For Hive compatibility, this must be all lowercase.
- `catalog_id` - (Optional) ID of the Glue Catalog and database to create the table in. If omitted, this defaults to the AWS Account ID plus the database name.
- `description` - (Optional) Description of the table.
- `owner` - (Optional) Owner of the table.
- `retention` - (Optional) Retention time for this table.
- `storage_descriptor` - (Optional) A storage descriptor object containing information about the physical storage of this table. You can refer to the Glue Developer Guide for a full explanation of this object.
- `partition_keys` - (Optional) A list of columns by which the table is partitioned. Only primitive types are supported as partition keys.
- `view_original_text` - (Optional) If the table is a view, the original text of the view; otherwise null.
- `view_expanded_text` - (Optional) If the table is a view, the expanded text of the view; otherwise null.
- `table_type` - (Optional) The type of this table (EXTERNAL_TABLE, VIRTUAL_VIEW, etc.).
- `parameters` - (Optional) Properties associated with this table, as a list of key-value pairs.

» storage_descriptor

- `columns` - (Optional) A list of the Columns in the table.
- `location` - (Optional) The physical location of the table. By default this takes the form of the warehouse location, followed by the database location in the warehouse, followed by the table name.
- `input_format` - (Optional) The input format: SequenceFileInputFormat (binary), or TextInputFormat, or a custom format.
- `output_format` - (Optional) The output format: SequenceFileOutputFormat (binary), or IgnoreKeyTextOutputFormat, or a custom format.
- `compressed` - (Optional) True if the data in the table is compressed, or False if not.

- `number_of_buckets` - (Optional) Must be specified if the table contains any dimension columns.
- `ser_de_info` - (Optional) Serialization/deserialization (SerDe) information.
- `bucket_columns` - (Optional) A list of reducer grouping columns, clustering columns, and bucketing columns in the table.
- `sort_columns` - (Optional) A list of Order objects specifying the sort order of each bucket in the table.
- `parameters` - (Optional) User-supplied properties in key-value form.
- `skewed_info` - (Optional) Information about values that appear very frequently in a column (skewed values).
- `stored_as_sub_directories` - (Optional) True if the table data is stored in subdirectories, or False if not.

» column

- `name` - (Required) The name of the Column.
- `type` - (Optional) The datatype of data in the Column.
- `comment` - (Optional) Free-form text comment.

» ser_de_info

- `name` - (Optional) Name of the SerDe.
- `parameters` - (Optional) A map of initialization parameters for the SerDe, in key-value form.
- `serialization_library` - (Optional) Usually the class that implements the SerDe. An example is: org.apache.hadoop.hive.serde2.columnar.ColumnarSerDe.

» sort_columns

- `column` - (Required) The name of the column.
- `sort_order` - (Required) Indicates that the column is sorted in ascending order (== 1), or in descending order (==0).

» skewed_info

- `skewed_column_names` - (Optional) A list of names of columns that contain skewed values.
- `skewed_column_value_location_maps` - (Optional) A list of values that appear so frequently as to be considered skewed.
- `skewed_column_values` - (Optional) A mapping of skewed values to the columns that contain them.

» Import

Glue Tables can be imported with their catalog ID (usually AWS account ID), database name, and table name, e.g.

```
$ terraform import aws_glue_catalog_table.MyTable 123456789012:MyDatabase:MyTable
```

» Resource: aws_glue_classifier

Provides a Glue Classifier resource.

NOTE: It is only valid to create one type of classifier (csv, grok, JSON, or XML). Changing classifier types will recreate the classifier.

» Example Usage

» Csv Classifier

```
resource "aws_glue_classifier" "example" {
  name = "example"

  csv_classifier {
    allow_single_column    = false
    contains_header        = "PRESENT"
    delimiter              = ","
    disable_value_trimming = false
    header                 = ["example1", "example2"]
    quote_symbol           = "'"
  }
}
```

» Grok Classifier

```
resource "aws_glue_classifier" "example" {
  name = "example"

  grok_classifier {
    classification = "example"
    grok_pattern   = "example"
  }
}
```

» JSON Classifier

```
resource "aws_glue_classifier" "example" {
  name = "example"

  json_classifier {
    json_path = "example"
  }
}
```

» XML Classifier

```
resource "aws_glue_classifier" "example" {
  name = "example"

  xml_classifier {
    classification = "example"
    row_tag        = "example"
  }
}
```

» Argument Reference

The following arguments are supported:

- `csv_classifier` - (Optional) A classifier for Csv content. Defined below.
- `grok_classifier` – (Optional) A classifier that uses grok patterns. Defined below.
- `json_classifier` – (Optional) A classifier for JSON content. Defined below.
- `name` – (Required) The name of the classifier.
- `xml_classifier` – (Optional) A classifier for XML content. Defined below.

» csv_classifier

- `allow_single_column` - (Optional) Enables the processing of files that contain only one column.
- `contains_header` - (Optional) Indicates whether the CSV file contains a header. This can be one of ”ABSENT”, ”PRESENT”, or ”UNKNOWN”.
- `delimiter` - (Optional) The delimiter used in the Csv to separate columns.
- `disable_value_trimming` - (Optional) Specifies whether to trim column values.
- `header` - (Optional) A list of strings representing column names.

- `quote_symbol` - (Optional) A custom symbol to denote what combines content into a single column value. It must be different from the column delimiter.

» grok_classifier

- `classification` - (Required) An identifier of the data format that the classifier matches, such as Twitter, JSON, Omniture logs, Amazon CloudWatch Logs, and so on.
- `custom_patterns` - (Optional) Custom grok patterns used by this classifier.
- `grok_pattern` - (Required) The grok pattern used by this classifier.

» json_classifier

- `json_path` - (Required) A `JsonPath` string defining the JSON data for the classifier to classify. AWS Glue supports a subset of `JsonPath`, as described in Writing JsonPath Custom Classifiers.

» xml_classifier

- `classification` - (Required) An identifier of the data format that the classifier matches.
- `row_tag` - (Required) The XML tag designating the element that contains each record in an XML document being parsed. Note that this cannot identify a self-closing element (closed by `/>`). An empty row element that contains only attributes can be parsed as long as it ends with a closing tag (for example, `<row item_a="A" item_b="B"></row>` is okay, but `<row item_a="A" item_b="B" />` is not).

» Attributes Reference

The following additional attributes are exported:

- `id` - Name of the classifier

» Import

Glue Classifiers can be imported using their name, e.g.

```
$ terraform import aws_glue_classifier.MyClassifier MyClassifier
```

» Resource: aws_glue_connection

Provides a Glue Connection resource.

» Example Usage

» Non-VPC Connection

```
resource "aws_glue_connection" "example" {
  connection_properties = {
    JDBC_CONNECTION_URL = "jdbc:mysql://example.com/exampledatabase"
    PASSWORD            = "examplepassword"
    USERNAME            = "exampleusername"
  }

  name = "example"
}
```

» VPC Connection

For more information, see the AWS Documentation.

```
resource "aws_glue_connection" "example" {
  connection_properties = {
    JDBC_CONNECTION_URL = "jdbc:mysql://${aws_rds_cluster.example.endpoint}/exampledatabase'
    PASSWORD            = "examplepassword"
    USERNAME            = "exampleusername"
  }

  name = "example"

  physical_connection_requirements {
    availability_zone      = "${aws_subnet.example.availability_zone}"
    security_group_id_list = ["${aws_security_group.example.id}"]
    subnet_id              = "${aws_subnet.example.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `catalog_id` – (Optional) The ID of the Data Catalog in which to create the connection. If none is supplied, the AWS account ID is used by default.

- `connection_properties` – (Required) A map of key-value pairs used as parameters for this connection.
- `connection_type` – (Optional) The type of the connection. Defaults to `JBDC`.
- `description` – (Optional) Description of the connection.
- `match_criteria` – (Optional) A list of criteria that can be used in selecting this connection.
- `name` – (Required) The name of the connection.
- `physical_connection_requirements` - (Optional) A map of physical connection requirements, such as VPC and SecurityGroup. Defined below.

» physical_connection_requirements

- `availability_zone` - (Optional) The availability zone of the connection. This field is redundant and implied by `subnet_id`, but is currently an api requirement.
- `security_group_id_list` - (Optional) The security group ID list used by the connection.
- `subnet_id` - (Optional) The subnet ID used by the connection.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Catalog ID and name of the connection

» Import

Glue Connections can be imported using the `CATALOG-ID` (AWS account ID if not custom) and `NAME`, e.g.

```
$ terraform import aws_glue_connection.MyConnection 123456789012:MyConnection
```

» Resource: aws_glue_crawler

Manages a Glue Crawler. More information can be found in the AWS Glue Developer Guide

» Example Usage

» DynamoDB Target

```
resource "aws_glue_crawler" "example" {
  database_name = "${aws_glue_catalog_database.example.name}"
  name          = "example"
  role          = "${aws_iam_role.example.arn}"

  dynamodb_target {
    path = "table-name"
  }
}
```

» JDBC Target

```
resource "aws_glue_crawler" "example" {
  database_name = "${aws_glue_catalog_database.example.name}"
  name          = "example"
  role          = "${aws_iam_role.example.arn}"

  jdbc_target {
    connection_name = "${aws_glue_connection.example.name}"
    path            = "database-name/%"
  }
}
```

» S3 Target

```
resource "aws_glue_crawler" "example" {
  database_name = "${aws_glue_catalog_database.example.name}"
  name          = "example"
  role          = "${aws_iam_role.example.arn}"

  s3_target {
    path = "s3://${aws_s3_bucket.example.bucket}"
  }
}
```

» Catalog Target

```
resource "aws_glue_crawler" "example" {
  database_name = "${aws_glue_catalog_database.example.name}"
  name          = "example"
  role          = "${aws_iam_role.example.arn}"

  catalog_target {
    database_name = "${aws_glue_catalog_database.example.name}"
    tables = ["${aws_glue_catalog_table.example.name}"]
  }

  schema_change_policy {
    delete_behavior = "LOG"
  }

  configuration = <<EOF
{
  "Version":1.0,
  "Grouping": {
    "TableGroupingPolicy": "CombineCompatibleSchemas"
  }
}
EOF
}
```

» Argument Reference

NOTE: Must specify at least one of `dynamodb_target`, `jdbc_target`, `s3_target` or `catalog_target`.

The following arguments are supported:

- `database_name` (Required) Glue database where results are written.
- `name` (Required) Name of the crawler.
- `role` (Required) The IAM role friendly name (including path without leading slash), or ARN of an IAM role, used by the crawler to access other resources.
- `classifiers` (Optional) List of custom classifiers. By default, all AWS classifiers are included in a crawl, but these custom classifiers always override the default classifiers for a given classification.
- `configuration` (Optional) JSON string of configuration information.
- `description` (Optional) Description of the crawler.
- `dynamodb_target` (Optional) List of nested DynamoDB target arguments. See below.
- `jdbc_target` (Optional) List of nested JBDC target arguments. See below.
- `s3_target` (Optional) List nested Amazon S3 target arguments. See below.

- `schedule` (Optional) A cron expression used to specify the schedule. For more information, see Time-Based Schedules for Jobs and Crawlers. For example, to run something every day at 12:15 UTC, you would specify: `cron(15 12 * * ? *)`.
- `schema_change_policy` (Optional) Policy for the crawler's update and deletion behavior.
- `security_configuration` (Optional) The name of Security Configuration to be used by the crawler
- `table_prefix` (Optional) The table prefix used for catalog tables that are created.
- `tags` - (Optional) Key-value mapping of resource tags

» dynamodb_target Argument Reference

- `path` - (Required) The name of the DynamoDB table to crawl.

» jdbc_target Argument Reference

- `connection_name` - (Required) The name of the connection to use to connect to the JDBC target.
- `path` - (Required) The path of the JDBC target.
- `exclusions` - (Optional) A list of glob patterns used to exclude from the crawl.

» s3_target Argument Reference

- `path` - (Required) The path to the Amazon S3 target.
- `exclusions` - (Optional) A list of glob patterns used to exclude from the crawl.

» catalog_target Argument Reference

- `database_name` - (Required) The name of the Glue database to be synchronized.
- `tables` - (Required) A list of catalog tables to be synchronized.

Note: `deletion_behavior` of catalog target doesn't support `DEPRECATE_IN_DATABASE`.

Note: `configuration` for catalog target crawlers will have `{ ... "Grouping": { "TableGroupingPolicy": "CombineCompatibleSchemas"} }` by default.

» schema_change_policy Argument Reference

- `delete_behavior` - (Optional) The deletion behavior when the crawler finds a deleted object. Valid values: `LOG`, `DELETE_FROM_DATABASE`, or `DEPRECATE_IN_DATABASE`. Defaults to `DEPRECATE_IN_DATABASE`.
- `update_behavior` - (Optional) The update behavior when the crawler finds a changed schema. Valid values: `LOG` or `UPDATE_IN_DATABASE`. Defaults to `UPDATE_IN_DATABASE`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Crawler name
- `arn` - The ARN of the crawler

» Import

Glue Crawlers can be imported using `name`, e.g.

```
$ terraform import aws_glue_crawler.MyJob MyJob
```

» Resource: aws_glue_job

Provides a Glue Job resource.

Glue functionality, such as monitoring and logging of jobs, is typically managed with the `default_arguments` argument. See the Special Parameters Used by AWS Glue topic in the Glue developer guide for additional information.

» Example Usage

» Python Job

```
resource "aws_glue_job" "example" {
  name     = "example"
  role_arn = "${aws_iam_role.example.arn}"

  command {
    script_location = "s3://${aws_s3_bucket.example.bucket}/example.py"
  }
}
```

» Scala Job

```
resource "aws_glue_job" "example" {
  name     = "example"
  role_arn = "${aws_iam_role.example.arn}"

  command {
    script_location = "s3://${aws_s3_bucket.example.bucket}/example.scala"
  }

  default_arguments = {
    "--job-language" = "scala"
  }
}
```

» Enabling CloudWatch Logs and Metrics

```
resource "aws_cloudwatch_log_group" "example" {
  name              = "example"
  retention_in_days = 14
}

resource "aws_glue_job" "example" {
  # ... other configuration ...

  default_arguments = {
    # ... potentially other arguments ...
    "--continuous-log-logGroup"          = "${aws_cloudwatch_log_group.example.name}"
    "--enable-continuous-cloudwatch-log" = "true"
    "--enable-continuous-log-filter"     = "true"
    "--enable-metrics"                   = ""
  }
}
```

» Argument Reference

The following arguments are supported:

NOTE: The `allocated_capacity` attribute has been deprecated and might be removed in future releases, please use `max_capacity` instead.

- `allocated_capacity` – **DEPRECATED** (Optional) The number of AWS Glue data processing units (DPUs) to allocate to this Job. At least 2 DPUs need to be allocated; the default is 10. A DPU is a relative measure of processing power that consists of 4 vCPUs of compute capacity and 16 GB of memory.

- `command` – (Required) The command of the job. Defined below.
- `connections` – (Optional) The list of connections used for this job.
- `default_arguments` – (Optional) The map of default arguments for this job. You can specify arguments here that your own job-execution script consumes, as well as arguments that AWS Glue itself consumes. For information about how to specify and consume your own Job arguments, see the Calling AWS Glue APIs in Python topic in the developer guide. For information about the key-value pairs that AWS Glue consumes to set up your job, see the Special Parameters Used by AWS Glue topic in the developer guide.
- `description` – (Optional) Description of the job.
- `execution_property` – (Optional) Execution property of the job. Defined below.
- `glue_version` - (Optional) The version of glue to use, for example "1.0". For information about available versions, see the AWS Glue Release Notes.
- `max_capacity` – (Optional) The maximum number of AWS Glue data processing units (DPUs) that can be allocated when this job runs.
- `max_retries` – (Optional) The maximum number of times to retry this job if it fails.
- `name` – (Required) The name you assign to this job. It must be unique in your account.
- `role_arn` – (Required) The ARN of the IAM role associated with this job.
- `tags` - (Optional) Key-value mapping of resource tags
- `timeout` – (Optional) The job timeout in minutes. The default is 2880 minutes (48 hours).
- `security_configuration` - (Optional) The name of the Security Configuration to be associated with the job.
- `worker_type` - (Optional) The type of predefined worker that is allocated when a job runs. Accepts a value of Standard, G.1X, or G.2X.
- `number_of_workers` - (Optional) The number of workers of a defined workerType that are allocated when a job runs.

» command Argument Reference

- `name` - (Optional) The name of the job command. Defaults to `glueetl`
- `script_location` - (Required) Specifies the S3 path to a script that executes a job.
- `python_version` - (Optional) The Python version being used to execute a Python shell job. Allowed values are 2 or 3.

» execution_property Argument Reference

- `max_concurrent_runs` - (Optional) The maximum number of concurrent runs allowed for a job. The default is 1.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of Glue Job
- `id` - Job name

» Import

Glue Jobs can be imported using `name`, e.g.

```
$ terraform import aws_glue_job.MyJob MyJob
```

» Resource: aws_glue_security_configuration

Manages a Glue Security Configuration.

» Example Usage

```
resource "aws_glue_security_configuration" "example" {
  name = "example"

  encryption_configuration {
    cloudwatch_encryption {
      cloudwatch_encryption_mode = "DISABLED"
    }

    job_bookmarks_encryption {
      job_bookmarks_encryption_mode = "DISABLED"
    }

    s3_encryption {
      kms_key_arn        = "${data.aws_kms_key.example.arn}"
      s3_encryption_mode = "SSE-KMS"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `encryption_configuration` – (Required) Configuration block containing encryption configuration. Detailed below.
- `name` – (Required) Name of the security configuration.

» encryption_configuration Argument Reference

- `cloudwatch_encryption` - (Required) A `cloudwatch_encryption` block as described below, which contains encryption configuration for CloudWatch.
- `job_bookmarks_encryption` - (Required) A `job_bookmarks_encryption` block as described below, which contains encryption configuration for job bookmarks.
- `s3_encryption` - (Required) A `s3_encryption` block as described below, which contains encryption configuration for S3 data.

» cloudwatch_encryption Argument Reference

- `cloudwatch_encryption_mode` - (Optional) Encryption mode to use for CloudWatch data. Valid values: `DISABLED`, `SSE-KMS`. Default value: `DISABLED`.
- `kms_key_arn` - (Optional) Amazon Resource Name (ARN) of the KMS key to be used to encrypt the data.

» job_bookmarks_encryption Argument Reference

- `job_bookmarks_encryption_mode` - (Optional) Encryption mode to use for job bookmarks data. Valid values: `CSE-KMS`, `DISABLED`. Default value: `DISABLED`.
- `kms_key_arn` - (Optional) Amazon Resource Name (ARN) of the KMS key to be used to encrypt the data.

» s3_encryption Argument Reference

- `s3_encryption_mode` - (Optional) Encryption mode to use for S3 data. Valid values: `DISABLED`, `SSE-KMS`, `SSE-S3`. Default value: `DISABLED`.
- `kms_key_arn` - (Optional) Amazon Resource Name (ARN) of the KMS key to be used to encrypt the data.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Glue security configuration name

» Import

Glue Security Configurations can be imported using `name`, e.g.

```
$ terraform import aws_glue_security_configuration.example example
```

» Resource: aws_glue_trigger

Manages a Glue Trigger resource.

» Example Usage

» Conditional Trigger

```
resource "aws_glue_trigger" "example" {
  name          = "example"
  type          = "CONDITIONAL"

  actions {
    job_name = "${aws_glue_job.example1.name}"
  }

  predicate {
    conditions {
      job_name = "${aws_glue_job.example2.name}"
      state    = "SUCCEEDED"
    }
  }
}
```

» On-Demand Trigger

```
resource "aws_glue_trigger" "example" {
  name = "example"
  type = "ON_DEMAND"
  actions {
    job_name = "${aws_glue_job.example.name}"
  }
}
```

» Scheduled Trigger

```
resource "aws_glue_trigger" "example" {
  name     = "example"
  schedule = "cron(15 12 * * ? *)"
  type     = "SCHEDULED"

  actions {
    job_name = "${aws_glue_job.example.name}"
  }
}
```

» Conditional Trigger with Crawler Action

Note: Triggers can have both a crawler action and a crawler condition, just no example provided.

```
resource "aws_glue_trigger" "example" {
  name = "example"
  type = "CONDITIONAL"

  actions {
    crawler_name = "${aws_glue_crawler.example1.name}"
  }

  predicate {
    conditions {
      job_name = "${aws_glue_job.example2.name}"
      state    = "SUCCEEDED"
    }
  }
}
```

» Conditional Trigger with Crawler Condition

Note: Triggers can have both a crawler action and a crawler condition, just no example provided.

```
resource "aws_glue_trigger" "example" {
  name = "example"
  type = "CONDITIONAL"

  actions {
    job_name = "${aws_glue_job.example1.name}"
  }

  predicate {
    conditions {
      crawler_name = "${aws_glue_crawler.example2.name}"
      crawl_state    = "SUCCEEDED"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `actions` – (Required) List of actions initiated by this trigger when it fires. Defined below.
- `description` – (Optional) A description of the new trigger.
- `enabled` – (Optional) Start the trigger. Defaults to `true`. Not valid to disable for `ON_DEMAND` type.
- `name` – (Required) The name of the trigger.
- `predicate` – (Optional) A predicate to specify when the new trigger should fire. Required when trigger type is `CONDITIONAL`. Defined below.
- `schedule` – (Optional) A cron expression used to specify the schedule. Time-Based Schedules for Jobs and Crawlers
- `tags` - (Optional) Key-value mapping of resource tags
- `type` – (Required) The type of trigger. Valid values are `CONDITIONAL`, `ON_DEMAND`, and `SCHEDULED`.
- `workflow_name` - (Optional) A workflow to which the trigger should be associated to. Every workflow graph (DAG) needs a starting trigger (`ON_DEMAND` or `SCHEDULED` type) and can contain multiple additional `CONDITIONAL` triggers.

» actions Argument Reference

- `arguments` - (Optional) Arguments to be passed to the job. You can specify arguments here that your own job-execution script consumes, as well as arguments that AWS Glue itself consumes.
- `crawler_name` - (Optional) The name of the crawler to be executed. Conflicts with `job_name`.

- `job_name` - (Optional) The name of a job to be executed. Conflicts with `crawler_name`.
- `timeout` - (Optional) The job run timeout in minutes. It overrides the timeout value of the job.

» predicate Argument Reference

- `conditions` - (Required) A list of the conditions that determine when the trigger will fire. Defined below.
- `logical` - (Optional) How to handle multiple conditions. Defaults to `AND`. Valid values are `AND` or `ANY`.

» conditions Argument Reference

- `job_name` - (Optional) The name of the job to watch. If this is specified, `state` must also be specified. Conflicts with `crawler_name`.
- `state` - (Optional) The condition job state. Currently, the values supported are `SUCCEEDED`, `STOPPED`, `TIMEOUT` and `FAILED`. If this is specified, `job_name` must also be specified. Conflicts with `crawler_state`.
- `crawler_name` - (Optional) The name of the crawler to watch. If this is specified, `crawl_state` must also be specified. Conflicts with `job_name`.
- `crawl_state` - (Optional) The condition crawl state. Currently, the values supported are `RUNNING`, `SUCCEEDED`, `CANCELLED`, and `FAILED`. If this is specified, `crawler_name` must also be specified. Conflicts with `state`.
- `logical_operator` - (Optional) A logical operator. Defaults to `EQUALS`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of Glue Trigger
- `id` - Trigger name

» Timeouts

`aws_glue_trigger` provides the following Timeouts configuration options:

- `create` - (Default `5m`) How long to wait for a trigger to be created.
- `delete` - (Default `5m`) How long to wait for a trigger to be deleted.

» Import

Glue Triggers can be imported using `name`, e.g.

```
$ terraform import aws_glue_trigger.MyTrigger MyTrigger
```

» Resource: aws_glue_workflow

Provides a Glue Workflow resource. The workflow graph (DAG) can be build using the `aws_glue_trigger` resource. See the example below for creating a graph with four nodes (two triggers and two jobs).

» Example Usage

```
resource "aws_glue_workflow" "example" {
  name = "example"
}

resource "aws_glue_trigger" "example-start" {
  name = "trigger-start"
  type = "ON_DEMAND"
  workflow_name = "${aws_glue_workflow.example.name}"

  actions {
    job_name = "example-job"
  }
}

resource "aws_glue_trigger" "example-inner" {
  name = "trigger-inner"
  type = "CONDITIONAL"
  workflow_name = "${aws_glue_workflow.example.name}"

  predicate {
    conditions {
      job_name = "example-job"
      state = "SUCCEEDED"
    }
  }

  actions {
    job_name = "another-example-job"
  }
}
```

» Argument Reference

The following arguments are supported:

- `default_run_properties` – (Optional) A map of default run properties for this workflow. These properties are passed to all jobs associated to the workflow.
- `description` – (Optional) Description of the workflow.
- `name` – (Required) The name you assign to this workflow.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Workflow name

» Import

Glue Workflows can be imported using `name`, e.g.

```
$ terraform import aws_glue_workflow.MyWorkflow MyWorkflow
```

» Data Source: aws_guardduty_detector

Retrieve information about a GuardDuty detector.

» Example Usage

```
data "aws_guardduty_detector" "example" {}
```

» Argument Reference

- `id` - (Optional) The ID of the detector.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `finding_publishing_frequency` - The frequency of notifications sent about subsequent finding occurrences.
- `service_role_arn` - The service-linked role that grants GuardDuty access to the resources in the AWS account.

- `status` - The current status of the detector.

» Resource: aws_guardduty_detector

Provides a resource to manage a GuardDuty detector.

NOTE: Deleting this resource is equivalent to "disabling" GuardDuty for an AWS region, which removes all existing findings. You can set the `enable` attribute to `false` to instead "suspend" monitoring and feedback reporting while keeping existing data. See the Suspending or Disabling Amazon GuardDuty documentation for more information.

» Example Usage

```
resource "aws_guardduty_detector" "MyDetector" {
  enable = true
}
```

» Argument Reference

The following arguments are supported:

- `enable` - (Optional) Enable monitoring and feedback reporting. Setting to `false` is equivalent to "suspending" GuardDuty. Defaults to `true`.
- `finding_publishing_frequency` - (Optional) Specifies the frequency of notifications sent for subsequent finding occurrences. If the detector is a GuardDuty member account, the value is determined by the GuardDuty master account and cannot be modified, otherwise defaults to `SIX_HOURS`. For standalone and GuardDuty master accounts, it must be configured in Terraform to enable drift detection. Valid values for standalone and master accounts: `FIFTEEN_MINUTES`, `ONE_HOUR`, `SIX_HOURS`. See AWS Documentation for more information.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the GuardDuty detector
- `account_id` - The AWS account ID of the GuardDuty detector

» Import

GuardDuty detectors can be imported using the detector ID, e.g.

```
$ terraform import aws_guardduty_detector.MyDetector 00b00fd5aecc0ab60a708659477e9617
```

» Resource: aws_guardduty_invite_accepter

Provides a resource to accept a pending GuardDuty invite on creation, ensure the detector has the correct master account on read, and disassociate with the master account upon removal.

» Example Usage

```
resource "aws_guardduty_detector" "master" {}

resource "aws_guardduty_detector" "member" {
  provider = "aws.dev"
}

resource "aws_guardduty_member" "dev" {
  account_id  = "${aws_guardduty_detector.member.account_id}"
  detector_id = "${aws_guardduty_detector.master.id}"
  email       = "required@example.com"
  invite      = true
}

resource "aws_guardduty_invite_accepter" "member" {
  depends_on = ["aws_guardduty_member.dev"]
  provider   = "aws.dev"

  detector_id       = "${aws_guardduty_detector.member.id}"
  master_account_id = "${aws_guardduty_detector.master.account_id}"
}
```

» Argument Reference

The following arguments are supported:

- `detector_id` - (Required) The detector ID of the member GuardDuty account.
- `master_account_id` - (Required) AWS account ID for master account.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - GuardDuty member detector ID

» Timeouts

`aws_guardduty_invite_accepter` provides the following Timeouts configuration options:

- `create` - (Default `1m`) How long to wait for an invite to accept.

» Import

`aws_guardduty_invite_accepter` can be imported using the the member GuardDuty detector ID, e.g.

```
$ terraform import aws_guardduty_invite_accepter.member 00b00fd5aecc0ab60a708659477e9617
```

» Resource: aws_guardduty_ipset

Provides a resource to manage a GuardDuty IPSet.

Note: Currently in GuardDuty, users from member accounts cannot upload and further manage IPSets. IPSets that are uploaded by the master account are imposed on GuardDuty functionality in its member accounts. See the GuardDuty API Documentation

» Example Usage

```
resource "aws_guardduty_detector" "master" {
  enable = true
}

resource "aws_s3_bucket" "bucket" {
  acl = "private"
}

resource "aws_s3_bucket_object" "MyIPSet" {
  acl     = "public-read"
  content = "10.0.0.0/8\n"
  bucket  = "${aws_s3_bucket.bucket.id}"
  key     = "MyIPSet"
}

resource "aws_guardduty_ipset" "MyIPSet" {
  activate    = true
  detector_id = "${aws_guardduty_detector.master.id}"
  format      = "TXT"
  location    = "https://s3.amazonaws.com/${aws_s3_bucket_object.MyIPSet.bucket}/${aws_s3_bu
  name        = "MyIPSet"
}
```

» Argument Reference

The following arguments are supported:

- `activate` - (Required) Specifies whether GuardDuty is to start using the uploaded IPSet.
- `detector_id` - (Required) The detector ID of the GuardDuty.
- `format` - (Required) The format of the file that contains the IPSet. Valid values: `TXT` | `STIX` | `OTX_CSV` | `ALIEN_VAULT` | `PROOF_POINT` | `FIRE_EYE`
- `location` - (Required) The URI of the file that contains the IPSet.
- `name` - (Required) The friendly name to identify the IPSet.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the GuardDuty IPSet.

» Import

GuardDuty IPSet can be imported using the the master GuardDuty detector ID and IPSet ID, e.g.

```
$ terraform import aws_guardduty_ipset.MyIPSet 00b00fd5aecc0ab60a708659477e9617:12345678901
```

» Resource: aws_guardduty_member

Provides a resource to manage a GuardDuty member. To accept invitations in member accounts, see the `aws_guardduty_invite_accepter` resource.

» Example Usage

```
resource "aws_guardduty_detector" "master" {
  enable = true
}

resource "aws_guardduty_detector" "member" {
  provider = "aws.dev"

  enable = true
}

resource "aws_guardduty_member" "member" {
  account_id         = "${aws_guardduty_detector.member.account_id}"
  detector_id        = "${aws_guardduty_detector.master.id}"
  email              = "required@example.com"
  invite             = true
  invitation_message = "please accept guardduty invitation"
}
```

» Argument Reference

The following arguments are supported:

- `account_id` - (Required) AWS account ID for member account.
- `detector_id` - (Required) The detector ID of the GuardDuty account where you want to create member accounts.
- `email` - (Required) Email address for member account.
- `invite` - (Optional) Boolean whether to invite the account to GuardDuty as a member. Defaults to `false`. To detect if an invitation needs to be (re-)sent, the Terraform state value is `true` based on a `relationship_status` of `Disabled`, `Enabled`, `Invited`, or `EmailVerificationInProgress`.
- `invitation_message` - (Optional) Message for invitation.
- `disable_email_notification` - (Optional) Boolean whether an email notification is sent to the accounts. Defaults to `false`.

» Timeouts

`aws_guardduty_member` provides the following Timeouts configuration options:

- `create` - (Default `60s`) How long to wait for a verification to be done against inviting GuardDuty member account.
- `update` - (Default `60s`) How long to wait for a verification to be done against inviting GuardDuty member account.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the GuardDuty member
- `relationship_status` - The status of the relationship between the member account and its master account. More information can be found in Amazon GuardDuty API Reference.

» Import

GuardDuty members can be imported using the the master GuardDuty detector ID and member AWS account ID, e.g.

```
$ terraform import aws_guardduty_member.MyMember 00b00fd5aecc0ab60a708659477e9617:1234567890
```

» Resource: aws_guardduty_threatintelset

Provides a resource to manage a GuardDuty ThreatIntelSet.

Note: Currently in GuardDuty, users from member accounts cannot upload and further manage ThreatIntelSets. ThreatIntelSets that are uploaded by the master account are imposed on GuardDuty functionality in its member accounts. See the GuardDuty API Documentation

» Example Usage

```
resource "aws_guardduty_detector" "master" {
  enable = true
}

resource "aws_s3_bucket" "bucket" {
  acl = "private"
}

resource "aws_s3_bucket_object" "MyThreatIntelSet" {
  acl     = "public-read"
  content = "10.0.0.0/8\n"
  bucket  = "${aws_s3_bucket.bucket.id}"
  key     = "MyThreatIntelSet"
}

resource "aws_guardduty_threatintelset" "MyThreatIntelSet" {
  activate    = true
  detector_id = "${aws_guardduty_detector.master.id}"
  format      = "TXT"
  location    = "https://s3.amazonaws.com/${aws_s3_bucket_object.MyThreatIntelSet.bucket}/$
  name        = "MyThreatIntelSet"
}
```

» Argument Reference

The following arguments are supported:

- activate - (Required) Specifies whether GuardDuty is to start using the uploaded ThreatIntelSet.
- detector_id - (Required) The detector ID of the GuardDuty.
- format - (Required) The format of the file that contains the ThreatIntelSet. Valid values: TXT | STIX | OTX_CSV | ALIEN_VAULT | PROOF_POINT | FIRE_EYE
- location - (Required) The URI of the file that contains the ThreatIntelSet.
- name - (Required) The friendly name to identify the ThreatIntelSet.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The ID of the GuardDuty ThreatIntelSet and the detector ID. Format: <DetectorID>:<ThreatIntelSetID>

» Import

GuardDuty ThreatIntelSet can be imported using the the master GuardDuty detector ID and ThreatIntelSetID, e.g.

```
$ terraform import aws_guardduty_threatintelset.MyThreatIntelSet 00b00fd5aecc0ab60a70865947
```

» Data Source: aws_iam_account_alias

The IAM Account Alias data source allows access to the account alias for the effective account in which Terraform is working.

» Example Usage

```
data "aws_iam_account_alias" "current" {}
```

```
output "account_id" {
  value = "${data.aws_iam_account_alias.current.account_alias}"
}
```

» Argument Reference

There are no arguments available for this data source.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `account_alias` - The alias associated with the AWS account.

» Data Source: aws_iam_group

This data source can be used to fetch information about a specific IAM group. By using this data source, you can reference IAM group properties without having to hard code ARNs as input.

» Example Usage

```
data "aws_iam_group" "example" {
  group_name = "an_example_group_name"
}
```

» Argument Reference

- `group_name` - (Required) The friendly IAM group name to match.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) specifying the group.
- `path` - The path to the group.
- `group_id` - The stable and unique string identifying the group.
- `users` - List of objects containing group member information. See supported fields below.

» users

- `arn` - The Amazon Resource Name (ARN) specifying the iam user.
- `user_id` - The stable and unique string identifying the iam user.
- `user_name` - The name of the iam user.
- `path` - The path to the iam user.

» Data Source: aws_iam_instance_profile

This data source can be used to fetch information about a specific IAM instance profile. By using this data source, you can reference IAM instance profile properties without having to hard code ARNs as input.

» Example Usage

```
data "aws_iam_instance_profile" "example" {
  name = "an_example_instance_profile_name"
}
```

» Argument Reference

- `name` - (Required) The friendly IAM instance profile name to match.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) specifying the instance profile.
- `create_date` - The string representation of the date the instance profile was created.
- `path` - The path to the instance profile.
- `role_arn` - The role arn associated with this instance profile.
- `role_id` - The role id associated with this instance profile.
- `role_name` - The role name associated with this instance profile.

» aws_iam_policy

This data source can be used to fetch information about a specific IAM policy.

» Example Usage

```
data "aws_iam_policy" "example" {
  arn = "arn:aws:iam::123456789012:policy/UsersManageOwnCredentials"
}
```

» Argument Reference

- `arn` - (Required) ARN of the IAM policy.

» Attributes Reference

- `name` - The name of the IAM policy.
- `arn` - The Amazon Resource Name (ARN) specifying the policy.
- `path` - The path to the policy.
- `description` - The description of the policy.
- `policy` - The policy document of the policy.

» Data Source: aws_iam_policy_document

Generates an IAM policy document in JSON format.

This is a data source which can be used to construct a JSON representation of an IAM policy document, for use with resources which expect policy documents, such as the `aws_iam_policy` resource.

For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

```
data "aws_iam_policy_document" "example" {
  statement {
    sid = "1"

    actions = [
      "s3:ListAllMyBuckets",
      "s3:GetBucketLocation",
    ]

    resources = [
      "arn:aws:s3:::*",
    ]
  }

  statement {
    actions = [
      "s3:ListBucket",
    ]

    resources = [
      "arn:aws:s3:::${var.s3_bucket_name}",
    ]

    condition {
      test     = "StringLike"
      variable = "s3:prefix"

      values = [
        "",
        "home/",
        "home/&{aws:username}/",
      ]
    }
  }

  statement {
    actions = [
      "s3:*",
    ]

    resources = [
      "arn:aws:s3:::${var.s3_bucket_name}/home/&{aws:username}",
      "arn:aws:s3:::${var.s3_bucket_name}/home/&{aws:username}/*",
    ]
  }
}

resource "aws_iam_policy" "example" {
  name   = "example_policy"
  path   = "/"
  policy = "${data.aws_iam_policy_document.example.json}"
}
```

Using this data source to generate policy documents is *optional.* It is also valid to use literal JSON strings within your configuration, or to use the `file` interpolation function to read a raw JSON policy document from a file.

» Argument Reference

The following arguments are supported:

- `policy_id` (Optional) - An ID for the policy document.
- `source_json` (Optional) - An IAM policy document to import as a base for the current policy document. Statements with non-blank `sid`s in the current policy document will overwrite statements with the same `sid` in the source json. Statements without an `sid` cannot be overwritten.
- `override_json` (Optional) - An IAM policy document to import and override the current policy document. Statements with non-blank `sid`s in the override document will overwrite statements with the same `sid` in the current document. Statements without an `sid` cannot be overwritten.
- `statement` (Optional) - A nested configuration block (described below) configuring one *statement* to be included in the policy document.
- `version` (Optional) - IAM policy document version. Valid values: `2008-10-17`, `2012-10-17`. Defaults to `2012-10-17`. For more information, see the AWS IAM User Guide.

Each document configuration may have one or more `statement` blocks, which each accept the following arguments:

- `sid` (Optional) - An ID for the policy statement.
- `effect` (Optional) - Either "Allow" or "Deny", to specify whether this statement allows or denies the given actions. The default is "Allow".
- `actions` (Optional) - A list of actions that this statement either allows or denies. For example, `["ec2:RunInstances", "s3:*"]`.
- `not_actions` (Optional) - A list of actions that this statement does *not* apply to. Used to apply a policy statement to all actions *except* those listed.
- `resources` (Optional) - A list of resource ARNs that this statement applies to. This is required by AWS if used for an IAM policy.
- `not_resources` (Optional) - A list of resource ARNs that this statement does *not* apply to. Used to apply a policy statement to all resources *except* those listed.
- `principals` (Optional) - A nested configuration block (described below) specifying a resource (or resource pattern) to which this statement applies.
- `not_principals` (Optional) - Like `principals` except gives resources that the statement does *not* apply to.
- `condition` (Optional) - A nested configuration block (described below) that defines a further, possibly-service-specific condition that constrains whether this statement applies.

Each policy may have either zero or more `principals` blocks or zero or more `not_principals` blocks, both of which each accept the following arguments:

- `type` (Required) The type of principal. For AWS ARNs this is "AWS". For AWS services (e.g. Lambda), this is "Service".
- `identifiers` (Required) List of identifiers for principals. When `type` is "AWS", these are IAM user or role ARNs. When `type` is "Service", these are AWS Service roles e.g. `lambda.amazonaws.com`.

Each policy statement may have zero or more `condition` blocks, which each accept the following arguments:

- `test` (Required) The name of the IAM condition operator to evaluate.
- `variable` (Required) The name of a Context Variable to apply the condition to. Context variables may either be standard AWS variables starting with `aws:`, or service-specific variables prefixed with the service name.
- `values` (Required) The values to evaluate the condition against. If multiple values are provided, the condition matches if at least one of them applies. (That is, the tests are combined with the ”OR” boolean operation.)

When multiple `condition` blocks are provided, they must *all* evaluate to true for the policy statement to apply. (In other words, the conditions are combined with the ”AND” boolean operation.)

» Context Variable Interpolation

The IAM policy document format allows context variables to be interpolated into various strings within a statement. The native IAM policy document format uses `${...}`-style syntax that is in conflict with Terraform's interpolation syntax, so this data source instead uses `&{...}` syntax for interpolations that should be processed by AWS rather than by Terraform.

» Wildcard Principal

In order to define wildcard principal (a.k.a. anonymous user) use `type = "*"` and `identifiers = ["*"]`. In that case the rendered json will contain `"Principal": "*"`. Note, that even though the IAM Documentation states that `"Principal": "*"` and `"Principal": {"AWS": "*"}` are equivalent, those principals have different behavior for IAM Role Trust Policy. Therefore Terraform will normalize the principal field only in above-mentioned case and principals like `type = "AWS"` and `identifiers = ["*"]` will be rendered as `"Principal": {"AWS": "*"}`.

» Attributes Reference

The following attribute is exported:

- `json` - The above arguments serialized as a standard JSON policy document.

» Example with Multiple Principals

Showing how you can use this as an assume role policy as well as showing how you can specify multiple principal blocks with different types.

```
data "aws_iam_policy_document" "event_stream_bucket_role_assume_role_policy" {
  statement {
    actions = ["sts:AssumeRole"]

    principals {
      type        = "Service"
      identifiers = ["firehose.amazonaws.com"]
    }

    principals {
      type        = "AWS"
      identifiers = ["${var.trusted_role_arn}"]
    }
  }
}
```

» Example with Source and Override

Showing how you can use `source_json` and `override_json`

```
data "aws_iam_policy_document" "source" {
  statement {
    actions   = ["ec2:*"]
    resources = ["*"]
  }

  statement {
    sid = "SidToOverwrite"

    actions   = ["s3:*"]
    resources = ["*"]
  }
}

data "aws_iam_policy_document" "source_json_example" {
  source_json = "${data.aws_iam_policy_document.source.json}"

  statement {
    sid = "SidToOverwrite"

    actions = ["s3:*"]
    resources = [
      "arn:aws:s3:::somebucket",
      "arn:aws:s3:::somebucket/*",
    ]
  }
}

data "aws_iam_policy_document" "override" {
  statement {
    sid = "SidToOverwrite"

    actions   = ["s3:*"]
    resources = ["*"]
  }
}

data "aws_iam_policy_document" "override_json_example" {
  override_json = "${data.aws_iam_policy_document.override.json}"

  statement {
    actions   = ["ec2:*"]
    resources = ["*"]
  }

  statement {
    sid = "SidToOverwrite"

    actions = ["s3:*"]

    resources = [
      "arn:aws:s3:::somebucket",
      "arn:aws:s3:::somebucket/*",
    ]
  }
}
```

`data.aws_iam_policy_document.source_json_example.json` will evaluate to:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*"
    },
    {
      "Sid": "SidToOverwrite",
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::somebucket/*",
        "arn:aws:s3:::somebucket"
      ]
    }
  ]
}
```

`data.aws_iam_policy_document.override_json_example.json` will evaluate to:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*"
    },
    {
      "Sid": "SidToOverwrite",
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "*"
    }
  ]
}
```

You can also combine `source_json` and `override_json` in the same document.

» Example without Statement

Use without a `statement`:

```
data "aws_iam_policy_document" "source" {
  statement {
    sid       = "OverridePlaceholder"
    actions   = ["ec2:DescribeAccountAttributes"]
    resources = ["*"]
  }
}

data "aws_iam_policy_document" "override" {
  statement {
    sid       = "OverridePlaceholder"
    actions   = ["s3:GetObject"]
    resources = ["*"]
  }
}

data "aws_iam_policy_document" "politik" {
  source_json   = "${data.aws_iam_policy_document.source.json}"
  override_json = "${data.aws_iam_policy_document.override.json}"
}
```

`data.aws_iam_policy_document.politik.json` will evaluate to:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "OverridePlaceholder",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "*"
    }
  ]
}
```

» Data Source: aws_iam_role

This data source can be used to fetch information about a specific IAM role. By using this data source, you can reference IAM role properties without having to hard code ARNs as input.

» Example Usage

```
data "aws_iam_role" "example" {
  name = "an_example_role_name"
}
```

» Argument Reference

- `name` - (Required) The friendly IAM role name to match.

» Attributes Reference

- `id` - The friendly IAM role name to match.
- `arn` - The Amazon Resource Name (ARN) specifying the role.
- `assume_role_policy` - The policy document associated with the role.
- `create_date` - Creation date of the role in RFC 3339 format.
- `description` - Description for the role.
- `max_session_duration` - Maximum session duration.
- `path` - The path to the role.
- `permissions_boundary` - The ARN of the policy that is used to set the permissions boundary for the role.
- `unique_id` - The stable and unique string identifying the role.

» Data Source: aws_iam_server_certificate

Use this data source to lookup information about IAM Server Certificates.

» Example Usage

```
data "aws_iam_server_certificate" "my-domain" {
  name_prefix = "my-domain.org"
  latest      = true
}

resource "aws_elb" "elb" {
  name = "my-domain-elb"

  listener {
    instance_port      = 8000
    instance_protocol  = "https"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "${data.aws_iam_server_certificate.my-domain.arn}"
  }
}
```

» Argument Reference

- `name_prefix` - prefix of cert to filter by
- `path_prefix` - prefix of path to filter by
- `name` - exact name of the cert to lookup
- `latest` - sort results by expiration date. returns the certificate with expiration date in furthest in the future.

» Attributes Reference

- `arn` is set to the ARN of the IAM Server Certificate
- `path` is set to the path of the IAM Server Certificate
- `expiration_date` is set to the expiration date of the IAM Server Certificate
- `upload_date` is the date when the server certificate was uploaded
- `certificate_body` is the public key certificate (PEM-encoded). This is useful when configuring back-end instance authentication policy for load balancer
- `certificate_chain` is the public key certificate chain (PEM-encoded) if exists, empty otherwise

» Import

The terraform import function will read in certificate body, certificate chain (if it exists), id, name, path, and arn. It will not retrieve the private key which is not available through the AWS API.

» Data Source: aws_iam_user

This data source can be used to fetch information about a specific IAM user. By using this data source, you can reference IAM user properties without having to hard code ARNs or unique IDs as input.

» Example Usage

```
data "aws_iam_user" "example" {
  user_name = "an_example_user_name"
}
```

» Argument Reference

- `user_name` - (Required) The friendly IAM user name to match.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) assigned by AWS for this user.
- `path` - Path in which this user was created.
- `permissions_boundary` - The ARN of the policy that is used to set the permissions boundary for the user.
- `user_id` - The unique ID assigned by AWS for this user.
- `user_name` - The name associated to this User

» Resource: aws_iam_access_key

Provides an IAM access key. This is a set of credentials that allow API requests to be made as an IAM user.

» Example Usage

```
resource "aws_iam_access_key" "lb" {
  user    = "${aws_iam_user.lb.name}"
  pgp_key = "keybase:some_person_that_exists"
}

resource "aws_iam_user" "lb" {
  name = "loadbalancer"
  path = "/system/"
}

resource "aws_iam_user_policy" "lb_ro" {
  name = "test"
  user = "${aws_iam_user.lb.name}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}

output "secret" {
  value = "${aws_iam_access_key.lb.encrypted_secret}"
}
```

» Argument Reference

The following arguments are supported:

- `user` - (Required) The IAM user to associate with this access key.
- `pgp_key` - (Optional) Either a base-64 encoded PGP public key, or a keybase username in the form `keybase:some_person_that_exists`, for use in the `encrypted_secret` output attribute.
- `status` - (Optional) The access key status to apply. Defaults to `Active`. Valid values are `Active` and `Inactive`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The access key ID.
- `user` - The IAM user associated with this access key.
- `key_fingerprint` - The fingerprint of the PGP key used to encrypt the secret
- `secret` - The secret access key. Note that this will be written to the state file. If you use this, please protect your backend state file judiciously. Alternatively, you may supply a `pgp_key` instead, which will prevent the secret from being stored in plaintext, at the cost of preventing the use of the secret key in automation.
- `encrypted_secret` - The encrypted secret, base64 encoded, if `pgp_key` was specified. ~> **NOTE:** The encrypted secret may be decrypted using the command line, for example: `terraform output encrypted_secret | base64 --decode | keybase pgp decrypt`.
- `ses_smtp_password` - The secret access key converted into an SES SMTP password by applying AWS's documented conversion algorithm.

» Resource: aws_iam_account_alias

Note: There is only a single account alias per AWS account.

Manages the account alias for the AWS Account.

» Example Usage

```
resource "aws_iam_account_alias" "alias" {
  account_alias = "my-account-alias"
}
```

» Argument Reference

The following arguments are supported:

- `account_alias` - (Required) The account alias

» Import

The current Account Alias can be imported using the `account_alias`, e.g.

```
$ terraform import aws_iam_account_alias.alias my-account-alias
```

» Resource: aws_iam_account_password_policy

Note: There is only a single policy allowed per AWS account. An existing policy will be lost when using this resource as an effect of this limitation.

Manages Password Policy for the AWS Account. See more about Account Password Policy in the official AWS docs.

» Example Usage

```
resource "aws_iam_account_password_policy" "strict" {
  minimum_password_length        = 8
  require_lowercase_characters   = true
  require_numbers                = true
  require_uppercase_characters   = true
  require_symbols                = true
  allow_users_to_change_password = true
}
```

» Argument Reference

The following arguments are supported:

- `allow_users_to_change_password` - (Optional) Whether to allow users to change their own password
- `hard_expiry` - (Optional) Whether users are prevented from setting a new password after their password has expired (i.e. require administrator reset)
- `max_password_age` - (Optional) The number of days that an user password is valid.
- `minimum_password_length` - (Optional) Minimum length to require for user passwords.
- `password_reuse_prevention` - (Optional) The number of previous passwords that users are prevented from reusing.
- `require_lowercase_characters` - (Optional) Whether to require lowercase characters for user passwords.
- `require_numbers` - (Optional) Whether to require numbers for user passwords.
- `require_symbols` - (Optional) Whether to require symbols for user passwords.
- `require_uppercase_characters` - (Optional) Whether to require uppercase characters for user passwords.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `expire_passwords` - Indicates whether passwords in the account expire. Returns `true` if `max_password_age` contains a value greater than `0`. Returns `false` if it is `0` or *not present*.

» Import

IAM Account Password Policy can be imported using the word `iam-account-password-policy`, e.g.

```
$ terraform import aws_iam_account_password_policy.strict iam-account-password-policy
```

» Resource: aws_iam_group

Provides an IAM group.

» Example Usage

```
resource "aws_iam_group" "developers" {
  name = "developers"
  path = "/users/"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The group's name. The name must consist of upper and lowercase alphanumeric characters with no spaces. You can also include any of the following characters: `=,.@-_.`. Group names are not distinguished by case. For example, you cannot create groups named both "ADMINS" and "admins".
- `path` - (Optional, default "/") Path in which to create the group.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The group's ID.
- `arn` - The ARN assigned by AWS for this group.
- `name` - The group's name.
- `path` - The path of the group in IAM.
- `unique_id` - The unique ID assigned by AWS.

» Import

IAM Groups can be imported using the `name`, e.g.

```
$ terraform import aws_iam_group.developers developers
```

» Resource: aws_iam_group_membership

WARNING: Multiple aws_iam_group_membership resources with the same group name will produce inconsistent behavior!

Provides a top level resource to manage IAM Group membership for IAM Users. For more information on managing IAM Groups or IAM Users, see IAM Groups or IAM Users

Note: `aws_iam_group_membership` will conflict with itself if used more than once with the same group. To non-exclusively manage the users in a group, see the `aws_iam_user_group_membership` resource.

» Example Usage

```
resource "aws_iam_group_membership" "team" {
  name = "tf-testing-group-membership"

  users = [
    "${aws_iam_user.user_one.name}",
    "${aws_iam_user.user_two.name}",
  ]

  group = "${aws_iam_group.group.name}"
}

resource "aws_iam_group" "group" {
  name = "test-group"
}

resource "aws_iam_user" "user_one" {
  name = "test-user"
}

resource "aws_iam_user" "user_two" {
  name = "test-user-two"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name to identify the Group Membership
- `users` - (Required) A list of IAM User names to associate with the Group
- `group` – (Required) The IAM Group name to attach the list of `users` to

» Attributes Reference

- `name` - The name to identify the Group Membership
- `users` - list of IAM User names
- `group` – IAM Group name

» Resource: aws_iam_group_policy

Provides an IAM policy attached to a group.

» Example Usage

```
resource "aws_iam_group_policy" "my_developer_policy" {
  name  = "my_developer_policy"
  group = "${aws_iam_group.my_developers.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}

resource "aws_iam_group" "my_developers" {
  name = "developers"
  path = "/users/"
}
```

» Argument Reference

The following arguments are supported:

- `policy` - (Required) The policy document. This is a JSON formatted string. For more information about building IAM policy documents with Terraform, see the AWS IAM Policy Document Guide
- `name` - (Optional) The name of the policy. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `group` - (Required) The IAM group to attach to the policy.

» Attributes Reference

- `id` - The group policy ID.
- `group` - The group to which this policy applies.
- `name` - The name of the policy.
- `policy` - The policy document attached to the group.

» Import

IAM Group Policies can be imported using the `group_name:group_policy_name`, e.g.

```
$ terraform import aws_iam_group_policy.mypolicy group_of_mypolicy_name:mypolicy_name
```

» Resource: aws_iam_group_policy_attachment

Attaches a Managed IAM Policy to an IAM group

NOTE: The usage of this resource conflicts with the `aws_iam_policy_attachment` resource and will permanently show a difference if both are defined.

» Example Usage

```
resource "aws_iam_group" "group" {
  name = "test-group"
}

resource "aws_iam_policy" "policy" {
  name        = "test-policy"
  description = "A test policy"
  policy      = ""              # insert policy here
}

resource "aws_iam_group_policy_attachment" "test-attach" {
  group      = "${aws_iam_group.group.name}"
  policy_arn = "${aws_iam_policy.policy.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `group` (Required) - The group the policy should be applied to

- `policy_arn` (Required) - The ARN of the policy you want to apply

» Import

IAM group policy attachments can be imported using the group name and policy arn separated by `/`.

```
$ terraform import aws_iam_group_policy_attachment.test-attach test-group/arn:aws:iam::xxxxx
```

» Resource: aws_iam_instance_profile

Provides an IAM instance profile.

» Example Usage

```
resource "aws_iam_instance_profile" "test_profile" {
  name = "test_profile"
  role = "${aws_iam_role.role.name}"
}

resource "aws_iam_role" "role" {
  name = "test_role"
  path = "/"

  assume_role_policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Principal": {
               "Service": "ec2.amazonaws.com"
            },
            "Effect": "Allow",
            "Sid": ""
        }
    ]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The profile's name. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `path` - (Optional, default "/") Path in which to create the profile.
- `roles` - (**Deprecated**) A list of role names to include in the profile. The current default is 1. If you see an error message similar to `Cannot exceed quota for InstanceSessionsPerInstanceProfile: 1`, then you must contact AWS support and ask for a limit increase. WARNING: This is deprecated since version 0.9.3 (April 12, 2017), as >= 2 roles are not possible. See issue #11575.
- `role` - (Optional) The role name to include in the profile.

» Attribute Reference

- `id` - The instance profile's ID.
- `arn` - The ARN assigned by AWS to the instance profile.
- `create_date` - The creation timestamp of the instance profile.
- `name` - The instance profile's name.
- `path` - The path of the instance profile in IAM.
- `role` - The role assigned to the instance profile.
- `roles` - The list of roles assigned to the instance profile. (**Deprecated**)
- `unique_id` - The unique ID assigned by AWS.

» Import

Instance Profiles can be imported using the `name`, e.g.

```
$ terraform import aws_iam_instance_profile.test_profile app-instance-profile-1
```

» Resource: aws_iam_openid_connect_provider

Provides an IAM OpenID Connect provider.

» Example Usage

```
resource "aws_iam_openid_connect_provider" "default" {
  url = "https://accounts.google.com"
  client_id_list = [
    "266362248691-342342xasdasdasda-apps.googleusercontent.com",
  ]

  thumbprint_list = []
}
```

» Argument Reference

The following arguments are supported:

- `url` - (Required) The URL of the identity provider. Corresponds to the *iss* claim.
- `client_id_list` - (Required) A list of client IDs (also known as audiences). When a mobile or web app registers with an OpenID Connect provider, they establish a value that identifies the application. (This is the value that's sent as the client_id parameter on OAuth requests.)
- `thumbprint_list` - (Required) A list of server certificate thumbprints for the OpenID Connect (OIDC) identity provider's server certificate(s).

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS for this provider.

» Import

IAM OpenID Connect Providers can be imported using the `arn`, e.g.

```
$ terraform import aws_iam_openid_connect_provider.default arn:aws:iam::123456789012:oidc-pr
```

» Resource: aws_iam_policy

Provides an IAM policy.

» Example Usage

```
resource "aws_iam_policy" "policy" {
  name        = "test_policy"
  path        = "/"
  description = "My test policy"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Optional, Forces new resource) Description of the IAM policy.
- `name` - (Optional, Forces new resource) The name of the policy. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `path` - (Optional, default "/") Path in which to create the policy. See IAM Identifiers for more information.
- `policy` - (Required) The policy document. This is a JSON formatted string. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The policy's ID.
- `arn` - The ARN assigned by AWS to this policy.
- `description` - The description of the policy.
- `name` - The name of the policy.
- `path` - The path of the policy in IAM.
- `policy` - The policy document.

» Import

IAM Policies can be imported using the `arn`, e.g.

```
$ terraform import aws_iam_policy.administrator arn:aws:iam::123456789012:policy/UsersManage
```

» Resource: aws_iam_policy_attachment

Attaches a Managed IAM Policy to user(s), role(s), and/or group(s)

WARNING: The aws_iam_policy_attachment resource creates **exclusive** attachments of IAM policies. Across the entire AWS account, all of the users/roles/groups to which a single policy is attached must be declared by a single aws_iam_policy_attachment resource. This means that even any users/roles/groups that have the attached policy via any other mechanism (including other Terraform resources) will have that attached policy revoked by this resource. Consider `aws_iam_role_policy_attachment`, `aws_iam_user_policy_attachment`, or `aws_iam_group_policy_attachment` instead. These resources do not enforce exclusive attachment of an IAM policy.

NOTE: The usage of this resource conflicts with the `aws_iam_group_policy_attachment`, `aws_iam_role_policy_attachment`, and `aws_iam_user_policy_attachment` resources and will permanently show a difference if both are defined.

» Example Usage

```
resource "aws_iam_user" "user" {
  name = "test-user"
}

resource "aws_iam_role" "role" {
  name = "test-role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_iam_group" "group" {
  name = "test-group"
}

resource "aws_iam_policy" "policy" {
  name        = "test-policy"
  description = "A test policy"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}

resource "aws_iam_policy_attachment" "test-attach" {
  name       = "test-attachment"
  users      = ["${aws_iam_user.user.name}"]
  roles      = ["${aws_iam_role.role.name}"]
  groups     = ["${aws_iam_group.group.name}"]
  policy_arn = "${aws_iam_policy.policy.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - The name of the attachment. This cannot be an empty string.
- `users` (Optional) - The user(s) the policy should be applied to
- `roles` (Optional) - The role(s) the policy should be applied to

- `groups` (Optional) - The group(s) the policy should be applied to
- `policy_arn` (Required) - The ARN of the policy you want to apply

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The policy's ID.
- `name` - The name of the attachment.

» Resource: aws_iam_role

Provides an IAM role.

NOTE: If policies are attached to the role via the `aws_iam_policy_attachment` resource and you are modifying the role `name` or `path`, the `force_detach_policies` argument must be set to `true` and applied before attempting the operation otherwise you will encounter a `DeleteConflict` error. The `aws_iam_role_policy_attachment` resource (recommended) does not have this requirement.

» Example Usage

```
resource "aws_iam_role" "test_role" {
  name = "test_role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF

  tags = {
    tag-key = "tag-value"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the role. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `assume_role_policy` - (Required) The policy that grants an entity permission to assume the role.

NOTE: This `assume_role_policy` is very similar but slightly different than just a standard IAM policy and cannot use an `aws_iam_policy` resource. It *can* however, use an `aws_iam_policy_document` data source, see example below for how this could work.

- `force_detach_policies` - (Optional) Specifies to force detaching any policies the role has before destroying it. Defaults to `false`.
- `path` - (Optional) The path to the role. See IAM Identifiers for more information.
- `description` - (Optional) The description of the role.
- `max_session_duration` - (Optional) The maximum session duration (in seconds) that you want to set for the specified role. If you do not specify a value for this setting, the default maximum of one hour is applied. This setting can have a value from 1 hour to 12 hours.
- `permissions_boundary` - (Optional) The ARN of the policy that is used to set the permissions boundary for the role.
- `tags` - Key-value mapping of tags for the IAM role

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) specifying the role.
- `create_date` - The creation date of the IAM role.
- `description` - The description of the role.
- `id` - The name of the role.
- `name` - The name of the role.
- `unique_id` - The stable and unique string identifying the role.

» Example of Using Data Source for Assume Role Policy

```
data "aws_iam_policy_document" "instance-assume-role-policy" {
  statement {
    actions = ["sts:AssumeRole"]

    principals {
      type        = "Service"
      identifiers = ["ec2.amazonaws.com"]
    }
  }
}

resource "aws_iam_role" "instance" {
  name               = "instance_role"
  path               = "/system/"
  assume_role_policy = "${data.aws_iam_policy_document.instance-assume-role-policy.json}"
}
```

» Import

IAM Roles can be imported using the `name`, e.g.

```
$ terraform import aws_iam_role.developer developer_name
```

» Resource: aws_iam_role_policy

Provides an IAM role policy.

» Example Usage

```
resource "aws_iam_role_policy" "test_policy" {
  name = "test_policy"
  role = "${aws_iam_role.test_role.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}

resource "aws_iam_role" "test_role" {
  name = "test_role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the role policy. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `policy` - (Required) The policy document. This is a JSON formatted string. For more information about building IAM policy documents with Terraform, see the AWS IAM Policy Document Guide
- `role` - (Required) The IAM role to attach to the policy.

» Attributes Reference

- `id` - The role policy ID, in the form of `role_name:role_policy_name`.

- `name` - The name of the policy.
- `policy` - The policy document attached to the role.
- `role` - The name of the role associated with the policy.

» Import

IAM Role Policies can be imported using the `role_name:role_policy_name`, e.g.

```
$ terraform import aws_iam_role_policy.mypolicy role_of_mypolicy_name:mypolicy_name
```

» Resource: aws_iam_role_policy_attachment

Attaches a Managed IAM Policy to an IAM role

NOTE: The usage of this resource conflicts with the `aws_iam_policy_attachment` resource and will permanently show a difference if both are defined.

» Example Usage

```
resource "aws_iam_role" "role" {
  name = "test-role"

  assume_role_policy = <<EOF
    {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          },
          "Effect": "Allow",
          "Sid": ""
        }
      ]
    }
EOF
}

resource "aws_iam_policy" "policy" {
  name        = "test-policy"
  description = "A test policy"
  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy_attachment" "test-attach" {
  role       = "${aws_iam_role.role.name}"
  policy_arn = "${aws_iam_policy.policy.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `role` (Required) - The role the policy should be applied to
- `policy_arn` (Required) - The ARN of the policy you want to apply

» Import

IAM role policy attachments can be imported using the role name and policy arn separated by `/`.

```
$ terraform import aws_iam_role_policy_attachment.test-attach test-role/arn:aws:iam::xxxxxxx
```

» Resource: aws_iam_saml_provider

Provides an IAM SAML provider.

» Example Usage

```
resource "aws_iam_saml_provider" "default" {
  name                   = "myprovider"
  saml_metadata_document = "${file("saml-metadata.xml")}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the provider to create.
- `saml_metadata_document` - (Required) An XML document generated by an identity provider that supports SAML 2.0.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS for this provider.
- `valid_until` - The expiration date and time for the SAML provider in RFC1123 format, e.g. `Mon, 02 Jan 2006 15:04:05 MST`.

» Import

IAM SAML Providers can be imported using the `arn`, e.g.

```
$ terraform import aws_iam_saml_provider.default arn:aws:iam::123456789012:saml-provider/SAM
```

» Resource: aws_iam_server_certificate

Provides an IAM Server Certificate resource to upload Server Certificates. Certs uploaded to IAM can easily work with other AWS services such as:

- AWS Elastic Beanstalk
- Elastic Load Balancing
- CloudFront
- AWS OpsWorks

For information about server certificates in IAM, see Managing Server Certificates in AWS Documentation.

Note: All arguments including the private key will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

Using certs on file:

```
resource "aws_iam_server_certificate" "test_cert" {
  name             = "some_test_cert"
  certificate_body = "${file("self-ca-cert.pem")}"
  private_key      = "${file("test-key.pem")}"
}
```

Example with cert in-line:

```
resource "aws_iam_server_certificate" "test_cert_alt" {
  name = "alt_test_cert"

  certificate_body = <<EOF
-----BEGIN CERTIFICATE-----
[......] # cert contents
-----END CERTIFICATE-----
EOF

  private_key = <<EOF
-----BEGIN RSA PRIVATE KEY-----
[......] # cert contents
-----END RSA PRIVATE KEY-----
EOF
}
```

Use in combination with an AWS ELB resource:

Some properties of an IAM Server Certificates cannot be updated while they are in use. In order for Terraform to effectively manage a Certificate in this situation, it is recommended you utilize the `name_prefix` attribute and enable the `create_before_destroy` lifecycle block. This will allow Terraform to create a new, updated `aws_iam_server_certificate` resource and replace it in dependant resources before attempting to destroy the old version.

```
resource "aws_iam_server_certificate" "test_cert" {
  name_prefix      = "example-cert"
  certificate_body = "${file("self-ca-cert.pem")}"
  private_key      = "${file("test-key.pem")}"

  lifecycle {
    create_before_destroy = true
  }
}

resource "aws_elb" "ourapp" {
  name                      = "terraform-asg-deployment-example"
  availability_zones        = ["us-west-2a"]
  cross_zone_load_balancing = true

  listener {
    instance_port      = 8000
    instance_protocol  = "http"
    lb_port            = 443
    lb_protocol        = "https"
    ssl_certificate_id = "${aws_iam_server_certificate.test_cert.arn}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the Server Certificate. Do not include the path in this value. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `certificate_body` – (Required) The contents of the public key certificate in PEM-encoded format.
- `certificate_chain` – (Optional) The contents of the certificate chain. This is typically a concatenation of the PEM-encoded public key certificates of the chain.
- `private_key` – (Required) The contents of the private key in PEM-encoded format.
- `path` - (Optional) The IAM path for the server certificate. If it is not included, it defaults to a slash (/). If this certificate is for use with AWS CloudFront, the path must be in format `/cloudfront/your_path_here`. See IAM Identifiers for more details on IAM Paths.

NOTE: AWS performs behind-the-scenes modifications to some certificate files if they do not adhere to a specific format. These modifications will result in terraform forever believing that it needs to update the resources since the local and AWS file contents will not match after theses modifications occur. In order to prevent this from happening you must ensure that all your PEM-encoded files use UNIX line-breaks and that `certificate_body` contains only one certificate. All other certificates should go in `certificate_chain`. It is common for some Certificate Authorities to issue certificate files that have DOS line-breaks and that are actually multiple certificates concatenated together in order to form a full certificate chain.

» Attributes Reference

- `id` - The unique Server Certificate name
- `name` - The name of the Server Certificate
- `arn` - The Amazon Resource Name (ARN) specifying the server certificate.

» Import

IAM Server Certificates can be imported using the `name`, e.g.

```
$ terraform import aws_iam_server_certificate.certificate example.com-certificate-until-2018
```

» Resource: aws_iam_service_linked_role

Provides an IAM service-linked role.

» Example Usage

```
resource "aws_iam_service_linked_role" "elasticbeanstalk" {
  aws_service_name = "elasticbeanstalk.amazonaws.com"
}
```

» Argument Reference

The following arguments are supported:

- `aws_service_name` - (Required, Forces new resource) The AWS service to which this role is attached. You use a string similar to a URL but without the `http://` in front. For example: `elasticbeanstalk.amazonaws.com`. To find the full list of services that support service-linked roles, check the docs.
- `custom_suffix` - (Optional, forces new resource) Additional string appended to the role name. Not all AWS services support custom suffixes.
- `description` - (Optional) The description of the role.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) of the role.
- `arn` - The Amazon Resource Name (ARN) specifying the role.
- `create_date` - The creation date of the IAM role.

- `name` - The name of the role.
- `path` - The path of the role.
- `unique_id` - The stable and unique string identifying the role.

» Import

IAM service-linked roles can be imported using role ARN, e.g.

```
$ terraform import aws_iam_service_linked_role.elasticbeanstalk arn:aws:iam::123456789012:ro
```

» Resource: aws_iam_user

Provides an IAM user.

NOTE: If policies are attached to the user via the `aws_iam_policy_attachment` resource and you are modifying the user `name` or `path`, the `force_destroy` argument must be set to `true` and applied before attempting the operation otherwise you will encounter a `DeleteConflict` error. The `aws_iam_user_policy_attachment` resource (recommended) does not have this requirement.

» Example Usage

```
resource "aws_iam_user" "lb" {
  name = "loadbalancer"
  path = "/system/"

  tags = {
    tag-key = "tag-value"
  }
}

resource "aws_iam_access_key" "lb" {
  user = "${aws_iam_user.lb.name}"
}

resource "aws_iam_user_policy" "lb_ro" {
  name = "test"
  user = "${aws_iam_user.lb.name}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The user's name. The name must consist of upper and lowercase alphanumeric characters with no spaces. You can also include any of the following characters: `=,.@-_.`. User names are not distinguished by case. For example, you cannot create users named both "TESTUSER" and "testuser".
- `path` - (Optional, default "/") Path in which to create the user.
- `permissions_boundary` - (Optional) The ARN of the policy that is used to set the permissions boundary for the user.
- `force_destroy` - (Optional, default false) When destroying this user, destroy even if it has non-Terraform-managed IAM access keys, login profile or MFA devices. Without `force_destroy` a user with non-Terraform-managed access keys and login profile will fail to be destroyed.
- `tags` - Key-value mapping of tags for the IAM user

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS for this user.
- `name` - The user's name.
- `unique_id` - The unique ID assigned by AWS.

» Import

IAM Users can be imported using the `name`, e.g.

```
$ terraform import aws_iam_user.lb loadbalancer
```

» Resource: aws_iam_user_group_membership

Provides a resource for adding an IAM User to IAM Groups. This resource can be used multiple times with the same user for non-overlapping groups.

To exclusively manage the users in a group, see the `aws_iam_group_membership` resource.

» Example usage

```
resource "aws_iam_user_group_membership" "example1" {
  user = "${aws_iam_user.user1.name}"

  groups = [
    "${aws_iam_group.group1.name}",
    "${aws_iam_group.group2.name}",
  ]
}

resource "aws_iam_user_group_membership" "example2" {
  user = "${aws_iam_user.user1.name}"

  groups = [
    "${aws_iam_group.group3.name}",
  ]
}

resource "aws_iam_user" "user1" {
  name = "user1"
}

resource "aws_iam_group" "group1" {
  name = "group1"
}

resource "aws_iam_group" "group2" {
  name = "group2"
}

resource "aws_iam_group" "group3" {
  name = "group3"
}
```

» Argument Reference

The following arguments are supported:

- `user` - (Required) The name of the IAM User to add to groups
- `groups` - (Required) A list of IAM Groups to add the user to

» Attributes Reference

- `user` - The name of the IAM User
- `groups` - The list of IAM Groups

» Import

IAM user group membership can be imported using the user name and group names separated by `/`.

```
$ terraform import aws_iam_user_group_membership.example1 user1/group1/group2
```

» Resource: aws_iam_user_login_profile

Manages an IAM User Login Profile with limited support for password creation during Terraform resource creation. Uses PGP to encrypt the password for safe transport to the user. PGP keys can be obtained from Keybase.

To reset an IAM User login password via Terraform, you can use the `terraform taint` command or change any of the arguments.

» Example Usage

```
resource "aws_iam_user" "example" {
  name          = "example"
  path          = "/"
  force_destroy = true
}

resource "aws_iam_user_login_profile" "example" {
  user    = "${aws_iam_user.example.name}"
  pgp_key = "keybase:some_person_that_exists"
}

output "password" {
  value = "${aws_iam_user_login_profile.example.encrypted_password}"
}
```

» Argument Reference

The following arguments are supported:

- `user` - (Required) The IAM user's name.
- `pgp_key` - (Required) Either a base-64 encoded PGP public key, or a keybase username in the form `keybase:username`. Only applies on resource creation. Drift detection is not possible with this argument.
- `password_length` - (Optional, default 20) The length of the generated password on resource creation. Only applies on resource creation. Drift detection is not possible with this argument.
- `password_reset_required` - (Optional, default "true") Whether the user should be forced to reset the generated password on resource creation. Only applies on resource creation. Drift detection is not possible with this argument.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `key_fingerprint` - The fingerprint of the PGP key used to encrypt the password. Only available if password was handled on Terraform resource creation, not import.
- `encrypted_password` - The encrypted password, base64 encoded. Only available if password was handled on Terraform resource creation, not import.

NOTE: The encrypted password may be decrypted using the command line, for example: `terraform output password | base64 --decode | keybase pgp decrypt`.

» Import

IAM User Login Profiles can be imported without password information support via the IAM User name, e.g.

```
$ terraform import aws_iam_user_login_profile.example myusername
```

Since Terraform has no method to read the PGP or password information during import, use the Terraform resource `lifecycle` configuration block `ignore_changes` argument to ignore them unless password recreation is desired. e.g.

```
resource "aws_iam_user_login_profile" "example" {
  # ... other configuration ...

  lifecycle {
    ignore_changes = ["password_length", "password_reset_required", "pgp_key"]
  }
}
```

» Resource: aws_iam_user_policy

Provides an IAM policy attached to a user.

» Example Usage

```
resource "aws_iam_user_policy" "lb_ro" {
  name = "test"
  user = "${aws_iam_user.lb.name}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}

resource "aws_iam_user" "lb" {
  name = "loadbalancer"
  path = "/system/"
}

resource "aws_iam_access_key" "lb" {
  user = "${aws_iam_user.lb.name}"
}
```

» Argument Reference

The following arguments are supported:

- `policy` - (Required) The policy document. This is a JSON formatted string. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.
- `name` - (Optional) The name of the policy. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `user` - (Required) IAM user to which to attach this policy.

» Attributes Reference

- `id` - The user policy ID, in the form of `user_name:user_policy_name`.
- `name` - The name of the policy (always set).

» Import

IAM User Policies can be imported using the `user_name:user_policy_name`, e.g.

```
$ terraform import aws_iam_user_policy.mypolicy user_of_mypolicy_name:mypolicy_name
```

» Resource: aws_iam_user_policy_attachment

Attaches a Managed IAM Policy to an IAM user

NOTE: The usage of this resource conflicts with the `aws_iam_policy_attachment` resource and will permanently show a difference if both are defined.

» Example Usage

```
resource "aws_iam_user" "user" {
  name = "test-user"
}

resource "aws_iam_policy" "policy" {
  name        = "test-policy"
  description = "A test policy"
  policy      = ""              # insert policy here
}
```

```
resource "aws_iam_user_policy_attachment" "test-attach" {
  user       = "${aws_iam_user.user.name}"
  policy_arn = "${aws_iam_policy.policy.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `user` (Required) - The user the policy should be applied to
- `policy_arn` (Required) - The ARN of the policy you want to apply

» Import

IAM user policy attachments can be imported using the user name and policy arn separated by `/`.

```
$ terraform import aws_iam_user_policy_attachment.test-attach test-user/arn:aws:iam::xxxxxxx
```

» Resource: aws_iam_user_ssh_key

Uploads an SSH public key and associates it with the specified IAM user.

» Example Usage

```
resource "aws_iam_user" "user" {
  name = "test-user"
  path = "/"
}

resource "aws_iam_user_ssh_key" "user" {
  username   = "${aws_iam_user.user.name}"
  encoding   = "SSH"
  public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD3F6tyPEFEzV0LX3X8BsXdMsQz1x2cEikKDEY
}
```

» Argument Reference

The following arguments are supported:

- `username` - (Required) The name of the IAM user to associate the SSH public key with.
- `encoding` - (Required) Specifies the public key encoding format to use in the response. To retrieve the public key in ssh-rsa format, use `SSH`. To retrieve the public key in PEM format, use `PEM`.
- `public_key` - (Required) The SSH public key. The public key must be encoded in ssh-rsa format or PEM format.
- `status` - (Optional) The status to assign to the SSH public key. Active means the key can be used for authentication with an AWS CodeCommit repository. Inactive means the key cannot be used. Default is `active`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `ssh_public_key_id` - The unique identifier for the SSH public key.
- `fingerprint` - The MD5 message digest of the SSH public key.

» Import

SSH public keys can be imported using the `username`, `ssh_public_key_id`, and `encoding` e.g.

```
$ terraform import aws_iam_user_ssh_key.user user:APKAJNCNNJICVN7CFKCA:SSH
```

» Data Source: aws_iot_endpoint

Returns a unique endpoint specific to the AWS account making the call.

» Example Usage

```
data "aws_iot_endpoint" "example" {}

resource "kubernetes_pod" "agent" {
  metadata {
    name = "my-device"
  }

  spec {
    container {
      image = "gcr.io/my-project/image-name"
      name  = "image-name"
      env = [
        {
          name  = "IOT_ENDPOINT"
          value = "${data.aws_iot_endpoint.example.endpoint_address}"
        },
      ]
    }
  }
}
```

» Argument Reference

- `endpoint_type` - (Optional) Endpoint type. Valid values: `iot:CredentialProvider`, `iot:Data`, `iot:Data-ATS`, `iot:Job`.

» Attributes Reference

- `endpoint_address` - The endpoint based on `endpoint_type`:
  - No `endpoint_type`: Either `iot:Data` or `iot:Data-ATS` depending on region
  - `iot:CredentialsProvider`: `IDENTIFIER.credentials.iot.REGION.amazonaws.com`
  - `iot:Data`: `IDENTIFIER.iot.REGION.amazonaws.com`
  - `iot:Data-ATS`: `IDENTIFIER-ats.iot.REGION.amazonaws.com`
  - `iot:Job`: `IDENTIFIER.jobs.iot.REGION.amazonaws.com`

» Resource: aws_iot_certificate

Creates and manages an AWS IoT certificate.

» Example Usage

» With CSR

```
resource "aws_iot_certificate" "cert" {
  csr    = "${file("/my/csr.pem")}"
  active = true
}
```

» Without CSR

```
resource "aws_iot_certificate" "cert" {
  active = true
}
```

» Argument Reference

- `active` - (Required) Boolean flag to indicate if the certificate should be active
- `csr` - (Optional) The certificate signing request. Review CreateCertificateFromCsr for more information on generating a certificate from a certificate signing request (CSR). If none is specified both the certificate and keys will be generated, review CreateKeysAndCertificate for more information on generating keys and a certificate.

» Attributes Reference

In addition to the arguments, the following attributes are exported:

- `id` - The internal ID assigned to this certificate.
- `arn` - The ARN of the created certificate.
- `certificate_pem` - The certificate data, in PEM format.
- `public_key` - When no CSR is provided, the public key.
- `private_key` - When no CSR is provided, the private key.

» Resource: aws_iot_policy

Provides an IoT policy.

» Example Usage

```
resource "aws_iot_policy" "pubsub" {
  name = "PubSubToAnyTopic"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iot:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the policy.
- `policy` - (Required) The policy document. This is a JSON formatted string. Use the IoT Developer Guide for more information on IoT Policies. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS to this policy.
- `name` - The name of this policy.
- `default_version_id` - The default version of this policy.
- `policy` - The policy document.

» Resource: aws_iot_policy_attachment

Provides an IoT policy attachment.

» Example Usage

```
resource "aws_iot_policy" "pubsub" {
  name = "PubSubToAnyTopic"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iot:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}

resource "aws_iot_certificate" "cert" {
  csr    = "${file("csr.pem")}"
  active = true
}

resource "aws_iot_policy_attachment" "att" {
  policy = "${aws_iot_policy.pubsub.name}"
  target = "${aws_iot_certificate.cert.arn}"
}
```

» Argument Reference

The following arguments are supported:

- policy - (Required) The name of the policy to attach.
- target - (Required) The identity to which the policy is attached.

» Resource: aws_iot_topic_rule

» Example Usage

```
resource "aws_iot_topic_rule" "rule" {
  name        = "MyRule"
  description = "Example rule"
  enabled     = true
  sql         = "SELECT * FROM 'topic/test'"
  sql_version = "2015-10-08"

  sns {
    message_format = "RAW"
    role_arn       = "${aws_iam_role.role.arn}"
    target_arn     = "${aws_sns_topic.mytopic.arn}"
  }
}

resource "aws_sns_topic" "mytopic" {
  name = "mytopic"
}

resource "aws_iam_role" "role" {
  name = "myrole"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "iam_policy_for_lambda" {
  name = "mypolicy"
  role = "${aws_iam_role.role.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "sns:Publish"
        ],
        "Resource": "${aws_sns_topic.mytopic.arn}"
    }
  ]
}
EOF
}
```

» Argument Reference

- `name` - (Required) The name of the rule.
- `description` - (Optional) The description of the rule.
- `enabled` - (Required) Specifies whether the rule is enabled.
- `sql` - (Required) The SQL statement used to query the topic. For more information, see AWS IoT SQL Reference (http://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html#aws-iot-sql-reference) in the AWS IoT Developer Guide.
- `sql_version` - (Required) The version of the SQL rules engine to use when evaluating the rule.

The `cloudwatch_alarm` object takes the following arguments:

- `alarm_name` - (Required) The CloudWatch alarm name.
- `role_arn` - (Required) The IAM role ARN that allows access to the CloudWatch alarm.
- `state_reason` - (Required) The reason for the alarm change.
- `state_value` - (Required) The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT_DATA.

The `cloudwatch_metric` object takes the following arguments:

- `metric_name` - (Required) The CloudWatch metric name.
- `metric_namespace` - (Required) The CloudWatch metric namespace name.
- `metric_timestamp` - (Optional) An optional Unix timestamp (http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/cloudwatch_concepts.html#about_timestamp).
- `metric_unit` - (Required) The metric unit (supported units can be found here: http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/cloudwatch_concepts.html#Unit)
- `metric_value` - (Required) The CloudWatch metric value.
- `role_arn` - (Required) The IAM role ARN that allows access to the CloudWatch metric.

The `dynamodb` object takes the following arguments:

- `hash_key_field` - (Required) The hash key name.
- `hash_key_type` - (Optional) The hash key type. Valid values are ”STRING” or ”NUMBER”.
- `hash_key_value` - (Required) The hash key value.
- `payload_field` - (Optional) The action payload.
- `range_key_field` - (Optional) The range key name.
- `range_key_type` - (Optional) The range key type. Valid values are ”STRING” or ”NUMBER”.
- `range_key_value` - (Optional) The range key value.

- `role_arn` - (Required) The ARN of the IAM role that grants access to the DynamoDB table.
- `table_name` - (Required) The name of the DynamoDB table.

The `elasticsearch` object takes the following arguments:

- `endpoint` - (Required) The endpoint of your Elasticsearch domain.
- `id` - (Required) The unique identifier for the document you are storing.
- `index` - (Required) The Elasticsearch index where you want to store your data.
- `role_arn` - (Required) The IAM role ARN that has access to Elasticsearch.
- `type` - (Required) The type of document you are storing.

The `firehose` object takes the following arguments:

- `delivery_stream_name` - (Required) The delivery stream name.
- `role_arn` - (Required) The IAM role ARN that grants access to the Amazon Kinesis Firehose stream.
- `separator` - (Optional) A character separator that is used to separate records written to the Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ',' (comma).

The `kinesis` object takes the following arguments:

- `partition_key` - (Optional) The partition key.
- `role_arn` - (Required) The ARN of the IAM role that grants access to the Amazon Kinesis stream.
- `stream_name` - (Required) The name of the Amazon Kinesis stream.

The `lambda` object takes the following arguments:

- `function_arn` - (Required) The ARN of the Lambda function.

The `republish` object takes the following arguments:

- `role_arn` - (Required) The ARN of the IAM role that grants access.
- `topic` - (Required) The name of the MQTT topic the message should be republished to.

The `s3` object takes the following arguments:

- `bucket_name` - (Required) The Amazon S3 bucket name.
- `key` - (Required) The object key.
- `role_arn` - (Required) The ARN of the IAM role that grants access.

The `sns` object takes the following arguments:

- `message_format` - (Required) The message format of the message to publish. Accepted values are "JSON" and "RAW".
- `role_arn` - (Required) The ARN of the IAM role that grants access.
- `target_arn` - (Required) The ARN of the SNS topic.

The `sqs` object takes the following arguments:

- `queue_url` - (Required) The URL of the Amazon SQS queue.
- `role_arn` - (Required) The ARN of the IAM role that grants access.
- `use_base64` - (Required) Specifies whether to use Base64 encoding.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the topic rule
- `arn` - The ARN of the topic rule

» Import

IoT Topic Rules can be imported using the `name`, e.g.

```
$ terraform import aws_iot_topic_rule.rule <name>
```

» Resource: aws_iot_thing

Creates and manages an AWS IoT Thing.

» Example Usage

```
resource "aws_iot_thing" "example" {
  name = "example"

  attributes = {
    First = "examplevalue"
  }
}
```

» Argument Reference

- `name` - (Required) The name of the thing.
- `attributes` - (Optional) Map of attributes of the thing.
- `thing_type_name` - (Optional) The thing type name.

» Attributes Reference

In addition to the arguments above, the following attributes are exported:

- `default_client_id` - The default client ID.
- `version` - The current version of the thing record in the registry.
- `arn` - The ARN of the thing.

» Import

IOT Things can be imported using the name, e.g.

```
$ terraform import aws_iot_thing.example example
```

» Resource: aws_iot_thing_principal_attachment

Attaches Principal to AWS IoT Thing.

» Example Usage

```
resource "aws_iot_thing" "example" {
  name = "example"
}

resource "aws_iot_certificate" "cert" {
  csr    = "${file("csr.pem")}"
  active = true
}

resource "aws_iot_thing_principal_attachment" "att" {
  principal = "${aws_iot_certificate.cert.arn}"
  thing     = "${aws_iot_thing.example.name}"
}
```

» Argument Reference

- `principal` - (Required) The AWS IoT Certificate ARN or Amazon Cognito Identity ID.
- `thing` - (Required) The name of the thing.

» Resource: aws_iot_thing_type

Creates and manages an AWS IoT Thing Type.

» Example Usage

```
resource "aws_iot_thing_type" "foo" {
  name = "my_iot_thing"
}
```

» Argument Reference

- `name` - (Required, Forces New Resource) The name of the thing type.
- `deprecated` - (Optional, Defaults to false) Whether the thing type is deprecated. If true, no new things could be associated with this type.
- `properties` - (Optional), Configuration block that can contain the following properties of the thing type:
  - `description` - (Optional, Forces New Resource) The description of the thing type.
  - `searchable_attributes` - (Optional, Forces New Resource) A list of searchable thing attribute names.

» Attributes Reference

In addition to the arguments above, the following attributes are exported:

- `arn` - The ARN of the created AWS IoT Thing Type.

» Import

IOT Thing Types can be imported using the name, e.g.

```
$ terraform import aws_iot_thing_type.example example
```

» Resource: aws_iot_role_alias

Provides an IoT role alias.

» Example Usage

```
resource "aws_iam_role" "role" {
  name = "dynamodb-access-role"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {"Service": "credentials.iot.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iot_role_alias" "alias" {
  alias    = "Thermostat-dynamodb-access-role-alias"
  role_arn = "${aws_iam_role.role.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `alias` - (Required) The name of the role alias.
- `role_arn` - (Required) The identity of the role to which the alias refers.
- `credential_duration` - (Optional) The duration of the credential, in seconds. If you do not specify a value for this setting, the default maximum of one hour is applied. This setting can have a value from 900 seconds (15 minutes) to 3600 seconds (60 minutes).

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS to this role alias.

» Import

IOT Role Alias can be imported via the alias, e.g.

```
$ terraform import aws_iot_role_alias.example myalias
```

» Data Source: aws_inspector_rules_packages

The AWS Inspector Rules Packages data source allows access to the list of AWS Inspector Rules Packages which can be used by AWS Inspector within the region configured in the provider.

» Example Usage

```
# Declare the data source
data "aws_inspector_rules_packages" "rules" {}

# e.g. Use in aws_inspector_assessment_template
resource "aws_inspector_resource_group" "group" {
  tags = {
    test = "test"
  }
}

resource "aws_inspector_assessment_target" "assessment" {
  name               = "test"
  resource_group_arn = "${aws_inspector_resource_group.group.arn}"
}

resource "aws_inspector_assessment_template" "assessment" {
  name       = "Test"
  target_arn = "${aws_inspector_assessment_target.assessment.arn}"
  duration   = "60"

  rules_package_arns = ["${data.aws_inspector_rules_packages.rules.arns}"]
}
```

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arns` - A list of the AWS Inspector Rules Packages arns available in the AWS region.

» Resource: aws_inspector_assessment_target

Provides a Inspector assessment target

» Example Usage

```
resource "aws_inspector_resource_group" "bar" {
  tags = {
    Name = "foo"
    Env  = "bar"
  }
}

resource "aws_inspector_assessment_target" "foo" {
  name               = "assessment target"
  resource_group_arn = "${aws_inspector_resource_group.bar.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the assessment target.
- `resource_group_arn` (Optional) Inspector Resource Group Amazon Resource Name (ARN) stating tags for instance matching. If not specified, all EC2 instances in the current AWS account and region are included in the assessment target.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The target assessment ARN.

» Import

Inspector Assessment Targets can be imported via their Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_inspector_assessment_target.example arn:aws:inspector:us-east-1:12345
```

» Resource: aws_inspector_assessment_template

Provides a Inspector assessment template

» Example Usage

```
resource "aws_inspector_assessment_template" "foo" {
  name       = "bar template"
  target_arn = "${aws_inspector_assessment_target.foo.arn}"
  duration   = 3600

  rules_package_arns = [
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-H5hpSawc",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-JJOtZiqQ",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-vg5GGHSD",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the assessment template.
- `target_arn` - (Required) The assessment target ARN to attach the template to.
- `duration` - (Required) The duration of the inspector run.
- `rules_package_arns` - (Required) The rules to be used during the run.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The template assessment ARN.

» Resource: aws_inspector_resource_group

Provides a Inspector resource group

» Example Usage

```
resource "aws_inspector_resource_group" "bar" {
  tags = {
    Name = "foo"
    Env  = "bar"
  }
}
```

» Argument Reference

The following arguments are supported:

- `tags` - (Required) The tags on your EC2 Instance.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The resource group ARN.

» Data Source: aws_kinesis_stream

Use this data source to get information about a Kinesis Stream for use in other resources.

For more details, see the Amazon Kinesis Documentation.

» Example Usage

```
data "aws_kinesis_stream" "stream" {
  name = "stream-name"
}
```

» Argument Reference

- `name` - (Required) The name of the Kinesis Stream.

» Attributes Reference

`id` is set to the Amazon Resource Name (ARN) of the Kinesis Stream. In addition, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the Kinesis Stream (same as id).
- `name` - The name of the Kinesis Stream.
- `creation_timestamp` - The approximate UNIX timestamp that the stream was created.
- `status` - The current status of the stream. The stream status is one of CREATING, DELETING, ACTIVE, or UPDATING.
- `retention_period` - Length of time (in hours) data records are accessible after they are added to the stream.
- `open_shards` - The list of shard ids in the OPEN state. See Shard State for more.
- `closed_shards` - The list of shard ids in the CLOSED state. See Shard State for more.
- `shard_level_metrics` - A list of shard-level CloudWatch metrics which are enabled for the stream. See Monitoring with CloudWatch for more.
- `tags` - A mapping of tags to assigned to the stream.

» Resource: aws_kinesis_analytics_application

Provides a Kinesis Analytics Application resource. Kinesis Analytics is a managed service that allows processing and analyzing streaming data using standard SQL.

For more details, see the Amazon Kinesis Analytics Documentation.

» Example Usage

```
resource "aws_kinesis_stream" "test_stream" {
  name        = "terraform-kinesis-test"
  shard_count = 1
}

resource "aws_kinesis_analytics_application" "test_application" {
  name = "kinesis-analytics-application-test"

  inputs {
    name_prefix = "test_prefix"

    kinesis_stream {
      resource_arn = "${aws_kinesis_stream.test_stream.arn}"
      role_arn     = "${aws_iam_role.test.arn}"
    }

    parallelism {
      count = 1
    }

    schema {
      record_columns {
        mapping  = "$.test"
        name     = "test"
        sql_type = "VARCHAR(8)"
      }

      record_encoding = "UTF-8"

      record_format {
        mapping_parameters {
          json {
            record_row_path = "$"
          }
        }
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the Kinesis Analytics Application.
- `code` - (Optional) SQL Code to transform input data, and generate output.
- `description` - (Optional) Description of the application.
- `cloudwatch_logging_options` - (Optional) The CloudWatch log stream options to monitor application errors. See CloudWatch Logging Options below for more details.
- `inputs` - (Optional) Input configuration of the application. See Inputs below for more details.
- `outputs` - (Optional) Output destination configuration of the application. See Outputs below for more details.
- `reference_data_sources` - (Optional) An S3 Reference Data Source for the application. See Reference Data Sources below for more details.
- `tags` - Key-value mapping of tags for the Kinesis Analytics Application.

» CloudWatch Logging Options

Configure a CloudWatch Log Stream to monitor application errors.

The `cloudwatch_logging_options` block supports the following:

- `log_stream_arn` - (Required) The ARN of the CloudWatch Log Stream.
- `role_arn` - (Required) The ARN of the IAM Role used to send application messages.

» Inputs

Configure an Input for the Kinesis Analytics Application. You can only have 1 Input configured.

The `inputs` block supports the following:

- `name_prefix` - (Required) The Name Prefix to use when creating an in-application stream.
- `schema` - (Required) The Schema format of the data in the streaming source. See Source Schema below for more details.
- `kinesis_firehose` - (Optional) The Kinesis Firehose configuration for the streaming source. Conflicts with `kinesis_stream`. See Kinesis Firehose below for more details.
- `kinesis_stream` - (Optional) The Kinesis Stream configuration for the streaming source. Conflicts with `kinesis_firehose`. See Kinesis Stream below for more details.
- `parallelism` - (Optional) The number of Parallel in-application streams to create. See Parallelism below for more details.
- `processing_configuration` - (Optional) The Processing Configuration to transform records as they are received from the stream. See Processing Configuration below for more details.

» Outputs

Configure Output destinations for the Kinesis Analytics Application. You can have a maximum of 3 destinations configured.

The `outputs` block supports the following:

- `name` - (Required) The Name of the in-application stream.
- `schema` - (Required) The Schema format of the data written to the destination. See Destination Schema below for more details.
- `kinesis_firehose` - (Optional) The Kinesis Firehose configuration for the destination stream. Conflicts with `kinesis_stream`. See Kinesis Firehose below for more details.

- `kinesis_stream` - (Optional) The Kinesis Stream configuration for the destination stream. Conflicts with `kinesis_firehose`. See Kinesis Stream below for more details.
- `lambda` - (Optional) The Lambda function destination. See Lambda below for more details.

» Reference Data Sources

Add a Reference Data Source to the Kinesis Analytics Application. You can only have 1 Reference Data Source.

The `reference_data_sources` block supports the following:

- `schema` - (Required) The Schema format of the data in the streaming source. See Source Schema below for more details.
- `table_name` - (Required) The in-application Table Name.
- `s3` - (Optional) The S3 configuration for the reference data source. See S3 Reference below for more details.

» Kinesis Firehose

Configuration for a Kinesis Firehose delivery stream.

The `kinesis_firehose` block supports the following:

- `resource_arn` - (Required) The ARN of the Kinesis Firehose delivery stream.
- `role_arn` - (Required) The ARN of the IAM Role used to access the stream.

» Kinesis Stream

Configuration for a Kinesis Stream.

The `kinesis_stream` block supports the following:

- `resource_arn` - (Required) The ARN of the Kinesis Stream.
- `role_arn` - (Required) The ARN of the IAM Role used to access the stream.

» Destination Schema

The Schema format of the data in the destination.

The `schema` block supports the following:

- `record_format_type` - (Required) The Format Type of the records on the output stream. Can be `CSV` or `JSON`.

» Source Schema

The Schema format of the data in the streaming source.

The `schema` block supports the following:

- `record_columns` - (Required) The Record Column mapping for the streaming source data element. See Record Columns below for more details.
- `record_format` - (Required) The Record Format and mapping information to schematize a record. See Record Format below for more details.
- `record_encoding` - (Optional) The Encoding of the record in the streaming source.

» Parallelism

Configures the number of Parallel in-application streams to create.

The `parallelism` block supports the following:

- `count` - (Required) The Count of streams.

» Processing Configuration

The Processing Configuration to transform records as they are received from the stream.

The `processing_configuration` block supports the following:

- `lambda` - (Required) The Lambda function configuration. See Lambda below for more details.

» Lambda

The Lambda function that pre-processes records in the stream.

The `lambda` block supports the following:

- `resource_arn` - (Required) The ARN of the Lambda function.
- `role_arn` - (Required) The ARN of the IAM Role used to access the Lambda function.

» Record Columns

The Column mapping of each data element in the streaming source to the corresponding column in the in-application stream.

The `record_columns` block supports the following:

- `name` - (Required) Name of the column.

- sql_type - (Required) The SQL Type of the column.
- mapping - (Optional) The Mapping reference to the data element.

» Record Format

The Record Format and relevant mapping information that should be applied to schematize the records on the stream.

The record_format block supports the following:

- record_format_type - (Required) The type of Record Format. Can be CSV or JSON.
- mapping_parameters - (Optional) The Mapping Information for the record format. See Mapping Parameters below for more details.

» Mapping Parameters

Provides Mapping information specific to the record format on the streaming source.

The mapping_parameters block supports the following:

- csv - (Optional) Mapping information when the record format uses delimiters. See CSV Mapping Parameters below for more details.
- json - (Optional) Mapping information when JSON is the record format on the streaming source. See JSON Mapping Parameters below for more details.

» CSV Mapping Parameters

Mapping information when the record format uses delimiters.

The csv block supports the following:

- record_column_delimiter - (Required) The Column Delimiter.
- record_row_delimiter - (Required) The Row Delimiter.

» JSON Mapping Parameters

Mapping information when JSON is the record format on the streaming source.

The json block supports the following:

- record_row_path - (Required) Path to the top-level parent that contains the records.

» S3 Reference

Identifies the S3 bucket and object that contains the reference data.

The `s3` blcok supports the following:

- `bucket_arn` - (Required) The S3 Bucket ARN.
- `file_key` - (Required) The File Key name containing reference data.
- `role_arn` - (Required) The IAM Role ARN to read the data.

» Attributes Reference

The following attributes are exported along with all argument references:

- `id` - The ARN of the Kinesis Analytics Application.
- `arn` - The ARN of the Kinesis Analytics Appliation.
- `create_timestamp` - The Timestamp when the application version was created.
- `last_update_timestamp` - The Timestamp when the application was last updated.
- `status` - The Status of the application.
- `version` - The Version of the application.

» Import

Kinesis Analytics Application can be imported by using ARN, e.g.

```
$ terraform import aws_kinesis_analytics_application.example arn:aws:kinesisanalytics:us-wes
```

» Resource: aws_kinesis_stream

Provides a Kinesis Stream resource. Amazon Kinesis is a managed service that scales elastically for real-time processing of streaming big data.

For more details, see the Amazon Kinesis Documentation.

» Example Usage

```
resource "aws_kinesis_stream" "test_stream" {
  name             = "terraform-kinesis-test"
  shard_count      = 1
  retention_period = 48

  shard_level_metrics = [
    "IncomingBytes",
    "OutgoingBytes",
  ]

  tags = {
    Environment = "test"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A name to identify the stream. This is unique to the AWS account and region the Stream is created in.
- `shard_count` – (Required) The number of shards that the stream will use. Amazon has guidelines for specifying the Stream size that should be referenced when creating a Kinesis stream. See Amazon Kinesis Streams for more.
- `retention_period` - (Optional) Length of time data records are accessible after they are added to the stream. The maximum value of a stream's retention period is 168 hours. Minimum value is 24. Default is 24.
- `shard_level_metrics` - (Optional) A list of shard-level CloudWatch metrics which can be enabled for the stream. See Monitoring with CloudWatch for more. Note that the value ALL should not be used; instead you should provide an explicit list of metrics you wish to enable.
- `enforce_consumer_deletion` - (Optional) A boolean that indicates all registered consumers should be deregistered from the stream so that the stream can be destroyed without error. The default value is `false`.
- `encryption_type` - (Optional) The encryption type to use. The only acceptable values are `NONE` or `KMS`. The default value is `NONE`.
- `kms_key_id` - (Optional) The GUID for the customer-managed KMS key to use for encryption. You can also use a Kinesis-owned master key by specifying the alias `alias/aws/kinesis`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

- `id` - The unique Stream id
- `name` - The unique Stream name
- `shard_count` - The count of Shards for this Stream
- `arn` - The Amazon Resource Name (ARN) specifying the Stream (same as `id`)

» Timeouts

`aws_kinesis_stream` provides the following Timeouts configuration options:

- `create` - (Default `5 minutes`) Used for Creating a Kinesis Stream
- `update` - (Default `120 minutes`) Used for Updating a Kinesis Stream
- `delete` - (Default `120 minutes`) Used for Destroying a Kinesis Stream

» Import

Kinesis Streams can be imported using the `name`, e.g.

```
$ terraform import aws_kinesis_stream.test_stream terraform-kinesis-test
```

» Resource: aws_kinesis_firehose_delivery_stream

Provides a Kinesis Firehose Delivery Stream resource. Amazon Kinesis Firehose is a fully managed, elastic service to easily deliver real-time data streams to destinations such as Amazon S3 and Amazon Redshift.

For more details, see the Amazon Kinesis Firehose Documentation.

» Example Usage

» Extended S3 Destination

```
resource "aws_kinesis_firehose_delivery_stream" "extended_s3_stream" {
  name        = "terraform-kinesis-firehose-extended-s3-test-stream"
  destination = "extended_s3"

  extended_s3_configuration {
    role_arn   = "${aws_iam_role.firehose_role.arn}"
    bucket_arn = "${aws_s3_bucket.bucket.arn}"

    processing_configuration {
      enabled = "true"

      processors {
        type = "Lambda"

        parameters {
          parameter_name  = "LambdaArn"
          parameter_value = "${aws_lambda_function.lambda_processor.arn}:$LATEST"
        }
      }
    }
  }
}

resource "aws_s3_bucket" "bucket" {
  bucket = "tf-test-bucket"
  acl    = "private"
}

resource "aws_iam_role" "firehose_role" {
  name = "firehose_test_role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "firehose.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_iam_role" "lambda_iam" {
  name = "lambda_iam"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
```

```
}
EOF
}

resource "aws_lambda_function" "lambda_processor" {
  filename      = "lambda.zip"
  function_name = "firehose_lambda_processor"
  role          = "${aws_iam_role.lambda_iam.arn}"
  handler       = "exports.handler"
  runtime       = "nodejs8.10"
}
```

» S3 Destination

```
resource "aws_s3_bucket" "bucket" {
  bucket = "tf-test-bucket"
  acl    = "private"
}

resource "aws_iam_role" "firehose_role" {
  name = "firehose_test_role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "firehose.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_kinesis_firehose_delivery_stream" "test_stream" {
  name        = "terraform-kinesis-firehose-test-stream"
  destination = "s3"

  s3_configuration {
    role_arn   = "${aws_iam_role.firehose_role.arn}"
    bucket_arn = "${aws_s3_bucket.bucket.arn}"
  }
}
```

» Redshift Destination

```
resource "aws_redshift_cluster" "test_cluster" {
  cluster_identifier = "tf-redshift-cluster-%d"
  database_name      = "test"
  master_username    = "testuser"
  master_password    = "T3stPass"
  node_type          = "dc1.large"
  cluster_type       = "single-node"
}

resource "aws_kinesis_firehose_delivery_stream" "test_stream" {
  name        = "terraform-kinesis-firehose-test-stream"
  destination = "redshift"

  s3_configuration {
    role_arn           = "${aws_iam_role.firehose_role.arn}"
    bucket_arn         = "${aws_s3_bucket.bucket.arn}"
    buffer_size        = 10
    buffer_interval    = 400
    compression_format = "GZIP"
  }

  redshift_configuration {
    role_arn           = "${aws_iam_role.firehose_role.arn}"
    cluster_jdbcurl    = "jdbc:redshift://${aws_redshift_cluster.test_cluster.endpoint}/${aw
    username           = "testuser"
    password           = "T3stPass"
    data_table_name    = "test-table"
    copy_options       = "delimiter '|'" # the default delimiter
    data_table_columns = "test-col"
    s3_backup_mode     = "Enabled"

    s3_backup_configuration {
      role_arn           = "${aws_iam_role.firehose_role.arn}"
      bucket_arn         = "${aws_s3_bucket.bucket.arn}"
      buffer_size        = 15
      buffer_interval    = 300
      compression_format = "GZIP"
    }
  }
}
```

» Elasticsearch Destination

```
resource "aws_elasticsearch_domain" "test_cluster" {
  domain_name = "firehose-es-test"
}

resource "aws_kinesis_firehose_delivery_stream" "test_stream" {
  name        = "terraform-kinesis-firehose-test-stream"
  destination = "elasticsearch"

  s3_configuration {
    role_arn           = "${aws_iam_role.firehose_role.arn}"
    bucket_arn         = "${aws_s3_bucket.bucket.arn}"
    buffer_size        = 10
    buffer_interval    = 400
    compression_format = "GZIP"
  }

  elasticsearch_configuration {
    domain_arn = "${aws_elasticsearch_domain.test_cluster.arn}"
    role_arn   = "${aws_iam_role.firehose_role.arn}"
    index_name = "test"
    type_name  = "test"

    processing_configuration {
      enabled = "true"

      processors {
        type = "Lambda"

        parameters {
          parameter_name  = "LambdaArn"
          parameter_value = "${aws_lambda_function.lambda_processor.arn}:$LATEST"
        }
      }
    }
  }
}
```

» Splunk Destination

```
resource "aws_kinesis_firehose_delivery_stream" "test_stream" {
  name        = "terraform-kinesis-firehose-test-stream"
  destination = "splunk"

  s3_configuration {
    role_arn           = "${aws_iam_role.firehose.arn}"
    bucket_arn         = "${aws_s3_bucket.bucket.arn}"
    buffer_size        = 10
    buffer_interval    = 400
    compression_format = "GZIP"
  }

  splunk_configuration {
    hec_endpoint               = "https://http-inputs-mydomain.splunkcloud.com:443"
    hec_token                  = "51D4DA16-C61B-4F5F-8EC7-ED4301342A4A"
    hec_acknowledgment_timeout = 600
    hec_endpoint_type          = "Event"
    s3_backup_mode             = "FailedEventsOnly"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A name to identify the stream. This is unique to the AWS account and region the Stream is created in.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `kinesis_source_configuration` - (Optional) Allows the ability to specify the kinesis stream that is used as the source of the firehose delivery stream.
- `server_side_encryption` - (Optional) Encrypt at rest options. Server-side encryption should not be enabled when a kinesis stream is configured as the source of the firehose delivery stream.
- `destination` – (Required) This is the destination to where the data is delivered. The only options are `s3` (Deprecated, use `extended_s3` instead), `extended_s3`, `redshift`, `elasticsearch`, and `splunk`.
- `s3_configuration` - (Optional) Required for non-S3 destinations. For S3 destination, use `extended_s3_configuration` instead. Configuration options for the s3 destination (or the intermediate bucket if the destination is redshift). More details are given below.
- `extended_s3_configuration` - (Optional, only Required when `destination` is `extended_s3`) Enhanced configuration options for the s3 destination. More details are given below.
- `redshift_configuration` - (Optional) Configuration options if redshift is the destination. Using `redshift_configuration` requires the user to

also specify a s3_configuration block. More details are given below.
- elasticsearch_configuration - (Optional) Configuration options if elasticsearch is the destination. More details are given below.

The kinesis_source_configuration object supports the following:

- kinesis_stream_arn (Required) The kinesis stream used as the source of the firehose delivery stream.
- role_arn (Required) The ARN of the role that provides access to the source Kinesis stream.

The server_side_encryption object supports the following:

- enabled - (Optional) Whether to enable encryption at rest. Default is false.

The s3_configuration object supports the following:

- role_arn - (Required) The ARN of the AWS credentials.
- bucket_arn - (Required) The ARN of the S3 bucket
- prefix - (Optional) The ”YYYY/MM/DD/HH” time format prefix is automatically used for delivered S3 files. You can specify an extra prefix to be added in front of the time format prefix. Note that if the prefix ends with a slash, it appears as a folder in the S3 bucket
- buffer_size - (Optional) Buffer incoming data to the specified size, in MBs, before delivering it to the destination. The default value is 5. We recommend setting SizeInMBs to a value greater than the amount of data you typically ingest into the delivery stream in 10 seconds. For example, if you typically ingest data at 1 MB/sec set SizeInMBs to be 10 MB or higher.
- buffer_interval - (Optional) Buffer incoming data for the specified period of time, in seconds, before delivering it to the destination. The default value is 300.
- compression_format - (Optional) The compression format. If no value is specified, the default is UNCOMPRESSED. Other supported values are GZIP, ZIP & Snappy. If the destination is redshift you cannot use ZIP or Snappy.
- kms_key_arn - (Optional) Specifies the KMS key ARN the stream will use to encrypt data. If not set, no encryption will be used.
- cloudwatch_logging_options - (Optional) The CloudWatch Logging Options for the delivery stream. More details are given below

The extended_s3_configuration object supports the same fields from s3_configuration as well as the following:

- data_format_conversion_configuration - (Optional) Nested argument for the serializer, deserializer, and schema for converting data from the JSON format to the Parquet or ORC format before writing it to Amazon S3. More details given below.

- `error_output_prefix` - (Optional) Prefix added to failed records before writing them to S3. This prefix appears immediately following the bucket name.
- `processing_configuration` - (Optional) The data processing configuration. More details are given below.
- `s3_backup_mode` - (Optional) The Amazon S3 backup mode. Valid values are `Disabled` and `Enabled`. Default value is `Disabled`.
- `s3_backup_configuration` - (Optional) The configuration for backup in Amazon S3. Required if `s3_backup_mode` is `Enabled`. Supports the same fields as `s3_configuration` object.

The `redshift_configuration` object supports the following:

- `cluster_jdbcurl` - (Required) The jdbcurl of the redshift cluster.
- `username` - (Required) The username that the firehose delivery stream will assume. It is strongly recommended that the username and password provided is used exclusively for Amazon Kinesis Firehose purposes, and that the permissions for the account are restricted for Amazon Redshift INSERT permissions.
- `password` - (Required) The password for the username above.
- `retry_duration` - (Optional) The length of time during which Firehose retries delivery after a failure, starting from the initial request and including the first attempt. The default value is 3600 seconds (60 minutes). Firehose does not retry if the value of DurationInSeconds is 0 (zero) or if the first delivery attempt takes longer than the current value.
- `role_arn` - (Required) The arn of the role the stream assumes.
- `s3_backup_mode` - (Optional) The Amazon S3 backup mode. Valid values are `Disabled` and `Enabled`. Default value is `Disabled`.
- `s3_backup_configuration` - (Optional) The configuration for backup in Amazon S3. Required if `s3_backup_mode` is `Enabled`. Supports the same fields as `s3_configuration` object.
- `data_table_name` - (Required) The name of the table in the redshift cluster that the s3 bucket will copy to.
- `copy_options` - (Optional) Copy options for copying the data from the s3 intermediate bucket into redshift, for example to change the default delimiter. For valid values, see the AWS documentation
- `data_table_columns` - (Optional) The data table columns that will be targeted by the copy command.
- `cloudwatch_logging_options` - (Optional) The CloudWatch Logging Options for the delivery stream. More details are given below
- `processing_configuration` - (Optional) The data processing configuration. More details are given below.

The `elasticsearch_configuration` object supports the following:

- `buffering_interval` - (Optional) Buffer incoming data for the specified period of time, in seconds between 60 to 900, before delivering it to the destination. The default value is 300s.

- buffering_size - (Optional) Buffer incoming data to the specified size, in MBs between 1 to 100, before delivering it to the destination. The default value is 5MB.
- domain_arn - (Required) The ARN of the Amazon ES domain. The IAM role must have permission for DescribeElasticsearchDomain, DescribeElasticsearchDomains, and DescribeElasticsearchDomainConfig after assuming RoleARN. The pattern needs to be arn:.*.
- index_name - (Required) The Elasticsearch index name.
- index_rotation_period - (Optional) The Elasticsearch index rotation period. Index rotation appends a timestamp to the IndexName to facilitate expiration of old data. Valid values are NoRotation, OneHour, OneDay, OneWeek, and OneMonth. The default value is OneDay.
- retry_duration - (Optional) After an initial failure to deliver to Amazon Elasticsearch, the total amount of time, in seconds between 0 to 7200, during which Firehose re-attempts delivery (including the first attempt). After this time has elapsed, the failed documents are written to Amazon S3. The default value is 300s. There will be no retry if the value is 0.
- role_arn - (Required) The ARN of the IAM role to be assumed by Firehose for calling the Amazon ES Configuration API and for indexing documents. The pattern needs to be arn:.*.
- s3_backup_mode - (Optional) Defines how documents should be delivered to Amazon S3. Valid values are FailedDocumentsOnly and AllDocuments. Default value is FailedDocumentsOnly.
- type_name - (Required) The Elasticsearch type name with maximum length of 100 characters.
- cloudwatch_logging_options - (Optional) The CloudWatch Logging Options for the delivery stream. More details are given below
- processing_configuration - (Optional) The data processing configuration. More details are given below.

The splunk_configuration objects supports the following:

- hec_acknowledgment_timeout - (Optional) The amount of time, in seconds between 180 and 600, that Kinesis Firehose waits to receive an acknowledgment from Splunk after it sends it data.
- hec_endpoint - (Required) The HTTP Event Collector (HEC) endpoint to which Kinesis Firehose sends your data.
- hec_endpoint_type - (Optional) The HEC endpoint type. Valid values are Raw or Event. The default value is Raw.
- hec_token - The GUID that you obtain from your Splunk cluster when you create a new HEC endpoint.
- s3_backup_mode - (Optional) Defines how documents should be delivered to Amazon S3. Valid values are FailedEventsOnly and AllEvents. Default value is FailedEventsOnly.
- retry_duration - (Optional) After an initial failure to deliver to Amazon Elasticsearch, the total amount of time, in seconds between 0 to 7200, during which Firehose re-attempts delivery (including the first attempt).

After this time has elapsed, the failed documents are written to Amazon S3. The default value is 300s. There will be no retry if the value is 0.
- cloudwatch_logging_options - (Optional) The CloudWatch Logging Options for the delivery stream. More details are given below.
- processing_configuration - (Optional) The data processing configuration. More details are given below.

The cloudwatch_logging_options object supports the following:

- enabled - (Optional) Enables or disables the logging. Defaults to false.
- log_group_name - (Optional) The CloudWatch group name for logging. This value is required if enabled is true.
- log_stream_name - (Optional) The CloudWatch log stream name for logging. This value is required if enabled is true.

The processing_configuration object supports the following:

- enabled - (Optional) Enables or disables data processing.
- processors - (Optional) Array of data processors. More details are given below

The processors array objects support the following:

- type - (Required) The type of processor. Valid Values: Lambda
- parameters - (Optional) Array of processor parameters. More details are given below

The parameters array objects support the following:

- parameter_name - (Required) Parameter name. Valid Values: LambdaArn, NumberOfRetries, RoleArn, BufferSizeInMBs, BufferIntervalInSeconds
- parameter_value - (Required) Parameter value. Must be between 1 and 512 length (inclusive). When providing a Lambda ARN, you should specify the resource version as well.

» data_format_conversion_configuration

NOTE: Once configured, the data format conversion configuration can only be disabled, in which the configuration values will remain, but will not be active. It is not currently possible to completely remove the configuration without recreating the resource.

Example:

```
resource "aws_kinesis_firehose_delivery_stream" "example" {
  # ... other configuration ...
  extended_s3_configuration {
    # Must be at least 64
    buffer_size = 128
    # ... other configuration ...
    data_format_conversion_configuration {
      input_format_configuration {
        deserializer {
          hive_json_ser_de {}
        }
      }

      output_format_configuration {
        serializer {
          orc_ser_de {}
        }
      }

      schema_configuration {
        database_name = "${aws_glue_catalog_table.example.database_name}"
        role_arn      = "${aws_iam_role.example.arn}"
        table_name    = "${aws_glue_catalog_table.example.name}"
      }
    }
  }
}
```

- `input_format_configuration` - (Required) Nested argument that specifies the deserializer that you want Kinesis Data Firehose to use to convert the format of your data from JSON. More details below.
- `output_format_configuration` - (Required) Nested argument that specifies the serializer that you want Kinesis Data Firehose to use to convert the format of your data to the Parquet or ORC format. More details below.
- `schema_configuration` - (Required) Nested argument that specifies the AWS Glue Data Catalog table that contains the column information. More details below.
- `enabled` - (Optional) Defaults to `true`. Set it to `false` if you want to disable format conversion while preserving the configuration details.

» input_format_configuration

- `deserializer` - (Required) Nested argument that specifies which deserializer to use. You can choose either the Apache Hive JSON SerDe or the OpenX JSON SerDe. More details below.

» deserializer

NOTE: One of the deserializers must be configured. If no nested configuration needs to occur simply declare as `XXX_json_ser_de = []` or `XXX_json_ser_de`

`{}`.

- `hive_json_ser_de` - (Optional) Nested argument that specifies the native Hive / HCatalog JsonSerDe. More details below.
- `open_x_json_ser_de` - (Optional) Nested argument that specifies the OpenX SerDe. More details below.

» hive_json_ser_de

- `timestamp_formats` - (Optional) A list of how you want Kinesis Data Firehose to parse the date and time stamps that may be present in your input data JSON. To specify these format strings, follow the pattern syntax of JodaTime's DateTimeFormat format strings. For more information, see Class DateTimeFormat. You can also use the special value millis to parse time stamps in epoch milliseconds. If you don't specify a format, Kinesis Data Firehose uses java.sql.Timestamp::valueOf by default.

» open_x_json_ser_de

- `case_insensitive` - (Optional) When set to true, which is the default, Kinesis Data Firehose converts JSON keys to lowercase before deserializing them.
- `column_to_json_key_mappings` - (Optional) A map of column names to JSON keys that aren't identical to the column names. This is useful when the JSON contains keys that are Hive keywords. For example, timestamp is a Hive keyword. If you have a JSON key named timestamp, set this parameter to `{ ts = "timestamp" }` to map this key to a column named ts.
- `convert_dots_in_json_keys_to_underscores` - (Optional) When set to `true`, specifies that the names of the keys include dots and that you want Kinesis Data Firehose to replace them with underscores. This is useful because Apache Hive does not allow dots in column names. For example, if the JSON contains a key whose name is "a.b", you can define the column name to be "a_b" when using this option. Defaults to `false`.

» output_format_configuration

- `serializer` - (Required) Nested argument that specifies which serializer to use. You can choose either the ORC SerDe or the Parquet SerDe. More details below.

» serializer

NOTE: One of the serializers must be configured. If no nested configuration needs to occur simply declare as `XXX_ser_de = []` or `XXX_ser_de {}`.

- `orc_ser_de` - (Optional) Nested argument that specifies converting data to the ORC format before storing it in Amazon S3. For more information,

see Apache ORC. More details below.
- `parquet_ser_de` - (Optional) Nested argument that specifies converting data to the Parquet format before storing it in Amazon S3. For more information, see Apache Parquet. More details below.

» orc_ser_de

- `block_size_bytes` - (Optional) The Hadoop Distributed File System (HDFS) block size. This is useful if you intend to copy the data from Amazon S3 to HDFS before querying. The default is 256 MiB and the minimum is 64 MiB. Kinesis Data Firehose uses this value for padding calculations.
- `bloom_filter_columns` - (Optional) A list of column names for which you want Kinesis Data Firehose to create bloom filters.
- `bloom_filter_false_positive_probability` - (Optional) The Bloom filter false positive probability (FPP). The lower the FPP, the bigger the Bloom filter. The default value is `0.05`, the minimum is `0`, and the maximum is `1`.
- `compression` - (Optional) The compression code to use over data blocks. The default is `SNAPPY`.
- `dictionary_key_threshold` - (Optional) A float that represents the fraction of the total number of non-null rows. To turn off dictionary encoding, set this fraction to a number that is less than the number of distinct keys in a dictionary. To always use dictionary encoding, set this threshold to `1`.
- `enable_padding` - (Optional) Set this to `true` to indicate that you want stripes to be padded to the HDFS block boundaries. This is useful if you intend to copy the data from Amazon S3 to HDFS before querying. The default is `false`.
- `format_version` - (Optional) The version of the file to write. The possible values are `V0_11` and `V0_12`. The default is `V0_12`.
- `padding_tolerance` - (Optional) A float between 0 and 1 that defines the tolerance for block padding as a decimal fraction of stripe size. The default value is `0.05`, which means 5 percent of stripe size. For the default values of 64 MiB ORC stripes and 256 MiB HDFS blocks, the default block padding tolerance of 5 percent reserves a maximum of 3.2 MiB for padding within the 256 MiB block. In such a case, if the available size within the block is more than 3.2 MiB, a new, smaller stripe is inserted to fit within that space. This ensures that no stripe crosses block boundaries and causes remote reads within a node-local task. Kinesis Data Firehose ignores this parameter when `enable_padding` is `false`.
- `row_index_stride` - (Optional) The number of rows between index entries. The default is `10000` and the minimum is `1000`.
- `stripe_size_bytes` - (Optional) The number of bytes in each stripe. The default is 64 MiB and the minimum is 8 MiB.

» parquet_ser_de

- `block_size_bytes` - (Optional) The Hadoop Distributed File System (HDFS) block size. This is useful if you intend to copy the data from Amazon S3 to HDFS before querying. The default is 256 MiB and the minimum is 64 MiB. Kinesis Data Firehose uses this value for padding calculations.
- `compression` - (Optional) The compression code to use over data blocks. The possible values are `UNCOMPRESSED`, `SNAPPY`, and `GZIP`, with the default being `SNAPPY`. Use `SNAPPY` for higher decompression speed. Use `GZIP` if the compression ratio is more important than speed.
- `enable_dictionary_compression` - (Optional) Indicates whether to enable dictionary compression.
- `max_padding_bytes` - (Optional) The maximum amount of padding to apply. This is useful if you intend to copy the data from Amazon S3 to HDFS before querying. The default is `0`.
- `page_size_bytes` - (Optional) The Parquet page size. Column chunks are divided into pages. A page is conceptually an indivisible unit (in terms of compression and encoding). The minimum value is 64 KiB and the default is 1 MiB.
- `writer_version` - (Optional) Indicates the version of row format to output. The possible values are `V1` and `V2`. The default is `V1`.

» schema_configuration

- `database_name` - (Required) Specifies the name of the AWS Glue database that contains the schema for the output data.
- `role_arn` - (Required) The role that Kinesis Data Firehose can use to access AWS Glue. This role must be in the same account you use for Kinesis Data Firehose. Cross-account roles aren't allowed.
- `table_name` - (Required) Specifies the AWS Glue table that contains the column information that constitutes your data schema.
- `catalog_id` - (Optional) The ID of the AWS Glue Data Catalog. If you don't supply this, the AWS account ID is used by default.
- `region` - (Optional) If you don't specify an AWS Region, the default is the current region.
- `version_id` - (Optional) Specifies the table version for the output data schema. Defaults to `LATEST`.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) specifying the Stream

» Import

Kinesis Firehose Delivery streams can be imported using the stream ARN, e.g.

```
$ terraform import aws_kinesis_firehose_delivery_stream.foo arn:aws:firehose:us-east-1:XXX:d
```

Note: Import does not work for stream destination `s3`. Consider using `extended_s3` since `s3` destination is deprecated.

» Data Source: aws_kms_alias

Use this data source to get the ARN of a KMS key alias. By using this data source, you can reference key alias without having to hard code the ARN as input.

» Example Usage

```
data "aws_kms_alias" "s3" {
  name = "alias/aws/s3"
}
```

» Argument Reference

- `name` - (Required) The display name of the alias. The name must start with the word "alias" followed by a forward slash (alias/)

» Attributes Reference

- `arn` - The Amazon Resource Name(ARN) of the key alias.
- `target_key_id` - Key identifier pointed to by the alias.
- `target_key_arn` - ARN pointed to by the alias.

» Data Source: aws_kms_ciphertext

The KMS ciphertext data source allows you to encrypt plaintext into ciphertext by using an AWS KMS customer master key. The value returned by this data source changes every apply. For a stable ciphertext value, see the `aws_kms_ciphertext` resource.

Note: All arguments including the plaintext be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_kms_key" "oauth_config" {
  description = "oauth config"
  is_enabled  = true
}

data "aws_kms_ciphertext" "oauth" {
  key_id = "${aws_kms_key.oauth_config.key_id}"

  plaintext = <<EOF
{
  "client_id": "e587dbae22222f55da22",
  "client_secret": "8289575d00000ace55e1815ec13673955721b8a5"
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `plaintext` - (Required) Data to be encrypted. Note that this may show up in logs, and it will be stored in the state file.
- `key_id` - (Required) Globally unique key ID for the customer master key.
- `context` - (Optional) An optional mapping that makes up the encryption context.

» Attributes Reference

All of the argument attributes are also exported as result attributes.

- `ciphertext_blob` - Base64 encoded ciphertext

» aws_kms_key

Use this data source to get detailed information about the specified KMS Key with flexible key id input. This can be useful to reference key alias without having to hard code the ARN as input.

» Example Usage

```
data "aws_kms_key" "foo" {
  key_id = "alias/my-key"
}

data "aws_kms_key" "foo" {
  key_id = "1234abcd-12ab-34cd-56ef-1234567890ab"
}

data "aws_kms_key" "foo" {
  key_id = "arn:aws:kms:us-east-1:111122223333:alias/my-key"
}

data "aws_kms_key" "foo" {
  key_id = "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

» Argument Reference

- `key_id` - (Required) Key identifier which can be one of the following format:
  - Key ID. E.g: `1234abcd-12ab-34cd-56ef-1234567890ab`
  - Key ARN. E.g.: `arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456`
  - Alias name. E.g.: `alias/my-key`
  - Alias ARN: E.g.: `arn:aws:kms:us-east-1:111122223333:alias/my-key`
- `grant_tokens` - (Optional) List of grant tokens

» Attributes Reference

- `id`: The globally unique identifier for the key
- `arn`: The Amazon Resource Name (ARN) of the key
- `aws_account_id`: The twelve-digit account ID of the AWS account that owns the key
- `creation_date`: The date and time when the key was created
- `deletion_date`: The date and time after which AWS KMS deletes the key. This value is present only when `key_state` is `PendingDeletion`, otherwise this value is 0
- `description`: The description of the key.
- `enabled`: Specifies whether the key is enabled. When `key_state` is `Enabled` this value is true, otherwise it is false
- `expiration_model`: Specifies whether the Key's key material expires. This value is present only when `origin` is `EXTERNAL`, otherwise this value is empty
- `key_manager`: The key's manager
- `key_state`: The state of the key
- `key_usage`: Currently the only allowed value is `ENCRYPT_DECRYPT`

- `origin`: When this value is `AWS_KMS`, AWS KMS created the key material. When this value is `EXTERNAL`, the key material was imported from your existing key management infrastructure or the CMK lacks key material
- `valid_to`: The time at which the imported key material expires. This value is present only when `origin` is `EXTERNAL` and whose `expiration_model` is `KEY_MATERIAL_EXPIRES`, otherwise this value is 0

» Data Source: aws_kms_secrets

Decrypt multiple secrets from data encrypted with the AWS KMS service.

NOTE: Using this data provider will allow you to conceal secret data within your resource definitions but does not take care of protecting that data in all Terraform logging and state output. Please take care to secure your secret data beyond just the Terraform configuration.

» Example Usage

If you do not already have a `CiphertextBlob` from encrypting a KMS secret, you can use the below commands to obtain one using the AWS CLI kms encrypt command. This requires you to have your AWS CLI setup correctly and replace the `--key-id` with your own. Alternatively you can use `--plaintext 'password'` instead of reading from a file.

If you have a newline character at the end of your file, it will be decrypted with this newline character intact. For most use cases this is undesirable and leads to incorrect passwords or invalid values, as well as possible changes in the plan. Be sure to use `echo -n` if necessary.

```
$ echo -n 'master-password' > plaintext-password
$ aws kms encrypt --key-id ab123456-c012-4567-890a-deadbeef123 --plaintext fileb://plaintext
AQECAHgaPa0J8WadplGCqqVAr4HNvDaFSQ+NaiwIBhmm6qDSFwAAAGIwYAYJKoZIhvcNAQcGoFMwUQIBADBMBgkqhki
```

That encrypted output can now be inserted into Terraform configurations without exposing the plaintext secret directly.

```
data "aws_kms_secrets" "example" {
  secret {
    # ... potentially other configuration ...
    name    = "master_password"
    payload = "AQECAHgaPa0J8WadplGCqqVAr4HNvDaFSQ+NaiwIBhmm6qDSFwAAAGIwYAYJKoZIhvcNAQcGoFMw

    context = {
      foo = "bar"
    }
  }
  secret {
    # ... potentially other configuration ...
    name    = "master_username"
    payload = "AQECAHgaPa0J8WadplGCqqVAr4HNvDaFSQ+NaiwIBhmm6qDSFwAAAGIwYAYJKoZIhvcNAQcGoFMwU
  }
}

resource "aws_rds_cluster" "example" {
  # ... other configuration ...
  master_password = "${data.aws_kms_secrets.example.plaintext["master_password"]}"
  master_username = "${data.aws_kms_secrets.example.plaintext["master_username"]}"
}
```

» Argument Reference

The following arguments are supported:

- `secret` - (Required) One or more encrypted payload definitions from the KMS service. See the Secret Definitions below.

» Secret Definitions

Each `secret` supports the following arguments:

- `name` - (Required) The name to export this secret under in the attributes.
- `payload` - (Required) Base64 encoded payload, as returned from a KMS encrypt operation.
- `context` - (Optional) An optional mapping that makes up the Encryption Context for the secret.
- `grant_tokens` (Optional) An optional list of Grant Tokens for the secret.

For more information on `context` and `grant_tokens` see the KMS Concepts

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `plaintext` - Map containing each `secret name` as the key with its decrypted plaintext value

» Resource: aws_kms_alias

Provides an alias for a KMS customer master key. AWS Console enforces 1-to-1 mapping between aliases & keys, but API (hence Terraform too) allows you to create as many aliases as the account limits allow you.

» Example Usage

```
resource "aws_kms_key" "a" {}

resource "aws_kms_alias" "a" {
  name          = "alias/my-key-alias"
  target_key_id = "${aws_kms_key.a.key_id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The display name of the alias. The name must start with the word "alias" followed by a forward slash (alias/)
- `name_prefix` - (Optional) Creates an unique alias beginning with the specified prefix. The name must start with the word "alias" followed by a forward slash (alias/). Conflicts with `name`.
- `target_key_id` - (Required) Identifier for the key for which the alias is for, can be either an ARN or key_id.

» Attributes Reference

In addition to the arguments, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the key alias.
- `target_key_arn` - The Amazon Resource Name (ARN) of the target key identifier.

» Import

KMS aliases can be imported using the `name`, e.g.

```
$ terraform import aws_kms_alias.a alias/my-key-alias
```

» Resource: aws_kms_external_key

Manages a KMS Customer Master Key that uses external key material. To instead manage a KMS Customer Master Key where AWS automatically generates and potentially rotates key material, see the `aws_kms_key` resource.

Note: All arguments including the key material will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_kms_external_key" "example" {
  description = "KMS EXTERNAL for AMI encryption"
}
```

» Argument Reference

The following arguments are supported:

- `deletion_window_in_days` - (Optional) Duration in days after which the key is deleted after destruction of the resource. Must be between `7` and `30` days. Defaults to `30`.
- `description` - (Optional) Description of the key.
- `enabled` - (Optional) Specifies whether the key is enabled. Keys pending import can only be `false`. Imported keys default to `true` unless expired.
- `key_material_base64` - (Optional) Base64 encoded 256-bit symmetric encryption key material to import. The CMK is permanently associated with this key material. The same key material can be reimported, but you cannot import different key material.
- `policy` - (Optional) A key policy JSON document. If you do not provide a key policy, AWS KMS attaches a default key policy to the CMK.
- `tags` - (Optional) A key-value map of tags to assign to the key.
- `valid_to` - (Optional) Time at which the imported key material expires. When the key material expires, AWS KMS deletes the key material and the CMK becomes unusable. If not specified, key material does not expire. Valid values: RFC3339 time string (`YYYY-MM-DDTHH:MM:SSZ`)

» Attributes Reference

The following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the key.

- `expiration_model` - Whether the key material expires. Empty when pending key material import, otherwise `KEY_MATERIAL_EXPIRES` or `KEY_MATERIAL_DOES_NOT_EXPIRE`.
- `id` - The unique identifier for the key.
- `key_state` - The state of the CMK.
- `key_usage` - The cryptographic operations for which you can use the CMK.

» Import

KMS External Keys can be imported using the `id`, e.g.

```
$ terraform import aws_kms_external_key.a arn:aws:kms:us-west-2:111122223333:key/1234abcd-12
```

» Resource: aws_kms_grant

Provides a resource-based access control mechanism for a KMS customer master key.

» Example Usage

```
resource "aws_kms_key" "a" {}

resource "aws_iam_role" "a" {
  name = "iam-role-for-grant"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}
```

```
resource "aws_kms_grant" "a" {
  name              = "my-grant"
  key_id            = "${aws_kms_key.a.key_id}"
  grantee_principal = "${aws_iam_role.a.arn}"
  operations        = ["Encrypt", "Decrypt", "GenerateDataKey"]

  constraints {
    encryption_context_equals = {
      Department = "Finance"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resources) A friendly name for identifying the grant.
- `key_id` - (Required, Forces new resources) The unique identifier for the customer master key (CMK) that the grant applies to. Specify the key ID or the Amazon Resource Name (ARN) of the CMK. To specify a CMK in a different AWS account, you must use the key ARN.
- `grantee_principal` - (Required, Forces new resources) The principal that is given permission to perform the operations that the grant permits in ARN format. Note that due to eventual consistency issues around IAM principals, terraform's state may not always be refreshed to reflect what is true in AWS.
- `operations` - (Required, Forces new resources) A list of operations that the grant permits. The permitted values are: `Decrypt, Encrypt, GenerateDataKey, GenerateDataKeyWithoutPlaintext, ReEncryptFrom, ReEncryptTo, CreateGrant, RetireGrant, DescribeKey`
- `retiring_principal` - (Optional, Forces new resources) The principal that is given permission to retire the grant by using RetireGrant operation in ARN format. Note that due to eventual consistency issues around IAM principals, terraform's state may not always be refreshed to reflect what is true in AWS.
- `constraints` - (Optional, Forces new resources) A structure that you can use to allow certain operations in the grant only when the desired encryption context is present. For more information about encryption context, see Encryption Context.
- `grant_creation_tokens` - (Optional, Forces new resources) A list of grant tokens to be used when creating the grant. See Grant Tokens for more information about grant tokens.
- `retire_on_delete` -(Defaults to false, Forces new resources) If set to false

(the default) the grants will be revoked upon deletion, and if set to true the grants will try to be retired upon deletion. Note that retiring grants requires special permissions, hence why we default to revoking grants. See RetireGrant for more information.

The `constraints` block supports the following arguments:

- `encryption_context_equals` - (Optional) A list of key-value pairs that must match the encryption context in subsequent cryptographic operation requests. The grant allows the operation only when the encryption context in the request is the same as the encryption context specified in this constraint. Conflicts with `encryption_context_subset`.
- `encryption_context_subset` - (Optional) A list of key-value pairs that must be included in the encryption context of subsequent cryptographic operation requests. The grant allows the cryptographic operation only when the encryption context in the request includes the key-value pairs specified in this constraint, although it can include additional key-value pairs. Conflicts with `encryption_context_equals`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `grant_id` - The unique identifier for the grant.
- `grant_token` - The grant token for the created grant. For more information, see Grant Tokens.

» Resource: aws_kms_ciphertext

The KMS ciphertext resource allows you to encrypt plaintext into ciphertext by using an AWS KMS customer master key. The value returned by this resource is stable across every apply. For a changing ciphertext value each apply, see the `aws_kms_ciphertext` data source.

Note: All arguments including the plaintext be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_kms_key" "oauth_config" {
  description = "oauth config"
  is_enabled  = true
}
```

```
resource "aws_kms_ciphertext" "oauth" {
  key_id = "${aws_kms_key.oauth_config.key_id}"

  plaintext = <<EOF
{
  "client_id": "e587dbae22222f55da22",
  "client_secret": "8289575d00000ace55e1815ec13673955721b8a5"
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `plaintext` - (Required) Data to be encrypted. Note that this may show up in logs, and it will be stored in the state file.
- `key_id` - (Required) Globally unique key ID for the customer master key.
- `context` - (Optional) An optional mapping that makes up the encryption context.

» Attributes Reference

All of the argument attributes are also exported as result attributes.

- `ciphertext_blob` - Base64 encoded ciphertext

» Resource: aws_kms_key

Provides a KMS customer master key.

» Example Usage

```
resource "aws_kms_key" "a" {
  description             = "KMS key 1"
  deletion_window_in_days = 10
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Optional) The description of the key as viewed in AWS console.
- `key_usage` - (Optional) Specifies the intended use of the key. Defaults to ENCRYPT_DECRYPT, and only symmetric encryption and decryption are supported.
- `policy` - (Optional) A valid policy JSON document. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.
- `deletion_window_in_days` - (Optional) Duration in days after which the key is deleted after destruction of the resource, must be between 7 and 30 days. Defaults to 30 days.
- `is_enabled` - (Optional) Specifies whether the key is enabled. Defaults to true.
- `enable_key_rotation` - (Optional) Specifies whether key rotation is enabled. Defaults to false.
- `tags` - (Optional) A mapping of tags to assign to the object.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the key.
- `key_id` - The globally unique identifier for the key.

» Import

KMS Keys can be imported using the `id`, e.g.

```
$ terraform import aws_kms_key.a 1234abcd-12ab-34cd-56ef-1234567890ab
```

» aws_lambda_function

Provides information about a Lambda Function.

» Example Usage

```
variable "function_name" {
  type = "string"
}

data "aws_lambda_function" "existing" {
  function_name = "${var.function_name}"
}
```

» Argument Reference

The following arguments are supported:

- `function_name` - (Required) Name of the lambda function.
- `qualifier` - (Optional) Alias name or version number of the lambda function. e.g. `$LATEST`, `my-alias`, or `1`

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Unqualified (no `:QUALIFIER` or `:VERSION` suffix) Amazon Resource Name (ARN) identifying your Lambda Function. See also `qualified_arn`.
- `dead_letter_config` - Configure the function's *dead letter queue*.
- `description` - Description of what your Lambda Function does.
- `environment` - The Lambda environment's configuration settings.
- `handler` - The function entrypoint in your code.
- `invoke_arn` - The ARN to be used for invoking Lambda Function from API Gateway.
- `kms_key_arn` - The ARN for the KMS encryption key.
- `last_modified` - The date this resource was last modified.
- `layers` - A list of Lambda Layer ARNs attached to your Lambda Function.
- `memory_size` - Amount of memory in MB your Lambda Function can use at runtime.
- `qualified_arn` - Qualified (`:QUALIFIER` or `:VERSION` suffix) Amazon Resource Name (ARN) identifying your Lambda Function. See also `arn`.
- `reserved_concurrent_executions` - The amount of reserved concurrent executions for this lambda function or `-1` if unreserved.
- `role` - IAM role attached to the Lambda Function.
- `runtime` - The runtime environment for the Lambda function..
- `source_code_hash` - Base64-encoded representation of raw SHA-256 sum of the zip file.
- `source_code_size` - The size in bytes of the function .zip file.
- `timeout` - The function execution time at which Lambda should terminate the function.
- `tracing_config` - Tracing settings of the function.
- `version` - The version of the Lambda function.
- `vpc_config` - VPC configuration associated with your Lambda function.

» Data Source: aws_lambda_invocation

Use this data source to invoke custom lambda functions as data source. The lambda function is invoked with RequestResponse invocation type.

» Example Usage

```
data "aws_lambda_invocation" "example" {
  function_name = "${aws_lambda_function.lambda_function_test.function_name}"

  input = <<JSON
{
  "key1": "value1",
  "key2": "value2"
}
JSON
}

output "result" {
  description = "String result of Lambda execution"
  value       = "${data.aws_lambda_invocation.example.result}"
}

# In Terraform 0.11 and earlier, the result_map attribute can be used
# to convert a result JSON string to a map of string keys to string values.
output "result_entry_tf011" {
  value = "${data.aws_lambda_invocation.example.result_map["key1"]}"
}

# In Terraform 0.12 and later, the jsondecode() function can be used
# to convert a result JSON string to native Terraform types.
output "result_entry_tf012" {
  value = jsondecode(data.aws_lambda_invocation.example.result)["key1"]
}
```

» Argument Reference

- `function_name` - (Required) The name of the lambda function.
- `input` - (Required) A string in JSON format that is passed as payload to the lambda function.
- `qualifier` - (Optional) The qualifier (a.k.a version) of the lambda function. Defaults to `$LATEST`.

» Attributes Reference

- `result` - String result of the lambda function invocation.
- `result_map` - This field is set only if result is a map of primitive types, where the map is string keys and string values. In Terraform 0.12 and later, use the `jsondecode()` function with the `result` attribute instead to convert the result to all supported native Terraform types.

» aws_lambda_layer_version

Provides information about a Lambda Layer Version.

» Example Usage

```
variable "layer_name" {
  type = "string"
}

data "aws_lambda_layer_version" "existing" {
  layer_name = "${var.layer_name}"
}
```

» Argument Reference

The following arguments are supported:

- `layer_name` - (Required) Name of the lambda layer.
- `version` - (Optional) Specific layer version. Conflicts with `compatible_runtime`. If omitted, the latest available layer version will be used.
- `compatible_runtime` (Optional) Specific runtime the layer version must support. Conflicts with `version`. If specified, the latest available layer version supporting the provided runtime will be used.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `description` - Description of the specific Lambda Layer version.
- `license_info` - License info associated with the specific Lambda Layer version.
- `compatible_runtimes` - A list of Runtimes the specific Lambda Layer version is compatible with.

- `arn` - The Amazon Resource Name (ARN) of the Lambda Layer with version.
- `layer_arn` - The Amazon Resource Name (ARN) of the Lambda Layer without version.
- `created_date` - The date this resource was created.
- `source_code_hash` - Base64-encoded representation of raw SHA-256 sum of the zip file.
- `source_code_size` - The size in bytes of the function .zip file.
- `version` - This Lamba Layer version.

» Resource: aws_lambda_alias

Creates a Lambda function alias. Creates an alias that points to the specified Lambda function version.

For information about Lambda and how to use it, see What is AWS Lambda? For information about function aliases, see CreateAlias and AliasRoutingConfiguration in the API docs.

» Example Usage

```
resource "aws_lambda_alias" "test_alias" {
  name             = "testalias"
  description      = "a sample description"
  function_name    = "${aws_lambda_function.lambda_function_test.arn}"
  function_version = "1"

  routing_config = {
    additional_version_weights = {
      "2" = 0.5
    }
  }
}
```

» Argument Reference

- `name` - (Required) Name for the alias you are creating. Pattern: `(?!^[0-9]+$)([a-zA-Z0-9-_]+)`
- `description` - (Optional) Description of the alias.
- `function_name` - (Required) The function ARN of the Lambda function for which you want to create an alias.
- `function_version` - (Required) Lambda function version for which you are creating the alias. Pattern: `(\$LATEST|[0-9]+)`.

- `routing_config` - (Optional) The Lambda alias' route configuration settings. Fields documented below

For **routing_config** the following attributes are supported:

- `additional_version_weights` - (Optional) A map that defines the proportion of events that should be sent to different versions of a lambda function.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) identifying your Lambda function alias.
- `invoke_arn` - The ARN to be used for invoking Lambda Function from API Gateway - to be used in `aws_api_gateway_integration`'s `uri`

» Import

Lambda Function Aliases can be imported using the `function_name/alias`, e.g.

```
$ terraform import aws_lambda_function_alias.test_lambda_alias my_test_lambda_function/my_al
```

» Resource: aws_lambda_event_source_mapping

Provides a Lambda event source mapping. This allows Lambda functions to get events from Kinesis, DynamoDB and SQS.

For information about Lambda and how to use it, see What is AWS Lambda?. For information about event source mappings, see CreateEventSourceMapping in the API docs.

» Example Usage

» DynamoDB

```
resource "aws_lambda_event_source_mapping" "example" {
  event_source_arn  = "${aws_dynamodb_table.example.stream_arn}"
  function_name     = "${aws_lambda_function.example.arn}"
  starting_position = "LATEST"
}
```

» Kinesis

```
resource "aws_lambda_event_source_mapping" "example" {
  event_source_arn  = "${aws_kinesis_stream.example.arn}"
  function_name     = "${aws_lambda_function.example.arn}"
  starting_position = "LATEST"
}
```

» SQS

```
resource "aws_lambda_event_source_mapping" "example" {
  event_source_arn = "${aws_sqs_queue.sqs_queue_test.arn}"
  function_name    = "${aws_lambda_function.example.arn}"
}
```

» Argument Reference

- `batch_size` - (Optional) The largest number of records that Lambda will retrieve from your event source at the time of invocation. Defaults to `100` for DynamoDB and Kinesis, `10` for SQS.
- `maximum_batching_window_in_seconds` - (Optional) The maximum amount of time to gather records before invoking the function, in seconds. Records will continue to buffer until either `maximum_batching_window_in_seconds` expires or `batch_size` has been met. Defaults to as soon as records are available in the stream. If the batch it reads from the stream only has one record in it, Lambda only sends one record to the function.
- `event_source_arn` - (Required) The event source ARN - can be a Kinesis stream, DynamoDB stream, or SQS queue.
- `enabled` - (Optional) Determines if the mapping will be enabled on creation. Defaults to `true`.
- `function_name` - (Required) The name or the ARN of the Lambda function that will be subscribing to events.
- `starting_position` - (Optional) The position in the stream where AWS Lambda should start reading. Must be one of `AT_TIMESTAMP` (Kinesis only), `LATEST` or `TRIM_HORIZON` if getting events from Kinesis or DynamoDB. Must not be provided if getting events from SQS. More information about these positions can be found in the AWS DynamoDB Streams API Reference and AWS Kinesis API Reference.
- `starting_position_timestamp` - (Optional) A timestamp in RFC3339 format of the data record which to start reading when using `starting_position` set to `AT_TIMESTAMP`. If a record with this exact timestamp does not exist, the next later record is chosen. If the

timestamp is older than the current trim horizon, the oldest available record is chosen.

» Attributes Reference

- `function_arn` - The the ARN of the Lambda function the event source mapping is sending events to. (Note: this is a computed value that differs from `function_name` above.)
- `last_modified` - The date this resource was last modified.
- `last_processing_result` - The result of the last AWS Lambda invocation of your Lambda function.
- `state` - The state of the event source mapping.
- `state_transition_reason` - The reason the event source mapping is in its current state.
- `uuid` - The UUID of the created event source mapping.

» Import

Lambda event source mappings can be imported using the `UUID` (event source mapping identifier), e.g.

```
$ terraform import aws_lambda_event_source_mapping.event_source_mapping 12345kxodurf3443
```

Note: Terraform will recreate the imported resource as AWS does not expose `startingPosition` information for existing Lambda event source mappings. For information about retrieving event source mappings, see GetEventSourceMapping in the API docs.

» Resource: aws_lambda_function

Provides a Lambda Function resource. Lambda allows you to trigger execution of code in response to events in AWS, enabling serverless backend solutions. The Lambda Function itself includes source code and runtime configuration.

For information about Lambda and how to use it, see What is AWS Lambda?

For a detailed example of setting up Lambda and API Gateway, see Serverless Applications with AWS Lambda and API Gateway.

NOTE: Due to AWS Lambda improved VPC networking changes that began deploying in September 2019, EC2 subnets and security groups associated with Lambda Functions can take up to 45 minutes to successfully delete. Terraform AWS Provider version 2.31.0 and later automatically handles this increased timeout, however prior versions require setting the customizable deletion timeouts of those Terraform resources to 45 minutes (`delete = "45m"`). AWS and

HashiCorp are working together to reduce the amount of time required for resource deletion and updates can be tracked in this GitHub issue.

» Example Usage

» Basic Example

```
resource "aws_iam_role" "iam_for_lambda" {
  name = "iam_for_lambda"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_lambda_function" "test_lambda" {
  filename      = "lambda_function_payload.zip"
  function_name = "lambda_function_name"
  role          = "${aws_iam_role.iam_for_lambda.arn}"
  handler       = "exports.test"

  # The filebase64sha256() function is available in Terraform 0.11.12 and later
  # For Terraform 0.11.11 and earlier, use the base64sha256() function and the file() functi
  # source_code_hash = "${base64sha256(file("lambda_function_payload.zip"))}"
  source_code_hash = "${filebase64sha256("lambda_function_payload.zip")}"

  runtime = "nodejs8.10"

  environment {
    variables = {
      foo = "bar"
    }
  }
}
```

» Lambda Layers

NOTE: The `aws_lambda_layer_version` attribute values for `arn` and `layer_arn` were swapped in version 2.0.0 of the Terraform AWS Provider. For version 1.x, use `layer_arn` references. For version 2.x, use `arn` references.

```
resource "aws_lambda_layer_version" "example" {
  # ... other configuration ...
}

resource "aws_lambda_function" "example" {
  # ... other configuration ...
  layers = ["${aws_lambda_layer_version.example.arn}"]
}
```

» CloudWatch Logging and Permissions

For more information about CloudWatch Logs for Lambda, see the Lambda User Guide.

```
resource "aws_lambda_function" "test_lambda" {
  function_name = "${var.lambda_function_name}"
  ...
  depends_on    = ["aws_iam_role_policy_attachment.lambda_logs", "aws_cloudwatch_log_group.e
}

# This is to optionally manage the CloudWatch Log Group for the Lambda Function.
# If skipping this resource configuration, also add "logs:CreateLogGroup" to the IAM policy
resource "aws_cloudwatch_log_group" "example" {
  name              = "/aws/lambda/${var.lambda_function_name}"
  retention_in_days = 14
}

# See also the following AWS managed policy: AWSLambdaBasicExecutionRole
resource "aws_iam_policy" "lambda_logging" {
  name = "lambda_logging"
  path = "/"
  description = "IAM policy for logging from a lambda"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*",
      "Effect": "Allow"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy_attachment" "lambda_logs" {
  role = "${aws_iam_role.iam_for_lambda.name}"
  policy_arn = "${aws_iam_policy.lambda_logging.arn}"
}
```

» Specifying the Deployment Package

AWS Lambda expects source code to be provided as a deployment package whose structure varies depending on which `runtime` is in use. See Runtimes for the valid values of `runtime`. The expected structure of the deployment package can be found in the AWS Lambda documentation for each runtime.

Once you have created your deployment package you can specify it either directly as a local file (using the `filename` argument) or indirectly via Amazon S3 (using the `s3_bucket`, `s3_key` and `s3_object_version` arguments). When providing the deployment package via S3 it may be useful to use the `aws_s3_bucket_object` resource to upload it.

For larger deployment packages it is recommended by Amazon to upload via S3, since the S3 API has better support for uploading large files efficiently.

» Argument Reference

- `filename` - (Optional) The path to the function's deployment package within the local filesystem. If defined, The `s3_`-prefixed options cannot be used.
- `s3_bucket` - (Optional) The S3 bucket location containing the function's deployment package. Conflicts with `filename`. This bucket must reside in the same AWS region where you are creating the Lambda function.
- `s3_key` - (Optional) The S3 key of an object containing the function's deployment package. Conflicts with `filename`.

- `s3_object_version` - (Optional) The object version containing the function's deployment package. Conflicts with `filename`.
- `function_name` - (Required) A unique name for your Lambda Function.
- `dead_letter_config` - (Optional) Nested block to configure the function's *dead letter queue*. See details below.
- `handler` - (Required) The function entrypoint in your code.
- `role` - (Required) IAM role attached to the Lambda Function. This governs both who / what can invoke your Lambda Function, as well as what resources our Lambda Function has access to. See Lambda Permission Model for more details.
- `description` - (Optional) Description of what your Lambda Function does.
- `layers` - (Optional) List of Lambda Layer Version ARNs (maximum of 5) to attach to your Lambda Function. See Lambda Layers
- `memory_size` - (Optional) Amount of memory in MB your Lambda Function can use at runtime. Defaults to `128`. See Limits
- `runtime` - (Required) See Runtimes for valid values.
- `timeout` - (Optional) The amount of time your Lambda Function has to run in seconds. Defaults to `3`. See Limits
- `reserved_concurrent_executions` - (Optional) The amount of reserved concurrent executions for this lambda function. A value of `0` disables lambda from being triggered and `-1` removes any concurrency limitations. Defaults to Unreserved Concurrency Limits `-1`. See Managing Concurrency
- `publish` - (Optional) Whether to publish creation/change as new Lambda Function Version. Defaults to `false`.
- `vpc_config` - (Optional) Provide this to allow your function to access your VPC. Fields documented below. See Lambda in VPC
- `environment` - (Optional) The Lambda environment's configuration settings. Fields documented below.
- `kms_key_arn` - (Optional) Amazon Resource Name (ARN) of the AWS Key Management Service (KMS) key that is used to encrypt environment variables. If this configuration is not provided when environment variables are in use, AWS Lambda uses a default service key. If this configuration is provided when environment variables are not in use, the AWS Lambda API does not save this configuration and Terraform will show a perpetual difference of adding the key. To fix the perpetual difference, remove this configuration.
- `source_code_hash` - (Optional) Used to trigger updates. Must be set to a base64-encoded SHA256 hash of the package file specified with either `filename` or `s3_key`. The usual way to set this is `filebase64sha256("file.zip")` (Terraform 0.11.12 and later) or `base64sha256(file("file.zip"))` (Terraform 0.11.11 and earlier), where "file.zip" is the local filename of the lambda function source archive.
- `tags` - (Optional) A mapping of tags to assign to the object.

dead_letter_config is a child block with a single argument:

- `target_arn` - (Required) The ARN of an SNS topic or SQS queue to notify when an invocation fails. If this option is used, the function's IAM role must be granted suitable access to write to the target object, which means allowing either the `sns:Publish` or `sqs:SendMessage` action on this ARN, depending on which service is targeted.

tracing_config is a child block with a single argument:

- `mode` - (Required) Can be either `PassThrough` or `Active`. If PassThrough, Lambda will only trace the request from an upstream service if it contains a tracing header with "sampled=1". If Active, Lambda will respect any tracing header it receives from an upstream service. If no tracing header is received, Lambda will call X-Ray for a tracing decision.

vpc_config requires the following:

- `subnet_ids` - (Required) A list of subnet IDs associated with the Lambda function.
- `security_group_ids` - (Required) A list of security group IDs associated with the Lambda function.

NOTE: if both `subnet_ids` and `security_group_ids` are empty then vpc_config is considered to be empty or unset.

For **environment** the following attributes are supported:

- `variables` - (Optional) A map that defines environment variables for the Lambda function.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) identifying your Lambda Function.
- `qualified_arn` - The Amazon Resource Name (ARN) identifying your Lambda Function Version (if versioning is enabled via `publish = true`).
- `invoke_arn` - The ARN to be used for invoking Lambda Function from API Gateway - to be used in `aws_api_gateway_integration`'s `uri`
- `version` - Latest published version of your Lambda Function.
- `last_modified` - The date this resource was last modified.
- `kms_key_arn` - (Optional) The ARN for the KMS encryption key.
- `source_code_hash` - Base64-encoded representation of raw SHA-256 sum of the zip file, provided either via `filename` or `s3_*` parameters.
- `source_code_size` - The size in bytes of the function .zip file.

» Timeouts

`aws_lambda_function` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for slow uploads or EC2 throttling errors.

» Import

Lambda Functions can be imported using the `function_name`, e.g.

```
$ terraform import aws_lambda_function.test_lambda my_test_lambda_function
```

» Resource: aws_lambda_function_event_invoke_config

Manages an asynchronous invocation configuration for a Lambda Function or Alias. More information about asynchronous invocations and the configurable values can be found in the Lambda Developer Guide.

» Example Usage

» Destination Configuration

NOTE: Ensure the Lambda Function IAM Role has necessary permissions for the destination, such as `sqs:SendMessage` or `sns:Publish`, otherwise the API will return a generic `InvalidParameterValueException: The destination ARN arn:PARTITION:SERVICE:REGION:ACCOUNT:RESOURCE is invalid.` error.

```
resource "aws_lambda_function_event_invoke_config" "example" {
  function_name = aws_lambda_alias.example.function_name

  destination_config {
    on_failure {
      destination = aws_sqs_queue.example.arn
    }

    on_success {
      destination = aws_sns_topic.example.arn
    }
  }
}
```

» Error Handling Configuration

```
resource "aws_lambda_function_event_invoke_config" "example" {
  function_name                = aws_lambda_alias.example.function_name
  maximum_event_age_in_seconds = 60
  maximum_retry_attempts       = 0
}
```

» Configuration for Alias Name

```
resource "aws_lambda_function_event_invoke_config" "example" {
  function_name = aws_lambda_alias.example.function_name
  qualifier     = aws_lambda_alias.example.name

  # ... other configuration ...
}
```

» Configuration for Function Latest Unpublished Version

```
resource "aws_lambda_function_event_invoke_config" "example" {
  function_name = aws_lambda_function.example.function_name
  qualifier     = "$LATEST"

  # ... other configuration ...
}
```

» Configuration for Function Published Version

```
resource "aws_lambda_function_event_invoke_config" "example" {
  function_name = aws_lambda_function.example.function_name
  qualifier     = aws_lambda_function.example.version

  # ... other configuration ...
}
```

» Argument Reference

The following arguments are required:

- `function_name` - (Required) Name or Amazon Resource Name (ARN) of the Lambda Function, omitting any version or alias qualifier.

The following arguments are optional:

- `destination_config` - (Optional) Configuration block with destination configuration. See below for details.
- `maximum_event_age_in_seconds` - (Optional) Maximum age of a request that Lambda sends to a function for processing in seconds. Valid values between 60 and 21600.
- `maximum_retry_attempts` - (Optional) Maximum number of times to retry when the function returns an error. Valid values between 0 and 2. Defaults to 2.
- `qualifier` - (Optional) Lambda Function published version, `$LATEST`, or Lambda Alias name.

» destination_config Configuration Block

NOTE: At least one of `on_failure` or `on_success` must be configured when using this configuration block, otherwise remove it completely to prevent perpetual differences in Terraform runs.

The following arguments are optional:

- `on_failure` - (Optional) Configuration block with destination configuration for failed asynchronous invocations. See below for details.
- `on_success` - (Optional) Configuration block with destination configuration for successful asynchronous invocations. See below for details.

» destination_config on_failure Configuration Block

The following arguments are required:

- `destination` - (Required) Amazon Resource Name (ARN) of the destination resource. See the Lambda Developer Guide for acceptable resource types and associated IAM permissions.

» destination_config on_success Configuration Block

The following arguments are required:

- `destination` - (Required) Amazon Resource Name (ARN) of the destination resource. See the Lambda Developer Guide for acceptable resource types and associated IAM permissions.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Fully qualified Lambda Function name or Amazon Resource Name (ARN)

» Import

Lambda Function Event Invoke Configs can be imported using the fully qualified Function name or Amazon Resource Name (ARN), e.g.

ARN without qualifier (all versions and aliases):

```
$ terraform import aws_lambda_function_event_invoke_config.example arn:aws:us-east-1:1234567
```

ARN with qualifier:

```
$ terraform import aws_lambda_function_event_invoke_config.example arn:aws:us-east-1:1234567
```

Name without qualifier (all versions and aliases):

```
$ terraform import aws_lambda_function_event_invoke_config.example my_function
```

Name with qualifier:

```
$ terraform import aws_lambda_function_event_invoke_config.example my_function:production
```

» Resource: aws_lambda_layer_version

Provides a Lambda Layer Version resource. Lambda Layers allow you to reuse shared bits of code across multiple lambda functions.

For information about Lambda Layers and how to use them, see AWS Lambda Layers

» Example Usage

```
resource "aws_lambda_layer_version" "lambda_layer" {
  filename   = "lambda_layer_payload.zip"
  layer_name = "lambda_layer_name"

  compatible_runtimes = ["nodejs8.10"]
}
```

» Specifying the Deployment Package

AWS Lambda Layers expect source code to be provided as a deployment package whose structure varies depending on which `compatible_runtimes` this layer specifies. See Runtimes for the valid values of `compatible_runtimes`.

Once you have created your deployment package you can specify it either directly as a local file (using the `filename` argument) or indirectly via Amazon S3 (using the `s3_bucket`, `s3_key` and `s3_object_version` arguments).

When providing the deployment package via S3 it may be useful to use the `aws_s3_bucket_object` resource to upload it.

For larger deployment packages it is recommended by Amazon to upload via S3, since the S3 API has better support for uploading large files efficiently.

» Argument Reference

- `layer_name` (Required) A unique name for your Lambda Layer
- `filename` (Optional) The path to the function's deployment package within the local filesystem. If defined, The `s3_`-prefixed options cannot be used.
- `s3_bucket` - (Optional) The S3 bucket location containing the function's deployment package. Conflicts with `filename`. This bucket must reside in the same AWS region where you are creating the Lambda function.
- `s3_key` - (Optional) The S3 key of an object containing the function's deployment package. Conflicts with `filename`.
- `s3_object_version` - (Optional) The object version containing the function's deployment package. Conflicts with `filename`.
- `compatible_runtimes` - (Optional) A list of Runtimes this layer is compatible with. Up to 5 runtimes can be specified.
- `description` - (Optional) Description of what your Lambda Layer does.
- `license_info` - (Optional) License info for your Lambda Layer. See License Info.
- `source_code_hash` - (Optional) Used to trigger updates. Must be set to a base64-encoded SHA256 hash of the package file specified with either `filename` or `s3_key`. The usual way to set this is `${filebase64sha256("file.zip")}` (Terraform 0.11.12 or later) or `${base64sha256(file("file.zip"))}` (Terraform 0.11.11 and earlier), where "file.zip" is the local filename of the lambda layer source archive.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) of the Lambda Layer with version.
- `layer_arn` - The Amazon Resource Name (ARN) of the Lambda Layer without version.
- `created_date` - The date this resource was created.
- `source_code_size` - The size in bytes of the function .zip file.
- `version` - This Lamba Layer version.

» Import

Lambda Layers can be imported using `arn`.

```
$ terraform import \
    aws_lambda_layer_version.test_layer \
    arn:aws:lambda:_REGION_:_ACCOUNT_ID_:layer:_LAYER_NAME_:_LAYER_VERSION_
```

» Resource: aws_lambda_permission

Creates a Lambda permission to allow external sources invoking the Lambda function (e.g. CloudWatch Event Rule, SNS or S3).

» Example Usage

```
resource "aws_lambda_permission" "allow_cloudwatch" {
  statement_id  = "AllowExecutionFromCloudWatch"
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.test_lambda.function_name}"
  principal     = "events.amazonaws.com"
  source_arn    = "arn:aws:events:eu-west-1:111122223333:rule/RunDaily"
  qualifier     = "${aws_lambda_alias.test_alias.name}"
}

resource "aws_lambda_alias" "test_alias" {
  name             = "testalias"
  description      = "a sample description"
  function_name    = "${aws_lambda_function.test_lambda.function_name}"
  function_version = "$LATEST"
}

resource "aws_lambda_function" "test_lambda" {
  filename      = "lambdatest.zip"
  function_name = "lambda_function_name"
  role          = "${aws_iam_role.iam_for_lambda.arn}"
  handler       = "exports.handler"
  runtime       = "nodejs8.10"
}

resource "aws_iam_role" "iam_for_lambda" {
  name = "iam_for_lambda"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}
```

» Usage with SNS

```
resource "aws_lambda_permission" "with_sns" {
  statement_id  = "AllowExecutionFromSNS"
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.func.function_name}"
  principal     = "sns.amazonaws.com"
  source_arn    = "${aws_sns_topic.default.arn}"
}

resource "aws_sns_topic" "default" {
  name = "call-lambda-maybe"
}

resource "aws_sns_topic_subscription" "lambda" {
  topic_arn = "${aws_sns_topic.default.arn}"
  protocol  = "lambda"
  endpoint  = "${aws_lambda_function.func.arn}"
}

resource "aws_lambda_function" "func" {
  filename      = "lambdatest.zip"
  function_name = "lambda_called_from_sns"
  role          = "${aws_iam_role.default.arn}"
  handler       = "exports.handler"
  runtime       = "python2.7"
}

resource "aws_iam_role" "default" {
  name = "iam_for_lambda_with_sns"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}
```

» Specify Lambda permissions for API Gateway REST API

```
resource "aws_api_gateway_rest_api" "MyDemoAPI" {
  name        = "MyDemoAPI"
  description = "This is my API for demonstration purposes"
}

resource "aws_lambda_permission" "lambda_permission" {
  statement_id  = "AllowMyDemoAPIInvoke"
  action        = "lambda:InvokeFunction"
  function_name = "MyDemoFunction"
  principal     = "apigateway.amazonaws.com"

  # The /*/*/* part allows invocation from any stage, method and resource path
  # within API Gateway REST API.
  source_arn = "${aws_api_gateway_rest_api.MyDemoAPI.execution_arn}/*/*/*"
}
```

» Argument Reference

- `action` - (Required) The AWS Lambda action you want to allow in this statement. (e.g. `lambda:InvokeFunction`)
- `event_source_token` - (Optional) The Event Source Token to validate. Used with Alexa Skills.
- `function_name` - (Required) Name of the Lambda function whose resource policy you are updating
- `principal` - (Required) The principal who is getting this permission. e.g. `s3.amazonaws.com`, an AWS account ID, or any valid AWS service principal such as `events.amazonaws.com` or `sns.amazonaws.com`.

- `qualifier` - (Optional) Query parameter to specify function version or alias name. The permission will then apply to the specific qualified ARN. e.g. `arn:aws:lambda:aws-region:acct-id:function:function-name:2`
- `source_account` - (Optional) This parameter is used for S3 and SES. The AWS account ID (without a hyphen) of the source owner.
- `source_arn` - (Optional) When granting Amazon S3 or CloudWatch Events permission to invoke your function, you should specify this field with the Amazon Resource Name (ARN) for the S3 Bucket or CloudWatch Events Rule as its value. This ensures that only events generated from the specified bucket or rule can invoke the function. API Gateway ARNs have a unique structure described here.
- `statement_id` - (Optional) A unique statement identifier. By default generated by Terraform.
- `statement_id_prefix` - (Optional) A statement identifier prefix. Terraform will generate a unique suffix. Conflicts with `statement_id`.

» Import

Lambda permission statements can be imported using function_name/statement_id, with an optional qualifier, e.g.

```
$ terraform import aws_lambda_permission.test_lambda_permission my_test_lambda_function/Allo

$ terraform import aws_lambda_permission.test_lambda_permission my_test_lambda_function:qual
```

» Resource: aws_lambda_provisioned_concurrency_config

Manages a Lambda Provisioned Concurrency Configuration.

» Example Usage

» Alias Name

```
resource "aws_lambda_provisioned_concurrency_config" "example" {
  function_name                     = aws_lambda_alias.example.function_name
  provisioned_concurrent_executions = 1
  qualifier                         = aws_lambda_alias.example.name
}
```

» Function Version

```
resource "aws_lambda_provisioned_concurrency_config" "example" {
  function_name                     = aws_lambda_function.example.function_name
  provisioned_concurrent_executions = 1
  qualifier                         = aws_lambda_function.example.version
}
```

» Argument Reference

The following arguments are required:

- `function_name` - (Required) Name or Amazon Resource Name (ARN) of the Lambda Function.
- `provisioned_concurrent_executions` - (Required) Amount of capacity to allocate. Must be greater than or equal to `1`.
- `qualifier` - (Required) Lambda Function version or Lambda Alias name.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Lambda Function name and qualifier separated by a colon (`:`).

» Timeouts

`aws_lambda_provisioned_concurrency_config` provides the following Timeouts configuration options:

- `create` - (Default `15 minutes`) How long to wait for the Lambda Provisioned Concurrency Config to be ready on creation.
- `update` - (Default `15 minutes`) How long to wait for the Lambda Provisioned Concurrency Config to be ready on update.

» Import

Lambda Provisioned Concurrency Configs can be imported using the `function_name` and `qualifier` separated by a colon (`:`), e.g.

```
$ terraform import aws_lambda_provisioned_concurrency_config.example my_function:production
```

» Resource: aws_licensemanager_association

Provides a License Manager association.

Note: License configurations can also be associated with launch templates by specifying the `license_specifications` block for an `aws_launch_template`.

» Example Usage

```
data "aws_ami" "example" {
  most_recent = true
  owners      = ["amazon"]

  filter {
    name   = "name"
    values = ["amzn-ami-vpc-nat*"]
  }
}

resource "aws_instance" "example" {
  ami           = "${data.aws_ami.example.id}"
  instance_type = "t2.micro"
}

resource "aws_licensemanager_license_configuration" "example" {
  name                  = "Example"
  license_counting_type = "Instance"
}

resource "aws_licensemanager_association" "example" {
  license_configuration_arn = "${aws_licensemanager_license_configuration.example.arn}"
  resource_arn              = "${aws_instance.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `license_configuration_arn` - (Required) ARN of the license configuration.
- `resource_arn` - (Required) ARN of the resource associated with the license configuration.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The license configuration ARN.

» Import

License configurations can be imported in the form `resource_arn,license_configuration_arn`, e.g.

```
$ terraform import aws_licensemanager_association.example arn:aws:ec2:eu-west-1:123456789012
```

» Resource: aws_licensemanager_license_configuration

Provides a License Manager license configuration resource.

Note: Removing the `license_count` attribute is not supported by the License Manager API - use `terraform taint aws_licensemanager_license_configuration.<id>` to recreate the resource instead.

» Example Usage

```
resource "aws_licensemanager_license_configuration" "example" {
  name                     = "Example"
  description              = "Example"
  license_count            = 10
  license_count_hard_limit = true
  license_counting_type    = "Socket"

  license_rules = [
    "#minimumSockets=2",
  ]

  tags = {
    foo = "barr"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the license configuration.
- `description` - (Optional) Description of the license configuration.
- `license_count` - (Optional) Number of licenses managed by the license configuration.
- `license_count_hard_limit` - (Optional) Sets the number of available licenses as a hard limit.

- `license_counting_type` - (Required) Dimension to use to track license inventory. Specify either `vCPU`, `Instance`, `Core` or `Socket`.
- `license_rules` - (Optional) Array of configured License Manager rules.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Rules

License rules should be in the format of `#RuleType=RuleValue`. Supported rule types:

- `minimumVcpus` - Resource must have minimum vCPU count in order to use the license. Default: 1
- `maximumVcpus` - Resource must have maximum vCPU count in order to use the license. Default: unbounded, limit: 10000
- `minimumCores` - Resource must have minimum core count in order to use the license. Default: 1
- `maximumCores` - Resource must have maximum core count in order to use the license. Default: unbounded, limit: 10000
- `minimumSockets` - Resource must have minimum socket count in order to use the license. Default: 1
- `maximumSockets` - Resource must have maximum socket count in order to use the license. Default: unbounded, limit: 10000
- `allowedTenancy` - Defines where the license can be used. If set, restricts license usage to selected tenancies. Specify a comma delimited list of `EC2-Default`, `EC2-DedicatedHost`, `EC2-DedicatedInstance`

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The license configuration ARN.

» Import

License configurations can be imported using the `id`, e.g.

```
$ terraform import aws_licensemanager_license_configuration.example arn:aws:license-manager:
```

» Resource: aws_lightsail_domain

Creates a domain resource for the specified domain (e.g., example.com). You cannot register a new domain name using Lightsail. You must register a domain name using Amazon Route 53 or another domain name registrar. If you have already registered your domain, you can enter its name in this parameter to manage the DNS records for that domain.

Note: Lightsail is currently only supported in a limited number of AWS Regions, please see "Regions and Availability Zones in Amazon Lightsail" for more details

» Example Usage, creating a new domain

```
resource "aws_lightsail_domain" "domain_test" {
  domain_name = "mydomain.com"
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Required) The name of the Lightsail domain to manage

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The name used for this domain
- `arn` - The ARN of the Lightsail domain

» Resource: aws_lightsail_instance

Provides a Lightsail Instance. Amazon Lightsail is a service to provide easy virtual private servers with custom software already setup. See What is Amazon Lightsail? for more information.

Note: Lightsail is currently only supported in a limited number of AWS Regions, please see "Regions and Availability Zones in Amazon Lightsail" for more details

» Example Usage

```
# Create a new GitLab Lightsail Instance
resource "aws_lightsail_instance" "gitlab_test" {
  name              = "custom_gitlab"
  availability_zone = "us-east-1b"
  blueprint_id      = "string"
  bundle_id         = "string"
  key_pair_name     = "some_key_name"
  tags = {
    foo = "bar"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Lightsail Instance. Names be unique within each AWS Region in your Lightsail account.
- `availability_zone` - (Required) The Availability Zone in which to create your instance (see list below)
- `blueprint_id` - (Required) The ID for a virtual private server image (see list below)
- `bundle_id` - (Required) The bundle of specification information (see list below)
- `key_pair_name` - (Optional) The name of your key pair. Created in the Lightsail console (cannot use `aws_key_pair` at this time)
- `user_data` - (Optional) launch script to configure server with additional user data
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Availability Zones

Lightsail currently supports the following Availability Zones (e.g. `us-east-1a`):

- `ap-northeast-1{a,c,d}`
- `ap-northeast-2{a,c}`
- `ap-south-1{a,b}`
- `ap-southeast-1{a,b,c}`
- `ap-southeast-2{a,b,c}`
- `ca-central-1{a,b}`
- `eu-central-1{a,b,c}`
- `eu-west-1{a,b,c}`
- `eu-west-2{a,b,c}`
- `eu-west-3{a,b,c}`
- `us-east-1{a,b,c,d,e,f}`
- `us-east-2{a,b,c}`
- `us-west-2{a,b,c}`

» Blueprints

Lightsail currently supports the following Blueprint IDs:

» OS Only

- `amazon_linux_2018_03_0_2`
- `centos_7_1901_01`
- `debian_8_7`
- `debian_9_5`
- `freebsd_11_1`
- `opensuse_42_2`
- `ubuntu_16_04_2`
- `ubuntu_18_04`

» Apps and OS

- `drupal_8_5_6`
- `gitlab_11_1_4_1`
- `joomla_3_8_11`
- `lamp_5_6_37_2`
- `lamp_7_1_20_1`
- `magento_2_2_5`
- `mean_4_0_1`
- `nginx_1_14_0_1`
- `nodejs_10_8_0`
- `plesk_ubuntu_17_8_11_1`
- `redmine_3_4_6`
- `wordpress_4_9_8`
- `wordpress_multisite_4_9_8`

» Bundles

Lightsail currently supports the following Bundle IDs (e.g. an instance in `ap-northeast-1` would use `small_2_0`):

» Prefix

A Bundle ID starts with one of the below size prefixes:

- `nano_`
- `micro_`
- `small_`
- `medium_`
- `large_`
- `xlarge_`
- `2xlarge_`

» Suffix

A Bundle ID ends with one of the following suffixes depending on Availability Zone:

- ap-northeast-1: `2_0`
- ap-northeast-2: `2_0`
- ap-south-1: `2_1`
- ap-southeast-1: `2_0`
- ap-southeast-2: `2_2`
- ca-central-1: `2_0`
- eu-central-1: `2_0`
- eu-west-1: `2_0`
- eu-west-2: `2_0`
- eu-west-3: `2_0`
- us-east-1: `2_0`
- us-east-2: `2_0`
- us-west-2: `2_0`

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The ARN of the Lightsail instance (matches `arn`).
- `arn` - The ARN of the Lightsail instance (matches `id`).
- `created_at` - The timestamp when the instance was created.
- `availability_zone`
- `blueprint_id`
- `bundle_id`
- `key_pair_name`
- `user_data`

» Import

Lightsail Instances can be imported using their name, e.g.

```
$ terraform import aws_lightsail_instance.gitlab_test 'custom gitlab'
```

» Resource: aws_lightsail_key_pair

Provides a Lightsail Key Pair, for use with Lightsail Instances. These key pairs are separate from EC2 Key Pairs, and must be created or imported for use with Lightsail.

Note: Lightsail is currently only supported in a limited number of AWS Regions, please see ”Regions and Availability Zones in Amazon Lightsail” for more details

» Example Usage, creating a new Key Pair

```
# Create a new Lightsail Key Pair
resource "aws_lightsail_key_pair" "lg_key_pair" {
  name = "lg_key_pair"
}
```

» Create new Key Pair, encrypting the private key with a PGP Key

```
resource "aws_lightsail_key_pair" "lg_key_pair" {
  name    = "lg_key_pair"
  pgp_key = "keybase:keybaseusername"
}
```

» Import an existing public key

```
resource "aws_lightsail_key_pair" "lg_key_pair" {
  name       = "importing"
  public_key = "${file("~/.ssh/id_rsa.pub")}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the Lightsail Key Pair. If omitted, a unique name will be generated by Terraform
- `pgp_key` – (Optional) An optional PGP key to encrypt the resulting private key material. Only used when creating a new key pair
- `public_key` - (Required) The public key material. This public key will be imported into Lightsail

NOTE: a PGP key is not required, however it is strongly encouraged. Without a PGP key, the private key material will be stored in state unencrypted. `pgp_key` is ignored if `public_key` is supplied.

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The name used for this key pair
- `arn` - The ARN of the Lightsail key pair
- `fingerprint` - The MD5 public key fingerprint as specified in section 4 of RFC 4716.
- `public_key` - the public key, base64 encoded
- `private_key` - the private key, base64 encoded. This is only populated when creating a new key, and when no `pgp_key` is provided
- `encrypted_private_key` – the private key material, base 64 encoded and encrypted with the given `pgp_key`. This is only populated when creating a new key and `pgp_key` is supplied
- `encrypted_fingerprint` - The MD5 public key fingerprint for the encrypted private key

» Import

Lightsail Key Pairs cannot be imported, because the private and public key are only available on initial creation.

» Resource: aws_lightsail_static_ip

Allocates a static IP address.

Note: Lightsail is currently only supported in a limited number of AWS Regions, please see ”Regions and Availability Zones in Amazon Lightsail” for more details

» Example Usage

```
resource "aws_lightsail_static_ip" "test" {
  name = "example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name for the allocated static IP

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `arn` - The ARN of the Lightsail static IP
- `ip_address` - The allocated static IP address
- `support_code` - The support code.

» Resource: aws_lightsail_static_ip_attachment

Provides a static IP address attachment - relationship between a Lightsail static IP & Lightsail instance.

Note: Lightsail is currently only supported in a limited number of AWS Regions, please see "Regions and Availability Zones in Amazon Lightsail" for more details

» Example Usage

```
resource "aws_lightsail_static_ip_attachment" "test" {
  static_ip_name = "${aws_lightsail_static_ip.test.id}"
  instance_name  = "${aws_lightsail_instance.test.id}"
}

resource "aws_lightsail_static_ip" "test" {
  name = "example"
}

resource "aws_lightsail_instance" "test" {
  name              = "example"
  availability_zone = "us-east-1b"
  blueprint_id      = "string"
  bundle_id         = "string"
  key_pair_name     = "some_key_name"
}
```

» Argument Reference

The following arguments are supported:

- `static_ip_name` - (Required) The name of the allocated static IP
- `instance_name` - (Required) The name of the Lightsail instance to attach the IP to

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `ip_address` - The allocated static IP address

» Resource: aws_macie_member_account_association

Associates an AWS account with Amazon Macie as a member account.

NOTE: Before using Amazon Macie for the first time it must be enabled manually. Instructions are here.

» Example Usage

```
resource "aws_macie_member_account_association" "example" {
  member_account_id = "123456789012"
}
```

» Argument Reference

The following arguments are supported:

- `member_account_id` - (Required) The ID of the AWS account that you want to associate with Amazon Macie as a member account.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the association.

» Resource: aws_macie_s3_bucket_association

Associates an S3 resource with Amazon Macie for monitoring and data classification.

NOTE: Before using Amazon Macie for the first time it must be enabled manually. Instructions are here.

» Example Usage

```
resource "aws_macie_s3_bucket_association" "example" {
  bucket_name = "tf-macie-example"
  prefix      = "data"

  classification_type {
    one_time = "FULL"
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket_name` - (Required) The name of the S3 bucket that you want to associate with Amazon Macie.
- `classification_type` - (Optional) The configuration of how Amazon Macie classifies the S3 objects.
- `member_account_id` - (Optional) The ID of the Amazon Macie member account whose S3 resources you want to associate with Macie. If `member_account_id` isn't specified, the action associates specified S3 resources with Macie for the current master account.
- `prefix` - (Optional) Object key prefix identifying one or more S3 objects to which the association applies.

The `classification_type` object supports the following:

- `continuous` - (Optional) A string value indicating that Macie perform a one-time classification of all of the existing objects in the bucket. The only valid value is the default value, `FULL`.
- `one_time` - (Optional) A string value indicating whether or not Macie performs a one-time classification of all of the existing objects in the bucket. Valid values are `NONE` and `FULL`. Defaults to `NONE` indicating that Macie only classifies objects that are added after the association was created.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the association.

» Data Source: aws_mq_broker

Provides information about a MQ Broker.

» Example Usage

```
variable "broker_id" {
  type    = "string"
  default = ""
}

variable "broker_name" {
  type    = "string"
  default = ""
}

data "aws_mq_broker" "by_id" {
  broker_id = "${var.broker_id}"
}

data "aws_mq_broker" "by_name" {
  broker_name = "${var.broker_name}"
}
```

» Argument Reference

The following arguments are supported:

- `broker_id` - (Optional) The unique id of the mq broker.
- `broker_name` - (Optional) The unique name of the mq broker.

» Attributes Reference

See the `aws_mq_broker` resource for details on the returned attributes. They are identical except for user password, which is not returned when describing broker.

» Resource: aws_mq_broker

Provides an MQ Broker Resource. This resources also manages users for the broker.

For more information on Amazon MQ, see Amazon MQ documentation.

Changes to an MQ Broker can occur when you change a parameter, such as `configuration` or `user`, and are reflected in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because a modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately (see documentation below).

Note: using `apply_immediately` can result in a brief downtime as the broker reboots.

Note: All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_mq_broker" "example" {
  broker_name = "example"

  configuration {
    id       = "${aws_mq_configuration.test.id}"
    revision = "${aws_mq_configuration.test.latest_revision}"
  }

  engine_type        = "ActiveMQ"
  engine_version     = "5.15.0"
  host_instance_type = "mq.t2.micro"
  security_groups    = ["${aws_security_group.test.id}"]

  user {
    username = "ExampleUser"
    password = "MindTheGap"
  }
}
```

» Argument Reference

The following arguments are supported:

- `apply_immediately` - (Optional) Specifies whether any broker modifications are applied immediately, or during the next maintenance window. Default is `false`.
- `auto_minor_version_upgrade` - (Optional) Enables automatic upgrades to new minor versions for brokers, as Apache releases the versions.
- `broker_name` - (Required) The name of the broker.

- `configuration` - (Optional) Configuration of the broker. See below.
- `deployment_mode` - (Optional) The deployment mode of the broker. Supported: `SINGLE_INSTANCE` and `ACTIVE_STANDBY_MULTI_AZ`. Defaults to `SINGLE_INSTANCE`.
- `encryption_options` - (Optional) Configuration block containing encryption options. See below.
- `engine_type` - (Required) The type of broker engine. Currently, Amazon MQ supports only `ActiveMQ`.
- `engine_version` - (Required) The version of the broker engine. See the AmazonMQ Broker Engine docs for supported versions.
- `host_instance_type` - (Required) The broker's instance type. e.g. `mq.t2.micro` or `mq.m4.large`
- `publicly_accessible` - (Optional) Whether to enable connections from applications outside of the VPC that hosts the broker's subnets.
- `security_groups` - (Required) The list of security group IDs assigned to the broker.
- `subnet_ids` - (Optional) The list of subnet IDs in which to launch the broker. A `SINGLE_INSTANCE` deployment requires one subnet. An `ACTIVE_STANDBY_MULTI_AZ` deployment requires two subnets.
- `maintenance_window_start_time` - (Optional) Maintenance window start time. See below.
- `logs` - (Optional) Logging configuration of the broker. See below.
- `user` - (Optional) The list of all ActiveMQ usernames for the specified broker. See below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Nested Fields

» `configuration`

- `id` - (Optional) The Configuration ID.
- `revision` - (Optional) Revision of the Configuration.

» `encryption_options`

- `kms_key_id` - (Optional) Amazon Resource Name (ARN) of Key Management Service (KMS) Customer Master Key (CMK) to use for encryption at rest. Requires setting `use_aws_owned_key` to `false`. To perform drift detection when AWS managed CMKs or customer managed CMKs are in use, this value must be configured.
- `use_aws_owned_key` - (Optional) Boolean to enable an AWS owned Key Management Service (KMS) Customer Master Key (CMK) that is not in your account. Defaults to `true`. Setting to `false` without configuring `kms_key_id` will create an AWS managed Customer Master Key (CMK) aliased to `aws/mq` in your account.

» `maintenance_window_start_time`

- `day_of_week` - (Required) The day of the week. e.g. `MONDAY`, `TUESDAY`, or `WEDNESDAY`
- `time_of_day` - (Required) The time, in 24-hour format. e.g. `02:00`
- `time_zone` - (Required) The time zone, UTC by default, in either the Country/City format, or the UTC offset format. e.g. `CET`

NOTE: AWS currently does not support updating the maintenance window beyond resource creation.

» `logs`

- `general` - (Optional) Enables general logging via CloudWatch. Defaults to `false`.
- `audit` - (Optional) Enables audit logging. User management action made using JMX or the ActiveMQ Web Console is logged. Defaults to `false`.

» `user`

- `console_access` - (Optional) Whether to enable access to the ActiveMQ Web Console for the user.
- `groups` - (Optional) The list of groups (20 maximum) to which the ActiveMQ user belongs.
- `password` - (Required) The password of the user. It must be 12 to 250 characters long, at least 4 unique characters, and must not contain commas.
- `username` - (Required) The username of the user.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The unique ID that Amazon MQ generates for the broker.
- `arn` - The ARN of the broker.
- `instances` - A list of information about allocated brokers (both active & standby).
  - `instances.0.console_url` - The URL of the broker's ActiveMQ Web Console.
  - `instances.0.ip_address` - The IP Address of the broker.
  - `instances.0.endpoints` - The broker's wire-level protocol endpoints in the following order & format referenceable e.g. as `instances.0.endpoints.0` (SSL):
    - `ssl://broker-id.mq.us-west-2.amazonaws.com:61617`
    - `amqp+ssl://broker-id.mq.us-west-2.amazonaws.com:5671`

- * stomp+ssl://broker-id.mq.us-west-2.amazonaws.com:61614
  * mqtt+ssl://broker-id.mq.us-west-2.amazonaws.com:8883
  * wss://broker-id.mq.us-west-2.amazonaws.com:61619

» Import

MQ Broker is currently not importable.

» Resource: aws_mq_configuration

Provides an MQ Configuration Resource.

For more information on Amazon MQ, see Amazon MQ documentation.

» Example Usage

```
resource "aws_mq_configuration" "example" {
  description    = "Example Configuration"
  name           = "example"
  engine_type    = "ActiveMQ"
  engine_version = "5.15.0"

  data = <<DATA
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<broker xmlns="http://activemq.apache.org/schema/core">
  <plugins>
    <forcePersistencyModeBrokerPlugin persistenceFlag="true"/>
    <statisticsBrokerPlugin/>
    <timeStampingBrokerPlugin ttlCeiling="86400000" zeroExpirationOverride="86400000"/>
  </plugins>
</broker>
DATA
}
```

» Argument Reference

The following arguments are supported:

- `data` - (Required) The broker configuration in XML format. See official docs for supported parameters and format of the XML.
- `description` - (Optional) The description of the configuration.
- `engine_type` - (Required) The type of broker engine.

- `engine_version` - (Required) The version of the broker engine.
- `name` - (Required) The name of the configuration
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The unique ID that Amazon MQ generates for the configuration.
- `arn` - The ARN of the configuration.
- `latest_revision` - The latest revision of the configuration.

» Import

MQ Configurations can be imported using the configuration ID, e.g.

```
$ terraform import aws_mq_configuration.example c-0187d1eb-88c8-475a-9b79-16ef5a10c94f
```

» Resource: aws_media_convert_queue

Provides an AWS Elemental MediaConvert Queue.

» Example Usage

```
resource "aws_media_convert_queue" "test" {
  name = "tf-test-queue"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique identifier describing the queue
- `description` - (Optional) A description of the queue
- `pricing_plan` - (Optional) Specifies whether the pricing plan for the queue is on-demand or reserved. Valid values are `ON_DEMAND` or `RESERVED`. Default to `ON_DEMAND`.
- `reservation_plan_settings` - (Optional) A detail pricing plan of the reserved queue. See below.
- `status` - (Optional) A status of the queue. Valid values are `ACTIVE` or `RESERVED`. Default to `PAUSED`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Nested Fields

» `reservation_plan_settings`

- `commitment` - (Required) The length of the term of your reserved queue pricing plan commitment. Valid value is `ONE_YEAR`.
- `renewal_type` - (Required) Specifies whether the term of your reserved queue pricing plan. Valid values are `AUTO_RENEW` or `EXPIRE`.
- `reserved_slots` - (Required) Specifies the number of reserved transcode slots (RTS) for queue.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The same as `name`
- `arn` - The Arn of the queue

» Import

Media Convert Queue can be imported via the queue name, e.g.

```
$ terraform import aws_media_convert_queue.test tf-test-queue
```

» Resource: aws_media_package_channel

Provides an AWS Elemental MediaPackage Channel.

» Example Usage

```
resource "aws_media_package_channel" "kittens" {
  channel_id  = "kitten-channel"
  description = "A channel dedicated to amusing videos of kittens."
}
```

» Argument Reference

The following arguments are supported:

- `channel_id` - (Required) A unique identifier describing the channel
- `description` - (Optional) A description of the channel
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The same as `channel_id`
- `arn` - The ARN of the channel
- `hls_ingest` - A single item list of HLS ingest information
  - `ingest_endpoints` - A list of the ingest endpoints
  - `password` - The password
  - `url` - The URL
  - `username` - The username

» Import

Media Package Channels can be imported via the channel ID, e.g.

```
$ terraform import aws_media_package_channel.kittens kittens-channel
```

» Resource: aws_media_store_container

Provides a MediaStore Container.

» Example Usage

```
resource "aws_media_store_container" "example" {
  name = "example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the container. Must contain alphanumeric characters or underscores.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the container.
- `endpoint` - The DNS endpoint of the container.

» Import

MediaStore Container can be imported using the MediaStore Container Name, e.g.

```
$ terraform import aws_media_store_container.example example
```

» Resource: aws_media_store_container_policy

Provides a MediaStore Container Policy.

» Example Usage

```
data "aws_region" "current" {}

data "aws_caller_identity" "current" {}

resource "aws_media_store_container" "example" {
  name = "example"
}

resource "aws_media_store_container_policy" "example" {
  container_name = "${aws_media_store_container.example.name}"

  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "MediaStoreFullAccess",
        "Action": [ "mediastore:*" ],
        "Principal": {"AWS" : "arn:aws:iam::${data.aws_caller_identity.current.account_id}:r
        "Effect": "Allow",
        "Resource": "arn:aws:mediastore:${data.aws_caller_identity.current.account_id}:${dat
        "Condition": {
            "Bool": { "aws:SecureTransport": "true" }
        }
    }]
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `container_name` - (Required) The name of the container.
- `policy` - (Required) The contents of the policy. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

» Import

MediaStore Container Policy can be imported using the MediaStore Container Name, e.g.

```
$ terraform import aws_media_store_container_policy.example example
```

» Data Source: aws_msk_cluster

Get information on an Amazon MSK Cluster.

» Example Usage

```
data "aws_msk_cluster" "example" {
  cluster_name = "example"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_name` - (Required) Name of the cluster.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the MSK cluster.
- `bootstrap_brokers` - A comma separated list of one or more hostname:port pairs of Kafka brokers suitable to boostrap connectivity to the Kafka cluster.
- `bootstrap_brokers_tls` - A comma separated list of one or more DNS names (or IPs) and TLS port pairs kafka brokers suitable to boostrap connectivity to the kafka cluster.

- `kafka_version` - Apache Kafka version.
- `number_of_broker_nodes` - Number of broker nodes in the cluster.
- `tags` - Map of key-value pairs assigned to the cluster.
- `zookeeper_connect_string` - A comma separated list of one or more IP:port pairs to use to connect to the Apache Zookeeper cluster.

» Data Source: aws_msk_configuration

Get information on an Amazon MSK Configuration.

» Example Usage

```
data "aws_msk_configuration" "example" {
  name = "example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the configuration.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the configuration.
- `latest_revision` - Latest revision of the configuration.
- `description` - Description of the configuration.
- `kafka_versions` - List of Apache Kafka versions which can use this configuration.
- `server_properties` - Contents of the server.properties file.

» Resource: aws_msk_cluster

Manages AWS Managed Streaming for Kafka cluster

» Example Usage

```
resource "aws_vpc" "vpc" {
  cidr_block = "192.168.0.0/22"
}

data "aws_availability_zones" "azs" {
  state = "available"
}

resource "aws_subnet" "subnet_az1" {
  availability_zone = "${data.aws_availability_zones.azs.names[0]}"
  cidr_block        = "192.168.0.0/24"
  vpc_id            = "${aws_vpc.vpc.id}"
}

resource "aws_subnet" "subnet_az2" {
  availability_zone = "${data.aws_availability_zones.azs.names[1]}"
  cidr_block        = "192.168.1.0/24"
  vpc_id            = "${aws_vpc.vpc.id}"
}

resource "aws_subnet" "subnet_az3" {
  availability_zone = "${data.aws_availability_zones.azs.names[2]}"
  cidr_block        = "192.168.2.0/24"
  vpc_id            = "${aws_vpc.vpc.id}"
}

resource "aws_security_group" "sg" {
  vpc_id = "${aws_vpc.vpc.id}"
}

resource "aws_kms_key" "kms" {
  description = "example"
}

resource "aws_msk_cluster" "example" {
  cluster_name           = "example"
  kafka_version          = "2.1.0"
  number_of_broker_nodes = 3

  broker_node_group_info {
    instance_type  = "kafka.m5.large"
    ebs_volume_size = 1000
    client_subnets = [
      "${aws_subnet.subnet_az1.id}",
      "${aws_subnet.subnet_az2.id}",
      "${aws_subnet.subnet_az3.id}",
    ]
    security_groups = [ "${aws_security_group.sg.id}" ]
  }

  encryption_info {
    encryption_at_rest_kms_key_arn = "${aws_kms_key.kms.arn}"
  }

  tags = {
    foo = "bar"
  }
}

output "zookeeper_connect_string" {
  value = "${aws_msk_cluster.example.zookeeper_connect_string}"
}

output "bootstrap_brokers" {
  description = "Plaintext connection host:port pairs"
  value       = "${aws_msk_cluster.example.bootstrap_brokers}"
}

output "bootstrap_brokers_tls" {
  description = "TLS connection host:port pairs"
  value       = "${aws_msk_cluster.example.bootstrap_brokers_tls}"
}
```

» Argument Reference

The following arguments are supported:

- `broker_node_group_info` - (Required) Configuration block for the broker nodes of the Kafka cluster.
- `cluster_name` - (Required) Name of the MSK cluster.
- `kafka_version` - (Required) Specify the desired Kafka software version.
- `number_of_broker_nodes` - (Required) The desired total number of broker nodes in the kafka cluster. It must be a multiple of the number of specified client subnets.
- `client_authentication` - (Optional) Configuration block for specifying a client authentication. See below.
- `configuration_info` - (Optional) Configuration block for specifying a MSK Configuration to attach to Kafka brokers. See below.

- `encryption_info` - (Optional) Configuration block for specifying encryption. See below.
- `enhanced_monitoring` - (Optional) Specify the desired enhanced MSK CloudWatch monitoring level. See Monitoring Amazon MSK with Amazon CloudWatch
- `tags` - (Optional) A mapping of tags to assign to the resource

» broker_node_group_info Argument Reference

- `client_subnets` - (Required) A list of subnets to connect to in client VPC (documentation).
- `ebs_volume_size` - (Required) The size in GiB of the EBS volume for the data drive on each broker node.
- `instance_type` - (Required) Specify the instance type to use for the kafka brokers. e.g. kafka.m5.large. (Pricing info)
- `security_groups` - (Required) A list of the security groups to associate with the elastic network interfaces to control who can communicate with the cluster.
- `az_distribution` - (Optional) The distribution of broker nodes across availability zones (documentation). Currently the only valid value is `DEFAULT`.

» client_authentication Argument Reference

- `tls` - (Optional) Configuration block for specifying TLS client authentication. See below.

» client_authentication tls Argument Reference

- `certificate_authority_arns` - (Optional) List of ACM Certificate Authority Amazon Resource Names (ARNs).

» configuration_info Argument Reference

- `arn` - (Required) Amazon Resource Name (ARN) of the MSK Configuration to use in the cluster.
- `revision` - (Required) Revision of the MSK Configuration to use in the cluster.

» encryption_info Argument Reference

- `encryption_in_transit` - (Optional) Configuration block to specify encryption in transit. See below.

- `encryption_at_rest_kms_key_arn` - (Optional) You may specify a KMS key short ID or ARN (it will always output an ARN) to use for encrypting your data at rest. If no key is specified, an AWS managed KMS ('aws/msk' managed service) key will be used for encrypting the data at rest.

» encryption_info encryption_in_transit Argument Reference

- `client_broker` - (Optional) Encryption setting for data in transit between clients and brokers. Valid values: `TLS`, `TLS_PLAINTEXT`, and `PLAINTEXT`. Default value: `TLS_PLAINTEXT`.
- `in_cluster` - (Optional) Whether data communication among broker nodes is encrypted. Default value: `true`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the MSK cluster.
- `bootstrap_brokers` - A comma separated list of one or more hostname:port pairs of kafka brokers suitable to boostrap connectivity to the kafka cluster. Only contains value if `client_broker` encryption in transit is set to `PLAINTEXT` or `TLS_PLAINTEXT`.
- `bootstrap_brokers_tls` - A comma separated list of one or more DNS names (or IPs) and TLS port pairs kafka brokers suitable to boostrap connectivity to the kafka cluster. Only contains value if `client_broker` encryption in transit is set to `TLS_PLAINTEXT` or `TLS`.
- `current_version` - Current version of the MSK Cluster used for updates, e.g. `K13V1IB3VIYZZH`
- `encryption_info.0.encryption_at_rest_kms_key_arn` - The ARN of the KMS key used for encryption at rest of the broker data volumes.
- `zookeeper_connect_string` - A comma separated list of one or more IP:port pairs to use to connect to the Apache Zookeeper cluster.

» Import

MSK clusters can be imported using the cluster `arn`, e.g.

```
$ terraform import aws_msk_cluster.example arn:aws:kafka:us-west-2:123456789012:cluster/exam
```

» Resource: aws_msk_configuration

Manages an Amazon Managed Streaming for Kafka configuration. More information can be found on the MSK Developer Guide.

NOTE: The API does not support deleting MSK configurations. Removing this Terraform resource will only remove the Terraform state for it.

» Example Usage

```
resource "aws_msk_configuration" "example" {
  kafka_versions = ["2.1.0"]
  name           = "example"

  server_properties = <<PROPERTIES
auto.create.topics.enable = true
delete.topic.enable = true
PROPERTIES
}
```

» Argument Reference

The following arguments are supported:

- `server_properties` - (Required) Contents of the server.properties file. Supported properties are documented in the MSK Developer Guide.
- `kafka_versions` - (Required) List of Apache Kafka versions which can use this configuration.
- `name` - (Required) Name of the configuration.
- `description` - (Optional) Description of the configuration.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the configuration.
- `latest_revision` - Latest revision of the configuration.

» Import

MSK configurations can be imported using the configuration ARN, e.g.

```
$ terraform import aws_msk_cluster.example arn:aws:kafka:us-west-2:123456789012:configuratio
```

» Resource: aws_neptune_parameter_group

Manages a Neptune Parameter Group

» Example Usage

```
resource "aws_neptune_parameter_group" "example" {
  family = "neptune1"
  name   = "example"

  parameter {
    name  = "neptune_query_timeout"
    value = "25"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required, Forces new resource) The name of the Neptune parameter group.
- `family` - (Required) The family of the Neptune parameter group.
- `description` - (Optional) The description of the Neptune parameter group. Defaults to ”Managed by Terraform”.
- `parameter` - (Optional) A list of Neptune parameters to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Parameter blocks support the following:

- `name` - (Required) The name of the Neptune parameter.
- `value` - (Required) The value of the Neptune parameter.
- `apply_method` - (Optional) The apply method of the Neptune parameter. Valid values are `immediate` and `pending-reboot`. Defaults to `pending-reboot`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Neptune parameter group name.
- `arn` - The Neptune parameter group Amazon Resource Name (ARN).

» Import

Neptune Parameter Groups can be imported using the `name`, e.g.

```
$ terraform import aws_neptune_parameter_group.some_pg some-pg
```

» Resource: aws_neptune_subnet_group

Provides an Neptune subnet group resource.

» Example Usage

```
resource "aws_neptune_subnet_group" "default" {
  name       = "main"
  subnet_ids = ["${aws_subnet.frontend.id}", "${aws_subnet.backend.id}"]

  tags = {
    Name = "My neptune subnet group"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the neptune subnet group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `description` - (Optional) The description of the neptune subnet group. Defaults to ”Managed by Terraform”.
- `subnet_ids` - (Required) A list of VPC subnet IDs.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The neptune subnet group name.
- `arn` - The ARN of the neptune subnet group.

» Import

Neptune Subnet groups can be imported using the `name`, e.g.

```
$ terraform import aws_neptune_subnet_group.default production-subnet-group
```

» Resource: aws_neptune_cluster_parameter_group

Manages a Neptune Cluster Parameter Group

» Example Usage

```
resource "aws_neptune_cluster_parameter_group" "example" {
  family      = "neptune1"
  name        = "example"
  description = "neptune cluster parameter group"

  parameter {
    name  = "neptune_enable_audit_log"
    value = 1
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the neptune cluster parameter group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `family` - (Required) The family of the neptune cluster parameter group.
- `description` - (Optional) The description of the neptune cluster parameter group. Defaults to ”Managed by Terraform”.
- `parameter` - (Optional) A list of neptune parameters to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Parameter blocks support the following:

- `name` - (Required) The name of the neptune parameter.
- `value` - (Required) The value of the neptune parameter.
- `apply_method` - (Optional) Valid values are `immediate` and `pending-reboot`. Defaults to `pending-reboot`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The neptune cluster parameter group name.
- `arn` - The ARN of the neptune cluster parameter group.

» Import

Neptune Cluster Parameter Groups can be imported using the `name`, e.g.

```
$ terraform import aws_neptune_cluster_parameter_group.cluster_pg production-pg-1
```

» Resource: aws_neptune_cluster

Provides an Neptune Cluster Resource. A Cluster Resource defines attributes that are applied to the entire cluster of Neptune Cluster Instances.

Changes to a Neptune Cluster can occur when you manually change a parameter, such as `backup_retention_period`, and are reflected in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because a modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately (see documentation below).

» Example Usage

```
resource "aws_neptune_cluster" "default" {
  cluster_identifier                  = "neptune-cluster-demo"
  engine                              = "neptune"
  backup_retention_period             = 5
  preferred_backup_window             = "07:00-09:00"
  skip_final_snapshot                 = true
  iam_database_authentication_enabled = true
  apply_immediately                   = true
}
```

Note: AWS Neptune does not support user name/password–based access control. See the AWS Docs for more information.

» Argument Reference

The following arguments are supported:

- `apply_immediately` - (Optional) Specifies whether any cluster modifications are applied immediately, or during the next maintenance window. Default is `false`.
- `availability_zones` - (Optional) A list of EC2 Availability Zones that instances in the Neptune cluster can be created in.
- `backup_retention_period` - (Optional) The days to retain backups for. Default `1`

- `cluster_identifier` - (Optional, Forces new resources) The cluster identifier. If omitted, Terraform will assign a random, unique identifier.
- `cluster_identifier_prefix` - (Optional, Forces new resource) Creates a unique cluster identifier beginning with the specified prefix. Conflicts with `cluster_identifier`.
- `engine` - (Optional) The name of the database engine to be used for this Neptune cluster. Defaults to `neptune`.
- `engine_version` - (Optional) The database engine version.
- `final_snapshot_identifier` - (Optional) The name of your final Neptune snapshot when this Neptune cluster is deleted. If omitted, no final snapshot will be made.
- `iam_roles` - (Optional) A List of ARNs for the IAM roles to associate to the Neptune Cluster.
- `iam_database_authentication_enabled` - (Optional) Specifies whether or mappings of AWS Identity and Access Management (IAM) accounts to database accounts is enabled.
- `kms_key_arn` - (Optional) The ARN for the KMS encryption key. When specifying `kms_key_arn`, `storage_encrypted` needs to be set to true.
- `neptune_subnet_group_name` - (Optional) A Neptune subnet group to associate with this Neptune instance.
- `neptune_cluster_parameter_group_name` - (Optional) A cluster parameter group to associate with the cluster.
- `preferred_backup_window` - (Optional) The daily time range during which automated backups are created if automated backups are enabled using the BackupRetentionPeriod parameter. Time in UTC. Default: A 30-minute window selected at random from an 8-hour block of time per region. e.g. 04:00-09:00
- `preferred_maintenance_window` - (Optional) The weekly time range during which system maintenance can occur, in (UTC) e.g. wed:04:00-wed:04:30
- `port` - (Optional) The port on which the Neptune accepts connections. Default is `8182`.
- `replication_source_identifier` - (Optional) ARN of a source Neptune cluster or Neptune instance if this Neptune cluster is to be created as a Read Replica.
- `skip_final_snapshot` - (Optional) Determines whether a final Neptune snapshot is created before the Neptune cluster is deleted. If true is specified, no Neptune snapshot is created. If false is specified, a Neptune snapshot is created before the Neptune cluster is deleted, using the value from `final_snapshot_identifier`. Default is `false`.
- `snapshot_identifier` - (Optional) Specifies whether or not to create this cluster from a snapshot. You can use either the name or ARN when specifying a Neptune cluster snapshot, or the ARN when specifying a Neptune snapshot.
- `storage_encrypted` - (Optional) Specifies whether the Neptune cluster is encrypted. The default is `false` if not specified.

- `tags` - (Optional) A mapping of tags to assign to the Neptune cluster.
- `vpc_security_group_ids` - (Optional) List of VPC security groups to associate with the Cluster

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Neptune Cluster Amazon Resource Name (ARN)
- `cluster_resource_id` - The Neptune Cluster Resource ID
- `cluster_members` – List of Neptune Instances that are a part of this cluster
- `endpoint` - The DNS address of the Neptune instance
- `hosted_zone_id` - The Route53 Hosted Zone ID of the endpoint
- `id` - The Neptune Cluster Identifier
- `reader_endpoint` - A read-only endpoint for the Neptune cluster, automatically load-balanced across replicas
- `status` - The Neptune instance status

» Timeouts

`aws_neptune_cluster` provides the following Timeouts configuration options:

- `create` - (Default `120 minutes`) Used for Cluster creation
- `update` - (Default `120 minutes`) Used for Cluster modifications
- `delete` - (Default `120 minutes`) Used for destroying cluster. This includes any cleanup task during the destroying process.

» Import

`aws_neptune_cluster` can be imported by using the cluster identifier, e.g.

```
$ terraform import aws_neptune_cluster.example my-cluster
```

» Resource: aws_neptune_cluster_instance

A Cluster Instance Resource defines attributes that are specific to a single instance in a Neptune Cluster.

You can simply add neptune instances and Neptune manages the replication. You can use the count meta-parameter to make multiple instances and join them all to the same Neptune Cluster, or you may specify different Cluster Instance resources with various `instance_class` sizes.

» Example Usage

The following example will create a neptune cluster with two neptune instances(one writer and one reader).

```
resource "aws_neptune_cluster" "default" {
  cluster_identifier                  = "neptune-cluster-demo"
  engine                              = "neptune"
  backup_retention_period             = 5
  preferred_backup_window             = "07:00-09:00"
  skip_final_snapshot                 = true
  iam_database_authentication_enabled = true
  apply_immediately                   = true
}

resource "aws_neptune_cluster_instance" "example" {
  count              = 2
  cluster_identifier = "${aws_neptune_cluster.default.id}"
  engine             = "neptune"
  instance_class     = "db.r4.large"
  apply_immediately  = true
}
```

» Argument Reference

The following arguments are supported:

- `apply_immediately` - (Optional) Specifies whether any instance modifications are applied immediately, or during the next maintenance window. Default is`false`.
- `auto_minor_version_upgrade` - (Optional) Indicates that minor engine upgrades will be applied automatically to the instance during the maintenance window. Default is `true`.
- `availability_zone` - (Optional) The EC2 Availability Zone that the neptune instance is created in.
- `cluster_identifier` - (Required) The identifier of the `aws_neptune_cluster` in which to launch this instance.
- `engine` - (Optional) The name of the database engine to be used for the neptune instance. Defaults to `neptune`. Valid Values: `neptune`.
- `engine_version` - (Optional) The neptune engine version.
- `identifier` - (Optional, Forces new resource) The indentifier for the neptune instance, if omitted, Terraform will assign a random, unique identifier.
- `identifier_prefix` - (Optional, Forces new resource) Creates a unique identifier beginning with the specified prefix. Conflicts with `identifier`.
- `instance_class` - (Required) The instance class to use.

- `neptune_subnet_group_name` - (Required if `publicly_accessible = false`, Optional otherwise) A subnet group to associate with this neptune instance. **NOTE:** This must match the `neptune_subnet_group_name` of the attached `aws_neptune_cluster`.
- `neptune_parameter_group_name` - (Optional) The name of the neptune parameter group to associate with this instance.
- `port` - (Optional) The port on which the DB accepts connections. Defaults to `8182`.
- `preferred_backup_window` - (Optional) The daily time range during which automated backups are created if automated backups are enabled. Eg: "04:00-09:00"
- `preferred_maintenance_window` - (Optional) The window to perform maintenance in. Syntax: "ddd:hh24:mi-ddd:hh24:mi". Eg: "Mon:00:00-Mon:03:00".
- `promotion_tier` - (Optional) Default 0. Failover Priority setting on instance level. The reader who has lower tier has higher priority to get promoter to writer.
- `publicly_accessible` - (Optional) Bool to control if instance is publicly accessible. Default is `false`.
- `tags` - (Optional) A mapping of tags to assign to the instance.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `address` - The hostname of the instance. See also `endpoint` and `port`.
- `arn` - Amazon Resource Name (ARN) of neptune instance
- `dbi_resource_id` - The region-unique, immutable identifier for the neptune instance.
- `endpoint` - The connection endpoint in `address:port` format.
- `id` - The Instance identifier
- `kms_key_arn` - The ARN for the KMS encryption key if one is set to the neptune cluster.
- `storage_encrypted` - Specifies whether the neptune cluster is encrypted.
- `writer` – Boolean indicating if this instance is writable. `False` indicates this instance is a read replica.

» Timeouts

`aws_neptune_cluster_instance` provides the following Timeouts configuration options:

- `create` - (Default `90 minutes`) How long to wait for creating instances to become available.

- `update` - (Default `90 minutes`) How long to wait for updating instances to complete updates.
- `delete` - (Default `90 minutes`) How long to wait for deleting instances to become fully deleted.

» Import

`aws_neptune_cluster_instance` can be imported by using the instance identifier, e.g.

```
$ terraform import aws_neptune_cluster_instance.example my-instance
```

» Resource: aws_neptune_cluster_snapshot

Manages a Neptune database cluster snapshot.

» Example Usage

```
resource "aws_neptune_cluster_snapshot" "example" {
  db_cluster_identifier          = "${aws_neptune_cluster.example.id}"
  db_cluster_snapshot_identifier = "resourcetestsnapshot1234"
}
```

» Argument Reference

The following arguments are supported:

- `db_cluster_identifier` - (Required) The DB Cluster Identifier from which to take the snapshot.
- `db_cluster_snapshot_identifier` - (Required) The Identifier for the snapshot.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `allocated_storage` - Specifies the allocated storage size in gigabytes (GB).
- `availability_zones` - List of EC2 Availability Zones that instances in the DB cluster snapshot can be restored in.
- `db_cluster_snapshot_arn` - The Amazon Resource Name (ARN) for the DB Cluster Snapshot.

- `engine` - Specifies the name of the database engine.
- `engine_version` - Version of the database engine for this DB cluster snapshot.
- `kms_key_id` - If storage_encrypted is true, the AWS KMS key identifier for the encrypted DB cluster snapshot.
- `license_model` - License model information for the restored DB cluster.
- `port` - Port that the DB cluster was listening on at the time of the snapshot.
- `source_db_cluster_snapshot_identifier` - The DB Cluster Snapshot Arn that the DB Cluster Snapshot was copied from. It only has value in case of cross customer or cross region copy.
- `storage_encrypted` - Specifies whether the DB cluster snapshot is encrypted.
- `status` - The status of this DB Cluster Snapshot.
- `vpc_id` - The VPC ID associated with the DB cluster snapshot.

» Timeouts

`aws_neptune_cluster_snapshot` provides the following Timeouts configuration options:

- `create` - (Default `20m`) How long to wait for the snapshot to be available.

» Import

`aws_neptune_cluster_snapshot` can be imported by using the cluster snapshot identifier, e.g.

```
$ terraform import aws_neptune_cluster_snapshot.example my-cluster-snapshot
```

» Resource: aws_neptune_event_subscription

» Example Usage

```
resource "aws_neptune_cluster" "default" {
  cluster_identifier                  = "neptune-cluster-demo"
  engine                              = "neptune"
  backup_retention_period             = 5
  preferred_backup_window             = "07:00-09:00"
  skip_final_snapshot                 = true
  iam_database_authentication_enabled = "true"
  apply_immediately                   = "true"
}
```

```
resource "aws_neptune_cluster_instance" "example" {
  count              = 1
  cluster_identifier = "${aws_neptune_cluster.default.id}"
  engine             = "neptune"
  instance_class     = "db.r4.large"
  apply_immediately  = "true"
}

resource "aws_sns_topic" "default" {
  name = "neptune-events"
}

resource "aws_neptune_event_subscription" "default" {
  name          = "neptune-event-sub"
  sns_topic_arn = "${aws_sns_topic.default.arn}"

  source_type = "db-instance"
  source_ids  = ["${aws_neptune_cluster_instance.example.id}"]

  event_categories = [
    "maintenance",
    "availability",
    "creation",
    "backup",
    "restoration",
    "recovery",
    "deletion",
    "failover",
    "failure",
    "notification",
    "configuration change",
    "read replica",
  ]

  tags = {
    "env" = "test"
  }
}
```

» Argument Reference

The following arguments are supported:

- `enabled` - (Optional) A boolean flag to enable/disable the subscription.

Defaults to true.

- `event_categories` - (Optional) A list of event categories for a `source_type` that you want to subscribe to. Run `aws neptune describe-event-categories` to find all the event categories.
- `name` - (Optional) The name of the Neptune event subscription. By default generated by Terraform.
- `name_prefix` - (Optional) The name of the Neptune event subscription. Conflicts with `name`.
- `sns_topic_arn` - (Required) The ARN of the SNS topic to send events to.
- `source_ids` - (Optional) A list of identifiers of the event sources for which events will be returned. If not specified, then all sources are included in the response. If specified, a `source_type` must also be specified.
- `source_type` - (Optional) The type of source that will be generating the events. Valid options are `db-instance`, `db-security-group`, `db-parameter-group`, `db-snapshot`, `db-cluster` or `db-cluster-snapshot`. If not set, all sources will be subscribed to.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes

The following additional atttributes are provided:

- `id` - The name of the Neptune event notification subscription.
- `arn` - The Amazon Resource Name of the Neptune event notification subscription.
- `customer_aws_id` - The AWS customer account associated with the Neptune event notification subscription.

» Timeouts

`aws_neptune_event_subscription` provides the following Timeouts configuration options:

- `create` - (Default `40m`) How long to wait for creating event subscription to become available.
- `delete` - (Default `40m`) How long to wait for deleting event subscription to become fully deleted.
- `update` - (Default `40m`) How long to wait for updating event subscription to complete updates.

» Import

`aws_neptune_event_subscription` can be imported by using the event subscription name, e.g.

```
$ terraform import aws_neptune_event_subscription.example my-event-subscription
```

» Resource: aws_opsworks_application

Provides an OpsWorks application resource.

» Example Usage

```
resource "aws_opsworks_application" "foo-app" {
  name        = "foobar application"
  short_name  = "foobar"
  stack_id    = "${aws_opsworks_stack.main.id}"
  type        = "rails"
  description = "This is a Rails application"

  domains = [
    "example.com",
    "sub.example.com",
  ]

  environment {
    key    = "key"
    value  = "value"
    secure = false
  }

  app_source {
    type     = "git"
    revision = "master"
    url      = "https://github.com/example.git"
  }

  enable_ssl = true

  ssl_configuration {
    private_key = "${file("./foobar.key")}"
    certificate = "${file("./foobar.crt")}"
  }
  document_root         = "public"
  auto_bundle_on_deploy = true
  rails_env             = "staging"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A human-readable name for the application.
- `short_name` - (Required) A short, machine-readable name for the application. This can only be defined on resource creation and ignored on resource update.
- `stack_id` - (Required) The id of the stack the application will belong to.
- `type` - (Required) Opsworks application type. One of `aws-flow-ruby`, `java`, `rails`, `php`, `nodejs`, `static` or `other`.
- `description` - (Optional) A description of the app.
- `environment` - (Optional) Object to define environment variables. Object is described below.
- `enable_ssl` - (Optional) Whether to enable SSL for the app. This must be set in order to let `ssl_configuration.private_key`, `ssl_configuration.certificate` and `ssl_configuration.chain` take effect.
- `ssl_configuration` - (Optional) The SSL configuration of the app. Object is described below.
- `app_source` - (Optional) SCM configuration of the app as described below.
- `data_source_arn` - (Optional) The data source's ARN.
- `data_source_type` - (Optional) The data source's type one of `AutoSelectOpsworksMysqlInstance`, `OpsworksMysqlInstance`, or `RdsDbInstance`.
- `data_source_database_name` - (Optional) The database name.
- `domains` - (Optional) A list of virtual host alias.
- `document_root` - (Optional) Subfolder for the document root for application of type `rails`.
- `auto_bundle_on_deploy` - (Optional) Run bundle install when deploying for application of type `rails`.
- `rails_env` - (Required if `type` = `rails`) The name of the Rails environment for application of type `rails`.
- `aws_flow_ruby_settings -` (Optional) Specify activity and workflow workers for your app using the aws-flow gem.

An `app_source` block supports the following arguments (can only be defined once per resource):

- `type` - (Required) The type of source to use. For example, ”archive”.
- `url` - (Required) The URL where the app resource can be found.

- `username` - (Optional) Username to use when authenticating to the source.
- `password` - (Optional) Password to use when authenticating to the source. Terraform cannot perform drift detection of this configuration.
- `ssh_key` - (Optional) SSH key to use when authenticating to the source. Terraform cannot perform drift detection of this configuration.
- `revision` - (Optional) For sources that are version-aware, the revision to use.

An `environment` block supports the following arguments:

- `key` - (Required) Variable name.
- `value` - (Required) Variable value.
- `secure` - (Optional) Set visibility of the variable value to `true` or `false`.

A `ssl_configuration` block supports the following arguments (can only be defined once per resource):

- `private_key` - (Required) The private key; the contents of the certificate's domain.key file.
- `certificate` - (Required) The contents of the certificate's domain.crt file.
- `chain` - (Optional) Can be used to specify an intermediate certificate authority key or client authentication.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the application.

» Resource: aws_opsworks_custom_layer

Provides an OpsWorks custom layer resource.

» Example Usage

```
resource "aws_opsworks_custom_layer" "custlayer" {
  name       = "My Awesome Custom Layer"
  short_name = "awesome"
  stack_id   = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A human-readable name for the layer.
- `short_name` - (Required) A short, machine-readable name for the layer, which will be used to identify it in the Chef node JSON.
- `stack_id` - (Required) The id of the stack the layer will belong to.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- `custom_json` - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- `mount_point` - (Required) The path to mount the EBS volume on the layer's instances.
- `size` - (Required) The size of the volume in gigabytes.
- `number_of_disks` - (Required) The number of disks to use for the EBS volume.

- `raid_level` - (Required) The RAID level to use for the volume.
- `type` - (Optional) The type of volume to create. This may be `standard` (the default), `io1` or `gp2`.
- `iops` - (Optional) For PIOPS volumes, the IOPS per disk.
- `encrypted` - (Optional) Encrypt the volume.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Import

OpsWorks Custom Layers can be imported using the `id`, e.g.

```
$ terraform import aws_opsworks_custom_layer.bar 00000000-0000-0000-0000-000000000000
```

» Resource: aws_opsworks_ganglia_layer

Provides an OpsWorks Ganglia layer resource.

» Example Usage

```
resource "aws_opsworks_ganglia_layer" "monitor" {
  stack_id = "${aws_opsworks_stack.main.id}"
  password = "foobarbaz"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `password` - (Required) The password to use for Ganglia.
- `name` - (Optional) A human-readable name for the layer.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.

- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `url` - (Optional) The URL path to use for Ganglia. Defaults to "/ganglia".
- `username` - (Optiona) The username to use for Ganglia. Defaults to "opsworks".
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- `custom_json` - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- `mount_point` - (Required) The path to mount the EBS volume on the layer's instances.
- `size` - (Required) The size of the volume in gigabytes.
- `number_of_disks` - (Required) The number of disks to use for the EBS volume.
- `raid_level` - (Required) The RAID level to use for the volume.
- `type` - (Optional) The type of volume to create. This may be `standard` (the default), `io1` or `gp2`.
- `iops` - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Resource: aws_opsworks_haproxy_layer

Provides an OpsWorks haproxy layer resource.

» Example Usage

```
resource "aws_opsworks_haproxy_layer" "lb" {
  stack_id       = "${aws_opsworks_stack.main.id}"
  stats_password = "foobarbaz"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `stats_password` - (Required) The password to use for HAProxy stats.
- `name` - (Optional) A human-readable name for the layer.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `healthcheck_method` - (Optional) HTTP method to use for instance healthchecks. Defaults to ”OPTIONS”.
- `healthcheck_url` - (Optional) URL path to use for instance healthchecks. Defaults to ”/”.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.

- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `stats_enabled` - (Optional) Whether to enable HAProxy stats.
- `stats_url` - (Optional) The HAProxy stats URL. Defaults to "/haproxy?stats".
- `stats_user` - (Optional) The username for HAProxy stats. Defaults to "opsworks".
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- `custom_json` - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- `mount_point` - (Required) The path to mount the EBS volume on the layer's instances.
- `size` - (Required) The size of the volume in gigabytes.
- `number_of_disks` - (Required) The number of disks to use for the EBS volume.
- `raid_level` - (Required) The RAID level to use for the volume.
- `type` - (Optional) The type of volume to create. This may be `standard` (the default), `io1` or `gp2`.
- `iops` - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Resource: aws_opsworks_instance

Provides an OpsWorks instance resource.

» Example Usage

```
resource "aws_opsworks_instance" "my-instance" {
  stack_id = "${aws_opsworks_stack.main.id}"

  layer_ids = [
    "${aws_opsworks_custom_layer.my-layer.id}",
  ]

  instance_type = "t2.micro"
  os            = "Amazon Linux 2015.09"
  state         = "stopped"
}
```

» Argument Reference

The following arguments are supported:

- `instance_type` - (Required) The type of instance to start
- `stack_id` - (Required) The id of the stack the instance will belong to.
- `layer_ids` - (Required) The ids of the layers the instance will belong to.
- `state` - (Optional) The desired state of the instance. Can be either `"running"` or `"stopped"`.
- `install_updates_on_boot` - (Optional) Controls where to install OS and package updates when the instance boots. Defaults to `true`.
- `auto_scaling_type` - (Optional) Creates load-based or time-based instances. If set, can be either: `"load"` or `"timer"`.
- `availability_zone` - (Optional) Name of the availability zone where instances will be created by default.
- `ebs_optimized` - (Optional) If true, the launched EC2 instance will be EBS-optimized.
- `hostname` - (Optional) The instance's host name.
- `architecture` - (Optional) Machine architecture for created instances. Can be either `"x86_64"` (the default) or `"i386"`
- `ami_id` - (Optional) The AMI to use for the instance. If an AMI is specified, `os` must be `"Custom"`.
- `os` - (Optional) Name of operating system that will be installed.
- `root_device_type` - (Optional) Name of the type of root device instances will have by default. Can be either `"ebs"` or `"instance-store"`

- ssh_key_name - (Optional) Name of the SSH keypair that instances will have by default.
- agent_version - (Optional) The AWS OpsWorks agent to install. Defaults to "INHERIT".
- subnet_id - (Optional) Subnet ID to attach to
- tenancy - (Optional) Instance tenancy to use. Can be one of "default", "dedicated" or "host"
- virtualization_type - (Optional) Keyword to choose what virtualization mode created instances will use. Can be either "paravirtual" or "hvm".
- root_block_device - (Optional) Customize details about the root block device of the instance. See Block Devices below for details.
- ebs_block_device - (Optional) Additional EBS block devices to attach to the instance. See Block Devices below for details.
- ephemeral_block_device - (Optional) Customize Ephemeral (also known as ”Instance Store”) volumes on the instance. See Block Devices below for details.

» Block devices

Each of the *_block_device attributes controls a portion of the AWS Instance's ”Block Device Mapping”. It's a good idea to familiarize yourself with AWS's Block Device Mapping docs to understand the implications of using these attributes.

The root_block_device mapping supports the following:

- volume_type - (Optional) The type of volume. Can be "standard", "gp2", or "io1". (Default: "standard").
- volume_size - (Optional) The size of the volume in gigabytes.
- iops - (Optional) The amount of provisioned IOPS. This must be set with a volume_type of "io1".
- delete_on_termination - (Optional) Whether the volume should be destroyed on instance termination (Default: true).

Modifying any of the root_block_device settings requires resource replacement.

Each ebs_block_device supports the following:

- device_name - The name of the device to mount.
- snapshot_id - (Optional) The Snapshot ID to mount.
- volume_type - (Optional) The type of volume. Can be "standard", "gp2", or "io1". (Default: "standard").
- volume_size - (Optional) The size of the volume in gigabytes.
- iops - (Optional) The amount of provisioned IOPS. This must be set with a volume_type of "io1".

- `delete_on_termination` - (Optional) Whether the volume should be destroyed on instance termination (Default: `true`).

Modifying any `ebs_block_device` currently requires resource replacement.

Each `ephemeral_block_device` supports the following:

- `device_name` - The name of the block device to mount on the instance.
- `virtual_name` - The Instance Store Device Name (e.g. `"ephemeral0"`)

Each AWS Instance type has a different set of Instance Store block devices available for attachment. AWS publishes a list of which ephemeral devices are available on each type. The devices are always identified by the `virtual_name` in the format `"ephemeral{0..N}"`.

NOTE: Currently, changes to `*_block_device` configuration of *existing* resources cannot be automatically detected by Terraform. After making updates to block device configuration, resource recreation can be manually triggered by using the `taint` command.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the OpsWorks instance.
- `agent_version` - The AWS OpsWorks agent version.
- `availability_zone` - The availability zone of the instance.
- `ec2_instance_id` - EC2 instance ID
- `ssh_key_name` - The key name of the instance
- `public_dns` - The public DNS name assigned to the instance. For EC2-VPC, this is only available if you've enabled DNS hostnames for your VPC
- `public_ip` - The public IP address assigned to the instance, if applicable.
- `private_dns` - The private DNS name assigned to the instance. Can only be used inside the Amazon EC2, and only available if you've enabled DNS hostnames for your VPC
- `private_ip` - The private IP address assigned to the instance
- `subnet_id` - The VPC subnet ID.
- `tenancy` - The Instance tenancy
- `security_group_ids` - The associated security groups.

» Timeouts

`aws_opsworks_instance` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used when the instance is created. It should cover the time needed for the instance to start successfully.
- `delete` - (Default `10 minutes`) Used when the instance is deleted. It should cover the time needed for the instance to stop successfully.
- `update` - (Default `10 minutes`) Used when the instance is changed. It should cover the time needed to either start or stop the instance.

» Import

Opsworks Instances can be imported using the `instance id`, e.g.

```
$ terraform import aws_opsworks_instance.my_instance 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

» Resource: aws_opsworks_java_app_layer

Provides an OpsWorks Java application layer resource.

» Example Usage

```
resource "aws_opsworks_java_app_layer" "app" {
  stack_id = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `name` - (Optional) A human-readable name for the layer.
- `app_server` - (Optional) Keyword for the application container to use. Defaults to ”tomcat”.
- `app_server_version` - (Optional) Version of the selected application container to use. Defaults to ”7”.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.

- auto_healing - (Optional) Whether to enable auto-healing for the layer.
- install_updates_on_boot - (Optional) Whether to install OS and package updates on each instance when it boots.
- instance_shutdown_timeout - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- jvm_type - (Optional) Keyword for the type of JVM to use. Defaults to openjdk.
- jvm_options - (Optional) Options to set for the JVM.
- jvm_version - (Optional) Version of JVM to use. Defaults to "7".
- elastic_load_balancer - (Optional) Name of an Elastic Load Balancer to attach to this layer
- drain_elb_on_shutdown - (Optional) Whether to enable Elastic Load Balancing connection draining.
- system_packages - (Optional) Names of a set of system packages to install on the layer's instances.
- use_ebs_optimized_instances - (Optional) Whether to use EBS-optimized instances.
- ebs_volume - (Optional) ebs_volume blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- custom_json - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- custom_configure_recipes
- custom_deploy_recipes
- custom_setup_recipes
- custom_shutdown_recipes
- custom_undeploy_recipes

An ebs_volume block supports the following arguments:

- mount_point - (Required) The path to mount the EBS volume on the layer's instances.
- size - (Required) The size of the volume in gigabytes.
- number_of_disks - (Required) The number of disks to use for the EBS volume.
- raid_level - (Required) The RAID level to use for the volume.
- type - (Optional) The type of volume to create. This may be standard (the default), io1 or gp2.
- iops - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Resource: aws_opsworks_memcached_layer

Provides an OpsWorks memcached layer resource.

» Example Usage

```
resource "aws_opsworks_memcached_layer" "cache" {
  stack_id = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `name` - (Optional) A human-readable name for the layer.
- `allocated_memory` - (Optional) Amount of memory to allocate for the cache on each instance, in megabytes. Defaults to 512MB.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.

- ebs_volume - (Optional) ebs_volume blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- custom_json - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- custom_configure_recipes
- custom_deploy_recipes
- custom_setup_recipes
- custom_shutdown_recipes
- custom_undeploy_recipes

An ebs_volume block supports the following arguments:

- mount_point - (Required) The path to mount the EBS volume on the layer's instances.
- size - (Required) The size of the volume in gigabytes.
- number_of_disks - (Required) The number of disks to use for the EBS volume.
- raid_level - (Required) The RAID level to use for the volume.
- type - (Optional) The type of volume to create. This may be standard (the default), io1 or gp2.
- iops - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The id of the layer.

» Resource: aws_opsworks_mysql_layer

Provides an OpsWorks MySQL layer resource.

Note: All arguments including the root password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_opsworks_mysql_layer" "db" {
  stack_id = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `name` - (Optional) A human-readable name for the layer.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `root_password` - (Optional) Root password to use for MySQL.
- `root_password_on_all_instances` - (Optional) Whether to set the root user password to all instances in the stack so they can access the instances in this layer.
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- `custom_json` - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- `mount_point` - (Required) The path to mount the EBS volume on the layer's instances.
- `size` - (Required) The size of the volume in gigabytes.
- `number_of_disks` - (Required) The number of disks to use for the EBS volume.
- `raid_level` - (Required) The RAID level to use for the volume.
- `type` - (Optional) The type of volume to create. This may be `standard` (the default), `io1` or `gp2`.
- `iops` - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Resource: aws_opsworks_nodejs_app_layer

Provides an OpsWorks NodeJS application layer resource.

» Example Usage

```
resource "aws_opsworks_nodejs_app_layer" "app" {
  stack_id = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `name` - (Optional) A human-readable name for the layer.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.

- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `nodejs_version` - (Optional) The version of NodeJS to use. Defaults to "0.10.38".
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- `custom_json` - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- `mount_point` - (Required) The path to mount the EBS volume on the layer's instances.
- `size` - (Required) The size of the volume in gigabytes.
- `number_of_disks` - (Required) The number of disks to use for the EBS volume.
- `raid_level` - (Required) The RAID level to use for the volume.
- `type` - (Optional) The type of volume to create. This may be `standard` (the default), `io1` or `gp2`.
- `iops` - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Resource: aws_opsworks_permission

Provides an OpsWorks permission resource.

» Example Usage

```
resource "aws_opsworks_permission" "my_stack_permission" {
  allow_ssh  = true
  allow_sudo = true
  level      = "iam_only"
  user_arn   = "${aws_iam_user.user.arn}"
  stack_id   = "${aws_opsworks_stack.stack.id}"
}
```

» Argument Reference

The following arguments are supported:

- `allow_ssh` - (Optional) Whether the user is allowed to use SSH to communicate with the instance
- `allow_sudo` - (Optional) Whether the user is allowed to use sudo to elevate privileges
- `user_arn` - (Required) The user's IAM ARN to set permissions for
- `level` - (Optional) The users permission level. Mus be one of `deny`, `show`, `deploy`, `manage`, `iam_only`
- `stack_id` - (Required) The stack to set the permissions for

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The computed id of the permission. Please note that this is only used internally to identify the permission. This value is not used in aws.

» Resource: aws_opsworks_php_app_layer

Provides an OpsWorks PHP application layer resource.

» Example Usage

```
resource "aws_opsworks_php_app_layer" "app" {
  stack_id = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `name` - (Optional) A human-readable name for the layer.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- `custom_json` - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- mount_point - (Required) The path to mount the EBS volume on the layer's instances.
- size - (Required) The size of the volume in gigabytes.
- number_of_disks - (Required) The number of disks to use for the EBS volume.
- raid_level - (Required) The RAID level to use for the volume.
- type - (Optional) The type of volume to create. This may be standard (the default), io1 or gp2.
- iops - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The id of the layer.

» Resource: aws_opsworks_rails_app_layer

Provides an OpsWorks Ruby on Rails application layer resource.

» Example Usage

```
resource "aws_opsworks_rails_app_layer" "app" {
  stack_id = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- stack_id - (Required) The id of the stack the layer will belong to.
- name - (Optional) A human-readable name for the layer.
- app_server - (Optional) Keyword for the app server to use. Defaults to "apache_passenger".
- auto_assign_elastic_ips - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- auto_assign_public_ips - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- bundler_version - (Optional) When OpsWorks is managing Bundler, which version to use. Defaults to "1.5.3".

- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer
- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `manage_bundler` - (Optional) Whether OpsWorks should manage bundler. On by default.
- `passenger_version` - (Optional) The version of Passenger to use. Defaults to "4.0.46".
- `ruby_version` - (Optional) The version of Ruby to use. Defaults to "2.0.0".
- `rubygems_version` - (Optional) The version of RubyGems to use. Defaults to "2.2.2".
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.
- `custom_json` - (Optional) Custom JSON attributes to apply to the layer.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- `mount_point` - (Required) The path to mount the EBS volume on the layer's instances.
- `size` - (Required) The size of the volume in gigabytes.
- `number_of_disks` - (Required) The number of disks to use for the EBS volume.
- `raid_level` - (Required) The RAID level to use for the volume.

- `type` - (Optional) The type of volume to create. This may be `standard` (the default), `io1` or `gp2`.
- `iops` - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Resource: aws_opsworks_rds_db_instance

Provides an OpsWorks RDS DB Instance resource.

Note: All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_opsworks_rds_db_instance" "my_instance" {
  stack_id            = "${aws_opsworks_stack.my_stack.id}"
  rds_db_instance_arn = "${aws_db_instance.my_instance.arn}"
  db_user             = "someUser"
  db_password         = "somePass"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The stack to register a db instance for. Changing this will force a new resource.
- `rds_db_instance_arn` - (Required) The db instance to register for this stack. Changing this will force a new resource.
- `db_user` - (Required) A db username
- `db_password` - (Required) A db password

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The computed id. Please note that this is only used internally to identify the stack <-> instance relation. This value is not used in aws.

» Resource: aws_opsworks_stack

Provides an OpsWorks stack resource.

» Example Usage

```
resource "aws_opsworks_stack" "main" {
  name                         = "awesome-stack"
  region                       = "us-west-1"
  service_role_arn             = "${aws_iam_role.opsworks.arn}"
  default_instance_profile_arn = "${aws_iam_instance_profile.opsworks.arn}"

  tags = {
    Name = "foobar-terraform-stack"
  }

  custom_json = <<EOT
{
 "foobar": {
    "version": "1.0.0"
  }
}
EOT
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the stack.
- `region` - (Required) The name of the region where the stack will exist.
- `service_role_arn` - (Required) The ARN of an IAM role that the OpsWorks service will act as.
- `default_instance_profile_arn` - (Required) The ARN of an IAM Instance Profile that created instances will have by default.
- `agent_version` - (Optional) If set to `"LATEST"`, OpsWorks will automatically install the latest version.
- `berkshelf_version` - (Optional) If `manage_berkshelf` is enabled, the version of Berkshelf to use.
- `color` - (Optional) Color to paint next to the stack's resources in the OpsWorks console.
- `default_availability_zone` - (Optional) Name of the availability zone where instances will be created by default. This is required unless you set `vpc_id`.

- `configuration_manager_name` - (Optional) Name of the configuration manager to use. Defaults to "Chef".
- `configuration_manager_version` - (Optional) Version of the configuration manager to use. Defaults to "11.4".
- `custom_cookbooks_source` - (Optional) When `use_custom_cookbooks` is set, provide this sub-object as described below.
- `custom_json` - (Optional) User defined JSON passed to "Chef". Use a "here doc" for multiline JSON.
- `default_os` - (Optional) Name of OS that will be installed on instances by default.
- `default_root_device_type` - (Optional) Name of the type of root device instances will have by default.
- `default_ssh_key_name` - (Optional) Name of the SSH keypair that instances will have by default.
- `default_subnet_id` - (Optional) Id of the subnet in which instances will be created by default. Mandatory if `vpc_id` is set, and forbidden if it isn't.
- `hostname_theme` - (Optional) Keyword representing the naming scheme that will be used for instance hostnames within this stack.
- `manage_berkshelf` - (Optional) Boolean value controlling whether Opsworks will run Berkshelf for this stack.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `use_custom_cookbooks` - (Optional) Boolean value controlling whether the custom cookbook settings are enabled.
- `use_opsworks_security_groups` - (Optional) Boolean value controlling whether the standard OpsWorks security groups apply to created instances.
- `vpc_id` - (Optional) The id of the VPC that this stack belongs to.
- `custom_json` - (Optional) Custom JSON attributes to apply to the entire stack.

The `custom_cookbooks_source` block supports the following arguments:

- `type` - (Required) The type of source to use. For example, "archive".
- `url` - (Required) The URL where the cookbooks resource can be found.
- `username` - (Optional) Username to use when authenticating to the source.
- `password` - (Optional) Password to use when authenticating to the source. Terraform cannot perform drift detection of this configuration.
- `ssh_key` - (Optional) SSH key to use when authenticating to the source. Terraform cannot perform drift detection of this configuration.
- `revision` - (Optional) For sources that are version-aware, the revision to use.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the stack.

» Import

OpsWorks stacks can be imported using the `id`, e.g.

```
$ terraform import aws_opsworks_stack.bar 00000000-0000-0000-0000-000000000000
```

» Resource: aws_opsworks_static_web_layer

Provides an OpsWorks static web server layer resource.

» Example Usage

```
resource "aws_opsworks_static_web_layer" "web" {
  stack_id = "${aws_opsworks_stack.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `stack_id` - (Required) The id of the stack the layer will belong to.
- `name` - (Optional) A human-readable name for the layer.
- `auto_assign_elastic_ips` - (Optional) Whether to automatically assign an elastic IP address to the layer's instances.
- `auto_assign_public_ips` - (Optional) For stacks belonging to a VPC, whether to automatically assign a public IP address to each of the layer's instances.
- `custom_instance_profile_arn` - (Optional) The ARN of an IAM profile that will be used for the layer's instances.
- `custom_security_group_ids` - (Optional) Ids for a set of security groups to apply to the layer's instances.
- `auto_healing` - (Optional) Whether to enable auto-healing for the layer.
- `install_updates_on_boot` - (Optional) Whether to install OS and package updates on each instance when it boots.
- `instance_shutdown_timeout` - (Optional) The time, in seconds, that OpsWorks will wait for Chef to complete after triggering the Shutdown event.
- `elastic_load_balancer` - (Optional) Name of an Elastic Load Balancer to attach to this layer

- `drain_elb_on_shutdown` - (Optional) Whether to enable Elastic Load Balancing connection draining.
- `system_packages` - (Optional) Names of a set of system packages to install on the layer's instances.
- `use_ebs_optimized_instances` - (Optional) Whether to use EBS-optimized instances.
- `ebs_volume` - (Optional) `ebs_volume` blocks, as described below, will each create an EBS volume and connect it to the layer's instances.

The following extra optional arguments, all lists of Chef recipe names, allow custom Chef recipes to be applied to layer instances at the five different lifecycle events, if custom cookbooks are enabled on the layer's stack:

- `custom_configure_recipes`
- `custom_deploy_recipes`
- `custom_setup_recipes`
- `custom_shutdown_recipes`
- `custom_undeploy_recipes`

An `ebs_volume` block supports the following arguments:

- `mount_point` - (Required) The path to mount the EBS volume on the layer's instances.
- `size` - (Required) The size of the volume in gigabytes.
- `number_of_disks` - (Required) The number of disks to use for the EBS volume.
- `raid_level` - (Required) The RAID level to use for the volume.
- `type` - (Optional) The type of volume to create. This may be `standard` (the default), `io1` or `gp2`.
- `iops` - (Optional) For PIOPS volumes, the IOPS per disk.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The id of the layer.

» Resource: aws_opsworks_user_profile

Provides an OpsWorks User Profile resource.

» Example Usage

```
resource "aws_opsworks_user_profile" "my_profile" {
  user_arn     = "${aws_iam_user.user.arn}"
  ssh_username = "my_user"
}
```

» Argument Reference

The following arguments are supported:

- `user_arn` - (Required) The user's IAM ARN
- `allow_self_management` - (Optional) Whether users can specify their own SSH public key through the My Settings page
- `ssh_username` - (Required) The ssh username, with witch this user wants to log in
- `ssh_public_key` - (Optional) The users public key

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Same value as `user_arn`

» Data Source: aws_organizations_organization

Get information about the organization that the user's account belongs to

» Example Usage

» List all account IDs for the organization

```
# Terraform 0.12 syntax
data "aws_organizations_organization" "example" {}

output "account_ids" {
  value = data.aws_organizations_organization.example.accounts[*].id
}
```

» SNS topic that can be interacted by the organization only

```
data "aws_organizations_organization" "example" {}

resource "aws_sns_topic" "sns_topic" {
  name = "my-sns-topic"
}
```

```
resource "aws_sns_topic_policy" "sns_topic_policy" {
  arn = "${aws_sns_topic.sns_topic.arn}"

  policy = "${data.aws_iam_policy_document.sns_topic_policy.json}"
}

data "aws_iam_policy_document" "sns_topic_policy" {
  statement {
    effect = "Allow"

    actions = [
      "SNS:Subscribe",
      "SNS:Publish",
    ]

    condition {
      test     = "StringEquals"
      variable = "aws:PrincipalOrgID"

      values = [
        "${data.aws_organizations_organization.example.id}",
      ]
    }

    principals {
      type        = "AWS"
      identifiers = ["*"]
    }

    resources = [
      "${aws_sns_topic.sns_topic.arn}",
    ]
  }
}
```

» Argument Reference

There are no arguments available for this data source.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the organization.
- `feature_set` - The FeatureSet of the organization.
- `id` - The ID of the organization.
- `master_account_arn` - The Amazon Resource Name (ARN) of the account that is designated as the master account for the organization.
- `master_account_email` - The email address that is associated with the AWS account that is designated as the master account for the organization.
- `master_account_id` - The unique identifier (ID) of the master account of an organization.

» Master Account Attributes Reference

If the account is the master account for the organization, the following attributes are also exported:

- `accounts` - List of organization accounts including the master account. For a list excluding the master account, see the `non_master_accounts` attribute. All elements have these attributes:
  - `arn` - ARN of the account
  - `email` - Email of the account
  - `id` - Identifier of the account
  - `name` - Name of the account
- `aws_service_access_principals` - A list of AWS service principal names that have integration enabled with your organization. Organization must have `feature_set` set to `ALL`. For additional information, see the AWS Organizations User Guide.
- `enabled_policy_types` - A list of Organizations policy types that are enabled in the Organization Root. Organization must have `feature_set` set to `ALL`. For additional information about valid policy types (e.g. `SERVICE_CONTROL_POLICY`), see the AWS Organizations API Reference.
- `non_master_accounts` - List of organization accounts excluding the master account. For a list including the master account, see the `accounts` attribute. All elements have these attributes:
  - `arn` - ARN of the account
  - `email` - Email of the account
  - `id` - Identifier of the account
  - `name` - Name of the account
- `roots` - List of organization roots. All elements have these attributes:
  - `arn` - ARN of the root
  - `id` - Identifier of the root
  - `name` - Name of the root
  - `policy_types` - List of policy types enabled for this root. All elements have these attributes:
  - `name` - The name of the policy type
  - `status` - The status of the policy type as it relates to the associated

root

» Data Source: aws_organizations_organizational_units

Get all direct child organizational units under a parent organizational unit. This only provides immediate children, not all children.

» Example Usage

```
data "aws_organizations_organization" "org" {}

data "aws_organizations_organizational_units" "ou" {
  parent_id = data.aws_organizations_organization.org.roots[0].id
}
```

» Argument Reference

- `parent_id` - (Required) The parent ID of the organizational unit.

» Attributes Reference

- `children` - List of child organizational units, which have the following attributes:
  - `arn` - ARN of the organizational unit
  - `name` - Name of the organizational unit
  - `id` - ID of the organizational unit

» Resource: aws_organizations_account

Provides a resource to create a member account in the current organization.

Note: Account management must be done from the organization's master account.

WARNING: Deleting this Terraform resource will only remove an AWS account from an organization. Terraform will not close the account. The member account must be prepared to be a standalone account beforehand. See the AWS Organizations documentation for more information.

» Example Usage:

```
resource "aws_organizations_account" "account" {
  name  = "my_new_account"
  email = "john@doe.org"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A friendly name for the member account.
- `email` - (Required) The email address of the owner to assign to the new member account. This email address must not already be associated with another AWS account.
- `iam_user_access_to_billing` - (Optional) If set to `ALLOW`, the new account enables IAM users to access account billing information if they have the required permissions. If set to `DENY`, then only the root user of the new account can access account billing information.
- `parent_id` - (Optional) Parent Organizational Unit ID or Root ID for the account. Defaults to the Organization default Root ID. A configuration must be present for this argument to perform drift detection.
- `role_name` - (Optional) The name of an IAM role that Organizations automatically preconfigures in the new member account. This role trusts the master account, allowing users in the master account to assume the role, as permitted by the master account administrator. The role has administrator permissions in the new member account. The Organizations API provides no method for reading this information after account creation, so Terraform cannot perform drift detection on its value and will always show a difference for a configured value after import unless `ignore_changes` is used.
- `tags` - (Optional) Key-value mapping of resource tags.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN for this account.
- `id` - The AWS account id

» Import

The AWS member account can be imported by using the `account_id`, e.g.

```
$ terraform import aws_organizations_account.my_org 111111111111
```

Certain resource arguments, like `role_name`, do not have an Organizations API method for reading the information after account creation. If the argument is set in the Terraform configuration on an imported resource, Terraform will always show a difference. To workaround this behavior, either omit the argument from the Terraform configuration or use `ignore_changes` to hide the difference, e.g.

```
resource "aws_organizations_account" "account" {
  name      = "my_new_account"
  email     = "john@doe.org"
  role_name = "myOrganizationRole"

  # There is no AWS Organizations API for reading role_name
  lifecycle {
    ignore_changes = ["role_name"]
  }
}
```

» Resource: aws_organizations_organization

Provides a resource to create an organization.

» Example Usage:

```
resource "aws_organizations_organization" "org" {
  aws_service_access_principals = [
    "cloudtrail.amazonaws.com",
    "config.amazonaws.com",
  ]

  feature_set = "ALL"
}
```

» Argument Reference

The following arguments are supported:

- `aws_service_access_principals` - (Optional) List of AWS service principal names for which you want to enable integration with your organization. This is typically in the form of a URL, such as service-abbreviation.amazonaws.com. Organization must have `feature_set` set to `ALL`. For additional information, see the AWS Organizations User Guide.

- `enabled_policy_types` - (Optional) List of Organizations policy types to enable in the Organization Root. Organization must have `feature_set` set to `ALL`. For additional information about valid policy types (e.g. `SERVICE_CONTROL_POLICY`), see the AWS Organizations API Reference.
- `feature_set` - (Optional) Specify "ALL" (default) or "CONSOLIDATED_BILLING".

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `accounts` - List of organization accounts including the master account. For a list excluding the master account, see the `non_master_accounts` attribute. All elements have these attributes:
  - `arn` - ARN of the account
  - `email` - Email of the account
  - `id` - Identifier of the account
  - `name` - Name of the account
- `arn` - ARN of the organization
- `id` - Identifier of the organization
- `master_account_arn` - ARN of the master account
- `master_account_email` - Email address of the master account
- `master_account_id` - Identifier of the master account
- `non_master_accounts` - List of organization accounts excluding the master account. For a list including the master account, see the `accounts` attribute. All elements have these attributes:
  - `arn` - ARN of the account
  - `email` - Email of the account
  - `id` - Identifier of the account
  - `name` - Name of the account
- `roots` - List of organization roots. All elements have these attributes:
  - `arn` - ARN of the root
  - `id` - Identifier of the root
  - `name` - Name of the root
  - `policy_types` - List of policy types enabled for this root. All elements have these attributes:
  - `name` - The name of the policy type
  - `status` - The status of the policy type as it relates to the associated root

» Import

The AWS organization can be imported by using the `id`, e.g.

```
$ terraform import aws_organizations_organization.my_org o-1234567
```

» Resource: aws_organizations_organizational_unit

Provides a resource to create an organizational unit.

» Example Usage

```
resource "aws_organizations_organizational_unit" "example" {
  name      = "example"
  parent_id = "${aws_organizations_organization.example.roots.0.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - The name for the organizational unit
- `parent_id` - ID of the parent organizational unit, which may be the root

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `accounts` - List of child accounts for this Organizational Unit. Does not return account information for child Organizational Units. All elements have these attributes:
  - `arn` - ARN of the account
  - `email` - Email of the account
  - `id` - Identifier of the account
  - `name` - Name of the account
- `arn` - ARN of the organizational unit
- `id` - Identifier of the organization unit

» Import

AWS Organizations Organizational Units can be imported by using the `id`, e.g.

```
$ terraform import aws_organizations_organizational_unit.example ou-1234567
```

» Resource: aws_organizations_policy

Provides a resource to manage an AWS Organizations policy.

» Example Usage

```
resource "aws_organizations_policy" "example" {
  name = "example"

  content = <<CONTENT
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "*",
    "Resource": "*"
  }
}
CONTENT
}
```

» Argument Reference

The following arguments are supported:

- `content` - (Required) The policy content to add to the new policy. For example, if you create a service control policy (SCP), this string must be JSON text that specifies the permissions that admins in attached accounts can delegate to their users, groups, and roles. For more information about the SCP syntax, see the Service Control Policy Syntax documentation.
- `name` - (Required) The friendly name to assign to the policy.
- `description` - (Optional) A description to assign to the policy.
- `type` - (Optional) The type of policy to create. Currently, the only valid value is `SERVICE_CONTROL_POLICY` (SCP).

» Attribute Reference

- `id` - The unique identifier (ID) of the policy.
- `arn` - Amazon Resource Name (ARN) of the policy.

» Import

`aws_organizations_policy` can be imported by using the policy ID, e.g.

```
$ terraform import aws_organizations_policy.example p-12345678
```

» Resource: aws_organizations_policy_attachment

Provides a resource to attach an AWS Organizations policy to an organization account, root, or unit.

» Example Usage

» Organization Account

```
resource "aws_organizations_policy_attachment" "account" {
  policy_id = "${aws_organizations_policy.example.id}"
  target_id = "123456789012"
}
```

» Organization Root

```
resource "aws_organizations_policy_attachment" "root" {
  policy_id = "${aws_organizations_policy.example.id}"
  target_id = "${aws_organizations_organization.example.roots.0.id}"
}
```

» Organization Unit

```
resource "aws_organizations_policy_attachment" "unit" {
  policy_id = "${aws_organizations_policy.example.id}"
  target_id = "${aws_organizations_organizational_unit.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `policy_id` - (Required) The unique identifier (ID) of the policy that you want to attach to the target.
- `target_id` - (Required) The unique identifier (ID) of the root, organizational unit, or account number that you want to attach the policy to.

» Import

`aws_organizations_policy_attachment` can be imported by using the target ID and policy ID, e.g. with an account target

```
$ terraform import aws_organizations_policy_attachment.account 123456789012:p-12345678
```

» Resource: aws_pinpoint_app

Provides a Pinpoint App resource.

» Example Usage

```
resource "aws_pinpoint_app" "example" {
  name = "test-app"

  limits {
    maximum_duration = 600
  }

  quiet_time {
    start = "00:00"
    end   = "06:00"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The application name. By default generated by Terraform
- `name_prefix` - (Optional) The name of the Pinpoint application. Conflicts with `name`
- `campaign_hook` - (Optional) The default campaign limits for the app. These limits apply to each campaign for the app, unless the campaign overrides the default with limits of its own
- `limits` - (Optional) The default campaign limits for the app. These limits apply to each campaign for the app, unless the campaign overrides the default with limits of its own
- `quiet_time` - (Optional) The default quiet time for the app. Each campaign for this app sends no messages during this time unless the campaign overrides the default with a quiet time of its own
- `tags` - (Optional) Key-value mapping of resource tags

`campaign_hook` supports the following:

- `lambda_function_name` - (Optional) Lambda function name or ARN to be called for delivery. Conflicts with `web_url`

- `mode` - (Required if `lambda_function_name` or `web_url` are provided) What mode Lambda should be invoked in. Valid values for this parameter are `DELIVERY`, `FILTER`.

- `web_url` - (Optional) Web URL to call for hook. If the URL has authentication specified it will be added as authentication to the request. Conflicts with `lambda_function_name`

`limits` supports the following:

- `daily` - (Optional) The maximum number of messages that the campaign can send daily.
- `maximum_duration` - (Optional) The length of time (in seconds) that the campaign can run before it ends and message deliveries stop. This duration begins at the scheduled start time for the campaign. The minimum value is 60.
- `messages_per_second` - (Optional) The number of messages that the campaign can send per second. The minimum value is 50, and the maximum is 20000.
- `total` - (Optional) The maximum total number of messages that the campaign can send.

`quiet_time` supports the following:

- `end` - (Optional) The default end time for quiet time in ISO 8601 format. Required if `start` is set
- `start` - (Optional) The default start time for quiet time in ISO 8601 format. Required if `end` is set

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `application_id` - The Application ID of the Pinpoint App.
- `arn` - Amazon Resource Name (ARN) of the PinPoint Application

» Import

Pinpoint App can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_app.name application-id
```

» Resource: aws_pinpoint_adm_channel

Provides a Pinpoint ADM (Amazon Device Messaging) Channel resource.

Note: All arguments including the Client ID and Client Secret will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_pinpoint_app" "app" {}

resource "aws_pinpoint_adm_channel" "channel" {
  application_id = "${aws_pinpoint_app.app.application_id}"
  client_id      = ""
  client_secret  = ""
  enabled        = true
}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `client_id` - (Required) Client ID (part of OAuth Credentials) obtained via Amazon Developer Account.
- `client_secret` - (Required) Client Secret (part of OAuth Credentials) obtained via Amazon Developer Account.
- `enabled` - (Optional) Specifies whether to enable the channel. Defaults to `true`.

» Import

Pinpoint ADM Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_adm_channel.channel application-id
```

» Resource: aws_pinpoint_apns_channel

Provides a Pinpoint APNs Channel resource.

Note: All arguments, including certificates and tokens, will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_pinpoint_apns_channel" "apns" {
  application_id = "${aws_pinpoint_app.app.application_id}"

  certificate = "${file("./certificate.pem")}"
  private_key = "${file("./private_key.key")}"
}

resource "aws_pinpoint_app" "app" {}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `enabled` - (Optional) Whether the channel is enabled or disabled. Defaults to `true`.
- `default_authentication_method` - (Optional) The default authentication method used for APNs. **NOTE**: Amazon Pinpoint uses this default for every APNs push notification that you send using the console. You can override the default when you send a message programmatically using the Amazon Pinpoint API, the AWS CLI, or an AWS SDK. If your default authentication type fails, Amazon Pinpoint doesn't attempt to use the other authentication type.

One of the following sets of credentials is also required.

If you choose to use **Certificate credentials** you will have to provide: * `certificate` - (Required) The pem encoded TLS Certificate from Apple. * `private_key` - (Required) The Certificate Private Key file (ie. `.key` file).

If you choose to use **Key credentials** you will have to provide: * `bundle_id` - (Required) The ID assigned to your iOS app. To find this value, choose Certificates, IDs & Profiles, choose App IDs in the Identifiers section, and choose your app. * `team_id` - (Required) The ID assigned to your Apple developer account team. This value is provided on the Membership page. * `token_key` - (Required) The `.p8` file that you download from your Apple developer account when you create an authentication key. * `token_key_id` - (Required) The ID assigned to your signing key. To find this value, choose Certificates, IDs & Profiles, and choose your key in the Keys section.

» Import

Pinpoint APNs Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_apns_channel.apns application-id
```

» Resource: aws_pinpoint_apns_sandbox_channel

Provides a Pinpoint APNs Sandbox Channel resource.

Note: All arguments, including certificates and tokens, will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_pinpoint_apns_sandbox_channel" "apns_sandbox" {
  application_id = "${aws_pinpoint_app.app.application_id}"

  certificate = "${file("./certificate.pem")}"
  private_key = "${file("./private_key.key")}"
}

resource "aws_pinpoint_app" "app" {}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `enabled` - (Optional) Whether the channel is enabled or disabled. Defaults to `true`.
- `default_authentication_method` - (Optional) The default authentication method used for APNs Sandbox. **NOTE**: Amazon Pinpoint uses this default for every APNs push notification that you send using the console. You can override the default when you send a message programmatically using the Amazon Pinpoint API, the AWS CLI, or an AWS SDK. If your default authentication type fails, Amazon Pinpoint doesn't attempt to use the other authentication type.

One of the following sets of credentials is also required.

If you choose to use **Certificate credentials** you will have to provide: * `certificate` - (Required) The pem encoded TLS Certificate from Apple. * `private_key` - (Required) The Certificate Private Key file (ie. `.key` file).

If you choose to use **Key credentials** you will have to provide: * `bundle_id` - (Required) The ID assigned to your iOS app. To find this value, choose Certificates, IDs & Profiles, choose App IDs in the Identifiers section, and choose your app. * `team_id` - (Required) The ID assigned to your Apple developer account team. This value is provided on the Membership page. * `token_key` - (Required) The `.p8` file that you download from your Apple developer account when you create an authentication key. * `token_key_id` - (Required) The ID assigned to your signing key. To find this value, choose Certificates, IDs & Profiles, and choose your key in the Keys section.

» Import

Pinpoint APNs Sandbox Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_apns_sandbox_channel.apns_sandbox application-id
```

» Resource: aws_pinpoint_apns_voip_channel

Provides a Pinpoint APNs VoIP Channel resource.

Note: All arguments, including certificates and tokens, will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_pinpoint_apns_voip_channel" "apns_voip" {
  application_id = "${aws_pinpoint_app.app.application_id}"

  certificate = "${file("./certificate.pem")}"
  private_key = "${file("./private_key.key")}"
}

resource "aws_pinpoint_app" "app" {}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `enabled` - (Optional) Whether the channel is enabled or disabled. Defaults to `true`.
- `default_authentication_method` - (Optional) The default authentication method used for APNs. **NOTE**: Amazon Pinpoint uses this default for every APNs push notification that you send using the console. You can override the default when you send a message programmatically using the Amazon Pinpoint API, the AWS CLI, or an AWS SDK. If your default authentication type fails, Amazon Pinpoint doesn't attempt to use the other authentication type.

One of the following sets of credentials is also required.

If you choose to use **Certificate credentials** you will have to provide: * `certificate` - (Required) The pem encoded TLS Certificate from Apple. * `private_key` - (Required) The Certificate Private Key file (ie. `.key` file).

If you choose to use **Key credentials** you will have to provide: * `bundle_id` - (Required) The ID assigned to your iOS app. To find this value, choose Certificates, IDs & Profiles, choose App IDs in the Identifiers section, and choose your app. * `team_id` - (Required) The ID assigned to your Apple developer account team. This value is provided on the Membership page. * `token_key` - (Required) The `.p8` file that you download from your Apple developer account when you create an authentication key. * `token_key_id` - (Required) The ID assigned to your signing key. To find this value, choose Certificates, IDs & Profiles, and choose your key in the Keys section.

» Import

Pinpoint APNs VoIP Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_apns_voip_channel.apns_voip application-id
```

» Resource: aws_pinpoint_apns_voip_sandbox_channel

Provides a Pinpoint APNs VoIP Sandbox Channel resource.

Note: All arguments, including certificates and tokens, will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_pinpoint_apns_voip_sandbox_channel" "apns_voip_sandbox" {
  application_id = "${aws_pinpoint_app.app.application_id}"

  certificate = "${file("./certificate.pem")}"
  private_key = "${file("./private_key.key")}"
}

resource "aws_pinpoint_app" "app" {}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `enabled` - (Optional) Whether the channel is enabled or disabled. Defaults to `true`.
- `default_authentication_method` - (Optional) The default authentication method used for APNs. **NOTE**: Amazon Pinpoint uses this default for every APNs push notification that you send using the console. You can override the default when you send a message programmatically using the Amazon Pinpoint API, the AWS CLI, or an AWS SDK. If your default authentication type fails, Amazon Pinpoint doesn't attempt to use the other authentication type.

One of the following sets of credentials is also required.

If you choose to use **Certificate credentials** you will have to provide: * `certificate` - (Required) The pem encoded TLS Certificate from Apple. * `private_key` - (Required) The Certificate Private Key file (ie. `.key` file).

If you choose to use **Key credentials** you will have to provide: * `bundle_id` - (Required) The ID assigned to your iOS app. To find this value, choose Certificates, IDs & Profiles, choose App IDs in the Identifiers section, and choose your app. * `team_id` - (Required) The ID assigned to your Apple developer account team. This value is provided on the Membership page. * `token_key` - (Required) The `.p8` file that you download from your Apple developer account when you create an authentication key. * `token_key_id` - (Required) The ID assigned to your signing key. To find this value, choose Certificates, IDs & Profiles, and choose your key in the Keys section.

» Import

Pinpoint APNs VoIP Sandbox Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_apns_voip_sandbox_channel.apns_voip_sandbox application-id
```

» Resource: aws_pinpoint_baidu_channel

Provides a Pinpoint Baidu Channel resource.

Note: All arguments including the Api Key and Secret Key will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_pinpoint_app" "app" {}
resource "aws_pinpoint_baidu_channel" "channel" {
  application_id = "${aws_pinpoint_app.app.application_id}"
  api_key        = ""
  secret_key     = ""
}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `enabled` - (Optional) Specifies whether to enable the channel. Defaults to `true`.
- `api_key` - (Required) Platform credential API key from Baidu.
- `secret_key` - (Required) Platform credential Secret key from Baidu.

» Import

Pinpoint Baidu Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_baidu_channel.channel application-id
```

» Resource: aws_pinpoint_email_channel

Provides a Pinpoint SMS Channel resource.

» Example Usage

```
resource "aws_pinpoint_email_channel" "email" {
  application_id = "${aws_pinpoint_app.app.application_id}"
  from_address   = "user@example.com"
  identity       = "${aws_ses_domain_identity.identity.arn}"
  role_arn       = "${aws_iam_role.role.arn}"
}

resource "aws_pinpoint_app" "app" {}

resource "aws_ses_domain_identity" "identity" {
  domain = "example.com"
}

resource "aws_iam_role" "role" {
  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "role_policy" {
  name = "role_policy"
  role = "${aws_iam_role.role.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": {
    "Action": [
      "mobileanalytics:PutEvents",
      "mobileanalytics:PutItems"
    ],
    "Effect": "Allow",
    "Resource": [
      "*"
    ]
  }
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `enabled` - (Optional) Whether the channel is enabled or disabled. Defaults to `true`.

- `from_address` - (Required) The email address used to send emails from.
- `identity` - (Required) The ARN of an identity verified with SES.
- `role_arn` - (Required) The ARN of an IAM Role used to submit events to Mobile Analytics' event ingestion service.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `messages_per_second` - Messages per second that can be sent.

» Import

Pinpoint Email Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_email_channel.email application-id
```

» Resource: aws_pinpoint_event_stream

Provides a Pinpoint Event Stream resource.

» Example Usage

```
resource "aws_pinpoint_event_stream" "stream" {
  application_id         = "${aws_pinpoint_app.app.application_id}"
  destination_stream_arn = "${aws_kinesis_stream.test_stream.arn}"
  role_arn               = "${aws_iam_role.test_role.arn}"
}

resource "aws_pinpoint_app" "app" {}

resource "aws_kinesis_stream" "test_stream" {
  name        = "pinpoint-kinesis-test"
  shard_count = 1
}

resource "aws_iam_role" "test_role" {
  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "pinpoint.us-east-1.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "test_role_policy" {
  name = "test_policy"
  role = "${aws_iam_role.test_role.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": {
    "Action": [
      "kinesis:PutRecords",
      "kinesis:DescribeStream"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:kinesis:us-east-1:*:*/*"
    ]
  }
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `destination_stream_arn` - (Required) The Amazon Resource Name (ARN) of the Amazon Kinesis stream or Firehose delivery stream to which you want to publish events.
- `role_arn` - (Required) The IAM role that authorizes Amazon Pinpoint to publish events to the stream in your account.

» Import

Pinpoint Event Stream can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_event_stream.stream application-id
```

» Resource: aws_pinpoint_gcm_channel

Provides a Pinpoint GCM Channel resource.

Note: Api Key argument will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_pinpoint_gcm_channel" "gcm" {
  application_id = "${aws_pinpoint_app.app.application_id}"
  api_key        = "api_key"
}

resource "aws_pinpoint_app" "app" {}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `api_key` - (Required) Platform credential API key from Google.
- `enabled` - (Optional) Whether the channel is enabled or disabled. Defaults to `true`.

» Import

Pinpoint GCM Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_gcm_channel.gcm application-id
```

» Resource: aws_pinpoint_sms_channel

Provides a Pinpoint SMS Channel resource.

» Example Usage

```
resource "aws_pinpoint_sms_channel" "sms" {
  application_id = "${aws_pinpoint_app.app.application_id}"
}

resource "aws_pinpoint_app" "app" {}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The application ID.
- `enabled` - (Optional) Whether the channel is enabled or disabled. Defaults to `true`.
- `sender_id` - (Optional) Sender identifier of your messages.
- `short_code` - (Optional) The Short Code registered with the phone provider.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `promotional_messages_per_second` - Promotional messages per second that can be sent.
- `transactional_messages_per_second` - Transactional messages per second that can be sent.

» Import

Pinpoint SMS Channel can be imported using the `application-id`, e.g.

```
$ terraform import aws_pinpoint_sms_channel.sms application-id
```

» Data Source: aws_pricing_product

Use this data source to get the pricing information of all products in AWS. This data source is only available in a us-east-1 or ap-south-1 provider.

» Example Usage

```
data "aws_pricing_product" "example" {
  service_code = "AmazonEC2"

  filters {
    field = "instanceType"
    value = "c5.xlarge"
  }

  filters {
    field = "operatingSystem"
    value = "Linux"
  }

  filters {
    field = "location"
    value = "US East (N. Virginia)"
  }

  filters {
    field = "preInstalledSw"
    value = "NA"
  }

  filters {
    field = "licenseModel"
    value = "No License required"
  }

  filters {
    field = "tenancy"
    value = "Shared"
  }
}
```

```
data "aws_pricing_product" "example" {
  service_code = "AmazonRedshift"

  filters {
    field = "instanceType"
    value = "ds1.xlarge"
  }

  filters {
    field = "location"
    value = "US East (N. Virginia)"
  }
}
```

» Argument Reference

- `service_code` - (Required) The code of the service. Available service codes can be fetched using the DescribeServices pricing API call.
- `filters` - (Required) A list of filters. Passed directly to the API (see GetProducts API reference). These filters must describe a single product, this resource will fail if more than one product is returned by the API.

» filters

- `field` (Required) The product attribute name that you want to filter on.
- `value` (Required) The product attribute value that you want to filter on.

» Attributes Reference

- `result` - Set to the product returned from the API.

» Data Source: aws_qldb_ledger

Use this data source to fetch information about a Quantum Ledger Database.

» Example Usage

```
data "aws_qldb_ledger" "example" {
  name = "an_example_ledger"
}
```

» Argument Reference

- `name` - (Required) The friendly name of the ledger to match.

» Attributes Reference

- `arn` - Amazon Resource Name (ARN) of the ledger.
- `deletion_protection` - Deletion protection on the QLDB Ledger instance. Set to `true` by default.

» Resource: aws_qldb_ledger

Provides an AWS Quantum Ledger Database (QLDB) resource

NOTE: Deletion protection is enabled by default. To successfully delete this resource via Terraform, `deletion_protection = false` must be applied before attempting deletion.

» Example Usage

```
resource "aws_qldb_ledger" "sample-ledger" {
  name = "sample-ledger"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The friendly name for the QLDB Ledger instance. By default generated by Terraform.
- `deletion_protection` - (Optional) The deletion protection for the QLDB Ledger instance. By default it is `true`. To delete this resource via Terraform, this value must be configured to `false` and applied first before attempting deletion.
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Name of the QLDB Ledger
- `arn` - The ARN of the QLDB Ledger

» Import

QLDB Ledgers can be imported using the `name`, e.g.

```
$ terraform import aws_qldb_ledger.sample-ledger sample-ledger
```

» Resource: aws_quicksight_group

Resource for managing QuickSight Group

» Example Usage

```
resource "aws_quicksight_group" "example" {
    group_name = "tf-example"
}
```

» Argument Reference

The following arguments are supported:

- `group_name` - (Required) A name for the group.
- `aws_account_id` - (Optional) The ID for the AWS account that the group is in. Currently, you use the ID for the AWS account that contains your Amazon QuickSight account.
- `description` - (Optional) A description for the group.
- `namespace` - (Optional) The namespace. Currently, you should set this to `default`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of group

» Import

QuickSight Group can be imported using the aws account id, namespace and group name separated by `/`.

```
$ terraform import aws_quicksight_group.example 123456789123/default/tf-example
```

» Resource: aws_quicksight_user

Resource for managing QuickSight User

» Example Usage

```
resource "aws_quicksight_user" "example" {
  user_name      = "an-author"
  email          = "author@example.com"
  identity_type  = "IAM"
  user_role      = "AUTHOR"
}
```

» Argument Reference

The following arguments are supported:

- `email` - (Required) The email address of the user that you want to register.
- `identity_type` - (Required) Amazon QuickSight supports several ways of managing the identity of users. This parameter accepts two values: `IAM` and `QUICKSIGHT`.
- `user_role` - (Required) The Amazon QuickSight role of the user. The user role can be one of the following: `READER`, `AUTHOR`, or `ADMIN`
- `user_name` - (Optional) The Amazon QuickSight user name that you want to create for the user you are registering.
- `aws_account_id` - (Optional) The ID for the AWS account that the group is in. Currently, you use the ID for the AWS account that contains your Amazon QuickSight account.
- `iam_arn` - (Optional) The ARN of the IAM user or role that you are registering with Amazon QuickSight.
- `namespace` - (Optional) The namespace. Currently, you should set this to `default`.
- `session_name` - (Optional) The name of the IAM session to use when assuming roles that can embed QuickSight dashboards.

» Attributes Reference

All above attributes except for `session_name` and `identity_type` are exported as well as:

- `arn` - Amazon Resource Name (ARN) of the user

» Import

Importing is currently not supported on this resource.

» Data Source: aws_ram_resource_share

`aws_ram_resource_share` Retrieve information about a RAM Resource Share.

» Example Usage

```
data "aws_ram_resource_share" "example" {
  name = "example"
  resource_owner = "SELF"
}
```

» Search by filters

```
data "aws_ram_resource_share" "tag_filter" {
  name           = "MyResourceName"
  resource_owner = "SELF"

  filter {
    name   = "NameOfTag"
    values = ["exampleNameTagValue"]
  }
}
```

» Argument Reference

The following Arguments are supported

- `name` - (Required) The name of the resource share to retrieve.
- `resource_owner` (Required) The owner of the resource share. Valid values are SELF or OTHER-ACCOUNTS
- `filter` - (Optional) A filter used to scope the list e.g. by tags. See related docs.
  - `name` - (Required) The name of the tag key to filter on.
  - `values` - (Required) The value of the tag key.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the resource share.
- `id` - The Amazon Resource Name (ARN) of the resource share.
- `status` - The Status of the RAM share.
- `tags` - The Tags attached to the RAM share

» Resource: aws_ram_principal_association

Provides a Resource Access Manager (RAM) principal association. Depending if RAM Sharing with AWS Organizations is enabled, the RAM behavior with different principal types changes.

When RAM Sharing with AWS Organizations is enabled:

- For AWS Account ID, Organization, and Organizational Unit principals within the same AWS Organization, no resource share invitation is sent and resources become available automatically after creating the association.
- For AWS Account ID principals outside the AWS Organization, a resource share invitation is sent and must be accepted before resources become available. See the `aws_ram_resource_share_accepter` resource to accept these invitations.

When RAM Sharing with AWS Organizations is not enabled:

- Organization and Organizational Unit principals cannot be used.
- For AWS Account ID principals, a resource share invitation is sent and must be accepted before resources become available. See the `aws_ram_resource_share_accepter` resource to accept these invitations.

» Example Usage

» AWS Account ID

```
resource "aws_ram_resource_share" "example" {
  # ... other configuration ...
  allow_external_principals = true
}

resource "aws_ram_principal_association" "example" {
  principal          = "111111111111"
  resource_share_arn = "${aws_ram_resource_share.example.arn}"
}
```

» AWS Organization

```
resource "aws_ram_principal_association" "example" {
  principal          = "${aws_organizations_organization.example.arn}"
  resource_share_arn = "${aws_ram_resource_share.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `principal` - (Required) The principal to associate with the resource share. Possible values are an AWS account ID, an AWS Organizations Organization ARN, or an AWS Organizations Organization Unit ARN.
- `resource_share_arn` - (Required) The Amazon Resource Name (ARN) of the resource share.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) of the Resource Share and the principal, separated by a comma.

» Import

RAM Principal Associations can be imported using their Resource Share ARN and the `principal` separated by a comma, e.g.

```
$ terraform import aws_ram_principal_association.example arn:aws:ram:eu-west-1:123456789012:
```

» Resource: aws_ram_resource_association

Manages a Resource Access Manager (RAM) Resource Association.

NOTE: Certain AWS resources (e.g. EC2 Subnets) can only be shared in an AWS account that is a member of an AWS Organizations organization with organization-wide Resource Access Manager functionality enabled. See the Resource Access Manager User Guide and AWS service specific documentation for additional information.

» Example Usage

```
resource "aws_ram_resource_association" "example" {
  resource_arn       = "${aws_subnet.example.arn}"
  resource_share_arn = "${aws_ram_resource_share.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `resource_arn` - (Required) Amazon Resource Name (ARN) of the resource to associate with the RAM Resource Share.
- `resource_share_arn` - (Required) Amazon Resource Name (ARN) of the RAM Resource Share.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) of the resource share.

» Import

RAM Resource Associations can be imported using their Resource Share ARN and Resource ARN separated by a comma, e.g.

```
$ terraform import aws_ram_resource_association.example arn:aws:ram:eu-west-1:123456789012:r
```

» Resource: aws_ram_resource_share

Manages a Resource Access Manager (RAM) Resource Share. To association principals with the share, see the `aws_ram_principal_association` resource. To associate resources with the share, see the `aws_ram_resource_association` resource.

» Example Usage

```
resource "aws_ram_resource_share" "example" {
  name                      = "example"
  allow_external_principals = true
  tags = {
    Environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the resource share.
- `allow_external_principals` - (Optional) Indicates whether principals outside your organization can be associated with a resource share.
- `tags` - (Optional) A mapping of tags to assign to the resource share.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) of the resource share.
- `id` - The Amazon Resource Name (ARN) of the resource share.

» Import

Resource shares can be imported using the `id`, e.g.

```
$ terraform import aws_ram_resource_share.example arn:aws:ram:eu-west-1:123456789012:resourc
```

» Resource: aws_ram_resource_share_accepter

Manage accepting a Resource Access Manager (RAM) Resource Share invitation. From a *receiver* AWS account, accept an invitation to share resources that were shared by a *sender* AWS account. To create a resource share in the *sender*, see the `aws_ram_resource_share` resource.

Note: If both AWS accounts are in the same Organization and RAM Sharing with AWS Organizations is enabled, this resource is not necessary as RAM Resource Share invitations are not used.

» Example Usage

This configuration provides an example of using multiple Terraform AWS providers to configure two different AWS accounts. In the *sender* account, the configuration creates a `aws_ram_resource_share` and uses a data source in the *receiver* account to create a `aws_ram_principal_association` resource with the *receiver's* account ID. In the *receiver* account, the configuration accepts the invitation to share resources with the `aws_ram_resource_share_accepter`.

```
provider "aws" {
  profile = "profile2"
}

provider "aws" {
  alias   = "alternate"
  profile = "profile1"
}

resource "aws_ram_resource_share" "sender_share" {
  provider = "aws.alternate"

  name                      = "tf-test-resource-share"
  allow_external_principals = true

  tags = {
      Name = "tf-test-resource-share"
  }
}

resource "aws_ram_principal_association" "sender_invite" {
  provider = "aws.alternate"

  principal          = "${data.aws_caller_identity.receiver.account_id}"
  resource_share_arn = "${aws_ram_resource_share.test.arn}"
}

data "aws_caller_identity" "receiver" {}

resource "aws_ram_resource_share_accepter" "receiver_accept" {
  share_arn = "${aws_ram_principal_association.test.resource_share_arn}"
}
```

» Argument Reference

The following arguments are supported:

- `share_arn` - (Required) The ARN of the resource share.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `invitation_arn` - The ARN of the resource share invitation.
- `share_id` - The ID of the resource share as displayed in the console.
- `status` - The status of the invitation (e.g., ACCEPTED, REJECTED).
- `receiver_account_id` - The account ID of the receiver account which accepts the invitation.
- `sender_account_id` - The account ID of the sender account which extends the invitation.
- `share_name` - The name of the resource share.
- `resources` - A list of the resource ARNs shared via the resource share.

» Import

Resource share accepters can be imported using the resource share ARN, e.g.

```
$ terraform import aws_ram_resource_share_accepter.example arn:aws:ram:us-east-1:12345678901
```

» Data Source: aws_db_cluster_snapshot

Use this data source to get information about a DB Cluster Snapshot for use when provisioning DB clusters.

NOTE: This data source does not apply to snapshots created on DB Instances. See the `aws_db_snapshot` data source for DB Instance snapshots.

» Example Usage

```
data "aws_db_cluster_snapshot" "development_final_snapshot" {
  db_cluster_identifier = "development_cluster"
  most_recent           = true
}

# Use the last snapshot of the dev database before it was destroyed to create
# a new dev database.
resource "aws_rds_cluster" "aurora" {
  cluster_identifier   = "development_cluster"
  snapshot_identifier  = "${data.aws_db_cluster_snapshot.development_final_snapshot.id}"
  db_subnet_group_name = "my_db_subnet_group"

  lifecycle {
    ignore_changes = ["snapshot_identifier"]
  }
}

resource "aws_rds_cluster_instance" "aurora" {
  cluster_identifier   = "${aws_rds_cluster.aurora.id}"
  instance_class       = "db.t2.small"
  db_subnet_group_name = "my_db_subnet_group"
}
```

» Argument Reference

The following arguments are supported:

- `most_recent` - (Optional) If more than one result is returned, use the most recent Snapshot.
- `db_cluster_identifier` - (Optional) Returns the list of snapshots created by the specific db_cluster
- `db_cluster_snapshot_identifier` - (Optional) Returns information on a specific snapshot_id.
- `snapshot_type` - (Optional) The type of snapshots to be returned. If you don't specify a SnapshotType value, then both automated and manual DB cluster snapshots are returned. Shared and public DB Cluster Snapshots are not included in the returned results by default. Possible values are, `automated`, `manual`, `shared` and `public`.
- `include_shared` - (Optional) Set this value to true to include shared manual DB Cluster Snapshots from other AWS accounts that this AWS account has been given permission to copy or restore, otherwise set this value to false. The default is `false`.
- `include_public` - (Optional) Set this value to true to include manual DB Cluster Snapshots that are public and can be copied or restored by any AWS account, otherwise set this value to false. The default is `false`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `allocated_storage` - Specifies the allocated storage size in gigabytes (GB).
- `availability_zones` - List of EC2 Availability Zones that instances in the DB cluster snapshot can be restored in.
- `db_cluster_identifier` - Specifies the DB cluster identifier of the DB cluster that this DB cluster snapshot was created from.

- db_cluster_snapshot_arn - The Amazon Resource Name (ARN) for the DB Cluster Snapshot.
- engine_version - Version of the database engine for this DB cluster snapshot.
- engine - Specifies the name of the database engine.
- id - The snapshot ID.
- kms_key_id - If storage_encrypted is true, the AWS KMS key identifier for the encrypted DB cluster snapshot.
- license_model - License model information for the restored DB cluster.
- port - Port that the DB cluster was listening on at the time of the snapshot.
- snapshot_create_time - Time when the snapshot was taken, in Universal Coordinated Time (UTC).
- source_db_cluster_snapshot_identifier - The DB Cluster Snapshot Arn that the DB Cluster Snapshot was copied from. It only has value in case of cross customer or cross region copy.
- status - The status of this DB Cluster Snapshot.
- storage_encrypted - Specifies whether the DB cluster snapshot is encrypted.
- vpc_id - The VPC ID associated with the DB cluster snapshot.
- tags - A mapping of tags for the resource.

» Data Source: aws_db_event_categories

» Example Usage

List the event categories of all the RDS resources.

```
data "aws_db_event_categories" "example" {}

output "example" {
  value = "${data.aws_db_event_categories.example.event_categories}"
}
```

List the event categories specific to the RDS resource db-snapshot.

```
data "aws_db_event_categories" "example" {
  source_type = "db-snapshot"
}

output "example" {
  value = "${data.aws_db_event_categories.example.event_categories}"
}
```

» Argument Reference

The following arguments are supported:

- `source_type` - (Optional) The type of source that will be generating the events. Valid options are db-instance, db-security-group, db-parameter-group, db-snapshot, db-cluster or db-cluster-snapshot.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `event_categories` - A list of the event categories.

» Data Source: aws_db_instance

Use this data source to get information about an RDS instance

» Example Usage

```
data "aws_db_instance" "database" {
  db_instance_identifier = "my-test-database"
}
```

» Argument Reference

The following arguments are supported:

- `db_instance_identifier` - (Required) The name of the RDS instance

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `address` - The hostname of the RDS instance. See also `endpoint` and `port`.
- `allocated_storage` - Specifies the allocated storage size specified in gigabytes.
- `auto_minor_version_upgrade` - Indicates that minor version patches are applied automatically.
- `availability_zone` - Specifies the name of the Availability Zone the DB instance is located in.

- `backup_retention_period` - Specifies the number of days for which automatic DB snapshots are retained.
- `db_cluster_identifier` - If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.
- `db_instance_arn` - The Amazon Resource Name (ARN) for the DB instance.
- `db_instance_class` - Contains the name of the compute and memory capacity class of the DB instance.
- `db_name` - Contains the name of the initial database of this instance that was provided at create time, if one was specified when the DB instance was created. This same name is returned for the life of the DB instance.
- `db_parameter_groups` - Provides the list of DB parameter groups applied to this DB instance.
- `db_security_groups` - Provides List of DB security groups associated to this DB instance.
- `db_subnet_group` - Specifies the name of the subnet group associated with the DB instance.
- `db_instance_port` - Specifies the port that the DB instance listens on.
- `enabled_cloudwatch_logs_exports` - List of log types to export to cloudwatch.
- `endpoint` - The connection endpoint in `address:port` format.
- `engine` - Provides the name of the database engine to be used for this DB instance.
- `engine_version` - Indicates the database engine version.
- `hosted_zone_id` - The canonical hosted zone ID of the DB instance (to be used in a Route 53 Alias record).
- `iops` - Specifies the Provisioned IOPS (I/O operations per second) value.
- `kms_key_id` - If StorageEncrypted is true, the KMS key identifier for the encrypted DB instance.
- `license_model` - License model information for this DB instance.
- `master_username` - Contains the master username for the DB instance.
- `monitoring_interval` - The interval, in seconds, between points when Enhanced Monitoring metrics are collected for the DB instance.
- `monitoring_role_arn` - The ARN for the IAM role that permits RDS to send Enhanced Monitoring metrics to CloudWatch Logs.
- `multi_az` - Specifies if the DB instance is a Multi-AZ deployment.
- `option_group_memberships` - Provides the list of option group memberships for this DB instance.
- `port` - The database port.
- `preferred_backup_window` - Specifies the daily time range during which automated backups are created.
- `preferred_maintenance_window` - Specifies the weekly time range during which system maintenance can occur in UTC.
- `publicly_accessible` - Specifies the accessibility options for the DB instance.
- `resource_id` - The RDS Resource ID of this instance.

- `storage_encrypted` - Specifies whether the DB instance is encrypted.
- `storage_type` - Specifies the storage type associated with DB instance.
- `timezone` - The time zone of the DB instance.
- `vpc_security_groups` - Provides a list of VPC security group elements that the DB instance belongs to.
- `replicate_source_db` - The identifier of the source DB that this is a replica of.
- `ca_cert_identifier` - Specifies the identifier of the CA certificate for the DB instance.

» Data Source: aws_db_snapshot

Use this data source to get information about a DB Snapshot for use when provisioning DB instances

NOTE: This data source does not apply to snapshots created on Aurora DB clusters. See the `aws_db_cluster_snapshot` data source for DB Cluster snapshots.

» Example Usage

```
resource "aws_db_instance" "prod" {
  allocated_storage    = 10
  engine               = "mysql"
  engine_version       = "5.6.17"
  instance_class       = "db.t2.micro"
  name                 = "mydb"
  username             = "foo"
  password             = "bar"
  db_subnet_group_name = "my_database_subnet_group"
  parameter_group_name = "default.mysql5.6"
}

data "aws_db_snapshot" "latest_prod_snapshot" {
  db_instance_identifier = "${aws_db_instance.prod.id}"
  most_recent            = true
}

# Use the latest production snapshot to create a dev instance.
resource "aws_db_instance" "dev" {
  instance_class      = "db.t2.micro"
  name                = "mydbdev"
  snapshot_identifier = "${data.aws_db_snapshot.latest_prod_snapshot.id}"
  lifecycle {
    ignore_changes = ["snapshot_identifier"]
  }
}
```

» Argument Reference

NOTE: One of either `db_instance_identifier` or `db_snapshot_identifier` is required.

The following arguments are supported:

- `most_recent` - (Optional) If more than one result is returned, use the most recent Snapshot.
- `db_instance_identifier` - (Optional) Returns the list of snapshots created by the specific db_instance
- `db_snapshot_identifier` - (Optional) Returns information on a specific snapshot_id.
- `snapshot_type` - (Optional) The type of snapshots to be returned. If you don't specify a SnapshotType value, then both automated and manual snapshots are returned. Shared and public DB snapshots are not included in the returned results by default. Possible values are, `automated`, `manual`, `shared` and `public`.
- `include_shared` - (Optional) Set this value to true to include shared manual DB snapshots from other AWS accounts that this AWS account has been given permission to copy or restore, otherwise set this value to false. The default is `false`.
- `include_public` - (Optional) Set this value to true to include manual DB snapshots that are public and can be copied or restored by any AWS account, otherwise set this value to false. The default is `false`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The snapshot ID.
- `allocated_storage` - Specifies the allocated storage size in gigabytes (GB).
- `availability_zone` - Specifies the name of the Availability Zone the DB instance was located in at the time of the DB snapshot.
- `db_snapshot_arn` - The Amazon Resource Name (ARN) for the DB snapshot.
- `encrypted` - Specifies whether the DB snapshot is encrypted.

- `engine` - Specifies the name of the database engine.
- `engine_version` - Specifies the version of the database engine.
- `iops` - Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.
- `kms_key_id` - The ARN for the KMS encryption key.
- `license_model` - License model information for the restored DB instance.
- `option_group_name` - Provides the option group name for the DB snapshot.
- `source_db_snapshot_identifier` - The DB snapshot Arn that the DB snapshot was copied from. It only has value in case of cross customer or cross region copy.
- `source_region` - The region that the DB snapshot was created in or copied from.
- `status` - Specifies the status of this DB snapshot.
- `storage_type` - Specifies the storage type associated with DB snapshot.
- `vpc_id` - Specifies the ID of the VPC associated with the DB snapshot.
- `snapshot_create_time` - Provides the time when the snapshot was taken, in Universal Coordinated Time (UTC).

» Data Source: aws_rds_cluster

Provides information about a RDS cluster.

» Example Usage

```
data "aws_rds_cluster" "clusterName" {
  cluster_identifier = "clusterName"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_identifier` - (Required) The cluster identifier of the RDS cluster.

» Attributes Reference

See the RDS Cluster Resource for details on the returned attributes - they are identical.

» Resource: aws_db_cluster_snapshot

Manages a RDS database cluster snapshot for Aurora clusters. For managing RDS database instance snapshots, see the `aws_db_snapshot` resource.

» Example Usage

```
resource "aws_db_cluster_snapshot" "example" {
  db_cluster_identifier          = "${aws_rds_cluster.example.id}"
  db_cluster_snapshot_identifier = "resourcetestsnapshot1234"
}
```

» Argument Reference

The following arguments are supported:

- `db_cluster_identifier` - (Required) The DB Cluster Identifier from which to take the snapshot.
- `db_cluster_snapshot_identifier` - (Required) The Identifier for the snapshot.
- `tags` - (Optional) A mapping of tags to assign to the DB cluster.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `allocated_storage` - Specifies the allocated storage size in gigabytes (GB).
- `availability_zones` - List of EC2 Availability Zones that instances in the DB cluster snapshot can be restored in.
- `db_cluster_snapshot_arn` - The Amazon Resource Name (ARN) for the DB Cluster Snapshot.
- `engine` - Specifies the name of the database engine.
- `engine_version` - Version of the database engine for this DB cluster snapshot.
- `kms_key_id` - If storage_encrypted is true, the AWS KMS key identifier for the encrypted DB cluster snapshot.
- `license_model` - License model information for the restored DB cluster.
- `port` - Port that the DB cluster was listening on at the time of the snapshot.
- `source_db_cluster_snapshot_identifier` - The DB Cluster Snapshot Arn that the DB Cluster Snapshot was copied from. It only has value in case of cross customer or cross region copy.

- `storage_encrypted` - Specifies whether the DB cluster snapshot is encrypted.
- `status` - The status of this DB Cluster Snapshot.
- `vpc_id` - The VPC ID associated with the DB cluster snapshot.

» Timeouts

`aws_db_cluster_snapshot` provides the following Timeouts configuration options:

- `create` - (Default `20m`) How long to wait for the snapshot to be available.

» Import

`aws_db_cluster_snapshot` can be imported by using the cluster snapshot identifier, e.g.

```
$ terraform import aws_db_cluster_snapshot.example my-cluster-snapshot
```

» Resource: aws_db_event_subscription

Provides a DB event subscription resource.

» Example Usage

```
resource "aws_db_instance" "default" {
  allocated_storage    = 10
  engine               = "mysql"
  engine_version       = "5.6.17"
  instance_class       = "db.t2.micro"
  name                 = "mydb"
  username             = "foo"
  password             = "bar"
  db_subnet_group_name = "my_database_subnet_group"
  parameter_group_name = "default.mysql5.6"
}

resource "aws_sns_topic" "default" {
  name = "rds-events"
}

resource "aws_db_event_subscription" "default" {
  name      = "rds-event-sub"
  sns_topic = "${aws_sns_topic.default.arn}"

  source_type = "db-instance"
  source_ids  = ["${aws_db_instance.default.id}"]

  event_categories = [
    "availability",
    "deletion",
    "failover",
    "failure",
    "low storage",
    "maintenance",
    "notification",
    "read replica",
    "recovery",
    "restoration",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the DB event subscription. By default generated by Terraform.
- `name_prefix` - (Optional) The name of the DB event subscription. Conflicts with `name`.
- `sns_topic` - (Required) The SNS topic to send events to.
- `source_ids` - (Optional) A list of identifiers of the event sources for which events will be returned. If not specified, then all sources are included in the response. If specified, a source_type must also be specified.
- `source_type` - (Optional) The type of source that will be generating the events. Valid options are `db-instance`, `db-security-group`, `db-parameter-group`, `db-snapshot`, `db-cluster` or `db-cluster-snapshot`. If not set, all sources will be subscribed to.
- `event_categories` - (Optional) A list of event categories for a SourceType that you want to subscribe to. See http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_Events.html or run `aws rds describe-event-categories`.
- `enabled` - (Optional) A boolean flag to enable/disable the subscription. Defaults to true.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes

The following additional atttributes are provided:

- `id` - The name of the RDS event notification subscription
- `arn` - The Amazon Resource Name of the RDS event notification subscription
- `customer_aws_id` - The AWS customer account associated with the RDS event notification subscription

» Timeouts

`aws_db_event_subscription` provides the following Timeouts configuration options:

- `create` - (Default `40m`) How long to wait for a RDS event notification subscription to be ready.
- `delete` - (Default `40m`) How long to wait for a RDS event notification subscription to be deleted.
- `update` - (Default `40m`) How long to wait for a RDS event notification subscription to be updated.

» Import

DB Event Subscriptions can be imported using the `name`, e.g.

```
$ terraform import aws_db_event_subscription.default rds-event-sub
```

» Resource: aws_db_instance

Provides an RDS instance resource. A DB instance is an isolated database environment in the cloud. A DB instance can contain multiple user-created databases.

Changes to a DB instance can occur when you manually change a parameter, such as `allocated_storage`, and are reflected in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because a modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately (see documentation below).

When upgrading the major version of an engine, `allow_major_version_upgrade` must be set to `true`.

Note: using `apply_immediately` can result in a brief downtime as the server reboots. See the AWS Docs on RDS Maintenance for more information.

Note: All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» RDS Instance Class Types

Amazon RDS supports three types of instance classes: Standard, Memory Optimized, and Burstable Performance. For more information please read the AWS RDS documentation about DB Instance Class Types

» Example Usage

» Basic Usage

```
resource "aws_db_instance" "default" {
  allocated_storage    = 20
  storage_type         = "gp2"
  engine               = "mysql"
  engine_version       = "5.7"
  instance_class       = "db.t2.micro"
  name                 = "mydb"
  username             = "foo"
  password             = "foobarbaz"
  parameter_group_name = "default.mysql5.7"
}
```

» Storage Autoscaling

To enable Storage Autoscaling with instances that support the feature, define the `max_allocated_storage` argument higher than the `allocated_storage` argument. Terraform will automatically hide differences with the `allocated_storage` argument value if autoscaling occurs.

```
resource "aws_db_instance" "example" {
  # ... other configuration ...

  allocated_storage     = 50
  max_allocated_storage = 100
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- `allocated_storage` - (Required unless a `snapshot_identifier` or `replicate_source_db` is provided) The allocated storage in gibibytes. If `max_allocated_storage` is configured, this argument represents the initial storage allocation and differences from the configuration will be ignored automatically when Storage Autoscaling occurs.
- `allow_major_version_upgrade` - (Optional) Indicates that major version upgrades are allowed. Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible.
- `apply_immediately` - (Optional) Specifies whether any database modifications are applied immediately, or during the next maintenance window. Default is `false`. See Amazon RDS Documentation for more information.
- `auto_minor_version_upgrade` - (Optional) Indicates that minor engine upgrades will be applied automatically to the DB instance during the maintenance window. Defaults to true.
- `availability_zone` - (Optional) The AZ for the RDS instance.
- `backup_retention_period` - (Optional) The days to retain backups for. Must be between `0` and `35`. Must be greater than `0` if the database is used as a source for a Read Replica. See Read Replica.
- `backup_window` - (Optional) The daily time range (in UTC) during which automated backups are created if they are enabled. Example: ”09:46-10:16”. Must not overlap with `maintenance_window`.
- `ca_cert_identifier` - (Optional) The identifier of the CA certificate for the DB instance.
- `character_set_name` - (Optional) The character set name to use for DB encoding in Oracle instances. This can’t be changed. See Oracle Character Sets Supported in Amazon RDS for more information.
- `copy_tags_to_snapshot` – (Optional, boolean) Copy all Instance `tags` to snapshots. Default is `false`.
- `db_subnet_group_name` - (Optional) Name of DB subnet group. DB instance will be created in the VPC associated with the DB subnet group. If unspecified, will be created in the `default` VPC, or in EC2 Classic, if available. When working with read replicas, it should be specified only if the source database specifies an instance in another AWS Region. See DBSubnetGroupName in API action CreateDBInstanceReadReplica for additional read replica contraints.
- `deletion_protection` - (Optional) If the DB instance should have deletion protection enabled. The database can’t be deleted when this value is set to `true`. The default is `false`.
- `domain` - (Optional) The ID of the Directory Service Active Directory

domain to create the instance in.

- `domain_iam_role_name` - (Optional, but required if domain is provided) The name of the IAM role to be used when making API calls to the Directory Service.
- `enabled_cloudwatch_logs_exports` - (Optional) List of log types to enable for exporting to CloudWatch logs. If omitted, no logs will be exported. Valid values (depending on `engine`): `alert`, `audit`, `error`, `general`, `listener`, `slowquery`, `trace`, `postgresql` (PostgreSQL), `upgrade` (PostgreSQL).
- `engine` - (Required unless a `snapshot_identifier` or `replicate_source_db` is provided) The database engine to use. For supported values, see the Engine parameter in API action CreateDBInstance. Note that for Amazon Aurora instances the engine must match the DB cluster's engine'. For information on the difference between the available Aurora MySQL engines see Comparison between Aurora MySQL 1 and Aurora MySQL 2 in the Amazon RDS User Guide.
- `engine_version` - (Optional) The engine version to use. If `auto_minor_version_upgrade` is enabled, you can provide a prefix of the version such as `5.7` (for `5.7.10`) and this attribute will ignore differences in the patch version automatically (e.g. `5.7.17`). For supported values, see the EngineVersion parameter in API action CreateDBInstance. Note that for Amazon Aurora instances the engine version must match the DB cluster's engine version'.
- `final_snapshot_identifier` - (Optional) The name of your final DB snapshot when this DB instance is deleted. Must be provided if `skip_final_snapshot` is set to `false`.
- `iam_database_authentication_enabled` - (Optional) Specifies whether or mappings of AWS Identity and Access Management (IAM) accounts to database accounts is enabled.
- `identifier` - (Optional, Forces new resource) The name of the RDS instance, if omitted, Terraform will assign a random, unique identifier.
- `identifier_prefix` - (Optional, Forces new resource) Creates a unique identifier beginning with the specified prefix. Conflicts with `identifier`.
- `instance_class` - (Required) The instance type of the RDS instance.
- `iops` - (Optional) The amount of provisioned IOPS. Setting this implies a storage_type of "io1".
- `kms_key_id` - (Optional) The ARN for the KMS encryption key. If creating an encrypted replica, set this to the destination KMS ARN.
- `license_model` - (Optional, but required for some DB engines, i.e. Oracle SE1) License model information for this DB instance.
- `maintenance_window` - (Optional) The window to perform maintenance in. Syntax: "ddd:hh24:mi-ddd:hh24:mi". Eg: "Mon:00:00-Mon:03:00". See RDS Maintenance Window docs for more information.
- `max_allocated_storage` - (Optional) When configured, the upper limit to which Amazon RDS can automatically scale the storage of the DB instance. Configuring this will automatically ignore differences to `allocated_storage`. Must be greater than or equal to

allocated_storage or 0 to disable Storage Autoscaling.

- monitoring_interval - (Optional) The interval, in seconds, between points when Enhanced Monitoring metrics are collected for the DB instance. To disable collecting Enhanced Monitoring metrics, specify 0. The default is 0. Valid Values: 0, 1, 5, 10, 15, 30, 60.
- monitoring_role_arn - (Optional) The ARN for the IAM role that permits RDS to send enhanced monitoring metrics to CloudWatch Logs. You can find more information on the AWS Documentation what IAM permissions are needed to allow Enhanced Monitoring for RDS Instances.
- multi_az - (Optional) Specifies if the RDS instance is multi-AZ
- name - (Optional) The name of the database to create when the DB instance is created. If this parameter is not specified, no database is created in the DB instance. Note that this does not apply for Oracle or SQL Server engines. See the AWS documentation for more details on what applies for those engines.
- option_group_name - (Optional) Name of the DB option group to associate.
- parameter_group_name - (Optional) Name of the DB parameter group to associate.
- password - (Required unless a snapshot_identifier or replicate_source_db is provided) Password for the master DB user. Note that this may show up in logs, and it will be stored in the state file.
- port - (Optional) The port on which the DB accepts connections.
- publicly_accessible - (Optional) Bool to control if instance is publicly accessible. Default is false.
- replicate_source_db - (Optional) Specifies that this resource is a Replicate database, and to use this value as the source database. This correlates to the identifier of another Amazon RDS Database to replicate. Note that if you are creating a cross-region replica of an encrypted database you will also need to specify a kms_key_id. See DB Instance Replication and Working with PostgreSQL and MySQL Read Replicas for more information on using Replication.
- security_group_names - (Optional/Deprecated) List of DB Security Groups to associate. Only used for DB Instances on the *EC2-Classic* Platform.
- skip_final_snapshot - (Optional) Determines whether a final DB snapshot is created before the DB instance is deleted. If true is specified, no DBSnapshot is created. If false is specified, a DB snapshot is created before the DB instance is deleted, using the value from final_snapshot_identifier. Default is false.
- snapshot_identifier - (Optional) Specifies whether or not to create this database from a snapshot. This correlates to the snapshot ID you'd find in the RDS console, e.g: rds:production-2015-06-26-06-05.
- storage_encrypted - (Optional) Specifies whether the DB instance is encrypted. Note that if you are creating a cross-region read replica this field is ignored and you should instead declare kms_key_id with a valid

ARN. The default is `false` if not specified.

- `storage_type` - (Optional) One of "standard" (magnetic), "gp2" (general purpose SSD), or "io1" (provisioned IOPS SSD). The default is "io1" if `iops` is specified, "gp2" if not.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `timezone` - (Optional) Time zone of the DB instance. `timezone` is currently only supported by Microsoft SQL Server. The `timezone` can only be set on creation. See MSSQL User Guide for more information.
- `username` - (Required unless a `snapshot_identifier` or `replicate_source_db` is provided) Username for the master DB user.
- `vpc_security_group_ids` - (Optional) List of VPC security groups to associate.
- `s3_import` - (Optional) Restore from a Percona Xtrabackup in S3. See Importing Data into an Amazon RDS MySQL DB Instance
- `performance_insights_enabled` - (Optional) Specifies whether Performance Insights are enabled. Defaults to false.
- `performance_insights_kms_key_id` - (Optional) The ARN for the KMS key to encrypt Performance Insights data. When specifying `performance_insights_kms_key_id`, `performance_insights_enabled` needs to be set to true. Once KMS key is set, it can never be changed.
- `performance_insights_retention_period` - (Optional) The amount of time in days to retain Performance Insights data. Either 7 (7 days) or 731 (2 years). When specifying `performance_insights_retention_period`, `performance_insights_enabled` needs to be set to true. Defaults to '7'.

NOTE: Removing the `replicate_source_db` attribute from an existing RDS Replicate database managed by Terraform will promote the database to a fully standalone database.

» S3 Import Options

Full details on the core parameters and impacts are in the API Docs: RestoreDBInstanceFromS3. Sample

```
resource "aws_db_instance" "db" {
  s3_import {
    source_engine         = "mysql"
    source_engine_version = "5.6"
    bucket_name           = "mybucket"
    bucket_prefix         = "backups"
    ingestion_role        = "arn:aws:iam::1234567890:role/role-xtrabackup-rds-restore"
  }
}
```

- `bucket_name` - (Required) The bucket name where your backup is stored
- `bucket_prefix` - (Optional) Can be blank, but is the path to your backup

- `ingestion_role` - (Required) Role applied to load the data.
- `source_engine` - (Required, as of Feb 2018 only 'mysql' supported) Source engine for the backup
- `source_engine_version` - (Required, as of Feb 2018 only '5.6' supported) Version of the source engine used to make the backup

This will not recreate the resource if the S3 object changes in some way. It's only used to initialize the database

» Timeouts

`aws_db_instance` provides the following Timeouts configuration options:

- `create` - (Default `40 minutes`) Used for Creating Instances, Replicas, and restoring from Snapshots.
- `update` - (Default `80 minutes`) Used for Database modifications.
- `delete` - (Default `40 minutes`) Used for destroying databases. This includes the time required to take snapshots.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `address` - The hostname of the RDS instance. See also `endpoint` and `port`.
- `arn` - The ARN of the RDS instance.
- `allocated_storage` - The amount of allocated storage.
- `availability_zone` - The availability zone of the instance.
- `backup_retention_period` - The backup retention period.
- `backup_window` - The backup window.
- `ca_cert_identifier` - Specifies the identifier of the CA certificate for the DB instance.
- `domain` - The ID of the Directory Service Active Directory domain the instance is joined to
- `domain_iam_role_name` - The name of the IAM role to be used when making API calls to the Directory Service.
- `endpoint` - The connection endpoint in `address:port` format.
- `engine` - The database engine.
- `engine_version` - The database engine version.
- `hosted_zone_id` - The canonical hosted zone ID of the DB instance (to be used in a Route 53 Alias record).
- `id` - The RDS instance ID.
- `instance_class`- The RDS instance class.
- `maintenance_window` - The instance maintenance window.
- `multi_az` - If the RDS instance is multi AZ enabled.

- `name` - The database name.
- `port` - The database port.
- `resource_id` - The RDS Resource ID of this instance.
- `status` - The RDS instance status.
- `storage_encrypted` - Specifies whether the DB instance is encrypted.
- `username` - The master username for the database.

On Oracle instances the following is exported additionally:

- `character_set_name` - The character set used on Oracle instances.

» Import

DB Instances can be imported using the `identifier`, e.g.

```
$ terraform import aws_db_instance.default mydb-rds-instance
```

» Resource: aws_db_instance_role_association

Manages a RDS DB Instance association with an IAM Role. Example use cases:

- Amazon RDS Oracle integration with Amazon S3
- Importing Amazon S3 Data into an RDS PostgreSQL DB Instance

To manage the RDS DB Instance IAM Role for Enhanced Monitoring, see the `aws_db_instance` resource `monitoring_role_arn` argument instead.

» Example Usage

```
resource "aws_db_instance_role_association" "example" {
  db_instance_identifier = "${aws_db_instance.example.id}"
  feature_name           = "S3_INTEGRATION"
  role_arn               = "${aws_iam_role.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `db_instance_identifier` - (Required) DB Instance Identifier to associate with the IAM Role.
- `feature_name` - (Required) Name of the feature for association. This can be found in the AWS documentation relevant to the integration or a full list is available in the `SupportedFeatureNames` list returned by AWS CLI rds describe-db-engine-versions.

- `role_arn` - (Required) Amazon Resource Name (ARN) of the IAM Role to associate with the DB Instance.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - DB Instance Identifier and IAM Role ARN separated by a comma (`,`)

» Import

`aws_db_instance_role_association` can be imported using the DB Instance Identifier and IAM Role ARN separated by a comma (`,`), e.g.

```
$ terraform import aws_db_instance_role_association.example my-db-instance,arn:aws:iam::1234
```

» Resource: aws_db_option_group

Provides an RDS DB option group resource. Documentation of the available options for various RDS engines can be found at: * MariaDB Options * Microsoft SQL Server Options * MySQL Options * Oracle Options

» Example Usage

```
resource "aws_db_option_group" "example" {
  name                     = "option-group-test-terraform"
  option_group_description = "Terraform Option Group"
  engine_name              = "sqlserver-ee"
  major_engine_version     = "11.00"

  option {
    option_name = "Timezone"

    option_settings {
      name  = "TIME_ZONE"
      value = "UTC"
    }
  }

  option {
    option_name = "SQLSERVER_BACKUP_RESTORE"
    option_settings {
      name  = "IAM_ROLE_ARN"
      value = "${aws_iam_role.example.arn}"
    }
  }

  option {
    option_name = "TDE"
  }
}
```

Note: Any modifications to the `db_option_group` are set to happen immediately as we default to applying immediately.

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the option group. If omitted, Terraform will assign a random, unique name. Must be lowercase, to match as it is stored in AWS.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`. Must be lowercase, to match as it is stored in AWS.
- `option_group_description` - (Optional) The description of the option group. Defaults to ”Managed by Terraform”.
- `engine_name` - (Required) Specifies the name of the engine that this option group should be associated with.
- `major_engine_version` - (Required) Specifies the major version of the engine that this option group should be associated with.
- `option` - (Optional) A list of Options to apply.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Option blocks support the following:

- `option_name` - (Required) The Name of the Option (e.g. MEMCACHED).
- `option_settings` - (Optional) A list of option settings to apply.
- `port` - (Optional) The Port number when connecting to the Option (e.g. 11211).
- `version` - (Optional) The version of the option (e.g. 13.1.0.0).
- `db_security_group_memberships` - (Optional) A list of DB Security Groups for which the option is enabled.
- `vpc_security_group_memberships` - (Optional) A list of VPC Security Groups for which the option is enabled.

Option Settings blocks support the following:

- `name` - (Optional) The Name of the setting.
- `value` - (Optional) The Value of the setting.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The db option group name.
- `arn` - The ARN of the db option group.

» Timeouts

`aws_db_option_group` provides the following Timeouts configuration options:

- `delete` - (Default `15 minutes`)

» Import

DB Option groups can be imported using the `name`, e.g.

```
$ terraform import aws_db_option_group.bar mysql-option-group
```

» Resource: aws_db_parameter_group

Provides an RDS DB parameter group resource .Documentation of the available parameters for various RDS engines can be found at: * Aurora MySQL Parameters * Aurora PostgreSQL Parameters * MariaDB Parameters * Oracle Parameters * PostgreSQL Parameters

» Example Usage

```
resource "aws_db_parameter_group" "default" {
  name   = "rds-pg"
  family = "mysql5.6"

  parameter {
    name  = "character_set_server"
    value = "utf8"
  }

  parameter {
    name  = "character_set_client"
    value = "utf8"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the DB parameter group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `family` - (Required) The family of the DB parameter group.
- `description` - (Optional) The description of the DB parameter group. Defaults to ”Managed by Terraform”.
- `parameter` - (Optional) A list of DB parameters to apply. Note that parameters may differ from a family to an other. Full list of all parameters can be discovered via `aws rds describe-db-parameters` after initial creation of the group.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Parameter blocks support the following:

- `name` - (Required) The name of the DB parameter.
- `value` - (Required) The value of the DB parameter.
- `apply_method` - (Optional) ”immediate” (default), or ”pending-reboot”. Some engines can't apply some parameters without a reboot, and you will need to specify ”pending-reboot” here.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The db parameter group name.
- `arn` - The ARN of the db parameter group.

» Import

DB Parameter groups can be imported using the `name`, e.g.

```
$ terraform import aws_db_parameter_group.rds_pg rds-pg
```

» Resource: aws_db_security_group

Provides an RDS security group resource. This is only for DB instances in the EC2-Classic Platform. For instances inside a VPC, use the `aws_db_instance.vpc_security_group_ids` attribute instead.

» Example Usage

```
resource "aws_db_security_group" "default" {
  name = "rds_sg"

  ingress {
    cidr = "10.0.0.0/24"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the DB security group.
- `description` - (Optional) The description of the DB security group. Defaults to "Managed by Terraform".
- `ingress` - (Required) A list of ingress rules.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Ingress blocks support the following:

- `cidr` - The CIDR block to accept
- `security_group_name` - The name of the security group to authorize
- `security_group_id` - The ID of the security group to authorize
- `security_group_owner_id` - The owner Id of the security group provided by `security_group_name`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The db security group ID.
- `arn` - The arn of the DB security group.

» Import

DB Security groups can be imported using the `name`, e.g.

```
$ terraform import aws_db_security_group.default aws_rds_sg-1
```

» Resource: aws_db_snapshot

Manages a RDS database instance snapshot. For managing RDS database cluster snapshots, see the `aws_db_cluster_snapshot` resource.

» Example Usage

```
resource "aws_db_instance" "bar" {
  allocated_storage = 10
  engine            = "MySQL"
  engine_version    = "5.6.21"
  instance_class    = "db.t2.micro"
  name              = "baz"
  password          = "barbarbarbar"
  username          = "foo"

  maintenance_window      = "Fri:09:00-Fri:09:30"
  backup_retention_period = 0
  parameter_group_name    = "default.mysql5.6"
}

resource "aws_db_snapshot" "test" {
  db_instance_identifier = "${aws_db_instance.bar.id}"
  db_snapshot_identifier = "testsnapshot1234"
}
```

» Argument Reference

The following arguments are supported:

- `db_instance_identifier` - (Required) The DB Instance Identifier from which to take the snapshot.
- `db_snapshot_identifier` - (Required) The Identifier for the snapshot.
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `allocated_storage` - Specifies the allocated storage size in gigabytes (GB).

- availability_zone - Specifies the name of the Availability Zone the DB instance was located in at the time of the DB snapshot.
- db_snapshot_arn - The Amazon Resource Name (ARN) for the DB snapshot.
- encrypted - Specifies whether the DB snapshot is encrypted.
- engine - Specifies the name of the database engine.
- engine_version - Specifies the version of the database engine.
- iops - Specifies the Provisioned IOPS (I/O operations per second) value of the DB instance at the time of the snapshot.
- kms_key_id - The ARN for the KMS encryption key.
- license_model - License model information for the restored DB instance.
- option_group_name - Provides the option group name for the DB snapshot.
- source_db_snapshot_identifier - The DB snapshot Arn that the DB snapshot was copied from. It only has value in case of cross customer or cross region copy.
- source_region - The region that the DB snapshot was created in or copied from.
- status - Specifies the status of this DB snapshot.
- storage_type - Specifies the storage type associated with DB snapshot.
- vpc_id - Specifies the storage type associated with DB snapshot.

» Resource: aws_db_subnet_group

Provides an RDS DB subnet group resource.

» Example Usage

```
resource "aws_db_subnet_group" "default" {
  name       = "main"
  subnet_ids = ["${aws_subnet.frontend.id}", "${aws_subnet.backend.id}"]

  tags = {
    Name = "My DB subnet group"
  }
}
```

» Argument Reference

The following arguments are supported:

- name - (Optional, Forces new resource) The name of the DB subnet group. If omitted, Terraform will assign a random, unique name.

- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `description` - (Optional) The description of the DB subnet group. Defaults to "Managed by Terraform".
- `subnet_ids` - (Required) A list of VPC subnet IDs.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The db subnet group name.
- `arn` - The ARN of the db subnet group.

» Import

DB Subnet groups can be imported using the `name`, e.g.

```
$ terraform import aws_db_subnet_group.default production-subnet-group
```

» Resource: aws_rds_cluster

Manages a RDS Aurora Cluster. To manage cluster instances that inherit configuration from the cluster (when not running the cluster in `serverless` engine mode), see the `aws_rds_cluster_instance` resource. To manage non-Aurora databases (e.g. MySQL, PostgreSQL, SQL Server, etc.), see the `aws_db_instance` resource.

For information on the difference between the available Aurora MySQL engines see Comparison between Aurora MySQL 1 and Aurora MySQL 2 in the Amazon RDS User Guide.

Changes to a RDS Cluster can occur when you manually change a parameter, such as `port`, and are reflected in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because a modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately (see documentation below).

Note: using `apply_immediately` can result in a brief downtime as the server reboots. See the AWS Docs on RDS Maintenance for more information.

Note: All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

» Aurora MySQL 2.x (MySQL 5.7)

```
resource "aws_rds_cluster" "default" {
  cluster_identifier      = "aurora-cluster-demo"
  engine                  = "aurora-mysql"
  engine_version          = "5.7.mysql_aurora.2.03.2"
  availability_zones      = ["us-west-2a", "us-west-2b", "us-west-2c"]
  database_name           = "mydb"
  master_username         = "foo"
  master_password         = "bar"
  backup_retention_period = 5
  preferred_backup_window = "07:00-09:00"
}
```

» Aurora MySQL 1.x (MySQL 5.6)

```
resource "aws_rds_cluster" "default" {
  cluster_identifier      = "aurora-cluster-demo"
  availability_zones      = ["us-west-2a", "us-west-2b", "us-west-2c"]
  database_name           = "mydb"
  master_username         = "foo"
  master_password         = "bar"
  backup_retention_period = 5
  preferred_backup_window = "07:00-09:00"
}
```

» Aurora with PostgreSQL engine

```
resource "aws_rds_cluster" "postgresql" {
  cluster_identifier      = "aurora-cluster-demo"
  engine                  = "aurora-postgresql"
  availability_zones      = ["us-west-2a", "us-west-2b", "us-west-2c"]
  database_name           = "mydb"
  master_username         = "foo"
  master_password         = "bar"
  backup_retention_period = 5
  preferred_backup_window = "07:00-09:00"
}
```

» Aurora Multi-Master Cluster

More information about Aurora Multi-Master Clusters can be found in the RDS User Guide.

```
resource "aws_rds_cluster" "example" {
  cluster_identifier   = "example"
  db_subnet_group_name = "${aws_db_subnet_group.example.name}"
  engine_mode          = "multimaster"
  master_password      = "barbarbarbar"
  master_username      = "foo"
  skip_final_snapshot  = true
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- `cluster_identifier` - (Optional, Forces new resources) The cluster identifier. If omitted, Terraform will assign a random, unique identifier.
- `cluster_identifier_prefix` - (Optional, Forces new resource) Creates a unique cluster identifier beginning with the specified prefix. Conflicts with `cluster_identifier`.
- `copy_tags_to_snapshot` – (Optional, boolean) Copy all Cluster `tags` to snapshots. Default is `false`.
- `database_name` - (Optional) Name for an automatically created database on cluster creation. There are different naming restrictions per database engine: RDS Naming Constraints
- `deletion_protection` - (Optional) If the DB instance should have deletion protection enabled. The database can't be deleted when this value is set to `true`. The default is `false`.
- `master_password` - (Required unless a `snapshot_identifier` or `global_cluster_identifier` is provided) Password for the master DB user. Note that this may show up in logs, and it will be stored in the state file. Please refer to the RDS Naming Constraints
- `master_username` - (Required unless a `snapshot_identifier` or `global_cluster_identifier` is provided) Username for the master DB user. Please refer to the RDS Naming Constraints. This argument does not support in-place updates and cannot be changed during a restore from snapshot.
- `final_snapshot_identifier` - (Optional) The name of your final DB snapshot when this DB cluster is deleted. If omitted, no final snapshot will be made.

- `skip_final_snapshot` - (Optional) Determines whether a final DB snapshot is created before the DB cluster is deleted. If true is specified, no DB snapshot is created. If false is specified, a DB snapshot is created before the DB cluster is deleted, using the value from `final_snapshot_identifier`. Default is `false`.
- `availability_zones` - (Optional) A list of EC2 Availability Zones for the DB cluster storage where DB cluster instances can be created. RDS automatically assigns 3 AZs if less than 3 AZs are configured, which will show as a difference requiring resource recreation next Terraform apply. It is recommended to specify 3 AZs or use the `lifecycle` configuration block `ignore_changes` argument if necessary.
- `backtrack_window` - (Optional) The target backtrack window, in seconds. Only available for `aurora` engine currently. To disable backtracking, set this value to `0`. Defaults to `0`. Must be between `0` and `259200` (72 hours)
- `backup_retention_period` - (Optional) The days to retain backups for. Default `1`
- `preferred_backup_window` - (Optional) The daily time range during which automated backups are created if automated backups are enabled using the BackupRetentionPeriod parameter.Time in UTC Default: A 30-minute window selected at random from an 8-hour block of time per region. e.g. 04:00-09:00
- `preferred_maintenance_window` - (Optional) The weekly time range during which system maintenance can occur, in (UTC) e.g. wed:04:00-wed:04:30
- `port` - (Optional) The port on which the DB accepts connections
- `vpc_security_group_ids` - (Optional) List of VPC security groups to associate with the Cluster
- `snapshot_identifier` - (Optional) Specifies whether or not to create this cluster from a snapshot. You can use either the name or ARN when specifying a DB cluster snapshot, or the ARN when specifying a DB snapshot.
- `global_cluster_identifier` - (Optional) The global cluster identifier specified on `aws_rds_global_cluster`.
- `storage_encrypted` - (Optional) Specifies whether the DB cluster is encrypted. The default is `false` for `provisioned` `engine_mode` and `true` for `serverless` `engine_mode`.
- `replication_source_identifier` - (Optional) ARN of a source DB cluster or DB instance if this DB cluster is to be created as a Read Replica.
- `apply_immediately` - (Optional) Specifies whether any cluster modifications are applied immediately, or during the next maintenance window. Default is `false`. See Amazon RDS Documentation for more information.
- `db_subnet_group_name` - (Optional) A DB subnet group to associate with this DB instance. **NOTE:** This must match the `db_subnet_group_name` specified on every `aws_rds_cluster_instance` in the cluster.
- `db_cluster_parameter_group_name` - (Optional) A cluster parameter group to associate with the cluster.

- `kms_key_id` - (Optional) The ARN for the KMS encryption key. When specifying `kms_key_id`, `storage_encrypted` needs to be set to true.
- `iam_roles` - (Optional) A List of ARNs for the IAM roles to associate to the RDS Cluster.
- `iam_database_authentication_enabled` - (Optional) Specifies whether or mappings of AWS Identity and Access Management (IAM) accounts to database accounts is enabled. Please see AWS Documentation for availability and limitations.
- `engine` - (Optional) The name of the database engine to be used for this DB cluster. Defaults to `aurora`. Valid Values: `aurora`, `aurora-mysql`, `aurora-postgresql`
- `engine_mode` - (Optional) The database engine mode. Valid values: `global`, `multimaster`, `parallelquery`, `provisioned`, `serverless`. Defaults to: `provisioned`. See the RDS User Guide for limitations when using `serverless`.
- `engine_version` - (Optional) The database engine version. Updating this argument results in an outage. See the Aurora MySQL and Aurora Postgres documentation for your configured engine to determine this value. For example with Aurora MySQL 2, a potential value for this argument is `5.7.mysql_aurora.2.03.2`.
- `source_region` - (Optional) The source region for an encrypted replica DB cluster.
- `enabled_cloudwatch_logs_exports` - (Optional) List of log types to export to cloudwatch. If omitted, no logs will be exported. The following log types are supported: `audit`, `error`, `general`, `slowquery`, `postgresql` (PostgreSQL).
- `scaling_configuration` - (Optional) Nested attribute with scaling properties. Only valid when `engine_mode` is set to `serverless`. More details below.
- `enable_http_endpoint` - (Optional) Enable HTTP endpoint (data API). Only valid when `engine_mode` is set to `serverless`.
- `tags` - (Optional) A mapping of tags to assign to the DB cluster.

» S3 Import Options

Full details on the core parameters and impacts are in the API Docs: RestoreDBClusterFromS3. Requires that the S3 bucket be in the same region as the RDS cluster you're trying to create. Sample:

NOTE: RDS Aurora Serverless does not support loading data from S3, so its not possible to directly use `engine_mode` set to `serverless` with `s3_import`.

```
resource "aws_rds_cluster" "db" {
  engine = "aurora"

  s3_import {
    source_engine         = "mysql"
    source_engine_version = "5.6"
    bucket_name           = "mybucket"
    bucket_prefix         = "backups"
    ingestion_role        = "arn:aws:iam::1234567890:role/role-xtrabackup-rds-restore"
  }
}
```

- `bucket_name` - (Required) The bucket name where your backup is stored
- `bucket_prefix` - (Optional) Can be blank, but is the path to your backup
- `ingestion_role` - (Required) Role applied to load the data.
- `source_engine` - (Required) Source engine for the backup
- `source_engine_version` - (Required) Version of the source engine used to make the backup

This will not recreate the resource if the S3 object changes in some way. It's only used to initialize the database. This only works currently with the aurora engine. See AWS for currently supported engines and options. See Aurora S3 Migration Docs.

» scaling_configuration Argument Reference

NOTE: `scaling_configuration` configuration is only valid when `engine_mode` is set to `serverless`.

Example:

```
resource "aws_rds_cluster" "example" {
  # ... other configuration ...

  engine_mode = "serverless"

  scaling_configuration {
    auto_pause               = true
    max_capacity             = 256
    min_capacity             = 2
    seconds_until_auto_pause = 300
    timeout_action           = "ForceApplyCapacityChange"
  }
}
```

- `auto_pause` - (Optional) Whether to enable automatic pause. A DB cluster can be paused only when it's idle (it has no connections). If a DB cluster is paused for more than seven days, the DB cluster might be backed up with a snapshot. In this case, the DB cluster is restored when there is a request to connect to it. Defaults to `true`.

- `max_capacity` - (Optional) The maximum capacity. The maximum capacity must be greater than or equal to the minimum capacity. Valid capacity values are `1`, `2`, `4`, `8`, `16`, `32`, `64`, `128`, and `256`. Defaults to `16`.
- `min_capacity` - (Optional) The minimum capacity. The minimum capacity must be lesser than or equal to the maximum capacity. Valid capacity values are `1`, `2`, `4`, `8`, `16`, `32`, `64`, `128`, and `256`. Defaults to `2`.
- `seconds_until_auto_pause` - (Optional) The time, in seconds, before an Aurora DB cluster in serverless mode is paused. Valid values are `300` through `86400`. Defaults to `300`.
- `timeout_action` - (Optional) The action to take when the timeout is reached. Valid values: `ForceApplyCapacityChange`, `RollbackCapacityChange`. Defaults to `RollbackCapacityChange`. See documentation.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of cluster
- `id` - The RDS Cluster Identifier
- `cluster_identifier` - The RDS Cluster Identifier
- `cluster_resource_id` - The RDS Cluster Resource ID
- `cluster_members` – List of RDS Instances that are a part of this cluster
- `allocated_storage` - The amount of allocated storage
- `availability_zones` - The availability zone of the instance
- `backup_retention_period` - The backup retention period
- `preferred_backup_window` - The daily time range during which the backups happen
- `preferred_maintenance_window` - The maintenance window
- `endpoint` - The DNS address of the RDS instance
- `reader_endpoint` - A read-only endpoint for the Aurora cluster, automatically load-balanced across replicas
- `engine` - The database engine
- `engine_version` - The database engine version
- `maintenance_window` - The instance maintenance window
- `database_name` - The database name
- `port` - The database port
- `status` - The RDS instance status
- `master_username` - The master username for the database
- `storage_encrypted` - Specifies whether the DB cluster is encrypted
- `replication_source_identifier` - ARN of the source DB cluster or DB instance if this DB cluster is created as a Read Replica.
- `hosted_zone_id` - The Route53 Hosted Zone ID of the endpoint

» Timeouts

`aws_rds_cluster` provides the following Timeouts configuration options:

- `create` - (Default `120 minutes`) Used for Cluster creation
- `update` - (Default `120 minutes`) Used for Cluster modifications
- `delete` - (Default `120 minutes`) Used for destroying cluster. This includes any cleanup task during the destroying process.

» Import

RDS Clusters can be imported using the `cluster_identifier`, e.g.

```
$ terraform import aws_rds_cluster.aurora_cluster aurora-prod-cluster
```

» Resource: aws_rds_cluster_endpoint

Manages a RDS Aurora Cluster Endpoint. You can refer to the User Guide.

» Example Usage

```
resource "aws_rds_cluster" "default" {
  cluster_identifier      = "aurora-cluster-demo"
  availability_zones      = ["us-west-2a", "us-west-2b", "us-west-2c"]
  database_name           = "mydb"
  master_username         = "foo"
  master_password         = "bar"
  backup_retention_period = 5
  preferred_backup_window = "07:00-09:00"
}

resource "aws_rds_cluster_instance" "test1" {
  apply_immediately  = true
  cluster_identifier = "${aws_rds_cluster.default.id}"
  identifier         = "test1"
  instance_class     = "db.t2.small"
}

resource "aws_rds_cluster_instance" "test2" {
  apply_immediately  = true
  cluster_identifier = "${aws_rds_cluster.default.id}"
  identifier         = "test2"
  instance_class     = "db.t2.small"
}

resource "aws_rds_cluster_instance" "test3" {
  apply_immediately  = true
  cluster_identifier = "${aws_rds_cluster.default.id}"
  identifier         = "test3"
  instance_class     = "db.t2.small"
}

resource "aws_rds_cluster_endpoint" "eligible" {
  cluster_identifier          = "${aws_rds_cluster.default.id}"
  cluster_endpoint_identifier = "reader"
  custom_endpoint_type        = "READER"

  excluded_members = [
    "${aws_rds_cluster_instance.test1.id}",
    "${aws_rds_cluster_instance.test2.id}",
  ]
}

resource "aws_rds_cluster_endpoint" "static" {
  cluster_identifier          = "${aws_rds_cluster.default.id}"
  cluster_endpoint_identifier = "static"
  custom_endpoint_type        = "READER"

  static_members = [
    "${aws_rds_cluster_instance.test1.id}",
    "${aws_rds_cluster_instance.test3.id}",
  ]
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- `cluster_identifier` - (Required, Forces new resources) The cluster identifier.
- `cluster_endpoint_identifier` - (Required, Forces new resources) The identifier to use for the new endpoint. This parameter is stored as a lowercase string.
- `custom_endpoint_type` - (Required) The type of the endpoint. One of: READER , ANY .

- `static_members` - (Optional) List of DB instance identifiers that are part of the custom endpoint group. Conflicts with `excluded_members`.
- `excluded_members` - (Optional) List of DB instance identifiers that aren't part of the custom endpoint group. All other eligible instances are reachable through the custom endpoint. Only relevant if the list of static members is empty. Conflicts with `static_members`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of cluster
- `id` - The RDS Cluster Endpoint Identifier
- `endpoint` - A custom endpoint for the Aurora cluster

» Import

RDS Clusters Endpoint can be imported using the `cluster_endpoint_identifier`, e.g.

```
$ terraform import aws_rds_cluster_endpoint.custom_reader aurora-prod-cluster-custom-reader
```

» Resource: aws_rds_cluster_instance

Provides an RDS Cluster Instance Resource. A Cluster Instance Resource defines attributes that are specific to a single instance in a RDS Cluster, specifically running Amazon Aurora.

Unlike other RDS resources that support replication, with Amazon Aurora you do not designate a primary and subsequent replicas. Instead, you simply add RDS Instances and Aurora manages the replication. You can use the count meta-parameter to make multiple instances and join them all to the same RDS Cluster, or you may specify different Cluster Instance resources with various `instance_class` sizes.

For more information on Amazon Aurora, see Aurora on Amazon RDS in the Amazon RDS User Guide.

NOTE: Deletion Protection from the RDS service can only be enabled at the cluster level, not for individual cluster instances. You can still add the `prevent_destroy` lifecycle behavior to your Terraform resource configuration if you desire protection from accidental deletion.

» Example Usage

```
resource "aws_rds_cluster_instance" "cluster_instances" {
  count              = 2
  identifier         = "aurora-cluster-demo-${count.index}"
  cluster_identifier = "${aws_rds_cluster.default.id}"
  instance_class     = "db.r4.large"
}

resource "aws_rds_cluster" "default" {
  cluster_identifier = "aurora-cluster-demo"
  availability_zones = ["us-west-2a", "us-west-2b", "us-west-2c"]
  database_name      = "mydb"
  master_username    = "foo"
  master_password    = "barbut8chars"
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- `identifier` - (Optional, Forces new resource) The identifier for the RDS instance, if omitted, Terraform will assign a random, unique identifier.
- `identifier_prefix` - (Optional, Forces new resource) Creates a unique identifier beginning with the specified prefix. Conflicts with `identifier`.
- `cluster_identifier` - (Required) The identifier of the `aws_rds_cluster` in which to launch this instance.
- `engine` - (Optional) The name of the database engine to be used for the RDS instance. Defaults to `aurora`. Valid Values: `aurora`, `aurora-mysql`, `aurora-postgresql`. For information on the difference between the available Aurora MySQL engines see Comparison between Aurora MySQL 1 and Aurora MySQL 2 in the Amazon RDS User Guide.
- `engine_version` - (Optional) The database engine version.
- `instance_class` - (Required) The instance class to use. For details on CPU and memory, see Scaling Aurora DB Instances. Aurora uses `db.*` instance classes/types. Please see AWS Documentation for currently available instance classes and complete details.
- `publicly_accessible` - (Optional) Bool to control if instance is publicly accessible. Default `false`. See the documentation on Creating DB Instances for more details on controlling this property.
- `db_subnet_group_name` - (Required if `publicly_accessible = false`, Optional otherwise) A DB subnet group to associate with this DB instance.

NOTE: This must match the `db_subnet_group_name` of the attached `aws_rds_cluster`.

- `db_parameter_group_name` - (Optional) The name of the DB parameter group to associate with this instance.
- `apply_immediately` - (Optional) Specifies whether any database modifications are applied immediately, or during the next maintenance window. Default is`false`.
- `monitoring_role_arn` - (Optional) The ARN for the IAM role that permits RDS to send enhanced monitoring metrics to CloudWatch Logs. You can find more information on the AWS Documentation what IAM permissions are needed to allow Enhanced Monitoring for RDS Instances.
- `monitoring_interval` - (Optional) The interval, in seconds, between points when Enhanced Monitoring metrics are collected for the DB instance. To disable collecting Enhanced Monitoring metrics, specify 0. The default is 0. Valid Values: 0, 1, 5, 10, 15, 30, 60.
- `promotion_tier` - (Optional) Default 0. Failover Priority setting on instance level. The reader who has lower tier has higher priority to get promoted to writer.
- `availability_zone` - (Optional, Computed) The EC2 Availability Zone that the DB instance is created in. See docs about the details.
- `preferred_backup_window` - (Optional) The daily time range during which automated backups are created if automated backups are enabled. Eg: "04:00-09:00"
- `preferred_maintenance_window` - (Optional) The window to perform maintenance in. Syntax: "ddd:hh24:mi-ddd:hh24:mi". Eg: "Mon:00:00-Mon:03:00".
- `auto_minor_version_upgrade` - (Optional) Indicates that minor engine upgrades will be applied automatically to the DB instance during the maintenance window. Default `true`.
- `performance_insights_enabled` - (Optional) Specifies whether Performance Insights is enabled or not.
- `performance_insights_kms_key_id` - (Optional) The ARN for the KMS key to encrypt Performance Insights data. When specifying `performance_insights_kms_key_id`, `performance_insights_enabled` needs to be set to true.
- `copy_tags_to_snapshot` – (Optional, boolean) Indicates whether to copy all of the user-defined tags from the DB instance to snapshots of the DB instance. Default `false`.
- `ca_cert_identifier` - (Optional) The identifier of the CA certificate for the DB instance.
- `tags` - (Optional) A mapping of tags to assign to the instance.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of cluster instance
- `cluster_identifier` - The RDS Cluster Identifier
- `identifier` - The Instance identifier
- `id` - The Instance identifier
- `writer` – Boolean indicating if this instance is writable. `False` indicates this instance is a read replica.
- `allocated_storage` - The amount of allocated storage
- `availability_zone` - The availability zone of the instance
- `endpoint` - The DNS address for this instance. May not be writable
- `engine` - The database engine
- `engine_version` - The database engine version
- `database_name` - The database name
- `port` - The database port
- `status` - The RDS instance status
- `storage_encrypted` - Specifies whether the DB cluster is encrypted.
- `kms_key_id` - The ARN for the KMS encryption key if one is set to the cluster.
- `dbi_resource_id` - The region-unique, immutable identifier for the DB instance.
- `performance_insights_enabled` - Specifies whether Performance Insights is enabled or not.
- `performance_insights_kms_key_id` - The ARN for the KMS encryption key used by Performance Insights.

» Timeouts

`aws_rds_cluster_instance` provides the following Timeouts configuration options:

- `create` - (Default `90 minutes`) Used for Creating Instances, Replicas, and restoring from Snapshots
- `update` - (Default `90 minutes`) Used for Database modifications
- `delete` - (Default `90 minutes`) Used for destroying databases. This includes the time required to take snapshots

» Import

RDS Cluster Instances can be imported using the `identifier`, e.g.

```
$ terraform import aws_rds_cluster_instance.prod_instance_1 aurora-cluster-instance-1
```

» Resource: aws_rds_cluster_parameter_group

Provides an RDS DB cluster parameter group resource. Documentation of the available parameters for various Aurora engines can be found at:

- Aurora MySQL Parameters
- Aurora PostgreSQL Parameters

» Example Usage

```
resource "aws_rds_cluster_parameter_group" "default" {
  name        = "rds-cluster-pg"
  family      = "aurora5.6"
  description = "RDS default cluster parameter group"

  parameter {
    name  = "character_set_server"
    value = "utf8"
  }

  parameter {
    name  = "character_set_client"
    value = "utf8"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the DB cluster parameter group. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `family` - (Required) The family of the DB cluster parameter group.
- `description` - (Optional) The description of the DB cluster parameter group. Defaults to ”Managed by Terraform”.
- `parameter` - (Optional) A list of DB parameters to apply. Note that parameters may differ from a family to an other. Full list of all parameters can be discovered via `aws rds describe-db-cluster-parameters` after initial creation of the group.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Parameter blocks support the following:

- `name` - (Required) The name of the DB parameter.

- `value` - (Required) The value of the DB parameter.
- `apply_method` - (Optional) "immediate" (default), or "pending-reboot". Some engines can't apply some parameters without a reboot, and you will need to specify "pending-reboot" here.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The db cluster parameter group name.
- `arn` - The ARN of the db cluster parameter group.

» Import

RDS Cluster Parameter Groups can be imported using the `name`, e.g.

```
$ terraform import aws_rds_cluster_parameter_group.cluster_pg production-pg-1
```

» Resource: aws_rds_global_cluster

Manages a RDS Global Cluster, which is an Aurora global database spread across multiple regions. The global database contains a single primary cluster with read-write capability, and a read-only secondary cluster that receives data from the primary cluster through high-speed replication performed by the Aurora storage subsystem.

More information about Aurora global databases can be found in the Aurora User Guide.

NOTE: RDS only supports the `aurora` engine (MySQL 5.6 compatible) for Global Clusters at this time.

» Example Usage

```
provider "aws" {
  alias  = "primary"
  region = "us-east-2"
}

provider "aws" {
  alias  = "secondary"
  region = "us-west-2"
}
```

```
resource "aws_rds_global_cluster" "example" {
  provider = "aws.primary"

  global_cluster_identifier = "example"
}

resource "aws_rds_cluster" "primary" {
  provider = "aws.primary"

  # ... other configuration ...
  engine_mode               = "global"
  global_cluster_identifier = "${aws_rds_global_cluster.example.id}"
}

resource "aws_rds_cluster_instance" "primary" {
  provider = "aws.primary"

  # ... other configuration ...
  cluster_identifier = "${aws_rds_cluster.primary.id}"
}

resource "aws_rds_cluster" "secondary" {
  depends_on = ["aws_rds_cluster_instance.primary"]
  provider   = "aws.secondary"

  # ... other configuration ...
  engine_mode               = "global"
  global_cluster_identifier = "${aws_rds_global_cluster.example.id}"
}

resource "aws_rds_cluster_instance" "secondary" {
  provider = "aws.secondary"

  # ... other configuration ...
  cluster_identifier = "${aws_rds_cluster.secondary.id}"
}
```

» Argument Reference

The following arguments are supported:

- `global_cluster_identifier` - (Required, Forces new resources) The global cluster identifier.
- `database_name` - (Optional, Forces new resources) Name for an automat-

ically created database on cluster creation.

- `deletion_protection` - (Optional) If the Global Cluster should have deletion protection enabled. The database can't be deleted when this value is set to `true`. The default is `false`.
- `engine` - (Optional, Forces new resources) Name of the database engine to be used for this DB cluster. Valid values: `aurora`. Defaults to `aurora`.
- `engine_version` - (Optional, Forces new resources) Engine version of the Aurora global database.
- `storage_encrypted` - (Optional, Forces new resources) Specifies whether the DB cluster is encrypted. The default is `false`.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - RDS Global Cluster Amazon Resource Name (ARN)
- `global_cluster_resource_id` - AWS Region-unique, immutable identifier for the global database cluster. This identifier is found in AWS CloudTrail log entries whenever the AWS KMS key for the DB cluster is accessed
- `id` - RDS Global Cluster identifier

» Import

`aws_rds_global_cluster` can be imported by using the RDS Global Cluster identifier, e.g.

```
$ terraform import aws_rds_global_cluster.example example
```

» Data Source: aws_redshift_cluster

Provides details about a specific redshift cluster.

» Example Usage

```
data "aws_redshift_cluster" "test_cluster" {
  cluster_identifier = "test-cluster"
}

resource "aws_kinesis_firehose_delivery_stream" "test_stream" {
  name        = "terraform-kinesis-firehose-test-stream"
  destination = "redshift"
  s3_configuration {
    role_arn           = "${aws_iam_role.firehose_role.arn}"
    bucket_arn         = "${aws_s3_bucket.bucket.arn}"
    buffer_size        = 10
    buffer_interval    = 400
    compression_format = "GZIP"
  }

  redshift_configuration {
    role_arn           = "${aws_iam_role.firehose_role.arn}"
    cluster_jdbcurl    = "jdbc:redshift://${data.aws_redshift_cluster.test_cluster.endpoint}
    username           = "testuser"
    password           = "T3stPass"
    data_table_name    = "test-table"
    copy_options       = "delimiter '|'" # the default delimiter
    data_table_columns = "test-col"
  }
}
```

» Argument Reference

The following arguments are supported:

- `cluster_identifier` - (Required) The cluster identifier

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `allow_version_upgrade` - Whether major version upgrades can be applied during maintenance period
- `automated_snapshot_retention_period` - The backup retention period
- `availability_zone` - The availability zone of the cluster
- `bucket_name` - The name of the S3 bucket where the log files are to be stored
- `cluster_identifier` - The cluster identifier
- `cluster_parameter_group_name` - The name of the parameter group to be associated with this cluster
- `cluster_public_key` - The public key for the cluster
- `cluster_revision_number` - The cluster revision number
- `cluster_security_groups` - The security groups associated with the cluster
- `cluster_subnet_group_name` - The name of a cluster subnet group to be associated with this cluster
- `cluster_type` - The cluster type

- database_name - The name of the default database in the cluster
- elastic_ip - The Elastic IP of the cluster
- enable_logging - Whether cluster logging is enabled
- encrypted - Whether the cluster data is encrypted
- endpoint - The cluster endpoint
- enhanced_vpc_routing - Whether enhanced VPC routing is enabled
- iam_roles - The IAM roles associated to the cluster
- kms_key_id - The KMS encryption key associated to the cluster
- master_username - Username for the master DB user
- node_type - The cluster node type
- number_of_nodes - The number of nodes in the cluster
- port - The port the cluster responds on
- preferred_maintenance_window - The maintenance window
- publicly_accessible - Whether the cluster is publicly accessible
- s3_key_prefix - The folder inside the S3 bucket where the log files are stored
- tags - The tags associated to the cluster
- vpc_id - The VPC Id associated with the cluster
- vpc_security_group_ids - The VPC security group Ids associated with the cluster

» Data Source: aws_redshift_service_account

Use this data source to get the Account ID of the AWS Redshift Service Account in a given region for the purpose of allowing Redshift to store audit data in S3.

» Example Usage

```
data "aws_redshift_service_account" "main" {}

resource "aws_s3_bucket" "bucket" {
  bucket        = "tf-redshift-logging-test-bucket"
  force_destroy = true

  policy = <<EOF
{
    "Version": "2008-10-17",
    "Statement": [
        {
                    "Sid": "Put bucket policy needed for audit logging",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "${data.aws_redshift_service_account.main.arn}"
```

```
                },
                "Action": "s3:PutObject",
                "Resource": "arn:aws:s3:::tf-redshift-logging-test-bucket/*"
            },
            {
                "Sid": "Get bucket policy needed for audit logging ",
                "Effect": "Allow",
                "Principal": {
                    "AWS": "${data.aws_redshift_service_account.main.arn}"
                },
                "Action": "s3:GetBucketAcl",
                "Resource": "arn:aws:s3:::tf-redshift-logging-test-bucket"
            }
    ]
}
EOF
}
```

» Argument Reference

- `region` - (Optional) Name of the region whose AWS Redshift account ID is desired. Defaults to the region from the AWS provider configuration.

» Attributes Reference

- `id` - The ID of the AWS Redshift service account in the selected region.
- `arn` - The ARN of the AWS Redshift service account in the selected region.

» Resource: aws_redshift_cluster

Provides a Redshift Cluster Resource.

Note: All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "aws_redshift_cluster" "default" {
  cluster_identifier = "tf-redshift-cluster"
  database_name      = "mydb"
  master_username    = "foo"
  master_password    = "Mustbe8characters"
  node_type          = "dc1.large"
  cluster_type       = "single-node"
}
```

» Argument Reference

For more detailed documentation about each argument, refer to the AWS official documentation.

The following arguments are supported:

- cluster_identifier - (Required) The Cluster Identifier. Must be a lower case string.
- database_name - (Optional) The name of the first database to be created when the cluster is created. If you do not provide a name, Amazon Redshift will create a default database called dev.
- node_type - (Required) The node type to be provisioned for the cluster.
- cluster_type - (Optional) The cluster type to use. Either single-node or multi-node.
- master_password - (Required unless a snapshot_identifier is provided) Password for the master DB user. Note that this may show up in logs, and it will be stored in the state file. Password must contain at least 8 chars and contain at least one uppercase letter, one lowercase letter, and one number.
- master_username - (Required unless a snapshot_identifier is provided) Username for the master DB user.
- cluster_security_groups - (Optional) A list of security groups to be associated with this cluster.
- vpc_security_group_ids - (Optional) A list of Virtual Private Cloud (VPC) security groups to be associated with the cluster.
- cluster_subnet_group_name - (Optional) The name of a cluster subnet group to be associated with this cluster. If this parameter is not provided the resulting cluster will be deployed outside virtual private cloud (VPC).
- availability_zone - (Optional) The EC2 Availability Zone (AZ) in which you want Amazon Redshift to provision the cluster. For example, if you have several EC2 instances running in a specific Availability Zone, then you might want the cluster to be provisioned in the same zone in order to decrease network latency.
- preferred_maintenance_window - (Optional) The weekly time range (in UTC) during which automated cluster maintenance can occur. Format: ddd:hh24:mi-ddd:hh24:mi

- `cluster_parameter_group_name` - (Optional) The name of the parameter group to be associated with this cluster.
- `automated_snapshot_retention_period` - (Optional) The number of days that automated snapshots are retained. If the value is 0, automated snapshots are disabled. Even if automated snapshots are disabled, you can still create manual snapshots when you want with create-cluster-snapshot. Default is 1.
- `port` - (Optional) The port number on which the cluster accepts incoming connections. The cluster is accessible only via the JDBC and ODBC connection strings. Part of the connection string requires the port on which the cluster will listen for incoming connections. Default port is 5439.
- `cluster_version` - (Optional) The version of the Amazon Redshift engine software that you want to deploy on the cluster. The version selected runs on all the nodes in the cluster.
- `allow_version_upgrade` - (Optional) If true , major version upgrades can be applied during the maintenance window to the Amazon Redshift engine that is running on the cluster. Default is true
- `number_of_nodes` - (Optional) The number of compute nodes in the cluster. This parameter is required when the ClusterType parameter is specified as multi-node. Default is 1.
- `publicly_accessible` - (Optional) If true, the cluster can be accessed from a public network. Default is `true`.
- `encrypted` - (Optional) If true , the data in the cluster is encrypted at rest.
- `enhanced_vpc_routing` - (Optional) If true , enhanced VPC routing is enabled.
- `kms_key_id` - (Optional) The ARN for the KMS encryption key. When specifying `kms_key_id`, `encrypted` needs to be set to true.
- `elastic_ip` - (Optional) The Elastic IP (EIP) address for the cluster.
- `skip_final_snapshot` - (Optional) Determines whether a final snapshot of the cluster is created before Amazon Redshift deletes the cluster. If true , a final cluster snapshot is not created. If false , a final cluster snapshot is created before the cluster is deleted. Default is false.
- `final_snapshot_identifier` - (Optional) The identifier of the final snapshot that is to be created immediately before deleting the cluster. If this parameter is provided, `skip_final_snapshot` must be false.
- `snapshot_identifier` - (Optional) The name of the snapshot from which to create the new cluster.

- `snapshot_cluster_identifier` - (Optional) The name of the cluster the source snapshot was created from.
- `owner_account` - (Optional) The AWS customer account used to create or copy the snapshot. Required if you are restoring a snapshot you do not own, optional if you own the snapshot.
- `iam_roles` - (Optional) A list of IAM Role ARNs to associate with the cluster. A Maximum of 10 can be associated to the cluster at any time.
- `logging` - (Optional) Logging, documented below.
- `snapshot_copy` - (Optional) Configuration of automatic copy of snapshots from one region to another. Documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Timeouts

`aws_redshift_cluster` provides the following Timeouts configuration options:

- `create` - (Default `75 minutes`) Used for creating Clusters.
- `update` - (Default `40 minutes`) Used for Cluster Argument changes.
- `delete` - (Default `40 minutes`) Used for destroying Clusters.

» Nested Blocks

» `logging`

- `enable` - (Required) Enables logging information such as queries and connection attempts, for the specified Amazon Redshift cluster.
- `bucket_name` - (Optional, required when `enable` is `true`) The name of an existing S3 bucket where the log files are to be stored. Must be in the same region as the cluster and the cluster must have read bucket and put object permissions. For more information on the permissions required for the bucket, please read the AWS documentation
- `s3_key_prefix` - (Optional) The prefix applied to the log file names.

» `snapshot_copy`

- `destination_region` - (Required) The destination region that you want to copy snapshots to.
- `retention_period` - (Optional) The number of days to retain automated snapshots in the destination region after they are copied from the source region. Defaults to `7`.
- `grant_name` - (Optional) The name of the snapshot copy grant to use when snapshots of an AWS KMS-encrypted cluster are copied to the destination region.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of cluster
- `id` - The Redshift Cluster ID.
- `cluster_identifier` - The Cluster Identifier
- `cluster_type` - The cluster type
- `node_type` - The type of nodes in the cluster
- `database_name` - The name of the default database in the Cluster
- `availability_zone` - The availability zone of the Cluster
- `automated_snapshot_retention_period` - The backup retention period
- `preferred_maintenance_window` - The backup window
- `endpoint` - The connection endpoint
- `encrypted` - Whether the data in the cluster is encrypted
- `cluster_security_groups` - The security groups associated with the cluster
- `vpc_security_group_ids` - The VPC security group Ids associated with the cluster
- `dns_name` - The DNS name of the cluster
- `port` - The Port the cluster responds on
- `cluster_version` - The version of Redshift engine software
- `cluster_parameter_group_name` - The name of the parameter group to be associated with this cluster
- `cluster_subnet_group_name` - The name of a cluster subnet group to be associated with this cluster
- `cluster_public_key` - The public key for the cluster
- `cluster_revision_number` - The specific revision number of the database in the cluster

» Import

Redshift Clusters can be imported using the `cluster_identifier`, e.g.

```
$ terraform import aws_redshift_cluster.myprodcluster tf-redshift-cluster-12345
```

» Resource: aws_redshift_event_subscription

Provides a Redshift event subscription resource.

» Example Usage

```
resource "aws_redshift_cluster" "default" {
  cluster_identifier = "default"
  database_name      = "default"

  # ...
}

resource "aws_sns_topic" "default" {
  name = "redshift-events"
}

resource "aws_redshift_event_subscription" "default" {
  name          = "redshift-event-sub"
  sns_topic_arn = "${aws_sns_topic.default.arn}"

  source_type = "cluster"
  source_ids  = ["${aws_redshift_cluster.default.id}"]

  severity = "INFO"

  event_categories = [
    "configuration",
    "management",
    "monitoring",
    "security",
  ]

  tags = {
    Name = "default"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Redshift event subscription.
- `sns_topic_arn` - (Required) The ARN of the SNS topic to send events to.
- `source_ids` - (Optional) A list of identifiers of the event sources for which events will be returned. If not specified, then all sources are included in the response. If specified, a source_type must also be specified.
- `source_type` - (Optional) The type of source that will be generating the events. Valid options are `cluster`, `cluster-parameter-group`, `cluster-security-group`, or `cluster-snapshot`. If not set, all sources will be subscribed to.

- `severity` - (Optional) The event severity to be published by the notification subscription. Valid options are `INFO` or `ERROR`.
- `event_categories` - (Optional) A list of event categories for a SourceType that you want to subscribe to. See https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-event-notifications.html or run `aws redshift describe-event-categories`.
- `enabled` - (Optional) A boolean flag to enable/disable the subscription. Defaults to true.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes

The following additional atttributes are provided:

- `arn` - Amazon Resource Name (ARN) of the Redshift event notification subscription
- `id` - The name of the Redshift event notification subscription
- `customer_aws_id` - The AWS customer account associated with the Redshift event notification subscription

» Import

Redshift Event Subscriptions can be imported using the `name`, e.g.

```
$ terraform import aws_redshift_event_subscription.default redshift-event-sub
```

» Resource: aws_redshift_parameter_group

Provides a Redshift Cluster parameter group resource.

» Example Usage

```
resource "aws_redshift_parameter_group" "bar" {
  name   = "parameter-group-test-terraform"
  family = "redshift-1.0"

  parameter {
    name  = "require_ssl"
    value = "true"
  }

  parameter {
    name  = "query_group"
    value = "example"
  }

  parameter {
    name  = "enable_user_activity_logging"
    value = "true"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Redshift parameter group.
- `family` - (Required) The family of the Redshift parameter group.
- `description` - (Optional) The description of the Redshift parameter group. Defaults to ”Managed by Terraform”.
- `parameter` - (Optional) A list of Redshift parameters to apply.

Parameter blocks support the following:

- `name` - (Required) The name of the Redshift parameter.
- `value` - (Required) The value of the Redshift parameter.
- `tags` - (Optional) A mapping of tags to assign to the resource.

You can read more about the parameters that Redshift supports in the documentation

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of parameter group
- `id` - The Redshift parameter group name.

» Import

Redshift Parameter Groups can be imported using the `name`, e.g.

```
$ terraform import aws_redshift_parameter_group.paramgroup1 parameter-group-test-terraform
```

» Resource: aws_redshift_security_group

Creates a new Amazon Redshift security group. You use security groups to control access to non-VPC clusters

» Example Usage

```
resource "aws_redshift_security_group" "default" {
  name = "redshift-sg"

  ingress {
    cidr = "10.0.0.0/24"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Redshift security group.
- `description` - (Optional) The description of the Redshift security group. Defaults to ”Managed by Terraform”.
- `ingress` - (Optional) A list of ingress rules.

Ingress blocks support the following:

- `cidr` - The CIDR block to accept
- `security_group_name` - The name of the security group to authorize
- `security_group_owner_id` - The owner Id of the security group provided by `security_group_name`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Redshift security group ID.

» Import

Redshift security groups can be imported using the `name`, e.g.

```
$ terraform import aws_redshift_security_group.testgroup1 redshift_test_group
```

» Resource: aws_redshift_snapshot_copy_grant

Creates a snapshot copy grant that allows AWS Redshift to encrypt copied snapshots with a customer master key from AWS KMS in a destination region.

Note that the grant must exist in the destination region, and not in the region of the cluster.

» Example Usage

```
resource "aws_redshift_snapshot_copy_grant" "test" {
  snapshot_copy_grant_name = "my-grant"
}

resource "aws_redshift_cluster" "test" {
  # ... other configuration ...
  snapshot_copy {
    destination_region = "us-east-2"
    grant_name         = "${aws_redshift_snapshot_copy_grant.test.snapshot_copy_grant_name}'
  }
}
```

» Argument Reference

The following arguments are supported:

- `snapshot_copy_grant_name` - (Required, Forces new resource) A friendly name for identifying the grant.
- `kms_key_id` - (Optional, Forces new resource) The unique identifier for the customer master key (CMK) that the grant applies to. Specify the key ID or the Amazon Resource Name (ARN) of the CMK. To specify a CMK in a different AWS account, you must use the key ARN. If not specified, the default key is used.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

- `arn` - Amazon Resource Name (ARN) of snapshot copy grant

» Resource: aws_redshift_snapshot_schedule_association

» Example Usage

```
resource "aws_redshift_cluster" "default" {
  cluster_identifier = "tf-redshift-cluster"
  database_name      = "mydb"
  master_username    = "foo"
  master_password    = "Mustbe8characters"
  node_type          = "dc1.large"
  cluster_type       = "single-node"
}

resource "aws_redshift_snapshot_schedule" "default" {
    identifier = "tf-redshift-snapshot-schedule"
    definitions = [
        "rate(12 hours)",
    ]
}

resource "aws_redshift_snapshot_schedule_association" "default" {
      cluster_identifier  = "${aws_redshift_cluster.default.id}"
    schedule_identifier = "${aws_redshift_snapshot_schedule.default.id}"
}
```

» Argument Reference

The following arguments are supported:

- `cluster_identifier` - (Required, Forces new resource) The cluster identifier.
- `schedule_identifier` - (Required, Forces new resource) The snapshot schedule identifier.

» Import

Redshift Snapshot Schedule Association can be imported using the `<cluster-identifier>/<schedule-identifier>`, e.g.

```
$ terraform import aws_redshift_snapshot_schedule_association.default tf-redshift-cluster/tf
```

» Resource: aws_redshift_snapshot_schedule

» Example Usage

```
resource "aws_redshift_snapshot_schedule" "default" {
    identifier = "tf-redshift-snapshot-schedule"
    definitions = [
        "rate(12 hours)",
    ]
}
```

» Argument Reference

The following arguments are supported:

- `identifier` - (Optional, Forces new resource) The snapshot schedule identifier. If omitted, Terraform will assign a random, unique identifier.
- `identifier_prefix` - (Optional, Forces new resource) Creates a unique identifier beginning with the specified prefix. Conflicts with `identifier`.
- `description` - (Optional) The description of the snapshot schedule.
- `definitions` - (Optional) The definition of the snapshot schedule. The definition is made up of schedule expressions, for example `cron(30 12 *)` or `rate(12 hours)`.
- `force_destroy` - (Optional) Whether to destroy all associated clusters with this snapshot schedule on deletion. Must be enabled and applied before attempting deletion.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Import

Redshift Snapshot Schedule can be imported using the `identifier`, e.g.

```
$ terraform import aws_redshift_snapshot_schedule.default tf-redshift-snapshot-schedule
```

» Resource: aws_redshift_subnet_group

Creates a new Amazon Redshift subnet group. You must provide a list of one or more subnets in your existing Amazon Virtual Private Cloud (Amazon VPC) when creating Amazon Redshift subnet group.

» Example Usage

```
resource "aws_vpc" "foo" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_subnet" "foo" {
  cidr_block        = "10.1.1.0/24"
  availability_zone = "us-west-2a"
  vpc_id            = "${aws_vpc.foo.id}"

  tags = {
    Name = "tf-dbsubnet-test-1"
  }
}

resource "aws_subnet" "bar" {
  cidr_block        = "10.1.2.0/24"
  availability_zone = "us-west-2b"
  vpc_id            = "${aws_vpc.foo.id}"

  tags = {
    Name = "tf-dbsubnet-test-2"
  }
}

resource "aws_redshift_subnet_group" "foo" {
  name       = "foo"
  subnet_ids = ["${aws_subnet.foo.id}", "${aws_subnet.bar.id}"]

  tags = {
    environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Redshift Subnet group.
- `description` - (Optional) The description of the Redshift Subnet group. Defaults to ”Managed by Terraform”.
- `subnet_ids` - (Required) An array of VPC subnet IDs.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the Redshift Subnet group name
- `id` - The Redshift Subnet group ID.

» Import

Redshift subnet groups can be imported using the `name`, e.g.

```
$ terraform import aws_redshift_subnet_group.testgroup1 test-cluster-subnet-group
```

» Resource: aws_resourcegroups_group

Provides a Resource Group.

» Example Usage

```
resource "aws_resourcegroups_group" "test" {
  name = "test-group"

  resource_query {
    query = <<JSON
{
  "ResourceTypeFilters": [
    "AWS::EC2::Instance"
  ],
  "TagFilters": [
    {
      "Key": "Stage",
      "Values": ["Test"]
    }
  ]
}
JSON
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource group's name. A resource group name can have a maximum of 127 characters, including letters, numbers, hyphens, dots, and underscores. The name cannot start with `AWS` or `aws`.
- `description` - (Optional) A description of the resource group.
- `resource_query` - (Required) A `resource_query` block. Resource queries are documented below.
- `tags` - (Optional) Key-value mapping of resource tags

An `resource_query` block supports the following arguments:

- `query` - (Required) The resource query as a JSON string.
- `type` - (Required) The type of the resource query. Defaults to `TAG_FILTERS_1_0`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN assigned by AWS for this resource group.

» Import

Resource groups can be imported using the `name`, e.g.

```
$ terraform import aws_resourcegroups_group.foo resource-group-name
```

» Data Source: aws_route53_delegation_set

`aws_route53_delegation_set` provides details about a specific Route 53 Delegation Set.

This data source allows to find a list of name servers associated with a specific delegation set.

» Example Usage

The following example shows how to get a delegation set from its id.

```
data "aws_route53_delegation_set" "dset" {
  id = "MQWGHCBFAKEID"
}
```

» Argument Reference

- `id` - (Required) The Hosted Zone id of the desired delegation set.

The following attribute is additionally exported:

- `caller_reference` - Caller Reference of the delegation set.
- `name_servers` - The list of DNS name servers for the delegation set.

» Data Source: aws_route53_zone

`aws_route53_zone` provides details about a specific Route 53 Hosted Zone.

This data source allows to find a Hosted Zone ID given Hosted Zone name and certain search criteria.

» Example Usage

The following example shows how to get a Hosted Zone from its name and from this data how to create a Record Set.

```
data "aws_route53_zone" "selected" {
  name         = "test.com."
  private_zone = true
}

resource "aws_route53_record" "www" {
  zone_id = "${data.aws_route53_zone.selected.zone_id}"
  name    = "www.${data.aws_route53_zone.selected.name}"
  type    = "A"
  ttl     = "300"
  records = ["10.0.0.1"]
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available Hosted Zone. You have to use `zone_id` or `name`, not both of them. The given filter must match exactly one Hosted Zone. If you use `name` field for private Hosted Zone, you need to add `private_zone` field to `true`

- `zone_id` - (Optional) The Hosted Zone id of the desired Hosted Zone.

- `name` - (Optional) The Hosted Zone name of the desired Hosted Zone.

- `private_zone` - (Optional) Used with `name` field to get a private Hosted Zone.
- `vpc_id` - (Optional) Used with `name` field to get a private Hosted Zone associated with the vpc_id (in this case, private_zone is not mandatory).
- `tags` - (Optional) Used with `name` field. A mapping of tags, each pair of which must exactly match a pair on the desired Hosted Zone.

» Attributes Reference

All of the argument attributes are also exported as result attributes. This data source will complete the data by populating any fields that are not included in the configuration with the data for the selected Hosted Zone.

The following attribute is additionally exported:

- `caller_reference` - Caller Reference of the Hosted Zone.
- `comment` - The comment field of the Hosted Zone.
- `name_servers` - The list of DNS name servers for the Hosted Zone.
- `resource_record_set_count` - The number of Record Set in the Hosted Zone.
- `linked_service_principal` - The service that created the Hosted Zone (e.g. `servicediscovery.amazonaws.com`).
- `linked_service_description` - The description provided by the service that created the Hosted Zone (e.g. `arn:aws:servicediscovery:us-east-1:1234567890:namespace/ns-`

» Resource: aws_route53_delegation_set

Provides a Route53 Delegation Set resource.

» Example Usage

```
resource "aws_route53_delegation_set" "main" {
  reference_name = "DynDNS"
}

resource "aws_route53_zone" "primary" {
  name              = "hashicorp.com"
  delegation_set_id = "${aws_route53_delegation_set.main.id}"
}

resource "aws_route53_zone" "secondary" {
  name              = "terraform.io"
  delegation_set_id = "${aws_route53_delegation_set.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `reference_name` - (Optional) This is a reference name used in Caller Reference (helpful for identifying single delegation set amongst others)

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The delegation set ID
- `name_servers` - A list of authoritative name servers for the hosted zone (effectively a list of NS records).

» Import

Route53 Delegation Sets can be imported using the `delegation set id`, e.g.

```
$ terraform import aws_route53_delegation_set.set1 N1PA6795SAMPLE
```

» Resource: aws_route53_health_check

Provides a Route53 health check.

» Example Usage

» Connectivity and HTTP Status Code Check

```
resource "aws_route53_health_check" "example" {
  fqdn              = "example.com"
  port              = 80
  type              = "HTTP"
  resource_path     = "/"
  failure_threshold = "5"
  request_interval  = "30"

  tags = {
    Name = "tf-test-health-check"
  }
}
```

» Connectivity and String Matching Check

```
resource "aws_route53_health_check" "example" {
  failure_threshold = "5"
  fqdn              = "example.com"
  port              = 443
  request_interval  = "30"
  resource_path     = "/"
  search_string     = "example"
  type              = "HTTPS_STR_MATCH"
}
```

» Aggregate Check

```
resource "aws_route53_health_check" "parent" {
  type                   = "CALCULATED"
  child_health_threshold = 1
  child_healthchecks     = ["${aws_route53_health_check.child.id}"]

  tags = {
    Name = "tf-test-calculated-health-check"
  }
}
```

» CloudWatch Alarm Check

```
resource "aws_cloudwatch_metric_alarm" "foobar" {
  alarm_name          = "terraform-test-foobar5"
  comparison_operator = "GreaterThanOrEqualToThreshold"
  evaluation_periods  = "2"
  metric_name         = "CPUUtilization"
  namespace           = "AWS/EC2"
  period              = "120"
  statistic           = "Average"
  threshold           = "80"
  alarm_description   = "This metric monitors ec2 cpu utilization"
}

resource "aws_route53_health_check" "foo" {
  type                            = "CLOUDWATCH_METRIC"
  cloudwatch_alarm_name           = "${aws_cloudwatch_metric_alarm.foobar.alarm_name}"
  cloudwatch_alarm_region         = "us-west-2"
  insufficient_data_health_status = "Healthy"
}
```

» Argument Reference

The following arguments are supported:

- `reference_name` - (Optional) This is a reference name used in Caller Reference (helpful for identifying single health_check set amongst others)
- `fqdn` - (Optional) The fully qualified domain name of the endpoint to be checked.
- `ip_address` - (Optional) The IP address of the endpoint to be checked.
- `port` - (Optional) The port of the endpoint to be checked.
- `type` - (Required) The protocol to use when performing health checks. Valid values are `HTTP`, `HTTPS`, `HTTP_STR_MATCH`, `HTTPS_STR_MATCH`, `TCP`, `CALCULATED` and `CLOUDWATCH_METRIC`.
- `failure_threshold` - (Required) The number of consecutive health checks that an endpoint must pass or fail.
- `request_interval` - (Required) The number of seconds between the time that Amazon Route 53 gets a response from your endpoint and the time that it sends the next health-check request.
- `resource_path` - (Optional) The path that you want Amazon Route 53 to request when performing health checks.
- `search_string` - (Optional) String searched in the first 5120 bytes of the response body for check to be considered healthy. Only valid with `HTTP_STR_MATCH` and `HTTPS_STR_MATCH`.
- `measure_latency` - (Optional) A Boolean value that indicates whether you want Route 53 to measure the latency between health checkers in multiple AWS regions and your endpoint and to display CloudWatch latency graphs in the Route 53 console.
- `invert_healthcheck` - (Optional) A boolean value that indicates whether the status of health check should be inverted. For example, if a health check is healthy but Inverted is True , then Route 53 considers the health check to be unhealthy.
- `enable_sni` - (Optional) A boolean value that indicates whether Route53 should send the `fqdn` to the endpoint when performing the health check. This defaults to AWS' defaults: when the `type` is "HTTPS" `enable_sni`

defaults to `true`, when `type` is anything else `enable_sni` defaults to `false`.

- `child_healthchecks` - (Optional) For a specified parent health check, a list of HealthCheckId values for the associated child health checks.
- `child_health_threshold` - (Optional) The minimum number of child health checks that must be healthy for Route 53 to consider the parent health check to be healthy. Valid values are integers between 0 and 256, inclusive
- `cloudwatch_alarm_name` - (Optional) The name of the CloudWatch alarm.
- `cloudwatch_alarm_region` - (Optional) The CloudWatchRegion that the CloudWatch alarm was created in.
- `insufficient_data_health_status` - (Optional) The status of the health check when CloudWatch has insufficient data about the state of associated alarm. Valid values are `Healthy` , `Unhealthy` and `LastKnownStatus`.
- `regions` - (Optional) A list of AWS regions that you want Amazon Route 53 health checkers to check the specified endpoint from.
- `tags` - (Optional) A mapping of tags to assign to the health check.

At least one of either `fqdn` or `ip_address` must be specified.

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The id of the health check

» Import

Route53 Health Checks can be imported using the `health check id`, e.g.

```
$ terraform import aws_route53_health_check.http_check abcdef11-2222-3333-4444-555555fedcba
```

» Resource: aws_route53_query_log

Provides a Route53 query logging configuration resource.

NOTE: There are restrictions on the configuration of query logging. Notably, the CloudWatch log group must be in the `us-east-1` region, a permissive CloudWatch log resource policy must be in place, and the Route53 hosted zone must be public. See Configuring Logging for DNS Queries for additional details.

» Example Usage

```
# Example CloudWatch log group in us-east-1

provider "aws" {
  alias  = "us-east-1"
  region = "us-east-1"
}

resource "aws_cloudwatch_log_group" "aws_route53_example_com" {
  provider = "aws.us-east-1"

  name              = "/aws/route53/${aws_route53_zone.example_com.name}"
  retention_in_days = 30
}

# Example CloudWatch log resource policy to allow Route53 to write logs
# to any log group under /aws/route53/*

data "aws_iam_policy_document" "route53-query-logging-policy" {
  statement {
    actions = [
      "logs:CreateLogStream",
      "logs:PutLogEvents",
    ]

    resources = ["arn:aws:logs:*:*:log-group:/aws/route53/*"]

    principals {
      identifiers = ["route53.amazonaws.com"]
      type        = "Service"
    }
  }
}

resource "aws_cloudwatch_log_resource_policy" "route53-query-logging-policy" {
  provider = "aws.us-east-1"

  policy_document = "${data.aws_iam_policy_document.route53-query-logging-policy.json}"
  policy_name     = "route53-query-logging-policy"
}

# Example Route53 zone with query logging

resource "aws_route53_zone" "example_com" {
  name = "example.com"
}

resource "aws_route53_query_log" "example_com" {
  depends_on = ["aws_cloudwatch_log_resource_policy.route53-query-logging-policy"]

  cloudwatch_log_group_arn = "${aws_cloudwatch_log_group.aws_route53_example_com.arn}"
  zone_id                  = "${aws_route53_zone.example_com.zone_id}"
}
```

» Argument Reference

The following arguments are supported:

- `cloudwatch_log_group_arn` - (Required) CloudWatch log group ARN to send query logs.
- `zone_id` - (Required) Route53 hosted zone ID to enable query logs.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The query logging configuration ID

» Import

Route53 query logging configurations can be imported using their ID, e.g.

```
$ terraform import aws_route53_query_log.example_com xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

» Resource: aws_route53_record

Provides a Route53 record resource.

» Example Usage

» Simple routing policy

```
resource "aws_route53_record" "www" {
  zone_id = "${aws_route53_zone.primary.zone_id}"
  name    = "www.example.com"
  type    = "A"
  ttl     = "300"
  records = ["${aws_eip.lb.public_ip}"]
}
```

» Weighted routing policy

Other routing policies are configured similarly. See AWS Route53 Developer Guide for details.

```
resource "aws_route53_record" "www-dev" {
  zone_id = "${aws_route53_zone.primary.zone_id}"
  name    = "www"
  type    = "CNAME"
  ttl     = "5"

  weighted_routing_policy {
    weight = 10
  }

  set_identifier = "dev"
  records        = ["dev.example.com"]
}

resource "aws_route53_record" "www-live" {
  zone_id = "${aws_route53_zone.primary.zone_id}"
  name    = "www"
  type    = "CNAME"
  ttl     = "5"

  weighted_routing_policy {
    weight = 90
  }

  set_identifier = "live"
  records        = ["live.example.com"]
}
```

» Alias record

See related part of AWS Route53 Developer Guide to understand differences between alias and non-alias records.

TTL for all alias records is 60 seconds, you cannot change this, therefore `ttl` has to be omitted in alias records.

```
resource "aws_elb" "main" {
  name               = "foobar-terraform-elb"
  availability_zones = ["us-east-1c"]

  listener {
    instance_port     = 80
    instance_protocol = "http"
    lb_port           = 80
    lb_protocol       = "http"
  }
}

resource "aws_route53_record" "www" {
  zone_id = "${aws_route53_zone.primary.zone_id}"
  name    = "example.com"
  type    = "A"

  alias {
    name                   = "${aws_elb.main.dns_name}"
    zone_id                = "${aws_elb.main.zone_id}"
    evaluate_target_health = true
  }
}
```

» NS and SOA Record Management

When creating Route 53 zones, the `NS` and `SOA` records for the zone are automatically created. Enabling the `allow_overwrite` argument will allow managing these records in a single Terraform run without the requirement for `terraform import`.

```
resource "aws_route53_zone" "example" {
  name = "test.example.com"
}

resource "aws_route53_record" "example" {
  allow_overwrite = true
  name            = "test.example.com"
  ttl             = 30
  type            = "NS"
  zone_id         = "${aws_route53_zone.example.zone_id}"

  records = [
    "${aws_route53_zone.example.name_servers.0}",
    "${aws_route53_zone.example.name_servers.1}",
    "${aws_route53_zone.example.name_servers.2}",
    "${aws_route53_zone.example.name_servers.3}",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The ID of the hosted zone to contain this record.
- `name` - (Required) The name of the record.
- `type` - (Required) The record type. Valid values are `A`, `AAAA`, `CAA`, `CNAME`, `MX`, `NAPTR`, `NS`, `PTR`, `SOA`, `SPF`, `SRV` and `TXT`.
- `ttl` - (Required for non-alias records) The TTL of the record.
- `records` - (Required for non-alias records) A string list of records. To specify a single record value longer than 255 characters such as a TXT record for DKIM, add `\"\"` inside the Terraform configuration string (e.g. `"first255characters\"\"morecharacters"`).
- `set_identifier` - (Optional) Unique identifier to differentiate records with routing policies from one another. Required if using `failover`, `geolocation`, `latency`, or `weighted` routing policies documented below.
- `health_check_id` - (Optional) The health check the record should be associated with.
- `alias` - (Optional) An alias block. Conflicts with `ttl` & `records`. Alias record documented below.
- `failover_routing_policy` - (Optional) A block indicating the routing behavior when associated health check fails. Conflicts with any other routing policy. Documented below.
- `geolocation_routing_policy` - (Optional) A block indicating a routing policy based on the geolocation of the requestor. Conflicts with any other routing policy. Documented below.
- `latency_routing_policy` - (Optional) A block indicating a routing policy based on the latency between the requestor and an AWS region. Conflicts with any other routing policy. Documented below.
- `weighted_routing_policy` - (Optional) A block indicating a weighted routing policy. Conflicts with any other routing policy. Documented below.
- `multivalue_answer_routing_policy` - (Optional) Set to `true` to indi-

cate a multivalue answer routing policy. Conflicts with any other routing policy.
- `allow_overwrite` - (Optional) Allow creation of this record in Terraform to overwrite an existing record, if any. This does not affect the ability to update the record in Terraform and does not prevent other resources within Terraform or manual Route 53 changes outside Terraform from overwriting this record. `false` by default. This configuration is not recommended for most environments.

Exactly one of `records` or `alias` must be specified: this determines whether it's an alias record.

Alias records support the following:

- `name` - (Required) DNS domain name for a CloudFront distribution, S3 bucket, ELB, or another resource record set in this hosted zone.
- `zone_id` - (Required) Hosted zone ID for a CloudFront distribution, S3 bucket, ELB, or Route 53 hosted zone. See `resource_elb.zone_id` for example.
- `evaluate_target_health` - (Required) Set to `true` if you want Route 53 to determine whether to respond to DNS queries using this resource record set by checking the health of the resource record set. Some resources have special requirements, see related part of documentation.

Failover routing policies support the following:

- `type` - (Required) `PRIMARY` or `SECONDARY`. A `PRIMARY` record will be served if its healthcheck is passing, otherwise the `SECONDARY` will be served. See http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-configuring-options.html#dns-failover-failover-rrsets

Geolocation routing policies support the following:

- `continent` - A two-letter continent code. See http://docs.aws.amazon.com/Route53/latest/APIReference/API_GetGeoLocation.html for code details. Either `continent` or `country` must be specified.
- `country` - A two-character country code or `*` to indicate a default resource record set.
- `subdivision` - (Optional) A subdivision code for a country.

Latency routing policies support the following:

- `region` - (Required) An AWS region from which to measure latency. See http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-latency

Weighted routing policies support the following:

- `weight` - (Required) A numeric value indicating the relative weight of the record. See http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-weighted.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `name` - The name of the record.
- `fqdn` - FQDN built using the zone domain and `name`.

» Import

Route53 Records can be imported using ID of the record. The ID is made up as ZONEID_RECORDNAME_TYPE_SET-IDENTIFIER

e.g.

```
Z4KAPRWWNC7JR_dev.example.com_NS_dev
```

In this example, `Z4KAPRWWNC7JR` is the ZoneID, `dev.example.com` is the Record Name, `NS` is the Type and `dev` is the Set Identifier. Only the Set Identifier is actually optional in the ID

To import the ID above, it would look as follows:

```
$ terraform import aws_route53_record.myrecord Z4KAPRWWNC7JR_dev.example.com_NS_dev
```

» Resource: aws_route53_zone

Manages a Route53 Hosted Zone.

» Example Usage

» Public Zone

```
resource "aws_route53_zone" "primary" {
  name = "example.com"
}
```

» Public Subdomain Zone

For use in subdomains, note that you need to create a `aws_route53_record` of type `NS` as well as the subdomain zone.

```
resource "aws_route53_zone" "main" {
  name = "example.com"
}
```

```
resource "aws_route53_zone" "dev" {
  name = "dev.example.com"

  tags = {
    Environment = "dev"
  }
}

resource "aws_route53_record" "dev-ns" {
  zone_id = "${aws_route53_zone.main.zone_id}"
  name    = "dev.example.com"
  type    = "NS"
  ttl     = "30"

  records = [
    "${aws_route53_zone.dev.name_servers.0}",
    "${aws_route53_zone.dev.name_servers.1}",
    "${aws_route53_zone.dev.name_servers.2}",
    "${aws_route53_zone.dev.name_servers.3}",
  ]
}
```

» Private Zone

NOTE: Terraform provides both exclusive VPC associations defined in-line in this resource via `vpc` configuration blocks and a separate Zone VPC Association resource. At this time, you cannot use in-line VPC associations in conjunction with any `aws_route53_zone_association` resources with the same zone ID otherwise it will cause a perpetual difference in plan output. You can optionally use the generic Terraform resource lifecycle configuration block with `ignore_changes` to manage additional associations via the `aws_route53_zone_association` resource.

NOTE: Private zones require at least one VPC association at all times.

```
resource "aws_route53_zone" "private" {
  name = "example.com"

  vpc {
    vpc_id = "${aws_vpc.example.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) This is the name of the hosted zone.
- `comment` - (Optional) A comment for the hosted zone. Defaults to 'Managed by Terraform'.
- `delegation_set_id` - (Optional) The ID of the reusable delegation set whose NS records you want to assign to the hosted zone. Conflicts with `vpc` as delegation sets can only be used for public zones.
- `force_destroy` - (Optional) Whether to destroy all records (possibly managed outside of Terraform) in the zone when destroying the zone.
- `tags` - (Optional) A mapping of tags to assign to the zone.
- `vpc` - (Optional) Configuration block(s) specifying VPC(s) to associate with a private hosted zone. Conflicts with the `delegation_set_id` argument in this resource and any `aws_route53_zone_association` resource specifying the same zone ID. Detailed below.

» vpc Argument Reference

- `vpc_id` - (Required) ID of the VPC to associate.
- `vpc_region` - (Optional) Region of the VPC to associate. Defaults to AWS provider region.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `zone_id` - The Hosted Zone ID. This can be referenced by zone records.
- `name_servers` - A list of name servers in associated (or default) delegation set. Find more about delegation sets in AWS docs.

» Import

Route53 Zones can be imported using the `zone id`, e.g.

```
$ terraform import aws_route53_zone.myzone Z1D633PJN98FT9
```

» Resource: aws_route53_zone_association

Manages a Route53 Hosted Zone VPC association. VPC associations can only be made on private zones.

NOTE: Unless explicit association ordering is required (e.g. a separate cross-account association authorization), usage of this resource is not recommended. Use the `vpc` configuration blocks available within the `aws_route53_zone` resource instead.

NOTE: Terraform provides both this standalone Zone VPC Association resource and exclusive VPC associations defined in-line in the `aws_route53_zone` resource via `vpc` configuration blocks. At this time, you cannot use those in-line VPC associations in conjunction with this resource and the same zone ID otherwise it will cause a perpetual difference in plan output. You can optionally use the generic Terraform resource lifecycle configuration block with `ignore_changes` in the `aws_route53_zone` resource to manage additional associations via this resource.

» Example Usage

```
resource "aws_vpc" "primary" {
  cidr_block           = "10.6.0.0/16"
  enable_dns_hostnames = true
  enable_dns_support   = true
}

resource "aws_vpc" "secondary" {
  cidr_block           = "10.7.0.0/16"
  enable_dns_hostnames = true
  enable_dns_support   = true
}

resource "aws_route53_zone" "example" {
  name = "example.com"

  # NOTE: The aws_route53_zone vpc argument accepts multiple configuration
  #       blocks. The below usage of the single vpc configuration, the
  #       lifecycle configuration, and the aws_route53_zone_association
  #       resource is for illustrative purposes (e.g. for a separate
  #       cross-account authorization process, which is not shown here).
  vpc {
    vpc_id = "${aws_vpc.primary.id}"
  }

  lifecycle {
    ignore_changes = ["vpc"]
  }
}
```

```
resource "aws_route53_zone_association" "secondary" {
  zone_id = "${aws_route53_zone.example.zone_id}"
  vpc_id  = "${aws_vpc.secondary.id}"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The private hosted zone to associate.
- `vpc_id` - (Required) The VPC to associate with the private hosted zone.
- `vpc_region` - (Optional) The VPC's region. Defaults to the region of the AWS provider.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The calculated unique identifier for the association.
- `zone_id` - The ID of the hosted zone for the association.
- `vpc_id` - The ID of the VPC for the association.
- `vpc_region` - The region in which the VPC identified by `vpc_id` was created.

» Import

Route 53 Hosted Zone Associations can be imported via the Hosted Zone ID and VPC ID, separated by a colon (`:`), e.g.

```
$ terraform import aws_route53_zone_association.example Z123456ABCDEFG:vpc-12345678
```

» Data Source: aws_route53_resolver_rule

`aws_route53_resolver_rule` provides details about a specific Route53 Resolver rule.

» Example Usage

The following example shows how to get a Route53 Resolver rule based on its associated domain name and rule type.

```
data "aws_route53_resolver_rule" "example" {
  domain_name = "subdomain.example.com"
  rule_type   = "SYSTEM"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available resolver rules in the current region. The given filters must match exactly one resolver rule whose data will be exported as attributes.

- `domain_name` - (Optional) The domain name the desired resolver rule forwards DNS queries for. Conflicts with `resolver_rule_id`.
- `name` - (Optional) The friendly name of the desired resolver rule. Conflicts with `resolver_rule_id`.
- `resolver_endpoint_id` (Optional) The ID of the outbound resolver endpoint of the desired resolver rule. Conflicts with `resolver_rule_id`.
- `resolver_rule_id` (Optional) The ID of the desired resolver rule. Conflicts with `domain_name`, `name`, `resolver_endpoint_id` and `rule_type`.
- `rule_type` - (Optional) The rule type of the desired resolver rule. Valid values are `FORWARD`, `SYSTEM` and `RECURSIVE`. Conflicts with `resolver_rule_id`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the resolver rule.
- `arn` - The ARN (Amazon Resource Name) for the resolver rule.
- `owner_id` - When a rule is shared with another AWS account, the account ID of the account that the rule is shared with.
- `share_status` - Whether the rules is shared and, if so, whether the current account is sharing the rule with another account, or another account is sharing the rule with the current account. Values are `NOT_SHARED`, `SHARED_BY_ME` or `SHARED_WITH_ME`
- `tags` - A mapping of tags assigned to the resolver rule.

» Data Source: aws_route53_resolver_rules

`aws_route53_resolver_rules` provides details about a set of Route53 Resolver rules.

» Example Usage

The following example shows how to get Route53 Resolver rules based on tags.

```
data "aws_route53_resolver_rules" "example" {
  tags = {
    Environment = "dev"
  }
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available resolver rules in the current region.

- `owner_id` (Optional) When the desired resolver rules are shared with another AWS account, the account ID of the account that the rules are shared with.
- `resolver_endpoint_id` (Optional) The ID of the outbound resolver endpoint for the desired resolver rules.
- `rule_type` (Optional) The rule type of the desired resolver rules. Valid values are `FORWARD`, `SYSTEM` and `RECURSIVE`.
- `share_status` (Optional) Whether the desired resolver rules are shared and, if so, whether the current account is sharing the rules with another account, or another account is sharing the rules with the current account. Values are `NOT_SHARED`, `SHARED_BY_ME` or `SHARED_WITH_ME`

» Attributes Reference

The following attributes are exported:

- `resolver_rule_ids` - The IDs of the matched resolver rules.

» Resource: aws_route53_resolver_endpoint

Provides a Route 53 Resolver endpoint resource.

» Example Usage

```
resource "aws_route53_resolver_endpoint" "foo" {
  name      = "foo"
  direction = "INBOUND"
  security_group_ids = [
    "${aws_security_group.sg1.id}",
    "${aws_security_group.sg2.id}",
  ]

  ip_address {
    subnet_id = "${aws_subnet.sn1.id}"
  }

  ip_address {
    subnet_id = "${aws_subnet.sn2.id}"
    ip        = "10.0.64.4"
  }

  tags {
    Environment = "Prod"
  }
}
```

» Argument Reference

The following arguments are supported:

- `direction` - (Required) The direction of DNS queries to or from the Route 53 Resolver endpoint. Valid values are `INBOUND` (resolver forwards DNS queries to the DNS service for a VPC from your network or another VPC) or `OUTBOUND` (resolver forwards DNS queries from the DNS service for a VPC to your network or another VPC).
- `ip_address` - (Required) The subnets and IP addresses in your VPC that you want DNS queries to pass through on the way from your VPCs to your network (for outbound endpoints) or on the way from your network to your VPCs (for inbound endpoints). Described below.
- `security_group_ids` - (Required) The ID of one or more security groups that you want to use to control access to this VPC.
- `name` - (Optional) The friendly name of the Route 53 Resolver endpoint.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `ip_address` object supports the following:

- `subnet_id` - (Required) The ID of the subnet that contains the IP address.
- `ip` - (Optional) The IP address in the subnet that you want to use for DNS queries.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the Route 53 Resolver endpoint.
- `arn` - The ARN of the Route 53 Resolver endpoint.
- `host_vpc_id` - The ID of the VPC that you want to create the resolver endpoint in.

» Timeouts

`aws_route53_resolver_endpoint` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating Route 53 Resolver endpoint
- `update` - (Default `10 minutes`) Used for updating Route 53 Resolver endpoint
- `delete` - (Default `10 minutes`) Used for destroying Route 53 Resolver endpoint

» Import

Route 53 Resolver endpoints can be imported using the Route 53 Resolver endpoint ID, e.g.

```
$ terraform import aws_route53_resolver_endpoint.foo rslvr-in-abcdef01234567890
```

» Resource: aws_route53_resolver_rule

Provides a Route53 Resolver rule.

» Example Usage

» System rule

```
resource "aws_route53_resolver_rule" "sys" {
  domain_name = "subdomain.example.com"
  rule_type   = "SYSTEM"
}
```

» Forward rule

```
resource "aws_route53_resolver_rule" "fwd" {
  domain_name          = "example.com"
  name                 = "example"
  rule_type            = "FORWARD"
  resolver_endpoint_id = "${aws_route53_resolver_endpoint.foo.id}"

  target_ip {
    ip = "123.45.67.89"
  }

  tags {
    Environment = "Prod"
  }
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Required) DNS queries for this domain name are forwarded to the IP addresses that are specified using `target_ip`.
- `rule_type` - (Required) The rule type. Valid values are `FORWARD`, `SYSTEM` and `RECURSIVE`.
- `name` - (Optional) A friendly name that lets you easily find a rule in the Resolver dashboard in the Route 53 console.
- `resolver_endpoint_id` (Optional) The ID of the outbound resolver endpoint that you want to use to route DNS queries to the IP addresses that you specify using `target_ip`. This argument should only be specified for `FORWARD` type rules.
- `target_ip` - (Optional) Configuration block(s) indicating the IPs that you want Resolver to forward DNS queries to (documented below). This argument should only be specified for `FORWARD` type rules.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `target_ip` object supports the following:

- `ip` - (Required) One IP address that you want to forward DNS queries to. You can specify only IPv4 addresses.
- `port` - (Optional) The port at `ip` that you want to forward DNS queries to. Default value is `53`

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the resolver rule.
- `arn` - The ARN (Amazon Resource Name) for the resolver rule.
- `owner_id` - When a rule is shared with another AWS account, the account ID of the account that the rule is shared with.
- `share_status` - Whether the rules is shared and, if so, whether the current account is sharing the rule with another account, or another account is sharing the rule with the current account. Values are `NOT_SHARED`, `SHARED_BY_ME` or `SHARED_WITH_ME`

» Import

Route53 Resolver rules can be imported using the `id`, e.g.

```
$ terraform import aws_route53_resolver_rule.sys rslvr-rr-0123456789abcdef0
```

» Resource: aws_route53_resolver_rule_association

Provides a Route53 Resolver rule association.

» Example Usage

```
resource "aws_route53_resolver_rule_association" "example" {
  resolver_rule_id = "${aws_route53_resolver_rule.sys.id}"
  vpc_id           = "${aws_vpc.foo.id}"
}
```

» Argument Reference

The following arguments are supported:

- `resolver_rule_id` - (Required) The ID of the resolver rule that you want to associate with the VPC.
- `vpc_id` - (Required) The ID of the VPC that you want to associate the resolver rule with.
- `name` - (Optional) A name for the association that you're creating between a resolver rule and a VPC.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the resolver rule association.

» Import

Route53 Resolver rule associations can be imported using the `id`, e.g.

```
$ terraform import aws_route53_resolver_rule_association.example rslvr-rrassoc-97242eaf88exa
```

» Data Source: aws_canonical_user_id

The Canonical User ID data source allows access to the canonical user ID for the effective account in which Terraform is working.

» Example Usage

```
data "aws_canonical_user_id" "current" {}

output "canonical_user_id" {
  value = "${data.aws_canonical_user_id.current.id}"
}
```

» Argument Reference

There are no arguments available for this data source.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The canonical user ID associated with the AWS account.
- `display_name` - The human-friendly name linked to the canonical user ID. The bucket owner's display name. **NOTE:** This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions.

» Data Source: aws_s3_bucket

Provides details about a specific S3 bucket.

This resource may prove useful when setting up a Route53 record, or an origin for a CloudFront Distribution.

» Example Usage

» Route53 Record

```
data "aws_s3_bucket" "selected" {
  bucket = "bucket.test.com"
}

data "aws_route53_zone" "test_zone" {
  name = "test.com."
}

resource "aws_route53_record" "example" {
  zone_id = "${data.aws_route53_zone.test_zone.id}"
  name    = "bucket"
  type    = "A"

  alias {
    name    = "${data.aws_s3_bucket.selected.website_domain}"
    zone_id = "${data.aws_s3_bucket.selected.hosted_zone_id}"
  }
}
```

» CloudFront Origin

```
data "aws_s3_bucket" "selected" {
  bucket = "a-test-bucket"
}

resource "aws_cloudfront_distribution" "test" {
  origin {
    domain_name = "${data.aws_s3_bucket.selected.bucket_domain_name}"
    origin_id   = "s3-selected-bucket"
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the bucket.
- `arn` - The ARN of the bucket. Will be of format `arn:aws:s3:::bucketname`.
- `bucket_domain_name` - The bucket domain name. Will be of format `bucketname.s3.amazonaws.com`.
- `bucket_regional_domain_name` - The bucket region-specific domain name. The bucket domain name including the region name, please refer here for format. Note: The AWS CloudFront allows specifying S3 region-specific endpoint when creating S3 origin, it will prevent redirect issues from CloudFront to S3 Origin URL.
- `hosted_zone_id` - The Route 53 Hosted Zone ID for this bucket's region.
- `region` - The AWS region this bucket resides in.
- `website_endpoint` - The website endpoint, if the bucket is configured with a website. If not, this will be an empty string.
- `website_domain` - The domain of the website endpoint, if the bucket is configured with a website. If not, this will be an empty string. This is used to create Route 53 alias records.

» Data Source: aws_s3_bucket_object

The S3 object data source allows access to the metadata and *optionally* (see below) content of an object stored inside S3 bucket.

Note: The content of an object (`body` field) is available only for objects which have a human-readable `Content-Type` (`text/*` and `application/json`). This is to prevent printing unsafe characters and potentially downloading large amount of data which would be thrown away in favour of metadata.

» Example Usage

The following example retrieves a text object (which must have a `Content-Type` value starting with `text/`) and uses it as the `user_data` for an EC2 instance:

```
data "aws_s3_bucket_object" "bootstrap_script" {
  bucket = "ourcorp-deploy-config"
  key    = "ec2-bootstrap-script.sh"
}

resource "aws_instance" "example" {
  instance_type = "t2.micro"
  ami           = "ami-2757f631"
  user_data     = "${data.aws_s3_bucket_object.bootstrap_script.body}"
}
```

The following, more-complex example retrieves only the metadata for a zip file stored in S3, which is then used to pass the most recent `version_id` to AWS Lambda for use as a function implementation. More information about Lambda functions is available in the documentation for `aws_lambda_function`.

```
data "aws_s3_bucket_object" "lambda" {
  bucket = "ourcorp-lambda-functions"
  key    = "hello-world.zip"
}

resource "aws_lambda_function" "test_lambda" {
  s3_bucket         = "${data.aws_s3_bucket_object.lambda.bucket}"
  s3_key            = "${data.aws_s3_bucket_object.lambda.key}"
  s3_object_version = "${data.aws_s3_bucket_object.lambda.version_id}"
  function_name     = "lambda_function_name"
  role              = "${aws_iam_role.iam_for_lambda.arn}"             # (not shown)
  handler           = "exports.test"
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to read the object from
- `key` - (Required) The full path to the object inside the bucket
- `version_id` - (Optional) Specific version ID of the object returned (defaults to latest version)

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `body` - Object data (see **limitations above** to understand cases in which this field is actually available)
- `cache_control` - Specifies caching behavior along the request/reply chain.
- `content_disposition` - Specifies presentational information for the object.

- `content_encoding` - Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
- `content_language` - The language the content is in.
- `content_length` - Size of the body in bytes.
- `content_type` - A standard MIME type describing the format of the object data.
- `etag` - ETag generated for the object (an MD5 sum of the object content in case it's not encrypted)
- `expiration` - If the object expiration is configured (see object lifecycle management), the field includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.
- `expires` - The date and time at which the object is no longer cacheable.
- `last_modified` - Last modified date of the object in RFC1123 format (e.g. `Mon, 02 Jan 2006 15:04:05 MST`)
- `metadata` - A map of metadata stored with the object in S3
- `object_lock_legal_hold_status` - Indicates whether this object has an active legal hold. This field is only returned if you have permission to view an object's legal hold status.
- `object_lock_mode` - The object lock retention mode currently in place for this object.
- `object_lock_retain_until_date` - The date and time when this object's object lock will expire.
- `server_side_encryption` - If the object is stored using server-side encryption (KMS or Amazon S3-managed encryption key), this field includes the chosen encryption and algorithm used.
- `sse_kms_key_id` - If present, specifies the ID of the Key Management Service (KMS) master encryption key that was used for the object.
- `storage_class` - Storage class information of the object. Available for all objects except for `Standard` storage class objects.
- `version_id` - The latest version ID of the object returned.
- `website_redirect_location` - If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
- `tags` - A mapping of tags assigned to the object.

Note: Terraform ignores all leading `/`s in the object's `key` and treats multiple `/`s in the rest of the object's `key` as a single `/`, so values of `/index.html` and `index.html` correspond to the same S3 object as do `first//second///third//` and `first/second/third/`.

» Data Source: aws_s3_bucket_objects

NOTE on `max_keys`: Retrieving very large numbers of keys can adversely affect Terraform's performance.

The bucket-objects data source returns keys (i.e., file names) and other metadata about objects in an S3 bucket.

» Example Usage

The following example retrieves a list of all object keys in an S3 bucket and creates corresponding Terraform object data sources:

```
data "aws_s3_bucket_objects" "my_objects" {
  bucket = "ourcorp"
}

data "aws_s3_bucket_object" "object_info" {
  count  = "${length(data.aws_s3_bucket_objects.my_objects.keys)}"
  key    = "${element(data.aws_s3_bucket_objects.my_objects.keys, count.index)}"
  bucket = "${data.aws_s3_bucket_objects.my_objects.bucket}"
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) Lists object keys in this S3 bucket
- `prefix` - (Optional) Limits results to object keys with this prefix (Default: none)
- `delimiter` - (Optional) A character used to group keys (Default: none)
- `encoding_type` - (Optional) Encodes keys using this method (Default: none; besides none, only "url" can be used)
- `max_keys` - (Optional) Maximum object keys to return (Default: 1000)
- `start_after` - (Optional) Returns key names lexicographically after a specific object key in your bucket (Default: none; S3 lists object keys in UTF-8 character encoding in lexicographical order)
- `fetch_owner` - (Optional) Boolean specifying whether to populate the owner list (Default: false)

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `keys` - List of strings representing object keys

- common_prefixes - List of any keys between prefix and the next occurrence of delimiter (i.e., similar to subdirectories of the prefix "directory"); the list is only returned when you specify delimiter
- owners - List of strings representing object owner IDs (see fetch_owner above)

» Resource: aws_s3_account_public_access_block

Manages S3 account-level Public Access Block configuration. For more information about these settings, see the AWS S3 Block Public Access documentation.

NOTE: Each AWS account may only have one S3 Public Access Block configuration. Multiple configurations of the resource against the same AWS account will cause a perpetual difference.

Advanced usage: To use a custom API endpoint for this Terraform resource, use the s3control endpoint provider configuration, not the s3 endpoint provider configuration.

» Example Usage

```
resource "aws_s3_account_public_access_block" "example" {
  block_public_acls   = true
  block_public_policy = true
}
```

» Argument Reference

The following arguments are supported:

- account_id - (Optional) AWS account ID to configure. Defaults to automatically determined account ID of the Terraform AWS provider.
- block_public_acls - (Optional) Whether Amazon S3 should block public ACLs for buckets in this account. Defaults to false. Enabling this setting does not affect existing policies or ACLs. When set to true causes the following behavior:
  - PUT Bucket acl and PUT Object acl calls will fail if the specified ACL allows public access.
  - PUT Object calls will fail if the request includes an object ACL.
- block_public_policy - (Optional) Whether Amazon S3 should block public bucket policies for buckets in this account. Defaults to false. Enabling this setting does not affect existing bucket policies. When set to true causes Amazon S3 to:

    - Reject calls to PUT Bucket policy if the specified bucket policy allows public access.
- `ignore_public_acls` - (Optional) Whether Amazon S3 should ignore public ACLs for buckets in this account. Defaults to `false`. Enabling this setting does not affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set. When set to `true` causes Amazon S3 to:
    - Ignore all public ACLs on buckets in this account and any objects that they contain.
- `restrict_public_buckets` - (Optional) Whether Amazon S3 should restrict public bucket policies for buckets in this account. Defaults to `false`. Enabling this setting does not affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked. When set to `true`:
    - Only the bucket owner and AWS Services can access buckets with public policies.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - AWS account ID

» Import

`aws_s3_account_public_access_block` can be imported by using the AWS account ID, e.g.

```
$ terraform import aws_s3_account_public_access_block.example 123456789012
```

» Resource: aws_s3_bucket

Provides a S3 bucket resource.

» Example Usage

» Private Bucket w/ Tags

```
resource "aws_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"
  tags = {
    Name        = "My bucket"
    Environment = "Dev"
  }
}
```

» Static Website Hosting

```
resource "aws_s3_bucket" "b" {
  bucket = "s3-website-test.hashicorp.com"
  acl    = "public-read"
  policy = "${file("policy.json")}"

  website {
    index_document = "index.html"
    error_document = "error.html"

    routing_rules = <<EOF
[{
    "Condition": {
        "KeyPrefixEquals": "docs/"
    },
    "Redirect": {
        "ReplaceKeyPrefixWith": "documents/"
    }
}]
EOF
  }
}
```

» Using CORS

```
resource "aws_s3_bucket" "b" {
  bucket = "s3-website-test.hashicorp.com"
  acl    = "public-read"

  cors_rule {
    allowed_headers = ["*"]
    allowed_methods = ["PUT", "POST"]
    allowed_origins = ["https://s3-website-test.hashicorp.com"]
    expose_headers  = ["ETag"]
    max_age_seconds = 3000
  }
}
```

» Using versioning

```
resource "aws_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  versioning {
    enabled = true
  }
}
```

» Enable Logging

```
resource "aws_s3_bucket" "log_bucket" {
  bucket = "my-tf-log-bucket"
  acl    = "log-delivery-write"
}

resource "aws_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  logging {
    target_bucket = "${aws_s3_bucket.log_bucket.id}"
    target_prefix = "log/"
  }
}
```

» Using object lifecycle

```
resource "aws_s3_bucket" "bucket" {
  bucket = "my-bucket"
  acl    = "private"

  lifecycle_rule {
    id      = "log"
    enabled = true

    prefix = "log/"

    tags = {
      "rule"      = "log"
      "autoclean" = "true"
    }

    transition {
      days          = 30
      storage_class = "STANDARD_IA" # or "ONEZONE_IA"
    }

    transition {
      days          = 60
      storage_class = "GLACIER"
    }

    expiration {
      days = 90
    }
  }

  lifecycle_rule {
    id      = "tmp"
    prefix  = "tmp/"
    enabled = true

    expiration {
      date = "2016-01-12"
    }
  }
}

resource "aws_s3_bucket" "versioning_bucket" {
  bucket = "my-versioning-bucket"
  acl    = "private"

  versioning {
    enabled = true
  }

  lifecycle_rule {
    prefix  = "config/"
    enabled = true

    noncurrent_version_transition {
      days          = 30
      storage_class = "STANDARD_IA"
    }
    noncurrent_version_transition {
      days          = 60
      storage_class = "GLACIER"
    }

    noncurrent_version_expiration {
      days = 90
    }
  }
}
```

» Using replication configuration

```
provider "aws" {
  region = "eu-west-1"
}

provider "aws" {
  alias  = "central"
  region = "eu-central-1"
}

resource "aws_iam_role" "replication" {
  name = "tf-iam-role-replication-12345"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
POLICY
}

resource "aws_iam_policy" "replication" {
  name = "tf-iam-role-policy-replication-12345"
  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetReplicationConfiguration",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "${aws_s3_bucket.bucket.arn}"
      ]
    },
    {
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl"
      ],
      "Effect": "Allow",
      "Resource": [
        "${aws_s3_bucket.bucket.arn}/*"
      ]
    },
    {
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete"
      ],
      "Effect": "Allow",
      "Resource": "${aws_s3_bucket.destination.arn}/*"
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy_attachment" "replication" {
  role       = "${aws_iam_role.replication.name}"
  policy_arn = "${aws_iam_policy.replication.arn}"
}

resource "aws_s3_bucket" "destination" {
  bucket = "tf-test-bucket-destination-12345"
  region = "eu-west-1"
  versioning {
    enabled = true
  }
}

resource "aws_s3_bucket" "bucket" {
  provider = "aws.central"
  bucket   = "tf-test-bucket-12345"
  acl      = "private"
  region   = "eu-central-1"

  versioning {
    enabled = true
  }

  replication_configuration {
    role = "${aws_iam_role.replication.arn}"

    rules {
      id     = "foobar"
      prefix = "foo"
      status = "Enabled"

      destination {
        bucket        = "${aws_s3_bucket.destination.arn}"
        storage_class = "STANDARD"
      }
    }
  }
}
```

» Enable Default Server Side Encryption

```
resource "aws_kms_key" "mykey" {
  description             = "This key is used to encrypt bucket objects"
  deletion_window_in_days = 10
}

resource "aws_s3_bucket" "mybucket" {
  bucket = "mybucket"

  server_side_encryption_configuration {
    rule {
      apply_server_side_encryption_by_default {
        kms_master_key_id = "${aws_kms_key.mykey.arn}"
        sse_algorithm     = "aws:kms"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Optional, Forces new resource) The name of the bucket. If omitted, Terraform will assign a random, unique name.
- `bucket_prefix` - (Optional, Forces new resource) Creates a unique bucket name beginning with the specified prefix. Conflicts with `bucket`.
- `acl` - (Optional) The canned ACL to apply. Defaults to ”private”.
- `policy` - (Optional) A valid bucket policy JSON document. Note that if the policy document is not specific enough (but still valid), Terraform may view the policy as constantly changing in a `terraform plan`. In this case, please make sure you use the verbose/specific version of the policy. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.
- `tags` - (Optional) A mapping of tags to assign to the bucket.
- `force_destroy` - (Optional, Default:`false`) A boolean that indicates all objects (including any locked objects) should be deleted from the bucket so that the bucket can be destroyed without error. These objects are *not* recoverable.
- `website` - (Optional) A website object (documented below).
- `cors_rule` - (Optional) A rule of Cross-Origin Resource Sharing (documented below).
- `versioning` - (Optional) A state of versioning (documented below)
- `logging` - (Optional) A settings of bucket logging (documented below).
- `lifecycle_rule` - (Optional) A configuration of object lifecycle management (documented below).
- `acceleration_status` - (Optional) Sets the accelerate configuration of an existing bucket. Can be `Enabled` or `Suspended`.
- `region` - (Optional) If specified, the AWS region this bucket should reside in. Otherwise, the region used by the callee.

- `request_payer` - (Optional) Specifies who should bear the cost of Amazon S3 data transfer. Can be either `BucketOwner` or `Requester`. By default, the owner of the S3 bucket would incur the costs of any data transfer. See Requester Pays Buckets developer guide for more information.

- `replication_configuration` - (Optional) A configuration of replication configuration (documented below).

- `server_side_encryption_configuration` - (Optional) A configuration of server-side encryption configuration (documented below)

- `object_lock_configuration` - (Optional) A configuration of S3 object locking (documented below)

NOTE: You cannot use `acceleration_status` in `cn-north-1` or `us-gov-west-1`

The `website` object supports the following:

- `index_document` - (Required, unless using `redirect_all_requests_to`) Amazon S3 returns this index document when requests are made to the root domain or any of the subfolders.
- `error_document` - (Optional) An absolute path to the document to return in case of a 4XX error.
- `redirect_all_requests_to` - (Optional) A hostname to redirect all website requests for this bucket to. Hostname can optionally be prefixed with a protocol (`http://` or `https://`) to use when redirecting requests. The default is the protocol that is used in the original request.
- `routing_rules` - (Optional) A json array containing routing rules describing redirect behavior and when redirects are applied.

The `CORS` object supports the following:

- `allowed_headers` (Optional) Specifies which headers are allowed.
- `allowed_methods` (Required) Specifies which methods are allowed. Can be `GET`, `PUT`, `POST`, `DELETE` or `HEAD`.
- `allowed_origins` (Required) Specifies which origins are allowed.
- `expose_headers` (Optional) Specifies expose header in the response.
- `max_age_seconds` (Optional) Specifies time in seconds that browser can cache the response for a preflight request.

The `versioning` object supports the following:

- `enabled` - (Optional) Enable versioning. Once you version-enable a bucket, it can never return to an unversioned state. You can, however, suspend versioning on that bucket.
- `mfa_delete` - (Optional) Enable MFA delete for either `Change the versioning state of your bucket` or `Permanently delete an object version`. Default is `false`.

The `logging` object supports the following:

- `target_bucket` - (Required) The name of the bucket that will receive the log objects.
- `target_prefix` - (Optional) To specify a key prefix for log objects.

The `lifecycle_rule` object supports the following:

- `id` - (Optional) Unique identifier for the rule.
- `prefix` - (Optional) Object key prefix identifying one or more objects to which the rule applies.
- `tags` - (Optional) Specifies object tags key and value.
- `enabled` - (Required) Specifies lifecycle rule status.
- `abort_incomplete_multipart_upload_days` (Optional) Specifies the number of days after initiating a multipart upload when the multipart upload must be completed.
- `expiration` - (Optional) Specifies a period in the object's expire (documented below).
- `transition` - (Optional) Specifies a period in the object's transitions (documented below).
- `noncurrent_version_expiration` - (Optional) Specifies when noncurrent object versions expire (documented below).
- `noncurrent_version_transition` - (Optional) Specifies when noncurrent object versions transitions (documented below).

At least one of `expiration`, `transition`, `noncurrent_version_expiration`, `noncurrent_version_transition` must be specified.

The `expiration` object supports the following

- `date` (Optional) Specifies the date after which you want the corresponding action to take effect.
- `days` (Optional) Specifies the number of days after object creation when the specific rule action takes effect.
- `expired_object_delete_marker` (Optional) On a versioned bucket (versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to direct Amazon S3 to delete expired object delete markers.

The `transition` object supports the following

- `date` (Optional) Specifies the date after which you want the corresponding action to take effect.
- `days` (Optional) Specifies the number of days after object creation when the specific rule action takes effect.
- `storage_class` (Required) Specifies the Amazon S3 storage class to which you want the object to transition. Can be `ONEZONE_IA`, `STANDARD_IA`, `INTELLIGENT_TIERING`, `GLACIER`, or `DEEP_ARCHIVE`.

The `noncurrent_version_expiration` object supports the following

- `days` (Required) Specifies the number of days an object is noncurrent object versions expire.

The `noncurrent_version_transition` object supports the following

- `days` (Required) Specifies the number of days an object is noncurrent object versions expire.
- `storage_class` (Required) Specifies the Amazon S3 storage class to which you want the noncurrent versions object to transition. Can be `ONEZONE_IA`, `STANDARD_IA`, `INTELLIGENT_TIERING`, `GLACIER`, or `DEEP_ARCHIVE`.

The `replication_configuration` object supports the following:

- `role` - (Required) The ARN of the IAM role for Amazon S3 to assume when replicating the objects.
- `rules` - (Required) Specifies the rules managing the replication (documented below).

The `rules` object supports the following:

- `id` - (Optional) Unique identifier for the rule.
- `priority` - (Optional) The priority associated with the rule.
- `destination` - (Required) Specifies the destination for the rule (documented below).
- `source_selection_criteria` - (Optional) Specifies special object selection criteria (documented below).
- `prefix` - (Optional) Object keyname prefix identifying one or more objects to which the rule applies.
- `status` - (Required) The status of the rule. Either `Enabled` or `Disabled`. The rule is ignored if status is not Enabled.
- `filter` - (Optional) Filter that identifies subset of objects to which the replication rule applies (documented below).

NOTE on `prefix` and `filter`: Amazon S3's latest version of the replication configuration is V2, which includes the `filter` attribute for replication rules. With the `filter` attribute, you can specify object filters based on the object key prefix, tags, or both to scope the objects that the rule applies to. Replication configuration V1 supports filtering based on only the `prefix` attribute. For backwards compatibility, Amazon S3 continues to support the V1 configuration. * For a specific rule, `prefix` conflicts with `filter` * If any rule has `filter` specified then they all must * `priority` is optional (with a default value of `0`) but must be unique between multiple rules

The `destination` object supports the following:

- `bucket` - (Required) The ARN of the S3 bucket where you want Amazon S3 to store replicas of the object identified by the rule.
- `storage_class` - (Optional) The class of storage used to store the object. Can be `STANDARD`, `REDUCED_REDUNDANCY`, `STANDARD_IA`, `ONEZONE_IA`, `INTELLIGENT_TIERING`, `GLACIER`, or `DEEP_ARCHIVE`.

- replica_kms_key_id - (Optional) Destination KMS encryption key ARN for SSE-KMS replication. Must be used in conjunction with sse_kms_encrypted_objects source selection criteria.
- access_control_translation - (Optional) Specifies the overrides to use for object owners on replication. Must be used in conjunction with account_id owner override configuration.
- account_id - (Optional) The Account ID to use for overriding the object owner on replication. Must be used in conjunction with access_control_translation override configuration.

The source_selection_criteria object supports the following:

- sse_kms_encrypted_objects - (Optional) Match SSE-KMS encrypted objects (documented below). If specified, replica_kms_key_id in destination must be specified as well.

The sse_kms_encrypted_objects object supports the following:

- enabled - (Required) Boolean which indicates if this criteria is enabled.

The filter object supports the following:

- prefix - (Optional) Object keyname prefix that identifies subset of objects to which the rule applies.
- tags - (Optional) A mapping of tags that identifies subset of objects to which the rule applies. The rule applies only to objects having all the tags in its tagset.

The server_side_encryption_configuration object supports the following:

- rule - (required) A single object for server-side encryption by default configuration. (documented below)

The rule object supports the following:

- apply_server_side_encryption_by_default - (required) A single object for setting server-side encryption by default. (documented below)

The apply_server_side_encryption_by_default object supports the following:

- sse_algorithm - (required) The server-side encryption algorithm to use. Valid values are AES256 and aws:kms
- kms_master_key_id - (optional) The AWS KMS master key ID used for the SSE-KMS encryption. This can only be used when you set the value of sse_algorithm as aws:kms. The default aws/s3 AWS KMS master key is used if this element is absent while the sse_algorithm is aws:kms.

The access_control_translation object supports the following:

- owner - (Required) The override value for the owner on replicated objects. Currently only Destination is supported.

The `object_lock_configuration` object supports the following:

- `object_lock_enabled` - (Required) Indicates whether this bucket has an Object Lock configuration enabled. Valid value is `Enabled`.
- `rule` - (Optional) The Object Lock rule in place for this bucket.

The `rule` object supports the following:

- `default_retention` - (Required) The default retention period that you want to apply to new objects placed in this bucket.

The `default_retention` object supports the following:

- `mode` - (Required) The default Object Lock retention mode you want to apply to new objects placed in this bucket. Valid values are `GOVERNANCE` and `COMPLIANCE`.
- `days` - (Optional) The number of days that you want to specify for the default retention period.
- `years` - (Optional) The number of years that you want to specify for the default retention period.

Either `days` or `years` must be specified, but not both.

NOTE on `object_lock_configuration`: You can only enable S3 Object Lock for new buckets. If you need to turn on S3 Object Lock for an existing bucket, please contact AWS Support. When you create a bucket with S3 Object Lock enabled, Amazon S3 automatically enables versioning for the bucket. Once you create a bucket with S3 Object Lock enabled, you can't disable Object Lock or suspend versioning for the bucket.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the bucket.
- `arn` - The ARN of the bucket. Will be of format `arn:aws:s3:::bucketname`.
- `bucket_domain_name` - The bucket domain name. Will be of format `bucketname.s3.amazonaws.com`.
- `bucket_regional_domain_name` - The bucket region-specific domain name. The bucket domain name including the region name, please refer here for format. Note: The AWS CloudFront allows specifying S3 region-specific endpoint when creating S3 origin, it will prevent redirect issues from CloudFront to S3 Origin URL.
- `hosted_zone_id` - The Route 53 Hosted Zone ID for this bucket's region.
- `region` - The AWS region this bucket resides in.
- `website_endpoint` - The website endpoint, if the bucket is configured with a website. If not, this will be an empty string.

- `website_domain` - The domain of the website endpoint, if the bucket is configured with a website. If not, this will be an empty string. This is used to create Route 53 alias records.

» Import

S3 bucket can be imported using the `bucket`, e.g.

```
$ terraform import aws_s3_bucket.bucket bucket-name
```

» Resource: aws_s3_bucket_inventory

Provides a S3 bucket inventory configuration resource.

» Example Usage

» Add inventory configuration

```
resource "aws_s3_bucket" "test" {
  bucket = "my-tf-test-bucket"
}

resource "aws_s3_bucket" "inventory" {
  bucket = "my-tf-inventory-bucket"
}

resource "aws_s3_bucket_inventory" "test" {
  bucket = "${aws_s3_bucket.test.id}"
  name   = "EntireBucketDaily"

  included_object_versions = "All"

  schedule {
    frequency = "Daily"
  }

  destination {
    bucket {
      format     = "ORC"
      bucket_arn = "${aws_s3_bucket.inventory.arn}"
    }
  }
}
```

» Add inventory configuration with S3 bucket object prefix

```
resource "aws_s3_bucket" "test" {
  bucket = "my-tf-test-bucket"
}

resource "aws_s3_bucket" "inventory" {
  bucket = "my-tf-inventory-bucket"
}

resource "aws_s3_bucket_inventory" "test-prefix" {
  bucket = "${aws_s3_bucket.test.id}"
  name   = "DocumentsWeekly"

  included_object_versions = "All"

  schedule {
    frequency = "Daily"
  }

  filter {
    prefix = "documents/"
  }

  destination {
    bucket {
      format     = "ORC"
      bucket_arn = "${aws_s3_bucket.inventory.arn}"
      prefix     = "inventory"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket where the inventory configuration will be stored.
- `name` - (Required) Unique identifier of the inventory configuration for the bucket.
- `included_object_versions` - (Required) Object versions to include in the inventory list. Valid values: `All`, `Current`.
- `schedule` - (Required) Specifies the schedule for generating inventory results (documented below).

- destination - (Required) Contains information about where to publish the inventory results (documented below).
- enabled - (Optional, Default: true) Specifies whether the inventory is enabled or disabled.
- filter - (Optional) Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria (documented below).
- optional_fields - (Optional) List of optional fields that are included in the inventory results. Valid values: Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode, ObjectLockLegalHoldStatus, IntelligentTieringAccessTier.

The filter configuration supports the following:

- prefix - (Optional) The prefix that an object must have to be included in the inventory results.

The schedule configuration supports the following:

- frequency - (Required) Specifies how frequently inventory results are produced. Valid values: Daily, Weekly.

The destination configuration supports the following:

- bucket - (Required) The S3 bucket configuration where inventory results are published (documented below).

The bucket configuration supports the following:

- bucket_arn - (Required) The Amazon S3 bucket ARN of the destination.
- format - (Required) Specifies the output format of the inventory results. Can be CSV, ORC or Parquet.
- account_id - (Optional) The ID of the account that owns the destination bucket. Recommended to be set to prevent problems if the destination bucket ownership changes.
- prefix - (Optional) The prefix that is prepended to all inventory results.
- encryption - (Optional) Contains the type of server-side encryption to use to encrypt the inventory (documented below).

The encryption configuration supports the following:

- sse_kms - (Optional) Specifies to use server-side encryption with AWS KMS-managed keys to encrypt the inventory file (documented below).
- sse_s3 - (Optional) Specifies to use server-side encryption with Amazon S3-managed keys (SSE-S3) to encrypt the inventory file.

The sse_kms configuration supports the following:

- key_id - (Required) The ARN of the KMS customer master key (CMK) used to encrypt the inventory file.

» Import

S3 bucket inventory configurations can be imported using `bucket:inventory`, e.g.

```
$ terraform import aws_s3_bucket_inventory.my-bucket-entire-bucket my-bucket:EntireBucket
```

» Resource: aws_s3_bucket_metric

Provides a S3 bucket metrics configuration resource.

» Example Usage

» Add metrics configuration for entire S3 bucket

```
resource "aws_s3_bucket" "example" {
  bucket = "example"
}

resource "aws_s3_bucket_metric" "example-entire-bucket" {
  bucket = "${aws_s3_bucket.example.bucket}"
  name   = "EntireBucket"
}
```

» Add metrics configuration with S3 bucket object filter

```
resource "aws_s3_bucket" "example" {
  bucket = "example"
}

resource "aws_s3_bucket_metric" "example-filtered" {
  bucket = "${aws_s3_bucket.example.bucket}"
  name   = "ImportantBlueDocuments"

  filter {
    prefix = "documents/"

    tags = {
      priority = "high"
      class    = "blue"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to put metric configuration.
- `name` - (Required) Unique identifier of the metrics configuration for the bucket.
- `filter` - (Optional) Object filtering that accepts a prefix, tags, or a logical AND of prefix and tags (documented below).

The `filter` metric configuration supports the following:

- `prefix` - (Optional) Object prefix for filtering (singular).
- `tags` - (Optional) Object tags for filtering (up to 10).

» Import

S3 bucket metric configurations can be imported using `bucket:metric`, e.g.

```
$ terraform import aws_s3_bucket_metric.my-bucket-entire-bucket my-bucket:EntireBucket
```

» Resource: aws_s3_bucket_notification

Manages a S3 Bucket Notification Configuration. For additional information, see the Configuring S3 Event Notifications section in the Amazon S3 Developer Guide.

NOTE: S3 Buckets only support a single notification configuration. Declaring multiple `aws_s3_bucket_notification` resources to the same S3 Bucket will cause a perpetual difference in configuration. See the example ”Trigger multiple Lambda functions” for an option.

» Example Usage

» Add notification configuration to SNS Topic

```
resource "aws_sns_topic" "topic" {
  name = "s3-event-notification-topic"

  policy = <<POLICY
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect": "Allow",
        "Principal": {"AWS":"*"},
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:*:*:s3-event-notification-topic",
        "Condition":{
            "ArnLike":{"aws:SourceArn":"${aws_s3_bucket.bucket.arn}"}
        }
    }]
}
POLICY
}

resource "aws_s3_bucket" "bucket" {
  bucket = "your_bucket_name"
}

resource "aws_s3_bucket_notification" "bucket_notification" {
  bucket = "${aws_s3_bucket.bucket.id}"

  topic {
    topic_arn     = "${aws_sns_topic.topic.arn}"
    events        = ["s3:ObjectCreated:*"]
    filter_suffix = ".log"
  }
}
```

» Add notification configuration to SQS Queue

```
resource "aws_sqs_queue" "queue" {
  name = "s3-event-notification-queue"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:*:*:s3-event-notification-queue",
      "Condition": {
        "ArnEquals": { "aws:SourceArn": "${aws_s3_bucket.bucket.arn}" }
      }
    }
  ]
}
POLICY
}

resource "aws_s3_bucket" "bucket" {
  bucket = "your_bucket_name"
}

resource "aws_s3_bucket_notification" "bucket_notification" {
  bucket = "${aws_s3_bucket.bucket.id}"

  queue {
    queue_arn     = "${aws_sqs_queue.queue.arn}"
    events        = ["s3:ObjectCreated:*"]
    filter_suffix = ".log"
  }
}
```

» Add notification configuration to Lambda Function

```
resource "aws_iam_role" "iam_for_lambda" {
  name = "iam_for_lambda"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow"
    }
  ]
}
EOF
}

resource "aws_lambda_permission" "allow_bucket" {
  statement_id  = "AllowExecutionFromS3Bucket"
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.func.arn}"
  principal     = "s3.amazonaws.com"
  source_arn    = "${aws_s3_bucket.bucket.arn}"
}
```

```
resource "aws_lambda_function" "func" {
  filename      = "your-function.zip"
  function_name = "example_lambda_name"
  role          = "${aws_iam_role.iam_for_lambda.arn}"
  handler       = "exports.example"
  runtime       = "go1.x"
}

resource "aws_s3_bucket" "bucket" {
  bucket = "your_bucket_name"
}

resource "aws_s3_bucket_notification" "bucket_notification" {
  bucket = "${aws_s3_bucket.bucket.id}"

  lambda_function {
    lambda_function_arn = "${aws_lambda_function.func.arn}"
    events              = ["s3:ObjectCreated:*"]
    filter_prefix       = "AWSLogs/"
    filter_suffix       = ".log"
  }
}
```

» Trigger multiple Lambda functions

```
resource "aws_iam_role" "iam_for_lambda" {
  name = "iam_for_lambda"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow"
    }
  ]
}
EOF
}

resource "aws_lambda_permission" "allow_bucket1" {
  statement_id  = "AllowExecutionFromS3Bucket1"
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.func1.arn}"
  principal     = "s3.amazonaws.com"
  source_arn    = "${aws_s3_bucket.bucket.arn}"
}

resource "aws_lambda_function" "func1" {
  filename      = "your-function1.zip"
  function_name = "example_lambda_name1"
  role          = "${aws_iam_role.iam_for_lambda.arn}"
  handler       = "exports.example"
  runtime       = "go1.x"
}

resource "aws_lambda_permission" "allow_bucket2" {
  statement_id  = "AllowExecutionFromS3Bucket2"
  action        = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function.func2.arn}"
  principal     = "s3.amazonaws.com"
  source_arn    = "${aws_s3_bucket.bucket.arn}"
}

resource "aws_lambda_function" "func2" {
  filename      = "your-function2.zip"
  function_name = "example_lambda_name2"
  role          = "${aws_iam_role.iam_for_lambda.arn}"
  handler       = "exports.example"
}

resource "aws_s3_bucket" "bucket" {
  bucket = "your_bucket_name"
}

resource "aws_s3_bucket_notification" "bucket_notification" {
  bucket = "${aws_s3_bucket.bucket.id}"

  lambda_function {
    lambda_function_arn = "${aws_lambda_function.func1.arn}"
    events              = ["s3:ObjectCreated:*"]
    filter_prefix       = "AWSLogs/"
    filter_suffix       = ".log"
  }

  lambda_function {
    lambda_function_arn = "${aws_lambda_function.func2.arn}"
    events              = ["s3:ObjectCreated:*"]
    filter_prefix       = "OtherLogs/"
    filter_suffix       = ".log"
  }
}
```

» Add multiple notification configurations to SQS Queue

```
resource "aws_sqs_queue" "queue" {
  name = "s3-event-notification-queue"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:*:*:s3-event-notification-queue",
      "Condition": {
        "ArnEquals": { "aws:SourceArn": "${aws_s3_bucket.bucket.arn}" }
      }
    }
  ]
}
POLICY
}

resource "aws_s3_bucket" "bucket" {
  bucket = "your_bucket_name"
}

resource "aws_s3_bucket_notification" "bucket_notification" {
  bucket = "${aws_s3_bucket.bucket.id}"

  queue {
    id            = "image-upload-event"
    queue_arn     = "${aws_sqs_queue.queue.arn}"
    events        = ["s3:ObjectCreated:*"]
    filter_prefix = "images/"
  }

  queue {
    id            = "video-upload-event"
    queue_arn     = "${aws_sqs_queue.queue.arn}"
    events        = ["s3:ObjectCreated:*"]
    filter_prefix = "videos/"
  }
}
```

For Terraform's JSON syntax, use an array instead of defining the `queue` key twice.

```
{
    "bucket": "${aws_s3_bucket.bucket.id}",
    "queue": [
        {
            "id": "image-upload-event",
            "queue_arn": "${aws_sqs_queue.queue.arn}",
            "events": ["s3:ObjectCreated:*"],
            "filter_prefix": "images/"
        },
        {
            "id": "video-upload-event",
            "queue_arn": "${aws_sqs_queue.queue.arn}",
            "events": ["s3:ObjectCreated:*"],
            "filter_prefix": "videos/"
        }
    ]
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to put notification configuration.
- `topic` - (Optional) The notification configuration to SNS Topic (documented below).
- `queue` - (Optional) The notification configuration to SQS Queue (documented below).
- `lambda_function` - (Optional, Multiple) Used to configure notifications to a Lambda Function (documented below).

The `topic` notification configuration supports the following:

- `id` - (Optional) Specifies unique identifier for each of the notification configurations.
- `topic_arn` - (Required) Specifies Amazon SNS topic ARN.
- `events` - (Required) Specifies event for which to send notifications.
- `filter_prefix` - (Optional) Specifies object key name prefix.

- `filter_suffix` - (Optional) Specifies object key name suffix.

The `queue` notification configuration supports the following:

- `id` - (Optional) Specifies unique identifier for each of the notification configurations.
- `queue_arn` - (Required) Specifies Amazon SQS queue ARN.
- `events` - (Required) Specifies event for which to send notifications.
- `filter_prefix` - (Optional) Specifies object key name prefix.
- `filter_suffix` - (Optional) Specifies object key name suffix.

The `lambda_function` notification configuration supports the following:

- `id` - (Optional) Specifies unique identifier for each of the notification configurations.
- `lambda_function_arn` - (Required) Specifies Amazon Lambda function ARN.
- `events` - (Required) Specifies event for which to send notifications.
- `filter_prefix` - (Optional) Specifies object key name prefix.
- `filter_suffix` - (Optional) Specifies object key name suffix.

» Import

S3 bucket notification can be imported using the `bucket`, e.g.

```
$ terraform import aws_s3_bucket_notification.bucket_notification bucket-name
```

» Resource: aws_s3_bucket_object

Provides a S3 bucket object resource.

» Example Usage

» Uploading a file to a bucket

```
resource "aws_s3_bucket_object" "object" {
  bucket = "your_bucket_name"
  key    = "new_object_key"
  source = "path/to/file"

  # The filemd5() function is available in Terraform 0.11.12 and later
  # For Terraform 0.11.11 and earlier, use the md5() function and the file() function:
  # etag = "${md5(file("path/to/file"))}"
  etag = "${filemd5("path/to/file")}"
}
```

» Encrypting with KMS Key

```
resource "aws_kms_key" "examplekms" {
  description             = "KMS key 1"
  deletion_window_in_days = 7
}

resource "aws_s3_bucket" "examplebucket" {
  bucket = "examplebuckettftest"
  acl    = "private"
}

resource "aws_s3_bucket_object" "examplebucket_object" {
  key        = "someobject"
  bucket     = "${aws_s3_bucket.examplebucket.id}"
  source     = "index.html"
  kms_key_id = "${aws_kms_key.examplekms.arn}"
}
```

» Server Side Encryption with S3 Default Master Key

```
resource "aws_s3_bucket" "examplebucket" {
  bucket = "examplebuckettftest"
  acl    = "private"
}

resource "aws_s3_bucket_object" "examplebucket_object" {
  key                    = "someobject"
  bucket                 = "${aws_s3_bucket.examplebucket.id}"
  source                 = "index.html"
  server_side_encryption = "aws:kms"
}
```

» Server Side Encryption with AWS-Managed Key

```
resource "aws_s3_bucket" "examplebucket" {
  bucket = "examplebuckettftest"
  acl    = "private"
}

resource "aws_s3_bucket_object" "examplebucket_object" {
  key                    = "someobject"
  bucket                 = "${aws_s3_bucket.examplebucket.id}"
  source                 = "index.html"
  server_side_encryption = "AES256"
}
```

» S3 Object Lock

```
resource "aws_s3_bucket" "examplebucket" {
  bucket = "examplebuckettftest"
  acl    = "private"

  versioning {
    enabled = true
  }

  object_lock_configuration {
    object_lock_enabled = "Enabled"
  }
}

resource "aws_s3_bucket_object" "examplebucket_object" {
  key    = "someobject"
  bucket = "${aws_s3_bucket.examplebucket.id}"
  source = "important.txt"

  object_lock_legal_hold_status = "ON"
  object_lock_mode              = "GOVERNANCE"
  object_lock_retain_until_date = "2021-12-31T23:59:60Z"

  force_destroy = true
}
```

» Argument Reference

Note: If you specify `content_encoding` you are responsible for encoding the body appropriately. `source`, `content`, and `content_base64` all expect already encoded/compressed bytes.

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to put the file in.
- `key` - (Required) The name of the object once it is in the bucket.
- `source` - (Optional, conflicts with `content` and `content_base64`) The path to a file that will be read and uploaded as raw bytes for the object content.

- `content` - (Optional, conflicts with `source` and `content_base64`) Literal string value to use as the object content, which will be uploaded as UTF-8-encoded text.
- `content_base64` - (Optional, conflicts with `source` and `content`) Base64-encoded data that will be decoded and uploaded as raw bytes for the object content. This allows safely uploading non-UTF8 binary data, but is recommended only for small content such as the result of the `gzipbase64` function with small text strings. For larger objects, use `source` to stream the content from a disk file.
- `acl` - (Optional) The canned ACL to apply. Defaults to "private".
- `cache_control` - (Optional) Specifies caching behavior along the request/reply chain Read w3c cache_control for further details.
- `content_disposition` - (Optional) Specifies presentational information for the object. Read w3c content_disposition for further information.
- `content_encoding` - (Optional) Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field. Read w3c content encoding for further information.
- `content_language` - (Optional) The language the content is in e.g. en-US or en-GB.
- `content_type` - (Optional) A standard MIME type describing the format of the object data, e.g. application/octet-stream. All Valid MIME Types are valid for this input.
- `website_redirect` - (Optional) Specifies a target URL for website redirect.
- `storage_class` - (Optional) Specifies the desired Storage Class for the object. Can be either "`STANDARD`", "`REDUCED_REDUNDANCY`", "`ONEZONE_IA`", "`INTELLIGENT_TIERING`", "`GLACIER`", "`DEEP_ARCHIVE`", or "`STANDARD_IA`". Defaults to "`STANDARD`".
- `etag` - (Optional) Used to trigger updates. The only meaningful value is `${filemd5("path/to/file")}` (Terraform 0.11.12 or later) or `${md5(file("path/to/file"))}` (Terraform 0.11.11 or earlier). This attribute is not compatible with KMS encryption, `kms_key_id` or `server_side_encryption = "aws:kms"`.
- `server_side_encryption` - (Optional) Specifies server-side encryption of the object in S3. Valid values are "`AES256`" and "`aws:kms`".
- `kms_key_id` - (Optional) Specifies the AWS KMS Key ARN to use for object encryption. This value is a fully qualified **ARN** of the KMS Key. If using `aws_kms_key`, use the exported `arn` attribute: `kms_key_id = "${aws_kms_key.foo.arn}"`
- `metadata` - (Optional) A mapping of keys/values to provision metadata (will be automatically prefixed by `x-amz-meta-`, note that only lowercase label are currently supported by the AWS Go API).
- `tags` - (Optional) A mapping of tags to assign to the object.
- `force_destroy` - (Optional) Allow the object to be deleted by removing any legal hold on any object version. Default is `false`. This value should

be set to `true` only if the bucket has S3 object lock enabled.
- `object_lock_legal_hold_status` - (Optional) The legal hold status that you want to apply to the specified object. Valid values are `ON` and `OFF`.
- `object_lock_mode` - (Optional) The object lock retention mode that you want to apply to this object. Valid values are `GOVERNANCE` and `COMPLIANCE`.
- `object_lock_retain_until_date` - (Optional) The date and time, in RFC3339 format, when this object's object lock will expire.

If no content is provided through `source`, `content` or `content_base64`, then the object will be empty.

Note: Terraform ignores all leading `/`s in the object's `key` and treats multiple `/`s in the rest of the object's `key` as a single `/`, so values of `/index.html` and `index.html` correspond to the same S3 object as do `first//second///third//` and `first/second/third/`.

» Attributes Reference

The following attributes are exported

- `id` - the `key` of the resource supplied above
- `etag` - the ETag generated for the object (an MD5 sum of the object content). For plaintext objects or objects encrypted with an AWS-managed key, the hash is an MD5 digest of the object data. For objects encrypted with a KMS key or objects created by either the Multipart Upload or Part Copy operation, the hash is not an MD5 digest, regardless of the method of encryption. More information on possible values can be found on Common Response Headers.
- `version_id` - A unique version ID value for the object, if bucket versioning is enabled.

» Resource: aws_s3_bucket_policy

Attaches a policy to an S3 bucket resource.

» Example Usage

» Basic Usage

```
resource "aws_s3_bucket" "b" {
  bucket = "my_tf_test_bucket"
}
```

```
resource "aws_s3_bucket_policy" "b" {
  bucket = "${aws_s3_bucket.b.id}"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Id": "MYBUCKETPOLICY",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::my_tf_test_bucket/*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "8.8.8.8/32"}
      }
    }
  ]
}
POLICY
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to which to apply the policy.
- `policy` - (Required) The text of the policy. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

» Import

S3 bucket policies can be imported using the bucket name, e.g.

```
$ terraform import aws_s3_bucket_policy.example my-bucket-name
```

» Resource: aws_s3_bucket_public_access_block

Manages S3 bucket-level Public Access Block configuration. For more information about these settings, see the AWS S3 Block Public Access documentation.

» Example Usage

```
resource "aws_s3_bucket" "example" {
  bucket = "example"
}

resource "aws_s3_bucket_public_access_block" "example" {
  bucket = "${aws_s3_bucket.example.id}"

  block_public_acls   = true
  block_public_policy = true
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) S3 Bucket to which this Public Access Block configuration should be applied.
- `block_public_acls` - (Optional) Whether Amazon S3 should block public ACLs for this bucket. Defaults to `false`. Enabling this setting does not affect existing policies or ACLs. When set to `true` causes the following behavior:
  - PUT Bucket acl and PUT Object acl calls will fail if the specified ACL allows public access.
  - PUT Object calls will fail if the request includes an object ACL.
- `block_public_policy` - (Optional) Whether Amazon S3 should block public bucket policies for this bucket. Defaults to `false`. Enabling this setting does not affect the existing bucket policy. When set to `true` causes Amazon S3 to:
  - Reject calls to PUT Bucket policy if the specified bucket policy allows public access.
- `ignore_public_acls` - (Optional) Whether Amazon S3 should ignore public ACLs for this bucket. Defaults to `false`. Enabling this setting does not affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set. When set to `true` causes Amazon S3 to:
  - Ignore public ACLs on this bucket and any objects that it contains.
- `restrict_public_buckets` - (Optional) Whether Amazon S3 should restrict public bucket policies for this bucket. Defaults to `false`. Enabling this setting does not affect the previously stored bucket policy, except that public and cross-account access within the public bucket policy, including non-public delegation to specific accounts, is blocked. When set to `true`:
  - Only the bucket owner and AWS Services can access this buckets if it has a public policy.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Name of the S3 bucket the configuration is attached to

» Import

`aws_s3_bucket_public_access_block` can be imported by using the bucket name, e.g.

```
$ terraform import aws_s3_bucket_public_access_block.example my-bucket
```

» Resource: aws_sagemaker_endpoint

Provides a SageMaker Endpoint resource.

» Example Usage

Basic usage:

```
resource "aws_sagemaker_endpoint" "e" {
  name                 = "my-endpoint"
  endpoint_config_name = "${aws_sagemaker_endpoint_configuration.ec.name}"

  tags = {
    Name = "foo"
  }
}
```

» Argument Reference

The following arguments are supported:

- `endpoint_config_name` - (Required) The name of the endpoint configuration to use.
- `name` - (Optional) The name of the endpoint. If omitted, Terraform will assign a random, unique name.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) assigned by AWS to this endpoint.
- `name` - The name of the endpoint.

» Import

Endpoints can be imported using the `name`, e.g.

```
$ terraform import aws_sagemaker_endpoint.test_endpoint my-endpoint
```

» Resource: aws_sagemaker_endpoint_configuration

Provides a SageMaker endpoint configuration resource.

» Example Usage

Basic usage:

```
resource "aws_sagemaker_endpoint_configuration" "ec" {
  name = "my-endpoint-config"

  production_variants {
    variant_name           = "variant-1"
    model_name             = "${aws_sagemaker_model.m.name}"
    initial_instance_count = 1
    instance_type          = "ml.t2.medium"
  }

  tags {
    Name = "foo"
  }
}
```

» Argument Reference

The following arguments are supported:

- `production_variants` - (Required) Fields are documented below.

- kms_key_arn - (Optional) Amazon Resource Name (ARN) of a AWS Key Management Service key that Amazon SageMaker uses to encrypt data on the storage volume attached to the ML compute instance that hosts the endpoint.
- name - (Optional) The name of the endpoint configuration. If omitted, Terraform will assign a random, unique name.
- tags - (Optional) A mapping of tags to assign to the resource.

The production_variants block supports:

- initial_instance_count - (Required) Initial number of instances used for auto-scaling.
- instance_type (Required) - The type of instance to start.
- accelerator_type (Optional) - The size of the Elastic Inference (EI) instance to use for the production variant.
- initial_variant_weight (Optional) - Determines initial traffic distribution among all of the models that you specify in the endpoint configuration. If unspecified, it defaults to 1.0.
- model_name - (Required) The name of the model to use.
- variant_name - (Optional) The name of the variant. If omitted, Terraform will assign a random, unique name. ## Attributes Reference

The following attributes are exported:

- arn - The Amazon Resource Name (ARN) assigned by AWS to this endpoint configuration.
- name - The name of the endpoint configuration.

» Import

Endpoint configurations can be imported using the name, e.g.

```
$ terraform import aws_sagemaker_endpoint_configuration.test_endpoint_config endpoint-config
```

» Resource: aws_sagemaker_model

Provides a SageMaker model resource.

» Example Usage

Basic usage:

```
resource "aws_sagemaker_model" "m" {
  name               = "my-model"
  execution_role_arn = "${aws_iam_role.foo.arn}"
  primary_container {
    image = "174872318107.dkr.ecr.us-west-2.amazonaws.com/kmeans:1"
  }
}

resource "aws_iam_role" "r" {
  assume_role_policy = "${data.aws_iam_policy_document.assume_role.json}"
}

data "aws_iam_policy_document" "assume_role" {
  statement {
    actions = ["sts:AssumeRole"]

    principals {
      type        = "Service"
      identifiers = ["sagemaker.amazonaws.com"]
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the model (must be unique). If omitted, Terraform will assign a random, unique name.
- `primary_container` - (Optional) The primary docker image containing inference code that is used when the model is deployed for predictions. If not specified, the `container` argument is required. Fields are documented below.
- `execution_role_arn` - (Required) A role that SageMaker can assume to access model artifacts and docker images for deployment.
- `container` (Optional) - Specifies containers in the inference pipeline. If not specified, the `primary_container` argument is required. Fields are documented below.
- `enable_network_isolation` (Optional) - Isolates the model container. No inbound or outbound network calls can be made to or from the model container.
- `vpc_config` (Optional) - Specifies the VPC that you want your model to connect to. VpcConfig is used in hosting services and in batch transform.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `primary_container` and `container` block both support:

- `image` - (Required) The registry path where the inference code image is

stored in Amazon ECR.
- `model_data_url` - (Optional) The URL for the S3 location where model artifacts are stored.
- `container_hostname` - (Optional) The DNS host name for the container.
- `environment` - (Optional) Environment variables for the Docker container. A list of key value pairs.

» Attributes Reference

The following attributes are exported:

- `name` - The name of the model.
- `arn` - The Amazon Resource Name (ARN) assigned by AWS to this model.

» Import

Models can be imported using the `name`, e.g.

```
$ terraform import aws_sagemaker_model.test_model model-foo
```

» Resource: aws_sagemaker_notebook_instance

Provides a Sagemaker Notebook Instance resource.

» Example Usage

Basic usage:

```
resource "aws_sagemaker_notebook_instance" "ni" {
  name          = "my-notebook-instance"
  role_arn      = "${aws_iam_role.role.arn}"
  instance_type = "ml.t2.medium"

  tags = {
    Name = "foo"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the notebook instance (must be unique).

- `role_arn` - (Required) The ARN of the IAM role to be used by the notebook instance which allows SageMaker to call other services on your behalf.
- `instance_type` - (Required) The name of ML compute instance type.
- `subnet_id` - (Optional) The VPC subnet ID.
- `security_groups` - (Optional) The associated security groups.
- `kms_key_id` - (Optional) The AWS Key Management Service (AWS KMS) key that Amazon SageMaker uses to encrypt the model artifacts at rest using Amazon S3 server-side encryption.
- `lifecycle_config_name` - (Optional) The name of a lifecycle configuration to associate with the notebook instance.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `id` - The name of the notebook instance.
- `arn` - The Amazon Resource Name (ARN) assigned by AWS to this notebook instance.

» Import

Sagemaker Notebook Instances can be imported using the `name`, e.g.

```
$ terraform import aws_sagemaker_notebook_instance.test_notebook_instance my-notebook-instan
```

» Resource: aws_sagemaker_notebook_instance_lifecycle_configur

Provides a lifecycle configuration for SageMaker Notebook Instances.

» Example Usage

Usage:

```
resource "aws_sagemaker_notebook_instance_lifecycle_configuration" "lc" {
  name      = "foo"
  on_create = "${base64encode('echo foo')}"
  on_start  = "${base64encode('echo bar')}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the lifecycle configuration (must be unique). If omitted, Terraform will assign a random, unique name.
- `on_create` - (Optional) A shell script (base64-encoded) that runs only once when the SageMaker Notebook Instance is created.
- `on_start` - (Optional) A shell script (base64-encoded) that runs every time the SageMaker Notebook Instance is started including the time it's created.

» Attributes Reference

The following attributes are exported:

- `arn` - The Amazon Resource Name (ARN) assigned by AWS to this lifecycle configuration.

» Import

Models can be imported using the `name`, e.g.

```
$ terraform import aws_sagemaker_notebook_instance_lifecycle_configuration.lc foo
```

» Data Source: aws_secretsmanager_secret

Retrieve metadata information about a Secrets Manager secret. To retrieve a secret value, see the `aws_secretsmanager_secret_version` data source.

» Example Usage

» ARN

```
data "aws_secretsmanager_secret" "by-arn" {
  arn = "arn:aws:secretsmanager:us-east-1:123456789012:secret:example-123456"
}
```

» Name

```
data "aws_secretsmanager_secret" "by-name" {
  name = "example"
}
```

» Argument Reference

- `arn` - (Optional) The Amazon Resource Name (ARN) of the secret to retrieve.
- `name` - (Optional) The name of the secret to retrieve.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) of the secret.
- `description` - A description of the secret.
- `kms_key_id` - The Key Management Service (KMS) Customer Master Key (CMK) associated with the secret.
- `id` - The Amazon Resource Name (ARN) of the secret.
- `rotation_enabled` - Whether rotation is enabled or not.
- `rotation_lambda_arn` - Rotation Lambda function Amazon Resource Name (ARN) if rotation is enabled.
- `rotation_rules` - Rotation rules if rotation is enabled.
- `tags` - Tags of the secret.
- `policy` - The resource-based policy document that's attached to the secret.

» Data Source: aws_secretsmanager_secret_version

Retrieve information about a Secrets Manager secret version, including its secret value. To retrieve secret metadata, see the `aws_secretsmanager_secret` data source.

» Example Usage

» Retrieve Current Secret Version

By default, this data sources retrieves information based on the `AWSCURRENT` staging label.

```
data "aws_secretsmanager_secret_version" "example" {
  secret_id = "${data.aws_secretsmanager_secret.example.id}"
}
```

» Retrieve Specific Secret Version

```
data "aws_secretsmanager_secret_version" "by-version-stage" {
  secret_id     = "${data.aws_secretsmanager_secret.example.id}"
  version_stage = "example"
}
```

» Handling Key-Value Secret Strings in JSON

Reading key-value pairs from JSON back into a native Terraform map can be accomplished in Terraform 0.12 and later with the `jsondecode()` function:

```
output "example" {
  value = jsondecode(data.aws_secretsmanager_secret_version.example.secret_string)["key1"]
}
```

» Argument Reference

- `secret_id` - (Required) Specifies the secret containing the version that you want to retrieve. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret.
- `version_id` - (Optional) Specifies the unique identifier of the version of the secret that you want to retrieve. Overrides `version_stage`.
- `version_stage` - (Optional) Specifies the secret version that you want to retrieve by the staging label attached to the version. Defaults to `AWSCURRENT`.

» Attributes Reference

- `arn` - The ARN of the secret.
- `id` - The unique identifier of this version of the secret.
- `secret_string` - The decrypted part of the protected secret information that was originally provided as a string.
- `secret_binary` - The decrypted part of the protected secret information that was originally provided as a binary. Base64 encoded.
- `version_id` - The unique identifier of this version of the secret.

» Resource: aws_secretsmanager_secret

Provides a resource to manage AWS Secrets Manager secret metadata. To manage a secret value, see the `aws_secretsmanager_secret_version` resource.

» Example Usage

» Basic

```
resource "aws_secretsmanager_secret" "example" {
  name = "example"
}
```

» Rotation Configuration

To enable automatic secret rotation, the Secrets Manager service requires usage of a Lambda function. The Rotate Secrets section in the Secrets Manager User Guide provides additional information about deploying a prebuilt Lambda functions for supported credential rotation (e.g. RDS) or deploying a custom Lambda function.

NOTE: Configuring rotation causes the secret to rotate once as soon as you store the secret. Before you do this, you must ensure that all of your applications that use the credentials stored in the secret are updated to retrieve the secret from AWS Secrets Manager. The old credentials might no longer be usable after the initial rotation and any applications that you fail to update will break as soon as the old credentials are no longer valid.

NOTE: If you cancel a rotation that is in progress (by removing the `rotation` configuration), it can leave the VersionStage labels in an unexpected state. Depending on what step of the rotation was in progress, you might need to remove the staging label AWSPENDING from the partially created version, specified by the SecretVersionId response value. You should also evaluate the partially rotated new version to see if it should be deleted, which you can do by removing all staging labels from the new version's VersionStage field.

```
resource "aws_secretsmanager_secret" "rotation-example" {
  name                = "rotation-example"
  rotation_lambda_arn = "${aws_lambda_function.example.arn}"

  rotation_rules {
    automatically_after_days = 7
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) Specifies the friendly name of the new secret. The secret name can consist of uppercase letters, lowercase letters, digits, and any of the following characters: `/_+=.@-` Conflicts with `name_prefix`.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `description` - (Optional) A description of the secret.
- `kms_key_id` - (Optional) Specifies the ARN or alias of the AWS KMS customer master key (CMK) to be used to encrypt the secret values in the versions stored in this secret. If you don't specify this value, then Secrets Manager defaults to using the AWS account's default CMK (the one named `aws/secretsmanager`). If the default KMS CMK with that name doesn't yet exist, then AWS Secrets Manager creates it for you automatically the first time.
- `policy` - (Optional) A valid JSON document representing a resource policy. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.
- `recovery_window_in_days` - (Optional) Specifies the number of days that AWS Secrets Manager waits before it can delete the secret. This value can be `0` to force deletion without recovery or range from `7` to `30` days. The default value is `30`.
- `rotation_lambda_arn` - (Optional) Specifies the ARN of the Lambda function that can rotate the secret.
- `rotation_rules` - (Optional) A structure that defines the rotation configuration for this secret. Defined below.
- `tags` - (Optional) Specifies a key-value map of user-defined tags that are attached to the secret.

» rotation_rules

- `automatically_after_days` - (Required) Specifies the number of days between automatic scheduled rotations of the secret.

» Attribute Reference

- `id` - Amazon Resource Name (ARN) of the secret.
- `arn` - Amazon Resource Name (ARN) of the secret.
- `rotation_enabled` - Specifies whether automatic rotation is enabled for this secret.

» Import

`aws_secretsmanager_secret` can be imported by using the secret Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_secretsmanager_secret.example arn:aws:secretsmanager:us-east-1:123456
```

» Resource: aws_secretsmanager_secret_version

Provides a resource to manage AWS Secrets Manager secret version including its secret value. To manage secret metadata, see the `aws_secretsmanager_secret` resource.

NOTE: If the `AWSCURRENT` staging label is present on this version during resource deletion, that label cannot be removed and will be skipped to prevent errors when fully deleting the secret. That label will leave this secret version active even after the resource is deleted from Terraform unless the secret itself is deleted. Move the `AWSCURRENT` staging label before or after deleting this resource from Terraform to fully trigger version deprecation if necessary.

» Example Usage

» Simple String Value

```
resource "aws_secretsmanager_secret_version" "example" {
  secret_id     = "${aws_secretsmanager_secret.example.id}"
  secret_string = "example-string-to-protect"
}
```

» Key-Value Pairs

Secrets Manager also accepts key-value pairs in JSON.

```
# The map here can come from other supported configurations
# like locals, resource attribute, map() built-in, etc.
variable "example" {
  default = {
    key1 = "value1"
    key2 = "value2"
  }

  type = "map"
}

resource "aws_secretsmanager_secret_version" "example" {
  secret_id     = "${aws_secretsmanager_secret.example.id}"
  secret_string = "${jsonencode(var.example)}"
}
```

Reading key-value pairs from JSON back into a native Terraform map can be accomplished in Terraform 0.12 and later with the `jsondecode()` function:

```
output "example" {
  value = jsondecode(aws_secretsmanager_secret_version.example.secret_string)["key1"]
}
```

» Argument Reference

The following arguments are supported:

- `secret_id` - (Required) Specifies the secret to which you want to add a new version. You can specify either the Amazon Resource Name (ARN) or the friendly name of the secret. The secret must already exist.
- `secret_string` - (Optional) Specifies text data that you want to encrypt and store in this version of the secret. This is required if secret_binary is not set.
- `secret_binary` - (Optional) Specifies binary data that you want to encrypt and store in this version of the secret. This is required if secret_string is not set. Needs to be encoded to base64.
- `version_stages` - (Optional) Specifies a list of staging labels that are attached to this version of the secret. A staging label must be unique to a single version of the secret. If you specify a staging label that's already associated with a different version of the same secret then that staging label is automatically removed from the other version and attached to this version. If you do not specify a value, then AWS Secrets Manager automatically moves the staging label `AWSCURRENT` to this new version on creation.

NOTE: If `version_stages` is configured, you must include the `AWSCURRENT` staging label if this secret version is the only version or if the label is currently present on this secret version, otherwise Terraform will show a perpetual difference.

» Attribute Reference

- `arn` - The ARN of the secret.
- `id` - A pipe delimited combination of secret ID and version ID.
- `version_id` - The unique identifier of the version of the secret.

» Import

`aws_secretsmanager_secret_version` can be imported by using the secret ID and version ID, e.g.

```
$ terraform import aws_secretsmanager_secret_version.example arn:aws:secretsmanager:us-east-
```

» Resource: aws_securityhub_account

Enables Security Hub for this AWS account.

NOTE: Destroying this resource will disable Security Hub for this AWS account.

NOTE: This AWS service is in Preview and may change before General Availability release. Backwards compatibility is not guaranteed between Terraform AWS Provider releases.

» Example Usage

```
resource "aws_securityhub_account" "example" {}
```

» Argument Reference

The resource does not support any arguments.

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - AWS Account ID.

» Import

An existing Security Hub enabled account can be imported using the AWS account ID, e.g.

```
$ terraform import aws_securityhub_account.example 123456789012
```

» Resource: aws_securityhub_product_subscription

Subscribes to a Security Hub product.

NOTE: This AWS service is in Preview and may change before General Availability release. Backwards compatibility is not guaranteed between Terraform AWS Provider releases.

» Example Usage

```
resource "aws_securityhub_account" "example" {}

data "aws_region" "current" {}

resource "aws_securityhub_product_subscription" "example" {
  depends_on  = ["aws_securityhub_account.example"]
  product_arn = "arn:aws:securityhub:${data.aws_region.current.name}:733251395267:product/al
}
```

» Argument Reference

The following arguments are supported:

- `product_arn` - (Required) The ARN of the product that generates findings that you want to import into Security Hub - see below.

Currently available products (remember to replace `${var.region}` as appropriate):

- `arn:aws:securityhub:${var.region}::product/aws/guardduty`
- `arn:aws:securityhub:${var.region}::product/aws/inspector`
- `arn:aws:securityhub:${var.region}::product/aws/macie`
- `arn:aws:securityhub:${var.region}:733251395267:product/alertlogic/althreatmanagement`
- `arn:aws:securityhub:${var.region}:679703615338:product/armordefense/armoranywhere`
- `arn:aws:securityhub:${var.region}:151784055945:product/barracuda/cloudsecurityguardian`
- `arn:aws:securityhub:${var.region}:758245563457:product/checkpoint/cloudguard-iaas`
- `arn:aws:securityhub:${var.region}:634729597623:product/checkpoint/dome9-arc`
- `arn:aws:securityhub:${var.region}:517716713836:product/crowdstrike/crowdstrike-falcon`
- `arn:aws:securityhub:${var.region}:749430749651:product/cyberark/cyberark-pta`
- `arn:aws:securityhub:${var.region}:250871914685:product/f5networks/f5-advanced-waf`
- `arn:aws:securityhub:${var.region}:123073262904:product/fortinet/fortigate`
- `arn:aws:securityhub:${var.region}:324264561773:product/guardicore/aws-infection-monkey`
- `arn:aws:securityhub:${var.region}:324264561773:product/guardicore/guardicore`
- `arn:aws:securityhub:${var.region}:949680696695:product/ibm/qradar-siem`
- `arn:aws:securityhub:${var.region}:955745153808:product/imperva/imperva-attack-analytics`
- `arn:aws:securityhub:${var.region}:297986523463:product/mcafee-skyhigh/mcafee-mvision-cl`
- `arn:aws:securityhub:${var.region}:188619942792:product/paloaltonetworks/redlock`
- `arn:aws:securityhub:${var.region}:122442690527:product/paloaltonetworks/vm-series`
- `arn:aws:securityhub:${var.region}:805950163170:product/qualys/qualys-pc`
- `arn:aws:securityhub:${var.region}:805950163170:product/qualys/qualys-vm`
- `arn:aws:securityhub:${var.region}:336818582268:product/rapid7/insightvm`
- `arn:aws:securityhub:${var.region}:062897671886:product/sophos/sophos-server-protection`
- `arn:aws:securityhub:${var.region}:112543817624:product/splunk/splunk-enterprise`
- `arn:aws:securityhub:${var.region}:112543817624:product/splunk/splunk-phantom`

- `arn:aws:securityhub:${var.region}:956882708938:product/sumologicinc/sumologic-mda`
- `arn:aws:securityhub:${var.region}:754237914691:product/symantec-corp/symantec-cwp`
- `arn:aws:securityhub:${var.region}:422820575223:product/tenable/tenable-io`
- `arn:aws:securityhub:${var.region}:679593333241:product/trend-micro/deep-security`
- `arn:aws:securityhub:${var.region}:453761072151:product/turbot/turbot`
- `arn:aws:securityhub:${var.region}:496947949261:product/twistlock/twistlock-enterprise`

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `arn` - The ARN of a resource that represents your subscription to the product that generates the findings that you want to import into Security Hub.

» Import

Security Hub product subscriptions can be imported in the form `product_arn,arn`, e.g.

```
$ terraform import aws_securityhub_product_subscription.example arn:aws:securityhub:eu-west-
```

» Resource: aws_securityhub_standards_subscription

Subscribes to a Security Hub standard.

NOTE: This AWS service is in Preview and may change before General Availability release. Backwards compatibility is not guaranteed between Terraform AWS Provider releases.

» Example Usage

```
resource "aws_securityhub_account" "example" {}

resource "aws_securityhub_standards_subscription" "example" {
  depends_on    = ["aws_securityhub_account.example"]
  standards_arn = "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0"
}
```

» Argument Reference

The following arguments are supported:

- `standards_arn` - (Required) The ARN of a standard - see below.

Currently available standards:

Name	ARN
CIS AWS Foundations	`arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0`

» Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `id` - The ARN of a resource that represents your subscription to a supported standard.

» Import

Security Hub standards subscriptions can be imported using the standards subscription ARN, e.g.

```
$ terraform import aws_securityhub_standards_subscription.example arn:aws:securityhub:eu-wes
```

» Resource: aws_ses_active_receipt_rule_set

Provides a resource to designate the active SES receipt rule set

» Example Usage

```
resource "aws_ses_active_receipt_rule_set" "main" {
  rule_set_name = "primary-rules"
}
```

» Argument Reference

The following arguments are supported:

- `rule_set_name` - (Required) The name of the rule set

» Resource: aws_ses_domain_identity

Provides an SES domain identity resource

» Argument Reference

The following arguments are supported:

- `domain` - (Required) The domain name to assign to SES

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the domain identity.
- `verification_token` - A code which when added to the domain as a TXT record will signal to SES that the owner of the domain has authorised SES to act on their behalf. The domain identity will be in state ”verification pending” until this is done. See below for an example of how this might be achieved when the domain is hosted in Route 53 and managed by Terraform. Find out more about verifying domains in Amazon SES in the AWS SES docs.

» Example Usage

```
resource "aws_ses_domain_identity" "example" {
  domain = "example.com"
}

resource "aws_route53_record" "example_amazonses_verification_record" {
  zone_id = "ABCDEFGHIJ123"
  name    = "_amazonses.example.com"
  type    = "TXT"
  ttl     = "600"
  records = ["${aws_ses_domain_identity.example.verification_token}"]
}
```

» Import

SES domain identities can be imported using the domain name.

```
$ terraform import aws_ses_domain_identity.example example.com
```

» Resource: aws_ses_domain_identity_verification

Represents a successful verification of an SES domain identity.

Most commonly, this resource is used together with `aws_route53_record` and `aws_ses_domain_identity` to request an SES domain identity, deploy the required DNS verification records, and wait for verification to complete.

WARNING: This resource implements a part of the verification workflow. It does not represent a real-world entity in AWS, therefore changing or deleting this resource on its own has no immediate effect.

» Example Usage

```
resource "aws_ses_domain_identity" "example" {
  domain = "example.com"
}

resource "aws_route53_record" "example_amazonses_verification_record" {
  zone_id = "${aws_route53_zone.example.zone_id}"
  name    = "_amazonses.${aws_ses_domain_identity.example.id}"
  type    = "TXT"
  ttl     = "600"
  records = ["${aws_ses_domain_identity.example.verification_token}"]
}

resource "aws_ses_domain_identity_verification" "example_verification" {
  domain = "${aws_ses_domain_identity.example.id}"

  depends_on = ["aws_route53_record.example_amazonses_verification_record"]
}
```

» Argument Reference

The following arguments are supported:

- `domain` - (Required) The domain name of the SES domain identity to verify.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The domain name of the domain identity.
- `arn` - The ARN of the domain identity.

» Timeouts

`acm_ses_domain_identity_verification` provides the following Timeouts configuration options:

- `create` - (Default `45m`) How long to wait for a domain identity to be verified.

» Resource: aws_ses_domain_dkim

Provides an SES domain DKIM generation resource.

Domain ownership needs to be confirmed first using ses_domain_identity Resource

» Argument Reference

The following arguments are supported:

- `domain` - (Required) Verified domain name to generate DKIM tokens for.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `dkim_tokens` - DKIM tokens generated by SES. These tokens should be used to create CNAME records used to verify SES Easy DKIM. See below for an example of how this might be achieved when the domain is hosted in Route 53 and managed by Terraform. Find out more about verifying domains in Amazon SES in the AWS SES docs.

» Example Usage

```
resource "aws_ses_domain_identity" "example" {
  domain = "example.com"
}

resource "aws_ses_domain_dkim" "example" {
  domain = "${aws_ses_domain_identity.example.domain}"
}

resource "aws_route53_record" "example_amazonses_dkim_record" {
  count   = 3
  zone_id = "ABCDEFGHIJ123"
  name    = "${element(aws_ses_domain_dkim.example.dkim_tokens, count.index)}._domainkey.exa
  type    = "CNAME"
  ttl     = "600"
  records = ["${element(aws_ses_domain_dkim.example.dkim_tokens, count.index)}.dkim.amazonse
}
```

» Import

DKIM tokens can be imported using the `domain` attribute, e.g.

```
$ terraform import aws_ses_domain_dkim.example example.com
```

» Resource: aws_ses_domain_mail_from

Provides an SES domain MAIL FROM resource.

NOTE: For the MAIL FROM domain to be fully usable, this resource should be paired with the aws_ses_domain_identity resource. To validate the MAIL FROM domain, a DNS MX record is required. To pass SPF checks, a DNS TXT record may also be required. See the Amazon SES MAIL FROM documentation for more information.

» Example Usage

```
resource "aws_ses_domain_mail_from" "example" {
  domain           = "${aws_ses_domain_identity.example.domain}"
  mail_from_domain = "bounce.${aws_ses_domain_identity.example.domain}"
}

# Example SES Domain Identity
resource "aws_ses_domain_identity" "example" {
  domain = "example.com"
}

# Example Route53 MX record
resource "aws_route53_record" "example_ses_domain_mail_from_mx" {
  zone_id = "${aws_route53_zone.example.id}"
  name    = "${aws_ses_domain_mail_from.example.mail_from_domain}"
  type    = "MX"
  ttl     = "600"
  records = ["10 feedback-smtp.us-east-1.amazonses.com"]           # Change to the region i
}
```

```
# Example Route53 TXT record for SPF
resource "aws_route53_record" "example_ses_domain_mail_from_txt" {
  zone_id = "${aws_route53_zone.example.id}"
  name    = "${aws_ses_domain_mail_from.example.mail_from_domain}"
  type    = "TXT"
  ttl     = "600"
  records = ["v=spf1 include:amazonses.com -all"]
}
```

» Argument Reference

The following arguments are required:

- `domain` - (Required) Verified domain name to generate DKIM tokens for.
- `mail_from_domain` - (Required) Subdomain (of above domain) which is to be used as MAIL FROM address (Required for DMARC validation)

The following arguments are optional:

- `behavior_on_mx_failure` - (Optional) The action that you want Amazon SES to take if it cannot successfully read the required MX record when you send an email. Defaults to `UseDefaultValue`. See the SES API documentation for more information.

» Attributes Reference

In addition to the arguments, which are exported, the following attributes are exported:

- `id` - The domain name.

» Import

MAIL FROM domain can be imported using the `domain` attribute, e.g.

```
$ terraform import aws_ses_domain_mail_from.example example.com
```

» Resource: aws_ses_email_identity

Provides an SES email identity resource

» Argument Reference

The following arguments are supported:

- `email` - (Required) The email address to assign to SES

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the email identity.

» Example Usage

```
resource "aws_ses_email_identity" "example" {
  email = "email@example.com"
}
```

» Import

SES email identities can be imported using the email address.

```
$ terraform import aws_ses_email_identity.example email@example.com
```

» Resource: aws_ses_receipt_filter

Provides an SES receipt filter resource

» Example Usage

```
resource "aws_ses_receipt_filter" "filter" {
  name   = "block-spammer"
  cidr   = "10.10.10.10"
  policy = "Block"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the filter

- `cidr` - (Required) The IP address or address range to filter, in CIDR notation
- `policy` - (Required) Block or Allow

» Resource: aws_ses_receipt_rule

Provides an SES receipt rule resource

» Example Usage

```
# Add a header to the email and store it in S3
resource "aws_ses_receipt_rule" "store" {
  name          = "store"
  rule_set_name = "default-rule-set"
  recipients    = ["karen@example.com"]
  enabled       = true
  scan_enabled  = true

  add_header_action {
    header_name  = "Custom-Header"
    header_value = "Added by SES"
    position     = 1
  }

  s3_action {
    bucket_name = "emails"
    position    = 2
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the rule
- `rule_set_name` - (Required) The name of the rule set
- `after` - (Optional) The name of the rule to place this rule after
- `enabled` - (Optional) If true, the rule will be enabled
- `recipients` - (Optional) A list of email addresses
- `scan_enabled` - (Optional) If true, incoming emails will be scanned for spam and viruses
- `tls_policy` - (Optional) Require or Optional

- `add_header_action` - (Optional) A list of Add Header Action blocks. Documented below.
- `bounce_action` - (Optional) A list of Bounce Action blocks. Documented below.
- `lambda_action` - (Optional) A list of Lambda Action blocks. Documented below.
- `s3_action` - (Optional) A list of S3 Action blocks. Documented below.
- `sns_action` - (Optional) A list of SNS Action blocks. Documented below.
- `stop_action` - (Optional) A list of Stop Action blocks. Documented below.
- `workmail_action` - (Optional) A list of WorkMail Action blocks. Documented below.

Add header actions support the following:

- `header_name` - (Required) The name of the header to add
- `header_value` - (Required) The value of the header to add
- `position` - (Required) The position of the action in the receipt rule

Bounce actions support the following:

- `message` - (Required) The message to send
- `sender` - (Required) The email address of the sender
- `smtp_reply_code` - (Required) The RFC 5321 SMTP reply code
- `status_code` - (Optional) The RFC 3463 SMTP enhanced status code
- `topic_arn` - (Optional) The ARN of an SNS topic to notify
- `position` - (Required) The position of the action in the receipt rule

Lambda actions support the following:

- `function_arn` - (Required) The ARN of the Lambda function to invoke
- `invocation_type` - (Optional) Event or RequestResponse
- `topic_arn` - (Optional) The ARN of an SNS topic to notify
- `position` - (Required) The position of the action in the receipt rule

S3 actions support the following:

- `bucket_name` - (Required) The name of the S3 bucket
- `kms_key_arn` - (Optional) The ARN of the KMS key
- `object_key_prefix` - (Optional) The key prefix of the S3 bucket
- `topic_arn` - (Optional) The ARN of an SNS topic to notify
- `position` - (Required) The position of the action in the receipt rule

SNS actions support the following:

- `topic_arn` - (Required) The ARN of an SNS topic to notify
- `position` - (Required) The position of the action in the receipt rule

Stop actions support the following:

- `scope` - (Required) The scope to apply
- `topic_arn` - (Optional) The ARN of an SNS topic to notify

- `position` - (Required) The position of the action in the receipt rule

WorkMail actions support the following:

- `organization_arn` - (Required) The ARN of the WorkMail organization
- `topic_arn` - (Optional) The ARN of an SNS topic to notify
- `position` - (Required) The position of the action in the receipt rule

» Import

SES receipt rules can be imported using the ruleset name and rule name separated by `:`.

```
$ terraform import aws_ses_receipt_rule.my_rule my_rule_set:my_rule
```

» Resource: aws_ses_receipt_rule_set

Provides an SES receipt rule set resource

» Example Usage

```
resource "aws_ses_receipt_rule_set" "main" {
  rule_set_name = "primary-rules"
}
```

» Argument Reference

The following arguments are supported:

- `rule_set_name` - (Required) The name of the rule set

» Import

SES receipt rule sets can be imported using the rule set name.

```
$ terraform import aws_ses_receipt_rule_set.my_rule_set my_rule_set_name
```

» Resource: aws_ses_configuration_set

Provides an SES configuration set resource

» Example Usage

```
resource "aws_ses_configuration_set" "test" {
  name = "some-configuration-set-test"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the configuration set

» Import

SES Configuration Sets can be imported using their `name`, e.g.

```
$ terraform import aws_ses_configuration_set.test some-configuration-set-test
```

» Resource: aws_ses_event_destination

Provides an SES event destination

» Example Usage

» CloudWatch Destination

```
resource "aws_ses_event_destination" "cloudwatch" {
  name                   = "event-destination-cloudwatch"
  configuration_set_name = "${aws_ses_configuration_set.example.name}"
  enabled                = true
  matching_types         = ["bounce", "send"]

  cloudwatch_destination {
    default_value  = "default"
    dimension_name = "dimension"
    value_source   = "emailHeader"
  }
}
```

» Kinesis Destination

```
resource "aws_ses_event_destination" "kinesis" {
  name                   = "event-destination-kinesis"
  configuration_set_name = "${aws_ses_configuration_set.example.name}"
  enabled                = true
  matching_types         = ["bounce", "send"]

  kinesis_destination {
    stream_arn = "${aws_kinesis_firehose_delivery_stream.example.arn}"
    role_arn   = "${aws_iam_role.example.arn}"
  }
}
```

» SNS Destination

```
resource "aws_ses_event_destination" "sns" {
  name                   = "event-destination-sns"
  configuration_set_name = "${aws_ses_configuration_set.example.name}"
  enabled                = true
  matching_types         = ["bounce", "send"]

  sns_destination {
    topic_arn = "${aws_sns_topic.example.arn}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the event destination
- `configuration_set_name` - (Required) The name of the configuration set
- `enabled` - (Optional) If true, the event destination will be enabled
- `matching_types` - (Required) A list of matching types. May be any of `"send"`, `"reject"`, `"bounce"`, `"complaint"`, `"delivery"`, `"open"`, `"click"`, or `"renderingFailure"`.
- `cloudwatch_destination` - (Optional) CloudWatch destination for the events
- `kinesis_destination` - (Optional) Send the events to a kinesis firehose destination
- `sns_destination` - (Optional) Send the events to an SNS Topic destination

NOTE: You can specify "`cloudwatch_destination`" or "`kinesis_destination`" but not both

» cloudwatch_destination Argument Reference

- `default_value` - (Required) The default value for the event
- `dimension_name` - (Required) The name for the dimension
- `value_source` - (Required) The source for the value. It can be either "`messageTag`" or "`emailHeader`"

» kinesis_destination Argument Reference

- `stream_arn` - (Required) The ARN of the Kinesis Stream
- `role_arn` - (Required) The ARN of the role that has permissions to access the Kinesis Stream

» sns_destination Argument Reference

- `topic_arn` - (Required) The ARN of the SNS topic

» Resource: ses_identity_notification_topic

Resource for managing SES Identity Notification Topics

» Example Usage

```
resource "aws_ses_identity_notification_topic" "test" {
  topic_arn                = "${aws_sns_topic.example.arn}"
  notification_type        = "Bounce"
  identity                 = "${aws_ses_domain_identity.example.domain}"
  include_original_headers = true
}
```

» Argument Reference

The following arguments are supported:

- `topic_arn` - (Optional) The Amazon Resource Name (ARN) of the Amazon SNS topic. Can be set to ”” (an empty string) to disable publishing.

- `notification_type` - (Required) The type of notifications that will be published to the specified Amazon SNS topic. Valid Values: *Bounce*, *Complaint* or *Delivery*.
- `identity` - (Required) The identity for which the Amazon SNS topic will be set. You can specify an identity by using its name or by using its Amazon Resource Name (ARN).
- `include_original_headers` - (Optional) Whether SES should include original email headers in SNS notifications of this type. *false* by default.

» Import

Identity Notification Topics can be imported using ID of the record. The ID is made up as IDENTITY|TYPE where IDENTITY is the SES Identity and TYPE is the Notification Type.

e.g.

`example.com|Bounce`

In this example, `example.com` is the SES Identity and `Bounce` is the Notification Type.

To import the ID above, it would look as follows:

```
$ terraform import aws_ses_identity_notification_topic.test 'example.com|Bounce'
```

» Resource: aws_ses_identity_policy

Manages a SES Identity Policy. More information about SES Sending Authorization Policies can be found in the SES Developer Guide.

» Example Usage

```
resource "aws_ses_domain_identity" "example" {
  domain = "example.com"
}

data "aws_iam_policy_document" "example" {
  statement {
    actions   = ["SES:SendEmail", "SES:SendRawEmail"]
    resources = ["${aws_ses_domain_identity.example.arn}"]

    principals {
      identifiers = ["*"]
      type        = "AWS"
    }
  }
}

resource "aws_ses_identity_policy" "example" {
  identity = "${aws_ses_domain_identity.example.arn}"
  name     = "example"
  policy   = "${data.aws_iam_policy_document.example.json}"
}
```

» Argument Reference

The following arguments are supported:

- `identity` - (Required) Name or Amazon Resource Name (ARN) of the SES Identity.
- `name` - (Required) Name of the policy.
- `policy` - (Required) JSON string of the policy. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

» Import

SES Identity Policies can be imported using the identity and policy name, separated by a pipe character (`|`), e.g.

```
$ terraform import aws_ses_identity_policy.example 'example.com|example'
```

» Resource: aws_ses_template

Provides a resource to create a SES template.

» Example Usage

```
resource "aws_ses_template" "MyTemplate" {
  name    = "MyTemplate"
  subject = "Greetings, {{name}}!"
  html    = "<h1>Hello {{name}},</h1><p>Your favorite animal is {{favoriteanimal}}.</p>"
  text    = "Hello {{name}},\r\nYour favorite animal is {{favoriteanimal}}."
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the template. Cannot exceed 64 characters. You will refer to this name when you send email.
- `html` - (Optional) The HTML body of the email. Must be less than 500KB in size, including both the text and HTML parts.
- `subject` - (Optional) The subject line of the email.
- `text` - (Optional) The email body that will be visible to recipients whose email clients do not display HTML. Must be less than 500KB in size, including both the text and HTML parts.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the SES template

» Import

SES templates can be imported using the template name, e.g.

```
$ terraform import aws_ses_template.MyTemplate MyTemplate
```

» Resource: aws_servicecatalog_portfolio

Provides a resource to create a Service Catalog Portfolio.

» Example Usage

```
resource "aws_servicecatalog_portfolio" "portfolio" {
  name          = "My App Portfolio"
  description   = "List of my organizations apps"
  provider_name = "Brett"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the portfolio.

- `description` - (Required) Description of the portfolio
- `provider_name` - (Required) Name of the person or organization who owns the portfolio.
- `tags` - (Optional) Tags to apply to the connection.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the Service Catalog Portfolio.

» Import

Service Catalog Portfolios can be imported using the `service catalog portfolio id`, e.g.

```
$ terraform import aws_servicecatalog_portfolio.testfolio port-12344321
```

» Resource: aws_service_discovery_http_namespace

» Example Usage

```
resource "aws_service_discovery_http_namespace" "example" {
  name        = "development"
  description = "example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the http namespace.
- `description` - (Optional) The description that you specify for the namespace when you create it.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of a namespace.
- `arn` - The ARN that Amazon Route 53 assigns to the namespace when you create it.

» Import

Service Discovery HTTP Namespace can be imported using the namespace ID, e.g.

```
$ terraform import aws_service_discovery_http_namespace.example ns-1234567890
```

» Resource: aws_service_discovery_private_dns_namespace

Provides a Service Discovery Private DNS Namespace resource.

» Example Usage

```
resource "aws_vpc" "example" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_service_discovery_private_dns_namespace" "example" {
  name        = "hoge.example.local"
  description = "example"
  vpc         = "${aws_vpc.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the namespace.
- `vpc` - (Required) The ID of VPC that you want to associate the namespace with.
- `description` - (Optional) The description that you specify for the namespace when you create it.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of a namespace.
- `arn` - The ARN that Amazon Route 53 assigns to the namespace when you create it.
- `hosted_zone` - The ID for the hosted zone that Amazon Route 53 creates when you create a namespace.

» Resource: aws_service_discovery_public_dns_namespace

Provides a Service Discovery Public DNS Namespace resource.

» Example Usage

```
resource "aws_service_discovery_public_dns_namespace" "example" {
  name        = "hoge.example.com"
  description = "example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the namespace.
- `description` - (Optional) The description that you specify for the namespace when you create it.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of a namespace.
- `arn` - The ARN that Amazon Route 53 assigns to the namespace when you create it.
- `hosted_zone` - The ID for the hosted zone that Amazon Route 53 creates when you create a namespace.

» Import

Service Discovery Public DNS Namespace can be imported using the namespace ID, e.g.

```
$ terraform import aws_service_discovery_public_dns_namespace.example 0123456789
```

» Resource: aws_service_discovery_service

Provides a Service Discovery Service resource.

» Example Usage

```
resource "aws_vpc" "example" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true
}

resource "aws_service_discovery_private_dns_namespace" "example" {
  name        = "example.terraform.local"
  description = "example"
  vpc         = "${aws_vpc.example.id}"
}

resource "aws_service_discovery_service" "example" {
  name = "example"

  dns_config {
    namespace_id = "${aws_service_discovery_private_dns_namespace.example.id}"

    dns_records {
      ttl  = 10
      type = "A"
    }

    routing_policy = "MULTIVALUE"
  }

  health_check_custom_config {
    failure_threshold = 1
  }
}
```

```
resource "aws_service_discovery_public_dns_namespace" "example" {
  name        = "example.terraform.com"
  description = "example"
}

resource "aws_service_discovery_service" "example" {
  name = "example"

  dns_config {
    namespace_id = "${aws_service_discovery_public_dns_namespace.example.id}"

    dns_records {
      ttl  = 10
      type = "A"
    }
  }

  health_check_config {
    failure_threshold = 10
    resource_path     = "path"
    type              = "HTTP"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required, ForceNew) The name of the service.
- `description` - (Optional) The description of the service.
- `dns_config` - (Optional) A complex type that contains information about the resource record sets that you want Amazon Route 53 to create when you register an instance.
- `health_check_config` - (Optional) A complex type that contains settings for an optional health check. Only for Public DNS namespaces.
- `health_check_custom_config` - (Optional, ForceNew) A complex type that contains settings for ECS managed health checks.
- `namespace_id` - (Optional) The ID of the namespace that you want to use to create the service.

» dns_config

The following arguments are supported:

- `namespace_id` - (Required, ForceNew) The ID of the namespace to use for DNS configuration.
- `dns_records` - (Required) An array that contains one DnsRecord object for each resource record set.
- `routing_policy` - (Optional) The routing policy that you want to apply to all records that Route 53 creates when you register an instance and specify the service. Valid Values: MULTIVALUE, WEIGHTED

» dns_records

The following arguments are supported:

- `ttl` - (Required) The amount of time, in seconds, that you want DNS resolvers to cache the settings for this resource record set.

- `type` - (Required, ForceNew) The type of the resource, which indicates the value that Amazon Route 53 returns in response to DNS queries. Valid Values: A, AAAA, SRV, CNAME

» health_check_config

The following arguments are supported:

- `failure_threshold` - (Optional) The number of consecutive health checks. Maximum value of 10.
- `resource_path` - (Optional) The path that you want Route 53 to request when performing health checks. Route 53 automatically adds the DNS name for the service. If you don't specify a value, the default value is /.
- `type` - (Optional, ForceNew) The type of health check that you want to create, which indicates how Route 53 determines whether an endpoint is healthy. Valid Values: HTTP, HTTPS, TCP

» health_check_custom_config

The following arguments are supported:

- `failure_threshold` - (Optional, ForceNew) The number of 30-second intervals that you want service discovery to wait before it changes the health status of a service instance. Maximum value of 10.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the service.
- `arn` - The ARN of the service.

» Import

Service Discovery Service can be imported using the service ID, e.g.

```
$ terraform import aws_service_discovery_service.example 0123456789
```

» Data Source: aws_servicequotas_service

Retrieve information about a Service Quotas Service.

» Example Usage

```
data "aws_servicequotas_service" "example" {
  service_name = "Amazon Virtual Private Cloud (Amazon VPC)"
}
```

» Argument Reference

- `service_name` - (Required) Service name to lookup within Service Quotas. Available values can be found with the AWS CLI service-quotas list-services command.

» Attributes Reference

- `id` - Code of the service.
- `service_code` - Code of the service.

» Data Source: aws_servicequotas_service_quota

Retrieve information about a Service Quota.

» Example Usage

```
data "aws_servicequotas_service_quota" "by_quota_code" {
  quota_code   = "L-F678F1CE"
  service_code = "vpc"
}

data "aws_servicequotas_service_quota" "by_quota_name" {
  quota_name   = "VPCs per Region"
  service_code = "vpc"
}
```

» Argument Reference

NOTE: Either `quota_code` or `quota_name` must be configured.

- `service_code` - (Required) Service code for the quota. Available values can be found with the `aws_servicequotas_service` data source or AWS CLI service-quotas list-services command.

- quota_code - (Optional) Quota code within the service. When configured, the data source directly looks up the service quota. Available values can be found with the AWS CLI service-quotas list-service-quotas command.
- quota_name - (Optional) Quota name within the service. When configured, the data source searches through all service quotas to find the matching quota name. Available values can be found with the AWS CLI service-quotas list-service-quotas command.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- adjustable - Whether the service quota is adjustable.
- arn - Amazon Resource Name (ARN) of the service quota.
- default_value - Default value of the service quota.
- global_quota - Whether the service quota is global for the AWS account.
- id - Amazon Resource Name (ARN) of the service quota.
- service_name - Name of the service.
- value - Current value of the service quota.

» Resource: aws_servicequotas_service_quota

Manages an individual Service Quota.

» Example Usage

```
resource "aws_servicequotas_service_quota" "example" {
  quota_code   = "L-F678F1CE"
  service_code = "vpc"
  value        = 75
}
```

» Argument Reference

The following arguments are supported:

- quota_code - (Required) Code of the service quota to track. For example: L-F678F1CE. Available values can be found with the AWS CLI service-quotas list-service-quotas command.
- service_code - (Required) Code of the service to track. For example: vpc. Available values can be found with the AWS CLI service-quotas list-services command.

- `value` - (Required) Float specifying the desired value for the service quota. If the desired value is higher than the current value, a quota increase request is submitted. When a known request is submitted and pending, the value reflects the desired value of the pending request.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `adjustable` - Whether the service quota can be increased.
- `arn` - Amazon Resource Name (ARN) of the service quota.
- `default_value` - Default value of the service quota.
- `id` - Service code and quota code, separated by a front slash (`/`)
- `quota_name` - Name of the quota.
- `service_name` - Name of the service.

» Import

NOTE This resource does not require explicit import and will assume management of an existing service quota on Terraform resource creation.

`aws_servicequotas_service_quota` can be imported by using the service code and quota code, separated by a front slash (`/`), e.g.

```
$ terraform import aws_servicequotas_service_quota.example vpc/L-F678F1CE
```

» Resource: aws_shield_protection

Enables AWS Shield Advanced for a specific AWS resource. The resource can be an Amazon CloudFront distribution, Elastic Load Balancing load balancer, AWS Global Accelerator accelerator, Elastic IP Address, or an Amazon Route 53 hosted zone.

» Example Usage

» Create protection

```
data "aws_availability_zones" "available" {}
data "aws_region" "current" {}
data "aws_caller_identity" "current" {}

resource "aws_eip" "foo" {
  vpc = true
```

```
}

resource "aws_shield_protection" "foo" {
  name         = "${var.name}"
  resource_arn = "arn:aws:ec2:${data.aws_region.current.name}:${data.aws_caller_identity.cur
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A friendly name for the Protection you are creating.
- `resource_arn` - (Required) The ARN (Amazon Resource Name) of the resource to be protected.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The unique identifier (ID) for the Protection object that is created.

» Import

Shield protection resources can be imported by specifying their ID e.g.

```
$ terraform import aws_shield_protection.foo ff9592dc-22f3-4e88-afa1-7b29fde9669a
```

» Resource: aws_simpledb_domain

Provides a SimpleDB domain resource

» Example Usage

```
resource "aws_simpledb_domain" "users" {
  name = "users"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the SimpleDB domain

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the SimpleDB domain

» Import

SimpleDB Domains can be imported using the `name`, e.g.

```
$ terraform import aws_simpledb_domain.users users
```

» Data Source: aws_sns_topic

Use this data source to get the ARN of a topic in AWS Simple Notification Service (SNS). By using this data source, you can reference SNS topics without having to hard code the ARNs as input.

» Example Usage

```
data "aws_sns_topic" "example" {
  name = "an_example_topic"
}
```

» Argument Reference

- `name` - (Required) The friendly name of the topic to match.

» Attributes Reference

- `arn` - Set to the ARN of the found topic, suitable for referencing in other resources that support SNS topics.

» Resource: aws_sns_platform_application

Provides an SNS platform application resource

» Example Usage

» Apple Push Notification Service (APNS)

```
resource "aws_sns_platform_application" "apns_application" {
  name                = "apns_application"
  platform            = "APNS"
  platform_credential = "<APNS PRIVATE KEY>"
  platform_principal  = "<APNS CERTIFICATE>"
}
```

» Google Cloud Messaging (GCM)

```
resource "aws_sns_platform_application" "gcm_application" {
  name                = "gcm_application"
  platform            = "GCM"
  platform_credential = "<GCM API KEY>"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The friendly name for the SNS platform application
- `platform` - (Required) The platform that the app is registered with. See Platform for supported platforms.
- `platform_credential` - (Required) Application Platform credential. See Credential for type of credential required for platform. The value of this attribute when stored into the Terraform state is only a hash of the real value, so therefore it is not practical to use this as an attribute for other resources.
- `event_delivery_failure_topic_arn` - (Optional) SNS Topic triggered when a delivery to any of the platform endpoints associated with your platform application encounters a permanent failure.
- `event_endpoint_created_topic_arn` - (Optional) SNS Topic triggered when a new platform endpoint is added to your platform application.
- `event_endpoint_deleted_topic_arn` - (Optional) SNS Topic triggered when an existing platform endpoint is deleted from your platform application.
- `event_endpoint_updated_topic_arn` - (Optional) SNS Topic triggered when an existing platform endpoint is changed from your platform application.
- `failure_feedback_role_arn` - (Optional) The IAM role permitted to receive failure feedback for this application.

- `platform_principal` - (Optional) Application Platform principal. See Principal for type of principal required for platform. The value of this attribute when stored into the Terraform state is only a hash of the real value, so therefore it is not practical to use this as an attribute for other resources.
- `success_feedback_role_arn` - (Optional) The IAM role permitted to receive success feedback for this application.
- `success_feedback_sample_rate` - (Optional) The percentage of success to sample (0-100)

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ARN of the SNS platform application
- `arn` - The ARN of the SNS platform application

» Import

SNS platform applications can be imported using the ARN, e.g.

```
$ terraform import aws_sns_platform_application.gcm_application arn:aws:sns:us-west-2:012345
```

» Resource: aws_sns_sms_preferences

Provides a way to set SNS SMS preferences.

» Example Usage

```
resource "aws_sns_sms_preferences" "update_sms_prefs" {}
```

» Argument Reference

The following arguments are supported:

- `monthly_spend_limit` - (Optional) The maximum amount in USD that you are willing to spend each month to send SMS messages.
- `delivery_status_iam_role_arn` - (Optional) The ARN of the IAM role that allows Amazon SNS to write logs about SMS deliveries in CloudWatch Logs.

- delivery_status_success_sampling_rate - (Optional) The percentage of successful SMS deliveries for which Amazon SNS will write logs in CloudWatch Logs. The value must be between 0 and 100.
- default_sender_id - (Optional) A string, such as your business brand, that is displayed as the sender on the receiving device.
- default_sms_type - (Optional) The type of SMS message that you will send by default. Possible values are: Promotional, Transactional
- usage_report_s3_bucket - (Optional) The name of the Amazon S3 bucket to receive daily SMS usage reports from Amazon SNS.

» Resource: aws_sns_topic

Provides an SNS topic resource

» Example Usage

```
resource "aws_sns_topic" "user_updates" {
  name = "user-updates-topic"
}
```

» Example with Delivery Policy

```
resource "aws_sns_topic" "user_updates" {
  name = "user-updates-topic"
  delivery_policy = <<EOF
{
  "http": {
    "defaultHealthyRetryPolicy": {
      "minDelayTarget": 20,
      "maxDelayTarget": 20,
      "numRetries": 3,
      "numMaxDelayRetries": 0,
      "numNoDelayRetries": 0,
      "numMinDelayRetries": 0,
      "backoffFunction": "linear"
    },
    "disableSubscriptionOverrides": false,
    "defaultThrottlePolicy": {
      "maxReceivesPerSecond": 1
    }
  }
}
EOF
}
```

» Example with Server-side encryption (SSE)

```
resource "aws_sns_topic" "user_updates" {
  name              = "user-updates-topic"
  kms_master_key_id = "alias/aws/sns"
}
```

» Message Delivery Status Arguments

The `<endpoint>_success_feedback_role_arn` and `<endpoint>_failure_feedback_role_arn` arguments are used to give Amazon SNS write access to use CloudWatch Logs on your behalf. The `<endpoint>_success_feedback_sample_rate` argument is for specifying the sample rate percentage (0-100) of successfully delivered messages. After you configure the `<endpoint>_failure_feedback_role_arn` argument, then all failed message deliveries generate CloudWatch Logs.

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The friendly name for the SNS topic. By default generated by Terraform.
- `name_prefix` - (Optional) The friendly name for the SNS topic. Conflicts with `name`.
- `display_name` - (Optional) The display name for the SNS topic
- `policy` - (Optional) The fully-formed AWS policy as JSON. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.
- `delivery_policy` - (Optional) The SNS delivery policy. More on AWS documentation
- `application_success_feedback_role_arn` - (Optional) The IAM role permitted to receive success feedback for this topic
- `application_success_feedback_sample_rate` - (Optional) Percentage of success to sample
- `application_failure_feedback_role_arn` - (Optional) IAM role for failure feedback
- `http_success_feedback_role_arn` - (Optional) The IAM role permitted to receive success feedback for this topic
- `http_success_feedback_sample_rate` - (Optional) Percentage of success to sample

- `http_failure_feedback_role_arn` - (Optional) IAM role for failure feedback
- `kms_master_key_id` - (Optional) The ID of an AWS-managed customer master key (CMK) for Amazon SNS or a custom CMK. For more information, see Key Terms
- `lambda_success_feedback_role_arn` - (Optional) The IAM role permitted to receive success feedback for this topic
- `lambda_success_feedback_sample_rate` - (Optional) Percentage of success to sample
- `lambda_failure_feedback_role_arn` - (Optional) IAM role for failure feedback
- `sqs_success_feedback_role_arn` - (Optional) The IAM role permitted to receive success feedback for this topic
- `sqs_success_feedback_sample_rate` - (Optional) Percentage of success to sample
- `sqs_failure_feedback_role_arn` - (Optional) IAM role for failure feedback
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ARN of the SNS topic
- `arn` - The ARN of the SNS topic, as a more obvious property (clone of id)

» Import

SNS Topics can be imported using the `topic arn`, e.g.

```
$ terraform import aws_sns_topic.user_updates arn:aws:sns:us-west-2:0123456789012:my-topic
```

» Resource: aws_sns_topic_policy

Provides an SNS topic policy resource

NOTE: If a Principal is specified as just an AWS account ID rather than an ARN, AWS silently converts it to the ARN for the root user, causing future terraform plans to differ. To avoid this problem, just specify the full ARN, e.g. `arn:aws:iam::123456789012:root`

» Example Usage

```
resource "aws_sns_topic" "test" {
  name = "my-topic-with-policy"
}

resource "aws_sns_topic_policy" "default" {
  arn = "${aws_sns_topic.test.arn}"

  policy = "${data.aws_iam_policy_document.sns-topic-policy.json}"
}

data "aws_iam_policy_document" "sns-topic-policy" {
  policy_id = "__default_policy_ID"

  statement {
    actions = [
      "SNS:Subscribe",
      "SNS:SetTopicAttributes",
      "SNS:RemovePermission",
      "SNS:Receive",
      "SNS:Publish",
      "SNS:ListSubscriptionsByTopic",
      "SNS:GetTopicAttributes",
      "SNS:DeleteTopic",
      "SNS:AddPermission",
    ]

    condition {
      test     = "StringEquals"
      variable = "AWS:SourceOwner"

      values = [
        "${var.account-id}",
      ]
    }

    effect = "Allow"

    principals {
      type        = "AWS"
      identifiers = ["*"]
    }

    resources = [
      "${aws_sns_topic.test.arn}",
    ]

    sid = "__default_statement_ID"
  }
}
```

» Argument Reference

The following arguments are supported:

- `arn` - (Required) The ARN of the SNS topic
- `policy` - (Required) The fully-formed AWS policy as JSON. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

» Import

SNS Topic Policy can be imported using the topic ARN, e.g.

```
$ terraform import aws_sns_topic_policy.user_updates arn:aws:sns:us-west-2:0123456789012:my-
```

» Resource: aws_sns_topic_subscription

Provides a resource for subscribing to SNS topics. Requires that an SNS topic exist for the subscription to attach to. This resource allows you to automatically place messages sent to SNS topics in SQS queues, send them as HTTP(S) POST requests to a given endpoint, send SMS messages, or notify devices / applications. The most likely use case for Terraform users will probably be SQS queues.

NOTE: If the SNS topic and SQS queue are in different AWS regions, it is important for the "aws_sns_topic_subscription" to use an AWS provider that is in the same region of the SNS topic. If the "aws_sns_topic_subscription" is using a provider with a different region than the SNS topic, terraform will fail to create the subscription.

NOTE: Setup of cross-account subscriptions from SNS topics to SQS queues requires Terraform to have access to BOTH accounts.

NOTE: If SNS topic and SQS queue are in different AWS accounts but the same region it is important for the "aws_sns_topic_subscription" to use the AWS provider of the account with the SQS queue. If "aws_sns_topic_subscription" is using a Provider with a different account than the SQS queue, terraform creates the subscriptions but does not keep state and tries to re-create the subscription at every apply.

NOTE: If SNS topic and SQS queue are in different AWS accounts and different AWS regions it is important to recognize that the subscription needs to be initiated from the account with the SQS queue but in the region of the SNS topic.

» Example Usage

You can directly supply a topic and ARN by hand in the `topic_arn` property along with the queue ARN:

```
resource "aws_sns_topic_subscription" "user_updates_sqs_target" {
  topic_arn = "arn:aws:sns:us-west-2:432981146916:user-updates-topic"
  protocol  = "sqs"
  endpoint  = "arn:aws:sqs:us-west-2:432981146916:terraform-queue-too"
}
```

Alternatively you can use the ARN properties of a managed SNS topic and SQS queue:

```
resource "aws_sns_topic" "user_updates" {
  name = "user-updates-topic"
}

resource "aws_sqs_queue" "user_updates_queue" {
  name = "user-updates-queue"
}

resource "aws_sns_topic_subscription" "user_updates_sqs_target" {
  topic_arn = "${aws_sns_topic.user_updates.arn}"
  protocol  = "sqs"
  endpoint  = "${aws_sqs_queue.user_updates_queue.arn}"
}
```

You can subscribe SNS topics to SQS queues in different Amazon accounts and regions:

```
/*
#
# Variables
#
*/
variable "sns" {
  default = {
    account-id   = "111111111111"
    role-name    = "service/service-hashicorp-terraform"
    name         = "example-sns-topic"
    display_name = "example"
    region       = "us-west-1"
  }
}

variable "sqs" {
  default = {
    account-id = "222222222222"
    role-name  = "service/service-hashicorp-terraform"
    name       = "example-sqs-queue"
    region     = "us-east-1"
  }
}

data "aws_iam_policy_document" "sns-topic-policy" {
  policy_id = "__default_policy_ID"

  statement {
    actions = [
      "SNS:Subscribe",
      "SNS:SetTopicAttributes",
      "SNS:RemovePermission",
      "SNS:Receive",
      "SNS:Publish",
      "SNS:ListSubscriptionsByTopic",
      "SNS:GetTopicAttributes",
      "SNS:DeleteTopic",
      "SNS:AddPermission",
    ]

    condition {
      test     = "StringEquals"
      variable = "AWS:SourceOwner"

      values = [
        "${var.sns["account-id"]}",
      ]
    }

    effect = "Allow"

    principals {
      type        = "AWS"
      identifiers = ["*"]
    }
    resources = [
      "arn:aws:sns:${var.sns["region"]}:${var.sns["account-id"]}:${var.sns["name"]}",
    ]

    sid = "__default_statement_ID"
  }

  statement {
    actions = [
      "SNS:Subscribe",
      "SNS:Receive",
    ]

    condition {
      test     = "StringLike"
      variable = "SNS:Endpoint"

      values = [
        "arn:aws:sqs:${var.sqs["region"]}:${var.sqs["account-id"]}:${var.sqs["name"]}",
      ]
    }

    effect = "Allow"

    principals {
      type        = "AWS"
      identifiers = ["*"]
    }

    resources = [
      "arn:aws:sns:${var.sns["region"]}:${var.sns["account-id"]}:${var.sns["name"]}",
    ]

    sid = "__console_sub_0"
  }
}

data "aws_iam_policy_document" "sqs-queue-policy" {
  policy_id = "arn:aws:sqs:${var.sqs["region"]}:${var.sqs["account-id"]}:${var.sqs["name"]}/

  statement {
    sid    = "example-sns-topic"
    effect = "Allow"

    principals {
      type        = "AWS"
      identifiers = ["*"]
    }

    actions = [
      "SQS:SendMessage",
    ]

    resources = [
      "arn:aws:sqs:${var.sqs["region"]}:${var.sqs["account-id"]}:${var.sqs["name"]}",
    ]

    condition {
      test     = "ArnEquals"
      variable = "aws:SourceArn"

      values = [
        "arn:aws:sns:${var.sns["region"]}:${var.sns["account-id"]}:${var.sns["name"]}",
      ]
    }
  }
}

# provider to manage SNS topics
provider "aws" {
  alias  = "sns"
  region = "${var.sns["region"]}"

  assume_role {
    role_arn     = "arn:aws:iam::${var.sns["account-id"]}:role/${var.sns["role-name"]}"
    session_name = "sns-${var.sns["region"]}"
  }
}

# provider to manage SQS queues
provider "aws" {
  alias  = "sqs"
  region = "${var.sqs["region"]}"

  assume_role {
    role_arn     = "arn:aws:iam::${var.sqs["account-id"]}:role/${var.sqs["role-name"]}"
    session_name = "sqs-${var.sqs["region"]}"
  }
}

# provider to subscribe SQS to SNS (using the SQS account but the SNS region)
```

```
provider "aws" {
  alias  = "sns2sqs"
  region = "${var.sns["region"]}"

  assume_role {
    role_arn     = "arn:aws:iam::${var.sqs["account-id"]}:role/${var.sqs["role-name"]}"
    session_name = "sns2sqs-${var.sns["region"]}"
  }
}

resource "aws_sns_topic" "sns-topic" {
  provider     = "aws.sns"
  name         = "${var.sns["name"]}"
  display_name = "${var.sns["display_name"]}"
  policy       = "${data.aws_iam_policy_document.sns-topic-policy.json}"
}

resource "aws_sqs_queue" "sqs-queue" {
  provider = "aws.sqs"
  name     = "${var.sqs["name"]}"
  policy   = "${data.aws_iam_policy_document.sqs-queue-policy.json}"
}

resource "aws_sns_topic_subscription" "sns-topic" {
  provider  = "aws.sns2sqs"
  topic_arn = "${aws_sns_topic.sns-topic.arn}"
  protocol  = "sqs"
  endpoint  = "${aws_sqs_queue.sqs-queue.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `topic_arn` - (Required) The ARN of the SNS topic to subscribe to
- `protocol` - (Required) The protocol to use. The possible values for this are: `sqs`, `sms`, `lambda`, `application`. (`http` or `https` are partially supported, see below) (`email` is option but unsupported, see below).
- `endpoint` - (Required) The endpoint to send data to, the contents will vary with the protocol. (see below for more information)
- `endpoint_auto_confirms` - (Optional) Boolean indicating whether the end point is capable of auto confirming subscription e.g., PagerDuty (default is false)
- `confirmation_timeout_in_minutes` - (Optional) Integer indicating number of minutes to wait in retying mode for fetching subscription arn before

marking it as failure. Only applicable for http and https protocols (default is 1 minute).
- `raw_message_delivery` - (Optional) Boolean indicating whether or not to enable raw message delivery (the original message is directly passed, not wrapped in JSON with the original message in the message property) (default is false).
- `filter_policy` - (Optional) JSON String with the filter policy that will be used in the subscription to filter messages seen by the target resource. Refer to the SNS docs for more details.
- `delivery_policy` - (Optional) JSON String with the delivery policy (retries, backoff, etc.) that will be used in the subscription - this only applies to HTTP/S subscriptions. Refer to the SNS docs for more details.

» Protocols supported

Supported SNS protocols include:

- `lambda` -- delivery of JSON-encoded message to a lambda function
- `sqs` -- delivery of JSON-encoded message to an Amazon SQS queue
- `application` -- delivery of JSON-encoded message to an EndpointArn for a mobile app and device
- `sms` -- delivery text message

Partially supported SNS protocols include:

- `http` -- delivery of JSON-encoded messages via HTTP. Supported only for the end points that auto confirms the subscription.
- `https` -- delivery of JSON-encoded messages via HTTPS. Supported only for the end points that auto confirms the subscription.

Unsupported protocols include the following:

- `email` -- delivery of message via SMTP
- `email-json` -- delivery of JSON-encoded message via SMTP

These are unsupported because the endpoint needs to be authorized and does not generate an ARN until the target email address has been validated. This breaks the Terraform model and as a result are not currently supported.

» Specifying endpoints

Endpoints have different format requirements according to the protocol that is chosen.

- SQS endpoints come in the form of the SQS queue's ARN (not the URL of the queue) e.g: `arn:aws:sqs:us-west-2:432981146916:terraform-queue-too`
- Application endpoints are also the endpoint ARN for the mobile app and device.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ARN of the subscription
- `topic_arn` - The ARN of the topic the subscription belongs to
- `protocol` - The protocol being used
- `endpoint` - The full endpoint to send data to (SQS ARN, HTTP(S) URL, Application ARN, SMS number, etc.)
- `arn` - The ARN of the subscription stored as a more user-friendly property

» Import

SNS Topic Subscriptions can be imported using the `subscription arn`, e.g.

```
$ terraform import aws_sns_topic_subscription.user_updates_sqs_target arn:aws:sns:us-west-2:
```

» Data Source: aws_sqs_queue

Use this data source to get the ARN and URL of queue in AWS Simple Queue Service (SQS). By using this data source, you can reference SQS queues without having to hardcode the ARNs as input.

» Example Usage

```
data "aws_sqs_queue" "example" {
  name = "queue"
}
```

» Argument Reference

- `name` - (Required) The name of the queue to match.

» Attributes Reference

- `arn` - The Amazon Resource Name (ARN) of the queue.
- `url` - The URL of the queue.

» Resource: aws_sqs_queue

» Example Usage

```
resource "aws_sqs_queue" "terraform_queue" {
  name                      = "terraform-example-queue"
  delay_seconds             = 90
  max_message_size          = 2048
  message_retention_seconds = 86400
  receive_wait_time_seconds = 10
  redrive_policy            = jsonencode({
    deadLetterTargetArn = aws_sqs_queue.terraform_queue_deadletter.arn
    maxReceiveCount     = 4
  })

  tags = {
    Environment = "production"
  }
}
```

» FIFO queue

```
resource "aws_sqs_queue" "terraform_queue" {
  name                        = "terraform-example-queue.fifo"
  fifo_queue                  = true
  content_based_deduplication = true
}
```

» Server-side encryption (SSE)

```
resource "aws_sqs_queue" "terraform_queue" {
  name                              = "terraform-example-queue"
  kms_master_key_id                 = "alias/aws/sqs"
  kms_data_key_reuse_period_seconds = 300
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) This is the human-readable name of the queue. If omitted, Terraform will assign a random name.

- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `visibility_timeout_seconds` - (Optional) The visibility timeout for the queue. An integer from 0 to 43200 (12 hours). The default for this attribute is 30. For more information about visibility timeout, see AWS docs.
- `message_retention_seconds` - (Optional) The number of seconds Amazon SQS retains a message. Integer representing seconds, from 60 (1 minute) to 1209600 (14 days). The default for this attribute is 345600 (4 days).
- `max_message_size` - (Optional) The limit of how many bytes a message can contain before Amazon SQS rejects it. An integer from 1024 bytes (1 KiB) up to 262144 bytes (256 KiB). The default for this attribute is 262144 (256 KiB).
- `delay_seconds` - (Optional) The time in seconds that the delivery of all messages in the queue will be delayed. An integer from 0 to 900 (15 minutes). The default for this attribute is 0 seconds.
- `receive_wait_time_seconds` - (Optional) The time for which a ReceiveMessage call will wait for a message to arrive (long polling) before returning. An integer from 0 to 20 (seconds). The default for this attribute is 0, meaning that the call will return immediately.
- `policy` - (Optional) The JSON policy for the SQS queue. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.
- `redrive_policy` - (Optional) The JSON policy to set up the Dead Letter Queue, see AWS docs. **Note:** when specifying `maxReceiveCount`, you must specify it as an integer (`5`), and not a string (`"5"`).
- `fifo_queue` - (Optional) Boolean designating a FIFO queue. If not set, it defaults to `false` making it standard.
- `content_based_deduplication` - (Optional) Enables content-based deduplication for FIFO queues. For more information, see the related documentation
- `kms_master_key_id` - (Optional) The ID of an AWS-managed customer master key (CMK) for Amazon SQS or a custom CMK. For more information, see Key Terms.
- `kms_data_key_reuse_period_seconds` - (Optional) The length of time, in seconds, for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. An integer representing seconds, between 60 seconds (1 minute) and 86,400 seconds (24 hours). The default is 300 (5 minutes).
- `tags` - (Optional) A mapping of tags to assign to the queue.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The URL for the created Amazon SQS queue.
- `arn` - The ARN of the SQS queue

» Import

SQS Queues can be imported using the `queue url`, e.g.

```
$ terraform import aws_sqs_queue.public_queue https://queue.amazonaws.com/80398EXAMPLE/MyQue
```

» Resource: aws_sqs_queue_policy

Allows you to set a policy of an SQS Queue while referencing ARN of the queue within the policy.

» Example Usage

```
resource "aws_sqs_queue" "q" {
  name = "examplequeue"
}

resource "aws_sqs_queue_policy" "test" {
  queue_url = "${aws_sqs_queue.q.id}"

  policy = <<POLICY
{
  "Version": "2012-10-17",
  "Id": "sqspolicy",
  "Statement": [
    {
      "Sid": "First",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:SendMessage",
      "Resource": "${aws_sqs_queue.q.arn}",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "${aws_sns_topic.example.arn}"
        }
      }
    }
  ]
}
POLICY
}
```

» Argument Reference

The following arguments are supported:

- `queue_url` - (Required) The URL of the SQS Queue to which to attach the policy
- `policy` - (Required) The JSON policy for the SQS queue. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.

» Import

SQS Queue Policies can be imported using the queue URL, e.g.

```
$ terraform import aws_sqs_queue_policy.test https://queue.amazonaws.com/0123456789012/myque
```

» Data Source: aws_ssm_document

Gets the contents of the specified Systems Manager document.

» Example Usage

To get the contents of the document owned by AWS.

```
data "aws_ssm_document" "foo" {
  name            = "AWS-GatherSoftwareInventory"
  document_format = "YAML"
}

output "content" {
  value = "${data.aws_ssm_document.foo.content}"
}
```

To get the contents of the custom document.

```
data "aws_ssm_document" "test" {
  name            = "${aws_ssm_document.test.name}"
  document_format = "JSON"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Systems Manager document.
- `document_format` - (Optional) Returns the document in the specified format. The document format can be either JSON or YAML. JSON is the default format.
- `document_version` - (Optional) The document version for which you want information.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the document.
- `content` - The contents of the document.
- `document_type` - The type of the document.

» Data Source: aws_ssm_parameter

Provides an SSM Parameter data source.

» Example Usage

```
data "aws_ssm_parameter" "foo" {
  name = "foo"
}
```

Note: The unencrypted value of a SecureString will be stored in the raw state as plain-text. Read more about sensitive data in state.

Note: The data source is currently following the behavior of the SSM API to return a string value, regardless of parameter type. For type `StringList`, we can use the built-in split() function to get values in a list. Example: `split(",", data.aws_ssm_parameter.subnets.value)`

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the parameter.
- `with_decryption` - (Optional) Whether to return decrypted `SecureString` value. Defaults to `true`.

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the parameter.
- `name` - The name of the parameter.
- `type` - The type of the parameter. Valid types are `String`, `StringList` and `SecureString`.
- `value` - The value of the parameter.
- `version` - The version of the parameter.

» Resource: aws_ssm_activation

Registers an on-premises server or virtual machine with Amazon EC2 so that it can be managed using Run Command.

» Example Usage

```
resource "aws_iam_role" "test_role" {
  name = "test_role"

  assume_role_policy = <<EOF
  {
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Principal": {"Service": "ssm.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  }
EOF
}

resource "aws_iam_role_policy_attachment" "test_attach" {
  role       = "${aws_iam_role.test_role.name}"
  policy_arn = "arn:aws:iam::aws:policy/service-role/AmazonEC2RoleforSSM"
}

resource "aws_ssm_activation" "foo" {
  name               = "test_ssm_activation"
  description        = "Test"
  iam_role           = "${aws_iam_role.test_role.id}"
  registration_limit = "5"
  depends_on         = ["aws_iam_role_policy_attachment.test_attach"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The default name of the registered managed instance.
- `description` - (Optional) The description of the resource that you want to register.
- `expiration_date` - (Optional) A timestamp in RFC3339 format by which this activation request should expire. The default value is 24 hours from resource creation time.
- `iam_role` - (Required) The IAM Role to attach to the managed instance.
- `registration_limit` - (Optional) The maximum number of managed instances you want to register. The default value is 1 instance.
- `tags` - (Optional) A mapping of tags to assign to the object.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The activation ID.
- `activation_code` - The code the system generates when it processes the activation.
- `name` - The default name of the registered managed instance.
- `description` - The description of the resource that was registered.
- `expired` - If the current activation has expired.
- `expiration_date` - The date by which this activation request should expire. The default value is 24 hours.
- `iam_role` - The IAM Role attached to the managed instance.
- `registration_limit` - The maximum number of managed instances you want to be registered. The default value is 1 instance.
- `registration_count` - The number of managed instances that are currently registered using this activation.

» Resource: aws_ssm_association

Associates an SSM Document to an instance or EC2 tag.

» Example Usage

```
resource "aws_ssm_association" "example" {
  name = "${aws_ssm_document.example.name}"

  targets {
    key    = "InstanceIds"
    values = ["${aws_instance.example.id}"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the SSM document to apply.
- `association_name` - (Optional) The descriptive name for the association.
- `document_version` - (Optional) The document version you want to associate with the target(s). Can be a specific version or the default version.
- `instance_id` - (Optional) The instance ID to apply an SSM document to. Use `targets` with key `InstanceIds` for document schema versions 2.0 and above.
- `output_location` - (Optional) An output location block. Output Location is documented below.
- `parameters` - (Optional) A block of arbitrary string parameters to pass to the SSM document.
- `schedule_expression` - (Optional) A cron expression when the association will be applied to the target(s).
- `targets` - (Optional) A block containing the targets of the SSM association. Targets are documented below. AWS currently supports a maximum of 5 targets.
- `compliance_severity` - (Optional) The compliance severity for the association. Can be one of the following: `UNSPECIFIED`, `LOW`, `MEDIUM`, `HIGH` or `CRITICAL`
- `max_concurrency` - (Optional) The maximum number of targets allowed to run the association at the same time. You can specify a number, for example 10, or a percentage of the target set, for example 10%.
- `max_errors` - (Optional) The number of errors that are allowed before the system stops sending requests to run the association on additional targets. You can specify a number, for example 10, or a percentage of the target set, for example 10%.

Output Location (`output_location`) is an S3 bucket where you want to store the results of this association:

- `s3_bucket_name` - (Required) The S3 bucket name.
- `s3_key_prefix` - (Optional) The S3 bucket prefix. Results stored in the root if not configured.

Targets specify what instance IDs or tags to apply the document to and has these keys:

- `key` - (Required) Either `InstanceIds` or `tag:Tag Name` to specify an EC2 tag.
- `values` - (Required) A list of instance IDs or tag values. AWS currently limits this list size to one value.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `association_id` - The ID of the SSM association.
- `instance_id` - The instance id that the SSM document was applied to.
- `name` - The name of the SSM document to apply.
- `parameters` - Additional parameters passed to the SSM document.

» Import

SSM associations can be imported using the `association_id`, e.g.

```
$ terraform import aws_ssm_association.test-association 10abcdef-0abc-1234-5678-90abcdef1234
```

» Resource: aws_ssm_document

Provides an SSM Document resource

NOTE on updating SSM documents: Only documents with a schema version of 2.0 or greater can update their content once created, see SSM Schema Features. To update a document with an older schema version you must recreate the resource.

» Example Usage

```
resource "aws_ssm_document" "foo" {
  name          = "test_document"
  document_type = "Command"

  content = <<DOC
  {
    "schemaVersion": "1.2",
    "description": "Check ip configuration of a Linux instance.",
    "parameters": {

    },
    "runtimeConfig": {
      "aws:runShellScript": {
        "properties": [
          {
            "id": "0.aws:runShellScript",
            "runCommand": ["ifconfig"]
          }
        ]
      }
    }
  }
DOC
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the document.
- `attachments_source` - (Optional) One or more configuration blocks describing attachments sources to a version of a document. Defined below.
- `content` - (Required) The JSON or YAML content of the document.
- `document_format` - (Optional, defaults to JSON) The format of the document. Valid document types include: `JSON` and `YAML`
- `document_type` - (Required) The type of the document. Valid document types include: `Automation`, `Command`, `Package`, `Policy`, and `Session`
- `permissions` - (Optional) Additional Permissions to attach to the document. See Permissions below for details.
- `tags` - (Optional) A mapping of tags to assign to the object.

» attachments_source

The `attachments_source` block supports the following:

- `key` - (Required) The key describing the location of an attachment to a document. Valid key types include: `SourceUrl` and `S3FileUrl`
- `values` - (Required) The value describing the location of an attachment to a document
- `name` - (Optional) The name of the document attachment file

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `created_date` - The date the document was created.

- description - The description of the document.
- schema_version - The schema version of the document.
- default_version - The default version of the document.
- hash - The sha1 or sha256 of the document content
- hash_type - ”Sha1” ”Sha256”. The hashing algorithm used when hashing the content.
- latest_version - The latest version of the document.
- owner - The AWS user account of the person who created the document.
- status - ”Creating”, ”Active” or ”Deleting”. The current status of the document.
- parameter - The parameters that are available to this document.
- platform_types - A list of OS platforms compatible with this SSM document, either ”Windows” or ”Linux”.

» Permissions

The permissions attribute specifies how you want to share the document. If you share a document privately, you must specify the AWS user account IDs for those people who can use the document. If you share a document publicly, you must specify All as the account ID.

The permissions mapping supports the following:

- type - The permission type for the document. The permission type can be Share.
- account_ids - The AWS user accounts that should have access to the document. The account IDs can either be a group of account IDs or All.

» Import

SSM Documents can be imported using the name, e.g.

```
$ terraform import aws_ssm_document.example example
```

The attachments_source argument does not have an SSM API method for reading the attachment information detail after creation. If the argument is set in the Terraform configuration on an imported resource, Terraform will always show a difference. To workaround this behavior, either omit the argument from the Terraform configuration or use ignore_changes to hide the difference, e.g.

```
resource "aws_ssm_document" "test" {
  name          = "test_document"
  document_type = "Package"

  attachments_source {
    key = "SourceUrl"
    values = ["s3://${aws_s3_bucket.object_bucket.bucket}/test.zip"]
  }

  # There is no AWS SSM API for reading attachments_source info directly
  lifecycle {
    ignore_changes = ["attachments_source"]
  }
}
```

» Resource: aws_ssm_maintenance_window

Provides an SSM Maintenance Window resource

» Example Usage

```
resource "aws_ssm_maintenance_window" "production" {
  name     = "maintenance-window-application"
  schedule = "cron(0 16 ? * TUE *)"
  duration = 3
  cutoff   = 1
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the maintenance window.
- `schedule` - (Required) The schedule of the Maintenance Window in the form of a cron or rate expression.
- `cutoff` - (Required) The number of hours before the end of the Maintenance Window that Systems Manager stops scheduling new tasks for execution.
- `duration` - (Required) The duration of the Maintenance Window in hours.
- `allow_unassociated_targets` - (Optional) Whether targets must be registered with the Maintenance Window before tasks can be defined for those targets.
- `enabled` - (Optional) Whether the maintenance window is enabled. Default: `true`.
- `end_date` - (Optional) Timestamp in ISO-8601 extended format when to no longer run the maintenance window.
- `schedule_timezone` - (Optional) Timezone for schedule in Internet Assigned Numbers Authority (IANA) Time Zone Database format. For example: `America/Los_Angeles`, `etc/UTC`, or `Asia/Seoul`.

- `start_date` - (Optional) Timestamp in ISO-8601 extended format when to begin the maintenance window.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the maintenance window.

» Import

SSM Maintenance Windows can be imported using the `maintenance window id`, e.g. `$ terraform import aws_ssm_maintenance_window.imported-window mw-0123456789`

» Resource: aws_ssm_maintenance_window_target

Provides an SSM Maintenance Window Target resource

» Example Usage

```
resource "aws_ssm_maintenance_window" "window" {
  name     = "maintenance-window-webapp"
  schedule = "cron(0 16 ? * TUE *)"
  duration = 3
  cutoff   = 1
}

resource "aws_ssm_maintenance_window_target" "target1" {
  window_id     = "${aws_ssm_maintenance_window.window.id}"
  name          = "maintenance-window-target"
  description   = "This is a maintenance window target"
  resource_type = "INSTANCE"

  targets {
    key    = "tag:Name"
    values = ["acceptance_test"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `window_id` - (Required) The Id of the maintenance window to register the target with.
- `name` - (Optional) The name of the maintenance window target.
- `description` - (Optional) The description of the maintenance window target.
- `resource_type` - (Required) The type of target being registered with the Maintenance Window. Possible values `INSTANCE`.
- `targets` - (Required) The targets (either instances or tags). Instances are specified using Key=InstanceIds,Values=InstanceId1,InstanceId2. Tags are specified using Key=tag name,Values=tag value.
- `owner_information` - (Optional) User-provided value that will be included in any CloudWatch events raised while running tasks for these targets in this Maintenance Window.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the maintenance window target.

» Resource: aws_ssm_maintenance_window_task

Provides an SSM Maintenance Window Task resource

» Example Usage

» Automation Tasks

```
resource "aws_ssm_maintenance_window_task" "example" {
  max_concurrency  = 2
  max_errors       = 1
  priority         = 1
  service_role_arn = "${aws_iam_role.example.arn}"
  task_arn         = "AWS-RestartEC2Instance"
  task_type        = "AUTOMATION"
  window_id        = "${aws_ssm_maintenance_window.example.id}"

  targets {
    key    = "InstanceIds"
    values = ["${aws_instance.example.id}"]
  }

  task_invocation_parameters {
    automation_parameters {
      document_version = "$LATEST"

      parameter {
        name   = "InstanceId"
        values = ["${aws_instance.example.id}"]
      }
    }
  }
}
```

» Lambda Tasks

```
resource "aws_ssm_maintenance_window_task" "example" {
  max_concurrency  = 2
  max_errors       = 1
  priority         = 1
  service_role_arn = "${aws_iam_role.example.arn}"
  task_arn         = "${aws_lambda_function.example.arn}"
  task_type        = "LAMBDA"
  window_id        = "${aws_ssm_maintenance_window.example.id}"

  targets {
    key    = "InstanceIds"
    values = ["${aws_instance.example.id}"]
  }

  task_invocation_parameters {
    lambda_parameters {
      client_context = "${base64encode("{\"key1\":\"value1\"}")}"
      payload        = "{\"key1\":\"value1\"}"
    }
  }
}
```

» Run Command Tasks

```
resource "aws_ssm_maintenance_window_task" "example" {
  max_concurrency  = 2
  max_errors       = 1
  priority         = 1
  service_role_arn = "${aws_iam_role.example.arn}"
  task_arn         = "AWS-RunShellScript"
  task_type        = "RUN_COMMAND"
  window_id        = "${aws_ssm_maintenance_window.example.id}"

  targets {
    key    = "InstanceIds"
    values = ["${aws_instance.example.id}"]
  }

  task_invocation_parameters {
    run_command_parameters {
      output_s3_bucket     = "${aws_s3_bucket.example.bucket}"
      output_s3_key_prefix = "output"
      service_role_arn     = "${aws_iam_role.example.arn}"
      timeout_seconds      = 600

      notification_config {
        notification_arn    = "${aws_sns_topic.example.arn}"
        notification_events = ["All"]
        notification_type   = ["Command"]
      }

      parameter {
        name   = "commands"
        values = ["date"]
      }
    }
  }
}
```

» Step Function Tasks

```
resource "aws_ssm_maintenance_window_task" "example" {
  max_concurrency  = 2
  max_errors       = 1
  priority         = 1
  service_role_arn = "${aws_iam_role.example.arn}"
  task_arn         = "${aws_sfn_activity.example.id}"
  task_type        = "STEP_FUNCTIONS"
  window_id        = "${aws_ssm_maintenance_window.example.id}"

  targets {
    key    = "InstanceIds"
    values = ["${aws_instance.example.id}"]
  }

  task_invocation_parameters {
    step_functions_parameters {
      input = "{\"key1\":\"value1\"}"
      name  = "example"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `window_id` - (Required) The Id of the maintenance window to register the task with.
- `max_concurrency` - (Required) The maximum number of targets this task can be run for in parallel.
- `max_errors` - (Required) The maximum number of errors allowed before this task stops being scheduled.
- `task_type` - (Required) The type of task being registered. The only allowed value is `RUN_COMMAND`.
- `task_arn` - (Required) The ARN of the task to execute.
- `service_role_arn` - (Required) The role that should be assumed when executing the task.
- `name` - (Optional) The name of the maintenance window task.
- `description` - (Optional) The description of the maintenance window task.
- `targets` - (Required) The targets (either instances or window target ids). Instances are specified using Key=InstanceIds,Values=instanceid1,instanceid2. Window target ids are specified using Key=WindowTargetIds,Values=window target id1, window target id2.
- `priority` - (Optional) The priority of the task in the Maintenance Window, the lower the number the higher the priority. Tasks in a Maintenance Window are scheduled in priority order with tasks that have the same priority scheduled in parallel.
- `logging_info` - (Optional, **Deprecated**) A structure containing information about an Amazon S3 bucket to write instance-level logs to. Use `task_invocation_parameters` configuration block `run_command_parameters` configuration block `output_s3_*` arguments instead. Conflicts with `task_invocation_parameters`. Documented below.
- `task_parameters` - (Optional, **Deprecated**) A structure containing information about parameters required by the particu-

lar `task_arn`. Use `parameter` configuration blocks under the `task_invocation_parameters` configuration block instead. Conflicts with `task_invocation_parameters`. Documented below.
- `task_invocation_parameters` - (Optional) The parameters for task execution. This argument is conflict with `task_parameters` and `logging_info`.

`logging_info` supports the following:

- `s3_bucket_name` - (Required)
- `s3_region` - (Required)
- `s3_bucket_prefix` - (Optional)

`task_parameters` supports the following:

- `name` - (Required)
- `values` - (Required)

`task_invocation_parameters` supports the following:

- `automation_parameters` - (Optional) The parameters for an AUTOMATION task type. Documented below.
- `lambda_parameters` - (Optional) The parameters for a LAMBDA task type. Documented below.
- `run_command_parameters` - (Optional) The parameters for a RUN_COMMAND task type. Documented below.
- `step_functions_parameters` - (Optional) The parameters for a STEP_FUNCTIONS task type. Documented below.

`automation_parameters` supports the following:

- `document_version` - (Optional) The version of an Automation document to use during task execution.
- `parameter` - (Optional) The parameters for the RUN_COMMAND task execution. Documented below.

`lambda_parameters` supports the following:

- `client_context` - (Optional) Pass client-specific information to the Lambda function that you are invoking.
- `payload` - (Optional) JSON to provide to your Lambda function as input.
- `qualifier` - (Optional) Specify a Lambda function version or alias name.

`run_command_parameters` supports the following:

- `comment` - (Optional) Information about the command(s) to execute.
- `document_hash` - (Optional) The SHA-256 or SHA-1 hash created by the system when the document was created. SHA-1 hashes have been deprecated.
- `document_hash_type` - (Optional) SHA-256 or SHA-1. SHA-1 hashes have been deprecated. Valid values: `Sha256` and `Sha1`

- `notification_config` - (Optional) Configurations for sending notifications about command status changes on a per-instance basis. Documented below.
- `output_s3_bucket` - (Optional) The name of the Amazon S3 bucket.
- `output_s3_key_prefix` - (Optional) The Amazon S3 bucket subfolder.
- `parameter` - (Optional) The parameters for the RUN_COMMAND task execution. Documented below.
- `service_role_arn` - (Optional) The IAM service role to assume during task execution.
- `timeout_seconds` - (Optional) If this time is reached and the command has not already started executing, it doesn't run.

`step_functions_parameters` supports the following:

- `input` - (Optional) The inputs for the STEP_FUNCTION task.
- `name` - (Optional) The name of the STEP_FUNCTION task.

`notification_config` supports the following:

- `notification_arn` - (Optional) An Amazon Resource Name (ARN) for a Simple Notification Service (SNS) topic. Run Command pushes notifications about command status changes to this topic.
- `notification_events` - (Optional) The different events for which you can receive notifications. Valid values: `All`, `InProgress`, `Success`, `TimedOut`, `Cancelled`, and `Failed`
- `notification_type` - (Optional) When specified with `Command`, receive notification when the status of a command changes. When specified with `Invocation`, for commands sent to multiple instances, receive notification on a per-instance basis when the status of a command changes. Valid values: `Command` and `Invocation`

`parameter` supports the following:

- `name` - (Required) The parameter name.
- `values` - (Required) The array of strings.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the maintenance window task.

» Import

AWS Maintenance Window Task can be imported using the `window_id` and `window_task_id` separated by `/`.

```
$ terraform import aws_ssm_maintenance_window_task.task <window_id>/<window_task_id>
```

» Resource: aws_ssm_patch_baseline

Provides an SSM Patch Baseline resource

NOTE on Patch Baselines: The `approved_patches` and `approval_rule` are both marked as optional fields, but the Patch Baseline requires that at least one of them is specified.

» Example Usage

Basic usage using `approved_patches` only

```
resource "aws_ssm_patch_baseline" "production" {
  name             = "patch-baseline"
  approved_patches = ["KB123456"]
}
```

Advanced usage, specifying patch filters

```
resource "aws_ssm_patch_baseline" "production" {
  name             = "patch-baseline"
  description      = "Patch Baseline Description"
  approved_patches = ["KB123456", "KB456789"]
  rejected_patches = ["KB987654"]

  global_filter {
    key    = "PRODUCT"
    values = ["WindowsServer2008"]
  }

  global_filter {
    key    = "CLASSIFICATION"
    values = ["ServicePacks"]
  }

  global_filter {
    key    = "MSRC_SEVERITY"
    values = ["Low"]
  }

  approval_rule {
    approve_after_days = 7
    compliance_level   = "HIGH"

    patch_filter {
      key    = "PRODUCT"
      values = ["WindowsServer2016"]
    }

    patch_filter {
      key    = "CLASSIFICATION"
      values = ["CriticalUpdates", "SecurityUpdates", "Updates"]
    }

    patch_filter {
      key    = "MSRC_SEVERITY"
      values = ["Critical", "Important", "Moderate"]
    }
  }

  approval_rule {
    approve_after_days = 7

    patch_filter {
      key    = "PRODUCT"
      values = ["WindowsServer2012"]
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the patch baseline.
- `description` - (Optional) The description of the patch baseline.
- `operating_system` - (Optional) Defines the operating system the patch baseline applies to. Supported operating systems include `WINDOWS`, `AMAZON_LINUX`, `AMAZON_LINUX_2`, `SUSE`, `UBUNTU`, `CENTOS`, and `REDHAT_ENTERPRISE_LINUX`. The Default value is `WINDOWS`.
- `approved_patches_compliance_level` - (Optional) Defines the compliance level for approved patches. This means that if an approved patch is reported as missing, this is the severity of the compliance violation. Valid compliance levels include the following: `CRITICAL`, `HIGH`, `MEDIUM`, `LOW`, `INFORMATIONAL`, `UNSPECIFIED`. The default value is `UNSPECIFIED`.
- `approved_patches` - (Optional) A list of explicitly approved patches for the baseline.
- `rejected_patches` - (Optional) A list of rejected patches.
- `global_filter` - (Optional) A set of global filters used to exclude patches from the baseline. Up to 4 global filters can be specified using Key/Value pairs. Valid Keys are `PRODUCT | CLASSIFICATION | MSRC_SEVERITY |`

PATCH_ID.
- approval_rule - (Optional) A set of rules used to include patches in the baseline. up to 10 approval rules can be specified. Each approval_rule block requires the fields documented below.

The approval_rule block supports:

- approve_after_days - (Required) The number of days after the release date of each patch matched by the rule the patch is marked as approved in the patch baseline. Valid Range: 0 to 100.
- patch_filter - (Required) The patch filter group that defines the criteria for the rule. Up to 4 patch filters can be specified per approval rule using Key/Value pairs. Valid Keys are PRODUCT | CLASSIFICATION | MSRC_SEVERITY | PATCH_ID.
- compliance_level - (Optional) Defines the compliance level for patches approved by this rule. Valid compliance levels include the following: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL, UNSPECIFIED. The default value is UNSPECIFIED.
- enable_non_security - (Optional) Boolean enabling the application of non-security updates. The default value is 'false'. Valid for Linux instances only.
- tags - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The ID of the patch baseline.

» Import

SSM Patch Baselines can be imported by their baseline ID, e.g.

```
$ terraform import aws_ssm_patch_baseline.example pb-12345678
```

» Resource: aws_ssm_patch_group

Provides an SSM Patch Group resource

» Example Usage

```
resource "aws_ssm_patch_baseline" "production" {
  name             = "patch-baseline"
  approved_patches = ["KB123456"]
}

resource "aws_ssm_patch_group" "patchgroup" {
  baseline_id = "${aws_ssm_patch_baseline.production.id}"
  patch_group = "patch-group-name"
}
```

» Argument Reference

The following arguments are supported:

- `baseline_id` - (Required) The ID of the patch baseline to register the patch group with.
- `patch_group` - (Required) The name of the patch group that should be registered with the patch baseline.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the patch baseline.

» Resource: aws_ssm_parameter

Provides an SSM Parameter resource.

» Example Usage

To store a basic string parameter:

```
resource "aws_ssm_parameter" "foo" {
  name  = "foo"
  type  = "String"
  value = "bar"
}
```

To store an encrypted string using the default SSM KMS key:

```
resource "aws_db_instance" "default" {
  allocated_storage    = 10
  storage_type         = "gp2"
  engine               = "mysql"
  engine_version       = "5.7.16"
  instance_class       = "db.t2.micro"
  name                 = "mydb"
  username             = "foo"
  password             = "${var.database_master_password}"
  db_subnet_group_name = "my_database_subnet_group"
  parameter_group_name = "default.mysql5.7"
}

resource "aws_ssm_parameter" "secret" {
  name        = "/${var.environment}/database/password/master"
  description = "The parameter description"
  type        = "SecureString"
  value       = "${var.database_master_password}"

  tags = {
    environment = "${var.environment}"
  }
}
```

Note: The unencrypted value of a SecureString will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the parameter. If the name contains a path (e.g. any forward slashes (`/`)), it must be fully qualified with a leading forward slash (`/`). For additional requirements and constraints, see the AWS SSM User Guide.
- `type` - (Required) The type of the parameter. Valid types are `String`, `StringList` and `SecureString`.
- `value` - (Required) The value of the parameter.
- `description` - (Optional) The description of the parameter.
- `tier` - (Optional) The tier of the parameter. If not specified, will default to `Standard`. Valid tiers are `Standard` and `Advanced`. For more information on parameter tiers, see the AWS SSM Parameter tier comparison and guide.
- `key_id` - (Optional) The KMS key id or arn for encrypting a SecureString.
- `overwrite` - (Optional) Overwrite an existing parameter. If not specified, will default to `false` if the resource has not been created by terraform to avoid overwrite of existing resource and will default to `true` otherwise (terraform lifecycle rules should then be used to manage the update behavior).
- `allowed_pattern` - (Optional) A regular expression used to validate the parameter value.

- `tags` - (Optional) A mapping of tags to assign to the object.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - The ARN of the parameter.
- `name` - (Required) The name of the parameter.
- `description` - (Required) The description of the parameter.
- `type` - (Required) The type of the parameter. Valid types are `String`, `StringList` and `SecureString`.
- `value` - (Required) The value of the parameter.
- `version` - The version of the parameter.

» Import

SSM Parameters can be imported using the `parameter store name`, e.g.

```
$ terraform import aws_ssm_parameter.my_param /my_path/my_paramname
```

» Resource: aws_ssm_resource_data_sync

Provides a SSM resource data sync.

» Example Usage

```
resource "aws_s3_bucket" "hoge" {
  bucket = "tf-test-bucket-1234"
  region = "us-east-1"
}

resource "aws_s3_bucket_policy" "hoge" {
  bucket = "${aws_s3_bucket.hoge.bucket}"

  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SSMBucketPermissionsCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::tf-test-bucket-1234"
        },
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": ["arn:aws:s3:::tf-test-bucket-1234/*"],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
EOF
}

resource "aws_ssm_resource_data_sync" "foo" {
  name = "foo"

  s3_destination = {
    bucket_name = "${aws_s3_bucket.hoge.bucket}"
    region      = "${aws_s3_bucket.hoge.region}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name for the configuration.
- `s3_destination` - (Required) Amazon S3 configuration details for the sync.

» s3_destination

`s3_destination` supports the following:

- `bucket_name` - (Required) Name of S3 bucket where the aggregated data is stored.
- `region` - (Required) Region with the bucket targeted by the Resource Data Sync.
- `kms_key_arn` - (Optional) ARN of an encryption key for a destination in Amazon S3.
- `prefix` - (Optional) Prefix for the bucket.
- `sync_format` - (Optional) A supported sync format. Only JsonSerDe is currently supported. Defaults to JsonSerDe.

» Import

SSM resource data sync can be imported using the `name`, e.g.

```
$ terraform import aws_ssm_resource_data_sync.example example-name
```

» Resource: aws_sfn_activity

Provides a Step Function Activity resource

» Example Usage

```
resource "aws_sfn_activity" "sfn_activity" {
  name = "my-activity"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the activity to create.
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Amazon Resource Name (ARN) that identifies the created activity.
- `name` - The name of the activity.
- `creation_date` - The date the activity was created.

» Import

Activities can be imported using the `arn`, e.g.

```
$ terraform import aws_sfn_activity.foo arn:aws:states:eu-west-1:123456789098:activity:bar
```

» Resource: aws_sfn_state_machine

Provides a Step Function State Machine resource

» Example Usage

```
# ...

resource "aws_sfn_state_machine" "sfn_state_machine" {
  name     = "my-state-machine"
  role_arn = "${aws_iam_role.iam_for_sfn.arn}"

  definition = <<EOF
{
  "Comment": "A Hello World example of the Amazon States Language using an AWS Lambda Functi
  "StartAt": "HelloWorld",
  "States": {
    "HelloWorld": {
      "Type": "Task",
      "Resource": "${aws_lambda_function.lambda.arn}",
      "End": true
    }
  }
}
EOF
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the state machine.
- `definition` - (Required) The Amazon States Language definition of the state machine.
- `role_arn` - (Required) The Amazon Resource Name (ARN) of the IAM role to use for this state machine.
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ARN of the state machine.
- `creation_date` - The date the state machine was created.
- `status` - The current status of the state machine. Either "ACTIVE" or "DELETING".

» Import

State Machines can be imported using the `arn`, e.g.

```
$ terraform import aws_sfn_state_machine.foo arn:aws:states:eu-west-1:123456789098:stateMach
```

» Data Source: aws_storagegateway_local_disk

Retrieve information about a Storage Gateway local disk. The disk identifier is useful for adding the disk as a cache or upload buffer to a gateway.

» Example Usage

```
data "aws_storagegateway_local_disk" "test" {
  disk_path   = "${aws_volume_attachment.test.device_name}"
  gateway_arn = "${aws_storagegateway_gateway.test.arn}"
}
```

» Argument Reference

- `gateway_arn` - (Required) The Amazon Resource Name (ARN) of the gateway.
- `disk_node` - (Optional) The device node of the local disk to retrieve. For example, `/dev/sdb`.
- `disk_path` - (Optional) The device path of the local disk to retrieve. For example, `/dev/xvdb` or `/dev/nvme1n1`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `disk_id` - The disk identifier. e.g. `pci-0000:03:00.0-scsi-0:0:0:0`
- `id` - The disk identifier. e.g. `pci-0000:03:00.0-scsi-0:0:0:0`

» Resource: aws_storagegateway_cache

Manages an AWS Storage Gateway cache.

NOTE: The Storage Gateway API provides no method to remove a cache disk. Destroying this Terraform resource does not perform any Storage Gateway actions.

» Example Usage

```
resource "aws_storagegateway_cache" "example" {
  disk_id     = "${data.aws_storagegateway_local_disk.example.id}"
  gateway_arn = "${aws_storagegateway_gateway.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `disk_id` - (Required) Local disk identifier. For example, `pci-0000:03:00.0-scsi-0:0:0:0`.
- `gateway_arn` - (Required) The Amazon Resource Name (ARN) of the gateway.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Combined gateway Amazon Resource Name (ARN) and local disk identifier.

» Import

`aws_storagegateway_cache` can be imported by using the gateway Amazon Resource Name (ARN) and local disk identifier separated with a colon (`:`), e.g.

```
$ terraform import aws_storagegateway_cache.example arn:aws:storagegateway:us-east-1:1234567
```

» Resource: aws_storagegateway_cached_iscsi_volume

Manages an AWS Storage Gateway cached iSCSI volume.

NOTE: The gateway must have cache added (e.g. via the `aws_storagegateway_cache` resource) before creating volumes otherwise the Storage Gateway API will return an error.

NOTE: The gateway must have an upload buffer added (e.g. via the `aws_storagegateway_upload_buffer` resource) before the volume is operational to clients, however the Storage Gateway API will allow volume creation without error in that case and return volume status as `UPLOAD BUFFER NOT CONFIGURED`.

» Example Usage

NOTE: These examples are referencing the `aws_storagegateway_cache` resource `gateway_arn` attribute to ensure Terraform properly adds cache before creating the volume. If you are not using this method, you may need to declare an expicit dependency (e.g. via `depends_on = ["aws_storagegateway_cache.example"]`) to ensure proper ordering.

» Create Empty Cached iSCSI Volume

```
resource "aws_storagegateway_cached_iscsi_volume" "example" {
  gateway_arn          = "${aws_storagegateway_cache.example.gateway_arn}"
  network_interface_id = "${aws_instance.example.private_ip}"
  target_name          = "example"
  volume_size_in_bytes = 5368709120                                       # 5 GB
}
```

» Create Cached iSCSI Volume From Snapshot

```
resource "aws_storagegateway_cached_iscsi_volume" "example" {
  gateway_arn          = "${aws_storagegateway_cache.example.gateway_arn}"
  network_interface_id = "${aws_instance.example.private_ip}"
  snapshot_id          = "${aws_ebs_snapshot.example.id}"
  target_name          = "example"
  volume_size_in_bytes = "${aws_ebs_snapshot.example.volume_size * 1024 * 1024 * 1024}"
}
```

» Create Cached iSCSI Volume From Source Volume

```
resource "aws_storagegateway_cached_iscsi_volume" "example" {
  gateway_arn          = "${aws_storagegateway_cache.example.gateway_arn}"
  network_interface_id = "${aws_instance.example.private_ip}"
  source_volume_arn    = "${aws_storagegateway_cached_iscsi_volume.existing.arn}"
  target_name          = "example"
  volume_size_in_bytes = "${aws_storagegateway_cached_iscsi_volume.existing.volume_size_in_b
}
```

» Argument Reference

The following arguments are supported:

- `gateway_arn` - (Required) The Amazon Resource Name (ARN) of the gateway.
- `network_interface_id` - (Required) The network interface of the gateway on which to expose the iSCSI target. Only IPv4 addresses are accepted.
- `target_name` - (Required) The name of the iSCSI target used by initiators to connect to the target and as a suffix for the target ARN. The target name must be unique across all volumes of a gateway.
- `volume_size_in_bytes` - (Required) The size of the volume in bytes.
- `snapshot_id` - (Optional) The snapshot ID of the snapshot to restore as the new cached volume. e.g. `snap-1122aabb`.
- `source_volume_arn` - (Optional) The ARN for an existing volume. Specifying this ARN makes the new volume into an exact copy of the specified existing volume's latest recovery point. The `volume_size_in_bytes` value for this new volume must be equal to or larger than the size of the existing volume, in bytes.
- `tags` - (Optional) Key-value mapping of resource tags

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Volume Amazon Resource Name (ARN), e.g. `arn:aws:storagegateway:us-east-1:123456789012`
- `chap_enabled` - Whether mutual CHAP is enabled for the iSCSI target.
- `id` - Volume Amazon Resource Name (ARN), e.g. `arn:aws:storagegateway:us-east-1:123456789012:`
- `lun_number` - Logical disk number.
- `network_interface_port` - The port used to communicate with iSCSI targets.
- `target_arn` - Target Amazon Resource Name (ARN), e.g. `arn:aws:storagegateway:us-east-1:123456`
- `volume_arn` - Volume Amazon Resource Name (ARN), e.g. `arn:aws:storagegateway:us-east-1:12345`
- `volume_id` - Volume ID, e.g. `vol-12345678`.

» Import

`aws_storagegateway_cached_iscsi_volume` can be imported by using the volume Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_storagegateway_cache.example arn:aws:storagegateway:us-east-1:1234567
```

» Resource: aws_storagegateway_gateway

Manages an AWS Storage Gateway file, tape, or volume gateway in the provider region.

NOTE: The Storage Gateway API requires the gateway to be connected to properly return information after activation. If you are receiving `The specified gateway is not connected` errors during resource creation (gateway activation), ensure your gateway instance meets the Storage Gateway requirements.

» Example Usage

» File Gateway

```
resource "aws_storagegateway_gateway" "example" {
  gateway_ip_address = "1.2.3.4"
  gateway_name       = "example"
  gateway_timezone   = "GMT"
  gateway_type       = "FILE_S3"
}
```

» Tape Gateway

```
resource "aws_storagegateway_gateway" "example" {
  gateway_ip_address = "1.2.3.4"
  gateway_name       = "example"
  gateway_timezone   = "GMT"
  gateway_type       = "VTL"
  media_changer_type = "AWS-Gateway-VTL"
  tape_drive_type    = "IBM-ULT3580-TD5"
}
```

» Volume Gateway (Cached)

```
resource "aws_storagegateway_gateway" "example" {
  gateway_ip_address = "1.2.3.4"
  gateway_name       = "example"
  gateway_timezone   = "GMT"
  gateway_type       = "CACHED"
}
```

» Volume Gateway (Stored)

```
resource "aws_storagegateway_gateway" "example" {
  gateway_ip_address = "1.2.3.4"
  gateway_name       = "example"
  gateway_timezone   = "GMT"
  gateway_type       = "STORED"
}
```

» Argument Reference

NOTE: One of `activation_key` or `gateway_ip_address` must be provided for resource creation (gateway activation). Neither is required for resource import. If using `gateway_ip_address`, Terraform must be able to make an HTTP (port 80) GET request to the specified IP address from where it is running.

The following arguments are supported:

- `gateway_name` - (Required) Name of the gateway.
- `gateway_timezone` - (Required) Time zone for the gateway. The time zone is of the format "GMT", "GMT-hr:mm", or "GMT+hr:mm". For example, `GMT-4:00` indicates the time is 4 hours behind GMT. The time zone is used, for example, for scheduling snapshots and your gateway's maintenance schedule.
- `activation_key` - (Optional) Gateway activation key during resource creation. Conflicts with `gateway_ip_address`. Additional information is available in the Storage Gateway User Guide.
- `gateway_ip_address` - (Optional) Gateway IP address to retrieve activation key during resource creation. Conflicts with `activation_key`. Gateway must be accessible on port 80 from where Terraform is running. Additional information is available in the Storage Gateway User Guide.
- `gateway_type` - (Optional) Type of the gateway. The default value is `STORED`. Valid values: `CACHED`, `FILE_S3`, `STORED`, `VTL`.
- `media_changer_type` - (Optional) Type of medium changer to use for tape gateway. Terraform cannot detect drift of this argument. Valid values: `STK-L700`, `AWS-Gateway-VTL`.
- `smb_active_directory_settings` - (Optional) Nested argument with Active Directory domain join information for Server Message Block (SMB) file shares. Only valid for `FILE_S3` gateway type. Must be set before creating `ActiveDirectory` authentication SMB file shares. More details below.
- `smb_guest_password` - (Optional) Guest password for Server Message Block (SMB) file shares. Only valid for `FILE_S3` gateway type. Must be set before creating `GuestAccess` authentication SMB file shares. Terraform can only detect drift of the existence of a guest password, not its

actual value from the gateway. Terraform can however update the password with changing the argument.
- `tape_drive_type` - (Optional) Type of tape drive to use for tape gateway. Terraform cannot detect drift of this argument. Valid values: `IBM-ULT3580-TD5`.
- `tags` - (Optional) Key-value mapping of resource tags

» smb_active_directory_settings

Information to join the gateway to an Active Directory domain for Server Message Block (SMB) file shares.

NOTE It is not possible to unconfigure this setting without recreating the gateway. Also, Terraform can only detect drift of the `domain_name` argument from the gateway.

- `domain_name` - (Required) The name of the domain that you want the gateway to join.
- `password` - (Required) The password of the user who has permission to add the gateway to the Active Directory domain.
- `username` - (Required) The user name of user who has permission to add the gateway to the Active Directory domain.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the gateway.
- `arn` - Amazon Resource Name (ARN) of the gateway.
- `gateway_id` - Identifier of the gateway.

» Timeouts

`aws_storagegateway_gateway` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for gateway activation and connection to Storage Gateway.

» Import

`aws_storagegateway_gateway` can be imported by using the gateway Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_storagegateway_gateway.example arn:aws:storagegateway:us-east-1:12345
```

» Resource: aws_storagegateway_nfs_file_share

Manages an AWS Storage Gateway NFS File Share.

» Example Usage

```
resource "aws_storagegateway_nfs_file_share" "example" {
  client_list  = ["0.0.0.0/0"]
  gateway_arn  = "${aws_storagegateway_gateway.example.arn}"
  location_arn = "${aws_s3_bucket.example.arn}"
  role_arn     = "${aws_iam_role.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `client_list` - (Required) The list of clients that are allowed to access the file gateway. The list must contain either valid IP addresses or valid CIDR blocks. Set to `["0.0.0.0/0"]` to not limit access. Minimum 1 item. Maximum 100 items.
- `gateway_arn` - (Required) Amazon Resource Name (ARN) of the file gateway.
- `location_arn` - (Required) The ARN of the backed storage used for storing file data.
- `role_arn` - (Required) The ARN of the AWS Identity and Access Management (IAM) role that a file gateway assumes when it accesses the underlying storage.
- `default_storage_class` - (Optional) The default storage class for objects put into an Amazon S3 bucket by the file gateway. Defaults to `S3_STANDARD`. Valid values: `S3_STANDARD`, `S3_STANDARD_IA`, `S3_ONEZONE_IA`.
- `guess_mime_type_enabled` - (Optional) Boolean value that enables guessing of the MIME type for uploaded objects based on file extensions. Defaults to `true`.
- `kms_encrypted` - (Optional) Boolean value if `true` to use Amazon S3 server side encryption with your own AWS KMS key, or `false` to use a key managed by Amazon S3. Defaults to `false`.
- `kms_key_arn` - (Optional) Amazon Resource Name (ARN) for KMS key used for Amazon S3 server side encryption. This value can only be set when `kms_encrypted` is true.
- `nfs_file_share_defaults` - (Optional) Nested argument with file share default values. More information below.

- `object_acl` - (Optional) Access Control List permission for S3 bucket objects. Defaults to `private`.
- `read_only` - (Optional) Boolean to indicate write status of file share. File share does not accept writes if `true`. Defaults to `false`.
- `requester_pays` - (Optional) Boolean who pays the cost of the request and the data download from the Amazon S3 bucket. Set this value to `true` if you want the requester to pay instead of the bucket owner. Defaults to `false`.
- `squash` - (Optional) Maps a user to anonymous user. Defaults to `RootSquash`. Valid values: `RootSquash` (only root is mapped to anonymous user), `NoSquash` (no one is mapped to anonymous user), `AllSquash` (everyone is mapped to anonymous user)
- `tags` - (Optional) Key-value mapping of resource tags

» nfs_file_share_defaults

Files and folders stored as Amazon S3 objects in S3 buckets don't, by default, have Unix file permissions assigned to them. Upon discovery in an S3 bucket by Storage Gateway, the S3 objects that represent files and folders are assigned these default Unix permissions.

- `directory_mode` - (Optional) The Unix directory mode in the string form ”nnnn”. Defaults to `"0777"`.
- `file_mode` - (Optional) The Unix file mode in the string form ”nnnn”. Defaults to `"0666"`.
- `group_id` - (Optional) The default group ID for the file share (unless the files have another group ID specified). Defaults to `65534` (`nfsnobody`). Valid values: `0` through `4294967294`.
- `owner_id` - (Optional) The default owner ID for the file share (unless the files have another owner ID specified). Defaults to `65534` (`nfsnobody`). Valid values: `0` through `4294967294`.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the NFS File Share.
- `arn` - Amazon Resource Name (ARN) of the NFS File Share.
- `fileshare_id` - ID of the NFS File Share.
- `path` - File share path used by the NFS client to identify the mount point.

» Timeouts

`aws_storagegateway_nfs_file_share` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for file share creation.
- `update` - (Default `10m`) How long to wait for file share updates.
- `delete` - (Default `10m`) How long to wait for file share deletion.

» Import

`aws_storagegateway_nfs_file_share` can be imported by using the NFS File Share Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_storagegateway_nfs_file_share.example arn:aws:storagegateway:us-east-
```

» Resource: aws_storagegateway_smb_file_share

Manages an AWS Storage Gateway SMB File Share.

» Example Usage

» Active Directory Authentication

NOTE: The gateway must have already joined the Active Directory domain prior to SMB file share creation. e.g. via "SMB Settings" in the AWS Storage Gateway console or `smb_active_directory_settings` in the `aws_storagegateway_gateway` resource.

```
resource "aws_storagegateway_smb_file_share" "example" {
  authentication = "ActiveDirectory"
  gateway_arn    = "${aws_storagegateway_gateway.example.arn}"
  location_arn   = "${aws_s3_bucket.example.arn}"
  role_arn       = "${aws_iam_role.example.arn}"
}
```

» Guest Authentication

NOTE: The gateway must have already had the SMB guest password set prior to SMB file share creation. e.g. via "SMB Settings" in the AWS Storage Gateway console or `smb_guest_password` in the `aws_storagegateway_gateway` resource.

```
resource "aws_storagegateway_smb_file_share" "example" {
  authentication = "GuestAccess"
  gateway_arn    = "${aws_storagegateway_gateway.example.arn}"
  location_arn   = "${aws_s3_bucket.example.arn}"
  role_arn       = "${aws_iam_role.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `gateway_arn` - (Required) Amazon Resource Name (ARN) of the file gateway.
- `location_arn` - (Required) The ARN of the backed storage used for storing file data.
- `role_arn` - (Required) The ARN of the AWS Identity and Access Management (IAM) role that a file gateway assumes when it accesses the underlying storage.
- `authentication` - (Optional) The authentication method that users use to access the file share. Defaults to `ActiveDirectory`. Valid values: `ActiveDirectory`, `GuestAccess`.
- `default_storage_class` - (Optional) The default storage class for objects put into an Amazon S3 bucket by the file gateway. Defaults to `S3_STANDARD`. Valid values: `S3_STANDARD`, `S3_STANDARD_IA`, `S3_ONEZONE_IA`.
- `guess_mime_type_enabled` - (Optional) Boolean value that enables guessing of the MIME type for uploaded objects based on file extensions. Defaults to `true`.
- `invalid_user_list` - (Optional) A list of users in the Active Directory that are not allowed to access the file share. Only valid if `authentication` is set to `ActiveDirectory`.
- `kms_encrypted` - (Optional) Boolean value if `true` to use Amazon S3 server side encryption with your own AWS KMS key, or `false` to use a key managed by Amazon S3. Defaults to `false`.
- `kms_key_arn` - (Optional) Amazon Resource Name (ARN) for KMS key used for Amazon S3 server side encryption. This value can only be set when `kms_encrypted` is true.
- `smb_file_share_defaults` - (Optional) Nested argument with file share default values. More information below.
- `object_acl` - (Optional) Access Control List permission for S3 bucket objects. Defaults to `private`.
- `read_only` - (Optional) Boolean to indicate write status of file share. File share does not accept writes if `true`. Defaults to `false`.
- `requester_pays` - (Optional) Boolean who pays the cost of the request and the data download from the Amazon S3 bucket. Set this value to `true`

if you want the requester to pay instead of the bucket owner. Defaults to `false`.

- `valid_user_list` - (Optional) A list of users in the Active Directory that are allowed to access the file share. Only valid if `authentication` is set to `ActiveDirectory`.
- `tags` - (Optional) Key-value mapping of resource tags

» smb_file_share_defaults

Files and folders stored as Amazon S3 objects in S3 buckets don't, by default, have Unix file permissions assigned to them. Upon discovery in an S3 bucket by Storage Gateway, the S3 objects that represent files and folders are assigned these default Unix permissions.

- `directory_mode` - (Optional) The Unix directory mode in the string form "nnnn". Defaults to `"0777"`.
- `file_mode` - (Optional) The Unix file mode in the string form "nnnn". Defaults to `"0666"`.
- `group_id` - (Optional) The default group ID for the file share (unless the files have another group ID specified). Defaults to `0`. Valid values: `0` through `4294967294`.
- `owner_id` - (Optional) The default owner ID for the file share (unless the files have another owner ID specified). Defaults to `0`. Valid values: `0` through `4294967294`.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Amazon Resource Name (ARN) of the SMB File Share.
- `arn` - Amazon Resource Name (ARN) of the SMB File Share.
- `fileshare_id` - ID of the SMB File Share.
- `path` - File share path used by the NFS client to identify the mount point.

» Timeouts

`aws_storagegateway_smb_file_share` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for file share creation.
- `update` - (Default `10m`) How long to wait for file share updates.
- `delete` - (Default `15m`) How long to wait for file share deletion.

» Import

`aws_storagegateway_smb_file_share` can be imported by using the SMB File Share Amazon Resource Name (ARN), e.g.

```
$ terraform import aws_storagegateway_smb_file_share.example arn:aws:storagegateway:us-east-
```

» Resource: aws_storagegateway_upload_buffer

Manages an AWS Storage Gateway upload buffer.

NOTE: The Storage Gateway API provides no method to remove an upload buffer disk. Destroying this Terraform resource does not perform any Storage Gateway actions.

» Example Usage

```
resource "aws_storagegateway_upload_buffer" "example" {
  disk_id     = "${data.aws_storagegateway_local_disk.example.id}"
  gateway_arn = "${aws_storagegateway_gateway.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `disk_id` - (Required) Local disk identifier. For example, `pci-0000:03:00.0-scsi-0:0:0:0`.
- `gateway_arn` - (Required) The Amazon Resource Name (ARN) of the gateway.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Combined gateway Amazon Resource Name (ARN) and local disk identifier.

» Import

`aws_storagegateway_upload_buffer` can be imported by using the gateway Amazon Resource Name (ARN) and local disk identifier separated with a colon (`:`), e.g.

```
$ terraform import aws_storagegateway_upload_buffer.example arn:aws:storagegateway:us-east-1
```

» Resource: aws_storagegateway_working_storage

Manages an AWS Storage Gateway working storage.

NOTE: The Storage Gateway API provides no method to remove a working storage disk. Destroying this Terraform resource does not perform any Storage Gateway actions.

» Example Usage

```
resource "aws_storagegateway_working_storage" "example" {
  disk_id     = "${data.aws_storagegateway_local_disk.example.id}"
  gateway_arn = "${aws_storagegateway_gateway.example.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `disk_id` - (Required) Local disk identifier. For example, `pci-0000:03:00.0-scsi-0:0:0:0`.
- `gateway_arn` - (Required) The Amazon Resource Name (ARN) of the gateway.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - Combined gateway Amazon Resource Name (ARN) and local disk identifier.

» Import

`aws_storagegateway_working_storage` can be imported by using the gateway Amazon Resource Name (ARN) and local disk identifier separated with a colon (`:`), e.g.

```
$ terraform import aws_storagegateway_working_storage.example arn:aws:storagegateway:us-east
```

» Resource: aws_swf_domain

Provides an SWF Domain resource.

» Example Usage

To register a basic SWF domain:

```
resource "aws_swf_domain" "foo" {
  name                                        = "foo"
  description                                 = "Terraform SWF Domain"
  workflow_execution_retention_period_in_days = 30
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the domain. If omitted, Terraform will assign a random, unique name.
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `description` - (Optional, Forces new resource) The domain description.
- `workflow_execution_retention_period_in_days` - (Required, Forces new resource) Length of time that SWF will continue to retain information about the workflow execution after the workflow execution is complete, must be between 0 and 90 days.
- `tags` - (Optional) Key-value mapping of resource tags

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The name of the domain.
- `arn` - Amazon Resource Name (ARN)

» Import

SWF Domains can be imported using the `name`, e.g.

```
$ terraform import aws_swf_domain.foo test-domain
```

» Data Source: aws_transfer_server

Use this data source to get the ARN of an AWS Transfer Server for use in other resources.

» Example Usage

```
data "aws_transfer_server" "example" {
  server_id = "s-1234567"
}
```

» Argument Reference

- `server_id` - (Required) ID for an SFTP server.

» Attributes Reference

- `arn` - Amazon Resource Name (ARN) of Transfer Server
- `endpoint` - The endpoint of the Transfer Server (e.g. `s-12345678.server.transfer.REGION.amazonaws.`
- `id` - The Server ID of the Transfer Server (e.g. `s-12345678`)
- `identity_provider_type` - The mode of authentication enabled for this service. The default value is `SERVICE_MANAGED`, which allows you to store and access SFTP user credentials within the service. `API_GATEWAY` indicates that user authentication requires a call to an API Gateway endpoint URL provided by you to integrate an identity provider of your choice.
- `invocation_role` - Amazon Resource Name (ARN) of the IAM role used to authenticate the user account with an `identity_provider_type` of `API_GATEWAY`.
- `logging_role` - Amazon Resource Name (ARN) of an IAM role that allows the service to write your SFTP users' activity to your Amazon CloudWatch logs for monitoring and auditing purposes.
- `url` - URL of the service endpoint used to authenticate users with an `identity_provider_type` of `API_GATEWAY`.

» Resource: aws_transfer_server

Provides a AWS Transfer Server resource.

```
resource "aws_iam_role" "foo" {
  name = "tf-test-transfer-server-iam-role"

  assume_role_policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal": {
            "Service": "transfer.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}
EOF
}

resource "aws_iam_role_policy" "foo" {
  name = "tf-test-transfer-server-iam-policy-%s"
  role = "${aws_iam_role.foo.id}"

  policy = <<POLICY
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Sid": "AllowFullAccesstoCloudWatchLogs",
        "Effect": "Allow",
        "Action": [
            "logs:*"
        ],
        "Resource": "*"
        }
    ]
}
POLICY
}

resource "aws_transfer_server" "foo" {
  identity_provider_type = "SERVICE_MANAGED"
  logging_role           = "${aws_iam_role.foo.arn}"

  tags = {
    NAME = "tf-acc-test-transfer-server"
    ENV  = "test"
  }
}
```

» Argument Reference

The following arguments are supported:

- `endpoint_details` - (Optional) The virtual private cloud (VPC) endpoint settings that you want to configure for your SFTP server. Fields documented below.
- `endpoint_type` - (Optional) The type of endpoint that you want your SFTP server connect to. If you connect to a `VPC_ENDPOINT`, your SFTP server isn't accessible over the public internet. If you want to connect your SFTP server via public internet, set `PUBLIC`. Defaults to `PUBLIC`.
- `invocation_role` - (Optional) Amazon Resource Name (ARN) of the IAM role used to authenticate the user account with an `identity_provider_type` of `API_GATEWAY`.
- `url` - (Optional) - URL of the service endpoint used to authenticate users with an `identity_provider_type` of `API_GATEWAY`.
- `identity_provider_type` - (Optional) The mode of authentication enabled for this service. The default value is `SERVICE_MANAGED`, which allows you to store and access SFTP user credentials within the service. `API_GATEWAY` indicates that user authentication requires a call to an API Gateway endpoint URL provided by you to integrate an identity provider of your choice.
- `logging_role` - (Optional) Amazon Resource Name (ARN) of an IAM role that allows the service to write your SFTP users' activity to your Amazon CloudWatch logs for monitoring and auditing purposes.
- `force_destroy` - (Optional) A boolean that indicates all users associated with the server should be deleted so that the Server can be destroyed without error. The default value is `false`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

endpoint_details requires the following:

- `vpc_endpoint_id` - (Required) The ID of the VPC endpoint.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of Transfer Server
- `id` - The Server ID of the Transfer Server (e.g. `s-12345678`)
- `endpoint` - The endpoint of the Transfer Server (e.g. `s-12345678.server.transfer.REGION.amazonaws`

» Import

Transfer Servers can be imported using the `server id`, e.g.

```
$ terraform import aws_transfer_server.bar s-12345678
```

» Resource: aws_transfer_ssh_key

Provides a AWS Transfer User SSH Key resource.

```
resource "aws_transfer_server" "foo" {
  identity_provider_type = "SERVICE_MANAGED"

  tags = {
    NAME = "tf-acc-test-transfer-server"
  }
}

resource "aws_iam_role" "foo" {
  name = "tf-test-transfer-user-iam-role-%s"

  assume_role_policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal": {
            "Service": "transfer.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}
EOF
}

resource "aws_iam_role_policy" "foo" {
  name = "tf-test-transfer-user-iam-policy-%s"
  role = "${aws_iam_role.foo.id}"

  policy = <<POLICY
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFullAccesstoS3",
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": "*"
        }
    ]
}
POLICY
}

resource "aws_transfer_user" "foo" {
  server_id = "${aws_transfer_server.foo.id}"
  user_name = "tftestuser"
  role      = "${aws_iam_role.foo.arn}"

  tags = {
    NAME = "tftestuser"
  }
}

resource "aws_transfer_ssh_key" "foo" {
  server_id = "${aws_transfer_server.foo.id}"
  user_name = "${aws_transfer_user.foo.user_name}"
  body      = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD3F6tyPEFEzV0LX3X8BsXdMsQz1x2cEikKDEY0a
}
```

» Argument Reference

The following arguments are supported:

- `server_id` - (Requirement) The Server ID of the Transfer Server (e.g. `s-12345678`)
- `user_name` - (Requirement) The name of the user account that is assigned to one or more servers.
- `body` - (Requirement) The public key portion of an SSH key pair.

» Import

Transfer SSH Public Key can be imported using the `server_id` and `user_name` and `ssh_public_key_id` separated by `/`.

```
$ terraform import aws_transfer_ssh_key.bar s-12345678/test-username/key-12345
```

» Resource: aws_transfer_user

Provides a AWS Transfer User resource. Managing SSH keys can be accomplished with the `aws_transfer_ssh_key` resource.

```
resource "aws_transfer_server" "foo" {
  identity_provider_type = "SERVICE_MANAGED"

  tags = {
    NAME = "tf-acc-test-transfer-server"
  }
}

resource "aws_iam_role" "foo" {
  name = "tf-test-transfer-user-iam-role"

  assume_role_policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal": {
            "Service": "transfer.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}
EOF
}

resource "aws_iam_role_policy" "foo" {
  name = "tf-test-transfer-user-iam-policy"
  role = "${aws_iam_role.foo.id}"

  policy = <<POLICY
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFullAccesstoS3",
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
      "Resource": "*"
    }
  ]
}
POLICY
}

resource "aws_transfer_user" "foo" {
  server_id = "${aws_transfer_server.foo.id}"
  user_name = "tftestuser"
  role      = "${aws_iam_role.foo.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `server_id` - (Requirement) The Server ID of the Transfer Server (e.g. `s-12345678`)
- `user_name` - (Requirement) The name used for log in to your SFTP server.
- `home_directory` - (Optional) The landing directory (folder) for a user when they log in to the server using their SFTP client. It should begin with a `/`. The first item in the path is the name of the home bucket (accessible as `${Transfer:HomeBucket}` in the policy) and the rest is the home directory (accessible as `${Transfer:HomeDirectory}` in the policy). For example, `/example-bucket-1234/username` would set the home bucket to `example-bucket-1234` and the home directory to `username`.
- `policy` - (Optional) An IAM JSON policy document that scopes down user access to portions of their Amazon S3 bucket. IAM variables you can use inside this policy include `${Transfer:UserName}`, `${Transfer:HomeDirectory}`, and `${Transfer:HomeBucket}`. Since the IAM variable syntax matches Terraform's interpolation syntax, they must be escaped inside Terraform configuration strings (`$${Transfer:UserName}`). These are evaluated on-the-fly when navigating the bucket.
- `role` - (Requirement) Amazon Resource Name (ARN) of an IAM role that allows the service to controls your user's access to your Amazon S3 bucket.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of Transfer User

» Import

Transfer Users can be imported using the `server_id` and `user_name` separated by `/`.

```
$ terraform import aws_transfer_user.bar s-12345678/test-username
```

» Data Source: aws_customer_gateway

Get an existing AWS Customer Gateway.

» Example Usage

```
data "aws_customer_gateway" "foo" {
    filter {
        name = "tag:Name"
    values = ["foo-prod"]
    }
}

resource "aws_vpn_gateway" "main" {
    vpc_id          = "${aws_vpc.main.id}"
    amazon_side_asn = 7224
}

resource "aws_vpn_connection" "transit" {
    vpn_gateway_id      = "${aws_vpn_gateway.main.id}"
    customer_gateway_id = "${data.aws_customer_gateway.foo.id}"
    type                = "${data.aws_customer_gateway.foo.type}"
    static_routes_only  = false
}
```

» Argument Reference

The following arguments are supported:

- `id` - (Optional) The ID of the gateway.
- `filter` - (Optional) One or more name-value pairs to filter by.

» Attribute Reference

In addition to the arguments above, the following attributes are exported:

- `bgp_asn` - (Optional) The gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN).
- `ip_address` - (Optional) The IP address of the gateway's Internet-routable external interface.
- `tags` - Map of key-value pairs assigned to the gateway.
- `type` - (Optional) The type of customer gateway. The only type AWS supports at this time is "ipsec.1".

» Data Source: aws_internet_gateway

`aws_internet_gateway` provides details about a specific Internet Gateway.

» Example Usage

```
variable "vpc_id" {}

data "aws_internet_gateway" "default" {
  filter {
    name   = "attachment.vpc-id"
    values = ["${var.vpc_id}"]
  }
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available Internet Gateway in the current region. The given filters must match exactly one Internet Gateway whose data will be exported as attributes.

- `internet_gateway_id` - (Optional) The id of the specific Internet Gateway to retrieve.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired Internet Gateway.
- `filter` - (Optional) Custom filter block as described below.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. An Internet Gateway will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes except `filter` block are also exported as result attributes. This data source will complete the data by populating any fields that are not included in the configuration with the data for the selected Internet Gateway.

`attachments` are also exported with the following attributes, when there are relevants: Each attachement supports the following:

- `owner_id` - The ID of the AWS account that owns the internet gateway.
- `state` - The current state of the attachment between the gateway and the VPC. Present only if a VPC is attached
- `vpc_id` - The ID of an attached VPC.

» Data Source: aws_nat_gateway

Provides details about a specific Nat Gateway.

» Example Usage

```
variable "subnet_id" {}

data "aws_nat_gateway" "default" {
  subnet_id = "${aws_subnet.public.id}"
}
```

Usage with tags:

```
data "aws_nat_gateway" "default" {
  subnet_id = "${aws_subnet.public.id}"

  tags = {
    Name = "gw NAT"
  }
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available Nat Gateways in the current region. The given filters must match exactly one Nat Gateway whose data will be exported as attributes.

- `id` - (Optional) The id of the specific Nat Gateway to retrieve.
- `subnet_id` - (Optional) The id of subnet that the Nat Gateway resides in.

- `vpc_id` - (Optional) The id of the VPC that the Nat Gateway resides in.
- `state` - (Optional) The state of the NAT gateway (pending | failed | available | deleting | deleted).
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired Nat Gateway.
- `filter` - (Optional) Custom filter block as described below.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. An Nat Gateway will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes except `filter` block are also exported as result attributes. This data source will complete the data by populating any fields that are not included in the configuration with the data for the selected Nat Gateway.

`addresses` are also exported with the following attributes, when they are relevant: Each attachement supports the following:

- `allocation_id` - The Id of the EIP allocated to the selected Nat Gateway.
- `network_interface_id` - The Id of the ENI allocated to the selected Nat Gateway.
- `private_ip` - The private Ip address of the selected Nat Gateway.
- `public_ip` - The public Ip (EIP) address of the selected Nat Gateway.

» Data Source: aws_network_acls

» Example Usage

The following shows outputing all network ACL ids in a vpc.

```
data "aws_network_acls" "example" {
  vpc_id = "${var.vpc_id}"
}

output "example" {
  value = "${data.aws_network_acls.example.ids}"
}
```

The following example retrieves a list of all network ACL ids in a VPC with a custom tag of `Tier` set to a value of ”Private”.

```
data "aws_network_acls" "example" {
  vpc_id = "${var.vpc_id}"

  tags = {
    Tier = "Private"
  }
}
```

The following example retrieves a network ACL id in a VPC which associated with specific subnet.

```
data "aws_network_acls" "example" {
  vpc_id = "${var.vpc_id}"

  filter {
    name   = "association.subnet-id"
    values = ["${aws_subnet.test.id}"]
  }
}
```

» Argument Reference

- `vpc_id` - (Optional) The VPC ID that you want to filter from.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired network ACLs.
- `filter` - (Optional) Custom filter block as described below.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. A VPC will be selected if any one of the given values matches.

» Attributes Reference

- `ids` - A list of all the network ACL ids found. This data source will fail if none are found.

» aws_network_interface

Use this data source to get information about a Network Interface.

» Example Usage

```
data "aws_network_interface" "bar" {
  id = "eni-01234567"
}
```

» Argument Reference

The following arguments are supported:

- `id` – (Optional) The identifier for the network interface.
- `filter` – (Optional) One or more name/value pairs to filter off of. There are several valid keys, for a full reference, check out describe-network-interfaces in the AWS CLI reference.

» Attributes Reference

See the Network Interface for details on the returned attributes.

Additionally, the following attributes are exported:

- `association` - The association information for an Elastic IP address (IPv4) associated with the network interface. See supported fields below.
- `availability_zone` - The Availability Zone.
- `description` - Description of the network interface.
- `interface_type` - The type of interface.
- `ipv6_addresses` - List of IPv6 addresses to assign to the ENI.
- `mac_address` - The MAC address.
- `owner_id` - The AWS account ID of the owner of the network interface.
- `private_dns_name` - The private DNS name.
- `private_ip` - The private IPv4 address of the network interface within the subnet.
- `private_ips` - The private IPv4 addresses associated with the network interface.
- `requester_id` - The ID of the entity that launched the instance on your behalf.
- `security_groups` - The list of security groups for the network interface.
- `subnet_id` - The ID of the subnet.
- `tags` - Any tags assigned to the network interface.
- `vpc_id` - The ID of the VPC.

» association

- allocation_id - The allocation ID.
- association_id - The association ID.
- ip_owner_id - The ID of the Elastic IP address owner.
- public_dns_name - The public DNS name.
- public_ip - The address of the Elastic IP address bound to the network interface.

» Import

Elastic Network Interfaces can be imported using the id, e.g.

```
$ terraform import aws_network_interface.test eni-12345
```

» Data Source: aws_network_interfaces

» Example Usage

The following shows outputing all network interface ids in a region.

```
data "aws_network_interfaces" "example" {}

output "example" {
  value = "${data.aws_network_interfaces.example.ids}"
}
```

The following example retrieves a list of all network interface ids with a custom tag of Name set to a value of test.

```
data "aws_network_interfaces" "example" {
  tags = {
    Name = "test"
  }
}

output "example1" {
  value = "${data.aws_network_interfaces.example.ids}"
}
```

The following example retrieves a network interface ids which associated with specific subnet.

```
data "aws_network_interfaces" "example" {
  filter {
    name   = "subnet-id"
    values = ["${aws_subnet.test.id}"]
  }
}

output "example" {
  value = "${data.aws_network_interfaces.example.ids}"
}
```

» Argument Reference

- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired network interfaces.
- `filter` - (Optional) Custom filter block as described below.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field.

» Attributes Reference

- `ids` - A list of all the network interface ids found. This data source will fail if none are found.

» Data Source: aws_prefix_list

`aws_prefix_list` provides details about a specific prefix list (PL) in the current region.

This can be used both to validate a prefix list given in a variable and to obtain the CIDR blocks (IP address ranges) for the associated AWS service. The latter may be useful e.g. for adding network ACL rules.

» Example Usage

```
resource "aws_vpc_endpoint" "private_s3" {
  vpc_id       = "${aws_vpc.foo.id}"
  service_name = "com.amazonaws.us-west-2.s3"
}
```

```
data "aws_prefix_list" "private_s3" {
  prefix_list_id = "${aws_vpc_endpoint.private_s3.prefix_list_id}"
}

resource "aws_network_acl" "bar" {
  vpc_id = "${aws_vpc.foo.id}"
}

resource "aws_network_acl_rule" "private_s3" {
  network_acl_id = "${aws_network_acl.bar.id}"
  rule_number    = 200
  egress         = false
  protocol       = "tcp"
  rule_action    = "allow"
  cidr_block     = "${data.aws_prefix_list.private_s3.cidr_blocks[0]}"
  from_port      = 443
  to_port        = 443
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available prefix lists. The given filters must match exactly one prefix list whose data will be exported as attributes.

- `prefix_list_id` - (Optional) The ID of the prefix list to select.
- `name` - (Optional) The name of the prefix list to select.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the selected prefix list.
- `name` - The name of the selected prefix list.
- `cidr_blocks` - The list of CIDR blocks for the AWS service associated with the prefix list.

» Data Source: aws_route

`aws_route` provides details about a specific Route.

This resource can prove useful when finding the resource associated with a CIDR. For example, finding the peering connection associated with a CIDR value.

» Example Usage

The following example shows how one might use a CIDR value to find a network interface id and use this to create a data source of that network interface.

```
variable "subnet_id" {}

data "aws_route_table" "selected" {
  subnet_id = "${var.subnet_id}"
}

data "aws_route" "route" {
  route_table_id         = "${aws_route_table.selected.id}"
  destination_cidr_block = "10.0.1.0/24"
}

data "aws_network_interface" "interface" {
  network_interface_id = "${data.aws_route.route.network_interface_id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available Route in the current region. The given filters must match exactly one Route whose data will be exported as attributes.

- `route_table_id` - (Required) The id of the specific Route Table containing the Route entry.
- `destination_cidr_block` - (Optional) The CIDR block of the Route belonging to the Route Table.
- `destination_ipv6_cidr_block` - (Optional) The IPv6 CIDR block of the Route belonging to the Route Table.
- `egress_only_gateway_id` - (Optional) The Egress Only Gateway ID of the Route belonging to the Route Table.
- `gateway_id` - (Optional) The Gateway ID of the Route belonging to the Route Table.
- `instance_id` - (Optional) The Instance ID of the Route belonging to the Route Table.

- `nat_gateway_id` - (Optional) The NAT Gateway ID of the Route belonging to the Route Table.
- `transit_gateway_id` - (Optional) The EC2 Transit Gateway ID of the Route belonging to the Route Table.
- `vpc_peering_connection_id` - (Optional) The VPC Peering Connection ID of the Route belonging to the Route Table.
- `network_interface_id` - (Optional) The Network Interface ID of the Route belonging to the Route Table.

» Attributes Reference

All of the argument attributes are also exported as result attributes when there is data available. For example, the `vpc_peering_connection_id` field will be empty when the route is attached to a Network Interface.

» Data Source: aws_route_table

`aws_route_table` provides details about a specific Route Table.

This resource can prove useful when a module accepts a Subnet id as an input variable and needs to, for example, add a route in the Route Table.

» Example Usage

The following example shows how one might accept a Route Table id as a variable and use this data source to obtain the data necessary to create a route.

```
variable "subnet_id" {}

data "aws_route_table" "selected" {
  subnet_id = "${var.subnet_id}"
}

resource "aws_route" "route" {
  route_table_id            = "${data.aws_route_table.selected.id}"
  destination_cidr_block    = "10.0.1.0/22"
  vpc_peering_connection_id = "pcx-45ff3dc1"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available Route Table in the current region. The given filters must match exactly one Route Table whose data will be exported as attributes.

- `filter` - (Optional) Custom filter block as described below.
- `route_table_id` - (Optional) The id of the specific Route Table to retrieve.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired Route Table.
- `vpc_id` - (Optional) The id of the VPC that the desired Route Table belongs to.
- `subnet_id` - (Optional) The id of a Subnet which is connected to the Route Table (not exported if not passed as a parameter).
- `gateway_id` - (Optional) The id of an Internet Gateway or Virtual Private Gateway which is connected to the Route Table (not exported if not passed as a parameter).

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. A Route Table will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes except the `filter` block are also exported as result attributes. This data source will complete the data by populating any fields that are not included in the configuration with the data for the selected Route Table. In addition the following attributes are exported:

- `owner_id` - The ID of the AWS account that owns the route table

`routes` are also exported with the following attributes, when there are relevants: Each route supports the following:

- `cidr_block` - The CIDR block of the route.
- `ipv6_cidr_block` - The IPv6 CIDR block of the route.
- `egress_only_gateway_id` - The ID of the Egress Only Internet Gateway.
- `gateway_id` - The Internet Gateway ID.
- `nat_gateway_id` - The NAT Gateway ID.
- `instance_id` - The EC2 instance ID.

- `transit_gateway_id` - The EC2 Transit Gateway ID.
- `vpc_peering_connection_id` - The VPC Peering ID.
- `network_interface_id` - The ID of the elastic network interface (eni) to use.

`associations` are also exported with the following attributes:

- `route_table_association_id` - The Association ID.
- `route_table_id` - The Route Table ID.
- `subnet_id` - The Subnet ID. Only set when associated with a Subnet.
- `gateway_id` - The Gateway ID. Only set when associated with an Internet Gateway or Virtual Private Gateway.
- `main` - If the Association due to the Main Route Table.

» Data Source: aws_route_tables

This resource can be useful for getting back a list of route table ids to be referenced elsewhere.

» Example Usage

The following adds a route for a particular cidr block to every (private kops) route table in a specified vpc to use a particular vpc peering connection.

```
data "aws_route_tables" "rts" {
  vpc_id = "${var.vpc_id}"

  filter {
    name   = "tag:kubernetes.io/kops/role"
    values = ["private*"]
  }
}

resource "aws_route" "r" {
  count                     = "${length(data.aws_route_tables.rts.ids)}"
  route_table_id            = "${data.aws_route_tables.rts.ids[count.index]}"
  destination_cidr_block    = "10.0.1.0/22"
  vpc_peering_connection_id = "pcx-0e9a7a9ecd137dc54"
}
```

» Argument Reference

- `filter` - (Optional) Custom filter block as described below.

- `vpc_id` - (Optional) The VPC ID that you want to filter from.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired route tables.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. A Route Table will be selected if any one of the given values matches.

» Attributes Reference

- `ids` - A list of all the route table ids found. This data source will fail if none are found.

» Data Source: aws_security_group

`aws_security_group` provides details about a specific Security Group.

This resource can prove useful when a module accepts a Security Group id as an input variable and needs to, for example, determine the id of the VPC that the security group belongs to.

» Example Usage

The following example shows how one might accept a Security Group id as a variable and use this data source to obtain the data necessary to create a subnet.

```
variable "security_group_id" {}

data "aws_security_group" "selected" {
  id = "${var.security_group_id}"
}

resource "aws_subnet" "subnet" {
  vpc_id     = "${data.aws_security_group.selected.vpc_id}"
  cidr_block = "10.0.1.0/24"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available security group in the current region. The given filters must match exactly one security group whose data will be exported as attributes.

- `filter` - (Optional) Custom filter block as described below.
- `id` - (Optional) The id of the specific security group to retrieve.
- `name` - (Optional) The name that the desired security group must have.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired security group.
- `vpc_id` - (Optional) The id of the VPC that the desired security group belongs to.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. A Security Group will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes except `filter` blocks are also exported as result attributes. This data source will complete the data by populating any fields that are not included in the configuration with the data for the selected Security Group.

The following fields are also exported:

- `description` - The description of the security group.
- `arn` - The computed ARN of the security group.

Note: The default security group for a VPC has the name `default`.

» Data Source: aws_security_groups

Use this data source to get IDs and VPC membership of Security Groups that are created outside of Terraform.

» Example Usage

```
data "aws_security_groups" "test" {
  tags = {
    Application = "k8s"
    Environment = "dev"
  }
}
```

```
data "aws_security_groups" "test" {
  filter {
    name   = "group-name"
    values = ["*nodes*"]
  }

  filter {
    name   = "vpc-id"
    values = ["${var.vpc_id}"]
  }
}
```

» Argument Reference

- `tags` - (Optional) A mapping of tags, each pair of which must exactly match for desired security groups.
- `filter` - (Optional) One or more name/value pairs to use as filters. There are several valid keys, for a full reference, check out describe-security-groups in the AWS CLI reference.

» Attributes Reference

- `ids` - IDs of the matches security groups.
- `vpc_ids` - The VPC IDs of the matched security groups. The data source's tag or filter *will span VPCs* unless the `vpc-id` filter is also used.

» Data Source: aws_subnet

`aws_subnet` provides details about a specific VPC subnet.

This resource can prove useful when a module accepts a subnet id as an input variable and needs to, for example, determine the id of the VPC that the subnet belongs to.

» Example Usage

The following example shows how one might accept a subnet id as a variable and use this data source to obtain the data necessary to create a security group that allows connections from hosts in that subnet.

```
variable "subnet_id" {}

data "aws_subnet" "selected" {
  id = "${var.subnet_id}"
}

resource "aws_security_group" "subnet" {
  vpc_id = "${data.aws_subnet.selected.vpc_id}"

  ingress {
    cidr_blocks = ["${data.aws_subnet.selected.cidr_block}"]
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
  }
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available subnets in the current region. The given filters must match exactly one subnet whose data will be exported as attributes.

- `availability_zone` - (Optional) The availability zone where the subnet must reside.
- `availability_zone_id` - (Optional) The ID of the Availability Zone for the subnet.
- `cidr_block` - (Optional) The cidr block of the desired subnet.
- `ipv6_cidr_block` - (Optional) The Ipv6 cidr block of the desired subnet
- `default_for_az` - (Optional) Boolean constraint for whether the desired subnet must be the default subnet for its associated availability zone.
- `filter` - (Optional) Custom filter block as described below.
- `id` - (Optional) The id of the specific subnet to retrieve.
- `state` - (Optional) The state that the desired subnet must have.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired subnet.

- `vpc_id` - (Optional) The id of the VPC that the desired subnet belongs to.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API. For example, if matching against tag `Name`, use:

```
data "aws_subnet" "selected" {
  filter {
    name   = "tag:Name"
    values = [""]       # insert value here
  }
}
```

- `values` - (Required) Set of values that are accepted for the given field. A subnet will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes except `filter` blocks are also exported as result attributes. This data source will complete the data by populating any fields that are not included in the configuration with the data for the selected subnet.

In addition the following attributes are exported:

- `arn` - The ARN of the subnet.
- `owner_id` - The ID of the AWS account that owns the subnet.

» Data Source: aws_subnet_ids

`aws_subnet_ids` provides a list of ids for a vpc_id

This resource can be useful for getting back a set of subnet ids for a vpc.

» Example Usage

The following shows outputing all cidr blocks for every subnet id in a vpc.

```
data "aws_subnet_ids" "example" {
  vpc_id = var.vpc_id
}

data "aws_subnet" "example" {
  for_each = data.aws_subnet_ids.example.ids
  id       = each.value
}

output "subnet_cidr_blocks" {
  value = [for s in data.aws_subnet.for-each-example : s.cidr_block]
}
```

The following example retrieves a list of all subnets in a VPC with a custom tag of `Tier` set to a value of ”Private” so that the `aws_instance` resource can loop through the subnets, putting instances across availability zones.

```
data "aws_subnet_ids" "private" {
  vpc_id = var.vpc_id

  tags = {
    Tier = "Private"
  }
}

resource "aws_instance" "app" {
  for_each      = data.aws_subnet_ids.example.ids
  ami           = var.ami
  instance_type = "t2.micro"
  subnet_id     = each.value
}
```

» Argument Reference

- `vpc_id` - (Required) The VPC ID that you want to filter from.
- `filter` - (Optional) Custom filter block as described below.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired subnets.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API. For example, if matching against tag `Name`, use:

```
data "aws_subnet_ids" "selected" {
  filter {
    name   = "tag:Name"
    values = [""] # insert values here
  }
}
```

- `values` - (Required) Set of values that are accepted for the given field. Subnet IDs will be selected if any one of the given values match.

» Attributes Reference

- `ids` - A set of all the subnet ids found. This data source will fail if none are found.

» Data Source: aws_vpc

`aws_vpc` provides details about a specific VPC.

This resource can prove useful when a module accepts a vpc id as an input variable and needs to, for example, determine the CIDR block of that VPC.

» Example Usage

The following example shows how one might accept a VPC id as a variable and use this data source to obtain the data necessary to create a subnet within it.

```
variable "vpc_id" {}

data "aws_vpc" "selected" {
  id = "${var.vpc_id}"
}

resource "aws_subnet" "example" {
  vpc_id            = "${data.aws_vpc.selected.id}"
  availability_zone = "us-west-2a"
  cidr_block        = "${cidrsubnet(data.aws_vpc.selected.cidr_block, 4, 1)}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VPCs in the current region. The given filters must match exactly one VPC whose data will be exported as attributes.

- `cidr_block` - (Optional) The cidr block of the desired VPC.
- `dhcp_options_id` - (Optional) The DHCP options id of the desired VPC.
- `default` - (Optional) Boolean constraint on whether the desired VPC is the default VPC for the region.

- `filter` - (Optional) Custom filter block as described below.
- `id` - (Optional) The id of the specific VPC to retrieve.
- `state` - (Optional) The current state of the desired VPC. Can be either `"pending"` or `"available"`.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired VPC.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. A VPC will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes except `filter` blocks are also exported as result attributes. This data source will complete the data by populating any fields that are not included in the configuration with the data for the selected VPC.

The following attribute is additionally exported:

- `arn` - Amazon Resource Name (ARN) of VPC
- `enable_dns_support` - Whether or not the VPC has DNS support
- `enable_dns_hostnames` - Whether or not the VPC has DNS hostname support
- `instance_tenancy` - The allowed tenancy of instances launched into the selected VPC. May be any of `"default"`, `"dedicated"`, or `"host"`.
- `ipv6_association_id` - The association ID for the IPv6 CIDR block.
- `ipv6_cidr_block` - The IPv6 CIDR block.
- `main_route_table_id` - The ID of the main route table associated with this VPC.
- `owner_id` - The ID of the AWS account that owns the VPC.

`cidr_block_associations` is also exported with the following attributes:

- `association_id` - The association ID for the the IPv4 CIDR block.
- `cidr_block` - The CIDR block for the association.
- `state` - The State of the association.

» Data Source: aws_vpc_dhcp_options

Retrieve information about an EC2 DHCP Options configuration.

» Example Usage

» Lookup by DHCP Options ID

```
data "aws_vpc_dhcp_options" "example" {
  dhcp_options_id = "dopts-12345678"
}
```

» Lookup by Filter

```
data "aws_vpc_dhcp_options" "example" {
  filter {
    name   = "key"
    values = ["domain-name"]
  }

  filter {
    name   = "value"
    values = ["example.com"]
  }
}
```

» Argument Reference

- `dhcp_options_id` - (Optional) The EC2 DHCP Options ID.
- `filter` - (Optional) List of custom filters as described below.

» filter

For more information about filtering, see the EC2 API documentation.

- `name` - (Required) The name of the field to filter.
- `values` - (Required) Set of values for filtering.

» Attributes Reference

- `dhcp_options_id` - EC2 DHCP Options ID
- `domain_name` - The suffix domain name to used when resolving non Fully Qualified Domain Names. e.g. the `search` value in the `/etc/resolv.conf` file.
- `domain_name_servers` - List of name servers.
- `id` - EC2 DHCP Options ID
- `netbios_name_servers` - List of NETBIOS name servers.

- netbios_node_type - The NetBIOS node type (1, 2, 4, or 8). For more information about these node types, see RFC 2132.
- ntp_servers - List of NTP servers.
- tags - A mapping of tags assigned to the resource.
- owner_id - The ID of the AWS account that owns the DHCP options set.

» Data Source: aws_vpc_endpoint

The VPC Endpoint data source provides details about a specific VPC endpoint.

» Example Usage

```
# Declare the data source
data "aws_vpc_endpoint" "s3" {
  vpc_id       = "${aws_vpc.foo.id}"
  service_name = "com.amazonaws.us-west-2.s3"
}

resource "aws_vpc_endpoint_route_table_association" "private_s3" {
  vpc_endpoint_id = "${data.aws_vpc_endpoint.s3.id}"
  route_table_id  = "${aws_route_table.private.id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VPC endpoints. The given filters must match exactly one VPC endpoint whose data will be exported as attributes.

- filter - (Optional) Custom filter block as described below.
- id - (Optional) The ID of the specific VPC Endpoint to retrieve.
- service_name - (Optional) The AWS service name of the specific VPC Endpoint to retrieve.
- state - (Optional) The state of the specific VPC Endpoint to retrieve.
- tags - (Optional) A mapping of tags, each pair of which must exactly match a pair on the specific VPC Endpoint to retrieve.
- vpc_id - (Optional) The ID of the VPC in which the specific VPC Endpoint is used.

More complex filters can be expressed using one or more filter sub-blocks, which take the following arguments:

- name - (Required) The name of the field to filter by, as defined by the underlying AWS API.

- `values` - (Required) Set of values that are accepted for the given field. A VPC Endpoint will be selected if any one of the given values matches.

» Attributes Reference

In addition to all arguments above except `filter`, the following attributes are exported:

- `cidr_blocks` - The list of CIDR blocks for the exposed AWS service. Applicable for endpoints of type `Gateway`.
- `dns_entry` - The DNS entries for the VPC Endpoint. Applicable for endpoints of type `Interface`. DNS blocks are documented below.
- `network_interface_ids` - One or more network interfaces for the VPC Endpoint. Applicable for endpoints of type `Interface`.
- `owner_id` - The ID of the AWS account that owns the VPC endpoint.
- `policy` - The policy document associated with the VPC Endpoint. Applicable for endpoints of type `Gateway`.
- `prefix_list_id` - The prefix list ID of the exposed AWS service. Applicable for endpoints of type `Gateway`.
- `private_dns_enabled` - Whether or not the VPC is associated with a private hosted zone - `true` or `false`. Applicable for endpoints of type `Interface`.
- `requester_managed` - Whether or not the VPC Endpoint is being managed by its service - `true` or `false`.
- `route_table_ids` - One or more route tables associated with the VPC Endpoint. Applicable for endpoints of type `Gateway`.
- `security_group_ids` - One or more security groups associated with the network interfaces. Applicable for endpoints of type `Interface`.
- `subnet_ids` - One or more subnets in which the VPC Endpoint is located. Applicable for endpoints of type `Interface`.
- `vpc_endpoint_type` - The VPC Endpoint type, `Gateway` or `Interface`.

DNS blocks (for `dns_entry`) support the following attributes:

- `dns_name` - The DNS name.
- `hosted_zone_id` - The ID of the private hosted zone.

» Data Source: aws_vpc_endpoint_service

The VPC Endpoint Service data source details about a specific service that can be specified when creating a VPC endpoint within the region configured in the provider.

» Example Usage

AWS service usage:

```
# Declare the data source
data "aws_vpc_endpoint_service" "s3" {
  service = "s3"
}

# Create a VPC
resource "aws_vpc" "foo" {
  cidr_block = "10.0.0.0/16"
}

# Create a VPC endpoint
resource "aws_vpc_endpoint" "ep" {
  vpc_id       = "${aws_vpc.foo.id}"
  service_name = "${data.aws_vpc_endpoint_service.s3.service_name}"
}
```

Non-AWS service usage:

```
data "aws_vpc_endpoint_service" "custome" {
  service_name = "com.amazonaws.vpce.us-west-2.vpce-svc-0e87519c997c63cd8"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VPC endpoint services. The given filters must match exactly one VPC endpoint service whose data will be exported as attributes.

- `service` - (Optional) The common name of an AWS service (e.g. `s3`).
- `service_name` - (Optional) The service name that can be specified when creating a VPC endpoint.

NOTE: One of `service` or `service_name` must be specified.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `acceptance_required` - Whether or not VPC endpoint connection requests to the service must be accepted by the service owner - `true` or `false`.
- `availability_zones` - The Availability Zones in which the service is available.

- base_endpoint_dns_names - The DNS names for the service.
- manages_vpc_endpoints - Whether or not the service manages its VPC endpoints - true or false.
- owner - The AWS account ID of the service owner or amazon.
- private_dns_name - The private DNS name for the service.
- service_id - The ID of the endpoint service.
- service_type - The service type, Gateway or Interface.
- tags - A mapping of tags assigned to the resource.
- vpc_endpoint_policy_supported - Whether or not the service supports endpoint policies - true or false.

» Data Source: aws_vpc_peering_connection

The VPC Peering Connection data source provides details about a specific VPC peering connection.

» Example Usage

```
# Declare the data source
data "aws_vpc_peering_connection" "pc" {
  vpc_id          = "${aws_vpc.foo.id}"
  peer_cidr_block = "10.0.1.0/22"
}

# Create a route table
resource "aws_route_table" "rt" {
  vpc_id = "${aws_vpc.foo.id}"
}

# Create a route
resource "aws_route" "r" {
  route_table_id            = "${aws_route_table.rt.id}"
  destination_cidr_block    = "${data.aws_vpc_peering_connection.pc.peer_cidr_block}"
  vpc_peering_connection_id = "${data.aws_vpc_peering_connection.pc.id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VPC peering connection. The given filters must match exactly one VPC peering connection whose data will be exported as attributes.

- id - (Optional) The ID of the specific VPC Peering Connection to retrieve.

- `status` - (Optional) The status of the specific VPC Peering Connection to retrieve.
- `vpc_id` - (Optional) The ID of the requester VPC of the specific VPC Peering Connection to retrieve.
- `owner_id` - (Optional) The AWS account ID of the owner of the requester VPC of the specific VPC Peering Connection to retrieve.
- `cidr_block` - (Optional) The CIDR block of the requester VPC of the specific VPC Peering Connection to retrieve.
- `region` - (Optional) The region of the requester VPC of the specific VPC Peering Connection to retrieve.
- `peer_vpc_id` - (Optional) The ID of the accepter VPC of the specific VPC Peering Connection to retrieve.
- `peer_owner_id` - (Optional) The AWS account ID of the owner of the accepter VPC of the specific VPC Peering Connection to retrieve.
- `peer_cidr_block` - (Optional) The CIDR block of the accepter VPC of the specific VPC Peering Connection to retrieve.
- `peer_region` - (Optional) The region of the accepter VPC of the specific VPC Peering Connection to retrieve.
- `filter` - (Optional) Custom filter block as described below.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired VPC Peering Connection.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. A VPC Peering Connection will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes except `filter` are also exported as result attributes.

- `accepter` - A configuration block that describes VPC Peering Connection options set for the accepter VPC.
- `requester` - A configuration block that describes VPC Peering Connection options set for the requester VPC.

» Accepter and Requester Attributes Reference

- `allow_remote_vpc_dns_resolution` - Indicates whether a local VPC can resolve public DNS hostnames to private IP addresses when queried from instances in a peer VPC.
- `allow_classic_link_to_remote_vpc` - Indicates whether a local ClassicLink connection can communicate with the peer VPC over the VPC peering connection.
- `allow_vpc_to_remote_classic_link` - Indicates whether a local VPC can communicate with a ClassicLink connection in the peer VPC over the VPC peering connection.

» Data Source: aws_vpcs

This resource can be useful for getting back a list of VPC Ids for a region.

The following example retrieves a list of VPC Ids with a custom tag of `service` set to a value of ”production”.

» Example Usage

The following shows outputing all VPC Ids.

```
data "aws_vpcs" "foo" {
  tags = {
    service = "production"
  }
}

output "foo" {
  value = "${data.aws_vpcs.foo.ids}"
}
```

An example use case would be interpolate the `aws_vpcs` output into `count` of an aws_flow_log resource.

```
data "aws_vpcs" "foo" {}

resource "aws_flow_log" "test_flow_log" {
  count = "${length(data.aws_vpcs.foo.ids)}"

  # ...
  vpc_id = "${element(data.aws_vpcs.foo.ids, count.index)}"

  # ...
}

output "foo" {
  value = "${data.aws_vpcs.foo.ids}"
}
```

» Argument Reference

- tags - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired vpcs.
- filter - (Optional) Custom filter block as described below.

More complex filters can be expressed using one or more filter sub-blocks, which take the following arguments:

- name - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- values - (Required) Set of values that are accepted for the given field. A VPC will be selected if any one of the given values matches.

» Attributes Reference

- ids - A list of all the VPC Ids found. This data source will fail if none are found.

» Data Source: aws_vpn_gateway

The VPN Gateway data source provides details about a specific VPN gateway.

» Example Usage

```
data "aws_vpn_gateway" "selected" {
  filter {
    name   = "tag:Name"
    values = ["vpn-gw"]
  }
}

output "vpn_gateway_id" {
  value = "${data.aws_vpn_gateway.selected.id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VPN gateways. The given filters must match exactly one VPN gateway whose data will be exported as attributes.

- `id` - (Optional) The ID of the specific VPN Gateway to retrieve.
- `state` - (Optional) The state of the specific VPN Gateway to retrieve.
- `availability_zone` - (Optional) The Availability Zone of the specific VPN Gateway to retrieve.
- `attached_vpc_id` - (Optional) The ID of a VPC attached to the specific VPN Gateway to retrieve.
- `filter` - (Optional) Custom filter block as described below.
- `tags` - (Optional) A mapping of tags, each pair of which must exactly match a pair on the desired VPN Gateway.
- `amazon_side_asn` - (Optional) The Autonomous System Number (ASN) for the Amazon side of the specific VPN Gateway to retrieve.

More complex filters can be expressed using one or more `filter` sub-blocks, which take the following arguments:

- `name` - (Required) The name of the field to filter by, as defined by the underlying AWS API.
- `values` - (Required) Set of values that are accepted for the given field. A VPN Gateway will be selected if any one of the given values matches.

» Attributes Reference

All of the argument attributes are also exported as result attributes.

» Resource: aws_customer_gateway

Provides a customer gateway inside a VPC. These objects can be connected to VPN gateways via VPN connections, and allow you to establish tunnels between your network and the VPC.

» Example Usage

```
resource "aws_customer_gateway" "main" {
  bgp_asn    = 65000
  ip_address = "172.83.124.10"
  type       = "ipsec.1"

  tags = {
    Name = "main-customer-gateway"
  }
}
```

» Argument Reference

The following arguments are supported:

- `bgp_asn` - (Required) The gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN).
- `ip_address` - (Required) The IP address of the gateway's Internet-routable external interface.
- `type` - (Required) The type of customer gateway. The only type AWS supports at this time is ”ipsec.1”.
- `tags` - (Optional) Tags to apply to the gateway.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The amazon-assigned ID of the gateway.
- `bgp_asn` - The gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN).
- `ip_address` - The IP address of the gateway's Internet-routable external interface.
- `type` - The type of customer gateway.
- `tags` - Tags applied to the gateway.

» Import

Customer Gateways can be imported using the `id`, e.g.

```
$ terraform import aws_customer_gateway.main cgw-b4dc3961
```

» Resource: aws_default_network_acl

Provides a resource to manage the default AWS Network ACL. VPC Only.

Each VPC created in AWS comes with a Default Network ACL that can be managed, but not destroyed. **This is an advanced resource**, and has special caveats to be aware of when using it. Please read this document in its entirety before using this resource.

The `aws_default_network_acl` behaves differently from normal resources, in that Terraform does not *create* this resource, but instead attempts to "adopt" it into management. We can do this because each VPC created has a Default Network ACL that cannot be destroyed, and is created with a known set of default rules.

When Terraform first adopts the Default Network ACL, it **immediately removes all rules in the ACL**. It then proceeds to create any rules specified in the configuration. This step is required so that only the rules specified in the configuration are created.

This resource treats its inline rules as absolute; only the rules defined inline are created, and any additions/removals external to this resource will result in diffs being shown. For these reasons, this resource is incompatible with the `aws_network_acl_rule` resource.

For more information about Network ACLs, see the AWS Documentation on Network ACLs.

» Basic Example Usage, with default rules

The following config gives the Default Network ACL the same rules that AWS includes, but pulls the resource under management by Terraform. This means that any ACL rules added or changed will be detected as drift.

```
resource "aws_vpc" "mainvpc" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_default_network_acl" "default" {
  default_network_acl_id = "${aws_vpc.mainvpc.default_network_acl_id}"

  ingress {
    protocol   = -1
    rule_no    = 100
    action     = "allow"
    cidr_block = # set a CIDR block here
    from_port  = 0
    to_port    = 0
  }

  egress {
    protocol   = -1
    rule_no    = 100
    action     = "allow"
    cidr_block = "0.0.0.0/0"
    from_port  = 0
    to_port    = 0
  }
}
```

» Example config to deny all Egress traffic, allowing Ingress

The following denies all Egress traffic by omitting any `egress` rules, while including the default `ingress` rule to allow all traffic.

```
resource "aws_vpc" "mainvpc" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_default_network_acl" "default" {
  default_network_acl_id = "${aws_vpc.mainvpc.default_network_acl_id}"

  ingress {
    protocol   = -1
    rule_no    = 100
    action     = "allow"
    cidr_block = # set a CIDR block here
    from_port  = 0
    to_port    = 0
  }
}
```

» Example config to deny all traffic to any Subnet in the Default Network ACL:

This config denies all traffic in the Default ACL. This can be useful if you want a locked down default to force all resources in the VPC to assign a non-default ACL.

```
resource "aws_vpc" "mainvpc" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_default_network_acl" "default" {
  default_network_acl_id = "${aws_vpc.mainvpc.default_network_acl_id}"
  # no rules defined, deny all traffic in this ACL
}
```

» Argument Reference

The following arguments are supported:

- `default_network_acl_id` - (Required) The Network ACL ID to manage. This attribute is exported from `aws_vpc`, or manually found via the AWS Console.
- `subnet_ids` - (Optional) A list of Subnet IDs to apply the ACL to. See the notes below on managing Subnets in the Default Network ACL
- `ingress` - (Optional) Specifies an ingress rule. Parameters defined below.
- `egress` - (Optional) Specifies an egress rule. Parameters defined below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Both `egress` and `ingress` support the following keys:

- `from_port` - (Required) The from port to match.
- `to_port` - (Required) The to port to match.
- `rule_no` - (Required) The rule number. Used for ordering.
- `action` - (Required) The action to take.
- `protocol` - (Required) The protocol to match. If using the `-1` 'all' protocol, you must specify a from and to port of 0.
- `cidr_block` - (Optional) The CIDR block to match. This must be a valid network mask.
- `ipv6_cidr_block` - (Optional) The IPv6 CIDR block.
- `icmp_type` - (Optional) The ICMP type to be used. Default 0.
- `icmp_code` - (Optional) The ICMP type code to be used. Default 0.

Note: For more information on ICMP types and codes, see here: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

» Managing Subnets in the Default Network ACL

Within a VPC, all Subnets must be associated with a Network ACL. In order to "delete" the association between a Subnet and a non-default Network ACL, the association is destroyed by replacing it with an association between the Subnet and the Default ACL instead.

When managing the Default Network ACL, you cannot "remove" Subnets. Instead, they must be reassigned to another Network ACL, or the Subnet itself must be destroyed. Because of these requirements, removing the `subnet_ids` attribute from the configuration of a `aws_default_network_acl` resource may result in a reoccurring plan, until the Subnets are reassigned to another Network ACL or are destroyed.

Because Subnets are by default associated with the Default Network ACL, any non-explicit association will show up as a plan to remove the Subnet. For example: if you have a custom `aws_network_acl` with two subnets attached, and you remove the `aws_network_acl` resource, after successfully destroying this resource future plans will show a diff on the managed `aws_default_network_acl`, as those two Subnets have been orphaned by the now destroyed network acl and thus adopted by the Default Network ACL. In order to avoid a reoccurring plan, they will need to be reassigned, destroyed, or added to the `subnet_ids` attribute of the `aws_default_network_acl` entry.

As an alternative to the above, you can also specify the following lifecycle configuration in your `aws_default_network_acl` resource:

```
lifecycle {
  ignore_changes = ["subnet_ids"]
}
```

» Removing `aws_default_network_acl` from your configuration

Each AWS VPC comes with a Default Network ACL that cannot be deleted. The `aws_default_network_acl` allows you to manage this Network ACL, but Terraform cannot destroy it. Removing this resource from your configuration will remove it from your statefile and management, **but will not destroy the Network ACL.** All Subnets associations and ingress or egress rules will be left as they are at the time of removal. You can resume managing them via the AWS Console.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the Default Network ACL
- `vpc_id` - The ID of the associated VPC
- `ingress` - Set of ingress rules
- `egress` - Set of egress rules
- `subnet_ids` – IDs of associated Subnets
- `owner_id` - The ID of the AWS account that owns the Default Network ACL

» Resource: aws_default_route_table

Provides a resource to manage a Default VPC Routing Table.

Each VPC created in AWS comes with a Default Route Table that can be managed, but not destroyed. **This is an advanced resource**, and has special caveats to be aware of when using it. Please read this document in its entirety before using this resource. It is recommended you **do not** use both `aws_default_route_table` to manage the default route table **and** use the `aws_main_route_table_association`, due to possible conflict in routes.

The `aws_default_route_table` behaves differently from normal resources, in that Terraform does not *create* this resource, but instead attempts to "adopt" it into management. We can do this because each VPC created has a Default Route Table that cannot be destroyed, and is created with a single route.

When Terraform first adopts the Default Route Table, it **immediately removes all defined routes**. It then proceeds to create any routes specified in the configuration. This step is required so that only the routes specified in the configuration present in the Default Route Table.

For more information about Route Tables, see the AWS Documentation on Route Tables.

For more information about managing normal Route Tables in Terraform, see our documentation on aws_route_table.

NOTE on Route Tables and Routes: Terraform currently provides both a standalone Route resource and a Route Table resource with routes defined in-line. At this time you cannot use a Route Table with in-line routes in conjunction with any Route resources. Doing so will cause a conflict of rule settings and will overwrite routes.

» Example usage with tags:

```
resource "aws_default_route_table" "r" {
  default_route_table_id = "${aws_vpc.foo.default_route_table_id}"

  route {
    # ...
  }

  tags = {
    Name = "default table"
  }
}
```

» Argument Reference

The following arguments are supported:

- `default_route_table_id` - (Required) The ID of the Default Routing Table.
- `route` - (Optional) A list of route objects. Their keys are documented below. This argument is processed in attribute-as-blocks mode.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `propagating_vgws` - (Optional) A list of virtual gateways for propagation.

» route Argument Reference

One of the following destination arguments must be supplied:

- `cidr_block` - (Required) The CIDR block of the route.
- `ipv6_cidr_block` - (Optional) The Ipv6 CIDR block of the route

One of the following target arguments must be supplied:

- `egress_only_gateway_id` - (Optional) Identifier of a VPC Egress Only Internet Gateway.
- `gateway_id` - (Optional) Identifier of a VPC internet gateway or a virtual private gateway.
- `instance_id` - (Optional) Identifier of an EC2 instance.
- `nat_gateway_id` - (Optional) Identifier of a VPC NAT gateway.
- `network_interface_id` - (Optional) Identifier of an EC2 network interface.
- `transit_gateway_id` - (Optional) Identifier of an EC2 Transit Gateway.
- `vpc_peering_connection_id` - (Optional) Identifier of a VPC peering connection.

Note that the default route, mapping the VPC's CIDR block to "local", is created implicitly and cannot be specified.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the routing table
- `owner_id` - The ID of the AWS account that owns the route table

» Resource: aws_default_security_group

Provides a resource to manage the default AWS Security Group.

For EC2 Classic accounts, each region comes with a Default Security Group. Additionally, each VPC created in AWS comes with a Default Security Group that can be managed, but not destroyed. **This is an advanced resource**, and has special caveats to be aware of when using it. Please read this document in its entirety before using this resource.

The `aws_default_security_group` behaves differently from normal resources, in that Terraform does not *create* this resource, but instead "adopts" it into management. We can do this because these default security groups cannot be destroyed, and are created with a known set of default ingress/egress rules.

When Terraform first adopts the Default Security Group, it **immediately removes all ingress and egress rules in the Security Group**. It then proceeds to create any rules specified in the configuration. This step is required so that only the rules specified in the configuration are created.

This resource treats its inline rules as absolute; only the rules defined inline are created, and any additions/removals external to this resource will result in diff shown. For these reasons, this resource is incompatible with the `aws_security_group_rule` resource.

For more information about Default Security Groups, see the AWS Documentation on Default Security Groups.

» Basic Example Usage, with default rules

The following config gives the Default Security Group the same rules that AWS provides by default, but pulls the resource under management by Terraform. This means that any ingress or egress rules added or changed will be detected as drift.

```
resource "aws_vpc" "mainvpc" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_default_security_group" "default" {
  vpc_id = "${aws_vpc.mainvpc.id}"

  ingress {
    protocol  = -1
    self      = true
    from_port = 0
    to_port   = 0
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

» Example config to deny all Egress traffic, allowing Ingress

The following denies all Egress traffic by omitting any `egress` rules, while including the default `ingress` rule to allow all traffic.

```
resource "aws_vpc" "mainvpc" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_default_security_group" "default" {
  vpc_id = "${aws_vpc.mainvpc.id}"

  ingress {
    protocol  = -1
    self      = true
    from_port = 0
    to_port   = 0
  }
}
```

» Argument Reference

The arguments of an `aws_default_security_group` differ slightly from `aws_security_group` resources. Namely, the `name` argument is computed, and the `name_prefix` attribute removed. The following arguments are still supported:

- `ingress` - (Optional) Can be specified multiple times for each ingress rule. Each ingress block supports fields documented below.
- `egress` - (Optional, VPC only) Can be specified multiple times for each egress rule. Each egress block supports fields documented below.
- `vpc_id` - (Optional, Forces new resource) The VPC ID. **Note that changing the `vpc_id` will *not* restore any default security group rules that were modified, added, or removed.** It will be left in its current state
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Usage

With the exceptions mentioned above, `aws_default_security_group` should identical behavior to `aws_security_group`. Please consult AWS_SECURITY_GROUP for further usage documentation.

» Removing `aws_default_security_group` from your configuration

Each AWS VPC (or region, if using EC2 Classic) comes with a Default Security Group that cannot be deleted. The `aws_default_security_group` allows you to manage this Security Group, but Terraform cannot destroy it. Removing this resource from your configuration will remove it from your statefile and management, but will not destroy the Security Group. All ingress or egress rules will be left as they are at the time of removal. You can resume managing them via the AWS Console.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the security group
- `vpc_id` - The VPC ID.
- `owner_id` - The owner ID.
- `name` - The name of the security group
- `description` - The description of the security group
- `ingress` - The ingress rules. See above for more.
- `egress` - The egress rules. See above for more.

» Resource: aws_default_subnet

Provides a resource to manage a default AWS VPC subnet in the current region.

The `aws_default_subnet` behaves differently from normal resources, in that Terraform does not *create* this resource, but instead "adopts" it into management.

» Example Usage

Basic usage with tags:

```
resource "aws_default_subnet" "default_az1" {
  availability_zone = "us-west-2a"

  tags = {
    Name = "Default subnet for us-west-2a"
  }
}
```

» Argument Reference

The arguments of an aws_default_subnet differ from aws_subnet resources. Namely, the availability_zone argument is required and the availability_zone_id, vpc_id, cidr_block, ipv6_cidr_block, and assign_ipv6_address_on_creation arguments are computed. The following arguments are still supported:

- map_public_ip_on_launch - (Optional) Specify true to indicate that instances launched into the subnet should be assigned a public IP address.
- tags - (Optional) A mapping of tags to assign to the resource.

» Removing aws_default_subnet from your configuration

The aws_default_subnet resource allows you to manage a region's default VPC subnet, but Terraform cannot destroy it. Removing this resource from your configuration will remove it from your statefile and management, but will not destroy the subnet. You can resume managing the subnet via the AWS Console.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The ID of the subnet
- availability_zone- The AZ for the subnet.
- availability_zone_id- The AZ ID of the subnet.
- cidr_block - The CIDR block for the subnet.
- vpc_id - The VPC ID.
- ipv6_association_id - The association ID for the IPv6 CIDR block.
- ipv6_cidr_block - The IPv6 CIDR block.
- owner_id - The ID of the AWS account that owns the subnet.

» Resource: aws_default_vpc

Provides a resource to manage the default AWS VPC in the current region.

For AWS accounts created after 2013-12-04, each region comes with a Default VPC. **This is an advanced resource**, and has special caveats to be aware of when using it. Please read this document in its entirety before using this resource.

The aws_default_vpc behaves differently from normal resources, in that Terraform does not *create* this resource, but instead "adopts" it into management.

» Example Usage

Basic usage with tags:

```
resource "aws_default_vpc" "default" {
  tags = {
    Name = "Default VPC"
  }
}
```

» Argument Reference

The arguments of an `aws_default_vpc` differ slightly from `aws_vpc` resources. Namely, the `cidr_block`, `instance_tenancy` and `assign_generated_ipv6_cidr_block` arguments are computed. The following arguments are still supported:

- `enable_dns_support` - (Optional) A boolean flag to enable/disable DNS support in the VPC. Defaults true.
- `enable_dns_hostnames` - (Optional) A boolean flag to enable/disable DNS hostnames in the VPC. Defaults false.
- `enable_classiclink` - (Optional) A boolean flag to enable/disable ClassicLink for the VPC. Only valid in regions and accounts that support EC2 Classic. See the ClassicLink documentation for more information. Defaults false.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Removing `aws_default_vpc` from your configuration

The `aws_default_vpc` resource allows you to manage a region's default VPC, but Terraform cannot destroy it. Removing this resource from your configuration will remove it from your statefile and management, but will not destroy the VPC. You can resume managing the VPC via the AWS Console.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of VPC
- `id` - The ID of the VPC
- `cidr_block` - The CIDR block of the VPC
- `instance_tenancy` - Tenancy of instances spin up within VPC.
- `enable_dns_support` - Whether or not the VPC has DNS support
- `enable_dns_hostnames` - Whether or not the VPC has DNS hostname support
- `enable_classiclink` - Whether or not the VPC has Classiclink enabled

- `assign_generated_ipv6_cidr_block` - Whether or not an Amazon-provided IPv6 CIDR block with a /56 prefix length for the VPC was assigned
- `main_route_table_id` - The ID of the main route table associated with this VPC. Note that you can change a VPC's main route table by using an `aws_main_route_table_association`
- `default_network_acl_id` - The ID of the network ACL created by default on VPC creation
- `default_security_group_id` - The ID of the security group created by default on VPC creation
- `default_route_table_id` - The ID of the route table created by default on VPC creation
- `ipv6_association_id` - The association ID for the IPv6 CIDR block of the VPC
- `ipv6_cidr_block` - The IPv6 CIDR block of the VPC
- `owner_id` - The ID of the AWS account that owns the VPC.

» Import

Default VPCs can be imported using the `vpc id`, e.g.

```
$ terraform import aws_default_vpc.default vpc-a01106c2
```

» Resource: aws_default_vpc_dhcp_options

Provides a resource to manage the default AWS DHCP Options Set in the current region.

Each AWS region comes with a default set of DHCP options. **This is an advanced resource**, and has special caveats to be aware of when using it. Please read this document in its entirety before using this resource.

The `aws_default_vpc_dhcp_options` behaves differently from normal resources, in that Terraform does not *create* this resource, but instead "adopts" it into management.

» Example Usage

Basic usage with tags:

```
resource "aws_default_vpc_dhcp_options" "default" {
  tags = {
    Name = "Default DHCP Option Set"
  }
}
```

» Argument Reference

The arguments of an `aws_default_vpc_dhcp_options` differ slightly from `aws_vpc_dhcp_options` resources. Namely, the `domain_name`, `domain_name_servers` and `ntp_servers` arguments are computed. The following arguments are still supported:

- `netbios_name_servers` - (Optional) List of NETBIOS name servers.
- `netbios_node_type` - (Optional) The NetBIOS node type (1, 2, 4, or 8). AWS recommends to specify 2 since broadcast and multicast are not supported in their network. For more information about these node types, see RFC 2132.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Removing `aws_default_vpc_dhcp_options` from your configuration

The `aws_default_vpc_dhcp_options` resource allows you to manage a region's default DHCP Options Set, but Terraform cannot destroy it. Removing this resource from your configuration will remove it from your statefile and management, but will not destroy the DHCP Options Set. You can resume managing the DHCP Options Set via the AWS Console.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the DHCP Options Set.
- `owner_id` - The ID of the AWS account that owns the DHCP options set.

» Resource: aws_egress_only_internet_gateway

[IPv6 only] Creates an egress-only Internet gateway for your VPC. An egress-only Internet gateway is used to enable outbound communication over IPv6 from instances in your VPC to the Internet, and prevents hosts outside of your VPC from initiating an IPv6 connection with your instance.

» Example Usage

```
resource "aws_vpc" "foo" {
  cidr_block                       = "10.1.0.0/16"
  assign_generated_ipv6_cidr_block = true
}

resource "aws_egress_only_internet_gateway" "foo" {
  vpc_id = "${aws_vpc.foo.id}"
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Required) The VPC ID to create in.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the Egress Only Internet Gateway.

» Resource: aws_flow_log

Provides a VPC/Subnet/ENI Flow Log to capture IP traffic for a specific network interface, subnet, or VPC. Logs are sent to a CloudWatch Log Group or a S3 Bucket.

» Example Usage

» CloudWatch Logging

```
resource "aws_flow_log" "example" {
  iam_role_arn    = "${aws_iam_role.example.arn}"
  log_destination = "${aws_cloudwatch_log_group.example.arn}"
  traffic_type    = "ALL"
  vpc_id          = "${aws_vpc.example.id}"
}

resource "aws_cloudwatch_log_group" "example" {
  name = "example"
}

resource "aws_iam_role" "example" {
  name = "example"
  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "vpc-flow-logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "example" {
  name = "example"
  role = "${aws_iam_role.example.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}
```

» S3 Logging

```
resource "aws_flow_log" "example" {
  log_destination      = "${aws_s3_bucket.example.arn}"
  log_destination_type = "s3"
  traffic_type         = "ALL"
  vpc_id               = "${aws_vpc.example.id}"
}

resource "aws_s3_bucket" "example" {
  bucket = "example"
}
```

» Argument Reference

NOTE: One of `eni_id`, `subnet_id`, or `vpc_id` must be specified.

The following arguments are supported:

- `traffic_type` - (Required) The type of traffic to capture. Valid values: `ACCEPT`,`REJECT`, `ALL`.
- `eni_id` - (Optional) Elastic Network Interface ID to attach to
- `iam_role_arn` - (Optional) The ARN for the IAM role that's used to post flow logs to a CloudWatch Logs log group
- `log_destination_type` - (Optional) The type of the logging destination. Valid values: `cloud-watch-logs`, `s3`. Default: `cloud-watch-logs`.
- `log_destination` - (Optional) The ARN of the logging destination.
- `log_group_name` - (Optional) *Deprecated:* Use `log_destination` instead. The name of the CloudWatch log group.
- `subnet_id` - (Optional) Subnet ID to attach to
- `vpc_id` - (Optional) VPC ID to attach to
- `log_format` - (Optional) The fields to include in the flow log record, in the order in which they should appear.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The Flow Log ID

» Import

Flow Logs can be imported using the `id`, e.g.

```
$ terraform import aws_flow_log.test_flow_log fl-1a2b3c4d
```

» Resource: aws_internet_gateway

Provides a resource to create a VPC Internet Gateway.

» Example Usage

```
resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.main.id}"

  tags = {
    Name = "main"
  }
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Required) The VPC ID to create in.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Note: It's recommended to denote that the AWS Instance or Elastic IP depends on the Internet Gateway. For example:

```
resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.main.id}"
}

resource "aws_instance" "foo" {
  depends_on = ["aws_internet_gateway.gw"]
}
```

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the Internet Gateway.
- `owner_id` - The ID of the AWS account that owns the internet gateway.

» Import

Internet Gateways can be imported using the `id`, e.g.

```
$ terraform import aws_internet_gateway.gw igw-c0a643a9
```

» Resource: aws_main_route_table_association

Provides a resource for managing the main routing table of a VPC.

» Example Usage

```
resource "aws_main_route_table_association" "a" {
  vpc_id         = "${aws_vpc.foo.id}"
  route_table_id = "${aws_route_table.bar.id}"
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Required) The ID of the VPC whose main route table should be set
- `route_table_id` - (Required) The ID of the Route Table to set as the new main route table for the target VPC

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the Route Table Association
- `original_route_table_id` - Used internally, see **Notes** below

» Notes

On VPC creation, the AWS API always creates an initial Main Route Table. This resource records the ID of that Route Table under `original_route_table_id`. The "Delete" action for a `main_route_table_association` consists of resetting this original table as the Main Route Table for the VPC. You'll see this additional Route Table in the AWS console; it must remain intact in order for the `main_route_table_association` delete to work properly.

» Resource: aws_nat_gateway

Provides a resource to create a VPC NAT Gateway.

» Example Usage

```
resource "aws_nat_gateway" "gw" {
  allocation_id = "${aws_eip.nat.id}"
  subnet_id     = "${aws_subnet.public.id}"
}
```

Usage with tags:

```
resource "aws_nat_gateway" "gw" {
  allocation_id = "${aws_eip.nat.id}"
  subnet_id     = "${aws_subnet.public.id}"

  tags = {
    Name = "gw NAT"
  }
}
```

» Argument Reference

The following arguments are supported:

- `allocation_id` - (Required) The Allocation ID of the Elastic IP address for the gateway.
- `subnet_id` - (Required) The Subnet ID of the subnet in which to place the gateway.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Note: It's recommended to denote that the NAT Gateway depends on the Internet Gateway for the VPC in which the NAT Gateway's subnet is located. For example:

```
resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.main.id}"
}

resource "aws_nat_gateway" "gw" {
  //other arguments

  depends_on = ["aws_internet_gateway.gw"]
}
```

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the NAT Gateway.

- `allocation_id` - The Allocation ID of the Elastic IP address for the gateway.
- `subnet_id` - The Subnet ID of the subnet in which the NAT gateway is placed.
- `network_interface_id` - The ENI ID of the network interface created by the NAT gateway.
- `private_ip` - The private IP address of the NAT Gateway.
- `public_ip` - The public IP address of the NAT Gateway.

» Import

NAT Gateways can be imported using the `id`, e.g.

```
$ terraform import aws_nat_gateway.private_gw nat-05dba92075d71c408
```

» Resource: aws_network_acl

Provides an network ACL resource. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

NOTE on Network ACLs and Network ACL Rules: Terraform currently provides both a standalone Network ACL Rule resource and a Network ACL resource with rules defined in-line. At this time you cannot use a Network ACL with in-line rules in conjunction with any Network ACL Rule resources. Doing so will cause a conflict of rule settings and will overwrite rules.

» Example Usage

```
resource "aws_network_acl" "main" {
  vpc_id = "${aws_vpc.main.id}"

  egress {
    protocol   = "tcp"
    rule_no    = 200
    action     = "allow"
    cidr_block = "10.3.0.0/18"
    from_port  = 443
    to_port    = 443
  }

  ingress {
    protocol   = "tcp"
    rule_no    = 100
    action     = "allow"
    cidr_block = "10.3.0.0/18"
    from_port  = 80
    to_port    = 80
  }

  tags = {
    Name = "main"
  }
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Required) The ID of the associated VPC.
- `subnet_ids` - (Optional) A list of Subnet IDs to apply the ACL to
- `ingress` - (Optional) Specifies an ingress rule. Parameters defined below. This argument is processed in attribute-as-blocks mode.
- `egress` - (Optional) Specifies an egress rule. Parameters defined below. This argument is processed in attribute-as-blocks mode.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Both `egress` and `ingress` support the following keys:

- `from_port` - (Required) The from port to match.
- `to_port` - (Required) The to port to match.
- `rule_no` - (Required) The rule number. Used for ordering.
- `action` - (Required) The action to take.
- `protocol` - (Required) The protocol to match. If using the -1 'all' protocol, you must specify a from and to port of 0.
- `cidr_block` - (Optional) The CIDR block to match. This must be a valid network mask.
- `ipv6_cidr_block` - (Optional) The IPv6 CIDR block.
- `icmp_type` - (Optional) The ICMP type to be used. Default 0.
- `icmp_code` - (Optional) The ICMP type code to be used. Default 0.

Note: For more information on ICMP types and codes, see here: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the network ACL

- `owner_id` - The ID of the AWS account that owns the network ACL.

» Import

Network ACLs can be imported using the `id`, e.g.

```
$ terraform import aws_network_acl.main acl-7aaabd18
```

» Resource: aws_network_acl_rule

Creates an entry (a rule) in a network ACL with the specified rule number.

NOTE on Network ACLs and Network ACL Rules: Terraform currently provides both a standalone Network ACL Rule resource and a Network ACL resource with rules defined in-line. At this time you cannot use a Network ACL with in-line rules in conjunction with any Network ACL Rule resources. Doing so will cause a conflict of rule settings and will overwrite rules.

» Example Usage

```
resource "aws_network_acl" "bar" {
  vpc_id = "${aws_vpc.foo.id}"
}

resource "aws_network_acl_rule" "bar" {
  network_acl_id = "${aws_network_acl.bar.id}"
  rule_number    = 200
  egress         = false
  protocol       = "tcp"
  rule_action    = "allow"
  # Opening to 0.0.0.0/0 can lead to security vulnerabilities.
  cidr_block = # add a CIDR block here
  from_port      = 22
  to_port        = 22
}
```

Note: One of either `cidr_block` or `ipv6_cidr_block` is required.

» Argument Reference

The following arguments are supported:

- `network_acl_id` - (Required) The ID of the network ACL.

- `rule_number` - (Required) The rule number for the entry (for example, 100). ACL entries are processed in ascending order by rule number.
- `egress` - (Optional, bool) Indicates whether this is an egress rule (rule is applied to traffic leaving the subnet). Default `false`.
- `protocol` - (Required) The protocol. A value of -1 means all protocols.
- `rule_action` - (Required) Indicates whether to allow or deny the traffic that matches the rule. Accepted values: `allow` | `deny`
- `cidr_block` - (Optional) The network range to allow or deny, in CIDR notation (for example 172.16.0.0/24).
- `ipv6_cidr_block` - (Optional) The IPv6 CIDR block to allow or deny.
- `from_port` - (Optional) The from port to match.
- `to_port` - (Optional) The to port to match.
- `icmp_type` - (Optional) ICMP protocol: The ICMP type. Required if specifying ICMP for the protocol. e.g. -1
- `icmp_code` - (Optional) ICMP protocol: The ICMP code. Required if specifying ICMP for the protocol. e.g. -1

NOTE: If the value of `protocol` is `-1` or `all`, the `from_port` and `to_port` values will be ignored and the rule will apply to all ports.

NOTE: If the value of `icmp_type` is `-1` (which results in a wildcard ICMP type), the `icmp_code` must also be set to `-1` (wildcard ICMP code).

Note: For more information on ICMP types and codes, see here: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the network ACL Rule

» Resource: aws_network_interface

Provides an Elastic network interface (ENI) resource.

» Example Usage

```
resource "aws_network_interface" "test" {
  subnet_id       = "${aws_subnet.public_a.id}"
  private_ips     = ["10.0.0.50"]
  security_groups = ["${aws_security_group.web.id}"]

  attachment {
    instance     = "${aws_instance.test.id}"
    device_index = 1
  }
}
```

» Argument Reference

The following arguments are supported:

- `subnet_id` - (Required) Subnet ID to create the ENI in.
- `description` - (Optional) A description for the network interface.
- `private_ips` - (Optional) List of private IPs to assign to the ENI.
- `private_ips_count` - (Optional) Number of secondary private IPs to assign to the ENI. The total number of private IPs will be 1 + private_ips_count, as a primary private IP will be assiged to an ENI by default.
- `security_groups` - (Optional) List of security group IDs to assign to the ENI.
- `attachment` - (Optional) Block to define the attachment of the ENI. Documented below.
- `source_dest_check` - (Optional) Whether to enable source destination checking for the ENI. Default true.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `attachment` block supports:

- `instance` - (Required) ID of the instance to attach to.
- `device_index` - (Required) Integer to define the devices index.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the network interface.
- `subnet_id` - Subnet ID the ENI is in.
- `description` - A description for the network interface.
- `private_ips` - List of private IPs assigned to the ENI.
- `security_groups` - List of security groups attached to the ENI.
- `attachment` - Block defining the attachment of the ENI.
- `source_dest_check` - Whether source destination checking is enabled
- `tags` - Tags assigned to the ENI.

» Import

Network Interfaces can be imported using the `id`, e.g.

```
$ terraform import aws_network_interface.test eni-e5aa89a3
```

» Resource: aws_network_interface_attachment

Attach an Elastic network interface (ENI) resource with EC2 instance.

» Example Usage

```
resource "aws_network_interface_attachment" "test" {
  instance_id          = "${aws_instance.test.id}"
  network_interface_id = "${aws_network_interface.test.id}"
  device_index         = 0
}
```

» Argument Reference

The following arguments are supported:

- `instance_id` - (Required) Instance ID to attach.
- `network_interface_id` - (Required) ENI ID to attach.
- `device_index` - (Required) Network interface index (int).

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `instance_id` - Instance ID.
- `network_interface_id` - Network interface ID.
- `attachment_id` - The ENI Attachment ID.
- `status` - The status of the Network Interface Attachment.

» Resource: aws_route

Provides a resource to create a routing table entry (a route) in a VPC routing table.

NOTE on Route Tables and Routes: Terraform currently provides both a standalone Route resource and a Route Table resource with routes defined in-line. At this time you cannot use a Route Table with in-line routes in conjunction with any Route resources. Doing so will cause a conflict of rule settings and will overwrite rules.

» Example usage:

```
resource "aws_route" "r" {
  route_table_id            = "rtb-4fbb3ac4"
  destination_cidr_block    = "10.0.1.0/22"
  vpc_peering_connection_id = "pcx-45ff3dc1"
  depends_on                = ["aws_route_table.testing"]
}
```

» Example IPv6 Usage:

```
resource "aws_vpc" "vpc" {
  cidr_block                       = "10.1.0.0/16"
  assign_generated_ipv6_cidr_block = true
}

resource "aws_egress_only_internet_gateway" "egress" {
  vpc_id = "${aws_vpc.vpc.id}"
}

resource "aws_route" "r" {
  route_table_id              = "rtb-4fbb3ac4"
  destination_ipv6_cidr_block = "::/0"
  egress_only_gateway_id      = "${aws_egress_only_internet_gateway.egress.id}"
}
```

» Argument Reference

The following arguments are supported:

- `route_table_id` - (Required) The ID of the routing table.

One of the following destination arguments must be supplied:

- `destination_cidr_block` - (Optional) The destination CIDR block.
- `destination_ipv6_cidr_block` - (Optional) The destination IPv6 CIDR block.

One of the following target arguments must be supplied:

- `egress_only_gateway_id` - (Optional) Identifier of a VPC Egress Only Internet Gateway.
- `gateway_id` - (Optional) Identifier of a VPC internet gateway or a virtual private gateway.
- `instance_id` - (Optional) Identifier of an EC2 instance.
- `nat_gateway_id` - (Optional) Identifier of a VPC NAT gateway.

- `network_interface_id` - (Optional) Identifier of an EC2 network interface.
- `transit_gateway_id` - (Optional) Identifier of an EC2 Transit Gateway.
- `vpc_peering_connection_id` - (Optional) Identifier of a VPC peering connection.

Note that the default route, mapping the VPC's CIDR block to "local", is created implicitly and cannot be specified.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

NOTE: Only the arguments that are configured (one of the above) will be exported as an attribute once the resource is created.

- `id` - Route Table identifier and destination

» Timeouts

`aws_route` provides the following Timeouts configuration options:

- `create` - (Default `2 minutes`) Used for route creation
- `delete` - (Default `5 minutes`) Used for route deletion

» Import

Individual routes can be imported using `ROUTETABLEID_DESTINATION`.

For example, import a route in route table `rtb-656C65616E6F72` with an IPv4 destination CIDR of `10.42.0.0/16` like this:

```
$ terraform import aws_route.my_route rtb-656C65616E6F72_10.42.0.0/16
```

Import a route in route table `rtb-656C65616E6F72` with an IPv6 destination CIDR of `2620:0:2d0:200::8/125` similarly:

```
$ terraform import aws_route.my_route rtb-656C65616E6F72_2620:0:2d0:200::8/125
```

» Resource: aws_route_table

Provides a resource to create a VPC routing table.

NOTE on Route Tables and Routes: Terraform currently provides both a standalone Route resource and a Route Table resource with routes defined in-line. At this time you cannot use a Route Table with in-line routes in conjunction with any Route resources. Doing so will cause a conflict of rule settings and will overwrite rules.

NOTE on `gateway_id` and `nat_gateway_id`: The AWS API is very forgiving with these two attributes and the `aws_route_table` resource can be created with a NAT ID specified as a Gateway ID attribute. This *will* lead to a permanent diff between your configuration and statefile, as the API returns the correct parameters in the returned route table. If you're experiencing constant diffs in your `aws_route_table` resources, the first thing to check is whether or not you're specifying a NAT ID instead of a Gateway ID, or vice-versa.

NOTE on `propagating_vgws` and the `aws_vpn_gateway_route_propagation` resource: If the `propagating_vgws` argument is present, it's not supported to *also* define route propagations using `aws_vpn_gateway_route_propagation`, since this resource will delete any propagating gateways not explicitly listed in `propagating_vgws`. Omit this argument when defining route propagation using the separate resource.

» Example usage with tags:

```
resource "aws_route_table" "r" {
  vpc_id = "${aws_vpc.default.id}"

  route {
    cidr_block = "10.0.1.0/24"
    gateway_id = "${aws_internet_gateway.main.id}"
  }

  route {
    ipv6_cidr_block        = "::/0"
    egress_only_gateway_id = "${aws_egress_only_internet_gateway.foo.id}"
  }

  tags = {
    Name = "main"
  }
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Required) The VPC ID.
- `route` - (Optional) A list of route objects. Their keys are documented below. This argument is processed in attribute-as-blocks mode.

- `tags` - (Optional) A mapping of tags to assign to the resource.
- `propagating_vgws` - (Optional) A list of virtual gateways for propagation.

» route Argument Reference

One of the following destination arguments must be supplied:

- `cidr_block` - (Required) The CIDR block of the route.
- `ipv6_cidr_block` - (Optional) The Ipv6 CIDR block of the route.

One of the following target arguments must be supplied:

- `egress_only_gateway_id` - (Optional) Identifier of a VPC Egress Only Internet Gateway.
- `gateway_id` - (Optional) Identifier of a VPC internet gateway or a virtual private gateway.
- `instance_id` - (Optional) Identifier of an EC2 instance.
- `nat_gateway_id` - (Optional) Identifier of a VPC NAT gateway.
- `network_interface_id` - (Optional) Identifier of an EC2 network interface.
- `transit_gateway_id` - (Optional) Identifier of an EC2 Transit Gateway.
- `vpc_peering_connection_id` - (Optional) Identifier of a VPC peering connection.

Note that the default route, mapping the VPC's CIDR block to ”local”, is created implicitly and cannot be specified.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

NOTE: Only the target that is entered is exported as a readable attribute once the route resource is created.

- `id` - The ID of the routing table.
- `owner_id` - The ID of the AWS account that owns the route table.

» Import

Route Tables can be imported using the route table `id`. For example, to import route table `rtb-4e616f6d69`, use this command:

```
$ terraform import aws_route_table.public_rt rtb-4e616f6d69
```

» Resource: aws_route_table_association

Provides a resource to create an association between a route table and a subnet or a route table and an internet gateway or virtual private gateway.

» Example Usage

```
resource "aws_route_table_association" "a" {
  subnet_id      = aws_subnet.foo.id
  route_table_id = aws_route_table.bar.id
}

resource "aws_route_table_association" "b" {
  gateway_id     = aws_internet_gateway.foo.id
  route_table_id = aws_route_table.bar.id
}
```

» Argument Reference

NOTE: Please note that one of either `subnet_id` or `gateway_id` is required.

The following arguments are supported:

- `subnet_id` - (Optional) The subnet ID to create an association. Conflicts with `gateway_id`.
- `gateway_id` - (Optional) The gateway ID to create an association. Conflicts with `subnet_id`.
- `route_table_id` - (Required) The ID of the routing table to associate with.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the association

» Import

NOTE: Attempting to associate a route table with a subnet or gateway, where either is already associated, will result in an error (e.g., `Resource.AlreadyAssociated: the specified association for route table rtb-4176657279 conflicts with an existing association`) unless you first import the original association.

EC2 Route Table Associations can be imported using the sassociated resource ID and Route Table ID separated by a forward slash (`/`).

For example with EC2 Subnets:

```
$ terraform import aws_route_table_association.assoc subnet-6777656e646f6c796e/rtb-656c65616
```

For example with EC2 Internet Gateways:

```
$ terraform import aws_route_table_association.assoc igw-01b3a60780f8d034a/rtb-656c65616e6f7
```

» Resource: aws_security_group

Provides a security group resource.

NOTE on Security Groups and Security Group Rules: Terraform currently provides both a standalone Security Group Rule resource (a single `ingress` or `egress` rule), and a Security Group resource with `ingress` and `egress` rules defined in-line. At this time you cannot use a Security Group with in-line rules in conjunction with any Security Group Rule resources. Doing so will cause a conflict of rule settings and will overwrite rules.

NOTE: Referencing Security Groups across VPC peering has certain restrictions. More information is available in the VPC Peering User Guide.

NOTE: Due to AWS Lambda improved VPC networking changes that began deploying in September 2019, security groups associated with Lambda Functions can take up to 45 minutes to successfully delete. Terraform AWS Provider version 2.31.0 and later automatically handles this increased timeout, however prior versions require setting the customizable deletion timeout to 45 minutes (`delete = "45m"`). AWS and HashiCorp are working together to reduce the amount of time required for resource deletion and updates can be tracked in this GitHub issue.

» Example Usage

Basic usage

```
resource "aws_security_group" "allow_tls" {
  name        = "allow_tls"
  description = "Allow TLS inbound traffic"
  vpc_id      = "${aws_vpc.main.id}"

  ingress {
    # TLS (change to whatever ports you need)
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    # Please restrict your ingress to only necessary IPs and ports.
    # Opening to 0.0.0.0/0 can lead to security vulnerabilities.
    cidr_blocks = # add a CIDR block here
  }

  egress {
    from_port       = 0
    to_port         = 0
    protocol        = "-1"
    cidr_blocks     = ["0.0.0.0/0"]
    prefix_list_ids = ["pl-12c4e678"]
  }
}
```

Basic usage with tags:

```
resource "aws_security_group" "allow_tls" {
  name        = "allow_tls"
  description = "Allow TLS inbound traffic"
  vpc_id      = "${aws_vpc.main.id}"

  ingress {
    # TLS (change to whatever ports you need)
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    # Please restrict your ingress to only necessary IPs and ports.
    # Opening to 0.0.0.0/0 can lead to security vulnerabilities.
    cidr_blocks = # add your IP address here
  }

  tags = {
    Name = "allow_all"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional, Forces new resource) The name of the security group. If omitted, Terraform will assign a random, unique name
- `name_prefix` - (Optional, Forces new resource) Creates a unique name beginning with the specified prefix. Conflicts with `name`.

- `description` - (Optional, Forces new resource) The security group description. Defaults to "Managed by Terraform". Cannot be "". **NOTE**: This field maps to the AWS `GroupDescription` attribute, for which there is no Update API. If you'd like to classify your security groups in a way that can be updated, use `tags`.
- `ingress` - (Optional) Can be specified multiple times for each ingress rule. Each ingress block supports fields documented below. This argument is processed in attribute-as-blocks mode.
- `egress` - (Optional, VPC only) Can be specified multiple times for each egress rule. Each egress block supports fields documented below. This argument is processed in attribute-as-blocks mode.
- `revoke_rules_on_delete` - (Optional) Instruct Terraform to revoke all of the Security Groups attached ingress and egress rules before deleting the rule itself. This is normally not needed, however certain AWS services such as Elastic Map Reduce may automatically add required rules to security groups used with the service, and those rules may contain a cyclic dependency that prevent the security groups from being destroyed without removing the dependency first. Default `false`
- `vpc_id` - (Optional, Forces new resource) The VPC ID.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `ingress` block supports:

- `cidr_blocks` - (Optional) List of CIDR blocks.
- `ipv6_cidr_blocks` - (Optional) List of IPv6 CIDR blocks.
- `prefix_list_ids` - (Optional) List of prefix list IDs.
- `from_port` - (Required) The start port (or ICMP type number if protocol is "icmp")
- `protocol` - (Required) The protocol. If you select a protocol of "-1" (semantically equivalent to `"all"`, which is not a valid value here), you must specify a "from_port" and "to_port" equal to 0. If not icmp, tcp, udp, or "-1" use the protocol number
- `security_groups` - (Optional) List of security group Group Names if using EC2-Classic, or Group IDs if using a VPC.
- `self` - (Optional) If true, the security group itself will be added as a source to this ingress rule.
- `to_port` - (Required) The end range port (or ICMP code if protocol is "icmp").
- `description` - (Optional) Description of this ingress rule.

The `egress` block supports:

- `cidr_blocks` - (Optional) List of CIDR blocks.
- `ipv6_cidr_blocks` - (Optional) List of IPv6 CIDR blocks.
- `prefix_list_ids` - (Optional) List of prefix list IDs (for allowing access to VPC endpoints)
- `from_port` - (Required) The start port (or ICMP type number if protocol is "icmp")

- `protocol` - (Required) The protocol. If you select a protocol of "-1" (semantically equivalent to `"all"`, which is not a valid value here), you must specify a "from_port" and "to_port" equal to 0. If not icmp, tcp, udp, or "-1" use the protocol number
- `security_groups` - (Optional) List of security group Group Names if using EC2-Classic, or Group IDs if using a VPC.
- `self` - (Optional) If true, the security group itself will be added as a source to this egress rule.
- `to_port` - (Required) The end range port (or ICMP code if protocol is "icmp").
- `description` - (Optional) Description of this egress rule.

NOTE on Egress rules: By default, AWS creates an `ALLOW ALL` egress rule when creating a new Security Group inside of a VPC. When creating a new Security Group inside a VPC, **Terraform will remove this default rule**, and require you specifically re-create it if you desire that rule. We feel this leads to fewer surprises in terms of controlling your egress rules. If you desire this rule to be in place, you can use this `egress` block:

```
egress {
  from_port   = 0
  to_port     = 0
  protocol    = "-1"
  cidr_blocks = ["0.0.0.0/0"]
}
```

» Usage with prefix list IDs

Prefix list IDs are managed by AWS internally. Prefix list IDs are associated with a prefix list name, or service name, that is linked to a specific region. Prefix list IDs are exported on VPC Endpoints, so you can use this format:

```
# ...
egress {
  from_port       = 0
  to_port         = 0
  protocol        = "-1"
  prefix_list_ids = ["${aws_vpc_endpoint.my_endpoint.prefix_list_id}"]
}

# ...
resource "aws_vpc_endpoint" "my_endpoint" {
  # ...
}
```

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the security group
- `arn` - The ARN of the security group
- `vpc_id` - The VPC ID.
- `owner_id` - The owner ID.
- `name` - The name of the security group
- `description` - The description of the security group
- `ingress` - The ingress rules. See above for more.
- `egress` - The egress rules. See above for more.

» Timeouts

`aws_security_group` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for a security group to be created.
- `delete` - (Default `10m`) How long to retry on `DependencyViolation` errors during security group deletion from lingering ENIs left by certain AWS services such as Elastic Load Balancing. NOTE: Lambda ENIs can take up to 45 minutes to delete, which is not affected by changing this customizable timeout (in version 2.31.0 and later of the Terraform AWS Provider) unless it is increased above 45 minutes.

» Import

Security Groups can be imported using the `security group id`, e.g.

```
$ terraform import aws_security_group.elb_sg sg-903004f8
```

» Resource: aws_network_interface_sg_attachment

This resource attaches a security group to an Elastic Network Interface (ENI). It can be used to attach a security group to any existing ENI, be it a secondary ENI or one attached as the primary interface on an instance.

NOTE on instances, interfaces, and security groups: Terraform currently provides the capability to assign security groups via the `aws_instance` and the `aws_network_interface` resources. Using this resource in conjunction with security groups provided in-line in those resources will cause conflicts, and will lead to spurious diffs and undefined behavior - please use one or the other.

» Example Usage

The following provides a very basic example of setting up an instance (provided by instance) in the default security group, creating a security group (provided by sg) and then attaching the security group to the instance's primary network interface via the aws_network_interface_sg_attachment resource, named sg_attachment:

```
data "aws_ami" "ami" {
  most_recent = true

  filter {
    name   = "name"
    values = ["amzn-ami-hvm-*"]
  }

  owners = ["amazon"]
}

resource "aws_instance" "instance" {
  instance_type = "t2.micro"
  ami           = "${data.aws_ami.ami.id}"

  tags = {
    "type" = "terraform-test-instance"
  }
}

resource "aws_security_group" "sg" {
  tags = {
    "type" = "terraform-test-security-group"
  }
}

resource "aws_network_interface_sg_attachment" "sg_attachment" {
  security_group_id    = "${aws_security_group.sg.id}"
  network_interface_id = "${aws_instance.instance.primary_network_interface_id}"
}
```

In this example, instance is provided by the aws_instance data source, fetching an external instance, possibly not managed by Terraform. sg_attachment then attaches to the output instance's network_interface_id:

```
data "aws_instance" "instance" {
  instance_id = "i-1234567890abcdef0"
}
```

```
resource "aws_security_group" "sg" {
  tags = {
    "type" = "terraform-test-security-group"
  }
}

resource "aws_network_interface_sg_attachment" "sg_attachment" {
  security_group_id    = "${aws_security_group.sg.id}"
  network_interface_id = "${data.aws_instance.instance.network_interface_id}"
}
```

» Argument Reference

- `security_group_id` - (Required) The ID of the security group.
- `network_interface_id` - (Required) The ID of the network interface to attach to.

» Output Reference

There are no outputs for this resource.

» Resource: aws_security_group_rule

Provides a security group rule resource. Represents a single `ingress` or `egress` group rule, which can be added to external Security Groups.

NOTE on Security Groups and Security Group Rules: Terraform currently provides both a standalone Security Group Rule resource (a single `ingress` or `egress` rule), and a Security Group resource with `ingress` and `egress` rules defined in-line. At this time you cannot use a Security Group with in-line rules in conjunction with any Security Group Rule resources. Doing so will cause a conflict of rule settings and will overwrite rules.

NOTE: Setting `protocol = "all"` or `protocol = -1` with `from_port` and `to_port` will result in the EC2 API creating a security group rule with all ports open. This API behavior cannot be controlled by Terraform and may generate warnings in the future.

NOTE: Referencing Security Groups across VPC peering has certain restrictions. More information is available in the VPC Peering User Guide.

» Example Usage

Basic usage

```
resource "aws_security_group_rule" "allow_all" {
  type            = "ingress"
  from_port       = 0
  to_port         = 65535
  protocol        = "tcp"
  # Opening to 0.0.0.0/0 can lead to security vulnerabilities.
  cidr_blocks = # add a CIDR block here
  prefix_list_ids = ["pl-12c4e678"]

  security_group_id = "sg-123456"
}
```

» Argument Reference

The following arguments are supported:

- `type` - (Required) The type of rule being created. Valid options are `ingress` (inbound) or `egress` (outbound).
- `cidr_blocks` - (Optional) List of CIDR blocks. Cannot be specified with `source_security_group_id`.
- `ipv6_cidr_blocks` - (Optional) List of IPv6 CIDR blocks.
- `prefix_list_ids` - (Optional) List of prefix list IDs (for allowing access to VPC endpoints).
- `from_port` - (Required) The start port (or ICMP type number if protocol is ”icmp”).
- `protocol` - (Required) The protocol. If not icmp, tcp, udp, or all use the protocol number
- `security_group_id` - (Required) The security group to apply this rule to.
- `source_security_group_id` - (Optional) The security group id to allow access to/from, depending on the `type`. Cannot be specified with `cidr_blocks` and `self`.
- `self` - (Optional) If true, the security group itself will be added as a source to this ingress rule. Cannot be specified with `source_security_group_id`.
- `to_port` - (Required) The end port (or ICMP code if protocol is ”icmp”).
- `description` - (Optional) Description of the rule.

» Usage with prefix list IDs

Prefix list IDs are manged by AWS internally. Prefix list IDs are associated with a prefix list name, or service name, that is linked to a specific region. Prefix list

IDs are exported on VPC Endpoints, so you can use this format:

```
resource "aws_security_group_rule" "allow_all" {
  type              = "egress"
  to_port           = 0
  protocol          = "-1"
  prefix_list_ids   = ["${aws_vpc_endpoint.my_endpoint.prefix_list_id}"]
  from_port         = 0
  security_group_id = "sg-123456"
}

# ...
resource "aws_vpc_endpoint" "my_endpoint" {
  # ...
}
```

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the security group rule
- `type` - The type of rule, `ingress` or `egress`
- `from_port` - The start port (or ICMP type number if protocol is ”icmp”)
- `to_port` - The end port (or ICMP code if protocol is ”icmp”)
- `protocol` – The protocol used
- `description` – Description of the rule

» Import

Security Group Rules can be imported using the `security_group_id`, `type`, `protocol`, `from_port`, `to_port`, and source(s)/destination(s) (e.g. `cidr_block`) separated by underscores (`_`). All parts are required.

Not all rule permissions (e.g., not all of a rule’s CIDR blocks) need to be imported for Terraform to manage rule permissions. However, importing some of a rule’s permissions but not others, and then making changes to the rule will result in the creation of an additional rule to capture the updated permissions. Rule permissions that were not imported are left intact in the original rule.

» Examples

Import an ingress rule in security group `sg-6e616f6d69` for TCP port 8000 with an IPv4 destination CIDR of `10.0.3.0/24`:

```
$ terraform import aws_security_group_rule.ingress sg-6e616f6d69_ingress_tcp_8000_8000_10.0.
```

Import a rule with various IPv4 and IPv6 source CIDR blocks:

```
$ terraform import aws_security_group_rule.ingress sg-4973616163_ingress_tcp_100_121_10.1.0.
```

Import a rule, applicable to all ports, with a protocol other than TCP/UDP/ICMP/ALL, e.g., Multicast Transport Protocol (MTP), using the IANA protocol number, e.g., 92.

```
$ terraform import aws_security_group_rule.ingress sg-6777656e646f6c796e_ingress_92_0_65536_
```

Import an egress rule with a prefix list ID destination:

```
$ terraform import aws_security_group_rule.egress sg-62726f6479_egress_tcp_8000_8000_pl-646
```

Import a rule applicable to all protocols and ports with a security group source:

```
$ terraform import aws_security_group_rule.ingress_rule sg-7472697374616e_ingress_all_0_655
```

Import a rule that has itself and an IPv6 CIDR block as sources:

```
$ terraform import aws_security_group_rule.rule_name sg-656c65616e6f72_ingress_tcp_80_80_se
```

» Resource: aws_subnet

Provides an VPC subnet resource.

NOTE: Due to AWS Lambda improved VPC networking changes that began deploying in September 2019, subnets associated with Lambda Functions can take up to 45 minutes to successfully delete. Terraform AWS Provider version 2.31.0 and later automatically handles this increased timeout, however prior versions require setting the customizable deletion timeout to 45 minutes (`delete = "45m"`). AWS and HashiCorp are working together to reduce the amount of time required for resource deletion and updates can be tracked in this GitHub issue.

» Example Usage

» Basic Usage

```
resource "aws_subnet" "main" {
  vpc_id     = "${aws_vpc.main.id}"
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "Main"
  }
}
```

» Subnets In Secondary VPC CIDR Blocks

When managing subnets in one of a VPC's secondary CIDR blocks created using a `aws_vpc_ipv4_cidr_block_association` resource, it is recommended to reference that resource's `vpc_id` attribute to ensure correct dependency ordering.

```
resource "aws_vpc_ipv4_cidr_block_association" "secondary_cidr" {
  vpc_id     = "${aws_vpc.main.id}"
  cidr_block = "172.2.0.0/16"
}

resource "aws_subnet" "in_secondary_cidr" {
  vpc_id     = "${aws_vpc_ipv4_cidr_block_association.secondary_cidr.vpc_id}"
  cidr_block = "172.2.0.0/24"
}
```

» Argument Reference

The following arguments are supported:

- `availability_zone` - (Optional) The AZ for the subnet.
- `availability_zone_id` - (Optional) The AZ ID of the subnet.
- `cidr_block` - (Required) The CIDR block for the subnet.
- `ipv6_cidr_block` - (Optional) The IPv6 network range for the subnet, in CIDR notation. The subnet size must use a /64 prefix length.
- `map_public_ip_on_launch` - (Optional) Specify true to indicate that instances launched into the subnet should be assigned a public IP address. Default is `false`.
- `assign_ipv6_address_on_creation` - (Optional) Specify true to indicate that network interfaces created in the specified subnet should be assigned an IPv6 address. Default is `false`
- `vpc_id` - (Required) The VPC ID.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the subnet
- `arn` - The ARN of the subnet.
- `ipv6_cidr_block_association_id` - The association ID for the IPv6 CIDR block.
- `owner_id` - The ID of the AWS account that owns the subnet.

» Timeouts

`aws_subnet` provides the following Timeouts configuration options:

- `create` - (Default `10m`) How long to wait for a subnet to be created.
- `delete` - (Default `20m`) How long to retry on `DependencyViolation` errors during subnet deletion from lingering ENIs left by certain AWS services such as Elastic Load Balancing. NOTE: Lambda ENIs can take up to 45 minutes to delete, which is not affected by changing this customizable timeout (in version 2.31.0 and later of the Terraform AWS Provider) unless it is increased above 45 minutes.

» Import

Subnets can be imported using the `subnet id`, e.g.

```
$ terraform import aws_subnet.public_subnet subnet-9d4a7b6c
```

» Resource: aws_vpc

Provides a VPC resource.

» Example Usage

Basic usage:

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}
```

Basic usage with tags:

```
resource "aws_vpc" "main" {
  cidr_block       = "10.0.0.0/16"
  instance_tenancy = "dedicated"

  tags = {
    Name = "main"
  }
}
```

» Argument Reference

The following arguments are supported:

- `cidr_block` - (Required) The CIDR block for the VPC.
- `instance_tenancy` - (Optional) A tenancy option for instances launched into the VPC
- `enable_dns_support` - (Optional) A boolean flag to enable/disable DNS support in the VPC. Defaults true.
- `enable_dns_hostnames` - (Optional) A boolean flag to enable/disable DNS hostnames in the VPC. Defaults false.
- `enable_classiclink` - (Optional) A boolean flag to enable/disable ClassicLink for the VPC. Only valid in regions and accounts that support EC2 Classic. See the ClassicLink documentation for more information. Defaults false.
- `enable_classiclink_dns_support` - (Optional) A boolean flag to enable/disable ClassicLink DNS Support for the VPC. Only valid in regions and accounts that support EC2 Classic.
- `assign_generated_ipv6_cidr_block` - (Optional) Requests an Amazon-provided IPv6 CIDR block with a /56 prefix length for the VPC. You cannot specify the range of IP addresses, or the size of the CIDR block. Default is `false`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of VPC
- `id` - The ID of the VPC
- `cidr_block` - The CIDR block of the VPC
- `instance_tenancy` - Tenancy of instances spin up within VPC.
- `enable_dns_support` - Whether or not the VPC has DNS support
- `enable_dns_hostnames` - Whether or not the VPC has DNS hostname support
- `enable_classiclink` - Whether or not the VPC has Classiclink enabled
- `main_route_table_id` - The ID of the main route table associated with this VPC. Note that you can change a VPC's main route table by using an `aws_main_route_table_association`.
- `default_network_acl_id` - The ID of the network ACL created by default on VPC creation
- `default_security_group_id` - The ID of the security group created by default on VPC creation
- `default_route_table_id` - The ID of the route table created by default on VPC creation
- `ipv6_association_id` - The association ID for the IPv6 CIDR block.
- `ipv6_cidr_block` - The IPv6 CIDR block.
- `owner_id` - The ID of the AWS account that owns the VPC.

» Import

VPCs can be imported using the `vpc id`, e.g.

```
$ terraform import aws_vpc.test_vpc vpc-a01106c2
```

» Resource: aws_vpc_dhcp_options

Provides a VPC DHCP Options resource.

» Example Usage

Basic usage:

```
resource "aws_vpc_dhcp_options" "dns_resolver" {
  domain_name_servers = ["8.8.8.8", "8.8.4.4"]
}
```

Full usage:

```
resource "aws_vpc_dhcp_options" "foo" {
  domain_name          = "service.consul"
  domain_name_servers  = ["127.0.0.1", "10.0.0.2"]
  ntp_servers          = ["127.0.0.1"]
  netbios_name_servers = ["127.0.0.1"]
  netbios_node_type    = 2

  tags = {
    Name = "foo-name"
  }
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Optional) the suffix domain name to use by default when resolving non Fully Qualified Domain Names. In other words, this is what ends up being the `search` value in the `/etc/resolv.conf` file.
- `domain_name_servers` - (Optional) List of name servers to configure in `/etc/resolv.conf`. If you want to use the default AWS nameservers you should set this to `AmazonProvidedDNS`.
- `ntp_servers` - (Optional) List of NTP servers to configure.
- `netbios_name_servers` - (Optional) List of NETBIOS name servers.

- `netbios_node_type` - (Optional) The NetBIOS node type (1, 2, 4, or 8). AWS recommends to specify 2 since broadcast and multicast are not supported in their network. For more information about these node types, see RFC 2132.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Remarks

- Notice that all arguments are optional but you have to specify at least one argument.
- `domain_name_servers`, `netbios_name_servers`, `ntp_servers` are limited by AWS to maximum four servers only.
- To actually use the DHCP Options Set you need to associate it to a VPC using `aws_vpc_dhcp_options_association`.
- If you delete a DHCP Options Set, all VPCs using it will be associated to AWS's `default` DHCP Option Set.
- In most cases unless you're configuring your own DNS you'll want to set `domain_name_servers` to `AmazonProvidedDNS`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the DHCP Options Set.
- `owner_id` - The ID of the AWS account that owns the DHCP options set.

You can find more technical documentation about DHCP Options Set in the official AWS User Guide.

» Import

VPC DHCP Options can be imported using the `dhcp options id`, e.g.

```
$ terraform import aws_vpc_dhcp_options.my_options dopt-d9070ebb
```

» Resource: aws_vpc_dhcp_options_association

Provides a VPC DHCP Options Association resource.

» Example Usage

```
resource "aws_vpc_dhcp_options_association" "dns_resolver" {
  vpc_id          = "${aws_vpc.foo.id}"
  dhcp_options_id = "${aws_vpc_dhcp_options.foo.id}"
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Required) The ID of the VPC to which we would like to associate a DHCP Options Set.
- `dhcp_options_id` - (Required) The ID of the DHCP Options Set to associate to the VPC.

» Remarks

- You can only associate one DHCP Options Set to a given VPC ID.
- Removing the DHCP Options Association automatically sets AWS's `default` DHCP Options Set to the VPC.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the DHCP Options Set Association.

» Resource: aws_vpc_endpoint

Provides a VPC Endpoint resource.

NOTE on VPC Endpoints and VPC Endpoint Associations: Terraform provides both standalone VPC Endpoint Associations for Route Tables - (an association between a VPC endpoint and a single `route_table_id`) and Subnets - (an association between a VPC endpoint and a single `subnet_id`) and a VPC Endpoint resource with `route_table_ids` and `subnet_ids` attributes. Do not use the same resource ID in both a VPC Endpoint resource and a VPC Endpoint Association resource. Doing so will cause a conflict of associations and will overwrite the association.

» Example Usage

» Basic

```
resource "aws_vpc_endpoint" "s3" {
  vpc_id       = "${aws_vpc.main.id}"
  service_name = "com.amazonaws.us-west-2.s3"
}
```

» Basic w/ Tags

```
resource "aws_vpc_endpoint" "s3" {
  vpc_id       = "${aws_vpc.main.id}"
  service_name = "com.amazonaws.us-west-2.s3"

  tags = {
    Environment = "test"
  }
}
```

» Interface Endpoint Type

```
resource "aws_vpc_endpoint" "ec2" {
  vpc_id            = "${aws_vpc.main.id}"
  service_name      = "com.amazonaws.us-west-2.ec2"
  vpc_endpoint_type = "Interface"

  security_group_ids = [
    "${aws_security_group.sg1.id}",
  ]

  private_dns_enabled = true
}
```

» Custom Service

```
resource "aws_vpc_endpoint" "ptfe_service" {
  vpc_id            = "${var.vpc_id}"
  service_name      = "${var.ptfe_service}"
  vpc_endpoint_type = "Interface"

  security_group_ids = [
    "${aws_security_group.ptfe_service.id}",
  ]

  subnet_ids          = ["${local.subnet_ids}"]
  private_dns_enabled = false
}

data "aws_route53_zone" "internal" {
  name         = "vpc.internal."
  private_zone = true
  vpc_id       = "${var.vpc_id}"
}

resource "aws_route53_record" "ptfe_service" {
  zone_id = "${data.aws_route53_zone.internal.zone_id}"
  name    = "ptfe.${data.aws_route53_zone.internal.name}"
  type    = "CNAME"
  ttl     = "300"
  records = ["${lookup(aws_vpc_endpoint.ptfe_service.dns_entry[0], "dns_name")}"]
}
```

NOTE The `dns_entry` output is a list of maps: Terraform interpolation support for lists of maps requires the `lookup` and `[]` until full support of lists of maps is available

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The service name, in the form `com.amazonaws.region.service` for AWS services.
- `vpc_id` - (Required) The ID of the VPC in which the endpoint will be used.
- `auto_accept` - (Optional) Accept the VPC endpoint (the VPC endpoint and service need to be in the same AWS account).
- `policy` - (Optional) A policy to attach to the endpoint that controls access to the service. Defaults to full access. All `Gateway` and some `Interface` endpoints support policies - see the relevant AWS documentation for more details. For more information about building AWS IAM policy documents with Terraform, see the AWS IAM Policy Document Guide.
- `private_dns_enabled` - (Optional; AWS services and AWS Marketplace partner services only) Whether or not to associate a private hosted zone with the specified VPC. Applicable for endpoints of type `Interface`. Defaults to `false`.
- `route_table_ids` - (Optional) One or more route table IDs. Applicable for endpoints of type `Gateway`.

- `subnet_ids` - (Optional) The ID of one or more subnets in which to create a network interface for the endpoint. Applicable for endpoints of type `Interface`.
- `security_group_ids` - (Optional) The ID of one or more security groups to associate with the network interface. Required for endpoints of type `Interface`.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `vpc_endpoint_type` - (Optional) The VPC endpoint type, `Gateway` or `Interface`. Defaults to `Gateway`.

» Timeouts

`aws_vpc_endpoint` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating a VPC endpoint
- `update` - (Default `10 minutes`) Used for VPC endpoint modifications
- `delete` - (Default `10 minutes`) Used for destroying VPC endpoints

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPC endpoint.
- `cidr_blocks` - The list of CIDR blocks for the exposed AWS service. Applicable for endpoints of type `Gateway`.
- `dns_entry` - The DNS entries for the VPC Endpoint. Applicable for endpoints of type `Interface`. DNS blocks are documented below.
- `network_interface_ids` - One or more network interfaces for the VPC Endpoint. Applicable for endpoints of type `Interface`.
- `owner_id` - The ID of the AWS account that owns the VPC endpoint.
- `prefix_list_id` - The prefix list ID of the exposed AWS service. Applicable for endpoints of type `Gateway`.
- `requester_managed` - Whether or not the VPC Endpoint is being managed by its service - `true` or `false`.
- `state` - The state of the VPC endpoint.

DNS blocks (for `dns_entry`) support the following attributes:

- `dns_name` - The DNS name.
- `hosted_zone_id` - The ID of the private hosted zone.

» Import

VPC Endpoints can be imported using the `vpc endpoint id`, e.g.

```
$ terraform import aws_vpc_endpoint.endpoint1 vpce-3ecf2a57
```

» Resource: aws_vpc_endpoint_connection_notification

Provides a VPC Endpoint connection notification resource. Connection notifications notify subscribers of VPC Endpoint events.

» Example Usage

```
resource "aws_sns_topic" "topic" {
  name = "vpce-notification-topic"

  policy = <<POLICY
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect": "Allow",
        "Principal": {
            "Service": "vpce.amazonaws.com"
        },
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:*:*:vpce-notification-topic"
    }]
}
POLICY
}

resource "aws_vpc_endpoint_service" "foo" {
  acceptance_required        = false
  network_load_balancer_arns = ["${aws_lb.test.arn}"]
}

resource "aws_vpc_endpoint_connection_notification" "foo" {
  vpc_endpoint_service_id     = "${aws_vpc_endpoint_service.foo.id}"
  connection_notification_arn = "${aws_sns_topic.topic.arn}"
  connection_events           = ["Accept", "Reject"]
}
```

» Argument Reference

The following arguments are supported:

- `vpc_endpoint_service_id` - (Optional) The ID of the VPC Endpoint Service to receive notifications for.
- `vpc_endpoint_id` - (Optional) The ID of the VPC Endpoint to receive notifications for.

- `connection_notification_arn` - (Required) The ARN of the SNS topic for the notifications.
- `connection_events` - (Required) One or more endpoint events for which to receive notifications.

NOTE: One of `vpc_endpoint_service_id` or `vpc_endpoint_id` must be specified.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPC connection notification.
- `state` - The state of the notification.
- `notification_type` - The type of notification.

» Import

VPC Endpoint connection notifications can be imported using the `VPC endpoint connection notification id`, e.g.

```
$ terraform import aws_vpc_endpoint_connection_notification.foo vpce-nfn-09e6ed3b4efba2263
```

» Resource: aws_vpc_endpoint_route_table_association

Manages a VPC Endpoint Route Table Association

» Example Usage

```
resource "aws_vpc_endpoint_route_table_association" "example" {
  route_table_id  = "${aws_route_table.example.id}"
  vpc_endpoint_id = "${aws_vpc_endpoint.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- `route_table_id` - (Required) Identifier of the EC2 Route Table to be associated with the VPC Endpoint.
- `vpc_endpoint_id` - (Required) Identifier of the VPC Endpoint with which the EC2 Route Table will be associated.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - A hash of the EC2 Route Table and VPC Endpoint identifiers.

» Import

VPC Endpoint Route Table Associations can be imported using `vpc_endpoint_id` together with `route_table_id`, e.g.

```
$ terraform import aws_vpc_endpoint_route_table_association.example vpce-aaaaaaaa/rt-bbbbbbb
```

» Resource: aws_vpc_endpoint_service

Provides a VPC Endpoint Service resource. Service consumers can create an *Interface* VPC Endpoint to connect to the service.

NOTE on VPC Endpoint Services and VPC Endpoint Service Allowed Principals: Terraform provides both a standalone VPC Endpoint Service Allowed Principal resource and a VPC Endpoint Service resource with an `allowed_principals` attribute. Do not use the same principal ARN in both a VPC Endpoint Service resource and a VPC Endpoint Service Allowed Principal resource. Doing so will cause a conflict and will overwrite the association.

» Example Usage

» Basic

```
resource "aws_vpc_endpoint_service" "example" {
  acceptance_required        = false
  network_load_balancer_arns = ["${aws_lb.example.arn}"]
}
```

» Basic w/ Tags

```
resource "aws_vpc_endpoint_service" "example" {
  acceptance_required        = false
  network_load_balancer_arns = ["${aws_lb.example.arn}"]

  tags = {
    Environment = "test"
  }
}
```

» Argument Reference

The following arguments are supported:

- `acceptance_required` - (Required) Whether or not VPC endpoint connection requests to the service must be accepted by the service owner - `true` or `false`.
- `network_load_balancer_arns` - (Required) The ARNs of one or more Network Load Balancers for the endpoint service.
- `allowed_principals` - (Optional) The ARNs of one or more principals allowed to discover the endpoint service.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPC endpoint service.
- `availability_zones` - The Availability Zones in which the service is available.
- `base_endpoint_dns_names` - The DNS names for the service.
- `manages_vpc_endpoints` - Whether or not the service manages its VPC endpoints - `true` or `false`.
- `private_dns_name` - The private DNS name for the service.
- `service_name` - The service name.
- `service_type` - The service type, `Gateway` or `Interface`.
- `state` - The state of the VPC endpoint service.

» Import

VPC Endpoint Services can be imported using the `VPC endpoint service id`, e.g.

```
$ terraform import aws_vpc_endpoint_service.foo vpce-svc-0f97a19d3fa8220bc
```

» Resource: aws_vpc_endpoint_service_allowed_principal

Provides a resource to allow a principal to discover a VPC endpoint service.

NOTE on VPC Endpoint Services and VPC Endpoint Service Allowed Principals: Terraform provides both a standalone VPC Endpoint Service Allowed Principal resource and a VPC Endpoint Service resource with an `allowed_principals` attribute. Do not use the same principal ARN in both a VPC Endpoint Service resource and a VPC Endpoint Service Allowed Principal resource. Doing so will cause a conflict and will overwrite the association.

» Example Usage

Basic usage:

```
data "aws_caller_identity" "current" {}

resource "aws_vpc_endpoint_service_allowed_principal" "allow_me_to_foo" {
  vpc_endpoint_service_id = "${aws_vpc_endpoint_service.foo.id}"
  principal_arn           = "${data.aws_caller_identity.current.arn}"
}
```

» Argument Reference

The following arguments are supported:

- `vpc_endpoint_service_id` - (Required) The ID of the VPC endpoint service to allow permission.
- `principal_arn` - (Required) The ARN of the principal to allow permissions.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the association.

» Resource: aws_vpc_endpoint_subnet_association

Provides a resource to create an association between a VPC endpoint and a subnet.

NOTE on VPC Endpoints and VPC Endpoint Subnet Associations: Terraform provides both a standalone VPC Endpoint Subnet Association (an association between a VPC endpoint and a single `subnet_id`) and a VPC Endpoint resource with a `subnet_ids` attribute. Do not use the same subnet ID in both a VPC Endpoint resource and a VPC Endpoint Subnet Association resource. Doing so will cause a conflict of associations and will overwrite the association.

» Example Usage

Basic usage:

```
resource "aws_vpc_endpoint_subnet_association" "sn_ec2" {
  vpc_endpoint_id = "${aws_vpc_endpoint.ec2.id}"
  subnet_id       = "${aws_subnet.sn.id}"
}
```

» Argument Reference

The following arguments are supported:

- `vpc_endpoint_id` - (Required) The ID of the VPC endpoint with which the subnet will be associated.
- `subnet_id` - (Required) The ID of the subnet to be associated with the VPC endpoint.

» Timeouts

`aws_vpc_endpoint_subnet_association` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating the association
- `delete` - (Default `10 minutes`) Used for destroying the association

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the association.

» Resource: aws_vpc_ipv4_cidr_block_association

Provides a resource to associate additional IPv4 CIDR blocks with a VPC.

When a VPC is created, a primary IPv4 CIDR block for the VPC must be specified. The `aws_vpc_ipv4_cidr_block_association` resource allows further IPv4 CIDR blocks to be added to the VPC.

» Example Usage

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_vpc_ipv4_cidr_block_association" "secondary_cidr" {
  vpc_id     = "${aws_vpc.main.id}"
  cidr_block = "172.2.0.0/16"
}
```

» Argument Reference

The following arguments are supported:

- `cidr_block` - (Required) The additional IPv4 CIDR block to associate with the VPC.
- `vpc_id` - (Required) The ID of the VPC to make the association with.

» Timeouts

`aws_vpc_ipv4_cidr_block_association` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating the association
- `delete` - (Default `10 minutes`) Used for destroying the association

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPC CIDR association

» Import

`aws_vpc_ipv4_cidr_block_association` can be imported by using the VPC CIDR Association ID, e.g.

```
$ terraform import aws_vpc_ipv4_cidr_block_association.example vpc-cidr-assoc-xxxxxxxx
```

» Resource: aws_vpc_peering_connection

Provides a resource to manage a VPC peering connection.

NOTE on VPC Peering Connections and VPC Peering Connection Options: Terraform provides both a standalone VPC Peering Connection Options and a VPC Peering Connection resource with `accepter` and `requester` attributes. Do not manage options for the same VPC peering connection in both a VPC Peering Connection resource and a VPC Peering Connection Options resource. Doing so will cause a conflict of options and will overwrite the options. Using a VPC Peering Connection Options resource decouples management of the connection options from management of the VPC Peering Connection and allows options to be set correctly in cross-account scenarios.

Note: For cross-account (requester's AWS account differs from the accepter's AWS account) or inter-region VPC Peering Connections use the `aws_vpc_peering_connection` resource to manage the requester's side of the connection and use the `aws_vpc_peering_connection_accepter` resource to manage the accepter's side of the connection.

» Example Usage

```
resource "aws_vpc_peering_connection" "foo" {
  peer_owner_id = "${var.peer_owner_id}"
  peer_vpc_id   = "${aws_vpc.bar.id}"
  vpc_id        = "${aws_vpc.foo.id}"
}
```

Basic usage with connection options:

```
resource "aws_vpc_peering_connection" "foo" {
  peer_owner_id = "${var.peer_owner_id}"
  peer_vpc_id   = "${aws_vpc.bar.id}"
  vpc_id        = "${aws_vpc.foo.id}"

  accepter {
    allow_remote_vpc_dns_resolution = true
  }

  requester {
    allow_remote_vpc_dns_resolution = true
  }
}
```

Basic usage with tags:

```
resource "aws_vpc_peering_connection" "foo" {
  peer_owner_id = "${var.peer_owner_id}"
  peer_vpc_id   = "${aws_vpc.bar.id}"
  vpc_id        = "${aws_vpc.foo.id}"
  auto_accept   = true

  tags = {
    Name = "VPC Peering between foo and bar"
  }
}

resource "aws_vpc" "foo" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_vpc" "bar" {
  cidr_block = "10.2.0.0/16"
}
```

Basic usage with region:

```
resource "aws_vpc_peering_connection" "foo" {
  peer_owner_id = "${var.peer_owner_id}"
  peer_vpc_id   = "${aws_vpc.bar.id}"
  vpc_id        = "${aws_vpc.foo.id}"
  peer_region   = "us-east-1"
}

resource "aws_vpc" "foo" {
  provider   = "aws.us-west-2"
  cidr_block = "10.1.0.0/16"
}

resource "aws_vpc" "bar" {
  provider   = "aws.us-east-1"
  cidr_block = "10.2.0.0/16"
}
```

» Argument Reference

Note: Modifying the VPC Peering Connection options requires peering to be active. An automatic activation can be done using the `auto_accept` attribute. Alternatively, the VPC Peering Connection has to be made active manually using other means. See notes below for more information.

The following arguments are supported:

- `peer_owner_id` - (Optional) The AWS account ID of the owner of the peer

VPC. Defaults to the account ID the AWS provider is currently connected to.

- `peer_vpc_id` - (Required) The ID of the VPC with which you are creating the VPC Peering Connection.
- `vpc_id` - (Required) The ID of the requester VPC.
- `auto_accept` - (Optional) Accept the peering (both VPCs need to be in the same AWS account).
- `peer_region` - (Optional) The region of the accepter VPC of the [VPC Peering Connection]. `auto_accept` must be `false`, and use the `aws_vpc_peering_connection_accepter` to manage the accepter side.
- `accepter` (Optional) - An optional configuration block that allows for VPC Peering Connection options to be set for the VPC that accepts the peering connection (a maximum of one).
- `requester` (Optional) - A optional configuration block that allows for VPC Peering Connection options to be set for the VPC that requests the peering connection (a maximum of one).
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Accepter and Requester Arguments

Note: When enabled, the DNS resolution feature requires that VPCs participating in the peering must have support for the DNS hostnames enabled. This can be done using the `enable_dns_hostnames` attribute in the `aws_vpc` resource. See Using DNS with Your VPC user guide for more information.

- `allow_remote_vpc_dns_resolution` - (Optional) Allow a local VPC to resolve public DNS hostnames to private IP addresses when queried from instances in the peer VPC. This is not supported for inter-region VPC peering.
- `allow_classic_link_to_remote_vpc` - (Optional) Allow a local linked EC2-Classic instance to communicate with instances in a peer VPC. This enables an outbound communication from the local ClassicLink connection to the remote VPC.
- `allow_vpc_to_remote_classic_link` - (Optional) Allow a local VPC to communicate with a linked EC2-Classic instance in a peer VPC. This enables an outbound communication from the local VPC to the remote ClassicLink connection.

» Timeouts

`aws_vpc_peering_connection` provides the following Timeouts configuration options:

- `create` - (Default `1 minute`) Used for creating a peering connection
- `update` - (Default `1 minute`) Used for peering connection modifications
- `delete` - (Default `1 minute`) Used for destroying peering connections

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPC Peering Connection.
- `accept_status` - The status of the VPC Peering Connection request.

» Notes

If both VPCs are not in the same AWS account do not enable the `auto_accept` attribute. The accepter can manage its side of the connection using the `aws_vpc_peering_connection_accepter` resource or accept the connection manually using the AWS Management Console, AWS CLI, through SDKs, etc.

» Import

VPC Peering resources can be imported using the `vpc peering id`, e.g.

```
$ terraform import aws_vpc_peering_connection.test_connection pcx-111aaa111
```

» Resource: aws_vpc_peering_connection_accepter

Provides a resource to manage the accepter's side of a VPC Peering Connection.

When a cross-account (requester's AWS account differs from the accepter's AWS account) or an inter-region VPC Peering Connection is created, a VPC Peering Connection resource is automatically created in the accepter's account. The requester can use the `aws_vpc_peering_connection` resource to manage its side of the connection and the accepter can use the `aws_vpc_peering_connection_accepter` resource to "adopt" its side of the connection into management.

» Example Usage

```
provider "aws" {
  region = "us-east-1"

  # Requester's credentials.
}

provider "aws" {
  alias  = "peer"
  region = "us-west-2"
  # Accepter's credentials.
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_vpc" "peer" {
  provider   = "aws.peer"
  cidr_block = "10.1.0.0/16"
}

data "aws_caller_identity" "peer" {
  provider = "aws.peer"
}

# Requester's side of the connection.
resource "aws_vpc_peering_connection" "peer" {
  vpc_id        = "${aws_vpc.main.id}"
  peer_vpc_id   = "${aws_vpc.peer.id}"
  peer_owner_id = "${data.aws_caller_identity.peer.account_id}"
  peer_region   = "us-west-2"
  auto_accept   = false

  tags = {
    Side = "Requester"
  }
}

# Accepter's side of the connection.
resource "aws_vpc_peering_connection_accepter" "peer" {
  provider                  = "aws.peer"
  vpc_peering_connection_id = "${aws_vpc_peering_connection.peer.id}"
  auto_accept               = true

  tags = {
    Side = "Accepter"
  }
}
```

» Argument Reference

The following arguments are supported:

- `vpc_peering_connection_id` - (Required) The VPC Peering Connection ID to manage.
- `auto_accept` - (Optional) Whether or not to accept the peering request. Defaults to `false`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Removing `aws_vpc_peering_connection_accepter` from your configuration

AWS allows a cross-account VPC Peering Connection to be deleted from either the requester's or accepter's side. However, Terraform only allows the VPC Peering Connection to be deleted from the requester's side by removing the corresponding `aws_vpc_peering_connection` resource from your configuration. Removing a `aws_vpc_peering_connection_accepter` resource from your configuration will remove it from your statefile and management, **but will not destroy the VPC Peering Connection.**

» Attributes Reference

All of the argument attributes except `auto_accept` are also exported as result attributes.

- `id` - The ID of the VPC Peering Connection.
- `accept_status` - The status of the VPC Peering Connection request.
- `vpc_id` - The ID of the accepter VPC.
- `peer_vpc_id` - The ID of the requester VPC.
- `peer_owner_id` - The AWS account ID of the owner of the requester VPC.
- `peer_region` - The region of the accepter VPC.
- `accepter` - A configuration block that describes VPC Peering Connection options set for the accepter VPC.
- `requester` - A configuration block that describes VPC Peering Connection options set for the requester VPC.

» Accepter and Requester Attributes Reference

- `allow_remote_vpc_dns_resolution` - Indicates whether a local VPC can resolve public DNS hostnames to private IP addresses when queried from instances in a peer VPC.
- `allow_classic_link_to_remote_vpc` - Indicates whether a local ClassicLink connection can communicate with the peer VPC over the VPC Peering Connection.
- `allow_vpc_to_remote_classic_link` - Indicates whether a local VPC can communicate with a ClassicLink connection in the peer VPC over the VPC Peering Connection.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPC Peering Connection.

» Import

VPC Peering Connection Accepters can be imported by using the Peering Connection ID, e.g.

```
$ terraform import aws_vpc_peering_connection_accepter.example pcx-12345678
```

Certain resource arguments, like `auto_accept`, do not have an EC2 API method for reading the information after peering connection creation. If the argument is set in the Terraform configuration on an imported resource, Terraform will always show a difference. To workaround this behavior, either omit the argument from the Terraform configuration or use `ignore_changes` to hide the difference, e.g.

```
resource "aws_vpc_peering_connection_accepter" "example" {
  # ... other configuration ...

  # There is no AWS EC2 API for reading auto_accept
  lifecycle {
    ignore_changes = ["auto_accept"]
  }
}
```

» Resource: aws_vpc_peering_connection_options

Provides a resource to manage VPC peering connection options.

NOTE on VPC Peering Connections and VPC Peering Connection Options: Terraform provides both a standalone VPC Peering Connection Options and a VPC Peering Connection resource with `accepter` and `requester` attributes. Do not manage options for the same VPC peering connection in both a VPC Peering Connection resource and a VPC Peering Connection Options resource. Doing so will cause a conflict of options and will overwrite the options. Using a VPC Peering Connection Options resource decouples management of the connection options from management of the VPC Peering Connection and allows options to be set correctly in cross-region and cross-account scenarios.

Basic usage:

```
resource "aws_vpc" "foo" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_vpc" "bar" {
  cidr_block = "10.1.0.0/16"
}

resource "aws_vpc_peering_connection" "foo" {
  vpc_id      = "${aws_vpc.foo.id}"
  peer_vpc_id = "${aws_vpc.bar.id}"
  auto_accept = true
}

resource "aws_vpc_peering_connection_options" "foo" {
  vpc_peering_connection_id = "${aws_vpc_peering_connection.foo.id}"

  accepter {
    allow_remote_vpc_dns_resolution = true
  }

  requester {
    allow_vpc_to_remote_classic_link = true
    allow_classic_link_to_remote_vpc = true
  }
}
```

Basic cross-account usage:

```
provider "aws" {
  alias = "requester"

  # Requester's credentials.
}

provider "aws" {
  alias = "accepter"

  # Accepter's credentials.
}

resource "aws_vpc" "main" {
  provider = "aws.requester"

  cidr_block = "10.0.0.0/16"

  enable_dns_support   = true
  enable_dns_hostnames = true
}
```

```
resource "aws_vpc" "peer" {
  provider = "aws.accepter"

  cidr_block = "10.1.0.0/16"

  enable_dns_support   = true
  enable_dns_hostnames = true
}

data "aws_caller_identity" "peer" {
  provider = "aws.accepter"
}

# Requester's side of the connection.
resource "aws_vpc_peering_connection" "peer" {
  provider = "aws.requester"

  vpc_id        = "${aws_vpc.main.id}"
  peer_vpc_id   = "${aws_vpc.peer.id}"
  peer_owner_id = "${data.aws_caller_identity.peer.account_id}"
  auto_accept   = false

  tags = {
    Side = "Requester"
  }
}

# Accepter's side of the connection.
resource "aws_vpc_peering_connection_accepter" "peer" {
  provider = "aws.accepter"

  vpc_peering_connection_id = "${aws_vpc_peering_connection.peer.id}"
  auto_accept               = true

  tags = {
    Side = "Accepter"
  }
}

resource "aws_vpc_peering_connection_options" "requester" {
  provider = "aws.requester"

  # As options can't be set until the connection has been accepted
  # create an explicit dependency on the accepter.
  vpc_peering_connection_id = "${aws_vpc_peering_connection_accepter.peer.id}"
  requester {
    allow_remote_vpc_dns_resolution = true
  }
}

resource "aws_vpc_peering_connection_options" "accepter" {
  provider = "aws.accepter"

  vpc_peering_connection_id = "${aws_vpc_peering_connection_accepter.peer.id}"

  accepter {
    allow_remote_vpc_dns_resolution = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `vpc_peering_connection_id` - (Required) The ID of the requester VPC peering connection.
- `accepter` (Optional) - An optional configuration block that allows for VPC Peering Connection options to be set for the VPC that accepts the peering connection (a maximum of one).
- `requester` (Optional) - A optional configuration block that allows for VPC Peering Connection options to be set for the VPC that requests the peering connection (a maximum of one).

» Accepter and Requester Arguments

Note: When enabled, the DNS resolution feature requires that VPCs participating in the peering must have support for the DNS hostnames enabled. This can be done using the `enable_dns_hostnames` attribute in the `aws_vpc` resource. See Using DNS with Your VPC user guide for more information.

- `allow_remote_vpc_dns_resolution` - (Optional) Allow a local VPC to resolve public DNS hostnames to private IP addresses when queried from instances in the peer VPC.
- `allow_classic_link_to_remote_vpc` - (Optional) Allow a local linked EC2-Classic instance to communicate with instances in a peer VPC. This enables an outbound communication from the local ClassicLink connection to the remote VPC. This option is not supported for inter-region VPC peering.
- `allow_vpc_to_remote_classic_link` - (Optional) Allow a local VPC to communicate with a linked EC2-Classic instance in a peer VPC. This

enables an outbound communication from the local VPC to the remote ClassicLink connection. This option is not supported for inter-region VPC peering.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPC Peering Connection Options.

» Import

VPC Peering Connection Options can be imported using the `vpc peering id`, e.g.

```
$ terraform import aws_vpc_peering_connection_options.foo pcx-111aaa111
```

» Resource: aws_vpn_connection

Manages an EC2 VPN connection. These objects can be connected to customer gateways, and allow you to establish tunnels between your network and Amazon.

Note: All arguments including `tunnel1_preshared_key` and `tunnel2_preshared_key` will be stored in the raw state as plain-text. Read more about sensitive data in state.

Note: The CIDR blocks in the arguments `tunnel1_inside_cidr` and `tunnel2_inside_cidr` must have a prefix of /30 and be a part of a specific range. Read more about this in the AWS documentation.

» Example Usage

» EC2 Transit Gateway

```
resource "aws_ec2_transit_gateway" "example" {}

resource "aws_customer_gateway" "example" {
  bgp_asn    = 65000
  ip_address = "172.0.0.1"
  type       = "ipsec.1"
}

resource "aws_vpn_connection" "example" {
  customer_gateway_id = "${aws_customer_gateway.example.id}"
  transit_gateway_id  = "${aws_ec2_transit_gateway.example.id}"
  type                = "${aws_customer_gateway.example.type}"
}
```

» Virtual Private Gateway

```
resource "aws_vpc" "vpc" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_vpn_gateway" "vpn_gateway" {
  vpc_id = "${aws_vpc.vpc.id}"
}

resource "aws_customer_gateway" "customer_gateway" {
  bgp_asn    = 65000
  ip_address = "172.0.0.1"
  type       = "ipsec.1"
}

resource "aws_vpn_connection" "main" {
  vpn_gateway_id      = "${aws_vpn_gateway.vpn_gateway.id}"
  customer_gateway_id = "${aws_customer_gateway.customer_gateway.id}"
  type                = "ipsec.1"
  static_routes_only  = true
}
```

» Argument Reference

The following arguments are required:

- `customer_gateway_id` - (Required) The ID of the customer gateway.
- `type` - (Required) The type of VPN connection. The only type AWS supports at this time is ”ipsec.1”.

One of the following arguments is required:

- `transit_gateway_id` - (Optional) The ID of the EC2 Transit Gateway.
- `vpn_gateway_id` - (Optional) The ID of the Virtual Private Gateway.

Other arguments:

- `static_routes_only` - (Optional, Default `false`) Whether the VPN connection uses static routes exclusively. Static routes must be used for devices that don’t support BGP.

- `tags` - (Optional) Tags to apply to the connection.
- `tunnel1_inside_cidr` - (Optional) The CIDR block of the inside IP addresses for the first VPN tunnel.
- `tunnel2_inside_cidr` - (Optional) The CIDR block of the inside IP addresses for the second VPN tunnel.
- `tunnel1_preshared_key` - (Optional) The preshared key of the first VPN tunnel.
- `tunnel2_preshared_key` - (Optional) The preshared key of the second VPN tunnel.

Note: The preshared key must be between 8 and 64 characters in length and cannot start with zero(0). Allowed characters are alphanumeric characters, periods(.) and underscores(_).

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The amazon-assigned ID of the VPN connection.
- `customer_gateway_configuration` - The configuration information for the VPN connection's customer gateway (in the native XML format).
- `customer_gateway_id` - The ID of the customer gateway to which the connection is attached.
- `static_routes_only` - Whether the VPN connection uses static routes exclusively.
- `tags` - Tags applied to the connection.
- `transit_gateway_attachment_id` - When associated with an EC2 Transit Gateway (`transit_gateway_id` argument), the attachment ID.
- `tunnel1_address` - The public IP address of the first VPN tunnel.
- `tunnel1_cgw_inside_address` - The RFC 6890 link-local address of the first VPN tunnel (Customer Gateway Side).
- `tunnel1_vgw_inside_address` - The RFC 6890 link-local address of the first VPN tunnel (VPN Gateway Side).
- `tunnel1_preshared_key` - The preshared key of the first VPN tunnel.
- `tunnel1_bgp_asn` - The bgp asn number of the first VPN tunnel.
- `tunnel1_bgp_holdtime` - The bgp holdtime of the first VPN tunnel.
- `tunnel2_address` - The public IP address of the second VPN tunnel.
- `tunnel2_cgw_inside_address` - The RFC 6890 link-local address of the second VPN tunnel (Customer Gateway Side).
- `tunnel2_vgw_inside_address` - The RFC 6890 link-local address of the second VPN tunnel (VPN Gateway Side).
- `tunnel2_preshared_key` - The preshared key of the second VPN tunnel.
- `tunnel2_bgp_asn` - The bgp asn number of the second VPN tunnel.
- `tunnel2_bgp_holdtime` - The bgp holdtime of the second VPN tunnel.
- `type` - The type of VPN connection.

- vpn_gateway_id - The ID of the virtual private gateway to which the connection is attached.

» Import

VPN Connections can be imported using the vpn connection id, e.g.

```
$ terraform import aws_vpn_connection.testvpnconnection vpn-40f41529
```

» Resource: aws_vpn_connection_route

Provides a static route between a VPN connection and a customer gateway.

» Example Usage

```
resource "aws_vpc" "vpc" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_vpn_gateway" "vpn_gateway" {
  vpc_id = "${aws_vpc.vpc.id}"
}

resource "aws_customer_gateway" "customer_gateway" {
  bgp_asn    = 65000
  ip_address = "172.0.0.1"
  type       = "ipsec.1"
}

resource "aws_vpn_connection" "main" {
  vpn_gateway_id      = "${aws_vpn_gateway.vpn_gateway.id}"
  customer_gateway_id = "${aws_customer_gateway.customer_gateway.id}"
  type                = "ipsec.1"
  static_routes_only  = true
}

resource "aws_vpn_connection_route" "office" {
  destination_cidr_block = "192.168.10.0/24"
  vpn_connection_id      = "${aws_vpn_connection.main.id}"
}
```

» Argument Reference

The following arguments are supported:

- `destination_cidr_block` - (Required) The CIDR block associated with the local subnet of the customer network.
- `vpn_connection_id` - (Required) The ID of the VPN connection.

» Attribute Reference

In addition to all arguments above, the following attributes are exported:

- `destination_cidr_block` - The CIDR block associated with the local subnet of the customer network.
- `vpn_connection_id` - The ID of the VPN connection.

» Resource: aws_vpn_gateway

Provides a resource to create a VPC VPN Gateway.

» Example Usage

```
resource "aws_vpn_gateway" "vpn_gw" {
  vpc_id = "${aws_vpc.main.id}"

  tags = {
    Name = "main"
  }
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Optional) The VPC ID to create in.
- `availability_zone` - (Optional) The Availability Zone for the virtual private gateway.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `amazon_side_asn` - (Optional) The Autonomous System Number (ASN) for the Amazon side of the gateway. If you don't specify an ASN, the virtual private gateway is created with the default ASN.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the VPN Gateway.

» Import

VPN Gateways can be imported using the `vpn gateway id`, e.g.

```
$ terraform import aws_vpn_gateway.testvpngateway vgw-9a4cacf3
```

» Resource: aws_vpn_gateway_attachment

Provides a Virtual Private Gateway attachment resource, allowing for an existing hardware VPN gateway to be attached and/or detached from a VPC.

Note: The `aws_vpn_gateway` resource can also automatically attach the Virtual Private Gateway it creates to an existing VPC by setting the `vpc_id` attribute accordingly.

» Example Usage

```
resource "aws_vpc" "network" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_vpn_gateway" "vpn" {
  tags = {
    Name = "example-vpn-gateway"
  }
}

resource "aws_vpn_gateway_attachment" "vpn_attachment" {
  vpc_id         = "${aws_vpc.network.id}"
  vpn_gateway_id = "${aws_vpn_gateway.vpn.id}"
}
```

See Virtual Private Cloud and Virtual Private Gateway user guides for more information.

» Argument Reference

The following arguments are supported:

- `vpc_id` - (Required) The ID of the VPC.
- `vpn_gateway_id` - (Required) The ID of the Virtual Private Gateway.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `vpc_id` - The ID of the VPC that Virtual Private Gateway is attached to.
- `vpn_gateway_id` - The ID of the Virtual Private Gateway.

» Import

This resource does not support importing.

» Resource: aws_vpn_gateway_route_propagation

Requests automatic route propagation between a VPN gateway and a route table.

Note: This resource should not be used with a route table that has the `propagating_vgws` argument set. If that argument is set, any route propagation not explicitly listed in its value will be removed.

» Example Usage

```
resource "aws_vpn_gateway_route_propagation" "example" {
  vpn_gateway_id = "${aws_vpn_gateway.example.id}"
  route_table_id = "${aws_route_table.example.id}"
}
```

» Argument Reference

The following arguments are required:

- `vpn_gateway_id` - The id of the `aws_vpn_gateway` to propagate routes from.
- `route_table_id` - The id of the `aws_route_table` to propagate routes into.

» Attributes Reference

This resource does not export any additional attributes.

» Data Source: aws_waf_web_acl

`aws_waf_web_acl` Retrieves a WAF Web ACL Resource Id.

» Example Usage

```
data "aws_waf_web_acl" "example" {
  name = "tfWAFWebACL"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF Web ACL.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Web ACL.

» Data Source: aws_waf_ipset

`aws_waf_ipset` Retrieves a WAF IP Set Resource Id.

» Example Usage

```
data "aws_waf_ipset" "example" {
  name = "tfWAFIPSet"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF IP set.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF IP set.

» Data Source: aws_waf_rule

`aws_waf_rule` Retrieves a WAF Rule Resource Id.

» Example Usage

```
data "aws_waf_rule" "example" {
  name = "tfWAFRule"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF rule.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF rule.

» Data Source: aws_waf_rate_based_rule

`aws_waf_rate_based_rule` Retrieves a WAF Rate Based Rule Resource Id.

» Example Usage

```
data "aws_waf_rate_based_rule" "example" {
  name = "tfWAFRateBasedRule"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF rate based rule.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF rate based rule.

» Resource: aws_waf_byte_match_set

Provides a WAF Byte Match Set Resource

» Example Usage

```
resource "aws_waf_byte_match_set" "byte_set" {
  name = "tf_waf_byte_match_set"

  byte_match_tuples {
    text_transformation   = "NONE"
    target_string         = "badrefer1"
    positional_constraint = "CONTAINS"

    field_to_match {
      type = "HEADER"
      data = "referer"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Byte Match Set.
- `byte_match_tuples` - Specifies the bytes (typically a string that corresponds with ASCII characters) that you want to search for in web requests, the location in requests that you want to search, and other settings.

» Nested blocks

» `byte_match_tuples`

» Arguments

- `field_to_match` - (Required) The part of a web request that you want to search, such as a specified header or a query string.
- `positional_constraint` - (Required) Within the portion of a web request that you want to search (for example, in the query string, if any), specify where you want to search. e.g. `CONTAINS`, `CONTAINS_WORD` or `EXACTLY`. See docs for all supported values.
- `target_string` - (Optional) The value that you want to search for. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.
- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. If you specify a transformation, AWS WAF performs the transformation on `target_string` before inspecting a request for a match. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values.

» `field_to_match`

» Arguments

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Byte Match Set.

» Import

WAF Byte Match Set can be imported using the id, e.g.

```
$ terraform import aws_waf_byte_match_set.byte_set a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_waf_geo_match_set

Provides a WAF Geo Match Set Resource

» Example Usage

```
resource "aws_waf_geo_match_set" "geo_match_set" {
  name = "geo_match_set"

  geo_match_constraint {
    type  = "Country"
    value = "US"
  }

  geo_match_constraint {
    type  = "Country"
    value = "CA"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the GeoMatchSet.
- `geo_match_constraint` - (Optional) The GeoMatchConstraint objects which contain the country that you want AWS WAF to search for.

» Nested Blocks

» `geo_match_constraint`

» Arguments

- `type` - (Required) The type of geographical area you want AWS WAF to search for. Currently Country is the only valid value.
- `value` - (Required) The country that you want AWS WAF to search for. This is the two-letter country code, e.g. `US`, `CA`, `RU`, `CN`, etc. See docs for all supported values.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF GeoMatchSet.
- `arn` - Amazon Resource Name (ARN)

» Import

WAF Geo Match Set can be imported using their ID, e.g.

```
$ terraform import aws_waf_geo_match_set.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_waf_ipset

Provides a WAF IPSet Resource

» Example Usage

```
resource "aws_waf_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptors {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }

  ip_set_descriptors {
    type  = "IPV4"
    value = "10.16.16.0/16"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the IPSet.
- `ip_set_descriptors` - (Optional) One or more pairs specifying the IP address type (IPV4 or IPV6) and the IP address range (in CIDR format) from which web requests originate.

» Nested Blocks

» `ip_set_descriptors`

» Arguments

- `type` - (Required) Type of the IP address - `IPV4` or `IPV6`.
- `value` - (Required) An IPv4 or IPv6 address specified via CIDR notation. e.g. `192.0.2.44/32` or `1111:0000:0000:0000:0000:0000:0000:0000/64`

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF IPSet.
- `arn` - The ARN of the WAF IPSet.

» Import

WAF IPSets can be imported using their ID, e.g.

```
$ terraform import aws_waf_ipset.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_waf_rate_based_rule

Provides a WAF Rate Based Rule Resource

» Example Usage

```
resource "aws_waf_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptors {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }
}

resource "aws_waf_rate_based_rule" "wafrule" {
  depends_on  = ["aws_waf_ipset.ipset"]
  name        = "tfWAFRule"
  metric_name = "tfWAFRule"

  rate_key   = "IP"
  rate_limit = 100

  predicates {
    data_id = "${aws_waf_ipset.ipset.id}"
    negated = false
    type    = "IPMatch"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metric_name` - (Required) The name or description for the Amazon CloudWatch metric of this rule.
- `name` - (Required) The name or description of the rule.
- `rate_key` - (Required) Valid value is IP.
- `rate_limit` - (Required) The maximum number of requests, which have an identical value in the field specified by the RateKey, allowed in a five-minute period. Minimum value is 100.
- `predicates` - (Optional) The objects to include in a rule (documented below).
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Blocks

» predicates

See the WAF Documentation for more information.

» Arguments

- `negated` - (Required) Set this to `false` if you want to allow, block, or count requests based on the settings in the specified `ByteMatchSet`, `IPSet`, `SqlInjectionMatchSet`, `XssMatchSet`, or `SizeConstraintSet`. For example, if an IPSet includes the IP address `192.0.2.44`, AWS WAF will allow or block requests based on that IP address. If set to `true`, AWS WAF will allow, block, or count requests based on all IP addresses *except* `192.0.2.44`.
- `data_id` - (Required) A unique identifier for a predicate in the rule, such as Byte Match Set ID or IPSet ID.
- `type` - (Required) The type of predicate in a rule. Valid values: `ByteMatch`, `GeoMatch`, `IPMatch`, `RegexMatch`, `SizeConstraint`, `SqlInjectionMatch`, or `XssMatch`.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF rule.
- `arn` - Amazon Resource Name (ARN)

» Import

WAF Rated Based Rule can be imported using the id, e.g.

```
$ terraform import aws_waf_rate_based_rule.wafrule a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_waf_regex_match_set

Provides a WAF Regex Match Set Resource

» Example Usage

```
resource "aws_waf_regex_match_set" "example" {
  name = "example"

  regex_match_tuple {
    field_to_match {
      data = "User-Agent"
      type = "HEADER"
    }

    regex_pattern_set_id = "${aws_waf_regex_pattern_set.example.id}"
    text_transformation  = "NONE"
  }
}

resource "aws_waf_regex_pattern_set" "example" {
  name                  = "example"
  regex_pattern_strings = ["one", "two"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Regex Match Set.
- `regex_match_tuple` - (Required) The regular expression pattern that you want AWS WAF to search for in web requests, the location in requests that you want AWS WAF to search, and other settings. See below.

» Nested Arguments

» `regex_match_tuple`

- `field_to_match` - (Required) The part of a web request that you want to search, such as a specified header or a query string.
- `regex_pattern_set_id` - (Required) The ID of a Regex Pattern Set.
- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values.

» `field_to_match`

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regex Match Set.
- `arn` - Amazon Resource Name (ARN)

» Import

WAF Regex Match Set can be imported using their ID, e.g.

```
$ terraform import aws_waf_regex_match_set.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_waf_regex_pattern_set

Provides a WAF Regex Pattern Set Resource

» Example Usage

```
resource "aws_waf_regex_pattern_set" "example" {
  name                  = "tf_waf_regex_pattern_set"
  regex_pattern_strings = ["one", "two"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Regex Pattern Set.
- `regex_pattern_strings` - (Optional) A list of regular expression (regex) patterns that you want AWS WAF to search for, such as `B[a@]dB[o0]t`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regex Pattern Set.
- `arn` - Amazon Resource Name (ARN)

» Import

AWS WAF Regex Pattern Set can be imported using their ID, e.g.

```
$ terraform import aws_waf_regex_pattern_set.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbccc
```

» Resource: aws_waf_rule

Provides a WAF Rule Resource

» Example Usage

```
resource "aws_waf_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptors {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }
}

resource "aws_waf_rule" "wafrule" {
  depends_on  = ["aws_waf_ipset.ipset"]
  name        = "tfWAFRule"
  metric_name = "tfWAFRule"

  predicates {
    data_id = "${aws_waf_ipset.ipset.id}"
    negated = false
    type    = "IPMatch"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metric_name` - (Required) The name or description for the Amazon CloudWatch metric of this rule. The name can contain only alphanumeric characters (A-Z, a-z, 0-9); the name can't contain whitespace.
- `name` - (Required) The name or description of the rule.
- `predicates` - (Optional) The objects to include in a rule (documented below).
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Blocks

» `predicates`

See the WAF Documentation for more information.

» Arguments

- `negated` - (Required) Set this to `false` if you want to allow, block, or count requests based on the settings in the specified waf_byte_match_set, waf_ipset, aws_waf_size_constraint_set, aws_waf_sql_injection_match_set or aws_waf_xss_match_set. For example, if an IPSet includes the IP address `192.0.2.44`, AWS WAF will allow or block requests based on that IP address. If set to `true`, AWS WAF will allow, block, or count requests based on all IP addresses *except* `192.0.2.44`.
- `data_id` - (Required) A unique identifier for a predicate in the rule, such as Byte Match Set ID or IPSet ID.
- `type` - (Required) The type of predicate in a rule. Valid values: `ByteMatch`, `GeoMatch`, `IPMatch`, `RegexMatch`, `SizeConstraint`, `SqlInjectionMatch`, or `XssMatch`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF rule.
- `arn` - The ARN of the WAF rule.

» Import

WAF rules can be imported using the id, e.g.

```
$ terraform import aws_waf_rule.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_waf_rule_group

Provides a WAF Rule Group Resource

» Example Usage

```
resource "aws_waf_rule" "example" {
  name        = "example"
  metric_name = "example"
}

resource "aws_waf_rule_group" "example" {
  name        = "example"
  metric_name = "example"

  activated_rule {
    action {
      type = "COUNT"
    }

    priority = 50
    rule_id  = "${aws_waf_rule.example.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A friendly name of the rule group
- `metric_name` - (Required) A friendly name for the metrics from the rule group
- `activated_rule` - (Optional) A list of activated rules, see below
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Blocks

» `activated_rule`

» Arguments

- `action` - (Required) Specifies the action that CloudFront or AWS WAF takes when a web request matches the conditions in the rule.
  - `type` - (Required) e.g. `BLOCK`, `ALLOW`, or `COUNT`

- `priority` - (Required) Specifies the order in which the rules are evaluated. Rules with a lower value are evaluated before rules with a higher value.
- `rule_id` - (Required) The ID of a rule
- `type` - (Optional) The rule type, either `REGULAR`, `RATE_BASED`, or `GROUP`. Defaults to `REGULAR`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF rule group.
- `arn` - The ARN of the WAF rule group.

» Import

WAF Rule Group can be imported using the id, e.g.

```
$ terraform import aws_waf_rule_group.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_waf_size_constraint_set

Provides a WAF Size Constraint Set Resource

» Example Usage

```
resource "aws_waf_size_constraint_set" "size_constraint_set" {
  name = "tfsize_constraints"

  size_constraints {
    text_transformation = "NONE"
    comparison_operator = "EQ"
    size                = "4096"

    field_to_match {
      type = "BODY"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Size Constraint Set.
- `size_constraints` - (Optional) Specifies the parts of web requests that you want to inspect the size of.

» Nested Blocks

» `size_constraints`

» Arguments

- `field_to_match` - (Required) Specifies where in a web request to look for the size constraint.
- `comparison_operator` - (Required) The type of comparison you want to perform. e.g. `EQ`, `NE`, `LT`, `GT`. See docs for all supported values.
- `size` - (Required) The size in bytes that you want to compare against the size of the specified `field_to_match`. Valid values are between 0 - 21474836480 bytes (0 - 20 GB).
- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. If you specify a transformation, AWS WAF performs the transformation on `field_to_match` before inspecting a request for a match. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values. **Note:** if you choose `BODY` as `type`, you must choose `NONE` because CloudFront forwards only the first 8192 bytes for inspection.

» `field_to_match`

» Arguments

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Size Constraint Set.
- `arn` - Amazon Resource Name (ARN)

» Import

AWS WAF Size Constraint Set can be imported using their ID, e.g.

```
$ terraform import aws_waf_size_constraint_set.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbc
```

» Resource: aws_waf_sql_injection_match_set

Provides a WAF SQL Injection Match Set Resource

» Example Usage

```
resource "aws_waf_sql_injection_match_set" "sql_injection_match_set" {
  name = "tf-sql_injection_match_set"

  sql_injection_match_tuples {
    text_transformation = "URL_DECODE"

    field_to_match {
      type = "QUERY_STRING"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the SQL Injection Match Set.
- `sql_injection_match_tuples` - (Optional) The parts of web requests that you want AWS WAF to inspect for malicious SQL code and, if you want AWS WAF to inspect a header, the name of the header.

» Nested Blocks

» `sql_injection_match_tuples`

- `field_to_match` - (Required) Specifies where in a web request to look for snippets of malicious SQL code.

- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. If you specify a transformation, AWS WAF performs the transformation on `field_to_match` before inspecting a request for a match. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values.

» `field_to_match`

» Arguments

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF SQL Injection Match Set.

» Resource: aws_waf_web_acl

Provides a WAF Web ACL Resource

» Example Usage

```
resource "aws_waf_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptors {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }
}

resource "aws_waf_rule" "wafrule" {
  depends_on  = ["aws_waf_ipset.ipset"]
  name        = "tfWAFRule"
  metric_name = "tfWAFRule"

  predicates {
    data_id = "${aws_waf_ipset.ipset.id}"
    negated = false
    type    = "IPMatch"
  }
}

resource "aws_waf_web_acl" "waf_acl" {
  depends_on  = ["aws_waf_ipset.ipset", "aws_waf_rule.wafrule"]
  name        = "tfWebACL"
  metric_name = "tfWebACL"

  default_action {
    type = "ALLOW"
  }

  rules {
    action {
      type = "BLOCK"
    }

    priority = 1
    rule_id  = "${aws_waf_rule.wafrule.id}"
    type     = "REGULAR"
  }
}
```

» Logging

NOTE: The Kinesis Firehose Delivery Stream name must begin with `aws-waf-logs-` and be located in `us-east-1` region. See the AWS WAF Developer Guide for more information about enabling WAF logging.

```
resource "aws_waf_web_acl" "example" {
  # ... other configuration ...
  logging_configuration {
    log_destination = "${aws_kinesis_firehose_delivery_stream.example.arn}"

    redacted_fields {
      field_to_match {
        type = "URI"
      }

      field_to_match {
        data = "referer"
        type = "HEADER"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `default_action` - (Required) Configuration block with action that you want AWS WAF to take when a request doesn't match the criteria in any of the rules that are associated with the web ACL. Detailed below.
- `metric_name` - (Required) The name or description for the Amazon CloudWatch metric of this web ACL.
- `name` - (Required) The name or description of the web ACL.
- `rules` - (Optional) Configuration blocks containing rules to associate with the web ACL and the settings for each rule. Detailed below.
- `logging_configuration` - (Optional) Configuration block to enable WAF logging. Detailed below.
- `tags` - (Optional) Key-value mapping of resource tags

» `default_action` Configuration Block

- `type` - (Required) Specifies how you want AWS WAF to respond to requests that don't match the criteria in any of the `rules`. e.g. `ALLOW`, `BLOCK` or `COUNT`

» `logging_configuration` Configuration Block

- `log_destination` - (Required) Amazon Resource Name (ARN) of Kinesis Firehose Delivery Stream
- `redacted_fields` - (Optional) Configuration block containing parts of the request that you want redacted from the logs. Detailed below.

» `redacted_fields` Configuration Block

- `field_to_match` - (Required) Set of configuration blocks for fields to redact. Detailed below.

» `field_to_match` Configuration Block

Additional information about this configuration can be found in the AWS WAF Regional API Reference.

- `data` - (Optional) When the value of `type` is `HEADER`, enter the name of the header that you want the WAF to search, for example, `User-Agent` or `Referer`. If the value of `type` is any other value, omit `data`.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER` or `METHOD`

» `rules` Configuration Block

See docs for all details and supported values.

- `action` - (Optional) The action that CloudFront or AWS WAF takes when a web request matches the conditions in the rule. Not used if `type` is `GROUP`.
  - `type` - (Required) valid values are: `BLOCK`, `ALLOW`, or `COUNT`
- `override_action` - (Optional) Override the action that a group requests CloudFront or AWS WAF takes when a web request matches the conditions in the rule. Only used if `type` is `GROUP`.
  - `type` - (Required) valid values are: `NONE` or `COUNT`
- `priority` - (Required) Specifies the order in which the rules in a WebACL are evaluated. Rules with a lower value are evaluated before rules with a higher value.
- `rule_id` - (Required) ID of the associated WAF (Global) rule (e.g. `aws_waf_rule`). WAF (Regional) rules cannot be used.
- `type` - (Optional) The rule type, either `REGULAR`, as defined by Rule, `RATE_BASED`, as defined by RateBasedRule, or `GROUP`, as defined by RuleGroup. The default is REGULAR. If you add a RATE_BASED rule, you need to set `type` as `RATE_BASED`. If you add a GROUP rule, you need to set `type` as `GROUP`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF WebACL.
- `arn` - The ARN of the WAF WebACL.

» Import

WAF Web ACL can be imported using the `id`, e.g.

```
$ terraform import aws_waf_web_acl.main 0c8e583e-18f3-4c13-9e2a-67c4805d2f94
```

» Resource: aws_waf_xss_match_set

Provides a WAF XSS Match Set Resource

» Example Usage

```
resource "aws_waf_xss_match_set" "xss_match_set" {
  name = "xss_match_set"

  xss_match_tuples {
    text_transformation = "NONE"

    field_to_match {
      type = "URI"
    }
  }

  xss_match_tuples {
    text_transformation = "NONE"

    field_to_match {
      type = "QUERY_STRING"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the SizeConstraintSet.
- `xss_match_tuples` - (Optional) The parts of web requests that you want to inspect for cross-site scripting attacks.

» Nested Blocks

» `xss_match_tuples`

- `field_to_match` - (Required) Specifies where in a web request to look for cross-site scripting attacks.
- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. If you specify a transformation, AWS WAF performs

the transformation on `target_string` before inspecting a request for a match. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values.

» `field_to_match`

» Arguments

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF XssMatchSet.
- `arn` - Amazon Resource Name (ARN)

» Import

WAF XSS Match Set can be imported using their ID, e.g.

```
$ terraform import aws_waf_xss_match_set.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Data Source: aws_wafregional_ipset

`aws_wafregional_ipset` Retrieves a WAF Regional IP Set Resource Id.

» Example Usage

```
data "aws_wafregional_ipset" "example" {
  name = "tfWAFRegionalIPSet"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF Regional IP set.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional IP set.

» Data Source: aws_wafregional_rule

`aws_wafregional_rule` Retrieves a WAF Regional Rule Resource Id.

» Example Usage

```
data "aws_wafregional_rule" "example" {
  name = "tfWAFRegionalRule"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF Regional rule.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional rule.

» Data Source: aws_wafregional_web_acl

`aws_wafregional_web_acl` Retrieves a WAF Regional Web ACL Resource Id.

» Example Usage

```
data "aws_wafregional_web_acl" "example" {
  name = "tfWAFRegionalWebACL"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF Regional Web ACL.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional Web ACL.

» Data Source: aws_wafregional_rate_based_rule

`aws_wafregional_rate_based_rule` Retrieves a WAF Regional Rate Based Rule Resource Id.

» Example Usage

```
data "aws_wafregional_rate_based_rule" "example" {
  name = "tfWAFRegionalRateBasedRule"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the WAF Regional rate based rule.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional rate based rule.

» Resource: aws_wafregional_byte_match_set

Provides a WAF Regional Byte Match Set Resource for use with Application Load Balancer.

» Example Usage

```
resource "aws_wafregional_byte_match_set" "byte_set" {
  name = "tf_waf_byte_match_set"

  byte_match_tuples {
    text_transformation   = "NONE"
    target_string         = "badrefer1"
    positional_constraint = "CONTAINS"

    field_to_match {
      type = "HEADER"
      data = "referer"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the ByteMatchSet.
- `byte_match_tuples` - (Optional)Settings for the ByteMatchSet, such as the bytes (typically a string that corresponds with ASCII characters) that you want AWS WAF to search for in web requests. ByteMatchTuple documented below.

ByteMatchTuples(byte_match_tuples) support the following:

- `field_to_match` - (Required) Settings for the ByteMatchTuple. FieldToMatch documented below.
- `positional_constraint` - (Required) Within the portion of a web request that you want to search.
- `target_string` - (Required) The value that you want AWS WAF to search for. The maximum length of the value is 50 bytes.
- `text_transformation` - (Required) The formatting way for web request.

FieldToMatch(field_to_match) support following:

- `data` - (Optional) When the value of Type is HEADER, enter the name of the header that you want AWS WAF to search, for example, User-Agent or Referer. If the value of Type is any other value, omit Data.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF ByteMatchSet.

» Import

WAF Regional Byte Match Set can be imported using the id, e.g.

```
$ terraform import aws_wafregional_byte_match_set.byte_set a1b2c3d4-d5f6-7777-8888-9999aaaab
```

» Resource: aws_wafregional_geo_match_set

Provides a WAF Regional Geo Match Set Resource

» Example Usage

```
resource "aws_wafregional_geo_match_set" "geo_match_set" {
  name = "geo_match_set"

  geo_match_constraint {
    type  = "Country"
    value = "US"
  }

  geo_match_constraint {
    type  = "Country"
    value = "CA"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Geo Match Set.
- `geo_match_constraint` - (Optional) The Geo Match Constraint objects which contain the country that you want AWS WAF to search for.

» Nested Blocks

» `geo_match_constraint`

» Arguments

- `type` - (Required) The type of geographical area you want AWS WAF to search for. Currently Country is the only valid value.
- `value` - (Required) The country that you want AWS WAF to search for. This is the two-letter country code, e.g. `US`, `CA`, `RU`, `CN`, etc. See docs for all supported values.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional Geo Match Set.

» Import

WAF Regional Geo Match Set can be imported using the id, e.g.

```
$ terraform import aws_wafregional_geo_match_set.geo_match_set a1b2c3d4-d5f6-7777-8888-9999a
```

» Resource: aws_wafregional_ipset

Provides a WAF Regional IPSet Resource for use with Application Load Balancer.

» Example Usage

```
resource "aws_wafregional_ipset" "ipset" {
  name = "tfIPSet"
  ip_set_descriptor {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }

  ip_set_descriptor {
    type  = "IPV4"
    value = "10.16.16.0/16"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the IPSet.
- `ip_set_descriptor` - (Optional) One or more pairs specifying the IP address type (IPV4 or IPV6) and the IP address range (in CIDR notation) from which web requests originate.

» Nested Blocks

» `ip_set_descriptor`

» Arguments

- `type` - (Required) The string like IPV4 or IPV6.
- `value` - (Required) The CIDR notation.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF IPSet.
- `arn` - The ARN of the WAF IPSet.

» Import

WAF Regional IPSets can be imported using their ID, e.g.

```
$ terraform import aws_wafregional_ipset.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_wafregional_rate_based_rule

Provides a WAF Rate Based Rule Resource

» Example Usage

```
resource "aws_wafregional_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptors {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }
}

resource "aws_wafregional_rate_based_rule" "wafrule" {
  depends_on  = ["aws_wafregional_ipset.ipset"]
  name        = "tfWAFRule"
  metric_name = "tfWAFRule"

  rate_key   = "IP"
  rate_limit = 100

  predicate {
    data_id = "${aws_wafregional_ipset.ipset.id}"
    negated = false
    type    = "IPMatch"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metric_name` - (Required) The name or description for the Amazon CloudWatch metric of this rule.
- `name` - (Required) The name or description of the rule.
- `rate_key` - (Required) Valid value is IP.
- `rate_limit` - (Required) The maximum number of requests, which have an identical value in the field specified by the RateKey, allowed in a five-minute period. Minimum value is 100.
- `predicate` - (Optional) The objects to include in a rule (documented below).
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Blocks

» `predicate`

See the WAF Documentation for more information.

» Arguments

- `negated` - (Required) Set this to `false` if you want to allow, block, or count requests based on the settings in the specified `ByteMatchSet`, `IPSet`, `SqlInjectionMatchSet`, `XssMatchSet`, or `SizeConstraintSet`. For example, if an IPSet includes the IP address `192.0.2.44`, AWS WAF will allow or block requests based on that IP address. If set to `true`, AWS WAF will allow, block, or count requests based on all IP addresses *except* `192.0.2.44`.
- `data_id` - (Required) A unique identifier for a predicate in the rule, such as Byte Match Set ID or IPSet ID.
- `type` - (Required) The type of predicate in a rule. Valid values: `ByteMatch`, `GeoMatch`, `IPMatch`, `RegexMatch`, `SizeConstraint`, `SqlInjectionMatch`, or `XssMatch`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional Rate Based Rule.
- `arn` - The ARN of the WAF Regional Rate Based Rule.

» Import

WAF Regional Rate Based Rule can be imported using the id, e.g.

```
$ terraform import aws_wafregional_rate_based_rule.wafrule a1b2c3d4-d5f6-7777-8888-9999aaaab
```

» Resource: aws_wafregional_regex_match_set

Provides a WAF Regional Regex Match Set Resource

» Example Usage

```
resource "aws_wafregional_regex_match_set" "example" {
  name = "example"
  regex_match_tuple {
    field_to_match {
      data = "User-Agent"
      type = "HEADER"
    }

    regex_pattern_set_id = "${aws_wafregional_regex_pattern_set.example.id}"
    text_transformation  = "NONE"
  }
}

resource "aws_wafregional_regex_pattern_set" "example" {
  name                  = "example"
  regex_pattern_strings = ["one", "two"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Regex Match Set.
- `regex_match_tuple` - (Required) The regular expression pattern that you want AWS WAF to search for in web requests, the location in requests that you want AWS WAF to search, and other settings. See below.

» Nested Arguments

» `regex_match_tuple`

- `field_to_match` - (Required) The part of a web request that you want to search, such as a specified header or a query string.
- `regex_pattern_set_id` - (Required) The ID of a Regex Pattern Set.
- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values.

» `field_to_match`

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.

- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional Regex Match Set.

» Import

WAF Regional Regex Match Set can be imported using the id, e.g.

```
$ terraform import aws_wafregional_regex_match_set.example a1b2c3d4-d5f6-7777-8888-9999aaaab
```

» Resource: aws_wafregional_regex_pattern_set

Provides a WAF Regional Regex Pattern Set Resource

» Example Usage

```
resource "aws_wafregional_regex_pattern_set" "example" {
  name                  = "example"
  regex_pattern_strings = ["one", "two"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Regex Pattern Set.
- `regex_pattern_strings` - (Optional) A list of regular expression (regex) patterns that you want AWS WAF to search for, such as `B[a@]dB[o0]t`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional Regex Pattern Set.

» Import

WAF Regional Regex Pattern Set can be imported using the id, e.g.

```
$ terraform import aws_wafregional_regex_pattern_set.example a1b2c3d4-d5f6-7777-8888-9999aaa
```

» Resource: aws_wafregional_rule

Provides an WAF Regional Rule Resource for use with Application Load Balancer.

» Example Usage

```
resource "aws_wafregional_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptor {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }
}

resource "aws_wafregional_rule" "wafrule" {
  name        = "tfWAFRule"
  metric_name = "tfWAFRule"

  predicate {
    type    = "IPMatch"
    data_id = "${aws_wafregional_ipset.ipset.id}"
    negated = false
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the rule.
- `metric_name` - (Required) The name or description for the Amazon CloudWatch metric of this rule.
- `predicate` - (Optional) The objects to include in a rule (documented below).
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Fields

» predicate

See the WAF Documentation for more information.

» Arguments

- `type` - (Required) The type of predicate in a rule. Valid values: `ByteMatch`, `GeoMatch`, `IPMatch`, `RegexMatch`, `SizeConstraint`, `SqlInjectionMatch`, or `XssMatch`
- `data_id` - (Required) The unique identifier of a predicate, such as the ID of a `ByteMatchSet` or `IPSet`.
- `negated` - (Required) Whether to use the settings or the negated settings that you specified in the objects.

» Remarks

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional Rule.
- `arn` - The ARN of the WAF Regional Rule.

» Import

WAF Regional Rule can be imported using the id, e.g.

```
$ terraform import aws_wafregional_rule.wafrule a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_wafregional_rule_group

Provides a WAF Regional Rule Group Resource

» Example Usage

```
resource "aws_wafregional_rule" "example" {
  name        = "example"
  metric_name = "example"
}
```

```
resource "aws_wafregional_rule_group" "example" {
  name        = "example"
  metric_name = "example"

  activated_rule {
    action {
      type = "COUNT"
    }

    priority = 50
    rule_id  = "${aws_wafregional_rule.example.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A friendly name of the rule group
- `metric_name` - (Required) A friendly name for the metrics from the rule group
- `activated_rule` - (Optional) A list of activated rules, see below
- `tags` - (Optional) Key-value mapping of resource tags

» Nested Blocks

» `activated_rule`

» Arguments

- `action` - (Required) Specifies the action that CloudFront or AWS WAF takes when a web request matches the conditions in the rule.
  - `type` - (Required) e.g. `BLOCK`, `ALLOW`, or `COUNT`
- `priority` - (Required) Specifies the order in which the rules are evaluated. Rules with a lower value are evaluated before rules with a higher value.
- `rule_id` - (Required) The ID of a rule
- `type` - (Optional) The rule type, either `REGULAR`, `RATE_BASED`, or `GROUP`. Defaults to `REGULAR`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Regional Rule Group.

- `arn` - The ARN of the WAF Regional Rule Group.

» Import

WAF Regional Rule Group can be imported using the id, e.g.

```
$ terraform import aws_wafregional_rule_group.example a1b2c3d4-d5f6-7777-8888-9999aaaabbbbc
```

» Resource: aws_wafregional_size_constraint_set

Provides a WAF Regional Size Constraint Set Resource for use with Application Load Balancer.

» Example Usage

```
resource "aws_wafregional_size_constraint_set" "size_constraint_set" {
  name = "tfsize_constraints"

  size_constraints {
    text_transformation = "NONE"
    comparison_operator = "EQ"
    size                = "4096"

    field_to_match {
      type = "BODY"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the Size Constraint Set.
- `size_constraints` - (Optional) Specifies the parts of web requests that you want to inspect the size of.

» Nested Blocks

» `size_constraints`

» Arguments

- `field_to_match` - (Required) Specifies where in a web request to look for the size constraint.
- `comparison_operator` - (Required) The type of comparison you want to perform. e.g. `EQ`, `NE`, `LT`, `GT`. See docs for all supported values.
- `size` - (Required) The size in bytes that you want to compare against the size of the specified `field_to_match`. Valid values are between 0 - 21474836480 bytes (0 - 20 GB).
- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. If you specify a transformation, AWS WAF performs the transformation on `field_to_match` before inspecting a request for a match. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values. **Note:** if you choose `BODY` as `type`, you must choose `NONE` because CloudFront forwards only the first 8192 bytes for inspection.

» `field_to_match`

» Arguments

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF Size Constraint Set.

» Import

WAF Size Constraint Set can be imported using the id, e.g.

```
$ terraform import aws_wafregional_size_constraint_set.size_constraint_set a1b2c3d4-d5f6-777
```

» Resource: aws_wafregional_sql_injection_match_set

Provides a WAF Regional SQL Injection Match Set Resource for use with Application Load Balancer.

» Example Usage

```
resource "aws_wafregional_sql_injection_match_set" "sql_injection_match_set" {
  name = "tf-sql_injection_match_set"

  sql_injection_match_tuple {
    text_transformation = "URL_DECODE"

    field_to_match {
      type = "QUERY_STRING"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name or description of the SizeConstraintSet.
- `sql_injection_match_tuple` - (Optional) The parts of web requests that you want AWS WAF to inspect for malicious SQL code and, if you want AWS WAF to inspect a header, the name of the header.

» Nested fields

» `sql_injection_match_tuple`

- `field_to_match` - (Required) Specifies where in a web request to look for snippets of malicious SQL code.
- `text_transformation` - (Required) Text transformations used to eliminate unusual formatting that attackers use in web requests in an effort to bypass AWS WAF. If you specify a transformation, AWS WAF performs the transformation on `field_to_match` before inspecting a request for a match. e.g. `CMD_LINE`, `HTML_ENTITY_DECODE` or `NONE`. See docs for all supported values.

» `field_to_match`

- `data` - (Optional) When `type` is `HEADER`, enter the name of the header that you want to search, e.g. `User-Agent` or `Referer`. If `type` is any other value, omit this field.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER`, `METHOD` or `BODY`. See docs for all supported values.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the WAF SqlInjectionMatchSet.

» Import

WAF Regional Sql Injection Match Set can be imported using the id, e.g.

```
$ terraform import aws_wafregional_sql_injection_match_set.sql_injection_match_set a1b2c3d4-
```

» Resource: aws_wafregional_web_acl

Provides a WAF Regional Web ACL Resource for use with Application Load Balancer.

» Example Usage

» Regular Rule

```
resource "aws_wafregional_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptor {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }
}

resource "aws_wafregional_rule" "wafrule" {
  name        = "tfWAFRule"
  metric_name = "tfWAFRule"

  predicate {
    data_id = "${aws_wafregional_ipset.ipset.id}"
    negated = false
    type    = "IPMatch"
  }
}

resource "aws_wafregional_web_acl" "wafacl" {
  name        = "tfWebACL"
  metric_name = "tfWebACL"

  default_action {
    type = "ALLOW"
  }

  rule {
    action {
      type = "BLOCK"
    }

    priority = 1
    rule_id  = "${aws_wafregional_rule.wafrule.id}"
    type     = "REGULAR"
  }
}
```

» Group Rule

```
resource "aws_wafregional_web_acl" "example" {
  name        = "example"
  metric_name = "example"

  default_action {
    type = "ALLOW"
  }

  rule {
    priority = 1
    rule_id  = "${aws_wafregional_rule_group.example.id}"
    type     = "GROUP"

    override_action {
      type = "NONE"
    }
  }
}
```

» Logging

NOTE: The Kinesis Firehose Delivery Stream name must begin with `aws-waf-logs-`. See the AWS WAF Developer Guide for more information about enabling WAF logging.

```
resource "aws_wafregional_web_acl" "example" {
  # ... other configuration ...

  logging_configuration {
    log_destination = "${aws_kinesis_firehose_delivery_stream.example.arn}"

    redacted_fields {
      field_to_match {
        type = "URI"
      }

      field_to_match {
        data = "referer"
        type = "HEADER"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `default_action` - (Required) The action that you want AWS WAF Regional to take when a request doesn't match the criteria in any of the rules that are associated with the web ACL.
- `metric_name` - (Required) The name or description for the Amazon CloudWatch metric of this web ACL.
- `name` - (Required) The name or description of the web ACL.
- `logging_configuration` - (Optional) Configuration block to enable WAF logging. Detailed below.
- `rule` - (Optional) Set of configuration blocks containing rules for the web ACL. Detailed below.
- `tags` - (Optional) Key-value mapping of resource tags

» `default_action` Configuration Block

- `type` - (Required) Specifies how you want AWS WAF Regional to respond to requests that match the settings in a rule. e.g. `ALLOW`, `BLOCK` or `COUNT`

» `logging_configuration` Configuration Block

- `log_destination` - (Required) Amazon Resource Name (ARN) of Kinesis Firehose Delivery Stream

- `redacted_fields` - (Optional) Configuration block containing parts of the request that you want redacted from the logs. Detailed below.

» `redacted_fields` Configuration Block

- `field_to_match` - (Required) Set of configuration blocks for fields to redact. Detailed below.

» `field_to_match` Configuration Block

Additional information about this configuration can be found in the AWS WAF Regional API Reference.

- `data` - (Optional) When the value of `type` is `HEADER`, enter the name of the header that you want the WAF to search, for example, `User-Agent` or `Referer`. If the value of `type` is any other value, omit `data`.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER` or `METHOD`

» `rule` Configuration Block

Additional information about this configuration can be found in the AWS WAF Regional API Reference.

- `priority` - (Required) Specifies the order in which the rules in a WebACL are evaluated. Rules with a lower value are evaluated before rules with a higher value.
- `rule_id` - (Required) ID of the associated WAF (Regional) rule (e.g. `aws_wafregional_rule`). WAF (Global) rules cannot be used.
- `action` - (Optional) Configuration block of the action that CloudFront or AWS WAF takes when a web request matches the conditions in the rule. Not used if `type` is `GROUP`. Detailed below.
- `override_action` - (Optional) Configuration block of the override the action that a group requests CloudFront or AWS WAF takes when a web request matches the conditions in the rule. Only used if `type` is `GROUP`. Detailed below.
- `type` - (Optional) The rule type, either `REGULAR`, as defined by Rule, `RATE_BASED`, as defined by RateBasedRule, or `GROUP`, as defined by RuleGroup. The default is REGULAR. If you add a RATE_BASED rule, you need to set `type` as `RATE_BASED`. If you add a GROUP rule, you need to set `type` as `GROUP`.

» `action` / `override_action` Configuration Block

- `type` - (Required) Specifies how you want AWS WAF Regional to respond to requests that match the settings in a rule. e.g. `ALLOW`, `BLOCK` or `COUNT`

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `arn` - Amazon Resource Name (ARN) of the WAF Regional WebACL.
- `id` - The ID of the WAF Regional WebACL.

» Import

WAF Regional Web ACL can be imported using the id, e.g.

```
$ terraform import aws_wafregional_web_acl.wafacl a1b2c3d4-d5f6-7777-8888-9999aaaabbbbcccc
```

» Resource: aws_wafregional_web_acl_association

Manages an association with WAF Regional Web ACL.

Note: An Application Load Balancer can only be associated with one WAF Regional WebACL.

» Example Usage

```
resource "aws_wafregional_ipset" "ipset" {
  name = "tfIPSet"

  ip_set_descriptor {
    type  = "IPV4"
    value = "192.0.7.0/24"
  }
}

resource "aws_wafregional_rule" "foo" {
  name        = "tfWAFRule"
  metric_name = "tfWAFRule"

  predicate {
    data_id = "${aws_wafregional_ipset.ipset.id}"
    negated = false
    type    = "IPMatch"
  }
}

resource "aws_wafregional_web_acl" "foo" {
  name        = "foo"
  metric_name = "foo"

  default_action {
    type = "ALLOW"
  }

  rule {
    action {
      type = "BLOCK"
    }

    priority = 1
    rule_id  = "${aws_wafregional_rule.foo.id}"
  }
}

resource "aws_vpc" "foo" {
  cidr_block = "10.1.0.0/16"
}

data "aws_availability_zones" "available" {}

resource "aws_subnet" "foo" {
  vpc_id            = "${aws_vpc.foo.id}"
  cidr_block        = "10.1.1.0/24"
  availability_zone = "${data.aws_availability_zones.available.names[0]}"
}

resource "aws_subnet" "bar" {
  vpc_id            = "${aws_vpc.foo.id}"
  cidr_block        = "10.1.2.0/24"
  availability_zone = "${data.aws_availability_zones.available.names[1]}"
}

resource "aws_alb" "foo" {
  internal = true
  subnets  = ["${aws_subnet.foo.id}", "${aws_subnet.bar.id}"]
}

resource "aws_wafregional_web_acl_association" "foo" {
  resource_arn = "${aws_alb.foo.arn}"
  web_acl_id   = "${aws_wafregional_web_acl.foo.id}"
}
```

» Argument Reference

The following arguments are supported:

- `web_acl_id` - (Required) The ID of the WAF Regional WebACL to create an association.
- `resource_arn` - (Required) ARN of the resource to associate with. For example, an Application Load Balancer or API Gateway Stage.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the association

» Import

WAF Regional Web ACL Association can be imported using their `web_acl_id:resource_arn`, e.g.

```
$ terraform import aws_wafregional_web_acl_association.foo web_acl_id:resource_arn
```

» Resource: aws_wafregional_xss_match_set

Provides a WAF Regional XSS Match Set Resource for use with Application Load Balancer.

» Example Usage

```
resource "aws_wafregional_xss_match_set" "xss_match_set" {
  name = "xss_match_set"

  xss_match_tuple {
    text_transformation = "NONE"

    field_to_match {
      type = "URI"
    }
  }

  xss_match_tuple {
    text_transformation = "NONE"
    field_to_match {
      type = "QUERY_STRING"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the set
- `xss_match_tuple` - (Optional) The parts of web requests that you want to inspect for cross-site scripting attacks.

» Nested fields

» `xss_match_tuple`

- `field_to_match` - (Required) Specifies where in a web request to look for cross-site scripting attacks.
- `text_transformation` - (Required) Which text transformation, if any, to perform on the web request before inspecting the request for cross-site scripting attacks.

» `field_to_match`

- `data` - (Optional) When the value of `type` is `HEADER`, enter the name of the header that you want the WAF to search, for example, `User-Agent` or `Referer`. If the value of `type` is any other value, omit `data`.
- `type` - (Required) The part of the web request that you want AWS WAF to search for a specified string. e.g. `HEADER` or `METHOD`

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the Regional WAF XSS Match Set.

» Import

AWS WAF Regional XSS Match can be imported using the `id`, e.g.

```
$ terraform import aws_wafregional_xss_match_set.example 12345abcde
```

» Resource: aws_workspaces_directory

Provides a directory registration in AWS WorkSpaces Service

» Example Usage

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "private-a" {
  vpc_id = "${aws_vpc.main.id}"
  availability_zone = "us-east-1a"
  cidr_block = "10.0.0.0/24"
}

resource "aws_subnet" "private-b" {
  vpc_id = "${aws_vpc.main.id}"
  availability_zone = "us-east-1b"
  cidr_block = "10.0.1.0/24"
}

resource "aws_directory_service_directory" "main" {
  name = "corp.example.com"
  password = "#S1ncerely"
  size = "Small"
  vpc_settings {
    vpc_id = "${aws_vpc.main.id}"
    subnet_ids = ["${aws_subnet.private-a.id}","${aws_subnet.private-b.id}"]
  }
}

resource "aws_workspaces_directory" "main" {
  directory_id = "${aws_directory_service_directory.main.id}"

  self_service_permissions = {
    increase_volume_size = true
    rebuild_workspace = true
  }
}
```

» Argument Reference

The following arguments are supported:

- directory_id - (Required) The directory identifier for registration in WorkSpaces service.
- subnet_ids - (Optional) The identifiers of the subnets where the directory resides.
- tags – (Optional) A mapping of tags assigned to the WorkSpaces directory.
- self_service_permissions – (Optional) The permissions to enable or disable self-service capabilities.

self_service_permissions supports the following:

- change_compute_type – (Optional) Whether WorkSpaces directory users can change the compute type (bundle) for their workspace. Default false.
- increase_volume_size – (Optional) Whether WorkSpaces directory users can increase the volume size of the drives on their workspace. Default false.
- rebuild_workspace – (Optional) Whether WorkSpaces directory users can rebuild the operating system of a workspace to its original state. Default false.
- restart_workspace – (Optional) Whether WorkSpaces directory users can restart their workspace. Default true.
- switch_running_mode – (Optional) Whether WorkSpaces directory users can switch the running mode of their workspace. Default false.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The WorkSpaces directory identifier.

» Import

Workspaces directory can be imported using the directory ID, e.g.

```
$ terraform import aws_workspaces_directory.main d-4444444444
```

» Resource: aws_worklink_fleet

» Example Usage

Basic usage:

```
resource "aws_worklink_fleet" "example" {
  name = "terraform-example"
}
```

Network Configuration Usage:

```
resource "aws_worklink_fleet" "example" {
  name = "terraform-example"

  network {
    vpc_id             = "${aws_vpc.test.id}"
    subnet_ids         = ["${aws_subnet.test.*.id}"]
    security_group_ids = ["${aws_security_group.test.id}"]
  }
}
```

Identity Provider Configuration Usage:

```
resource "aws_worklink_fleet" "test" {
  name = "tf-worklink-fleet-%s"

  identity_provider {
    type          = "SAML"
    saml_metadata = "${file("saml-metadata.xml")}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A region-unique name for the AMI.
- `audit_stream_arn` - (Optional) The ARN of the Amazon Kinesis data stream that receives the audit events.
- `device_ca_certificate` - (Optional) The certificate chain, including intermediate certificates and the root certificate authority certificate used to issue device certificates.
- `identity_provider` - (Optional) Provide this to allow manage the identity provider configuration for the fleet. Fields documented below.
- `display_name` - (Optional) The name of the fleet.
- `network` - (Optional) Provide this to allow manage the company network configuration for the fleet. Fields documented below.
- `optimize_for_end_user_location` - (Optional) The option to optimize for better performance by routing traffic through the closest AWS Region to users, which may be outside of your home Region. Defaults to `true`.

network requires the following:

NOTE: `network` is cannot removed without forece recreating by `terraform taint`.

- `vpc_id` - (Required) The VPC ID with connectivity to associated websites.

- `subnet_ids` - (Required) A list of subnet IDs used for X-ENI connections from Amazon WorkLink rendering containers.
- `security_group_ids` - (Required) A list of security group IDs associated with access to the provided subnets.

identity_provider requires the following:

NOTE: `identity_provider` is cannot removed without forece recreating by `terraform taint`.

- `type` - (Required) The type of identity provider.
- `saml_metadata` - (Required) The SAML metadata document provided by the customer's identity provider.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ARN of the created WorkLink Fleet.
- `arn` - The ARN of the created WorkLink Fleet.
- `company_code` - The identifier used by users to sign in to the Amazon WorkLink app.
- `created_time` - The time that the fleet was created.
- `last_updated_time` - The time that the fleet was last updated.

» Import

WorkLink can be imported using the ARN, e.g.

```
$ terraform import aws_worklink_fleet.test arn:aws:worklink::123456789012:fleet/example
```

» Resource: aws_worklink_website_certificate_authority_associati

» Example Usage

```
resource "aws_worklink_fleet" "example" {
  name = "terraform-example"
}

resource "aws_worklink_website_certificate_authority_association" "test" {
  fleet_arn   = "${aws_worklink_fleet.test.arn}"
  certificate = "${file("certificate.pem")}"
}
```

» Argument Reference

The following arguments are supported:

- `fleet_arn` - (Required, ForceNew) The ARN of the fleet.
- `certificate` - (Required, ForceNew) The root certificate of the Certificate Authority.
- `display_name` - (Optional, ForceNew) The certificate name to display.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `website_ca_id` - A unique identifier for the Certificate Authority.

» Import

WorkLink Website Certificate Authority can be imported using `FLEET-ARN,WEBSITE-CA-ID`, e.g.

```
$ terraform import aws_worklink_website_certificate_authority_association.example arn:aws:wo
```

» Data Source: aws_workspaces_bundle

Use this data source to get information about a WorkSpaces Bundle.

» Example Usage

```
data "aws_workspaces_bundle" "example" {
  bundle_id = "wsb-b0s22j3d7"
}
```

» Argument Reference

The following arguments are supported:

- `bundle_id` – (Required) The ID of the bundle.

» Attributes Reference

The following attributes are exported:

- `description` – The description of the bundle.
- `name` – The name of the bundle.
- `owner` – The owner of the bundle.
- `compute_type` – The compute type. See supported fields below.
- `root_storage` – The root volume. See supported fields below.
- `user_storage` – The user storage. See supported fields below.

» `compute_type`

- `name` - The name of the compute type.

» `root_storage`

- `capacity` - The size of the root volume.

» `user_storage`

- `capacity` - The size of the user storage.

» Resource: aws_workspaces_directory

Provides a directory registration in AWS WorkSpaces Service

» Example Usage

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "private-a" {
  vpc_id = "${aws_vpc.main.id}"
  availability_zone = "us-east-1a"
  cidr_block = "10.0.0.0/24"
}

resource "aws_subnet" "private-b" {
  vpc_id = "${aws_vpc.main.id}"
  availability_zone = "us-east-1b"
  cidr_block = "10.0.1.0/24"
}

resource "aws_directory_service_directory" "main" {
  name = "corp.example.com"
  password = "#S1ncerely"
  size = "Small"
  vpc_settings {
    vpc_id = "${aws_vpc.main.id}"
    subnet_ids = ["${aws_subnet.private-a.id}","${aws_subnet.private-b.id}"]
  }
}

resource "aws_workspaces_directory" "main" {
  directory_id = "${aws_directory_service_directory.main.id}"

  self_service_permissions = {
    increase_volume_size = true
    rebuild_workspace = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `directory_id` - (Required) The directory identifier for registration in WorkSpaces service.
- `subnet_ids` - (Optional) The identifiers of the subnets where the directory resides.
- `tags` – (Optional) A mapping of tags assigned to the WorkSpaces directory.
- `self_service_permissions` – (Optional) The permissions to enable or disable self-service capabilities.

`self_service_permissions` supports the following:

- `change_compute_type` – (Optional) Whether WorkSpaces directory users can change the compute type (bundle) for their workspace. Default `false`.
- `increase_volume_size` – (Optional) Whether WorkSpaces directory users can increase the volume size of the drives on their workspace. Default `false`.
- `rebuild_workspace` – (Optional) Whether WorkSpaces directory users can rebuild the operating system of a workspace to its original state. Default `false`.

- `restart_workspace` – (Optional) Whether WorkSpaces directory users can restart their workspace. Default `true`.
- `switch_running_mode` – (Optional) Whether WorkSpaces directory users can switch the running mode of their workspace. Default `false`.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The WorkSpaces directory identifier.

» Import

Workspaces directory can be imported using the directory ID, e.g.

```
$ terraform import aws_workspaces_directory.main d-4444444444
```

» Resource: aws_workspaces_ip_group

Provides an IP access control group in AWS WorkSpaces Service

» Example Usage

```
resource "aws_workspaces_ip_group" "contractors" {
  name = "Contractors"
  description = "Contractors IP access control group"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the IP group.
- `description` - (Optional) The description of the IP group.
- `rules` - (Optional) One or more pairs specifying the IP group rule (in CIDR format) from which web requests originate.

» Nested Blocks

» rules

» Arguments

- source - (Required) The IP address range, in CIDR notation, e.g. 10.0.0.0/16
- description - (Optional) The description.

» Attributes Reference

In addition to all arguments above, the following attributes are exported:

- id - The IP group identifier.

» Import

WorkSpaces IP groups can be imported using their GroupID, e.g.

```
$ terraform import aws_workspaces_ip_group.example wsipg-488lrtl3k
```

» Resource: aws_xray_sampling_rule

Creates and manages an AWS XRay Sampling Rule.

» Example Usage

```
resource "aws_xray_sampling_rule" "example" {
  rule_name      = "example"
  priority       = 10000
  version        = 1
  reservoir_size = 1
  fixed_rate     = 0.05
  url_path       = "*"
  host           = "*"
  http_method    = "*"
  service_type   = "*"
  service_name   = "*"
  resource_arn   = "*"

  attributes = {
    Hello = "Tris"
  }
}
```

» Argument Reference

- `rule_name` - (Required) The name of the sampling rule.
- `resource_arn` - (Required) Matches the ARN of the AWS resource on which the service runs.
- `priority` - (Required) The priority of the sampling rule.
- `fixed_rate` - (Required) The percentage of matching requests to instrument, after the reservoir is exhausted.
- `reservoir_size` - (Required) A fixed number of matching requests to instrument per second, prior to applying the fixed rate. The reservoir is not used directly by services, but applies to all services using the rule collectively.
- `service_name` - (Required) Matches the `name` that the service uses to identify itself in segments.
- `service_type` - (Required) Matches the `origin` that the service uses to identify its type in segments.
- `host` - (Required) Matches the hostname from a request URL.
- `http_method` - (Required) Matches the HTTP method of a request.
- `url_path` - (Required) Matches the path from a request URL.
- `version` - (Required) The version of the sampling rule format (`1`)
- `attributes` - (Optional) Matches attributes derived from the request.

» Attributes Reference

In addition to the arguments above, the following attributes are exported:

- `id` - The name of the sampling rule.
- `arn` - The ARN of the sampling rule.

» Import

XRay Sampling Rules can be imported using the name, e.g.

```
$ terraform import aws_xray_sampling_rule.example example
```