Terraform AWS Resource Arguments Cheat Sheet

Terraform AWS Resource Arguments Cheat Sheet

aws_vpc

```
resource "aws_vpc" "main" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true
  instance_tenancy     = "default"
  tags                 = { Name = "main-vpc" }
}
```

aws_subnet

```
resource "aws_subnet" "public" {
  vpc_id                  = aws_vpc.main.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "us-east-1a"
  map_public_ip_on_launch = true
  tags                    = { Name = "public-subnet" }
}
```

aws_instance

```
resource "aws_instance" "web" {
  ami                         = "ami-0abc1234567890"
  instance_type               = "t3.micro"
  subnet_id                   = aws_subnet.public.id
  vpc_security_group_ids      = [aws_security_group.web_sg.id]
  key_name                    = "my-key"
  associate_public_ip_address = true
  tags                        = { Name = "web-server" }
}
```

aws_lb

```
resource "aws_lb" "app_alb" {
  name               = "app-alb"
  internal           = false
  load_balancer_type = "application"
  subnets            = [aws_subnet.public.id]
  security_groups    = [aws_security_group.lb_sg.id]
  tags               = { Name = "app-alb" }
}
```

aws_eks_cluster

Terraform AWS Resource Arguments Cheat Sheet

```
resource "aws_eks_cluster" "eks" {
  name     = "eks-cluster"
  role_arn = aws_iam_role.eks_role.arn
  vpc_config {
    subnet_ids = [aws_subnet.private1.id, aws_subnet.private2.id]
  }
}
```

aws_ecr_repository

```
resource "aws_ecr_repository" "repo" {
  name                 = "my-app"
  image_tag_mutability = "MUTABLE"
  image_scanning_configuration {
    scan_on_push = true
  }
}
```

aws_cloudwatch_log_group

```
resource "aws_cloudwatch_log_group" "log_group" {
  name              = "/aws/my-app"
  retention_in_days = 7
  tags              = { Name = "my-log-group" }
}
```

aws_ecs_service

```
resource "aws_ecs_service" "app" {
  name            = "app-service"
  cluster         = aws_ecs_cluster.main.id
  task_definition = aws_ecs_task_definition.app.arn
  desired_count   = 2
  launch_type     = "FARGATE"
  network_configuration {
    subnets          = [aws_subnet.private1.id]
    assign_public_ip = true
    security_groups  = [aws_security_group.web_sg.id]
  }
  load_balancer {
    target_group_arn = aws_lb_target_group.tg.arn
    container_name   = "app"
    container_port   = 80
  }
}
```

Terraform AWS Resource Arguments Cheat Sheet

Advanced AWS Network Resources

aws_ec2_transit_gateway

```
resource "aws_ec2_transit_gateway" "tgw" {
  description = "Main transit gateway"
  amazon_side_asn = 64512
  auto_accept_shared_attachments = "enable"
  default_route_table_association = "enable"
  default_route_table_propagation = "enable"
  dns_support = "enable"
  vpn_ecmp_support = "enable"
  tags = { Name = "main-tgw" }
}
```

aws_ec2_transit_gateway_vpc_attachment

```
resource "aws_ec2_transit_gateway_vpc_attachment" "example" {
  subnet_ids         = [aws_subnet.private1.id, aws_subnet.private2.id]
  transit_gateway_id = aws_ec2_transit_gateway.tgw.id
  vpc_id             = aws_vpc.main.id
  tags               = { Name = "tgw-attachment" }
}
```

aws_vpn_gateway

```
resource "aws_vpn_gateway" "vpn" {
  vpc_id = aws_vpc.main.id
  tags   = { Name = "vpn-gateway" }
}
```

aws_customer_gateway

```
resource "aws_customer_gateway" "example" {
  bgp_asn    = 65000
  ip_address = "203.0.113.12"
  type       = "ipsec.1"
  tags       = { Name = "customer-gateway" }
}
```

aws_vpn_connection

```
resource "aws_vpn_connection" "vpn_conn" {
  vpn_gateway_id      = aws_vpn_gateway.vpn.id
  customer_gateway_id = aws_customer_gateway.example.id
  type                = "ipsec.1"
```

Terraform AWS Resource Arguments Cheat Sheet

```
  static_routes_only  = true
  tags                = { Name = "vpn-connection" }
}
```

aws_vpc_peering_connection

```
resource "aws_vpc_peering_connection" "peer" {
  vpc_id        = aws_vpc.main.id
  peer_vpc_id   = "vpc-0abcd1234ef567890"
  peer_region   = "us-west-2"
  auto_accept   = false
  tags          = { Name = "cross-region-peering" }
}
```

aws_vpc_endpoint

```
resource "aws_vpc_endpoint" "s3_endpoint" {
  vpc_id       = aws_vpc.main.id
  service_name = "com.amazonaws.${var.region}.s3"
  vpc_endpoint_type = "Gateway"
  route_table_ids = [aws_route_table.public.id]
  tags = { Name = "s3-endpoint" }
}
```

aws_ec2_transit_gateway_peering_attachment

```
resource "aws_ec2_transit_gateway_peering_attachment" "peer_attachment" {
  peer_account_id         = "123456789012"
  peer_region             = "us-west-2"
  peer_transit_gateway_id = "tgw-0abcd123456ef789"
  transit_gateway_id      = aws_ec2_transit_gateway.tgw.id
  tags                    = { Name = "tgw-peering" }
}
```