

» `spotinst__elastigroup__aws`

Provides a Spotinst AWS group resource.

» Example Usage

```
# Create an Elastigroup
resource "spotinst_elastigroup_aws" "default-elastigroup" {

  name          = "default-elastigroup"
  description   = "created by Terraform"
  product       = "Linux/UNIX"

  max_size      = 0
  min_size      = 0
  desired_capacity = 0
  capacity_unit  = "weight"

  region      = "us-west-2"
  subnet_ids  = ["sb-123456", "sb-456789"]

  image_id          = "ami-a27d8fda"
  iam_instance_profile = "iam-profile"
  key_name          = "my-key.ssh"
  security_groups    = ["sg-123456"]
  user_data          = "echo hello world"
  enable_monitoring  = false
  ebs_optimized      = false
  placement_tenancy  = "default"

  instance_types_ondemand      = "m3.2xlarge"
  instance_types_spot          = ["m3.xlarge", "m3.2xlarge"]
  instance_types_preferred_spot = ["m3.xlarge"]

  instance_types_weights = [
    {
      instance_type = "c3.large"
      weight        = 10
    },
    {
      instance_type = "c4.xlarge"
      weight        = 16
    }
  ]
}
```

```

orientation          = "balanced"
fallback_to_ondemand = false

scaling_up_policy {
  policy_name      = "Default Scaling Up Policy"
  metric_name      = "DefaultQueuesDepth"
  statistic        = "average"
  unit            = "none"
  adjustment       = 1
  namespace        = "custom"
  threshold        = 100
  period          = 60
  evaluation_periods = 5
  cooldown         = 300
}

scaling_down_policy {
  policy_name      = "Default Scaling Down Policy"
  metric_name      = "DefaultQueuesDepth"
  statistic        = "average"
  unit            = "none"
  adjustment       = 1
  namespace        = "custom"
  threshold        = 10
  period          = 60
  evaluation_periods = 10
  cooldown         = 300
}

tags = [
{
  key   = "Env"
  value = "production"
},
{
  key   = "Name"
  value = "default-production"
},
{
  key   = "Project"
  value = "app_v2"
}
]

lifecycle {
  ignore_changes = [

```

```

        "desired_capacity",
    ]
}
}

```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The group name.
- **description** - (Required) The group description.
- **product** - (Required) Operation system type. Valid values: "Linux/UNIX", "SUSE Linux", "Windows". For EC2 Classic instances: "Linux/UNIX (Amazon VPC)", "SUSE Linux (Amazon VPC)", "Windows (Amazon VPC)".
- **availability_zones** - (Optional) List of Strings of availability zones. Note: When this parameter is set, **subnet_ids** should be left unused.
- **subnet_ids** - (Optional) List of Strings of subnet identifiers. Note: When this parameter is set, **availability_zones** should be left unused.
- **region** - (Optional) The AWS region your group will be created in. Note: This parameter is required if you specify subnets (through **subnet_ids**). This parameter is optional if you specify Availability Zones (through **availability_zones**).
- **preferred_availability_zones** - The AZs to prioritize when launching Spot instances. If no markets are available in the Preferred AZs, Spot instances are launched in the non-preferred AZs. Note: Must be a sublist of **availability_zones** and **orientation** value must not be "equalAzDistribution".
- **max_size** - (Optional; Required if using scaling policies) The maximum number of instances the group should have at any time.
- **min_size** - (Optional; Required if using scaling policies) The minimum number of instances the group should have at any time.
- **desired_capacity** - (Optional) The desired number of instances the group should have at any time.
- **capacity_unit** - (Optional, Default: "instance") The capacity unit to launch instances by. If not specified, when choosing the weight unit, each instance will weight as the number of its vCPUs.
- **security_groups** - (Required) A list of associated security group IDs.
- **image_id** - (Optional) The ID of the AMI used to launch the instance.

- **iam_instance_profile** - (Optional) The ARN or name of an IAM instance profile to associate with launched instances.
- **key_name** - (Optional) The key name that should be used for the instance.
- **enable_monitoring** - (Optional) Indicates whether monitoring is enabled for the instance.
- **user_data** - (Optional) The user data to provide when launching the instance.
- **shutdown_script** - (Optional) The Base64-encoded shutdown script that executes prior to instance termination, for more information please see: Shutdown Script
- **ebs_optimized** - (Optional) Enable high bandwidth connectivity between instances and AWS's Elastic Block Store (EBS). For instance types that are EBS-optimized by default this parameter will be ignored.
- **placement_tenancy** - (Optional) Enable dedicated tenancy. Note: There is a flat hourly fee for each region in which dedicated tenancy is used.
- **instance_types_ondemand** - (Required) The type of instance determines your instance's CPU capacity, memory and storage (e.g., m1.small, c1.xlarge).
- **instance_types_spot** - (Required) One or more instance types.
- **instance_types_preferred_spot** - (Optional) Prioritize a subset of spot instance types. Must be a subset of the selected spot instance types.
- **instance_types_weights** - (Optional) List of weights per instance type for weighted groups. Each object in the list should have the following attributes:
 - **weight** - (Required) Weight per instance type (Integer).
 - **instance_type** - (Required) Name of instance type (String).
- **fallback_to_ondemand** - (Required) In a case of no Spot instances available, Elastigroup will launch on-demand instances instead.
- **orientation** - (Required, Default: "balanced") Select a prediction strategy. Valid values: "balanced", "costOriented", "equalAzDistribution", "availabilityOriented".
- **spot_percentage** - (Optional; Required if not using **ondemand_count**) The percentage of Spot instances that would spin up from the **desired_capacity** number.
- **ondemand_count** - (Optional; Required if not using **spot_percentage**) Number of on demand instances to launch in the group. All other instances will be spot instances. When this parameter is set the **spot_percentage** parameter is being ignored.

- **draining_timeout** - (Optional) The time in seconds, the instance is allowed to run while detached from the ELB. This is to allow the instance time to be drained from incoming TCP connections before terminating it, during a scale down operation.
- **utilize_reserved_instances** - (Optional) In a case of any available reserved instances, Elastigroup will utilize them first before purchasing Spot instances.
- **health_check_type** - (Optional) The service that will perform health checks for the instance. Valid values: "ELB", "HCS", "TARGET_GROUP", "MLB", "EC2", "MULTAI_TARGET_SET", "MLB_RUNTIME", "K8S_NODE", "NOMAD_NODE", "ECS_CLUSTER_INSTANCE".
- **health_check_grace_period** - (Optional) The amount of time, in seconds, after the instance has launched to start and check its health.
- **health_check_unhealthy_duration_before_replacement** - (Optional) The amount of time, in seconds, that we will wait before replacing an instance that is running and became unhealthy (this is only applicable for instances that were once healthy).
- **tags** - (Optional) A key/value mapping of tags to assign to the resource.
- **elastic_ips** - (Optional) A list of AWS Elastic IP allocation IDs to associate to the group instances.
- **revert_to_spot** - (Optional) Hold settings for strategy correction – replacing On-Demand for Spot instances. Supported Values: "never", "always", "timeWindow"
 - **perform_at** - (Required) In the event of a fallback to On-Demand instances, select the time period to revert back to Spot. Supported Arguments – always (default), timeWindow, never. For timeWindow or never to be valid the group must have availabilityOriented OR persistence defined.
 - **time_windows** - (Optional) Specify a list of time windows for to execute revertToSpot strategy. Time window format: ddd:hh:mm-ddd:hh:mm. Example: Mon:03:00-Wed:02:30

» Load Balancers

- **elastic_load_balancers** - (Optional) List of Elastic Load Balancers names (ELB).
- **target_group_arns** - (Optional) List of Target Group ARNs to register the instances to.
- **multai_target_sets** - (Optional) Set of targets to register.
 - **target_set_id** - (Required) ID of Multai target set.
 - **balancer_id** - (Required) ID of Multai Load Balancer.

Usage:

```
elastic_load_balancers = ["bal15", "bal12"]
target_group_arns = ["tg-arn"]
multitarget_target_sets = [{
    target_set_id = "ts-123",
    balancer_id   = "bal-123"
},
{
    target_set_id = "ts-234",
    balancer_id   = "bal-234"
}]
```

» Signals

Each `signal` supports the following:

- **name** - (Required) The name of the signal defined for the group. Valid Values: "INSTANCE_READY", "INSTANCE_READY_TO_SHUTDOWN"
- **timeout** - (Optional) The signals defined timeout- default is 40 minutes (1800 seconds).

Usage:

```
signal = {
    name      = "INSTANCE_READY_TO_SHUTDOWN"
    timeout = 100
}
```

» Scheduled Tasks

Each `scheduled_task` supports the following:

- **task_type** - (Required) The task type to run. Supported task types are: "scale", "backup_ami", "roll", "scaleUp", "percentageScaleUp", "scaleDown", "percentageScaleDown", "statefulUpdateCapacity".
- **cron_expression** - (Optional; Required if not using **frequency**) A valid cron expression. The cron is running in UTC time zone and is in Unix cron format.
- **start_time** - (Optional; Format: ISO 8601) Set a start time for one time tasks.
- **frequency** - (Optional; Required if not using **cron_expression**) The recurrence frequency to run this task. Supported values are "hourly", "daily", "weekly" and "continuous".
- **scale_target_capacity** - (Optional) The desired number of instances the group should have.

- `scale_min_capacity` - (Optional) The minimum number of instances the group should have.
- `scale_max_capacity` - (Optional) The maximum number of instances the group should have.
- `is_enabled` - (Optional, Default: `false`) Setting the task to being enabled or disabled. Valid values: `true`, `false`.
- `target_capacity` - (Optional; Only valid for `statefulUpdateCapacity`) The desired number of instances the group should have.
- `min_capacity` - (Optional; Only valid for `statefulUpdateCapacity`) The minimum number of instances the group should have.
- `max_capacity` - (Optional; Only valid for `statefulUpdateCapacity`) The maximum number of instances the group should have.
- `batch_size_percentage` - (Optional; Required when the `task_type` is `"roll"`.) The percentage size of each batch in the scheduled deployment roll.
- `grace_period` - (Optional) The period of time (seconds) to wait before checking a batch's health after it's deployment.

Usage:

```
scheduled_task = [{
    task_type           = "backup_ami"
    cron_expression     = ""
    start_time          = "1970-01-01T01:00:00Z"
    frequency           = "hourly"
    scale_target_capacity = 5
    scale_min_capacity  = 0
    scale_max_capacity  = 10
    is_enabled          = false
    target_capacity     = 5
    min_capacity        = 0
    max_capacity        = 10
    batch_size_percentage = 33
    grace_period        = 300
}]
```

» Scaling Policies

Each `scaling_*_policy` supports the following:

- `namespace` - (Required) The namespace for the alarm's associated metric.
- `metric_name` - (Required) The name of the metric, with or without spaces.
- `threshold` - (Required) The value against which the specified statistic is compared.
- `policy_name` - (Required) The name of the policy.

- **statistic** - (Optional, Default: `"average"`) The metric statistics to return. For information about specific statistics go to Statistics in the Amazon CloudWatch Developer Guide.
- **unit** - (Required) The unit for the alarm's associated metric. Valid values: `"percent", "seconds", "microseconds", "milliseconds", "bytes", "kilobytes", "megabytes", "gigabytes", "terabytes", "bits", "kilobits", "megabits", "gigabits", "terabits", "count", "bytes/second", "kilobytes/second", "megabytes/second", "gigabytes/second", "terabytes/second", "bits/second", "kilobits/second", "megabits/second", "gigabits/second", "terabits/second", "count/second", "none"`.
- **period** - (Optional, Default: 300) The granularity, in seconds, of the returned datapoints. Period must be at least 60 seconds and must be a multiple of 60.
- **evaluation_periods** - (Optional, Default: 1) The number of periods over which data is compared to the specified threshold.
- **cooldown** - (Optional, Default: 300) The amount of time, in seconds, after a scaling activity completes and before the next scaling activity can start. If this parameter is not specified, the default cooldown period for the group applies.
- **dimensions** - (Optional) A mapping of dimensions describing qualities of the metric.
- **operator** - (Optional, Scale Up Default: `gte`, Scale Down Default: `lte`) The operator to use in order to determine if the scaling policy is applicable. Valid values: `"gt", "gte", "lt", "lte"`.
- **source** - (Optional) The source of the metric. Valid values: `"cloudWatch", "spectrum"`.
- **action_type** - (Optional; if not using `min_target_capacity` or `max_target_capacity`) The type of action to perform for scaling. Valid values: `"adjustment", "percentageAdjustment", "setMaxTarget", "setMinTarget", "updateCapacity"`.

If you do not specify an action type, you can only use `adjustment`, `minTargetCapacity`, `maxTargetCapacity`. While using `action_type`, please also set the following:

When using `adjustment` – set the field `adjustment`
 When using `percentageAdjustment` – set the field `adjustment`
 When using `setMaxTarget` – set the field `max_target_capacity`
 When using `setMinTarget` – set the field `min_target_capacity`
 When using `updateCapacity` – set the fields `minimum`, `maximum`, and `target`

- **adjustment** - (Optional; if not using `min_target_capacity` or `max_target_capacity`;) The number of instances to add/remove

to/from the target capacity when scale is needed. Can be used as advanced expression for scaling of instances to add/remove to/from the target capacity when scale is needed. You can see more information here: [Advanced expression](#). Example value: "MAX(currCapacity / 5, value * 10)"

- **min_target_capacity** - (Optional; if not using `adjustment`; available only for scale up). The number of the desired target (and minimum) capacity
- **max_target_capacity** - (Optional; if not using `adjustment`; available only for scale down). The number of the desired target (and maximum) capacity
- **minimum** - (Optional; if using `updateCapacity`) The minimal number of instances to have in the group.
- **maximum** - (Optional; if using `updateCapacity`) The maximal number of instances to have in the group.
- **target** - (Optional; if using `updateCapacity`) The target number of instances to have in the group.

Usage:

```
scaling_up_policy = [{
    policy_name = "policy-name"
    metric_name = "CPUUtilization"
    namespace   = "AWS/EC2"
    source       = ""
    statistic    = "average"
    unit         = ""
    cooldown     = 60

    dimensions = {
        name = "name-1"
        value = "value-1"
    }

    threshold      = 10
    operator       = "gt"
    evaluation_periods = 10
    period         = 60

    // === MIN TARGET =====
    action_type     = "setMinTarget"
    min_target_capacity = 1
    // =====
```

```

// === ADJUSTMENT =====
# action_type = "adjustment"
# action_type = "percentageAdjustment"
# adjustment = "MAX(5,10)"
// =====

// === UPDATE CAPACITY =====
# action_type = "updateCapacity"
# minimum = 0
# maximum = 10
# target = 5
// =====

}]

scaling_target_policy = [{
    policy_name = ""
    metric_name = ""
    namespace = ""
    source = ""
    statistic = ""
    unit = ""
    cooldown = 10
    target = 1

    dimensions = [{
        name = ""
        value = ""
    }]
}]

```

» Network Interfaces

Each of the `network_interface` attributes controls a portion of the AWS Instance's "Elastic Network Interfaces". It's a good idea to familiarize yourself with AWS's Elastic Network Interfaces docs to understand the implications of using these attributes.

- `network_interface_id` - (Optional) The ID of the network interface.
- `device_index` - (Required) The index of the device on the instance for the network interface attachment.
- `description` - (Required) The description of the network interface.
- `private_ip_address` - (Optional) The private IP address of the network interface.
- `delete_on_termination` - (Optional) If set to true, the interface is deleted when the instance is terminated.

- `secondary_private_ip_address_count` - (Optional) The number of secondary private IP addresses.
- `associate_public_ip_address` - (Optional) Indicates whether to assign a public IP address to an instance you launch in a VPC. The public IP address can only be assigned to a network interface for `eth0`, and can only be assigned to a new network interface, not an existing one.

Usage:

```
network_interface = [{
    network_interface_id      = ""
    device_index              = 1
    description                = "nic description in here"
    private_ip_address        = "1.1.1.1"
    delete_on_termination    = false
    secondary_private_ip_address_count = 1
    associate_public_ip_address = true
}]
```

» 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.

Each `efs_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, Default: `"standard"`) The type of volume. Can be `"standard"`, `"gp2"`, `"io1"`, `"st1"` or `"sc1"`.
- `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.
- `encrypted` - (Optional) Enables EBS encryption on the volume.
- `kms_key_id` - (Optional) ID for a user managed CMK under which the EBS Volume is encrypted

Modifying any `efs_block_device` currently requires resource replacement.

Usage:

```
efs_block_device = [{
    device_name      = "/dev/sdb"
    snapshot_id      = ""
}]
```

```

        volume_type      = "gp2"
        volume_size      = 8
        iops              = 1
        delete_on_termination = true
        encrypted         = false
        kms_key_id        = "kms-key-01"
    },
    {
        device_name       = "/dev/sdc"
        snapshot_id       = ""
        volume_type       = "gp2"
        volume_size       = 8
        iops              = 1
        delete_on_termination = true
        encrypted         = true
        kms_key_id        = "kms-key-02"
    }
}]

```

Each `ephemeral_block_device` supports the following:

- `device_name` - (Required) The name of the block device to mount on the instance.
- `virtual_name` - (Required) The Instance Store Device Name (e.g. "ephemeral0").

Usage:

```

ephemeral_block_device = [{
    device_name = "/dev/xvdc"
    virtual_name = "ephemeral0"
}]

```

» Stateful

We support instance persistence via the following configurations. all values are boolean. For more information on instance persistence please see: Stateful configuration

- `persist_root_device` - (Optional) Boolean, should the instance maintain its root device volumes.
- `persist_block_devices` - (Optional) Boolean, should the instance maintain its Data volumes.
- `persist_private_ip` - (Optional) Boolean, should the instance maintain its private IP.
- `block_devices_mode` - (Optional) String, determine the way we attach the data volumes to the data devices, possible values: "**reattach**" and "**onLaunch**" (default is onLaunch).

- **private_ips** - (Optional) List of Private IPs to associate to the group instances.(e.g. "172.1.1.0"). Please note: This setting will only apply if `persistence.persist_private_ip` is set to true.

Usage:

```
persist_root_device   = false
persist_block_devices = false
persist_private_ip    = true
block_devices_mode    = "onLaunch"
private_ips           = ["1.1.1.1", "2.2.2.2"]
```

» Stateful Deallocation

- **stateful_deallocation** - (Optional)
 - **should_delete_images** - (Optional) For stateful groups: remove persistent images.
 - **should_delete_network_interfaces** - (Optional) For stateful groups: remove network interfaces.
 - **should_delete_volumes** - (Optional) For stateful groups: remove persistent volumes.
 - **should_delete_snapshots** - (Optional) For stateful groups: remove snapshots.

Usage:

```
stateful_deallocation = {
  should_delete_images           = false
  should_delete_network_interfaces = false
  should_delete_volumes         = false
  should_delete_snapshots       = false
}
```

» Health Check

- **health_check_type** - (Optional) The service that will perform health checks for the instance. Supported values : "ELB", "HCS", "TARGET_GROUP", "CUSTOM", "K8S_NODE", "MLB", "EC2", "MULTAI_TARGET_SET", "MLB_RUNTIME", "K8S_NODE", "NOMAD_NODE", "ECS_CLUSTER_INSTANCE".
- **health_check_grace_period** - (Optional) The amount of time, in seconds, after the instance has launched to starts and check its health
- **health_check_unhealthy_duration_before_replacement** - (Optional) The amount of time, in seconds, that we will wait before replacing an instance that is running and became unhealthy (this is only applicable for instances that were once healthy)

Usage:

```
health_check_type           = "ELB"
health_check_grace_period   = 100
health_check_unhealthy_duration_before_replacement = 120
```

» Third-Party Integrations

- `integration_rancher` - (Optional) Describes the Rancher integration.
 - `master_host` - (Required) The URL of the Rancher Master host.
 - `access_key` - (Required) The access key of the Rancher API.
 - `secret_key` - (Required) The secret key of the Rancher API.

Usage:

```
integration_rancher = {
  master_host = "master_host"
  access_key  = "access_key"
  secret_key  = "secret_key"
}
```

- `integration_ecs` - (Optional) Describes the EC2 Container Service integration.
 - `cluster_name` - (Required) The name of the EC2 Container Service cluster.
 - `autoscale_is_enabled` - (Optional, Default: `false`) Specifies whether the auto scaling feature is enabled.
 - `autoscale_cooldown` - (Optional, Default: 300) The amount of time, in seconds, after a scaling activity completes before any further trigger-related scaling activities can start.
 - `autoscale_is_auto_config` - (Optional, Default: `false`) Enabling the automatic auto-scaler functionality. For more information please see: ECS auto scaler.
 - `autoscale_headroom` - (Optional) Headroom for the cluster.
 - * `cpu_per_unit` - (Optional, Default: 0) Cpu units for compute.
 - * `memory_per_unit` - (Optional, Default: 0) RAM units for compute.
 - * `num_of_units` - (Optional, Default: 0) Amount of units for compute.
 - `autoscale_down` - (Optional) Enabling scale down.
 - * `evaluation_periods` - (Optional, Default: 5) Amount of cooldown evaluation periods for scale down.
 - `autoscale_attributes` - (Optional) A key/value mapping of tags to assign to the resource.

Usage:

```

integration_ecs = {
    cluster_name      = "ecs-cluster"
    autoscale_is_enabled = false
    autoscale_cooldown = 300

    autoscale_headroom = {
        cpu_per_unit      = 1024
        memory_per_unit   = 512
        num_of_units      = 2
    }

    autoscale_down = {
        evaluation_periods = 300
    }

    autoscale_attributes = [{
        key   = "test.ecs.key"
        value = "test.ecs.value"
    }]
}

```

- **integration_codedeploy** - (Optional) Describes the Code Deploy integration.
 - **cleanup_on_failure** - (Optional) Cleanup automatically after a failed deploy.
 - **terminate_instance_on_failure** - (Optional) Terminate the instance automatically after a failed deploy.
 - **deployment_groups** - (Optional) Specify the deployment groups details.
 - * **application_name** - (Optional) The application name.
 - * **deployment_group_name** - (Optional) The deployment group name.

Usage:

```

integration_codedeploy = {
    cleanup_on_failure      = false
    terminate_instance_on_failure = false

    deployment_groups = {
        application_name      = "my-app"
        deployment_group_name = "my-group"
    }
}

```

- **integration_route53** - (Optional) Describes the Route53 integration.
 - **domains** - (Required) Collection of one or more domains to register.

- * **hosted_zone_id** - (Required) The id associated with a hosted zone.
- * **record_sets** - (Required) Collection of records containing authoritative DNS information for the specified domain name.
 - **name** - (Required) The record set name.
 - **use_public_ip** - (Optional, Default: **false**) - Designates if the IP address should be exposed to connections outside the VPC.

Usage:

```
integration_route53 = {
  domains = {
    hosted_zone_id = "zone-id"

    record_sets = {
      name          = "foo.example.com"
      use_public_ip = true
    }
  }
}
```

- **integration_docker_swarm** - (Optional) Describes the Docker Swarm integration.
 - **master_host** - (Required) IP or FQDN of one of your swarm managers.
 - **master_port** - (Required) Network port used by your swarm.
 - **autoscale_is_enabled** - (Optional, Default: **false**) Specifies whether the auto scaling feature is enabled.
 - **autoscale_cooldown** - (Optional, Default: 300) The amount of time, in seconds, after a scaling activity completes before any further trigger-related scaling activities can start. Minimum 180, must be a multiple of 60.
 - **autoscale_headroom** - (Optional) An option to set compute reserve for the cluster.
 - * **cpu_per_unit** - (Optional, Default: 0) How much CPU to allocate for headroom unit.
 - * **memory_per_unit** - (Optional, Default: 0) The amount of memory in each headroom unit. Measured in MiB.
 - * **num_of_units** - (Optional, Default: 0) How many units to allocate for headroom unit.
 - **autoscale_down** - (Optional) Setting for scale down actions.
 - * **evaluation_periods** - (Optional, Default: 5) Number of periods over which data is compared. Minimum 3, Measured in consecutive minutes.

Usage:


```

integration_docker_swarm = {
    master_host      = "10.10.10.10"
    master_port      = 2376
    autoscale_is_enabled = true
    autoscale_cooldown = 180

    autoscale_headroom = {
        cpu_per_unit    = 2048
        memory_per_unit = 2048
        num_of_units    = 1
    }

    autoscale_down = {
        evaluation_periods = 3
    }
}

```

- **integration_kubernetes** - (Optional) Describes the Kubernetes integration.
 - **integration_mode** - (Required) Valid values: "saas", "pod".
 - **cluster_identifier** - (Required; if using integration_mode as pod)
 - **api_server** - (Required; if using integration_mode as saas)
 - **token** - (Required; if using integration_mode as saas) Kubernetes Token
 - **autoscale_is_enabled** - (Optional, Default: **false**) Specifies whether the auto scaling feature is enabled.
 - **autoscale_is_auto_config** - (Optional, Default: **false**) Enabling the automatic k8s auto-scaler functionality. For more information please see: Kubernetes auto scaler.
 - **autoscale_cooldown** - (Optional, Default: 300) The amount of time, in seconds, after a scaling activity completes before any further trigger-related scaling activities can start.
 - **autoscale_headroom** - (Optional) An option to set compute reserve for the cluster.
 - * **cpu_per_unit** - (Optional, Default: 0) How much CPU to allocate for headroom unit.
 - * **memory_per_unit** - (Optional, Default: 0) How much Memory allocate for headroom unit.
 - * **num_of_units** - (Optional, Default: 0) How many units to allocate for headroom unit.
 - **autoscale_down** - (Optional) Setting for scale down actions.
 - * **evaluation_periods** - (Optional, Default: 5) How many evaluation periods should accumulate before a scale down action takes place.
 - **autoscale_labels** - (Optional) A key/value mapping of tags to assign to the resource.

Usage:

```
integration_kubernetes = {
  integration_mode      = "pod"
  cluster_identifier    = "my-identifier.ek8s.com"

  // === SAAS =====
  # integration_mode    = "saas"
  # api_server          = "https://api.my-identifier.ek8s.com/api/v1/namespaces/kube-system/s
  # token               = "top-secret"
  // =====

  autoscale_is_enabled    = false
  autoscale_is_auto_config = false
  autoscale_cooldown      = 300

  autoscale_headroom = {
    cpu_per_unit      = 1024
    memory_per_unit   = 512
    num_of_units      = 1
  }

  autoscale_down = {
    evaluation_periods = 300
  }

  autoscale_labels = [{
    key   = "test.k8s.key"
    value = "test.k8s.value"
  }]
}
```

- **integration_nomad** - (Optional) Describes the Nomad integration.
 - **master_host** - (Required) The URL for the Nomad master host.
 - **master_port** - (Required) The network port for the master host.
 - **acl_token** - (Required) Nomad ACL Token
 - **autoscale_is_enabled** - (Optional, Default: **false**) Specifies whether the auto scaling feature is enabled.
 - **autoscale_cooldown** - (Optional, Default: 300) The amount of time, in seconds, after a scaling activity completes before any further trigger-related scaling activities can start.
 - **autoscale_headroom** - (Optional) An option to set compute reserve for the cluster.
 - * **cpu_per_unit** - (Optional, Default: 0) How much CPU (MHz) to allocate for headroom unit.
 - * **memory_per_unit** - (Optional, Default: 0) How much Memory

- allocate for headroom unit.
- * `num_of_units` - (Optional, Default: 0) How many units of headroom to allocate.
- `autoscale_down` - (Optional) Settings for scale down actions.
- * `evaluation_periods` - (Optional, Default: 5) How many evaluation periods should accumulate before a scale down action takes place.
- `autoscale_constraints` - (Optional) A key/value mapping of tags to assign to the resource.

Usage:

```
integration_nomad = {
  master_host      = "my-nomad-host"
  master_port      = 9000
  acl_token        = "top-secret"
  autoscale_is_enabled = false
  autoscale_cooldown = 300

  autoscale_headroom = {
    cpu_per_unit    = 1024
    memory_per_unit = 512
    num_of_units    = 2
  }

  autoscale_down = {
    evaluation_periods = 300
  }

  autoscale_constraints = [{
    key   = "test.nomad.key"
    value = "test.nomad.value"
  }]
}
```

- `integration_mesosphere` - (Optional) Describes the Mesosphere integration.
- `api_server` - (Optional) The public IP of the DC/OS Master.

Usage:

```
integration_mesosphere = {
  api_server = ""
}
```

- `integration_multai_runtime` - (Optional) Describes the Multai Runtime integration.
- `deployment_id` - (Optional) The deployment id you want to get

Usage:

```
integration_multai_runtime = {  
  deployment_id = ""  
}
```

- `integration_gitlab` - (Optional) Describes the Gitlab integration.
 - `runner` - (Optional) Settings for Gitlab runner.
 - * `is_enabled` - (Optional, Default: `false`) Specifies whether the integration is enabled.

Usage:

```
integration_gitlab = {  
  runner = {  
    is_enabled = true  
  }  
}
```

» Update Policy

- `update_policy` - (Optional)
 - `should_resume_stateful` - (Required) This will apply resuming action for Stateful instances in the Elastigroup upon scale up or capacity changes. Example usage will be for Elastigroups that will have scheduling rules to set a target capacity of 0 instances in the night and automatically restore the same state of the instances in the morning.
 - `should_roll` - (Required) Sets the enablement of the roll option.
 - `roll_config` - (Required) While used, you can control whether the group should perform a deployment after an update to the configuration.
 - * `batch_size_percentage` - (Required) Sets the percentage of the instances to deploy in each batch.
 - * `health_check_type` - (Optional) Sets the health check type to use. Valid values: "EC2", "ECS_CLUSTER_INSTANCE", "ELB", "HCS", "MLB", "TARGET_GROUP", "MULTAI_TARGET_SET", "NONE".
 - * `grace_period` - (Optional) Sets the grace period for new instances to become healthy.

```
update_policy = {  
  should_resume_stateful = false  
  should_roll            = false  
  
  roll_config = {  
    batch_size_percentage = 33  
  }  
}
```

```

        health_check_type    = "ELB"
        grace_period         = 300
    }
}

```

» Attributes Reference

The following attributes are exported:

- `id` - The group ID.

» `spotinst_subscription`

Provides a Spotinst subscription resource.

» Example Usage

```

# Create a Subscription
resource "spotinst_subscription" "default-subscription" {

    resource_id = "${spotinst_elastigroup_aws.my-eg.id}"
    event_type  = "AWS_EC2_INSTANCE_LAUNCH"
    protocol    = "http"
    endpoint    = "http://endpoint.com"

    format = {
        event          = "%event%"
        instance_id    = "%instance-id%"
        resource_id     = "%resource-id%"
        resource_name  = "%resource-name%"
        tags            = "foo,baz,baz"
    }
}

```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) Spotinst Resource ID (Elastigroup ID).
- `event_type` - (Required) The event to send the notification when triggered.
Valid values: "AWS_EC2_INSTANCE_TERMINATE", "AWS_EC2_INSTANCE_TERMINATED",
"AWS_EC2_INSTANCE_LAUNCH", "AWS_EC2_INSTANCE_UNHEALTHY_IN_ELB",

"GROUP_ROLL_FAILED", "GROUP_ROLL_FINISHED", "CANT_SCALE_UP_GROUP_MAX_CAPACITY",
"GROUP_UPDATED", "AWS_EC2_CANT_SPIN_OD", "AWS_EMR_PROVISION_TIMEOUT",
"AWS_EC2_INSTANCE_READY_SIGNAL_TIMEOUT".

- **protocol** - (Required) The protocol to send the notification. Valid values: "http", "https", "email", "email-json", "aws-sns".
- **endpoint** - (Required) The endpoint the notification will be sent to: url in case of "http"/"https", email address in case of "email"/"email-json", sns-topic-arn in case of "aws-sns".
- **format** - (Optional) The format of the notification content (JSON Format - Key+Value). Valid values: "%instance-id%", "%event%", "%resource-id%", "%resource-name%".

» Attributes Reference

The following attributes are exported:

- **id** - The subscription ID.