» spotinst_aws_group

Provides a Spotinst AWS group resource.

» Example Usage

```
# Create an AWS group
resource "spotinst_aws_group" "workers" {
  name = "workers-group"
  description = "created by Terraform"
  product = "Linux/UNIX"

  capacity {
    target = 50
    minimum = 25
    maximum = 100
  }

  strategy {
    risk = 100
  }

  scheduled_task {
    task_type = "scale"
    cron_expression = "0 5 * * 0-4"
    scale_target_capacity = 80
  }

  scheduled_task {
    task_type = "backup_ami"
    frequency = "hourly"
  }

  instance_types {
    ondemand = "c3.large"
    spot = ["m3.large", "m4.large", "c3.large", "c4.large"]
  }

  availability_zone {
    name = "us-west-2b"
    subnet_id = "subnet-7bbbf51e"
  }

  launch_specification {
```

```
  monitoring = false
  image_id = "ami-f0091d91"
  key_pair = "pemfile"
  security_group_ids = ["default", "allow-ssh"]
}

tags {
  foo = "bar"
  bar = "baz"
}

scaling_up_policy {
  policy_name = "Scaling Policy 1"
  metric_name = "CPUUtilization"
  statistic = "average"
  unit = "percent"
  threshold = 80
  adjustment = 1
  namespace = "AWS/EC2"
  period = 300
  evaluation_periods = 2
  cooldown = 300
}

scaling_down_policy {
  policy_name = "Scaling Policy 2"
  metric_name = "CPUUtilization"
  statistic = "average"
  unit = "percent"
  threshold = 40
  adjustment = 1
  namespace = "AWS/EC2"
  period = 300
  evaluation_periods = 2
  cooldown = 300
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The group description.
- `description` - (Optional) The group description.
- `product` - (Required) Operation system type.

- `capacity` - (Required) The group capacity. Only a single block is allowed.
  - `target` - (Required) The desired number of instances the group should have at any time.
  - `minimum` - (Optional; Required if using scaling policies) The minimum number of instances the group should have at any time.
  - `maximum` - (Optional; Required if using scaling policies) The maximum number of instances the group should have at any time.
- `strategy` - (Required) This determines how your group request is fulfilled from the possible On-Demand and Spot pools selected for launch. Only a single block is allowed.
  - `risk` - (Optional; Required if not using `ondemand_count`) The percentage of Spot instances that would spin up from the `capcity.target` number.
  - `ondemand_count` - (Optional; Required if not using `risk`) Number of on demand instances to launch in the group. All other instances will be spot instances. When this parameter is set the "risk" parameter is being ignored.
  - `availability_vs_cost` - (Optional) The percentage of Spot instances that would spin up from the `capcity.target` number.
  - `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.
- `instance_types` - The type of instance determines your instance's CPU capacity, memory and storage (e.g., m1.small, c1.xlarge).
  - `ondemand` - (Required) The base instance type.
  - `spot` - (Required) One or more instance types.
- `launch_specification` - (Required) Describes the launch specification for an instance.
  - `image_id` - (Required) The ID of the AMI used to launch the instance.
  - `key_pair` - (Optional) The key name that should be used for the instance.
  - `security_group_ids` - (Optional) A list of associated security group IDS.
  - `monitoring` - (Optional) Indicates whether monitoring is enabled for the instance.
  - `user_data` - (Optional) The user data to provide when launching the instance.
  - `iam_instance_profile` - (Optional) The ARN of an IAM instance profile to associate with launched instances.

  - `load_balancer_names` - (Optional) Registers each instance with the specified Elastic Load Balancers.

- `tags` - (Optional) A 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.

» Availability Zone

Each `availability_zone` supports the following:

- `name` - The name of the availability zone.
- `subnet_id` - (Optional) A specific subnet ID within the given availability zone. If not specified, the default subnet will be used.

» Scheduled Tasks

Each `scheduled_task` supports the following:

- `task_type` - (Required) The task type to run. Supported task types are `scale` and `backup_ami`.
- `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.
- `frequency` - (Optional; Required if not using `cron_expression`) The recurrence frequency to run this task. Supported values are `hourly`, `daily` and `weekly`.
- `scale_target_capcity` - (Optional) The desired number of instances the group should have.
- `scale_min_capcity` - (Optional) The minimum number of instances the group should have.
- `scale_max_capcity` - (Optional) The maximum number of instances the group should have.

» 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` - (Optional) The name of the policy.

- `statistic` - (Optional) The metric statistics to return. For information about specific statistics go to Statistics in the Amazon CloudWatch Developer Guide.
- `unit` - (Optional) The unit for the alarm's associated metric.
- `adjustment` - (Optional) The number of instances to add/remove to/from the target capacity when scale is needed.
- `period` - (Optional) 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) The number of periods over which data is compared to the specified threshold.
- `cooldown` - (Optional) 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.

» 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` - (Optional) The index of the device on the instance for the network interface attachment.
- `subnet_id` - (Optional) The ID of the subnet associated with the network string.
- `description` - (Optional) The description of the network interface.
- `private_ip_address` - (Optional) The private IP address of the network interface.
- `security_group_ids` - (Optional) The IDs of the security groups for 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.

» 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 `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"`.
- `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.

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"`).

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.

» Third-Party Integrations

- `rancher_integration` - (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.
- `elastic_beanstalk_integration` - (Optional) Describes the Elastic Beanstalk integration.
  - `environment_id` - (Required) The ID of the Elastic Beanstalk environment.
- `nirmata_integration` - (Optional) Describes the Nirmata integration.
  - `api_key` - (Required) The API key of the Nirmata API.

» Attributes Reference

The following attributes are exported:

- `id` - The group ID.