» stackpath_compute_workload

A computing application deployed to StackPath's edge network.

» Example Usage

» Containers

```
resource "stackpath_compute_workload" "my-compute-workload" {
  name = "my-compute-workload"
  slug = "my-compute-workload"

  annotations = {
    # request an anycast IP
    "anycast.platform.stackpath.net" = "true"
  }

  network_interface {
    network = "default"
  }

  container {
    # Name that should be given to the container
    name = "app"

    # Nginx image to use for the container
    image = "nginx:latest"

    # Override the command that's used to execute the container. If this option
    # is not provided the default entrypoint and command defined by the docker
    # image will be used.
    # command = []
    resources {
      requests = {
        "cpu"    = "1"
        "memory" = "2Gi"
      }
    }

    env {
      key   = "VARIABLE_NAME"
      value = "VALUE"
    }
  }
```

```
  target {
    name         = "us"
    min_replicas = 1
    max_replicas = 2
    scale_settings {
      metrics {
        metric = "cpu"
        # Scale up when CPU averages 50%.
        average_utilization = 50
      }
    }
    # Deploy these 1 to 2 instances in Dallas, TX, USA and Amsterdam, NL.
    deployment_scope = "cityCode"
    selector {
      key      = "cityCode"
      operator = "in"
      values   = [
        "DFW", "AMS"
      ]
    }
  }
}
```

» Virtual Machines

```
resource "stackpath_compute_workload" "my-compute-workload" {
  name = "my-compute-workload"
  slug = "my-compute-workload"

  annotations = {
    # request an anycast IP
    "anycast.platform.stackpath.net" = "true"
  }

  network_interface {
    network = "default"
  }

  virtual_machine {
    # Name that should be given to the VM
    name = "app"

    # StackPath image to use for the VM
    image = "stackpath-edge/ubuntu-1804-bionic:v201909061930"
```

```
    # Cloud-init user data.
    #
    # Provide at least a public key so you can SSH into VM instances after
    # they're started. See https://cloudinit.readthedocs.io/en/latest/topics/examples.html
    # for more information.
    user_data = <<EOT
#cloud-config
ssh_authorized_keys:
 - ssh-rsa <your public key>
EOT

    resources {
      requests = {
        "cpu"    = "1"
        "memory" = "2Gi"
      }
    }
  }

  target {
    name         = "us"
    min_replicas = 1
    max_replicas = 2
    scale_settings {
      metrics {
        metric = "cpu"
        # Scale up when CPU averages 50%.
        average_utilization = 50
      }
    }
    # Deploy these 1 to 2 instances in Dallas, TX, USA and Amsterdam, NL.
    deployment_scope = "cityCode"
    selector {
      key      = "cityCode"
      operator = "in"
      values   = [
        "DFW", "AMS"
      ]
    }
  }
}
```

» Argument Reference

- `name` - (Required) A human readable name.
- `slug` - (Required) A programmatic name for the workload. Workload slugs are used to build the workload's instance names and cannot be changed after creation.
- `labels` - (Optional) Key/value pairs of arbitrary label names and values that can be referenced as selectors by network policies.
- `annotations` - (Optional) Key/value pairs that define StackPath-specific workload configuration.
- `network_interface` - (Required) Networks to place the compute instance on. See Network Interfaces below for details.
- `image_pull_credentials` - (Optional) Credentials to pull container images with. See Image Pull Credentials below for details.
- `virtual_machine` - (Optional) Virtual machine configuration. StackPath supports a single virtual machine specification in a workload. At least one of `virtual_machine` or `container` must be provided. See Virtual Machines below for details.
- `container` - (Optional) Container configuration. At least one of `virtual_machine` or `container` must be provided. See Containers below for details.
- `volume_claim` - (Optional) Storage that is mounted to a compute workload's instances. See Volume Claims below for details.
- `target` - (Required) How the compute workload should be deployed across the StackPath edge platform. See Deployment Targets below for details.

» Network Interfaces

`network_interfaces` supports the following arguments:

- `network` - (Required) A name that can be that can be referenced by a selector by network policies. Currently, only the value "default" is supported.

» Image Pull Credentials

`image_pull_credentials` supports the following arguments:

- `docker_registry` - (Required) Authentication configuration needed to pull images from a Docker registry. See Docker Registry Credentials below for details.

» Docker Registry Credentials

`docker_registry` supports the following arguments:

- `server` - (Optional) The address of a Docker registry server. Defaults to "hub.docker.com".
- `username` - (Required) A username to connect the Docker registry.
- `password` - (Required) A password to connect to the Docker registry.
- `email` - (Optional) An email address to use with the Docker registry account.

» Virtual Machines

`virtual_machine` supports the following arguments:

- `name` - (Required) A virtual machine's name.
- `image` - (Required) The disk image to run as a virtual machine.
- `port` - (Optional) Network ports to expose from the virtual machine. Ports can also be used for internal DNS-based service discovery. See Network Ports below for details.
- `liveness_probe` - (Optional) Criteria to determine if the compute workload is online. See Probes below for details.
- `readiness_probe` - (Optional) Criteria to determine if the compute workload is ready to serve requests. See Probes below for details.
- `resources` - (Required) Hardware resources required by the virtual machine. See Resources below for details.
- `volume_mount` - (Optional) Storage volumes to mount in the virtual machine. See Volume Mounts below for details.
- `user_data` - (Optional) Cloud-init user data.

» Containers

`container` supports the following arguments:

- `name` - (Required) A container's name.
- `image` - (Required) A container's image location.
- `command` - (Optional) The command to execute a container.
- `env` - (Optional) Environment variables to set in the container instance. See Environment Variables below for details.
- `port` - (Optional) Networking ports to expose from the container. Ports can also be used for internal DNS-based service discovery. See Network Ports below for details.
- `liveness_probe` - (Optional) Criteria to determine if the compute workload is online. See Probes below for details.
- `readiness_probe` - (Optional) Criteria to determine if the compute workload is ready to serve requests. See Probes below for details.
- `resources` - (Required) Hardware resources required by the container. See Resources below for details.

- `volume_mount` - (Optional) Storage volumes to mount in the container. See Volume Mounts below for details.

» Environment Variables

`env` supports the following arguments:

- `key` - (Required) The environment variable name.
- `value` - (Optional) The environment variable value. One of `value` or `secret_value` must be provided.
- `secret_value` - (Optional) A sensitive environment variable value. This value cannot be read after it is set. One of `value` or `secret_value` must be provided.

» Network Ports

`port` supports the following arguments:

- `name` - (Required) The network port's name.
- `port` - (Required) The network port's number.
- `protocol` - (Optional) The network port's protocol, either "tcp" or "udp". Defaults to "tcp".
- `enable_implicit_network_policy` - (Optional) Whether or not the network port is accessible from the public Internet. Defaults to `false`.

» Volume Claims

`volume_claim` supports the following arguments:

- `name` - (Required) A human readable name.
- `slug` - (Optional) A programmatic slug. Reference this slug when mounting the claim into a workload's instances.
- `resources` - (Required) Hardware resources to allocate to the volume claim. See Resources below for details.

» Probes

`liveness_probe` and `readiness_probe` take the following arguments:

- `http_get` - (Optional) HTTP request information. One of `http_get` or `tcp_socket` must be provided. See HTTP probes below for details
- `tcp_socket` - (Optional) TCP socket information. One of `http_get` or `tcp_socket` must be provided. See TCP probes below for details
- `initial_delay_seconds` - (Optional) The initial delay before the probe starts. Defaults to 0.

- `timeout_seconds` - (Optional) The number of seconds before the probe times out and is considered a failure. Defaults to 10.
- `period` - (Optional) The frequency of the probe. Defaults to 60.
- `success_threshold` - (Required) The minimum consecutive successes required before a probe is considered successful after a failure. This must be 1 for liveness probes.
- `failure_threshold` - (Required) The amount of failures seen before the probe is considered a failure.

» HTTP Probes

`http_get` takes the following arguments:

- `path` - (Optional) The URL path to request from the application. Defaults to "/".
- `port` - (Required) The TCP port the HTTP service listens on.
- `scheme` - (Optional) The URL scheme to query the application with. Defaults to "http".
- `http_headers` - (Optional) HTTP header names and values to send to the HTTP service.

» TCP probes

`tcp_socket` takes the following arguments:

- `port` - (Required) The TCP port number to connect to.

» Resources

`resources` takes the following arguments:

- `requests` - (Required) Key/value pairs of hardware resource types and values.

» Volume Mounts

`volume-mount` takes the following arguments:

- `slug` - (Required) The slug of the volume claim to mount into the workload's instances.
- `mount_path` - (Required) The path the volume is mounted to in a workload's instances.

» Deployment Targets

`target` takes the following arguments:

- `name` - (Required) A human readable name.
- `min_replicas` - (Required) The minimum number of instances that should be deployed to a target.
- `max_replicas` - (Optional) The maximum number of instances that should be deployed to a target.
- `scale_settings` - (Optional) How to auto-scale the number of instances in the deployment target. See Scaling Settings below for details.
- `deployment_scope` - (Optional) Criteria that defines a deployment target. Defaults to "cityCode".
- `selector` - (Required) The value of the deployment scope to deploy to. See Selectors below for details.

» Scaling Settings

`scale_settings` takes the following arguments:

- `metrics` - (Required) Scaling metrics. See Scaling Metrics below for details.

» Scaling Metrics

`metrics` takes the following arguments:

- `metric` - (Required) A hardware metric to use as a scaling basis. Currently, only the "cpu" metric is supported.
- `average_utilization` - (Optional) The `metric`'s average utilization that should trigger scaling. One of `average_utilization` or `average_value` must be provided.
- `average_value` - (Optional) The `metric`'s average value that should trigger scaling. One of `average_utilization` or `average_value` must be provided.

» Selectors

`selector` takes the following arguments:

- `key` - (Required) The name of the data that a selector is based on.
- `operator` - (Required) A logical operator to apply to a selector. Only the "in" operator is supported.
- `values` - (Required) Data values to look for in a label selector.

» Instances

A workload instance is a collection of containers or a virtual machine created based on the template provided in a workload. Instances are accessed via a `stackpath_resource_compute_workload`'s computed `instances` field.

» Example Usage

```
# Output a StackPath compute workload's instances' name, internal IP addresses,
# and status
output "my-compute-workload-instances" {
  value = {
    for instance in stackpath_compute_workload.my-compute-workload.instances:
    instance.name => {
      ip_address = instance.external_ip_address
      phase      = instance.phase
    }
  }
}
```

» Instance Fields

- `name` - (Required) An instance's name. Names are generated from their corresponding workload's slug, followed by a unique hash.
- `metadata` - (Optional) Metadata associated with a running instance, including the workload's `labels` and annotations, both supplied by the user and generated by StackPath.
- `location` - (Optional) The instance's physical location. See Locations below for details.
- `external_ip_address` - (Optional) An IP address bound to the instance.
- `ip_address` - (Optional) An instance's internal IP address.
- `network_interface` - (Optional) A network interface bound to an instance. See Instance Network Interfaces below for details.
- `virtual_machine` - (Optional) An instance's virtual machine specification. An instance has either a `virtual_machine` or `container`.
- `container` - (Optional) An instance's container specification. An instance has either a `virtual_machine` or `container`.
- `phase` - (Optional) An instance's current status, such as "STARTING", "RUNNING", "FAILED", or "STOPPED".
- `reason` - (Optional) A short reason why an instance is in its current phase.
- `message` - (Optional) A longer message with details why an instance is in its current phase.

» Locations

`locaton` has the following fields:

- `name` - (Optional) A location's name.
- `city`- (Optional) A location's city.
- `city_code` - (Optional) A city's IATA code.
- `subdivision` - (Optional) A location's subdivision.
- `subdivision_code` - (Optional) A subdivision's ISO 3166-2 code.
- `country` - (Optional) A location's country.
- `country_code` - (Optional) A country's ISO 3166-1 alpha-2 code.
- `region` - (Optional) A location's region.
- `region_code` - (Optional) A region's GeoIP code.
- `continent` - (Optional) A location's continent.
- `continent_code` - (Optional) A continent's GeoIP code.
- `latitude` - (Optional) A location's latitude coordinate.
- `longitude` - (Optional) A location's longitude coordinate.

» Instance Network Interfaces

`network_interface` has the following fields:

- `network` - (Required) The name of the workload network interface.
- `ip_address` - (Required) A network interface's primary IP address.
- `ip_address_aliases` - (Optional) Additional IP addresses bound to a network interface.
- `gateway` - (Required) A network interface subnet's gateway IP address.

» Import

StackPath compute workloads can be imported by their UUID v4 formatted id.
e.g.

```
$ terraform import stackpath_compute_workload.terraform bdb77768-2938-4ad8-a736-be5290add801
```

» stackpath_compute_network_policy

Network ingress and egress control of StackPath computing workloads.

» Example Usage

```
resource "stackpath_compute_network_policy" "web-server" {
  name        = "Allow HTTP traffic to web servers"
```

```
  slug        = "web-servers-allow-http"
  description = "A network policy that allows HTTP access to instances with the web server r
  priority    = 20000

  # Apply this network policy to every workload instance on the stack with the
  # "web-server" role. Instance selectors can also use any labels present on
  # workloads or instances in the stack.
  instance_selector {
    key      = "role"
    operator = "in"
    values   = ["web-server"]
  }

  # Apply this network policy to specific workload instances. Use the key
  # "workload.platform.stackpath.net/workload-slug" to target instances by slug
  # or use the key "workload.platform.stackpath.net/workload-id" to target
  # instances by ID.
  #
  # Use the priority value 65534 to define multiple workload-specific policies
  # to avoid priority collisions.
  instance_selector {
    key      = "workload.platform.stackpath.net/workload-slug"
    operator = "in"
    values   = ["my-workload-slug"]
  }

  policy_types = ["INGRESS"]

  ingress {
    action      = "ALLOW"
    description = "Allow port 80 traffic from all IPs"
    protocol {
      tcp {
        destination_ports = [80]
      }
    }
    from {
      ip_block {
        cidr = "0.0.0.0/0"
      }
    }
  }
}
```

» Argument Reference

- `name` - (Required) A human readable name.
- `slug` - (Required) A programmatic name for the network policy.
- `description` - (Optional) A brief description.
- `labels` - (Optional) Key/value pairs of arbitrary label names and values that can be referenced as selectors.
- `annotations` - (Optional) Key/value pairs that define StackPath-specific network policy configuration.
- `instance_selector` - (Optional) A compute workload instance that the network policy applies to. A network policy with no selectors applies to all networks and all instances in the stack. See Selectors below for details.
- `network_selector` - (Optional) A network that the network policy applies to. A network policy with no selectors applies to all networks and all instances in the stack. See Selectors below for details.
- `policy_types` - (Required) A list of network policy types, either "INGRESS" and/or "EGRESS".
- `priority` - (Required) A priority value between 1 and 65000. Higher priority network policies override lower priority policies, and priorities must be unique across the stack.
- `egress` - (Optional) Outbound networking information. See Egress below for details.
- `ingress` - (Optional) Inbound networking information. See Ingress below for details.

» Egress

`egress` takes the following arguments:

- `action` - (Required) How a network policy treats outbound traffic, either "ALLOW" or "BLOCK".
- `description` - (Optional) A brief description.
- `protocol` - (Optional) Network protocol specific information. See Network Protocols below for details.
- `to` - (Optional) Allow or block outbound traffic to the specified targets. See Network Selectors below for details.

» Ingress

`ingress` takes the following arguments:

- `action` - (Required) How a network policy treats outbound traffic, either "ALLOW" or "BLOCK".
- `description` - (Optional) A brief description.

- `protocol` - (Optional) Network protocol specific information. See Network Protocols below for details.
- `from` - (Optional) Allow or block inbound traffic from the specified targets. See Network Selectors below for details.

» Network Protocols

`protocol` takes the following arguments:

- `ah` - (Optional) Allow or block the IPSec Authentication Header protocol. This argument block has no configuration.
- `esp` - (Optional) Allow or block the IPSec Encapsulating Security Payload protocol. This argument block has no configuration.
- `gre` - (Optional) Allow or block the Generic Routing Encapsulation protocol. This argument block has no configuration.
- `icmp` - (Optional) Allow or block the Internet Control Message Protocol. This argument block has no configuration.
- `tcp` - (Optional) Allow or block Transmission Control Protocol connections. See Network Ports below for details.
- `udp` - (Optional) Allow or block User Datagram Protocol connections. See Network Ports below for details.
- `tcp_udp` - (Optional) Allow or block both TCP and UDP connections. See Network Ports below for details.

» Network Ports

`tcp`, `udp`, and `tcp_udp` take the following arguments:

- `source_ports` - (Optional) A list of destination ports.
- `destination_ports` - (Optional) A list of destination ports.

» Network Selectors

`to` and `from` take the following arguments:

- `instance_selector` - (Optional) Target the given compute workload instances. See Selectors below for details.
- `network_selector` - (Optional) Target the given networks. See Selectors below for details.
- `ip_block` - (Optional) Target the given IP address blocks. See IP Blocks below for details.

» IP Blocks

`ip_block` takes the following arguments:

- `cidr` - (Required) A CIDR formatted subnet.
- `except` - (Optional) A list of CIDR formatted subnets to exclude from the `cidr` subnet.

» Selectors

`instance_selector` and `network_selector` take the following arguments:

- `key` - (Required) The name of the data that a selector is based on.
- `operator` - (Required) A logical operator to apply to a selector. Only the "in" operator is supported.
- `values` - (Required) Data values to look for in a label selector.

» Import

StackPath compute network policies can be imported by their UUID v4 formatted id. e.g.

```
$ terraform import stackpath_compute_network_policy.terraform bdb77768-2938-4ad8-a736-be529(
```

» stackpath_object_storage_bucket

An S3 compatible object storage bucket deployed to StackPath's edge network.

» Example Usage

```
resource "stackpath_object_storage_bucket" "my-bucket" {
  label = "my-bucket"
  region = "us-west-1"
  visibility = "PRIVATE"
}
```

» Argument Reference

- `label` - (Required) A human readable label for the bucket. Bucket label only supports (a-z, 0-9, -) and must start/end with a letter or number.
- `region` - (Required) Bucket region (us-east-2 us-west-1 eu-central-1)
- `visibility` - (Optional) PRIVATE or PUBLIC, defaults to PRIVATE

» Attributes Reference

- `endpoint_url` - S3 compatible region endpoint for the bucket, e.g. https://s3.us-east-2.stackpathstorage.com

» Import

StackPath object storage buckets can be imported by their UUID v4 formatted id. e.g.

```
$ terraform import stackpath_object_storage_bucket.bucket bdb77768-2938-4ad8-a736-be5290add8
```