Google Cloud Platform

Autoscaling

v 1.0

Infrastructure Automation

Agenda

1. Autoscaling
2. Policies
3. Configuration
4. Lab
5. Review

Autoscaling

- Available as part of the Compute Engine API
- Used to automatically scale number of instances in a *managed instance group* based on workload
  - Helps reduce costs by shutting down instances when not required
- Create one autoscaler per managed instance group
- Autoscalers can be used with zone-based managed instance groups or regional managed instance groups
- Autoscaler is fast, typically ~ 1 min moving window

You can enable autoscaling capabilities for a regional managed instance group. If you enable autoscaling for a regional managed instance group, it behaves as described below:

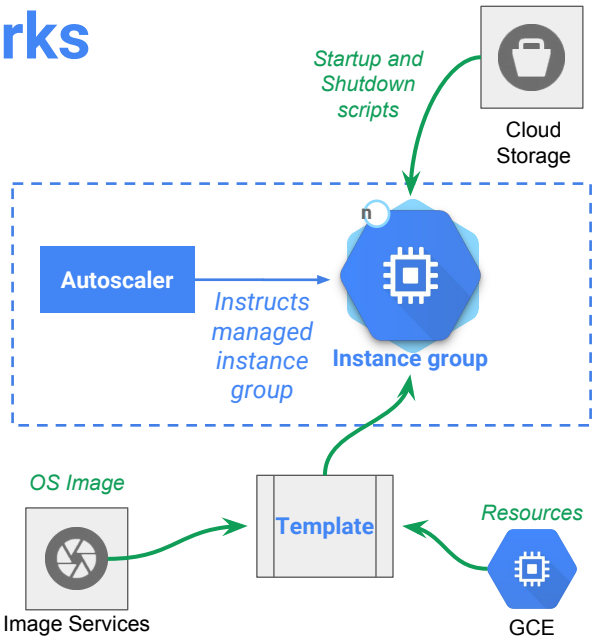
- An autoscaling policy is applied to the group as a whole (not to individual zones). For example, if you enable autoscaler to target 66% CPU utilization, the autoscaler will keep track of all instances in the group to maintain an average 66% utilization across all instances in all zones.
- Autoscaling attempts to spread instances evenly across available zones when possible. In general, the autoscaler keeps zones balanced in size by growing smaller zones and expecting that load will get redirected from bigger zones. We do not recommend configuring a custom load balancer that prefers one zone as this could cause unexpected behavior.
- If a zone experiences a failure, or a group of instances within a zone fails, ⅓ of capacity may be lost but ⅔ of the capacity will remain in the other zones. However, we recommend setting your autoscaling policy to overprovision your autoscaled managed instance group to avoid overloading surviving servers during the time from when a zone is lost until the autoscaler is able to spin up additional machines in the remaining zones.

Speed of our autoscaler and time for booting new instances means that GCP adapts very well to spikes in traffic.

For more information on using autoscalers with regional instance groups, see: https://cloud.google.com/compute/docs/instance-groups/distributing-instances-with-regional-instance-groups#autoscaling_a_regional_managed_instance_group.

How Autoscaling Works

- Autoscaler controls managed instance group
- Adds, removes instances using policies
- Policy includes number of replicas
  - Max number
  - Min number



In the diagram, n is any number of instance replicas based on a template. The template requisitions resources from GCE, identifies an OS image to boot, and starts new VMs. Startup and shutdown scripts can be used for many purposes. However, in cases where speed of autoscaling is required, it is a common practice to "bake" customizations and software installation into the OS Image, and use scripts for graceful entry and exit. For example, a shutdown script can be used to notify a load balancer that the instance is going out of service, and not to send any more traffic.

Agenda

1. Autoscaling
2. Policies
3. Configuration
4. Lab
5. Review

Policies determine behavior

- Policy options
  - Average CPU utilization
    - If average usage of total vCPU cores in instance group exceeds target, autoscaler adds more instances
  - HTTP load balancing serving capacity (defined in the backend service)
    - Maximum CPU utilization
    - Maximum requests per second/instance
  - Stackdriver standard and custom metrics
  - Google Cloud Pub/Sub queuing workload (Alpha)

To create an autoscaler, you must specify the autoscaling policy and a target utilization level that the autoscaler uses to determine when to scale the group. You can choose to scale using the following policies:

- Average CPU utilization
- Stackdriver Monitoring metrics
- HTTP load balancing serving capacity, which can be based on either utilization or requests per second.

Note: Autoscaler does not yet support Google Stackdriver Monitoring V3.

The autoscaler will collect information based on the policy, compare it to your desired target utilization, and determine if it needs to perform scaling.

The target utilization level is the level at which you want to maintain your virtual machine instances. For example, if you scale based on CPU utilization, you can set your target utilization level at 75% and the autoscaler will maintain the CPU utilization of the specified group of instances at or close to 75%. The utilization level for each metric is interpreted differently based on the autoscaling policy.

For more information on the autoscaler, see: https://cloud.google.com/compute/docs/autoscaler/.

Multiple Policies

- Autoscaler allows multiple policies (up to 5)
- Autoscaler handles multiple policies by calculating recommended number of virtual machines for each policy and picking policy that leaves the largest number of virtual machines in the group
  - Ensures enough virtual machines to handle application workloads and allows you to scale applications that have multiple possible bottlenecks

For more information on using multiple policies with an autoscaler, see: https://cloud.google.com/compute/docs/autoscaler/multiple-policies.

Policy Example: CPU Utilization

gcloud

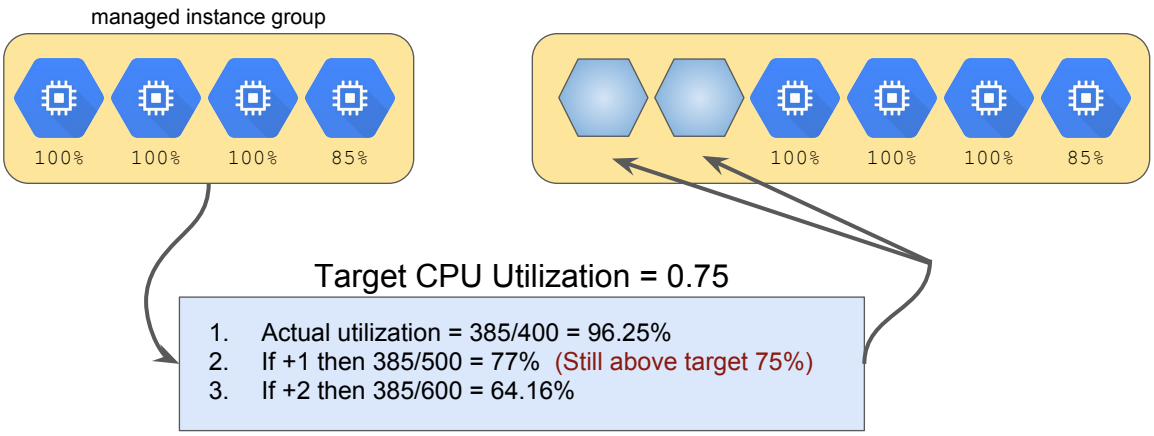
```
# Enable autoscaling for a managed instance group
# using CPU utilization

gcloud compute instance-groups managed set-autoscaling
example-managed-instance-group \
    --max-num-replicas 20 \
    --target-cpu-utilization 0.75 \
    --cool-down-period 90
```


In this example, the command creates an autoscaler that has a target CPU utilization of 75%. Setting a 0.75 target utilization tells the autoscaler to maintain an average usage of 75% among all cores in the instance group. The cool down period is the number of seconds the autoscaler should wait after a virtual machine has started before the autoscaler starts collecting information from it. This accounts for the amount of time it can take for a virtual machine to initialize. During this time, the instance group may exceed the scaling threshold but it won't launch new instances until the starting instance is available for measurement, at which point it will re-evaluate load.

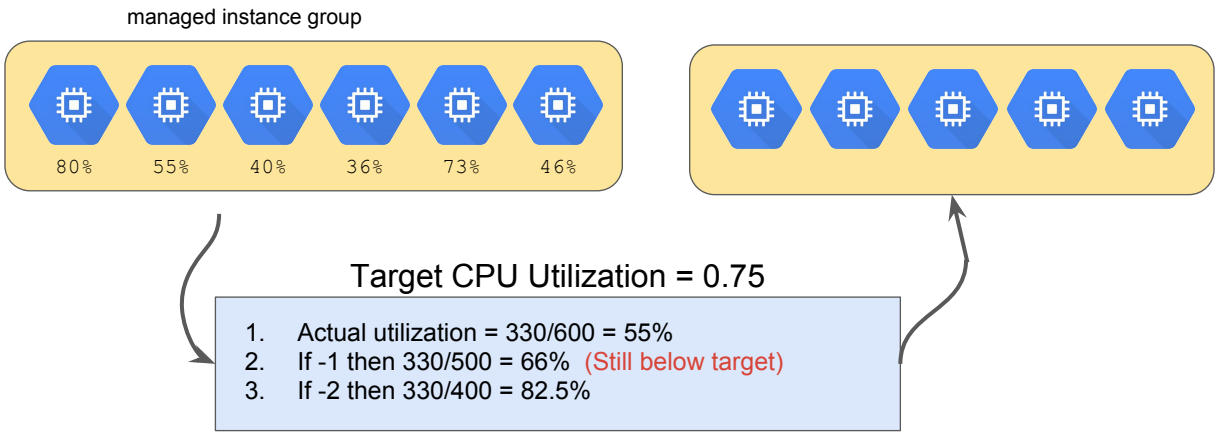
To see a list of parameters for the set-autoscaling subcommand, go to: https://cloud.google.com/sdk/gcloud/reference/compute/instance-groups/managed/set-autoscaling.

Scale-Up Policy Decision



The percentage utilization that an additional VM contributes depends on the size of the group. The 4th VM added to a group offers 25% increase in capacity to the group. The 10th VM added to a group only offers 10% more capacity, even though the VMs are the same size.
In this case shown in the diagram Autoscaler is conservative and rounds up. In other words, it would prefer to start an extra VM that isn't really needed than to possibly run out of capacity,

Scale-Down Policy Decision



In this example, removing one VM doesn't get close enough to the target of 75%. Removing a second VM would exceed the target. Autoscaler behaves conservatively. So it will shut down one VM rather than two VMs. It would prefer underutilization over running out of resource when its needed.

Agenda

1. Autoscaling
2. Policies
3. Configuration
4. Lab
5. Review

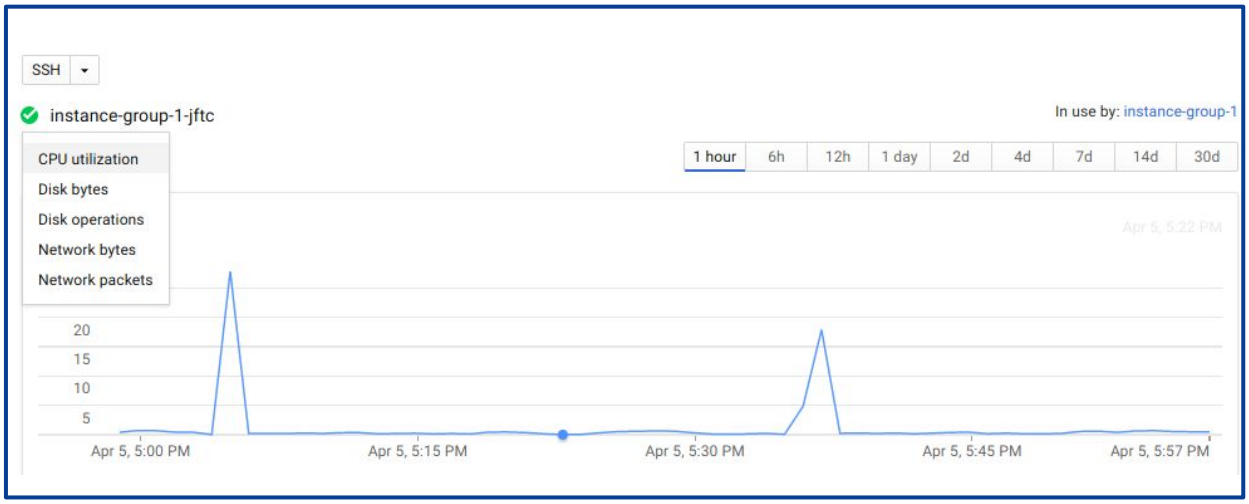
Configuring an Autoscaler

1. Create instance template (including startup, shutdown scripts) to automate tasks that must take place while instance is unattended
   - Software installation
   - Software startup and shutdown
   - Log archiving
2. Create managed instance group
3. Create autoscaler
4. Optionally, define multiple policies for autoscalers

- Autoscaler
  - https://cloud.google.com/compute/docs/autoscaler/
- Managing autoscalers
  - https://cloud.google.com/compute/docs/autoscaler/
- Managing-autoscalers
  - Understanding autoscaler decisions
- https://cloud.google.com/compute/docs/autoscaler/
- understanding-autoscaler-decisions

Advice: start small, test, and build-up to production scale.

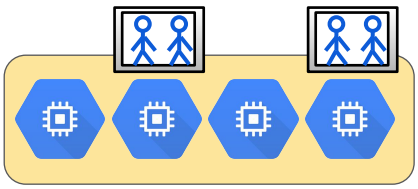
VM Graph helps set CPU utilization



When you click on an instance group (or an individual VM) a graph is presented. The default display is CPU utilization. And you can look at the graph over different time frames. This is very handy for viewing actual traffic, looking for patterns, and determining how best to configure Autoscaling policy to meet changing demand.

Custom Metrics

You can write custom Metrics to the Stackdriver Monitoring service.
You can use custom Metrics from Stackdriver Monitoring for Autoscaling.



Example

- Custom metric: **Game Users**
- Each VM has a capacity for 500 users
  - Number of users may be a better indicator of user experience than CPU load
- Custom Metric from application to Stackdriver
- Autoscaling policy set on Stackdriver monitoring custom metric

https://cloud.google.com/compute/docs/autoscaler/scaling-stackdriver-monitoring-metrics

In this example, the scaling of game servers could be based on CPU load or perhaps network traffic or GPU load.
However, each server only has a capacity for 500 game users. So perhaps a better metric would be number of users.
In this case you could write a custom metric to pass the number of users on a VM from the application to Stackdriver.
Then you could set your Autoscaling policy based on number of users.

Agenda

1. Autoscaling
2. Policies
3. Configuration
4. Lab
5. Review

Lab

In this lab you will configure autoscaling and verify that it works.

Agenda

1. Autoscaling
2. Policies
3. Configuration
4. Lab
5. Review

More...

- Autoscaling
  - https://cloud.google.com/compute/docs/autoscaler/

More to learn on this subject. Here are some suggestions and links.


cloud.google.com