|  |
| --- |
| gcloud config get-value project |
|  |  |
|  | gcloud compute instances list |
|  |  |
|  | gcloud services list |
|  | gcloud services list -h |
|  | gcloud services list --enabled |
|  | gcloud services list --available |
|  |  |
|  | gcloud services list --available | grep compute |
|  | gcloud services -h |
|  |  |
|  | gcloud compute instances list |
|  |  |
|  | gcloud services list |
|  |  |
|  | gcloud compute instances create myvm |
|  | gcloud compute instances delete myvm |
|  | gcloud compute instances list |

Worker Script

|  |
| --- |
| #! /bin/bash |
|  |  |
|  | # |
|  | # Echo commands as they are run, to make debugging easier. |
|  | # GCE startup script output shows up in "/var/log/syslog" . |
|  | # |
|  | set -x |
|  |  |
|  |  |
|  | # |
|  | # Make sure installed packages are up to date with all security patches. |
|  | # |
|  | apt-get update |
|  |  |
|  |  |
|  | # |
|  | # Install Google's Stackdriver logging agent, as per |
|  | # https://cloud.google.com/logging/docs/agent/installation |
|  | # |
|  | curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh |
|  | bash install-logging-agent.sh |
|  |  |
|  |  |
|  | # |
|  | # Install and run the "stress" tool to max the CPU load for a while. |
|  | # |
|  | apt-get install stress |
|  | stress -c 8 -t 120 |
|  |  |
|  |  |
|  | # |
|  | # Report that we're done. |
|  | # |
|  |  |
|  | # Metadata should be set in the "lab-logs-bucket" attribute using the "gs://mybucketname/" format. |
|  | log\_bucket\_metadata\_name=lab-logs-bucket |
|  | log\_bucket\_metadata\_url="http://metadata.google.internal/computeMetadata/v1/instance/attributes/${log\_bucket\_metadata\_name}" |
|  | worker\_log\_bucket=$(curl -H "Metadata-Flavor: Google" "${log\_bucket\_metadata\_url}") |
|  |  |
|  | # We write a file named after this machine. |
|  | worker\_log\_file="machine-$(hostname)-finished.txt" |
|  | echo "Phew! Work completed at $(date)" >"${worker\_log\_file}" |
|  |  |
|  | # And we copy that file to the bucket specified in the metadata. |
|  | echo "Copying the log file to the bucket..." |
|  | gsutil cp "${worker\_log\_file}" "${worker\_log\_bucket} |

GCE Script

|  |  |
| --- | --- |
|  | # Check the elected project |
|  | gcloud config list |
|  |  |
|  | # Show any .ssh folder |
|  | pwd |
|  | ls |
|  | ls -a .ssh |
|  |  |
|  | # Get our bearings in Cloud Shell |
|  | whoami |
|  | hostname |
|  | curl api.ipify.org |
|  |  |
|  | # Check that we have nothing running |
|  | gcloud compute instances list |
|  |  |
|  | # Don't create a default VM |
|  | # Cancel: gcloud compute instances create myhappyvm |
|  |  |
|  | # Look at how to set the machine type |
|  | gcloud compute instances create myhappyvm -h |
|  | gcloud compute instances create myhappyvm --help |
|  | gcloud compute machine-types list |
|  |  |
|  | # See how to filter |
|  | gcloud topic filters |
|  |  |
|  | # Show some free-tier-eligible options |
|  | gcloud compute machine-types list --filter="NAME:f1-micro" |
|  | gcloud compute machine-types list --filter="NAME:f1-micro AND ZONE~us-west" |
|  |  |
|  | # Set our defaults to Los Angeles |
|  | gcloud config set compute/zone us-west2-b |
|  | gcloud config set compute/region us-west2 |
|  |  |
|  | # Start our instance |
|  | gcloud compute instances create --machine-type=f1-micro myhappyvm |
|  | ping -c 3 myhappyvm |
|  | ping -c 3 internalipaddress |
|  | ping -c 3 externalipaddress |
|  |  |
|  | # Connect to the VM |
|  | ssh externalipaddress |
|  | gcloud compute ssh myhappyvm |
|  |  |
|  | # Get our bearings -- Skip? |
|  | whoami |
|  | hostname |
|  | curl api.ipify.org |
|  |  |
|  | # Get back to Cloud Shell |
|  | exit |
|  | curl api.ipify.org |
|  |  |
|  | # Look at the Cloud Shell .ssh files |
|  | cd .ssh |
|  | ls |
|  | cat google\_compute\_engine.pub |
|  | head -n 10 google\_compute\_engine |
|  |  |
|  | # Log back onto the VM |
|  | gcloud compute ssh myhappyvm |
|  |  |
|  | # See that our key is authorized |
|  | cd .ssh |
|  | ls |
|  | cat authorized\_keys |
|  | cd .. |
|  |  |
|  | # Check out the metadata |
|  | curl metadata.google.internal/computeMetadata/v1/ |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/ |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/project/ |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/project/project-id |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/project/attributes/ |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/project/attributes/ssh-keys |
|  |  |
|  | # Look at some instance metadata |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/instance/ |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/instance/name |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/ |
|  | curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/email |
|  |  |
|  | # See what gcloud knows |
|  | gcloud config list |
|  |  |
|  | # Look at our buckets |
|  | gsutil ls |
|  | gsutil ls gs://storage-lab-cli/ |
|  |  |
|  | # Attempt to delete the VM from within the VM |
|  | gcloud compute instances delete myhappyvm |
|  |  |
|  | # Exit back to Cloud Shell and actually delete the VM |
|  | exit |
|  | gcloud compute instances delete myhappyvm |