DevOps Lab

CLOUD COMPUTE - GCP

Compute: Virtual Machine creation

Home task

TASK 1

It's aiming to gain knowledge about the mechanisms of VMs creation in Google Cloud.

Create Virtual Machine in Google cloud by the following ways:
- gcp-ui (complete Lab: google codelabs: VM creation)
- gcloud
- terraform (all settings should be provided via variables (tfvars), add URL ("http://IP. Address/" to output)

Virtual Machine Properties:
- InstanceName : nginx-${creation-way}
- Region: us-central1
- Zone: us-central1-c
- Type: General Purpose, n1, 1CPU 4.5GB RAM, Disk SSD 35Gb;
- OS: Centos7
- Allow http,https traffic
- Labels:
  - ServerType=NginxServer
  - OSFamily=RedHat
  - WayOfInstallation=${creation-way}

✅ nginx-gcloud

Details   Monitoring   Screenshot

Remote access

SSH ▾

Connect to serial console ▾

☐ Enable connecting to serial ports ❓

Logs

Cloud Logging

Serial port 1 (console)

≫ More

Instance Id
4431992715019850924

Machine type
custom (1 vCPU, 4.5 GB memory)

Reservation
Automatically choose (default)

CPU platform
Intel Haswell

Display device
Turn on a display device if you want to use screen capturing and recording tools.

☐ Turn on display device

Zone
us-central1-c

Labels

osfamily : redhat   servertype : nginxserver   wayofinstallation : gcloud

- The instance should be protected for deletion.
- VM should have the up and running Nginx (automatically provisioned after VM is started via yum, default configuration)
- The instance is running in **default** network

VM instances

	Name ^	Zone	Recommendation	In use by	Internal IP
☐	✅ nginx-gcloud	us-central1-c			10.128.0.3 (nic0)
☐	✅ nginx-gcp-ui	us-central1-c			10.128.0.2 (nic0)
☐	✅ nginx-terraform	us-central1-c			10.128.0.7 (nic0)

P.S.:
For **gcp-ui** way please use the following guide for reference:
https://cloud.google.com/compute/docs/quickstart-linux

For **gcloud** way please use the following guide for reference:
https://cloud.google.com/ai-platform/deep-learning-vm/docs/quickstart-cli

For **terraform** way please use the following guide for reference:
https://www.terraform.io/docs/providers/google/guides/getting_started.html

```
Warning: Interpolation-only expressions are deprecated

  on main.tf line 5, in provider "google":
   5:   credentials = "${file("vladimir-project-01-7b7039c36bc6.json")}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

google_compute_disk.nginx-terraform: Creating...
google_compute_instance.default: Creating...
google_compute_disk.nginx-terraform: Still creating... [10s elapsed]
google_compute_instance.default: Still creating... [10s elapsed]
google_compute_disk.nginx-terraform: Creation complete after 15s [id=projects/vladimir-project-01/zones/us-central1-c/disks/test-terraform]
google_compute_instance.default: Creation complete after 16s [id=projects/vladimir-project-01/zones/us-central1-c/instances/nginx-terraform]
google_compute_attached_disk.nginx-terraform: Creating...
google_compute_attached_disk.nginx-terraform: Still creating... [10s elapsed]
google_compute_attached_disk.nginx-terraform: Creation complete after 12s [id=projects/vladimir-project-01/zones/us-central1-c/instances/nginx-terraform/test-terraform]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

ip = 35.202.56.142
vs@iMac-Vladimir google-cloud-module %
```
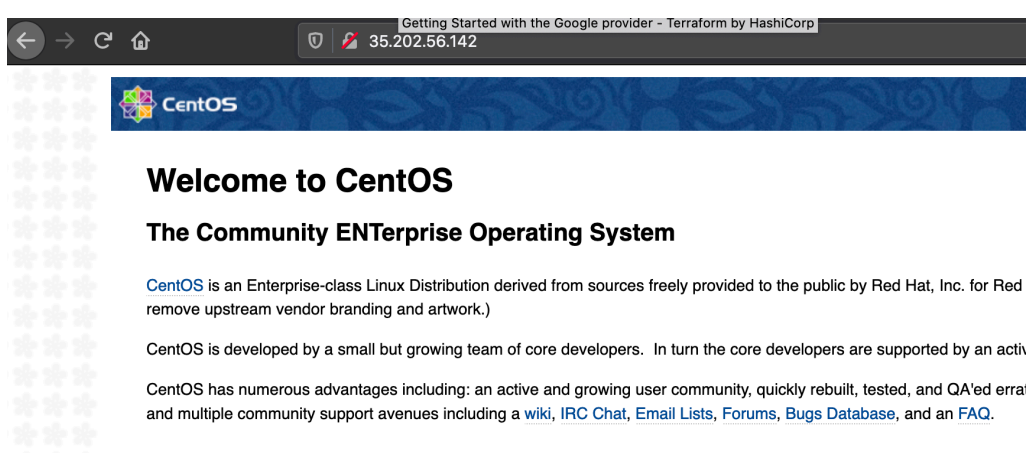
Welcome to CentOS

The Community ENTerprise Operating System

CentOS is an Enterprise-class Linux Distribution derived from sources freely provided to the public by Red Hat, Inc. for Red H remove upstream vendor branding and artwork.)

CentOS is developed by a small but growing team of core developers. In turn the core developers are supported by an active

CentOS has numerous advantages including: an active and growing user community, quickly rebuilt, tested, and QA'ed errata and multiple community support avenues including a wiki, IRC Chat, Email Lists, Forums, Bugs Database, and an FAQ.

Task 2

It's aiming to gain knowledge about the mechanisms of Persistent disk creation in Google Cloud.

Create Persistent Disk and attach it to one of existing Virtual machine (nginx-gcp-ui).
1) Complete Lab goolge codelabs: persistent disk with using your VM name
2) Create terraform configuration to do the same via terraform (use VM: nginx-gcp-terraform).
   For reference:
   - google_compute_disk
   - google_compute_attached_disk

Boot disk

Name	Image	Size (GB)	Device name	Type	Encryption	Mode	When deleting instance
nginx-terraform	centos-7-v20200811	35	persistent-disk-0	SSD persistent disk	Google managed	Boot, read/write	Delete disk

Additional disks

Name	Image	Size (GB)	Device name	Type	Encryption	Mode	When deleting instance
test-terraform	-	100	persistent-disk-1	SSD persistent disk	Google managed	Read/write	Keep disk

Local disks

All **reports**/code please place into repository:
https://github.com/MNT-Lab/google-cloud-module into appropriate branches: *first char of name + surname*.

For example:
Student: Siarhei Ivanou
Branch Name: **sivanou**

Format depends on case: README.md/scripts/terraform files

Email pattern: [MNT-CD-8.2]-FirstName-LastName

Email should contain the link to personalized branch.