

Migrate WebLogic Instance running on AZURE to GCE

Table of Contents

1. Introduction	2
2. “The Requirement”	3
3. Revision History.....	4
4. Challenge/ Problem Statement.....	5
5. Current way of Working in on-prem.....	8
6. Considerations	9
7. Proposed Solution &Transformation Details	36
8. Scenarios	37
9. Technology Road Map / Blueprints.....	38
10. GCP Service and Products	39
11. Implementation Diagram & Terraform Scripts	40
12. Conclusion.....	41

1. Introduction

In our test case we are going to Migrate the WebLogic Application running on AZURE Cloud to Google cloud. we have used the migration applicane Velostrata.

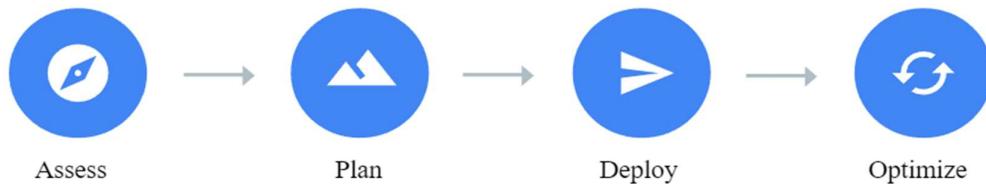
Velostrata gets enterprise applications running in Google Cloud within minutes, while data migrates transparently in the background. With Velostrata, enterprises can validate, run, and migrate applications into Google Cloud without rewriting them, modifying the image, or changing management processes.

Design the migration to Google Cloud

To migrate your VMs from your source environment to Google Cloud, we recommend that you follow the framework described in the Migration to Google Cloud series.

The framework illustrated in the preceding diagram has four phases:

1. **Assess.** In this phase, you assess your source environment, assess the workloads that you want to migrate to Google Cloud, and assess which VMs support each workload.
2. **Plan.** In this phase, you create the basic infrastructure for Migrate to VMs, such as provisioning the resource hierarchy and setting up network access.
3. **Deploy.** In this phase, you migrate the VMs from the source environment to Compute Engine.
4. **Optimize.** In this phase, you begin to take advantage of the cloud technologies and capabilities.



2. “The Requirement”

A) Objective:

Currently Customer running their WebLogic Application on AZURE, they are planning to migrate VM to Google Cloud Environment.

B) Requirements:

- Create CentOS VM in Azure cloud
- Deploy WebLogic Container using Docker.
- Required IAM for service account in Azure and GCP.
- Create a Google Cloud Compute Engine instance.
- Create an AZURE Local network and connections.
- Create a custom Virtual Private Cloud (VPC) in Google Cloud.
- Create VPN connection on both AZURE and GCP.
- Enable appropriate firewall rules for the Google Cloud VPC network.
- Set up the Compute Engine migration Manager -Velostrata Manager.
- Create Source Code and Target Code.
- Generate Migration Waves to migrate the vms

3. Revision History

Revision Date	Version No.	Comment	Status
27/11/2022	0.1	Initial Draft	In Progress
06/12/2022	0.2	Revised	Completed

4. Challenge/ Problem Statement

As per concept we have planned to migrate the weblogic server running on Azure to Google cloud environment, we have created the source machine in azure cloud and created the VPN and established the connection between Azure and GCP. We have also successfully launched the velostrate migration manager and update the source and target code.

While performing Migration we got this error - migrate for compute engine instance null is not accessible on port 443.

We have checked and identify the error due to using free google Cloud subscription then please increase your compute engine Persistent SSD quota by raising the request. The default quota is 100 GB but to deploy the cloud extension we need 500 GB for one node. Due to that deployment has getting failed.

We have limitation in Google cloud free tier, due to this we are unable to Perform our POC I.e... migrate VM from Azure to Google, but we have followed the Qwiklab and completed the migration activity from AWS to GCP.

5. Current way of Working in on-prem

6. Considerations

- Virtual Machines needs to be up and running on source cloud.
- Network connectivity between the source server and target/Cloud
- Enable all required API's services.
- VPN Connection between Source & Destination should be in same subnet.
- The amount of time that is available for migration.
- Business challenges, Required Down time for this activity.

7. Proposed Solution & Transformation Details

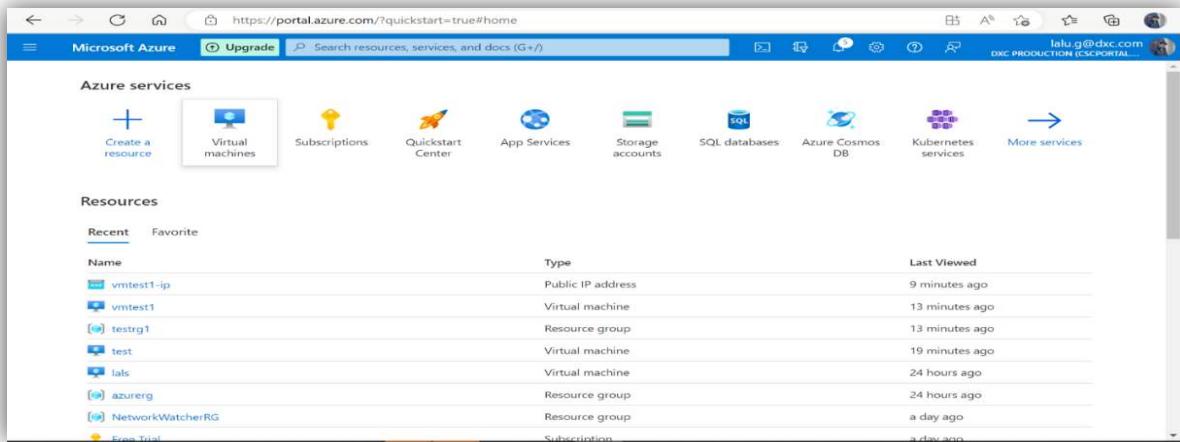
Create the Source VM running web logic application on Azure

Sign into Azure

Sign into the Azure portal at <https://portal.azure.com>.

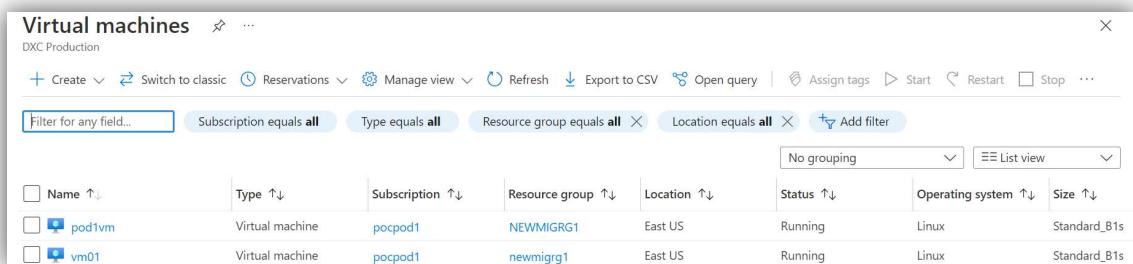
Create virtual machine

- Under **Azure Services**, select **Virtual machines**.
- In the **Virtual machines**, select **Create** and then **Azure virtual machine**.
- Under **Instance details**, enter name for the **Virtual machine** and choose operating system as CentOS.
- Leave the remaining configuration as defaults and then select the **Review +create** button at the bottom of the page.
- After validation runs, select the **Create** button at the bottom of the page.



The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with links for 'Upgrade', 'Search resources, services, and docs (G+)', and user information ('lu.g@dx.com DDC PRODUCTION (CSCPORTAL...)'). Below the bar, the 'Azure services' section is visible, featuring a 'Create a resource' button, a 'Virtual machines' button highlighted with a yellow box, and other service icons like Subscriptions, Quickstart Center, App Services, Storage accounts, SQL databases, Azure Cosmos DB, Kubernetes services, and More services. The main area is titled 'Resources' with a 'Recent' tab selected. It lists several resources: 'vmtest1-ip' (Public IP address), 'vmtest1' (Virtual machine), 'testrg1' (Resource group), 'test' (Virtual machine), 'tals' (Virtual machine), 'azurerg' (Resource group), and 'NetworkWatcherRG' (Resource group). Each item has a small icon, a name, a type, and a 'Last Viewed' timestamp. At the bottom of the list, there are buttons for 'Subscription' and 'Billing'.

- After deployment is complete, select **Go to resource**



The screenshot shows the 'Virtual machines' blade in the Azure portal. The title bar says 'Virtual machines' and 'DXC Production'. The top navigation bar includes 'Create', 'Switch to classic', 'Reservations', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', 'Start', 'Restart', 'Stop', and more. Below the bar are filter buttons for 'Subscription equals all', 'Type equals all', 'Resource group equals all', 'Location equals all', and an 'Add filter' button. A dropdown menu shows 'No grouping' and 'List view'. The main table lists two virtual machines: 'pod1vm' and 'vm01'. The columns are: Name, Type, Subscription, Resource group, Location, Status, Operating system, and Size. Both machines are listed as 'Virtual machine' under 'Type', connected to 'pocpod1' under 'Subscription', and located in 'NEWMIGRG1' under 'Resource group'. They are both in 'East US' under 'Location' and 'Running' under 'Status'. Both are 'Linux' under 'Operating system' and 'Standard_B1s' under 'Size'.

Name	Type	Subscription	Resource group	Location	Status	Operating system	Size
pod1vm	Virtual machine	pocpod1	NEWMIGRG1	East US	Running	Linux	Standard_B1s
vm01	Virtual machine	pocpod1	newmigrg1	East US	Running	Linux	Standard_B1s

Connect to virtual machine

Using Putty login New Virtual machine using public ip address 172.173.229.117 to deploy the Weblogic application.

```
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
student@vm01:~$ uname -a
Linux vm01 5.15.0-1023-azure #29~20.04.1-Ubuntu SMP Wed Oct 26 19:18:25 UTC 2022
x86_64 x86_64 x86_64 GNU/Linux
student@vm01:~$
```

Install Docker Engine on VM.

To get started with Docker Engine on your VM, make sure you [meet the prerequisites](#), then [install Docker](#).

DOCKER Installation methods

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

Set up the repository

Install the `yum-utils` package (which provides the `yum-config-manager` utility) and set up the repository.

```
$ sudo yum install -y yum-utils
$ sudo yum-config-manager \
--add-repo \
https://download.docker.com/linux/centos/docker-ce.repo
```

```
administrator123@azure-mig:~$ login as: administrator123
administrator123@188.168.153's password:
Thank you for choosing this Microsoft sponsored CentOS image from OpenLogic!
[REDACTED]
While OpenLogic support is not included with this image, OpenLogic does
offer Silver (12x5) & Gold (24x7) support options and consulting for
enterprise and/or mission critical systems as well as over 400 open-source
packages. If interested, email info@openlogic.com or call +1 612.517.2100.

[administrator123@azure-mig ~]$ uname -a
Linux azure-mig 3.10.0-1160.76.1.el7.x86_64 #1 SMP Wed Aug 10 16:21:17 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
[administrator123@azure-mig ~]$ date
Tue Nov 29 14:18:26 UTC 2022
[administrator123@azure-mig ~]$ 

[administrator123@azure-mig ~]$ sudo yum install -y yum-utils
Loaded plugins: langpacks
base-openlogic
extras-openlogic
updates-openlogic
base
base
extras
extras-openlogic
updates
updates
(1/9): extras-openlogic/7/x86_64/primary_db
(2/9): base-openlogic/7/x86_64/group_gz
(3/9): base/7/x86_64/group_gz
(4/9): openlogic/7/x86_64/primary_db
(5/9): base-openlogic/7/x86_64/primary_db
(6/9): extras/7/x86_64/primary_db
(7/9): updates-openlogic/7/x86_64/primary_db
(8/9): updates/7/x86_64/primary_db
(9/9): base/7/x86_64/primary_db
Package yum-utils-1.1.31-54.el7_8.noarch already installed and latest version
Nothing to do
[administrator123@azure-mig ~]$ 
```

Install Docker Engine

1. Install the *latest version* of Docker Engine, containers, and Docker Compose or go to the next step to install a specific version:

```
$ sudo yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

```
-bash-4.2# sudo yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

```
[administrator123@azure-mig ~]$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: langpacks
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
[administrator123@azure-mig ~]$ 
```

If prompted to accept the GPG key, verify that the fingerprint matches **060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35**, and if so, accept it. This command installs Docker, but it doesn't start Docker. It also creates a **docker** group, however, it doesn't add any users to the group by default. To install a *specific version* of Docker Engine, list the available versions in the repo, then select and install:

(a) List and sort the versions available in your repo. This example sorts results by version number, highest to lowest, and is truncated:

```
$ yum list docker-ce --showduplicates | sort -r
```

docker-ce.x86_64 3:18.09.1-3.el7	docker-ce-stable
docker-ce.x86_64 3:18.09.0-3.el7	docker-ce-stable
docker-ce.x86_64 18.06.1.ce-3.el7	docker-ce-stable
docker-ce.x86_64 18.06.0.ce-3.el7	docker-ce-stable

The list returned depends on which repositories are enabled and is specific to your version of CentOS (indicated by the **.el7** suffix in this example).

Install a specific version by its fully qualified package name, which is the package name (**docker-ce**) plus the version string (2nd column) starting at the first colon (:), up to the first hyphen, separated by a hyphen (-). For example, **docker-ce-18.09.1**.

```
$ sudo yum install docker-ce-<VERSION_STRING> docker-ce-cli-<VERSION_STRING> containerd.io docker-compose-plugin
```



3:19.03.12-3.el7

```
$ sudo yum install docker-ce-20.10.21.-3.el7.x86_64 docker-ce-cli-20.10.21.-3.el7.x86_64 containerd.io docker-compose-plugin
```

```
[administrator123@azure-mig ~]$ rpm -qa | grep -i docker
docker-compose-plugin-2.12.2-3.el7.x86_64
docker-ce-cli-20.10.21-3.el7.x86_64
docker-ce-rootless-extras-20.10.21-3.el7.x86_64
docker-scan-plugin-0.21.0-3.el7.x86_64
docker-ce-20.10.21-3.el7.x86_64
[administrator123@azure-mig ~]$ docker --version
Docker version 20.10.21, build baedalf
```

This command installs Docker, but it doesn't start Docker. It also creates a **docker** group, however, it doesn't add any users to the group by default.

2. Start Docker.

```
$ sudo systemctl status docker.service
$ sudo systemctl start docker.service
$ sudo systemctl enable docker.service
$ sudo systemctl start docker
$ sudo systemctl list-unit-files --type=service |grep -I docker
```

```
[administrator123@azure-mig ~]$ sudo systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
     Active: active (running) since Tue 2022-11-29 14:27:58 UTC; 1min 33s ago
       Docs: https://docs.docker.com
 Main PID: 8425 (dockerd)
   Groups: /system.slice/docker.service
          └─8425 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.403694247Z" level=info msg="scheme \"unix\" not registered, fallback to default...dule=grpc"
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.403702047Z" level=info msg="c2bResteaverWrapper: sending update to cc: {{(unix:/:...dule=grpc
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.403712047Z" level=info msg="ccBResteaverWrapper: switching balancer to \"pick_first\" module=grpc
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.444662223Z" level=info msg="Loading containers: start."
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.444662223Z" level=info msg="Default bridge (dockero) is assigned with an IP ad... address"
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.637269412Z" level=info msg="Loading containers: done."
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.659998199Z" level=info msg="Daemon has completed initialization"
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.660134499Z" level=info msg="Daemon has completed initialization"
Nov 29 14:27:58 azure-mig dockerd[8425]: time="2022-11-29T14:27:58.696773177Z" level=info msg="API listen on /var/run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
```

Please check that docker is installed or not by using below command.

docker –version

```
[administrator123@azure-mig ~]$ rpm -qa | grep -i docker
docker-compose-plugin-2.12.2-3.el7.x86_64
docker-ce-cli-20.10.21-3.el7.x86_64
docker-ce-rootless-extras-20.10.21-3.el7.x86_64
docker-scan-plugin-0.21.0-3.el7.x86_64
docker-ce-20.10.21-3.el7.x86_64
[administrator123@azure-mig ~]$ docker --version
Docker version 20.10.21, build baedalf
```

```
-bash-4.2# systemctl list-unit-files --type=service |grep -i docker
docker.service
enabled
-bash-4.2# 
```

Using the below link you can navigate the docker hub which has docker images available.

<https://hub.docker.com/r/ismaleiva90/weblogic12>

Use the below command to pull the docker image from docker hub, It may take some time as it has downloaded all the dependencies.

docker pull ismaleiva90/weblogic12

```
[administrator123@azure-mig ~]$ docker pull ismaleiva90/weblogic12
Using default tag: latest
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://<Fvar%2Frun%2Fdocker.sock/v1.24/images/create?fromImage=ismaleiva90%2Fweblogic12&tag=latest": dial unix /var/run/docker.sock: connect: permission denied
[administrator123@azure-mig ~]$ sudo docker pull ismaleiva90/weblogic12
Using default tag: latest
latest: Pulling from ismaleiva90/weblogic12
Image docker.io/ismaleiva90/weblogic12:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
a3ed95cae02: Pull complete
b8c01449ct4a: Pull complete
b82337cd1027: Pull complete
9e37151bd6be: Pull complete
0247d9683c11: Pull complete
b8c01449ct4a: Pull complete
91c32b48ce3b: Pull complete
c617a682e3f2: Pull complete
03083f09438c: Pull complete
fe83b45f7a9f: Pull complete
db497d879495: Pull complete
2a44b564aacf: Pull complete
cf65f5fb4ce9: Pull complete
377f0691a3c4: Pull complete
59fd638f77cf: Pull complete
c049d25582e2: Pull complete
916614cf1ec2: Pull complete
5cf15f6b208: Pull complete
ccb5273b4f71: Pull complete
```

To start the docker machine with 7001, 7002 and 5556 ports opened:

docker run -d -p 49164:7002 -p 49165:5556 ismaleiva90/weblogic12:latest

```
[administrator123@azure-mig ~]$ sudo docker run -d -p 49164:7002 -p 49165:5556 ismaleiva90/weblogic12:latest
27511305a73a13e0806162146c9127af6891caa61d7a3f9b2418bbc99e462b
[administrator123@azure-mig ~]$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
27511305a73 ismaleiva90/weblogic12:latest "/u01/oracle/weblogi..." 7 seconds ago Up 7 seconds 7001/tcp, 0.0.0.0:49165->5556/tcp, :::49165->5556/tcp, 0.0.0.0:49164->7002/tcp, :::49164->7002/tcp stoic_chebyshev
[administrator123@azure-mig ~]$
```

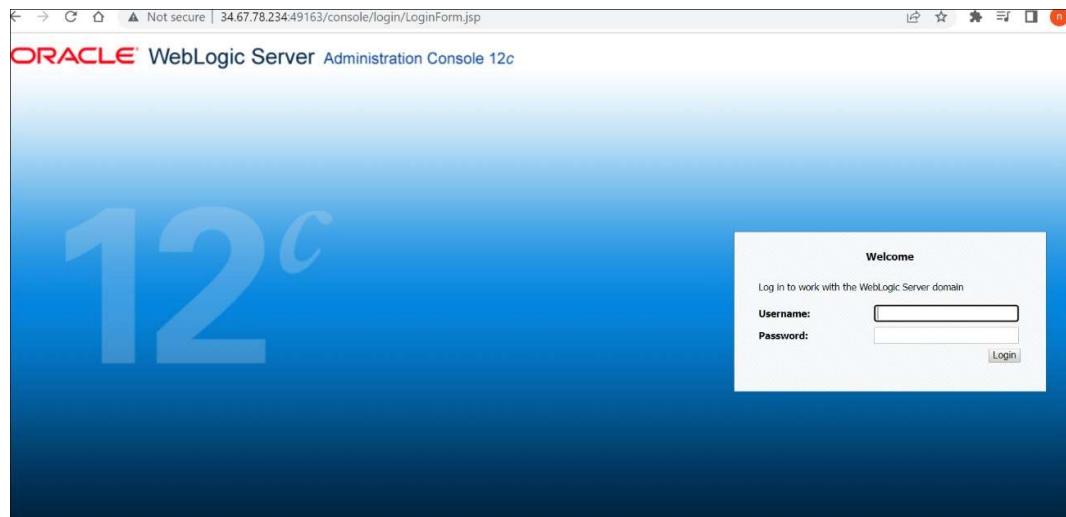
To verify the Docker container in your virtual machine use below command.

```
[administrator123@azure-mig ~]$ sudo docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
227511305a73 ismaleiva90/weblogic12:latest "/u01/oracle/weblogi..." 13 minutes ago Up 13 minutes 7001/tcp, 0.0.0.0:49165->5556/tcp, :::49165->5556/tcp, 0.0.0.0:49164->7002/tcp, :::49164->7002/tcp stoic_chebyshev
[administrator123@azure-mig ~]$
```

To login your web console use below link as mentioned, you can give your vm public ip to access the GUI.

<http://20.121.32.194:49164/console>

Please provide the username and password to access your web logic server, Here I would like t to give the default credentials provided by Oracle.



User: weblogic

Pass: welcome1

Once you provided the credentials you can be able to go to the Oracle Web logic server.

Note: If you are not able to allow, then you may need to check from your firewall rule and need to enable firewall rule if it is not allowed.

The screenshot shows the Oracle WebLogic Server Administration Console 12c. The left sidebar includes a 'Change Center' section with 'View changes and restarts' and a 'Domain Structure' section for 'base_domain' which lists Environment, Deployments, Services, Security Realms, Interoperability, and Diagnostics. A 'How do I...' panel on the left provides links for configuration tasks. The main content area is titled 'Home Page' and contains sections for 'Information and Resources' (Helpful Tools: Configure applications, GridLink for RAC Data Source, Dynamic Cluster, Recent Task Status, Set your console preferences; General Information: Common Admin Task Descriptions, Read the documentation, Ask a question on My Oracle Support) and 'Domain Configurations' (Domain, Environment, Clusters, Coherence Clusters, Machines, Virtual Hosts, Work Managers, Startup And Shutdown Classes). The 'Services' section lists Messaging (JMS Servers, Store-and-Forward Agents, JMS Modules, Path Services, Bridges), Data Sources, Persistent Stores, XML Registries, XML Entity Caches, Foreign JNDI Providers, Work Contexts, and JCOM. The 'Interoperability' section lists WTC Servers, Jolt Connection Pools. The 'Diagnostics' section lists Log Files, Diagnostic Modules, Built-in Diagnostic Modules, Diagnostic Images, Request Performance, Archives, Context, and SNMP.

Configure Virtual Network Gateway:

A virtual network gateway is composed of two or more VMs that are automatically configured and deployed to a specific subnet you create called the gateway subnet. The gateway VMs contain routing tables and run specific gateway services. You can't directly configure the VMs that are part of the virtual network gateway, although the settings that you select when configuring your gateway impact the gateway VMs that are created.

The screenshot shows the AWS Migration Service console. The top navigation bar includes 'Home > Virtual network gateways > migration'. The main view shows a summary of the 'migration' virtual network gateway, including its Resource group (newmigrgr1), Location (East US), Subscription (Free Trial), and Subscription ID (8e72ccb8-9c3d-4610-b41d-1ea22bfd7989). On the left, there are tabs for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Configuration, and Connections. The right side displays detailed information under the 'Essentials' section, including SKU (VpnGw2AZ), Gateway type (VPN), VPN type (Route-based), Virtual network (newmigrgr1-vnet), and Public IP address (20.121.247.25 (migip)). A 'JSON View' button is located at the bottom right of the details pane.

Home > Connections

DXC Production

+ Create Manage view Refresh Export to CSV Open query Assign tags

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

No grouping List view

Name	Status	Peer 1	Peer 2	Resource group	Location	Subscription	...
mig	Connected	migration	localnet	newmigr1	East US	Free Trial	...

Configure VPN Gateway:

When you configure a virtual network gateway, you configure a setting that specifies the gateway type. The gateway type determines how the virtual network gateway will be used and the actions that the gateway takes. The gateway type 'Vpn' specifies that the type of virtual network gateway created is a 'VPN gateway'.

Azure: localnet (34.71.197.135) need to update in GCP cloud VPN gateway (IP) Public IP address of migration manager 20.121.247.25 (migip) need to update as PEER VPN gateway

Google Cloud My First Project vpn

Hybrid Connectivity VPN tunnel details

VPN tunnel details

vpn-1-tunnel-1 Status Tunnel is up and running.

DETAILS MONITORING

Remote peer gateway

IP address 20.121.247.25

Cloud VPN gateway

Name	vpn-1 (classic gateway)
VPC network	default
Region	us-central1
IP address	34.71.197.135
Logs	View

Connection Established between Azure and GCP

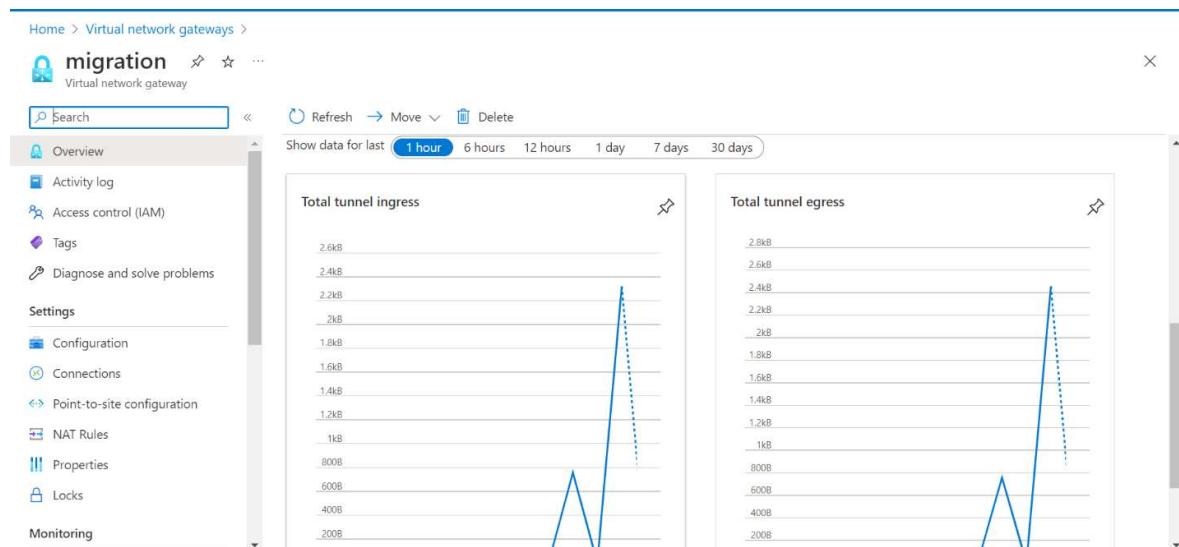
VPN connectivity check:

Ping test from GCP to Azure:

```
narrababu297@testvm:~$ ping 10.4.0.5
PING 10.4.0.5 (10.4.0.5) 56(84) bytes of data.
64 bytes from 10.4.0.5: icmp_seq=1 ttl=63 time=34.4 ms
64 bytes from 10.4.0.5: icmp_seq=2 ttl=63 time=31.7 ms
64 bytes from 10.4.0.5: icmp_seq=3 ttl=63 time=32.4 ms
64 bytes from 10.4.0.5: icmp_seq=4 ttl=63 time=31.1 ms
^C
--- 10.4.0.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 31.130/32.418/34.440/1.248 ms
narrababu297@testvm:~$
```

Ping test from Azure to GCP:

```
{student@vm01:~$ ping 10.128.0.4
PING 10.128.0.4 (10.128.0.4) 56(84) bytes of data.
64 bytes from 10.128.0.4: icmp_seq=1 ttl=63 time=34.5 ms
64 bytes from 10.128.0.4: icmp_seq=2 ttl=63 time=31.2 ms
64 bytes from 10.128.0.4: icmp_seq=3 ttl=63 time=31.8 ms
64 bytes from 10.128.0.4: icmp_seq=4 ttl=63 time=31.4 ms
64 bytes from 10.128.0.4: icmp_seq=5 ttl=63 time=32.3 ms
^C
--- 10.128.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 31.156/32.219/34.489/1.196 ms
student@vm01:~$
```



Creating Migration Manager:

To configure the Migrate for Compute Engine Manager on Google Cloud. The migration manager instance manages all migration infrastructure components and orchestrates migrations.

- Configure required network connectivity.
- Configure firewall rules.
- Create service accounts.
- Create migration manager passwords
- Enable Required APIs.

Refer below document to configure - **Configuring the migration manager**

https://cloud.google.com/migrate/compute-engine/docs/4.11/how-to/configure-manager/configuring-migration-manager?_gl=1*1bp76hh*_ga*MTM1MDM4MTIzMjMy4xNjYxMjQ5ODc5*_ga_WH2QY8WWF5*MTY3MDMzNDE3MC4yMS4xLjE2NzAzMzQ1MTYuMC4wLjA.&_ga=2.45972324.-1350381233.1661249879&_gac=1.81212773.1670334171.Cj0KCQiA7bucBhCeARIsAIOWr-9CCar82Cb2wa5T_QlIuG32I7OD5w2Wu8Bud-Wj2K9WK0-GtPDgDiIaAtgsEALw_wcB

A new version of Migrate for Compute Engine is available! TRY THE NEW VERSION

The migration manager instance manages all migration infrastructure components and orchestrates migrations. Migration console is served from the migration manager. [Learn more](#)

Typically you will need exactly one migration manager to perform the whole migration to Google Cloud Platform. You will need to create more only if you want to migrate multiple on-premises vCenter deployments. This is because a single migration manager is capable of migrating from only one vCenter deployment, although it can handle it together with sources from multiple cloud providers at the same time.

[DOCUMENTATION](#)

Instance Name	Region	Zone	Network	Subnetwork	Internal IP	Action	Instance Details
migvm	us-east1-b					Edit	View
mimvm1	us-central1	us-central1-c	default	default	10.128.0.5	Edit	View

Google Cloud Compute Engine - Migration manager details

mmvm1

VM INSTANCE SERVICE ACCOUNTS

Migration manager name	mmvm1
Region	us-central1
Zone	us-central1-c
Network	default
Subnetwork	default
Network tags	No network tags
Machine type	n1-standard-4

Google Cloud Secret Manager

my-secret-password

SECRET Manager lets you store, manage, and secure access to your application secrets.

Name	Location	Encryption	Labels	Created	Expiration	Actions
my-secret-password	Automatically replicated	Google-managed	None	12/3/22, 3:29 PM		⋮

Google Cloud Secret details

Secret: "my-secret-password"

OVERVIEW VERSIONS PERMISSIONS LOGS

Versions

Version	Status	Encryption	Created on	Actions
1	Enabled	Google-managed	12/3/22, 3:29 PM	⋮

Microsoft Azure - Notifications

More events in the activity log → Dismiss all

Added Role assignment

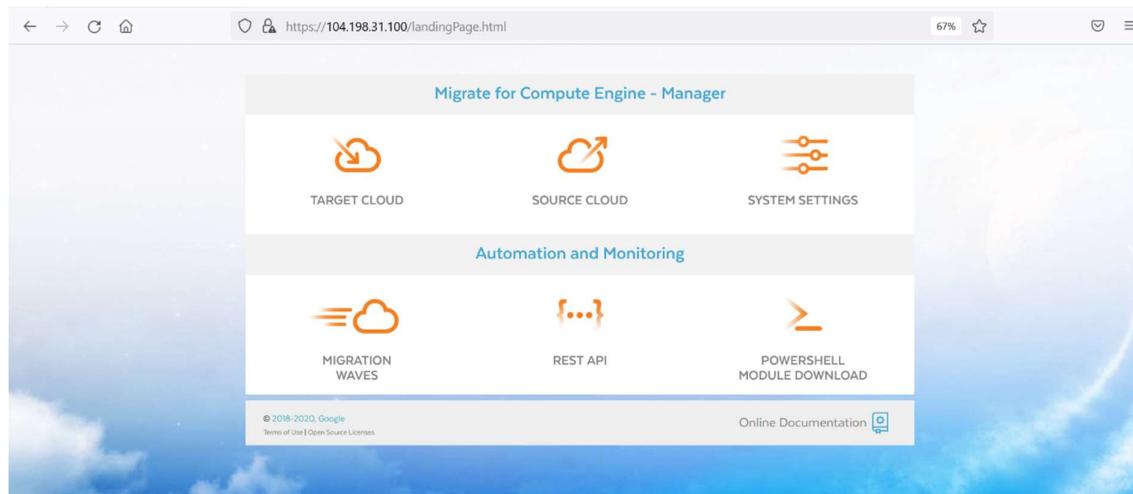
gcpvpn was added as Owner for Free Trial. a few seconds ago

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure', 'Upgrade' (with a 'Get started' button), a search bar ('Search resources, services, and docs (G+)'), and user information ('lalu.g@dx.com DXC PRODUCTION (SCPORTAL...)'). Below the navigation bar, the path 'Home > Subscriptions > Free Trial' is visible. The main content area is titled 'Free Trial | Access control (IAM)' and shows a table of role assignments. The table has columns for 'Name', 'Type', 'Role', 'Scope', and 'Condition'. One entry is shown: 'gcpvpn' (App) is assigned the 'Owner' role with 'This resource' scope and 'None' condition.

Velostrata tool for Migration:

we have created migration manager with velostrata-mgmt-4-11-11-36321-pantheon and secret manager in GCP successfully and able to access the migration appliance.

Please find the below details to access Migrate for Compute Engine-Manager.



Due to Limitiataion in free trial we are unable to proceed the VM migration from azure to GCP using Valostrata migration tool, but we have followed the Qwiklabs and completed the VM migration from AWS to GCP successfully.

MIGRATE FOR COMPUTE ENGINE - HEALTH CHECKS			
Cloud Extension Name : ext1			Search 
Component	Health check	Result	Reason
Edge A	Instance Provisioning Status	Failed	instance null is not accessible on port 443
Edge B	Instance Provisioning Status	Failed	instance null is not accessible on port 443
Cloud Api	Cloud Control Access	Passed	
Cloud Extension	Pending Upgrade	Passed	
Manager Control	Critical Disk Space Utilization	Passed	
Manager Control	Backend Process	Passed	
Manager Control	Backend Functionality	Passed	
Manager Control	FE Connection Status From BE	Passed	
Manager Control	VC Connection Status From BE	Passed	
Manager Control	VC Credentials	Passed	

© 2018-2020, Google
Terms of Use | Open Source Licenses

Showing 1-10 of 11     

VM Migration from AWS to Google Cloud

- Set your Google Cloud Project ID and retrieve the Terraform scripts.

```
student_04_67df50181fe59cloudshell: (qwiklabs-gcp-03-25ee7e304eac) $ gcloud config set project $DEVSHHELL_PROJECT_ID
Updated property [core/project].
student_04_67df50181fe59cloudshell: (qwiklabs-gcp-03-25ee7e304eac) $ gsutil cp gs://cloud-training/OCEL213/autonetdeploy-multicloudvpn2.tar .
tar xvf autonetdeploy-multicloudvpn2.tar
Copying gs://cloud-training/OCEL213/autonetdeploy-multicloudvpn2.tar...
- [1 files] [726.6 KiB/726.6 KiB]
Operation completed over 1 objects/726.6 KiB.
autonetdeploy-multicloudvpn/
autonetdeploy-multicloudvpn/.terraform
autonetdeploy-multicloudvpn/aws_set_credentials.sh
autonetdeploy-multicloudvpn/CONTRIBUTING
autonetdeploy-multicloudvpn/create_instance.sh
autonetdeploy-multicloudvpn/gcp_set_project.sh
autonetdeploy-multicloudvpn/terraform/autonetdeploy_multicloudvpn_terraform.sh
autonetdeploy-multicloudvpn/terraform.sh
autonetdeploy-multicloudvpn/images/
autonetdeploy-multicloudvpn/images/.autonetdeploy_gcpawsvpn_arch.png
autonetdeploy-multicloudvpn/images/.gcpawsvpn_plan_graph.png
autonetdeploy-multicloudvpn/images/autonetdeploy_gcpawsvpn_arch.png
autonetdeploy-multicloudvpn/images/gcpawsvpn_plan_graph.png
autonetdeploy-multicloudvpn/LICENSES
autonetdeploy-multicloudvpn/migrate_sa_roles.sh
autonetdeploy-multicloudvpn/README.md
autonetdeploy-multicloudvpn/terraform/
autonetdeploy-multicloudvpn/terraform/aws_compute.tf
autonetdeploy-multicloudvpn/terraform/aws_networking.tf
autonetdeploy-multicloudvpn/terraform/aws_outputs.tf
autonetdeploy-multicloudvpn/terraform/aws_security.tf
autonetdeploy-multicloudvpn/terraform/aws_variables.tf
autonetdeploy-multicloudvpn/terraform/gcp_compute.tf
autonetdeploy-multicloudvpn/terraform/gcp_networking.tf
autonetdeploy-multicloudvpn/terraform/gcp_outputs.tf
autonetdeploy-multicloudvpn/terraform/gcp_security.tf
autonetdeploy-multicloudvpn/terraform/gcp_variables.tf
autonetdeploy-multicloudvpn/terraform/main.tf
autonetdeploy-multicloudvpn/terraform/maingraph.sh
autonetdeploy-multicloudvpn/terraform/main_userdata.sh
student_04_67df50181fe59cloudshell: (qwiklabs-gcp-03-25ee7e304eac) $ 
```

➤ APIs & Services > Credentials and Uploaded the JSON File

The screenshot shows the Google Cloud Platform's APIs & Services page under the 'Credentials' tab. On the left, there's a sidebar with options like 'Enabled APIs & services', 'Library', 'Credentials' (which is selected), 'OAuth consent screen', and 'Page usage agreements'. The main area has sections for 'API Keys', 'OAuth 2.0 Client IDs', and 'Service Accounts'. Under 'Service Accounts', there are three entries:

Name	Type	Client ID	Actions
611338654076-compute@developer.gserviceaccount.com	Compute Engine default service account		
qwiklabs-gcp-03-25ee7e304eac@qwiklabs-gcp-03-25ee7e304eac.iam.serviceaccount.com	Qwiklabs User Service Account		

```
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac) $ ./gcp_set_credentials.sh ~/qwiklabs-gcp-03-25ee7e304eac-9ef8f1432d0f.json
Created /home/student_04_67df50181fe5/.config/gcloud/credentials_autonetdeploy.json from /home/student_04_67df50181fe5/qwiklabs-gcp-03-25ee7e304eac-9ef8f1432d0f.json.
Updated gcp_credentials file path in /home/student_04_67df50181fe5/autonetdeploy-multicloudvpn/terraform/terraform.tfvars.
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac) $ 
```

➤ Set AWS Account credentials and deploy the Terraform

```
export AWS_SECRET_KEY=AKIAZZ6JTHR74UN2NJUF
```

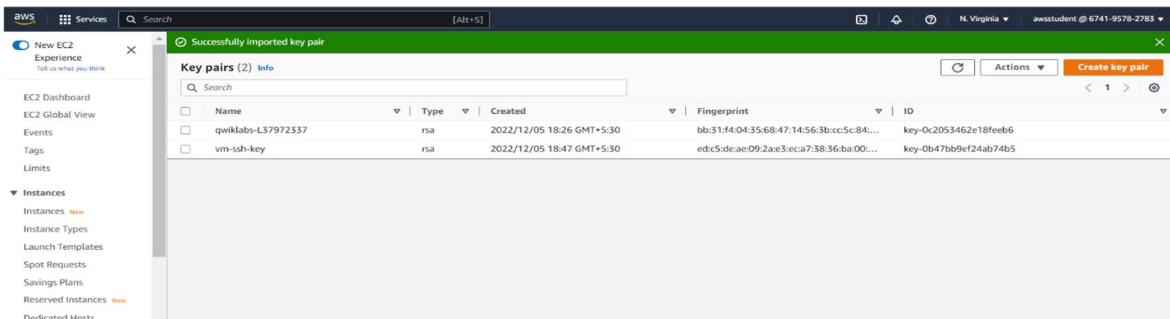
```
export AWS_SECRET_KEY=
4xZ+isy4M/SoLkGdTG0S2DHDpb+E0q1JOCzH0XsM
```

- To set your credentials – `./aws_set_credentials.sh $AWS_ACCESS_KEY $AWS_SECRET_KEY`
- configuration files for Deployment Manager and Terraform – `./gcp_set_project.sh`

➤ Generate Key Pairs - `ssh-keygen -t rsa -f ~/.ssh/vm-ssh-key -C $whoami`

```
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ export AWS_ACCESS_KEY=AKIAZZ6JTHR74UN2NJUFstudent_04_67df50181fe5@cloudshell:
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ export AWS_SECRET_KEY=4xZ+isy4M/S0lkGdtG0S2DHDb+E0q1JOCzH0XSM
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ ./aws_set_credentials.sh $AWS_ACCESS_KEY $AWS_SECRET_KEY
Created ./accessKeys.csv.
cp: cannot stat '/home/student_04_67df50181fe5/.aws/credentials_autonetdeploy': No such file or directory
Created backup (/home/student_04_67df50181fe5/.aws/credentials_autonetdeploy.bak).
Created /home/student_04_67df50181fe5/.aws/credentials_autonetdeploy.
Updated aws credentials file path in /home/student_04_67df50181fe5/autonetdeploy-multicloudvpn/terraform/terraform.tfvars.
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ ./gcp_set_project.sh
Updated gcp_project_id in /home/student_04_67df50181fe5/autonetdeploy-multicloudvpn/terraform/terraform.tfvars.
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ ssh-keygen -t rsa -f ~/.ssh/vm-ssh-key -C $whoami
Generating public/private rsa key pair.
Created directory '/home/student_04_67df50181fe5/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/student_04_67df50181fe5/.ssh/vm-ssh-key
Your public key has been saved in /home/student_04_67df50181fe5/.ssh/vm-ssh-key.pub
The key fingerprint is:
SHA256:1OMNjQkRVlQ0gbImcRznFT7KZvpM/y2IMPZhRdZ7g student_04_67df50181fe5
The key's randomart image is:
+---[RSA 3072]----+
|       .++OOB|
|       O+B+|
|       . @+.|
|       . o +o|
| S . . . o|
| . = *   |
| . X.o |
| o+B+ |
| B*-*|
+---[SHA256]-----+
```

```
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ chmod 400 ~/.ssh/vm-ssh-key
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ gcloud compute config-ssh --ssh-key-file=~/.ssh/vm-ssh-key
Updating project ssh metadata...working..Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-25ee7e304eac].
Updating project ssh metadata...done.
WARNING: No host aliases were added to your SSH configs because you do not have any running instances. Try running this command again after running some instances.
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ readlink -f ~/.ssh/vm-ssh-key.pub
/home/student_04_67df50181fe5/.ssh/vm-ssh-key.pub
```



Deploy the Terraform

Install the Terraform providers and verify your credentials –

```
pushd ./terraform && terraform init && popd >/dev/null
```

```
pushd ./terraform && terraform plan && popd >/dev/null
```

```
student_04_67df5018fe5@cloudshell:~/autonetdeploy-multicloudvpn (gwiklabs-gcp-03-25ee7e304eac)$ pushd ./terraform && terraform init && popd > /dev/null
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Finding hashicorp/aws versions matching "4.10.0"...
- Finding hashicorp/google versions matching "4.18.0"...
- Installing hashicorp/aws v4.10.0...
- Installed hashicorp/aws v4.10.0 (signed by HashiCorp)
- Installing hashicorp/google v4.18.0...
- Installed hashicorp/google v4.18.0 (signed by HashiCorp)
```

```
Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.
```

```
Warning: Version constraints inside provider configuration blocks are deprecated
```

```
on main.tf line 22, in provider "google":
22:   version = "4.18.0"
```

```
Terraform 0.13 and earlier allowed provider version constraints inside the provider configuration block, but that is now deprecated and will be removed in a future version of Terraform. To silence this warning, move the provider version constraint into the required_providers block.
```

```
(and one more similar warning elsewhere)
```

```
Terraform has been successfully initialized!
```

```
student_04_67df5018fe5@cloudshell:~/autonetdeploy-multicloudvpn (gwiklabs-gcp-03-25ee7e304eac)$ pushd terraform
```

```
./autonetdeploy-multicloudvpn/terraform ~$ autonetdeploy-multicloudvpn/terraform (gwiklabs-gcp-03-25ee7e304eac)$ terraform validate
```

```
Warning: Version constraints inside provider configuration blocks are deprecated
```

```
on main.tf line 22, in provider "google":
22:   version = "4.18.0"
```

```
Terraform 0.13 and earlier allowed provider version constraints inside the provider configuration block, but that is now deprecated and will be removed in a future version of Terraform. To silence this warning, move the provider version constraint into the required_providers block.
```

```
(and one more similar warning elsewhere)
```

```
Warning: Argument is deprecated
```

```
with provider("registry.terraform.io/hashicorp/aws"),
on main.tf line 36, in provider "aws":
 36:   shared_credentials_file = pathexpand(var.aws_credentials_file_path)
```

```
Use shared_credentials_files instead.
```

```
Success! The configuration is valid, but there were some validation warnings as shown above.
```

```
student_04_67df5018fe5@cloudshell:~/autonetdeploy-multicloudvpn/terraform (gwiklabs-gcp-03-25ee7e304eac)$ []
```

Configure AWS EC2 Instance for Migration

AWS Instance :

The screenshot shows the AWS EC2 Instances page. At the top left, there's a 'New EC2 Experience' button and a search bar. The main area displays a table with one row. The first column contains a checkbox, the second column is 'Name' with the value 'aws-vm-us-ea...', and the third column is 'Instance ID' with the value 'i-01d5d8931b24afe3b'. Below the table, there are several navigation links: 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', and 'Limits'.

```
Apply complete! Resources: 44 added, 0 changed, 0 destroyed.
```

Outputs:

```
aws_instance_external_ip = "54.159.161.6"
aws_instance_internal_ip = "172.16.0.100"
gcp_instance_external_ip = <<EOT
34.172.78.68
```

54.159.161.6 – AWS Instance public IP

- In Cloud Shell, run the following [AWS_INSTANCE_EXTERNAL_IP] with your instance's public IP address:

```
ssh -i ~/.ssh/vm-ssh-key ubuntu@[AWS_INSTANCE_EXTERNAL_IP]
```

The screenshot shows a terminal window with the following content:

```
student 04:67df50181fc58cloudshell:/automated-deploy-multi-cloudvm/terraform (qwiklabs-gcp-03-25ee7e304eac)$ ssh -i ~/.ssh/vm-ssh-key ubuntu@54.159.161.6
The authenticity of host '54.159.161.6' (54.159.161.6) can't be established.
ECDSA key fingerprint is SHA256:EM4+he6G5Y3euwBuoIWyu9QRDxalqOH-njYrph8y14.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.159.161.6' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-1128-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

UA Infra: Extended Security Maintenance (ESM) is not enabled.
0 updates can be applied immediately.

152 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-16-0-100:~$ date
Mon Dec  5 13:34:58 UTC 2022
ubuntu@ip-172-16-0-100:~$
```

- Install and configure Apache:

```
sudo bash -c "apt-get update"
sudo bash -c "apt-get install apache2 -y"
echo "Hello World" > MyText.txt
```

- When the Linux image runs on Google Cloud, it expects to find kernel drivers for the Migrate for Compute Engine mapped disks. These must be downloaded and installed before the migration up to Google Cloud. The driver installation must be performed on the EC2 machine.
- To download the installation package –

```
wget https://storage.googleapis.com/velostrata-
release/4.5/4.5.1/velostrata-prep-0.9-3.deb
```

```
sudo dpkg -i velostrata-prep-0.9-3.deb
sudo apt-get update && sudo apt-get install -f -y
```

- You have now completed the setup of the EC2 instance. Type exit to log out of the AWS VM instance

```

ubuntu@ip-172-16-0-100:~$ echo "Hello World" > MyText.txt
ubuntu@ip-172-16-0-100:~$ wget https://storage.googleapis.com/velostrata-release/4.5/4.5.1/velostrata-prep-0.9-3.deb
--2022-12-05 13:36:52 -- https://storage.googleapis.com/velostrata-release/4.5/4.5.1/velostrata-prep-0.9-3.deb
Resolving storage.googleapis.com (storage.googleapis.com)... 142.251.16.128, 142.251.33.208, 142.251.111.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|142.251.16.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 370346 (362K) [application/octet-stream]
Saving to: 'velostrata-prep-0.9-3.deb'

velostrata-prep-0.9-3.deb          100%[=====]   361.67K --.-KB/s   in 0.01s

2022-12-05 13:36:52 (35.1 MB/s) - 'velostrata-prep-0.9-3.deb' saved [370346/370346]

ubuntu@ip-172-16-0-100:~$ sudo dpkg -i velostrata-prep-0.9-3.deb
Selecting previously unselected package velostrata-prep.
(Reading database ... 52269 files and directories currently installed.)
Preparing to unpack velostrata-prep-0.9-3-3.deb ...
Unpacking velostrata-prep (0.9-3) ...
dpkg: dependency problems prevent configuration of velostrata-prep:
 velostrata-prep depends on multipath-tools; however:
  Package multipath-tools is not installed.
 velostrata-prep depends on multipath-tools-boot; however:
  Package multipath-tools-boot is not installed.
 velostrata-prep depends on sg3-utils; however:
  Package sg3-utils is not installed.

dpkg: error processing package velostrata-prep (--install):
 dependency problems - leaving unconfigured
Processing triggers for ureadahead (0.100.0-19.1) ...
Errors were encountered while processing:
 velostrata-prep
ubuntu@ip-172-16-0-100:~$ sudo apt-get update && sudo apt-get install -f -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [99.8 kB]

```

Set up Migration Service Accounts

Here, we create the service accounts that you will assign to Migrate for Compute Engine.

- To create the service accounts that you will assign to Migrate for Compute Engine –

```

cd ~/autonetdeploy-multicloudvpn/
sh migrate_sa_roles.sh

```

```

student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn/terraform (qwiklabs-gcp-03-25ee7e304eac)$ cd ~/autonetdeploy-multicloudvpn/
sh migrate_sa_roles.sh
Created service account [migration-manager].
Updated IAM policy for serviceAccount [migration-manager@qwiklabs-gcp-03-25ee7e304eac.iam.gserviceaccount.com].
bindings:
- members:
  - serviceAccount:migration-manager@qwiklabs-gcp-03-25ee7e304eac.iam.gserviceaccount.com
    role: roles/iam.serviceAccountTokenCreator
etag: BwXvFNJ6b4c=
version: 1
Created service account [migration-cloud-extension].
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ 

```

Set up Migration Manager

- To create the instance - bash create_instance.sh

```

student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ bash create_instance.sh
WARNING: You have selected a disk size of under [200GB]. This may result in poor I/O performance. For more information, see: https://developers.google.com/compute/docs/disks#performance.
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-25ee7e304eac/zones/us-central1-a/instances/velo-mgr].
NAME: velo-mgr
ZONE: us-central1-a
MACHINE_TYPE: n1-standard-4
PREEMPTIBLE:
INTERNAL_IP: 10.240.0.2
EXTERNAL_IP: 104.198.79.1
STATUS: RUNNING
student_04_67df50181fe5@cloudshell:~/autonetdeploy-multicloudvpn (qwiklabs-gcp-03-25ee7e304eac)$ 

```

- External IP: 104.198.79.1 - velo-mgr: <https://104.198.79.1/>
- Username: administrator123
- password: [administrator@123](#)

Configure AWS as a Source

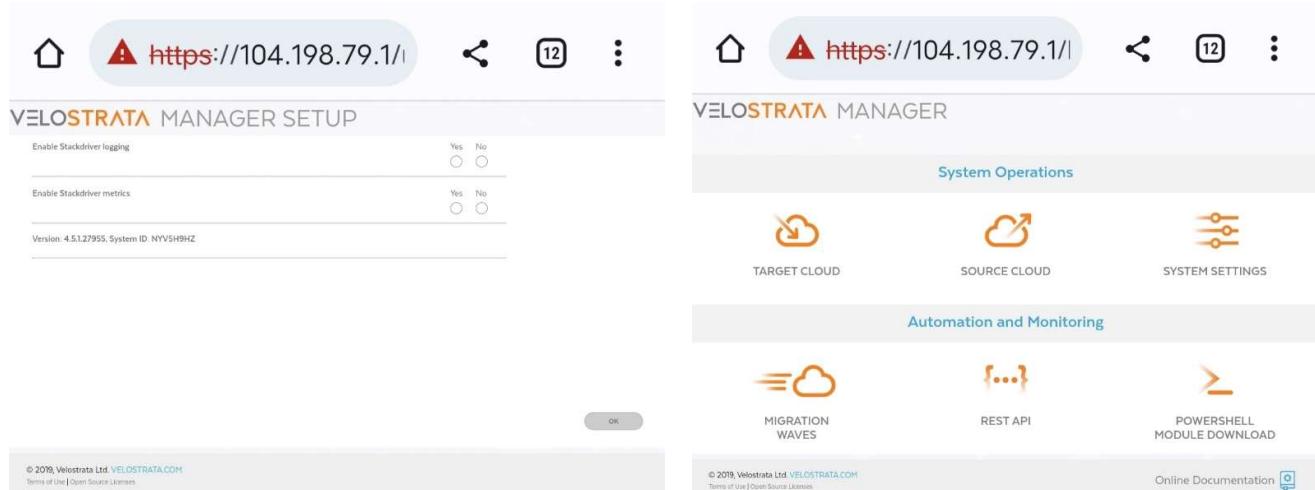
```
curl -k https:// 104.198.79.1
```

- The Velostrata Manager can take up to 2 minutes to launch before you can connect to it.
- In **Sign in**, specify the following information

Username - apiuser

Password - **velo1234**

- On the initial setup screen, click **Yes** for both Stackdriver Logging and Stackdriver Metrics.



Now you provide source information about the AWS VM to Migrate for Compute Engine.

- In Source Cloud , Click the Cloud Credentials and Create New credential

VELOSTRATA CLOUD CREDENTIALS

Create New Credentials

Cloud Provider: AWS

Credentials Name: aws-credentials

Region: US East (N. Virginia)

Access Key: AKIAZZ6JTHR74UN2NJUF

Secret Key: (redacted)

- In Source Cloud , Click the Cloud Details and Create New Details.

VELOSTRATA SOURCE CLOUD

Create New Cloud Details

Cloud Provider: AWS

Name: AWS

Credentials: aws-credentials

Region: US East (N. Virginia)

VPC: aws-vpc | vpc-Odfb82baded8f104d

Security Group: default

Please select the subnets where Velostrata workers are created when migrating instances from the respective availability zones:

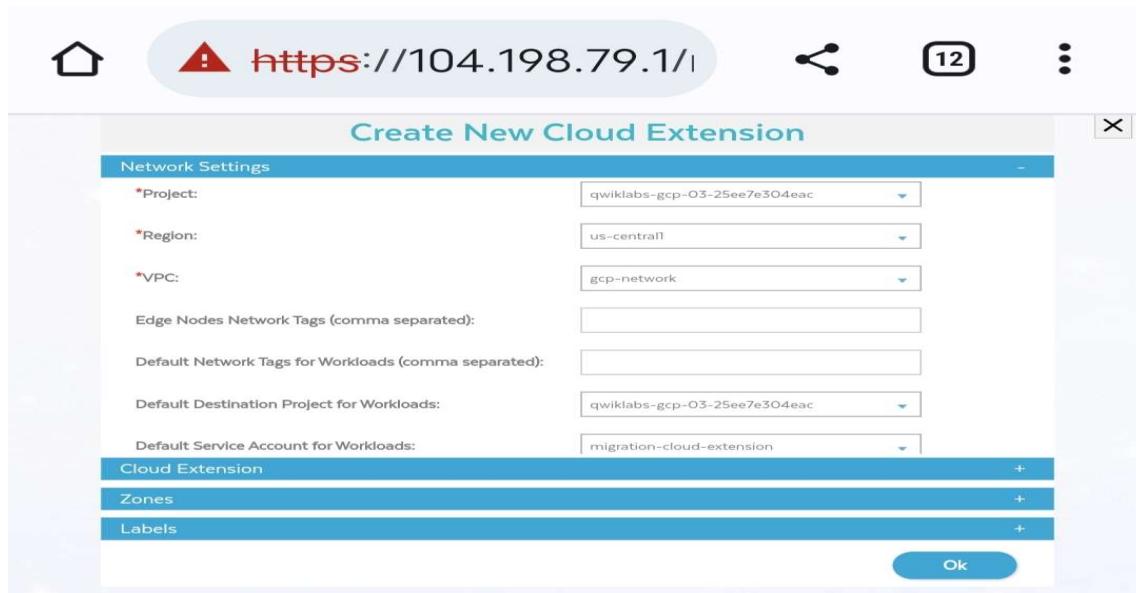
Worker subnet for availability zone: us-east-1e

172.16.0.0/24 | subnet-08698c60f91c36055

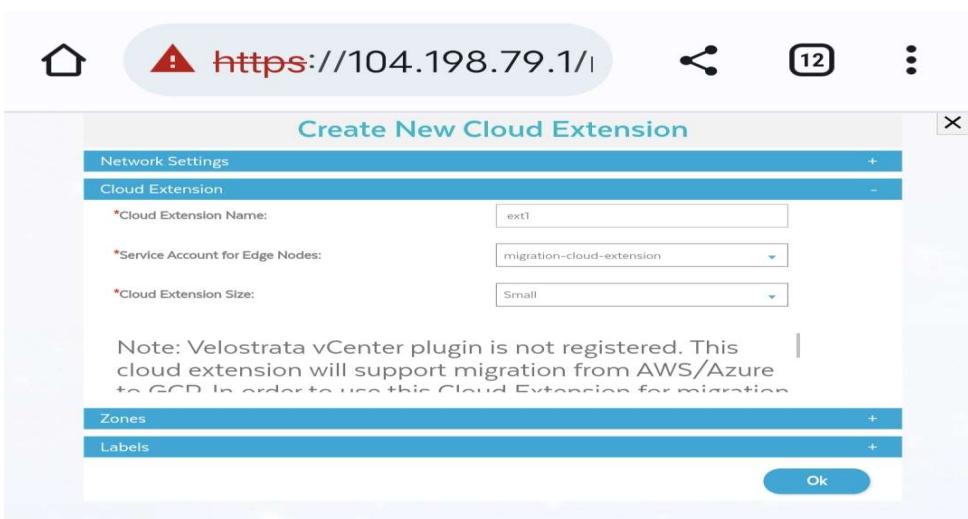
Set of Cloud Extensions

Set up Migrate for Compute Engine Cloud Extensions. A Cloud Extension is a channel for VM storage between two hosting environments, such as between an on-premises data center and Google Cloud, between AWS and Google Cloud, or between Azure and Google Cloud.

- Click the **Target Cloud** icon.
- On the **Cloud Extensions** tab, click **Create**.
- In **Create New Cloud Extension in Network Settings**, specify the following information and leave all other settings as default:



- In **Cloud Extension**, specify the following information:



- Update Zone Details and their subnets

Create New Cloud Extension

Network Settings	+
Cloud Extension	+
Zones	-
*Node A Zone:	us-central1-a
*Node B Zone:	us-central1-b
*Node A Subnet:	10.240.0.0/24
*Node B Subnet:	10.240.0.0/24
*Default Workload Subnet:	10.240.0.0/24
Labels	+

Ok

Note: ext1 should now be in the *Creating* state. Wait until it's in an *Active* state before continuing to the next task. This will take up to 4 minutes.

VELOSTRATA TARGET CLOUD

Cloud Extensions

Name	Cloud Provider	Size	Region	VPC	State	Health Checks
ext1	GCP	Small	us-central1	gcp-network	Creating	

VELOSTRATA TARGET CLOUD

Cloud Extensions

Name	Cloud Provider	Size	Region	VPC	State	Health Checks
ext1	GCP	Small	us-central1	gcp-network	Active	

Generating Migration Waves

In this task you will use a Migrate for Compute Engine feature called a *wave* that batches VMs for migration. Waves are made up of runbooks and jobs, which makes migration more manageable.

- Click **Migration Waves** from Home.



MIGRATION
WAVES

- Click **Generate Runbook**, and provide the following information:

The screenshot shows a user interface for managing migration waves. At the top, there's a blue header bar with the word "Waves". Below it, there are two blue buttons: "Generate Runbook" and "New Wave". A red arrow points to the "Generate Runbook" button. Underneath these buttons is a text input field labeled "Name" with a single character "A" typed into it.

The screenshot shows a detailed configuration dialog for generating a runbook. At the top, it says "Generate Runbook". Below that, there are dropdown menus for "Source" (set to "AWS") and "Source Cloud Details" (also set to "AWS"). There is also a section titled "Filter by Source Tags" with a table:

Name	Value
Name	*

Further down, there are fields for "Target Cloud Extension" (set to "ext1") and "Target Network". A checkbox labeled "Populate with Cloud Extension Defaults" is checked. At the bottom right is a blue "Create" button.

- Click **Create**, .csv file will be downloaded. You will edit this file to configure the migration settings.
- Open Google Sheets in a new tab, click **Blank** to start a new spreadsheet, Click **File > Import**, and Upload tab, **Velostrata Runbook.csv** file into the window.
- In the **Import file** box, select **Comma** as the Separator type and leave all other settings as their defaults, Click **Import data**.

Edit the runbook

1. Change the RunGroup column value from **-1** to **1** and set the **TargetInstanceType** column to be **n1-standard-1**.
2. Save the file back to your local file system: **File > Download > Comma-separated values (.csv, current sheet)**.

Create a new wave

1. Return to your Migrate for Compute Engine Manager (Velostrata Migration) tab, click **New Wave**, and specify the following information:

Property	Value
Wave Name	wave1
Runbook CSV	Your edited CSV file

Note: Be sure to select the CSV file that contains the edited changes and **not** the original one that was generated (assuming that you now have two files).

Validate the wave

- Click on the **wave1** line to select it.
- For **Action**, select **Validate** then **Run Validation**, Click **Yes**.

The status will change to *Validating*, and then to **Passed**.

- Ensure that wave1 line is still selected, and for **Action**, select **New Job**.
- In the **New Job** dialog, select **Full Migration**, and click **Start**.

The status will now change to *Full Migration (Running)*.

Name	Runbook Date	Runbook	Validation Status	Jobs	Last Job (Status)	Monitor	Source
wave1	Dec 5, 2022 8:09 PM		Passed	1	Full Migration (Running)		AWS

The migration is now progressing. Wait for the corresponding Compute Engine to become available.

The migration proceeds in two major phases:

- The first is the start up of the Compute Engine by bringing in enough of the original VM to start.

- From there, the remainder of the VM disk will be streamed in the background.

A record line for the VM being migrated should be displayed. This should take up to 10 minutes.

- Look at the AWS instance. The machine you're migrating should have an **Instance State** of *stopped*.

The screenshot shows the AWS EC2 Instances page. The search bar at the top contains the placeholder "[All S]". Below it, the "Instances (1) Info" section displays a single instance. The instance details are as follows:

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Monitoring	Security group name
aws-vm-us-ea...	i-01d5d8931b24afe3b	Stopped	t2.micro	-	us-east-1e	ec2-54-159-161-6.com...	54.159.161.6	disabled	aws-allow-internet,

- Refresh the browser session to see the most up-to-date information.
- In the Google Cloud Console, on the Compute Engine page, click Refresh until the **aws-vm-us-east-1** machine from AWS appears.

The screenshot shows the Google Cloud Compute Engine VM instances page. The left sidebar includes sections for Virtual machines, Storage, and Instance groups. The main area displays a table of VM instances:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
Green	aws-vm-us-east-1	us-central1-a			ip-velostrata-aws-vm-us-east-1 (10.240.0.5) (nic0)	34.134.148.11 (nic0)	SSH
Green	gcp-vm-us-central1	us-central1-a			10.240.0.100 (nic0)	35.239.66.199 (nic0)	SSH
Green	velo-mgr	us-central1-a			10.240.0.2 (nic0)	104.198.79.1 (nic0)	SSH
Green	velostrata-edge-node-a-f5567928-1d57-4863-a631-719df5490d93	us-central1-a			ip-velostrata-edge-node-a-f5567928-1d57-4863-a631-719df5490d93 (10.240.0.3) (nic0)	34.66.123.95 (nic0)	SSH
Green	velostrata-edge-node-b-f5567928-1d57-4863-a631-719df5490d93	us-central1-b			ip-velostrata-edge-node-b-f5567928-1d57-4863-a631-719df5490d93 (10.240.0.4) (nic0)	34.133.19.218 (nic0)	SSH

7. Monitor the **Migration Status** in the Velostrata Migration Waves dashboard until it changes from *empty* or *Moving to Target Cloud* to one of these:

When this status changes, the VM is available on Google Cloud and is ready to be used. When the VM has been **100%** migrated, the status will change to **Fully Migrated**.

```
student 04_67df50181fe5@cloudshell:~ (qwiklabs-gcp-03-25ee7e304eac)$ ssh -i ~/.ssh/vm-ssh-key ubuntu@34.134.148.11
The authenticity of host '34.134.148.11 (34.134.148.11)' can't be established.
ECDSA key fingerprint is SHA256:EM4a+h65Y3eumWBuoIWyu9QRDxzalqQH+nJYrph8yu4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.134.148.11' (ECDSA) to the list of known hosts.
Last login: Mon Dec  5 13:34:06 2022 from 34.126.125.83
ubuntu@ip-172-16-0-100:~$ date
Mon Dec  5 15:11:06 UTC 2022
ubuntu@ip-172-16-0-100:~$ ls
MyText.txt  velostrata-prep=0.9-3.deb
ubuntu@ip-172-16-0-100:~$ uptime
 15:11:22 up 4 min, 1 user, load average: 0.09, 0.25, 0.15
ubuntu@ip-172-16-0-100:~$
```

We have successfully Migrate the VM running from AWS to GCP and able to login to the server .

8. Scenarios

9. Technology Road Map / Blueprints

10. GCP Service and Products

GCP

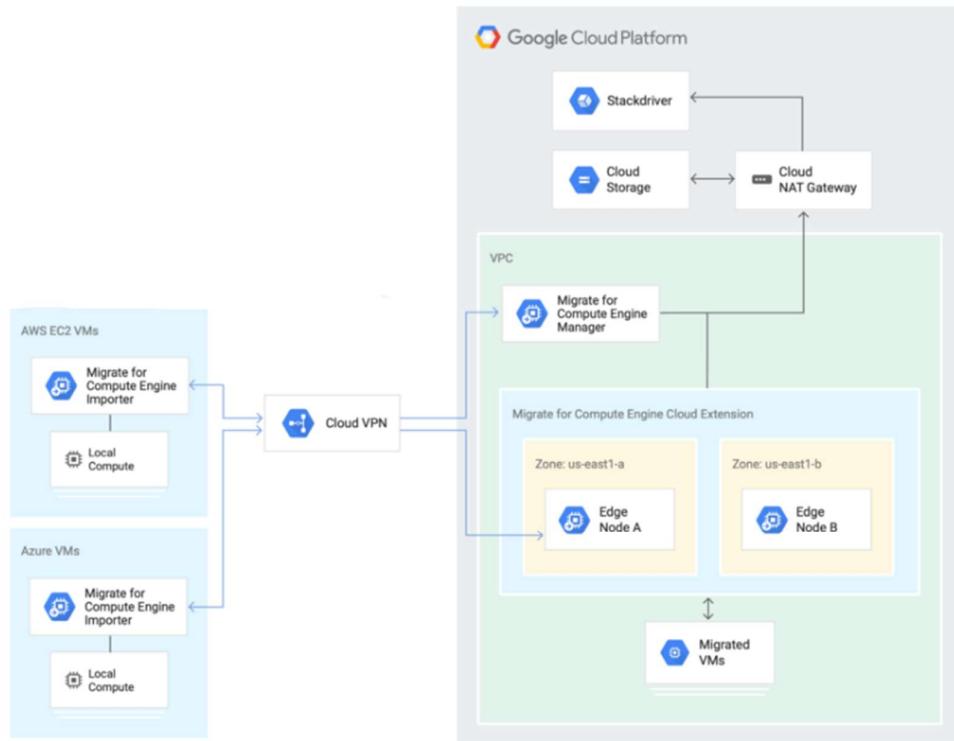
- Google Compute Engine
- IAM Service Accounts.
- API's Services
- Migration Manager - Velostrata
- Hybrid Connectivity - VPN

Other Tools

- Windows PowerShell
- PUTTY
- RDP
- Docker
- WebLogic Container

11. Implementation Diagram

Target Architecture Diagram



12. Conclusion

VM Migration from Azure/AWS to Google cloud can be done through migration manager **Velostrata** by establishing the connection between source and destination clouds .

With free trial we are unable to migrate the VMs from azure to GCP due to some limitation however we have completed our POC by migration VMs from AWS to GCP.

We successfully have migrated the VM from one cloud to another cloud .