Migrate WebLogic Instance running on AZURE to GCE

Table of Contents

[1. Introduction 2](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291857)

[2. “The Requirement” 3](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291858)

[3. Revision History 4](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291859)

[4. Challenge/ Problem Statement 5](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291860)

[5. Current way of Working in on-prem 8](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291861)

[6. Considerations](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291862) 9

[7. Proposed Solution &Transformation Details](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291863) 36

[8. Scenarios](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291864) 37

[9. Technology Road Map / Blueprints](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291865) 38

[10. GCP Service and Products](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291866) 39

[11. Implementation Diagram & Terraform Scripts](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291867) 40

[12. Conclusion 41](file:///C:\Users\vmatha\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XP3ELCPI\MSSQL%20on%20GCP-v0.3.docx#_Toc80291868)

# Introduction

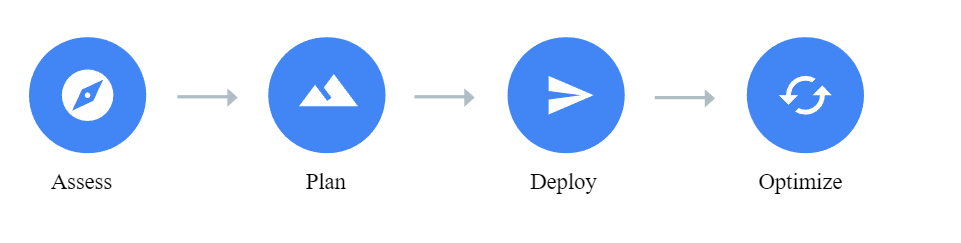
In our test case we are going to Migrate the WebLogic Application running on AZURE Cloud to Google cloud. we have used the migration applicane Velostrata.

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.

# “The Requirement”

1. **Objective**:

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

1. **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

# Revision History

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

# 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 limitiation 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.

# Current way of Working in on-prem

# 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.

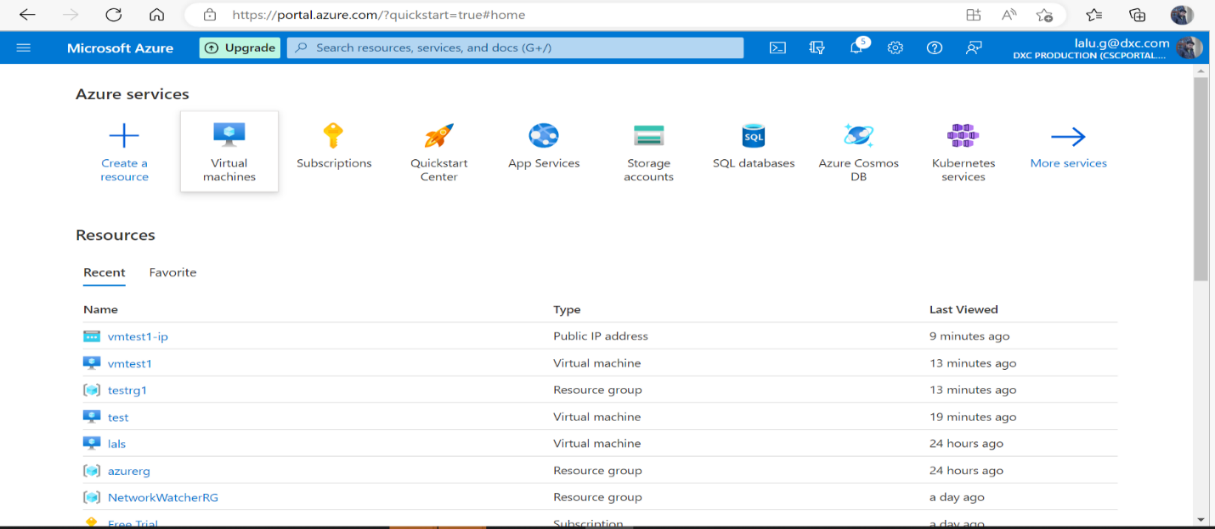
# Proposed Solution &Transformation Details

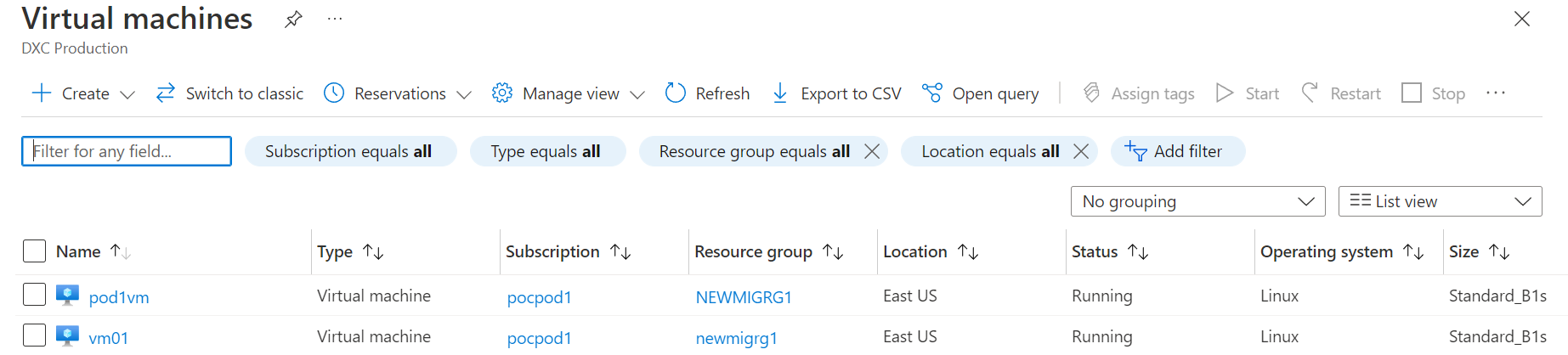
Create the Source VM running web logic application on Azure

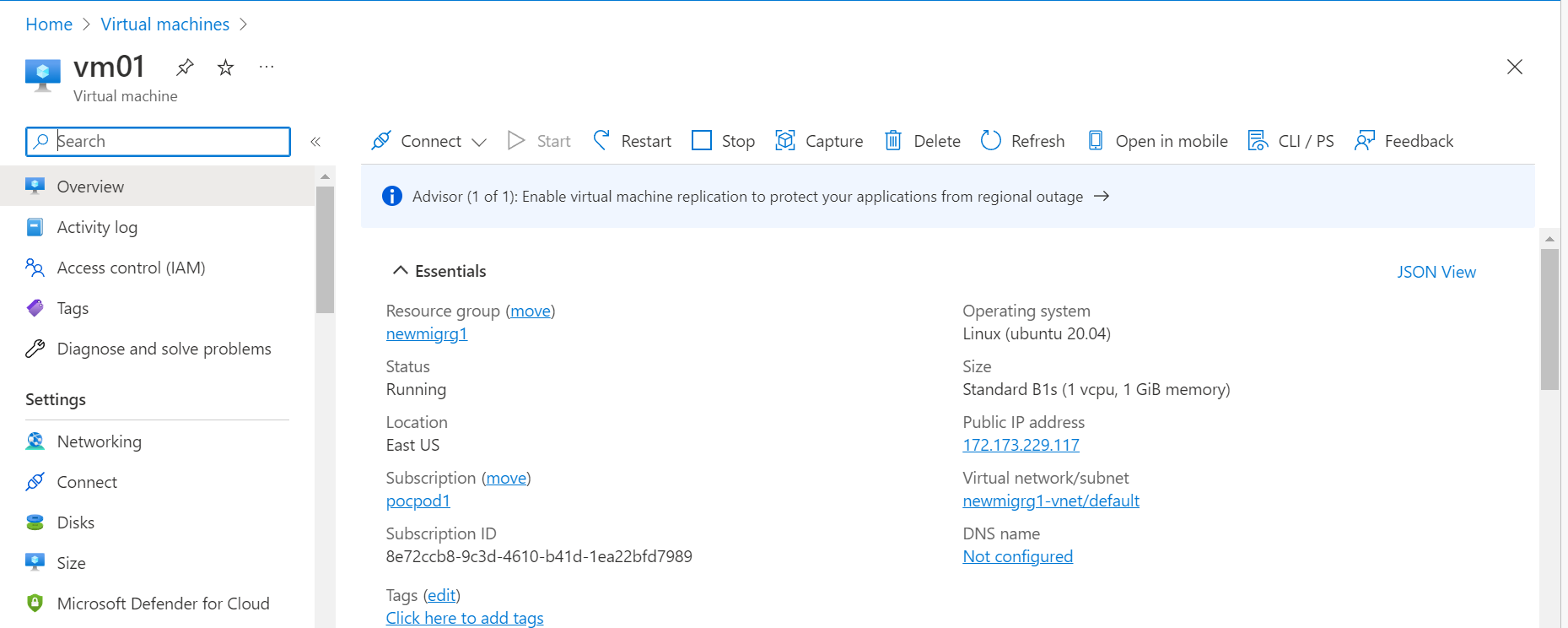
## Sign into Azure

Sign into the Azure portal at [https://portal.azure.com](https://portal.azure.com/).

## Create virtual machine

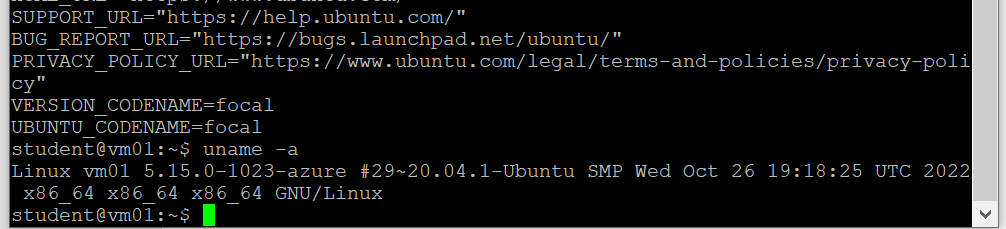
* Under Azure **Services**, select **Virtual machines**.
* In the **Virtual machines**, select **Create** and then **Azure virtual machine**.
* Under **Instance details**, enter name for the **Virtual machine** and choose operating system as CentOS.
* Leave the remaining configuration as defaults and 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.
* After deployment is complete, select **Go to resource**





## Connect to virtual machine

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



**Install Docker Engine on VM.**

To get started with Docker Engine on your VM, make sure you [meet the prerequisites](https://docs.docker.com/engine/install/centos/#prerequisites), then [install Docker](https://docs.docker.com/engine/install/centos/#installation-methods).

## 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

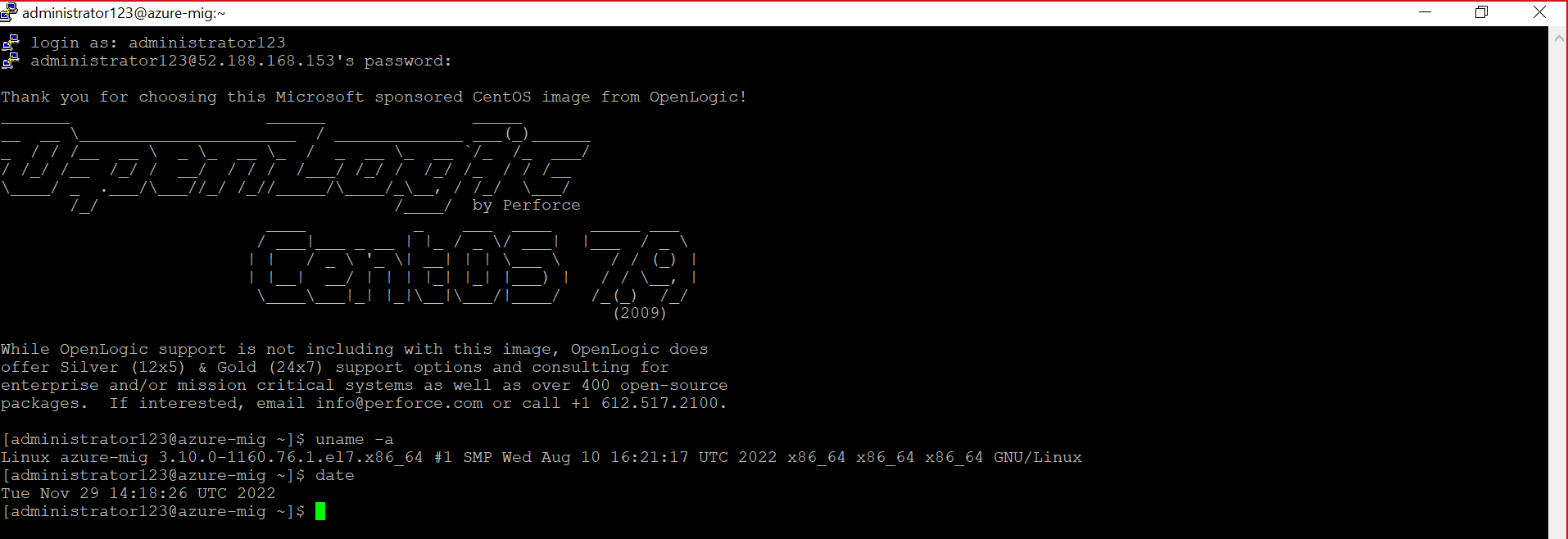
Install the yum-utils package (which provides the yum-config-manager utility) and set up the repository.

**$ sudo yum install -y yum-utils**

**$ sudo yum-config-manager \**

**--add-repo \**

[**https://download.docker.com/linux/centos/docker-ce.repo**](https://download.docker.com/linux/centos/docker-ce.repo)



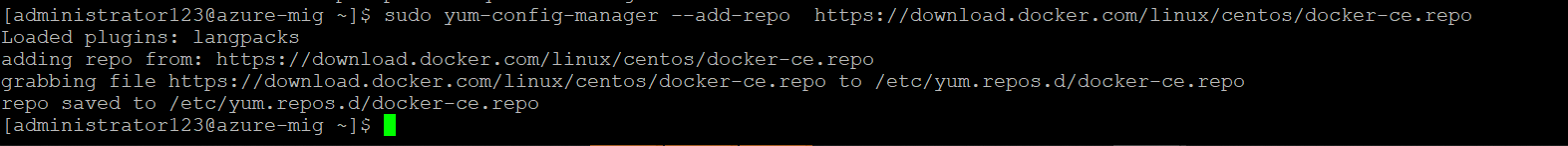
#### 

#### 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**





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**

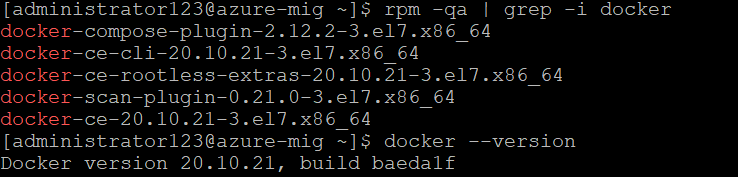
The list returned depends on which repositories are enabled and is specific to your version of CentOS (indicated by the.el7 suffix in this example).

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.e17**

**$ sudo yum install docker-ce-20.10.21.-3.e17. x86\_64 docker-ce-cli-20.10.21.-3.e17. x86\_64 containerd.io docker-compose-plugin**



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.

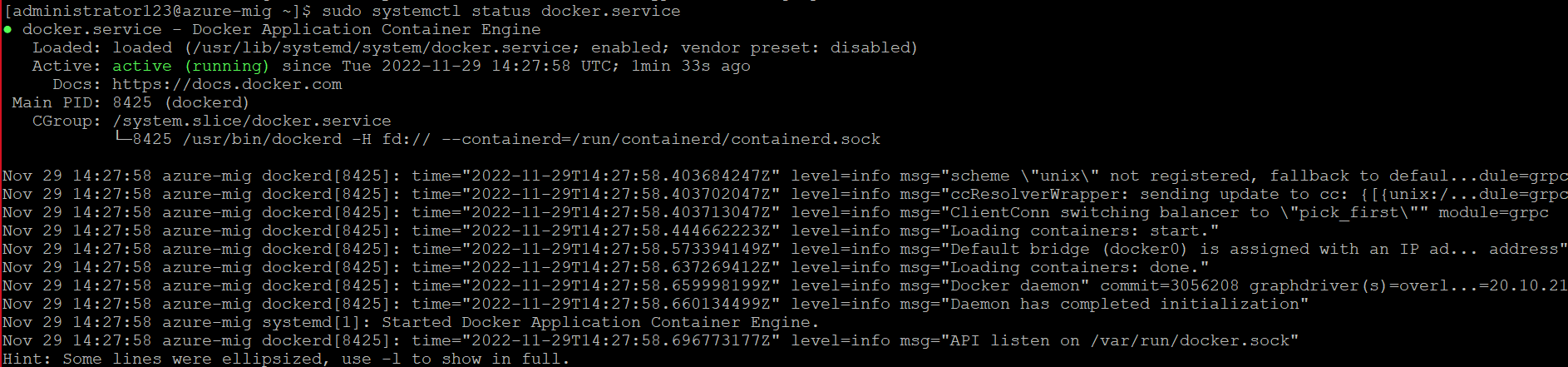
1. Start Docker.

**$ sudo systemctl status docker. service**

**$ sudo systemctl start docker. service**

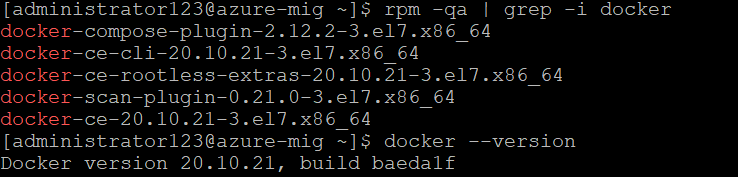
**$ sudo systemctl enable docker. service**

**$ sudo systemctl start docker**

**$ sudo systemctl list-unit-files –type=service |grep -I docker**

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

docker –version



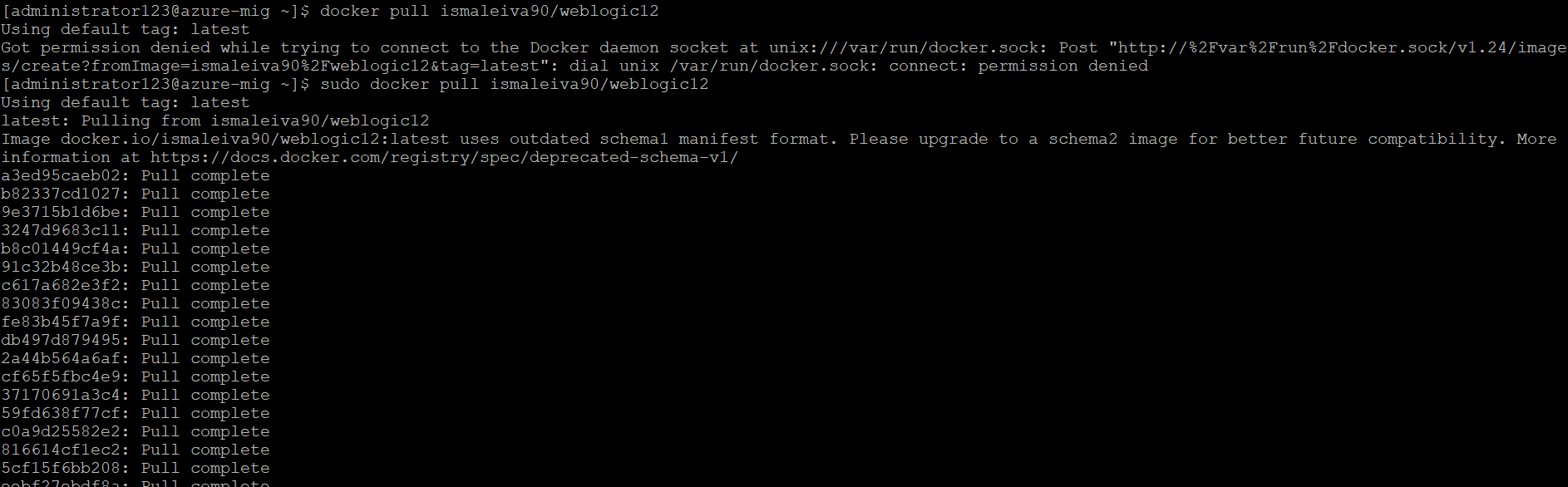


Using the below link you can navigate the docker hub which has docker images available.

[**https://hub.docker.com/r/ismaleiva90/weblogic12**](https://hub.docker.com/r/ismaleiva90/weblogic12)

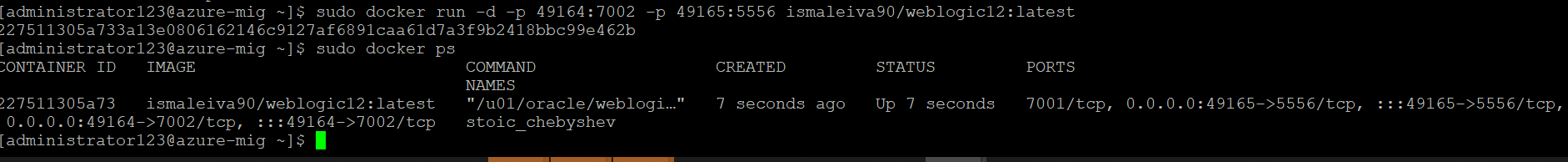
Use the below command to pull the docker image from docker hub, It may take some time as it has downloaded all the dependencies.

**docker pull ismaleiva90/weblogic12**

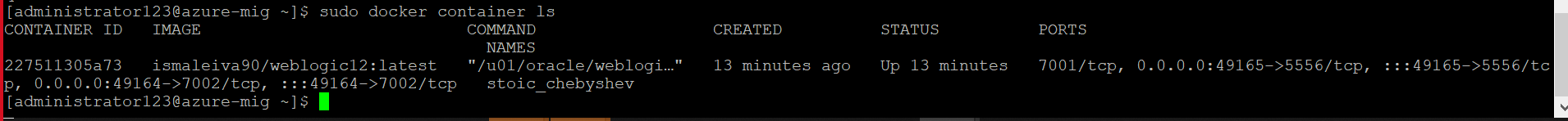


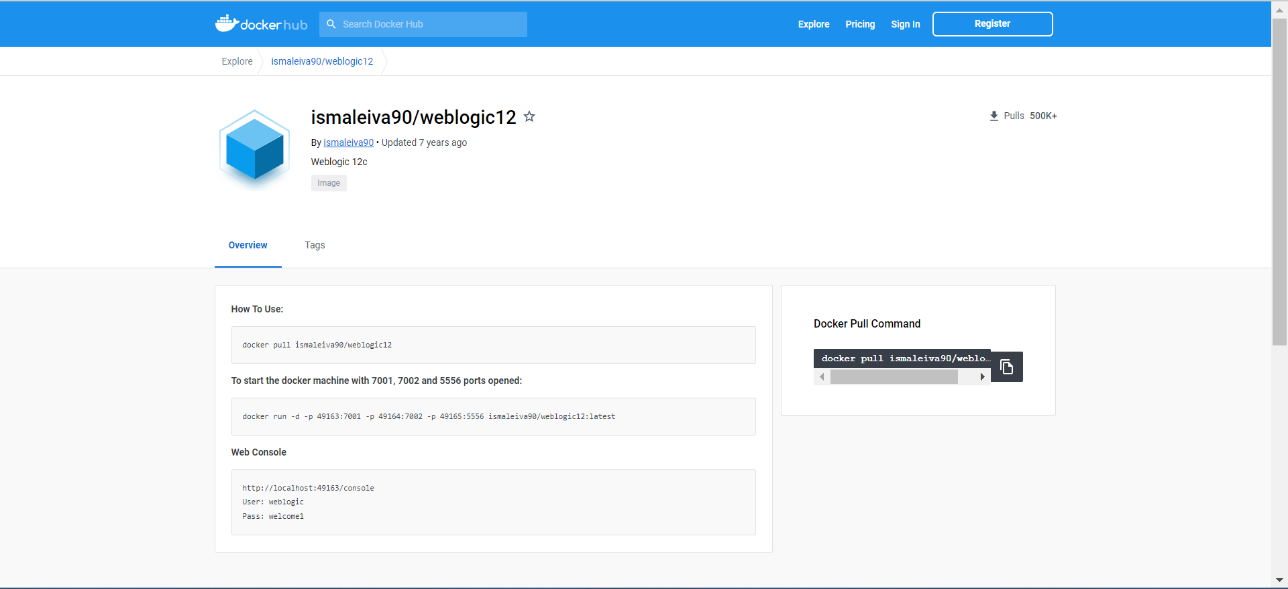
#### To start the docker machine with 7001, 7002 and 5556 ports opened:

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



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

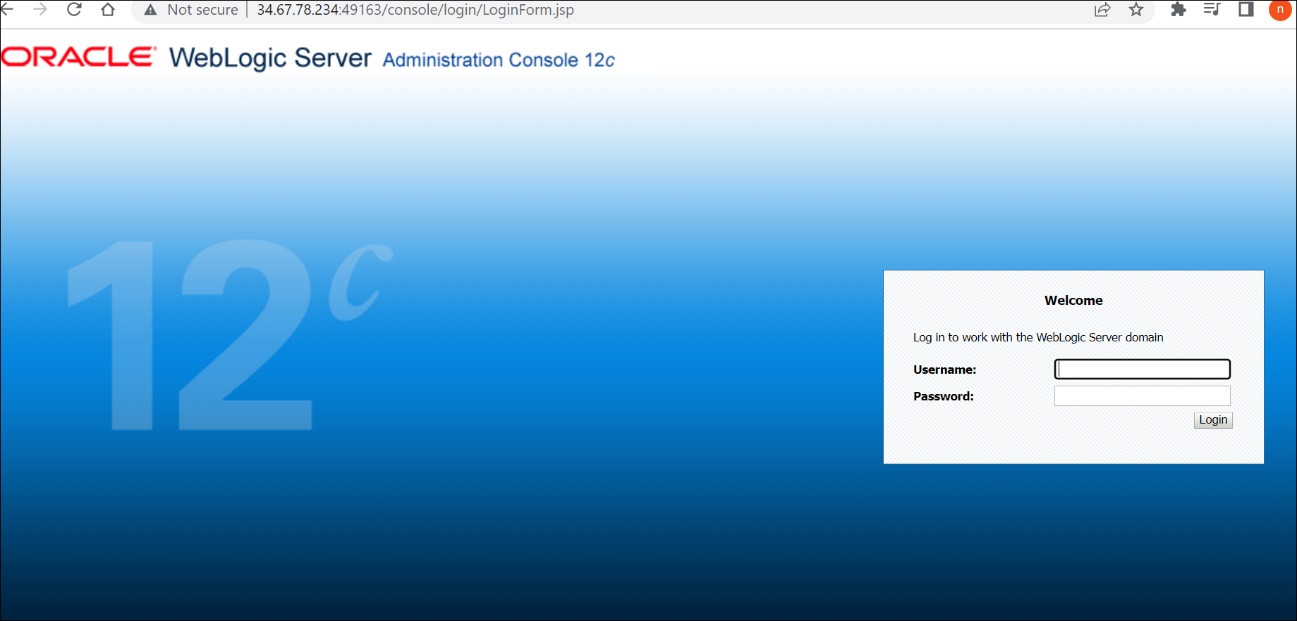




To login your web console use below link as mentioned, you can give your vm public ip to access the GUI.

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.

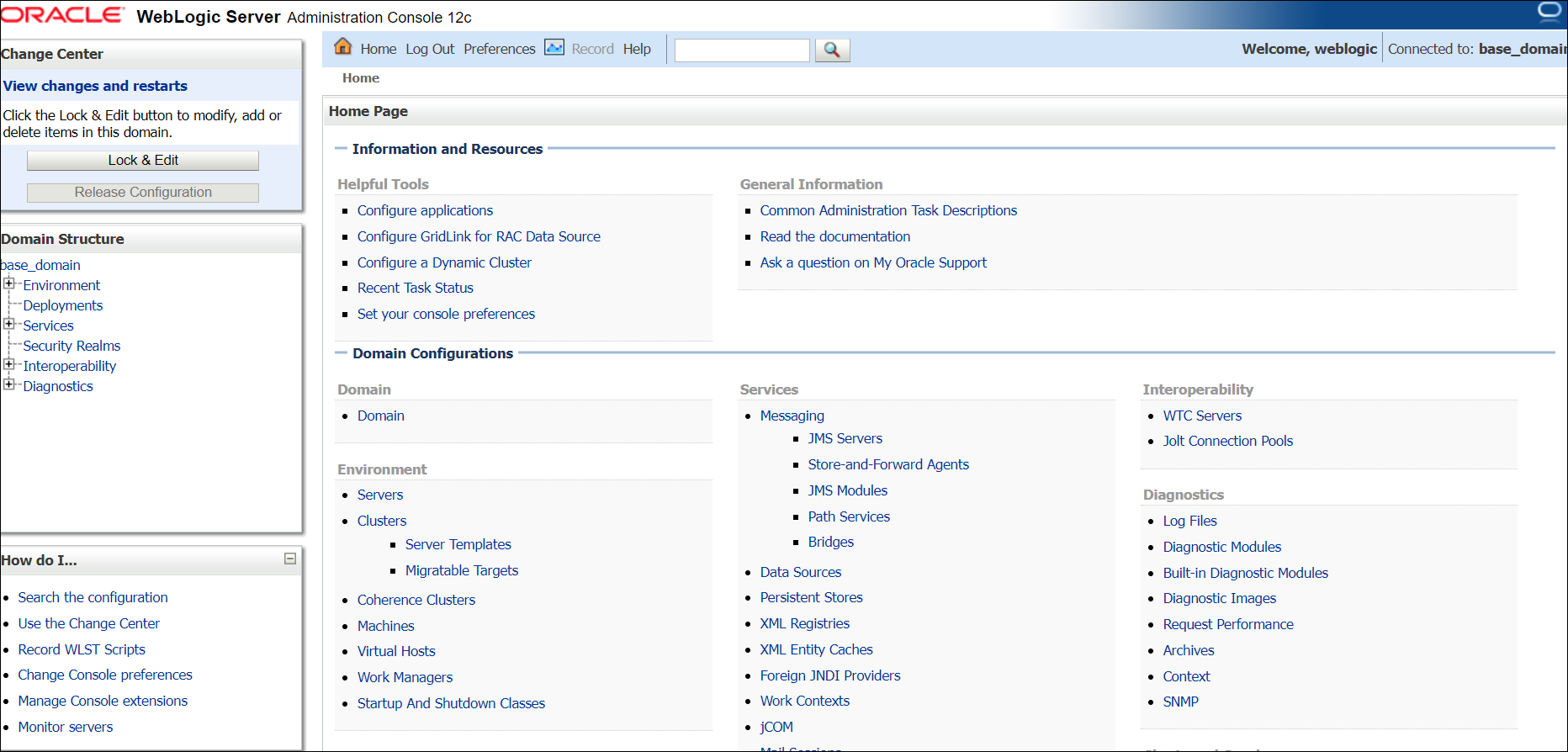


User: weblogic

Pass: welcome1

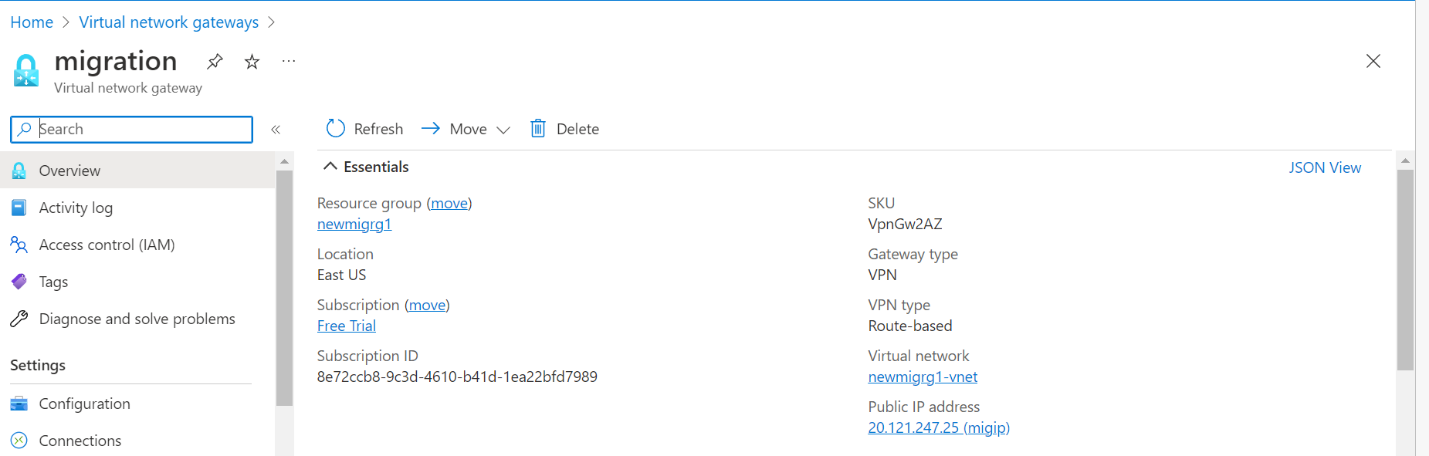
Once you provided the credentials you can be able to go to the Oracle Web logic server.

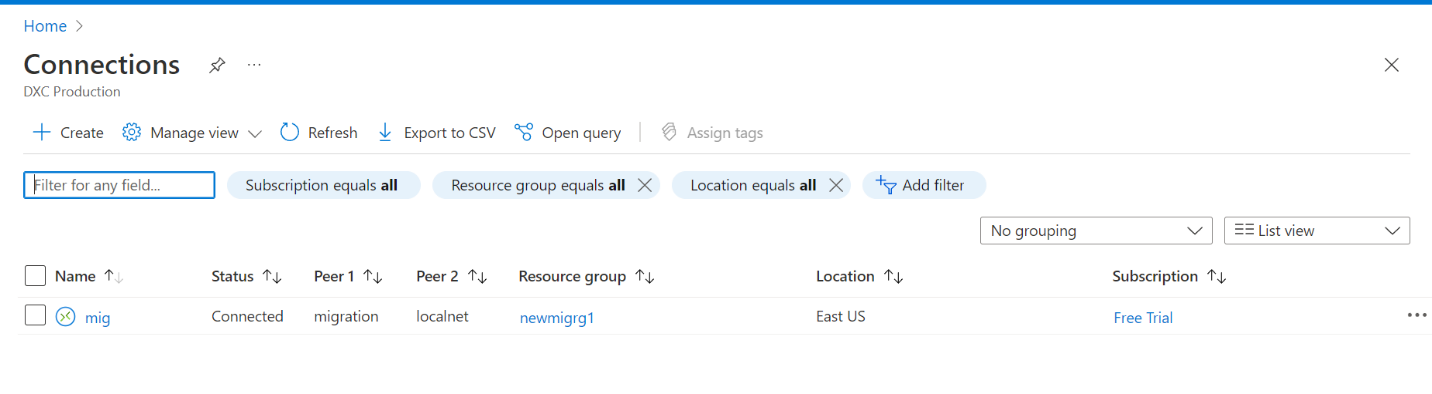
Note: If you are not able to allow, then you may need to check from your firewall rule and need to enable firewall rule if it is not allowed.



## Configure Virtual Network Gateway:

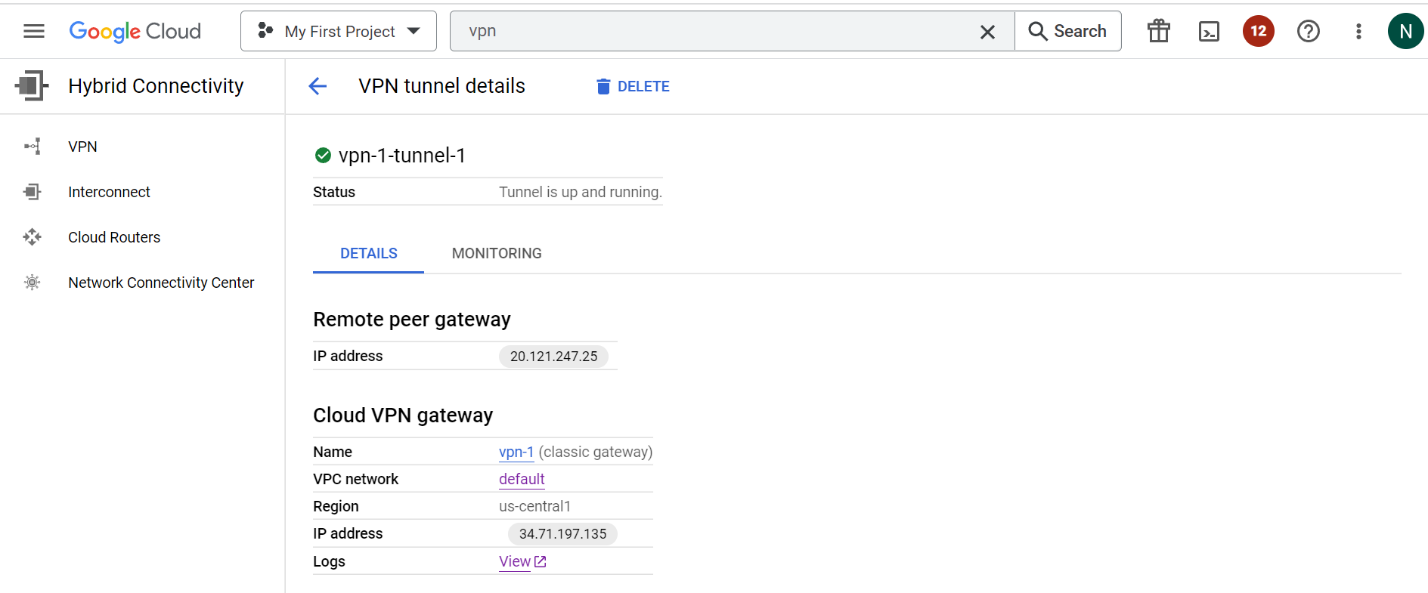
A virtual network gateway is composed of two or more VMs that are automatically configured and deployed to a specific subnet you create called the gateway subnet. The gateway VMs contain routing tables and run specific gateway services. You can't directly configure the VMs that are part of the virtual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



**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



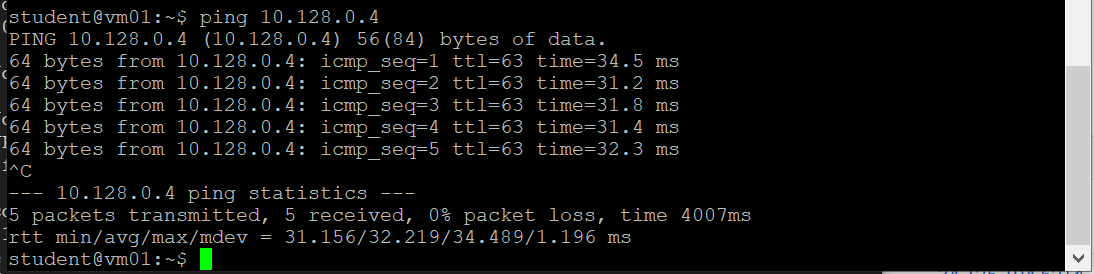
**Connection Established between Azure and GCP**

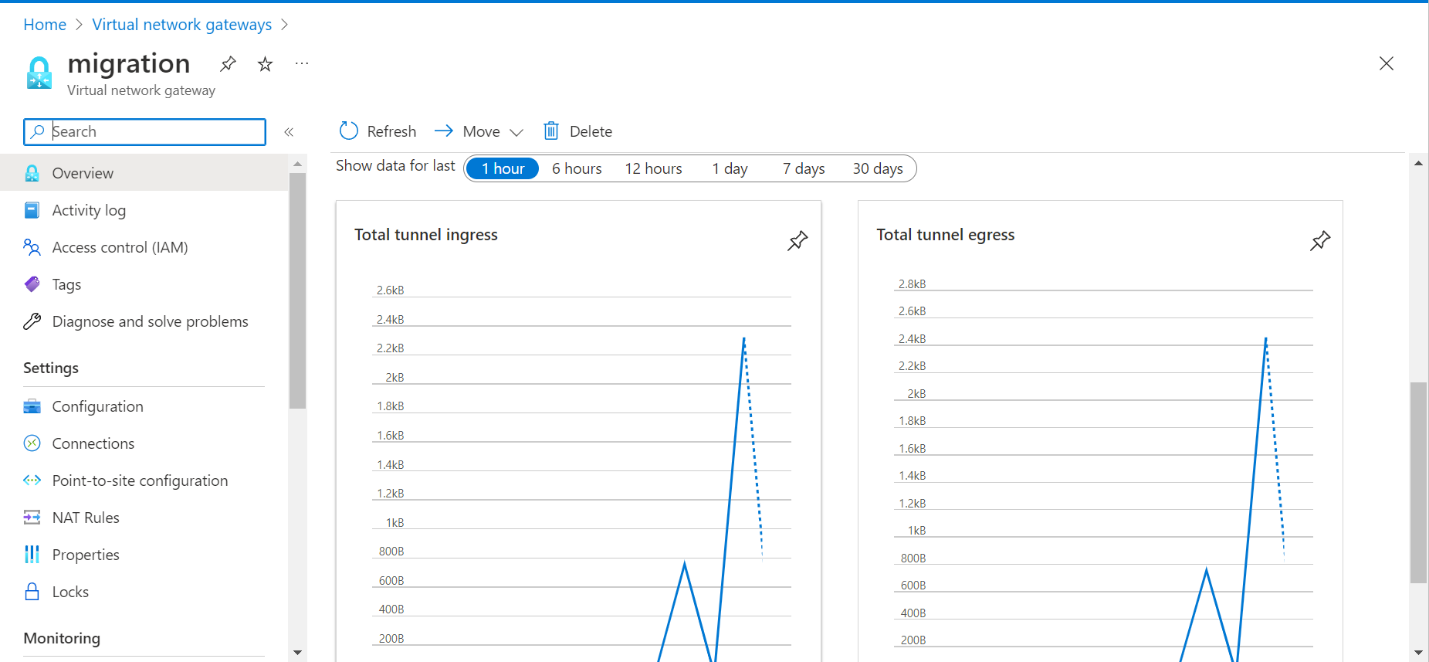
VPN connectivity check:

**Ping test from GCP to Azure:**



**Ping test from Azure to GCP:**





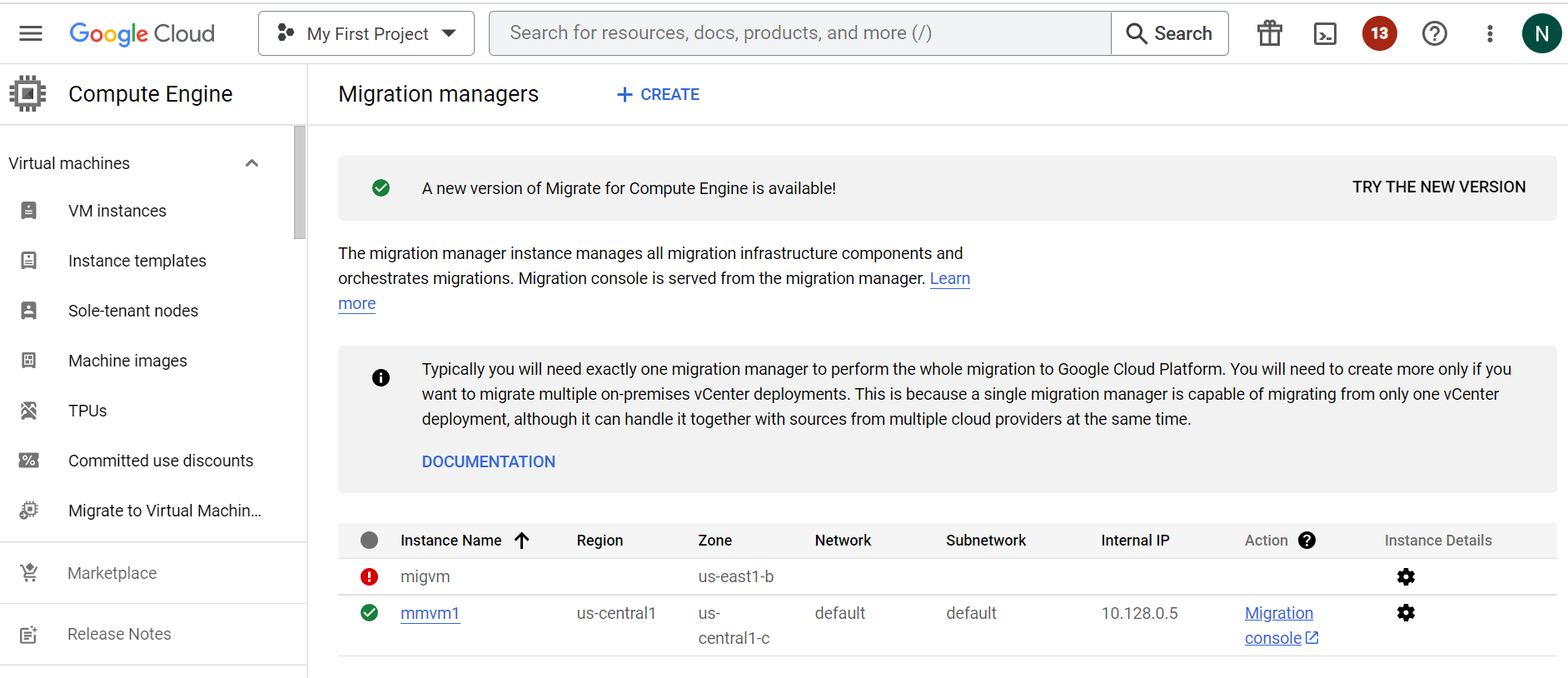
**Creating Migration Manager:**

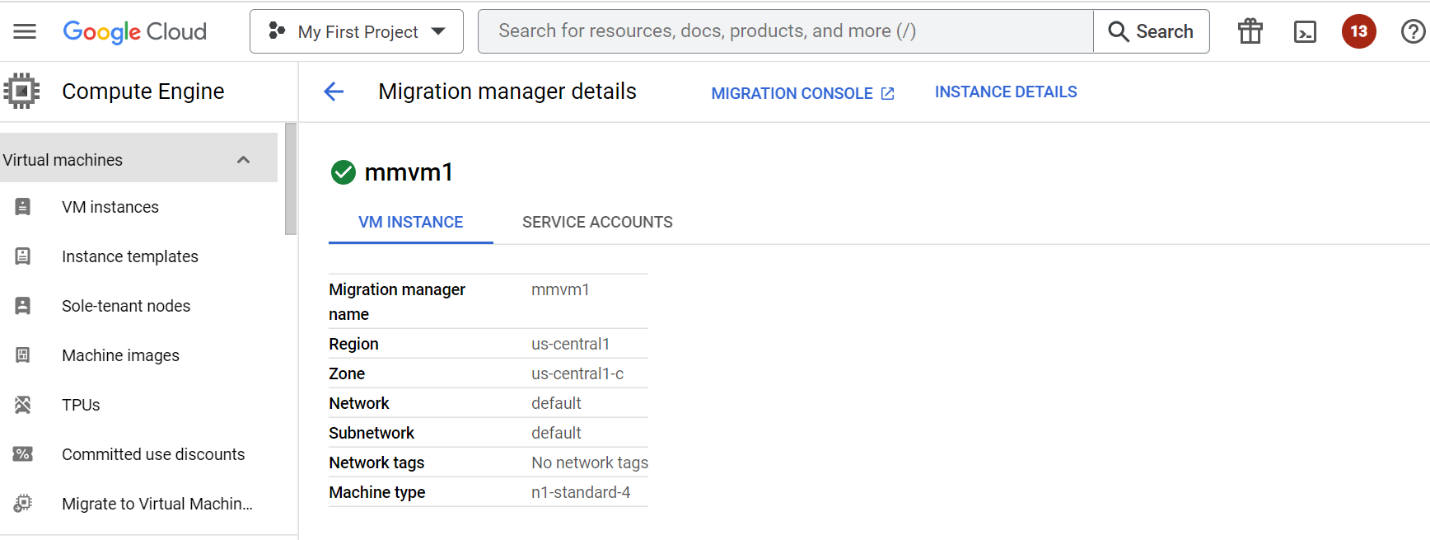
To configure the Migrate for Compute Engine Manager on Google Cloud. The migration manager instance manages all migration infrastructure components and orchestrates migrations.

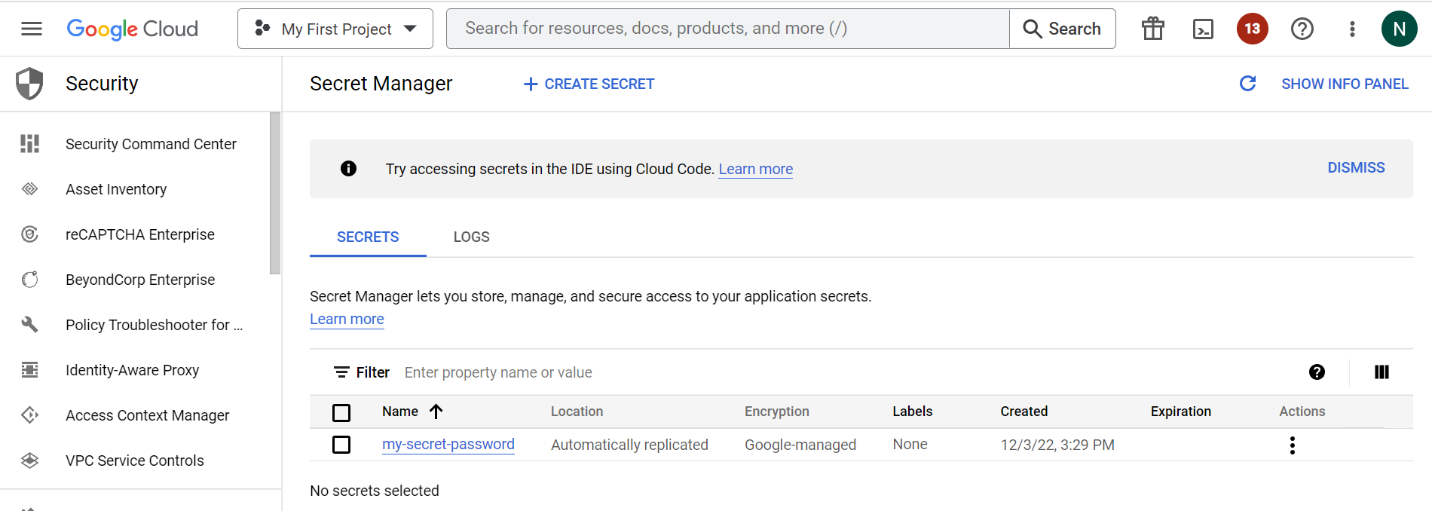
* Configure required network connectivity.
* Configure firewall rules.
* Create service accounts.
* Create migration manager passwords
* Enable Required APIs.

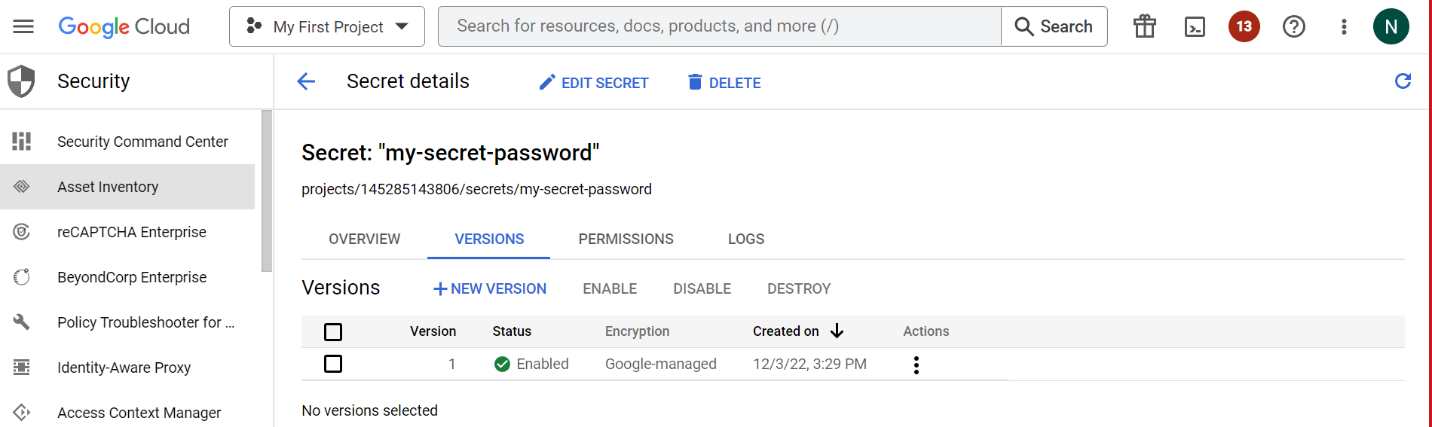
Refer below document to configure - **Configuring the migration manager**

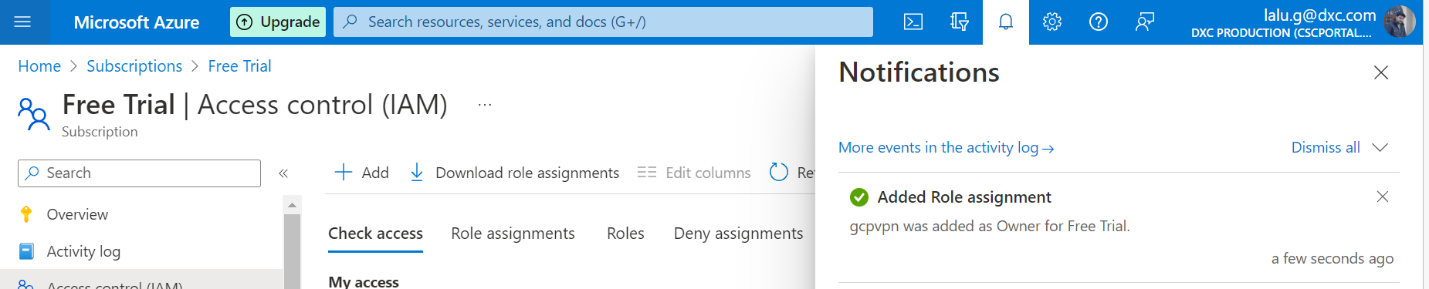
[https://cloud.google.com/migrate/compute-engine/docs/4.11/how-to/configure-manager/configuring-migration-manager?\_gl=1\*1bp76hh\*\_ga\*MTM1MDM4MTIzMy4xNjYxMjQ5ODc5\*\_ga\_WH2QY8WWF5\*MTY3MDMzNDE3MC4yMS4xLjE2NzAzMzQ1MTYuMC4wLjA.&\_ga=2.45972324.-1350381233.1661249879&\_gac=1.81212773.1670334171.Cj0KCQiA7bucBhCeARIsAIOwr-9CCar82Cb2wa5T\_QlIuG32I7OD5w2Wu8Bud-Wj2K9WK0-GtPDgDiIaAtgsEALw\_wcB](https://cloud.google.com/migrate/compute-engine/docs/4.11/how-to/configure-manager/configuring-migration-manager?_gl=1*1bp76hh*_ga*MTM1MDM4MTIzMy4xNjYxMjQ5ODc5*_ga_WH2QY8WWF5*MTY3MDMzNDE3MC4yMS4xLjE2NzAzMzQ1MTYuMC4wLjA.&_ga=2.45972324.-1350381233.1661249879&_gac=1.81212773.1670334171.Cj0KCQiA7bucBhCeARIsAIOwr-9CCar82Cb2wa5T_QlIuG32I7OD5w2Wu8Bud-Wj2K9WK0-GtPDgDiIaAtgsEALw_wcB)

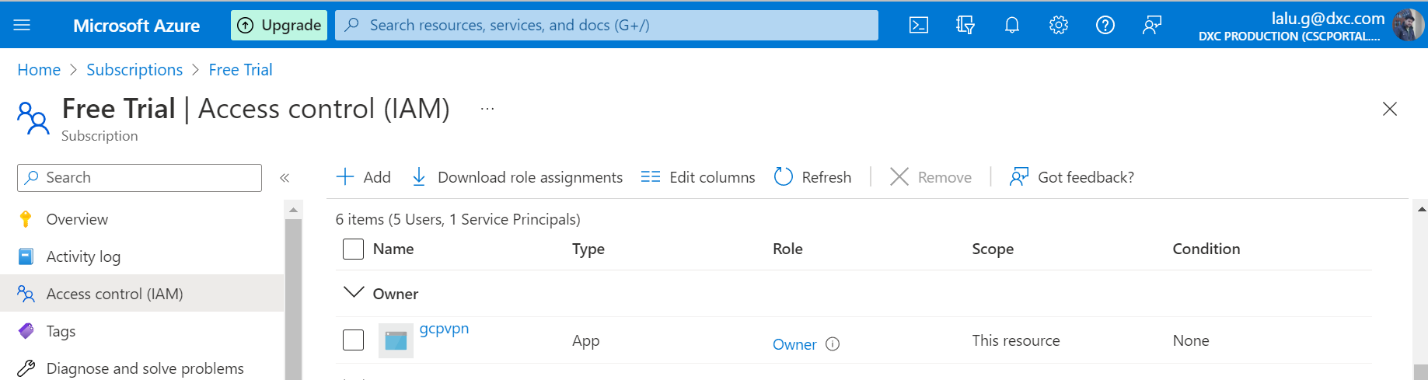






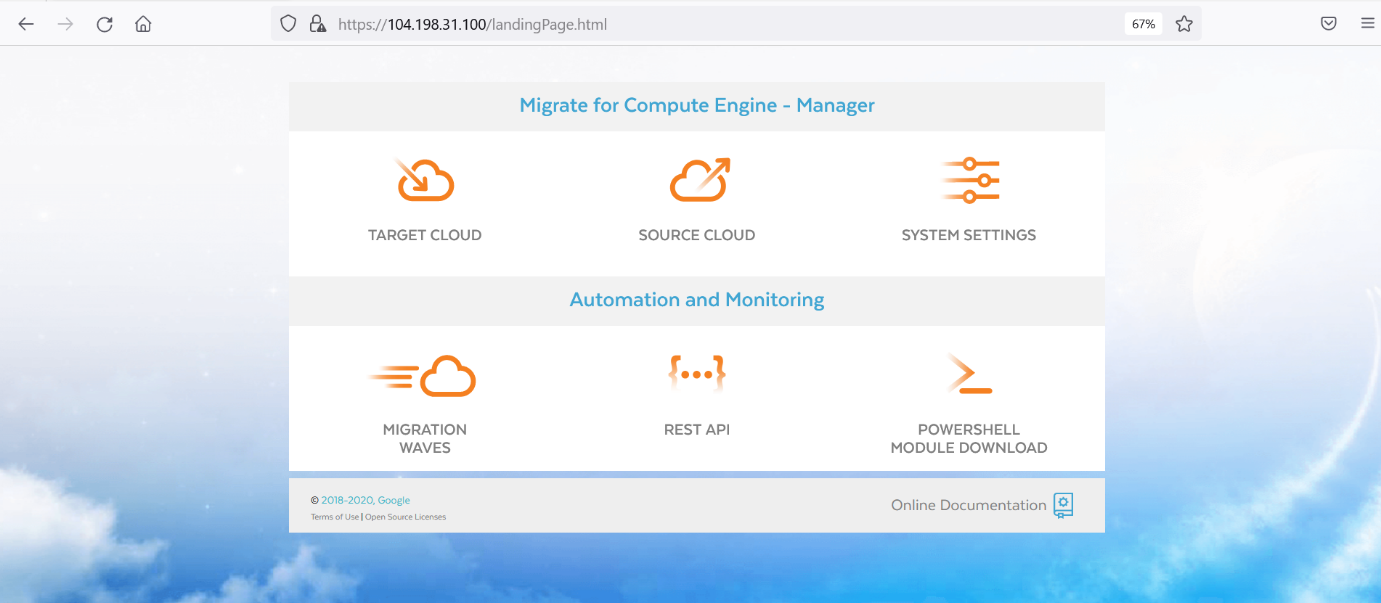






**Velostrata tool for Migration:**

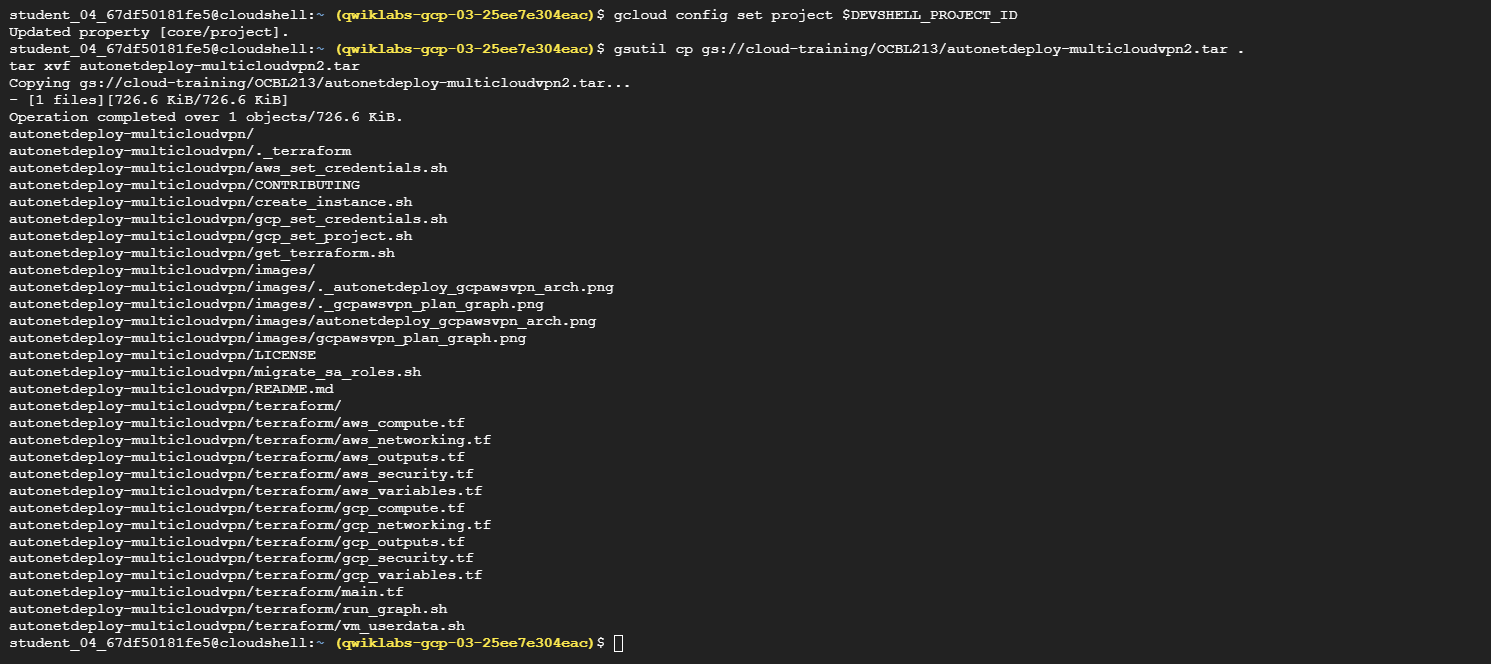
we have created migration manager with velostrata-mgmt-4-11-11-36321-pantheon and secret manager in GCP successfully and able to access the migration appliance.

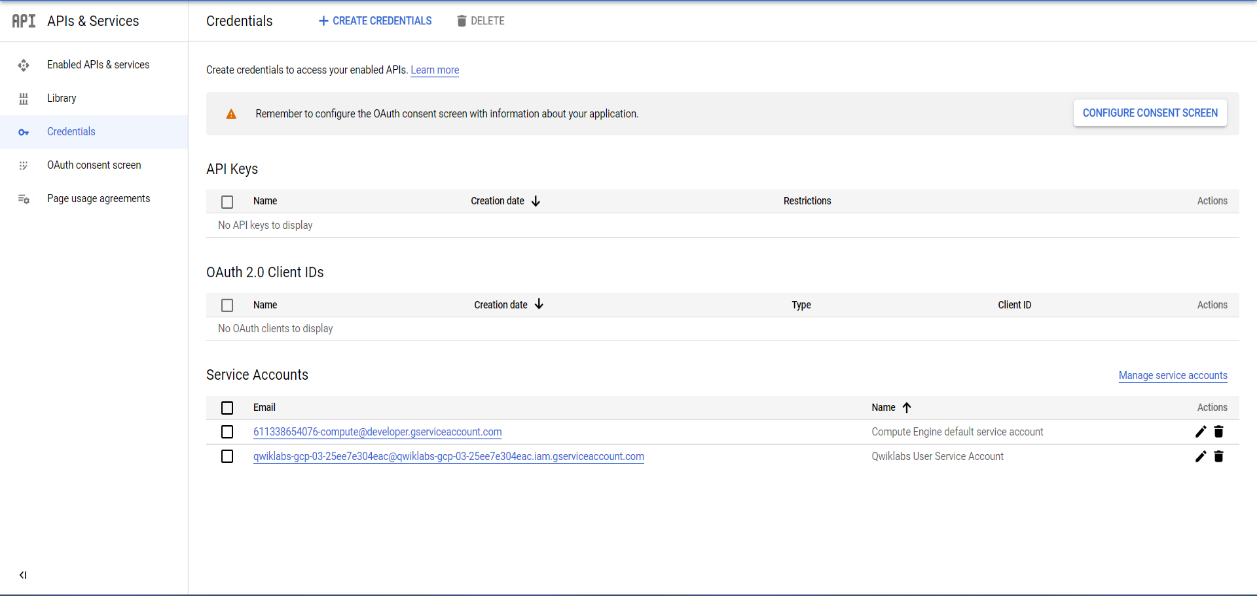
Please find the below details to access Migrate for Compute Engine-Manager.

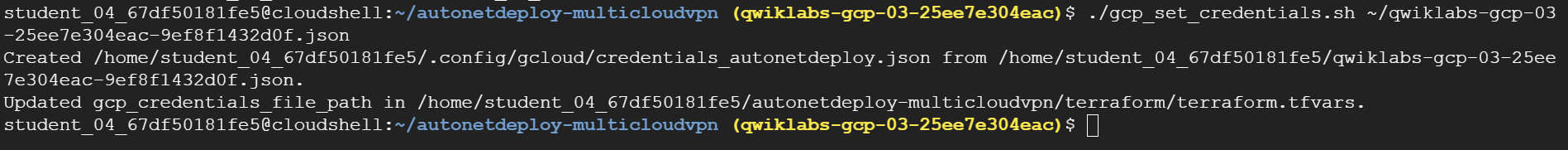
**Graphical user interface, table

Description automatically generatedDue 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.**

VM Migration from AWS to Google Cloud

* Set your Google Cloud Project ID and retrieve the Terraform scripts.
* APIs & Services> Credentials and Uploaded the JSON File

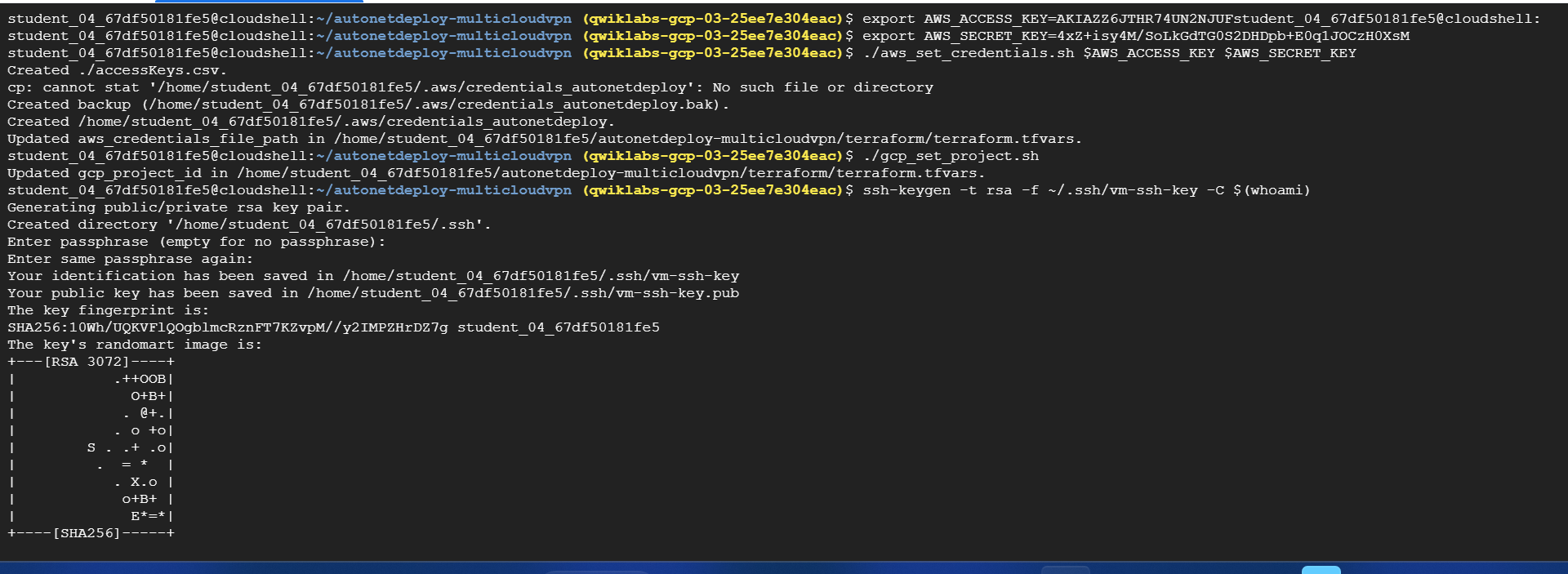


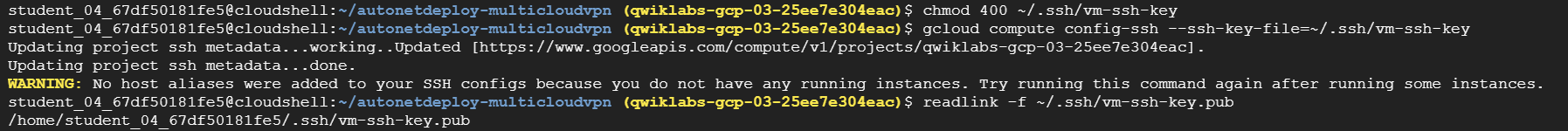


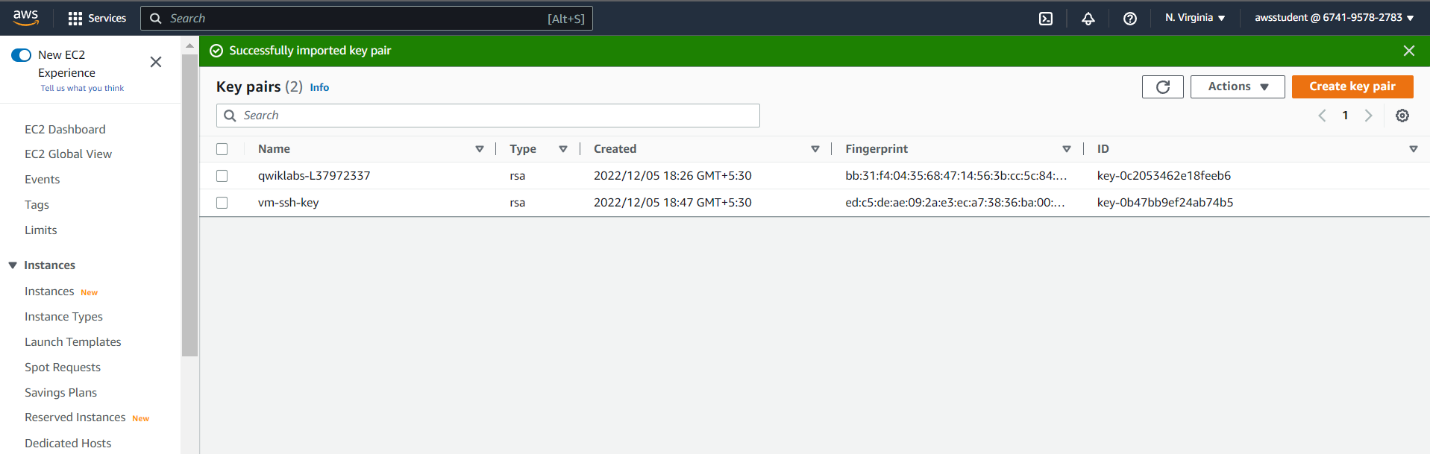
* Set AWS Account credentials and deploy the Terraform

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)



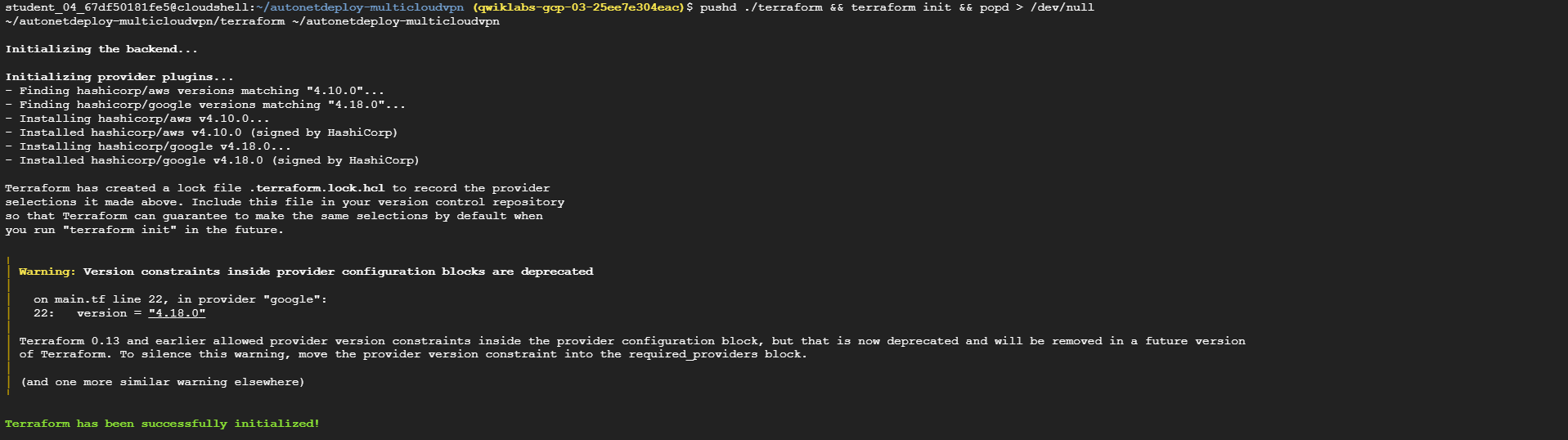


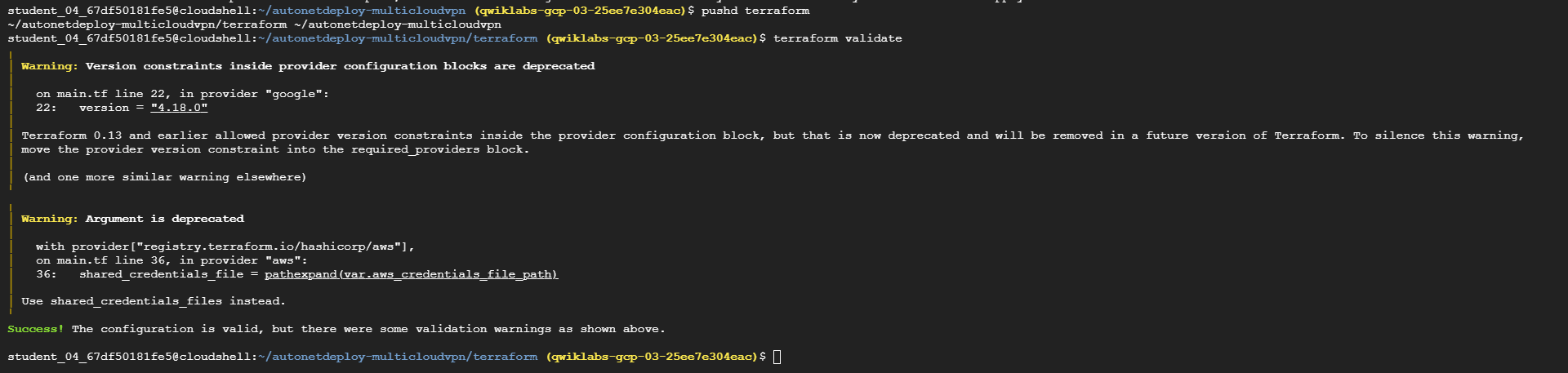
**Deploy the Terraform**

Install the Terraform providers and verify your credentials –

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

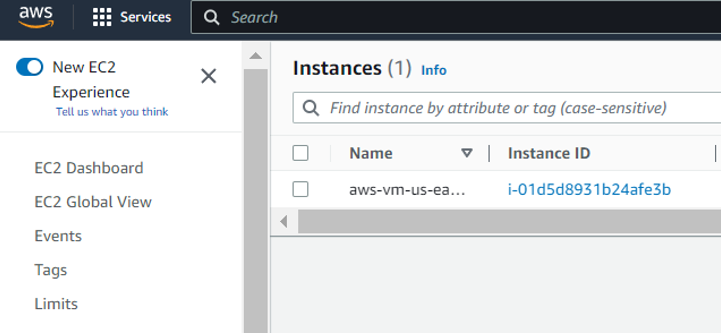
pushd ./terraform && terraform plan && popd > /dev/null

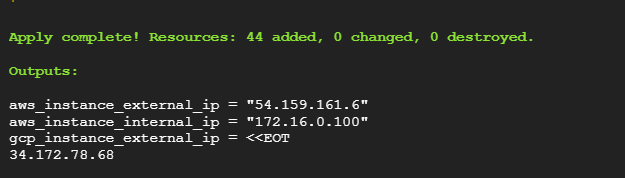




**Configure AWS EC2 Instance for Migration**

**AWS Instance :**





**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]



* **Install and configure Apache:**

sudo bash -c "apt-get update"

sudo bash -c "apt-get install apache2 -y"

echo "Hello World" > MyText.txt

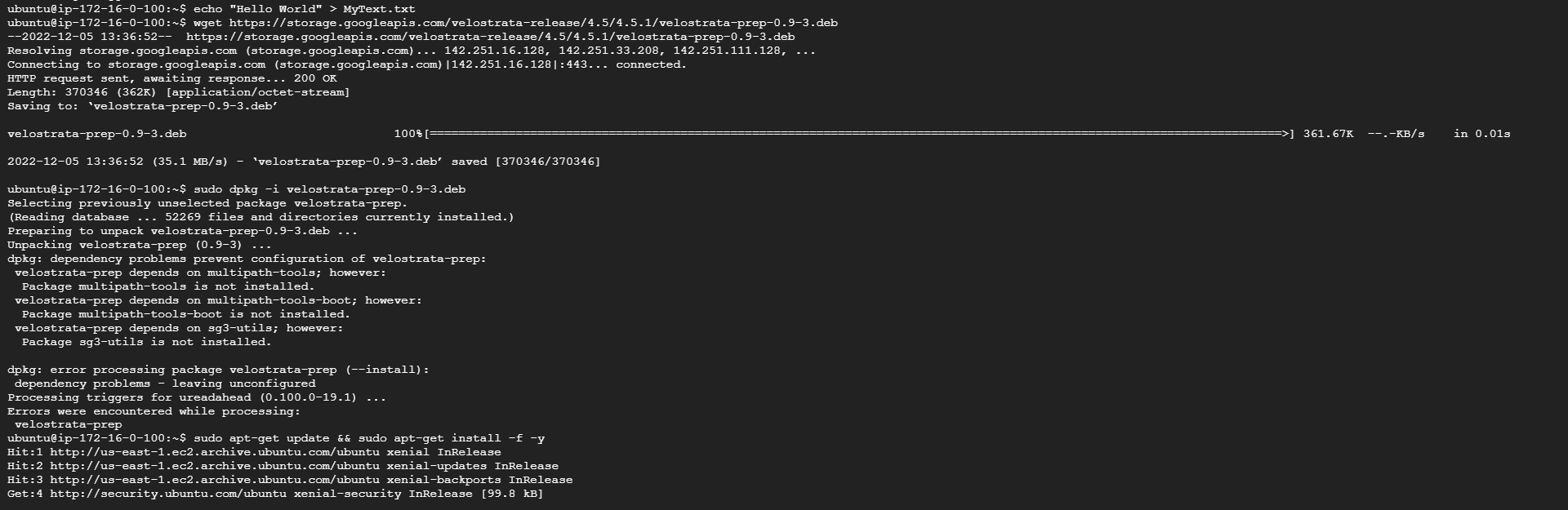
* When the Linux image runs on Google Cloud, it expects to find kernel drivers for the Migrate for Compute Engine mapped disks. These must be downloaded and installed before the migration up to Google Cloud. The driver installation must be performed on the EC2 machine.
* To download the installation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



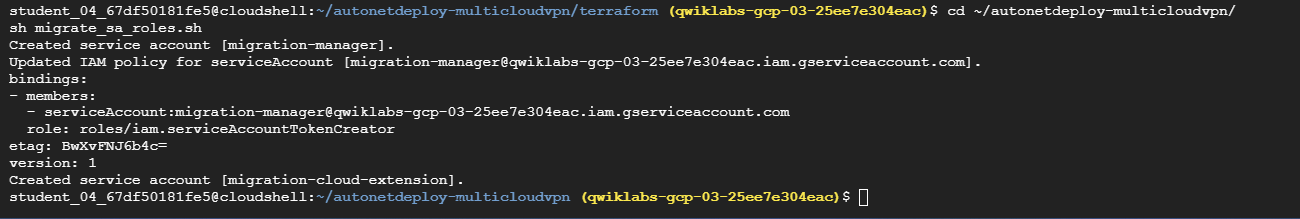
**Set up Migration Service Accounts**

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

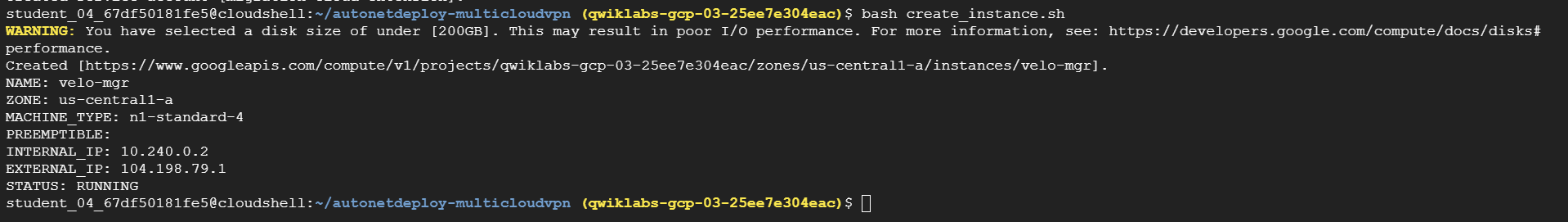
* To create the service accounts that you will assign to Migrate for Compute Engine –

cd ~/autonetdeploy-multicloudvpn/

sh migrate\_sa\_roles.sh



**Set up Migration Manager**

* To create the instance - bash create\_instance.sh
* 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

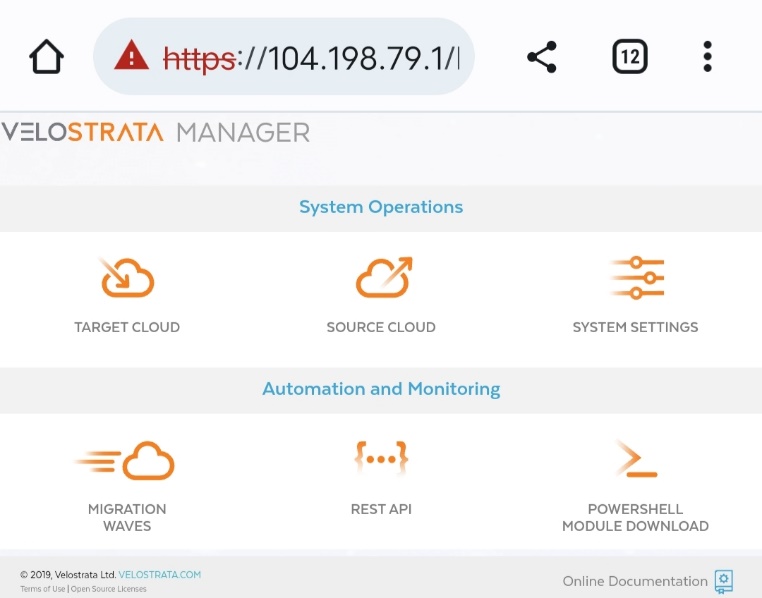
* The Velostrata Manager can take up to 2 minutes to launch before you can connect to it.
* In **Sign in**, specify the following information

Username - apiuser

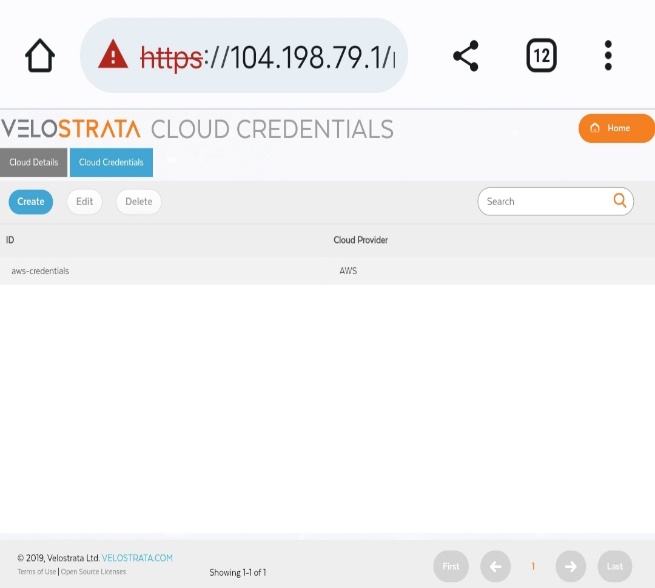
Password - **velo1234**

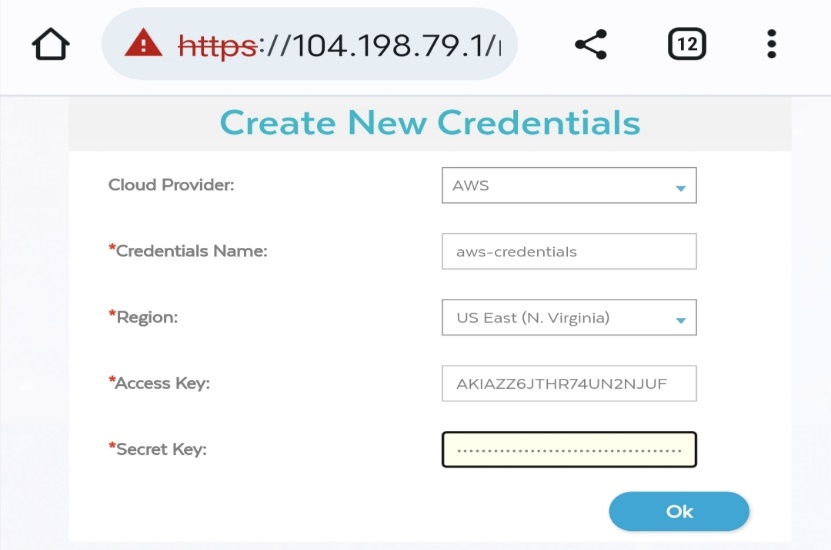
* On the initial setup screen, click **Yes** for both Stackdriver Logging and Stackdriver Metrics.

Graphical user interface, text, application, email

Description automatically generated

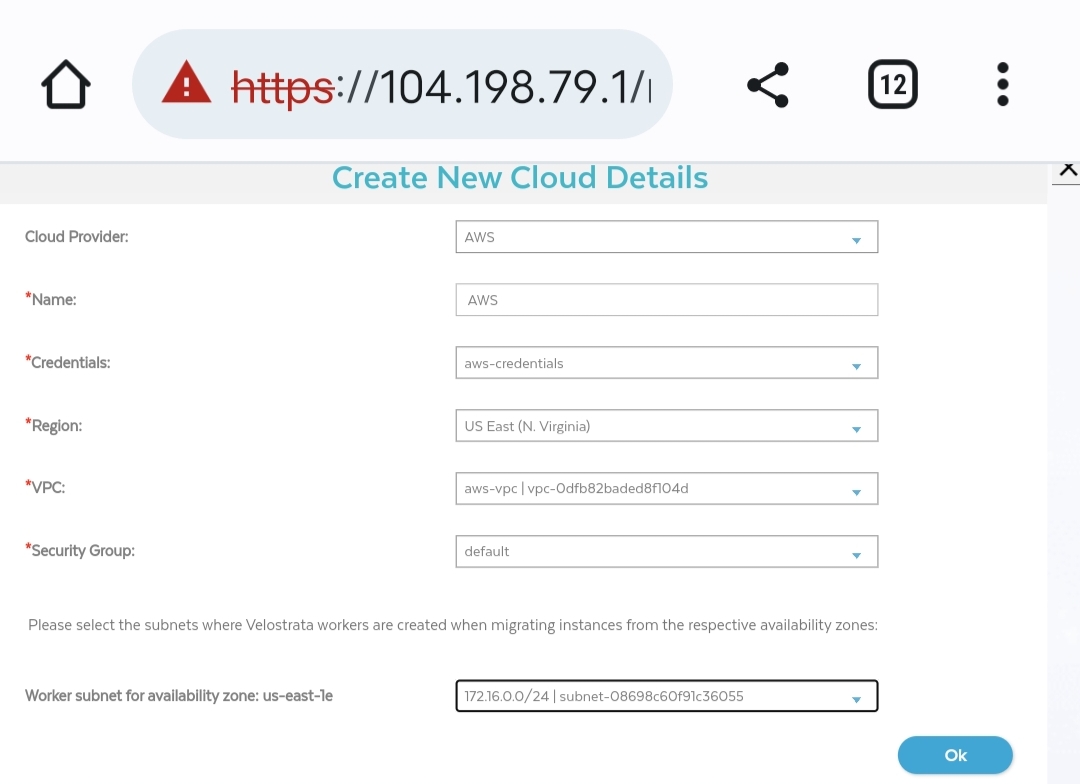
Now you provide source information about the AWS VM to Migrate for Compute Engine.

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



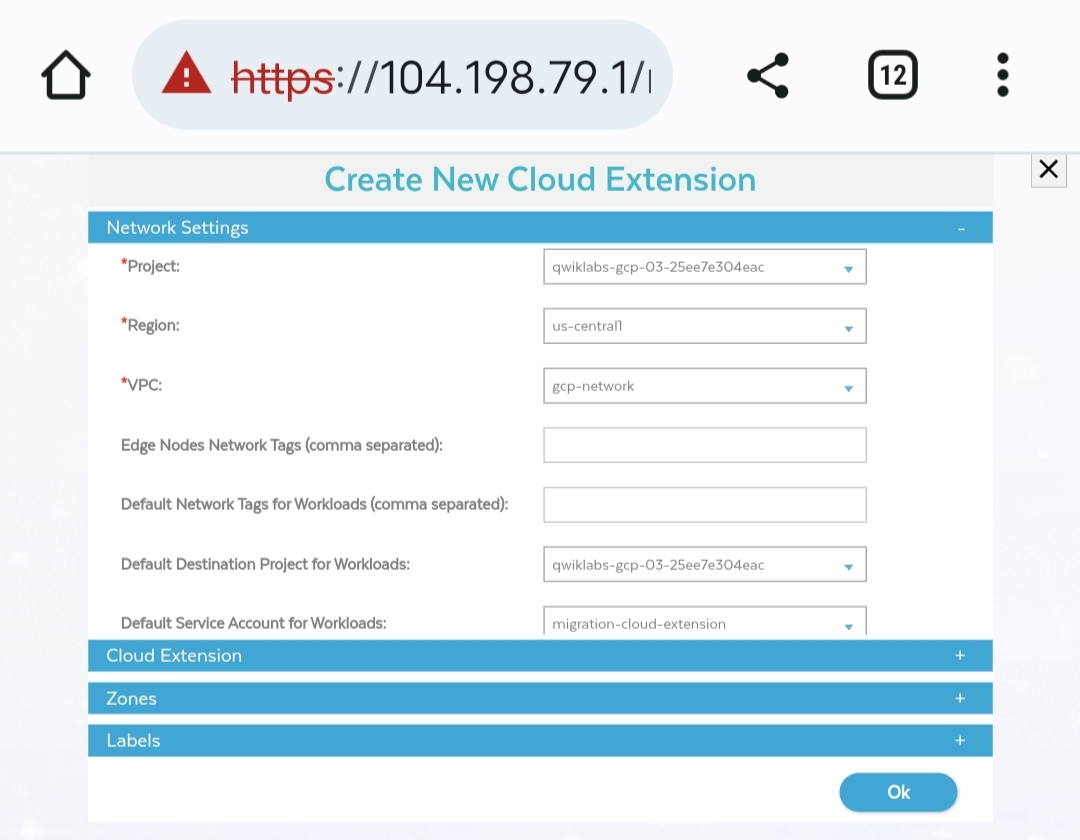
* In **Source Cloud** , Click the **Cloud Details** and **Create New Details.**

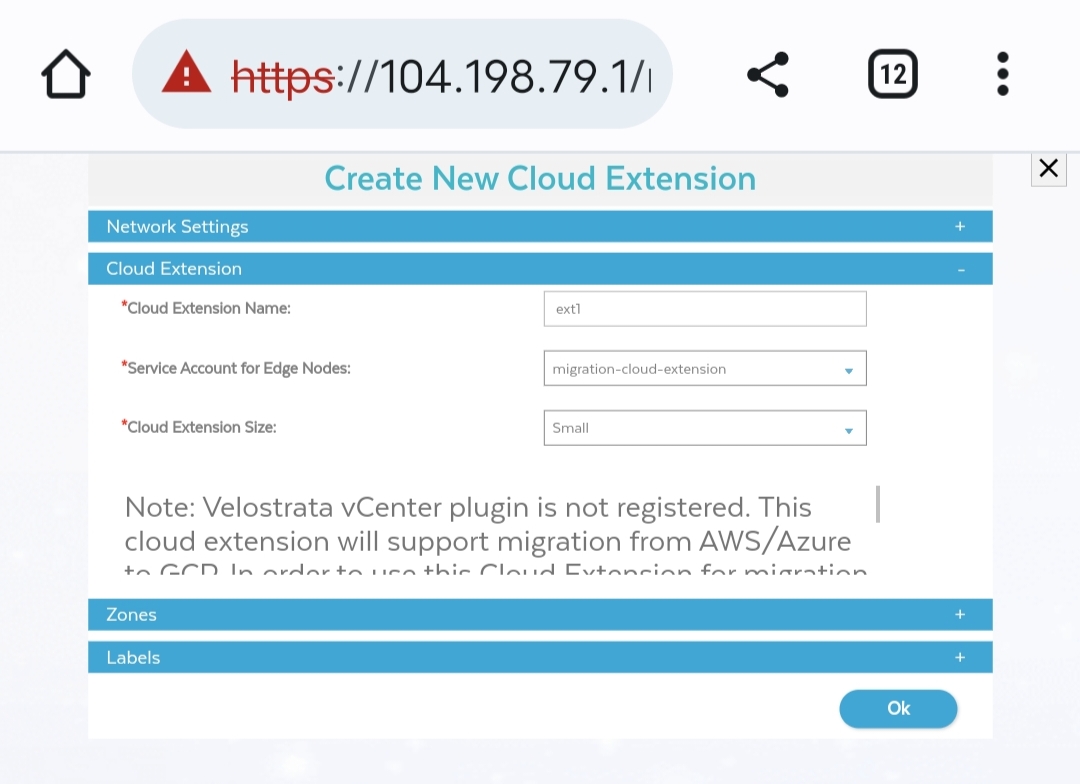


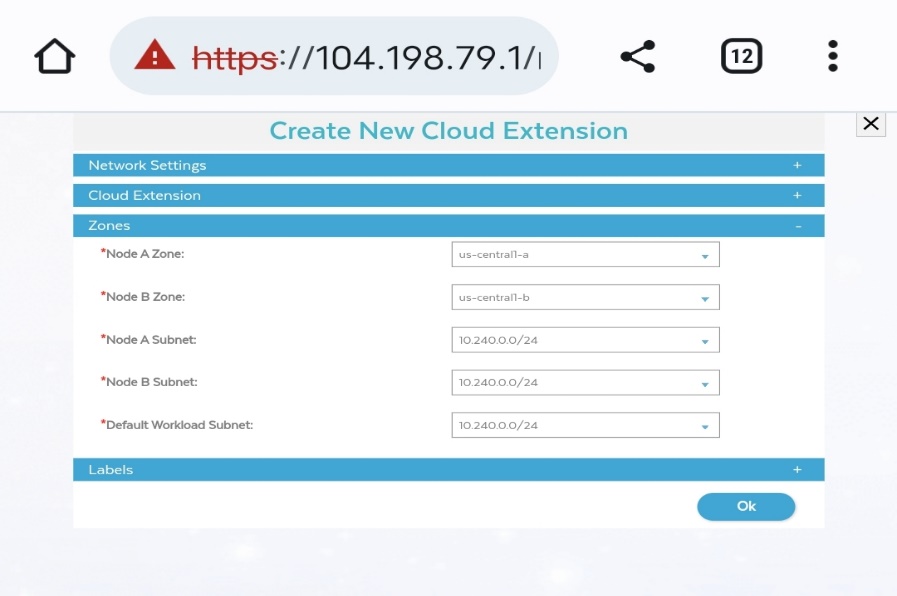


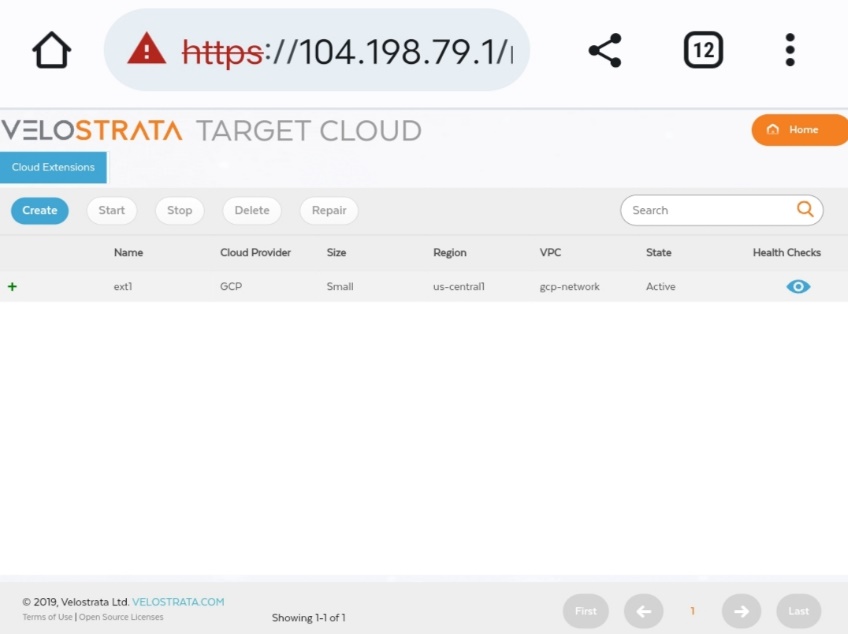
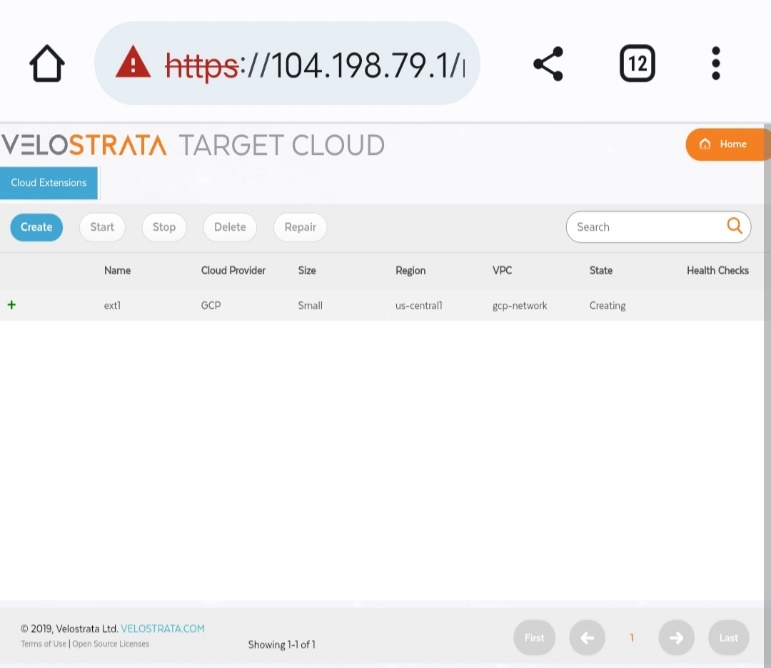
**Set of Cloud Extensions**

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

**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.

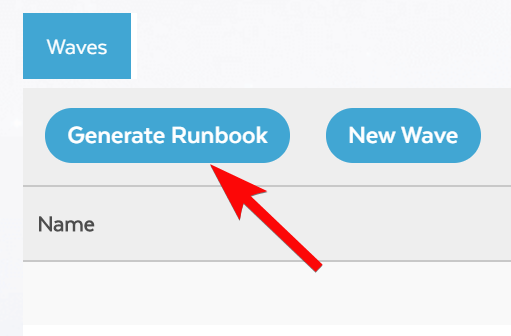
**Generating Migration Waves**

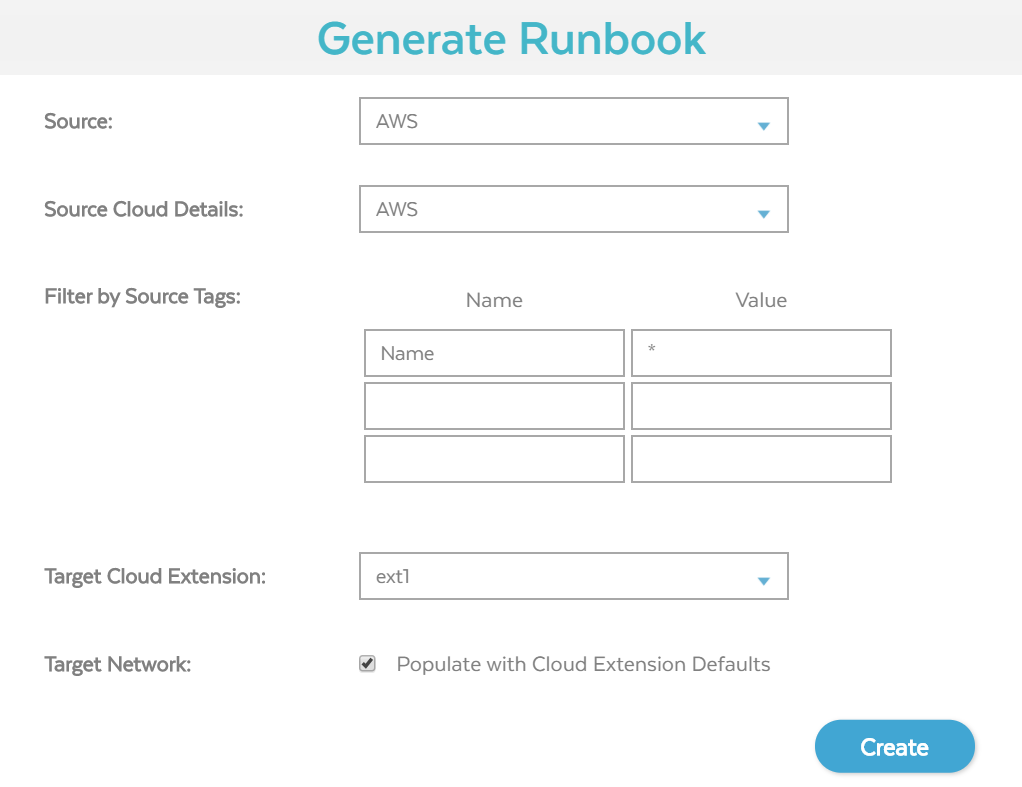
In this task you will use a Migrate for Compute Engine feature called a wave that batches VMs for migration. Waves are made up of runbooks and jobs, which makes migration more manageable.

* Click **Migration Waves from Home**.



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





* 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.

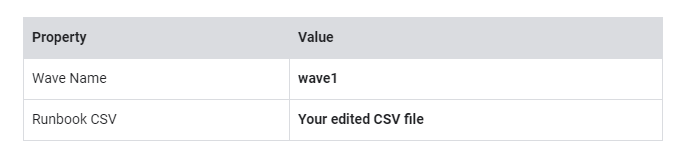
* In the **Import file** box, select **Comma** as the Separator type and leave all other settings as their defaults, Click **Import data**.

### 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:



**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**.

Graphical user interface, text, application

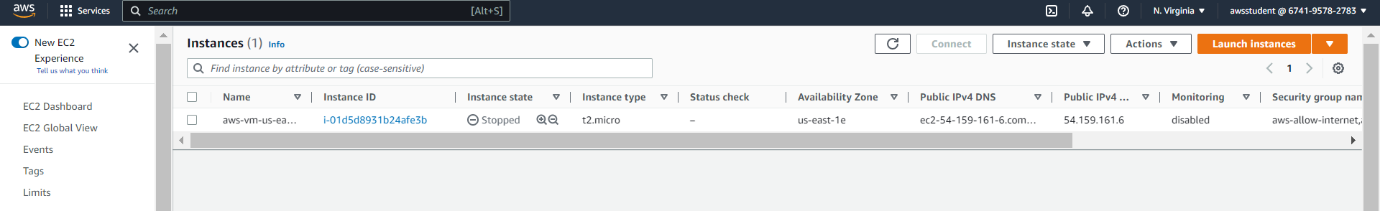
Description automatically generatedThe status will now change to Full Migration (Running).

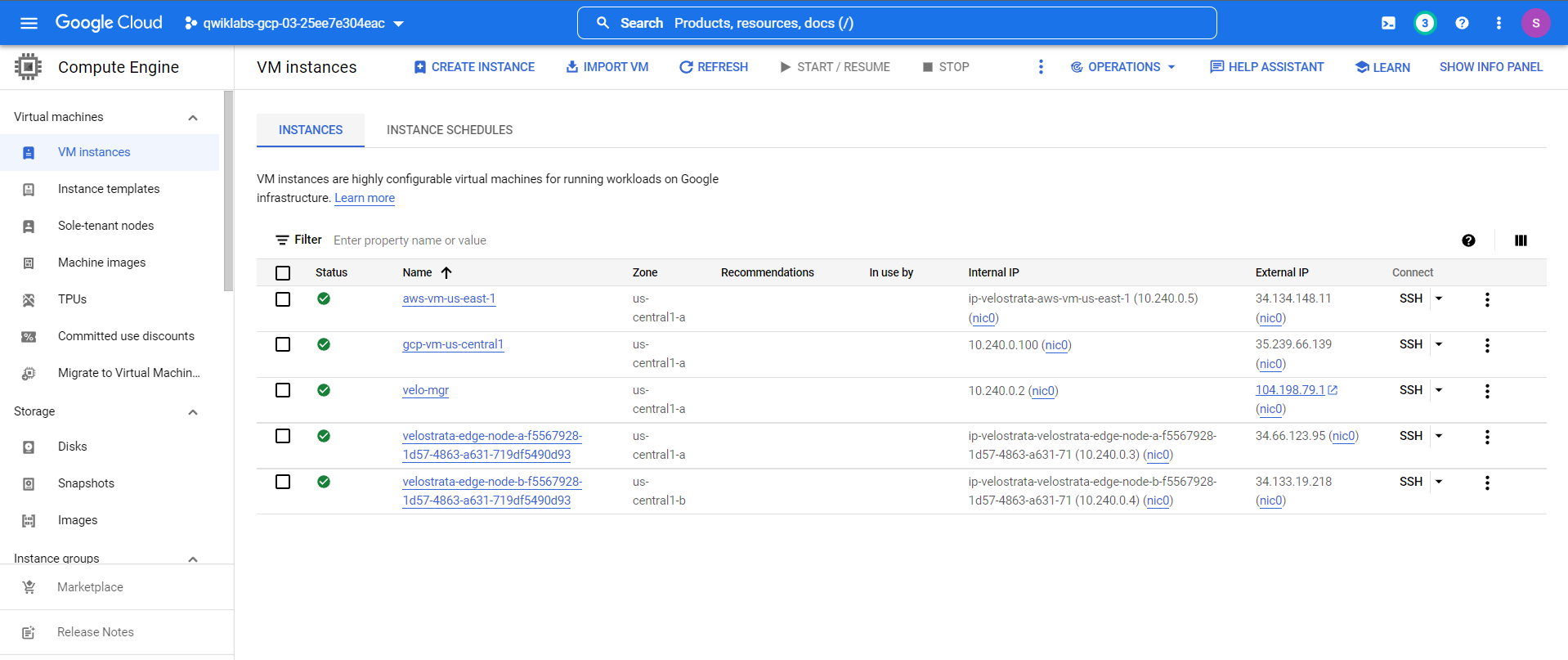
The migration is now progressing. Wait for the corresponding Compute Engine to become available.

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.

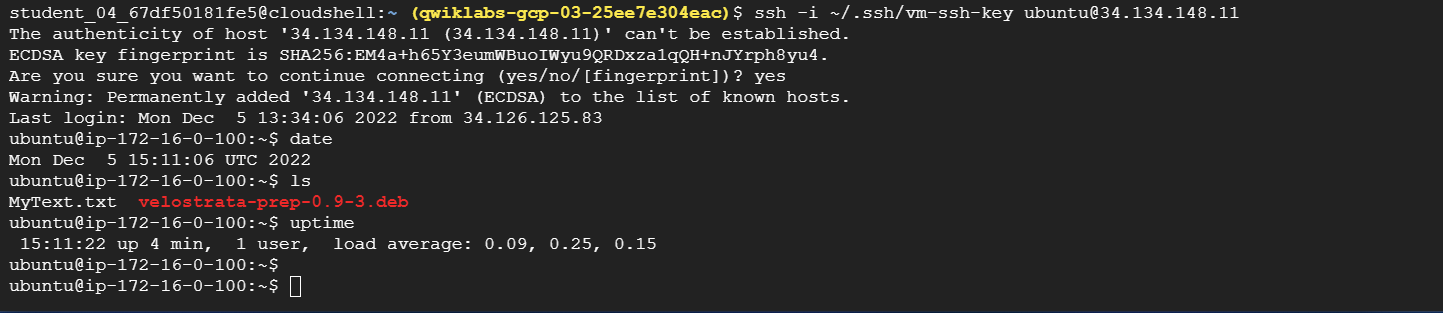
A record line for the VM being migrated should be displayed. This should take up to 10 minutes.

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



1. 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*.



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

# Scenarios

# Technology Road Map / Blueprints

# GCP Service and Products

**GCP**

* Google Compute Engine
* IAM Servcie Accounts.
* API’s Services
* Migration Manger -Velostrata
* Hybrid Connectivity - VPN

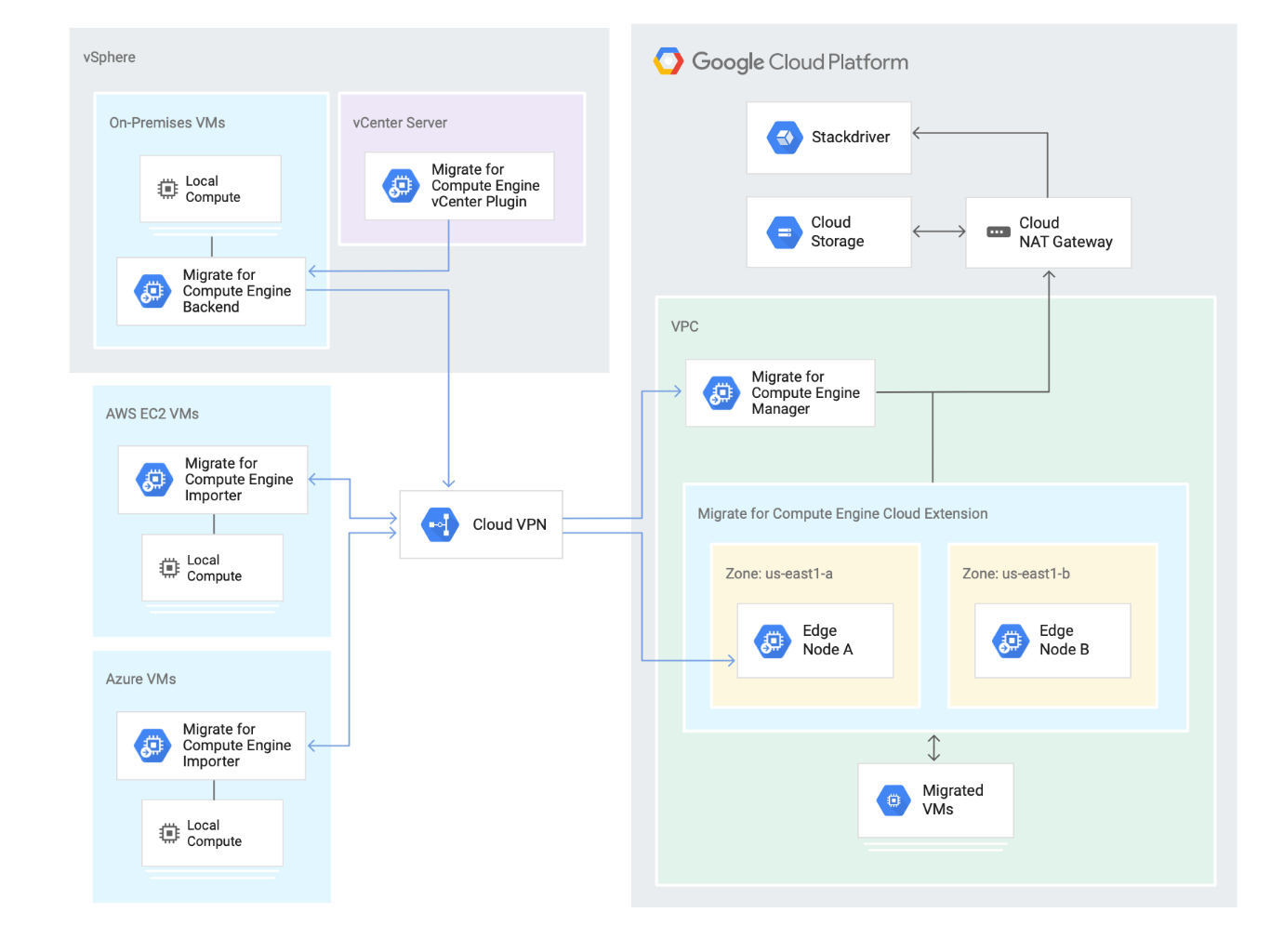
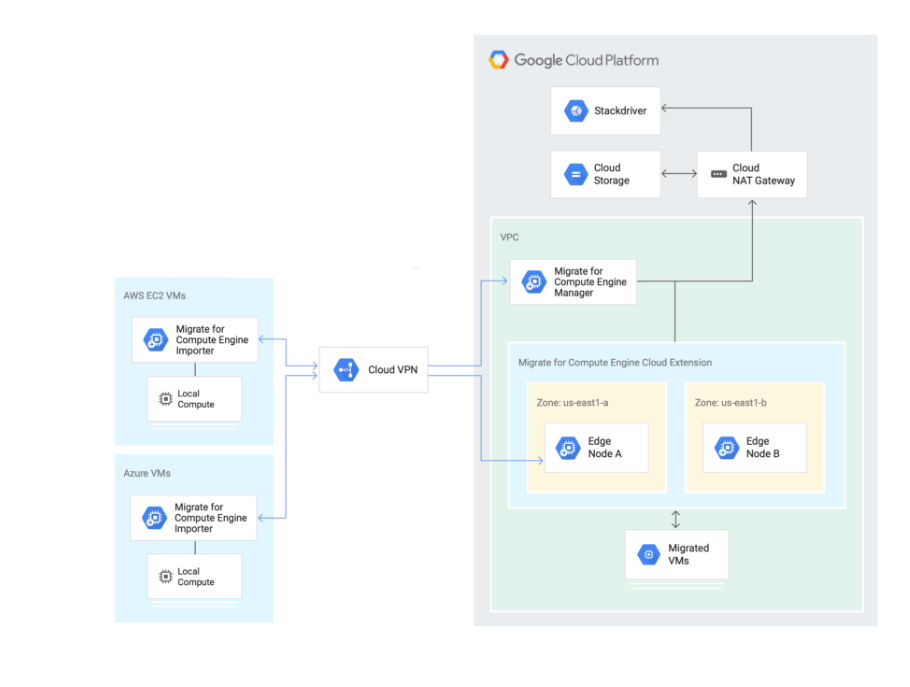
.

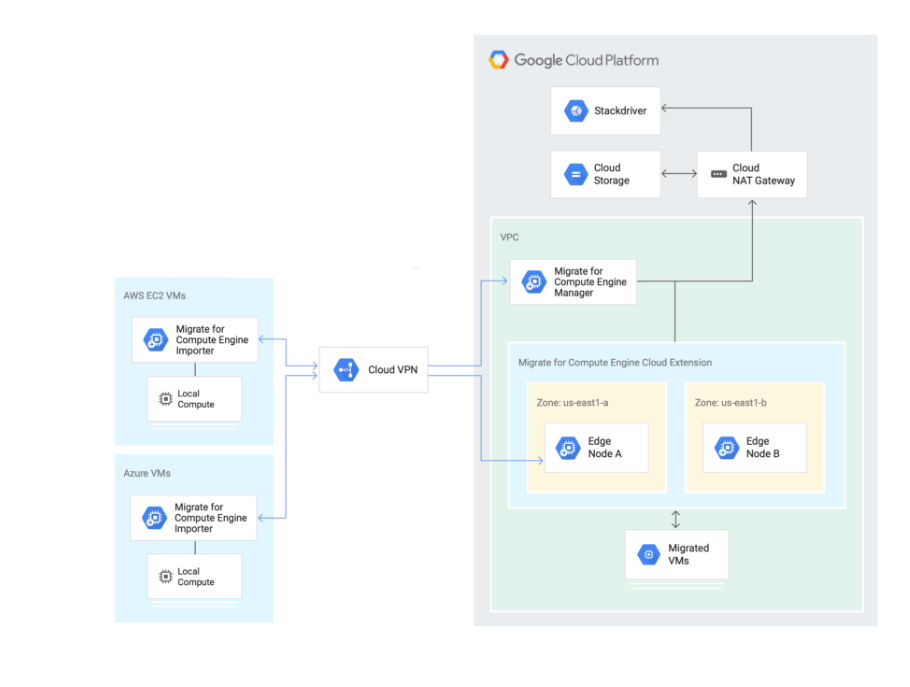
**Other Tools**

* Windows PowerShell
* PUTTY
* RDP
* Docker
* WebLogic Container

# Implementation Diagram

***Target Architecture Diagram***



******

# 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 dur 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 .