Sandbox Deployment

Table of Contents

Overview

Introduction

This document will guide you through an end-to-end deployment example, which you may use for product evaluation or as a reference for future deployment.

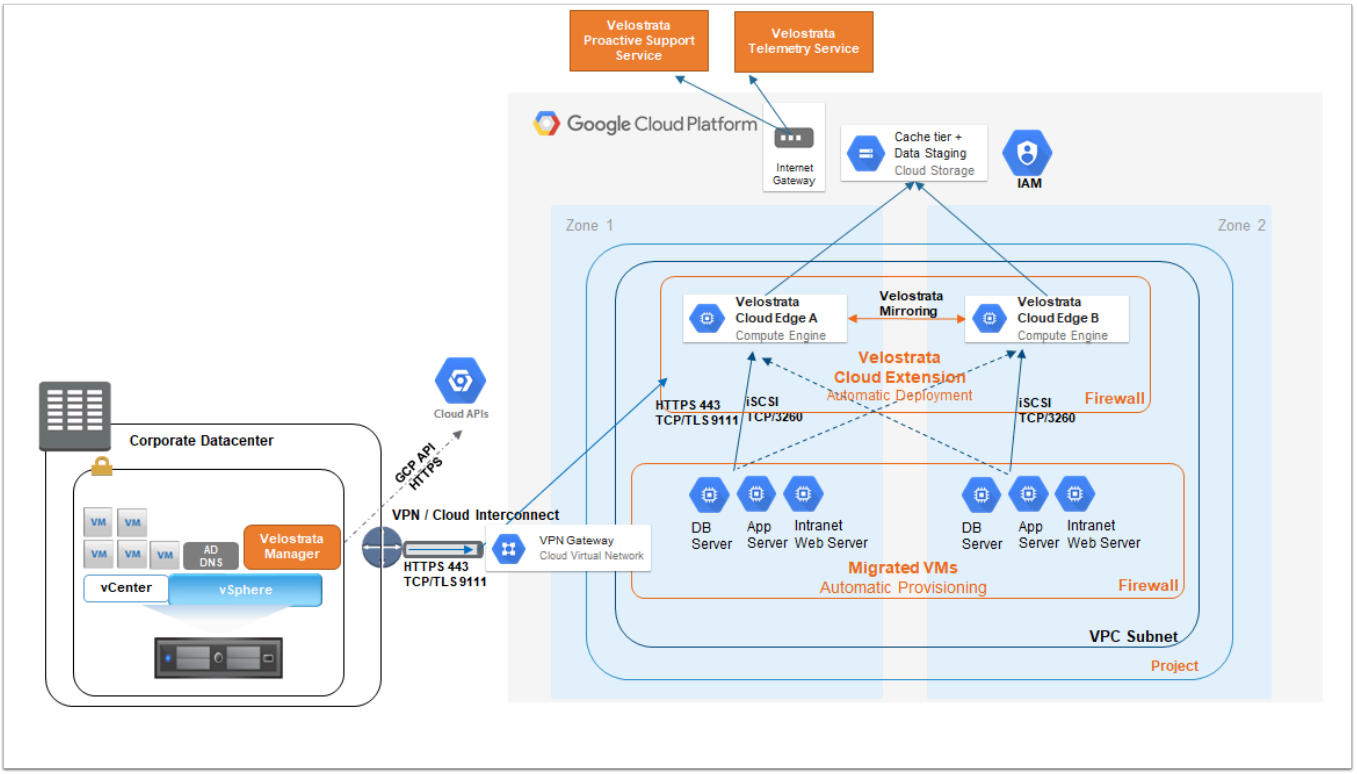
In this document, you will set up a GCP VPC environment in the cloud, to which you will connect your datacenter or lab environment using an open source software VPN and virtual router, velostrata will be deployed across on-premises and GCP to enable migration of workloads.

If you already have a VPC/VPN setup between on-prem/aws and GCP, and are looking for a production-caliber deployment, please see these guides here.

The following tasks will be completed using this guide:

1. **Setup Cloud Networking:**
    1. Setup the GCP environment in preparation for the Velostrata deployment.
    2. Set up a GCP Project including GCP VPC, subnets, firewall rules, network tags, VYOS VPN Gateway instance and Service Accounts with the permissions required for Velostrata operations. This task is automated based on interactive user input, using a PowerShell script, provided by Velostrata.
    3. Deploy and configure an on-prem VPN virtual appliance.
    4. Configure the cloud-side VPN virtual appliance and establish a VPN link.
    5. Configure a machine on-prem with a static route to reach virtual machines in the GCP network, over VPN.
2. **Setup Velostrata:**
    1. Deploy the Velostrata Manager virtual appliance on vSphere.
    2. Register the Velostrata vCenter Web Client Plug-in.
    3. Select a Datacenter in vCenter and create a Velostrata Cloud Extension.
3. **Run your first VM in Cloud with Velostrata:**
    1. Select a virtual machine on-prem and use the run-in-cloud wizard to execute it in the cloud.
    2. Connect to the virtual machine when running in the cloud.
    3. Monitor virtual machine performance when running in the cloud.
    4. Run the virtual machine back on-premises.

Reference Architecture



Important Note:

Velostrata recommends an **Internet connection** of no less than 20Mbit/sec symmetric or equivalent bandwidth and preferably 50-100Mbit/sec for production use. From an on-prem datacenter perspective, the connection uplink is used for virtual machine disk read activity (on cache miss) and the connection downlink is used for write-back activity.

For the full set of deployment prerequisites and architecture guidelines, see the **Velostrata User Guide**.

We also recommend that you familiarize yourself with the latest Velostrata release notes, see the **Velostrata Release Notes**.

Cloud Networking Setup

Preparing for the Setup of the Environment

In order for Velostrata to work with GCP, you'll need to make sure all of the following are configured before proceeding onto the next section:

1. Login as a user with administrative privileges in GCP.
    1. **NOTE**: This is required in order to create the deployment using Deployment Manager.
    2. **NOTE:** The user only requires Administrative privileges during the setup.
2. Create a project in GCP.
    1. Alternatively, if you just created your GCP account, you can also just rename "My First Project" to something else.
3. From the menu (top left) select "APIs & Services" > "Library"
    1. Search for all of the following and either verify they are enabled, or if not, enable them one by one:
        1. Identity and Access Management (IAM) API
        2. Cloud Resource Manager API
        3. Google Cloud Deployment Manager V2 API
        4. Compute Engine API
4. From the menu (top left) select "IAM & admin" > "IAM
    1. Find the member that contains "@cloudservices.gserviceaccount.com" and click the edit icon to the right of the name
    2. Click the Add Another Role link
    3. Select "Roles" from the left column and "Role Administrator" from the right column.
    4. Click save.
5. Download the GCP SDK [+Python if not already installed, too] from https://cloud.google.com/sdk/downloads and install it on your workstation (where you access VMware vCenter from).
    1. **NOTE:** that you may need to reboot your workstation after installing the SDK.
    2. **NOTE:** Powershell version 3 or higher is required on your workstation
6. After installing the GCP SDK, perform an update from the same command prompt by running the command: *gcloud components update*

Once you have completed these steps, you are ready to proceed to the next section.

Creating the Velostrata Reference Cloud Deployment Stack on GCP

The Velostrata Cloud Deployment template provided creates the full stack of resources required to deploy and use the Velostrata solution, including a GCP VPC, subnets, firewall rules, network tags, Velostrata VPN Gateway instance, and Service Accounts with the permissions required for Velostrata operations. Google Cloud Platform Deployment Manager allows you to manage these resources.

To create the Velostrata Reference Cloud Deployment Stack on GCP:

1. Download and extract the GCP deployment script: http://tiny.cc/velos-poc-gcp-v3-ps

   Open a PowerShell session and change directory to the extracted script folder(cd ...\fullDeployemnt):

   run **.\ gcp_poc_setup_v1.0.ps**

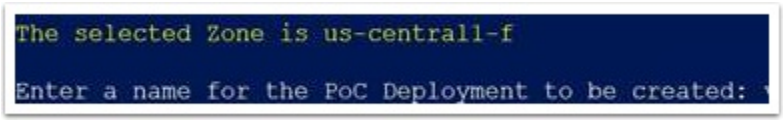
2. Enter the Administrative GCP Account.

3. Enter an existing Google Cloud Project Id for the Velostrata PoC setup.

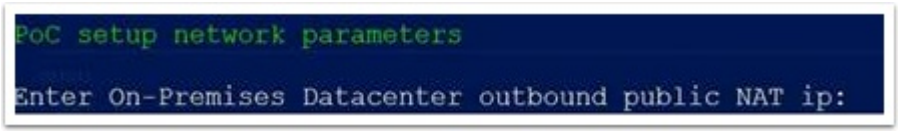
4. Enter the Google Cloud Platform Region.

5. Enter the Google Cloud Platform Region Zone.

6. Enter the name for the PoC Deployment, Lowercase, Minimum Length – 8 characters.



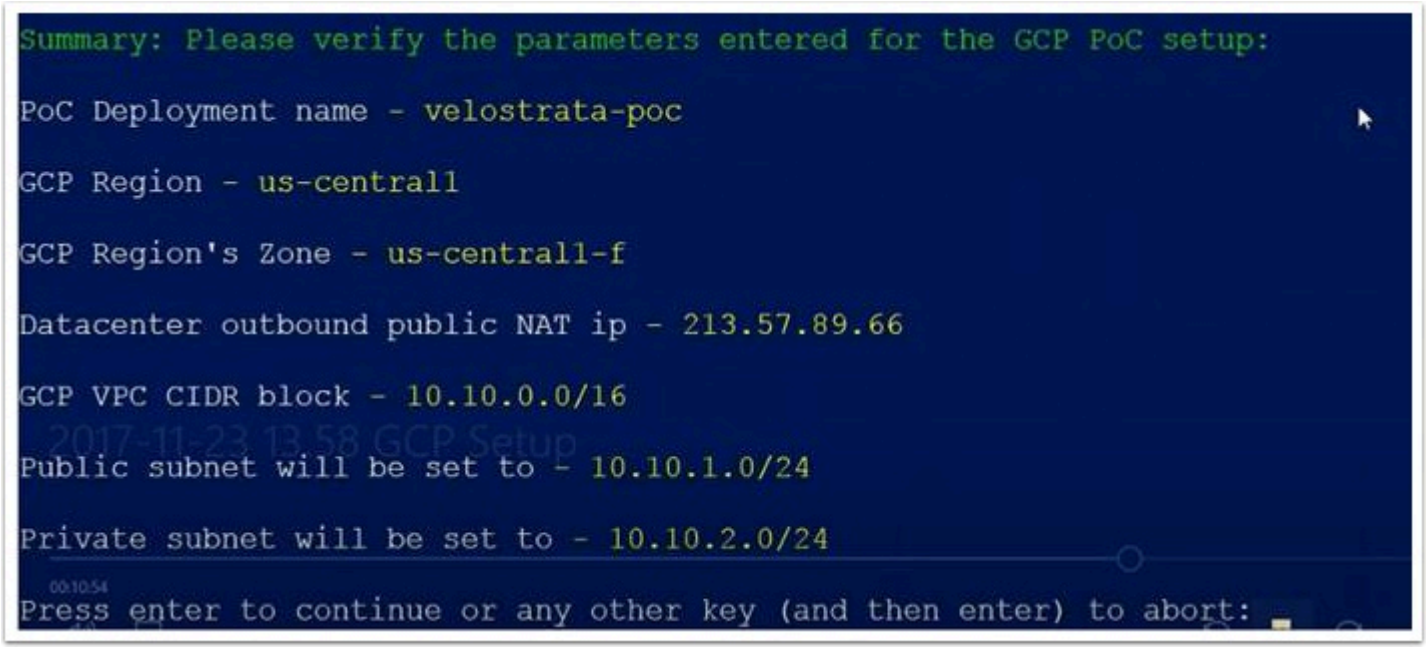
7. Enter the On-Premises Datacenter outbound **public** NAT IP, that is, the customer's public NAT IP.



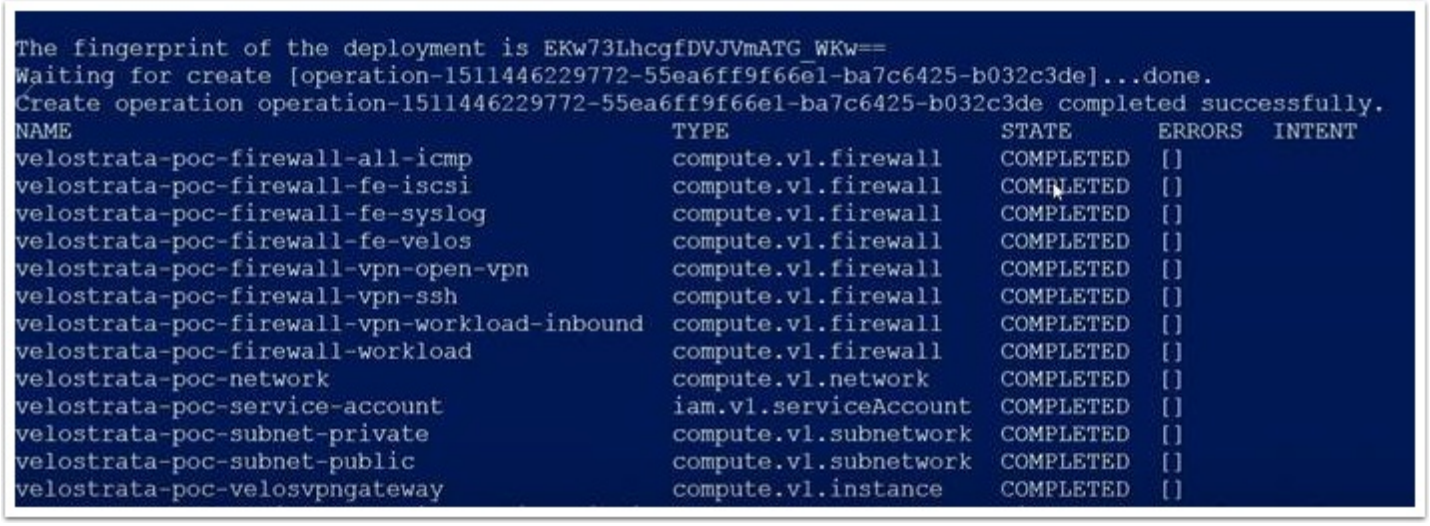
Enter the On-Premises Subnet CIDR (for example 192.168.0.0/16).

8. Enter the desired GCP VPC /16 CIDR block (for example **10.10.0.0/16)**.

9. A summary appears of the various parameters you entered.

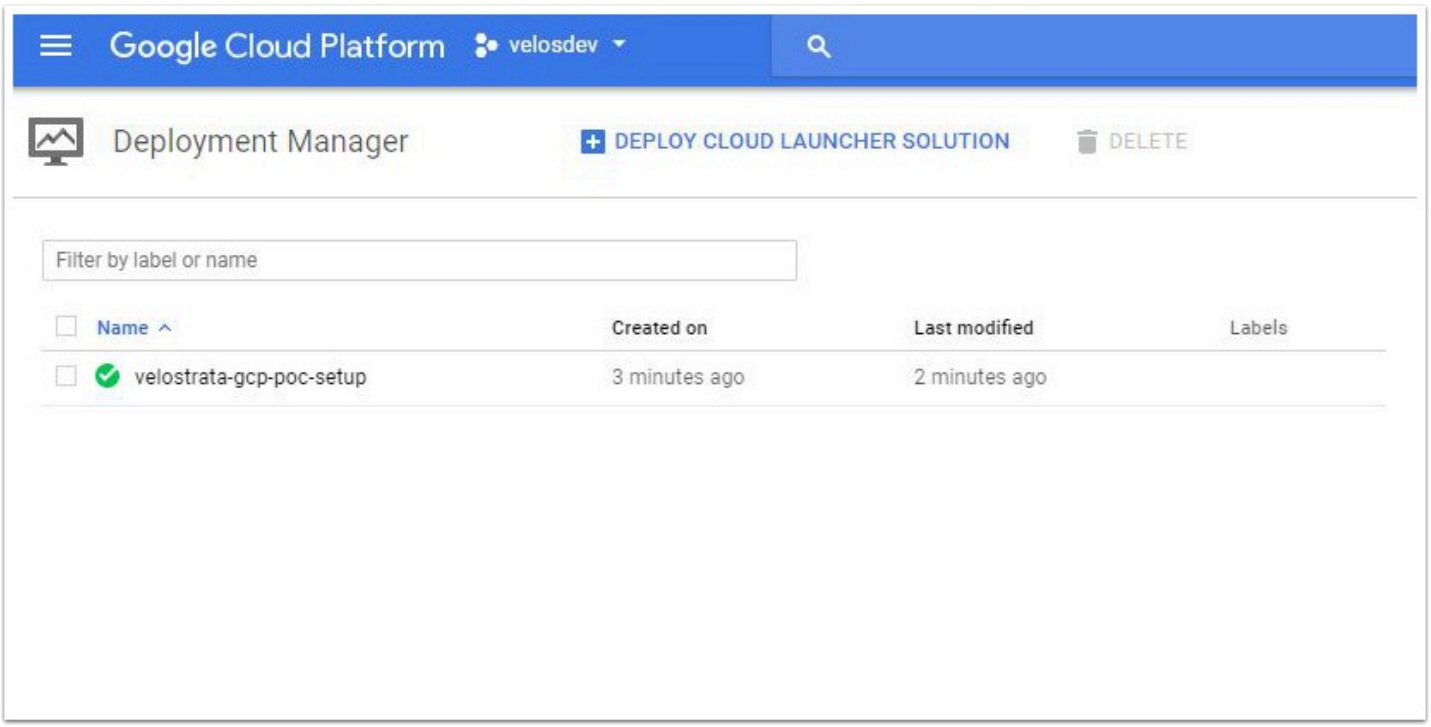


10. Review the parameters, and then press enter to continue. The public and private subnets are carved out of the CIDR block. The script takes approximately 5 minutes to complete and creates the following:
    - Deployment
    - VPC
    - Public and private subnets
    - Firewall tags configured with Firewall rules
    - Velostrata VPN Gateway instance
    - Service Accounts and assigns the required permissions for Velostrata operations

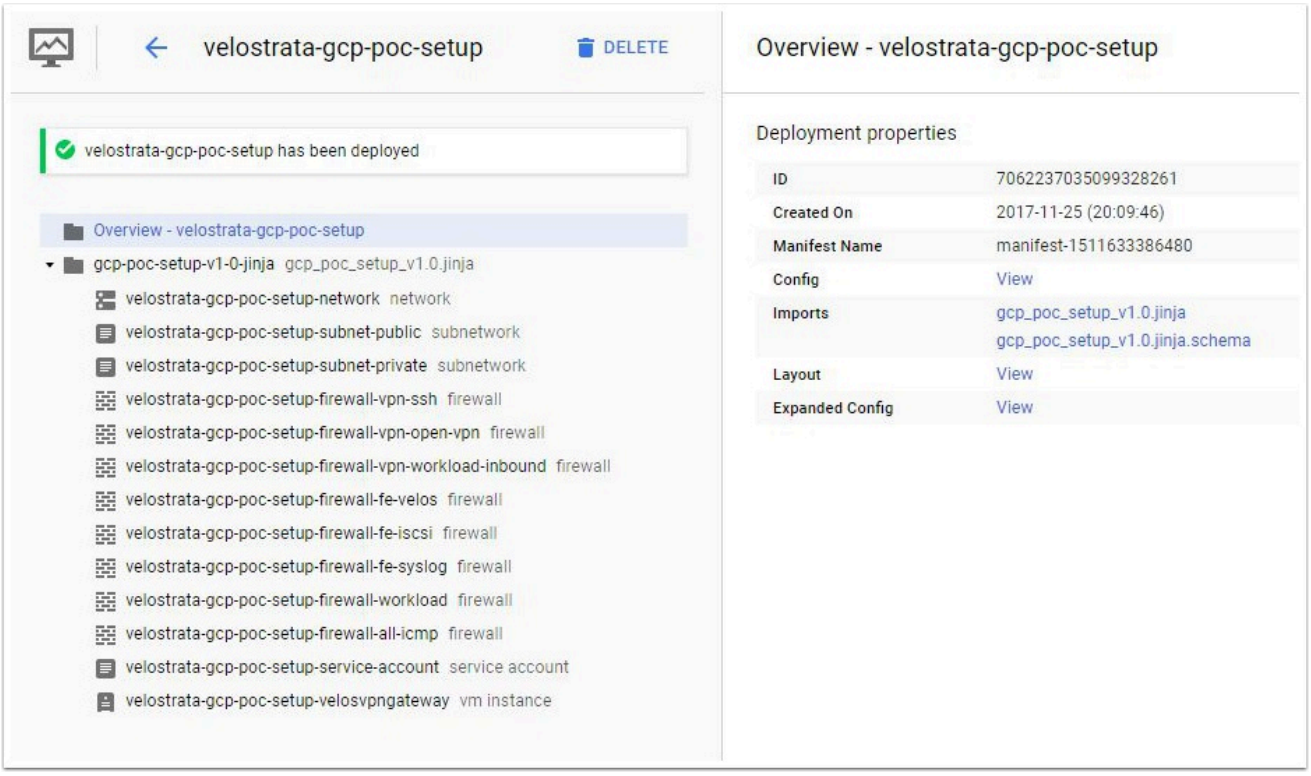


Monitor the script execution for any errors. up on successful execution a .json file will be created in the script directory, this file will be used later for the creation of the cloud extension

Optional : To view the created deployment, open the Deployment Manager in the GCP console.



12. Drill-down to review the deployment components and properties.

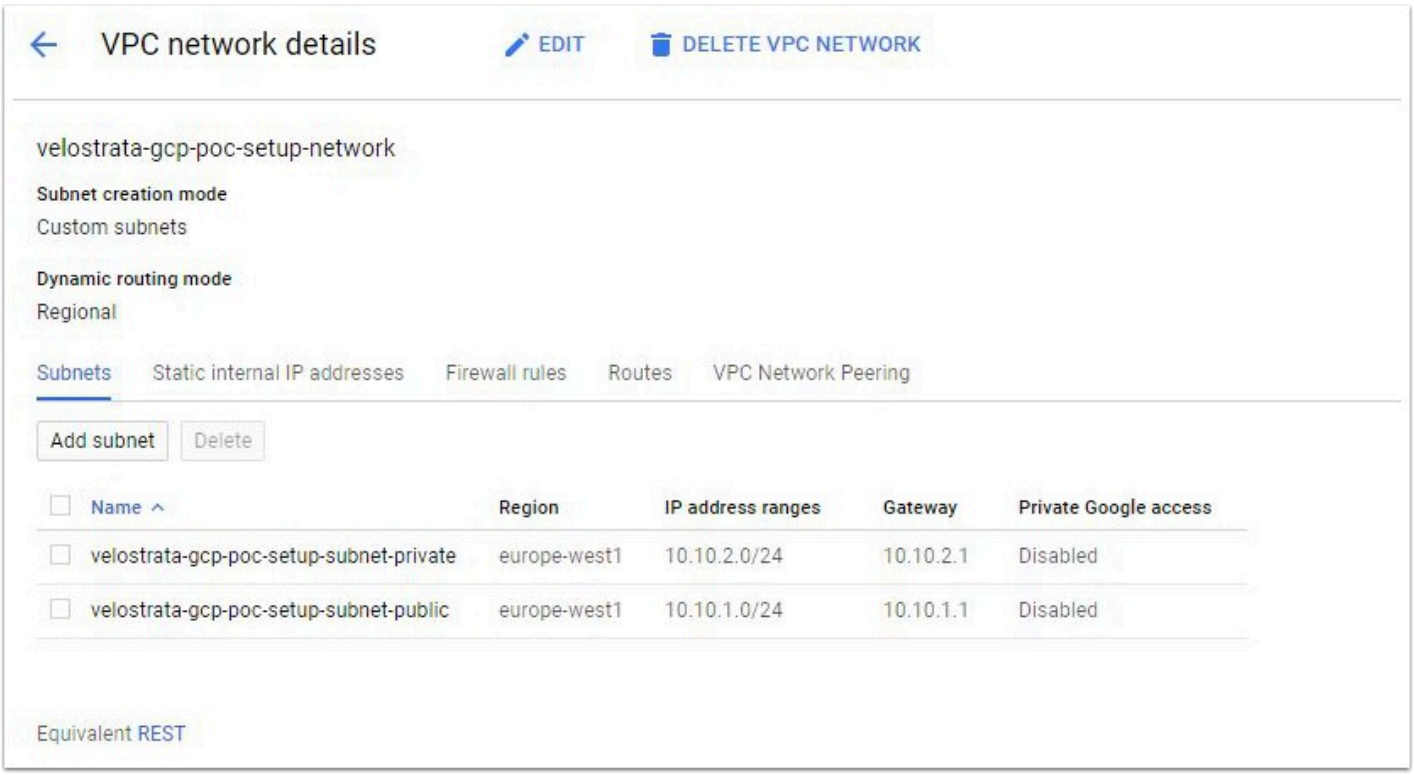


This includes the following:

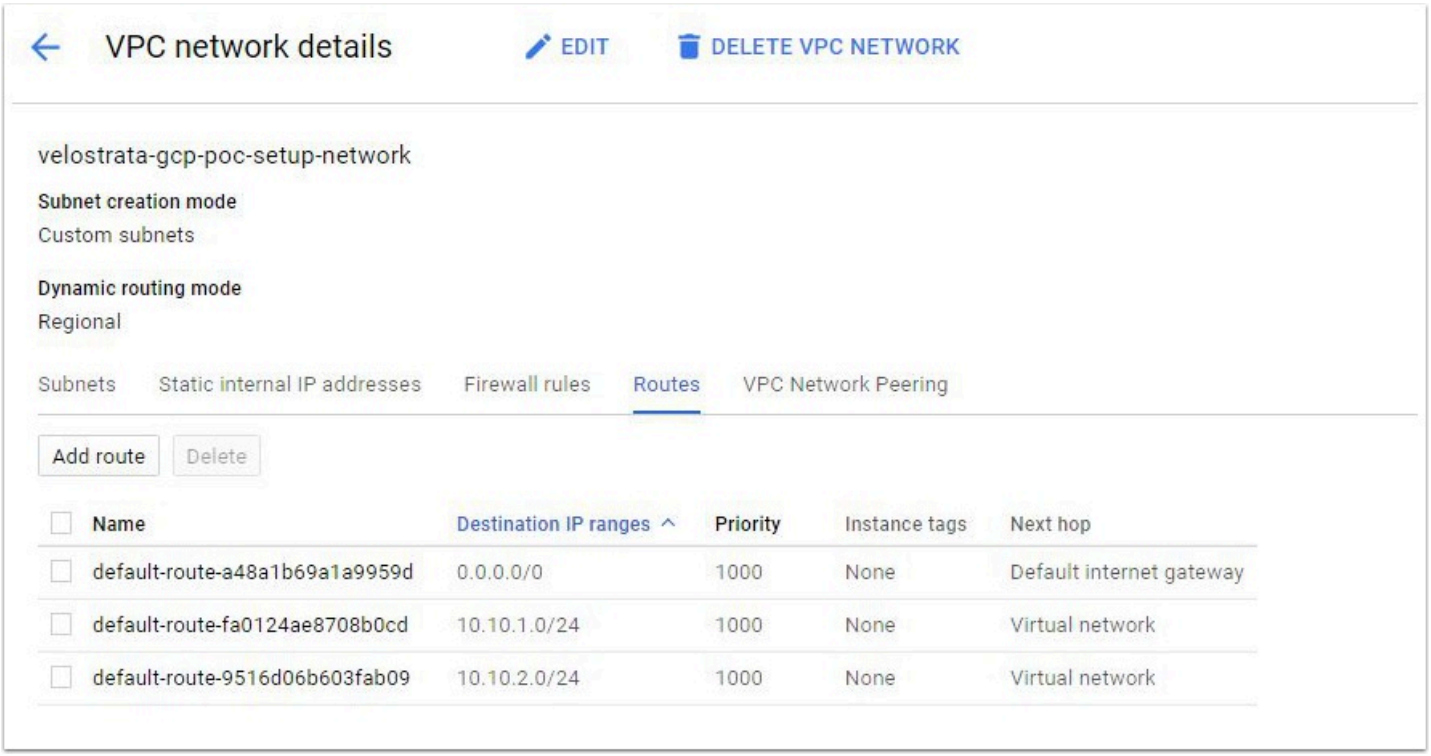
- VPC network
- VPC subnets (public and private)
- Firewall rules

- Service account
- VM instance for the Velostrata VPN Gateway

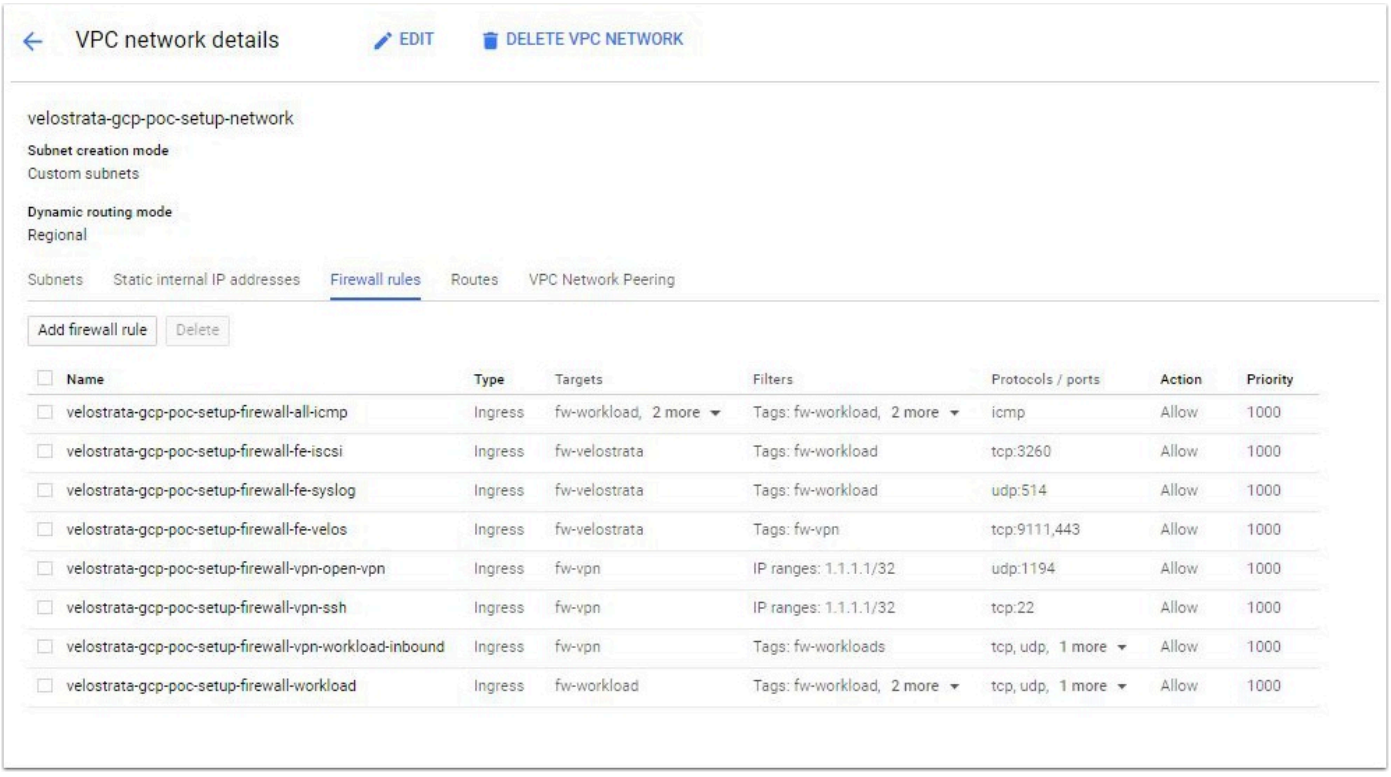
13. To view the VPC, select **velostrata-gcp-poc-network**.
14. To manage the resource, click **Manage Resource**. This shows us the VPN networks.



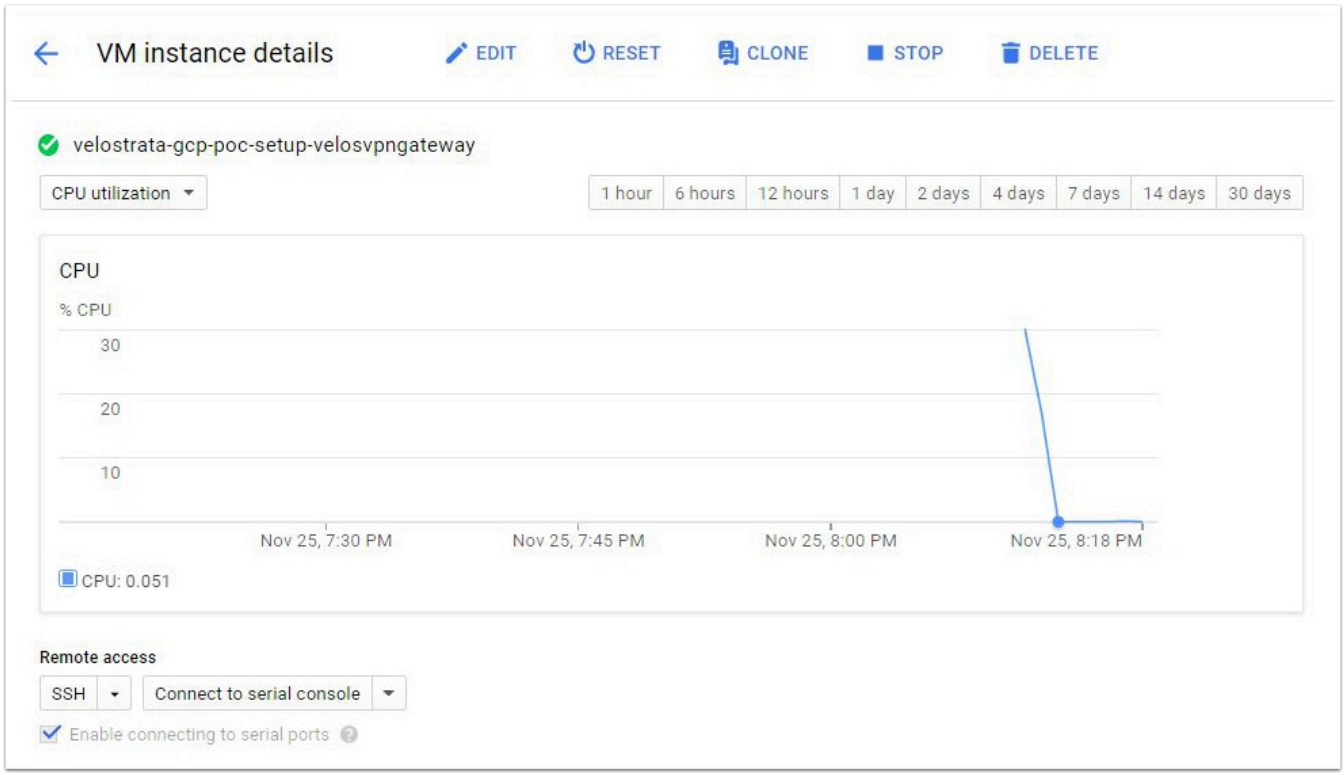
15. The name of the deployment is used as a prefix for the resource name, in this example, **velostrata-gcp-poc-setup**, and so on.
16. Select the **Routes** tab.



17. Select **Firewall rules**.



18. View the **velostrata-gcp-poc-setup-velosvpngateway** instance.


VM instance details
EDIT
RESET
CLONE
STOP
DELETE
velostrata-gcp-poc-setup-velosvpngateway
CPU utilization
1 hour
6 hours
12 hours
1 day
2 days
4 days
7 days
14 days
30 days
CPU
% CPU
30
20
10
Nov 25, 7:30 PM
Nov 25, 7:45 PM
Nov 25, 8:00 PM
Nov 25, 8:18 PM
CPU: 0.051
Remote access
SSH
Connect to serial console
Enable connecting to serial ports

Configuring the VyOS VPN Virtual Appliance On-Premises

To configure the VyOS VPN Virtual Appliance on-premises:

1. Download the latest VyOS-VMW OVA from the region nearest to you:

- US: http://tiny.cc/velos-us-vyos-ova
- EU: http://tiny.cc/velos-eu-vyos-ova

2. Login to the GCP portal and navigate to **Compute Engine -> VM Instances**. Find the **XXX.velosvpngateway** VM created by the script and record its **Public IP address**.
3. On the vCenter Server, click **File > Deploy OVF Templates**. Follow the instructions to deploy the VyOS virtual appliance and click **OK**.
4. Power on the VM.
5. Login to the on premise VYOS VPN Appliance (for example, using vCenter VM console) with user **vyos** and password **vyos**.
6. Set up a static IP for the appliance by entering the below commands. Replace the example 192.168.10.10/24 with your on your subnet setup on-prem, and netmask bits (CIDR notation). Replace 192.168.10.1 with the IP address of your local subnet default gateway. Use a gateway that can route to the internet, in order to allow the VPN tunnel to be created later on.

```
config
set interfaces ethernet eth0 address 192.168.10.10/24
set system gateway-address 192.168.10.1
```

note: if you had a DHCP address configured on the interface you will need to delete it by using the following command:

```
del interfaces ethernet eth0 address dhcp
```

- Change MTU value for eth0

```
set interfaces ethernet eth0 mtu 1460
```

8. Configure source-based NAT for the local network. This will allow virtual machines that run in the GCP VPC to communicate with corporate-side servers and services as needed without having to set up routing on these corporate servers.

```
set nat source rule 100 outbound-interface 'eth0'
set nat source rule 100 translation address masquerade
```

7. Enable SSH for remote management.

```
set service ssh port '22'
```

9. Commit, save the changes and exit the configuration mode.

```
commit
save
exit
```

Note: you can now use an ssh client like Putty to connect to the Vyos appliance, this will streamline copy-paste of the below commands.

10. Configure a VPN tunnel using OpenVPN. Generate a shared secret key file.

```
generate openvpn key /config/auth/secret
sudo chmod 640 /config/auth/secret
```

11. Configure the VPN tunnel interface.

Note: Replace 1.1.1.1 with the xxx.velosvpngateway instance public IP from GCP.

```
config
set interfaces openvpn vtun0 local-address 172.16.100.2
set interfaces openvpn vtun0 mode site-to-site
set interfaces openvpn vtun0 protocol udp
set interfaces openvpn vtun0 openvpn-option --comp-lzo
set interfaces openvpn vtun0 keep-alive interval 10
set interfaces openvpn vtun0 keep-alive failure-count 5
set interfaces openvpn vtun0 remote-address 172.16.100.1
set interfaces openvpn vtun0 remote-host 1.1.1.1
set interfaces openvpn vtun0 shared-secret-key-file /config/auth/secret
```

Note:

UDP is a preferred protocol, performance wise for VPNs, as the overhead on traffic sent across the vpn tunnel is lower and there is no nesting of congestion handling. For deployments where UDP is not an option due to firewall constraints, you may use a TCP based tunnel. If there is a requirement to use an HTTP proxy for outbound internet access, you may also configure that along with a TCP based tunnel. Do note that a significant performance impact is to be expected, especially when dealing with low bandwidth links. For TCP setup example, see **Configuring VPN using TCP**.

12. Set a static route to direct the local subnet traffic to the GCP network CIDR.

Note: Replace 10.10.0.0/16 with the VPC CIDR value you have entered in the GCP PoC script

```
set protocols static route 10.10.0.0/16 next-hop 172.16.100.1
```

13. Commit and save the configuration.

```
commit
save
exit
```

14. Copy the shared secret file from the on premise VyOS appliance (using PSCP for example) replace the IP example below 192.168.10.10 with the on premise VyOS address

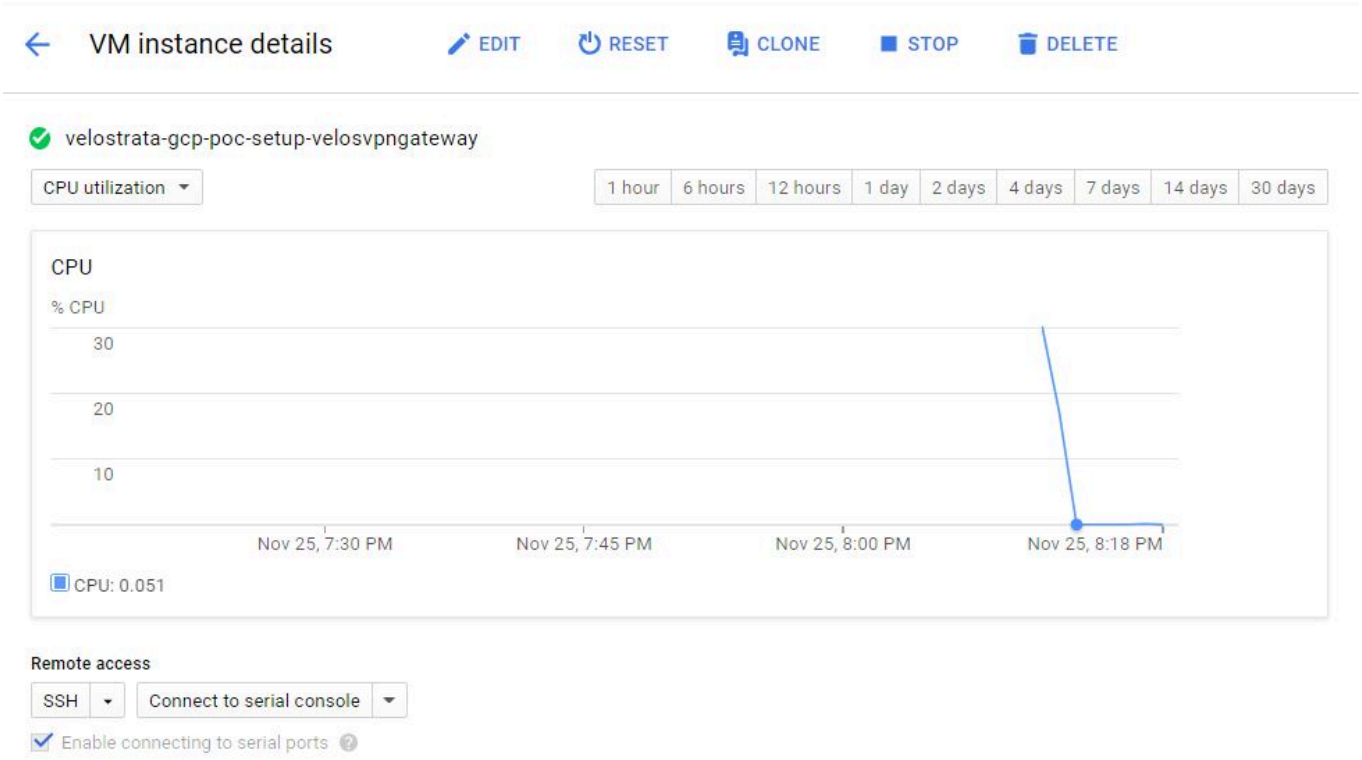
Pscp.exe -pw vyos vyos@192.168.10.10:/config/auth/secret .

Configuring the VyOS VPN Instance on GCP

To configure the VyOS VPN Instance on GCP:

As part of the PoC script that is run, the VM instance of a Velostrata VYOS VPN Gateway is created with its Serial console option enabled.

Also, The Velostrata VYOS VPN Gateway in GCP is assigned with FW Tag - 'fw-vpn', allowing inbound connectivity for the OpenVPN tunnel using port UDP/1194, as well as to allow SSH to the private IP address from the source on-premises environment (the outbound public IP configured during the deployment script execution).

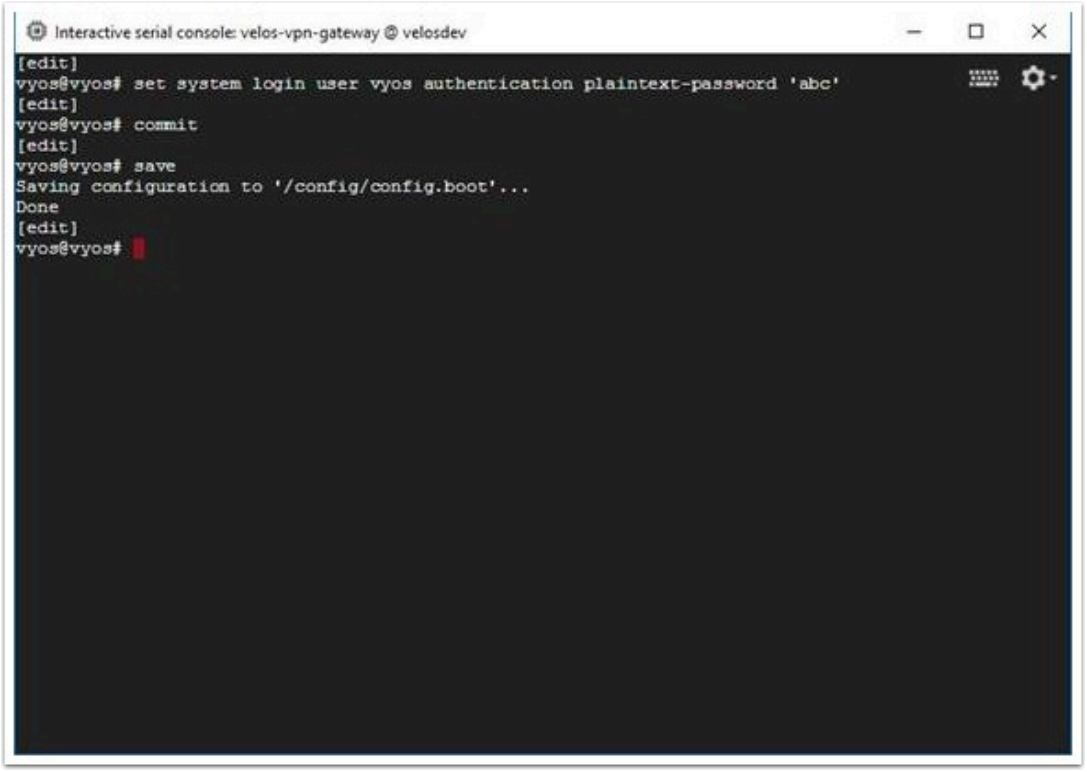


1. Open the google cloud console, and locate the velosvpngateway instance.
2. Connect to the VM serial console - under "remote access" click "connect to serial console". use the following credentials:

   User: vyos, Password: vyos

3. Change the Velos VPN Gateway default password by entering the following command:

```
conf
set system login user vyos authentication plaintext-password ‘password’
commit
save
exit
```



4. Delete all existing route table entries
5. To view current route table entries, enter the following command, and document the VM's default gateway IP:

```
sudo ip route show
```

6. Delete all existing route table entries:

```
sudo ip route delete xxx.xxx.xxx.xxx
sudo ip route delete 127.0.0.0/8
```

Using the **nano editor** and using the **sudo** - edit the config file **'vyatta-postconfig-bootup.script'** placed under **/opt/vyatta/etc/config/scripts/**

for example:

```
sudo nano -w /opt/vyatta/etc/config/scripts/vyatta-postconfig-bootup.script
```

Add the following lines to it (replace defaultGatewayIp with the public subnet gateway IP for the VPC, for example 10.10.1.1):

```
ip route add 'defaultGatewayIp' dev eth0
ip route append default via 'defaultGatewayIp'
```

save by clicking Ctrl+X -> Save -> Exit

7. Reboot.

8. Upgrade the VyOS image using the following command, Accept all defaults and when finished - Reboot.

```
add system image https://downloads.vyos.io/release/1.1.8/vyos-1.1.8-i586.iso
```

9. Change MTU value for eth0.

```
config
set interfaces ethernet eth0 mtu 1460
commit
save
exit
```

10. Next, we will need to copy the **shared secret key file** obtained previously from the VyOS virtual appliance on-prem to the velosvpngw appliance.

replace the VyosNewPass with the new password you have configured for the cloud appliance and the IP of the appliance

```
pscp -pw VyosNewPass secret vyos@xx.xx.xx.xx:/config/auth
```

11. Using PuTTY or the Serial console, connect to the VyOS instance on GCP to proceed with its configuration.

12. Configure source based NAT for the private subnet network. The VyOS instance will function as a NAT gateway, enabling private subnet instances to access the Internet.

```
config
set nat source rule 100 outbound-interface 'eth0'
set nat source rule 100 translation address masquerade
```

13. Setup the VPN tunnel using OpenVPN.

```
set interfaces openvpn vtun0 local-address 172.16.100.1
set interfaces openvpn vtun0 mode site-to-site
set interfaces openvpn vtun0 protocol udp
set interfaces openvpn vtun0 openvpn-option --comp-lzo
set interfaces openvpn vtun0 keep-alive interval 10
set interfaces openvpn vtun0 keep-alive failure-count 5
set interfaces openvpn vtun0 remote-address 172.16.100.2
set interfaces openvpn vtun0 shared-secret-key-file /config/auth/secret
```

14. Set a static route to direct workloads traffic from the private subnet to the corporate network. replace the 192.168.10.0/24 below with the on-premises network subnet

```
set protocols static route 192.168.10.0/24 next-hop 172.16.100.2
```

15. Commit and save the configuration.

```
commit
save
exit
```

16. Verify that the VPN tunnel has been established.

```
show openvpn site-to-site status
ping 172.16.100.2
```

Configure a Static Route to Reach Virtual Machines in the GCP VPC

With the setup example discussed in this document, there is no requirement to update or configure the on-prem subnet default gateway or core switch or router with routing into the VPC. This allows the PoC setup to be deployed in an existing networking environment without requiring advance planning and changes. However, when you run a virtual machine in cloud and wish to access it from on-prem using RDP (for Windows) or SSH (for Linux) you will need a client machine or a jump server to be configured with proper static routing to get to the VPC.

When working in a lab environment, where your own PC may not be on the same subnet as the VyOS VPN virtual appliance on vSphere, you may want to setup another virtual machine to serve as a jump server. You will then remote into that jump server from your PC and from there reach out to the VMs in the lab or in the cloud. In this case the static route to the VPC will be added on the jump server.

If your PC is on the same subnet as the VyOS VPN virtual appliance on vSphere, you may apply the static route on your PC directly to reach the cloud resources.

To configure the static route, you will need the GCP VPC CIDR configured in the Cloud Deployment section (e.g. 10.10.0.0/16), and the static IP of the VyOS virtual appliance on-prem (e.g. 192.168.10.10).

Note:The following example is for a Windows operating system. You may apply a similar approach on a Linux machine.

To configure a static route to reach Virtual Machines in the GCP VPC:

1. On a Windows PC or virtual machine, open a CMD.EXE command line as an administrator and run the following command to create a persistent route. replace the networking parameters with your specifics.

```
route add -p 10.10.0.0 mask 255.255.0.0  192.168.10.10
```

2. Check the reachability of the VyOS VPN Gateway instance in GCP by running a ping to the private IP (for example 10.10.X.X) address as show in the GCP Console.

Deploying and Configuring the Velostrata Manager Virtual Appliance

Download and Install On-Premises

Follow the instructions in **Deploying Velostrata On-Premises**. This will guide you through:

- **Configuration of the Velostrata service role in vCenter.**
- **Deployment of the Velostrata Manager virtual appliance on-prem.**
- **Registration of the Velostrata vCenter Web Client Plugin.**

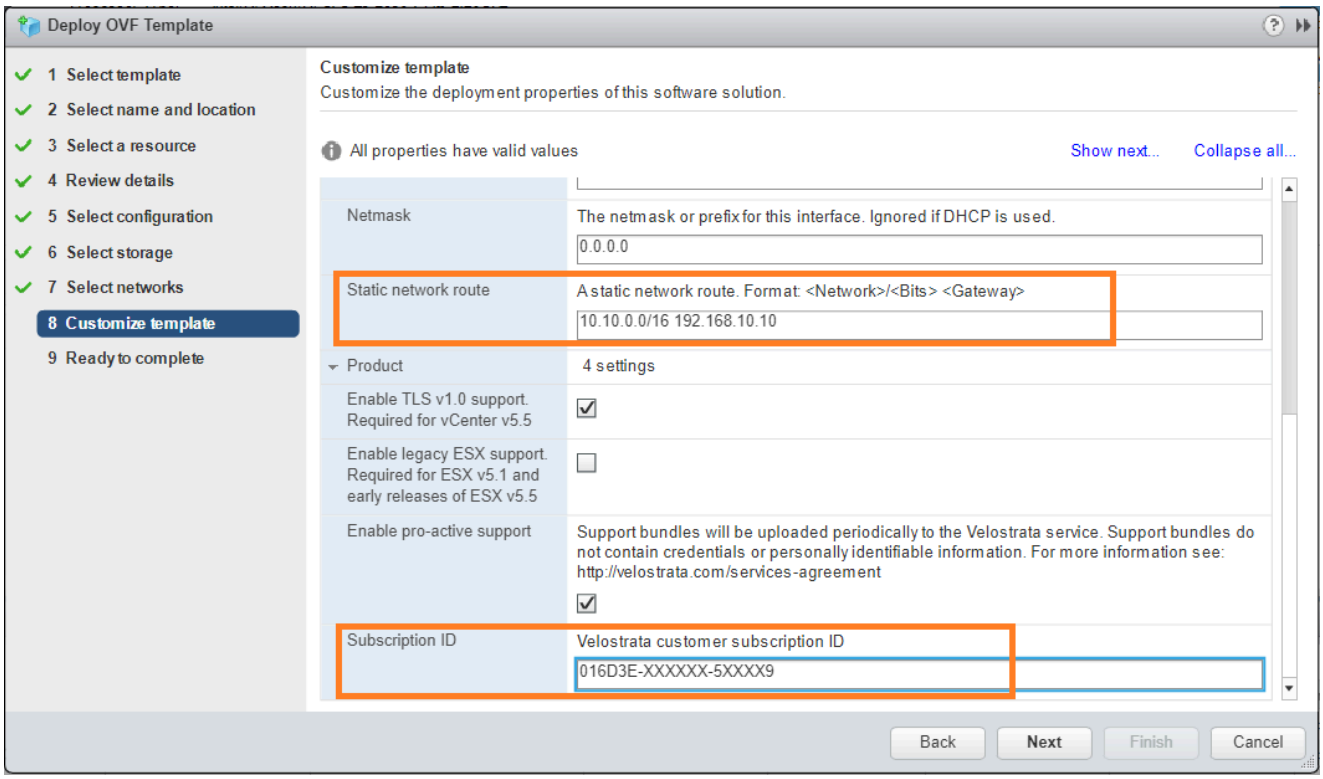
Note: Make sure to follow the PoC configuration parameters below, which will be needed during the deployment to configure the system according to the setup described in this document.

Configuration Parameters for this PoC Example:

- **Subscription ID:**You will need to provide a Subscription ID, which is the GCP Billing account ID.
- **Static Network Route:** For simplicity, in this PoC environment we do not require to change core routing switches or routers to be aware of the GCP VPC and the VPN gateway connecting to it. With this static network route, the Velostrata Manager virtual appliance will be able to reach the Velostrata Cloud Edge Nodes in the VPC as well as workload virtual machines for console probe checks.

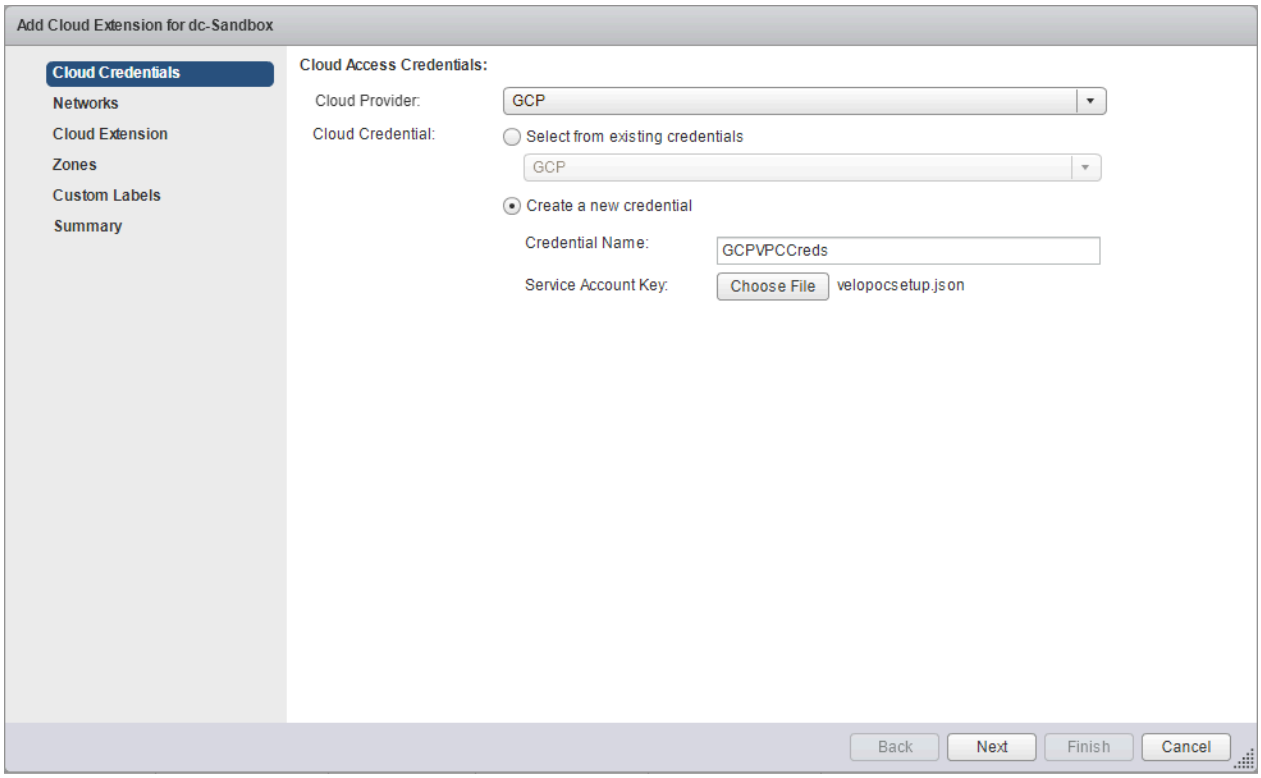
Enter the GCP VPC CIDR configured in the script, and the static IP of the VyOS virtul appliance on-prem. For this example:

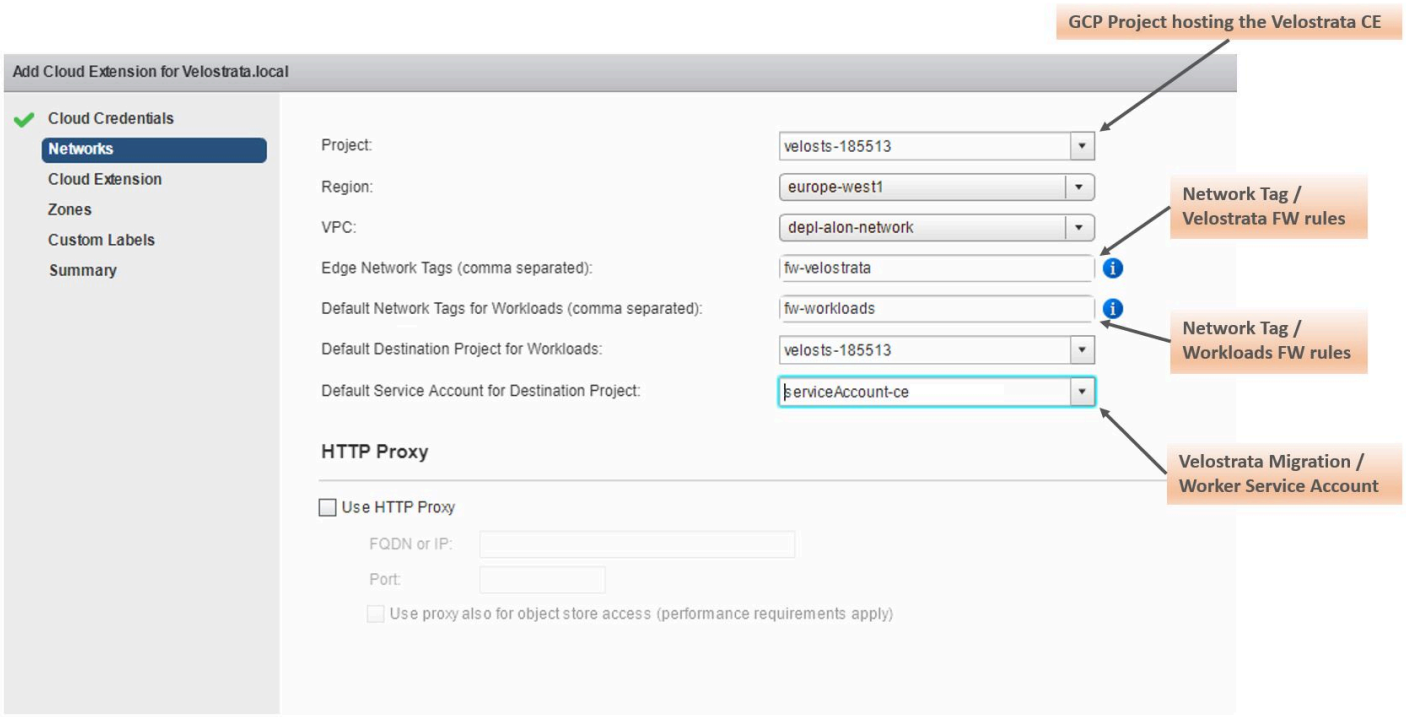
10.0.0.0/16 192.168.10.10

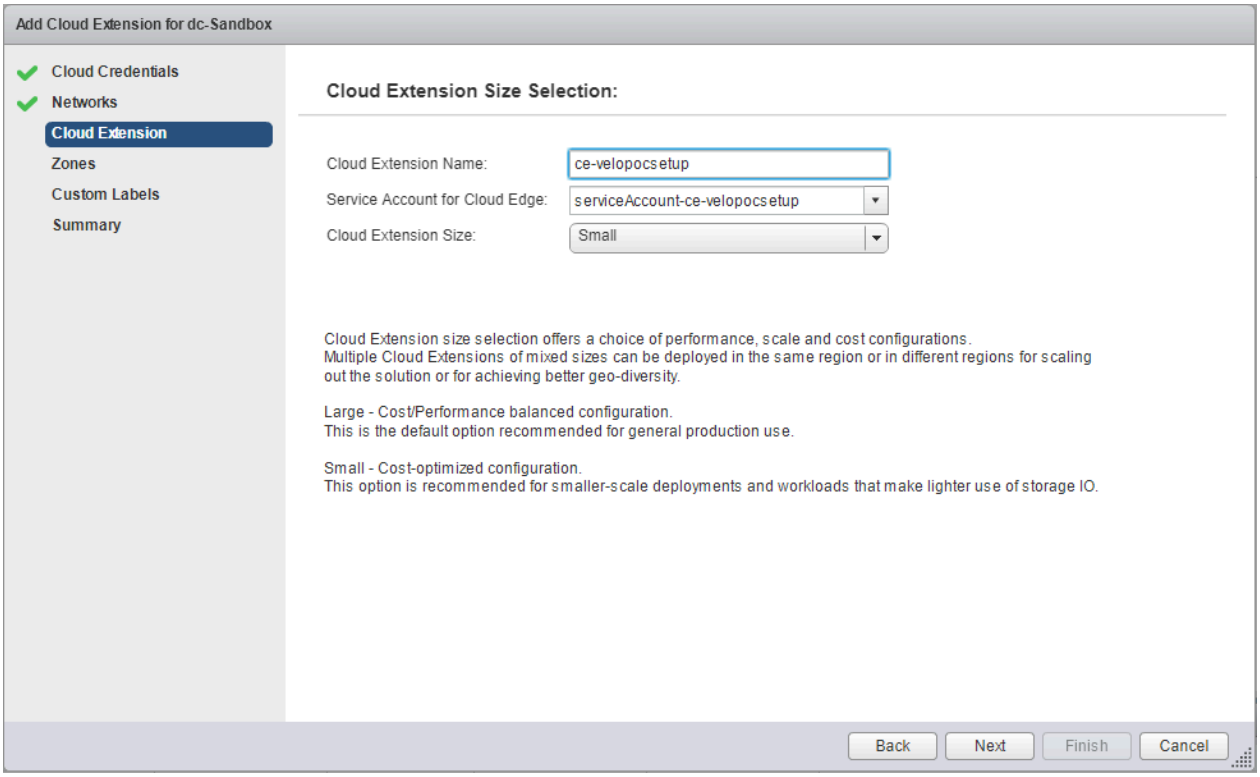

Deploy OVF Template
1 Select template
2 Select name and location
3 Select a resource
4 Review details
5 Select configuration
6 Select storage
7 Select networks
8 Customize template
9 Ready to complete
Customize template
Customize the deployment properties of this software solution.
All properties have valid values
Show next...
Collapse all...
Netmask
The netmask or prefix for this interface. Ignored if DHCP is used.
0.0.0.0
Static network route
A static network route. Format: <Network>/<Bits> <Gateway>
10.10.0.0/16 192.168.10.10
Product
4 settings
Enable TLS v1.0 support. Required for vCenter v5.5
Enable legacy ESX support. Required for ESX v5.1 and early releases of ESX v5.5
Enable pro-active support
Support bundles will be uploaded periodically to the Velostrata service. Support bundles do not contain credentials or personally identifiable information. For more information see: http://velostrata.com/services-agreement
Subscription ID
Velostrata customer subscription ID
016D3E-XXXXXX-5XXXX9
Back
Next
Finish
Cancel

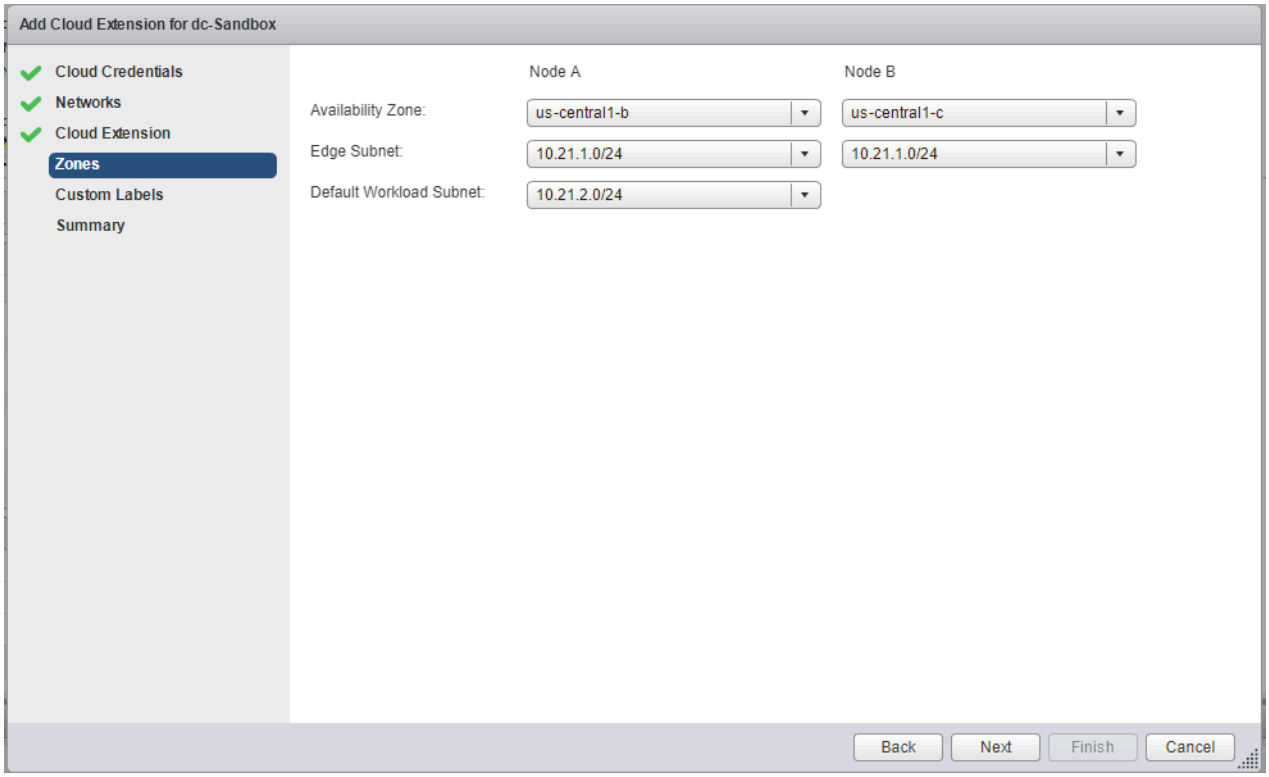
Creating a Velostrata GCP Cloud Extension

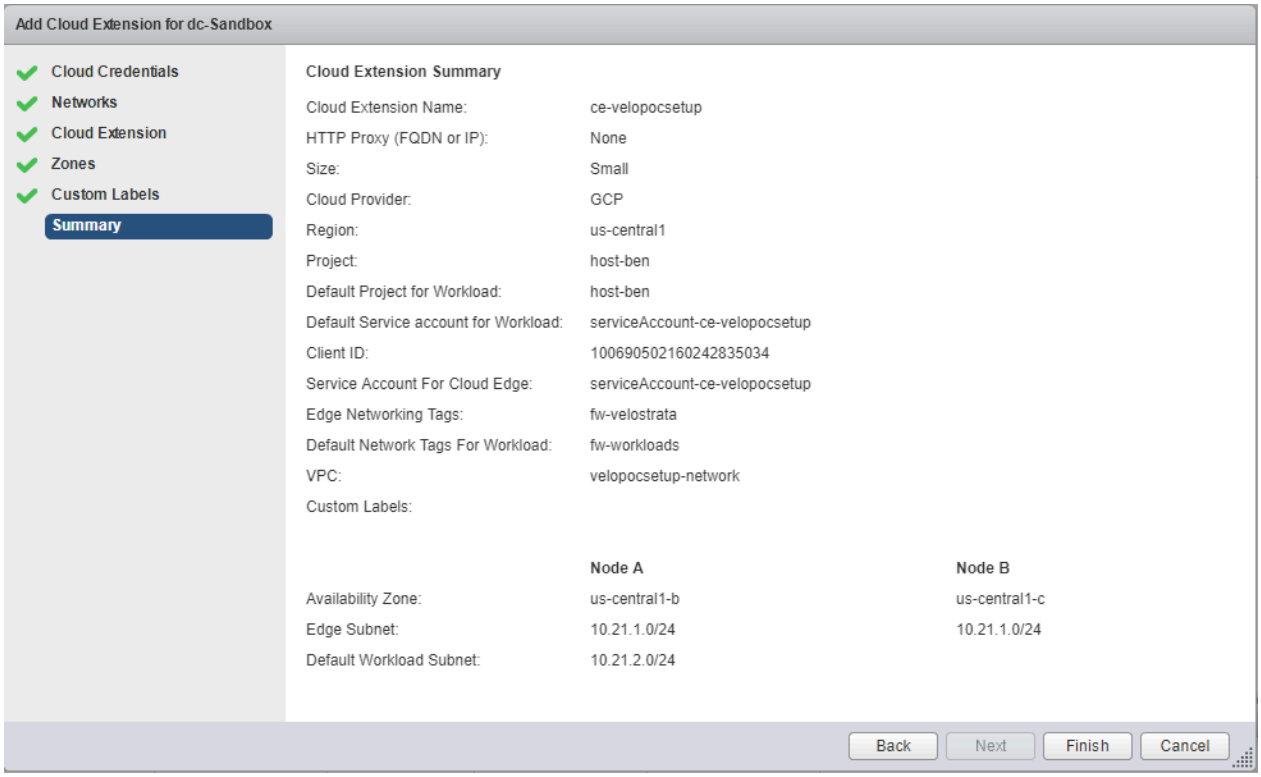
Follow the instructions in **Adding a Cloud Extension** in the User Guide, when creating the Cloud Extension - use the following for destination parameters:




GCP Project hosting the Velostrata CE
Add Cloud Extension for Velostrata.local
Cloud Credentials
Networks
Cloud Extension
Zones
Custom Labels
Summary
Project:
velosts-185513
Region:
europe-west1
VPC:
depl-alon-network
Edge Network Tags (comma separated):
fw-velostrata
Default Network Tags for Workloads (comma separated):
fw-workloads
Default Destination Project for Workloads:
velosts-185513
Default Service Account for Destination Project:
serviceAccount-ce
Network Tag / Velostrata FW rules
Network Tag / Workloads FW rules
Velostrata Migration / Worker Service Account
HTTP Proxy
Use HTTP Proxy
FQDN or IP:
Port:
Use proxy also for object store access (performance requirements apply)


Add Cloud Extension for dc-Sandbox
Cloud Credentials
Networks
Cloud Extension
Zones
Custom Labels
Summary
Cloud Extension Size Selection:
Cloud Extension Name:
ce-velopocsetup
Service Account for Cloud Edge:
serviceAccount-ce-velopocsetup
Cloud Extension Size:
Small
Cloud Extension size selection offers a choice of performance, scale and cost configurations.
Multiple Cloud Extensions of mixed sizes can be deployed in the same region or in different regions for scaling out the solution or for achieving better geo-diversity.
Large - Cost/Performance balanced configuration.
This is the default option recommended for general production use.
Small - Cost-optimized configuration.
This option is recommended for smaller-scale deployments and workloads that make lighter use of storage IO.
Back
Next
Finish
Cancel


Add Cloud Extension for dc-Sandbox
Cloud Credentials
Networks
Cloud Extension
Zones
Custom Labels
Summary
Node A
Node B
Availability Zone:
us-central1-b
us-central1-c
Edge Subnet:
10.21.1.0/24
10.21.1.0/24
Default Workload Subnet:
10.21.2.0/24
Back
Next
Finish
Cancel


Add Cloud Extension for dc-Sandbox
Cloud Credentials
Networks
Cloud Extension
Zones
Custom Labels
Summary
Cloud Extension Summary
Cloud Extension Name: ce-velopocsetup
HTTP Proxy (FQDN or IP): None
Size: Small
Cloud Provider: GCP
Region: us-central1
Project: host-ben
Default Project for Workload: host-ben
Default Service account for Workload: serviceAccount-ce-velopocsetup
Client ID: 100690502160242835034
Service Account For Cloud Edge: serviceAccount-ce-velopocsetup
Edge Networking Tags: fw-velostrata
Default Network Tags For Workload: fw-workloads
VPC: velopocsetup-network
Custom Labels:
Node A
Node B
Availability Zone: us-central1-b us-central1-c
Edge Subnet: 10.21.1.0/24 10.21.1.0/24
Default Workload Subnet: 10.21.2.0/24
Back
Next
Finish
Cancel

Running Your First VM in Cloud

Pre-requisites

No pre-installation is required for migration of Windows virtual machines (see release notes for supported versions).

If the virtual machine you select runs a Linux, you will need to first install the pre-requisite Velostrata preparation RPM package and its dependencies. visit Deploying Velostrata Prep Package for Linux Virtual Machines page to obtain the specific package and for installation instructions.

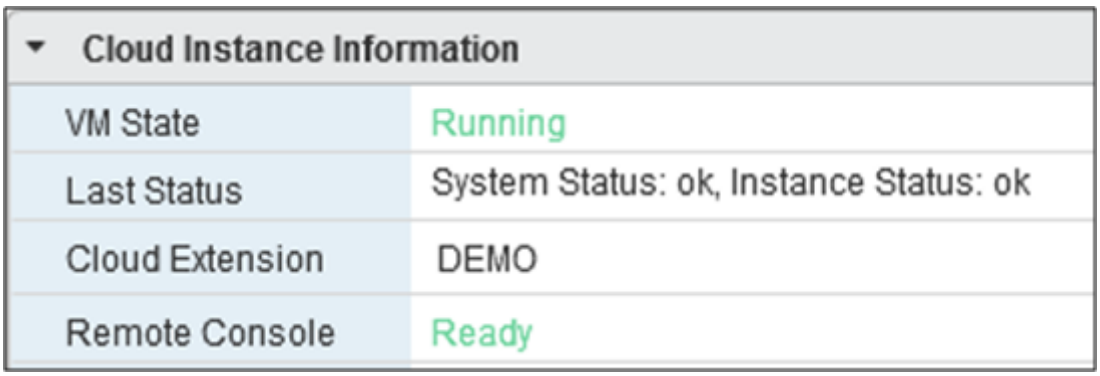
Wait for the Cloud Extension to be marked **Active** in the Datacenter summary page, in the vCenter Web Client.

You can now perform an end-to-end trial run of the system by selecting a virtual machine and running it in the cloud.

Select a virtual machine designated for test purposes.

1. Right-click the Virtual Machine, select **Velostrata operations** > **Run in Cloud**. A wizard appears.
2. In the wizard, select the **Velostrata Cloud Extension** you have created. Click **Next**.
3. Select the **GCP VM Size** for the workload. Click **Next**.
4. For **Storage policy**, leave the default selection (**Write back**). Click **Next**.
5. Select a **Subnet** for the workload. Click **Next**.
6. Review the **Summary** page and then click **Finish**.
7. Notice that the VM icon in the virtual machines inventory changesand it will be marked as **managed by Velostrata**.
8. To monitor the run-in-cloud task progress, navigate to the Virtual Machine summary page and review the **Cloud Instance Information** portlet. Wait for the **Remote Console** probe to show **Ready**.

Refer to the Velostrata User Guide for detailed instructions on Velostrata VM Operations.



Access the VM in cloud

To access the VM, now an instance running in GCP, we will need to make use of the jump server or PC you have prepared earlier with a static route to the VPN gateway.

1. Login or connect to the jump server using RDP.
2. RDP to the running instance on GCP using its FQDN (if dynamic DNS updates are enabled on the DNS server) or its Private IP address as shown in the Cloud Instance Information portlet.

3. Once logged into the virtual machine now in GCP, you may browse around, run your preferred test applications, or for example, create a test file on the desktop which you can inspect after moving the machine back on-premises, to experience the write-back feature.
4. You may review performance metrics for your virtual machine when running in cloud by selecting the **Monitoring** page in vCenter for the virtual machine , selecting the **Cloud Instance** tab, and then inspecting the various graphs shown, such as IOPS, IO latency, and so on. Additional monitoring graphs are available in the **Datacenter > Monitoring > Velostrata Service** tab. These statistics are at the Velostrata Cloud Extension level, and aggregate metrics across all virtual machines running in cloud using this Cloud Extension.

Run the VM back on-premises.

1. In vCenter, right-click on the virtual machine, then select **Velostrata operations** > **Run on Premises**. A dialog appears. Read through the on-screen notes and confirm the operation.
2. When the **Run on Premises** task is complete, the virtual machine no longer appears as managed by Velostrata, and remains shut down. **Start the virtual machine**.
3. When the virtual machine is back online, connect to it using RDP or its virtual console, and verify that all changes made while running in cloud are in place.

The Velostrata system is now ready for your PoC activities. We recommend that you work with a Velostrata sales engineer or a support representative to effectively plan your use PoC use cases, evaluation goals and success criteria.

Your Velostrata Setup is Ready!

Your Velostrata PoC setup is now ready for further testing and evaluation. Should you wish to retain the setup for production use, we recommend that you replace the VPN configuration with a production-ready solution such as GCP Cloud Interconnect..

Appendix A: Advanced VPN Configurations

Configuring VPN using TCP

In some environments, outbound UDP connections may be blocked. This will prevent the VPN configuration discussed earlier in this document from working correctly. In these cases, it is possible to set up the VPN tunnel over TCP.

Important Note: Using TCP for a VPN tunnel will significantly impact performance in case of link congestion. We recommend to apply QoS policies (example later in this appendix) to constrain the bandwidth consumed by the tunnel.

To configure VPN using TCP:

1. **VyOS virtual appliance on-premises:** Connect to the VyOS virtual appliance console on-premises and enter the following.

```
config
set interfaces openvpn vtun0 protocol tcp-active
set interfaces openvpn vtun0 openvpn-option "--sndbuf 131072"
set interfaces openvpn vtun0 openvpn-option "--rcvbuf 131072"
set interfaces openvpn vtun0 remote-port 443
commit
save
exit
```

2. **Update VPN Security Group:** Log in to the GCP console, and edit the firewall rules with the **fw-vpn** network tag. Add a rule to allow incoming connections on TCP port 443 (typically used for HTTPS). You may remove the pre-defined rule to allow connections on port 1194.
3. **VyOS instance in GCP**: Connect to the VyOS instance in GCP using PuTTY, and enter the following.

```
config
set interfaces openvpn vtun0 protocol tcp-passive
set interfaces openvpn vtun0 openvpn-option "--sndbuf 131072"
set interfaces openvpn vtun0 openvpn-option "--rcvbuf 131072"
set interfaces openvpn vtun0 local-port 443
commit
save
exit
```

Configuring VPN to Tunnel Through HTTP Proxy

In certain environments, internet access is only allowed through an HTTP proxy. In such environments you may setup the VPN tunnel to pass through the HTTP Proxy as follows.

Important Note: Using TCP for a VPN tunnel significantly impacts performance in case of link congestion. Tunneling through an HTTP proxy may introduce further response time delays and bandwidth constraints. We recommend applying QoS policies (there is an example later in this appendix) to constrain the bandwidth consumed by the tunnel.

To configure VPN to tunnel through HTTP proxy:

1. Follow the instructions to configure TCP-based VPN. See **Configuring VPN using TCP**.
2. Connect to the VyOS virtual appliance console **on-premises** and enter the following.
3. Replace **PROXY_IP** with the IP address of the HTTP proxy allowing outbound access to the internet, and replace **PROXY_PORT** with the applicable proxy port. The two are **separated by a single space** character.

```
config
set interfaces openvpn vtun0 openvpn-option "--http-proxy PROXY_IP PROXY_PORT"
commit
save
exit
```

Note: This configuration supports HTTP proxies that do not require authentication. It also requires that the HTTP proxy supports the HTTP CONNECT method.

Applying a QoS Policy to Constrain Bandwidth Usage

When using VPN over TCP, or when you wish to constrain the bandwidth usage of the VPN tunnel, it is recommended to set up a QoS policy on the VyOS machines on both sides on-premises and in GCP.

Connect to each VyOS machine and apply the following configuration.

- **Bandwidth Limit:**
  - For the **VyOS on-premises**, replace 100Mbit in the reference policy below with the **uplink** bandwidth you wish to constrain the VPN tunnel to, in context of your internet link.
  - For the **VyOS in GCP**, replace 100Mbit with the **downlink** bandwidth you wish to constrain the return path of the VPN to, in context of your internet link.
  - **Example Scenario:** You have an asymmetric internet connection on-premises, with 50Mbit/sec uplink and 100Mbit/sec downlink and you want to constrain the uplink allocation for the VPN to be max 40Mbit and downlink to be max 50Mbit. In the VyOS on-premises you replace the 100Mbit value with 40Mbit, and in the VyOS in GCP you replace the 100Mbit value with 50Mbit.
- **Velostrata Secure Datapath Limit:**
  - Within the bandwidth allocated to the VPN tunnel, you now have control on how much you would allocate for the Velostrata security Datapath channel. It is important to leave enough room for other application traffic and other control traffic. In the example below the split during congestion periods is 40% to Velostrata Datapath channel (class 100) and 60% for other traffic (default class). Each traffic class can burst higher when there is no congestion conflict.
  - Replace the 40% and 60% values respectively with your preference. Make sure to allocate a minimum of 20Mbit/sec for the Velostrata channel in either direction during congestion periods, and preferably higher when running in production.

```
config
set traffic-policy shaper WAN-OUT bandwidth '100Mbit'
set traffic-policy shaper WAN-OUT class 100 bandwidth '40%'
set traffic-policy shaper WAN-OUT class 100 burst '2kb'
set traffic-policy shaper WAN-OUT class 100 ceiling '90%'
set traffic-policy shaper WAN-OUT class 100 description 'VELOS'
set traffic-policy shaper WAN-OUT class 100 match VELOS ip protocol tcp
```

```
set traffic-policy shaper WAN-OUT class 100 match VELOS ip destination port '9111'
set traffic-policy shaper WAN-OUT class 100 queue-type 'drop-tail'
set traffic-policy shaper WAN-OUT default bandwidth '60%'
set traffic-policy shaper WAN-OUT default burst '2kb'
set traffic-policy shaper WAN-OUT default ceiling '100%'
set traffic-policy shaper WAN-OUT default queue-type 'fair-queue'
set interfaces openvpn vtun0 traffic-policy out 'WAN-OUT'
commit
save
exit
```