AVERE

vFXT Installation Guide for Google Cloud Platform (GCP)

Revised: 2017-09-12

Introduction

The vFXT Series Platform

The Avere virtual FXT (vFXT) Series Platform is a virtual server, or *node*, that runs the Avere OS software. Three or more nodes comprise a vFXT cluster.

The vFXT nodes are created using images for Google Compute Engine (GCE). The nodes run as virtual machines within GCE as an n1-highmem-8 instance or an n1-highmem-32 instance type. An instance is a virtual machine in GCE.

vFXT Instance Types

Instance Size	vCPUs	Memory (GiB)	Network
n1-highmem-8	8	52	8 Gb
n1-highmem-32*	32	208	16 Gb

* 32-core machine types are not available in zones us-central1-a and europe-west1-b. Contact support if this impacts you.

Avere uses persistent disks for the system drives while the data drives have the option to be either persistent (recommended) or local. Persistent SSD is recommended because the data on those drives will remain after reboot. The advantages of Local SSD are that it is less expensive and provides greater throughput (IOPS). Because of these advantages, customers choose Local SSD for burst-type workloads.

When starting out, the recommended cluster size is 3 nodes. Nodes may be added one at a time after cluster creation. If you know you want more nodes in the cluster, you can begin with a larger number and then add or remove nodes one at a time until you have the ideal cluster size. A vFXT cluster can be up to 20 nodes with 12 nodes being a suggested maximum.

Buckets for storage are automatically created during the vFXT cluster creation process. This cloud bucket is then used as the first core filer in your new vFXT cluster. The creation of this bucket is optional but recommended. If you plan to only use NFS core filers, you may omit this.

Avere Documentation

The following documentation is provided to supplement this guide:

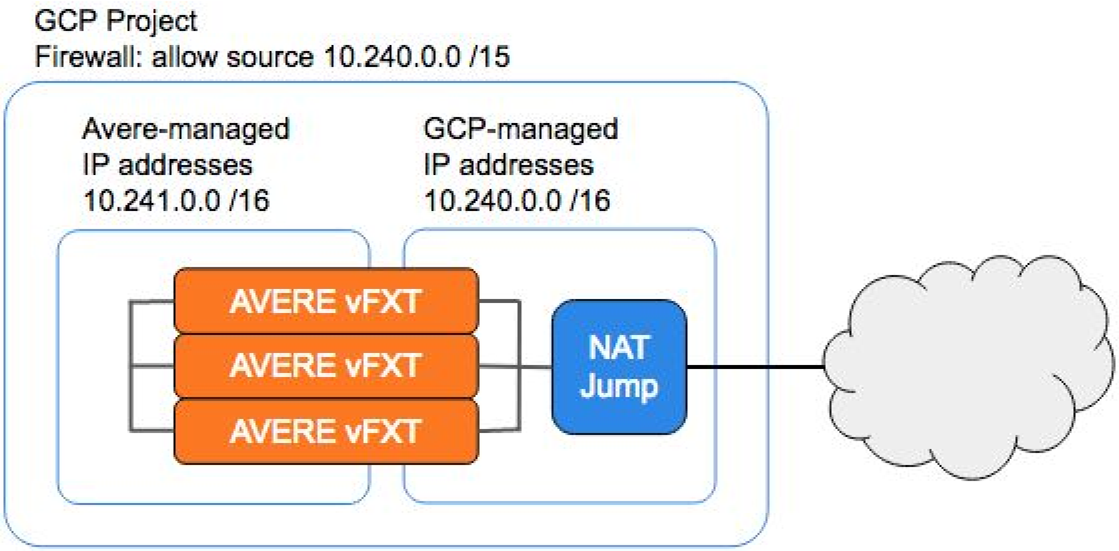
- Quick Start Guide – How to set up and configure an Avere production system.
- Operations Guide – How to administer the Avere OS software.
- Release Notes – Late-breaking information about the Avere product.
- Third-Party Licenses Reference – Licensing information for the third-party software used by the Avere product.

Product Requirements

There are several requirements for creating a vFXT cluster in GCP. A reference diagram and more Information about each can be found below.

1. Establish a Google Cloud Platform (GCP) account
2. Create a project in which the vFXT will run
3. Provide project number and/or compute ID to Avere representative
4. Decide which type of SSDs to use ("persistent" recommended)
5. Request quota increase for both CPU cores and SSD storage
6. Enable firewall rules to include Avere-managed IP addresses
7. Ensure API access is available for instances within the subnet (see Appendix)

Reference diagram



GCP Project

Projects form the basis for enabling and using the Google Cloud Platform services, including managing APIs, enabling billing, adding and removing collaborators and enabling other Google services. Each project is a totally compartmentalized world. Projects do not share resources, can have different owners and users, are billed separately, and are managed separately.

To create a project, first log in to the GCP Console. Create a project from the Dashboard.

GCP Regions and Zones

GCP has different regions, which are geographical locations (like Asia Northeast and US Central), and zones which are sites within that region. Each of those regions have at least 3 different zones. For example, the US Central region has zones 1-a, 1-b, 1-c, and 1-f.

For the latest information on GCP regions and zones, refer to Google Cloud Platform Docs.

Project Quotas

Google limits the number of resources users can leverage like number of CPU cores, public IP addresses, number of VM instances, and amount of SSD. Most quota limits are set per region although there are some that are defined per project. If any of the affected quota limits are exceeded, you won't be able to add more resources of the same type in that project or region.

Quota increase requests are processed by providing a total versus requesting additional. For example, if your project already allows up to 24 CPU cores and you need 30 more, then the request should include 54 cores or more, not 30.

Work with your Avere Systems Engineer or Avere Global Support to determine the additional project resources needed. The smallest vFXT installation will require over 30 cores and 4TB (4096GB) of persistent SSD in addition to any existing project resources.

To request a quota increase, go to the Quotas page, select a project (if needed), and click the Request increase button (pencil icon) to the right of the service. Previous requests have taken 24-48 hours to approve but some may be completed within minutes due to quota improvements.

GCP Networking

The Avere vFXT is designed to work in both subnet and legacy (non-subnet) networks. Subnet networks are the default installation. If needing a Legacy network, follow these instructions.

GCP manages a range of IP addresses based on the network range. Avere manages a set of IP addresses outside of the GCP-managed range in order to provide failover for client and management IP addresses.

If you created a custom subnet that was 10.10.0.0/20, for example, the vFXT would use the 10.10.16.0/20 address space for the Avere-managed IP addresses. If you were using an automatic network and subnets, Avere will scan the IP address range and will use an address outside of that range. For example, if you were using us-central1 (10.128.0.0/20) and the last CIDR block in that network was 10.148.0.0/20, then Avere would use 10.149.0.0/20 addresses.

Create a new firewall rule to include both Google-managed and Avere-managed IP traffic. For the first example, you could allow TCP, UDP, and ICMP 10.10.0.0/19. For the second example, you could allow TCP, UDP, and ICMP from the sources 10.128.0.0/20 and 10.149.0.0/20.

If using a Google VPN, review the Google VPN section in the Appendix.

External Access and Name Resolution

In order for the vFXT nodes to issue API commands to GCP and to access the licensing server, the vFXT must be able to resolve and access the following URLs:

- https://www.googleapis.com
- https://<bucketName>.storage.googleapis.com
- download.averesystems.com

See the Ports and Domain Whitelist sections in the appendix for a full list firewall considerations.

External access is often accomplished by implementing a NAT instance. See Common Configuration - Using NAT section below for more information. Other options include Carrier Interconnect (leased line), VPN, or Direct Peering (BGP at Google endpoint).

Review the Internet Exposure section for warnings.

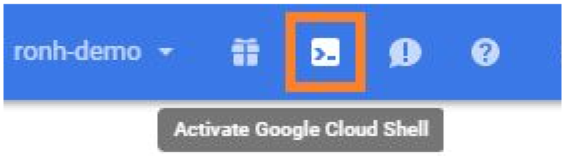
Accessing Avere Resources

Access to the Avere vFXT distribution project is granted to customers based on their project ID or project name. Email this information to your Avere SE to grant access to vFXT images.

Access Google Cloud Shell

Google Cloud Shell provides you with command-line access to computing resources hosted on GCP and is available in the GCP Console.

Activate Google Cloud Shell by clicking the icon in the upper right.



Issuing Commands

To launch Avere instances in Google Compute Engine (GCE), an administrator must execute a series of commands. These commands may be issued to GCP via a GCP instance.

Steps for installing the vFXT can be found in the Installing a vFXT series Edge Filer on GCP section in this document.

Precautions

Internet Exposure

Access to Google APIs is a requirement of vFXT nodes, but direct Internet exposure is not.

Just like other servers in your network, the vFXT nodes must sit behind a firewall to protect them against attacks. **Avere vFXT nodes are not hardened for direct Internet access.** Most Avere customers use a virtual NAT instance to allow designated traffic to traverse public (Internet) and private (project, internal) networks. Other customers extend their corporate network infrastructure to GCP. For example, all network traffic must pass through a firewall to traverse between the corporate network and the Internet.

For more information, refer to Google documentation on Networks, Firewalls, Routing, and NAT.

SSD Option

GCP offers two SSD options: Local and Persistent. Both are supported by the vFXT instance but before choosing, administrators must be aware of the consequences of using Local SSD.

Persistent SSD is recommended because the data on those drives will remain after reboot. Conversely, Local SSD data does not persist after a node reboot. This means that if a node or the cluster would be stopped, all node data and configuration will be lost.

The advantages of Local SSD are that it is less expensive and provides greater throughput (IOPS). Because of these advantages, customers choose Local SSD for burst-type workloads. For customers who want the vFXT cluster to survive a reboot, Persistent SSD must be chosen.

Contact your Avere Systems Engineer or Avere Global Support before choosing Local SSD.

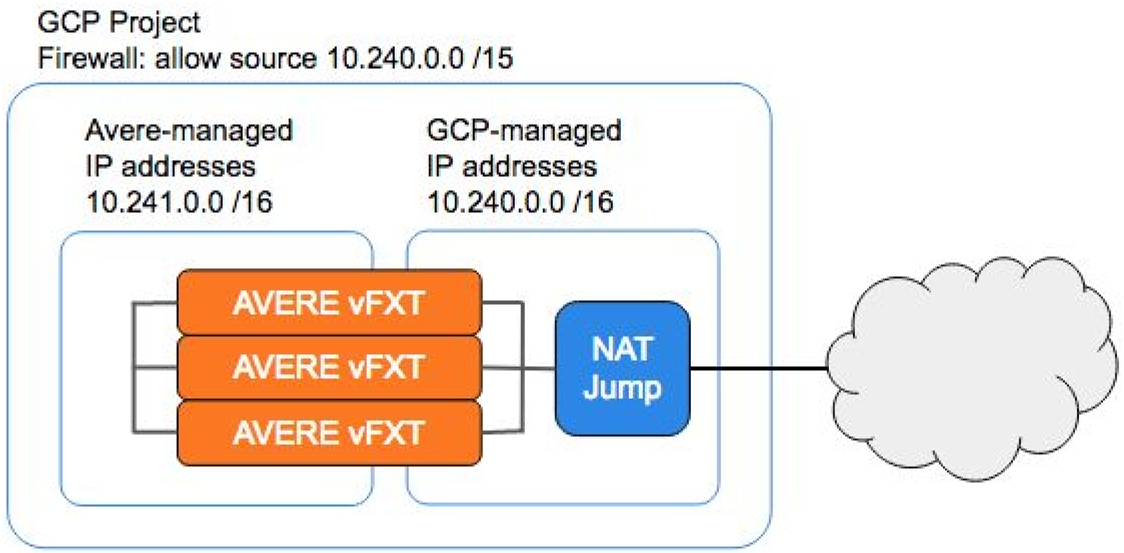
GCP Account Charges

Google Cloud Platform charges are incurred for, but are not limited to, use of GCE instances, disk storage (SSD) even when instances are not running, GCS, number of requests made to GCP, and for data entering and exiting GCP.

Avere recommends that administrators monitor all GCP charges.
For more information, refer to Google's pricing page as well as Google's documentation on Monitoring Estimated Charges.

Installing a vFXT Series Edge Filer on GCP

Reference diagram



Firewall Rules

If the firewall rules have not yet been edited to allow both Google-managed and Avere-managed IP traffic, then complete complete this step first. For example, allow TCP, UDP, and ICMP 10.240.0.0/15.

This process makes several assumptions about your project's networking and is only a guide for how you might choose to grant access. If you are unsure what network to choose or what specific changes to make, please work with your networking team.

1. Navigate to Firewall rules.
2. Click the rule to be edited. This is often the default-allow-internal rule.
3. Click the **Edit** button.
4. In the **Source IP ranges** field, decrement the number after the slash by 1. For example, if the field is "10.240.0.0/16" then change it to "10.240.0.0/15".
5. Click the **Save** button.

Common Configuration - Using NAT

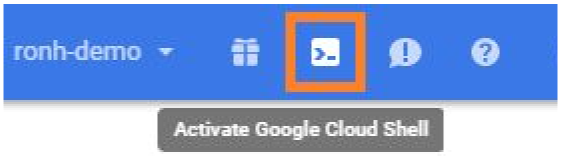
Many customers use a GCE instance to provide Network Address Translation (NAT) services for project instances to have external network access. Avere instances need Internet access in order to execute GCP API commands to set up and configure the vFXT cluster.

There are several steps in getting the NAT instance ready for the project. We will create the NAT, configure it (IP forwarding and masquerading), add a route so the Avere instances can use it to make API calls, and make its configuration persistent so that it functions after a reboot.

The following steps are for creating and configuring a NAT instance via Google Cloud Shell (command line). If you prefer to create the instance via the Google Cloud Console, instructions may be found in the Appendix.

Creating and Configuring a NAT Instance via Google Cloud Shell

1. If not already open, open the Google Cloud Shell by clicking the icon in the upper right.

2. Create the NAT instance replacing the `<fields>` with the information specific to your project:

   ```
   gcloud compute instances create nat-gateway --network <network>
   --subnet <subnet> --can-ip-forward --zone <zone> --machine-type
   n1-standard-4 --image-family=debian-8 --scopes=compute-rw,storage-rw
   --image-project=debian-cloud
   ```

   Example:

   ```
   gcloud compute instances create nat-gateway --network default
   --can-ip-forward --zone us-central1-f --machine-type n1-standard-4
   --image-family=debian-8 --scopes=compute-rw,storage-rw
   --image-project=debian-cloud
   ```

3. Create a route for the NAT instance replacing the `<fields>` with the information specific to your project:

   ```
   gcloud compute routes create nat-internet-route --network <network>
   --destination-range 0.0.0.0/0 --next-hop-instance nat-gateway
   --next-hop-instance-zone <zone> --tags use-nat --priority 800
   ```

   Example:

   ```
   gcloud compute routes create nat-internet-route --network default
   --destination-range 0.0.0.0/0 --next-hop-instance nat-gateway
   --next-hop-instance-zone us-central1-f --tags use-nat --priority 800
   ```

4. Set firewall rules to allow TCP, UDP, and ICMP among instances.

   ```
   gcloud compute firewall-rules create <ruleName> --network <network>
   --allow tcp,udp,icmp --source-ranges <subnetCIDR>
   ```

   Example:

   ```
   gcloud compute firewall-rules create allow-proj-traffic --network
   default --allow tcp,udp,icmp --source-ranges 10.128.0.0/9
   ```

5. Make the NAT services perpetual.

   Adding the NAT Startup Script:

   a. Navigate to the **VM instances** page.
   b. Click on the "nat-gateway" instance.
   c. Click the **Edit** button at the top of the page.
   d. Scroll down to **Custom metadata** and paste the following:

      Key: `startup-script`

      Value:

      ```
      #!/bin/sh
      sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
      iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
      ```

   e. Click "Save" at the bottom of the page.

6. Reboot and verify settings

   a. Navigate to the **VM instances** page.
   b. Check the box next to **nat-gateway** instance.
   c. Click the Reset button at the top. **This is required**.
   d. After the reboot, SSH into the nat-gateway (click SSH button)
   e. Run `sudo sysctl -a | grep ip_forward`

      and verify `net.ipv4.ip_forward = 1`
   f. Run `sudo iptables --list -t nat | grep MAS`

      and verify `MASQUERADE  all  --  anywhere  anywhere`
   g. If either of the responses in steps 6e or 6f do not verify, repeat steps 5 and 6.

Creating a vFXT Cluster

The vFXT creation process uses the vfxt.py script. The script streamlines the creation process by using GCP's APIs. vFXT.py does not require being used on a cloud instance; connectivity to GCP is all that is required. If being used on a cloud instance, it must have permissions to create storage and compute. `--scopes=compute-rw,storage-rw`

Instructions for using vfxt.py can be found in the [vfxt.py Usage Guide]() on the [Avere Library page]().

vfxt.py Software Requirements

For GNU/Debian instances use:

```
sudo apt-get update && sudo apt-get install libffi-dev libssl-dev python-dev
python-pip && sudo easy_install pip && sudo pip install --upgrade boto
requests google-api-python-client vFXT
```

For CentOS and RHEL instances, use:

```
sudo yum -y update && sudo yum -y install libffi-devel openssl-devel
python-devel python2-pip && sudo pip install --upgrade pip boto requests
google-api-python-client vFXT
```

Select Cluster Options

The flexibility of the vfxt.py script allows for several configuration options. Use the following information to decide which options are best for your cluster. Additional information about each of these options can be found in the [vFXT Series Platform]() section. Please work with your Systems Engineer (SE) or Avere Global Support (AGS) to determine the best options.

Run the vfxt.py Script

From the Google Cloud Shell, execute the following script replacing `<fields>` with options that are specific to your cluster and environment.

```
vfxt.py --cloud-type gce --on-instance --create --network <network> --zone
<zone> --cluster-name <clusterName> --admin-password <clusterPassword>
--instance-type <instanceType> --debug  --data-disk-type <diskType>
--node-cache-size <cacheSize> --nodes <nodeNumber> --gce-tag use-nat
```

	vfxt.py script option	Recommended	Option
VM instance type	`--instance-type`	`n1-highmem-8`	`n1-highmem-32`
SSD type	`--data-disk-type`	`pd-ssd`	`local-ssd` (local)
Node cache size	`--node-cache-size`	`1000` (for pd) `750` (for local)	for persistent, up to 8000 for local, multiples of 375 up to 3000
Number of nodes	`--nodes`	`3`	Up to `20`
Bucket creation	`--no-corefiler`	Omit this option	`--no-corefiler`

Example 1:

```
vfxt.py --cloud-type gce --on-instance --create --network default --zone
us-central1-f --cluster-name avere-customer --admin-password P@55w0rd
--instance-type n1-highmem-8 --debug --data-disk-type pd-ssd
--node-cache-size 1000 --nodes 3 --gce-tag use-nat
```

Example 2:

```
vfxt.py --cloud-type gce --on-instance --create --network default --zone
us-east4-a --cluster-name fxtcluster1 --admin-password P@55w0rd
--instance-type n1-highmem-32 --debug --data-disk-type local-ssd
--node-cache-size 750 --nodes 3 --gce-tag use-nat
```

The completed command will display the management IP address. Note this IP address.

Tunnel Traffic to Access Cluster

Access Avere's Control Panel by tunneling HTTPS traffic through the NAT to the local machine.

1. Copy your local machine's public key
    a. From your local machine, run: `cat ~/.ssh/id_rsa.pub`
       If the machine does not have the id_rsa.pub file, run `ssh-keygen` to create one
    b. Copy the RSA key
       Ex: `ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDktuJqf2gI4hZ6K1j abQKJhXI49v4y04y+ThisIsNotMyActualKey+0J3cgFcBwdmVU6aPw70i7GxVW65 Hx4AVF50vTECdqe/dK08OS8wA/ ronh@averesystems.com`
2. Add the local machine's public key to the NAT instance
    a. In Google Console, navigate to Compute Engine > VM Instances
    b. Click the nat-gateway instance
    c. Click the **Edit** button at the top
    d. Scroll down to the SSH Keys section and paste the RSA key in the field which says **Enter entire key data**
    e. Scroll to the bottom and click the **Save** button
3. Create SSH tunnel
    a. Navigate to Compute Engine > VM Instances
    b. Copy the public IP address of the NAT instance
    c. From the terminal/console of the local machine:
       `ssh -L 8443:<mgmtIpAddress>:443 <username>@<NATpublicIP>`
       where <username> is the name that appears before the "@" at the end of the ssh-rsa key - e.g. `ssh -L 8443:10.128.0.4:443 ronh@104.154.71.70`
    d. Type `yes` if prompted and press Enter. It should display
       `<username>@nat-gateway:~$`

Accessing vFXT Nodes (HTTPS and SSH)

Once the SSH tunnel is in place, access Avere's Control Panel via the local machine's browser.

1. In your browser's address bar, enter https://127.0.0.1:8443/
2. Enter **admin** for the username
3. Enter the password given when running the vfxt.py script

Managing Instances

Note: You cannot stop and restart an instance that has a local SSD. If using a read/write cache policy, set the writeback timer to 0 and wait for all data to flush before terminating instances.

Stopping Instances

If you need to stop a vFXT instance (one node) or the entire cluster and intend to restart it later, Avere recommends using the Avere Control Panel. Individual nodes or the entire cluster can be gracefully stopped by choosing the corresponding option within System Maintenance. For more information about using the System Maintenance options, refer to the Configuration Guide.

If you need to stop an instance or the entire cluster and do not intend to restart it, you may terminate the instance using the tools within the GCP console. See Deleting Instances for more information. Instances using local SSD cannot be stopped or restarted - only deleted.

NOTE: While GCE charges will not be incurred while instances are stopped, storage charges will continue for buckets and volumes.

Restarting Instances

If instances have been stopped and need to be restarted, they will need to be restarted from the GCP Console. Instances using local SSD cannot be stopped or restarted - only deleted.

To restart instances:

1. Navigate to Compute Engine > VM Instances
2. Click the checkboxes for the instances to be restarted
3. Click the Restart button at the top.

Deleting Instances

NOTE: Deleted instances cannot be restarted or retrieved. Deleting an instance is a permanent action and cannot be undone. If deleting a single instance, first remove the node using the Avere Control Panel prior to deleting it from GCE. See Removing a Node for more information.

To delete instances:

1. Navigate to Compute Engine > VM Instances
2. Click the checkboxes for the instances to be deleted
3. Click the Delete button at the top.

Adjusting Cluster Sizes

Cluster sizes can be increased by adding a node or decreased by removing a node. Adding a node should be done using the vfxt.py script. Removing a node should be done within the Avere Control Panel. Avere recommends removing nodes under the supervision of Avere Global Services (AGS).

Adding a Node

Adding a node should be done using the vfxt.py script.

From the NAT box or jump box, execute the following script replacing `<fields>` with options that are specific to your cluster.

```
vfxt.py --add-nodes --cloud-type gce --nodes <number> --network default
--zone us-central1-f --on-instance --management-address <mgmtIpAddress>
--admin-password <adminPassword>
```

Example:

```
vfxt.py --add-nodes --cloud-type gce --nodes 1 --network default --zone
us-central1-f --on-instance --management-address 10.128.0.4 --admin-password
P@55w0rd
```

Removing a Node

Removing a node should be done within the Avere Control Panel. Avere recommends removing nodes under the supervision of Avere Global Services (AGS). Removing a node might take several minutes and affect client access. Refer to the Configuration Guide for more detail.

To remove a vFXT node from a cluster:

1. Log in to the Avere Control Panel
2. Click the **Settings** tab
3. Click the **FXT Nodes** link under the Cluster section
4. Click the **Remove** button to the right of the node to be removed
5. Wait for the node removal process to complete and for all conditions to be cleared
6. Stop or delete the instance

Optional: Remove a vFXT Cluster

To remove a vFXT cluster from your Google Cloud project, you can use the vfxt.py script. You will need both the management IP address and the admin password for the cluster.

From the Google Cloud Shell, execute the following script replacing `<fields>` with options that are specific to your cluster.

```
vfxt.py --cloud-type gce --destroy --management-address <mgmtIPpAddress>
--admin-password <adminPassword> --on-instance
```

Example:

```
vfxt.py --cloud-type gce --destroy --management-address 10.128.0.4
--admin-password P@55w0rd --on-instance
```

Appendix

Ports

Ports required for vFXT inbound and outbound communication

API

Inbound

TCP	22	SSH

Outbound

TCP	80	HTTP
TCP	443	HTTPS

NFS

Inbound and Outbound

TCP/UDP	111	RPCBIND
TCP/UDP	2049	NFS
TCP/UDP	4045	NLOCKMGR
TCP/UDP	4046	MOUNTD
TCP/UDP	4047	STATUS

SMB/CIFS

Inbound

TCP	445	SMB
TCP	139	SMB
UDP	137	NETBIOS
UDP	138	NETBIOS

Outbound

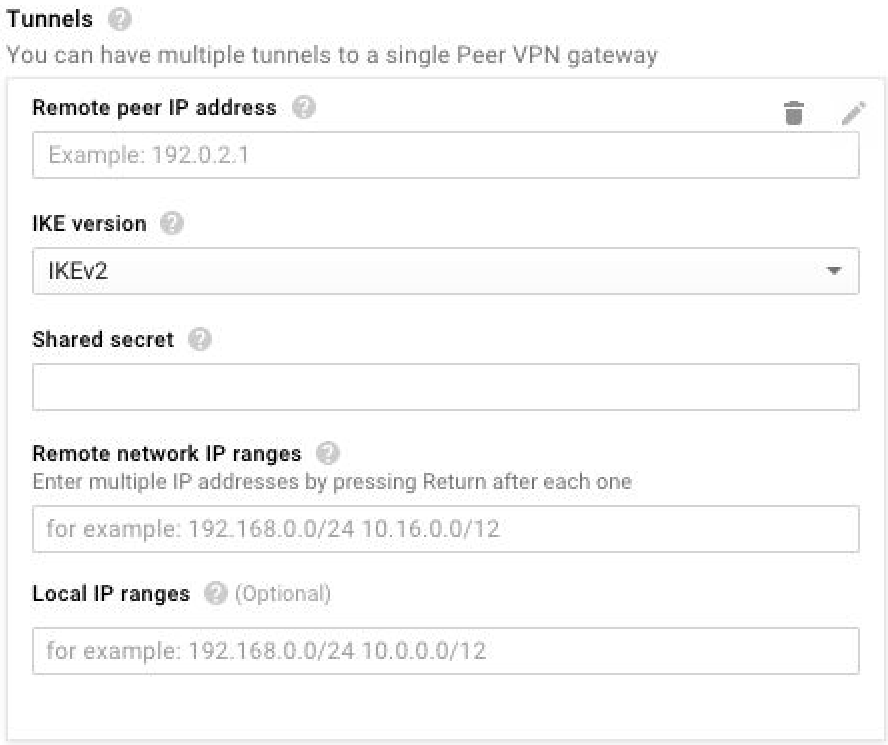
TCP/UDP	53	DNS
TCP/UDP	389	LDAP
TCP	686	LDAPS
TCP/UDP	88	Kerberos
UDP	123	NTP
TCP	445	SMB
TCP	139	SMB
UDP	137	NetBIOS
UDP	138	NetBIOS

Domain Whitelist

verisign.com
ocsp.verisign.com
SVRSecure-G3-crl.verisign.com
s3.amazonaws.com
sd.symcd.com
download.averesystems.com
www.googleapis.com
<bucket>.storage.googleapis.com

Optional: Google VPN

If using a Google VPN to connect the on-premises network with the GCP project, you will need to configure **Local IP ranges** in the Tunnels section of the GCP Console. This allows communication between on-prem IP addresses and Avere-managed IP addresses. Add all IP address ranges that need to communicate over the tunnel including Avere-managed and on-prem IP addresses. Note: If the tunnel is already established, you will need to recreate the tunnel with these settings. Configuration options are set during creation and cannot be changed while the tunnel is live but can be changed if the tunnel is re-negotiated. More information about setting up and configuring Google VPNs and tunnels can be found here.



Tunnels

You can have multiple tunnels to a single Peer VPN gateway

Remote peer IP address

Example: 192.0.2.1

IKE version

IKEv2

Shared secret

Remote network IP ranges

Enter multiple IP addresses by pressing Return after each one

for example: 192.168.0.0/24 10.16.0.0/12

Local IP ranges (Optional)

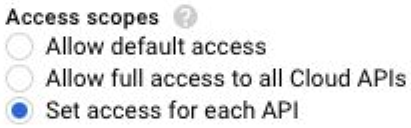
for example: 192.168.0.0/24 10.0.0.0/12

Creating and Configuring a NAT Instance via Google Cloud Console

Creating a Debian 8 instance in GCP will allow you to run the needed commands. The instance should be in the same VPC network where the cluster will be or it must at least have a route to communicate with the cluster's VPC network and subnetwork.

To set up an instance:

1. Navigate to the Compute Engine section of the Developer's Console
2. At the top, click the **CREATE INSTANCE** button
3. Name your instance - all lowercase, numbers, and hyphens (ex: ronh-linux1)
4. Choose your zone
5. Boot disk - verify it is the Debian GNU/Linux 8 (jessie) or 9 (stretch) image
6. Click **Set access for each API** unless you want to allow full access

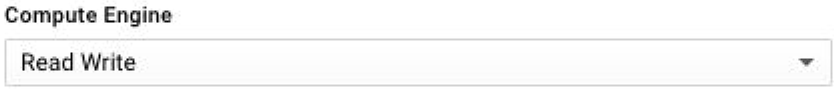
   Access scopes

   ○ Allow default access

   ○ Allow full access to all Cloud APIs

   ● Set access for each API

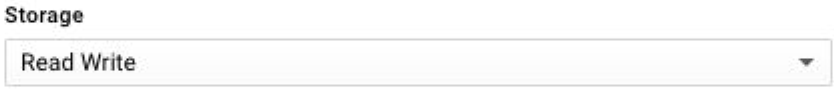
   a. Change the **Compute** field from None to **Read Write**

      Compute Engine

      Read Write

   b. Change the **Storage** field from Read Only to **Read Write**

      Storage

      Read Write

7. Click the **Management, disks, networking, SSH keys** link to expand it
8. Under the Management tab, add the following startup script:

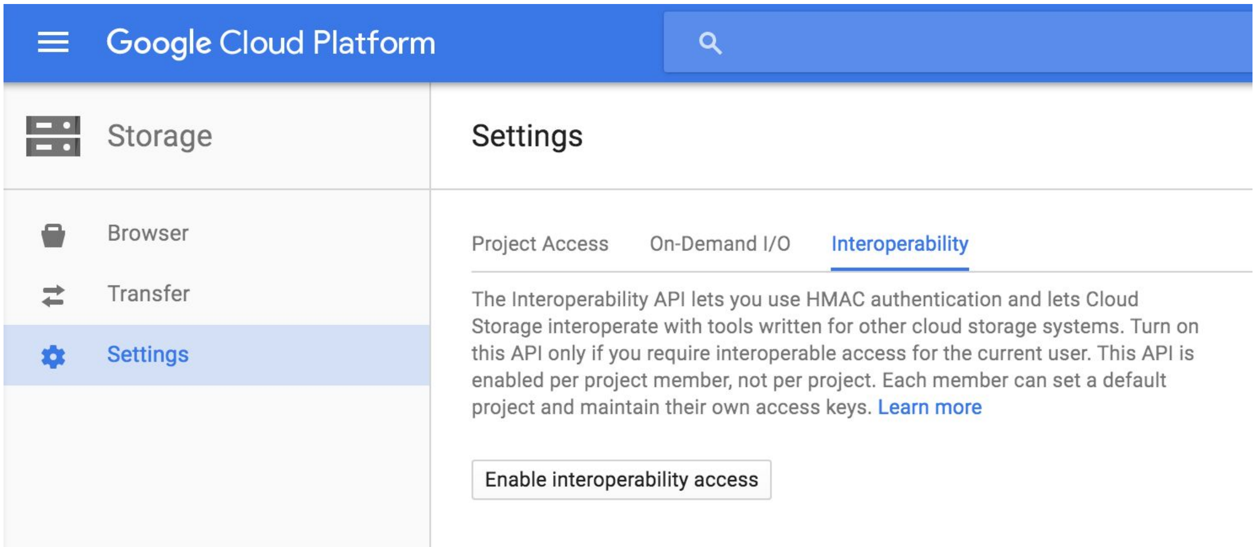
   ```
   #!/bin/sh
   sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
   iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
   ```

9. Under the HHS keys tab, add your SSH key. For example, copy and paste the contents of your `~/.ssh/id_rsa.pub` file.
10. Scroll to the bottom and click the **Create** button

Optional Installation: Create Core Filer for Google Cloud Storage Bucket

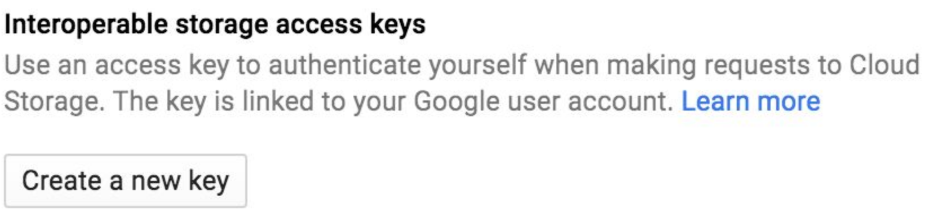
Enable S3 Compatibility with Google Storage and create create credentials

1. Within your project, use the left nav to go to Storage > Cloud Storage > Settings.
2. At the top, click the **Interoperability** tab.



The Interoperability API lets you use HMAC authentication and lets Cloud Storage interoperate with tools written for other cloud storage systems. Turn on this API only if you require interoperable access for the current user. This API is enabled per project member, not per project. Each member can set a default project and maintain their own access keys. Learn more

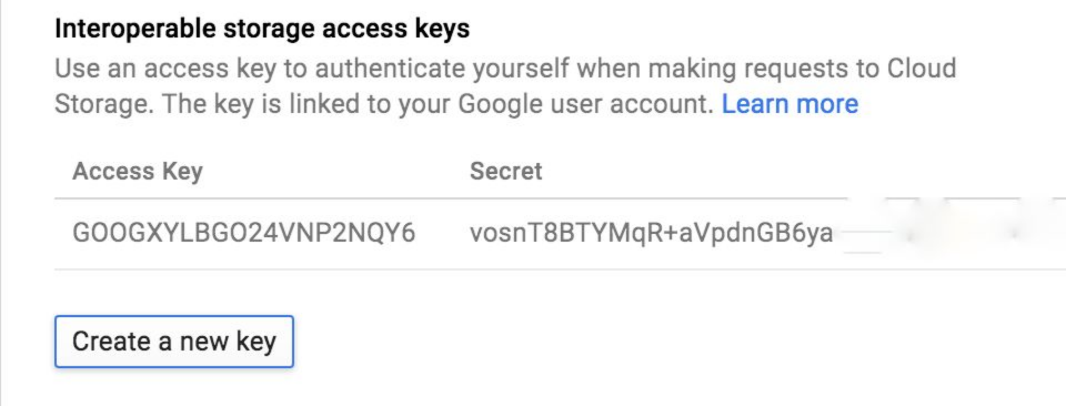
3. Click the **Enable interoperability access** button.
4. Click the **Create a New Key** button.



Interoperable storage access keys

Use an access key to authenticate yourself when making requests to Cloud Storage. The key is linked to your Google user account. Learn more

5. **Keep this tab open** to copy/paste keys.



Interoperable storage access keys

Use an access key to authenticate yourself when making requests to Cloud Storage. The key is linked to your Google user account. Learn more

Access Key	Secret
GOOGXYLBGO24VNP2NQY6	vosnT8BTYMqR+aVpdnGB6ya

Create bucket within Google

1. Within your project, Storage > Cloud Storage > Browser > Create Bucket button.
2. Provide unique name and leave storage class at Standard. This bucket name will be needed later.
3. Click Create button.

Add credentials to FXT

1. In new tab, log in the to FXT.
2. Settings tab > Cluster section on left > Cloud Credentials > Add Credential button.
3. Provide name, copy and paste keys from Google page.
4. Click Submit button.

Add cloud core filer

1. In FXT, Settings tab > Core Filer section on left > Manage Core Filers > Create button.
2. Choose Cloud and provide name.
3. Next.
4. Change storage type to Google Cloud Storage (S3).
5. Select the cloud credential.
6. Next.
7. Provide unique bucket name.
8. Click Add Filer button.