

GCP Administration Guide

FortiOS 7.6



FORTINET DOCUMENT LIBRARY

<https://docs.fortinet.com>

FORTINET VIDEO LIBRARY

<https://video.fortinet.com>

FORTINET BLOG

<https://blog.fortinet.com>

CUSTOMER SERVICE & SUPPORT

<https://support.fortinet.com>

FORTINET TRAINING & CERTIFICATION PROGRAM

<https://www.fortinet.com/training-certification>

FORTINET TRAINING INSTITUTE

<https://training.fortinet.com>

FORTIGUARD LABS

<https://www.fortiguard.com>

END USER LICENSE AGREEMENT

<https://www.fortinet.com/doc/legal/EULA.pdf>

FEEDBACK

Email: techdoc@fortinet.com



November 28, 2024

FortiOS 7.6 GCP Administration Guide

01-76-1054271-20241128

TABLE OF CONTENTS

About FortiGate-VM for GCP	6
Machine type support	6
Upgrading or downgrading a GCP instance to another machine type	10
Models	11
Licensing	13
Order types	13
Creating a support account	13
Migrating a FortiGate-VM instance between license types	15
Obtaining FortiCare-generated license and certificates for GCP PAYG instances	15
Obtaining FortiGate-VM image for GCP	17
Finding public FortiGate images	17
Using image family	18
Using private images	19
Copied private images	19
Single FortiGate-VM deployment	21
Google Cloud Marketplace	22
Initially deploying the FortiGate-VM	22
Registering and downloading your license	27
Connecting to the FortiGate-VM	27
Google Cloud Compute Engine	29
Obtaining the deployment image	29
Uploading the FortiGate deployment image to Google Cloud	29
Creating the FortiGate deployment image	30
Deploying the FortiGate-VM instance	32
Connecting to the FortiGate-VM	36
Configuring Google Cloud firewall rules	40
Configuring the second NIC on the FortiGate-VM	42
Configuring static routing in FortiGate-VM	43
Confidential computing	46
Google Cloud SDK	47
Obtaining the deployment image	47
Uploading the deployment image to Google Cloud	47
Creating a FortiGate custom image	47
Deploying a FortiGate-VM instance	48
Bootstrapping FortiGate at initial bootup	50
Deploying a FortiGate-VM on a T2A instance	51
Deploying a Shielded VM FortiGate-VM	53
Confidential computing	55
Deploying FortiGate-VM using Terraform	57
HA for FortiGate-VM on GCP	58
Deploying FortiGate-VM HA with SDN connector	58
Checking the prerequisites	60
Creating VPC networks and firewall rules	61
Deploying the primary FortiGate	62

Deploying the secondary FortiGate	63
Creating a GCP route table	65
Uploading the license and configuring network interfaces	66
Testing and troubleshooting	66
Protocol forwarding rule with SDN connector	69
Creating a target instance for each FortiGate-VM	69
Configuring the FortiGates	70
Testing the route and forwarding rule failover	71
Deploying FortiGate-VM HA with external and internal LB (web console)	73
FGCP in public cloud	73
Predeployment steps	73
Deploying FortiGate-VM instances	74
Reserving static addresses	75
Creating instance groups	76
Creating the external LB	76
Creating the internal LB	77
Creating a custom route	78
Configure FortiGates networking	78
Configuring FortiGate clustering	79
Configure health check probe responders	80
Best practices and next steps	81
Deploying FortiGate-VM HA with external and internal LB (GCloud CLI)	81
FGCP in public cloud	81
Predeployment steps	82
Reserving internal addresses	83
Deploying FortiGate-VM instances	83
Creating instance groups	84
Creating the external LB	84
Creating the internal LB and custom route	85
Configure FortiGates networking	86
Configuring FortiGate clustering	87
Configure health check probe responders	88
Best practices and next steps	89
Additional documentation	89
GCP IPv6 route support with nextHopInstance	90
SDN connector integration with GCP	91
Configuring GCP SDN connector using metadata IAM	91
GCP Kubernetes SDN connector	93
Configuring GCP SDN Connector using service account	93
Custom role permission guideline	94
API calls	94
Multiple GCP projects in a single SDN connector	95
Creating a GCP service account	98
Troubleshooting GCP SDN Connector	103
Pipelined automation using Google Cloud function	104
Deploying autoscaling on GCP	105
Requirements	105

Account permissions	105
Region requirements	105
Deploying FortiGate autoscale for GCP	106
Quotas	107
Terraform variables	108
Deployment information	111
Verifying the deployment	111
Verifying the instance group	113
Cluster monitoring	114
Adding instances to the protected subnet	115
Destroying the cluster	119
Troubleshooting	120
Debugging cloud-init	120
How to reset the elected primary FortiGate	121
Appendix	121
FortiGate Autoscale for GCP features	121
Architectural diagram	123
VPN for FortiGate-VM on GCP	124
Site-to-site IPsec VPNs between HA VPN on GCP	124
Packet mirroring	125
Creating VPC networks	125
Launching the FortiGate-VM instance	126
Creating an unmanaged instance group and load balancer	127
Configuring bidirectional VPC peering	128
Creating the packet mirroring policy	128
Verifying the configuration	129
Organization restrictions	130
SD-WAN transit routing with Google Network Connectivity Center	135
Prerequisites	135
Script execution for a single spoke	136
Configuring site-to-site VPN	139
Configuring the tunnel interfaces	140
Configuring BGP neighbors	141
Enabling dynamic routing mode	142
Completing post-deployment configuration	143
Deploying multiple spokes	144
Deploying resources in spoke VPC	144
Validating the configuration	145
Verifying site-to-site connectivity	146
GDC Virtual support	148
Enhancing FortiGate-VM performance	157
Deploying a gVNIC interface	157
DPDK support	158
Change log	167

About FortiGate-VM for GCP

By combining stateful inspection with a comprehensive suite of powerful security features, FortiGate next generation firewall technology delivers complete content and network protection. This solution is available for deployment on Google Cloud Platform (GCP).

There are several ways to deploy FortiGate-VM on GCP:

Deployment method	Description
Google Cloud Marketplace	See Google Cloud Marketplace on page 22 .
Google Cloud Compute Engine	Deploy a FortiGate-VM instance on Google Cloud Compute Engine from the custom image without using the Google Cloud Platform marketplace. See Google Cloud Compute Engine on page 29 . You must deploy FortiGate in this method when: <ul style="list-style-type: none">• FortiGate is required to be deployed inline across multiple networks and multiple network interfaces must be assigned to the instance. Google Cloud also does not allow changing the number of network interfaces after deploying VM instances.• You do not want to use the Google marketplace launcher. For example, you may want to use this deployment method if your organization does not allow you to browse marketplace websites in its IT policy.
Google Cloud SDK	Deploy a FortiGate-VM (BYOL) instance by using the Google Cloud SDK on your local PC. This is a method of deploying FortiGate-VM on GCP outside of the marketplace product listing and without creating an instance on the Google Cloud Compute Portal. This method also allows assigning multiple network interfaces to the VM instance. See Google Cloud SDK on page 47 .
Terraform	See Deploying FortiGate-VM using Terraform on page 57 .

Machine type support

You can deploy FortiGate for Google Cloud as a VM instance. Supported machine types may change without notice. FortiGate supports standard, high-memory, and high-CPU machine types with minimum 1 vCPU and 3.75 GB of RAM and maximum 96 vCPUs and 624 GB of RAM in the predefined machine type lineup. You can also customize the vCPU and RAM size combination within this range. See [Machine families resource and comparison guide](#).

FortiOS supports hot-adding vCPU and RAM. However, GCP may not support this. See [Edit the machine type of a compute instance](#).

Having at least 4 GB of RAM for proper FortiGate-VM operation is recommended, especially if unified threat management is enabled.

The following table summarizes a subset of machine type support for x64 instances. For a full list of supported machine types, see the [FortiGate Next-Generation Firewall \(BYOL\)](#). FortiOS 7.6.1 and later versions support the C4 instance type family.

Instance category	Instance type	vCPU	FortiGate minimum order (BYOL) to consume all instance CPU
General purpose	n1-standard-1	1	FG-VM01 or FG-VM01v
General purpose	n1-standard-2	2	FG-VM02 or FG-VM02v
General purpose	n1-standard-4	4	FG-VM04 or FG-VM04v
General purpose	n1-standard-8	8	FG-VM08 or FG-VM08v
General purpose	n1-standard-16	16	FG-VM16 or FG-VM16v
General purpose	n1-standard-32	32	FG-VMUL or FG-VMULv
General purpose	n2d-standard-1	1	FG-VM01 or FG-VM01v
General purpose	n2d-standard-2	2	FG-VM02 or FG-VM02v
General purpose	n2d-standard-4	4	FG-VM04 or FG-VM04v
General purpose	n2d-standard-8	8	FG-VM08 or FG-VM08v
General purpose	n2d-standard-16	16	FG-VM16 or FG-VM16v
General purpose	n2d-standard-32	32	FG-VMUL or FG-VMULv
General purpose	n2d-standard-1	1	FG-VM01 or FG-VM01v
General purpose	n2d-standard-2	2	FG-VM02 or FG-VM02v
General purpose	n2d-standard-4	4	FG-VM04 or FG-VM04v
General purpose	n2d-standard-8	8	FG-VM08 or FG-VM08v

Instance category	Instance type	vCPU	FortiGate minimum order (BYOL) to consume all instance CPU
General purpose	n2d-standard-16	16	FG-VM16 or FG-VM16v
General purpose	n2d-standard-32	32	FG-VMUL or FG-VMULv
General purpose	c3-highcpu-4	4	FG-VM04 or FG-VM04v
General purpose	c3-highcpu-8	8	FG-VM08 or FG-VM08v
General purpose	c3-highcpu-22	22	FG-VMUL or FG-VMULv
General purpose	c3-highcpu-44	44	FG-VMUL or FG-VMULv
General purpose	C3d-highcpu-4	4	FG-VM04 or FG-VM04v
General purpose	C3d-highcpu-8	8	FG-VM08 or FG-VM08v
General purpose	C3d-highcpu-16	16	FG-VM16 or FG-VM16v
General purpose	C3d-highcpu-30	30	FG-VMUL or FG-VMULv
General purpose	C3d-standard-4	4	FG-VM04 or FG-VM04v
General purpose	C3d-standard-8	8	FG-VM08 or FG-VM08v
General purpose	C3d-standard-16	16	FG-VM16 or FG-VM16v
General purpose	C3d-standard-30	30	FG-VMUL or FG-VMULv
General purpose	c4-standard-2	2	FG-VM02 or FG-VM02v
General purpose	c4-standard-4	4	FG-VM04 or FG-VM04v
General purpose	c4-standard-8	8	FG-VM08 or FG-VM08v
General purpose	c4-standard-16	16	FG-VM16 or FG-VM16v

Instance category	Instance type	vCPU	FortiGate minimum order (BYOL) to consume all instance CPU
General purpose	c4-standard-32	32	FG-VMUL or FG-VMULv
General purpose	c4-highcpu-2	2	FG-VM02 or FG-VM02v
General purpose	c4-highcpu-4	4	FG-VM04 or FG-VM04v
General purpose	c4-highcpu-8	8	FG-VM08 or FG-VM08v
General purpose	c4-highcpu-16	16	FG-VM16 or FG-VM16v
General purpose	c4-highcpu-32	32	FG-VMUL or FG-VMULv
General purpose	c4a-standard-2	2	FG-VM02 or FG-VM02v
General purpose	c4a-standard-4	4	FG-VM04 or FG-VM04v
General purpose	c4a-standard-8	8	FG-VM08 or FG-VM08v
General purpose	c4a-standard-16	16	FG-VM16 or FG-VM16v
General purpose	c4a-standard-32	32	FG-VMUL or FG-VMULv
General purpose	c4a-highcpu-2	2	FG-VM02 or FG-VM02v
General purpose	c4a-highcpu-4	4	FG-VM04 or FG-VM04v
General purpose	c4a-highcpu-8	8	FG-VM08 or FG-VM08v
General purpose	c4a-highcpu-16	16	FG-VM16 or FG-VM16v
General purpose	c4a-highcpu-32	32	FG-VMUL or FG-VMULv

The following table summarizes a subset of machine type support for ARM instances. FortiOS supports T2A (ARM). GCP marketplace deployments currently do not support T2A. T2A requires a manual CLI deployment. See [Deploying a FortiGate-VM on a T2A instance on page 51](#).

	Instance type	vCPU	Max NIC (enabled by GCP)	FortiGate minimum order (BYOL) to consume all instance CPU
General purpose	t2a-standard-1	1	2	FG-VM01 or FG-VM01v
	t2a-standard-2	2	4	FG-VM02 or FG-VM02v
	t2a-standard-4	4	4	FG-VM04 or FG-VM04v
	t2a-standard-8	8	8	FG-VM08 or FG-VM08v
	t2a-standard-16	16	8	FG-VM16 or FG-VM16v
	t2a-standard-32	32	8	FG-VMUL or FG-VMULv

For information about GCP machine types, see the following:

- [Predefined Machine Types](#)
- [Machine Resources](#)
- [Use Multiple Interfaces](#)

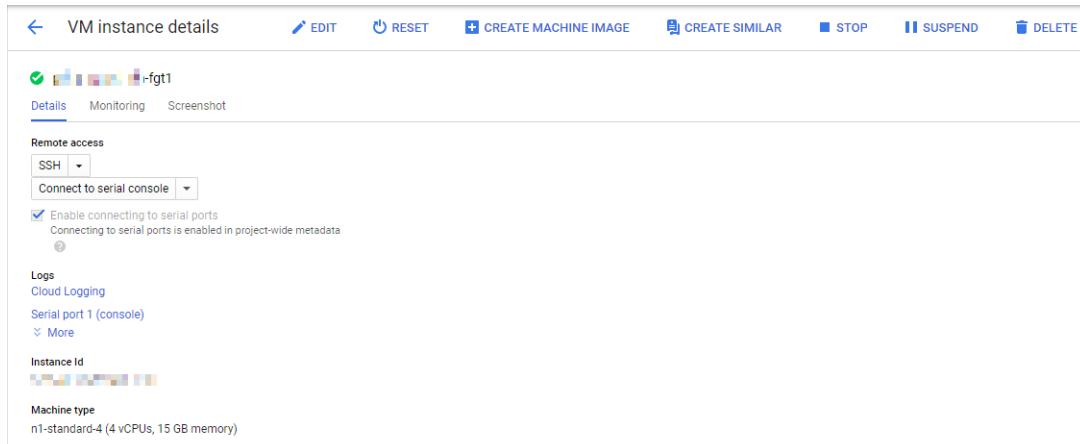
Upgrading or downgrading a GCP instance to another machine type

With FortiGate-VM bring your own license (BYOL) instances, you must source appropriate licenses to support the change in machine types for FortiGate-VM BYOL instances and add the licenses manually. You may have to add a new license to correspond to the new processor core count. See [How to upgrade FortiGate VM license](#).

Editing the instance configuration does not allow you to add or delete network interfaces.

To upgrade or downgrade a GCP instance to another machine type:

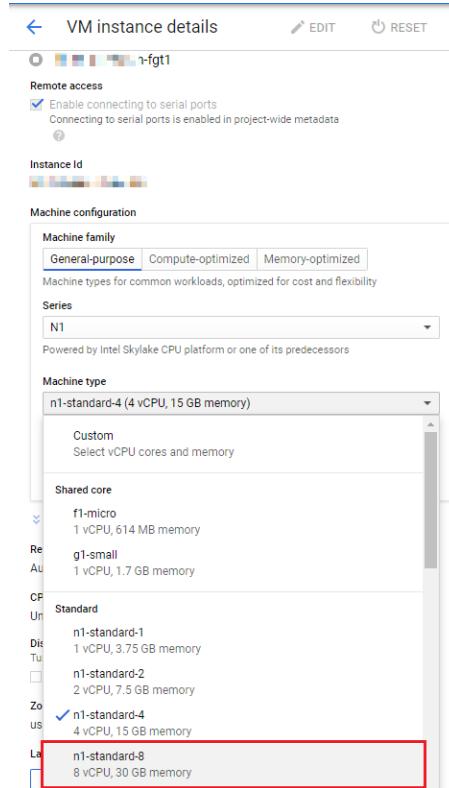
1. Go to *Compute Engine > Instances*.
2. Select the desired instance.
3. On the *VM instance details* page, click *STOP* to shut down the VM. In this example, the original machine type is n1-standard-4.



4. Once the VM powers off, click *EDIT* on the VM instance details page.

About FortiGate-VM for GCP

- From the *Series* and *Machine type* dropdown lists, select the desired new series and machine type. This example upgrades the VM instance from n1-standard-4 to n1-standard-8.



- Click **Save**.
- Select the VM and click **START/RESUME**.

Models

FortiGate-VM is available with different CPU and RAM sizes and can be deployed on various private and public cloud platforms. The following table shows the models conventionally available to order, also known as bring your own license (BYOL) models. See [Order types on page 13](#).

Model name	vCPU	
	Minimum	Maximum
FG-VM01/01v/01s	1	1
FG-VM02/02v/02s	1	2
FG-VM04/04v/04s	1	4
FG-VM08/08v/08s	1	8
FG-VM16/16v/16s	1	16
FG-VM32/32v/32s	1	32

Model name	vCPU	
	Minimum	Maximum
FG-VMUL/ULv/ULs	1	Unlimited



With the changes in the FortiGuard extended IPS database introduced in FortiOS 7.4.0, some workloads that depend on the extended IPS database must have the underlying VM resized to 8 vCPU or more to continue using the extended IPS database.

See [Support full extended IPS database for FortiGate VMs with eight cores or more](#).

For information about changing the instance type on an existing VM, see [Edit the machine type of a VM instance](#).

For more information about GCP Compute instances, see [Compute-optimized machine family for Compute Engine](#).



The v-series and s-series do not support virtual domains (VDOMs) by default. To add VDOMs, you must separately purchase perpetual VDOM addition licenses. You can add and stack VDOMs up to the maximum supported number after initial deployment.

Generally there are RAM size restrictions to FortiGate BYOL licenses. However, these restrictions are not applicable to GCP deployments. Any RAM size with certain CPU models are allowed. Licenses are based on the number of CPUs only.

Previously, platform-specific models such as FortiGate for GCP with a GCP-specific orderable menu existed. However, the common model is now applicable to all supported platforms.

For information about each model's order information, capacity limits, and adding VDOM, see the [FortiGate-VM datasheet](#).

The primary requirement for the provisioning of a virtual FortiGate may be the number of interfaces it can accommodate rather than its processing capabilities. In some cloud environments, the options with a high number of interfaces tend to have high numbers of vCPUs.

The licensing for FortiGate-VM does not restrict whether the FortiGate can work on a VM instance in a public cloud that uses more vCPUs than the license allows. The number of vCPUs indicated by the license does not restrict the FortiGate from working, regardless of how many vCPUs are included in the virtual instance. However, only the licensed number of vCPUs process traffic and management. The rest of the vCPUs are unused.

License	1 vCPU	2 vCPU	4 vCPU	8 vCPU	16 vCPU	32 vCPU
FGT-VM08	OK	OK	OK	OK	8 vCPUs used for traffic and management. The rest are not used.	8 vCPUs used for traffic and management. The rest are not used.

You can provision a VM instance based on the number of interfaces you need and license the FortiGate-VM for only the processors you need.

Licensing

You must have a license to deploy FortiGate for GCP. The following sections provide information on licensing FortiGate for GCP:

- [Order types on page 13](#)
- [Creating a support account on page 13](#)
- [Migrating a FortiGate-VM instance between license types on page 15](#)
- [Obtaining FortiCare-generated license and certificates for GCP PAYG instances on page 15](#)

Order types

On GCP, there are usually two order types: bring your own license (BYOL) and pay as you go (PAYG).

BYOL offers perpetual (normal series and v-series) and annual subscription (s-series) licensing as opposed to PAYG, which is an hourly subscription available with marketplace-listed products. BYOL licenses are available for purchase from resellers or your distributors, and the publicly available price list, which Fortinet updates quarterly, lists prices. BYOL licensing provides the same ordering practice across all private and public clouds, no matter what the platform is. You must activate a license for the first time you access the instance from the GUI or CLI before you can start using various features.

With a PAYG subscription, the FortiGate-VM becomes available for use immediately after the instance is created. The marketplace product page mentions term-based prices (hourly or annual).

In both BYOL and PAYG, cloud vendors charge separately for resource consumption on computing instances, storage, and so on, without use of software running on top of it (in this case FortiGate).

For BYOL, you typically order a combination of products and services including support entitlement. S-series SKUs contain the VM base and service bundle entitlements for easier ordering. PAYG includes support, for which you must contact Fortinet Support with your customer information.

To purchase PAYG, all you need to do is subscribe to the product on the marketplace. However, you must contact Fortinet Support with your customer information to obtain support entitlement. See [Creating a support account on page 13](#).



PAYG FortiGate instances do not support the use of virtual domains (VDOMs). If you plan to use VDOMs, deploy BYOL instances instead.



PAYG and BYOL licensing and payment models are not interchangeable. For example, once you spin up a FortiGate-VM PAYG instance, you cannot inject a BYOL license on the same VM. Likewise, you cannot convert a FortiGate-VM BYOL instance to PAYG.

Creating a support account

FortiGate for GCP supports pay as you go (PAYG) and bring your own license (BYOL) licensing models. See [Order types on page 13](#).

To make use of Fortinet technical support and ensure products function properly, you must complete certain steps to activate your entitlement. Our support team can identify your registration in the system thereafter.

First, if you do not have a Fortinet account, you can create one at [Customer Service & Support](#).

BYOL

You must obtain a license to activate the FortiGate. If you have not activated the license, you see the license upload screen when you log into the FortiGate and cannot proceed to configure the FortiGate.

You can obtain licenses for the BYOL licensing model through any Fortinet partner. After you purchase a license or obtain an evaluation license, you receive a PDF with an activation code.

The FortiOS permanent trial license requires a FortiCare account. This trial license has limited features and capacity. The trial license only applies to BYOL deployments for FortiGate-VM on GCP. See [VM license](#) for details.

To activate a BYOL license:

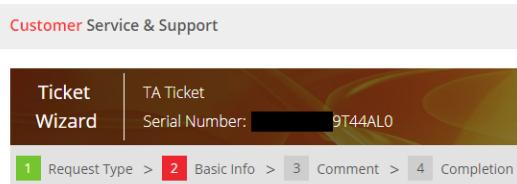
1. Go to [Customer Service & Support](#) and create a new account or log in with an existing account.
2. Go to *Asset > Register/Activate* to start the registration process.
3. In the *Registration* page, enter your license activation code, then select *Next* to continue registering the product.
4. If you register the s-series subscription model, the site prompts you to select one of the following:
 - Click *Register* to newly register the code to acquire a new serial number with a new license file.
 - Click *Renew* to renew and extend the licensed period on top of the existing serial number, so that all features on the VM node continue working uninterrupted upon license renewal.
5. At the end of the registration process, download the license (.lic) file to your computer. You upload this license later to activate the FortiGate-VM.

After registering a license, Fortinet servers may take up to 30 minutes to fully recognize the new license. When you upload the license (.lic) file to activate the FortiGate-VM, if you get an error that the license is invalid, wait 30 minutes and try again.

PAYG

To activate a PAYG license:

1. Deploy and boot the FortiGate PAYG VM and log into the FortiGate GUI management console.
2. From the Dashboard, copy the VM's serial number.
3. Go to [Customer Service & Support](#) and create a new account or log in with an existing account.
4. Go to *Asset > Register/Activate* to start the registration process.
5. In the *Registration* page, enter the serial number, and select *Next* to continue registering the product. Enter your details in the other fields.
6. After completing registration, contact [Fortinet Customer Support](#) and provide your FortiGate instance's serial number and the email address associated with your Fortinet account.



Migrating a FortiGate-VM instance between license types

When deploying a FortiGate-VM on public cloud, you determine the license type (pay as you go (PAYG) or bring your own license (BYOL)) during deployment. The license type is fixed for the VM's lifetime. The image that you use to deploy the FortiGate-VM on the public cloud marketplace predetermines the license type.

Migrating a FortiGate-VM instance from one license type to another requires a new deployment. You cannot simply switch license types on the same VM instance. However, you can migrate the configuration between two VMs running as different license types. There are also FortiOS feature differences between PAYG and BYOL license types. For example, a FortiGate-VM PAYG instance is packaged with Unified Threat Management protection and does not support VDOMs, whereas a FortiGate-VM BYOL instance supports greater protection levels and features depending on its contract.

To migrate FortiOS configuration to a FortiGate-VM of another license type:

1. Connect to the FortiOS GUI or CLI and back up the configuration. See [Configuration backups](#).
2. Deploy a new FortiGate-VM instance with the desired license type. If deploying a BYOL instance, you must purchase a new license from a Fortinet reseller. You can apply the license after deployment via the FortiOS GUI or bootstrap the license and configuration during initial bootup using custom data as described in [Bootstrapping FortiGate at initial bootup on page 50](#).
3. Restore the configuration on the FortiGate-VM instance that you deployed in step 2. As with the license, you can inject the configuration during initial bootup. Alternatively, you can restore the configuration in the FortiOS GUI as described in [Configuration backups](#).
4. If you deployed a PAYG instance in step 2, register the license. To receive support for a PAYG license, you must register the license as described in [Creating a support account on page 13](#).

Obtaining FortiCare-generated license and certificates for GCP PAYG instances

GCP PAYG instances can obtain FortiCare-generated licenses upon a new deployment, or in the CLI (`execute vm-license`) when upgrading from previous firmware. The process generates Fortinet_Factory and Fortinet_Factory_Backup certificates that contain the common name (CN) of the FortiGate serial number to uniquely identify this FortiGate.

Installing a new deployment

A newly deployed instance automatically retrieves the signed certificate from FortiCare. Appropriately 30 seconds after booting the instance, it gets the certificate and reboot once to install the new certificate.

To verify the installation in a new deployment:

1. Enable debugging and check the update status:

```
# diagnose debug enable
# diagnose debug update -1
Debug messages will be on for 30 minutes.
VM license install succeeded. Rebooting firewall.
```

2. After the reboot, verify the license information:

```
# diagnose debug vm-print-license
SerialNumber: FGVM04TM*****
CreateDate: Tue Jun 8 02:30:19 2021
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Model: PG (22)
CPU: 2147483647
MEM: 2147483647
```

3. Verify the Fortinet_Factory certificate information (the CN is the serial number):

```
config vpn certificate local
    # get Fortinet_Factory
    name          : Fortinet_Factory
    password      : *
    private-key   : *
    certificate   :
        Subject: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU =
FortiGate, CN = FGVM04TM*****, emailAddress = support@fortinet.com
        Issuer:   C = US, ST = California, L = Sunnyvale, O = Fortinet, OU =
Certificate Authority, CN = fortinet-subca2001, emailAddress = support@fortinet.com
        Valid from: 2021-06-08 02:30:19 GMT
        Valid to:   2056-01-19 03:14:07 GMT
    ...
...
```

Upgrading the firmware

To obtain a FortiCare-generated license during an upgrade:

1. Before upgrading, verify the Fortinet_Factory certificate information (the CN is FortiGate):

```
config vpn certificate local
    # get Fortinet_Factory
    name          : Fortinet_Factory
    password      : *
    private-key   : *
    certificate   :
        Subject: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU =
FortiGate, CN = FortiGate, emailAddress = support@fortinet.com
        Issuer:   C = US, ST = California, L = Sunnyvale, O = Fortinet, OU =
Certificate Authority, CN = fortinet-subca2001, emailAddress = support@fortinet.com
        Valid from: 2016-11-30 19:58:17 GMT
        Valid to:   2056-11-20 19:58:07 GMT
    ...
...
```

2. Verify the license information:

```
# diagnose debug vm-print-license
SerialNumber: FGTMCGPH*****
CreateDate: 1623112103
Model: PG (22)
CPU: 2147483647
MEM: 2147483647
```

Since there is no unique certificate from FortiCare, there are no Key, Cert, Key2, or Cert2 fields.

3. Upgrade the firmware and update the license:

```
# execute vm-license
This operation will reboot the system !
Do you want to continue? (y/n)y

Get instance JWT token
Requesting FortiCare license: FGTMCGPH*****
VM license install succeeded. Rebooting firewall.
```

4. Verify the new Fortinet_Factory certificate information (the CN is the serial number):

```
config vpn certificate local
    # get Fortinet_Factory
    name          : Fortinet_Factory
    password      : *
    private-key   : *
    certificate   :
        Subject: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU =
FortiGate, CN = FGTMCGPH***** , emailAddress = support@fortinet.com
        Issuer:   C = US, ST = California, L = Sunnyvale, O = Fortinet, OU =
Certificate Authority, CN = fortinet-subca2001, emailAddress = support@fortinet.com
        Valid from: 2021-06-08 02:30:19 GMT
        Valid to:   2056-01-19 03:14:07 GMT
        ...
...
```

5. Verify the license information (Key, Cert, Key2, or Cert2 fields are now available):

```
# diagnose debug vm-print-license
SerialNumber: FGTMCGPH*****
CreateDate: Tue Jun 8 02:30:19 2021
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Model: PG (22)
CPU: 2147483647
MEM: 2147483647
```

Obtaining FortiGate-VM image for GCP

When deploying instances using gcloud or templating tools, you must provide a base VM image. You can deploy an official image published by Fortinet or create your own image with a disk image downloaded from the [Fortinet Support site](#). Using an official image is recommended unless you must deploy a custom image.

Finding public FortiGate images

Fortinet publishes official images in the fortigcp-project-001 project. This is a special public project. Any GCP user can list images available there using the following command: `gcloud compute images list --project fortigcp-project-001`

The official FortiGate image names start with `fortinet-fgt-[VERSION]` for bring your own license (BYOL) images or `fortinet-fgtondemand-[VERSION]` for pay as you go (PAYG) images. Selecting the correct image is your responsibility if you deploy FortiGate-VM using gcloud or templates. Some templates provided by Fortinet can automatically find the image name based on version and licenses properties.

You can use gcloud command filter and format options to get a clean list. For example, `gcloud compute images list --project fortigcp-project-001 --filter="name ~ fortinet-fgtondemand AND status:READY" --format="get(selfLink)"` retrieves a list of image URLs for FortiGate PAYG. `FGT_IMG=$(gcloud compute images list --project fortigcp-project-001 --filter="name ~ fortinet-fgt- AND status:READY" --format="get(selfLink)" | sort -r | head -1)` saves the URL of the newest BYOL image into the `FGT_IMG` variable.

When deploying a VM, you can reference the base image by:

- Image project and image name
- Image project and image family
- Image URL

Using image family

All newly published versions since the end of 2021 support the image family attribute. Using predictable image family names makes it easier to deploy the newest image of given product's major version, because you no longer need to list all available images to see what is available. Instead, you can simply say "deploy newest image of FortiGate 7.0". The image family name consists of [product name]-[major version without dot]-[licensing option (if available)]

The following lists image families available at the time of writing:

- fortigate-64-byol
- fortigate-64-payg
- fortimanager-70
- fortianalyzer-70
- fortigate-70-byol
- fortigate-70-payg
- fortigate-70 (Do not use this image family.)
- fortigate-72-byol
- fortigate-72-payg

Using image family with gcloud

```
gcloud compute instances create my-fortigate \
--machine-type=e2-micro \
--image-family=fortigate-64-byol --image-project=fortigcp-project-001 \
--can-ip-forward \
--network-interface="network=default"
```

Using image family with Terraform

```
data "google_compute_image" "fgt_image" {
  project = "fortigcp-project-001"
  family  = "fortigate-72-byol"
```

```
}

resource "google_compute_instance" "my_fortigate" {
  name          = "my-fortigate"
  machine_type = "e2-micro"
  can_ip_forward = true
  boot_disk {
    initialize_params {
      image = data.google_compute_image.fgt_image.self_link
    }
  }
  network_interface {
    access_config {
    }
  }
}
```

Using image family with Deployment Manager

```
resources:
- name: my-fortigate
  type: compute.v1.instance
  properties:
    zone: europe-west6-b
    machineType: zones/europe-west6-b/machineTypes/e2-micro
    disks:
      - deviceName: boot
        type: PERSISTENT
        boot: true
        autoDelete: true
        initializeParams:
          sourceImage: projects/fortigcp-project-001/global/images/family/fortigate-64-byol
          diskSizeGb: 10.0
    networkInterfaces:
      - network: global/networks/default
        accessConfigs:
          - name: External NAT
            type: ONE_TO_ONE_NAT
```

Using private images

In some cases, you may want to use a private image to deploy FortiGates. For example, you may need to deploy an interim version obtained from [Fortinet Support](#) or want to use special features, such as MULTI_IP_SUBNET. You can create a private image by copying a public one or by downloading one from the [Fortinet Support](#) site.

Copied private images

To create a private copy of a public FortiGate GCP image with additional flags, issuing a single gcloud command is sufficient:

```
gcloud compute images create fgt-72-multi-ip --source-image-family fortigate-72-byol --source-image-project fortigcp-project-001 --guest-os-features=MULTI_IP_SUBNET
```

You can download a VM image from the [Fortinet Support site](#). The following provides instructions for using one as a base image for creating a VM instance in GCP.

To use a downloaded image as a base image for VM instance creation in GCP:

1. Download the image from [Fortinet Support site](#). The file name should end with .tar.gz
2. In the GCP console, go to *Cloud Storage* and create a new bucket.
3. Click *Upload files* and upload the *.tar.gz file downloaded from Fortinet website.
4. Once uploaded, go to *Compute > Images* and click the *Create image* icon at the top.
5. Select *Cloud storage file* as the source.
6. Do one of the following:
 - a. Browse the cloud storage and select the file.
 - b. Issue the following command to provide the cloud storage path to the file: `gcloud compute images create fgt-private-image --source-uri=gs://STORAGE_BUCKET_NAME/FORTIGATE_IMAGE.out.gcp.tar.gz`

Single FortiGate-VM deployment

Google Cloud Marketplace

Initially deploying the FortiGate-VM

GCP supports Terraform packages via marketplace deployments. The following document outlines the steps to deploy FortiGate-VM bring your own license (BYOL) and pay as you go (PAYG) via the GCP marketplace via Terraform packages.



Deleting the FortiGate-VM instance after deployment does not delete the log disk. However, deleting the entire deployment from the *Solution Deployment* section deletes all resources that the deployment created, including the log disk.

Do not rerun a broken deployment as Terraform state lock is enabled by default and you cannot disable it from the marketplace GUI. Delete the broken deployment and create a brand new deployment instead.

These are the limitations in the Terraform GCP provider.

Preparing a service account

For information about creating a service account, see [Create service accounts](#).

Deploying a FortiGate-VM requires the following permissions and roles:

- roles/config.agent
- roles/compute.networkAdmin
- roles/compute.admin
- roles/iam.serviceAccountUser
- roles/storage.objectViewer

Deployment name *
doc-example-deployment

Deployment Service Account ?

- Existing account
 New account

List of available Service Accounts that have the following roles:

- roles/config.agent
- roles/compute.networkAdmin
- roles/compute.admin
- roles/iam.serviceAccountUser
- roles/storage.objectViewer

Select a Service Account
marketplace-deployment (marketplace-deployment@ftnt-marketplace-publish-pr... ▾)

To perform initial deployment of the FortiGate-VM:

1. In the Google Cloud marketplace Cloud Launcher, find FortiGate Next-Generation Firewall. Select BYOL or PAYG according to your needs.
2. Click *LAUNCH*.
3. Configure the variables as required:

Machine type

Machine types for common workloads, optimized for cost and flexibility

Series — N2

Powered by Intel Cascade Lake and Ice Lake CPU platforms

Machine type — n2-standard-4 (4 vCPU, 2 core, 16 GB memory)

	vCPU 4	Memory 16 GB
---	-----------	-----------------

Custom Service Account



The service account in the *Custom Service Account* field is used for running the FortiGate-VM. Metadata Identity & Access Management (IAM) SDN connectors use this service account to interact with GCP APIs. For information about metadata IAM SDN Connectors, see [Configuring GCP SDN connector using metadata IAM on page 91](#). To use a custom service, use the email address of the service account, similar to <serviceaccount-name>@<project>.iam.gserviceaccount.com, in this field.

Boot Disk

Boot disk size in GB — 10

Boot disk type — SSD Persistent Disk

Log Disk

Enable Log Disk 

Log disk size in GB — 30

Log disk type — SSD Persistent Disk

See [Deployment variables](#) for descriptions of the deployment variables:

4. Add more networks and network interfaces if desired:

- a. Under *Network interfaces*, click **ADD NETWORK INTERFACE**.
- b. Select the desired network and subnetwork, then click **DONE**.

Networking

Network interfaces

unprotected-public unprotected-public-subnet (10.0.1.0/24) ▾

protected-private protected-private-subnet (10.0.2.0/24) ▾

Network interface ^

Networks in this project
 Networks shared with me (from host project: shared-vpc-project-301520)

Network ha-sync ▾ ?

Subnetwork ha-sync-subnet ▾ ?

External IP None ▾ ?

CANCEL DONE

ADD NETWORK INTERFACE

Networking

Network interfaces

unprotected-public unprotected-public-subnet (10.0.1.0/24)	▼
protected-private protected-private-subnet (10.0.2.0/24)	▼
ha-sync ha-sync-subnet (10.0.3.0/24)	▼
ha-mgmt ha-mgmt-subnet (10.0.4.0/24)	▼
ADD NETWORK INTERFACE	

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet



Creating certain firewall rules may expose your instance to the Internet.
Please check if the rules you are creating are aligned with your security preferences. [Learn more](#)

Allow TCP port 22 traffic

Source IP ranges for TCP port 22 traffic ?

Allow HTTPS traffic

Source IP ranges for HTTPS traffic ?

Allow HTTP traffic

Source IP ranges for HTTP traffic ?

Allow TCP port 541 traffic

Source IP ranges for TCP port 541 traffic ?

Allow TCP port 3000 traffic

Source IP ranges for TCP port 3000 traffic ?

Allow TCP port 8080 traffic

Source IP ranges for TCP port 8080 traffic ?



This example adds the HA-Sync and HA-Mgmt networks to NIC 3 and NIC 4 respectively to illustrate multiple network support. If you are not configuring high availability, you can select other networks for any NIC on the FortiGate deployment.



Google Cloud instances support a maximum of eight interfaces, based on the selected VM type.

- Click *Deploy*. When deployment is done, select *DETAILS* to review the temporary password and public IP address to access the FortiGate-VM.

◀ FortiGate Next-Generation Firewall (BYOL)

✓ doc-deployment-example

FortiGate Next-Generation Firewall (BYOL)

RESOURCES

DETAILS

Basic information

Deployment name	doc-deployment-example
Location	us-central1-b us-central1
Deployment date	Jun 6, 2024, 2:45:39 PM
Deployed from	Marketplace
Labels	goog-cloud... : true goog-cloud... : fc541fff-7...

Outputs

Admin Password
Admin User	admin
Arm64 Image	false
Confidential Vm Enabled	false
Has External Ip	true
Instance Machine Type	n2-standard-4
Instance Nat Ip	34.123.94.246
Instance Network	https://www.googleapis.com/compute/v1/projects/
Instance Self Link	https://www.googleapis.com/compute/v1/projects/
Instance Zone	us-central1-b
Shielded Vm Enabled	true

Deployment variables

Variable	Description
Deployment name	Enter the FortiGate-VM name to appear in the Compute Engine portal.
Deployment Service Account	Select <i>Existing account</i> .

Variable	Description
Select a Service Account	Autopopulated with service accounts that have the needed roles and permissions assigned.
Image Version	Select the FortiGate version. The latest version is the default.
Zone	Choose the zone to deploy the FortiGate to.
Enable Confidential VM	Enable the confidential VM feature. FortiOS 7.6 only supports this feature with an n2d machine family instance type x64 image.
Enable Shielded VM	Enable the shielded VM feature. FortiOS 7.6 supports this feature with any UEFI-based VMs with the x64 image.
Machine type	Choose the series and instance type required.
Custom Service Account	The service account in the <i>Custom Service Account</i> field is used for running the FortiGate-VM. Metadata Identity & Access Management (IAM) SDN connectors use this service account to interact with GCP APIs. For information about metadata IAM SDN Connectors, see Configuring GCP SDN connector using metadata IAM on page 91 . To use a custom service, use the email address of the service account, similar to <serviceaccount-name>@<project>.iam.gserviceaccount.com, in this field.
Boot disk size in GB	Leave as-is at 10 GB.
Boot disk type	Choose the desired boot disk type.
Enable Log Disk	Enable log disk.
Log disk size in GB	Select the desired log disk size or leave as-is at 30 GB.
Log disk type	Select the desired log disk type.
Network	Select the network located in the selected zone.
Subnetwork	Select the subnetwork where the FortiGate resides.
Enable IP Forward	Enable the VM to forward packets.
Firewall	Leave all selected as shown, or allow at least HTTPS if the strictest security is allowed in your network as the first setup. Change firewall settings as needed later on. These are the open ports allowed in Google Cloud to protect incoming access to the FortiGate instance over the Internet and are not part of FortiGate firewall features.
External IP	Select <i>Ephemeral</i> . You must access the FortiOS GUI via this public IP address.

Registering and downloading your license

Follow the instructions that BYOL on page 14 details, then continue to [Connecting to the FortiGate-VM on page 27](#).

Connecting to the FortiGate-VM

To connect to the FortiGate-VM, you need your login credentials and the FortiGate-VM's public DNS address. From the previous step, there is a temporary admin password that Google Cloud automatically generates.

To connect to the FortiGate-VM:

1. Connect to the FortiGate using your browser. Your browser displays a certificate error message, which is normal because browsers do not recognize the default self-signed FortiGate certificate. Proceed past this error.
2. If accessing the FortiGate for the first time via the GUI (HTTPS, port 443) or SSH (port 22), you may see a disclaimer. Click *Accept*.
3. Log in to the FortiGate-VM with the username *admin* and the supplied temporary password.
4. Change the password.
5. After logging in successfully, upload your license (.lic) file to activate the FortiGate-VM. The FortiGate-VM automatically restarts. After it restarts, wait about 30 minutes until the license is fully registered at Fortinet, then log in again.
6. After you log in, you see the FortiGate dashboard. The information in the dashboard varies depending on the instance type.

The screenshot shows the FortiGate-VM dashboard with the following sections:

- System Information:**
 - Hostname: docsdemo
 - Serial Number: FGVM02 [REDACTED]
 - Firmware: v7.0.0 build0059 (Interim)
 - Mode: NAT
 - System Time: 2021/03/24 14:02:47
 - Uptime: 00:02:13:39
 - WAN IP: 35.247.56.211
- Licenses:** (209.222.147.39)
 - FortiCare Support
 - Firmware & General Updates
 - IPS
 - AntiVirus
 - Web Filtering
 - FortiToken 0/2
- Virtual Machine:**
 - FGVM02 License
 - Allocated vCPUs: 2/2 (100%)
 - Allocated RAM: 2 GiB
 - Auto Scaling: Disabled
- Administrators:**
 - 2 HTTPS 1 HTTP 0 FortiExplorer
 - aharris super_admin
 - faz super_admin
- Security Fabric: fabric**
 - docsdemo (Fabric Root)
 - Branch_Office_01
 - S108DVCHTPD-GG54
 - Branch_Office_02

Google Cloud Compute Engine

Obtaining the deployment image

To obtain the deployment image:

1. Go to the [Fortinet support site](#) and log in.
2. Go to *Support > VM Images*.
3. From the *Select Product* dropdown list, select *FortiGate*.
4. From the *Select Platform* dropdown list, select *Google*.
5. Download the deployment package file. The deployment package file is named “FGT_VM64_GCP-vX-buildXXXX-FORTINET.out.gcp.tar.gz”, where vX is the major version number and XXXX is the build number.



This deployment method only applies for bring your own licensing.

Uploading the FortiGate deployment image to Google Cloud

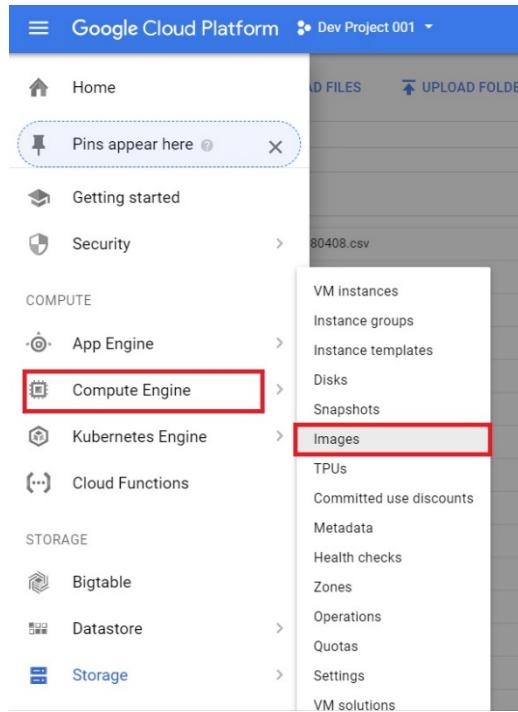
To upload the FortiGate deployment image to Google Cloud:

1. Log into Google Cloud.
2. Go to *Storage > Browser*.
3. Create a new bucket or go to an existing bucket.
4. Upload the newly downloaded deployment file.

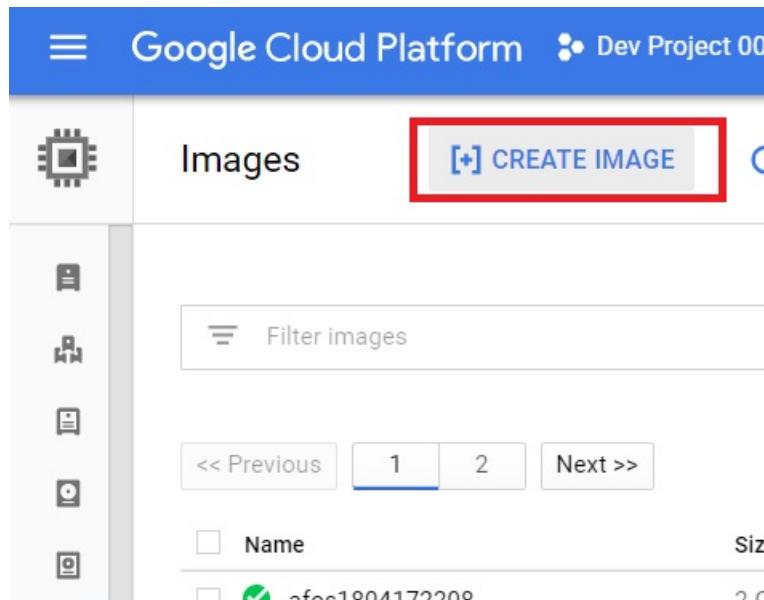
Creating the FortiGate deployment image

To create the FortiGate deployment image:

1. Go to *Compute Engine > Images*.



2. Click *CREATE IMAGE*.



3. On the *Create an image* page, enter the desired name. Under *Source*, select *Cloud Storage file*, then browse to the location of the deployment image file. Click *Create*.

Google Cloud Platform Dev Project 001

Create an image

Name

Family (Optional)

Description (Optional)

Encryption

Source

Cloud Storage file Your image source must use the .tar.gz extension and the file inside the archive must be named disk.raw. [Learn more](#)

Equivalent REST or command line

The image is listed on the *Images* pane.

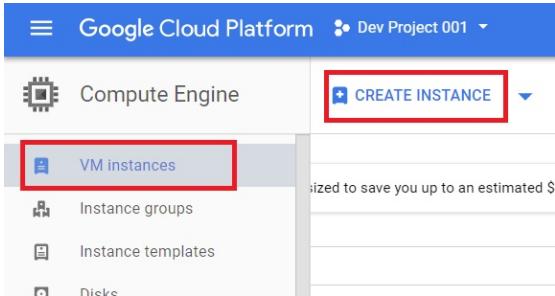
Google Cloud Platform Dev Project 001

	Images	[+ CREATE IMAGE]	REFRESH	CREATE INSTANCE	DEPRECATE	
<input type="checkbox"/>	<input checked="" type="checkbox"/> fortigatejkatoimage001	2 GB	Dev Project 001	Apr 20, 2018, 1:14:11 PM		
<input type="checkbox"/>	<input checked="" type="checkbox"/> fortinettechfortinet	2 GB	Dev Project 001	Feb 20, 2018, 11:02:51 AM		

Deploying the FortiGate-VM instance

To deploy the FortiGate-VM instance:

1. Go to *Compute Engine > VM Instances*. Click *CREATE INSTANCE*.



2. Configure the instance:

- a. In the *Name* field, enter the desired name. Select the desired zone and machine type.

A screenshot of the "Create an instance" configuration page. On the left, there's a sidebar with various icons. The main form area starts with a "Name" field containing "jkato-fgt-563-test007". Below it is a "Zone" dropdown set to "us-west1-c". Under "Machine type", there are dropdowns for "2 vCPUs" and "7.5 GB memory", with a "Customize" link. The "Container" section has a checkbox for deploying a container image, which is unchecked. The "Boot disk" section shows a "New 10 GB standard persistent disk" attached to "Debian GNU/Linux 9 (stretch)". There's a "Change" button next to the disk settings. The "Identity and API access" section includes a "Service account" dropdown set to "Compute Engine default service account".

- b. Under *Boot disk*, click *Change*.
- c. On the *Custom images* tab, select the newly created image. Change the boot disk type as needed, and enter 10 for the *Size*. Click *Select*.

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk

OS images Application images **Custom images** Snapshots Existing disks

Created from Dev Project 001 on Apr 12, 2018, 4:06:48 PM

fortigatejkatoimage001

Created from Dev Project 001 on Apr 20, 2018, 1:14:11 PM

fortinet-technote-001

Created from Dev Project 001 on Feb 20, 2018, 11:02:51 AM

fortipaq-golden

Created from Dev Project 001 on Jan 11, 2018, 3:29:40 PM

fortipoq-golden2

FortiBox Golden 2

Created from Dev Project 001 on Jan 29, 2018, 5:38:28 PM

fortipoq-golden3

FortiBox Golden 3

Created from Dev Project 001 on Mar 14, 2018, 10:17:40 AM

fortiooc-jordan

Created from Dev Project 001 on Feb 7, 2018, 7:1:58:20 AM

fos1533_1711142148

Created from Dev Project 001 on Nov 14, 2017, 1:43:17 PM

fwfb-brot-591

Created from Dev Project 001 on Mar 28, 2018, 7:27:00 PM

fwfb-on-demand-591

Created from Dev Project 001 on Mar 28, 2018, 8:10:30 PM

fwfb17

Created from Dev Project 001 on Apr 12, 2018, 3:04:28 AM

fwng-qyhy

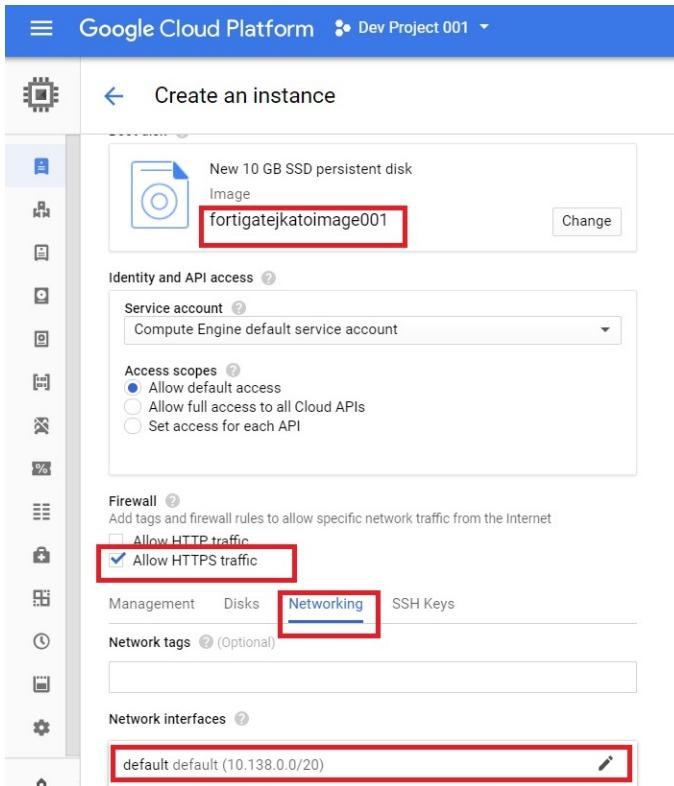
Created from Dev Project 001 on Mar 29, 2018, 11:46:17 PM

Can't find what you're looking for? Explore hundreds of VM solutions in [Cloud Launcher](#)

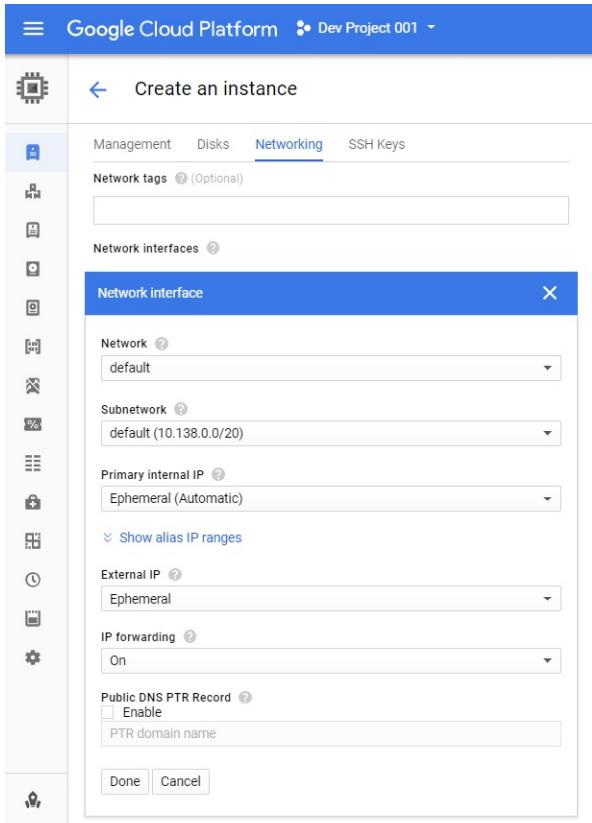
Boot disk type  Size (GB) 
SSD persistent disk  10

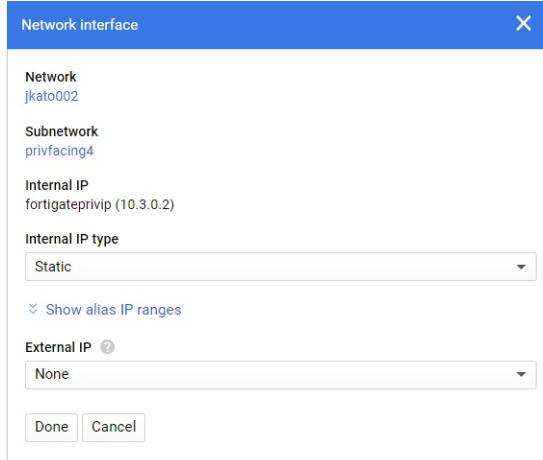
- d. Ensure the new image is selected.
- e. Select *Allow HTTPS traffic*. You access the FortiGate management console using HTTPS. If you allocate multiple network interfaces to the FortiGate, this is nullified at this stage. You can configure this later. See [Configuring Google Cloud firewall rules on page 40](#).
- f. Click *Networking*. Here you want to specify multiple network interfaces. One is located on the public-facing side of the Internet, the other facing a protected private network.



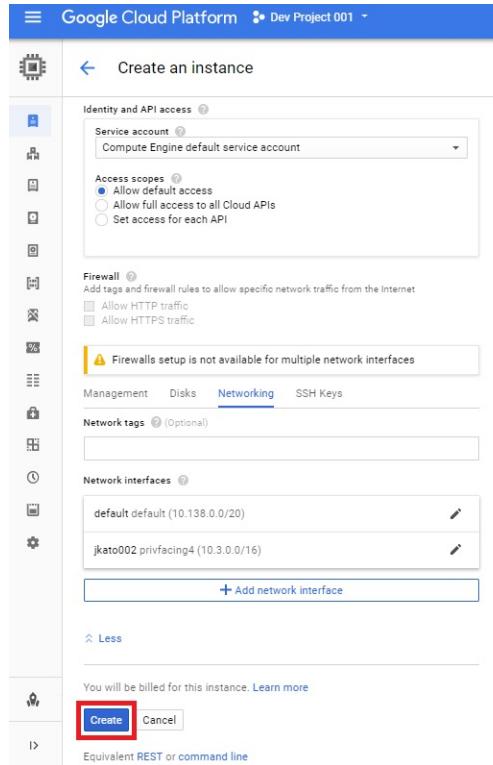
- g. Edit the first network interface. Preferably assign a static IP address. Under *IP Forwarding*, select *On*. Configure other items as needed and click *Done*.



- h. Click *Add network interface* to add the second interface for the private subnet. If you click *Network* there is the preconfigured network list. Select the one located in the same region as you chose to deploy the instance. Under *External IP*, select *None*.



3. After configuring all elements, click *Create*.



After 15-30 minutes, the instance should be up and running.

Google Cloud Platform Dev Project 001

VM instance details

jkato-fgt-563-test007

Remote access

SSH Connect to serial console

Enable connecting to serial ports

Logs

Machine type

n1-standard-2 (2 vCPUs, 7.5 GB memory)

CPU platform

Intel Broadwell

Zone

us-west1-c

Labels

None

Creation time

Apr 20, 2018, 3:26:03 PM

Network interfaces

Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	IP forwarding
default	default	10.138.0.8	—	35.197.98.220 (ephemeral)	On
jkato002	privfacing4	fortigatoprivip (10.3.0.2)	—	None	

Public DNS PTR Record

None

Connecting to the FortiGate-VM

To connect to the FortiGate-VM, you need your login credentials and its public DNS address.

To connect to the FortiGate-VM:

1. Choose the instance from the list of instances on the *VM Instances* page.
2. Depending on how you provisioned the instance, you must use the instance ID or the `fortigate_user_password` as the password. The instance ID is represented as a number that can be found after locating the instance in the GCP Compute Engine console.
 - a. There are two methods to obtain the instance ID. To use the instance ID as the password, do one of the following:

- i. Open *Serial port 1 (console)* as seen.

The screenshot shows the Google Cloud Platform Compute Engine VM instance details page. On the left, a sidebar lists various Compute Engine options: VM instances, Instance groups, Instance templates, Disks, Snapshots, Images, Cloud TPUs, Committed use discounts, Metadata, Health checks, Zones, Operations, Quotas, and Settings. The main area displays the CPU utilization chart for the selected instance. Below the chart, a section titled 'Remote access' shows connection methods: SSH (selected) and Connect to serial console. A dropdown menu under 'Connect to serial console' has 'Serial port 1 (console)' highlighted with a red box. Other options in the dropdown include Serial port 2, Serial port 3, and Serial port 4. At the bottom of the page, there are 'Logs' and 'Stackdriver Logging' sections.

The first time you access the serial console, you find the instance ID represented as a number. This is the login password.

```

(i) https://ssh.cloud.google.com/projects/dev.../zones/
serialport: Connected to dev-project-001-166400.us-west1-a.jkai
1420b361841fb8366e047bb2, active connections: 1).
Scanning /dev/sda1... (100%)
Scanning /dev/sda2... (100%)
Serial number is FGVM00UNLICENSED

FortiGate-VM64-GCP login: GCP instance id:244...5354

```

ii. Do the following:

- i. Select **View gcloud command** on the *VM instance details*.

The screenshot shows the 'VM instance details' page for a specific VM instance. On the right, there's a 'CPU utilization' chart and a table showing CPU usage over the last hour. Below the chart, there's a 'Serial port' section with four entries: Serial port 1 (console), Serial port 2, Serial port 3, and Serial port 4. A context menu is open over the 'Serial port 1 (console)' entry, with the 'View gcloud command' option highlighted by a red box. At the bottom of the page, there's a 'Remote access' section with 'SSH' and 'Connect to serial console' buttons, and a checked checkbox for enabling serial port connections.

- ii. Click **RUN IN CLOUD SHELL**.

gcloud command line

This is the gcloud command line with the parameters you have selected.

```
gcloud compute --project=dev-11-f connect-to-serial-port 11-f --zone=us-central1-f
```

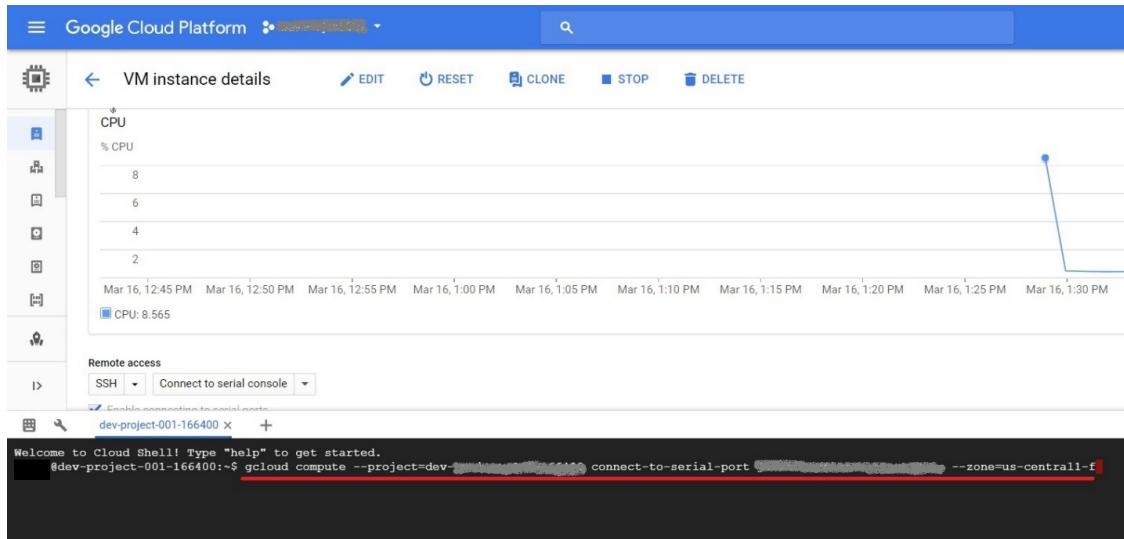
Line wrapping

gcloud reference

CLOSE

RUN IN CLOUD SHELL

- iii. By default, a command is shown as underlined in the following example. Delete the command shown underlined.



- iv. Enter the following command: `gcloud compute instances describe <instance_name>`.

```
Welcome to Cloud Shell! Type "help" to get started.
@dev-project-001-166400:~$ gcloud compute instances describe [REDACTED]
```

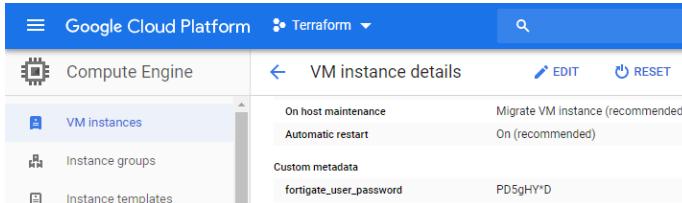
- v. You see a line starting with `id: '<number>'`. This is the FortiGate initial login password.

```
creationTimestamp: '2018-03-16T13:27:55.300-07:00'
deletionProtection: false
disks:
- autoDelete: true
boot: true
deviceName: [REDACTED]-tmp1-boot-disk
index: 0
interface: SCSI
kind: compute#attachedDisk
licenses:
- https://www.googleapis.com/compute/v1/projects/fortigcp-[REDACTED]/global/licenses/fortigate
mode: READ_WRITE
source: https://www.googleapis.com/compute/v1/projects/dev-[REDACTED]/zones/us-central1-f/disks/[REDACTED]
type: PERSISTENT
id: '504-[REDACTED]-35461'
kind: compute#instance
labelFingerprint: [REDACTED]
machineType: https://www.googleapis.com/compute/v1/projects/dev-[REDACTED]/zones/us-central1-f/machineTypes/n1-standard-2
metadata:
fingerprint: [REDACTED]
items:
- key: ssh-keys
value: |
```

You can also enter `gcloud compute instances describe <instance_name> | grep id:` This number is the login password.

```
@dev-project-001-166400:~$ gcloud compute instances describe [REDACTED] | grep id:
No zone specified. Using zone [us-central1-f] for instance: [REDACTED].
id: '504-[REDACTED]-35461'
@dev-project-001-166400:~$
```

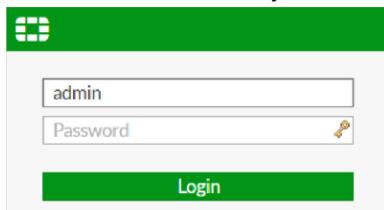
- b. To use the `fortigate_user_password` as the password, go to the *VM instance details* page and find the `fortigate_user_password` under *Custom metadata*.



3. Open an HTTPS session using the FortiGate-VM's public DNS address in your browser (https://<public_DNS>). You can find the FortiGate-VM's public IP address on the *VM instance details* page.

The screenshot shows the 'VM instance details' page for a Google Cloud VM instance named 'FGVM02'. The 'External IP' field, which contains '35.230.64.221 (ephemeral)', is highlighted with a red box. Other visible details include the machine type (n1-standard-1), CPU platform (Intel Broadwell), zone (us-west1-a), and creation time (Feb 20, 2018, 6:33:26 PM).

4. Access the FortiGate in your browser.



5. You see a certificate error message from the browser. This is expected since browsers do not recognize the default self-signed FortiGate certificate. Proceed past the error message.
6. Log into the FortiGate-VM with the username admin and the password (the instance ID or fortigate_user_password, depending on how you provisioned this instance).
7. Upload your license (.lic) file to activate the FortiGate-VM. The FortiGate-VM automatically restarts. After it restarts, wait about 30 minutes until the license is fully registered at Fortinet, and log in again. You now see the FortiOS dashboard. The information in the main dashboard varies depending on the instance type.

The screenshot shows the FortiOS 7.6 dashboard with several panels:

- System Information:** Hostname: docsdemo, Serial Number: FGVM02, Firmware: v7.0.0 build0059 (interim), Mode: NAT, System Time: 2021/03/24 14:02:47, Uptime: 00:02:13:39, WAN IP: 35.247.56.211.
- Licenses:** FortiCare Support, Firmware & General Updates, IPS, AntiVirus, Web Filtering.
- Virtual Machine:** Allocated vCPUs: 2/2 (100%), Allocated RAM: 2 GB, Auto Scaling: Disabled.
- Administrators:** 2 HTTPS, 1 HTTP, 0 FortiExplorer, aharris (super_admin), faz (super_admin).
- Security Fabric:** fabric, docsdemo (Fabric Root), Branch_Office_01 (S108DVCHTPD-GG54), Branch_Office_02.

You are encouraged to change the initial password at the top right corner of the FortiOS GUI.

Configuring Google Cloud firewall rules

You must open incoming port(s) to access FortiGate over the Internet.

HTTPS is the first port that is needed. Other ports are optional depending on what features are enabled. See *FortiGate open ports* in [FortiOS Ports and Protocols](#).

To configure Google Cloud firewall rules:

1. Go to the VPC where the public-facing subnet belongs for the FortiGate.

The screenshot shows the Google Cloud Platform interface for managing VPC network details. The 'Firewall rules' tab is selected. A red box highlights the 'Add firewall rule' button and the 'Firewall rules' tab. Another red box highlights the 'default-allow-https' rule, which is the last entry in the list. The table below shows the configuration for this rule.

Name	Type	Targets	Filters	Protocols / ports	Action	Priority
allow-internal	Ingress	App Engine	IP ranges: 10.128.0.0/24	ICMP, TCP, UDP	Allow	1000
allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000
dchao-ecotest-tcp-22	Ingress	dchao-ecotest-1.tcp.22	IP ranges: 0.0.0.0/0	tcp:22	Allow	1000
dchao-ecotest-tcp-443	Ingress	dchao-ecotest-tcp-443	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000
dchao-ecotest-tcp-514	Ingress	dchao-ecotest-tcp-514	IP ranges: 0.0.0.0/0	tcp:514	Allow	1000
allow-internal-tcp-80	Ingress	App Engine TCP 80	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000
allow-ecotest-tcp-8080	Ingress	dchao-ecotest-tcp-8080	IP ranges: 0.0.0.0/0	tcp:8080	Allow	1000
default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000
goingout	Outbound	App Engine	IP ranges: 0.0.0.0/0	all	Allow	1000

2. Select *Firewall rule*, then *Add firewall rule* if the required port is not open.

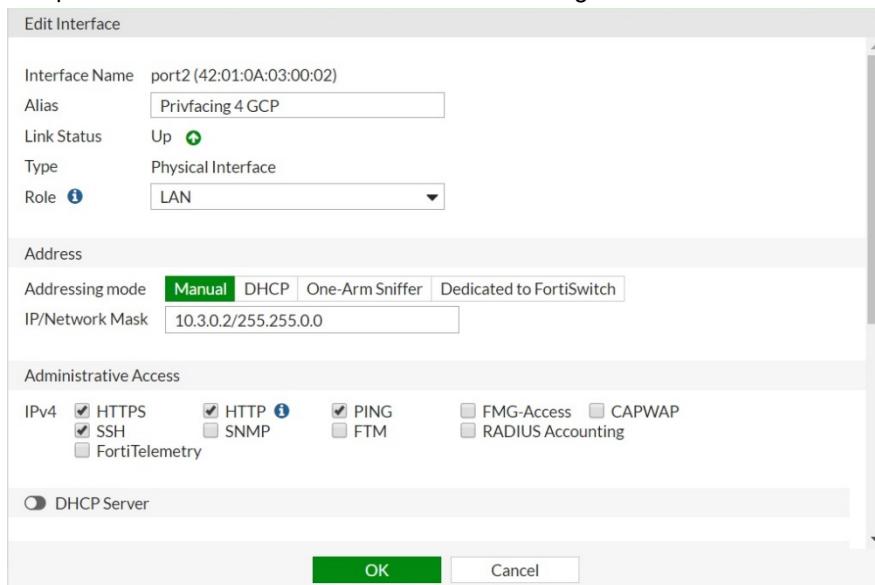
The screenshot shows the 'Create a firewall rule' interface in the Google Cloud Platform. The 'Name' field is set to 'httpkato001'. The 'Priority' is set to '1000'. The 'Direction of traffic' is 'Ingress'. The 'Action on match' is 'Allow'. The 'Targets' are set to 'All instances in the network'. The 'Source filter' is 'IP ranges', and the 'Source IP ranges' field contains 'for example, 0.0.0.0/0, 192.168.2.0/24'. The 'Protocols and ports' section has 'Specified protocols and ports' selected, with 'tcp:80' listed. At the bottom, there are 'Create' and 'Cancel' buttons.

Configuring the second NIC on the FortiGate-VM

After logging into the FortiGate management GUI, you must manually configure the second NIC. Otherwise, the configuration is empty.

To configure the second NIC on the FortiGate-VM:

1. Go to *Network > Interfaces*. port2's IP address/netmask is shown as *0.0.0.0 0.0.0*.
2. Edit port2. Enter the IP address and netmask. Configure other elements as needed, then click *OK*.



Configuring static routing in FortiGate-VM

By default, Google Compute virtual machine (VM) instances' network configuration use single host (/32 net mask) subnets regardless of the subnet CIDR configuration. The internal IP address and routes are assigned to the VM using the dynamic host configuration protocol (DHCP), but in some cases, you may need to configure addresses and routing statically in FortiOS. This guide describes configuring static IP addresses and routing for such requirements.



You can affect the way that subnets work on a per-VM basis during VM deployment using the `MULTI_IP_SUBNET` guest operating system feature, which is described at the end of this guide. As some Fortinet templates use this feature, confirm whether your deployment uses `MULTI_IP_SUBNET` or the standard networking scheme before continuing.

Configuring static network settings

Assigning a static internal IP address in GCP

By default, GCP assigns a VM instance an ephemeral internal IP address every time it is started. Before you configure a static IP address in FortiOS, ensure that Google Compute always uses the same IP address.

To assign a static internal IP address in GCP:

1. Open VM instance details for the FortiGate.
2. Click *Edit*.
3. Under *Network interfaces*, click the pencil icon to edit a desired network interface's properties.

4. From the *Internal IP type* dropdown list, select *Static*. This reserves the currently used internal IP address. This option is only available for instances that are currently running. You can assign a custom internal IP address for a stopped instance by changing its NIC properties.
5. Enter a name for the reserved internal IP address.
6. Repeat steps 3-6 for all desired network interfaces.

Configuring static addressing in FortiOS



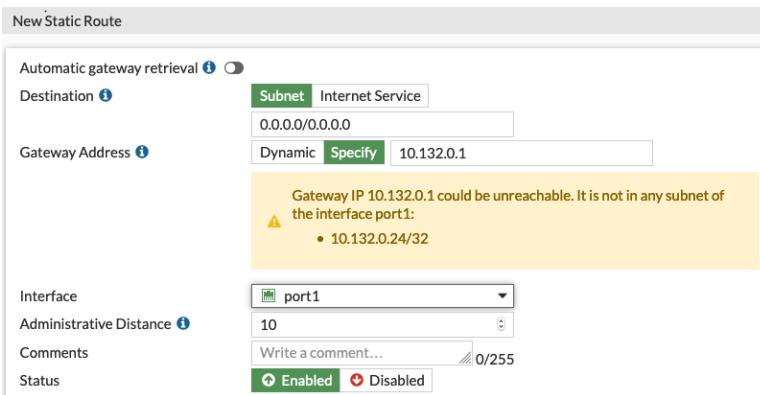
You must follow the proper order of actions as documented. Changing interface settings before configuring routing results in loss of communication with the FortiGate, which you can recover using CLI commands over a serial console.

To configure static addressing in FortiOS:

1. Log in to the FortiOS GUI.
2. Go to *Network > Static Routes*.
3. Configure a route to the local subnet CIDR:
 - a. Click *Create New*.
 - b. In the *Destination* field, enter the desired subnet.
 - c. For *Gateway Address*, select *Specify*. Enter the first IP address in the subnet. In this example, it is 10.132.0.1. The FortiOS GUI displays a warning that the gateway IP address is unreachable via the interface. You can disregard this error. Ensure that you selected the correct interface to be used for sending traffic towards next hop.
 - d. From the *Interface* dropdown list, select the desired interface.
 - e. Click *OK*.

4. If you are configuring the port1 interface, which FortiOS typically uses for egress traffic to the internet, metadata service, and the Google API, you must configure a default route using gateway settings:
 - a. Click *Create New*.
 - b. In the *Destination* field, enter 0.0.0.0/0.0.0.0.
 - c. For *Gateway Address*, select *Specify*. Enter the same IP address configured as the gateway address for the route to the local subnet CIDR. In this example, it is 10.132.0.1.
 - d. From the *Interface* dropdown list, select *port1*.

- e. Click OK.



5. Go to *Network > Interfaces*.
6. Double-click the desired interface.
7. Under *Addressing mode*, select *Manual*. FortiOS automatically populates the proper IP address with a 255.255.255.255 netmask.
8. Click OK.

Load balancer routes

If your FortiGate is accepting connections via a load balancer (LB), you must additionally configure routes to the health probes' IP ranges on each interface receiving traffic. This prevents the reverse path forwarding check from blocking the health probes. The IP ranges are different for different LB types. Google documents the ranges. For the internal LB, the ranges are 35.191.0.0/16 and 130.211.0.0/22.

The 0.0.0.0/0 route on the external interface covers the ranges that the external network LB uses.

MULTI_IP_SUBNET scheme

`MULTI_IP_SUBNET` is a guest operating system feature flag, which you can enable when creating the VM by using the command line, a deployment manager template, or Terraform. The following shows the commands:

```
gcloud compute instances create ...
--guest-os-features MULTI_IP_SUBNET
```

The following shows the deployment manager template:

```
- type: compute.v1.instance
  properties:
    disks:
      - boot: true
    guestOsFeatures:
      - type: MULTI_IP_SUBNET
```

You can verify that the instance was created using this option by clicking Equivalent REST at the bottom of the VM Instance details page or describing the instance using gcloud commands.

The `MULTI_IP_SUBNET` scheme simplifies configuring routing in FortiGates. It uses the subnet configuration known from on-premise networks, where the interface IP address is configured with the subnet's full netmask, instead of 255.255.255.255. Static route configuration in FortiOS is necessary only for the CIDRs not directly connected to the firewall.

Confidential computing

FortiOS 7.4.3 and later versions support confidential computing or confidential VM.

The GCP instance types that are eligible for confidential computing are N2D and C2D, which use the AMD Milan CPU. GCP Compute Engine prompts you to select the supported instance types.

For information on how confidential VM affects the price of running the FortiGate-VM, see [Confidential VM pricing](#).

When deploying a FortiGate-VM via Google Cloud Compute Engine, click *Enable* under *Confidential VM Service* to enable confidential computing/VM.

Confidential VM service

 Confidential Computing is disabled on this VM instance

ENABLE

Google Cloud SDK

You can deploy FortiGate-VM (bring your own license (BYOL)) by using the Google Cloud SDK on your local PC. This is a method of deploying FortiGate-VM on GCP outside of the marketplace product listing and without creating an instance on the Google Cloud Compute Portal.

For details, see [Cloud SDK](#).



This deployment method only applies for BYOL.

Obtaining the deployment image

To obtain the deployment image:

1. Go to the [Fortinet support site](#) and log in.
2. Go to *Support > VM Images*.
3. From the *Select Product* dropdown list, select *FortiGate*.
4. From the *Select Platform* dropdown list, select *Google*.
5. Download the deployment package file. The deployment package file is named “FGT_VM64_GCP-vX-buildXXXX-FORTINET.out.gcp.tar.gz”, where vX is the major version number and XXXX is the build number.



This deployment method only applies for bring your own licensing.

Uploading the deployment image to Google Cloud

To upload the FortiGate deployment image to Google Cloud:

1. Log into Google Cloud.
2. Go to *Storage > Browser*.
3. Create a new bucket or go to an existing bucket.
4. Upload the newly downloaded deployment file.

Creating a FortiGate custom image



This process uses environment variables with the GCloud SDK CLI commands.

To create a FortiGate custom image:

1. Obtain and place the latest FortiGate-VM 7.6 image in your desired bucket:
 - a. Download the FortiGate-VM image from the [Fortinet Support site](#). For more information, see [Obtaining the deployment image on page 47](#).
 - b. Place the obtained image in your desired bucket. For more information, see [Uploading the deployment image to Google Cloud on page 47](#).
2. Create a custom image via the Google Cloud CLI SDK. Assign environment variables with your project ID, the bucket where you placed the FortiGate-VM image, and the image name. This example uses the full name of the file downloaded from the [Fortinet Support site](#) in the image variable:

```
project=<your project id>
bucket=<name of your bucket>
source_image=FGT_VM64_GCP-v7.6.X.F-build1396-FORTINET.out.gcp.tar.gz
image_name=doc-fortigate-vm-image

gcloud compute images create $image_name \
--project=$project \
--source-uri=https://storage.googleapis.com/$bucket/$source_image \
--storage-location=us
```

```
@cloudshell:~ ( )$ project=
bucket=          -bucket
source_image=FGT_VM64_GCP-v7.2.4.F-build1396-FORTINET.out.gcp.tar.gz
image_name=doc-fortigate-vm-image
@cloudshell:~ ( )$ gcloud compute images create $image_name \
--project=$project \
--source-uri=https://storage.googleapis.com/$bucket/$source_image \
--storage-location=us
Created [https://www.googleapis.com/compute/v1/projects/           /global/images/doc-fortigate-vm-image].
NAME: doc-fortigate-vm-image
PROJECT:
FAMILY:
DEPRECATED:
STATUS: READY
@cloudshell:~ ( )$
```

Deploying a FortiGate-VM instance



The networks in this example are already setup. Use existing networks and subnets or create them prior to running the commands in this document. Edit all GCP environment-specific variables to fit your GCP environment.

This guide assumes familiarity with Linux distributions and the GCP environment and that Google Cloud CLI is already installed and configured for your project. For information about installing the Google Cloud CLI SDK, see [Install the gcloud CLI](#).



This process uses environment variables with the GCloud SDK CLI commands. Creating the FortiGate-VM instance references the custom image creation process is referenced.

To deploy a FortiGate-VM instance:

1. Define environment variables:

```
project=<your project id>
zone=us-central1-a
serviceaccount=<your service account>
image_name=doc-fortigate-vm-image
image=projects/$project/global/images/$image_name
```

For information about using publicly available images, see [Finding public FortiGate images on page 17](#).

2. Edit and run the following commands in GCP:

```
gcloud compute instances create doc-fortigate-vm \
--project=$project \
--zone=$zone \
--machine-type=n2d-standard-2 \
--network-interface=network-tier=PREMIUM,private-network-
ip=10.0.1.10,subnet=unprotected-public-subnet \
--network-interface=private-network-ip=10.0.2.10,subnet=protected-private-subnet,no-
address \
--can-ip-forward \
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=doc-fortigate-vm-
boot,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-
balanced \
--create-disk=auto-delete=yes,device-name=doc-fortigate-vm-
log,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced
```

```
@cloudshell:~ ( )$ project=
zone=us-central1-a
serviceaccount=.iam.gserviceaccount.com
image_name=doc-fortigate-vm-image
image=projects/$project/global/images/$image_name
@cloudshell:~ ( )$ gcloud compute instances create doc-fortigate-vm \
--project=$project \
--zone=$zone \
--machine-type=n2d-standard-2 \
--network-interface=network-tier=PREMIUM,private-network-ip=10.0.1.10,subnet=unprotected-public-subnet \
--network-interface=private-network-ip=10.0.2.10,subnet=protected-private-subnet,no-address \
--can-ip-forward \
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=doc-fortigate-vm-boot,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced \
--create-disk=auto-delete=yes,device-name=doc-fortigate-vm-log,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced
Created [https://www.googleapis.com/compute/v1/projects/
/zones/us-central1-a/instances/doc-fortigate-vm].
```



```
NAME: doc-fortigate-vm
ZONE: us-central1-a
MACHINE_TYPE: n2d-standard-2
PREEMPTIBLE:
INTERNAL IP: 10.0.1.10,10.0.2.10
EXTERNAL IP: 34.68. .
STATUS: RUNNING
@cloudshell:~ ( )$
```

3. Add the following lines to bootstrap the new instance with an existing configuration file and BYOL license file:

```
--metadata-from-file "license=<license text file>,user-data=<FortiGate CLI text file>".
--metadata-from-file "license=license.txt,user-data=config.txt".
```

```
@cloudshell:~ ( )$ gcloud compute instances describe doc-fortigate-vm --zone=$zone | grep id
id: '17552502682435'
@cloudshell:~ ( )$
```



This step requires a Linux distribution with the GCloud SDK CLI installed. It leverages the Linux file system to read the license and configuration files and pass them to the FortiGate-VM. See [Bootstrapping FortiGate at initial bootup on page 50](#).

- Obtain the newly deployed FortiGate-VM instance ID by running the following command: `gcloud compute instances describe doc-fortigate-vm -zone=$zone | grep id`. For more information, see [Get the ID of a VM instance](#).
- Access the newly deployed FortiGate-VM using the public IP address from step 2's output and the instance ID from step 4 as the password. If you did not bootstrap a license file, you can apply a license in the FortiOS GUI.

Bootstrapping FortiGate at initial bootup

This section explains how to add bootstrapping of FortiOS CLI commands and a BYOL license at the time of initial bootup as part of Google Cloud commands.

To bootstrap FortiGate at initial bootup:

- Create a text file that contains FortiGate CLI commands. In this example, let's save the file as config.txt. CRLF must be present. Therefore using a text editor that includes CRLF automatically is recommended. This example uses the following CLI commands:

```
config system global
    set timezone 03
end
```

This example sets the timezone as GMT-9 Alaska. You can replace these lines with your own CLI command set.

- You can download a license file from [Customer Service & Support](#) after registering your product code. Save the license file as a .txt file. FortiGate-VM license content resembles the following:

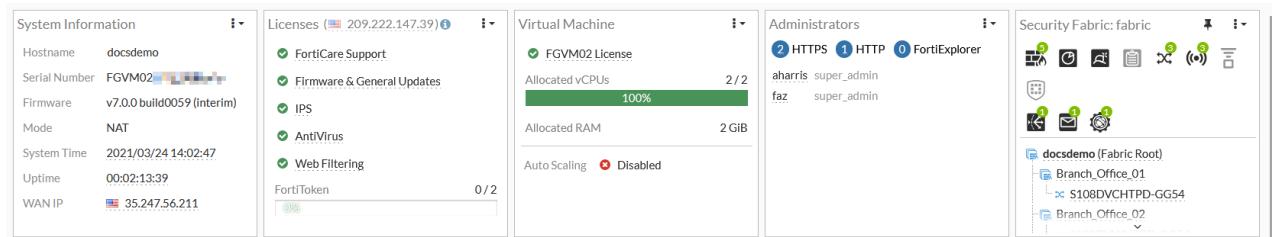
```
-----BEGIN FGT VM LICENSE-----
QAAAABUiZtrwJdUjEe/8C5dIVnOvnY1w/0ZwPPPTG7vm2KKwYvL4++qL0gED6/q
S0SpKwpTf1xjAurGtGyX1vaTpXg0AA1pwFjdjnSGTWJ6dVT7K1D8ncufaa3bcw
s8PmL1vzje4//+C9nqh4Fn/KyDweE1ptDMalscOm0B8r19hQ1DkHx+rgc3jQZ5
ELstrX1l/xQTB/gorG67ZdybdvxzPmvJYDS5AsI+QK88HJ+xGhJhkzbZ4ezU
Hd01ICSm7WKEYV5kaU43sZ9XEstxqPEInah3yXgYtd24pnV68346EHCKAdGyMTP
QdQBMKcT5aei0oogVAOX8D62CSZjh+r1tkdpR5YHoVYZHU9ShBCNjBrojbmnk7
NogQuadQhz28Mtpvzb24mlfDQM1TyQuCtwzJzmbBnvSB07xLqj1nTs20nfB
-----END FGT VM LICENSE-----
```

- Upload the config.txt and license files onto the Linux machine where you will run the Google Cloud SDK commands. Place the files in the same directory.
- Run the command as [Obtaining the deployment image on page 47](#) describes, adding the following:

```
--metadata-from-file "license=<license text file>,user-data=<FortiGate CLI text file>".
In this example, it will be --metadata-from-file "license=license.txt,user-
data=config.txt".
```

```
ubuntu@ip-172-31-33-147:~/BUILD-GCP/FGT/cloud-init$ ls
config.txt license.txt
ubuntu@ip-172-31-33-147:~/BUILD-GCP/FGT/cloud-init$ 
ubuntu@ip-172-31-33-147:~/BUILD-GCP/FGT/cloud-init$ 
ubuntu@ip-172-31-33-147:~/BUILD-GCP/FGT/cloud-init$ sudo gcloud compute instances create jkatoft603cloudinit2 --network-interface network=jkato001,subnet=publicfacing1 --network-interface network=jkato002,subnet=privfacing4,no-address --project dev --image jkato-fgt-603-10162018-001 --can-ip-forward --machine-type n1-highcpu-2 --zone us-west1-a --disk-name=jkatocloudinit2,device-name=jkatodevicecloudinit02,mode=rw,boot=no --metadata-from-file "license=license.txt,user-data=config.txt"
Created https://www.googleapis.com/compute/v1/projects/dev-zones/us-west1-a/instances/jkatoft603cloudinit2.
NAME          ZONE      MACHINE_TYPE  PREEMPTIBLE INTERNAL_IP  EXTERNAL_IP STATUS
jkatoft603cloudinit2  us-west1-a  n1-highcpu-2    PREEMPTIBLE  10.0.0.3.10.3.0.5  35.233.160.96  RUNNING
```

- After deployment, log into the FortiGate by accessing https://<IP_address> in your browser. The system displays the dashboard instead of a license upload window, since the license is already activated.



To see how bootstrapping went, check if the command was successfully run. Open the CLI console and enter `diag debug cloudbinit show`.

If the cloud-init was run successfully, the CLI shows `Finish running script` with no errors. If you see an error with this `diagnose` command, resolve it and try again by checking the license and config.txt files. Ensure that the text file contains CRLF.

- Check the timezone by running `config system global` and `get` commands.

```
Connected

FGVM01TM18000516 #
FGVM01TM18000516 #
FGVM01TM18000516 # diag debug cloudbinit show
>> Checking metadata source gcp
>> Run config script
>> Finish running script
>> FGVM01TM18000516 $ config system global
>> FGVM01TM18000516 (global) $ set timezone 03
>> FGVM01TM18000516 (global) $ end
```

The timezone was changed to Alaska as expected, meaning that the bootstrapping CLI command was successful. This assumes that you used the default FortiGate CLI command in step 1. If you modified the command, test it accordingly.

Deploying a FortiGate-VM on a T2A instance

FortiOS supports the GCP T2A instance family. A publicly available image has been created. You can use it to create a FortiGate-VM instance via the GCloud CLI, leveraging the ARM-based CPU instance family T2A. See [Expanding the Tau VM family with Arm-based processors](#).

To deploy a FortiGate-VM on GCP with the T2A family instance:

The example uses GCloud shell or Linux variables. Edit the variables to match your project ID, zone, and service account:

```
project=<GCP project ID>
zone=us-central1-a
serviceaccount=<your designated services account>
image=projects/fortigcp-project-001/global/images/fortinet-fgt-arm64-740-20230512-001-w-
license
```



Do not edit the image variable as this specifies the ARM specific image.

The following commands deploy a FortiGate-VM on a T2A instance type with two network adapters. The networks and subnets referenced in the GCloud SDK command must be in place and/or created prior to running the command. Edit the following command to fit your environment:

```
gcloud compute instances create fortigate-arm-t2a \
--project=$project \
--zone=$zone \
--machine-type=t2a-standard-2 \
--network-interface=private-network-ip=10.0.1.11,nic-type=GVNIC,subnet=unprotected-public-
subnet \
--network-interface=private-network-ip=10.0.2.11,nic-type=GVNIC,subnet=protected-private-
subnet,no-address \
--can-ip-forward \
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=fortigate-arm-
t2a,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced \
--create-disk=auto-delete=yes,device-name=fortigate-arm-t2a-log,mode=rw,name=fortigate-
arm-t2a-log,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced
```

The following shows the GCloud CLI output when the deployment succeeds:

```
Created [https://www.googleapis.com/compute/v1/projects/...../zones/us-central1-
a/instances/fortigate-arm-t2a].
NAME: fortigate-arm-t2a
ZONE: us-central1-a
MACHINE TYPE: t2a-standard-2
PREEMPTIBLE:
INTERNAL_IP: 10.0.1.11, 10.0.2.11
EXTERNAL_IP: 34.134.160.178
STATUS: RUNNING
```

The following shows the FortiGate-VM bring your own license instance details on the GCP portal:

[fortigate-arm-t2a](#) [EDIT](#) [RESET](#) [CREATE MACHINE IMAGE](#)

[DETAILS](#) [OBSERVABILITY](#) [OS INFO](#) [SCREENSHOT](#)

[SSH](#) [CONNECT TO SERIAL CONSOLE](#)

Connecting to serial ports is enabled in project-wide metadata [?](#)

Logs

[Logging](#)
[Serial port 1 \(console\)](#)

[SHOW MORE](#)

Basic information

Name	fortigate-arm-t2a
Instance Id	
Description	None
Type	Instance
Status	✓ Running
Creation time	Feb 15, 2023, 2:44:24 PM UTC-08:00
Zone	us-central1-a
Instance template	None
In use by	None
Reservations	Automatically choose
Labels	None
Tags ?	- edit
Deletion protection	Disabled
Confidential VM service ?	Disabled
Preserved state size	0 GB

Machine configuration

Machine type	t2a-standard-2
CPU platform	Ampere Altra

Deploying a Shielded VM FortiGate-VM

This is an example FortiGate-VM deployment leveraging Shielded VM features in GCP Compute Engine. The networks that this example uses have been created prior to running the command. Edit the commands to fit your environment prior to running the commands. This document assumes that the GCloud SDK CLI is installed and is ready to use for your environment. This method applies the bring your own license licensing method. For more information on GCP Shielded VM, see [Shielded VMs](#).



Downgrading to a previous GA version when using a UEFI-enabled FortiGate instance is not possible.



GCP marketplace deployments do not support enabling Shielded VM features.

To deploy FortiGate-VM leveraging Shielded VM in GCP Compute Engine:

1. Define environment variables:

```
project=<your project ID>
zone=us-central1-a
serviceaccount=<your service account>
image=projects/fortigcp-project-001/global/images/fortinet-fgt-724-20230310-001-w-
license
```

2. Edit and run the following commands in GCP:

```
gcloud compute instances create shielded-vm-fortigate \
--project=$project \
--zone=$zone \
--machine-type=n2d-standard-2 \
--network-interface=network-tier=PREMIUM,private-network-
ip=10.0.1.10,subnet=unprotected-public-subnet \
--network-interface=private-network-ip=10.0.2.10,subnet=protected-private-subnet,no-
address \
--shielded-secure-boot \
--can-ip-forward \
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=shielded-vm-
fortigate,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-
balanced \
--create-disk=auto-delete=yes,device-name=shielded-vm-fortigate-
log,mode=rw,name=shielded-vm-fortigate-
log,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced
```



The `--shielded-secure-boot` flag explicitly enables the Shielded VM features for the instance that the aforementioned command creates.

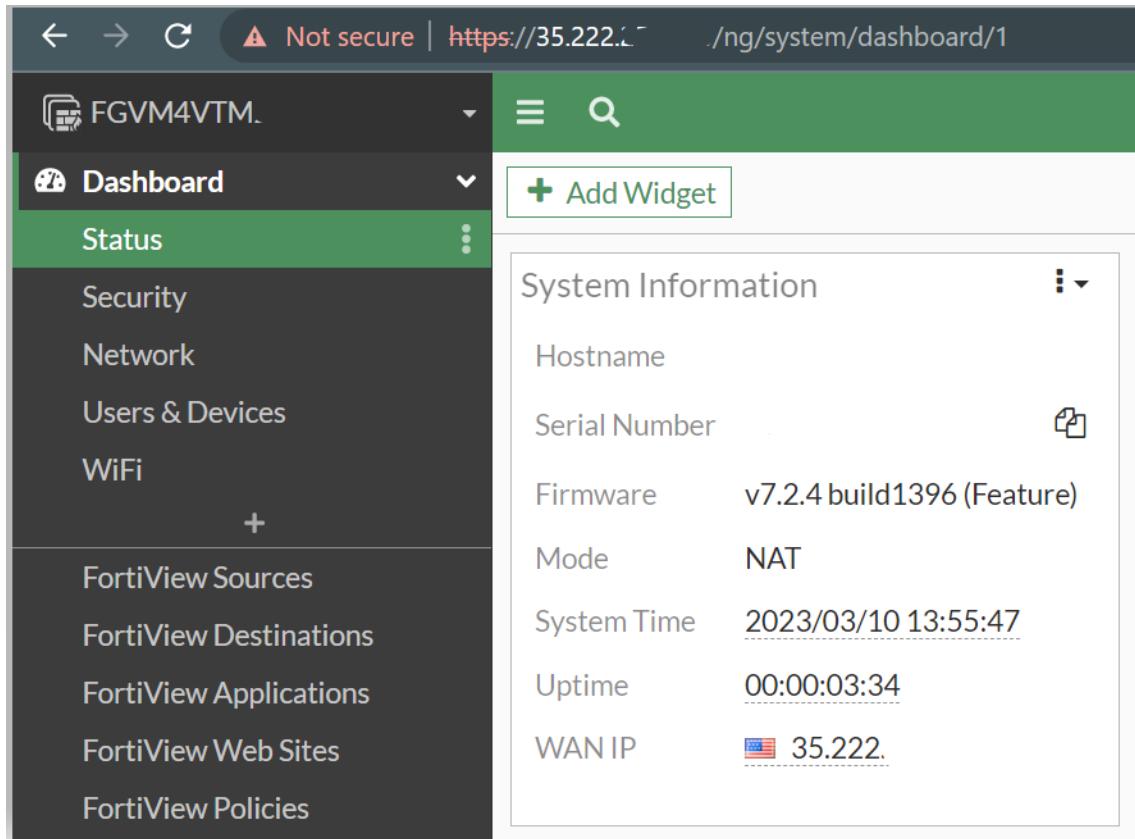
```
@cloudshell:~$ gcloud compute instances create shielded-vm-fortigate \
--project=$project \
--zone=$zone \
--machine-type=n2d-standard-2 \
--network-interface=network-tier=PREMIUM,private-network-ip=10.0.1.10,subnet=unprotected-public-subnet \
--network-interface=private-network-ip=10.0.2.10,subnet=protected-private-subnet,no-address \
--shielded-secure-boot \
--can-ip-forward \
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=shielded-vm-fortigate,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced \
--create-disk=auto-delete=yes,device-name=shielded-vm-fortigate-log,mode=rw,name=shielded-vm-fortigate-log,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced
Created [https://www.googleapis.com/compute/v1/projects/fortigcp-project-001/zones/us-central1-a/instances/shielded-vm-fortigate].
NAME: shielded-vm-fortigate
ZONE: us-central1-a
MACHINE_TYPE: n2d-standard-2
PREEMPTIBLE:
INTERNAL_IP: 10.0.1.10,10.0.2.10
EXTERNAL_IP: 35.222.144.111
STATUS: RUNNING
$
```

3. View instance details and verify Shielded VM features are enabled, including trusted platform module (TPM) support:

Security and access

Shielded VM ?

Secure Boot ?	On
vTPM ?	On
Integrity Monitoring ?	On



The screenshot shows the FortiGate-VM dashboard interface. The left sidebar has a 'Dashboard' section with 'Status' selected, and a '+' button to expand more options like 'FortiView Sources', 'Destinations', 'Applications', 'Web Sites', and 'Policies'. The main panel displays 'System Information' with the following details:

Hostname	
Serial Number	Edit
Firmware	v7.2.4 build1396 (Feature)
Mode	NAT
System Time	2023/03/10 13:55:47
Uptime	00:00:03:34
WAN IP	Edit 35.222.



To check TPM status in the FortiOS CLI, see [TPM support for FortiGate-VM](#).

Confidential computing

FortiOS 7.4.3 and later versions support confidential computing or confidential VM.

The GCP instance types that are eligible for confidential computing are N2D and C2D, which use the AMD Milan CPU. GCP Compute Engine prompts you to select the supported instance types.

For information on how confidential VM affects the price of running the FortiGate-VM, see [Confidential VM pricing](#).

The following highlights the steps needed to enable confidential computing/VM when deploying FortiGate-VM via the Google Cloud SDK.

You must include the following CLI options in the VM creation command string:

```
--min-cpu-platform="AMD Milan" \
--confidential-compute \
```

The following shows an example of the VM creation command:

```
gcloud compute instances create doc-conf-vm-fortigate-vm \
--project=$project \
--zone=$zone \
--machine-type=n2d-standard-2 \
--min-cpu-platform="AMD Milan" \
--confidential-compute \
--network-interface=network-tier=PREMIUM,private-network-ip=10.0.1.10,subnet=unprotected-public-subnet \
--network-interface=private-network-ip=10.0.2.10,subnet=protected-private-subnet,no-address \
 \
--can-ip-forward \
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=doc-fortigate-vm-boot,image=$image,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced \
--create-disk=auto-delete=yes,device-name=doc-fortigate-vm-log,mode=rw,size=10,type=projects/$project/zones/$zone/diskTypes/pd-balanced
```

Basic information

Name	doc-conf-vm-fortigate-vm
Instance Id	
Description	None
Type	Instance
Status	Running
Creation time	May 27, 2024, 2:21:52 PM UTC-07:00
Zone	us-central1-a
Instance template	None
In use by	None
Reservations	Automatically choose (default)
Labels	None
Tags ?	-
Deletion protection	Disabled
Confidential VM service ?	Enabled
Preserved state size	0 GB

Machine configuration

Machine type	n2d-standard-2
--------------	----------------

Deploying FortiGate-VM using Terraform

See the following:

- [Single FortiGate-VM deployment](#)
- [Active-passive HA cluster deployment](#)
- [Deploying FortiGate-VM Next Generation Firewall using Terraform](#)

HA for FortiGate-VM on GCP

The following topics provide overviews of high availability (HA) configurations when using FortiGate-VM for GCP:

- [Deploying FortiGate-VM HA with SDN connector on page 58](#)
- [Protocol forwarding rule with SDN connector on page 69](#)

The following summarizes minimum sufficient roles for active-passive HA deployments:

- Compute Instance Admin (v1)
- Compute Network Admin

Deploying FortiGate-VM HA with SDN connector

FortiGate-VM for Google Cloud Marketplace supports using the FortiGate Clustering Protocol (FGCP) in unicast form to provide an active-passive (A-P) high availability (HA) clustering solution for deployments in GCP. This feature shares a majority of the functionality, including configuration and session synchronization, that FGCP on FortiGate hardware provides with key changes to support GCP software-defined networking (SDN).

This solution works with two FortiGate instances configured as a primary and secondary pair, and requires that you deploy each instance with four network interfaces, within the same availability zone. These FortiGate instances act as a single logical instance and transfer interface public IP addressing.



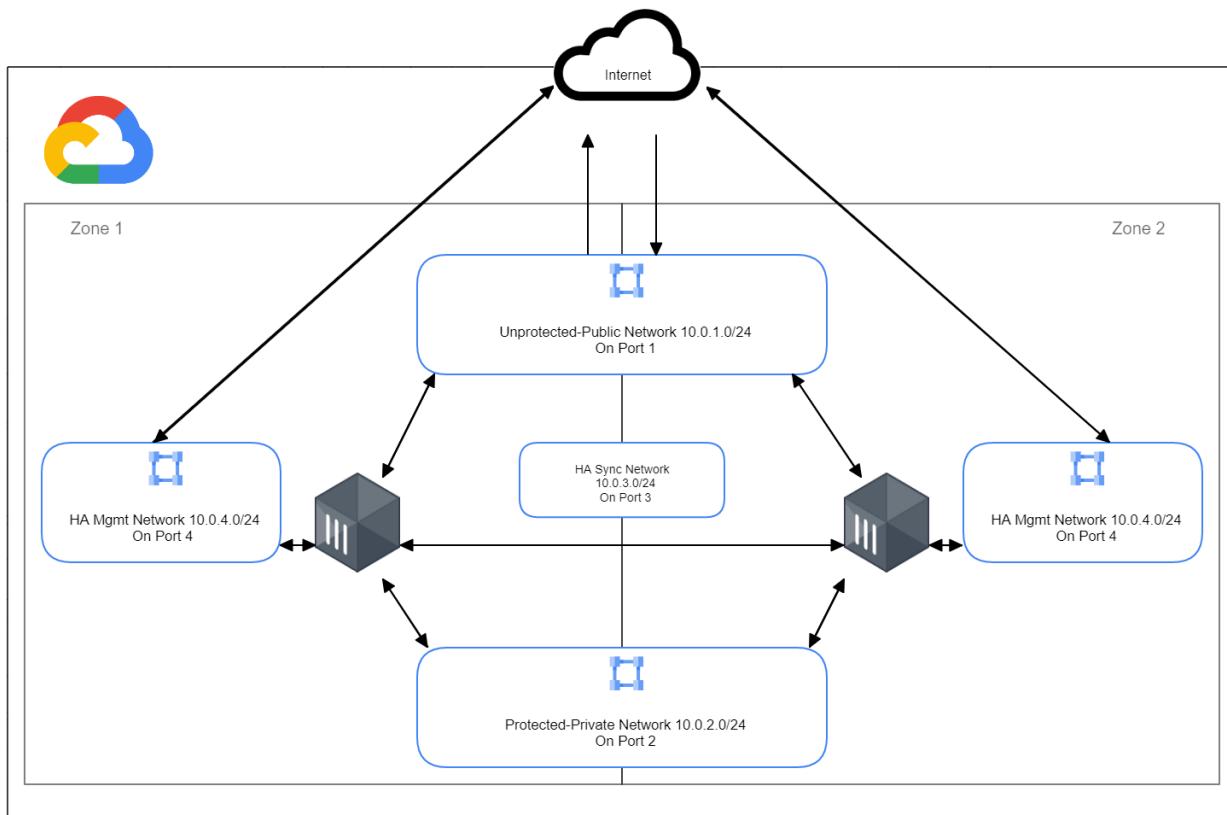
When deploying a FortiGate-VM HA cluster, choose a VM type that supports four or more network interfaces for each FortiGate-VM instance, as GCP does not allow adding network interfaces after you deploy the VMs. You can attach multiple network interfaces only when creating the VM instance on GCP.

Two FortiGate-VM instances must be the same machine type.

The main benefits of this solution are:

- Fast and stateful failover of FortiOS without external automation/services
- Automatic updates to route targets and IP addresses
- Native FortiOS session synchronization of firewall, IPsec/SSL VPN, and voice over IP sessions
- Native FortiOS configuration synchronization
- Ease of use as the cluster is treated as a single logical FortiGate

The following shows a network diagram of this deployment:



IPsec VPN phase 1 configuration does not synchronize between primary and secondary FortiGates across zones. Phase 2 configuration does synchronize.

This example uses four networks for the described purposes:

Network	Purpose
Default network (subnet default)	External Internet-facing network. This uses port1 on the FortiGate.
VPC2 (subnet internal)	Internal network where protected VMs are located. This uses port2 on the FortiGate.
VPC3 (subnet 3)	Subnet dedicated to the heartbeat between two FortiGates. This uses port3 on the FortiGate.
VPC4 (subnet 4)	Subnet dedicated to management access to the two FortiGates. This uses port4 on the FortiGate.

The following summarizes minimum sufficient roles for A-P HA deployments:

- Compute Instance Admin (v1)
- Compute Network Admin

The following summarizes bash environment variables used in the following gcloud commands:

- project=<GCP project ID>
- zone1=<zone for fortigate-a or primary/active FortiGate>
- zone2=<zone for fortigate-b or secondary/pассив FortiGate>
- reservedhaip=<HA Cluster IP to be moved in Failover event>
- reservedfgtahamgmtip=<Public IP to manage fortigate-a >
- reservedfgtbhamgmtip=<Public IP to manage fortigate-b >
- serviceaccount=<your designated services account with correct permissions>



You must set the aforementioned variables in the Linux bash environment before you can use them in gcloud SDK commands.

Check the prerequisites prior to attempting this deployment. This deployment method uses the SDN configuration that [Configuring GCP SDN connector using metadata IAM on page 91](#) describes.

Checking the prerequisites

To deploy and configure the FortiGate-VM as an active-passive high availability solution, you need the following items for this example walkthrough:

- GCP command interface. This example deploys two FortiGate-VMs using GCP. For more information about how to deploy FortiGate-VM using GCP, see [Google Cloud SDK on page 47](#).
- Availability to accommodate the required GCP resources:
 - Four networks/subnets
 - Ensure that the two FortiGates have connectivity to each other on each network.
 - Appropriate ingress/egress firewall rules for relevant networks (same as a single FortiGate-VM deployment). For detail on open ports that the FortiGate requires, see [FortiGate Open Ports](#).
 - Three public (external) IP addresses:
 - One for traffic to/through the active (primary) FortiGate. At the event of failover, this IP address moves from the primary FortiGate to the secondary. This must be a static external IP. It should be reserved/created before creating FortiGate instance. See [Reserving a Static External IP Address](#).
 - Two for management access to each FortiGate. They can be ephemeral IP address, but static ones are highly recommended. See [IP Addresses](#).
 - All internal IP addresses must be static, not DHCP. See [Reserving a Static Internal IP Address](#).
 - Two FortiGate-VM instances in multiple zones:
 - The two nodes must be deployed in the same region.
 - Each FortiGate-VM must have at least four network interfaces.
 - Each FortiGate-VM should have a log disk attached. This is the same requirement as when deploying a single FortiGate-VM.
 - Machine types that support at least four network interfaces. See [Creating Instances with Multiple Network Interfaces](#).
 - Two valid FortiGate-VM BYOL licenses. See [Licensing on page 13](#).
- You must configure an SDN connector for making GCP API calls on the primary FortiGate:
 - For SDN connector configuration on FortiOS 7.2, 7.0, and 6.4.7+ with metadata IAM, see [Configuring GCP SDN connector using metadata IAM on page 91](#).
 - For SDN connector configuration on FortiOS 7.2, 7.0, and 6.4.7+ with a service account, see [Configuring GCP SDN Connector using service account on page 93](#).

Creating VPC networks and firewall rules

This deployment requires four networks which you must create prior to deploying the FortiGates:

Network	Description
unprotected-network	Treated as unsafe and directly attached to the Internet.
protected-network	Commonly referred to as LAN in traditional physical network architectures.
ha-sync-network	All HA functionality, such as session and configuration synchronization, communicates with this network.
mgmt-network	Out of band management network. For A-P HA to properly manage IP addresses and route tables, the HA cluster must have a public IP address assigned to the HA mgmt interface. Without this configuration, failover does not complete successfully and results in failure of the cluster.

Additionally, you must set up the route tables and GCP firewall rules necessary to allow traffic flow through the FortiGates. The route tables and firewall rules are separate from those that you configure on the FortiGates. Name the GCP route tables and firewall rules according to the associated network and functionality.

To create VPC networks:

1. In the GCP console, go to *VPC Networks*, then click *CREATE VPC NETWORK*.
2. In the *Name* field, enter the desired name.
3. From the *Region* dropdown list, select the region appropriate for your deployment. All four networks must be in the same region.
4. From the *IP address range* field, enter the first network's subnet in CIDR format, such as 10.0.1.0/24.
5. Leave all other settings as-is, then click *Create*.
6. Repeat steps 1-5 to create the remaining three networks in your VPC.

GCP firewall rules are stateful, meaning that you only need to create one rule for the originating traffic. However, you may have traffic originate from both the Internet and your GCP resources. This requires you to create both an egress and ingress rule for each VPC network.

To create ingress rules:

1. In the GCP console, go to *VPC networks > Firewall Rules*. Click *Create Firewall Rule*.
2. In the *Name* field, enter the desired name.
3. From the *Network* dropdown list, select the desired network to associate with this firewall rule.
4. For *Direction of Traffic*, select *Ingress*.
5. For *Action on match*, select *Allow*.
6. From the *Targets* dropdown list, select *All instances in the network*.
7. In the *Source IP ranges* field, enter 0.0.0.0/0.
8. For *Protocols and ports*, click *Allow all*, then click *Create*.
9. Repeat steps 1-8 for the remaining three networks in your VPC.

To create egress rules:

1. In the GCP console, go to *VPC networks > Firewall Rules*. Click *Create Firewall Rule*.
2. In the *Name* field, enter the desired name.
3. From the *Network* dropdown list, select the desired network to associate with this firewall rule.
4. For *Direction of Traffic*, select *Egress*.
5. For *Action on match*, select *Allow*.
6. From the *Targets* dropdown list, select *All instances in the network*.
7. In the *Source IP ranges* field, enter 0.0.0.0/0.
8. For *Protocols and ports*, click *Allow all*, then click *Create*.
9. Repeat steps 1-8 for the remaining three networks in your VPC.

There should be a total of eight GCP firewall rules.

Deploying the primary FortiGate

Create the primary FortiGate A in zone1. The following command uses previously declared variables. See the prerequisites section for [Configuring GCP SDN connector using metadata IAM on page 91](#).

To deploy the primary FortiGate-VM instance:

1. Edit and run the following commands in GCP:

```
gcloud compute instances create fortigate-a \
--project=$project \
--zone=$zone1 \
--machine-type=e2-custom-4-8192 \
--network-interface=address=$reservedhaip,network-tier=PREMIUM,private-network-
    ip=10.0.1.10,subnet=unprotected-public-subnet \
--network-interface=private-network-ip=10.0.2.10,subnet=protected-private-subnet,no-
    address \
--network-interface=private-network-ip=10.0.3.10,subnet=ha-sync-subnet,no-address \
--network-interface=address=$reservedfgtahamgmtip,network-tier=PREMIUM,private-network-
    ip=10.0.4.10,subnet=ha-mgmt-subnet \
--can-ip-forward \
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=fortigate-a,image=projects/fortigcp-
    project-001/global/images/fortinet-fgt-723-20221110-001-w-
    license,mode=rw,size=10,type=projects/$project/zones/$zone1/diskTypes/pd-balanced \
--create-disk=auto-delete=yes,device-name=fgt-a-log,mode=rw,name=fgt-primary-
    log,size=10,type=projects/$project/zones/$zone1/diskTypes/pd-balanced
```

2. Gain access to the FortiGate-VM and license the VM.

3. Edit and run the following commands on FortiGate A:

```
config system global
    set hostname fortigate-a
end
config system ha
    set group-id 21
    set group-name <Name of Cluster>
    set mode a-p
    set hbdev "port3" 50
    set session-pickup enable
```

```

set session-pickup-connectionless enable
set ha-mgmt-status enable
config ha-mgmt-interfaces
    edit 1
        set interface "port4"
        set gateway <Gateway Address of the MGMT subnet>
    next
end
set override enable
set priority 200
set unicast-hb enable
set unicast-hb-peerip <HA Sync network Address of the First Fortigate>
set unicast-hb-netmask <subnet mask of the hasync network>
end
config system sdn-connector
    edit "gcp_ha"
        set type gcp
        set ha-status enable
        config external-ip
            edit "reserved-fgt-port1public"
        next
    end
config route
    edit " protected-private-rt"
    next
end
    set use-metadata-iam enable
next
end

```

- Configure a virtual domain (VDOM) exception. You must configure a VDOM exception to prevent interface synchronization between the two FortiGates:

```

config system vdom-exception
    edit 1
        set object system.interface
    next
    edit 2
        set object router.static
    next
    edit 3
        set object firewall.vip
    next
end

```

Deploying the secondary FortiGate

Create the secondary FortiGate B in zone us-central1-a by changing the zone variable to us-central1-a. The following command uses previously declared variables. See the prerequisites section for [Configuring GCP SDN connector using metadata IAM on page 91](#).



Port1 on FortiGate B does not have a reserved public IP address, as it is reassigned the port1/WAN reserved public IP address. Use the FortiGate B port1 ephemeral public IP address to license and configure the FortiGate, then release the ephemeral public IP address after you have configured high availability (HA) and before a failover is initiated.

To deploy the secondary FortiGate-VM instance:

1. Edit and run the following commands in GCP:

```
gcloud compute instances create fortigate-b \
--project=$project \
--zone=$zone2 \
--machine-type=e2-custom-4-8192 \
--network-interface=network-tier=PREMIUM,private-network-
ip=10.0.1.11,subnet=unprotected-public-subnet \
--network-interface=private-network-ip=10.0.2.11,subnet=protected-private-subnet,no-
address \
--network-interface=private-network-ip=10.0.3.11,subnet=ha-sync-subnet,no-address \
--network-interface=address=$reservedfgtbhamgmtip,network-tier=PREMIUM,private-network-
ip=10.0.4.11,subnet=ha-mgmt-subnet --can-ip-forward
--service-account=$serviceaccount \
--scopes=https://www.googleapis.com/auth/cloud-platform \
--create-disk=auto-delete=yes,boot=yes,device-name=fortigate-b,image=projects/fortigcp-
project-001/global/images/fortinet-fgt-723-20221110-001-w-
license,mode=rw,size=10,type=projects/$project/zones/$zone2/diskTypes/pd-balanced \
--create-disk=auto-delete=yes,device-name=fgt-b-log,mode=rw,name=fgt-secondary-
log,size=10,type=projects/$project/zones/$zone2/diskTypes/pd-balanced
```

2. Gain access to the FortiGate-VM and license the VM.

3. Edit and run the following commands on FortiGate B:

```
config system global
    set hostname fortigate-b
end
config system ha
    set group-id 21
    set group-name <Name of Cluster>
    set mode a-p
    set hbdev "port3" 50
    set session-pickup enable
    set session-pickup-connectionless enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port4"
            set gateway <Gateway Address of the MGMT subnet>
        next
    end
    set override enable
    set priority 150
    set unicast-hb enable
    set unicast-hb-peerip <HA Sync network Address of the First Fortigate>
    set unicast-hb-netmask <subnet mask of the hasync network>
end
```



After you have configured HA on the secondary FortiGate, you must remove the ephemeral public IP address from port1 from the secondary FortiGate. Otherwise, the HA failover and elastic IP address move fails due to the interface already having an assigned public IP address.

4. Configure a virtual domain (VDOM) exception. You must configure a VDOM exception to prevent interface synchronization between the two FortiGates:

```

config system vdom-exception
edit 1
    set object system.interface
next
edit 2
    set object router.static
next
edit 3
    set object firewall.vip
next
end

```

Creating a GCP route table

When you created your VPC networks, GCP automatically created several route tables. You must create one additional route table, which allows the protected network to use the FortiGates as the default gateway.

To create a GCP route table:

1. In the GCP console, click the primary FortiGate's instance details and note the IP address assigned to the protected network interface, nic1 if you followed the order of interface creation previously covered in this guide.

Network interfaces

Name ↑	Network	Subnetwork	Primary internal IP address	Alias IP ranges	Stack Type
nic0	unprotected-public	unprotected-public-subnet	10.0.1.10		IPv4
nic1	protected-private	protected-private-subnet	10.0.2.10		IPv4
nic2	ha-sync	ha-sync-subnet	10.0.3.10		IPv4
nic3	ha-mgmt	ha-mgmt-subnet	10.0.4.10		IPv4

2. Go to *VPC Networks > Routes*, then click *CREATE ROUTE*.
3. In the *Name* field, enter the route table name.
4. From the *Network* dropdown list, select the protected network.
5. In the *Destination* field, enter 0.0.0.0/0.
6. In the *Priority* field, enter 10. You can set this to any number less than 1000, which is the default priority for the GCP default route table. This ensures you route all traffic from the protected network through the FortiGate before leaving the VPC.
7. From the *Next hop* dropdown list, select *Specify an IP address*.
8. In the *Next hop IP address* field, enter the IP address of the FortiGate interface assigned to the protected network. In this example, the IP address is 10.0.2.10, but your IP address may be different.
9. Click *Create*.

Uploading the license and configuring network interfaces

To upload the license and configure network interfaces:

1. Go to *Compute Engine > VM instances*.
2. Note the external IP addresses assigned to each FortiGate's unprotected network interface.
3. Depending on how you provisioned the instance, you must use the instance ID or the `fortigate_user_password` (found in the GCP management console under VM instance details) as the password. The instance ID is represented as a number that can be found after locating the instance in the GCP Compute Engine console. Click the name of each instance and note the instance ID or the `fortigate_user_password`.
4. Configure the primary FortiGate:
 - a. Open a web browser window for the primary FortiGate. Go to `http://<FortiGate external IP address>`.
 - b. Log in with `admin` as the username and the FortiGate instance ID or `fortigate_user_password` as the password.
 - c. FortiOS prompts you to change the admin password immediately. Change the password as required.
 - d. Log back into the FortiGate using the `admin` username and the newly changed password.
 - e. Click *Upload* to install the license. Upload the license. The FortiGate reboots automatically.
 - f. Once the reboot is complete, FortiOS redirects you to the dashboard. Go to *Network > Interfaces*.
 - g. FortiGate port2, port3, and port4 show no IP addresses. Edit port2:
 - i. Under *Address*, ensure that *Manual* is selected under *Addressing Mode*.
 - ii. In the *IP/Network Mask* field, enter the IP address that GCP assigned to nic1 with a netmask of `255.255.255.255`. While the `255.255.255.255` netmask may seem different from what you would expect in a typical network, it works in GCP due to the SDN capabilities of the GCP VPC.
 - iii. Click *OK*.
 - h. Repeat step g for port3 and port4. Port3's IP address is the same as nic2 in GCP, while port4's IP address is the same as nic3 in GCP.
5. Repeat steps 4 for the secondary FortiGate.

Testing and troubleshooting

To optionally create an Ubuntu PC that can access the Internet via the FortiGates HA, edit and run the following commands in GCP:

```
gcloud compute instances create test-www --project=$project --zone=$zone --machine-type=e2-custom-4-8192 --network-interface=subnet=protected-private-subnet --maintenance-policy=MIGRATE --provisioning-model=STANDARD --service-account=$example-service-account
scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --create-disk=auto-delete=yes,boot=yes,device-name=test-www,image=projects/ubuntu-os-cloud/global/images/ubuntu-1804-bionic-v20221117,mode=rw,size=10,type=projects/$project/zones/$zone1/diskTypes/pd-balanced --no-shielded-secure-boot --shielded-vtpm --shielded-integrity-monitoring --reservation-affinity=any
```

To test FortiGate-VM HA:

1. Ensure that the HA status is in-sync and that the public external IP address is attached to the primary FortiGate:

```
HA Health Status: OK
Model: FortiGate-VM64-GCP
Mode: HA A-P
```

HA for FortiGate-VM on GCP

```

Group: 21
Primary selected using:
<2022/11/24 13:48:45> vcluster-1: FGVM4VTM22xxxxxx is selected as the primary
because its override priority is larger than peer member FGVM4VTM22xxxxxx.
ses_pickup: enable, ses_pickup_delay=disable
override: enable
unicast_hb: peerip=10.0.3.11, myip=10.0.3.10, hasync_port='port3'
Configuration Status:
    FGVM4VTM22xxxxxx (updated 4 seconds ago): in-sync
    FGVM4VTM22xxxxxx (updated 1 seconds ago): in-sync
Primary : fortigate-a , FGVM4VTM22xxxxxx, HA cluster index = 0
Secondary : fortigate-b , FGVM4VTM22xxxxxx, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 10.0.3.10
Primary: FGVM4VTM22xxxxxx, HA operating index = 0
Secondary: FGVM4VTM220xxxxxx, HA operating index = 1

```

								List	Faceplate	All	
Synchronized	Priority	Hostname	Serial No.	Role	Uptime	Sessions	Throughput				
	1 3 5 7 9 11 13 15 17 2 4 6 8 10 12 14 16 18	FGT-A	FGTGCRA2D	Master	00:00:03:51	96	273.00 kbps				
	1 3 5 7 9 11 13 15 17 2 4 6 8 10 12 14 16 18	FGT-B	FGTGCRA2D	Slave	00:03:05:39	40	21.00 kbps				

2. Log in to the Ubuntu PC if created.
3. Verify that the PC can access the Internet via FortiGate A, since FortiGate A is the primary FortiGate. Verify that the route-internal route gateway is set as 10.0.2.10, the FortiGate A IP address.

ALL DYNAMIC PEERING

One or more VPC networks in this project has been configured to import custom routes using VPC Network Peering. Any imported custom dynamic routes how Google Cloud resolves conflicts.

protected-private-rt Enter property name or value

<input type="checkbox"/>	Name ↑	Description	Destination IP range	Priority	Instance tags	Next hop	Network
<input type="checkbox"/>	protected-private-rt		0.0.0.0/0	1000	None	IP address 10.0.2.10	protected-private

4. Shut down FortiGate A.
5. Verify that FortiGate B is now the primary FortiGate.
6. Using an API call, ensure that the route-internal route was removed and replaced with a new one, which has set the gateway as 10.0.2.11, the FortiGate B IP address.

ALL DYNAMIC PEERING

One or more VPC networks in this project has been configured to import custom routes using VPC Network Peering. Any imported custom dynamic routes how Google Cloud resolves conflicts.

<input type="checkbox"/>	Name	Description	Destination IP range	Priority	Instance tags	Next hop	Network
<input type="checkbox"/>	protected-private-rt		0.0.0.0/0	1000	None	IP address 10.0.2.11	protected-private

7. Verify that the public IP address has detached from FortiGate A and attached to FortiGate B.
8. Log in to the PC.
9. Verify that the PC can access the Internet via FortiGate B, since FortiGate B is now the primary FortiGate.

To run diagnose commands:

After FortiGate A is shut down and FortiGate B becomes the new primary FortiGate, run the following diagnose command to see what happened to the route and public external IP address during the failover procedure:

```
FGT-B # diagnose debug application gcpd -1
```

The following shows the procedure of removing the old route (route-internal) and replacing it with a new route:

```
failover route: protected-private-rt, move next hop from 10.0.2.10 to 10.0.2.11
[pid 394]: failover route: protected-private-rt
[pid 394]: remove route protected-private-rt on next hop 10.0.2.10
[pid 394]: route protected-private-rt is updated to next hop 10.0.2.11 successfully.
```

The following shows the procedure of attaching a public external IP address to the new primary FortiGate B:

```
[pid 393]: nic0 of instance fortigate-a is using eip 34.135.xx8.xxx
[pid 393]: remove eip 34.135.xx8.xxx from instance fortigate-a(nic0).
[pid 393]: attach eip 34.135.xx8.xxx to instance fortigate-b(nic0).
[pid 393]: eip fgtprimaryip(34.135.xx8.xxx) is attached to local successfully.
```

Protocol forwarding rule with SDN connector

Compute Engine supports protocol forwarding, which lets you create forwarding rule objects that can send packets to a non-NATed target instance.

Each target instance contains a single virtual machine instance that receives and handles traffic from the corresponding forwarding rules.

In an active-passive (A-P) high availability (HA) configuration, when the failover occurs, the forwarding rules are updated to use the active/primary instance along with the route associated with the A-P configuration in the SDN connector.



This guide assumes that you have created all networks and FortiGate instances prior to starting the following instructions.

The Google CLI commands in this guide use Linux operating system variables.

This configuration requires the following:

- Networks and subnetworks created to support FortiGate A-P HA deployment
- Two FortiGate-VMs deployed, running, and configured as an A-P HA cluster
- Roles and Identity & Access Management permissions in the respective project to allow for changes to forwarding rule target instances to be updated on failover, such as the Compute load balancer (LB) admin role



Protocol forwarding and LB deployments can and do overlap in GCP resource configuration. However, they are not the same deployment method. For information about these deployment types, see the following:

- For FortiGate-VM in an A-P HA cluster, see [HA for FortiGate-VM on GCP on page 58](#).
- For FortiGate-VM HA with external and internal LB, see [Deploying FortiGate-VM HA with external and internal LB \(web console\) on page 73](#) and [Deploying FortiGate-VM HA with external and internal LB \(GCloud CLI\) on page 81](#).

Creating a target instance for each FortiGate-VM

To create a target instance for each FortiGate-VM:

1. Create a target instance around the primary FortiGate-VM instance:

```
gcloud compute target-instances create doc-pf-primary-instance \
--instance <INSTANCE_NAME>
```

```
@cloudshell:~ ( )$ gcloud compute target-instances create doc-pf-primary-instance --instance pf-fortigate-active --zone $zone1 /zones/us-central1-c/targetInstances/doc-pf-primary-instance.
Created [https://www.googleapis.com/compute/v1/projects/
NAME: doc-pf-primary-instance
ZONE: us-central1-c
INSTANCE: pf-fortigate-active
NAT_POLICY: NO_NAT
@cloudshell:~ ( )$
```

2. Create a target instance around the secondary FortiGate-VM instance:

```
gcloud compute target-instances create doc-pf-secondary-instance \
--instance <INSTANCE_NAME>
```

3. Create a forwarding rule:

```
gcloud compute forwarding-rules create doc-pf-rule \
--ip-protocol TCP \
--ports 80 \
--target-instance doc-pf-primary-instance \
--target-instance-zone=$zone1 \
--region us-central1
```

Configuring the FortiGates

To configure the FortiGates:

1. Edit the SDN connector on each FortiGate-VM to add the protocol forwarding rule configuration:

```
config system sdn-connector
  edit "gcp_conn"
    config forwarding-rule
      edit "doc-pf-rule"
        set target "doc-pf-primary-instance"
      next
    end
  next
end
```

2. For the secondary FortiGate-VM, specify the secondary target instance for the forwarding rule configuration:

```
config system sdn-connector
  edit "gcp_conn"
    config forwarding-rule
      edit "doc-pf-rule"
        set target "doc-pf-secondary-instance"
      next
    end
  next
end
```



The example configuration is not a full SDN configuration and only illustrates the changes needed to support protocol forwarding.

Testing the route and forwarding rule failover

To test the route and forwarding rule failover:

- Enter the following debug commands on each FortiGate prior to initiating failover:

```
diagnose debug reset
diagnose debug console timestamp enable
diagnose debug enable
diagnose debug application gcpd -1
```

The following shows the primary FortiGate debug output before failover is initiated:

```
2023-06-07 10:57:59 gcp_conn got 164 addresses
2023-06-07 10:57:59 gcpd sdn connector gcp_conn start updating IP addresses
2023-06-07 10:57:59 gcpd sdn connector gcp_conn finish updating IP addresses
2023-06-07 10:57:59 gcpd reap child pid: 2488
2023-06-07 10:58:12 In HA primary state
2023-06-07 10:58:12 get nics info for instance pf-fortigate-active
2023-06-07 10:58:12 get instance nic: nic0, 10.0.1.10, unprotected-public, accessConfig(external-nat), eip( 83.189), tier(PREMIUM)
2023-06-07 10:58:12 get instance nic: nic1, 10.0.2.10, protected-private
2023-06-07 10:58:12 get instance nic: nic2, 10.0.3.10, ha-sync
2023-06-07 10:58:12 get instance nic: nic3, 10.0.4.10, ha-mgmt, accessConfig(external-nat), eip( 176.216), tier(PREMIUM)
2023-06-07 10:58:12 gcpd checking eip: fgtpprimaryip
2023-06-07 10:58:12 eip: fgtpprimaryip(.83.189)
2023-06-07 10:58:12 attached instance: pf-fortigate-active, zone: us-central1-c
2023-06-07 10:58:12 eip fgtpprimaryip( 83.189) is attached in local instance already
2023-06-07 10:58:12 gcpd checking route: -protected-private
2023-06-07 10:58:12 route: aj-protected-private (next hop 10.0.2.10) is pointed to local instance already
2023-06-07 10:58:12 gcpd checking forwardrule: doc-pf-rule
2023-06-07 10:58:12 forwardrule: doc-pf-rule is pointed to target doc-pf-primary-instance already
```

- Before failover, confirm that the route and forwarding rule use the primary FortiGate IP address (10.0.2.10 in this example) and primary target instance.

Filter Enter property name or value

Name ↑	Type	Destination IP range	Priority	Instance tags	Next hop
protected-private	Static	0.0.0.0/0	1000	None	IP address 10.0.2.10


```
@cloudshell:~ ( )$ gcloud compute forwarding-rules list --filter="name=( 'doc-pf-rule')"
NAME: doc-pf-rule
REGION: us-central1
IP ADDRESS: 35.222.15.144
IP_PROTOCOL: TCP
TARGET: us-central1-c/targetInstances/doc-pf-primary-instance
@cloudshell:~ ( )$
```

The following shows the secondary FortiGate debug output after failover is initiated:

```

2023-06-07 11:23:34 [pid 2211]: route . protected-private is updated to next hop 10.0.2.11 successfully.
2023-06-07 11:23:34 gcpd ha failover waiting for pid [ 2209 2211 2212 ], ps = 3
2023-06-07 11:23:34 pid 2211 returns, ps = 2
2023-06-07 11:23:35 [pid 2209]: api status: PENDING
2023-06-07 11:23:35 [pid 2212]: api status: RUNNING
2023-06-07 11:23:36 [pid 2209]: api status: PENDING
2023-06-07 11:23:36 [pid 2212]: api status: RUNNING
2023-06-07 11:23:37 [pid 2209]: api status: PENDING
2023-06-07 11:23:37 [pid 2212]: api status: RUNNING
2023-06-07 11:23:38 [pid 2209]: api status: PENDING
2023-06-07 11:23:38 [pid 2212]: api status: RUNNING
2023-06-07 11:23:39 gcpd ha failover waiting for pid [ 2209 2212 ], ps = 2
2023-06-07 11:23:39 [pid 2209]: api status: PENDING
2023-06-07 11:23:39 [pid 2212]: api status: RUNNING
2023-06-07 11:23:40 [pid 2209]: api status: DONE
2023-06-07 11:23:40 [pid 2209]: leaving gcpd_ha_remove_eip_from_remote, rc = 0
2023-06-07 11:23:40 [pid 2209]: attach eip [ . . . . . 83.189 to instance pf-fortigate-passive(nic0).
2023-06-07 11:23:40 [pid 2212]: api status: RUNNING
2023-06-07 11:23:40 [pid 2209]: api status: RUNNING
2023-06-07 11:23:41 [pid 2212]: api status: RUNNING
2023-06-07 11:23:42 GCP guest environment update
2023-06-07 11:23:42 GCP metadata has new change, etag 97f1af80c8757b34
2023-06-07 11:23:42 GCP parse project ssh keys
2023-06-07 11:23:42 GCP parse instance ssh keys
2023-06-07 11:23:42 GCP account update finish
2023-06-07 11:23:42 GCP update done
2023-06-07 11:23:42 [pid 2209]: api status: RUNNING
2023-06-07 11:23:42 [pid 2212]: api status: RUNNING
2023-06-07 11:23:43 [pid 2209]: api status: RUNNING
2023-06-07 11:23:43 [pid 2212]: api status: RUNNING
2023-06-07 11:23:44 gcpd ha failover waiting for pid [ 2209 2212 ], ps = 2
2023-06-07 11:23:44 [pid 2209]: api status: DONE
2023-06-07 11:23:44 [pid 2209]: leaving gcpd_ha_add_eip_to_local, rc = 0
2023-06-07 11:23:44 [pid 2209]: eip fgtprimaryip( . . . . . 83.189) is attached to local successfully.
2023-06-07 11:23:44 gcpd ha failover waiting for pid [ 2209 2212 ], ps = 2
2023-06-07 11:23:44 pid 2209 returns, ps = 1
2023-06-07 11:23:44 [pid 2212]: api status: RUNNING
2023-06-07 11:23:45 [pid 2212]: api status: RUNNING
2023-06-07 11:23:47 [pid 2212]: api status: RUNNING
2023-06-07 11:23:48 [pid 2212]: api status: RUNNING
2023-06-07 11:23:49 [pid 2212]: api status: RUNNING
2023-06-07 11:23:49 gcpd ha failover waiting for pid [ 2212 ], ps = 1
2023-06-07 11:23:50 [pid 2212]: api status: RUNNING
2023-06-07 11:23:51 [pid 2212]: api status: RUNNING
2023-06-07 11:23:52 [pid 2212]: api status: RUNNING
2023-06-07 11:23:53 [pid 2212]: api status: RUNNING
2023-06-07 11:23:54 gcpd ha failover waiting for pid [ 2212 ], ps = 1
2023-06-07 11:23:54 [pid 2212]: api status: DONE
2023-06-07 11:23:54 [pid 2212]: leaving gcpd_ha_set_target, rc = 0
2023-06-07 11:23:54 [pid 2212]: forwardrule doc-pf-rule is updated to next hop doc-pf-secondary-instance successfully.

```

3. After failover, confirm that the route and forwarding rule use the secondary FortiGate IP address (10.0.2.11 in this example) and secondary target instance.

Name ↑	Type	Destination IP range	Priority	Instance tags	Next hop
protected-private	Static	0.0.0.0/0	1000	None	IP address 10.0.2.11


```

@cloudshell:~ ( )$ gcloud compute forwarding-rules list --filter="name=( 'doc-pf-rule')"
NAME: doc-pf-rule
REGION: us-central1
IP_ADDRESS: 35.222.15.144
IP_PROTOCOL: TCP
TARGET: us-central1-a/targetInstances/doc-pf-secondary-instance
@cloudshell:~ ( )$ 

```

For more information about this feature, see [Set up protocol forwarding](#).

Deploying FortiGate-VM HA with external and internal LB (web console)

Deploying FortiGates in a high availability (HA) cluster eliminates having a single point of failure and elevates Google Compute SLA to 99.98%. HA in a load balancer (LB) sandwich design features two FortiGate instances in an active-passive (A-P) cluster between a pair of GCP LBs (“LB sandwich” pattern) for fast and stateful failover.



This guide describes manually deploying the cluster. Fortinet publishes Terraform and deployment manager modules in its GitHub repositories. Consult documentation and examples in individual modules for automated deployments.

FGCP in public cloud

The FortiGate clustering protocol (FGCP) is a proprietary protocol used to create high availability clusters in hardware and virtual FortiGate deployments. Due to the way cloud networks work, you cannot take full advantage of the protocol capabilities and must use its unicast version, limiting the functionality to an active-passive cluster of two instances.

FGCP provides automatic synchronization of connection tables as well as synchronization of configuration from the primary to the secondary instance. You must apply all configuration changes to the primary instance. Using the priority option to statically assign the primary and secondary roles in the cluster is recommended.

For a more detailed description of the architecture, see [GitHub](#).

The following provides the configuration steps for this deployment:

1. [Predeployment steps on page 73](#)
2. [Deploying FortiGate-VM instances on page 74](#)
3. [Reserving static addresses on page 75](#)
4. [Creating instance groups on page 76](#)
5. [Creating the external LB on page 76](#)
6. [Creating the internal LB on page 77](#)
7. [Creating a custom route on page 78](#)
8. [Configure FortiGates networking on page 78](#)
9. [Configuring FortiGate clustering on page 79](#)
10. [Configure health check probe responders on page 80](#)
11. [Best practices and next steps on page 81](#)

Predeployment steps

To complete predeployment steps:

1. Decide the region, zones, licensing, firmware version, and machine type for the deployment.
2. Prepare three virtual private cloud (VPC) networks with non-overlapping subnets in the deployment region to use as external, internal, and heartbeat networks. You can optionally split the heartbeat and dedicated management network. This is obligatory for older firmware.

demo-vpc-ext	1	1460	Custom	None
us-west1	demo-ext			10.0.1.0/24
demo-vpc-hasync	1	1460	Custom	None
us-west1	demo-hasync			10.0.3.0/24
demo-vpc-int	1	1460	Custom	None
us-west1	demo-int			10.0.2.0/24

3. If desired, complete the following predeployment table. This allows you to easily access necessary information during the deployment. Mentions of these fields are bolded in subsequent sections of this guide:

REGION	Region to deploy to. It should be the same region your servers are using.
ZONE_1	Primary availability zone (AZ). This zone must be in REGION.
ZONE_2	Secondary AZ. This zone must be in REGION.
VPC_EXT / SUBNET_EXT	External VPC and subnet names.
VPC_INT / SUBNET_INT	Internal VPC and subnet names.
VPC_HASYNC / SUBNET_HASYNC	Heartbeat and management VPC and subnet names.
FIRMWARE_VERSION	Firmware version you plan to deploy. Some versions may be unavailable.
LICENSING	Bring your own license and pay as you go licensing use different images. You cannot change licensing without redeploying the FortiGates.
MACHINE_TYPE	Select the machine type that matches your performance requirements with a minimum of three vCPUs.

Deploying FortiGate-VM instances

Deploy two FortiGate-VM instances using any method that this document describes into two different zones of the same region. Both instances should have three network interfaces (NIC) connected to three subnets of three different virtual private cloud (VPC) networks:

- port1 – **SUBNET_EXT** (with a public IP address)
- port2 – **SUBNET_INT** (with no public IP address)
- port3 – **SUBNET_HASYNC** (optionally with public IP address for management)

Networking

Network interfaces

demo-vpc-ext demo-ext (10.0.1.0/24)	▼
demo-vpc-int demo-int (10.0.2.0/24)	▼
demo-vpc-hasync demo-hasync (10.0.3.0/24)	▼
ADD NETWORK INTERFACE	



You can use additional NICs to connect more networks. However, in a typical scenario, using VPC peering to connect to additional VPCs via a single internal NIC is recommended.



Remember to select a machine type with at least three vCPUs, such as n2-standard-4, to support three NICs.

The following provides example GCloud commands to deploy a primary FortiGate instance:

```
gcloud compute instance create example-vm-fgt1 --zone ZONE_1 \
--machine-type=e2-standard-4 \
--image-project=fortigcp-project-001 \
--image-family=fortigate-72-byol \
--can-ip-forward \
--network-interface="subnet=SUBNET_EXT" \
--network-interface="subnet=SUBNET_INT,no-address" \
--network-interface="subnet=SUBNET_HASYNC,private-network-ip=HA_IP_1" \
--service-account=SERVICE_ACCOUNT \
--scopes=cloud-platform
```

Reserving static addresses

Reserve static internal addresses for both FortiGate instances. A static internal IP address is required for the heartbeat interface (port3) and recommended for other NICs.

To reserve internal addresses:

1. In *Compute Engine > VM Instances*, click the instance name to open the instance details page.
2. Click *EDIT*.
3. In *Network interfaces*, open the network interface attributes by clicking the down arrow.
4. Open the *Primary internal IP* dropdown list and change the value from *Ephemeral* to *STATIC*.
5. Provide a name for the interface and click *RESERVE*.
6. Click *DONE* and *SAVE* to save the VM instance changes.
7. Repeat steps 3-6 for all NICs and for the second FortiGate.

Creating instance groups

Create two unmanaged instance groups in **ZONE_1** and **ZONE_2**. Each group will contain the FortiGate instance from the respective zone.

To create instance groups:

1. Go to *Compute Engine > Instance groups*.
2. Click **CREATE INSTANCE GROUP**.
3. Select *New unmanaged instance group*.
4. From the *Region* and *Zone* dropdown lists, select the region and zone where the primary FortiGate is deployed.
5. From the *Network* dropdown list, select the virtual private cloud hosting external subnet.
6. From the *Select VMs* dropdown list, select the FortiGate instance.
7. Repeat the steps 2-6 for the second zone and secondary FortiGate.

Creating the external LB



GCP web console does not support creating external load balancer (LB) frontends forwarding multiple protocols nor stateful failover options. For these advanced features, see [Deploying FortiGate-VM HA with external and internal LB \(GCloud CLI\) on page 81](#).

To create the external LB:

1. Go to *Network Services > Load balancing* and click **CREATE A LOAD BALANCER**.
2. In *TCP Load Balancing*, click **START CONFIGURATION**.
3. Leave all settings at their defaults and click **CONTINUE**.
4. Name your LB and select the region where FortiGates are running.
5. Configure the backend:
 - a. Under *Backends*, create a new backend. Select the first instance group and click **DONE**.
 - b. Click **ADD BACKEND** and add the second instance group to the backend list.
 - c. From the health check dropdown list, select *Create a health check* option.
 - d. Name your health check, configure it to use TCP protocol and port 8008, and click **SAVE**.
6. Switch to frontend configuration and change *Port* to *All*.

7. Click *CREATE*.

[← New TCP load balancer](#)

Name *	demo-elb	?
Lowercase, no spaces. Name is permanent		
Region *	us-west1 (Oregon)	?
<input checked="" type="checkbox"/> Backend configuration <input checked="" type="checkbox"/> Frontend configuration (i) Review and finalize (optional)		
CREATE CANCEL		

Frontend

Protocol ↑	IP version	IP:Port	Network Tier ?
TCP	IPv4	:all	Premium

Backend

Region	Endpoint protocol	Session affinity	Health check
us-west1	TCP	None	demo-healthcheck

▼ ADVANCED CONFIGURATIONS

Instance group ↑	IP stack type	Zone	Autoscaling	Use as failover group
demo-umig1	IPv4	us-west1-b	No configuration	No
demo-umig2	IPv4	us-west1-c	No configuration	No



Ensure that you allow connections from the Internet to the FortiGates by adding an appropriate cloud firewall rule in the external virtual private cloud network.

Creating the internal LB

To create the internal LB:

1. Go to *Network Services > Load balancing* and click *CREATE A LOAD BALANCER*.
2. In *TCP Load Balancing*, click *START CONFIGURATION*.
3. Change the *Internet-facing or internal only* option to *Only between my VMs* and click *CONTINUE*.
4. Name your load balancer (LB) and select the region where FortiGates are running.
5. From the *Network* dropdown list, select the INTERNAL_VPC network.
6. Configure the backend:
 - a. Under *Backends*, create a new backend. Select the first instance group and click *DONE*. Ignore the warning about the instance group having a different primary network.
 - b. Click *ADD BACKEND* and add the second instance group to the backend list.
 - c. From the health check dropdown list, select the health check that you created for the external LB.
7. Switch to frontend configuration and change *Port* to *All*.
8. From the *Subnetwork* dropdown list, select **SUBNET_INT**. Click *Done*.

9. Click *CREATE*.

New Internal load balancer

Protocol	IP version	Scope	Subnetwork	IP:Ports	Service label
TCP	IPv4	us-west1	demo-int	AUTOMATIC:all	

Frontend

Region	Network	Endpoint protocol	Session affinity	Health check
us-west1	demo-vpc-int	TCP	None	demo-healthcheck

Backend

Instance group	IP stack type	Zone	Autoscaling	Use as failover group
demo-umig1	IPv4	us-west1-b	No configuration	No
demo-umig2	IPv4	us-west1-c	No configuration	No

ADVANCED CONFIGURATIONS

The load balancer will forward traffic only to instances whose NICs are in network demo-vpc-int.

Review and finalize (optional)

CREATE **CANCEL**

Creating a custom route

To redirect traffic from the internal VPC network via FortiGates:

1. In the Google Cloud console, go to *VPC Networks*.
2. Click the name of your internal network.
3. On the *ROUTES* tab, delete the existing default route to the Internet by selecting it and clicking *DELETE*.
4. Click *ADD ROUTE*.
5. Provide a name for your custom route.
6. In the *Destination IP range* field, enter *0.0.0.0/0*.
7. From the *Next hop* dropdown list, select *Specify a forwarding rule of internal TCP/UDP load balancer*.
8. From the *Forwarding rule name* dropdown list, select the rule matching your internal load balancer.
9. Click *CREATE*.

Configure FortiGates networking

To configure FortiGates networking:

1. Log into both FortiGate instances. See the instructions in this document for single FortiGate deployment for instructions on initial login, licensing, and post-deployment steps.
2. Go to *Network > Static routes*.
3. Add a static route to *0.0.0.0/0* via the *port1* interface. Specify the first address in the external subnet as the gateway address. Ignore the warning that the gateway IP address may be unreachable.

The screenshot shows the FortiGate interface configuration screen. Under 'Destination' settings, a static route is being configured with a subnet of 0.0.0.0/0.0.0.0 and a gateway address of 10.0.1.1. A warning message is displayed: 'Gateway IP 10.0.1.1 could be unreachable. It is not in any subnet of the interface port1: • 10.0.1.2/32'. Below this, the 'Interface' section lists 'port1' with an administrative distance of 10. The 'Status' section shows 'Enabled'.

4. Add a static route to internal subnet via port2. Specify the first address in the internal subnet as the gateway address. Ignore the gateway reachability warning.
5. Add static routes to networks 35.191.0.0/16 and 130.211.0.0/22 via port2 with the first address of the internal subnet as the gateway address.
6. In *Network > Interfaces*, open port1 and port2 and change the *Addressing mode* from *DHCP* to *Manual*. The proper IP address and netmask are autopopulated.
7. In *Network > Interfaces*, open port3 and enable administrative (HTTPS and SSH) access.

Configuring FortiGate clustering



Currently you must configure FortiGate clustering protocol unicast clustering in the FortiOS CLI.

To configure FortiGate clustering:

1. Log in to the primary FortiGate.
2. In the CLI console, configure high availability:

```
config system ha
  set group-name "ha"
  set mode a-p
  set hbdev "port3" 50
  set session-pickup enable
  set ha-mgmt-status enable
  config ha-mgmt-interfaces
    edit 1
      set interface "port3"
      set gateway SUBNET_GW
```

```

        next
    end
    set override disable
    set priority 10
    set unicast-hb enable
    set unicast-hb-peerip PEER_IP
    set unicast-hb-netmask SUBNET_NETMASK_LONG
end

```

Replace the placeholders with the values for your deployment:

Placeholder	Value
SUBNET_GW	Gateway address (first IP address) for the heartbeat subnet.
PEER_IP	Secondary FortiGate internal IP address.
SUBNET_NETMASK_LONG	Heartbeat subnet mask in quad notation. For example, you could enter 255.255.255.0.

3. Repeat the same configuration on the secondary FortiGate, configuring the primary FortiGate internal IP address for PEER_IP and setting the priority to 5.
4. If the cluster is not built and FortiGates cannot connect to each other, ensure that the VPC network has a firewall rule allowing communication between FortiGate peers on the heartbeat network.



From this point, you should manage your FortiGate instances using the addresses associated with port3. The secondary FortiGate does not respond to requests on port1 when in passive mode.

Configure health check probe responders

To configure health check probe responders:

1. In Google Cloud console, go to *Network Services > Load balancing*.
2. Find the external and internal load balancers (LB) that you created and note their frontend IP addresses (public for external LB and private for internal LB).
3. In the FortiGate CLI, add a secondary IP address to port1, replacing ELB_FRONTEND with the external LB frontend public IP address:

```

config system interface
    edit port1
        set secondary-IP enable
        config secondaryip
            edit 0
            set ip ELB_FRONTEND/32
            set allowaccess probe-response
        next
    end
next
end

```

4. Repeat the same step for port2 and the internal LB frontend:

```
config system interface
  edit port2
    set secondary-IP enable
    config secondaryip
      edit 0
      set ip ILB_FRONTEND/32
      set allowaccess probe-response
    next
  end
next
end
```

5. Enable HTTP probe responses:

```
config system probe-response
  set mode http-probe
end
```

6. Enter `y` to confirm restarting the probe response daemon.

Best practices and next steps

Not publishing management interfaces on the public Internet is recommended. If using a public IP address, use cloud firewall rules to restrict access from trusted IP addresses ranges only.

By default, FortiGate-VM instances assigned to the Compute Engine default service account. Creating a dedicated service account and assigning it a custom role with minimum required permissions is recommended. See [Creating a GCP service account on page 98](#).

Deploying FortiGate-VM HA with external and internal LB (GCloud CLI)

Deploying FortiGates in a high availability (HA) cluster eliminates having a single point of failure and elevates Google Compute SLA to 99.98%. HA in a load balancer (LB) sandwich design features two FortiGate instances in an active-passive (A-P) cluster between a pair of GCP LBs ("LB sandwich" pattern) for fast and stateful failover.



This guide describes manually deploying the cluster. Fortinet publishes Terraform and deployment manager modules in its GitHub repositories. Consult documentation and examples in individual modules for automated deployments.

FGCP in public cloud

The FortiGate clustering protocol (FGCP) is a proprietary protocol used to create high availability clusters in hardware and virtual FortiGate deployments. Due to the way cloud networks work, you cannot take full advantage of the protocol

capabilities and must use its unicast version, limiting the functionality to an active-passive cluster of two instances.

FGCP provides automatic synchronization of connection tables as well as synchronization of configuration from the primary to the secondary instance. You must apply all configuration changes to the primary instance. Using the priority option to statically assign the primary and secondary roles in the cluster is recommended.

For a more detailed description of the architecture, see [GitHub](#).

The following provides the configuration steps for this deployment:

1. [Predeployment steps on page 82](#)
2. [Reserving internal addresses on page 83](#)
3. [Deploying FortiGate-VM instances on page 83](#)
4. [Creating instance groups on page 84](#)
5. [Creating the external LB on page 84](#)
6. [Creating the internal LB and custom route on page 85](#)
7. [Configure FortiGates networking on page 86](#)
8. [Configuring FortiGate clustering on page 87](#)
9. [Configure health check probe responders on page 88](#)
10. [Best practices and next steps on page 89](#)

Predeployment steps

To complete predeployment steps:

1. Decide the region, zones, licensing, firmware version, and machine type for the deployment.
2. Prepare three virtual private cloud (VPC) networks with non-overlapping subnets in the deployment region to use as external, internal, and heartbeat networks. You can optionally split the heartbeat and dedicated management network. This is obligatory for older firmware.

▼ demo-vpc-ext	1	1460	Custom	None
	us-west1	demo-ext		10.0.1.0/24
▼ demo-vpc-hasync	1	1460	Custom	None
	us-west1	demo-hasync		10.0.3.0/24
▼ demo-vpc-int	1	1460	Custom	None
	us-west1	demo-int		10.0.2.0/24

3. Identify the boot image to use to create instances. It can be a public or private image referred to by name, URL or family. See [Obtaining the deployment image on page 29](#).
4. If desired, complete the following predeployment table. This allows you to easily access necessary information during the deployment. Mentions of these fields are bolded in subsequent sections of this guide:

5. \$REGION	Region to deploy to. It should be the same region your servers are using.
\$ZONE_1	Primary availability zone (AZ). This zone must be in REGION.
\$ZONE_2	Secondary AZ. This zone must be in REGION.

\$VPC_EXT / \$SUBNET_EXT	External VPC and subnet names.
\$VPC_INT / \$SUBNET_INT	Internal VPC and subnet names.
\$VPC_HASYNC / \$SUBNET_HASYNC	Heartbeat and management VPC and subnet names.
\$IMAGE_NAME or \$IMAGE_FAMILY or \$IMAGE_URL	FortiGate image matching your licensing type and firmware version.
\$MACHINE_TYPE	Select the machine type that matches your performance requirements with a minimum of three vCPUs.

Reserving internal addresses

Reserve six static internal addresses for both FortiGate instances (three per instance). A static internal IP address is required for the heartbeat interface (port3) and recommended for other NICs. The following shows an example command to reserve a private address:

```
gcloud compute addresses create addr-fgt1-port1 --region=REGION \ --subnet=$SUBNET_EXT
```

Repeat this command six times for all three subnets and both FortiGate instances.

Deploying FortiGate-VM instances

Deploy two FortiGate-VM instances using any method that this document describes into two different zones of the same region. Both instances should have three network interfaces (NIC) connected to three subnets of three different virtual private cloud (VPC) networks:

- port1 – **SUBNET_EXT** (with a public IP address)
- port2 – **SUBNET_INT** (with no public IP address)
- port3 – **SUBNET_HASYNC** (optionally with public IP address for management)

If you have created a dedicated service account to be associated with FortiGates, you can indicate it during VM instances deployment.



You can use additional NICs to connect more networks. However, in a typical scenario, using VPC peering to connect to additional VPCs via a single internal NIC is recommended.



Remember to select a machine type with at least three vCPUs, such as n2-standard-4, to support three NICs.

The following provides example GCloud commands to deploy a primary FortiGate instance using a 7.6 bring your own license image:

```
gcloud compute instance create example-vm-fgt1 --zone ZONE_1 \
--machine-type=e2-standard-4 \
--image-project=fortigcp-project-001 \
--image-family=fortigate-74-byol \
--can-ip-forward \
--network-interface="subnet=SUBNET_EXT" \
--network-interface="subnet=SUBNET_INT,no-address" \
--network-interface="subnet=SUBNET_HASYNC,private-network-ip=HA_IP_1" \
--service-account=SERVICE_ACCOUNT \
--scopes=cloud-platform
```

Creating instance groups

Create two unmanaged instance groups in **ZONE_1** and **ZONE_2**. Each group will contain the FortiGate instance from the respective zone.

To create instance groups:

1. Create an unmanaged instance group: `gcloud compute instance-groups unmanaged create fgt-umig1 --zone=$ZONE1`
2. Add a VM instance to the instance group:

```
gcloud compute instance-groups unmanaged add-instances fgt-umig1 \
--instances=example-vm-fgt1 \
--zone=ZONE_1
```

3. Repeat steps 1 and 2 for the second zone.

Creating the external LB

Google Cloud load balancer (LB) is a set of multiple resources tied together to provide desired functionality. Some resources can be shared between external and internal LBs, while others cannot.

An external LB is responsible for sending packets between the Internet and the active FortiGate instance.

To create the external LB:

1. Create a shared health check:

```
gcloud compute health-checks create http fgt-hcheck-tcp8008 --region=$REGION \
--port=8008 \
--timeout=2s \
--healthy-threshold=1
```

2. Create a backend service and add instance groups to it:

```
gcloud compute backend-services create fgteleb-bes --region=$REGION \
--load-balancing-scheme=EXTERNAL \
--protocol=UNSPECIFIED \
--health-checks=fgt-hcheck-tcp8008 \
--health-checks-region=$REGION \
--connection-persistence-on-unhealthy-backends=NEVER_PERSISTS
gcloud compute backend-services add-backend fgteleb-bes --region=$REGION \
```

```
--instance-group=fgt-umig-$ZONE1_LABEL \
--instance-group-zone=$ZONE1
gcloud compute backend-services add-backend fgtelb-bes-$REGION_LABEL --region=$REGION \
--instance-group=fgt-umig-$ZONE2_LABEL \
--instance-group-zone=$ZONE2
```

3. Reserve a public IP address and create a forwarding rule:

```
gcloud compute addresses create fgtelb-serv1-eip-$REGION_LABEL --region=$REGION
gcloud compute forwarding-rules create fgtelb-serv1-fwd-$REGION_LABEL-13 --
region=$REGION \
--address=fgtelb-serv1-eip-$REGION_LABEL \
--ip-protocol=L3_DEFAULT \
--ports=ALL \
--load-balancing-scheme=EXTERNAL \
--backend-service=fgtelb-bes-$REGION_LABEL
```

4. Repeat step 3 if you need more public IP addresses attached to the cluster.

5. Enable connections to FortiGates using a cloud firewall ALLOW rule.

Creating the internal LB and custom route

The internal load balancer (LB) is used as the next hop for routing traffic originating from Google Cloud virtual private cloud networks to the active FortiGate instance.

To create the internal LB and custom route:

1. Create the internal backend service and add instance groups to it. You can reference the same health check as for the external LB:

```
gcloud compute backend-services create fgt-ilb-bes --region=$REGION \
--network=int-vpc \
--load-balancing-scheme=INTERNAL \
--health-checks=fgt-hcheck-tcp8008 \
--health-checks-region=$REGION \
--connection-persistence-on-unhealthy-backends=NEVER_PERSISTS
gcloud compute backend-services add-backend fgt-ilb-bes --region=$REGION \
--instance-group=fgt-umig1 \
--instance-group-zone=$ZONE1
gcloud compute backend-services add-backend fgt-ilb-bes --region=$REGION \
--instance-group=fgt-umig2 \
--instance-group-zone=$ZONE2
```

2. Create an internal forwarding rule in the internal subnet:

```
gcloud compute forwarding-rules create fgt-ilb-fwrule --region=$REGION \
--address=fgtilb-ip-int-$REGION_LABEL \
--ip-protocol=TCP \
--ports=ALL \
--load-balancing-scheme=INTERNAL \
--backend-service=fgtilb-int-bes-$REGION_LABEL \
--subnet=int-sb-$REGION_LABEL
```

3. Create the custom default route with forwarding rule as the next hop:

```
gcloud compute routes create rt-default-via-fgt \
--network=int-vpc \
--destination-range=0.0.0.0/0 \
--next-hop-ilb=fgt-ilb-fwrule \
--next-hop-ilb-region=$REGION \
--priority=10
```

Configure FortiGates networking

To configure FortiGates networking:

1. Log into both FortiGate instances. See the instructions in this document for single FortiGate deployment for instructions on initial login, licensing, and post-deployment steps.
2. Go to *Network > Static routes*.
3. Add a static route to 0.0.0.0/0 via the port1 interface. Specify the first address in the external subnet as the gateway address. Ignore the warning that the gateway IP address may be unreachable.

The screenshot shows the FortiGate configuration interface for adding a static route. The 'Destination' field is set to '0.0.0.0/0.0.0'. The 'Gateway Address' is set to '10.0.1.1' via 'Specify'. A warning message is displayed: 'Gateway IP 10.0.1.1 could be unreachable. It is not in any subnet of the interface port1: • 10.0.1.2/32'. The 'Interface' dropdown is set to 'port1'. Other settings include 'Administrative Distance' at 10, an empty 'Comments' field, and 'Status' set to 'Enabled'.

Automatic gateway retrieval	<input checked="" type="checkbox"/>
Destination	<input type="text"/> Subnet Internet Service 0.0.0.0/0.0.0.0
Gateway Address	Dynamic <input type="button" value="Specify"/> 10.0.1.1
<small>Gateway IP 10.0.1.1 could be unreachable. It is not in any subnet of the interface port1:</small> <small>• 10.0.1.2/32</small>	
Interface	<input type="button" value="port1"/> <input type="button" value="x"/> <input type="button" value="+"/>
Administrative Distance	<input type="text" value="10"/>
Comments	<input type="text" value="Write a comment..."/> 0/255
Status	<input type="button" value="Enabled"/> <input type="button" value="Disabled"/>

4. Add a static route to internal subnet via port2. Specify the first address in the internal subnet as the gateway address. Ignore the gateway reachability warning.

Automatic gateway retrieval  

Destination 

Subnet	Internet Service
10.0.2.0/24	
Gateway Address	10.0.2.1

Gateway IP 10.0.2.1 could be unreachable. It is not in any subnet of the interface port2:

- 10.0.2.2/32

Interface

 port2	
	

Administrative Distance 

Comments Write a comment... / 0/255

Status  Enabled  Disabled

- Add static routes to networks 35.191.0.0/16 and 130.211.0.0/22 via port2 with the first address of the internal subnet as the gateway address.

    Search 			
Destination	Gateway IP	Interface	Status
0.0.0.0/0	 10.0.1.1	 port1	 Enabled
10.0.2.0/24	 10.0.2.1	 port2	 Enabled
35.191.0.0/16	 10.0.2.1	 port2	 Enabled
130.211.0.0/22	 10.0.2.1	 port2	 Enabled

- In *Network > Interfaces*, open port1 and port2 and change the *Addressing mode* from *DHCP* to *Manual*. The proper IP address and netmask are autopopulated.
- In *Network > Interfaces*, open port3 and enable administrative (HTTPS and SSH) access.

Configuring FortiGate clustering



Currently you must configure FortiGate clustering protocol unicast clustering in the FortiOS CLI.

To configure FortiGate clustering:

- Log in to the primary FortiGate.
- In the CLI console, configure high availability:

```
config system ha
    set group-name "ha"
```

```

set mode a-p
set hbdev "port3" 50
set session-pickup enable
set ha-mgmt-status enable
config ha-mgmt-interfaces
    edit 1
        set interface "port3"
        set gateway SUBNET_GW
    next
end
set override disable
set priority 10
set unicast-hb enable
set unicast-hb-peerip PEER_IP
set unicast-hb-netmask SUBNET_NETMASK_LONG
end

```

Replace the placeholders with the values for your deployment:

Placeholder	Value
SUBNET_GW	Gateway address (first IP address) for the heartbeat subnet.
PEER_IP	Secondary FortiGate internal IP address.
SUBNET_NETMASK_LONG	Heartbeat subnet mask in quad notation. For example, you could enter 255.255.255.0.

3. Repeat the same configuration on the secondary FortiGate, configuring the primary FortiGate internal IP address for PEER_IP and setting the priority to 5.
4. If the cluster is not built and FortiGates cannot connect to each other, ensure that the VPC network has a firewall rule allowing communication between FortiGate peers on the heartbeat network.



From this point, you should manage your FortiGate instances using the addresses associated with port3. The secondary FortiGate does not respond to requests on port1 when in passive mode.

Configure health check probe responders

To configure health check probe responders:

1. In Google Cloud console, go to *Network Services > Load balancing*.
2. Find the external and internal load balancers (LB) that you created and note their frontend IP addresses (public for external LB and private for internal LB).
3. In the FortiGate CLI, add a secondary IP address to port1, replacing ELB_FRONTEND with the external LB frontend public IP address:

```

config system interface
edit port1
set secondary-IP enable

```

```
config secondaryip
    edit 0
    set ip ELB_FRONTEND/32
    set allowaccess probe-response
    next
end
next
end
```

4. Repeat the same step for port2 and the internal LB frontend:

```
config system interface
    edit port2
    set secondary-IP enable
    config secondaryip
        edit 0
        set ip ILB_FRONTEND/32
        set allowaccess probe-response
        next
    end
    next
end
```

5. Enable HTTP probe responses:

```
config system probe-response
    set mode http-probe
end
```

6. Enter *y* to confirm restarting the probe response daemon.

Best practices and next steps

Not publishing management interfaces on the public Internet is recommended. If using a public IP address, use cloud firewall rules to restrict access from trusted IP addresses ranges only.

By default, FortiGate-VM instances assigned to the Compute Engine default service account. Creating a dedicated service account and assigning it a custom role with minimum required permissions is recommended. See [Creating a GCP service account on page 98](#).

Additional documentation

See:

- [gcloud CLI overview](#)
- [GCP cloud shell](#)

GCP IPv6 route support with nextHopInstance

FortiOS 7.4.4 and later versions support IPv6 high availability (HA) routes. GCP IPv6 routes support the nextHopInstance attribute. When an HA active-passive failover occurs, the FortiGate-VM uses the nextHopInstance attribute to update IPv6 routes.



HA configuration on IPv6 is the same as IPv4 except for the key points that this document outlines in the following section.

The following lists prerequisites for using IPv6 HA routes:

- The following GCP elements must already exist. Otherwise, you must create them:
 - IPv6 networks
 - IPv6 route table for the internal subnet
- Two GCP FortiGate-VMs running FortiOS 7.4.4 installed and licensed with HA configuration working and in-sync

To add an IPv6 route to the SDN connector on both FortiGate HA nodes:

In the following example, "route-internal" is a nextHopIP route that is the IPv4 route on GCP routing table. "route-internal-ipv6" is a nextHopInstance route that is the IPv6 route on GCP routing table.

```
config system sdn-connector
  edit "gcp_conn"
    config route
      edit "route-internal"
        next
        edit "route-internal-ipv6"
        next
      end
    end
end
```

The FortiGate-VM detects the route table type and decides which route table attribute to use upon failover.

SDN connector integration with GCP

This guide describes configuring GCP SDN connector on FortiGate-VM for GCP.

The following summarizes minimum sufficient roles for this deployment:

- Compute Viewer
- Kubernetes Engine Viewer

You can also configure pipelined automation. See [Pipelined automation using Google Cloud function on page 104](#).

Configuring GCP SDN connector using metadata IAM

To populate dynamic objects, the FortiGate-VM must have API access to required resources on the Google Cloud Compute Engine.

To configure GCP SDN connector using metadata IAM:

1. In FortiOS, go to *Security Fabric > Fabric Connectors*.
2. Click *Create New*, and select *Google Cloud Platform (GCP)*. You can create only one SDN connector per connector type. For example, you can create one entry for GCP.
3. Configure the connector as follows:
 - a. **Name:** Enter the desired connector name.
 - b. Enable *Use metadata IAM*. The Google platform requires a certain authentication level to call APIs from the FortiGate. See [To check metadata API access: on page 93](#). The *Use metadata IAM* option is only available to FortiGate-VMs running on GCP. FortiGates running outside of GCP (including physical FortiGate units and FortiGate-VMs running on other cloud platforms) have a configuration that is equivalent to disabling this option.
 - c. **Update interval:** the default value is 60 seconds. You can enter a value between 1 and 3600 seconds.
 - d. **Status:** Green means that the connector is enabled. You can disable it at any time by toggling the switch.



4. Create a dynamic firewall address for the configured GCP SDN connector:
 - a. Go to *Policy & Objects > Addresses*. Click *Create New*, then select *Address*.
 - b. Configure the address:
 - i. **Name:** Enter the desired name.
 - ii. **Type:** Select *Fabric Connector Address*.
 - iii. **Fabric Connector Type:** Select *Google Cloud Platform (GCP)*.

- iv. Filter:** This means the SDN Connector automatically populates and updates only instances belonging to the specified VPN that match this filtering condition. Currently GCP supports the following filters:
- i. `id=<instance id>` : This matches an VM instance ID.
 - ii. `name=<instance name>` : This matches a VM instance name.
 - iii. `zone=<gcp zones>` : This matches a zone name.
 - iv. `network=<gcp network name>` : This matches a network name.
 - v. `subnet=<gcp subnet name>` : This matches a subnet name.
 - vi. `tag=<gcp network tags>` : This matches a network tag.
 - vii. `label.<gcp label key>=<gcp label value>` : This matches a free form GCP label key and its value.

The example configuration populates all IP addresses that belong to the default network in the zone us-central-1f.

Note that wildcards (such as the asterisk) are not allowed in filter values.

- v. Click OK.**

The address has been created. Wait for a few minutes before the setting takes effect. You will know that the address is in effect when the exclamation mark disappears from the address entry. When you hover over the address, you can see the list of populated IP addresses.

If the exclamation mark does not disappear, check the address settings.

To check metadata API access:

1. On the GCP Compute Engine, go to the FortiGate-VM.

The screenshot shows the 'VM instance details' page for a VM named 'jkatofgt603-fortigate-4-vm'. The left sidebar lists various Compute Engine resources: VM instances, Instance groups, Instance templates, Sole tenant nodes, Disks, Snapshots, Images, TPUs, Committed use discounts, Metadata, Health checks, Zones, Network endpoint groups, Operations, Quotas, and Marketplace. The main panel displays the following details:

- Details** tab selected.
- Remote access**: SSH is selected, and 'Connect to serial console' is also available.
- Logs**: Stackdriver Logging, Serial port 1 (console), and More.
- Machine type**: n1-standard-1 (1 vCPU, 3.75 GB memory).
- CPU platform**: Intel Broadwell.
- Zone**: us-west1-b.
- Labels**: goog-dm : jkatofgt60... (with a tooltip for 'jkatofgt60...').
- Creation time**: Dec 8, 2018, 11:19:09 PM.
- Network interfaces**: Name, Network, Subnetwork, Primary internal IP.

2. Scroll down to *Cloud API Access Scopes* and check the Compute Engine configuration. If Compute Engine is disabled, you must enable it:
 - a. Stop the VM.
 - b. Once the VM is completely stopped, click *Edit*.
 - c. From the *Compute Engine* dropdown list, select *Read/Write access*.
 - d. Save the change, then restart the VM.

GCP Kubernetes SDN connector

GCP SDN connectors support dynamic address groups based on GCP Kubernetes Engine filters. See the [FortiOS Administration Guide](#).

Configuring GCP SDN Connector using service account

See the [FortiOS Administration Guide](#).

Custom role permission guideline

The following provides the least privileged guideline for a custom role when using a GCP SDN connector with a service account for high availability (HA):

- compute.addresses.get
- compute.addresses.use
- compute.instances.addAccessConfig
- compute.instances.deleteAccessConfig
- compute.instances.get
- compute.instances.list
- compute.instances.updateNetworkInterface
- compute.networks.updatePolicy
- compute.networks.useExternalIp
- compute.subnetworks.use
- compute.subnetworks.useExternalIp
- compute.routes.create
- compute.routes.delete
- compute.routes.get
- compute.routes.list



This list is a guideline and focuses on the operation of HA between two FortiGate-VMs in a single zone and multizone deployment only. It allows for moving a single public IP address from the primary FortiGate to the secondary and updating the referenced GCP routing table in the FortiOS SDN connector configuration. Your custom role Identity and Access Management (IAM) permissions vary depending on your environment.



The predefined compute admin role includes the aforementioned IAM permissions. See [IAM permissions reference](#).

API calls

The SDN connector uses API calls to GCP API endpoints respective to its function. You can review the methods, calls, and error codes by using the following diagnostics commands:

Command	Description
diagnose debug reset	Clears filters or previous diagnostic configuration in the console or SSH session.
diagnose debug console timestamp enable	Enables timestamp of console output messages.
diagnose debug enable	Enables diagnostic output to the console.
diagnose debug application gcpd -1	Selects the GCP daemon or SDN connector.



For information about creating a GCP SDN connector, see [GCP SDN connector using service account](#).

The following are references for running a VM with a service account:

- [Creating and enabling service accounts for instances](#)
- [Permissions required for this task](#)

Multiple GCP projects in a single SDN connector

An option is added to specify multiple projects under a single GCP SDN connector. Previously, FortiOS allowed only one project per SDN connector, which limits the total projects to the number of SDN connectors (256). This enhancement also allows dynamic firewall address filters to filter on a project.

This example configures a GCP SDN connector (gcp_conn) with two projects. The example configures the first project, dev-project-001-166400, using the simple format. The example configures the second project, dev-project-002, using the advanced format.

To configure a GCP connector with multiple projects in the GUI:

1. Go to *Security Fabric > External Connectors* and click *Create New*.
2. Select *Google Cloud Platform (GCP)* and enter a name for the connector.
3. Configure the first project:
 - a. For *Projects*, select *Simple*.
 - b. Enter the project name, service account email, and private key.
4. Configure the second project:
 - a. For *Projects*, select *Advanced* (the projects are now displayed in a table) and click *Create New*. The *Add GCP Project* pane opens.
 - b. Enter a name.
 - c. Optionally, click the + to enter zones. If you do not select a zone, the SDN connector includes all zones. This example uses the *us-central1-a* zone.
 - d. Click *OK*.
5. Click *OK* to save the SDN connector.
6. Create a dynamic firewall address for the first project:
 - a. Go to *Policy & Objects > Addresses* and click *Create New > Address*.
 - b. Enter the following:

Name	project1_addresses
Type	Dynamic
Sub Type	Fabric Connector Address
SDN Connector	gcp_conn
Filter	Add a filter for the project, <i>Project=dev-project-001-166400</i> .

In this example, there are several instances for the first project, so add a filter for the ID, `Id=6266132824476267466`.
Change the logic operator to `and`.

- c. Click **OK**.
- 7. Create a dynamic firewall address for the second project:
 - a. Click **Create New > Address**.
 - b. Enter the following:

Name	project2_addresses
Type	Dynamic
Sub Type	Fabric Connector Address
SDN Connector	gcp_conn
Filter	Add a filter for the project, <code>Project=dev-project-002</code> .

- c. Click **OK**.
- The addresses have been created. Wait for a few minutes before the settings take effect.
- 8. Verify that the address resolve to the correct addresses. Hover over the address in the table to view the list of populated IP addresses.

To configure a GCP connector with multiple projects in the CLI:

1. Configure the SDN connector:

```
config system sdn-connector
  edit "gcp_conn"
    set status enable
    set type gcp
    config gcp-project-list
      edit "dev-project-001-166400"
        next
      edit "dev-project-002"
        set gcp-zone-list "us-central1-a"
        next
    end
    set service-account "xxxxxxxxxxxxx-compute@developer.gserviceaccount.com"
    set private-key *****
    set update-interval 30
  next
end
```

2. Create a dynamic firewall address for project one:

```
config firewall address
  edit "project1_addresses"
    set type dynamic
    set sdn "gcp_conn"
    set filter "Project=dev-project-001-166400 & Id=6266132824476267466"
  next
end
```

The dynamic firewall address IP is resolved by the SDN connector:

```
config firewall address
    edit "project1_addresses"
        show
            config firewall address
                edit "project1_addresses"
                    set uuid 38efbd88-fb08-51eb-8e6d-9b78a2a9bf49
                    set type dynamic
                    set sdn "gcp_conn"
                    set filter "Project=dev-project-001-166400 & Id=6266132824476267466"
                    config list
                        edit "172.16.16.3"
                        next
                        edit "172.16.24.3"
                        next
                        edit "172.16.8.4"
                        next
                    end
                end
            end
        end
    end
next
```

3. Create a dynamic firewall address for project two:

```
config firewall address
    edit "project2_addresses"
        set type dynamic
        set sdn "gcp_conn"
        set filter "Project=dev-project-002"
        set sdn-addr-type all
    next
end
```

The dynamic firewall address IP is resolved by the SDN connector:

```
config firewall address
    edit "project2_addresses"
        show
            config firewall address
                edit "project2_addresses"
                    set uuid 5ca9b2ba-fb08-51eb-57c0-12701b3d33c1
                    set type dynamic
                    set sdn "gcp_conn"
                    set filter "Project=dev-project-002"
                    set sdn-addr-type all
                    config list
                        edit "10.128.0.2"
                        next
                        edit "34.66.35.241"
                        next
                    end
                end
            end
        end
    end
next
```

Creating a GCP service account

This topic describes how to create a GCP service account and an API key pair, and provides guidelines on how to edit the private key for use in FortiOS. If you enabled metadata Identity and Access Management (IAM) in [Configuring GCP SDN Connector using service account on page 93](#), you do not need to create a service account.

To create a GCP service account:

1. Log into the GCP Compute Portal.
2. Go to *IAM & admin > Service accounts*.
3. Create a service account:
 - a. Select *Create a service account*.
 - b. Name the account.

- c. Click *CREATE* and *CONTINUE*.

Create service account

1 Service account details

Service account name
example-service-account

Display name for this service account

Service account ID *
example-service-account X C

Email address: example-service-account@dev-project-001-166400.iam.gserviceaccount.com ✉

Service account description
example-service-account

Describe what this service account will do

CREATE AND CONTINUE

2 Grant this service account access to project (optional)

3 Grant users access to this service account (optional)

DONE **CANCEL**

- d. From the *Role* dropdown list, select the desired role, then click *CONTINUE* or *DONE*.

Create service account

1 Service account details

2 Grant this service account access to project (optional)

Grant this service account access to Dev Project 001 so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role	fgt-ha-role	Condition Add condition
+ ADD ANOTHER ROLE		
CONTINUE		

3 Grant users access to this service account (optional)

DONE **CANCEL**



This example selects a custom role for high availability (HA). You can select the viewer role or another role if the FortiGate is on-premise or you do not need to configure HA.

- e. If you are configuring the service account for use in an SDN connector for HA or for running the VM, select the correct IAM role with the needed permissions.



For guidelines on the IAM role permissions for HA, see [Configuring GCP SDN Connector using service account on page 93](#).

For information about configuring a GCP IAM service account, see [Creating and managing service accounts](#).

- f. (Optional) Configure user access.

To create the service account key:

1. Edit the service account by selecting its email address.
2. On the Keys tab, click ADD KEY.

The screenshot shows the Google Cloud IAM & Admin interface. On the left sidebar, under the 'Service Accounts' section, the 'example-service-account' is selected. The main view shows the 'example-service-account' details with the 'KEYS' tab active. A modal window titled 'ADD KEY' is open, offering two options: 'Create new key' and 'Upload existing key'. A warning message at the top of the modal reads: '⚠ Service account keys could pose a security risk if compromised. We recommend using organization policies to manage service account keys.' Below the modal, there are links to 'organization policies' and 'Learn more about setting organization policies for service accounts'.

3. Select to import your existing key or generate another. If you create a new key, you can select a JSON formatted key or a P12, which includes the private and public keys. Once created, the key automatically downloads to your PC.

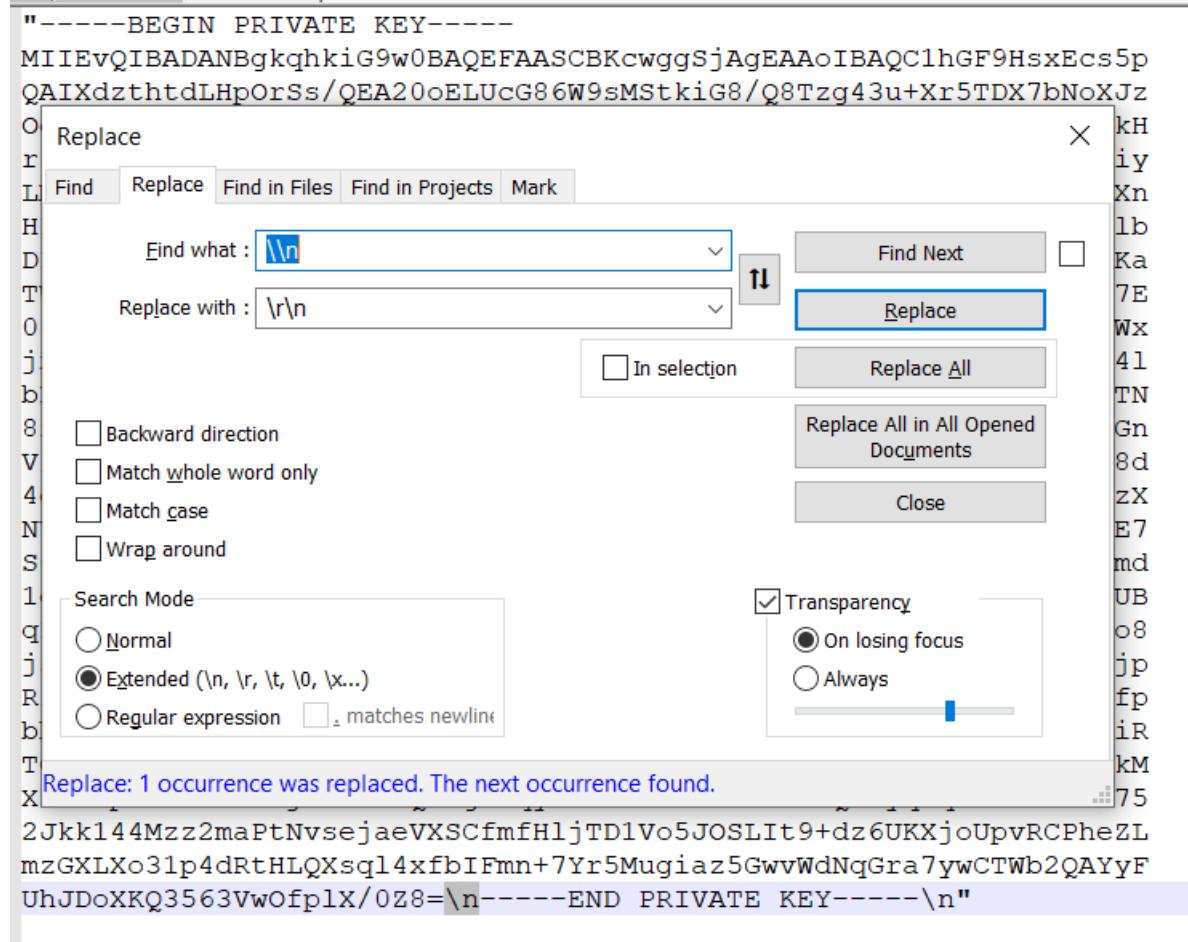


For information about creating service account keys, see [Create and manage service account keys](#).

To edit the private key:

1. Use a text editor to open the downloaded key.
2. Find the line ""private_key": "-----BEGIN PRIVATE KEY-----\n....."
3. Edit the key between "-----BEGIN PRIVATE KEY-----" and "-----END PRIVATE KEY-----".
4. Remove "\n" using a tool or command of your choice, for example by using the Find and Replace function in

Notepad++.



This replaces "\n" with the actual return line, rendering a correctly formatted private key.

5. Copy and paste the key content into the FortiOS GUI or CLI:

```
FortiWiFi-60E (gcp-connector-test) # set private-key "-----BEGIN PRIVATE KEY-----  
> MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggsjAgEAAoIBAQC1hGF9HsxEcs5p  
> QATVd++L+U-0S-OFAC0-EU-0C4W0-MO4U+GQ/00T--40...Y-ETDX7LNUyJz  
> Oe :H  
> rE :y  
> LM :n  
> HF :b  
> DF :a  
> TV :E  
> 0E :x  
> j> :l  
> bF :N  
> 8r :n  
> V/ :d  
> 4e :X  
> NV :7  
> Se :d  
> 1e :B  
> qF :8  
> js :p  
> Rf :p  
> bt :R  
> TC :M  
> Xz :5  
> 2. :L  
> MZuALau3 i...q4t4x1D1PmM17 / 11 Smug1az36wvWuNqG1 a/ywG1WbZQAtyF  
> UhJDoxKQ3563Vw0fp1X/0Z8=  
> -----END PRIVATE KEY-----  
> "
```

Troubleshooting GCP SDN Connector

You can check if API calls are made successfully by running the following commands in the CLI:

```
diagnose debug enable  
diagnose debug application gcpd -1
```

```
FGVM01TM18000516 # diagnose debug enable  
FGVM01TM18000516 # diagnose debug application gcpd -1  
Debug messages will be on for 30 minutes.
```

Wait a few minutes for the output. If the SDN connector was configured successfully, the API status shows 200 in communicating with the Google Cloud API server as shown. The host looks different depending on where you run the FortiGate instance (on or outside of GCP).

```
FGVM01TM18000517 (global) # diag debug enable
FGVM01TM18000517 (global) # diagnose debug application gcpd -1
Debug messages will be on for 30 minutes.

FGVM01TM18000517 (global) #
FGVM01TM18000517 (global) # gcpd api url: https://www.googleapis.com/compute/v1
host:www.googleapis.com:443:172.217.8.170
gcpd api result:200
host:www.googleapis.com:443:172.217.8.170
gcpd get instance list successfully
gcpd checking firewall address object jkatogcp001, vd 0
```

```
FGVM01TM18000516 # diagnose debug application gcpd -1
Debug messages will be on for 30 minutes.

FGVM01TM18000516 # gcpd exit
Unknown action 0

FGVM01TM18000516 #
FGVM01TM18000516 #
FGVM01TM18000516 # safeguard_fn() -1701
sync account, url: http://169.254.169.254/computeMetadata/v1/?alt=json&
gcpd api url: https://www.googleapis.com/compute/v1/projects/dev-project
host:www.googleapis.com:443:74.125.20.95
curl socket:11 vfid:0
https
{
  "error": {
    "errors": [
      {
        "domain": "global",
        "reason": "insufficientPermissions",
        "message": "Insufficient Permission"
      }
    ],
    "code": 403,
    "message": "Insufficient Permission"
  }
}

gcpd api result:403
gcpd get zones list failed
sync account, url: http://169.254.169.254/computeMetadata/v1/?alt=json&
```

If the CLI shows a failure, check the following and see if any required configuration is missing or incorrect:

- If using metadata IAM, can the FortiGate-VM access the API on Google Cloud Compute Engine?
- If the service account is specified:
 - Is the project name correct?
 - Is the service account email address correct?
 - Is the service account key correct?
 - Does the service account have the appropriate role/permissions?

Pipelined automation using Google Cloud function

See [GitHub](#).

Deploying autoscaling on GCP

You can deploy FortiGate virtual machines (VMs) to support autoscaling on Google Cloud Platform (GCP).

Multiple FortiGate-VM instances can form an autoscaling group to provide highly efficient clustering at times of high workloads. FortiGate-VM instances scale out automatically according to predefined workload levels. This deployment achieves autoscaling by using FortiGate-native high availability (HA) features such as `config-sync`, which synchronizes operating system configurations across multiple FortiGate-VM instances at the time of scaleout events.

FortiGate autoscale for GCP is available for on-demand (pay as you go) instances.

The standard deployment contains the following:

- HA architecture that spans two availability zones
- Virtual private cloud configured with public and private subnets
- Cloud NAT
- External-facing network load balancer (LB)
- Internal-facing network LB
- Cloud Functions, which runs Fortinet-provided scripts for running autoscaling. Functions are used to handle cluster creation and failover management.
- Firestore database which stores autoscaling configuration such as primary and secondary IP addresses. Firestore is a nosql database hosted on GCP.
- Managed instance group and instance template

Requirements

Installing and configuring FortiGate autoscale for GCP requires knowledge of the following:

- Configuring a FortiGate using the command line interface
- GCP
- Terraform 0.12

It is expected that DevOps engineers or advanced system administrators who are familiar with the aforementioned items will deploy FortiGate autoscale for GCP.

Account permissions

The default Compute service account should have sufficient Identity & Access Management permissions to deploy the cluster using Terraform. See [Access control for organization resources with IAM](#).

Region requirements

To deploy FortiGate autoscale for GCP, the region must support the following:

- Firestore
- Google bucket storage
- Cloud Functions
- Managed instance groups
- Cloud NAT

Deploying FortiGate autoscale for GCP

The easiest way to deploy FortiGate autoscale for GCP is with Terraform.

This deployment was tested with:

- Terraform 0.12
- Terraform Google Provider 2.20.1
- Terraform Google Provider Beta 2.20.1

To deploy FortiGate Autoscale for GCP:

1. Log in to your GCP account.
2. If you have not already done so, create an authentication token. The default Compute service account should have sufficient permissions. See [Authenticate for using client libraries](#).
3. Install Terraform. See [Install Terraform](#).
4. Clone the repository.
5. Change into the new directory and do one of the following:
 - Run the following commands:

```
npm install
npm run setup
```
 - Go to the [FortiGate Autoscale for GCP GitHub project release page](#) and download the latest gcp.zip from the releases tab. Create a folder named `dist` and place the gcp.zip file in that directory.

The following files and folders should be present:

```
.
├── assets
│   └── configset
│       ├── baseconfig
│       ├── httproutingpolicy
│       ├── httpsroutingpolicy
│       ├── internalelbweb
│       ├── port2config
│       ├── setuptgwvpn
│       └── storelogtofaz
├── cloud-function-package.json
└── dist
    └── gcp.zip
├── index.ts
├── main.tf
├── package.json
├── package-lock.json
├── README.md
└── tsconfig.json
```

```
└── tslint.json  
└── vars.tf
```

6. Open the vars.tf file and add values to the following variables:

Variable	Value
project	Google project ID
service_account	Service account that you will use to call Cloud Function
auth_key	GCP authentication key name and path. The default is account.json. Specify the path if the key is not in the current directory.

You can also do the aforementioned step from the command line using the following syntax:

```
terraform plan -var "<var_name>=<value>"
```

7. Customize other variables such as `cpu_ultilization` and `cooldown` period as needed. See [Terraform variables on page 108](#).

8. Initialize the providers and modules:

```
terraform init
```

9. Verify the plan:

```
terraform plan
```

10. Confirm and apply the plan:

```
terraform apply
```

Output is similar to the following. A randomly generated five-letter suffix is added to all resources. You can use the suffix to help identify your cluster resources.

```
InstanceTemplate = fortigateautoscale-instance-template-cehpm
LoadBalance_instances = []
LoadBalancer_Ip_Address = xxx.xxx.xxx.xxx
Notes = The Firestore Database must be deleted separately
Trigger_URL = https://us-central1-*****.cloudfunctions.net/fortigateautoscale-cehpm
google_compute_region_instance_group_manager = fortigateautoscale-fortigate-autoscale-cehpm
```



As part of the deployment, Terraform adjusts the `${fgt_secondary_ip}` value within the `baseconfig` file located in `assets/configset/`. The value should be the external load balancer IP address. See [Backend service-based external passthrough Network Load Balancer overview](#).

Quotas

FortiGate autoscale for GCP heavily uses Firestore to store cluster information. You can find Firestore quota information under [App Engine > Quotas](#). See [Quotas and limits](#).

You can adjust daily spending limits under [App Engine > Application settings](#).

See [Firestore pricing](#).

Terraform variables

The following table lists variables in the `vars.tf` file. You can change the values to suit your cluster needs.

Resource	Default	Description
project	Requires input	Project under which you will deploy the instance group. See Creating and managing projects .
auth_key	Requires input	File name of the authentication key you use to connect to GCP. See Adding credentials .
service_account	Requires input	Service account that you will use to call Cloud Functions. This allows Cloud Functions to be restricted to authorized calls.
region	us-central1	GCP region
zone	us-central1-c	GCP zone
nodejs_version	nodejs10	Node.js version to use in Cloud Functions

Resource	Default	Description
max_replicas	3	Maximum number of FortiGate-VM instances in the instance group. See Instance groups .
min_replicas	2	Minimum number of FortiGate-VM instances in the instance group.
cpu_utilization	0.5	Target CPU usage for the cluster to achieve. Instances scale out or in to meet this target.
		 Autoscaling is based on CPU utilization. This deployment does not support autoscaling using custom metrics.
cluster_name	FortigateAutoScale	Cluster name to use across objects (buckets, VPC, and so on)
bucket_name	fortigateautoscale	Blob storage bucket name
fortigate_image	projects/fortigcp-project-001/global/images/fortinet-fgtondemand-623-20191223-001-w-license	Source image for the instance group to use. The default image is FortiOS 6.2.3.
instance	n1-standard-1	Instance family type for the scaling configuration to use.
vpc_cidr	172.16.0.0/16	Classless inter-domain routing (CIDR) block for the FortiGate autoscale VPC divided into two /21 subnets.
public_subnet	172.16.0.0/21	Public subnet that the FortiGate cluster uses.
protected_subnet	172.16.8.0/21	Private subnet for VMs behind the FortiGate cluster.
firewall_allowed_range	0.0.0.0/0	GCP firewall range to allow.

Resource	Default	Description
		 • The default is to allow all. • If you use the GCP firewall policy to block incoming traffic, you must allow the load balancer to perform health checks and send data. For details on the IP addresses that will need access, see Probe IP ranges and firewall rules .
target_size	2	Autoscale cluster target size. See Autoscaling groups of instances .
SCRIPT_TIMEOUT	500	Timeout (in seconds) of a Cloud Functions invocation.
MASTER_ELECTION_TIMEOUT	400	Maximum time in seconds to wait for a primary election to complete. This variable should be less than the total script timeout (SCRIPT_TIMEOUT).
FORTIGATE_ADMIN_PORT	8443	Port number for FortiGate-VM administration. Do not use the FortiGate reserved ports 443, 541, 514, or 703. This value must be between 1 and 65535. This variable was previously used for the FortiGate Autoscale Cluster admin port.
HEARTBEAT_INTERVAL	25	Length of time in seconds that a FortiGate-VM waits between sending heartbeat requests to the function.
HEART_BEAT_DELAY_ALLOWANCE	10	Allowed variance in seconds before a heartbeat is considered out-of-sync and heartbeat loss is increased.
HEART_BEAT_LOSS_COUNT	10	Number of consecutively lost heartbeats. When the heartbeat loss count has been reached, the FortiGate-VM is deemed unhealthy and failover activities commence.

You can reference variables from the command line using the following:

```
terraform plan -var "<var name>=<value>"
```

Deployment information

Terraform deploys the following resources:

- VPC with two subnets split over two zones. You can choose more if the region supports it.
- Cloud NAT for egress traffic in the protected subnet
- Instance group
- Instance template
- Regional Autoscaler (autoscaling policy)
- Google Storage bucket
 - Template uploaded to the bucket at `assets/configset/baseconfig`
- Google Compute Function with an [HTTP trigger](#)
- Two GCP firewall rules: *Allow all* and *Allow only internal connections*
- An external-facing TCP network load balancer (LB)
- An internal LB

Additionally, the function creates a [Firestore](#) collection. The collection is not created during the Terraform deployment phase.

Verifying the deployment

To verify the deployment:

1. Log in to the GCP console and go to [Firestore](#).
2. Go to the *FortiGateMasterElection* table.
3. Note the primary FortiGate-VM IP address and ensure *VoteState* is *done*:

fortigateautoscale-fortigateautoscale-rnmlo	FORTIGATEMASTERELECTION
+ ADD DOCUMENT	+ START COLLECTION
FORTIANALYZER	+ ADD FIELD
FORTIGATEAUTOSCALE	▼ masterRecord
⋮ FORTIGATEMASTERELECTION >	InstanceId: "7723829953355373558" MasterIP: "172.16.0.3" SubnetId: "null" VoteState: "done" VpcId: "empty" voteEndTime: 1575577057464
LIFECYCLEITEM	
SETTINGS	

4. Go to the *FortiGateAutoscale* table and confirm that instances have been added to the cluster. Following is an example of a healthy cluster:

<p>fortigateautoscale-fortigateautoscale-rnmlo </p> <p>+ ADD DOCUMENT</p> <p>FORTIANALYZER</p> <p> FORTIGATEAUTOSCALE </p> <p>FORTIGATEMASTERELECTION</p> <p>LIFECYCLEITEM</p> <p>SETTINGS</p>	<p>FORTIGATEAUTOSCALE </p> <p>+ START COLLECTION</p> <p>+ ADD FIELD</p> <p>▼ 5075870911861937758</p> <pre>healthy: true heartBeatInterval: 25 heartBeatLossCount: "0" inSync: true instanceId: "5075870911861937758" ip: "172.16.0.7" masterIp: "172.16.0.3" nextHeartBeatTime: 1575580740272 syncState: "in-sync"</pre> <p>▼ 7244177209853008860</p> <pre>healthy: true heartBeatInterval: 25 heartBeatLossCount: "0" inSync: true instanceId: "7244177209853008860" ip: "172.16.0.4" masterIp: "172.16.0.3" nextHeartBeatTime: 1575580747567 syncState: "in-sync"</pre> <p>▼ 7723829953355373558</p> <pre>healthy: true heartBeatInterval: 25 heartBeatLossCount: "0" inSync: true instanceId: "7723829953355373558" ip: "172.16.0.3" masterIp: "172.16.0.3" nextHeartBeatTime: 1575580745865 syncState: "in-sync"</pre>
--	--



The *masterIp* field displays the primary FortiGate-VM IP address.
When an instance is removed from a cluster, its record is not deleted.

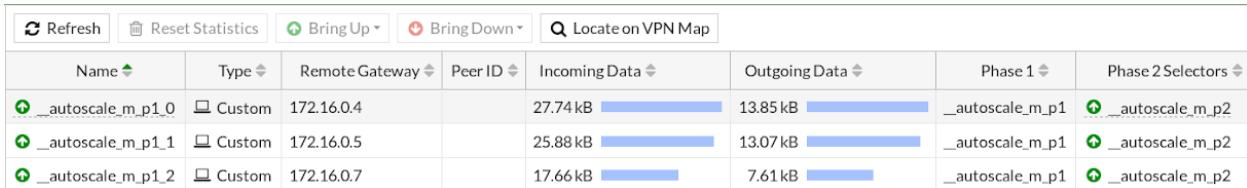
Verifying the instance group

To verify the instance group:

1. Log in to the primary FortiGate-VM instance using the public IP address from step 3 of [To verify the deployment: on page 111](#). The default admin port is 8443 and the default username/password is `admin/<instance-id>`. Cluster information displays on the main dashboard:

Virtual Machine	
Allocated vCPUs	2
Allocated RAM	4 GiB
Auto Scaling	<input checked="" type="checkbox"/> Enabled
Role	Master
Group Size	2

2. Go to *Dashboard > Network* and expand the *IPsec* widget to verify VPN status. You can view the current connections between the FortiGates in the cluster.



3. You can find additional settings in the *Firebase* collection under *SETTINGS*:

fortigateautoscale-fortigateautoscale-rnmlo	SETTINGS
+ ADD DOCUMENT	+ START COLLECTION
FORTIANALYZER	+ ADD FIELD
FORTIGATEAUTOSCALE	<ul style="list-style-type: none"> ▼ asset-storage-key-prefix <ul style="list-style-type: none"> description: "Asset storage key prefix." editable: false jsonEncoded: false settingValue: "empty"
FORTIGATEMASTERELECTION	<ul style="list-style-type: none"> ▼ asset-storage-name <ul style="list-style-type: none"> description: "Asset storage name." editable: false jsonEncoded: false settingValue: "fortigate-autoscale-rnmlo"
LIFECYCLEITEM	<ul style="list-style-type: none"> ▼ autoscale-handler-url <ul style="list-style-type: none"> description: "The FortiGate Autoscale handler UR..." editable: false jsonEncoded: false settingValue: "https://us-central1-..."
SETTINGS	<ul style="list-style-type: none"> ▼ byol-scaling-group-name <ul style="list-style-type: none"> description: "The name of the BYOL auto scaling ..." editable: false jsonEncoded: false settingValue: "fortigateautoscale-rnmlo"

Cluster monitoring

Various cluster metrics display in the GCP console under *Compute > Instance Groups > YOUR-FORTIGATE-AUTOSCALE_CLUSTER > Monitor*.

From here you can see the scale in and scale out actions that have been performed, as well as cluster health data.

Deploying autoscaling on GCP



Use [Google Cloud's operations suite \(formerly Stackdriver\)](#) for additional logging information, including Function scaling.

Adding instances to the protected subnet

When the deployment completes, you can create an instance group and add VMs to the protected subnet, behind the internal load balancer (LB).

In GCP, NICs must reside in separate VPCs. In this deployment, the FortiGate have two NICs: one in the exposed public subnet/VPC and the other in the protected subnet/VPC. By default, the protected subnet is called `fortigateautoscale-protected-subnet-CLUSTER-SUFFIX`.

The default FortiGate configuration located under `/assets/configset/baseconfig` specifies a virtual IP address (VIP) on port 80 and a VIP on port 443 with a policy that points to an internal LB.



Any VIPs created on the primary instance do not sync to the secondary instances. You must add any VIP you want to add as part of the baseconfig.

The following illustrates adding a basic unmanaged instance group into the protected subnet and internal LB.

To add instances to the protected subnet:

1. Create the VM, ensuring that it resides within the proper region, VPC, and subnet:

[← Create an instance](#)

To create a VM instance, select one of the options:

- New VM instance** [Create a single VM instance from scratch](#)
- New VM instance from template** [Create a single VM instance from an existing template](#)
- Marketplace** [Deploy a ready-to-go solution onto a VM instance](#)

Name [Name is permanent](#)

Region [Region is permanent](#) **Zone** [Zone is permanent](#)

us-central1 (Iowa)	us-central1-a
--------------------	---------------

Machine configuration [Machine types for common workloads, optimized for cost and flexibility](#)

Machine family [General-purpose](#) [Memory-optimized](#)

Series [N1](#)

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type [n1-standard-1 \(1 vCPU, 3.75 GB memory\)](#)

	vCPU	Memory
	1	3.75 GB

CPU platform and GPU

Container [Deploy a container image to this VM instance. Learn more](#)

Boot disk [New 10 GB standard persistent disk](#)

	Image	CentOS 7	Change
---	-------	----------	------------------------

Identity and API access [Service account](#)

Compute Engine default service account
--

Access scopes [Allow default access](#) [Allow full access to all Cloud APIs](#) [Set access for each API](#)

Firewall [Add tags and firewall rules to allow specific network traffic from the Internet](#)

<input type="checkbox"/> Allow HTTP traffic
<input type="checkbox"/> Allow HTTPS traffic

Networking [Management](#) [Security](#) [Disks](#) [Networking](#) [Sole Tenancy](#)

Network tags [\(Optional\)](#)

The screenshot shows the 'Create instance' wizard in the GCP console. The current step is 'Configure network interface'. The configuration includes:

- Hostname**: Set to "protected-instance.c.dev-project-001-166400.internal".
- Network interfaces**: One interface is defined with:
 - Network**: "fortigateautoscale-protected-vpc-kcjg".
 - Subnetwork**: "fortigateautoscale-protected-subnet-kcjg (172.16.8.0/24)".
 - Primary internal IP**: "Ephemeral (Automatic)".
 - External IP**: "None".
 - IP forwarding**: "Off".
- Add network interface**: A button to add more interfaces.

At the bottom, there are "Create" and "Cancel" buttons, and a link to "Equivalent REST or command line".

2. Create an instance group:

[← Create an instance group](#)

The screenshot shows the 'Create an instance group' dialog. On the left, there are two options: 'New managed instance group' and 'New unmanaged instance group'. The 'New unmanaged instance group' option is selected and highlighted in blue. It has a description: 'Create a group of unique VM instances without using a template. Add and remove VM instances manually.' To the right of the selection, there are several configuration fields:

- Name**: A text input field containing 'protected-instance-group'. A note says 'Name is permanent'.
- Description**: An optional text area.
- Location**:
 - Region**: A dropdown menu set to 'us-central1 (Iowa)'.
 - Zone**: A dropdown menu set to 'us-central1-a'.
- Specify port name mapping**: An optional text area.
- Network**: A dropdown menu set to 'fortigateautoscale-protected-vpc-kcjjg'.
- Subnetwork**: A dropdown menu set to 'fortigateautoscale-protected-subnet-kcjjg (172.16.8.0/24)'.
- VM instances**: A section showing a single instance named 'protected-instance' with a delete 'X' icon. Below it, a note says 'No available instances'.
- Billing Note**: A note stating 'You will be billed for VM instances in this group.' followed by links to 'Compute Engine pricing'.
- Buttons**: 'Create' and 'Cancel' buttons at the bottom.
- Equivalent Links**: Links to 'REST' or 'command line' documentation.

3. Under *Network services > Load balancing*, select *Internal load balancer > Backend configuration* and add the new instance group.

Destroying the cluster

The easiest way to destroy an autoscale cluster is to use Terraform.

To destroy the cluster:

- From your GCP directory, enter the following and confirm the resources are the ones to destroy.

```
terraform destroy
```

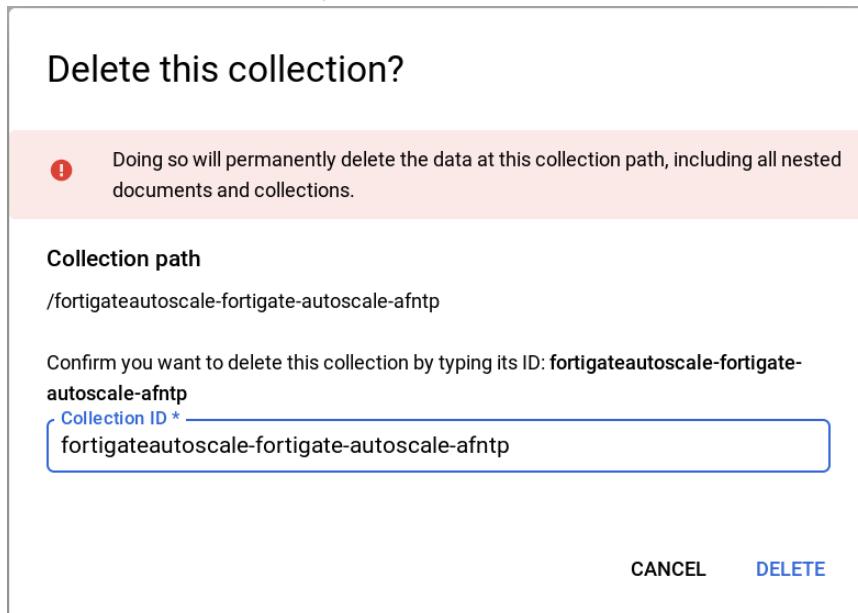
If you have VMs in the protected subnet you must manually remove these VMs before destroying the cluster. Output appears as follows after the cluster has been destroyed:

```
Destroy complete! Resources: 20 destroyed.
```

2. Erase the Firestore database by going to *Firebase*. Hover over the root collection and select *Delete collection*.



3. Enter the collection name to proceed.



Troubleshooting

Debugging cloud-init

Retrieving the `cloud-init` log can be useful when issues are occurring at bootup. To retrieve the log, log in to the FortiGate-VM and type the following into the CLI:

```
diag debug cloudinit show
```

Output will look similar to the following:

```
>> Checking metadata source gcp
>> GCP processing json format user-data
>> GCP trying to get config script from: https://us-central1-
*****.cloudfunctions.net/fortigateautoscale-rnmlo
>> GCP download config script successfully
>> Run config script
>> Finish running script
>> FortiGate-VM64-GCPON~AND $ config system dns
>> FortiGate-VM64-GCPON~AND (dns) $ unset primary
>> FortiGate-VM64-GCPON~AND (dns) $ unset secondary
>> FortiGate-VM64-GCPON~AND (dns) $ end
>> FortiGate-VM64-GCPON~AND $ config system auto-scale
>> FortiGate-VM64-GCPON~AND (auto-scale) $ set status enable
```

```
>> FortiGate-VM64-GCPON~AND (auto-scale) $ set sync-interface "port1"
>> FortiGate-VM64-GCPON~AND (auto-scale) $ set hb-interval 25
>> FortiGate-VM64-GCPON~AND (auto-scale) $ set role slave
>> FortiGate-VM64-GCPON~AND (auto-scale) $ set master-ip xxx.xxx.xxx.xxx
>> FortiGate-VM64-GCPON~AND (auto-scale) $ set callback-url https://us-central1-
*****.cloudfunctions.net/fortigateautoscale-rnml
>> FortiGate-VM64-GCPON~AND (auto-scale) $ set psksecret ****
>> FortiGate-VM64-GCPON~AND (auto-scale) $ end
```

How to reset the elected primary FortiGate

To reset the elected primary FortiGate, navigate to *FireStore > FortiGateMasterElection* and delete the only item. A new primary FortiGate is elected and a new record is created as a result.

For details on locating *FireStore > FortiGateMasterElection*, see [Verifying the deployment on page 111](#).

Appendix

FortiGate Autoscale for GCP features

Major components

Component	Description
Instance group	Contains one to many FortiGate-VMs (pay as you go licensing model). Dynamically scales out or scale in based on <code>cpu_utilization</code> .
configset folder	Contains files that are loaded as the initial configuration for a new FortiGate-VM instance.
baseconfig	Base configuration. You can modify this file as needed to meet your network requirements. Configset placeholders on page 121 explain placeholders such as <code>{SYNC_INTERFACE}</code> .
Tables in Firestore	Required to store information such as health check monitoring, primary election, state transitions, and so on. You should not modify these records unless required for troubleshooting purposes.

Configset placeholders

When the FortiGate-VM requests the configuration from the autoscaling function, the placeholders in the following table are replaced with associated environment variables stored in Cloud Functions:

Placeholder	Type	Description
<code>{SYNC_INTERFACE}</code>	Text	Interface for FortiGate-VMs to synchronize information. All characters must be lowercase.
<code>{CALLBACK_URL}</code>	URL	Cloud Functions URL to interact with the autoscaling handler script.

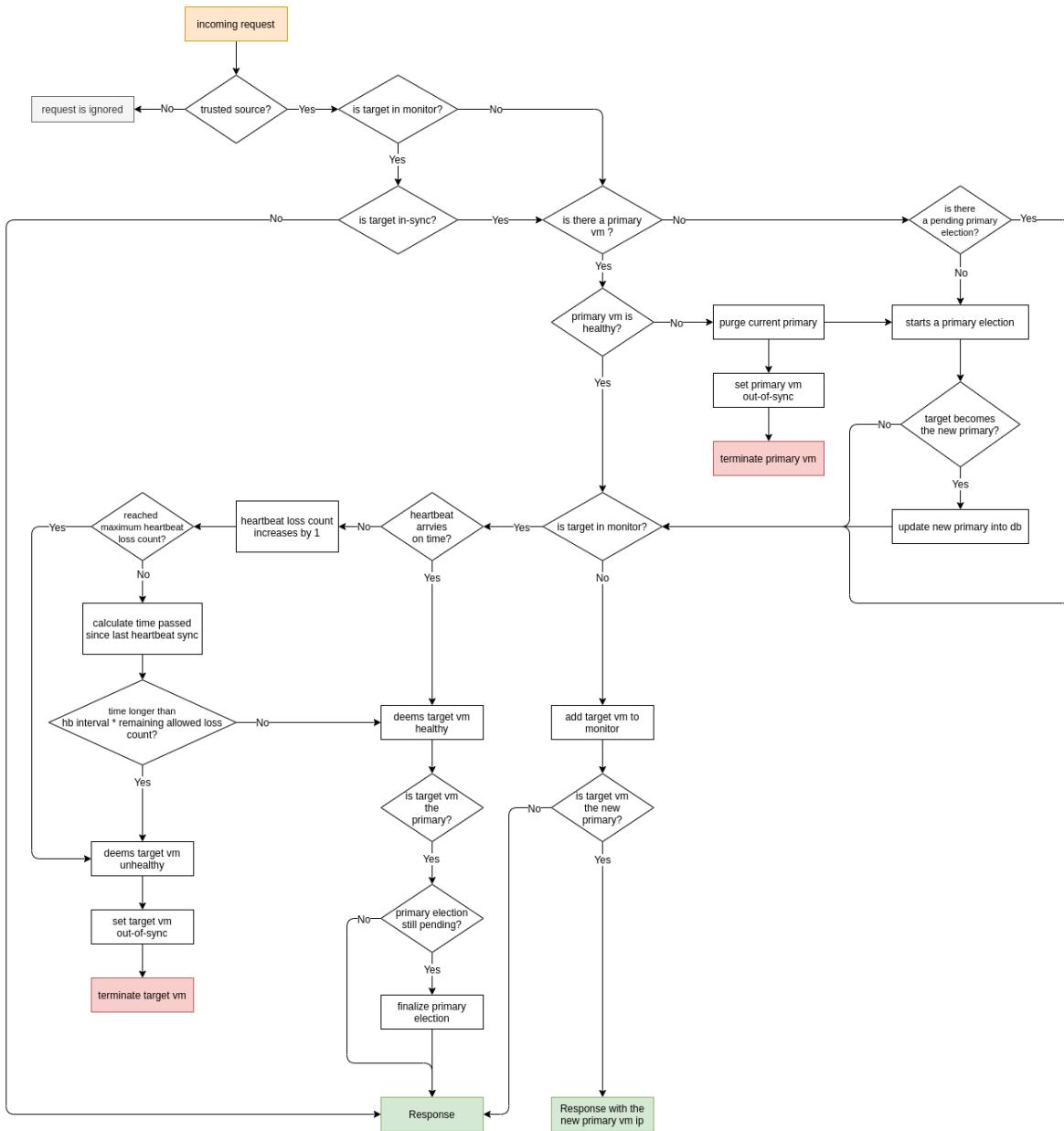
Placeholder	Type	Description
		Automatically generated during the Terraform deployment.
{ PSK_SECRET }	Text	<p>Preshared key used in FortiOS. Randomly generated during the Terraform deployment.</p> <p> Changes to the PSK secret after FortiGate Autoscale for GCP has been deployed are not reflected here. For new instances to be spawned with the changed PSK secret, this environment variable will need to be manually updated.</p>
{ADMIN_PORT}	Number	<p>A port number specified for administration login. A positive integer such as 443 etc. Default value: 8443.</p> <p> Changes to the admin port after deployment are not reflected here. For new instances to be spawned with the changed admin port, this environment variable will need to be updated.</p>

Architectural diagram

Election of the primary instance

FortiGate Autoscale

with heartbeat response & failover management



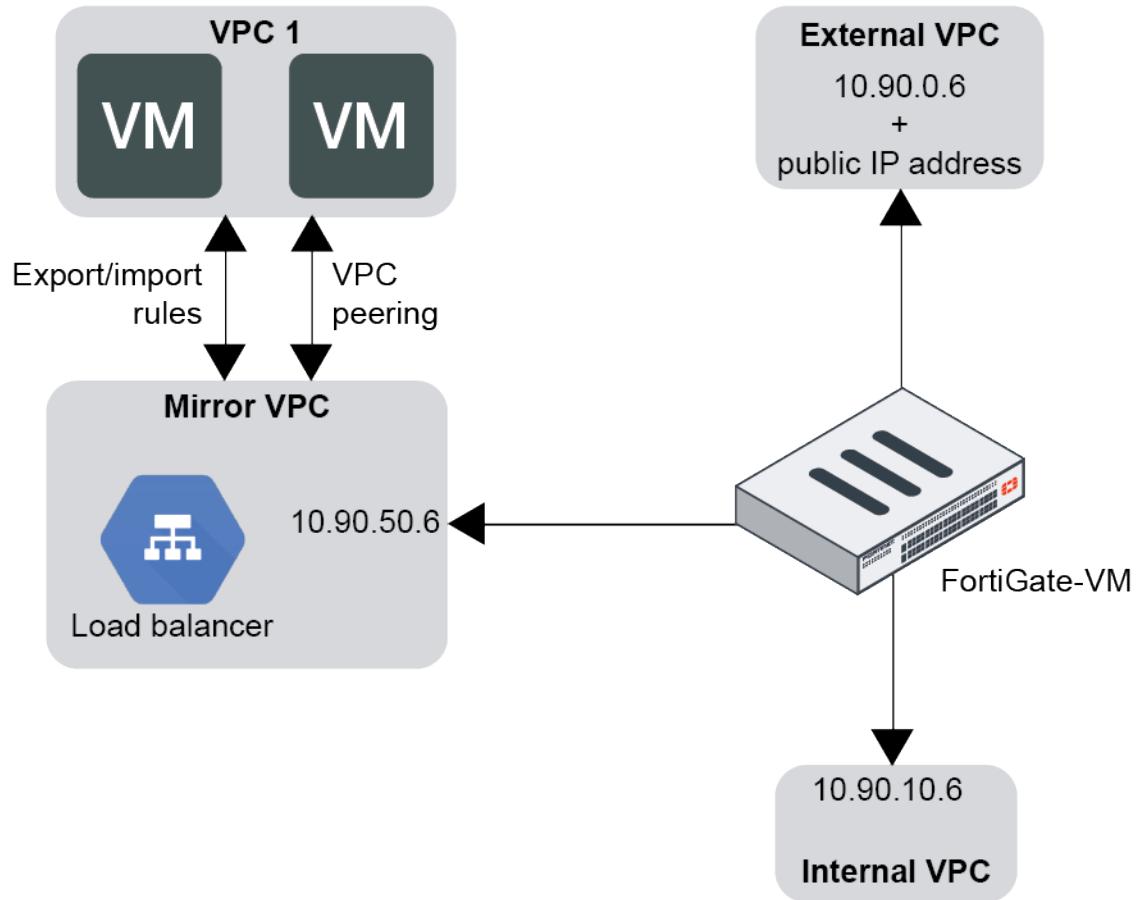
VPN for FortiGate-VM on GCP

Site-to-site IPsec VPNs between HA VPN on GCP

See [Google Cloud HA VPN interoperability guide for FortiGate](#).

Packet mirroring

You can use GCP's packet mirroring feature to capture all ingress and egress traffic and packet data, such as payloads and headers. As packet loading exports all traffic, not only the traffic between sampling periods, you may find it useful when monitoring and analyzing your security status. This configuration mirrors the traffic from a network interface or subnet in the specified VPC and sends it to the internal load balancer, which is specified as the destination in the packet mirroring policy. The following shows the topology for this configuration:



Creating VPC networks

This configuration requires three virtual private clouds (VPC) for the FortiGate: external, internal, and mirroring. It also requires a fourth VPC where you deploy the VM instances whose traffic will be mirrored. This guide refers to the fourth VPC as "VPC 1".

To create the VPC networks:

1. In the GCP console, go to *VPC Networks*, then click *CREATE VPC NETWORK*.
2. In the *Name* field, enter the desired name.
3. From the *Region* dropdown list, select the region appropriate for your deployment.
4. From the *IP address range* field, enter the first network's subnet in CIDR format, such as 10.0.1.0/24.
5. Leave all other settings as-is, then click *Create*.

packetmirroring-vpc1

Subnet creation mode
Custom subnets

Dynamic routing mode ?
 Regional
 Cloud Routers will learn routes only in the region in which they were created
 Global
 Global routing lets you dynamically learn routes to and from all regions with a single VPN or interconnect and Cloud Router

DNS server policy (Optional)
No server policy

Save **Cancel**

Subnets						Static internal IP addresses	Firewall rules	Routes	VPC Network Peering	Private service connection
Add subnet		Flow logs ▾								
<input type="checkbox"/>	Name ^	Region	IP address ranges	Gateway	Private Google access	Flow logs	<small>?</small>			
<input type="checkbox"/>	packetmirroring-vpc1-subnet1	us-west1	10.90.100.0/24	10.90.100.1	Off	Off				

Reserved subnets for internal HTTP(S) load balancers ?

<input type="checkbox"/>	Name	Region ^	IP address ranges	Gateway	Role
No matching results					

6. Repeat steps 1-5 to create the remaining three VPCs.
7. Go to *Compute Engine > Virtual machines > VM instances*. Deploy two VMs to VPC 1.

Launching the FortiGate-VM instance

Launch the FortiGate-VM instance from the marketplace as [Initially deploying the FortiGate-VM on page 22](#) describes. Ensure that you configure the FortiGate-VM with the network interfaces for the internal, external, and mirroring VPCs that you created in [Creating VPC networks on page 125](#).

VM instance details

Details Monitoring Screenshot

Remote access

SSH Connect to serial console

Enable connecting to serial ports
Connecting to serial ports is enabled in project-wide metadata

Logs

Cloud Logging Serial port 1 (console) More

Instance Id

Machine type

n1-standard-4 (4 vCPUs, 15 GB memory)

Warning: This instance is underutilized. You can save an estimated \$46 per month by switching to the machine type: custom (2 vCPUs, 5 GB memory). [Learn more](#) [Dismiss](#) [Resize](#)

Reservation

Automatically choose

In use by

packetmirror-instance-group1

CPU platform

Intel Broadwell

Display device

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

Turn on display device

Zone

us-west1-a

Labels

None

Creation time

Dec 4, 2020, 10:35:19 AM

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	ext	vpc-ext-subnet	10.90.0.6	—	34.82.224.172 (ephemeral)	Premium	On	View details
nic1	int	vpc-int-subnet	10.90.10.6	—	None			View details
nic2	mirror	vpc-mirror-subnet	10.90.50.6	—	None			View details

Creating an unmanaged instance group and load balancer

To create an unmanaged instance group:

1. Go to *Compute Engine > Instance groups > New unmanaged instance group*.
2. From the *Region* and *Zone* dropdown lists, select the same region and zone that the mirrored recipient, in this case the FortiGate-VM, is in.
3. From the *Network* dropdown list, select the FortiGate-VM external VPC network.
4. From the *Subnetwork* dropdown list, select the subnet in the external VPC where the FortiGate-VM interface is.
5. From the *VM instances* dropdown list, select the FortiGate-VM.
6. Click *Create*.

To create a health check:

1. Go to *Compute Engine > Instance groups > Health checks*.
2. From the *Protocol* dropdown list, select *TCP*.

3. In the *Port* field, enter 22.
 4. In the *Check interval* and *Timeout* fields, enter 5.
-



The health check fails unless you add a firewall rule to allow the following IP address ranges:
130.211.0.0/22 and 35.191.0.0/16.

To create an internal load balancer for the packet mirroring policy:

1. Go to *NETWORKING > Network services > Load balancing > Create load balancer*.
2. Under *TCP Load Balancing*, click *Start configuration*.
3. Under *Internet facing or internal only*, select *Only between my VMs*.
4. Under *Multiple regions or single region*, select *Single region only*.
5. Click *Continue*.
6. Complete backend configuration:
 - a. From the *Region* dropdown list, select the same region as the FortiGate-VM and instance group.
 - b. From the *Network* dropdown list, select the mirror VPC.
 - c. From the *Health check* dropdown list, select the health check that you created.
7. Complete frontend configuration:
 - a. From the *Subnetwork* dropdown list, select the mirror subnet.
 - b. Under *Advanced options*, select *Enable this load balancer for packet mirroring*.
 - c. Click *Done*.
8. Click *Create*.

Configuring bidirectional VPC peering

To configure bidirectional VPC peering:

1. Go to *VPC network > VPC network peering*.
2. Click *CREATE CONNECTION*, then Continue.
3. From the *Your VPC network* dropdown list, select the mirror VPC.
4. From the *VPC network name* dropdown list, select VPC 1.
5. Select all *Import* and *Export* options.
6. Click *CREATE*.
7. Repeat steps 2-6, this time selecting VPC 1 in the *Your VPC network* dropdown list and the mirror VPC in the *VPC network name* dropdown list. This allows bidirectional traffic flow.

Creating the packet mirroring policy

This policy mirrors the contents of VPC 1 and reflects them on the mirror VPC.

To create the packet mirroring policy:

1. Go to *VPC network > Packet mirroring > CREATE POLICY*.
2. From the *Region* dropdown list, select the same region selected for previous resources.
3. Under *Policy enforcement*, select *Enabled*. Click *CONTINUE*.
4. Select the VPC network:
 - a. Select *Mirrored source and collector destination are in separate, peered VPC networks*.
 - b. From the *Mirrored source VPC network* dropdown list, select VPC 1.
 - c. From the *Collector destination VPC network* dropdown list, select the mirror VPC. Click *CONTINUE*.
5. Click *Select one or more subnetworks*.
6. From the dropdown list, select VPC 1. Click *CONTINUE*.
7. The collector destination must be a GCP load balancer. From the *Collector destination* dropdown list, select the frontend name of the load balancer that you created in [To create an internal load balancer for the packet mirroring policy: on page 128](#). Click *CONTINUE*.
8. Select *Mirror all traffic*. Alternatively, you can monitor traffic between specific instances using instance tags.

Verifying the configuration

To verify the configuration:

1. On one of the VMs in VPC 1, ping the other VM. In this example, the VM IP addresses are 10.138.0.8 and 10.138.0.9. The following shows successful communication between the VMs:

```
ping 10.138.0.8 (10.138.0.8) 56(84) bytes of data.
64 bytes from 10.138.0.8: icmp_seq=1 ttl=64 time=1.48 ms
64 bytes from 10.138.0.8: icmp_seq=2 ttl=64 time=0.371 ms
64 bytes from 10.138.0.8: icmp_seq=3 ttl=64 time=0.382 ms
64 bytes from 10.138.0.8: icmp_seq=4 ttl=64 time=0.377 ms
^C
--- 10.138.0.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 42ms
rtt min/avg/max/mdev = 0.371/0.653/1.484/0.480 ms
```

2. If the packet mirroring configuration was successful, the packets are visible to the FortiGate. In FortiOS, run the `diag sniffer packet port3 'host <VM 1 IP address> and host <VM 2 IP address>' 4 0 3` command. Port3 is the FortiGate interface that is sitting in the mirror VPC. The output should look as follows:

```
fgt # diag sniffer packet port3 'host 10.138.0.8 and host 10.138.0.9' 4 0 3
interfaces=[port3]
filters=[host 10.138.0.8 and host 10.138.0.9]
pcap_lookupnet: port3: no IPv4 address assigned
6.783470 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
6.783623 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
6.784078 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
6.784310 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
7.784492 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
7.784519 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
7.784673 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
7.784687 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
8.797265 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
8.797290 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
8.797485 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
8.797494 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
9.821224 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
9.821246 port3 -- 10.138.0.9 -> 10.138.0.8: icmp: echo request
9.821393 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
9.821494 port3 -- 10.138.0.8 -> 10.138.0.9: icmp: echo reply
```

Organization restrictions

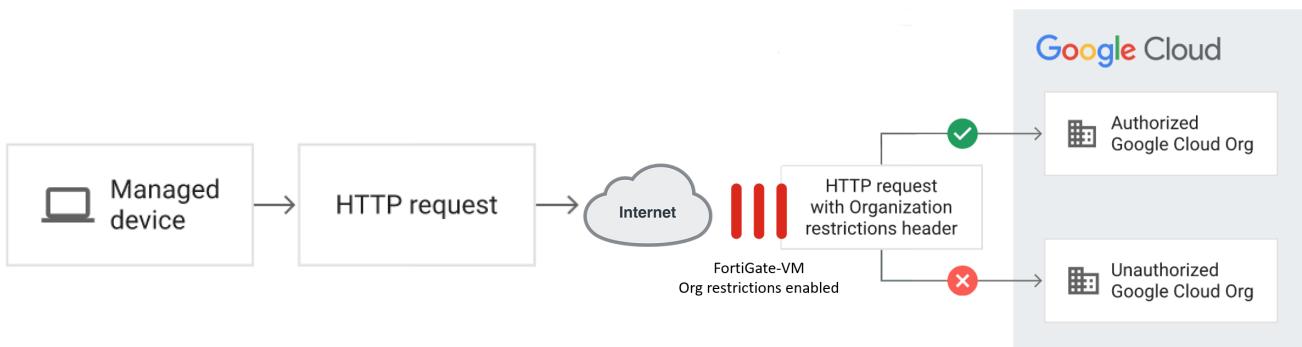
FortiGate-VM on GCP supports the organization restrictions feature. This guide is a walkthrough of how to configure FortiGate as a proxy and create the header insertion to use this GCP feature.

This guide assumes that your GCP environment has existing networks and resources.

The following provides an overview of using this feature with FortiGate-VM:

- Customer GCP organization ID is required. For information about finding your organization ID from a project ID, see [gcloud projects get-ancestors](#).
- You must set the FortiGate public IP address as the proxy server on the client web browser.
- The FortiGate-VM on GCP (proxy and header insertion) receives and processes the HTTP request.
- The FortiGate proxy firewall policy allows or denies GCP resource access based on header X-Goog-Allowed-Resources content of your organization ID in Base64 encoding.

For information about GCP organization restriction, see [Introduction to organization restrictions](#).



To configure the FortiGate-VM with organization restrictions:

1. On the FortiGate, enable web proxy:
 - a. Go to *Network > Explicit Proxy*.
 - b. Enable *Explicit Web Proxy*.

Organization restrictions

- c. Configure the proxy HTTP and HTTPS ports as desired. This example sets them to 8080.

Explicit Proxy

Explicit Web Proxy

Listen on Interfaces

HTTP Port High

HTTPS Port

FTP over HTTP

Proxy auto-config (PAC)

Proxy FQDN

Max HTTP request length KB

Max HTTP message length KB

Unknown HTTP version

Realm

Default Firewall Policy Action

Outgoing IP

Web Proxy Forwarding Servers

Server Name	Address	Port	Health Check	Server Down	Comments
No results					

URL Match List

Name	URL Pattern	Cache Exemption	Forward Server	Status	Comments
No results					

Explicit FTP Proxy

- d. Create address objects specifying allowed GCP endpoints. You will use the address objects in the web proxy profile header and the proxy policy configuration:

```
config firewall address
    edit allow_gcp_api_addr_obj
        set type fqdn
        set fqdn *.googleapis.com
    next
end
config firewall address
    edit allow_gcp_com_addr_obj
        set type fqdn
        set fqdn *.google.com
    next
end
```

```
config firewall address
    edit allow_gstatic_addr_obj
        set type fqdn
        set fqdn www.gstatic.com
    next
end
```

You may need more address objects for other Google services. You must add these address objects to the proxy header configuration and the proxy policy as destination addresses:



- *.gcr.io
- *.pkg.dev
- *.cloudfunctions.net
- *.run.app
- *.tunnel.cloudproxy.app
- *.datafusion.googleusercontent.com

2. Configure the web proxy profile:

```
config web-proxy profile
    edit gcp-org-restrict-profile
        config headers
            edit 1
                set name X-Goog-Allowed-Resources
                set dstaddr allow_gcp_api_addr_obj allow_gcp_com_addr_obj allow_gstatic_addr_obj
                set action add-to-request
                set content "{\"resources\": [\"organizations/<Customer Org ID>\"] , \"options\": \"strict\"}"
                set base64-encoding enable
                set add-option new
                set protocol http https
            end
        end
    end
```

3. Configure the proxy policy:



This is an example policy. Do not use it in a production environment.

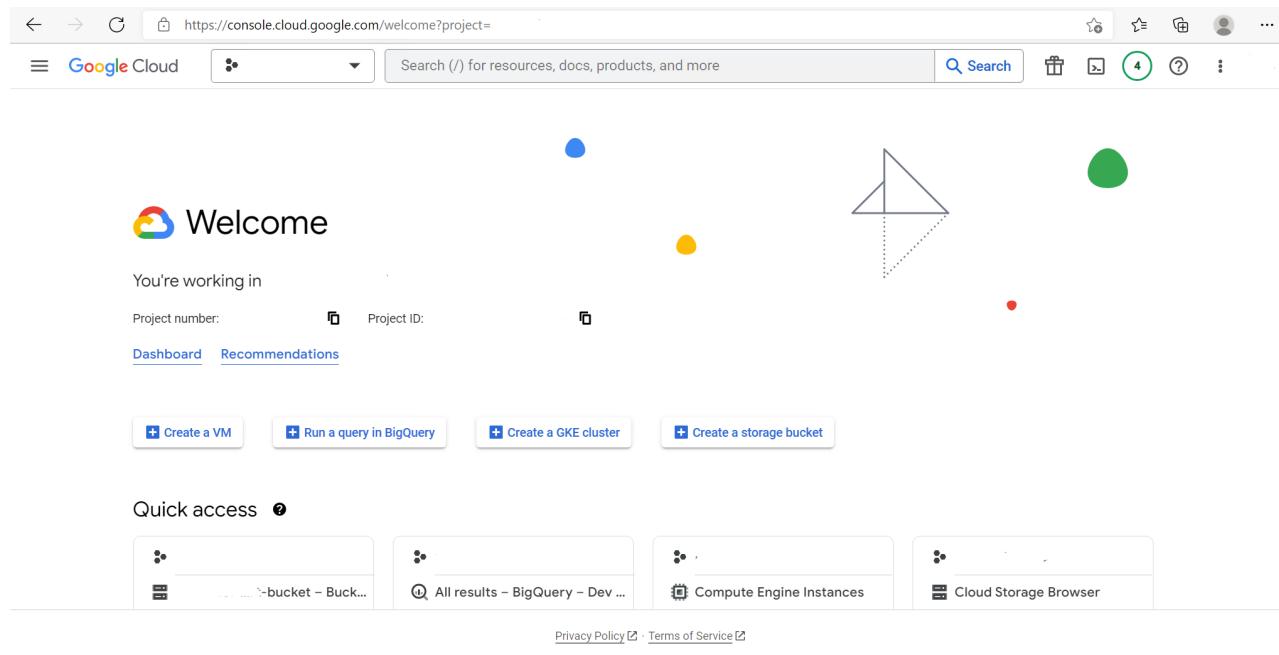
```
config firewall proxy-policy
    edit 1
        set name "gcp_restriction_policy"
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "allow_gcp_api_addr_obj" "allow_gcp_com_addr_obj" "allow_gstatic_addr_obj"
        set service "webproxy"
        set srcaddr-negate disable
        set dstaddr-negate disable
        set service-negate disable
        set action accept
        set status enable
        set schedule "always"
```

Organization restrictions

```
set logtraffic utm
set webproxy-forward-server ''
set webproxy-profile gcp-org-restrict-profile
set transparent disable
set disclaimer disable
set utm-status disable
set profile-protocol-options "default"
set ssl-ssh-profile "no-inspection"
set replacemsg-override-group ''
set logtraffic-start disable
set comments ''

next
end
```

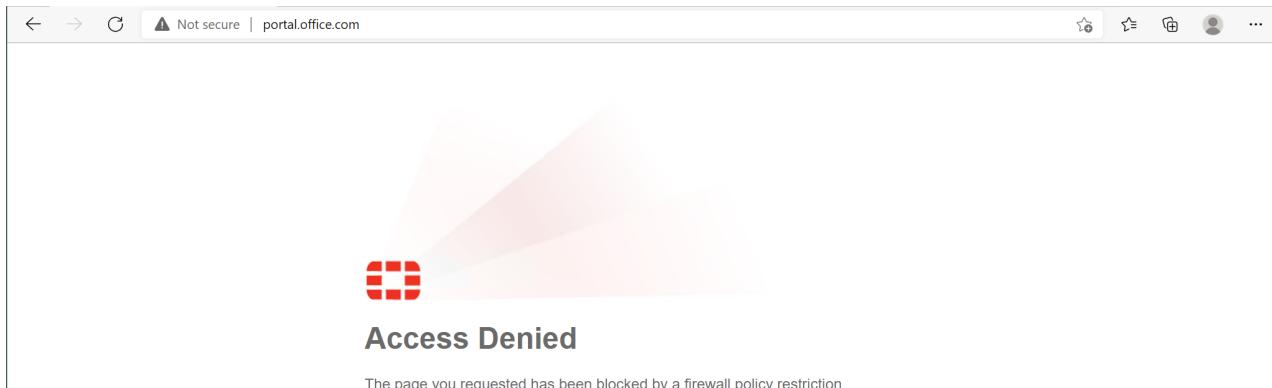
While a workstation is configured to use the FortiGate as a proxy, the web browser is allowed access to resources in the organization:



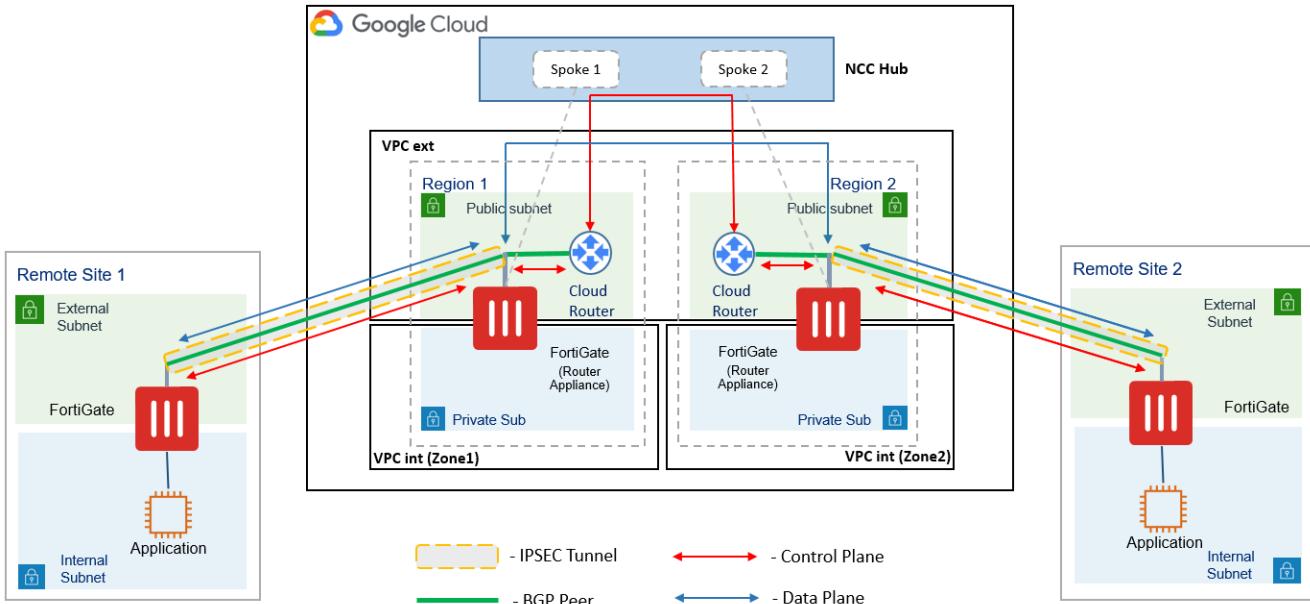
Organization restrictions

```
FGT-ORG-RESTRICTIONS # diagnose sniffer packet port1 'port 8080'
Using Original Sniffing Mode
interfaces=[port1]
filters=[port 8080]
4.177530      .51420 -> 10.0.1.36.8080: syn 2546082822
4.177596 10.0.1.36.8080 -> .51420: syn 1439644099 ack 2546082823
4.225628      .60541 -> 10.0.1.36.8080: syn 655859283
4.225677 10.0.1.36.8080 -> .60541: syn 2417995884 ack 655859284
4.238066      .51420 -> 10.0.1.36.8080: ack 1439644100
4.242790      .51420 -> 10.0.1.36.8080: psh 2546082823 ack 1439644100
4.242826 10.0.1.36.8080 -> .51420: ack 2546083105
4.243740 10.0.1.36.8080 -> .51420: psh 1439644100 ack 2546083105
4.291479      .60541 -> 10.0.1.36.8080: psh 655859284 ack 2417995885
4.291517 10.0.1.36.8080 -> .60541: ack 655859526
4.291530      .60541 -> 10.0.1.36.8080: ack 2417995885
4.291535 10.0.1.36.8080 -> .60541: ack 655859526
4.294958 10.0.1.36.8080 -> .60541: psh 2417995885 ack 655859526
4.306829      51420 -> 10.0.1.36.8080: psh 2546083105 ack 1439644172
4.308443 10.0.1.36.8080 -> .51420: 1439644172 ack 2546083622
```

The policy on the FortiGate proxy stops access to any other resources outside of the GCP organization:



SD-WAN transit routing with Google Network Connectivity Center



This guide assumes that the remote site (side office) is already in place with its application and FortiGate instances as the diagram shows. Therefore, this guide does not cover the steps for their deployment.

The remote site or side office is where customer workloads are present. This can be an on-premise or cloud deployment.

With an SD-WAN transit routing setup with Google Network Connectivity Center (NCC), you can route data and exchange border gateway protocol (BGP) routing information between two or more remote sites via GCP.

You can do this by configuring the NCC hub and an endpoint (spoke) for each remote site. To reduce network latency, you deploy a spoke in the GCP region that is located geographically closest to the remote site for which you created the spoke. The NCC hub itself is VPC-specific.

Prerequisites

To complete the prerequisites:

1. Download the required scripts from the [Fortinet GitHub repository](#).
2. Place the files in a GCP storage bucket:
 - a. Do one of the following:
 - i. To create a new bucket, in GCP, go to *Storage > Browser*, then click *Create Bucket*.
 - ii. To upload the files in an existing bucket, in GCP, go to *Storage > Browser*, and click the desired bucket.



You can store files securely by implementing measures such as disabling public access for the bucket and enforcing custom access control lists. See [Security and Privacy Considerations](#).

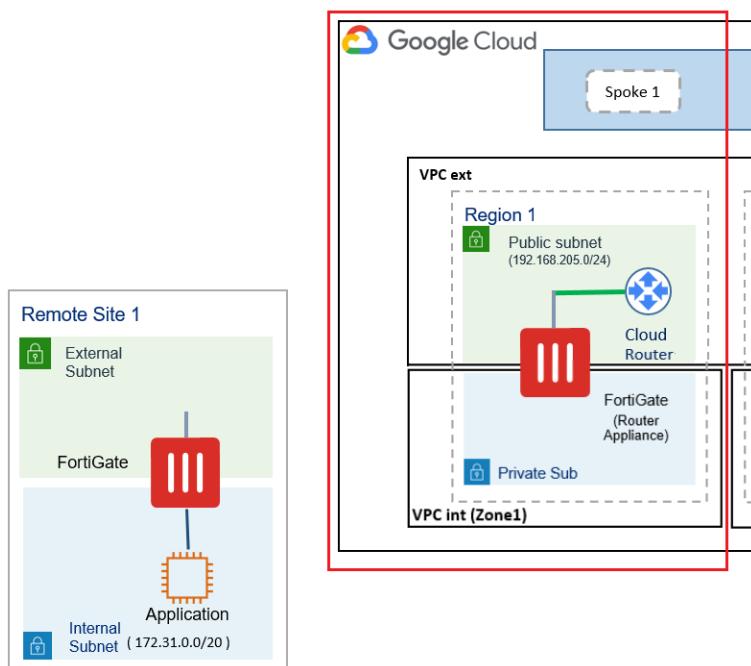
- b.** Upload the scripts that you downloaded to the bucket.
- 3.** Set up the environment. Do one of the following:
 - a.** Set up a Linux machine with python3 and PyJWT to execute the scripts. For details, see [Setting up a Python development environment](#).
 - b.** Use the Google Cloud shell to run the scripts. To execute the scripts successfully, you must install the following dependencies on the GCP cloud shell:


```
pip3 install --upgrade pip
pip3 install cryptography
pip3 install PyJWT
```

Script execution for a single spoke

To deploy a single spoke on the NCC by executing the script:

- 1.** The provided set of scripts deploys a single spoke on the NCC.



Filename	Description
deploy-fortigate-ncc.py	Script to deploy spoke in single region of the NCC.
Fortigate-ncc-param-zone1.json	Variables required for spoke deployment are stored here.

The variables listed in Fortigate-ncc-param-zone1.json except `project`, `ncc_vpc_ext`, and `ncc_hub` are unique for each spoke deployment. Ensure that you keep `ncc_vpc_int` unique. This allows you to deploy and access resources under the spoke FortiGate in its port 2 subnet. The following lists variables listed in Fortigate-ncc-param-zone1.json:

Variable	Description	Example
<code>project</code>	GCP project in which the infrastructure needs to be deployed.	project-001
<code>region</code>	Region in which spoke and cloud router are to be deployed.	us-west1
<code>zone</code>	Zone in which spoke and cloud router are to be deployed.	us-west1-a
<code>ncc_vpc_ext</code>	VPC for FortiGate external subnet.	demo-ext-1
<code>ncc_vpc_int</code>	VPC For FortiGate internal subnet.	demo-int-1
<code>ncc_hub</code>	Name of the NCC hub being created.	demo-ncc-hub
<code>Cloud_router</code>	Cloud router name for this zone.	zn1-cloudrouter
<code>Fortigate_spoke1</code>	Name of the spoke being created (shares name with FortiGate).	Fortigate-1
<code>sitetositeData</code>	Allows for exchange of site-to-site data and BGP routes between regions. This variable must be set to True.	True
<code>fortigate_spoke1_extip</code>	Unique name for static public IP address created for the FortiGate.	spoke1-publicip
<code>Ncc_vpc_ext_cidr</code>	Subnet used in <code>ncc_vpc_ext</code> (external).	192.168.205.0/24
<code>Ncc_vpc_int_cidr</code>	Subnet used in <code>ncc_vpc_int</code> (internal).	192.168.215.0/24
<code>fortigate_pwd</code>	Administrator password for FortiGate instance.	<string>
<code>cloud_router_ip1</code>	IP address assigned to cloud router interface 1.	192.168.205.101
<code>cloud_router_ip2</code>	IP address assigned to cloud router interface 2.	192.168.205.102
<code>cloud_router_asn</code>	Autonomous system number (ASN) set on cloud router.	65012
<code>fortigate_router_id</code>	Router ID set on FortiGate (spoke).	169.254.254.254
<code>fortigate_router_asn</code>	ASN set on FortiGate.	7252

2. Store the `Fortigate-ncc-param-zone1.json` text file in the GCP bucket.
3. Create an API key to authenticate and create resources on behalf of a GCP account. See [Authenticate using API keys](#) for details on creating an API key.
4. Copy `deploy-fortigate-ncc.py` and the API key file (`api_key`) locally for execution using the following command:

```
gsutil cp gs://<bucket-name>/<filename>.py <local_path>
```

The following shows an example of the command:

```
gsutil cp gs://test-bucket/deploy-fortigate-ncc.py
```

See [cp - Copy files and objects](#) for details.

- Execute the Python script, using the absolute path for the API key:

```
python3 deploy-fortigate-ncc.py <public_APIkey>.json <bucket_name> Fortigate-ncc-param-zone1.json
```

The following shows an example of the command:

```
python3 deploy-fortigate-ncc.py /home/pbapikey.json test-bucket1 Fortigate-ncc-param-zone1.json
```

- Use the same script to deploy the hub and other individual spokes by changing the given `Fortigate-ncc-param-zone1.json` file to reflect the correct variables for the new spoke deployment.

- Verify that the script ran successfully by running the following commands. The commands describe the infrastructure that the script deployed:

- To verify the hub, run `gcloud alpha network-connectivity hubs describe <ncc_hub>`. For example, if the NCC hub is named `testing-ncc-hub`, the command would be `gcloud alpha network-connectivity hubs describe testing-ncc-hub`.

```
@cloudshell:~ (dev-project-001-166400)$ gcloud alpha network-connectivity hubs describe testing-ncc-hub
createTime: '2021-04-06T16:52:47.634576969Z'
name: projects/dev-project-001-166400/locations/global/hubs/testing-ncc-hub
spoke: https://networkconnectivity.googleapis.com/v1alpha1/projects/966517025500/locations/us-west1/spokes/testing-fgt-1
- https://networkconnectivity.googleapis.com/v1alpha1/projects/966517025500/locations/us-west2/spokes/testing-fgt-2
state: ACTIVE
uniqueId: lc04device-1
updateTime: '2021-04-06T16:52:48.088581273Z'
@cloudshell:~ (dev-project-001-166400)$
```

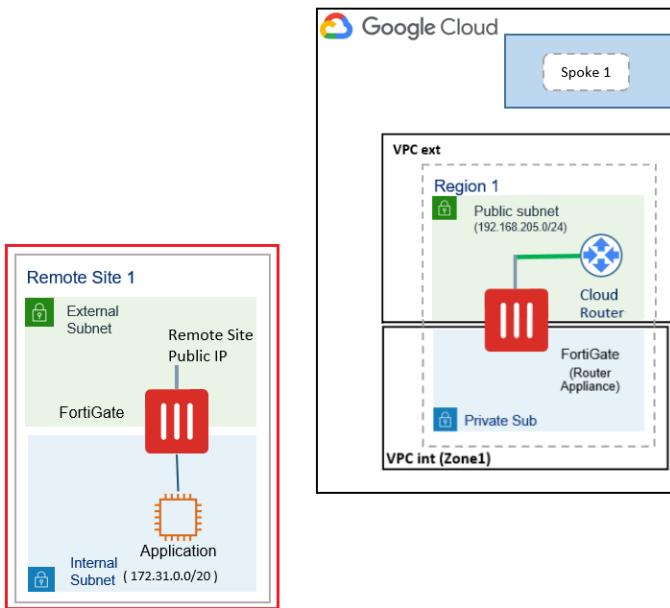
- To verify the spokes, run `gcloud alpha network-connectivity spokes describe <spoke_name> --<region_name>`. For example, if the spoke is named `testing-fgt-1` and the region is `us-west1`, the command would be `gcloud alpha network-connectivity spokes describe testing-fgt-1 --region=us-west1`.

```
@cloudshell:~ (dev-project-001-166400)$ gcloud alpha network-connectivity spokes describe testing-fgt-1 --region=us-west1
createTime: '2021-04-08T22:52:05.785607369Z'
hub: https://networkconnectivity.googleapis.com/v1alpha1/projects/966517025500/locations/global/hubs/testing-ncc-hub
linkAddressingType: IP
- ipAddress: 192.168.205.3
virtualMachine: https://www.googleapis.com/compute/v1/projects/dev-project-001-166400/zones/us-west1-a/instances/testing-fgt-1
name: projects/dev-project-001-166400/locations/us-west1/spokes/testing-fgt-1
state: ACTIVE
uniqueId: e5edc479-1
updateTime: '2021-04-08T22:52:11.400683322Z'
```

- To verify the cloud router, run `gcloud compute routers describe <cloud_router> --region=<region_name>`. For example, if the cloud router is named `testing-cr-zn1` and the region is `us-west1`, the command would be `gcloud compute routers describe testing-cr-zn1 --region=us-west1`.

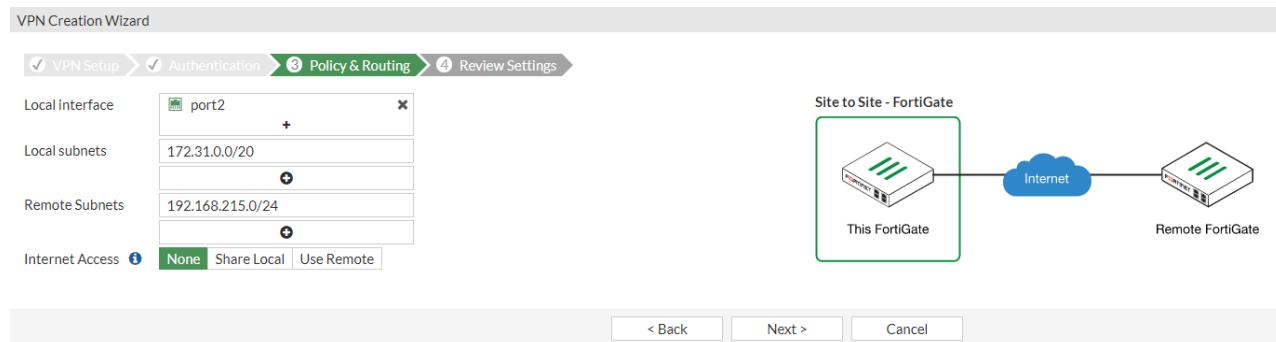
```
@cloudshell:~ (dev-project-001-166400)$ gcloud compute routers describe testing-cr-zn1 --region=us-west1
bgp:
  advertiseMode: DEFAULT
  asn: 65015
  bgpPeers:
    - interfaceName: testing-cr-zn1-0
      ipAddress: 192.168.205.101
      name: testing-cr-zn1-0-bgp0
      peerAsn: 7260
      peerIpAddress: 192.168.205.3
    - interfaceName: testing-cr-zn1-1
      ipAddress: 192.168.205.102
      name: testing-cr-zn1-1-bgp1
      peerAsn: 7260
      peerIpAddress: 192.168.205.3
  creationTimestamp: '2021-04-12T11:17:42.848-07:00'
  id: 104741354305744937
  interfaces:
    - ipRange: 192.168.205.101/24
      name: testing-cr-zn1-0
    - ipRange: 192.168.205.102/24
      name: testing-cr-zn1-1
  kind: compute#Router
  name: testing-cr-zn1
  network: https://www.googleapis.com/compute/v1/projects/dev-project-001-166400/global/networks/testing-ncc-ext-1
  region: https://www.googleapis.com/compute/v1/projects/dev-project-001-166400/regions/us-west1
  selfLink: https://www.googleapis.com/compute/v1/projects/dev-project-001-166400/regions/us-west1/routers/testing-cr-zn1
```

Configuring site-to-site VPN



To configure site-to-site VPN:

1. On the remote site 1 FortiGate, go to *VPN > IPsec Tunnels*, then click *Create New*.
2. On the *VPN Setup* tab, configure the following:
 - a. For *Template type*, select *Site to Site*.
 - b. For *NAT configuration*, select *No NAT between sites*.
 - c. Click *Next*.
3. On the *Authentication* tab, configure the following:
 - a. In the *Remote IP address* field, enter the destination FortiGate public IP address. This is the spoke1 public IP address.
 - b. Configure a signature or preshared key to secure the tunnel.
 - c. Click *Next*.
4. On the *Policy & Routing* tab, configure the local and remote subnets. Note that here, the local subnet refers to the remote site subnet, and the remote subnet refers to the NCC external and internal VPC subnets. Click *Next*.





Selecting all local and remote subnets should add the required firewall rules from port2 to the tunnel interface. If not, you must manually add the rules and set to allow all to try and debug the configuration. Ensure that you have added all the required local and remote subnets that need to be allowed through the tunnel.

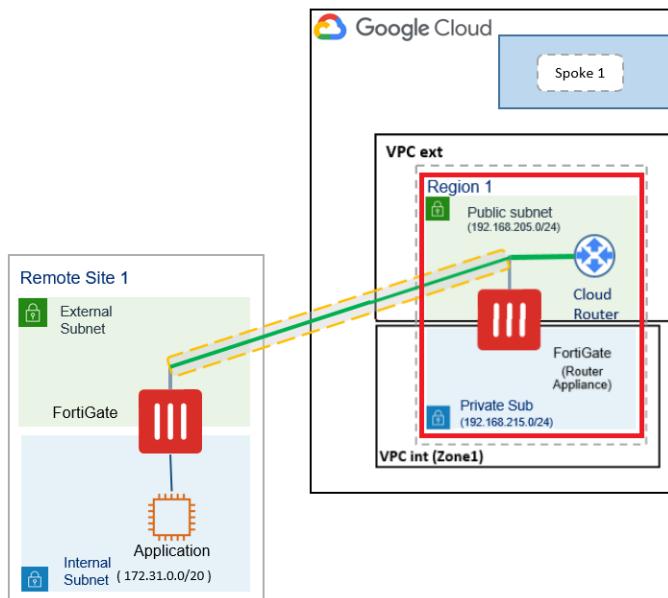
5. Review the configuration, then click *Create*.
6. Create a similar connection from the Region 1 spoke FortiGate to the remote site 1 FortiGate. When creating this connection, on the *Policy & Routing* tab, ensure that you add port1 and port2 as local interfaces when creating the tunnel interface.

Configuring the tunnel interfaces

The following instructions use the 169.254.110.0/29 subnet. This subnet is used only inside the site-to-site tunnel. You will use the IP addresses assigned in this configuration to configure BGP neighbors.

To configure the tunnel interface on the spoke 1 FortiGate to the remote site 1 FortiGate:

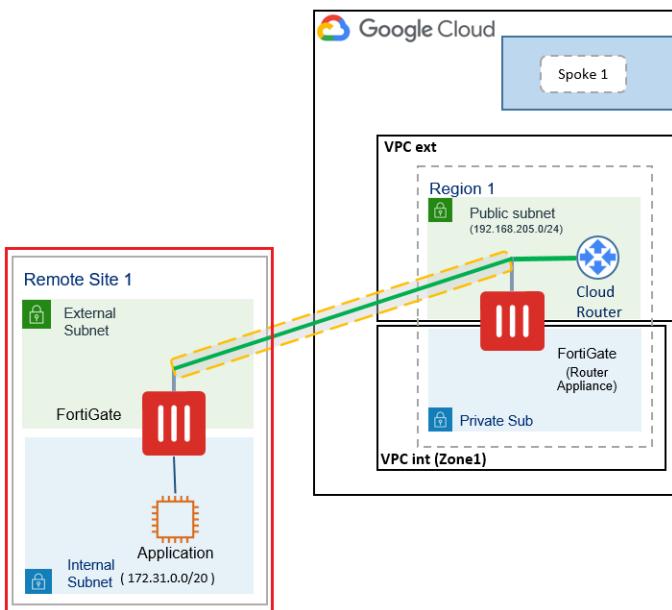
1. On the spoke 1 FortiGate, go to *Network > Interfaces*.
2. Extend the port 1 interface to reveal a new tunnel interface.
3. Edit the tunnel interface:
 - a. In the *IP* field, enter the local tunnel IP address. In this example, the value would be 169.254.110.1, the spoke 1 FortiGate IP address.
 - b. In the *Remote IP/Netmask* field, enter the remote tunnel IP address with netmask. In this example, the value would be 169.254.110.2 255.255.255.248. 169.254.110.2 is the remote site 1 FortiGate IP address, and 255.255.255.248 is the netmask.
 - c. Configure other settings as your network requires.
 - d. Click *OK*.



To configure the tunnel interface on the remote site 1 FortiGate to the spoke 1 FortiGate:

1. On the remote site 1 FortiGate, go to *Network > Interfaces*.
2. Extend the port 1 interface to reveal a new tunnel interface.
3. Edit the tunnel interface:
 - a. In the *IP* field, enter the local tunnel IP address. In this example, the value would be 169.254.110.2, the remote site 1 FortiGate IP address.
 - b. In the *Remote IP/Netmask* field, enter the remote tunnel IP address with netmask. In this example, the value would be 169.254.110.1 255.255.255.248. 169.254.110.2 is the spoke 1 FortiGate IP address, and 255.255.255.248 is the netmask.
 - c. Configure other settings as your network requires.
 - d. Click *OK*.

Configuring BGP neighbors



To configure BGP neighbors:

1. Configure the remote site 1 FortiGate:
 - a. Log in to the remote site 1 FortiGate.
 - b. Set a unique AS number and route ID:

```
config router bgp
  set as 7224
  set router-id 169.254.110.2
end
```

- c. Configure the BGP neighbor. Use the IP address set on the tunnel interface in [Configuring the tunnel interfaces on page 140](#) as the neighbor IP address. Set the AS obtained from the NCC spoke 1 FortiGate as the remote AS number. Enable `ebgp-enforce-multihop` and `soft-reconfiguration`:

```
config router bgp
```

```
config neighbor
    edit "169.254.110.1"
        set ebgp-enforce-multipath enable
        set soft-reconfiguration enable
        set remote-as 7252
    next
next
end
```

- d. Configure the network. The network prefix here is the FortiGate port 2 subnet in the remote office:

```
config network
    edit 1
        set prefix 172.31.0.0 255.255.240.0
    next
end
```

2. Configure the NCC spoke 1 FortiGate:

- a. Log in to the NCC spoke 1 FortiGate.
- b. Two BGP neighbors are already preconfigured from the initial script. o 192.168.205.101 and 192.168.205.102 are BGP neighbor associations for the cloud router located in the same region. Add a third BGP neighbor entry to peer the spoke 1 FortiGate to the remote site 1 FortiGate. Enable `ebgp-enforce-multipath` and `soft-reconfiguration`:

```
config router bgp
    config neighbor
        edit "169.254.110.2"
            set ebgp-enforce-multipath enable
            set soft-reconfiguration enable
            set remote-as 7224
        next
    next
end
```

Enabling dynamic routing mode

You must enable dynamic routing mode on the external VPC of the newly created NCC setup. This ensures that the cloud router dynamically advertises subnets and propagates learned routes in the region where the router is configured or throughout the entire VPC network.

To enable dynamic routing mode:

1. In the GCP management console, go to *Networking > VPC Network > (external_vpc_name) > EDIT*.
2. Under *Dynamic routing mode*, select *Global*.
3. Click *SAVE*.

4. Go to the external subnet's *ROUTES* tab. You should see the new dynamic routes.

Name	Description	Destination IP range	Priority	Instance tags	Next hop
default-route-3500738dbb9eec4b	Default local route to the subnetwork 192.168.225.0/24.	192.168.225.0/24	0	None	Virtual network pratheekb-ncc-ext-1
default-route-4d958c567e9428d5	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway
default-route-f55672078a7e0710	Default local route to the subnetwork 192.168.205.0/24	192.168.205.0/24	0	None	Virtual network pratheekb-ncc-ext-1
-cr-new-1-dynamic-route-1		172.31.0.0/20	0	None	IP address 192.168.205.2
-cr-new-1-dynamic-route-2		192.168.215.0/24	0	None	IP address 192.168.205.2
-cr-new-1-dynamic-route-3		172.31.0.0/20	0	None	IP address 192.168.205.2
-cr-new-1-dynamic-route-4		192.168.215.0/24	0	None	IP address 192.168.205.2
-cr-new-zn2-dynamic-route-1		192.168.235.0/24	0	None	IP address 192.168.225.2

Completing post-deployment configuration

The following shows the post-deployment configuration procedure on the GCP region 1 FortiGate. You would follow the same steps for other regions.

To complete post-deployment configuration:

- Configure the route map. Here, `nexthop1` and `nexthop2` are configured for the local cloud redundant interfaces by the script:

```
config router route-map
  edit "nexthop1"
    config rule
      edit 1
        set set-ip-nexthop 192.168.205.101
      next
    end
  next
  edit "nexthop2"
    config rule
      edit 1
        set set-ip-nexthop 192.168.205.102
      next
    end
  next
end
```

- Ensure that the route maps that the script created are reflected in their corresponding BGP neighbor entry:

```
config router bgp
  set as 7260
  set router-id 169.250.250.254
  config neighbor
    edit "192.168.205.101"
      ...
      set route-map-in "nexthop1"
    next
    edit "192.168.205.102"
      ...

```

```

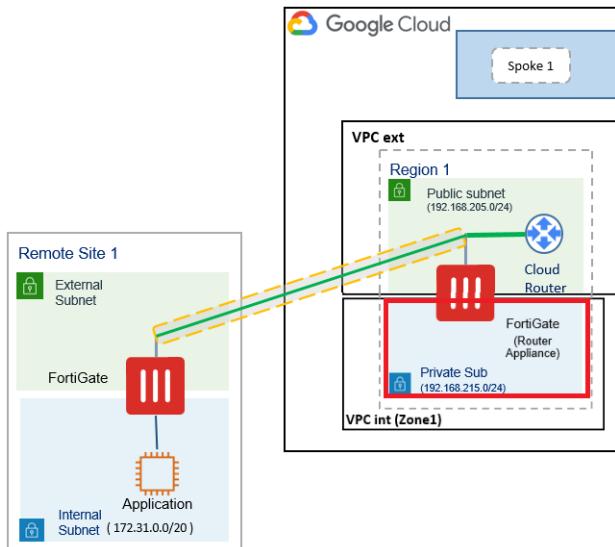
        set route-map-in "nexthop2"
        next
        ...
    end

```

Deploying multiple spokes

To deploy another spoke in a new zone, follow the procedure described in [Script execution for a single spoke on page 136](#). Ensure that all variables except `project`, `ncc_vpc_ext`, and `ncc_hub` are unique for each spoke deployment in the `Fortigate-ncc-param.json` file for zone 2. `ncc_vpc_int` should be a unique VPC for each region to effectively deploy resources under the spoke VPC.

Deploying resources in spoke VPC



To deploy resources in the internal VPC of a spoke:

1. To allow for traffic to flow out of the `ncc_vpc_int` private subnet, you must add routes to the VPC route table. This step allows the FortiGate to control traffic coming in and out of the internal VPC. Do the following:
 - a. On the GCP management console, go to `VPC Networks > ncc_vpc_int > ROUTES`.
 - b. Click `ADD ROUTE`.
 - c. In the `Destination IP range` field, enter `0.0.0.0/0`.
 - d. In the `Priority` field, enter `1000`.
 - e. In the `Next hop IP address` field, enter the internal port 2 IP address of the spoke FortiGate. In the example, this is `192.168.215.2`.
 - f. Click `CREATE`.
2. Go to `VPC Networks > ncc_vpc_int > Firewall Rules` and add firewall rules to allow and block the required traffic based on the type of service deployed.

Validating the configuration

You can run the `get router info bgp neighbors <neighbor_IP> received-routes` and `get router info bgp neighbors <neighbor_IP> advertised-routes` commands on the side office and spoke FortiGates to validate the configuration.

The following shows the desired output for the side office FortiGate:

```
FGT[...]:# get router info bgp neighbors 169.254.120.1 received-routes
VRF 0 BGP table version is 17, local router ID is 169.254.120.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network      Next Hop          Metric LocPrf Weight RouteTag Path
*> 10.200.2.0/24    169.254.120.1           0        0 7252 65012 7255 7225 ? <-/->
*> 192.168.235.0   169.254.120.1           0        0 7252 i <-/->

Total number of prefixes 2

FGT[...]:# get router info bgp neighbors 169.254.120.1 advertised-routes
VRF 0 BGP table version is 17, local router ID is 169.254.120.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network      Next Hop          Metric LocPrf Weight RouteTag Path
*> 172.31.0.0/28   169.254.120.2           100    32768       0 i <-/->

Total number of prefixes 1
```

The following shows the desired output for the spoke FortiGate:

```
spoke2fgt # get router info bgp neighbors 169.254.120.2 received-routes
VRF 0 BGP table version is 3, local router ID is 169.250.254.254
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network      Next Hop          Metric LocPrf Weight RouteTag Path
*> 172.31.0.0/20   169.254.120.2           0        0 7226 i <-/->

Total number of prefixes 1

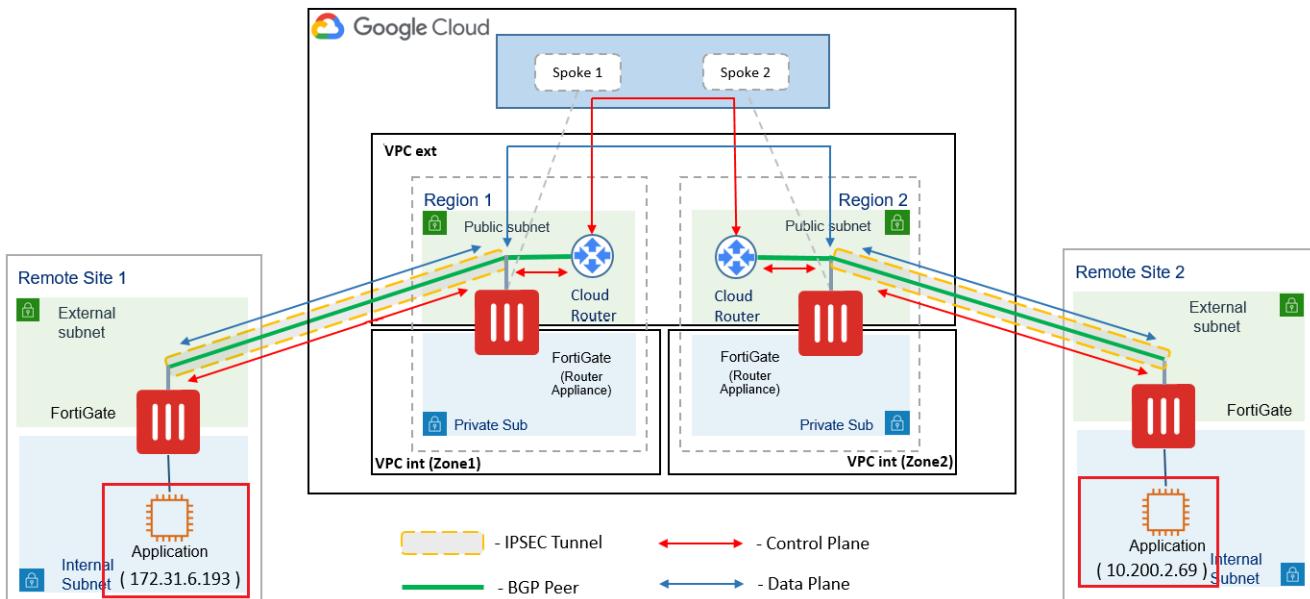
spoke2fgt # get router info bgp neighbors 169.254.120.2 advertised-routes
VRF 0 BGP table version is 3, local router ID is 169.250.254.254
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network      Next Hop          Metric LocPrf Weight RouteTag Path
*> 10.200.2.0/24   169.254.120.1           0        0 65012 7255 7225 ? <-/->
*> 192.168.235.0   169.254.120.1           100    32768       0 i <-/->

Total number of prefixes 2
```

To check the connected and BGP routes on the FortiGate, you can run the `get router info routing-table all` and `get router info routing-table database` commands. See [Technical Tip: FortiGate BGP configuration to announce specific routes and accept only a default route \(prefix list and route map\)](#).

Verifying site-to-site connectivity



You can verify site-to-site connectivity by pinging an application in remote site 2 from an application in remote site 1, and vice-versa. The following shows the desired output from a site 1 application instance:

```
ubuntu@ip-172-31-6-193:~$ ping 10.200.2.69 -c 5
PING 10.200.2.69 (10.200.2.69) 56(84) bytes of data.
64 bytes from 10.200.2.69: icmp_seq=1 ttl=60 time=52.8 ms
64 bytes from 10.200.2.69: icmp_seq=2 ttl=60 time=51.8 ms
64 bytes from 10.200.2.69: icmp_seq=3 ttl=60 time=51.5 ms
64 bytes from 10.200.2.69: icmp_seq=4 ttl=60 time=52.8 ms
64 bytes from 10.200.2.69: icmp_seq=5 ttl=60 time=51.9 ms

--- 10.200.2.69 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 51.581/52.231/52.875/0.594 ms
```

The following shows the desired output from a site 2 application instance:

```
ubuntu@ip-10-200-2-69:~$ ping 172.31.6.193 -c 5
PING 172.31.6.193 (172.31.6.193) 56(84) bytes of data.
64 bytes from 172.31.6.193: icmp_seq=1 ttl=60 time=52.7 ms
64 bytes from 172.31.6.193: icmp_seq=2 ttl=60 time=52.5 ms
64 bytes from 172.31.6.193: icmp_seq=3 ttl=60 time=52.0 ms
64 bytes from 172.31.6.193: icmp_seq=4 ttl=60 time=51.5 ms
64 bytes from 172.31.6.193: icmp_seq=5 ttl=60 time=51.8 ms

--- 172.31.6.193 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 51.534/52.128/52.749/0.531 ms
```

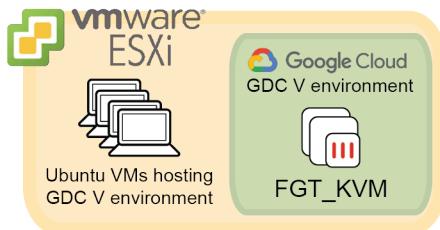
The following lists Google Cloud commands for deployment verification:

Resource	Describe command	Delete command
Cloud router	gcloud compute routers describe <cloud_router> --region=<region_name> Example: gcloud compute routers describe test-cloud-router --region=us-west1	gcloud compute routers delete <cloud_router> - --region=<region_name>

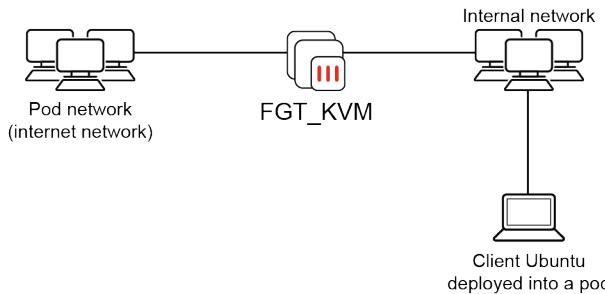
Resource	Describe command	Delete command
Hub	<pre>gcloud alpha network-connectivity hubs describe <ncc_hub></pre> <p>Example: gcloud alpha network-connectivity hubs describe test-hub</p>	<pre>gcloud alpha network-connectivity hubs delete <ncc_hub></pre>
Spoke	<pre>gcloud alpha network-connectivity spokes describe <spoke_name> --region=<region_name></pre> <p>Example: gcloud alpha network-connectivity spokes describe test-spoke --region=us-west1</p>	<pre>gcloud alpha network-connectivity spokes delete <spoke_name> --region=<region_name></pre>

GDC Virtual support

FortiGate-VM supports the Google Distributed Cloud Virtual (GDC V) environment. In the example, a KVM build of the FortiGate-VM is deployed into a GDC environment. The GDC V runs on a cluster of Ubuntu VMs.



The following diagram depicts traffic sent from the client through the FortiGate-VM to the internet:



The document divides the configuration into two procedures:

- Configuring the GDC V environment. See [To configure the GDC V environment: on page 148](#).
- Deploying and configuring the FortiGate-VM. See [To deploy and configure the FortiGate-VM: on page 154](#).

To configure the GDC V environment:

- Create four Ubuntu VMs as [Plan for a basic installation on your hardware](#) describes.
- Create the admin and user clusters on top of the four VM nodes as [Create basic clusters](#) describes. The following shows the example values for the information that you must gather before creating the clusters:

Information	Example value
Basic cluster information	
The name of the admin cluster you're creating. The location and naming of cluster artifacts on the admin workstation are based on the cluster name. The cluster namespace is derived from the cluster name.	admincluster
The name of the user cluster you're creating. The location and naming of cluster artifacts on the admin workstation are based on the cluster name. The cluster namespace is derived from the cluster name.	usercluster
The version of bmctl that you downloaded.	1.30.100-gke.96

Information	Example value
Account information	
The path to the SSH private key file on your admin workstation. By default, the path is /home/USERNAME/.ssh/id_rsa.	/home/aturner/.ssh/id_rsa
The ID of the Google Cloud project that you want to use for connecting your cluster to Google Cloud and viewing logs and metrics. This project is also referred to as the fleet host project.	dev-project-001-166400
The email address that is associated with your Google Cloud account. For example:alex@example.com.	aturner@example.com
Node machine IP addresses	
One IP address for the admin cluster control plane node.	172.16.200.71
One IP address for the user cluster control plane node.	172.16.200.72
One IP address for the user cluster worker node.	172.16.200.73
VIP addresses	
VIP for the Kubernetes API server of the admin cluster.	172.16.200.74
VIP for the Kubernetes API server of the user cluster.	172.16.200.75
One VIP to use as the external address for the ingress proxy.	172.16.200.76
Range of ten IP addresses for use as external IP addresses for Services of type LoadBalancer. Notice that this range includes the ingress VIP, which is required by MetalLB. No other IP addresses can overlap this range.	172.16.200.76-172.16.200.86
Pod and Service CIDRs	
Range of IP addresses in CIDR block notation for use by Pods on the admin cluster. The recommended starting value, which is pre-filled in the generated cluster configuration file is 192.168.0.0/16.	192.168.0.0/16
Range of IP addresses in CIDR block notation for use by Services on the admin cluster. The recommended starting value, which is pre-filled in the generated cluster configuration file is 10.96.0.0/20.	10.96.0.0/20
Range of IP addresses in CIDR block notation for use by Pods on the user cluster. The recommended starting value, which is pre-filled in the generated cluster configuration file and is the default value in the console is 192.168.0.0/16.	192.168.0.0/16
Range of IP addresses in CIDR block notation for use by Services on the user cluster. The recommended starting value, which is pre-filled in the generated cluster configuration file and is the default value in the console is 10.96.0.0/20.	10.96.0.0/20

3. In Google Cloud, go to *Clusters*. Select the clusters that you created and confirm that you can see the clusters connected on Google Kubernetes Engine (GKE).

The screenshot shows the Google Cloud Kubernetes Engine interface. The left sidebar is titled 'Kubernetes Engine' and has sections for 'ENTERPRISE', 'All Fleets', 'Fleet' (dev-project-001-166400 fleet), 'Resource Management' (Overview, Clusters, Workloads, Teams, Applications, Secrets & ConfigMaps, Storage, Object Browser, Rollout Sequencing NEW, Backup for GKE), 'Posture Management' (Security, Compliance PREVIEW, Policy), 'Networking' (Gateways, Services & Ingress, Network Function Optimizer), 'Features' (Feature Manager, Service Mesh, Config, Identity Service), 'Marketplace', and 'Release Notes'. The main content area is titled 'Clusters' and shows 'Admin cluster: admincluster'. It displays a summary box with 'Cluster status: running' (last operation finished 4 days ago) and a 'Show details' link. Below this are tabs for 'DETAILS' (selected), 'USER CLUSTERS', and 'LOGS'. The 'Cluster basics' section lists the following details:

Name	admincluster
Platform Type	Bare metal
Deployment type	Admin cluster
Google Cloud API location	us-central1
GDC software for bare metal version	1.30.100-gke.96
Kubernetes API version	v1.30.4-gke.1800
GKE Hub Membership	admincluster
Kubernetes API endpoint	172.16.200.74
Create time	Thu Nov 14 2024 20:32:50 GMT-0800 (Pacific Standard Time)
Maximum Pods per Node	250

The 'UPGRADE AVAILABLE' badge is present next to the GDC software version. The 'Authorization' section lists 'Read-only users'. The 'Networking' section lists:

Control plane node IPs	172.16.200.71
Load balancer type	Bundled with MetalLB
Control plane operating system	Linux
Control plane VIP	172.16.200.74
Control plane load balancer port	443
Service CIDRs	10.96.0.0/20
Pod CIDRs	192.168.0.0/16
Maximum Pods per node	250

The 'Storage' section lists:

Local volume provisioner node mount path	/mnt/localpv-disk
Local volume provisioner node storage class name	local-disks
Local volume provisioner share path	/mnt/localpv-share
Local volume provisioner share storage class name	local-shared
Number of subdirectories under shared path	5

- To enable multiple NICs for a pod or VM, you must enable it in usercluster.yaml as [Configure multiple network interfaces for Pods](#) describes, specifically to include the following:

```
apiVersion: v1
multipleNetworkInterfaces: true
enableDataplaneV2: true
```

- On the admin workstation, run the following to enable vmruntime on the user cluster to allow VM virtualization:

```
bmctl enable vmruntime --kubeconfig bmctl-workspace/usercluster/usercluster-kubeconfig
```

6. Create a separate yaml file to create the NetworkAttachmentDefinition (NAD) based on the following yaml. This creates a network definition that you can attach to pods or the FortiGate-VM so that they can communicate on the same internal subnet:

```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  name: test-bridge
spec:
  config: '{ "cniVersion": "0.3.1", "type": "bridge", "bridge": "br0", "ipam": { "type": "host-local", "subnet": "172.16.1.0/24" } }'
```

7. Create the DataVolume for the FortiGate-VM in a separate yaml file. You must download the qcow2 file from the KVM FortiGate-VM image from the [Fortinet Support site](#) and place it in an accessible location for the image creation to succeed:

```
apiVersion: cdi.kubevirt.io/v1beta1
kind: DataVolume
metadata:
  name: "fgt-boot-dv"
spec:
  source:
    http:
      url: "https://alextestbucket.s3.ap-southeast-1.amazonaws.com/fos3401.qcow2" # S3
or GCS
  pvc:
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: "5000Mi"
```

8. Create the FortiGate-VM for KVM instance using the boot disk created in step 7 and a secondary interface. The interface configuration in this yaml file uses `multus=test-bridge` which is defined in step 6 for eth1 and a default network name bridge, which is a system default and should not be changed in this configuration file.

```
apiVersion: vm.cluster.gke.io/v1
kind: VirtualMachine
metadata:
  creationTimestamp: null
  labels:
    kubevirt/vm: fgt
  name: fgt
  namespace: default
spec:
  compute:
    cpu:
      vcpus: 2
    memory:
      capacity: 4Gi
  disks:
    - boot: true
      driver: virtio
      virtualMachineDiskName: fgt-boot-dv
  guestEnvironment: {}
  interfaces:
    - default: true
      name: eth0
```

```
    networkName: bridge
```

```
    - name: eth1
```

```
        networkName: multus=test-bridge
```

```
        osType: Linux
```

```
        status: {}
```

9. Create an SSH server on an pod or container by creating a yaml file as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: ssh-pod
  labels:
    app: ssh-server
  annotations:
    k8s.v1.cni.cncf.io/networks: test-bridge
spec:
  containers:
    - name: ssh-server
      image: ubuntu:20.04
      command:
        - /bin/bash
        - -c
        - |
          apt-get update && \
          apt-get install -y openssh-server && \
          mkdir -p /run/sshd && \
          echo 'root:Fortinet123#' | chpasswd && \
          echo 'PermitRootLogin yes' >> /etc/ssh/sshd_config && \
          echo 'PasswordAuthentication yes' >> /etc/ssh/sshd_config && \
          service ssh start && \
          while true; do sleep 3600; done
  ports:
    - containerPort: 22
  securityContext:
    privileged: true # Needed for sshd
---
apiVersion: v1
kind: Service
metadata:
  name: ssh-service
spec:
  type: NodePort
  selector:
    app: ssh-server
  ports:
    - port: 22
      targetPort: 22
      nodePort: 30022 # You can change this port. If you change it, you must specify
                      the port number on your SSH connection string. For example: ssh root@172.16.200.73
```

-p 30022

10. From the admin workstation instance, apply the created yaml files from step 6 through 9 using `kubectl apply -f example.yaml`. Applying the yaml files creates the resources that the files define.
11. From the adminworkstation instance use `kubectl get vmi` to confirm that the VMs are visible and running, and that you can reach them from the worker node through their pod-network IP address:

```
aturner@adminworkstation:~$ kubectl get vmi
NAME          AGE   PHASE   IP           NODENAME
READY
ssh-pod        1/1   Running  0            8d
virt-launcher-fgt-6d5nh  2/2   Running  0            8d

aturner@userclusterworkernode:~$ ping 192.168.2.202
PING 192.168.2.202 (192.168.2.202) 56(84) bytes of data.
64 bytes from 192.168.2.202: icmp_seq=1 ttl=254 time=0.650 ms
^C
--- 192.168.2.202 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.650/0.650/0.650/0.000 ms
aturner@userclusterworkernode:~$ ping 192.168.2.29
PING 192.168.2.29 (192.168.2.29) 56(84) bytes of data.
64 bytes from 192.168.2.29: icmp_seq=1 ttl=63 time=0.438 ms
^C
--- 192.168.2.29 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.438/0.438/0.438/0.000 ms
aturner@userclusterworkernode:~$ ssh admin@192.168.2.202
admin@192.168.2.202's password:
FGVM08TM... # get sys stat
Version: FortiGate-VM64-KVM v7.6.0,build3401,240724 (GA.F)
First GA patch build date: 240724
Security Level: 2
Firmware Signature: certified
Virus-DB: 92.08924(2024-11-19 16:31)
Extended DB: 92.08924(2024-11-19 16:30)
Extreme DB: 1.00000(2018-04-09 18:07)
AV AI/ML Model: 3.01796(2024-11-19 15:50)
IPS-DB: 6.00741(2015-12-01 02:30)
IPS-ETDB: 6.00741(2015-12-01 02:30)
APP-DB: 6.00741(2015-12-01 02:30)
Proxy-IPS-DB: 6.00741(2015-12-01 02:30)
Proxy-IPS-ETDB: 6.00741(2015-12-01 02:30)
Proxy-APP-DB: 6.00741(2015-12-01 02:30)
FMWP-DB: 24.00111(2024-11-06 13:20)
IPS Malicious URL Database: 1.00001(2015-01-01 01:01)
IoT-Detect: 0.00000(2022-08-17 17:31)
OT-Detect-DB: 0.00000(2001-01-01 00:00)
OT-Patch-DB: 0.00000(2001-01-01 00:00)
OT-Threat-DB: 6.00741(2015-12-01 02:30)
IPS-Engine: 7.01014(2024-07-02 21:57)
Serial-Number: FGVM08TM...
License Status: Valid
License Expiration Date: 2025-08-24
VM Resources: 2 CPU/8 allowed, 3946 MB RAM
Log hard disk: Not available
```

```
Hostname: FGVM08TM...
Private Encryption: Disable
Operation Mode: NAT
Current virtual domain: root
Max number of virtual domains: 10
Virtual domains status: 1 in NAT mode, 0 in TP mode
Virtual domain configuration: disable
FIPS-CC mode: disable
Current HA mode: standalone
Branch point: 3401
Release Version Information: GA
FortiOS x86-64: Yes
System time: Tue Nov 19 17:07:49 2024
Last reboot reason: warm reboot
```

To deploy and configure the FortiGate-VM:

The test environment uses an SSH session to access the SSH server pod or container and through that session, triggers an EICAR test file download that flows through the FortiGate and triggers UTM processing via a firewall policy.

1. Upload a license to the FortiGate-VM:

```
FortiGate-VM64-KVM # execute restore vmlicense ftp workingfolder/FGVM08TM....lic
...86.126 **omitted**
This operation will overwrite the current VM license and reboot the system!
Do you want to continue? (y/n)y
```

Please wait...

```
Connect to ftp server ...86.126 ...
Get VM license from ftp server OK.
VM license install succeeded. Rebooting firewall.
```

2. The primary interface obtains its IP address using DHCP. Therefore, the NAD is the only address that you must configure. Configure the IP address in FortiOS and on the Ubuntu pod using the IP address that the NAD provides:

```
kubectl describe vmi fgt
...
    Ip Address: 172.16.1.250
    Ip Addresses:
        172.16.1.250
...
FGVM08TM24003117 (port2) # show
config system interface
    edit "port2"
        set vdom "root"
        set ip 172.16.1.250 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-response
fabric ftm speed-test
    set type physical
    set snmp-index 2
    set mtu-override enable
next
end
FGVM08TM24003117 # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
```

```
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
V - BGP VPNv4
* - candidate default
```

```
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 192.168.3.33, port1, [1/0]
C      172.16.1.0/24 is directly connected, port2
C      192.168.2.202/32 is directly connected, port1
S      192.168.3.33/32 [5/0] is directly connected, port1, [1/0]
```

3. Configure a firewall policy with unified threat management (UTM) and an antivirus (AV) profile:

```
config firewall policy
edit 1
set uuid 2864e7e4-a6d7-51ef-cc59-2a9e5ff5a48e
set srcintf "port2"
set dstintf "port1"
set action accept
set srcaddr "all"
set dstaddr "all"
set schedule "always"
set service "ALL"
set utm-status enable
set av-profile "default"
set nat enable
next
end
```

4. Configure the Ubuntu server with the route pointing to the FortiGate port2 address. In the example, the server IP address is ...86.126:

```
root@ssh-pod:~# ip route show
default via 192.168.3.33 dev eth0 mtu 1450
...86.126 via 172.16.1.250 dev net1
172.16.1.0/24 dev net1 proto kernel scope link src 172.16.1.253
192.168.3.33 dev eth0 scope link
root@ssh-pod:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: net1@if69: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 2a:a9:65:6f:lc:bc brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.16.1.253/24 brd 172.16.1.255 scope global net1
        valid_lft forever preferred_lft forever
    inet6 fe80::28a9:65ff:fe6f:1cbc/64 scope link
        valid_lft forever preferred_lft forever
67: eth0@if68: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether be:d5:28:86:c2:27 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.3.179/32 scope global eth0
```

```
    valid_lft forever preferred_lft forever
inet6 fe80::bcd5:28ff:fe86:c227/64 scope link
    valid_lft forever preferred_lft forever
```

5. To test the configuration, attempt to use cURL to download an Eicar file to the server. Confirm that the UTM and AV features are active and block download of the Eicar file:

```
root@ssh-pod:~# curl http://...86.126/samplevirus/eicar.txt
<!DOCTYPE html>
<html lang="en">
  **omitted**
  <h1>High Security Alert</h1>
  <p>You are not permitted to download the file "eicar.txt" because it is infected
with the virus "EICAR_TEST_FILE".</p>
  <table><tbody>
    <tr>
      <td>URL</td>
      <td>http://...86.126/samplevirus/eicar.txt</td>
    </tr>
    <tr>
      <td>Quarantined File Name</td>
      <td></td>
    </tr>
    <tr>
      <td>Reference URL</td>
      <td><a href="https://fortiguard.com/encyclopedia/virus/2172">https://fortiguard.com/encyclopedia/virus/2172</a></td>
    </tr>
  </tbody></table>
</div></body>
</html>
```

Enhancing FortiGate-VM performance

Deploying a gVNIC interface

The new GCP gVNIC interface is supported, which offers improved performance and bandwidth and is required on some VM shapes tuned for optimal performance.



A VM with gVNIC must be deployed with the CLI or API. Refer to the [Using Google Virtual NIC](#) documentation for other limitations. If you are upgrading from prior images that support virtIO, the images will remain that way.

Refer to [Creating a VM that uses gVNIC](#) for detailed instructions. The following example shows sample commands used to create an instance.

To deploy a gVNIC with the gcloud CLI:

1. Create a gVNIC enabled image using the FortiGate marketplace image.

```
gcloud compute --project=dev-project-000-000000 images create gcp-ond-700-gvnic --source-image=fortinet-fgtondemand-700-20210407-000-w-license --source-image-project=fortigcp-project-000 --guest-os-features=GVNIC
```

2. Deploy the instance with the gVNIC image and gVNIC specification in the parameter:

```
gcloud compute --project=dev-project-000-000000 instances create xxxxxxxx-script-ond-0128-gvnic --zone=us-central1-c --machine-type=n1-standard-1 --network-interface nic-type=GVNIC,subnet=xxxxxx-hapvc-port1external,private-network-ip=10.0.0.15,address=**.*.*.*** --network-interface nic-type=GVNIC,subnet=xxxxxx-hapvc-port2internal,private-network-ip=10.0.1.15,no-address --can-ip-forward --maintenance-policy=MIGRATE --service-account=*****@compute@developer.gserviceaccount.com --scopes=https://www.googleapis.com/auth/cloud-platform --image=gcp-ond-0128-gvnic --image-project=dev-project-000-000000 --boot-disk-type=pd-standard --boot-disk-device-name=xxxxxx-script-ond-0128  
Created [https://www.googleapis.com/compute/beta/projects/dev-project-000-000000/zones/us-central1-c/instances/xxxxxx-script-ond-0128-gvnic].  
NAME ZONE MACHINE_TYPE PREEMPTIBLE INTERNAL_IP  
EXTERNAL_IP STATUS  
xxxxxx-script-ond-0128-gvnic us-central1-c n1-standard-1 10.6.30.5  
**.*.*.*** RUNNING
```

3. Verify that gVNIC is enabled for the NIC:

```
gcloud compute instances describe xxxxxxxx-script-ond-0128-gvnic --zone=us-central1-c  
...  
guestOsFeatures:  
- type: GVNIC  
...  
name: xxxxxx-script-ond-0128-gvnic  
networkInterfaces:  
- accessConfigs:
```

```

- kind: compute#accessConfig
name: external-nat
natIP: **.*.*.**
networkTier: PREMIUM
type: ONE_TO_ONE_NAT
fingerprint: OiB_2ejfR-g=
kind: compute#networkInterface
name: nic0
network: https://www.googleapis.com/compute/v1/projects/xxx-xxxxxx-000-
000000/global/networks/xxxxxxxx
networkIP: 10.6.30.5
nicType: GVNIC
...

```

4. Log in to the FortiGate using SSH and verify that the drivers are correct:

```

# diagnose hardware lspci -v
00:04.0 Class 0200: Device 1ae0:0042
    Subsystem: Device 1ae0:0058
    Flags: bus master, fast devsel, latency 0, IRQ 11
    Memory at feb01000 (32-bit, non-prefetchable) [size=4K]
    Memory at feb02000 (32-bit, non-prefetchable) [size=64]
    Memory at fea00000 (32-bit, non-prefetchable) [size=1M]
    Capabilities: [80] MSI-X: Enable+ Count=3 Masked-
        Kernel driver in use: gvnic

# diagnose hardware deviceinfo nic port1
Name:          port1
Driver:        gve
Version:       1.2.0
Bus:          0000:00:04.0
Hwaddr:        **:**:**:**:**:**:***
Permanent Hwaddr: **:**:**:**:**:**:***
State:         up
Link:          up
Mtu:           1460
Supported:
Advertised:
Auto:          disabled

```

DPDK support

You can now enable DPDK on FortiGate-VMs deployed on the Google Cloud Platform. DPDK allows improved network performance.



Enabling DPDK in polling mode may result in high CPU usage. For GCP, a VM vCPU is allocated as a dedicated vSPU, resulting in one vCPU's usage to be higher or 100% for IPS in most cases.

The following example enables DPDK on a FortiGate-VM deployed on Google Cloud, passes UDP and TCP traffic with an antivirus (AV)/IPS/application firewall policy enabled, then checks the engine and vNP statistics.

To enable DPDK on a FortiGate-VM deployed on Google Cloud:

1. In the FortiOS CLI, enable DPDK, reboot, then check the DPDK status:

```

config dpdk global
(global) # set status enable
(global) # get
status          : enable
interface      :
multiqueue    : disable
sleep-on-idle  : disable
elasticbuffer  : disable
per-session-accounting: traffic-log-only
ipsec-offload   : disable
hugepage-percentage : 30
mbufpool-percentage : 25

(global) # set interface port1 port2 port3 port4
(global) # set multiqueue enable
(global) # set sleep-on-idle enable
(global) # set elasticbuffer enable
(global) # end
status, interface change will trigger system reboot and will take effect after the
reboot.
Enabling DPDK will adjust Tx/Rx ring size to max allowable value by PMD for the best
performance.
Do you want to continue? (y/n)y

config dpdk global
  set status enable
  set interface "port1" "port2" "port3" "port4"
  set multiqueue enable
  set sleep-on-idle enable
  set elasticbuffer enable
  set per-session-accounting traffic-log-only
  set ipsec-offload disable
  set hugepage-percentage 30
  set mbufpool-percentage 25
end

```

2. Check early initialization logs:

```

diagnose dpdk log show early-init
-----
DPDK early initialization starts at 2022-03-23 04:58:00(UTC)
-----
Content of DPDK configuration: (Use cmdb configuration)

config dpdk global
  set status enable
  set interface "port1" "port2" "port3" "port4"
  set multiqueue enable
  set sleep-on-idle enable
  set elasticbuffer enable
  set per-session-accounting traffic-log-only
  set ipsec-offload disable
  set hugepage-percentage 30

```

```
        set mbufpool-percentage 25
    end
    config dpdk cpus
        set rx-cpus "all"
        set vnp-cpus "all"
        set ips-cpus "all"
        set tx-cpus "all"
    end

Parse config success!

Check CPU definitions 'rx-cpus'
Check CPU definitions 'vnp-cpus'
Check CPU definitions 'ips-cpus'
Check CPU definitions 'tx-cpus'
Check CPU definitions 'isolated-cpus'
Check CPUs success!

Huge page allocation done

Ports enabled for DPDK:
    port1
    port2
    port3
    port4
Port name to device name mapping:
    port1: eth0
    port2: eth1
    port3: eth2
    port4: eth3
    port5: eth4
    port6: eth5
    port7: eth6
    port8: eth7
    port9: eth8
    port10: eth9
    port11: eth10
    port12: eth11
    port13: eth12
    port14: eth13
    port15: eth14
    port16: eth15
    port17: eth16
    port18: eth17
    port19: eth18
    port20: eth19
    port21: eth20
    port22: eth21
    port23: eth22
    port24: eth23

Start enabling DPDK kernel driver for port 'port1'...
Getting PCI device info for eth0...
reading pci dev /sys/class/net/eth0
link path: ../../devices/pci0000:00/0000:00:04.0/virtio1/net/eth0
Device info of eth0:
```

```
dev_name: eth0
macaddr: 42:01:0a:00:00:0f
pci_vendor: 0x1af4
pci_device: 0x1000
pci_id: 0000:00:04.0
pci_domain: 0
pci_bus: 0
pci_devid: 4
pci_function: 0
guid: n/a
Unbinding device eth0 from kernel driver...
Device eth0 unbind from kernel driver successful
Binding device eth0 to DPDK driver...
Device eth0 bind to DPDK driver successful
Creating DPDK kernel driver for device eth0...
Add VNP dev: eth0 PCI: 0000:00:04.0, Succeeded
DPDK kernel driver for eth0 successfully created
DPDK kernel driver enabled for port 'port1' (device name 'eth0')

Start enabling DPDK kernel driver for port 'port2'...
Getting PCI device info for eth1...
reading pci dev /sys/class/net/eth1
link path: ../../devices/pci0000:00/0000:00:05.0/virtio2/net/eth1
Device info of eth1:
  dev_name: eth1
  macaddr: 42:01:0a:00:01:0f
  pci_vendor: 0x1af4
  pci_device: 0x1000
  pci_id: 0000:00:05.0
  pci_domain: 0
  pci_bus: 0
  pci_devid: 5
  pci_function: 0
  guid: n/a
Unbinding device eth1 from kernel driver...
Device eth1 unbind from kernel driver successful
Binding device eth1 to DPDK driver...
Device eth1 bind to DPDK driver successful
Creating DPDK kernel driver for device eth1...
Add VNP dev: eth1 PCI: 0000:00:05.0, Succeeded
DPDK kernel driver for eth1 successfully created
DPDK kernel driver enabled for port 'port2' (device name 'eth1')

Start enabling DPDK kernel driver for port 'port3'...
Getting PCI device info for eth2...
reading pci dev /sys/class/net/eth2
link path: ../../devices/pci0000:00/0000:00:06.0/virtio3/net/eth2
Device info of eth2:
  dev_name: eth2
  macaddr: 42:01:0a:00:02:0f
  pci_vendor: 0x1af4
  pci_device: 0x1000
  pci_id: 0000:00:06.0
  pci_domain: 0
  pci_bus: 0
  pci_devid: 6
```

```
pci_function: 0
guid: n/a
Unbinding device eth2 from kernel driver...
Device eth2 unbind from kernel driver successful
Binding device eth2 to DPDK driver...
Device eth2 bind to DPDK driver successful
Creating DPDK kernel driver for device eth2...
Add VNP dev: eth2 PCI: 0000:00:06.0, Succeeded
DPDK kernel driver for eth2 successfully created
DPDK kernel driver enabled for port 'port3' (device name 'eth2')

Start enabling DPDK kernel driver for port 'port4',...
Getting PCI device info for eth3...
reading pci dev /sys/class/net/eth3
link path: ../../devices/pci0000:00/0000:00:07.0/virtio4/net/eth3
Device info of eth3:
  dev_name: eth3
  macaddr: 42:01:0a:00:03:0f
  pci_vendor: 0x1af4
  pci_device: 0x1000
  pci_id: 0000:00:07.0
  pci_domain: 0
  pci_bus: 0
  pci_devid: 7
  pci_function: 0
  guid: n/a
Unbinding device eth3 from kernel driver...
Device eth3 unbind from kernel driver successful
Binding device eth3 to DPDK driver...
Device eth3 bind to DPDK driver successful
Creating DPDK kernel driver for device eth3...
Add VNP dev: eth3 PCI: 0000:00:07.0, Succeeded
DPDK kernel driver for eth3 successfully created
DPDK kernel driver enabled for port 'port4' (device name 'eth3')
Bind ports success!

mknod for uio0 (254, 0) done.
mknod for uio1 (254, 1) done.
mknod for uio2 (254, 2) done.
mknod for uio3 (254, 3) done.
Make UIO nodes success!

-----EAL INIT-----
-----

#-----#
# port  oid      dev_name      pci_id
#-----#
  0    0      eth0 0000:00:04.0
  1    1      eth1 0000:00:05.0
  2    2      eth2 0000:00:06.0
  3    3      eth3 0000:00:07.0
#-----#
DPDK sanity test passed
```

3. Pass UDP and TCP traffic with AV/IPS/application firewall policy enabled, then check engine and vNP statistics:

```
diagnose dpdk statistics show engine
```

```
-----  
FortiOS DPDK Helper Engine Stats  
-----
```

Engine 2	Engine 3	Total	Engine 0	Engine 1
CPU ID:			0	1
2	3			
----- DPDK RX Stage -----				
dpdkrx_rx_pkts:	5	2610346	87916	2521121
	1304			
dpdkrx_tx_pkts:	5	2610346	87916	2521121
	1304			
dpdkrx_drop_pkts:	0	0	0	0
	0			
dpdkrx_drop_multiseg_pkts:	0	0	0	0
	0			
dpdkrx_elstcbuf_in_num:	0	0	0	0
	0			
dpdkrx_elstcbuf_out_num:	0	0	0	0
	0			
dpdkrx_monitor_rx_cnt:	0	0	0	0
	0			
----- VNP Stage -----				
vnp_rx_from_kernel_pkts:	10159	30974	6260	6161
	8394			
vnp_rx_pkts:	654737	2610346	720505	793687
	441417			
vnp_tx_pkts:	653882	2608246	723777	788462
	442125			
vnp_tx_drop_pkts:	0	0	0	0
	0			
vnp_to_ips_pkts:	652	2738	885	656
	545			
vnp_to_ips_drop_pkts:	0	0	0	0
	0			
vnp_to_vnp_pkts:	0	0	0	0
	0			
vnp_to_vnp_drop_pkts:	0	0	0	0
	0			
vnp_to_kernel_pkts:	10342	30289	2090	10709
	7148			
ipsec_dec_pkts:	0	0	0	0
	0			
ipsec_enc_pkts:	0	0	0	0
	0			
ipsec_sa_add:	0	0	0	0
	0			
ipsec_sa_upd:	0	0	0	0
	0			
ipsec_sa_del:		0	0	0

0	0	0	0	0
ipsec_spi_add:	0	0	0	0
0	0	0	0	0
ipsec_spi_add_fail:	0	0	0	0
0	0	0	0	0
ipsec_spi_del:	0	0	0	0
0	0	0	0	0
ipsec_spi_del_fail:	0	0	0	0
0	0	0	0	0
ipsec_spi_lookup:	0	0	0	0
0	0	0	0	0
ipsec_spi_lookup_fail:	0	0	0	0
0	0	0	0	0
ipsec_spi_reclaim:	0	0	0	0
0	0	0	0	0
ipsec_ib_sa_hit:	0	0	0	0
0	0	0	0	0
ipsec_ib_sa_miss:	0	0	0	0
0	0	0	0	0
ipsec_ib_headroom_err:	0	0	0	0
0	0	0	0	0
ipsec_ib_cryptodev_err:	0	0	0	0
0	0	0	0	0
ipsec_ib_post_proc_err:	0	0	0	0
0	0	0	0	0
ipsec_ib_uesp_dport_err:	0	0	0	0
0	0	0	0	0
ipsec_ib_uesp_not_enabled:	0	0	0	0
0	0	0	0	0
ipsec_ob_sa_hit:	0	0	0	0
0	0	0	0	0
ipsec_ob_sa_miss:	0	0	0	0
0	0	0	0	0
ipsec_ob_headroom_err:	0	0	0	0
0	0	0	0	0
ipsec_ob_cryptodev_err:	0	0	0	0
0	0	0	0	0
ipsec_ob_post_proc_err:	0	0	0	0
0	0	0	0	0
<hr/>				
----- IPS Stage -----				
ips_rx_pkts:	705	2738	657	698
678				
ips_tx_pkts:	705	2738	657	698
678				
ips_drop_pkts:	0	0	0	0
0				
ips_vdct_pkts:	0	0	0	0
0				
ips_inv_pkts:	0	0	0	0
0				
from_ips_rx_pkts:	0	0	0	0
0				
from_ips_tx_pkts:	0	0	0	0
0				
from_ips_drop_pkts:	0	0	0	0

```

0          0
from_ipsFallback_pkts:      0          0          0          0
0          0

----- DPDK TX Stage -----
dpdktx_rx_pkts:           2610984    2522231    86925
893         935
dpdktx_tx_pkts:           2610984    2522231    86925
893         935
dpdktx_drop_pkts:          0          0          0
0          0
dpdktx_drop_oversized_pkt: 0          0          0
0          0

diagnose dpdk statistics show vnp

-----
FortiOS DPDK Helper VNP Stats
-----

          Total     Engine 0     Engine 1
Engine 2     Engine 3
CPU ID:          0          1
2             3

----- VNP Internal -----
ctr_sse:           224038    68362    49639
50015        56022
ctr_sse_cmd:       168        62        34
39            33
ctr_sse_delmiss:   0          0          0
0            0
ctr_sse_msg:        113        48        11
17            37
ctr_sse_pruned:     0          0          0
0            0

vnp_st_rx_from_dpdkrx: 2610531    720527    793688
654804        441512
vnp_st_sse_proc:    2582749    718835    783299
644776        435839
vnp_st_tx_to_kernel: 30474      2112      10710
10409        7243
vnp_st_ipsec_ib:      0          0          0
0            0
vnp_st_ipsec_ob:      0          0          0
0            0
vnp_st_fpath_proc:   2580170    718463    782989
644412        434306
vnp_st_tx_to_dpdktx: 2608486    723802    788463
653996        442225
vnp_st_tx_to_ips:    2738       885       656
652          545
vnp_st_rx_from_kernel: 31222     6286      6164
10275        8497

```

vnp_st_sse_cmd:		168	62	34
39	33			
vnp_st_final:		33380	3059	11400
11100	7821			
ctr_sse_entries:		8	2	2
3	1			
err_sse_batch_size:		0	0	0
0	0			
err_sse_unknown_cmd:		0	0	0
0	0			
err_sse_full:		0	0	0
0	0			
err_sse_tbl_alloc_fail:		0	0	0
0	0			
err_sse_inv_oid:		0	0	0
0	0			
err_fp_no_act:		0	0	0
0	0			
err_fp_no_port:		0	0	0
0	0			
drop_inv_13:		0	0	0
0	0			
drop_inv_14:		0	0	0
0	0			
drop_fp_act:		0	0	0
0	0			
drop_inv_port:		0	0	0
0	0			
drop_inv_ip_cksum:		0	0	0
0	0			
drop_oversized_pkt:		0	0	0
0	0			
drop_unsupported:		0	0	0
0	0			
drop_looping_pkt:		0	0	0
0	0			
drop_ipsec_ob_fail:		0	0	0
0	0			

Change log

Date	Change description
2024-07-25	Initial release.
2024-11-27	Updated About FortiGate-VM for GCP on page 6.
2024-11-28	Added GDC Virtual support on page 148. Updated Machine type support on page 6.



www.fortinet.com

Copyright© 2024 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.