Document for the deployment of a Check Point gateway or management in Google Cloud using gcloud shell

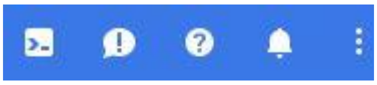
The gcloud shell is a useful tool for automation in Google Cloud, and this documentation goes over how to leverage this tool to deploy Check Point products in the cloud to allow for quick and efficient deployment and redeployment of next generation firewalls into the cloud.

This deployment guide assumes you have already installed gcloud command line. If not, you can install it by following the steps outline in this guide
https://cloud.google.com/compute/docs/gcloud-compute/
Also you should have putty installed or some other method of generating SSH keys and SSHing into instances.

Instead of the Google cloud launcher you can also choose to deploy R80.10 Check Point products using gcloud shell. (Note: these deployments will not show up in the deployment manager since they were not deployed with cloud launcher)

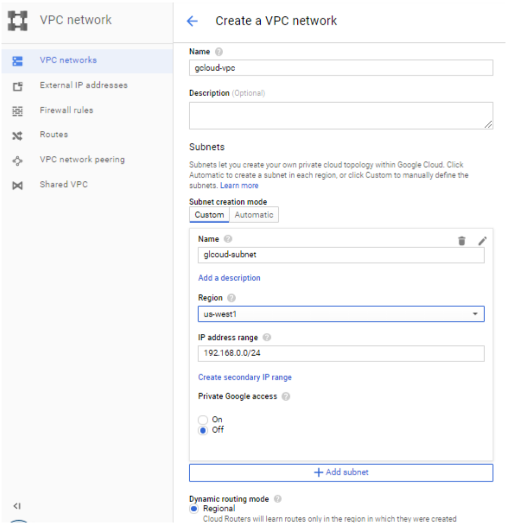
Important note: all gcloud commands are for your reference only and you will need to change them to suit your environment. More information about gcloud shell can be found at https://cloud.google.com/sdk/gcloud/reference/compute/

Click the gcloud icon in the top right hand corner is the left most icon on the following picture. A terminal will pop up at the bottom of the screen. This is where we will enter the gcloud commands.



Creating VPCs and subnets

Before we deploy the Check Point gateway/management make sure you have created the relevant VPCs and subnets. These can be created with the GUI in the VPC network section of GCP.



Name: gcloud-vpc

Description (Optional)

Subnets

Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. Learn more

Subnet creation mode: Custom | Automatic

Name: glcoud-subnet

Add a description

Region: us-west1

IP address range: 192.168.0.0/24

Create secondary IP range

Private Google access

○ On
● Off

+ Add subnet

Dynamic routing mode

● Regional
Cloud Routers will learn routes only in the region in which they were created

Or they can be created with the follow example gcloud commands:

```
gcloud compute networks create gcloud-vpc \
--project=gcloud-deployment \
--mode=subnet-mode

gcloud compute networks subnets create gcloud-subnet \
--project=gcloud-deployment \
--network=gcloud-vpc \
--region=us-west1 \
--range=192.168.0.0/24
```

Deploying the Check Point Gateway/Management

To deploy the Check Point use the following gcloud command but there are many options that can be changed to suit your project.

```
gcloud compute instances create "test-gateway" \
--can-ip-forward \
--project "checkpoint-cluster" \
--zone "us-west1-a" \
--boot-disk-size "100" \
--boot-disk-type "pd-standard" \
--boot-disk-device-name "test-gateway" \
--machine-type "n1-standard-2" \
--service-account "48435443454@cloudservices.gserviceaccount.com" \
--tags "checkpoint-gateway" \
--network-interface network='cluster-vpc-one',subnet='subnet-one',private-network-ip=192.168.1.12  \
--network-interface network='cluster-vpc-sync',subnet='subnet-sync',private-network-ip=192.168.5.12,no-address
\
--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
" \
--maintenance-policy "MIGRATE" \
--min-cpu-platform "Automatic" \
--image "check-point-r8010-byol-013-233-v20170906" \
--image-project "checkpoint-public" \
```

- “test-gateway” will be the name of the Check Point product
- --can-ip-forward should only be included for a gateway deployment. Delete this flag if you are deployment a management server
- Change the project name to your project
- Change the zone to your zone
- --boot-disk-size is storage size in GB. This must be greater than 100GB
- --boot-disk-type is the type of storage you can choose from a HHD or SSD use “pd-standard” for a HHD or “pd-ssd” for a SSD
- --boot-disk-device-name is the name for the storage device
- --machine-type is the type of instance.

        - You can choose from some predefined instances https://cloud.google.com/compute/docs/machine-types#custom_machine_types
        - or you can create a custom instance size using the –custom-cpu and –custom-memory flags (note: these must be integers) https://cloud.google.com/compute/docs/instances/creating-instance-with-custom-machine-type
            - Ex:
            - --custom-cpu 8
            - --custom-memory 8
    - --service-account is the service account you want to associate with the instance these can be found under IAM in GCP
    - --tags these are the tags for the instance for good practice add the tag checkpoint-gateway when creating a gateway and checkpoint-management when creating management
    - --network-interface specifies a network interface. You can have up to 8 interfaces, as long as the machine-type you choose supports it. There are multiple flags you can include for each network interface
        - network= This specifies the network/VPC the interface will be attached to
        - subnet= This specifies the subnet the interface will be attached to
        - private-network-ip= This specifies the private ip of the instance. If you wish for a private IP to be assigned automatically do not include this flag
        - for public IP there are three options (you can also change the public IP after creation)
            - do not include another flag: in this case an ephemeral public IP with be assigned
            - include the flag "no-address" in this case no public IP will be assigned
            - include address='reserved-address' in this case you can specify a public address. Where 'reserved-address' is a static IP you have already reserved.
    - Leave the last 5 flags as they are. You can also add other flags, but that is outside of the scope of this document. For further reference on this subject: https://cloud.google.com/sdk/gcloud/reference/compute/instances/create

Once you have changed all the settings you can paste the command into gcloud and it should create your instance.

Creating Google Cloud firewall rules

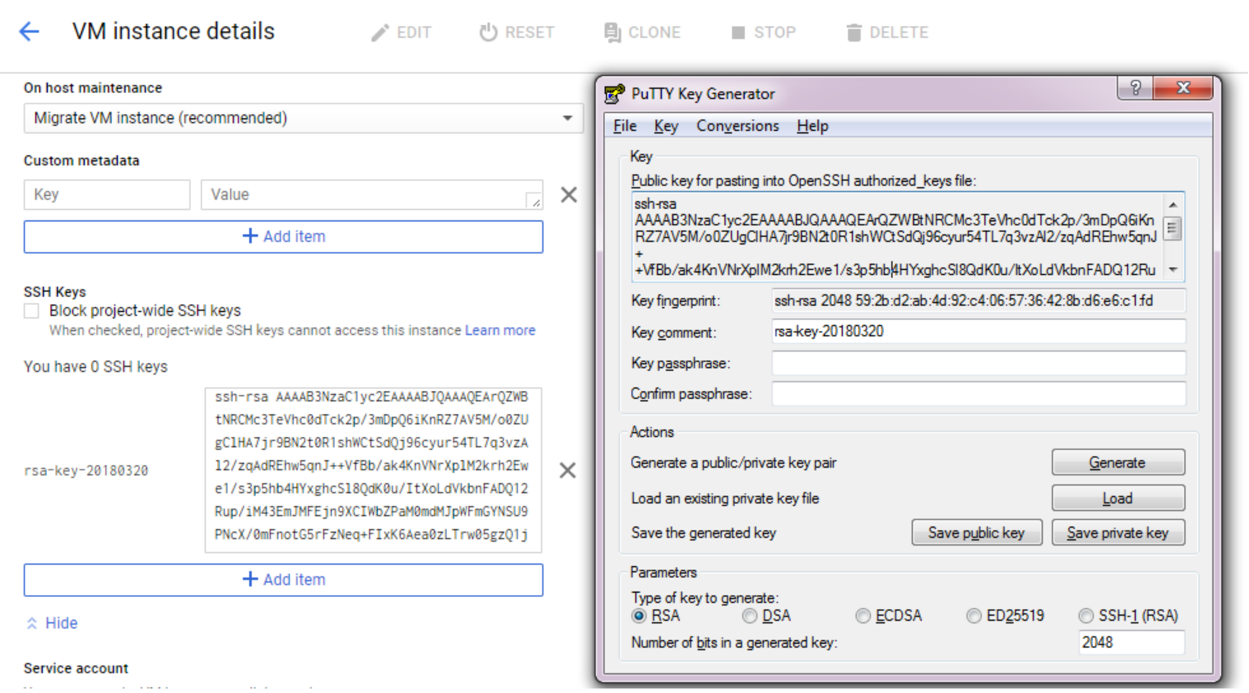
While the instance is starting up we can create the firewall rules. Without these rules we would not be able to SSH into the instance or run the first time wizard.

Below is an example gcloud command to create this firewall rule. You can customize it to fit your project, but make sure the –rules has the same protocols. Make sure that the tags are the same. This rule will only apply when the tags match, so if you forget to add the tags either on the instance or in the firewall rules the Google firewall will block it.

```
gcloud compute firewall-rules create 'allow-to-gtwy' \
--project "gcloud-deployment" \
--direction=INGRESS \
--priority=100 \
--network=gcloud-vpc \
--action=ALLOW \
--source-ranges=0.0.0.0/0 \
--target-tags=checkpoint-gateway,checkpoint-management \
--rules=tcp,icmp,udp,sctp,esp
```

Creating the SSH key and connecting to the instance

To create the SSH key we will use PuTTYgen. It should come installed when you install PuTTY. Open PuTTYgen and click generate to create a public/private key pair. Once you have generated the keypair we can go back and edit the new Check Point instance we just created. (Remember to save the private key file, we will need that to connect later) Click on “Compute Engine” and then “VM instances” in GCP. Then select the new instance we just created and click edit. Scroll down to SSH keys, and paste the public key into the box and click save.



On host maintenance: Migrate VM instance (recommended)

Custom metadata

SSH Keys

□ Block project-wide SSH keys

When checked, project-wide SSH keys cannot access this instance Learn more

You have 0 SSH keys

rsa-key-20180320

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEArQZWB
tNRCMc3TeVhc0dTck2p/3mDpQ6iKnRZ7AV5M/o0ZU
gC1HA7jr9BN2t0R1shWCtSdQj96cyur54TL7q3vzA
12/zqAdREhw5qnJ++VfBb/ak4KnVNrXplM2krh2Ew
e1/s3p5hb4HYxghcS18QdK0u/ItXoLdVkbnFADQ12
Rup/iM43EmJMFEjn9XCIWbZPaM0mdMJpWFmGYNSU9
PNcX/0mFnotG5rFzNeq+FIxK6Aea0zLTrw05gzQ1j
```

Service account

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEArQZWBtNRCMc3TeVhc0dTck2p/3mDpQ6iKn
RZ7AV5M/o0ZUgClHA7jr9BN2t0R1shWCtSdQj96cyur54TL7q3vzAl2/zqAdREhw5qnJ
+
+VfBb/ak4KnVNrXplM2krh2Ewe1/s3p5hb4HYxghcSl8QdK0u/ltXoLdVkbnFADQ12Ru
```

Key fingerprint: ssh-rsa 2048 59:2b:d2:ab:4d:92:c4:06:57:36:42:8b:d6:e6:c1:fd

Key comment: rsa-key-20180320

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

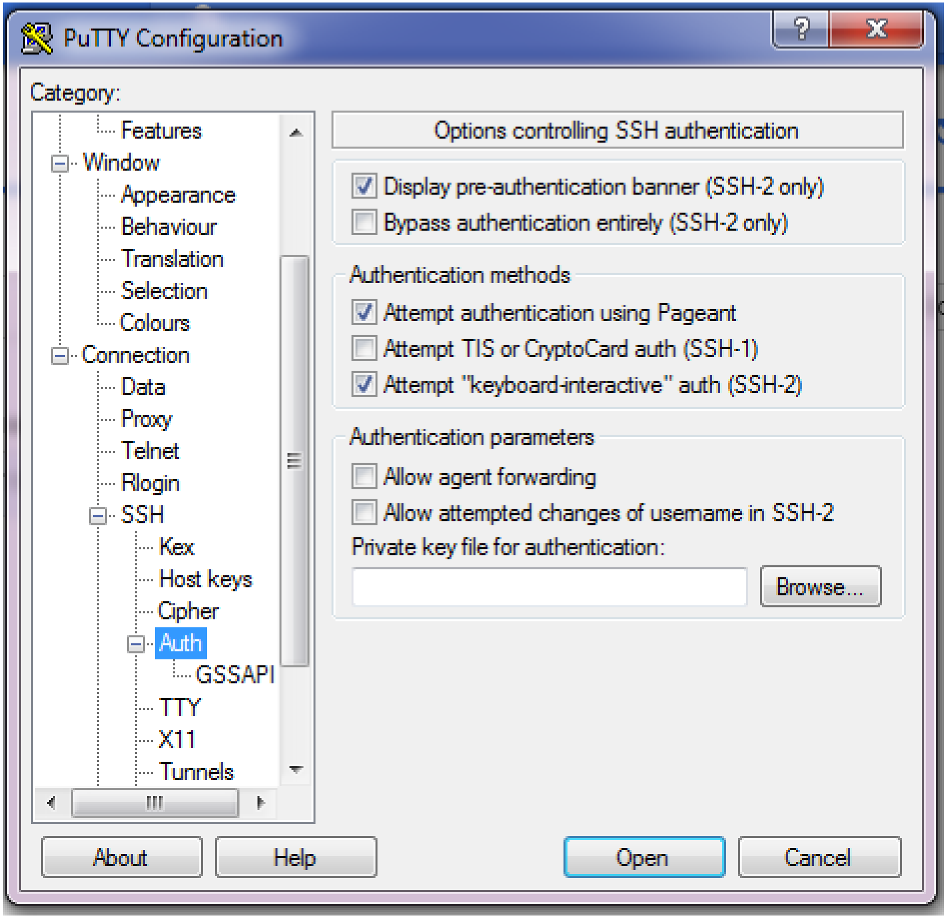
Parameters

Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

Now we can SSH into the instance. Open a new putty session. Under Connection – SSH – Auth we can select the private key file we saved from before.



PuTTY Configuration

Category:

Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)

□ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

□ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

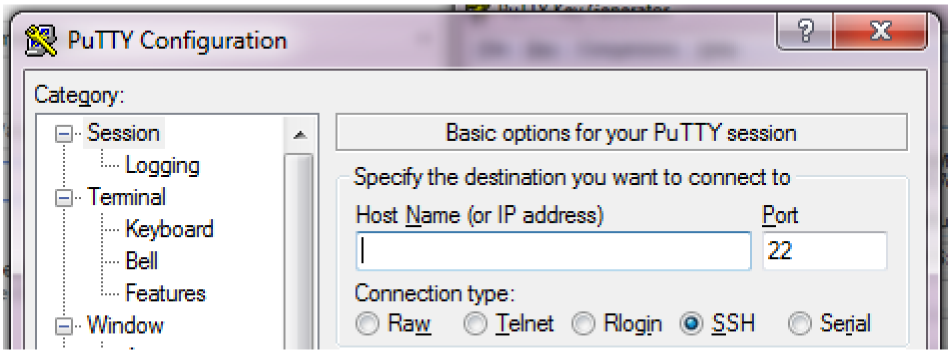
Authentication parameters

□ Allow agent forwarding

□ Allow attempted changes of username in SSH-2

Private key file for authentication:

Then we can go back to “Session” and in “Host Name or IP address” box we can fill in the public IP of the instance.



PuTTY Configuration

Category:

Basic options for your PuTTY session

Specify the destination you want to connect to

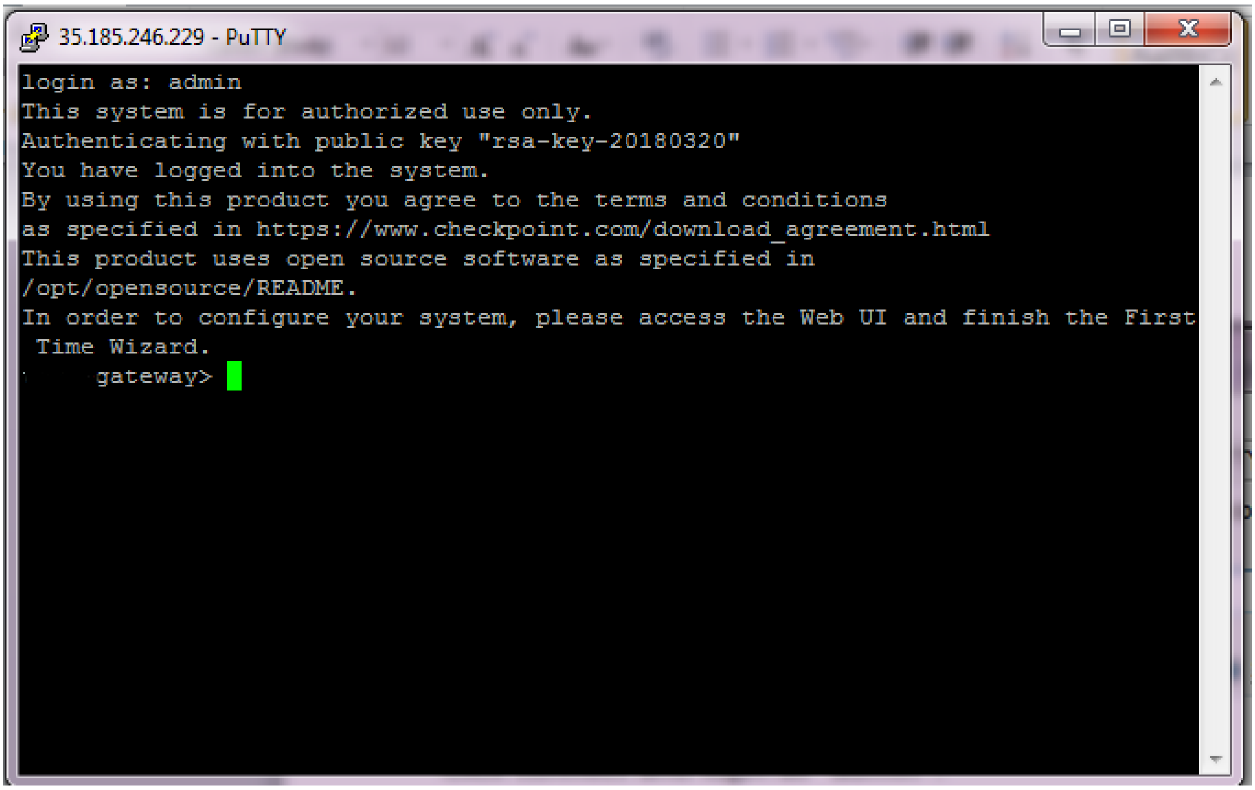
Host Name (or IP address)

Port: 22

Connection type:

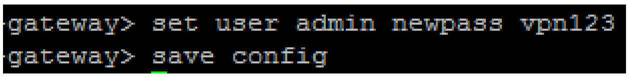
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Click connect and login as “admin”.



```
login as: admin
This system is for authorized use only.
Authenticating with public key "rsa-key-20180320"
You have logged into the system.
By using this product you agree to the terms and conditions
as specified in https://www.checkpoint.com/download_agreement.html
This product uses open source software as specified in
/opt/opensource/README.
In order to configure your system, please access the Web UI and finish the First
 Time Wizard.
     gateway>
```

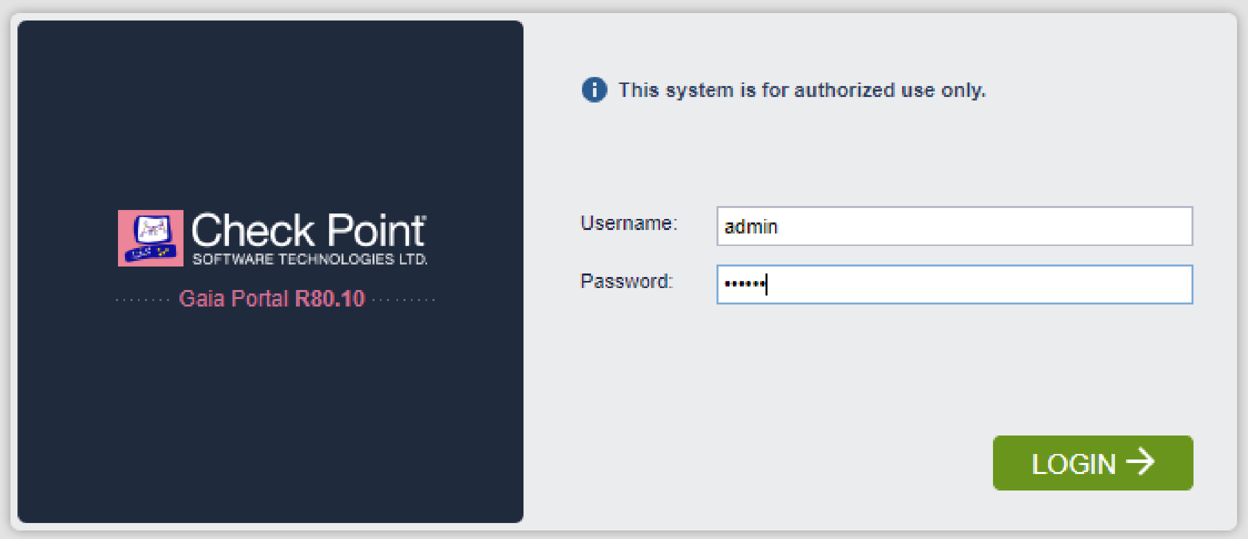
Since we created this using the gcloud command line and not the cloud launcher the first time wizard has not been completed, but first we need to set a new password for admin. Type “set user admin newpass NEW_PASSWORD” and then type save config. In this example the new password was vpn123.



```
gateway> set user admin newpass vpn123
gateway> save config
```

You can close PuTTY.

Now that we have set a password for admin we can connect to Gaia through the web interface to complete the first time wizard. Using your public address connect to the instance with https://. You will get a warning that the page is not secure, continue anyways.

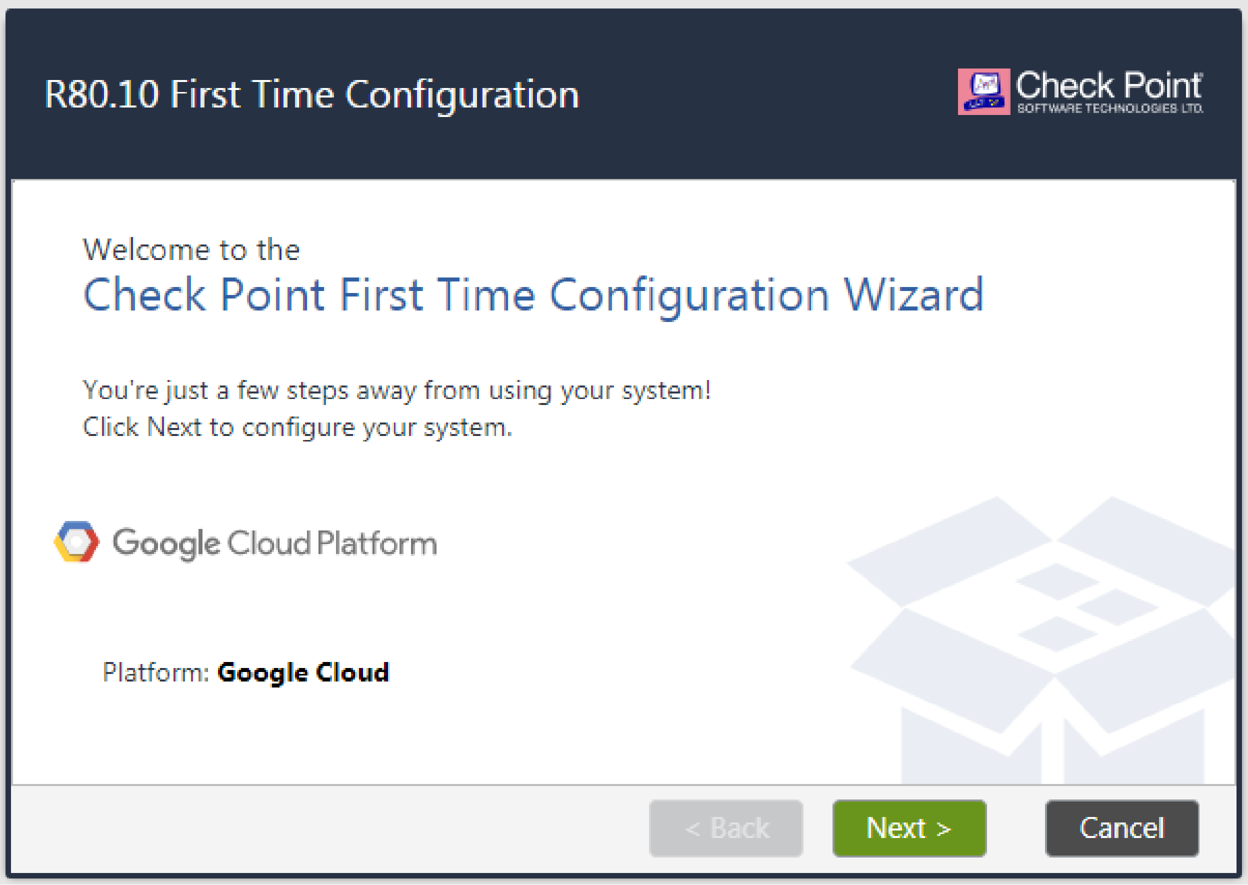


This system is for authorized use only.

Username: admin

Password: ••••••

Enter the credentials we created for admin, and follow the steps to complete the first time wizard.



R80.10 First Time Configuration

Welcome to the
Check Point First Time Configuration Wizard

You're just a few steps away from using your system!
Click Next to configure your system.

Platform: **Google Cloud**

Once the wizard is complete the instance will reboot.