

## Homework 2

### Google Cloud Platform

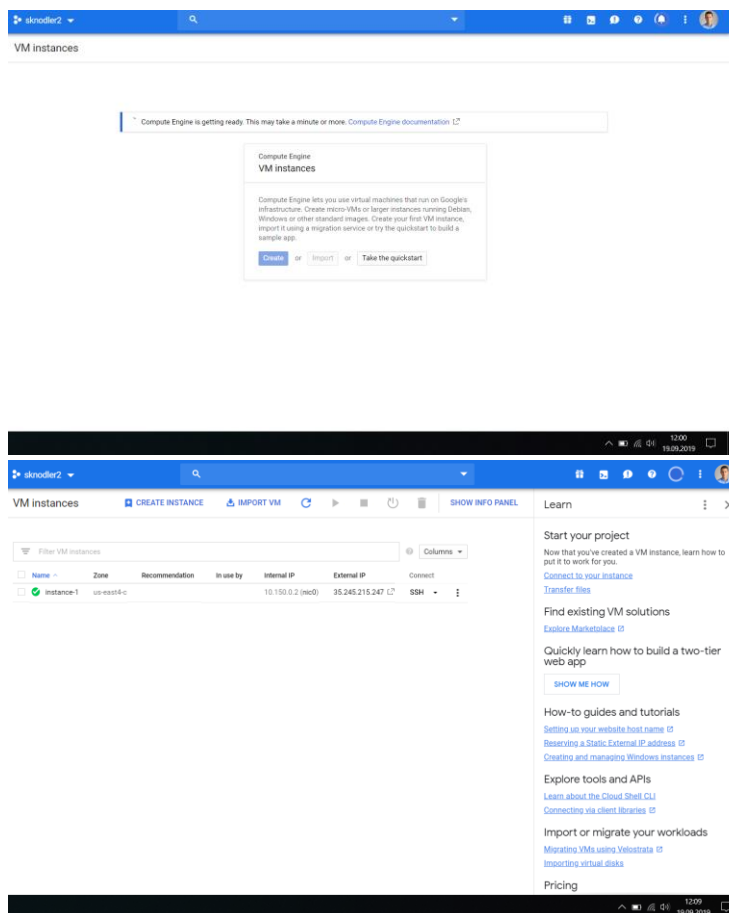
#### Section 4: Creating VM instances

<https://cloud.google.com/compute/docs/instances/create-start-instance>

Complete the following sub-modules in Creating and Starting a VM Instance:

A. Creating an instance from a public image

I went to the instance page and created a new instance. I specified the zone, chose my machine configuration and chose a public linux OS image to create my instance. Next, I allowed access via HTTP and HTTPS and clicked on create.



09/30/2019

## B. Creating an instance from a snapshot

### a. Create a snapshot

I went to my snapshot section to create a persistent disk snapshot. This can reduce the risk of unexpected data loss. I selected my disk in the “source disk” section and chose to store the snapshot in a multi-regional location.

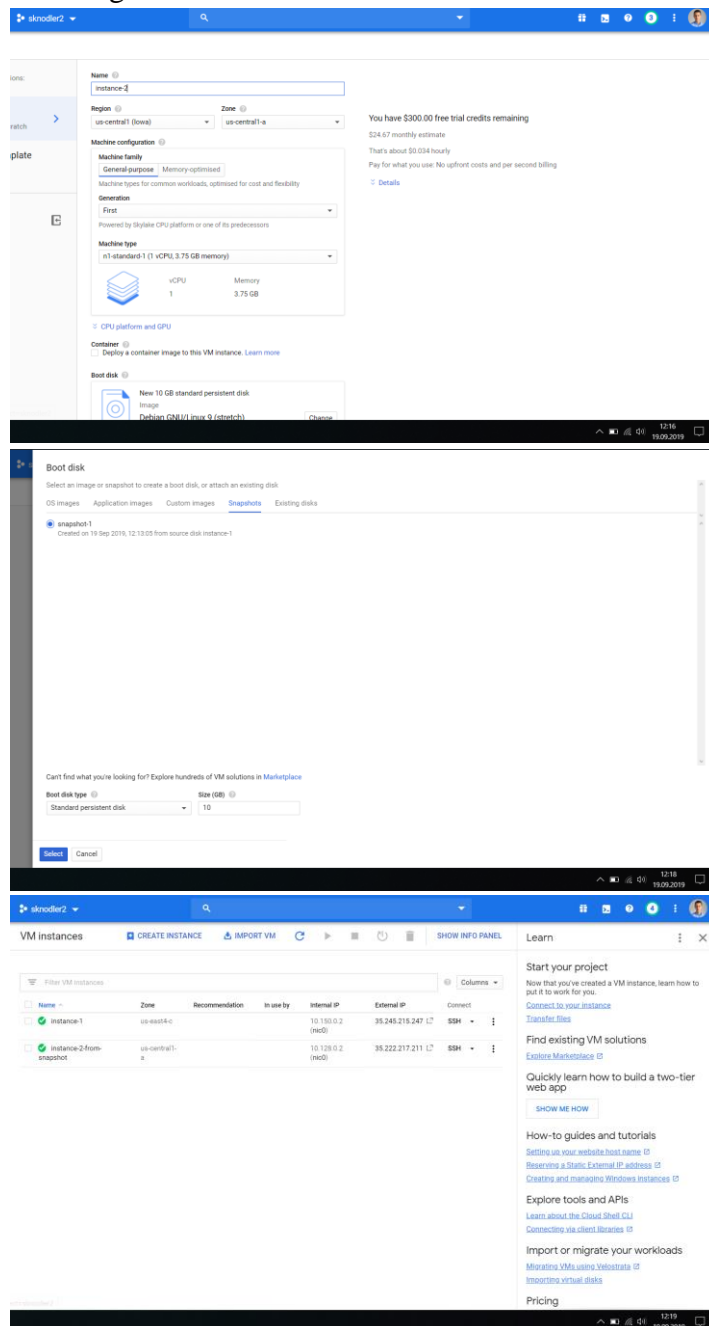
The image shows two screenshots from the Google Cloud Platform console. The top screenshot is the 'Create a snapshot' form. It includes fields for Name (snapshot-1), Description (optional), Source disk (selected), Location (Multi-regional), and Labels (+ Add label). A note mentions that a network transfer fee may apply if the snapshot is stored in a location other than the source disk. The bottom screenshot shows the 'Snapshots' page with a table of existing snapshots.

Name	Location	Snapshot size	Creation time	Creation type	Source disk	Disk size
snapshot-1	us	811.61 MB	19 Sep 2019, 12:13:05	Manual	instance-1	10 GB

The right sidebar of the bottom screenshot lists various resources and guides related to snapshots, including 'How-to guides and tutorials', 'Explore tools and APIs', and 'Resources'.

- b. Create an instance from the snapshot

To do this, I went back to the instance page and selected my snapshot when choosing the boot disk.

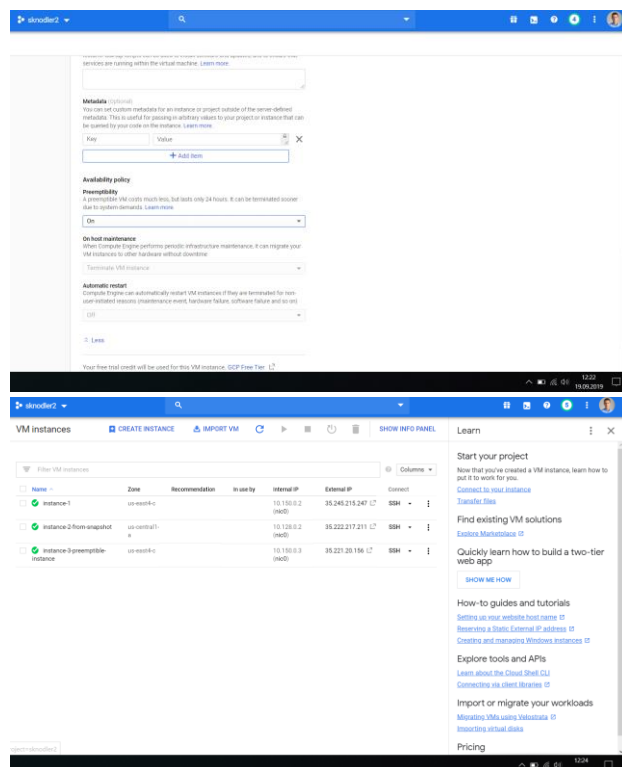


<https://cloud.google.com/compute/docs/instances/create-start-preemptible-instance>

Complete the following sub-module in Creating and Starting a Preemptible VM Instance:

A. Creating a preemptible instance

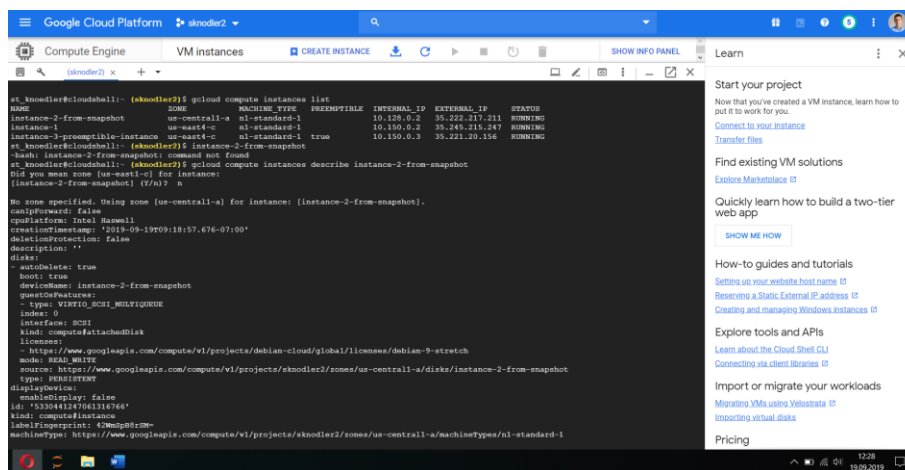
To do this, I created a new instance but chose Preemptibility “On”. This setting disables automatic restart for the instance, and sets the host maintenance action to Terminate.



<https://cloud.google.com/compute/docs/instances/instance-life-cycle>

Complete the following sub-module in Instance Life Cycle:

- A. Checking an instance's status (Check any instance that you created earlier)  
List all instances and their status: *gcloud compute instances list*  
Describe the status of a single instance: *gcloud compute instances describe instance-2-from-snapshot*  
→ I selected “n” as I did not mean the zone “us-east1-c” – the right instance was described in the following...



The screenshot shows the Google Cloud Platform console. The left sidebar displays the 'Compute Engine' section with 'VM instances' selected. The main panel shows a table of VM instances:

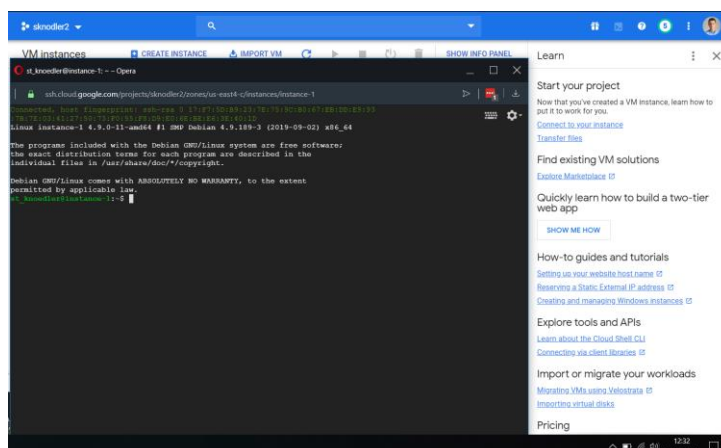
NAME	ZONE	MACHINE TYPE	PREEMPTIBLE	INTERNAL IP	EXTERNAL IP	STATUS
instance-2-from-snapshot	us-central1-a	n1-standard-1		10.128.0.2	35.222.217.211	RUNNING
instance-1	us-east1-a	n1-standard-1		10.130.0.2	35.243.211.241	RUNNING
instance-3-preemptible-instance	us-east1-c	n1-standard-1	true	10.130.0.3	35.221.20.156	RUNNING

Below the table, the details for 'instance-2-from-snapshot' are shown. The 'No zone specified' warning is visible. The instance is running on the 'us-central1-a' zone. The 'gcloud compute instances describe instance-2-from-snapshot' command output is displayed, showing the instance's configuration, including the machine type, disk, and network settings.

<https://cloud.google.com/compute/docs/instances/connecting-to-instance>

Complete the following sub-modules in Connecting to instances:

- A. Connecting to Linux instances  
In the list of virtual machine instances, I clicked on SSH in the row of the instance that I wanted to connect to.

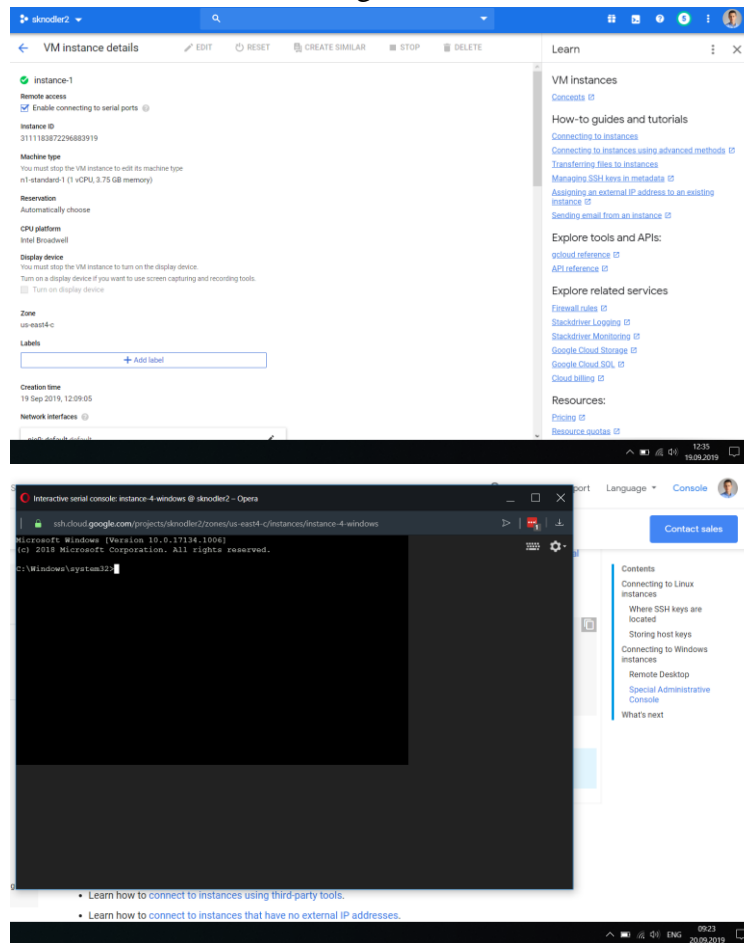


The screenshot shows the Google Cloud Platform console with the 'VM instances' page. The 'instance-2-from-snapshot' row is selected, and the 'SSH' button is clicked. A terminal window opens, showing the SSH connection to the instance. The terminal output displays the Debian GNU/Linux system version and the user 'sknodler' at the prompt. The terminal window is titled 'ssh.cloud.google.com/projects/sknodler/zones/us-east1-c/instances/instance-2-from-snapshot'.

## B. Connecting to Windows instances

### a. Special Administrative Console - Connecting to an instance through the command line

I created a windows instance. Then, I enabled connecting to serial ports in that instance. Under Remote access, I clicked the drop-down list next to Connect to serial console, and select Serial port 2. A Windows Special Administrative Console (SAC) opens. I entered 1. Cmd 2. ch -sn [CHANNEL\_NAME] and got connected:

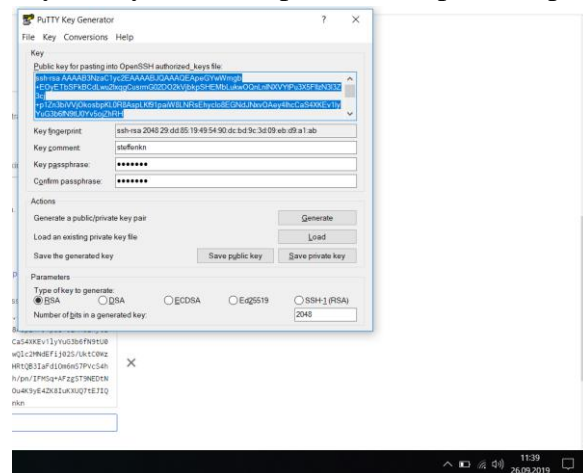


<https://cloud.google.com/compute/docs/instances/connecting-advanced>

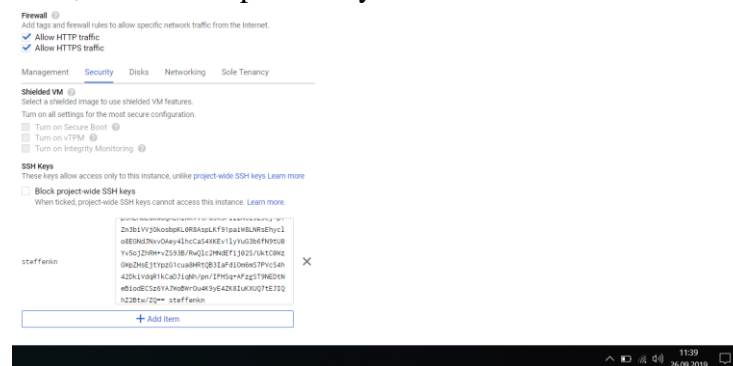
Complete the following sub-module in Connecting to instances using advanced methods:

A. Connecting using third-party tools (SSH)

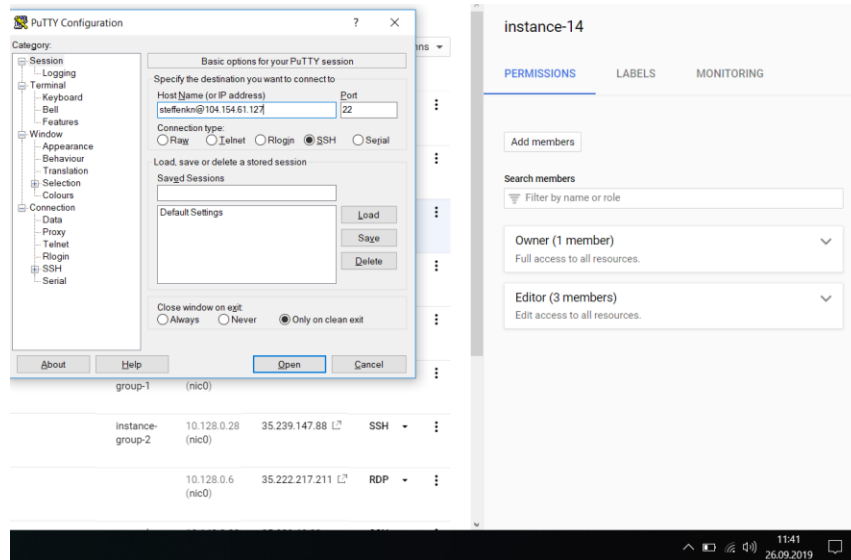
I used Putty key gen to create a key pair (public and private). I saved the private key on my local computer and copied the public key.



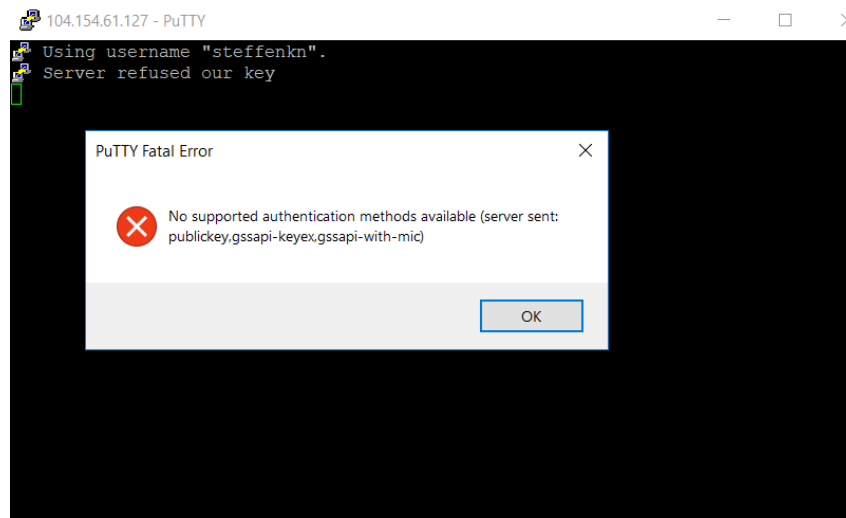
Next, I added the public key to the instance:



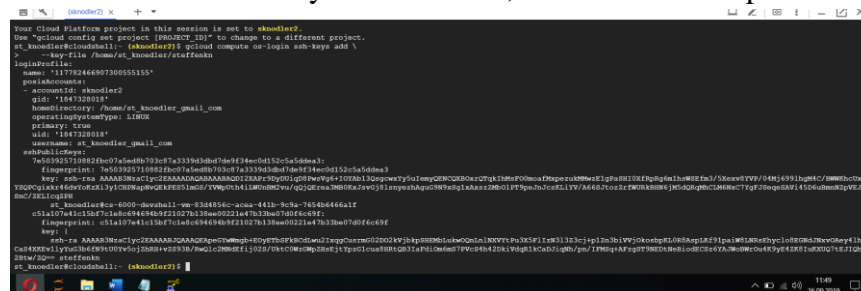
Next, I opened Putty and entered the public IP of the instance. Then, I opened SSH → Auth. And selected my private key that I have stored on my local computer earlier.



When connecting, I got an error message. I have tried this steps 100x with different keys and credentials but I could not connect with my computer.



Also tried to add the key via command, which did not help:





<https://cloud.google.com/compute/docs/disks/create-root-persistent-disks>

Complete the following sub-module in Creating Customized Boot Disks:

A. Creating a standalone boot persistent disk from an image

I went to new disk page and selected under source type an image from which I wanted to create the boot persistent disk.

The screenshot shows the Google Cloud Platform console. The top part displays the 'Create a disk' wizard. The 'Source type' is set to 'Image', and the 'Source image' is 'cd-common-gce-gpu-image-20190906'. The 'Size (GB)' is set to 30. The 'Region' is 'us-central1 (Iowa)' and the 'Zone' is 'us-central1-a'. The 'Snapshot schedule' is set to 'No schedule'. The 'Estimated performance' table shows sustained random IOPS limits of 22.50 (Read) and 45.00 (Write), and sustained throughput limits of 3.60 (Read) and 3.60 (Write). The 'Encryption' section shows 'Google-managed key' selected.

The bottom part shows the 'Disks' list table. The table has columns: Name, Status, Type, Size, Zone(s), In use by, Snapshot, and Actions. The first row is 'disk-1', which is a 'Standard persistent disk' of 30 GB in the 'us-central1-a' zone, not in use by any instance, and has no snapshot. The other rows show disks attached to various instances.

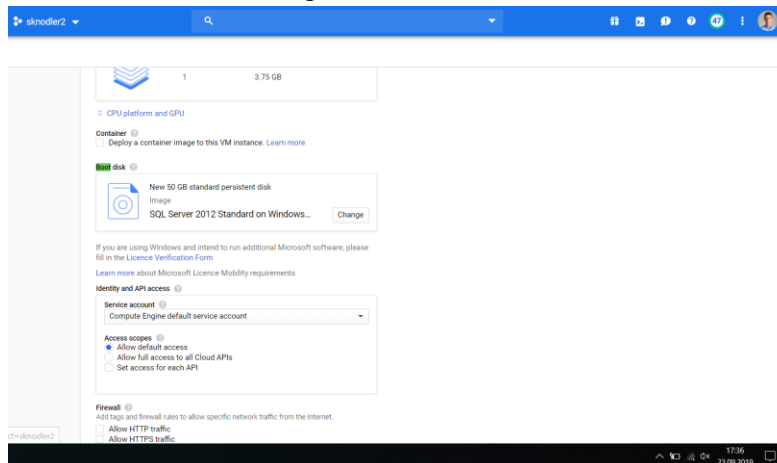
Name	Status	Type	Size	Zone(s)	In use by	Snapshot	Actions
disk-1	✓	Standard persistent disk	30 GB	us-central1-a		None	⋮
instance-1	✓	Standard persistent disk	10 GB	us-east4-c	Instance-1	None	⋮
instance-10	✓	Standard persistent disk	10 GB	us-east4-c	Instance-10	None	⋮
instance-11	✓	Standard persistent disk	10 GB	us-east4-c	Instance-11	None	⋮
instance-2-from-snapshot	✓	Standard persistent disk	10 GB	us-central1-a	Instance...	None	⋮
instance-3-preemptible-instance	✓	Standard persistent disk	10 GB	us-east4-c	Instance...	None	⋮
instance-4-windows	✓	Standard persistent disk	32 GB	us-east4-c	Instance...	None	⋮
instance-7	✓	Standard persistent disk	10 GB	us-east4-c	Instance-7	None	⋮

<https://cloud.google.com/compute/docs/instances/sql-server/creating-sql-server-instances>

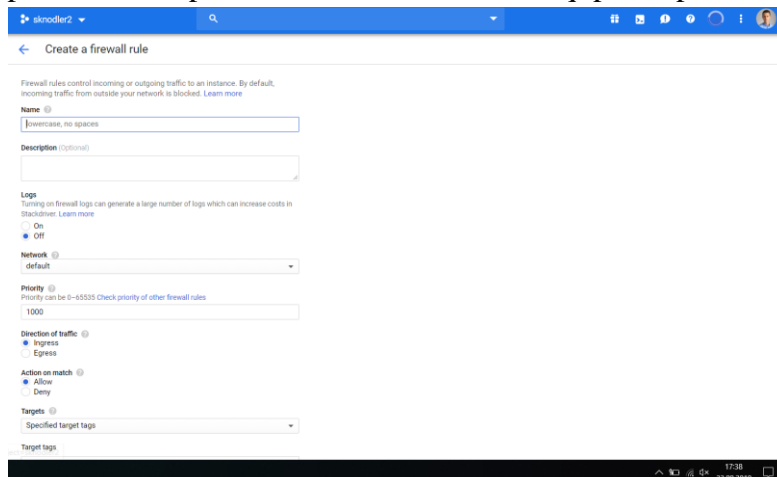
Complete the following sub-module in Creating SQL Server Instances:

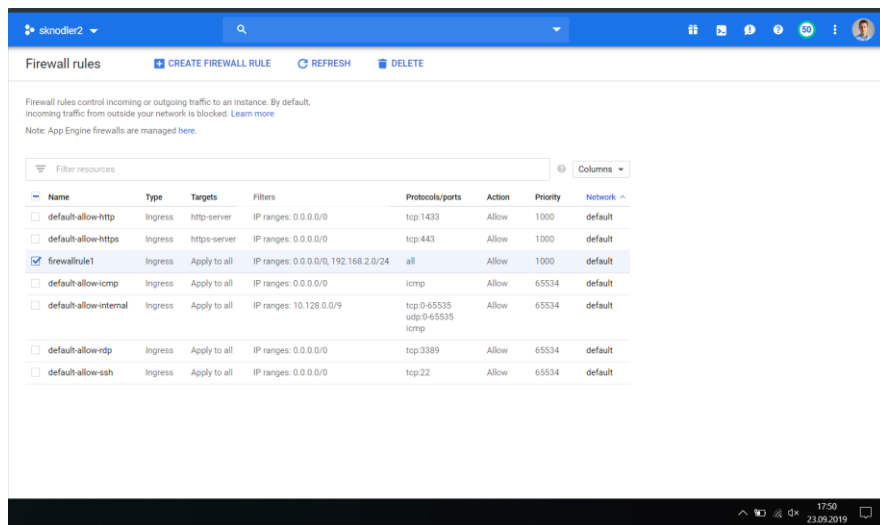
#### A. Creating a SQL Server Instance (Console)

I went to the instance again and, selected under the boot disk a SWL server image.



After I created the instance, I created a firewall rule to allow access to SQL Server on my new instance. Therefore, I used the default SQL Server port is 1433. I selected the VPC network where my SQL Server instance is located. I chose ingress traffic and I allowed all IPs to access with “Allow from any source”. In the protocols and port sections, I allowed the Sql port tcp:1433;





Name	Type	Targets	Filters	Protocols/ports	Action	Priority	Network
<input type="checkbox"/> default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:1433	Allow	1000	default
<input type="checkbox"/> default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	default
<input checked="" type="checkbox"/> firewallrule1	Ingress	Apply to all	IP ranges: 0.0.0.0/0, 192.168.2.0/24	all	Allow	1000	default
<input type="checkbox"/> default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default
<input type="checkbox"/> default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default
<input type="checkbox"/> default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default
<input type="checkbox"/> default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default

## Section 5: Managing your instances

[https://cloud.google.com/compute/docs/instances/stop-start-instance#stopping\\_an\\_instance](https://cloud.google.com/compute/docs/instances/stop-start-instance#stopping_an_instance)

Complete the following sub-module in Stopping and Starting an Instance:

### A. Stopping an Instance (GCloud)

Please see the code below that I entered:

```
st_knoedler@cloudshell:~ (sknodler2)$ gcloud compute instances stop instance-11
Did you mean zone [us-east1-b] for instance: [instance-11] (Y/n)? n

No zone specified. Using zone [us-east4-c] for instance: [instance-11].
Stopping instance(s) instance-11...done.
Updated [https://compute.googleapis.com/compute/v1/projects/sknodler2/zones/us-east4-c/instances/instance-11].
https://console.cloud.google.com/compute/soleTenancy/project=sknodler2
```

<https://cloud.google.com/compute/docs/instances/deleting-instance>

Complete the following sub-module in Deleting an Instance:

### A. Deleting an Instance (GCloud)

Please see the code below that I entered:

```
st_knoedler@cloudshell:~ (sknodler2)$ gcloud compute instances delete instance-11
Did you mean zone [us-east1-b] for instance: [instance-11] (Y/n)? n

No zone specified. Using zone [us-east4-c] for instance: [instance-11].
The following instances will be deleted. Any attached disks configured
to be auto-deleted will be deleted unless they are attached to any
other instances or the '--keep-disks' flag is given and specifies them
for keeping. Deleting a disk is irreversible and any data on the disk
will be lost.
- [instance-11] in [us-east4-c]

Do you want to continue (Y/n)? Y

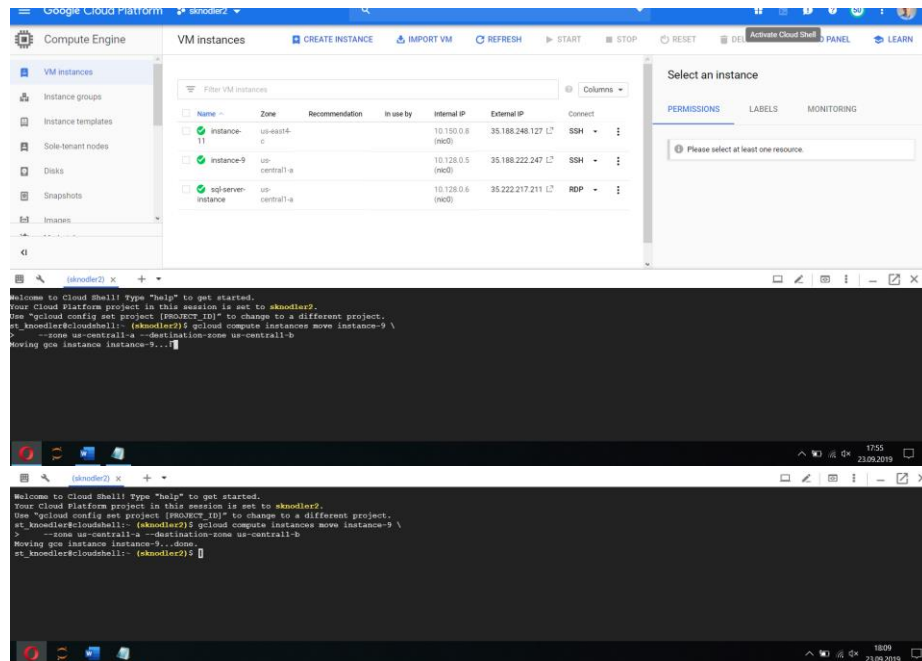
Deleted [https://www.googleapis.com/compute/v1/projects/sknodler2/zones/us-east4-c/instances/instance-11].
st_knoedler@cloudshell:~ (sknodler2)$
```

<https://cloud.google.com/compute/docs/instances/moving-instance-across-zones>

Complete the following sub-module in Moving an instance between zones:

A. Moving an instance automatically (GCloud)

Please see the code below that I entered in the command box:

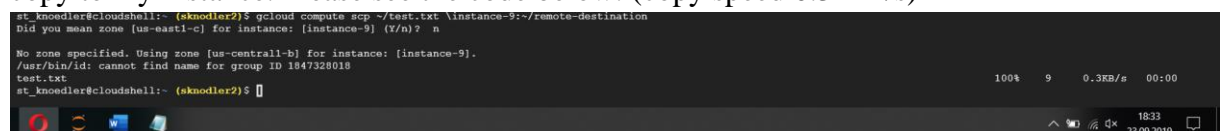


<https://cloud.google.com/compute/docs/instances/managing-instances>

Complete the following sub-module in Performing Other Tasks With Your Instances:

A. Copy files between an instance and local computer

I uploaded the file “test.txt” from my local computer and chose it in the directory to copy to my instance. Please see the code below: (copy speed 0.3 KB/s)



<https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances>

Complete the following sub-modules in Creating and enabling service accounts for instances:

A. Creating a new service account

I Opened the Service Accounts page in the GCP Console. I chose my project, a name for my account name, my service account id and granted the service account admin role.

The screenshot shows the 'Create service account' page in the GCP Console. The page has a blue header with the project name 'sknodler2' and a search bar. Below the header, there's a progress bar with three steps: 1. Service account details (active), 2. Grant this service account access to the project (optional), and 3. Grant users access to this service account (optional). The 'Service account details' section includes a text input for 'Service account name', a text input for 'Display name for this service account', a text input for 'Service account ID' (with a value '@sknodler2.iam.gserviceaccount.com' and a copy icon), and a text input for 'Service account description'. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

The screenshot shows the 'Create service account' page in the GCP Console, specifically the 'Service account permissions (optional)' step. The progress bar shows step 2 is active. The section title is 'Service account permissions (optional)'. Below it, there's a text input for 'Role' with a dropdown menu showing 'Service Account Admin'. There's also a '+ ADD ANOTHER ROLE' link. At the bottom, there are 'CONTINUE' and 'CANCEL' buttons.

Create service account

Service account details — Grant this service account access to the project (optional) — Grant users access to this service account (optional)

Grant users access to this service account (optional)

Grant access to users or groups that need to perform actions as this service account. [Learn more](#)

Service account users role  
st.knoedler@gmail.com

Grant users the permissions to deploy jobs and VMs with this service account

Service account admin role  
st.knoedler@gmail.com

Grant users the permission to administer this service account

Create key (optional)

Download a file that contains the private key. Store the file securely because this key can't be recovered if lost. However, if you are unsure why you need a key, skip this step for now.

+ CREATE KEY

DONE CANCEL

Permissions

Show inherited permissions

Filter tree

Role/Member ↑ Inheritance

- Compute Engine Service Agent (1)
- Editor (3)
- Owner (1)
- Security Admin (1)
- Service Account Admin (1)

Service accounts

+ CREATE SERVICE ACCOUNT DELETE

SHOW INFO PANEL

Service accounts for project 'sknodler2'

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more](#)

Filter table	Email	Status	Name ↑	Description	Key ID	Key creation date	Actions
<input type="checkbox"/>	sknodler2@appspot.gserviceaccount.com	✓	App Engine default service account		No keys		⋮
<input type="checkbox"/>	747099227955-compute@developer.gserviceaccount.com	✓	Compute Engine default service account		No keys		⋮
<input type="checkbox"/>	sknodler@sknodler2-iam.gserviceaccount.com	✓	sknodler	my account	No keys		⋮

I copied the service account email and granted IAM roles to the new service account “sknodler@sknodler2.iam.gserviceaccount.com”:

## B. Setting up a new instance to run as a service account

I created a new instance and chose under Identity and API access “sknodler” (the new service account)

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

Boot disk

New 10 GB standard persistent disk

Image

Debian GNU/Linux 9 (stretch) Change

Identity and API access

Service account

sknodler

Access scopes

Use IAM roles with service accounts to control VM access. [Learn more](#)

Firewall

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

- Allow HTTP traffic
- Allow HTTPS traffic

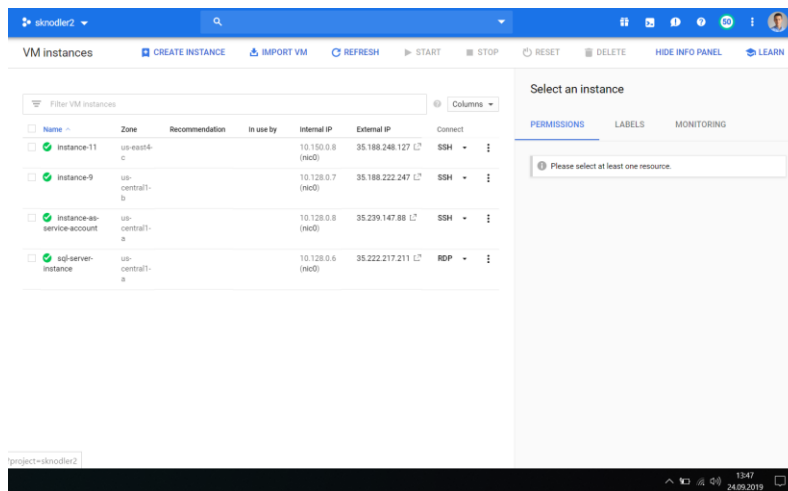
Management, security, disks, networking, sole tenancy

Your free trial credit will be used for this VM instance. [GCP Free Tier](#)

Create Cancel

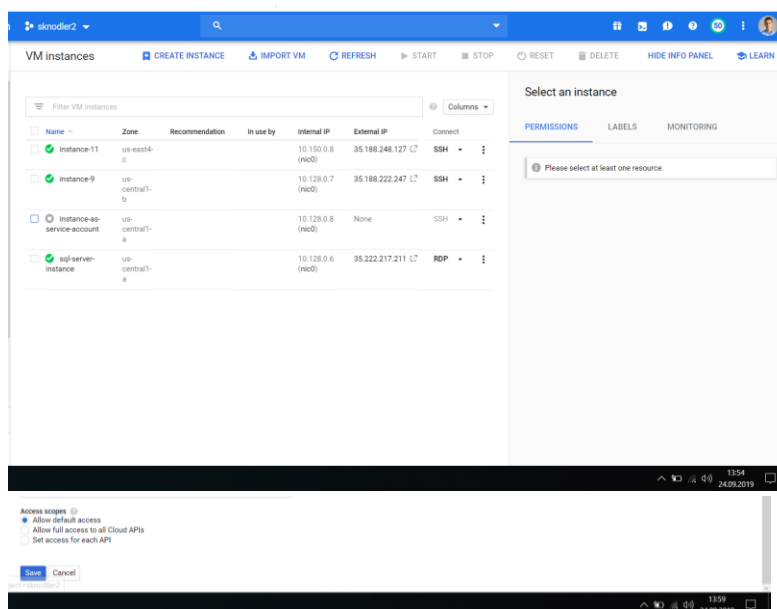
Equivalent REST or command line

09/30/2019



I also did the GCloud but forgot to take a screen shot, I ran:  
*gcloud compute instances create instance-12 \*  
*--service-account sknodler@sknodler2.iam.gserviceaccount.com*

- C. Changing the service account and access scopes for an instance  
I had to stop my instance first. Next, I clicked on the instance and chose “edit”. Then, I changed the access scope to “Allow full access to all Cloud APIs”



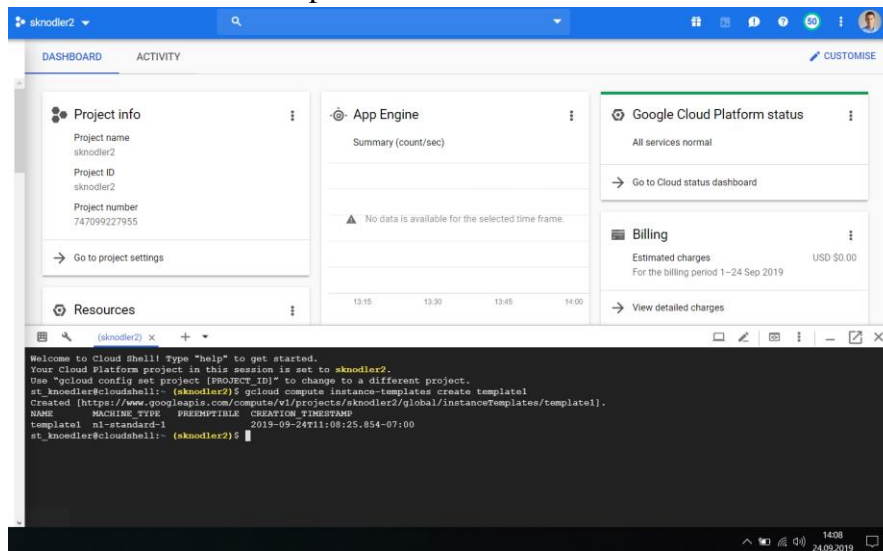
## Section 6: Creating and Managing Instance Templates

<https://cloud.google.com/compute/docs/instance-templates/create-instance-templates>

Complete the following sub-modules in Creating Instance Templates:

### A. Creating a new instance template (GCloud)

Please see the code/script below in the screenshot that I entered:



I did create a default template. When looking in the documentation, this is the following parameter that are chosen. However, The second screenshot shows how I could also specify the parameters:

If you do not provide explicit template settings, `gcloud compute` creates a template with the following default values:

- Machine type: n1-standard-1
- Image: The latest Debian image
- Boot disk: A new standard boot disk named after the instance
- Network: The default VPC network
- IP address: An ephemeral external IP address

You can also explicitly provide these configuration settings. For example:

```
gcloud compute instance-templates create example-template-custom \  
  --machine-type n1-standard-4 \  
  --image-family debian-9 \  
  --image-project debian-cloud \  
  --boot-disk-size 250GB
```



B. Creating an instance template that specifies a subnet

First, I listed my subnets from which I could choose. Next, I created a template with a subnet that was earlier listed. Please look at my code in the screenshot below:

```
st_knodler@cloudshell:~ (sknodler2)$ gcloud compute networks subnets list
NAME          REGION    NETWORK    RANGE
default-us-west2    default  10.168.0.0/20
default-asia-northeast1  default  10.146.0.0/20
default-asia-northeast2  default  10.174.0.0/20
default-us-west1    default  10.138.0.0/20
default-southamerica-east1  default  10.158.0.0/20
default-europe-west6    default  10.172.0.0/20
default-europe-west4    default  10.164.0.0/20
default-asia-east1    default  10.140.0.0/20
default-europe-north1    default  10.166.0.0/20
default-asia-southeast1  default  10.148.0.0/20
default-us-east4    default  10.150.0.0/20
default-europe-west1    default  10.132.0.0/20
default-europe-west2    default  10.154.0.0/20
default-europe-west3    default  10.156.0.0/20
default-australia-southeast1  default  10.152.0.0/20
default-asia-south1    default  10.160.0.0/20
default-us-east1    default  10.142.0.0/20
default-us-central1    default  10.128.0.0/20
default-asia-east2    default  10.170.0.0/20
default-northamerica-northeast1  default  10.162.0.0/20
st_knodler@cloudshell:~ (sknodler2)$ gcloud compute instance-templates create template-ga \
> --region us-central1 \
> --subnet default
Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/instanceTemplates/template-ga].
NAME          MACHINE_TYPE  PREEMPTIBLE  CREATION_TIMESTAMP
template-ga  nl-standard-1  2019-09-24T11:17:03.042-07:00
st_knodler@cloudshell:~ (sknodler2)$
```

## Section 7: Creating and Managing Groups of Instances

<https://cloud.google.com/compute/docs/instance-groups/creating-groups-of-managed-instances>

Complete the following sub-modules in Creating managed instance groups (MIG):

A. Creating a managed instance group (Console and GCloud)

Please look at the code below. I chose my earlier created template to create the instance group “example-group”

```
st_knodler@cloudshell:~ (sknodler2)$ gcloud compute instance-groups managed create example-group --base-instance-name test --size 3 --template template-ga
Did you mean zone [us-east1-c] for managed instance group:
[example-group] (Y/n)? Y
Created [https://www.googleapis.com/compute/v1/projects/sknodler2/zones/us-east1-c/instanceGroupManagers/example-group].
NAME          LOCATION    SCOPE  BASE_INSTANCE_NAME  SIZE  TARGET_SIZE  INSTANCE_TEMPLATE  AUTOSCALED
example-group  us-east1-c  zone  test               0     3            template1          no
st_knodler@cloudshell:~ (sknodler2)$
```

09/30/2019

I did the same in the console. I went to the instance group page and selected my earlier specified template-1 to create my managed instance group

The screenshot shows the Google Cloud Platform console interface. The top part displays the 'Create an instance group' wizard. The 'New managed instance group' option is selected. The wizard fields include: Name (example-group-1), Description (optional), Location (us-central1 (Iowa)), Region (us-central1 (Iowa)), Zone (us-central1-a), Instance template (template1), Auto-scaling (On), Auto-scaling policy (CPU utilisation), Target CPU utilisation (60%), Minimum number of instances (1), Maximum number of instances (10), Cool-down period (60 seconds), and Autohealing (No health check).

The bottom part of the screenshot shows the 'Instance groups' page. It features a table with the following data:

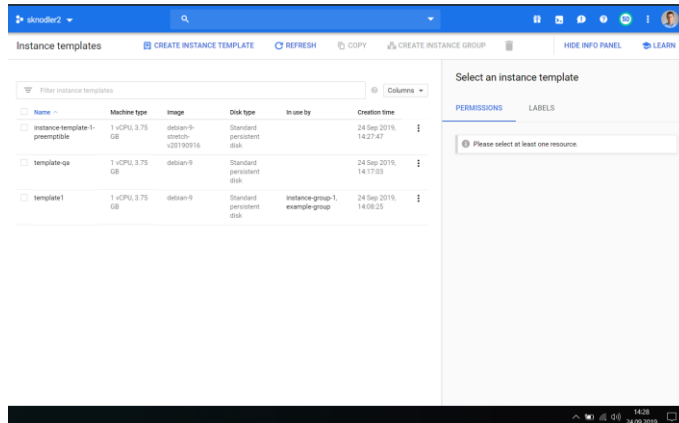
Name	Zone	Instances	Template	Creation time	Recommendation	Auto-scaling	In use by
example-group	us-east1-c	3	template1	24 Sep 2019, 14:19:57		Off	
instance-group-1	us-central1-a	1	template1	24 Sep 2019, 14:22:26		Target CPU utilisation 60%	

Please see above the two groups that I created through console and gcloud.

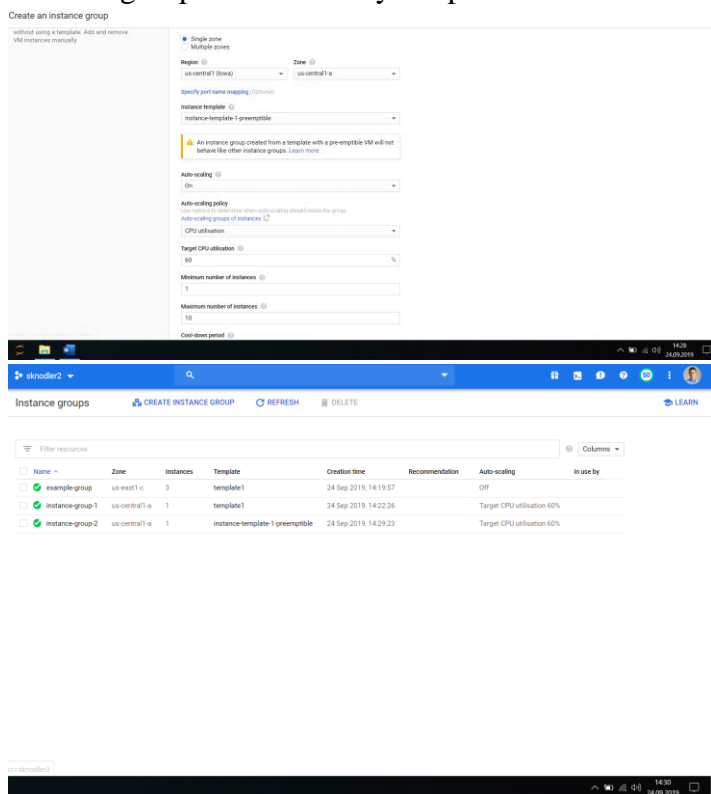
09/30/2019

## B. Creating groups of preemptible instances

I went to the instance template page, and clicked on “create instance template”. IN the settings, I chose Preemptibility to “On”



Then, I used the template to create a managed instance group. To do this, I went into instance groups and chose my template for the instance.



<https://cloud.google.com/compute/docs/instance-groups/autohealing-instances-in-migs>

Complete the following sub-module in Setting up health checking and autohealing for instances in

MIGs:

A. Setting up a health check and an autohealing policy (Console and GCloud)

Create a health check

In gcloud, I created a health check for autohealing that is more conservative than a load balancing health check. Therefore, this health check looks for a response on port 80 and tolerates some failures before it marks the instance as unhealthy. So if it does not get a response after 3 tries, the instance will be marked as unhealthy.

```
st_knodler@cloudshell:~ (sknodler2)$ gcloud compute health-checks create http example-check --port 80 \
> --check-interval 30s \
> --healthy-threshold 1 \
> --timeout 10s \
> --unhealthy-threshold 3
Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/healthChecks/example-check].
NAME      PROTOCOL
example-check HTTP
st_knodler@cloudshell:~ (sknodler2)$
```

stackdriver/19007

20:18 24.09.2019

Next, I create a firewall rule to allow health check to connect. As the health checks come from 130.211.0.0/22 and 35.191.0.0/16, I allowed those ip addresses. I using the default network.

```
st_knodler@cloudshell:~ (sknodler2)$ gcloud compute firewall-rules create allow-health-check \
> --allow tcp:80 \
> --source-ranges 130.211.0.0/22,35.191.0.0/16 \
> --network default
Creating firewall...Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/firewalls/allow-health-check].
Creating firewall...done.
NAME      NETWORK  DIRECTION  PRIORITY  ALLOW  DENY  DISABLED
allow-health-check default  INGRESS    1000      tcp:80  False
st_knodler@cloudshell:~ (sknodler2)$
```

nodler2

20:19 24.09.2019

Last, I applied the health check by configuring an autohealing policy for my regional or zonal managed instance group “instance-group-1”.

```
st_knodler@cloudshell:~ (sknodler2)$ gcloud compute instance-groups managed update instance-group-1 --health-check example-check --initial-delay 300 --zone us-central1-a
Updated [https://www.googleapis.com/compute/v1/projects/sknodler2/zones/us-central1-a/instanceGroupManagers/instance-group-1].
st_knodler@cloudshell:~ (sknodler2)$
```

st=sknodler2

20:25 24.09.2019

I did the same in the console. First, creating the health check and second, creating the new firewall rule that allows the access to the health check.

The screenshot shows the Google Cloud Platform console interface. The top section is titled 'Create a health check'. It contains a form with the following fields:

- Name:** example-check-2
- Description:** (optional)
- Protocol:** HTTP
- Port:** 80
- Proxy protocol:** NONE
- Request path:** /
- Health criteria:**
  - Check interval:** 5 seconds
  - Timeout:** 5 seconds
  - Healthy threshold:** 5 consecutive successes
  - Unhealthy threshold:** 3 consecutive failures

Below the form, there is a table titled 'Health checks' with the following data:

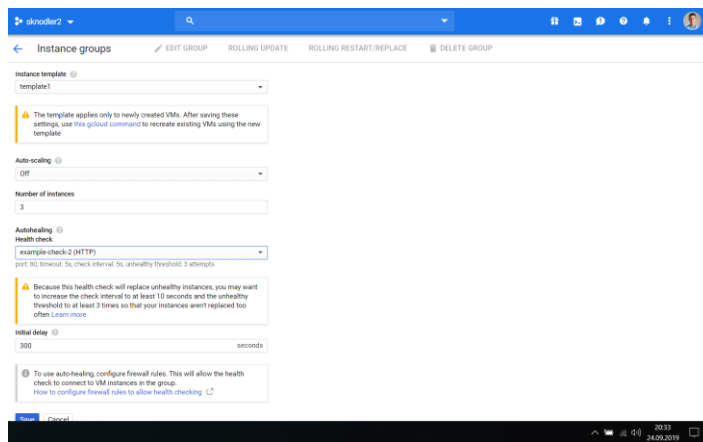
Name	Host	Path	Protocol	Port	Created by
example-check	/	/	HTTP	80	instance-group-1
example-check-2	/	/	HTTP	80	

The screenshot shows the Google Cloud Platform console interface for creating a firewall rule. The form is titled 'Create a firewall rule'. It contains the following fields:

- Action:** Deny
- Targets:** Specified target tags
- Target tags:** (empty)
- Source filter:** IP ranges
- Source IP ranges:** 192.211.0.0/22, 35.191.0.0/16
- Second source filter:** None
- Protocols and ports:** Allow all
- Specified protocols and ports:** tcp, 80
- Other protocols:** (empty)

At the bottom, there are buttons for 'Create', 'Cancel', and 'Equivalent REST or command line'.

Next, I edited my managed instance group to apply my health check. Therefore, I chose the health check under auto-healing and added a delay of 300 seconds so that the health check tries to contact the instance with some time in between. Otherwise, I could create a new instance even if not really necessary.



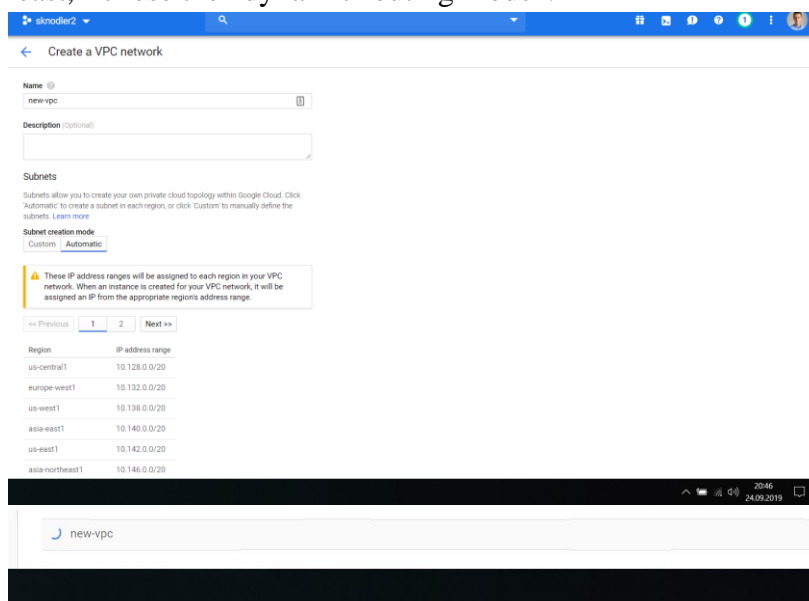
## Section 8: Virtual Private Cloud (VPC)

<https://cloud.google.com/vpc/docs/using-vpc>

Complete the following sub-modules in Using VPC networks:

### A. Creating an auto mode network (Console)

Auto mode networks create one subnet in each GCP region automatically when I create the network. Therefore, when a new regions become available, new subnets in those regions are automatically added to the auto mode network. I opened the vpc network page and created a new vpc network. I chose automatic for the subnet creation mode and in the firewall settings I chose my predefined firewall. Last but not least, I chose the “dynamic routing mode”.



## B. Creating a custom mode network (GCloud)

Please see the code below. I created a new custom mode network and chose dynamic routing = global.

```
st_knoedler@cloudshell:~ (sknodler2) $ gcloud compute networks create new-vpc-1 --subnet-mode=custom --bgp-routing-mode=global
Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/networks/new-vpc-1].
NAME      SUBNET_MODE  BGP_ROUTING_MODE  IPV4_RANGE  GATEWAY_IPV4
new-vpc-1  CUSTOM      GLOBAL
```

## C. Create subnets

I went to the vpc network page and clicked on my new vpc network. Then, I chose “add subnet”, chose a name “subnet1”, the iprange and region to for my subnet.

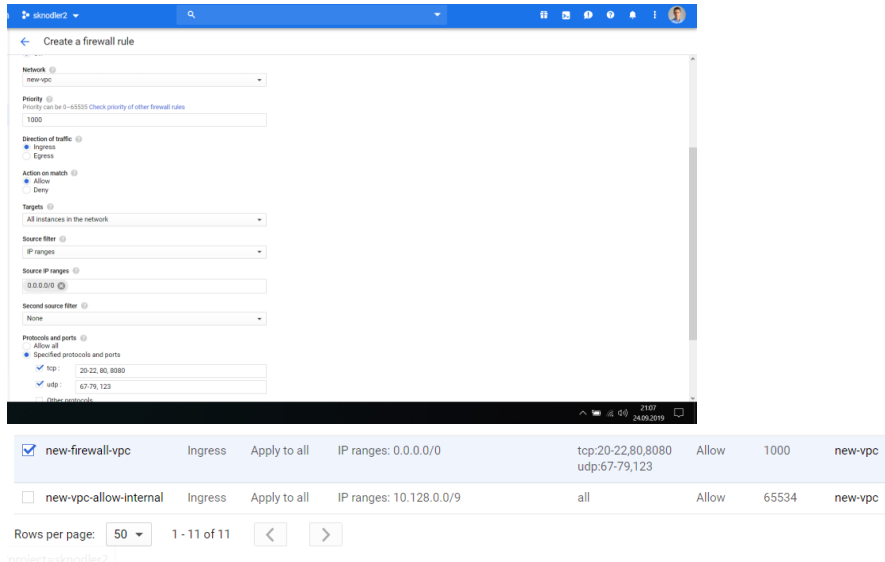
The first screenshot shows the 'VPC network details' page for a network named 'new-vpc'. The 'Subnets' tab is selected, displaying a table of existing subnets. The 'Add subnet' button is visible. The second screenshot shows the 'Add subnet' dialog box with 'subnet1' selected in the 'Name' field, 'us-central1' in the 'Region' field, and '10.0.0.0/9' in the 'IP address range' field. The 'Permissions' panel on the right shows the 'Owner' role assigned to the user.

Name	Region	IP address ranges	Gateway	Private Google access	Flow logs
new-vpc	us-central1	10.128.0.0/20	10.128.0.1	Off	Off
new-vpc	eu-west-1	10.132.0.0/20	10.132.0.1	Off	Off
new-vpc	us-west-1	10.138.0.0/20	10.138.0.1	Off	Off
new-vpc	asia-east-1	10.140.0.0/20	10.140.0.1	Off	Off
new-vpc	us-east-1	10.142.0.0/20	10.142.0.1	Off	Off
new-vpc	asia-northeast-1	10.146.0.0/20	10.146.0.1	Off	Off
new-vpc	asia-southeast-1	10.148.0.0/20	10.148.0.1	Off	Off

Name	Region	IP address ranges	Gateway	Private Google access	Flow logs
new-vpc	eu-west-3	10.156.0.0/20	10.156.0.1	Off	Off
new-vpc	southamerica-east-1	10.158.0.0/20	10.158.0.1	Off	Off
new-vpc	asia-south-1	10.160.0.0/20	10.160.0.1	Off	Off
new-vpc	northamerica-northeast-1	10.162.0.0/20	10.162.0.1	Off	Off
new-vpc	eu-west-4	10.164.0.0/20	10.164.0.1	Off	Off
new-vpc	eu-north-1	10.166.0.0/20	10.166.0.1	Off	Off
new-vpc	us-west-2	10.168.0.0/20	10.168.0.1	Off	Off
new-vpc	asia-east-2	10.170.0.0/20	10.170.0.1	Off	Off
new-vpc	eu-west-6	10.172.0.0/20	10.172.0.1	Off	Off
new-vpc	asia-northeast-2	10.174.0.0/20	10.174.0.1	Off	Off
subnet1	us-central1	10.0.0.0/9	10.0.0.1	Off	Off

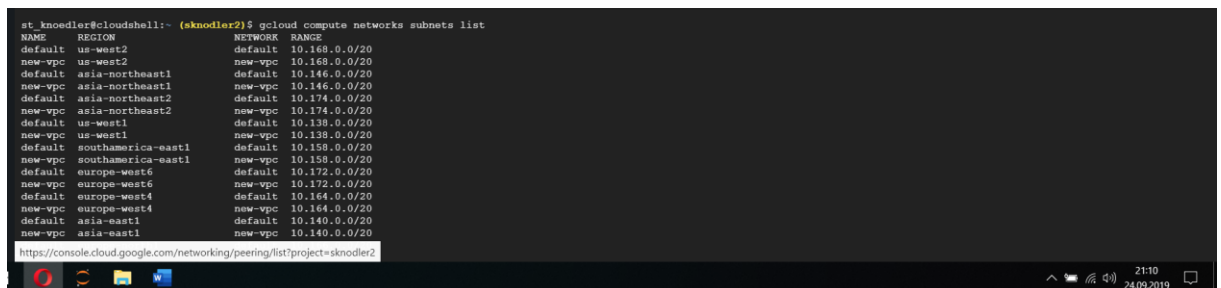
#### D. Create firewall rules

I went to my firewalls page and created a new firewall that allows ingress tcp ports: 20-22, 80, 8080 as well as udp ports 47-79, 123. I allowed all source ips with 0.0.0.0/0.



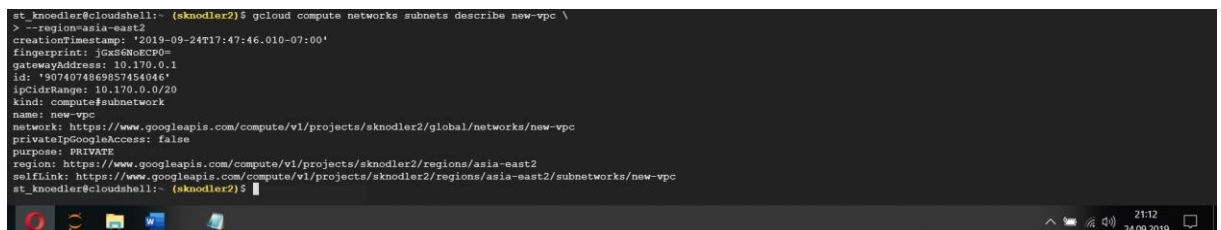
#### E. Listing subnets (GCloud)

Please see code below



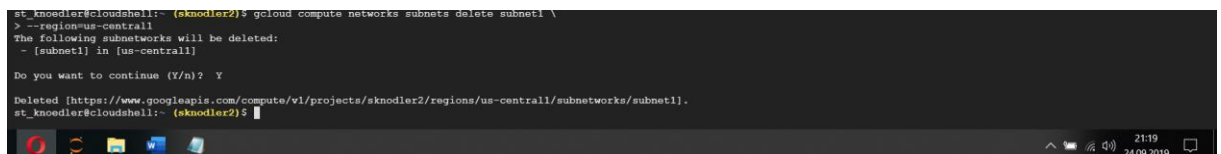
#### F. Describing a subnet (GCloud)

Please see code below



#### G. Deleting subnets (GCloud)

Please see code below:





## H. Deleting a network (GCloud)

Please see code below

```
st_knoedler@cloudshell:~ (sknodler2)$ gcloud compute networks delete new-vpc-1
The following networks will be deleted:
- [new-vpc-1]

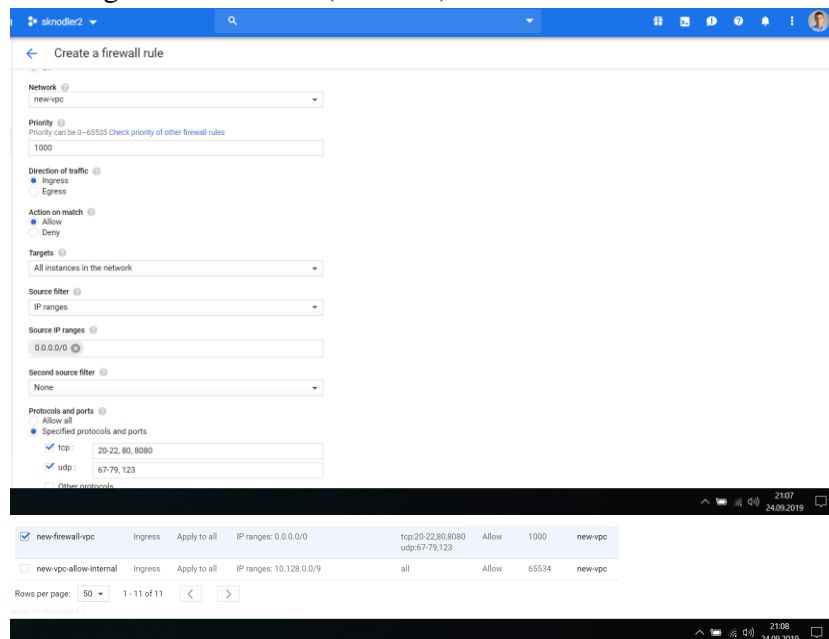
Do you want to continue (Y/n)? Y

Deleted [https://www.googleapis.com/compute/v1/projects/sknodler2/global/networks/new-vpc-1].
st_knoedler@cloudshell:~ (sknodler2)$
```

<https://cloud.google.com/vpc/docs/using-firewalls>

Complete the following sub-modules in Using firewall rules:

### A. Creating Firewall Rules (Console)



### B. Configuration examples - Recreate sample network configurations in the scenarios listed below

- Example 1: Deny all ingress TCP connections except those to port 80 from subnet1

```
st_knoedler@cloudshell:~ (sknodler2)$ gcloud compute firewall-rules create deny-subnet1-webserver-access \
--network new-vpc \
--action deny \
--direction ingress \
--rules tcp \
--source-ranges 0.0.0.0/0 \
--priority 1000 \
--target-tags webserver
Creating firewall...done.
Deny-subnet1-webserver-access new-vpc INGRESS 1000 ALLOW tcp False
```

```
st_knodler@cloudshell: (sknodler2) $ gcloud compute firewall-rules create vml-allow-ingress-tcp-port80-from-subnet1 \
> --network new-vpc \
> --action allow \
> --direction ingress \
> --rules tcp:80 \
> --source-ranges 10.240.10.0/24 \
> --priority 50 \
> --target-tags webserver
Creating firewall...Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/firewalls/vml-allow-ingress-tcp-port80-from-subnet1].
Creating firewall...done.
NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED
vml-allow-ingress-tcp-port80-from-subnet1 new-vpc INGRESS 50 tcp:80 False
```

- b. Example 2: Deny all egress TCP connections except those to port 80 of vm1

```
st_knodler@cloudshell: (sknodler2) $ gcloud compute firewall-rules create deny-all-access \
> --network new-vpc \
> --action deny \
> --direction egress \
> --rules tcp \
> --destination-ranges 0.0.0.0/0 \
> --priority 1000
Creating firewall...Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/firewalls/deny-all-access].
Creating firewall...done.
NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED
deny-all-access new-vpc EGRESS 1000 tcp False

st_knodler@cloudshell: (sknodler2) $ gcloud compute firewall-rules create vml-allow-egress-tcp-port80-to-vm1 \
> --network new-vpc \
> --action allow \
> --direction egress \
> --rules tcp:80 \
> --destination-ranges 192.168.10.2/32 \
> --priority 60
Creating firewall...Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/firewalls/vml-allow-egress-tcp-port80-to-vm1].
Creating firewall...done.
NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED
vml-allow-egress-tcp-port80-to-vm1 new-vpc EGRESS 60 tcp:80 False
```

- c. Example 3: Allow egress TCP connections to port 443 of an external host

```
st_knodler@cloudshell: (sknodler2) $ gcloud compute firewall-rules create vml-allow-egress-tcp-port443-to-192-0-2-5 \
> --network new-vpc \
> --action allow \
> --direction egress \
> --rules tcp:443 \
> --destination-ranges 192.0.2.5/32 \
> --priority 70 \
> --target-tags webserver
Creating firewall...Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/firewalls/vml-allow-egress-tcp-port443-to-192-0-2-5].
https://console.cloud.google.com/compute/disks?project=sknodler2
```

- d. Example 4: Allow SSH connections from vm2 to vm1

```
st_knodler@cloudshell: (sknodler2) $ gcloud compute firewall-rules create vml-allow-ingress-tcp-ssh-from-vm2 \
> --network new-vpc \
> --action allow \
> --direction ingress \
> --rules tcp:22 \
> --source-tags database \
> --priority 80 \
> --target-tags webserver
Creating firewall...Created [https://www.googleapis.com/compute/v1/projects/sknodler2/global/firewalls/vml-allow-ingress-tcp-ssh-from-vm2].
Creating firewall...done.
NAME NETWORK DIRECTION PRIORITY ALLOW DENY DISABLED
vml-allow-ingress-tcp-ssh-from-vm2 new-vpc INGRESS 80 tcp:22 False
```

## Section 9: Clean up

<https://cloud.google.com/shell/docs/quickstart#clean-up>

To avoid incurring charges to your Google Cloud Platform account for the resources used in this

I went to the Projects page in the console. Clicked the trash can icon next to the project I created. This shuts down the project and schedules it for deletion

09/30/2019

