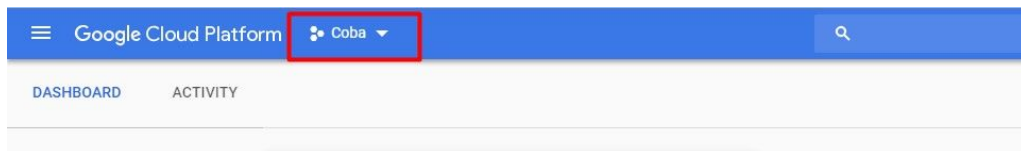


# Activity Log minggu 2

Pada minggu ke -2 kita memutuskan untuk berpidah dari offline host menjadi membuat vm instance di Google Cloud Platform

Caranya adalah :

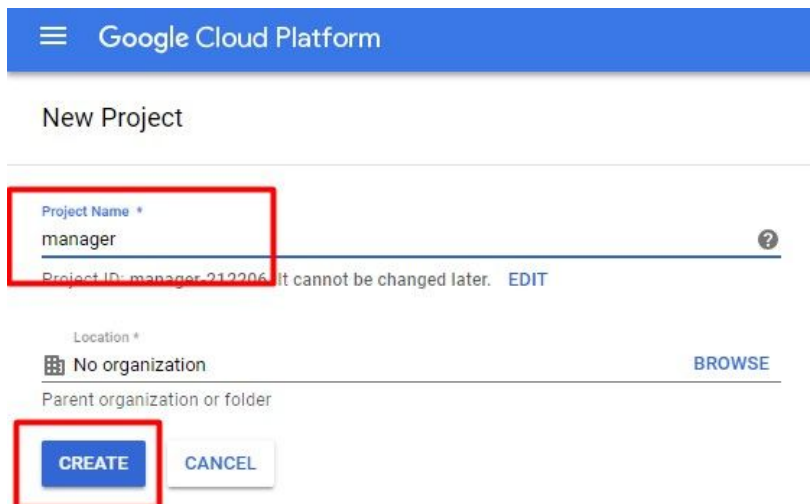
1. Kita masuk Ke Google Cloud Platform yang telah terdaftar, masuk ke **Console**
2. Kemudian klik bagian Project seperti bagian yang diberi kotak merah di bawah ini



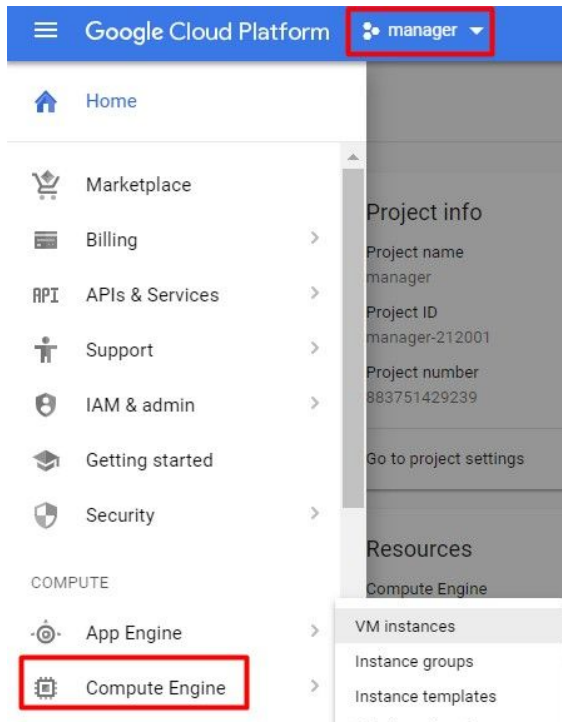
3. Kita buat project baru dengan klik **NEW PROJECT**



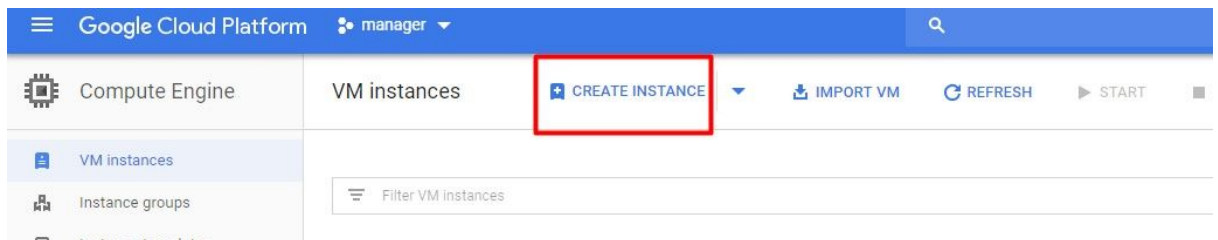
4. Kita isikan Project Namanya **manager** kemudian **CREATE**



5. Selanjutnya kita pilih Project **manager**, kemudian pada bagian menu, pilih **Compute Engine**



6. Kemudian klik **CREATE INSTANCE**



7. Disini kita buat VM dengan nama **maanager1** dengan OSnya adalah Ubuntu 16.04 TLS , jangan lupa untuk mencheck lis Allow HTTP, seperti gambar di bawah ini :

m manager

← Create an instance


Name <sup>?</sup>  
maanager1

Region <sup>?</sup> <sup>?</sup> Zone <sup>?</sup>  
us-east1 (South Carolina) us-east1-b \$2 Eff

Machine type <sup>?</sup>  
Customize to select cores, memory and GPUs. <sup>?</sup>

1 vCPU 3.75 GB memory Customize  
[Upgrade your account](#) to create instances with up to 96 cores

Container <sup>?</sup>  
☐ Deploy a container image to this VM instance. [Learn more](#)

Boot disk <sup>?</sup>  
 New 10 GB standard persistent disk  
Image  
Ubuntu 16.04 LTS [Change](#)

Identity and API access <sup>?</sup>

Service account <sup>?</sup>  
Compute Engine default service account

Access scopes <sup>?</sup>  
☒ Allow default access  
☐ Allow full access to all Cloud APIs  
☐ Set access for each API

Firewall <sup>?</sup>  
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 credits, if available, will be used for this instance

8. Kita juga melakukan hal yang sama membuat VM Instance dengan nama **work1** dan **work2**

The image shows two side-by-side screenshots of the Google Cloud Platform 'Create an instance' wizard. Both screenshots show the same configuration steps for creating a VM instance:

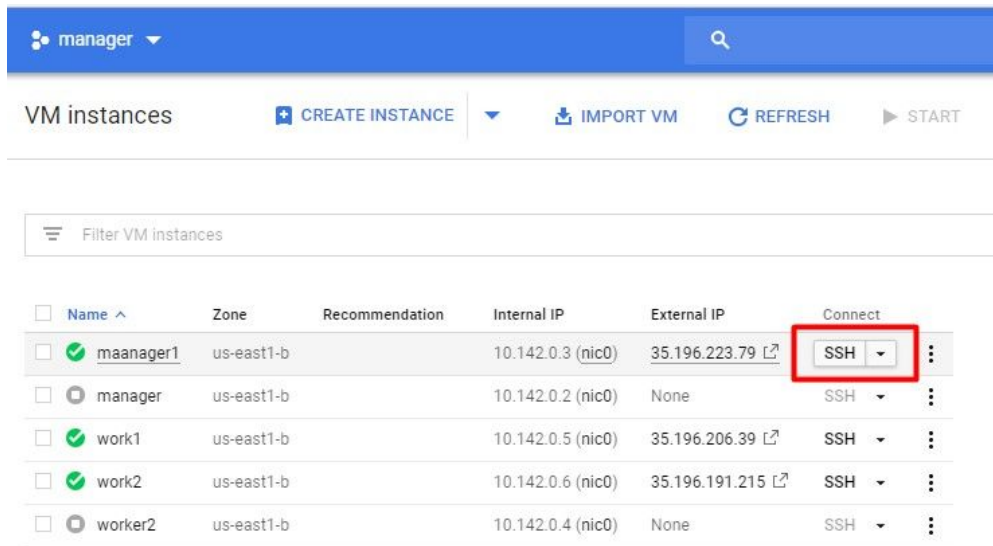
- Name:** work1 (left) / work2 (right)
- Region:** us-east1 (South Carolina)
- Zone:** us-east1-b
- Machine type:** 1 vCPU, 3.75 GB memory
- Boot disk:** New 10 GB standard persistent disk, Image: Ubuntu 16.04 LTS
- Identity and API access:** Service account: Compute Engine default service account, Access scopes: Allow default access
- Firewall:** Allow HTTP traffic

9. Maka sudah terbentuk VM baru dengan ip public secara otomatis

The image shows the Google Cloud Platform 'VM instances' page. The table below lists the VM instances created:

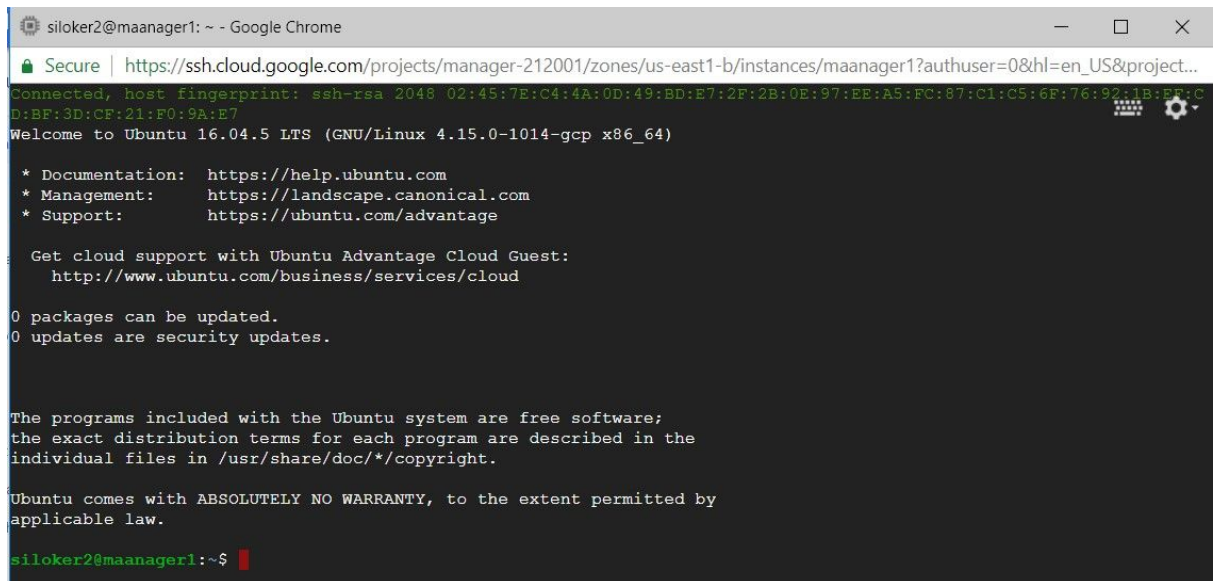
Name	Zone	Recommendation	Internal IP	External IP	Connect
maanager1	us-east1-b		10.142.0.3 (nic0)	35.196.223.79	SSH
manager	us-east1-b		10.142.0.2 (nic0)	None	SSH
work1	us-east1-b		10.142.0.5 (nic0)	35.196.206.39	SSH
work2	us-east1-b		10.142.0.6 (nic0)	35.196.191.215	SSH
worker2	us-east1-b		10.142.0.4 (nic0)	None	SSH

10. Untuk bisa mengakses VM, klik bisa klik SSH :



<input type="checkbox"/>	Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/>	maanager1	us-east1-b		10.142.0.3 (nic0)	35.196.223.79 ↗	SSH ▾
<input type="checkbox"/>	manager	us-east1-b		10.142.0.2 (nic0)	None	SSH ▾
<input type="checkbox"/>	work1	us-east1-b		10.142.0.5 (nic0)	35.196.206.39 ↗	SSH ▾
<input type="checkbox"/>	work2	us-east1-b		10.142.0.6 (nic0)	35.196.191.215 ↗	SSH ▾
<input type="checkbox"/>	worker2	us-east1-b		10.142.0.4 (nic0)	None	SSH ▾

11. Selanjutnya maka akan masuk ke console dari OS VM kita



```
siloker2@maanager1: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/manager-212001/zones/us-east1-b/instances/maanager1?authuser=0&hl=en_US&project...
Connected, host fingerprint: ssh-rsa 2048 02:45:7E:C4:4A:0D:49:BD:E7:2F:2B:0E:97:EE:A5:FC:87:C1:C5:6F:76:92:1B:FF:C
D:BF:3D:CF:21:F0:9A:E7
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-1014-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

siloker2@maanager1:~$
```

12 Kemudian kita install docker seperti pada Minggu Pertama di VM instance maanager1, work1 dan work2 sampai muncul test dengan DOcker image helloworld

Hello world di maanager1

```
siloker2@maanager1: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/manager-212001/zones/us-east1-b/instances/r
Preparing to unpack .../docker-ce_18.06.0~ce~3-0~ubuntu_amd64.deb ...
Unpacking docker-ce (18.06.0~ce~3-0~ubuntu) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for systemd (229-4ubuntu21.4) ...
Setting up pigz (2.3.1-2) ...
Setting up aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
Setting up cgroupfs-mount (1.2) ...
Setting up libltdl7:amd64 (2.4.6-0.1) ...
Setting up docker-ce (18.06.0~ce~3-0~ubuntu) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for systemd (229-4ubuntu21.4) ...
Processing triggers for ureadahead (0.100.0-19) ...
siloker2@maanager1:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
9db2ca6ccae0: Pull complete
Digest: sha256:4b8ff392a12ed9ea17784bd3c9a8b1fa3299cac44aca35a85c90c5e3c7afacdc
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/

siloker2@maanager1:~$
```



## Hello world di work1

```
siloker2@work1: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/manager-212001/zones/us-east1-b/instances/work1
Preparing to unpack .../docker-ce_18.06.0~ce~3-0~ubuntu_amd64.deb ...
Unpacking docker-ce (18.06.0~ce~3-0~ubuntu) ...
sudo docker run hello-worldProcessing triggers for man-db (2.7.5-1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for systemd (229-4ubuntu21.4) ...
Setting up pigz (2.3.1-2) ...
Setting up aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
Setting up cgroupfs-mount (1.2) ...
Setting up libltdl7:amd64 (2.4.6-0.1) ...
Setting up docker-ce (18.06.0~ce~3-0~ubuntu) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for systemd (229-4ubuntu21.4) ...
Processing triggers for ureadahead (0.100.0-19) ...
siloker2@work1:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
9db2ca6ccae0: Pull complete
Digest: sha256:4b8ff392a12ed9ea17784bd3c9a8b1fa3299cac44aca35a85c90c5e3c7afacdc
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/

siloker2@work1:~$
```

Hello world di work2

```
siloker2@work2: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/manager-212001/zones/us-east1-b/instances/work2
Preparing to unpack .../docker-ce_18.06.0~ce~3-0~ubuntu_amd64.deb ...
Unpacking docker-ce (18.06.0~ce~3-0~ubuntu) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for systemd (229-4ubuntu21.4) ...
Setting up pigz (2.3.1-2) ...
Setting up aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
Setting up cgroupfs-mount (1.2) ...
Setting up libltdl7:amd64 (2.4.6-0.1) ...
Setting up docker-ce (18.06.0~ce~3-0~ubuntu) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for systemd (229-4ubuntu21.4) ...
Processing triggers for ureadahead (0.100.0-19) ...
siloker2@work2:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
9db2ca6ccae0: Pull complete
Digest: sha256:4b8ff392a12ed9ea17784bd3c9a8b1fa3299cac44aca35a85c90c5e3c7afacdc
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/

siloker2@work2:~$
```