Deploy Hortonworks Sandbox on Google Cloud:

The following steps need to be done only for the first time:

1. Login to your Google Cloud account (https://cloud.google.com) and open a Google Cloud Shell. Make sure you are in the intended project.
2. Run the following script on **Google Cloud Shell** to launch a new instance named ambari. Copy this command to an editor of your choice and make sure to replace <PROJECT-ID> with your project-id (it can be found by clicking on the project name, under ID):

```
gcloud compute --project=<PROJECT-ID> instances create ambari \
--zone=us-east1-b --machine-type=n1-highmem-2 \
--subnet=default --network-tier=STANDARD --no-restart-on-failure \
--maintenance-policy=TERMINATE  --image=coreos-stable-1967-3-0-v20190108 \
--image-project=coreos-cloud --boot-disk-size=60GB \
--boot-disk-type=pd-standard --boot-disk-device-name=ambari \
#--preemptible
```

- This will give you a machine with 60 GB HHD and 13GB Memory (n1-highmem-2)

3. Once the instance is ready (External IP address will be populated) you should see the green sign next to its name under Compute Engine section. From there you can click on SSH button to SSH into your machine.
   - For the first time it may take a while until SSH button works. If that is the case from Cloud Shell run this command:
     - gcloud compute --project "< PROJECT-ID >" ssh --zone "us-east1-b" "ambari"
4. From the **SSH window** download the HDP2.5 image into your instance:
   - `wget https://storage.googleapis.com/is843/HDP_2.5_docker.tar.gz`
   - In case the above command wasn't available you can download from the original source, it could take much longer:
     - wget https://downloads-hortonworks.akamaized.net/sandbox-hdp-2.5/HDP_2.5_docker.tar.gz
5. Pass this image to docker:

```
docker load < HDP_2.5_docker.tar.gz
```

6. Now start this docker image by running the following command:

```
docker run -v hadoop:/hadoop --name sandbox --hostname \
"sandbox.hortonworks.com" --privileged -d -p 6080:6080 -p 9090:9090 \
-p 9000:9000  -p 8000:8000 -p 8020:8020 -p 42111:42111 -p 10500:10500 \
-p 16030:16030 -p 8042:8042 -p 8040:8040 -p 2100:2100 -p 4200:4200 \
-p 4040:4040 -p 8050:8050 -p 9996:9996 -p 9995:9995 -p 8080:8080 -p 8088:8088 \
-p 8886:8886 -p 8889:8889 -p 8443:8443 -p 8744:8744 -p 8888:8888 -p 8188:8188 \
-p 8983:8983 -p 1000:1000 -p 1100:1100 -p 11000:11000 -p 10001:10001 \
-p 15000:15000 -p 10000:10000 -p 8993:8993 -p 1988:1988 -p 5007:5007 \
-p 50070:50070 -p 19888:19888 -p 16010:16010 -p 50111:50111 -p 50075:50075 \
-p 50095:50095 -p 18080:18080 -p 60000:60000 -p 8090:8090 -p 8091:8091 \
-p 8005:8005 -p 8086:8086 -p 8082:8082 -p 60080:60080 -p 8765:8765 \
-p 5011:5011 -p 6001:6001 -p 6003:6003 -p 6008:6008 -p 1220:1220 \
-p 21000:21000 -p 6188:6188 -p 61888:61888 -p 2181:2181 -p 2222:22 \
sandbox /usr/sbin/sshd -D
```

- Now the container is running. You can confirm this by `docker ps`, it should return a non-empty list of running containers.

7. Start sandbox services, Ambari and HDP services do not start automatically when you start the Docker container. You need to start the processes with a script:
   - `ssh -p 2222 root@localhost`
     - the password for root is “hadoop”. You will be asked to set a new password. Remember it and write it down somewhere safe.
   - `/etc/init.d/startup_script start`
8. While you are here let’s set a password for the Ambari “admin” account:

   ```
   ambari-admin-password-reset
   ```
9. When you see all the services are OK and have been started we are done with the Ambari setup. We can close this window.
10. Go back to **Google Cloud Shell** and create a new firewall rule that allows INGRESS tcp:8080 with VMs containing tag 'allow-tcp-8080':

    ```
    gcloud compute firewall-rules create rule-allow-tcp-8080 \
    --source-ranges 0.0.0.0/0 --target-tags allow-tcp-8080 --allow tcp:8080
    ```
    - This needs to be done once per project. If you have done it before you can ignore it.
11. Set the zone that instance is located at

    ```
    gcloud config set compute/zone us-east1-b
    ```
12. Add the 'allow-tcp-8080' tag to your new VM:

    ```
    gcloud compute instances add-tags ambari --tags allow-tcp-8080
    ```
13. From Google Compute Engine page write down the external IP address of instance “ambari”. Have this you can log in to Ambari in your local browser by:
    - <External-IP-Address>.8080
    - Use maria_dev for username and password!
14. IMPORTANT: Stop the instance at the end of the session. You are paying $0.09/hour to use this instance and if you forget to stop it at the end of each session your $50 credit would last for a little over 3 weeks.

Resetting password for maria_dev

1. Log in to Ambari with “admin” username and the password you set in the previous step.
2. Go to Manage Ambari > Users
3. Reset the password for maria_dev to something secure.
4. It is recommended to deactivate the other users, or reset their passwords. You can find out about other roles and their privileges here: https://hortonworks.com/tutorial/learning-the-ropes-of-the-hortonworks-sandbox/

Restarting the Stopped Instance

- From Compute Engine select your “ambari“ instance, and from the top menu click Start.
- SSH into the machine and run the following so your container starts again
  ```
  docker restart sandbox
  ```
- Start sandbox services, Ambari and HDP services do not start automatically when you start the Docker container. You need to start the processes with a script:
  - `ssh -p 2222 root@localhost`
    - Use your “root” password that you set while setting up this instance.
  - `/etc/init.d/startup_script start`
- Because we stopped the instance the IP address has changed. Look up the new external IP address from Compute Engine and log in to Ambari from your local browser:
  - <External-IP-Address>.8080

Clean up

To make sure we won’t be charged for any of the resources delete the instance:

- From the Compute Engine page select the instance and click on the DELETE button.

Note: You won’t be charged for the instance while it’s stopped. But you will get charged for the disk, which is about $2/month for our 60GB configuration.