Module 2

Big Data Processing using Cloudera Quickstart with a Docker Container

Thanachart Numnonda, Executive Director, IMC Institute
Mr. Aekanun Thongtae, Big Data Consultant, IMC Institute

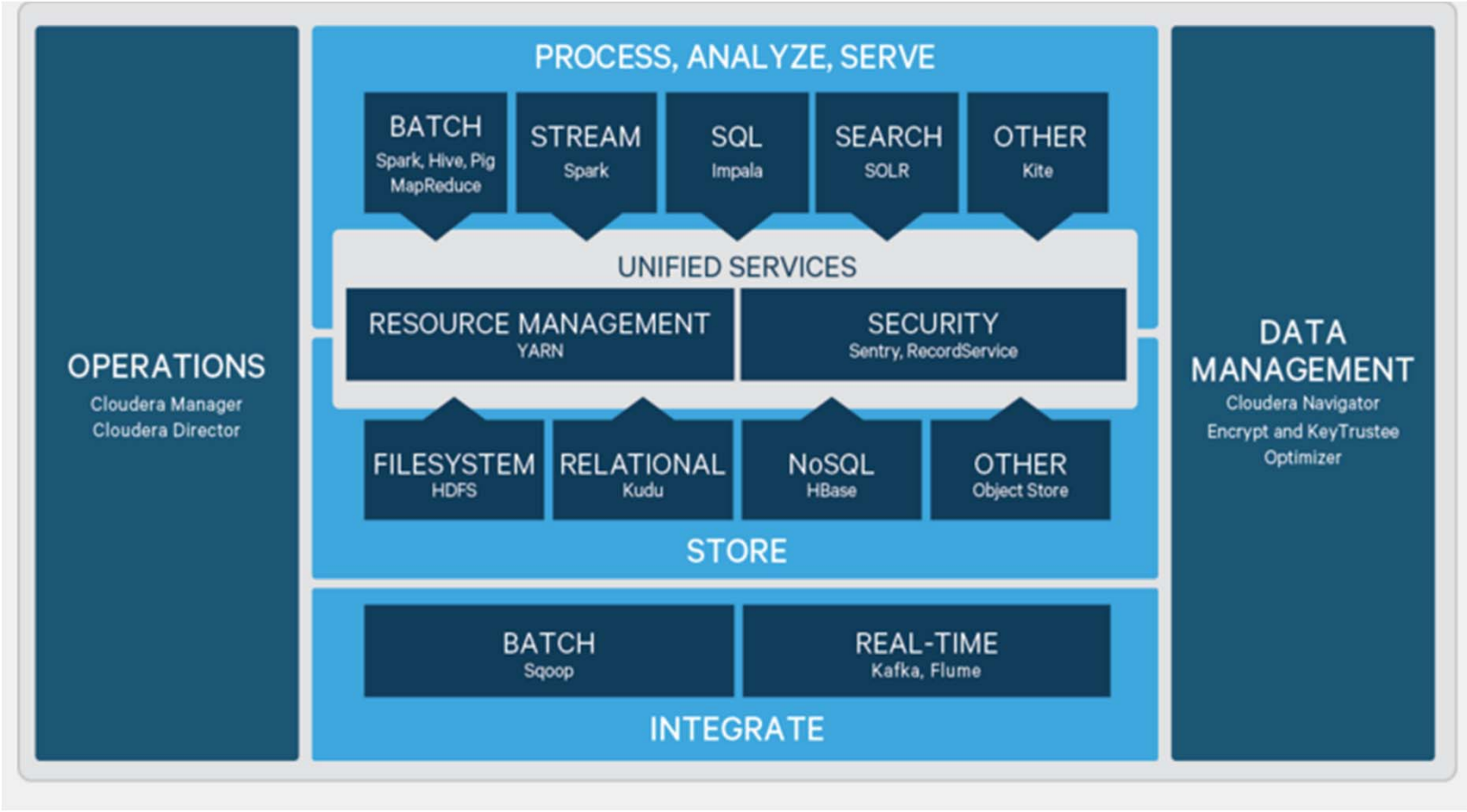
Thanisa Numnonda, Faculty of Information Technology,

King Mongkut's Institute of Technology Ladkrabang

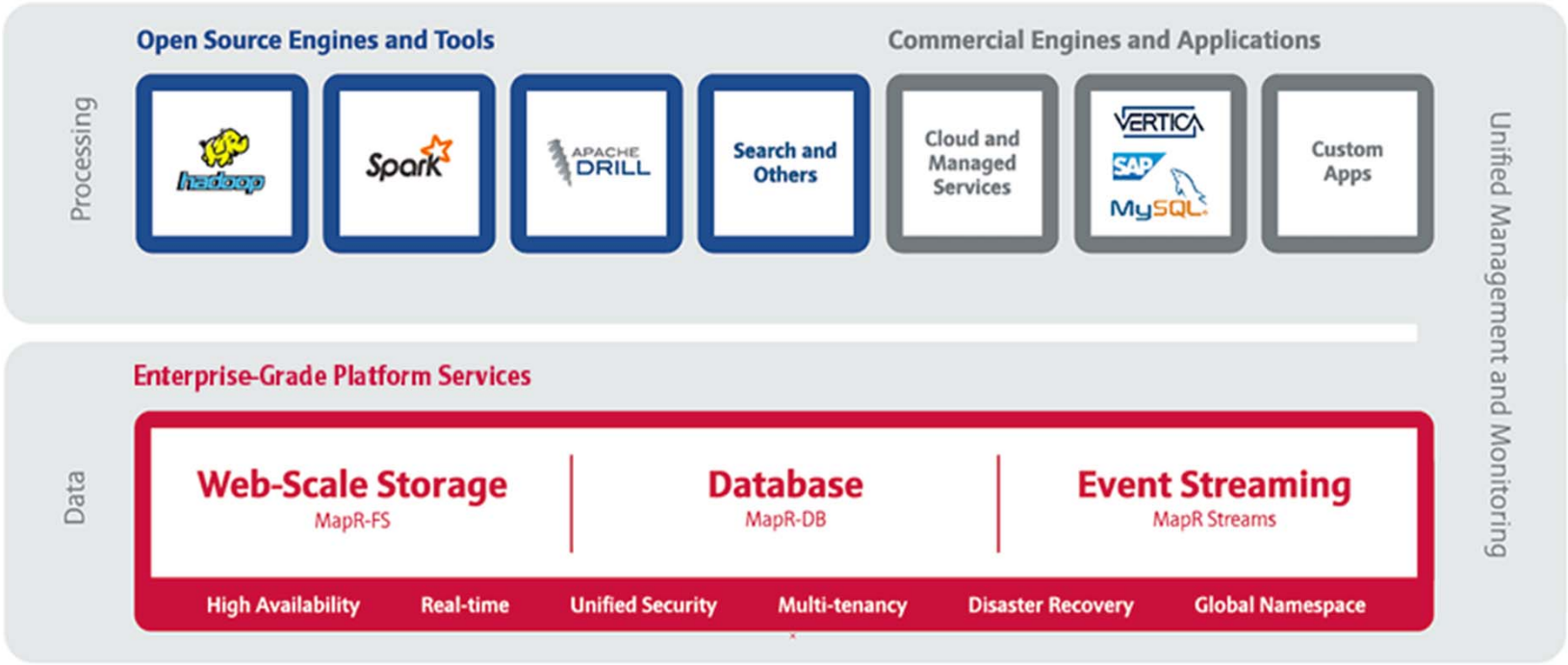
Hadoop Distribution

- **On-Premise**
  - Pure Apache Hadoop
  - Cloudera
  - MapR
  - Hortonworks
- **On-Cloud (Hadoop as a Service)**
  - Amazon EMR
  - Microsoft Azure HDInsight
  - Google Cloud

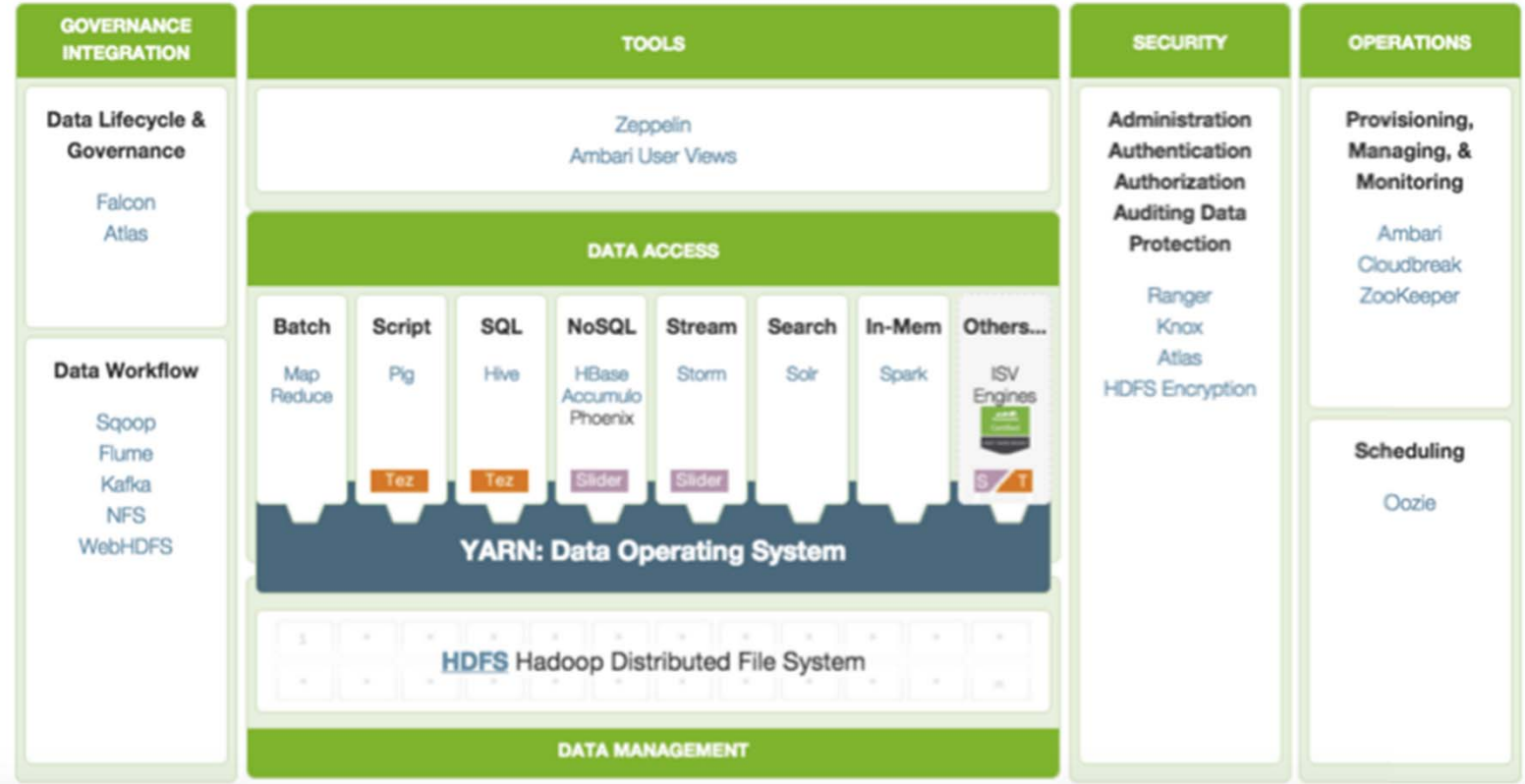
Cloudera


OPERATIONS
Cloudera Manager
Cloudera Director
PROCESS, ANALYZE, SERVE
BATCH
Spark, Hive, Pig
MapReduce
STREAM
Spark
SQL
Impala
SEARCH
SOLR
OTHER
Kite
UNIFIED SERVICES
RESOURCE MANAGEMENT
YARN
SECURITY
Sentry, RecordService
FILESYSTEM
HDFS
RELATIONAL
Kudu
NoSQL
HBase
OTHER
Object Store
STORE
BATCH
Sqoop
REAL-TIME
Kafka, Flume
INTEGRATE
DATA MANAGEMENT
Cloudera Navigator
Encrypt and KeyTrustee
Optimizer

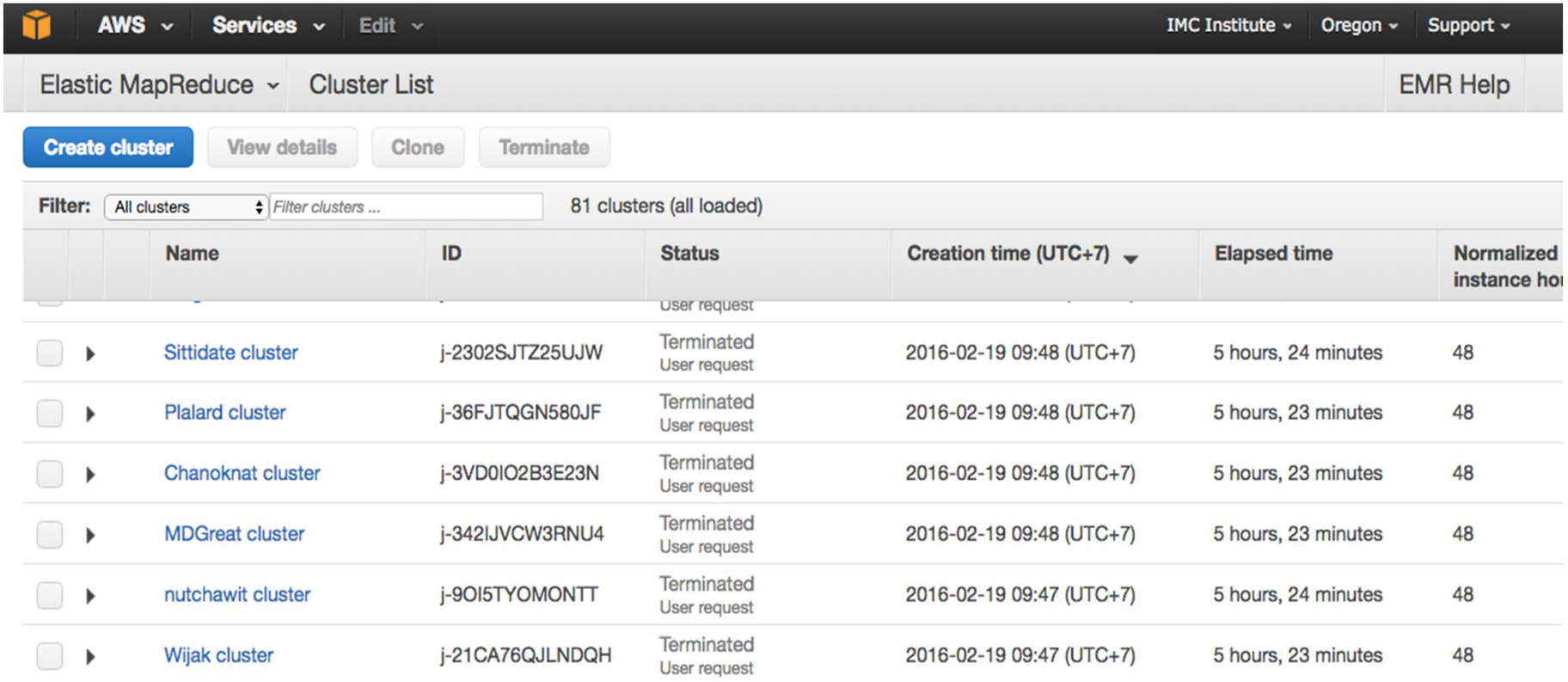
MapR


Processing
Open Source Engines and Tools
Commercial Engines and Applications
hadoop
Spark
APACHE DRILL
Search and Others
Cloud and Managed Services
VERTICA
SAP
MySQL
Custom Apps
Unified Management and Monitoring
Data
Enterprise-Grade Platform Services
Web-Scale Storage
MapR-FS
Database
MapR-DB
Event Streaming
MapR Streams
High Availability
Real-time
Unified Security
Multi-tenancy
Disaster Recovery
Global Namespace

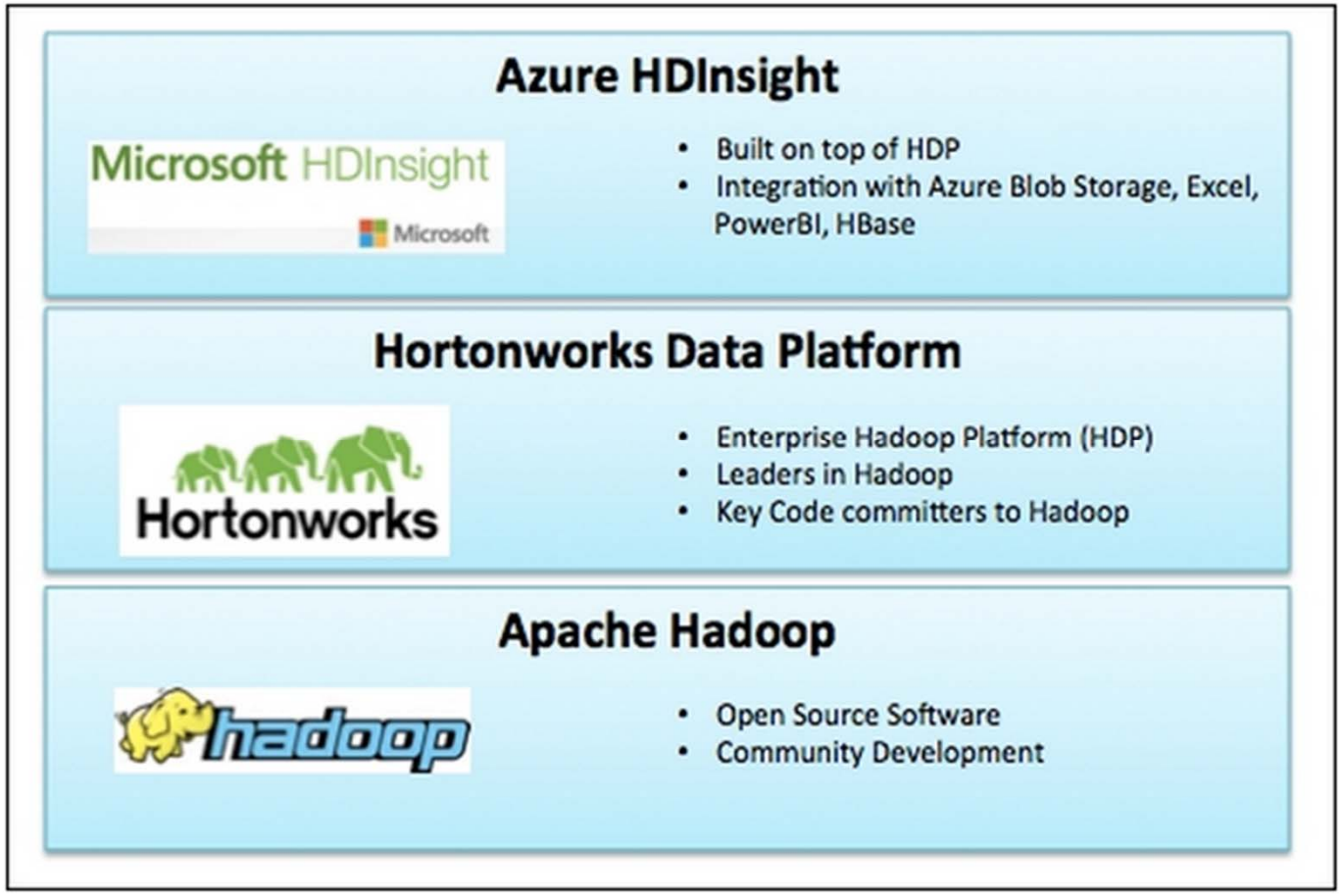
Hortonworks



Amazon EMR


AWS
Services
Edit
IMC Institute
Oregon
Support
Elastic MapReduce
Cluster List
EMR Help
Create cluster
View details
Clone
Terminate
Filter:
All clusters
Filter clusters ...
81 clusters (all loaded)
Name
ID
Status
Creation time (UTC+7)
Elapsed time
Normalized instance hou
User request
Sittidate cluster
j-2302SJTZ25UJW
Terminated
User request
2016-02-19 09:48 (UTC+7)
5 hours, 24 minutes
48
Plalard cluster
j-36FJTQGN580JF
Terminated
User request
2016-02-19 09:48 (UTC+7)
5 hours, 23 minutes
48
Chanoknat cluster
j-3VD0IO2B3E23N
Terminated
User request
2016-02-19 09:48 (UTC+7)
5 hours, 23 minutes
48
MDGreat cluster
j-342IJVCW3RNU4
Terminated
User request
2016-02-19 09:48 (UTC+7)
5 hours, 23 minutes
48
nutchawit cluster
j-9OI5TYOMONTT
Terminated
User request
2016-02-19 09:47 (UTC+7)
5 hours, 24 minutes
48
Wijak cluster
j-21CA76QJLNDQH
Terminated
User request
2016-02-19 09:47 (UTC+7)
5 hours, 23 minutes
48

Microsoft Azure HDInsight



Source: HDInsight Essentials - Second Edition

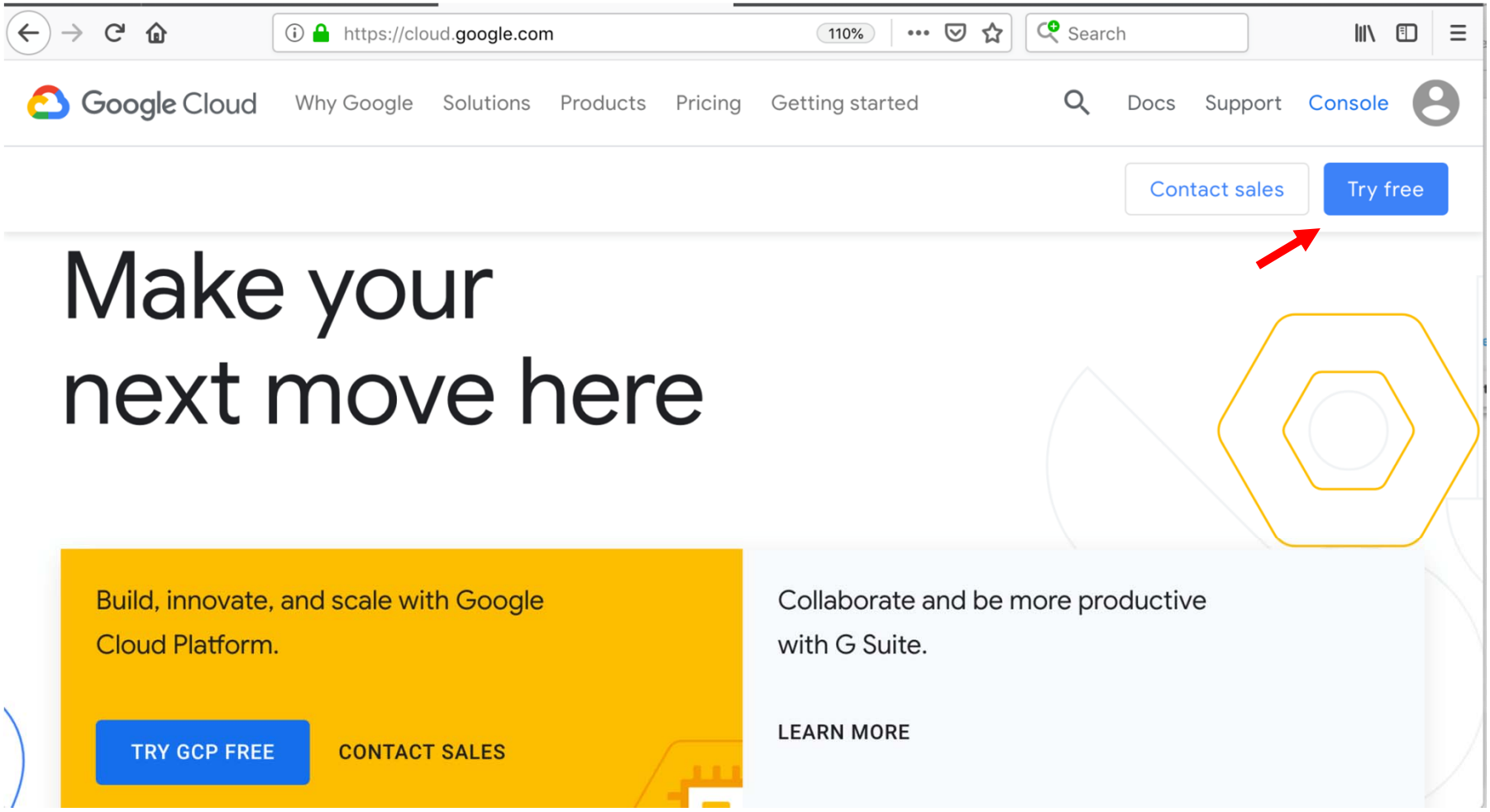
Running Cloudera Docker

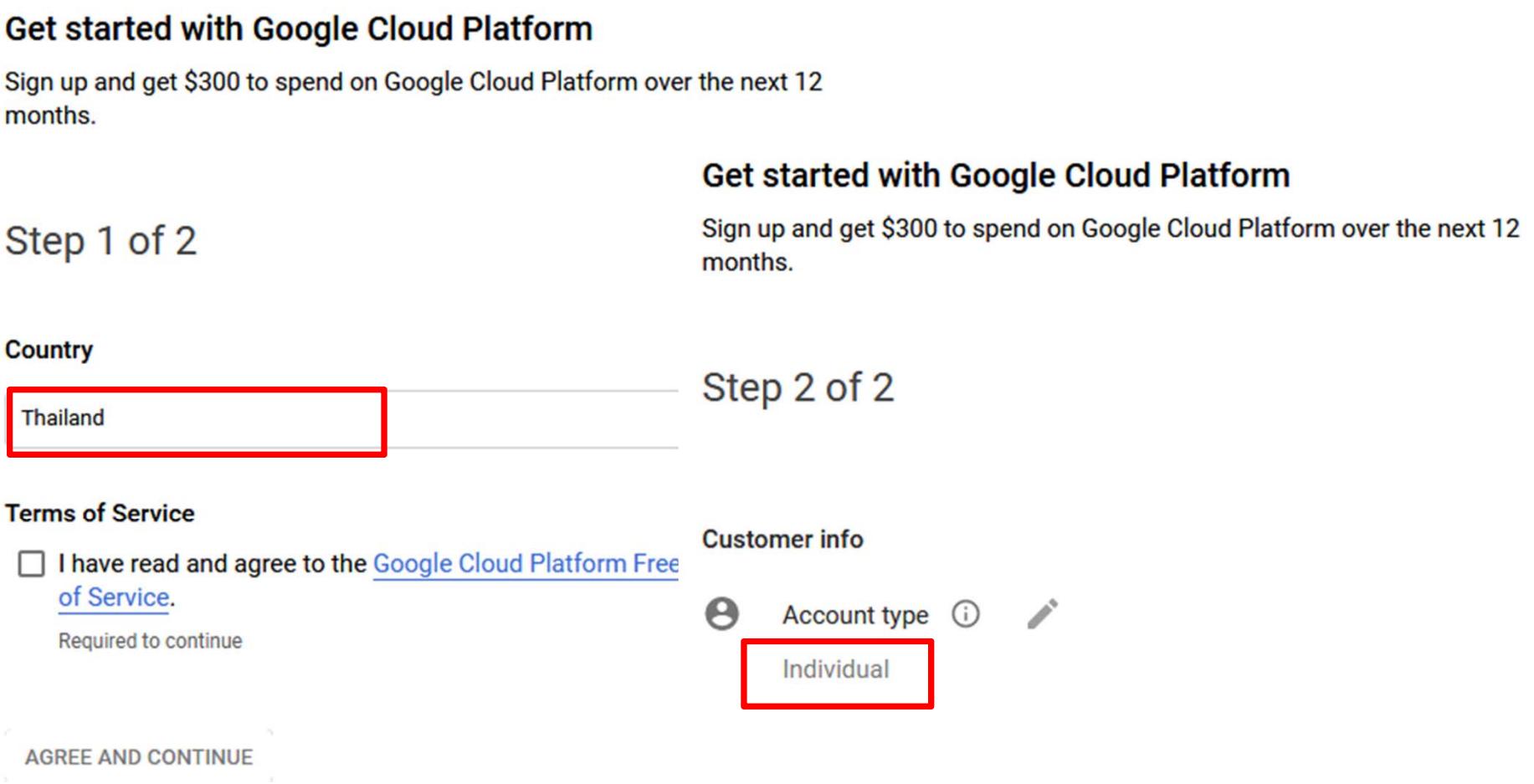
on Google Cloud

Our Works

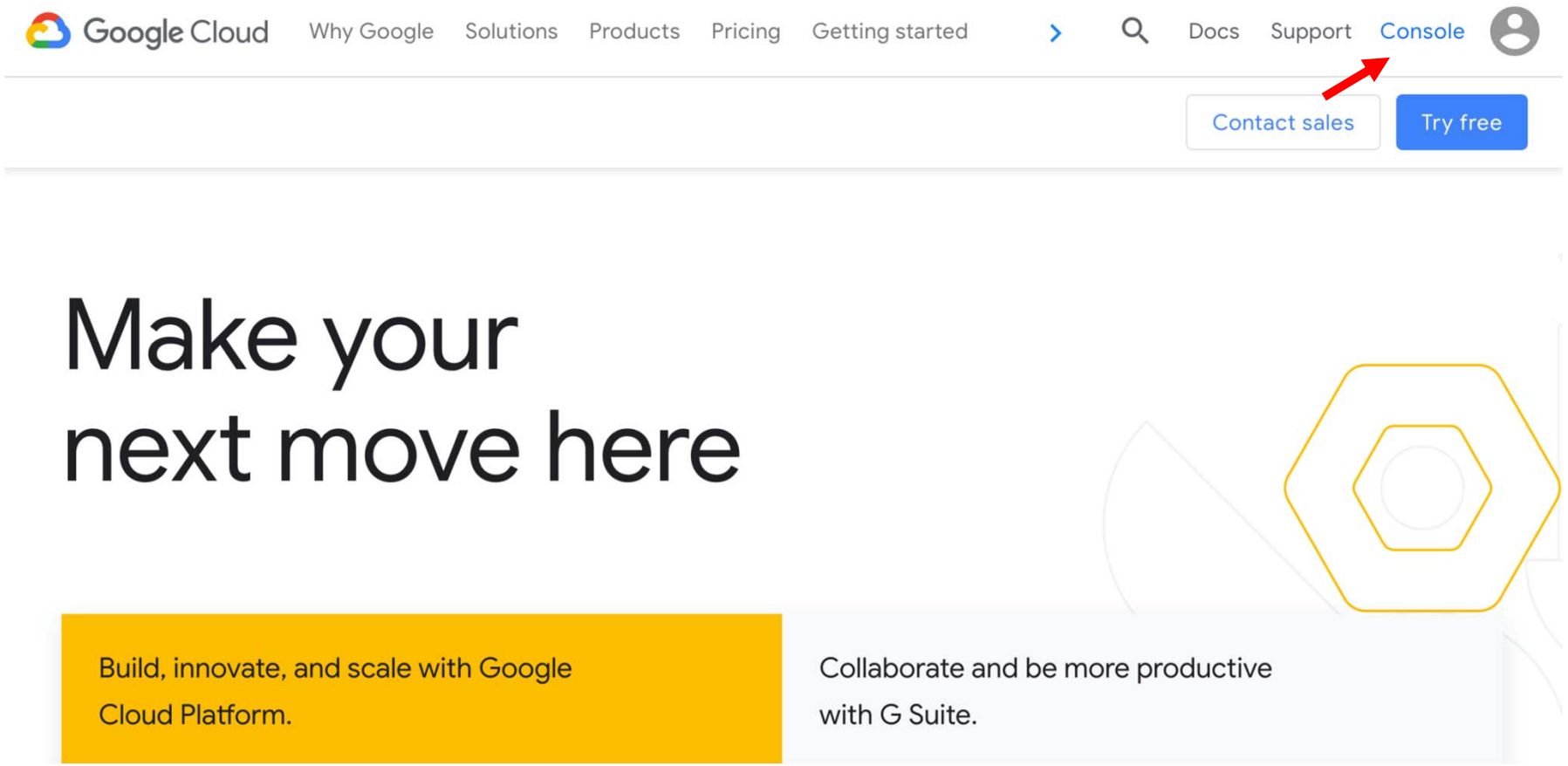
- Lauch a Virtual Server (Ubuntu) on Google Cloud
- Install Docker on Ubuntu
- Pull Cloudera QuickStart to the Docker

cloud.google.com => Try free

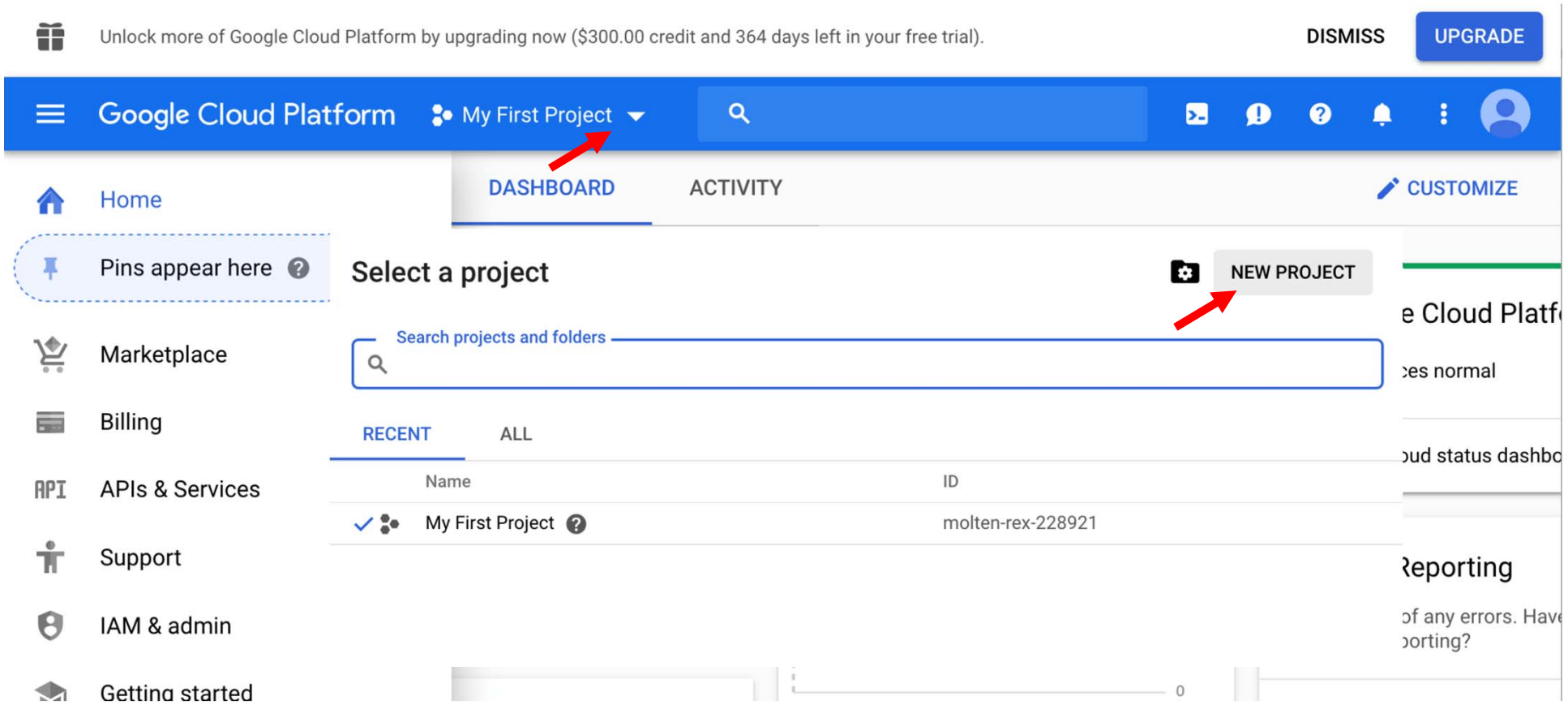



Get started with Google Cloud Platform
Sign up and get $300 to spend on Google Cloud Platform over the next 12 months.
Step 1 of 2
Country
Thailand
Terms of Service
I have read and agree to the Google Cloud Platform Free
of Service.
Required to continue
AGREE AND CONTINUE
Get started with Google Cloud Platform
Sign up and get $300 to spend on Google Cloud Platform over the next 12 months.
Step 2 of 2
Customer info
Account type
Individual

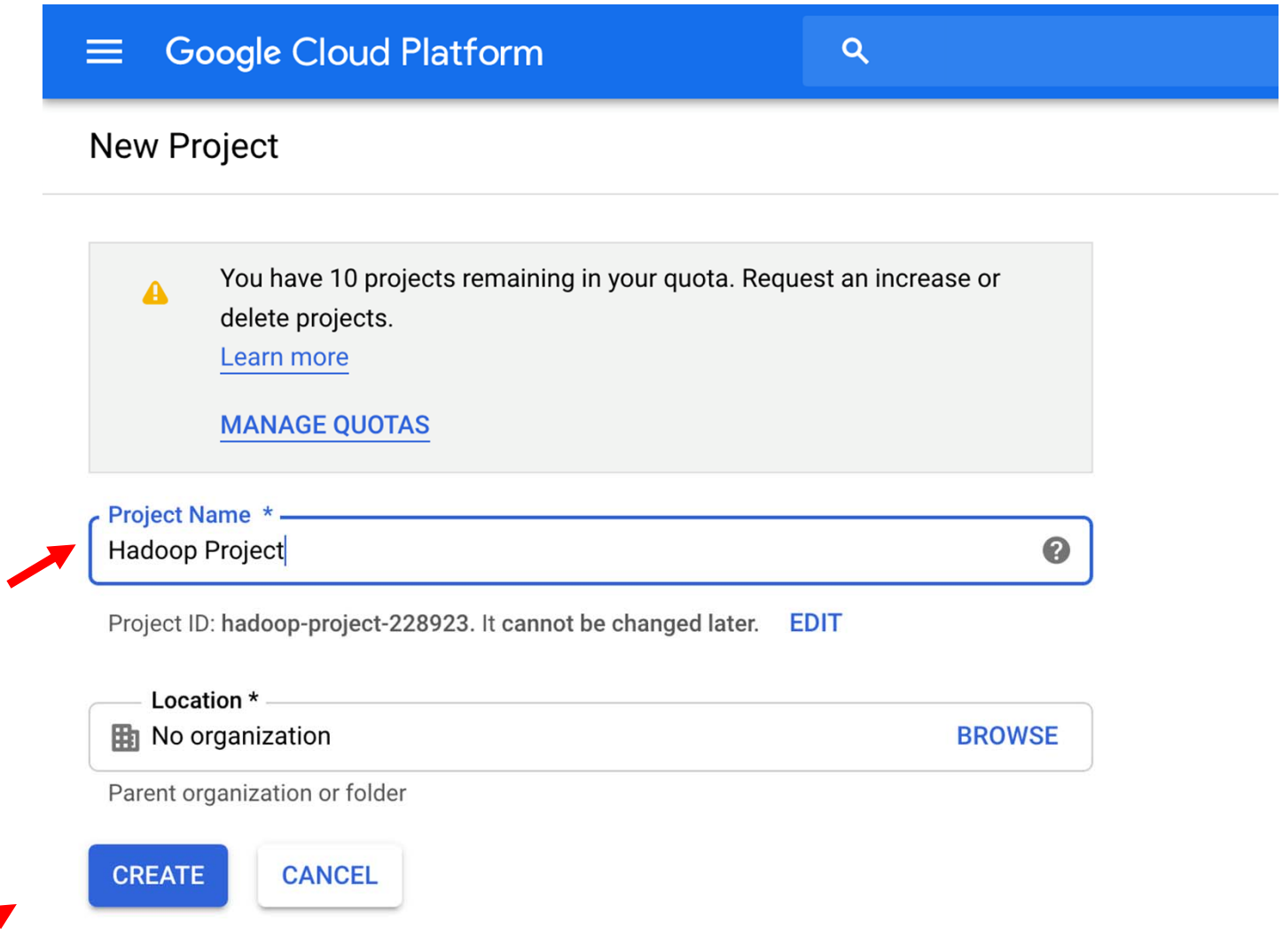
cloud.google.com => Console



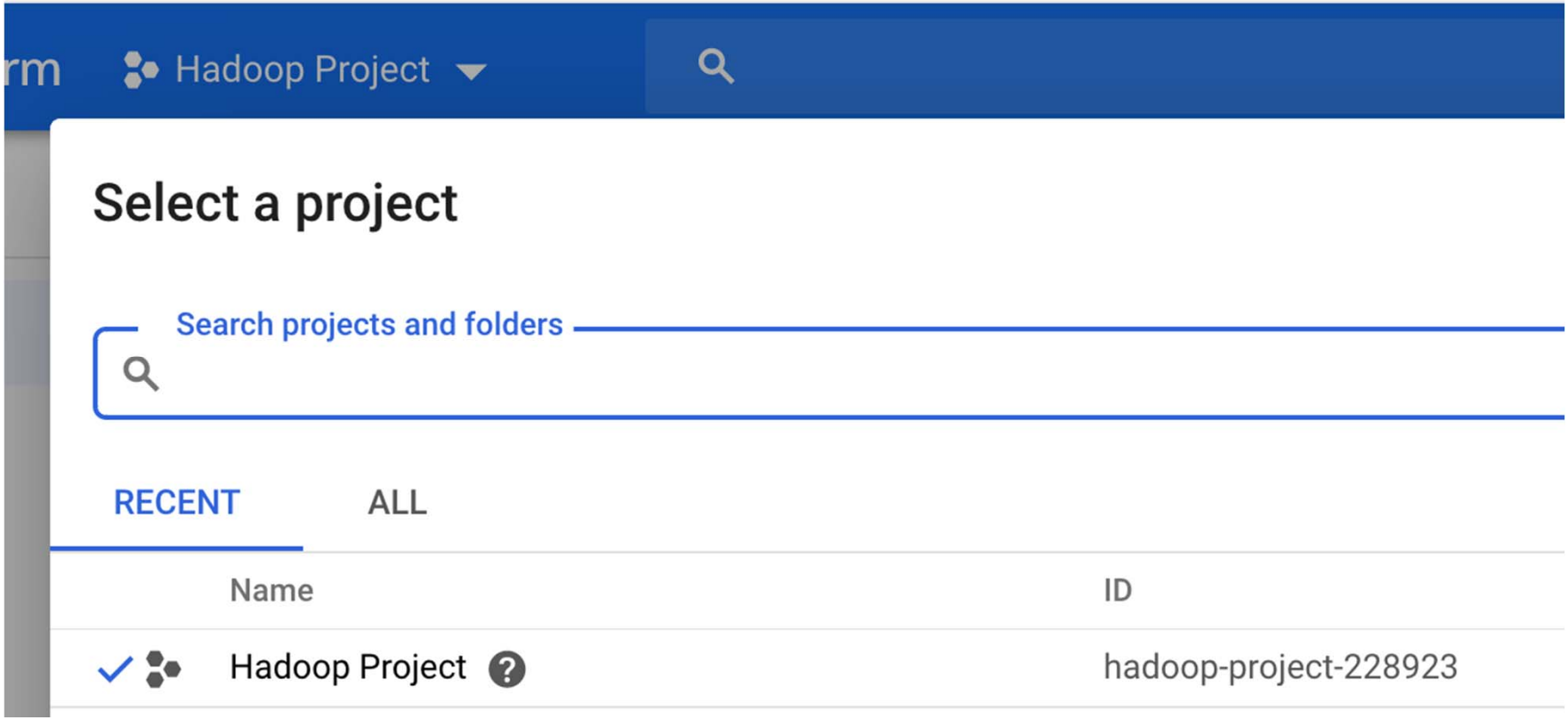
Create Google Cloud Project



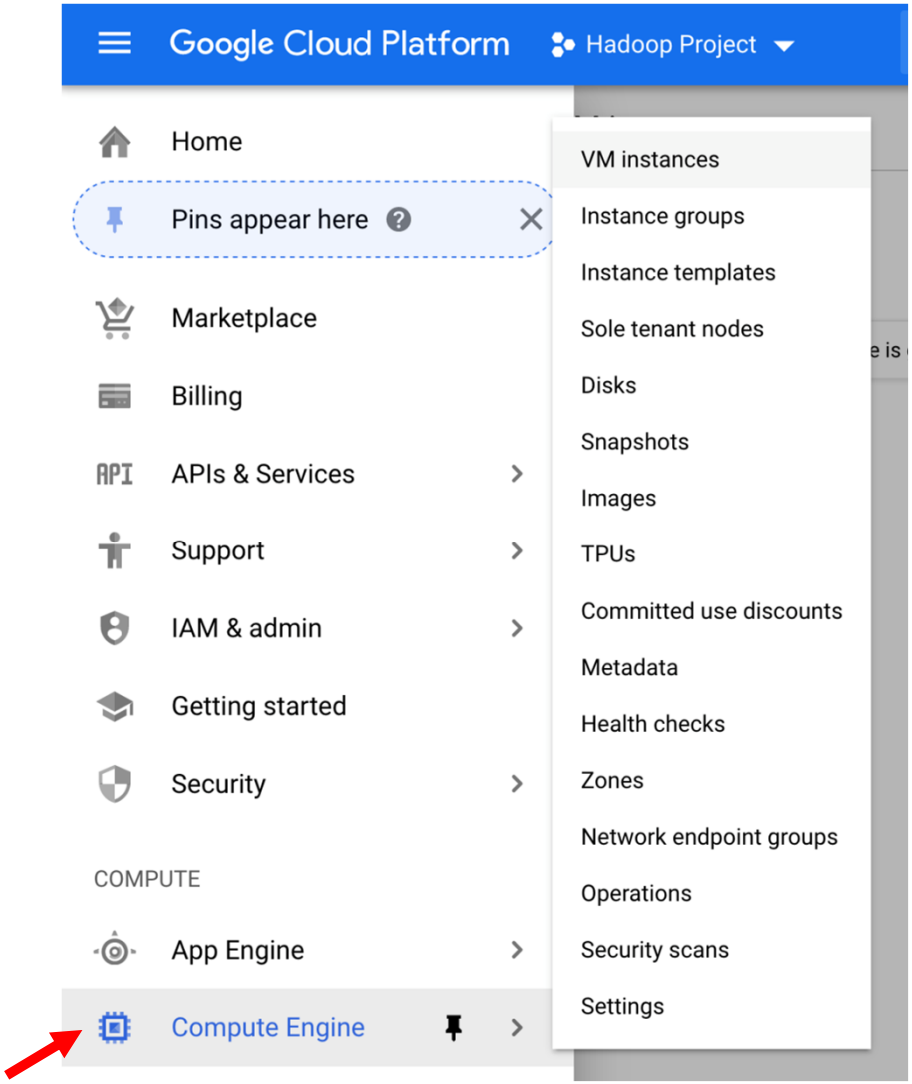
Create Google Cloud Project (Cont.)



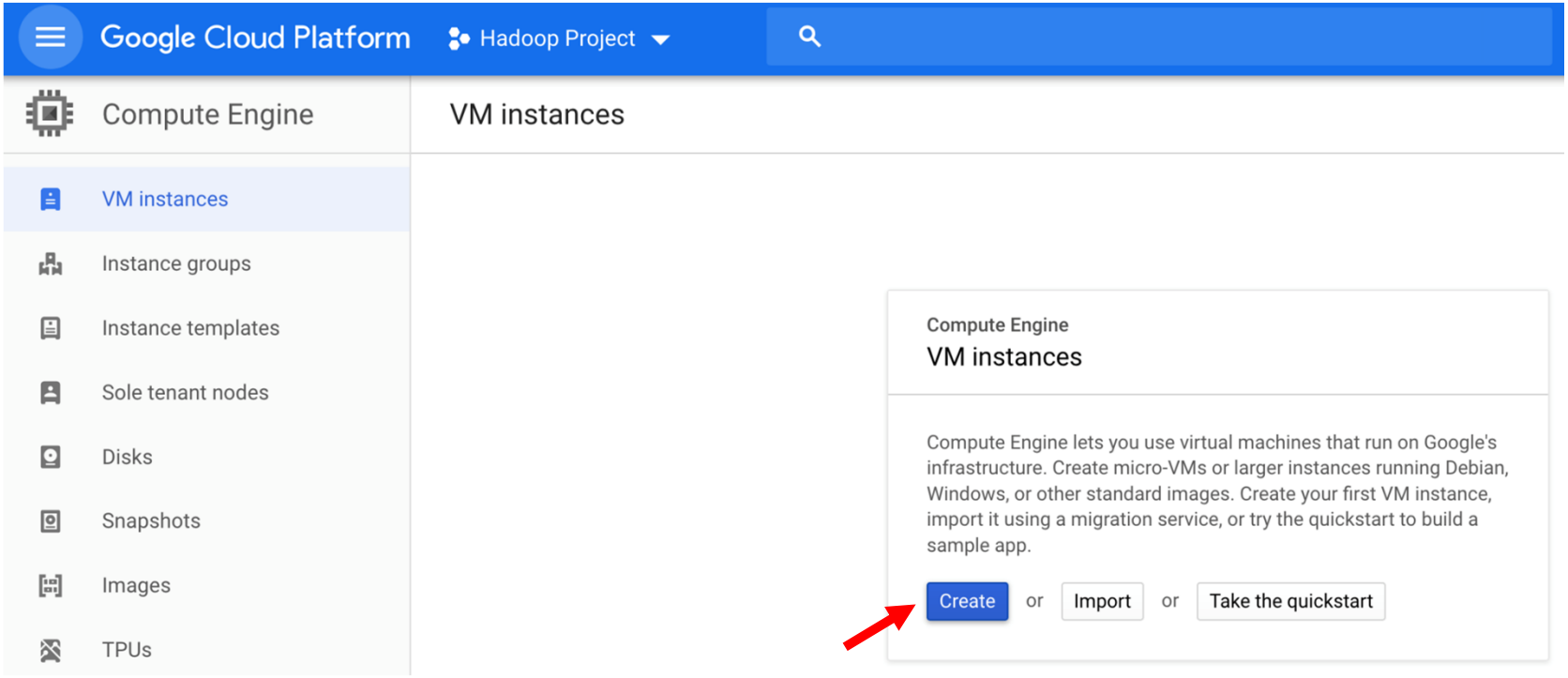
Select Google Cloud Project to **Hadoop Project**



Select **Compute Engine**

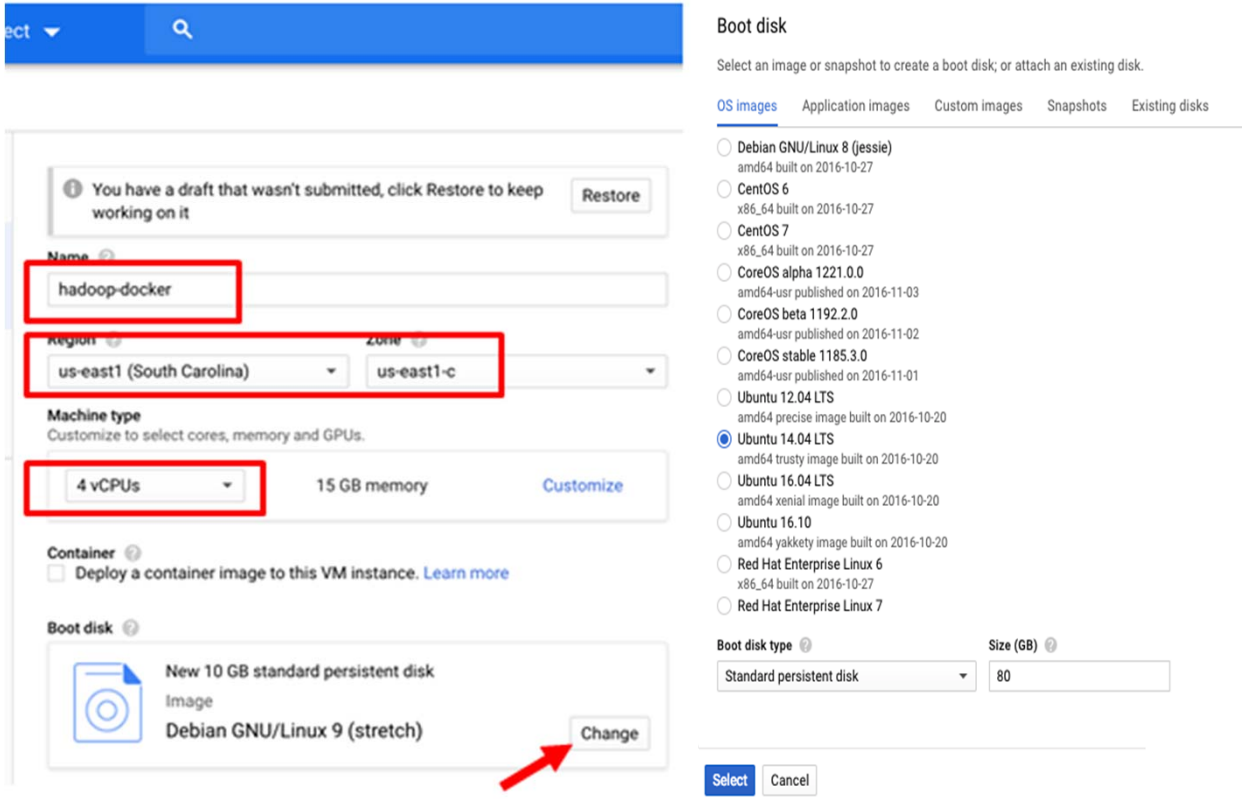


Create VM instances



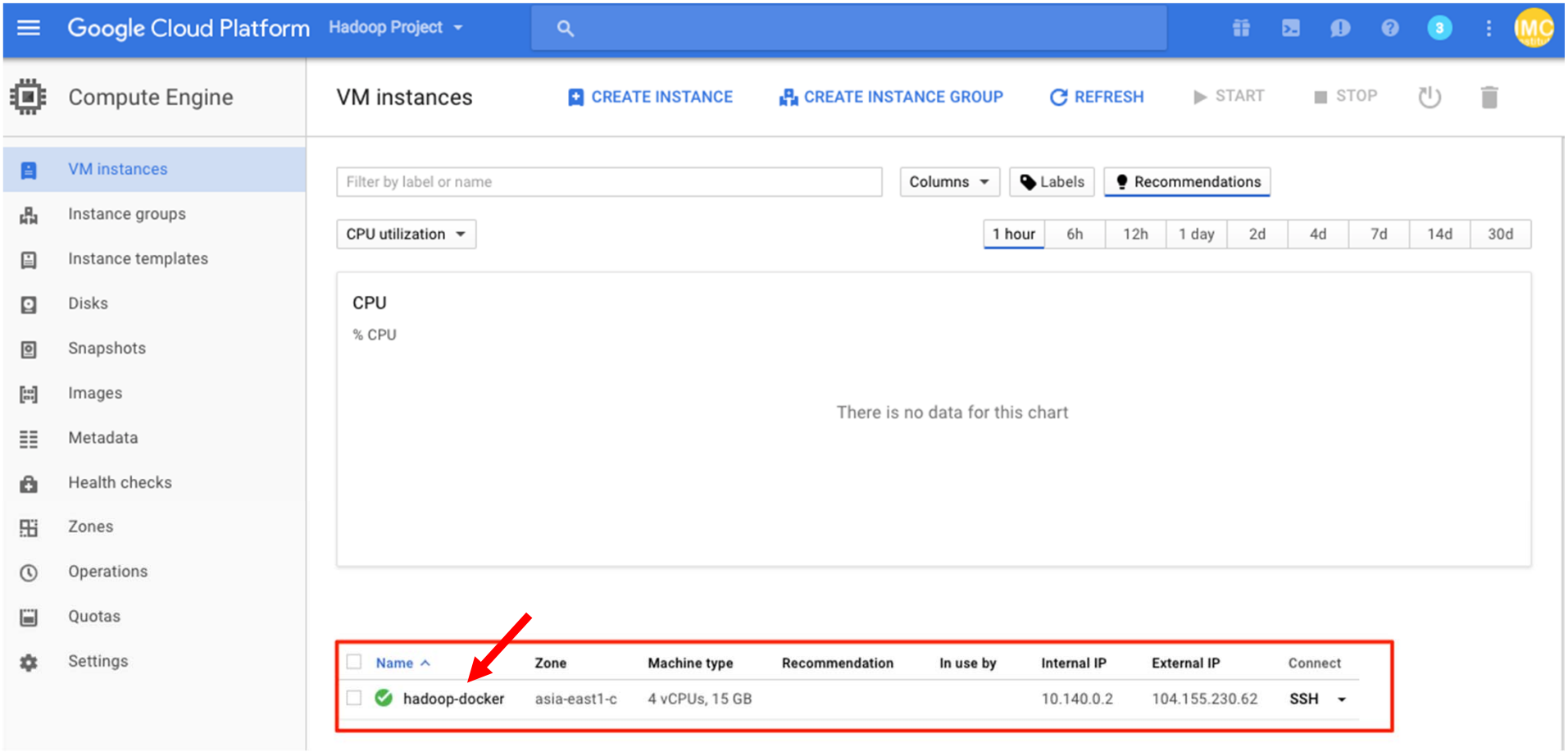
Create an instance with the following configuration

- Name: **hadoop-docker**
- Region : **us-east1**
- Zone : **us-east1-c**
- Machine type : **4vCPUs**

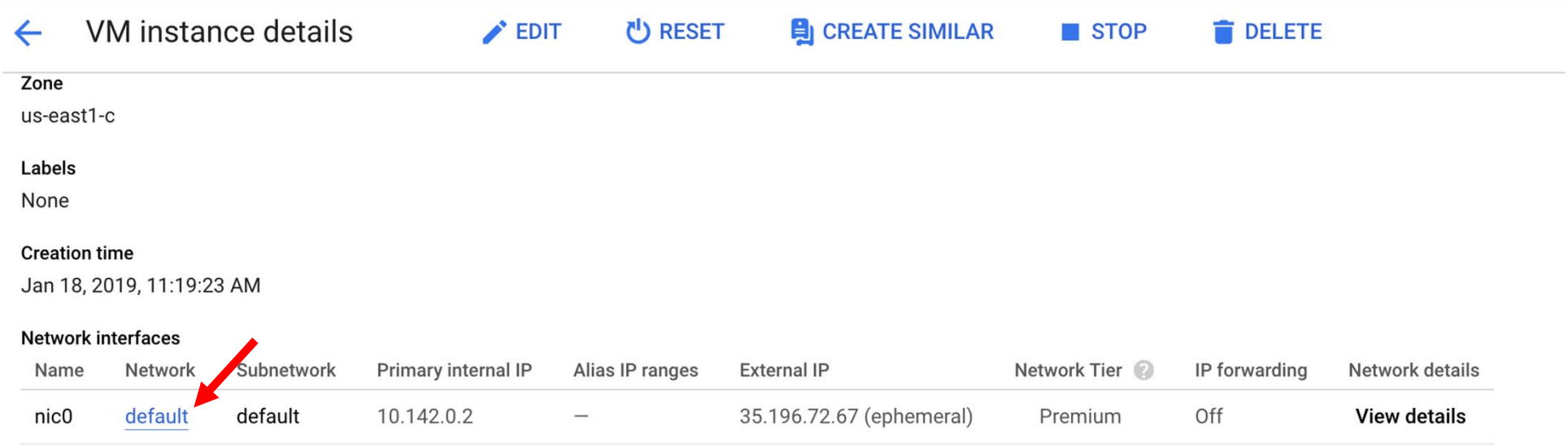


- Book disk : **Ubuntu Server 14.04 LTS**
- Size (GB) : **80**

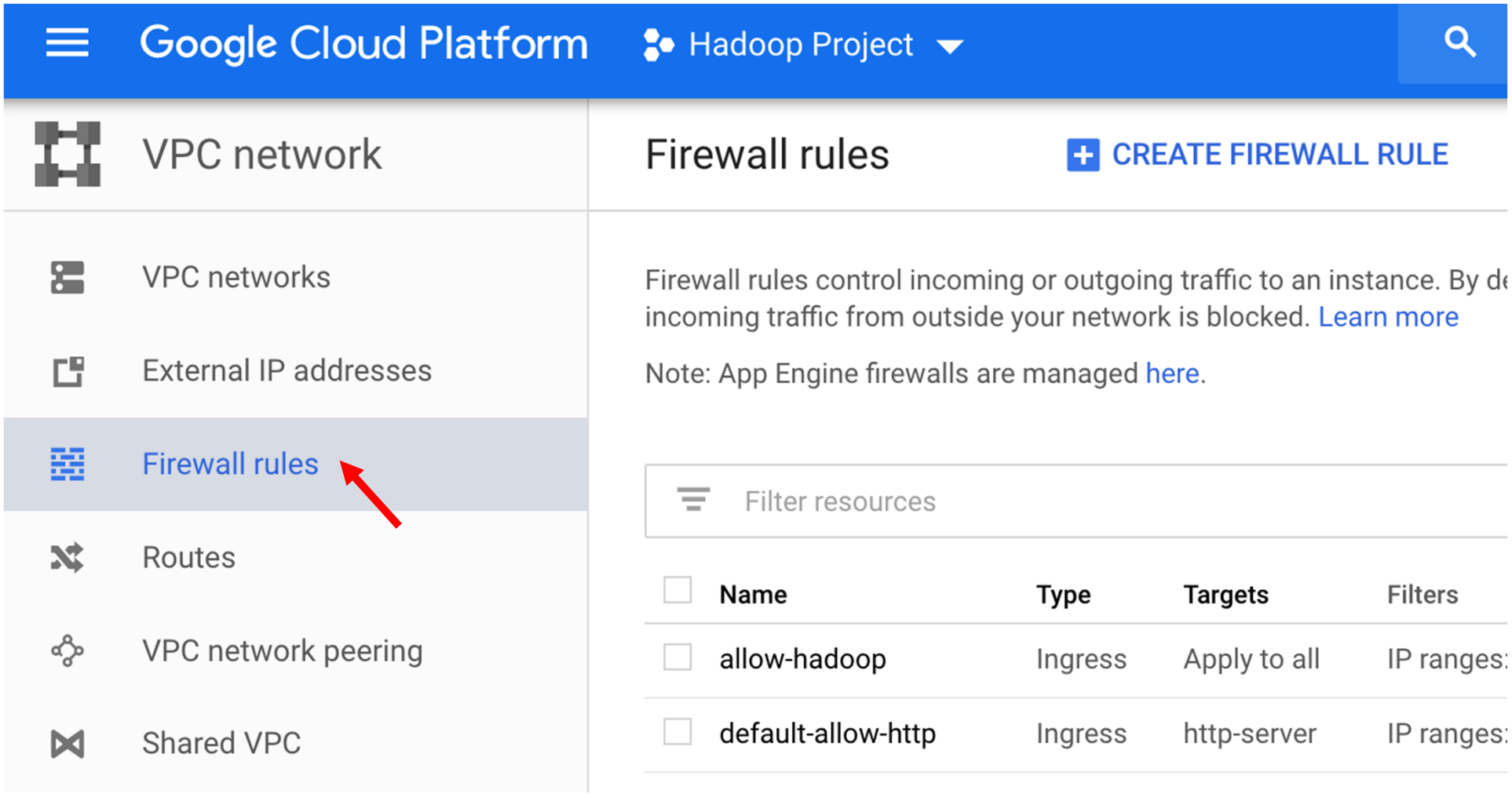
Config Firewall: Select the instance



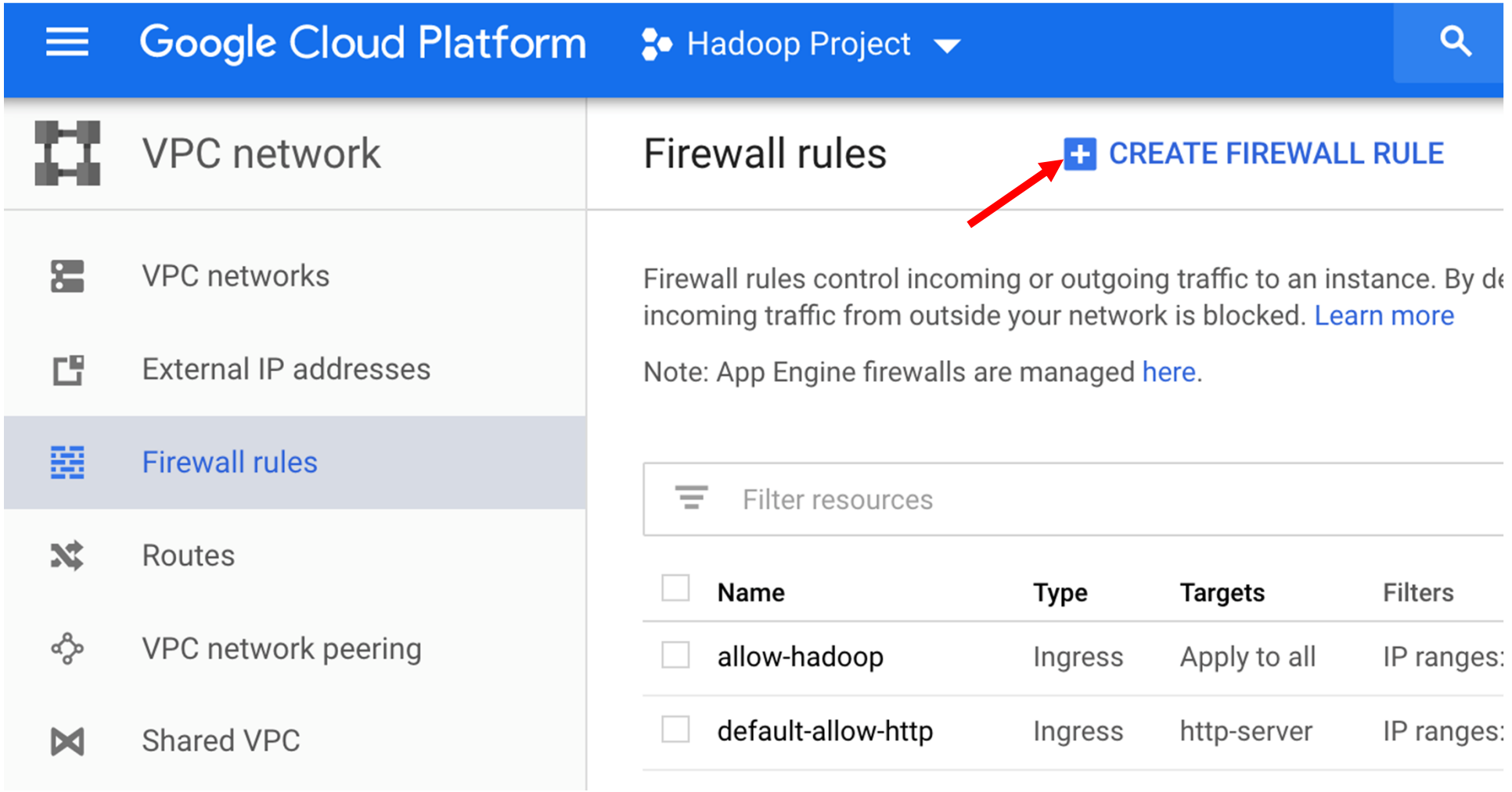
Config Firewall: Select the Network

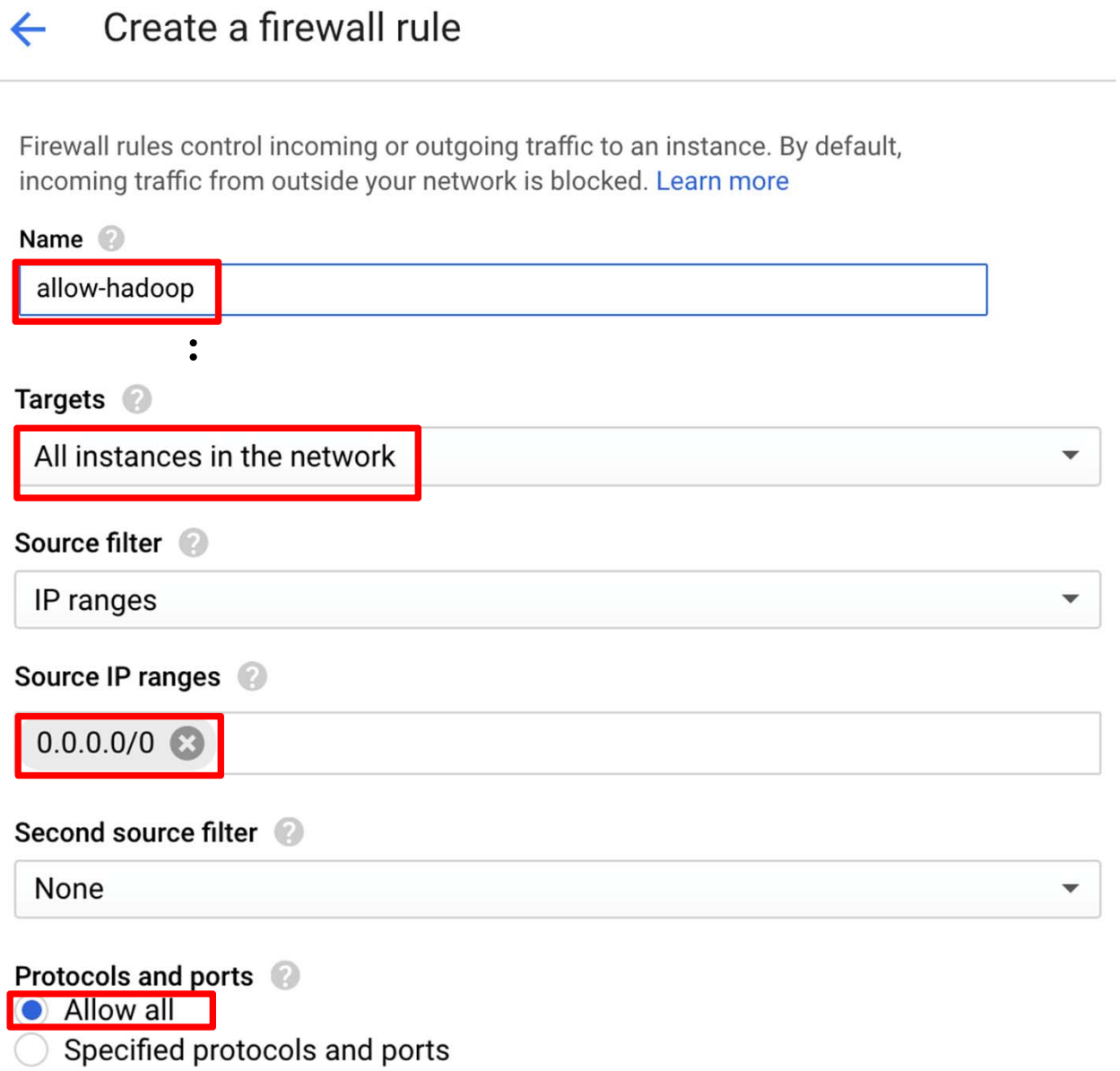


Config Firewall: Select Firewall rules

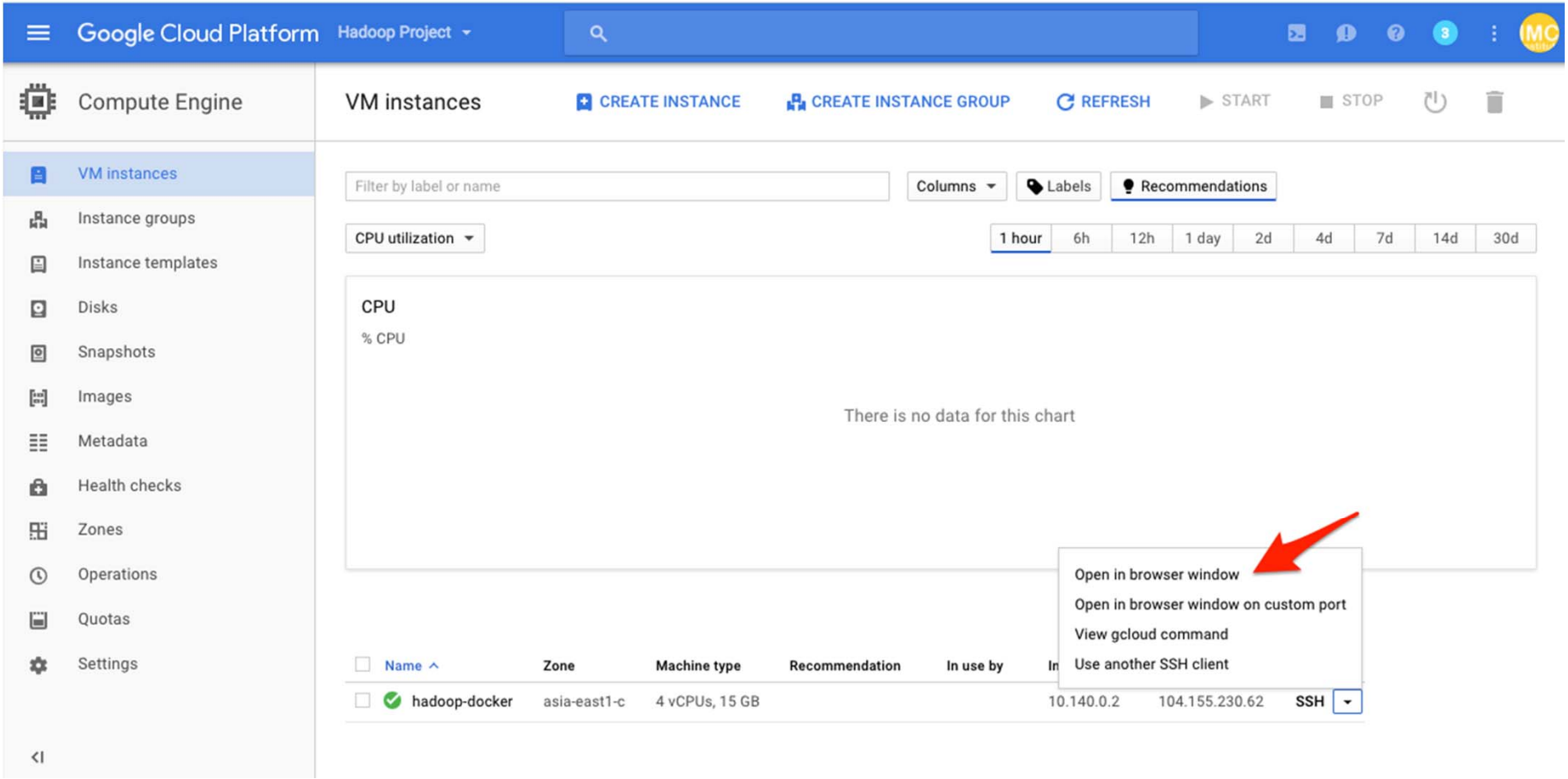


Create Firewall Rule




Create a firewall rule
Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more
Name
allow-hadoop
Targets
All instances in the network
Source filter
IP ranges
Source IP ranges
0.0.0.0/0
Second source filter
None
Protocols and ports
Allow all
Specified protocols and ports

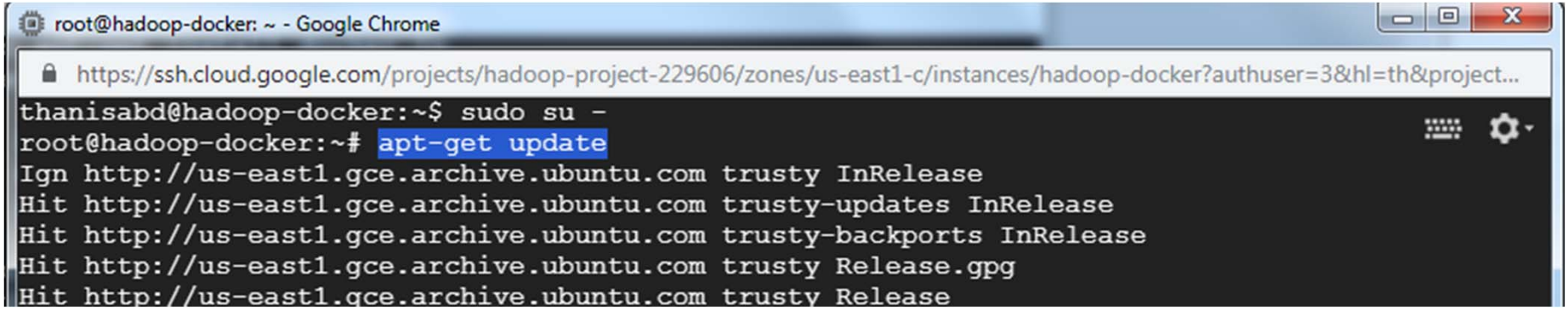
Connect via SSH in browser window



Update OS (Ubuntu)

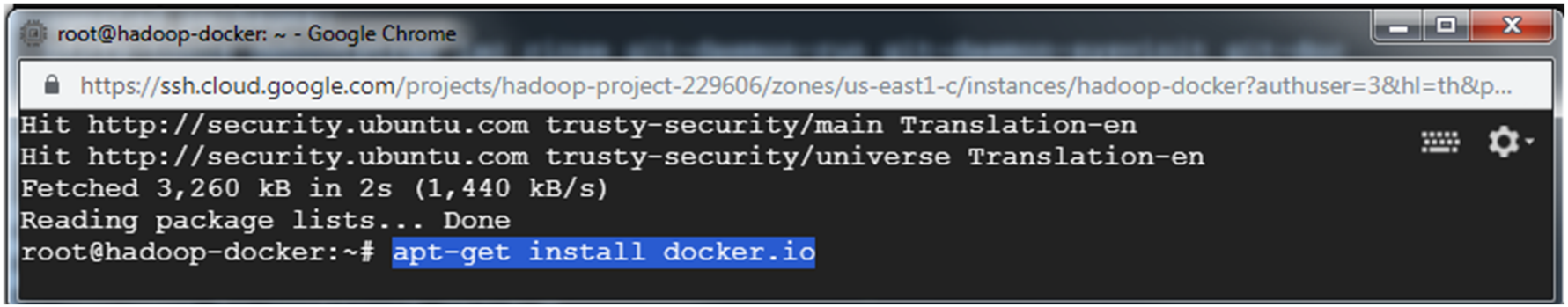
$ sudo su -

apt-get update



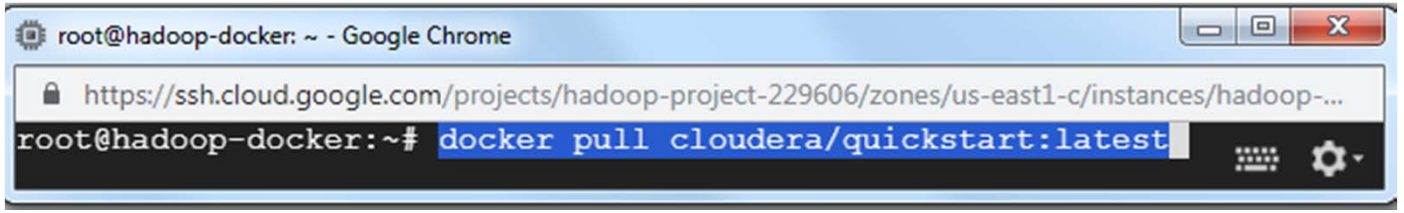
Docker Installation

apt-get install docker.io



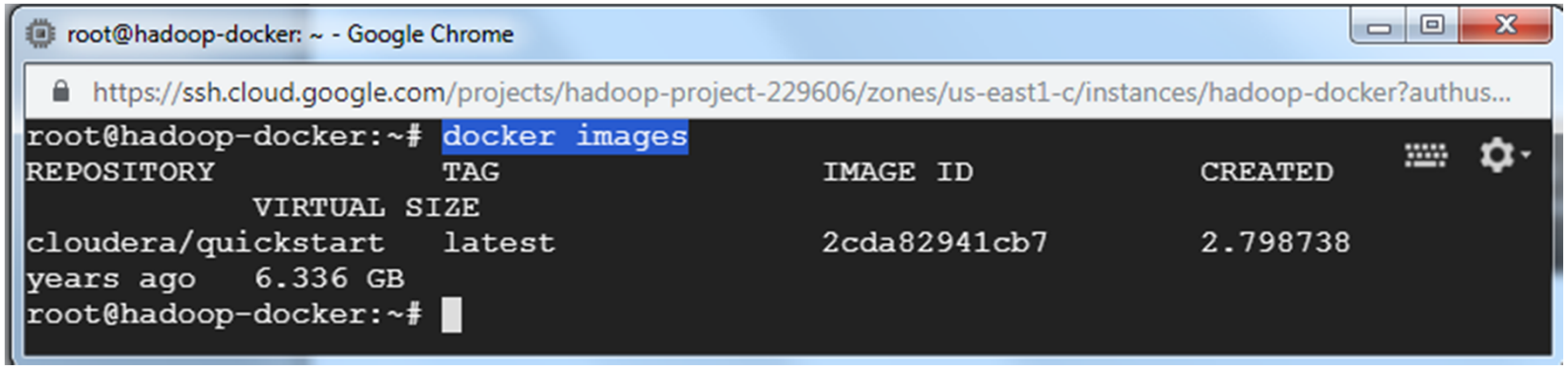
Pull Cloudera Quickstart =>
Install Cloudera Quickstart on Docker Container

docker pull cloudera/quickstart:latest



Verify the image was successfully pulled

docker images



Run Cloudera quickstart

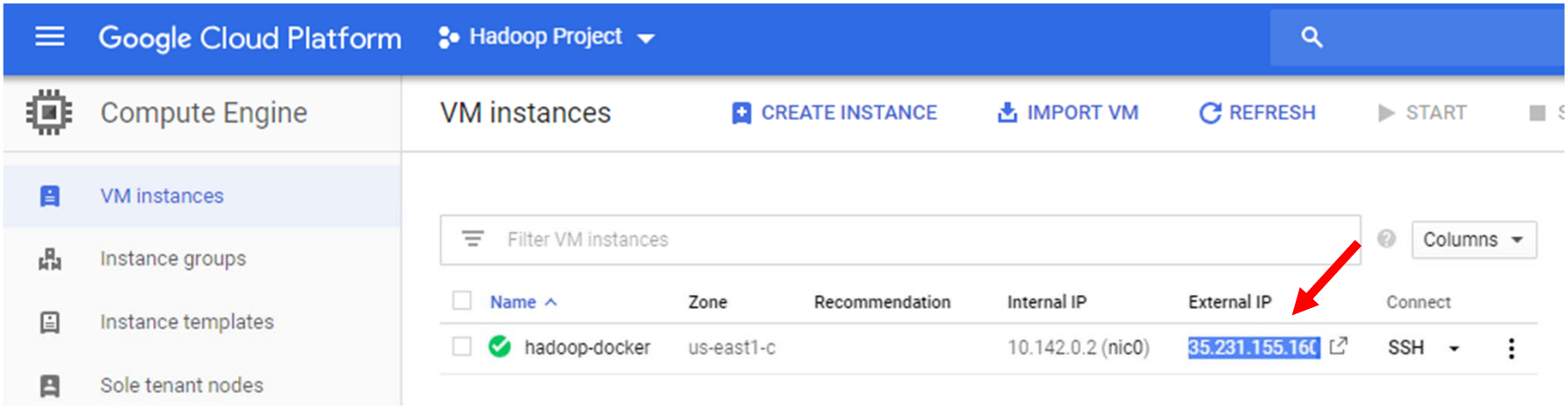
- SYNTAX: docker run --hostname=quickstart.cloudera --privileged=true -t -i [OPTIONS] [IMAGE] /usr/bin/docker-quickstart

```
# docker run -v /root:/mnt --hostname=quickstart.cloudera --privileged=true
-t -i -p 8888:8888 -p 8880:8880 -p 9092:9092 -p 2181:2181 -p 11122:11122
cloudera/quickstart /usr/bin/docker-quickstart
```

Successful running the Cloudera image

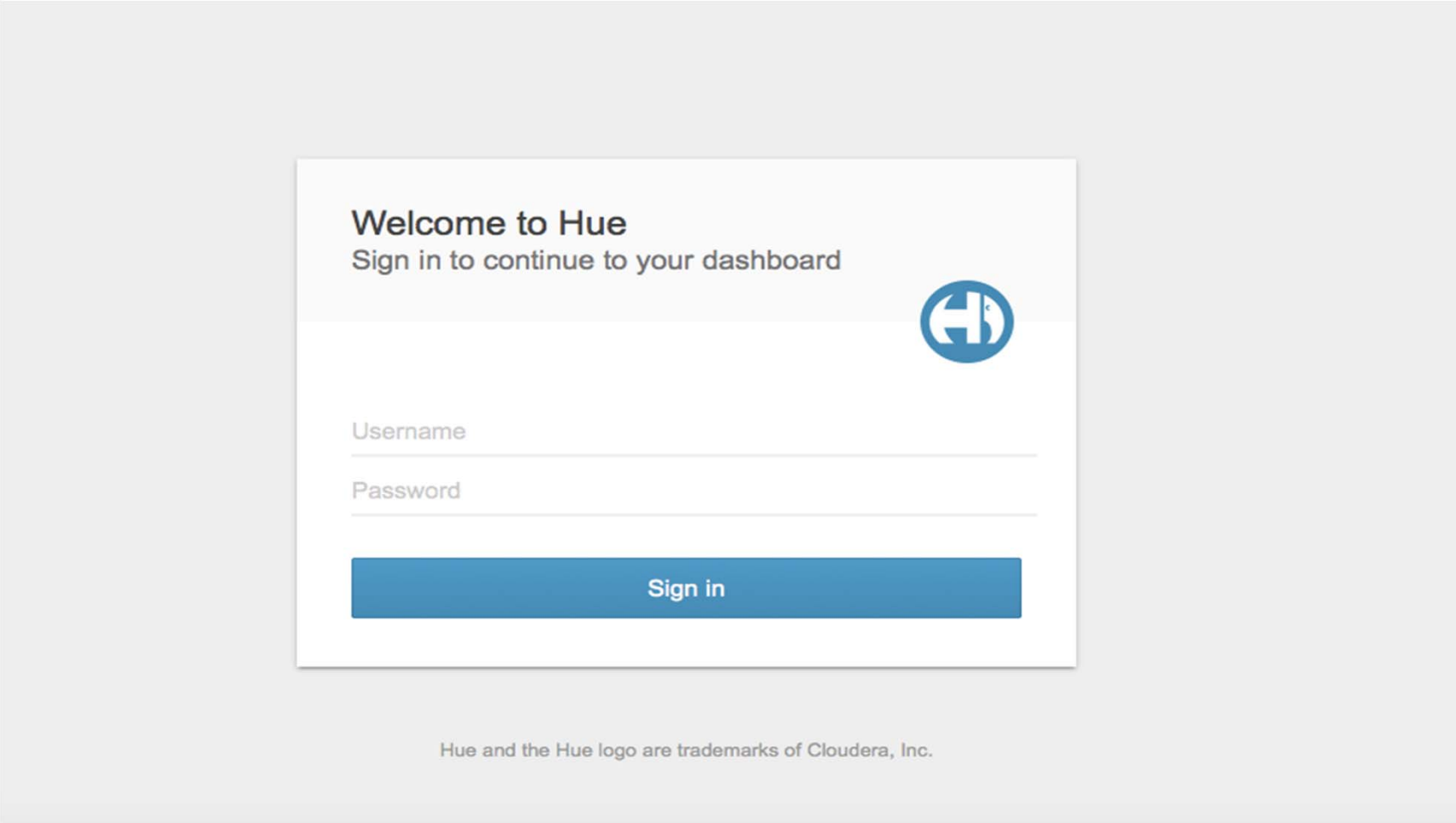
```
Using CATALINA_BASE:    /var/lib/oozie/tomcat-deployment
Using CATALINA_HOME:    /usr/lib/bigtop-tomcat
Using CATALINA_TMPDIR: /var/lib/oozie
Using JRE_HOME:         /usr/java/jdk1.7.0_67-cloudera
Using CLASSPATH:        /usr/lib/bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:     /var/run/oozie/oozie.pid
Starting Solr server daemon:                               [  OK  ]
Using CATALINA_BASE:    /var/lib/solr/tomcat-deployment
Using CATALINA_HOME:    /usr/lib/solr/../bigtop-tomcat
Using CATALINA_TMPDIR: /var/lib/solr/
Using JRE_HOME:         /usr/java/jdk1.7.0_67-cloudera
Using CLASSPATH:        /usr/lib/solr/../bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:     /var/run/solr/solr.pid
Started Impala Catalog Server (catalogd) :                 [  OK  ]
Started Impala Server (impalad):                           [  OK  ]
[root@quickstart /]#
```

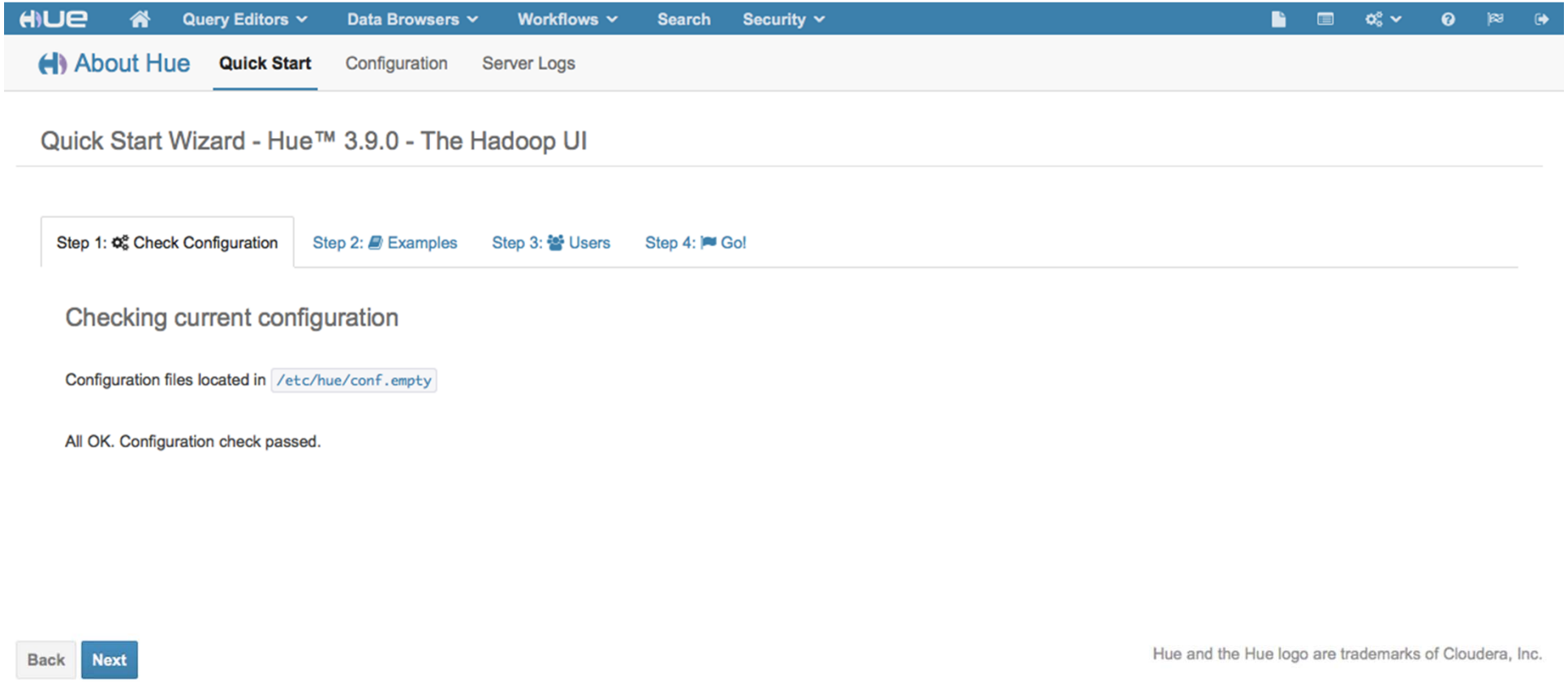
Hue: Finding the instance's external IP address



Login to Hue: **http://<<external-ip-address>>:8888**

Username: cloudera Password: cloudera




Query Editors
Data Browsers
Workflows
Search
Security
About Hue
Quick Start
Configuration
Server Logs
Quick Start Wizard - Hue™ 3.9.0 - The Hadoop UI
Step 1: Check Configuration
Step 2: Examples
Step 3: Users
Step 4: Go!
Checking current configuration
Configuration files located in /etc/hue/conf.empty
All OK. Configuration check passed.
Back
Next
Hue and the Hue logo are trademarks of Cloudera, Inc.

Step Conclusion

1. **cloud.google.com** -> Try free
2. Create Project
3. Create VM Instance
4. Add firewall rule
   - Target : **All instances in the network**
   - Source IP ranges: **0.0.0.0/0**
   - Allowed protocols and ports : Allow all
5. Connect to the instance (via SSH)
   - Update OS : **apt-get update**
   - Install Docker : **apt-get install docker.io**
   - Install Cloudera in Docker : **docker pull cloudera/quickstart:latest**
   - Run Docker : docker run -v /root:/mnt --hostname=quickstart.cloudera --privileged=true -t -i -p 8888:8888 -p 8880:8880 -p 9092:9092 -p 2181:2181 -p 11122:11122 cloudera/quickstart /usr/bin/docker-quickstart

Docker commands that are frequently used (Cont.)

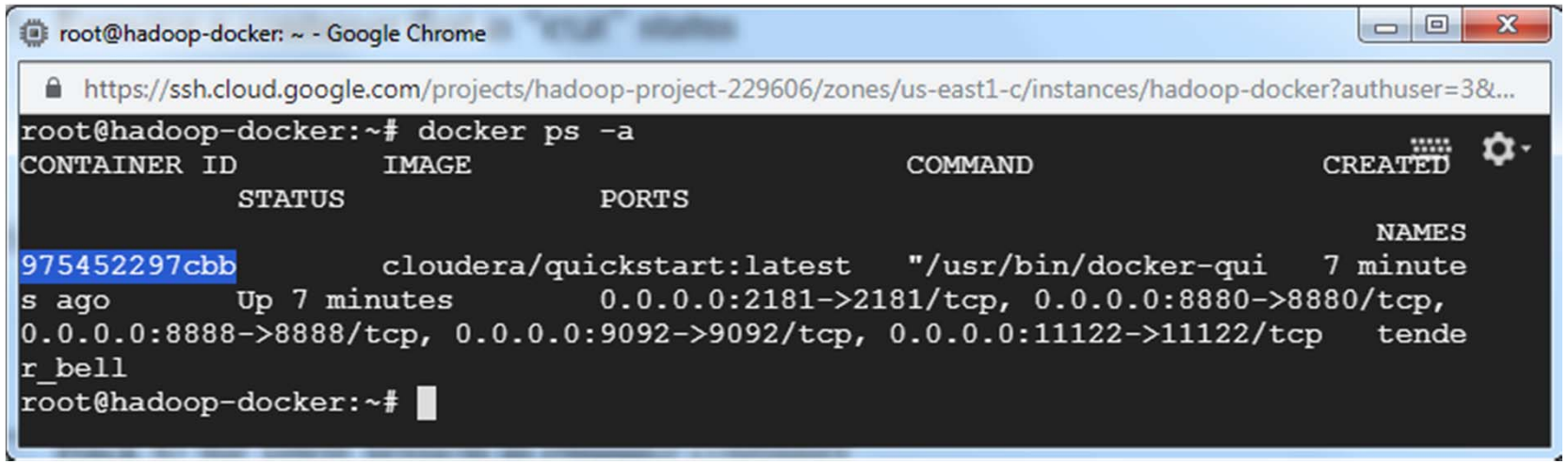
1. Exit from container (to -> root@hadoop-docker:~#)

   # exit (exit & kill the running image)

   Ctrl-P, Ctrl-Q (exit without killing the running image)

2. Check for Container ID and its status

   # docker ps -a -> *[container id]*



Docker commands that are frequently used

3. Force to stop a container

   # docker stop *[container id]*

4. Remove a container that is “exit” status

   # docker rm *[container id]*

5. Back to the running container (back to -> [root@quickstart /]#)

   # docker exec -it *[container id]* bash

6. Back to the latest session in running container

   # docker attach *[container id]*

Shell Script for

Complete stopping

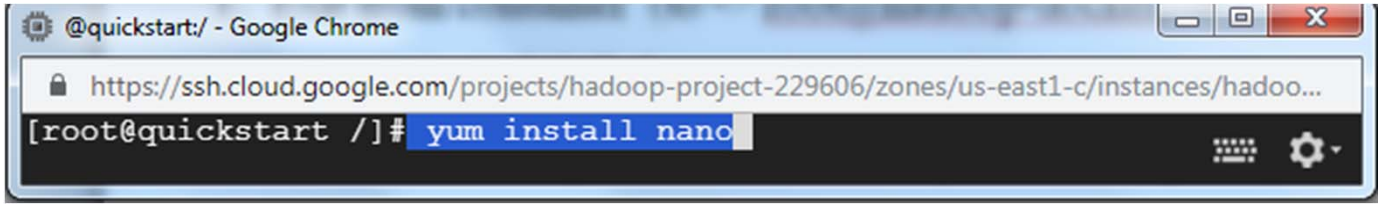
Hadoop’s all services

```
#! /usr/bin/env bash
/etc/init.d/zookeeper-server stop
/etc/init.d/hadoop-hdfs-datanode stop
/etc/init.d/hadoop-hdfs-journalnode stop
/etc/init.d/hadoop-hdfs-namenode stop
/etc/init.d/hadoop-hdfs-secondarynamenode stop
/etc/init.d/hadoop-httpfs stop
/etc/init.d/hadoop-mapreduce-historyserver stop
/etc/init.d/hadoop-yarn-nodemanager stop
/etc/init.d/hadoop-yarn-resourcemanager stop
/etc/init.d/hbase-master stop
/etc/init.d/hbase-rest stop
/etc/init.d/hbase-thrift stop
/etc/init.d/hive-metastore stop
/etc/init.d/hive-server2 stop
/etc/init.d/sqoop2-server stop
/etc/init.d/spark-history-server stop
/etc/init.d/hbase-regionserver stop
/etc/init.d/hue stop
/etc/init.d/impala-state-store stop
/etc/init.d/oozie stop
/etc/init.d/solr-server stop
/etc/init.d/impala-catalog stop
/etc/init.d/impala-server stop
```

Steps to stop all services

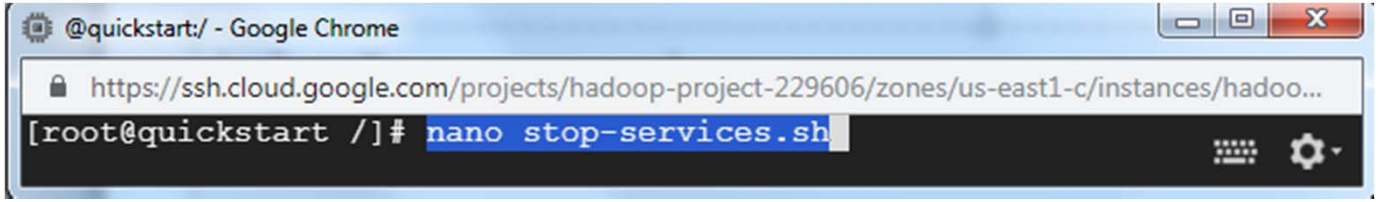
1. Install nano command

   # yum install nano



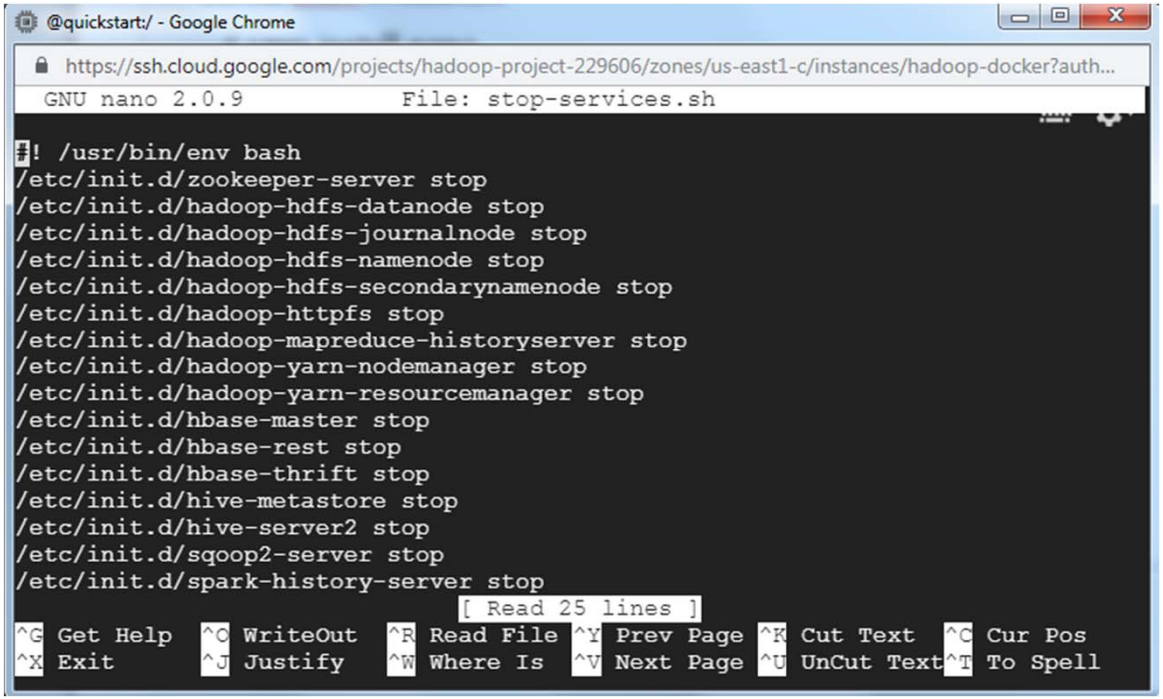
2. Create stop-services.sh

   # nano stop-services.sh



Steps to stop all services (Cont.)

3. Copy the content of the file and paste

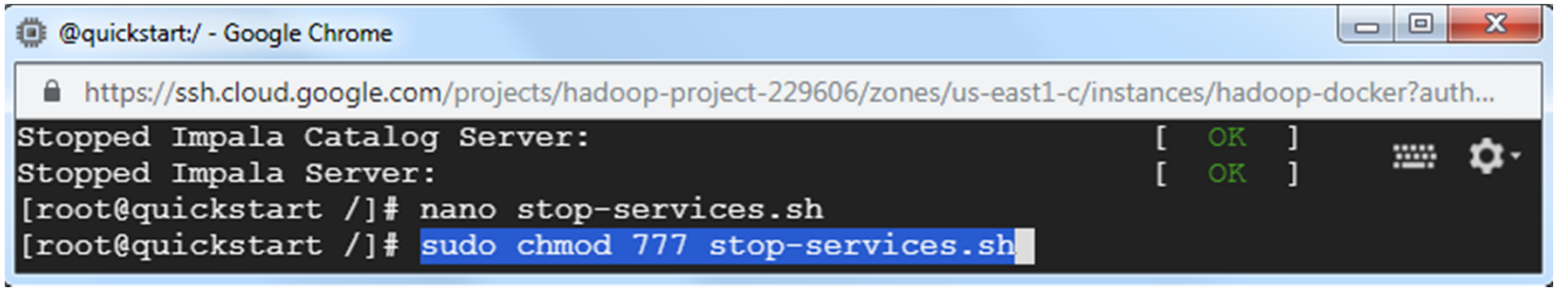


Ctrl-x, type Y and press Enter

Steps to stop all services (Cont.)

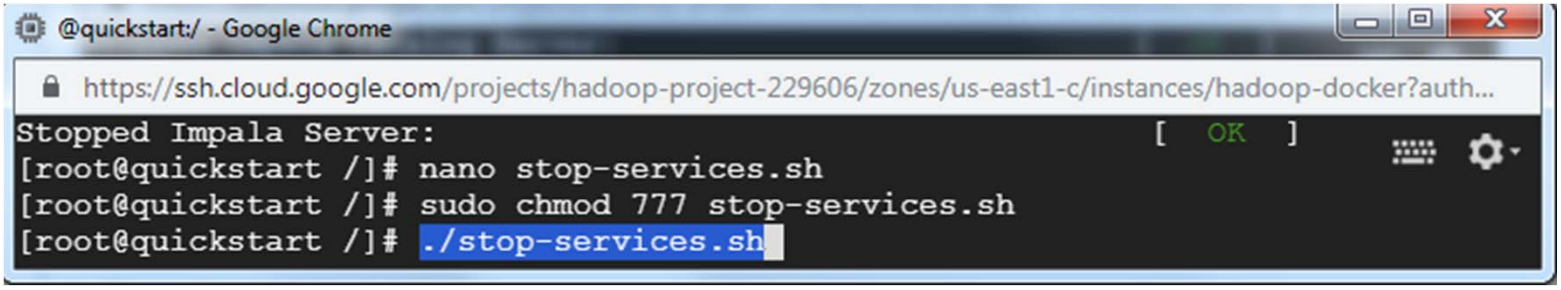
4. Give permission to the file

   # sudo chmod 777 stop-services.sh



5. Run the file

   # ./stop-services.sh



Steps to stop all services (Cont.)