Background

This section provides an overview of the EOSIO concepts covered in this lab.

EOSIO Blockchain

An EOSIO blockchain is a highly efficient, deterministic, distributed state machine that can operate in a decentralized fashion. The blockchain keeps track of transactions within a sequence of interchanged blocks. Each block cryptographically commits to the previous blocks along the same chain. It is therefore intractable to modify a transaction recorded on a given block without breaking the cryptographic checks of successive blocks. This simple fact makes blockchain transactions immutable and secure. Block production and block validation are performed by special nodes called Block Producers.

EOSIO Consensus

Block validation presents a challenge among any group of distributed nodes. A consensus model must be in place to validate such blocks in a fault tolerant way within the decentralized system. Consensus is the way for such distributed nodes and users to agree upon the current state of the blockchain. Two of the most common consensus models used in blockchains are Proof of Work (PoW) and Proof of Stake (PoS). In Proof of Stake, nodes that own the largest stake or percentage of some asset have equivalent decision power. One interesting variant is Delegated Proof-of-Stake (DPoS) in which a large number of participants or stakeholders elect a smaller number of delegates, which in turn make decisions for them. EOSIO uses Delegated Proof of Stake (DPoS) to elect the active producers who will then be authorized to produce blocks, validate them, and sign them to eventually be added to the blockchain.

EOSIO Accounts, Keys, and Permissions

An account identifies a participant in an EOSIO blockchain. A participant can be an individual or a group depending on the assigned permissions within the account. Accounts also represent the smart contract actors that push and receive actions to and from other accounts in the blockchain. Keys in EOSIO are binary strings represented in Base58 used for signing and verification of transactions, blocks, and other messages. Keys are created within a digital wallet associated with an account. Since account ownership is defined solely by the account name, the keys associated with an account can be updated without compromising security. A novel permission scheme involving accounts, permissions, and authority tables determine what accounts can do and how the actions that make a transaction are authorized. To that end, each account is assigned a hierarchical permission structure and each permission is assigned a pair of public and private keys used for signing and verification.

Create a virtual machine using the Google Cloud Console

1. In the **Navigation Menu** ☰, click **Compute Engine** > **VM instances**.
2. Click **Create Instance**.
3. For the **Name**, use my-vm-1.
4. For **Region** and **Zone**, use the defaults.
5. For **Machine Configuration**, select:

- Machine Family - **General Purpose**
- Series - **N1**

•Machine Type - **n1-standard-2**

6.For **Boot disk**, if the **Image** shown is not **Ubuntu 18.04 LTS**, click **Change** and select **Ubuntu** from the **Operating system** drop down menu and then select **Ubuntu 18.04 LTS** from **Version** drop down menu. Leave the size as default. Click on **Select**.

7.Leave the defaults for **Identity and API access** and all other fields unmodified.

8.Click **Create**.

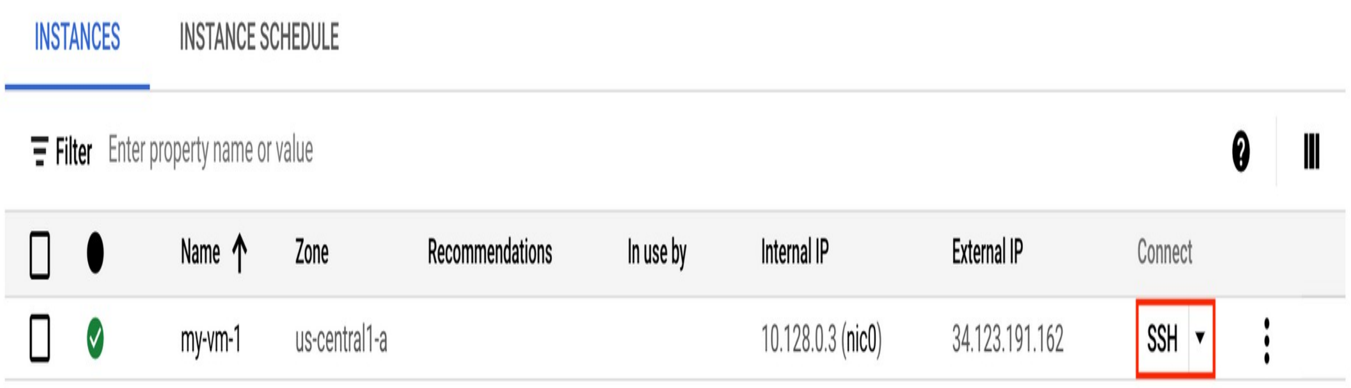
Note: The VM can take about two minutes to launch and be fully available for use.

Click Check my progress to verify the objective.

Install the EOSIO platform

1.In the **Navigation Menu** ☰, click **Compute Engine** > **VM instances**. You will see the VM instance you created.

2.Click the **SSH** button next to the my-vm-1 instance.


INSTANCES INSTANCE SCHEDULE

Filter Enter property name or value

		Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
☐	✓	my-vm-1	us-central1-a			10.128.0.3 (nic0)	34.123.191.162	SSH ▾

3.At the command prompt on my-vm-1, run the following to get the eosio binaries:

```
wget https://github.com/EOSIO/eos/releases/download/v2.0.9/eosio_2.0.9-1-ubuntu-18.04_amd64.deb
content_copy
```

4.Next, run the following to install the EOSIO platform:

```
sudo apt install ./eosio_2.0.9-1-ubuntu-18.04_amd64.deb
content_copy
```

5.Confirm nodeos is installed:

```
nodeos --version
content_copy
```

The response will be the nodeos version:

```
-v2.0.9
content_copy
```

6.Confirm cleos is installed:

```
cleos version client
content_copy
```

The response will be the cleos version:

```
-v2.0.9
content_copy
```

7.Confirm keosd is installed:

```
keosd -v
```

```
content_copy
```
The response will be the keosd version:

```
-v2.0.9

content_copy
```
Click Check my progress to verify the objective.

Run a local single node blockchain

1.In the **Navigation Menu** ☰, click **Compute Engine** > **VM instances**. You will see the VM instance you created.

2.Click the **SSH** button next to the my-vm-1 instance.

3.At the command prompt on my-vm-1, start the nodeos service daemon as a background task and launch a single node blockchain, directing the console output to a nodeos.log file:

```
nodeos -e -p eosio --plugin eosio::chain_api_plugin --plugin eosio::history_api_plugin --contracts-console >> nodeos.log 2>&1 &

content_copy
```
The response will be the PID for nodeos.

4.Verify that nodeos is running and producing blocks.

```
tail -f nodeos.log

content_copy
```
The output will look like this:

```
info  2021-03-08T02:44:11.900 nodeos    producer_plugin.cpp:2227      produce_block         ]
Produced block e0dc6324ce3c8f35... #588 @ 2021-03-08T02:44:12.000 signed by eosio [trxs: 0,
lib: 587, confirmed: 0]

content_copy
```
You will see the current block, and last irreversible block (lib) incrementing.

5.Press **Ctrl+C** to exit tail.

Click Check my progress to verify the objective.

Create wallet

1.In the **Navigation Menu** ☰, click **Compute Engine** > **VM instances**. You will see the VM instance you created.

2.Click the **SSH** button next to the my-vm-1 instance.

3.At the command prompt on my-vm-1, create a wallet, named **my_wallet**, and output the wallet password to a file named **my_wallet_password**:

```
cleos wallet create --name my_wallet --file my_wallet_password
content_copy
```

The response will show:

```
"/usr/opt/eosio/2.0.9/bin/keosd" launched
Creating wallet: my_wallet
Save password to use in the future to unlock this wallet.
Without password imported keys will not be retrievable.
saving password to wallet_password
content_copy
```

4.View the wallet password:

```
cat my_wallet_password
content_copy
```

5.To view the local wallets execute this command:

```
cleos wallet list
content_copy
```

The response will be:

```
Wallets:
[
  "my_wallet *"
]
content_copy
```
Click Check my progress to verify the objective.

Add the EOSIO system account private key to the new wallet

Every new EOSIO blockchain has a default system user called eosio. This account is used initially to set up the blockchain. It defaults to the private key, 5KQwrPbwdL6PhXujxW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3. Import this private key to your wallet to sign transactions on behalf of the eosio user.

1.In the **Navigation Menu** ☰, click **Compute Engine** > **VM instances**. You will see the VM instance you created.

2.Click the **SSH** button next to the my-vm-1 instance.

3.Use the cleos wallet open command to open the **my_wallet** wallet:

```
cleos wallet open --name my_wallet
content_copy
```
The response will be:

```
Opened: my_wallet
content_copy
```
4.Copy the wallet password from the **my_wallet_password** file.

5.Use the cleos wallet unlock command to unlock the **my_wallet** wallet, add the wallet password in to YOUR_PASSWORD:

```
cleos wallet unlock --name my_wallet --password YOUR_PASSWORD
content_copy
```

The response will be:

```
Unlocked: my_wallet
content_copy
```

6.Use the cleos wallet import command to import the EOSIO private key, 5KQwrPbwdL6PhXujxW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3, to the **my_wallet** wallet:

```
cleos wallet import --name my_wallet --private-key
5KQwrPbwdL6PhXujxW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3
content_copy
```

The response will be:

```
imported private key for: EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
content_copy
```

Click Check my progress to verify the objective.

Install the EOSIO Contract Development Toolkit (CDT)

1.In the **Navigation Menu** ☰, click **Compute Engine** > **VM instances**. You will see the VM instance you created.

2.Click the **SSH** button next to the my-vm-1 instance.

3.At the command prompt on my-vm-1, run the following to get the eosio binaries:

```
wget https://github.com/eosio/eosio.cdt/releases/download/v1.7.0/eosio.cdt_1.7.0-1-ubuntu-
18.04_amd64.deb
content_copy
```

4.At the command prompt on my-vm-1, install the CDT:

```
sudo apt install ./eosio.cdt_1.7.0-1-ubuntu-18.04_amd64.deb
content_copy
```

5.Confirm the CDT is installed:

```
eosio-cpp --version
content_copy
```

The response will be the CDT version:

```
eosio-cpp version 1.7.0
content_copy
```

Click Check my progress to verify the objective.

Create a blockchain account

1.In the **Navigation Menu** ☰, click **Compute Engine** > **VM instances**. You will see the VM instance you created.

2.Click the **SSH** button next to the my-vm-1 instance.

3.Use the cleos wallet open command to open the **my_wallet** wallet:

```
cleos wallet open --name my_wallet
content_copy
```

The response will be:

```
Opened: my_wallet
content_copy
```

4.Copy the wallet password from the my_wallet_password file:

```
export wallet_password=$(cat my_wallet_password)
echo $wallet_password
content_copy
```

5.Use the cleos wallet unlock command to unlock the **my_wallet** wallet, using the $wallet_password environment variable you just created:

```
cleos wallet unlock --name my_wallet --password $wallet_password

content_copy
```

The response will be:

```
Unlocked: my_wallet

content_copy
```

6.Use the cleos create key command to create a public/private key pair to use with the blockchain account:

```
cleos create key --file my_keypair1

content_copy
```

The response will be:

```
saving keys to my_keypair1

content_copy
```

7.Check the file my_keypair1:

```
cat my_keypair1

content_copy
```

The response will be the generated public and private keys and will look like:

```
Private key: 5J4drLKDPL6zHKSeMDQPNoSTsnBfkue757bR7CUxM21vJfgb9oR

Public key: EOS7puQYu36qyTbgPvJRgRoTU88BCREiqQTANTLZA44hxsdtxT9bY

content_copy
```

This shows **YOUR_PRIVATE_KEY** and **YOUR_PUBLIC_KEY**.

8.Use the cleos wallet import command to import the eosio private key created in the previous step **YOUR_PRIVATE_KEY**, to the **my_wallet** wallet:

```
cleos wallet import --name my_wallet --private-key YOUR_PRIVATE_KEY

content_copy
```

The response will be:

```
imported private key for: YOUR_PUBLIC_KEY

content_copy
```

9.Use the cleos create account command to create a blockchain account using **YOUR_PUBLIC_KEY**:

```
cleos create account eosio bob YOUR_PUBLIC_KEY

content_copy
```

The response will be:

```
executed transaction:
6a2aab0f2960c062652884e16e95c5313828d4630725bb2427f81f0d3dfe8f85  200 bytes  299 us
#         eosio <= eosio::newaccount            {"creator":"eosio","name":"bob","owner":
{"threshold":1,"keys":[{"key":"EOS7puQYu36qyTbgPvJRgRoTU88BC...
warning: transaction executed locally, but may not be confirmed by the network yet
content_copy
```
Click Check my progress to verify the objective.

Congratulations!

In this lab, you created a virtual machine (VM) instance, loaded and started a single node blockchain. You then created a wallet and loaded the default eosio private key used for development.

Next Steps / Learn More

Be sure to check out the following resources for more practice with Block.one:

•Block.one on the Google Cloud Marketplace!
•Learn more about the EOSIO platform at the developer portal.
•Get certified with Block.one training and certification.
•Training and Certification for EOSIO

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Last Tested Date: August 16, 2021
Last Updated Date: August 27, 2021