

# MyBlockChainExperts

*Advancement through blockchain*



## Libra Testnet on Google Cloud Platform

Deploying Facebook Libra testnet on Google Cloud Platform is a simple process that will take about an hour. The steps are broken into the following quick steps. Follow the workflow for deploying on Google Cloud Platform Compute Engine with a simple IAM posture for training and testing.

Note that this assumes you have a basic knowledge of Google Cloud Platform and is not a tutorial for Google Cloud Platform basics. For additional GCP and blockchain tutorials please go to [myblockchainexperts.org](http://myblockchainexperts.org) for more assistance.

- Deploy Compute Engine
- Configure Firewall
- Configure IAM
- Connect to Instance
- Download and Install git
- Install Stackdriver agents (optional for additional Monitoring and logging metrics)
- Install Libra testnet
- Create Accounts
- Mint Coins
- Validate Coins
- Transfer Coins
- Validate State
- Optional Appendix (Install Stackdriver agents for additional Monitoring and logging metrics)

# MyBlockChainExperts

*Advancement through blockchain*



## Deploy Compute Engine

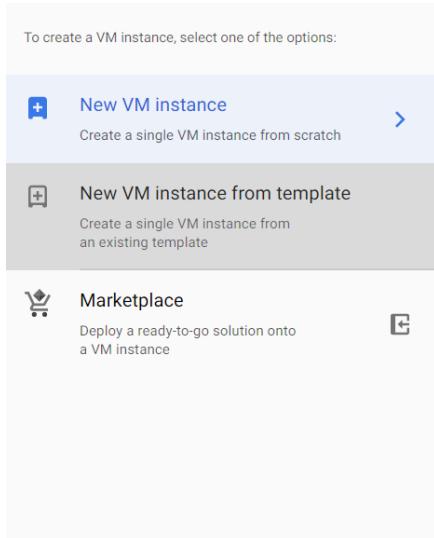
1. Deploy a Compute Engine Instance. This can be deployed with Free Credits or with the Free Tier if needed. I would recommend you use a single core and not a shared core. Also, you may want to deploy this is a separate project.

The screenshot shows the 'Create an instance' page in the Google Cloud Platform console. The URL in the address bar is https://console.cloud.google.com/compute/instancesAdd?project=cloud-developer-242517. The page has a blue header with the Google Cloud Platform logo and 'Cloud Developer'. Below the header, there's a breadcrumb trail: 'Create an instance'. The main content area is titled 'To create a VM instance, select one of the options:' and lists three options: 'New VM instance', 'New VM instance from template', and 'Marketplace'. The 'New VM instance' option is selected and expanded, showing fields for 'Name' (facebooklibratestnetvm1), 'Region' (us-central1 (Iowa)), 'Zone' (us-central1-a), 'Machine configuration' (Machine family: General-purpose), 'Generation' (First), 'Machine type' (n1-standard-1 (1 vCPU, 3.75 GB memory)), 'CPU platform and GPU' (Container: Deploy a container image to this VM instance, Learn more), and 'Boot disk' (New 10 GB standard persistent disk, Image: Debian GNU/Linux 9 (stretch), Change button). To the right of the form, there's a sidebar with resource usage information: 'You have \$6.396944 free trial credits remaining', '\$24.67 monthly estimate', 'That's about \$0.034 hourly', 'Pay for what you use: No upfront costs and per second billing', and a 'Details' link.

2. Enable Firewall for Http andHttps. Note that this is for web access. Not needed fully for initial demo. Also, you should determine your security posture and determine if you want to use VM based keys or project wide keys. This demo is using same keys for project.

# MyBlockChainExperts

*Advancement through blockchain*



To create a VM instance, select one of the options:

- New VM instance** >  
Create a single VM instance from scratch
- New VM instance from template**  
Create a single VM instance from an existing template
- Marketplace** [Edit]  
Deploy a ready-to-go solution onto a VM instance

**Identity and API access** ?

**Service account** ?  
Compute Engine default service account

**Access scopes** ?  
 Allow default access  
 Allow full access to all Cloud APIs  
 Set access for each API

**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

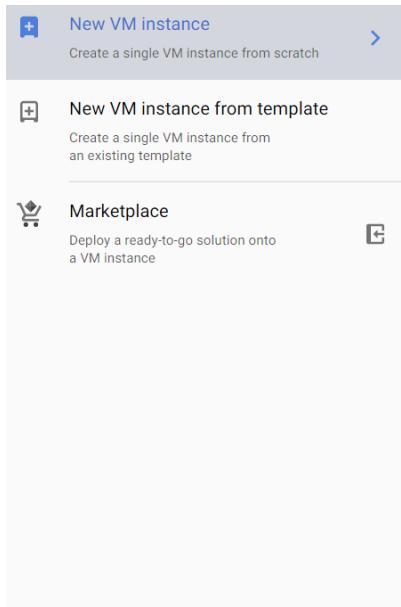
**Shielded VM** ?  
Select a shielded image to use shielded VM features.  
Turn on all settings for the most secure configuration.

Turn on Secure Boot ?  
 Turn on vTPM ?  
 Turn on Integrity Monitoring ?

**SSH Keys**  
These keys allow access only to this instance, unlike project-wide SSH keys [Learn more](#)

3. Deploy instance by selecting Create.

## Configure Firewall



**New VM instance** >  
Create a single VM instance from scratch

**New VM instance from template**  
Create a single VM instance from an existing template

**Marketplace** [Edit]  
Deploy a ready-to-go solution onto a VM instance

New 10 GB standard persistent disk  
Image  
Debian GNU/Linux 9 (stretch) Change

**Identity and API access** ?

**Service account** ?  
Compute Engine default service account

**Access scopes** ?  
 Allow default access  
 Allow full access to all Cloud APIs  
 Set access for each API

**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**

You will be billed for this instance. [Compute Engine pricing](#) [View]

**Create** Cancel

Equivalent REST or command line

# MyBlockChainExperts

*Advancement through blockchain*



4. Confirm Instance Creation in Compute Engine Dashboard which will show instance completed as a green checkmark.

The screenshot shows the Google Cloud Platform Compute Engine VM instances dashboard. The left sidebar is titled 'Compute Engine' and includes options like 'VM instances', 'Instance groups', 'Instance templates', 'Sole tenant nodes', 'Disks', 'Snapshots', 'Images', 'TPUs', 'Committed use discounts', 'Metadata', 'Health checks', 'Zones', and 'Network endpoint groups'. The main content area is titled 'VM instances' and shows a table with one row. The table columns are 'Name', 'Zone', 'Recommendation', 'In use by', 'Internal IP', 'External IP', and 'Connect'. The single row shows 'facebooklibratestnetvm1' in the 'Name' column, 'us-central1-a' in the 'Zone' column, and a green checkmark icon in the 'Recommendation' column. The 'External IP' column shows '34.66.186.191 (nic0)'. The 'Connect' column has an 'SSH' dropdown menu. To the right of the table is a sidebar titled 'Select an instance' with tabs for 'PERMISSIONS', 'LABELS', and 'MONITORING'. A message at the bottom of the sidebar says 'Please select at least one resource.'

5. Go back to [GCP home](#) dashboard.

# MyBlockChainExperts

*Advancement through blockchain*



The screenshot shows the Google Cloud Platform dashboard for the project 'Cloud Developer'. The left sidebar lists various services: Home, Kubernetes Engine, Marketplace, Billing, APIs & Services, Support, IAM & admin, Getting started, Security, COMPUTE (App Engine, Compute Engine, Kubernetes Engine, Cloud Functions), and LOGGING (Trace). The main dashboard area displays 'Project info' (Project name: Cloud Developer, Project ID: cloud-developer-242517, Project number: 394052344807) and 'Activity' (ADD PEOPLE TO THIS PROJECT, Go to project settings). It also features several cards: 'App Engine' (Summary (count/sec) chart, note: No data is available for the selected time frame), 'Compute Engine' (CPU (%) chart), 'Google Cloud Platform status' (All services normal, Go to Cloud status dashboard), 'Billing' (Estimated charges USD \$0.00 for Jul 1 – 20, 2019, View detailed charges), 'Error Reporting' (No application errors in the last 24 hours, Go to Error Reporting), and 'News' (How to use a Chromebook if you've switched from a PC, 2 days ago).

## Configure IAM

6. Select **IAM & admin** to Create Instance IAM posture and to add Permissions for Compute Engine user.

# MyBlockChainExperts

*Advancement through blockchain*



The screenshot shows the Google Cloud Platform IAM & admin interface. On the left, a sidebar lists various IAM-related options like Identity & Organization, Organization policies, Quotas, Service accounts, Labels, Settings, Privacy & Security, Cryptographic keys, Identity-Aware Proxy, Roles, Audit Logs, and Manage resources. The main area is titled "Permissions for project 'Cloud Developer'" and contains a table of users and their roles. The table has columns for Type, Member, Name, Role, and Inheritance. A row for "jholbrook2019@myblockchainexperts.net" is selected, showing the role "Compute Admin, Compute Instance Admin (beta), Owner". Other rows include service accounts and other users.

Type	Member	Name	Role	Inheritance
Compute Engine default service account	394052344807-compute@google.com	Compute Engine default service account	Editor	
Cloud Build Service Account	394052344807@cloudbuild.gserviceaccount.com	Cloud Build Service Account		
Google APIs Service Agent	394052344807@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor	
Viewer	auditorsonsite@cloud-developer-242517.iam.gserviceaccount.com	auditorsonsite	Viewer	
App Engine default service account	cloud-developer-242517.appspot.gserviceaccount.com	App Engine default service account	Editor	
Cloud SQL Client	jholbrook2015@switchsanguru.com	Cloud SQL Client		
Compute Admin, Compute Instance Admin (beta), Owner	<input checked="" type="checkbox"/> jholbrook2019@myblockchainexperts.net	Joseph Holbrook	Compute Admin, Compute Instance Admin (beta), Owner	
Cloud Build Service Agent	pearsontest@cloud-developer-242517.iam.gserviceaccount.com	pearsontest	Cloud Build Service Agent	
Compute Engine Service Agent	service-394052344807@compute-system.iam.gserviceaccount.com	Compute Engine Service Agent	Compute Engine Service Agent	
Kubernetes Engine Service Agent	service-394052344807@container-engine-robot.iam.gserviceaccount.com	Kubernetes Engine Service Agent	Kubernetes Engine Service Agent	

7. Select ADD to add user permissions.

# MyBlockChainExperts

*Advancement through blockchain*

Add members to "Cloud Developer"

## Add members, roles to "Cloud Developer" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. [Learn more](#)

New members

jholbrook2019@myblockchainexperts.net 



Role

Compute Admin 



Full control of all Compute Engine resources.

[+ ADD ANOTHER ROLE](#)

 SAVE

 CANCEL

8. Add Members by Entering your user name and then select Role of Compute Admin. Save to continue..

(To install git you will need to have admin access. This is the simplest way for a demo)

# MyBlockChainExperts

*Advancement through blockchain*

<input checked="" type="checkbox"/>		jholbrook2019@myblockchainexperts.net	Joseph Holbrook	Compute Admin Owner	
<input type="checkbox"/>		pearsontest@cloud-developer-242517.iam.gserviceaccount.com	pearsontest	Cloud Build Service Agent	
<input type="checkbox"/>		service-394052344807@compute-system.iam.gserviceaccount.com	Compute Engine Service Agent	Compute Engine Service Agent	
<input type="checkbox"/>		service-394052344807@container-engine-robot.iam.gserviceaccount.com	Kubernetes Engine Service Agent	Kubernetes Engine Service Agent	
<input type="checkbox"/>		service-394052344807@containerregistry.iam.gserviceaccount.com	Google Container Registry Service Agent	Editor	

9. Validate you are the owner of Compute Engine

## *Connect to Instance*

**Now we need to go back to Compute Engine Dashboard.**

10. Login into the VM you created by Selecting SSH Dropdown

# MyBlockChainExperts

*Advancement through blockchain*

The screenshot shows the Google Cloud Platform Compute Engine interface. On the left, there's a sidebar with various options like VM instances, Instance groups, Instance templates, Sole tenant nodes, Disks, Snapshots, Images, TPUs, Committed use discounts, and Metadata. The main area displays a table of VM instances. One row is selected, showing details: Name (facebooklibratestnetvm1), Zone (us-central1-a), Recommendation, In use by, Internal IP (10.128.0.19), External IP (34.66.186.191), and Connect (SSH dropdown). A context menu is open over the 'facebooklibratestnetvm1' row, listing five options: Open in browser window, Open in browser window on custom port, Open in browser window using provided private SSH key, View gcloud command, and Use another SSH client.

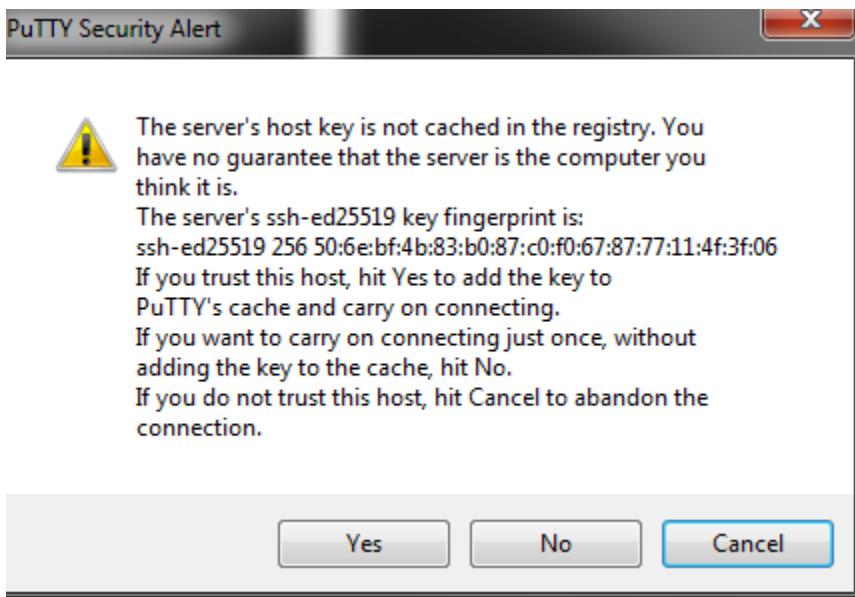
11. For the purpose of this demo I will use the SDK and will copy the gcloud command.

```
gcloud beta compute --project cloud-developer-242517 ssh --zone "us-central1-a" "facebooklibratestnetvm1"
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.
---
C:\Users\HPE Workstation\AppData\Local\Google\Cloud SDK>gcloud beta compute --project "cloud-developer-242517" ssh --zone "us-central1-a" "facebooklibratestnetvm1"
WARNING: Invalid characters in local username [HPE Workstation]. Using username corresponding to active account: [jholbrook2019]
```

Note.. You will receive a pop up like below if you have not logged into the VM before with the SDK

# MyBlockChainExperts

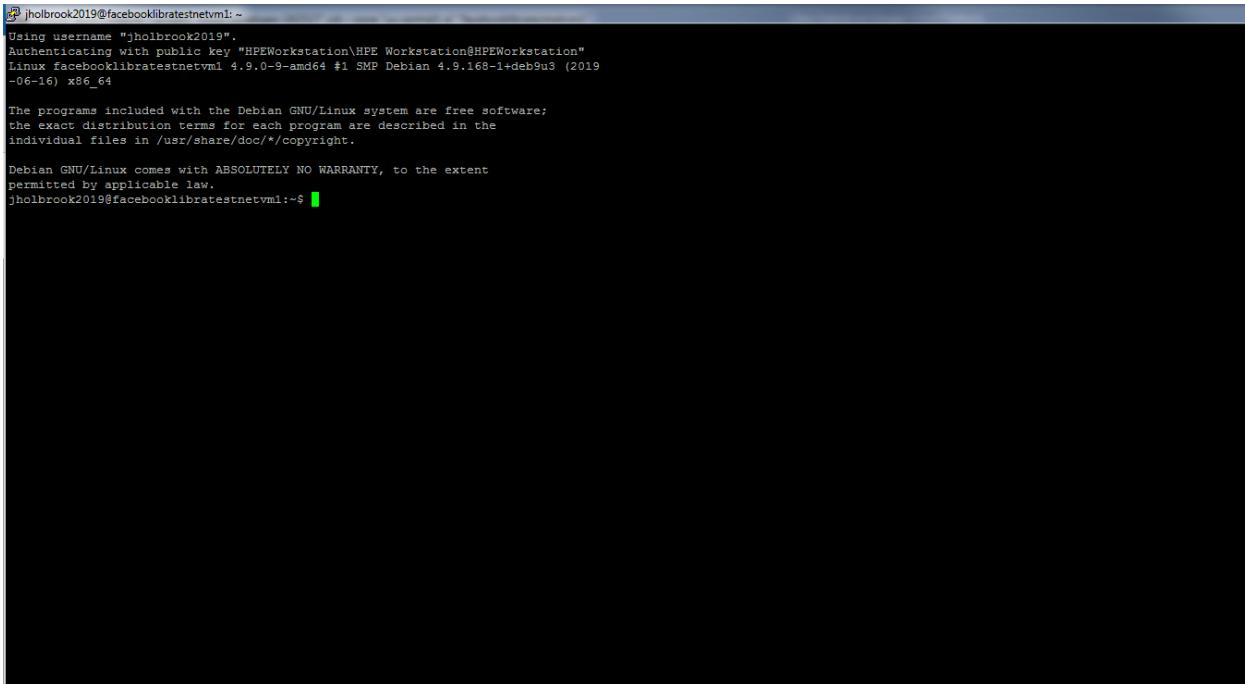
*Advancement through blockchain*



Select Yes to continue. You should see a putty or terminal show up.(depending on your config) Font will be changed to provide readability

# MyBlockChainExperts

*Advancement through blockchain*



A screenshot of a terminal window on a Linux system. The window title is "jholbrook2019@facebooklibratestnetvm1: ~". The terminal displays the following text:

```
jholbrook2019@facebooklibratestnetvm1: ~
Using username "jholbrook2019".
Authenticating with public key "HPEWorkstation\HPE Workstation@HPEWorkstation"
Linux facebooklibratestnetvm1 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u3 (2019-06-16) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jholbrook2019@facebooklibratestnetvm1:~$ █
```

We are logged in and ready to go. We now need to find git updates and download. Then we will install git since the Compute Engine image will not have git installed. (command : **which git** should be not found)

## Download and Install Git

# MyBlockChainExperts

## *Advancement through blockchain*



```
Using username "jholbrook2019".  
Authenticating with public key "HPEWorkstation\HPE Workstation@HPEWorkstation"  
Linux facebooklibratestnetvml 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u3 (2019  
-06-16) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
jholbrook2019@facebooklibratestnetvml:~$ sudo apt-get update  
Get:1 http://security.debian.org stretch/updates InRelease [94.3 kB]  
Ign:2 http://deb.debian.org/debian stretch InRelease  
Get:3 http://packages.cloud.google.com/apt cloud-sdk-stretch InRelease [6,377 B]  
Get:4 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]  
Get:5 http://packages.cloud.google.com/apt google-compute-engine-stretch-stable InRelease [3,843 B]  
Get:6 http://deb.debian.org/debian stretch-backports InRelease [91.8 kB]  
Get:7 http://deb.debian.org/debian stretch Release [118 kB]  
Hit:8 http://packages.cloud.google.com/apt google-cloud-packages-archive-keyring-stretch InRelease  
Get:9 http://deb.debian.org/debian stretch Release.gpg [2,434 B]  
Get:10 http://packages.cloud.google.com/apt cloud-sdk-stretch/main amd64 Packages [78.7 kB]  
Get:11 http://security.debian.org stretch/updates/main Sources [210 kB]  
Get:12 http://security.debian.org stretch/updates/main amd64 Packages [499 kB]  
Get:13 http://security.debian.org stretch/updates/main Translation-en [223 kB]  
Get:14 http://deb.debian.org/debian stretch-updates/main Sources.diff/Index [10.6 kB]  
Get:15 http://deb.debian.org/debian stretch-updates/main amd64 Packages.diff/Index [10.6 kB]  
Get:16 http://deb.debian.org/debian stretch-updates/main Translation-en.diff/Index [6,148 B]  
Get:17 http://deb.debian.org/debian stretch-updates/main Sources 2019-07-08-0821.07.pdiff [534 B]  
Get:18 http://deb.debian.org/debian stretch-updates/main amd64 Packages 2019-07-08-0821.07.pdiff [445 B]  
Get:17 http://deb.debian.org/debian stretch-updates/main Sources 2019-07-08-0821.07.pdiff [534 B]  
Get:18 http://deb.debian.org/debian stretch-updates/main amd64 Packages 2019-07-08-0821.07.pdiff [445 B]  
Get:19 http://deb.debian.org/debian stretch-updates/main Translation-en 2019-07-08-0821.07.pdiff [196 B]  
Get:19 http://deb.debian.org/debian stretch-updates/main Translation-en 2019-07-08-0821.07.pdiff [196 B]  
Get:20 http://packages.cloud.google.com/apt google-compute-engine-stretch-stable/main amd64 Packages [1,243 B]  
Get:21 http://deb.debian.org/debian stretch-backports/main Sources.diff/Index [27.8 kB]  
Get:22 http://deb.debian.org/debian stretch-backports/main amd64 Packages.diff/Index [27.8 kB]
```

12.Find packages for git by running command : sudo apt-get update

```
Get:55 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2019-07-16-1418.26.pdiff [363 B]  
Get:56 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2019-07-17-2020.08.pdiff [19.3 kB]  
Get:57 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2019-07-19-1411.35.pdiff [478 B]  
Get:58 http://deb.debian.org/debian stretch-backports/main Translation-en 2019-06-25-1418.21.pdiff [1,250 B]  
Get:59 http://deb.debian.org/debian stretch-backports/main Translation-en 2019-07-07-2020.35.pdiff [33 B]  
Get:60 http://deb.debian.org/debian stretch-backports/main Translation-en 2019-07-10-1415.27.pdiff [415 B]  
Get:61 http://deb.debian.org/debian stretch-backports/main Translation-en 2019-07-15-1418.25.pdiff [340 B]  
Get:57 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2019-07-19-1411.35.pdiff [478 B]  
Get:62 http://deb.debian.org/debian stretch-backports/main Translation-en 2019-07-19-1411.35.pdiff [367 B]  
Get:62 http://deb.debian.org/debian stretch-backports/main Translation-en 2019-07-19-1411.35.pdiff [367 B]  
Fetched 1,602 kB in 1s (1,243 kB/s)  
Reading package lists... Done
```

Package will complete and should reflect Done.

sudo apt-get install git

# MyBlockChainExperts



## Advancement through blockchain

```
jholbrook2019@facebooklibratestnetvm1:~$ sudo apt-get update
Hit:1 http://security.debian.org stretch/updates InRelease
Ign:2 http://deb.debian.org/debian stretch InRelease
Hit:3 http://deb.debian.org/debian stretch-updates InRelease
Hit:4 http://packages.cloud.google.com/apt cloud-sdk-stretch InRelease
Hit:5 http://deb.debian.org/debian stretch-backports InRelease
Hit:6 http://deb.debian.org/debian stretch Release
Hit:7 http://packages.cloud.google.com/apt google-compute-engine-stretch-stable InRelease
Hit:8 http://packages.cloud.google.com/apt google-cloud-packages-archive-keyring-stretch InRelease
Reading package lists... Done
jholbrook2019@facebooklibratestnetvm1:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man less libcurl3-gnutls liberror-perl libperl5.24 patch perl perl-modules-5.24 rename rsync
Suggested packages:
  git-daemon-run | git-daemon-svnserv git-doc git-email git-gui gitk gitweb git-arch git-cvs
  git-mediawiki git-svn ed diffutils-doc perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make
The following NEW packages will be installed:
  git git-man less libcurl3-gnutls liberror-perl libperl5.24 patch perl perl-modules-5.24 rename rsync
0 upgraded, 11 newly installed, 0 to remove and 12 not upgraded.
Need to get 13.0 MB of archives.
After this operation, 73.2 MB of additional disk space will be used.
Do you want to continue? [y/n] y
```

13. Install git.. 1. Run command: sudo apt-get install git and 2. also select Y which will run the install of git

```
Unpacking rsync (3.1.2-1+deb9u2) ...
Setting up perl-modules-5.24 (5.24.1-3+deb9u5) ...
Setting up libperl5.24:amd64 (5.24.1-3+deb9u5) ...
Setting up git-man (1:2.11.0-3+deb9u4) ...
Setting up less (481-2.1) ...
Processing triggers for mime-support (3.60) ...
Setting up libcurl3-gnutls:amd64 (7.52.1-5+deb9u9) ...
Setting up rsync (3.1.2-1+deb9u2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/rsync.service → /lib/systemd/system/rsync.service.
Setting up perl (5.24.1-3+deb9u5) ...
update-alternatives: using /usr/bin/prename to provide /usr/bin/rename (rename) in auto mode
Setting up patch (2.7.5-1+deb9u1) ...
Processing triggers for libc-bin (2.24-11+deb9u4) ...
Processing triggers for systemd (232-25+deb9u11) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up Liberror-perl (0.17024-1) ...
Setting up rename (0.20-4) ...
update-alternatives: using /usr/bin/file-rename to provide /usr/bin/rename (rename) in auto mode
Setting up git (1:2.11.0-3+deb9u4) ...
jholbrook2019@facebooklibratestnetvm1:~$
```

Download and install complete. Now lets validate our install of git

```
jholbrook2019@facebooklibratestnetvm1:~$ which git
/usr/bin/git
jholbrook2019@facebooklibratestnetvm1:~$
```

14. Validate git by running command: `which git`

# MyBlockChainExperts

*Advancement through blockchain*

Now we need to initialize

```
jholbrook2019@facebooklibratestnetvm1:~$ git init  
Initialized empty Git repository in /home/jholbrook2019/.git/  
jholbrook2019@facebooklibratestnetvm1:~$ █
```

15. Initialize git by running command: `git init`

Now we need to add to directory.

```
Initialized empty Git repository in /home/jholbrook2019/.git/  
jholbrook2019@facebooklibratestnetvm1:~$ git add .█
```

16. Add to directory tree. Run command: `git add .`

---

**Note.** It generally a good practice to commit, add origins and push to a master repository in a real development environment you git repositories. For time purposes and not being totally needed I will skip for this demo.

---

Clone the Facebook Libra Repository. We have our Compute Engine Instance ready and now can following the Libra page instructions are correct and would work. Following this link if you choose.  
<https://developers.libra.org/docs/my-first-transaction#the-transfer-command>

```
jholbrook2019@facebooklibratestnetvm1:~$ git clone https://github.com/libra/libra.git  
Cloning into 'libra'...  
remote: Enumerating objects: 966, done.  
remote: Counting objects: 100% (966/966), done.  
remote: Compressing objects: 100% (632/632), done.  
remote: Total 5433 (delta 409), reused 662 (delta 318), pack-reused 4467  
Receiving objects: 100% (5433/5433), 4.16 MiB | 0 bytes/s, done.  
Resolving deltas: 100% (2750/2750), done.  
jholbrook2019@facebooklibratestnetvm1:~$ █
```

# MyBlockChainExperts

*Advancement through blockchain*



17. Clone the Facebook Libra repository by running command: `git clone https://github.com/libra/libra.git`

## Setup Libra testnet

```
jholbrook2019@facebooklibratestnetvm1:~/libra$ ./scripts/dev_setup.sh
Welcome to Libra!

This script will download and install the necessary dependencies needed to
build Libra Core. This includes:
    * Rust (and the necessary components, e.g. rust-fmt, clippy)
    * CMake, protobuf, go (for building protobuf)

If you'd prefer to install these dependencies yourself, please exit this script
now with Ctrl-C.

Proceed with installing necessary dependencies? (y/N) > y
```

18. Setup Libra Core blockchain testnet by changing directory to libra and running the setup script.

Select Y

```
cd libra
./scripts/dev_setup.sh
```

Kick off by selecting Yes

```
jholbrook2019@facebooklibratestnetvm1:~/libra$ ./scripts/dev_setup.sh
Welcome to Libra!

This script will download and install the necessary dependencies needed to
build Libra Core. This includes:
    * Rust (and the necessary components, e.g. rust-fmt, clippy)
    * CMake, protobuf, go (for building protobuf)

If you'd prefer to install these dependencies yourself, please exit this script
now with Ctrl-C.

Proceed with installing necessary dependencies? (y/N) > y
```

19. Kick off by selecting Yes

# MyBlockChainExperts

*Advancement through blockchain*

The process will continue for a minute until you see the next screen.

```
(Reading database ... 53200 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-21+deb9u1_amd64.deb ...
Unpacking unzip (6.0-21+deb9u1) ...
Processing triggers for mime-support (3.60) ...
Setting up unzip (6.0-21+deb9u1) ...
Processing triggers for man-db (2.7.6.1-2) ...
  % Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100  164     0  164     0      0  1091      0  --::-- --::-- --::-- 1093
100  619     0  619     0      0  2245      0  --::-- --::-- --::--  9378
100 1513k  100 1513k     0      0 2296k      0  --::-- --::-- --::-- 2296k
Archive: protoc-3.8.0-linux-x86_64.zip
  inflating: /usr/local/bin/protoc
Archive: protoc-3.8.0-linux-x86_64.zip
  creating: /usr/local/include/google/
  creating: /usr/local/include/google/protobuf/
  inflating: /usr/local/include/google/protobuf/api.proto
  inflating: /usr/local/include/google/protobuf/type.proto
  inflating: /usr/local/include/google/protobuf/wrappers.proto
  creating: /usr/local/include/google/protobuf/compiler/
  inflating: /usr/local/include/google/protobuf/compiler/plugin.proto
  inflating: /usr/local/include/google/protobuf/timestamp.proto
  inflating: /usr/local/include/google/protobuf/duration.proto
  inflating: /usr/local/include/google/protobuf/descriptor.proto
  inflating: /usr/local/include/google/protobuf/any.proto
  inflating: /usr/local/include/google/protobuf/source_context.proto
  inflating: /usr/local/include/google/protobuf/field_mask.proto
  inflating: /usr/local/include/google/protobuf/struct.proto
  inflating: /usr/local/include/google/protobuf/empty.proto
protoc is installed to /usr/local/bin/
Finished installing all dependencies.

You should now be able to build the project by running:
  source /home/jholbrook2019/.cargo/env
  cargo build
jholbrook2019@facebooklibratestnetvml:~/libra$
```

We are done with dependencies.

We now need to run the setup script.

# MyBlockChainExperts

## Advancement through blockchain

```
jholbrook2019@facebooklibratestnetvm1:~/libra$ ./scripts/cli/start_cli_testnet.sh
Building and running client in debug mode.
  Updating crates.io index
  Updating git repository `https://github.com/pingcap/rust-rocksdb.git`
  Updating git repository `https://github.com/alexcrichton/bzip2-rs.git`
  Updating git repository `https://github.com/busyjay/lz4-rs.git`
  Updating git repository `https://github.com/busyjay/rust-snappy.git`
  Updating git repository `https://github.com/gyscos/zstd-rs.git`
Downloaded hex v0.3.2
Downloaded bincode v1.1.4
Downloaded num-traits v0.2.8
Downloaded hyper v0.12.33
Downloaded failure v0.1.5
Downloaded chrono v0.4.7
Downloaded rand v0.6.5
Downloaded slog v2.5.0
Downloaded digest v0.8.1
Downloaded proptest-derive v0.1.2
Downloaded rust_decimal v1.0.1
Downloaded quote v0.6.13
Downloaded proc-macro2 v0.4.30
Downloaded protobuf-codegen v2.7.0
Downloaded rayon v1.1.0
Downloaded ttl_cache v0.4.2
Downloaded hmac v0.7.1
Downloaded typed-arena v1.4.1
```

20. Setup the Libra Blockchain testnet by running the script.. `./scripts/cli/start_cli_testnet.sh`

Note: This is the longest part of the process. This could take up to 10 minutes since it will compile and build our local testnet.

```
Compiling vm_validator v0.1.0 (/home/jholbrook2019/libra/vm_validator)
Compiling vm_genesis v0.1.0 (/home/jholbrook2019/libra/language/vm/vm_genesis)
Compiling client v0.1.0 (/home/jholbrook2019/libra/client)
  Finished dev [unoptimized + debuginfo] target(s) in 18m 21s
    Running `target/debug/client --host ac.testnet.libra.org --port 8000 -s ./scripts/cli/trusted_peers.config
.toml`
Connected to validator at: ac.testnet.libra.org:8000
usage: <command> <args>

Use the following commands:

account | a          Account operations
query | q            Query operations
transfer | transferb | t | tb
  <sender_account_address>|<sender_account_ref_id> <receiver_account_address>|<receiver_account_ref_id> <
number_of_coins> [gas_unit_price_in_micro_libras (default=0)] [max_gas_amount_in_micro_libras (default 100000)]
  Suffix 'b' is for blocking.
  Transfer coins (in libra) from account to another.
dev               Local move development
help | h           Prints this help
quit | q!          Exit this client

Please, input commands:
libra%
```

Your install and script run has completed

# MyBlockChainExperts

*Advancement through blockchain*

Lets create accounts

## Create Accounts

We will create two accounts. I am using Joe and Frida as an example.

```
libra% account create joe
>> Creating/retrieving next account from wallet
Created/retrieved account #0 address 8b58d1ef97b6c73db9cb926da82cc44916f6329dfd09f47d85836effc72a4421
libra%
```

21. Create first account by typing command : **account create joe**

```
libra% account create frida
>> Creating/retrieving next account from wallet
Created/retrieved account #1 address 29cd1b9be594d771584630e83451a3ac4bcfd0c2000cc15bff5a0af4d7dd6c95
libra%
```

22. Create second account by typing command : **account create frida**

Now lets validate the accounts.

```
libra% account list
User account index: 0, address: 8b58d1ef97b6c73db9cb926da82cc44916f6329dfd09f47d85836effc72a4421, sequence number: 0, status: Local
User account index: 1, address: 29cd1b9be594d771584630e83451a3ac4bcfd0c2000cc15bff5a0af4d7dd6c95, sequence number: 0, status: Local
libra%
```

23. List account by running command: **account list**

Now we need to mint coins for these users.

Joe is user 0

Frida is user 1

For example:

- 1 is the index of Joes account.
- 1000 is the amount of Libra to be added to account.
- *A successful account mint command will also create Joe's account on the blockchain.* Another way to create Bob's account on the blockchain is to transfer money from Frida's account to Joe's account.

# MyBlockChainExperts

*Advancement through blockchain*



```
libra% account mint 0 1000
>> Minting coins
Mint request submitted
libra% account mint 1 1000
>> Minting coins
Mint request submitted
libra%
```

24. Mint coins. Run command for each user **account mint 0 1000** and then again **account mint 1 1000**

Now we need to validate our accounts are created.

```
libra% query balance 0
Balance is: 1000.000000
libra% query balance 1
Balance is: 1000.000000
libra%
```

25. Validate accounts by running commands **query balance 0** and **query balance 1**

Next lets validate we were at in sequence

```
>> Getting current sequence number
Sequence number is: 0
libra%
```

26. Validate sequence by running command

```
libra% query sequence 0
>> Getting current sequence number
Sequence number is: 0
libra%
```

We now will validate the account state.

# MyBlockChainExperts



*Advancement through blockchain*

```
libra% query account_state 0

>> Getting latest account state
Latest account state is:
Account: 8b58d1ef97b6c73db9cb926da82cc44916f6329dfd09f47d85836effc72a4421
State: Some(
    AccountStateBlob {
        Raw: 0x01000000210000001217da6c6b3e19f1825cfb2676daecce3bf3de03cf26647c78df00b371b25cc9744000000200000008
b58d1ef97b6c73db9cb926da82cc44916f6329dfd09f47d85836effc72a442100b211841700000000000000000000000000000000000000000000
0000000000000000
    Decoded: AccountResource {
        balance: 101000000000,
        sequence_number: 0,
        authentication_key: 0x8b58d1ef97b6c73db9cb926da82cc44916f6329dfd09f47d85836effc72a4421,
        sent_events_count: 0,
        received_events_count: 0,
    },
)
Blockchain Version: 171934
libra%
```

27. Validate the account state by running command: `query account_state 0`

This concludes the demo. The appendix provides instruction on how to setup Stackdriver Monitoring and Logging

There are also other activities you can perform on the Libra testnet page.

<https://developers.libra.org/docs/my-first-transaction#the-transfer-command>

**END OF DEMO**

# MyBlockChainExperts

*Advancement through blockchain*



## Appendix - Stackdriver

---

### Install Stackdriver Monitoring and Logging Agents (Optional)

*This is optional if you want the Stackdriver agents to give you deeper insight into your infrastructure and application data. Install both agents onto all the infrastructure you want deeper insight into.*

**Skip if you're not planning on monitoring or logging significantly.**

**Refer to appendix if you're looking for this insight.**

---

Monitoring can access some instance metrics without the Monitoring agent, including CPU utilization, some disk traffic metrics, network traffic, and uptime information. Monitoring uses the Monitoring agent to access additional system resources and application services in virtual machine (VM) instances. If you want these additional capabilities, you should install the Monitoring agent.

# MyBlockChainExperts

*Advancement through blockchain*



Stackdriver is under “Stackdriver” in the Cloud Console

The screenshot shows the Google Cloud Platform dashboard for the project "Cloud Developer". The left sidebar is titled "Stackdriver" and includes sections for Monitoring, Debug, Trace, Logging, Error Reporting, and Profiler. The main dashboard displays "Project info" (Project name: Cloud Developer, Project ID: cloud-developer-242517, Project number: 394052344807), "App Engine" metrics (Summary count/sec), "Compute Engine" metrics (CPU %), and "Google Cloud Platform status" (All services normal). There are also sections for Billing, Error Reporting, and News.

Select Monitoring from the Stackdriver menu. This will bring you to following interface. (Note if your project has not been imported into Stackdriver then you will be prompted to confirm)

The screenshot shows the Stackdriver Monitoring Overview page. The left sidebar lists Monitoring Overview, Resources, Alerting, Uptime Checks, Groups, Dashboards, Debug, Trace, Logging, Error Reporting, and Profiler. The main area features a "Welcome to Stackdriver Monitoring!" banner with links to Add GCP Projects, Email Reports, and Tutorials. It also includes sections for "Install Stackdriver agents", "Create uptime checks", "Create alerting policies", and "Create custom dashboards". Below these are "Resource dashboards" (listing Cloud Pub/Sub, Cloud Storage, Instances, App Engine, Google Cloud Load Balancers, and Block Storage Volumes) and "Incidents" (showing charts for GCE VM Instance - CPU utilization, Received bytes, Sent bytes, and Disk read bytes over time intervals of 1 min and 5 min).

# MyBlockChainExperts

*Advancement through blockchain*



Install the Agent. Select Install Stackdriver Agents

The screenshot shows the Stackdriver Cloud Developer interface. On the left, there's a sidebar with various monitoring and developer tools like Monitoring Overview, Resources, Alerting, Uptime Checks, Groups, Dashboards, Debug, Trace, Logging, Error Reporting, and Profiler. The 'Agent' tab is currently selected. The main content area has two sections: 'Stackdriver agents' and 'Monitoring Agent'. The 'Stackdriver agents' section says 'Set up the Stackdriver agents' and describes how they provide deeper insight into infrastructure and application data. Below it, the 'Monitoring Agent' section explains what it is and how it collects metrics from virtual machines. It includes a command-line script for installation:

```
$ curl -sS https://dl.google.com/cloudagents/install-monitoring-agent.sh
$ sudo bash install-monitoring-agent.sh
```

The 'Logging Agent' section also describes its function and includes another installation script:

```
$ curl -sS https://dl.google.com/cloudagents/install-logging-agent.sh
$ sudo bash install-logging-agent.sh
```

Copy the commands for monitoring and run the commands in the VM CLI(You will need to SSH in or use Cloud Shell to access VM you created )

Install Stackdriver Monitoring Agents

```
jholbrook2019@facebooklibratestnetvm1:~$ curl -sS https://dl.google.com/cloudagents/install-monitoring-agent.sh
jholbrook2019@facebooklibratestnetvm1:~$ sudo bash install-monitoring-agent.sh
=====
Starting installation of stackdriver-agent
=====

Installing agent for Debian or Ubuntu.
OK
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  libltdl7 libpython2.7 libyajl2
Suggested packages:
  default-jre libhiredis0.13 libmariadbclient18 libpq5
The following NEW packages will be installed:
  libltdl7 libpython2.7 libyajl2 stackdriver-agent
0 upgraded, 4 newly installed, 0 to remove and 12 not upgraded.
Need to get 3,222 kB of archives.
After this operation, 9,986 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 libltdl7 amd64 2.4.6-2 [389 kB]
Get:2 http://packages.cloud.google.com/apt google-cloud-monitoring-stretch/main amd64 stackdriver-agent amd64 5
  .5.2-384.stretch [1,738 kB]
Get:3 http://deb.debian.org/debian stretch/main amd64 libpython2.7 amd64 2.7.13-2+deb9u3 [1,071 kB]
```

1. Run the commands to install the Stackdriver Monitoring Agents by copying

# MyBlockChainExperts

*Advancement through blockchain*



**Run Commands :** curl -sSO https://dl.google.com/cloudagents/install-monitoring-agent.sh

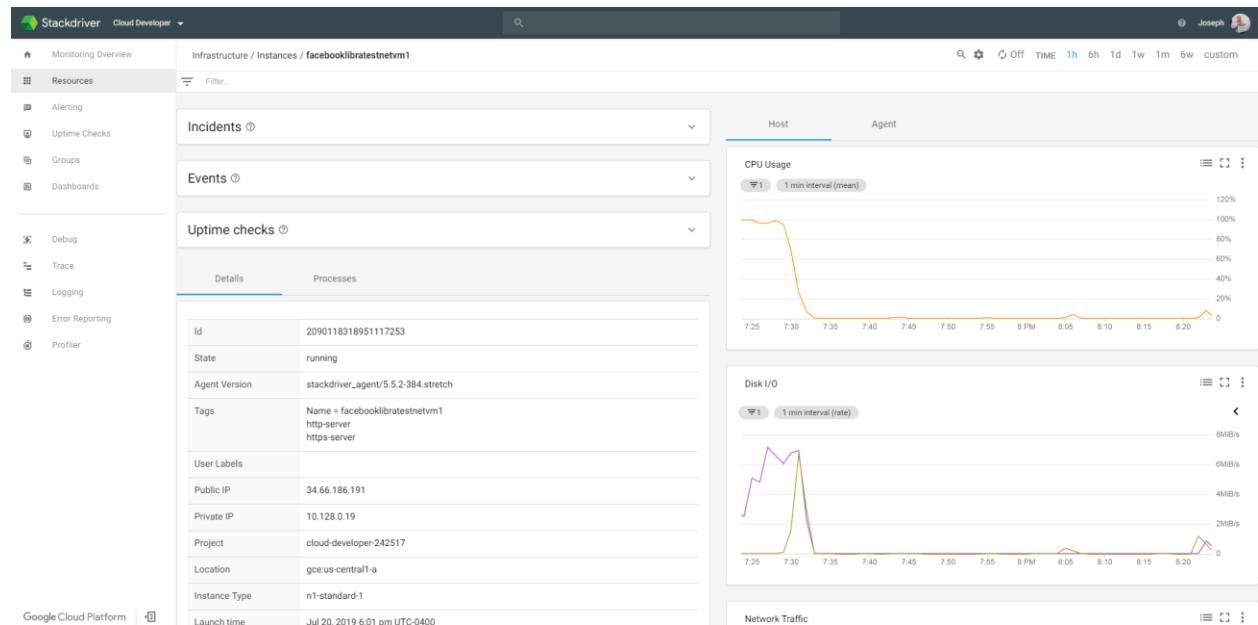
sudo bash install-monitoring-agent.sh

Repeat for Logging as well

**Run Commands:** curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh

sudo bash install-logging-agent.sh

After a few hours you VM should have activity. You can view under resources in Stackdriver Interface.



For more info on the Monitoring and Logging agents please refer to

<https://cloud.google.com/monitoring/agent/>

<https://cloud.google.com/logging/docs/agent/>