

Ethereum

Ethereum is an open-source, blockchain-based distributed computing platform.

Ethereum.org

Ethereum.org/foundation



Ethereum

Accounts in Ethereum:

- Externally owned Accounts [user accts]
- Contracts



Nodes:

- Computers that are part of the blockchain network
- Miners are nodes willing to participate in the validation and processing of transactions. Ideally, these nodes have more computing power than normal nodes.

Ethereum

Ether is the digital currency used by Ethereum.

10^{18} wei = 1 ether



Current market price of Ether?

Ethereum

- Gas
- Nonce
- Difficulty



Ethereum

Smart Contracts

A smart contract is a programmatic definition of business rules for 2 or more entities to transact with each other. It is written and deployed on a blockchain framework.

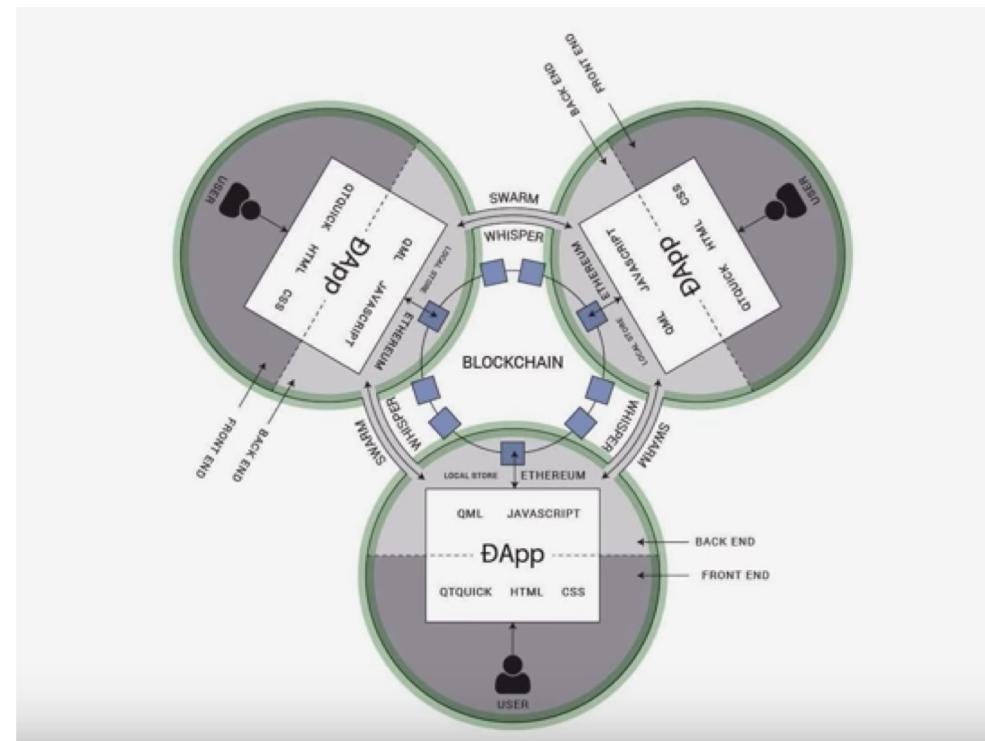


Ethereum

Decentralized App (Dapps)

As the name indicates, a Dapp is an application that has its code running on a decentralized peer-to-peer network. Traditional apps have their code on a centralized app server.

Dapp = UI + Smart Contract(s)



Validate Setup

Ensure that you have downloaded and set up:

- Go Ethereum (Geth) CLI

On linux – you can install homebrew and then use the commands:

`brew tap ethereum/ethereum`

`brew install ethereum`

Setup

- ✓ Create directory called Ethereum
- ✓ Create an environment var called ethereum_home and assign the full path of the directory you just created

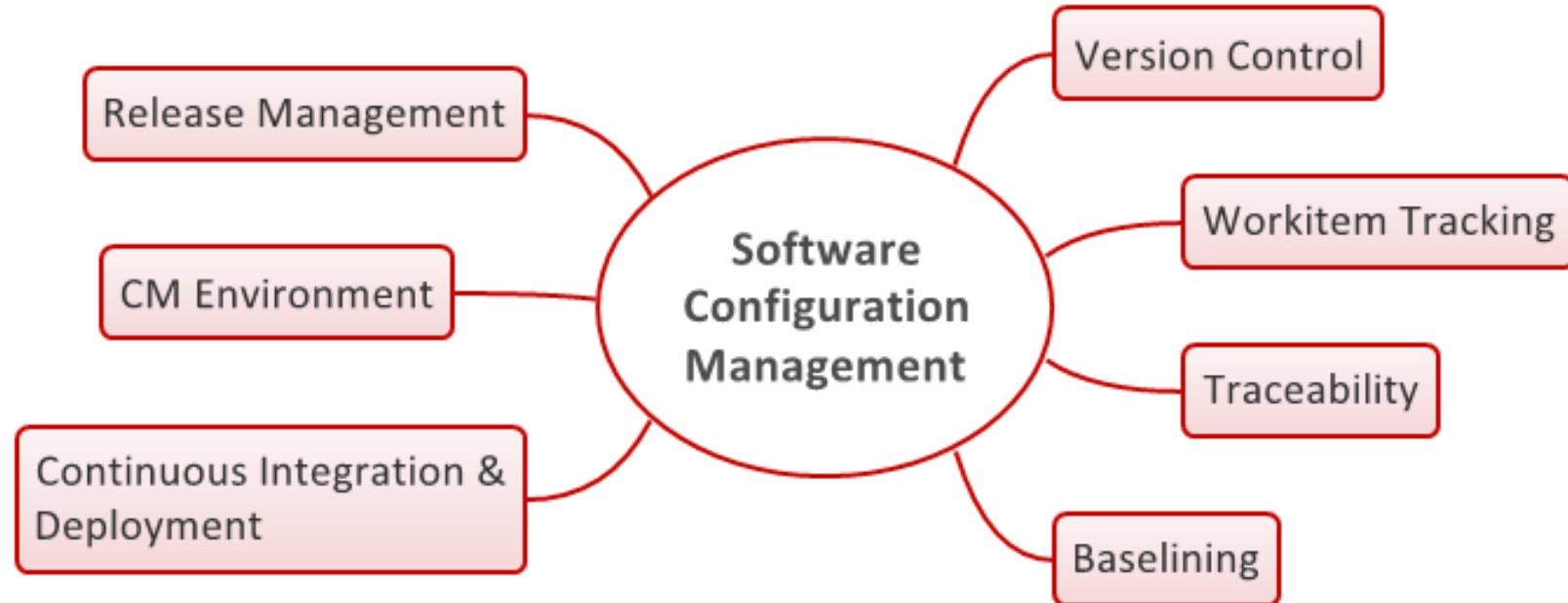
Setup

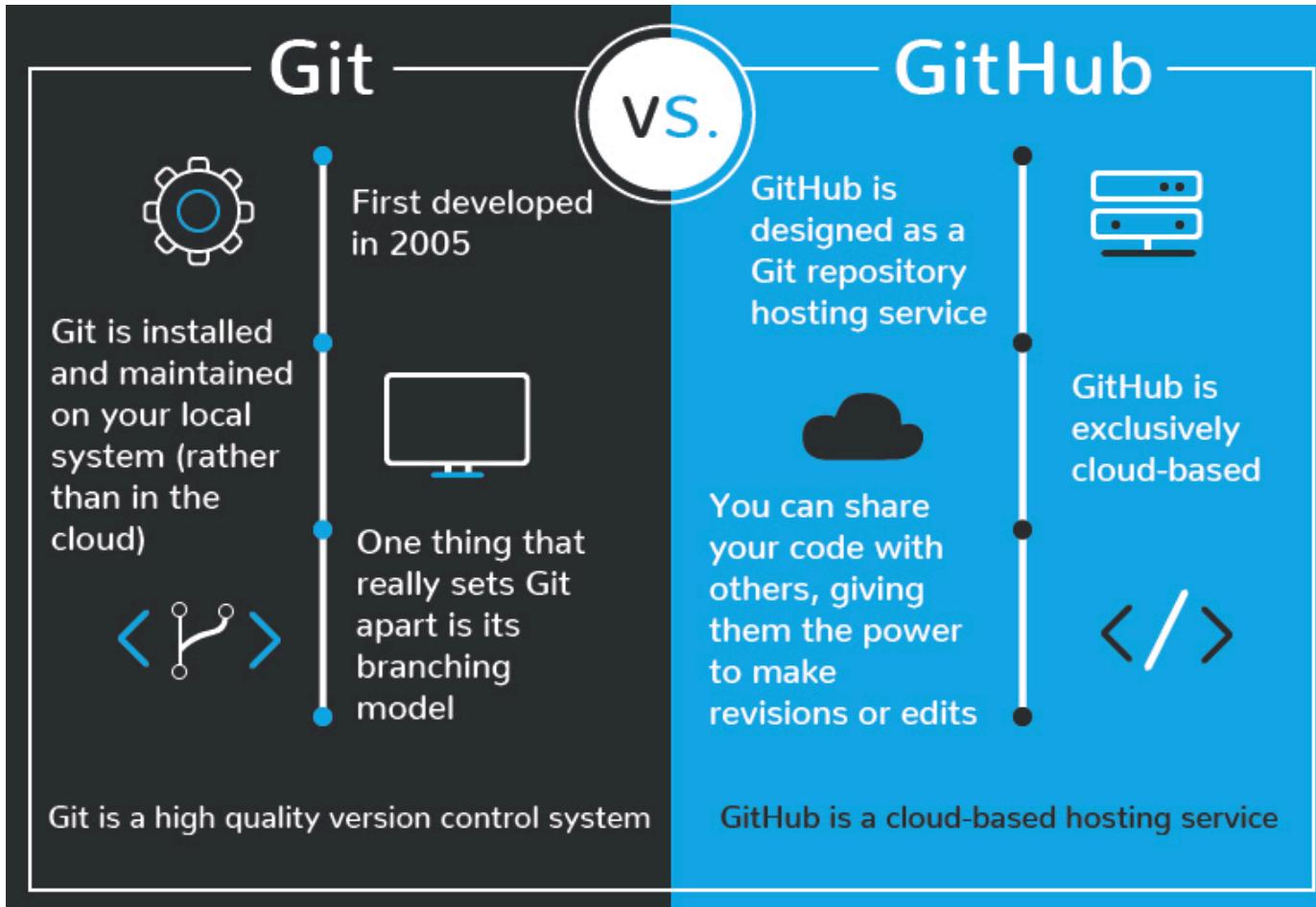
Create a file `genesis.json` in your `ethereum_home` directory

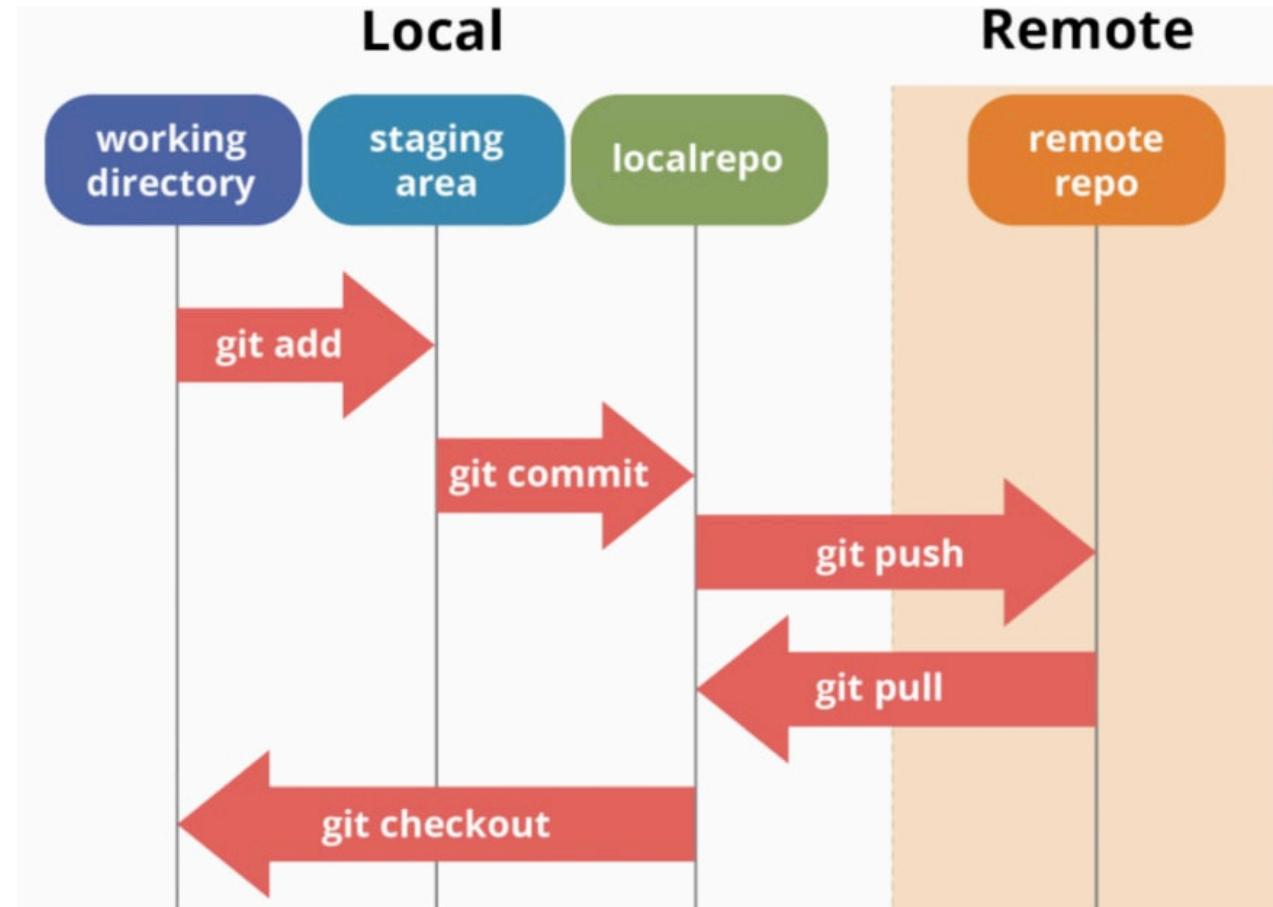
****Access source code at [github](#).**

Search for repository: nagenn/Blockchain**

Git







Download, install & configure Git

Download and Install Git:

<https://git-scm.com/downloads>

#Install git with screenshots for windows:

<http://robertovormittag.net/ebooks/git-and-github/git-for-windows-installation-screenshots/>

#On Mac - you can use homebrew

brew install git

#On Ubuntu - you can use apt-get

apt-get install git

git --version to verify installation

#Configure:

git config --global user.name "Nagen"

git config --global user.email "nagen@test.com"

#Many other config options:

<https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration>

#3 levels of config: System, Global & Local

git config --global user.name

git config --system user.name

#Check your config

git config --global --list

#Create repository:

mkdir <directory name>

cd < directory name>

git init

#OR

git init <directory name>

cd < directory name>

Creating a Git Repo

```
#Check status:  
git status  
  
#Create a new file  
touch file1.code  
#<OR> use an editor to create a simple text file in the current directory  
  
git status  
  
#Add file to staging  
git add <filename> OR '.' to add all files in the directory  
  
#Commit file:  
git commit -m "<comments>"  
  
git status  
  
#Modify file:  
echo "#Sample line of code" >> file1.code OR use an editor to add this line and save
```

Add content to your repo

#Try

```
git commit -m "Version 1.01"
```

Why does it not work?

#We forgot to add it to staging

```
git add .
```

#Now try commit but without -m "comment".

#What do you see?

#Now try

```
git commit -m "<comment>"
```

#modify the file again

#This time we will use one command for both adding to staging and committing

```
git commit -a -m "<comment>"
```

#For files that have already been added before and modified now, you can use

Commit your new content

```
#View all commits:
```

```
git log
```

```
#More concise:
```

```
git log --oneline
```

```
#try
```

```
git diff file1.code
```

```
#Edit & save file1.code
```

```
#Now try again
```

```
git diff file1.code
```

```
git add .
```

```
#try again
```

```
git diff file.code
```

View logs

Walk-thru Github.com

Working with github

```
#create an account or sign-in  
#create a new repo on github  
# now link your local repo and remote repo, name your name repo origin  
#this way you can push to origin instead of using the whole url  
git remote add origin https://github.com/<your repo>  
#(be sure to be in a local git repo when running this command)  
#you can also use git clone
```

```
#push your local to remote  
git push origin master
```

#TIP: Use StackOverflow for searching issues and resolution

#view commits on github Code->commits

Pulling & Pushing changes

```
#Pulling changes from remote to local  
#On github.com, open a file in your remote repository and edit, commit  
#on local  
git pull origin master  
git log --oneline  
#Observe that number of commits have increased and the change is reflecting  
  
#Pushing changes to remote from local  
#Edit and save one of the files on your local working directory  
git commit -a -m "<comment>"  
git push origin master
```

Quick exercise

```
#Make a change on the remote repo and commit  
#Make a change to the same file (but different line) on the local repo and commit  
#Now try git push origin master on the local repo  
#what is the outcome and why?
```

Day 4

Merging & Conflicts

```
#Merging the 2 changes  
git pull origin master  
#Observe the contents of the changed file - it will have a merge of both changes  
#But remote repo is still out of sync  
git status  
git commit -a -m "<comment>"  
git push origin master  
#now both have the merged file
```

```
#Merge conflict resolution  
#Make a change to a file on local repo and commit  
#make a change to same file and same line number on remote repo and commit  
#run git pull origin master on local repo  
#What do you see?
```

Cloning

```
#cloning in git
#change directory to one step above
cd ..
#create a clone
git clone <sourcereponame> <targetreponame>
cd <targetreponame>
git branch
#observe this shows only the current active branch
git branch -a
#this shows all the branches

#similary you can clone from github
cd ..
git clone <github repo url>
```

Ethereum & Geth

Init and start the console

To init your first ethereum node:

```
geth --datadir "$ethereum_home/NUclass" init "<ethereumdir>/genesis.json" [**run once only**]  
geth --datadir "$ethereum_home/NUclass" console 2>console.log
```

The javascript console starts with a chevron '>'.

admin. [tab key] to view all options possible with the admin command

admin.nodeInfo - observe the default port and the enode id

personal. [tab key]

personal.newAccount() to create a new account

Operations

personal.listAccounts() to list accounts

eth. [tab key]

eth.blockNumber should return 0 since we are yet to create our first block

Create a 2nd node:

```
geth --datadir "$ethereum_home/Nuclass-node2" init "<ethereumdir>/genesis.json"  
[**run once only**]
```

```
geth --datadir "$ethereum_home/Nuclass-node2" --port 30304 --nodiscover --  
networkid 1234 console 2>console.log
```

[**different port number – also, need to restart first node with same networkid**]

(**On Windows, add --ipcdisable, if you get access denied**)