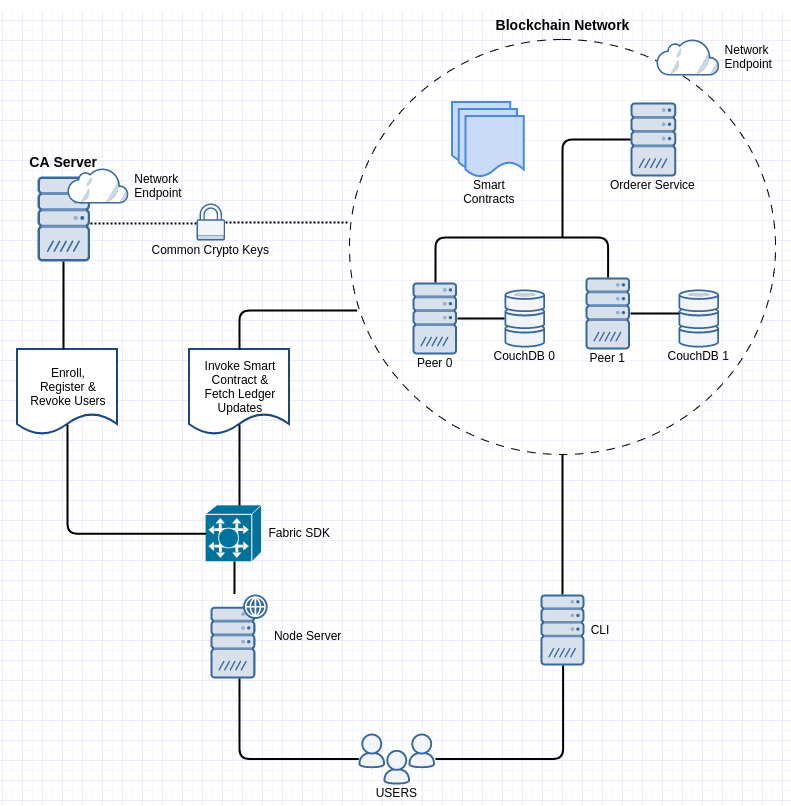
**Hyperledger Fabric on Multiple Instances**

Trial of article at: <https://medium.com/@wahabjawed/hyperledger-fabric-on-multiple-hosts-a33b08ef24f>

This is a simple guide to deploy multiple Hyperledger Fabric instances on different logical instances, following the [Building Your First Network](https://hyperledger-fabric.readthedocs.io/en/release-1.2/build_network.html) sample:



This layout is mapped to actual instances (VMs: **VM1** and **VM2**) as follows:

1. A Certificate Authority (CA) **— VM1** *(key material shared between both the VMs)*
2. An Orderer — **VM1**
3. 1 PEER (peer0 & couchdb0) on — **VM1**
4. 1 PEER (peer1 & couchdb1) on — **VM2**
5. CLI on — **VM2**

Procedure:

1. Setup multiple (2 here) instances on AWS; both instances are within the same VPC and share same subnet, with all traffic opened out between the two.
2. Install prerequisites as in here: <https://hyperledger-fabric.readthedocs.io/en/release-1.2/prereqs.html>
3. Clone the Fabric samples directory from:

git clone -b issue-6978 https://github.com/sstone1/fabric-samples.git

1. Install the Fabric platform specific binaries (needed for crypto tools like cryptogen, configtxgen) using the following command:

curl **-**sSL http:**//**bit**.**ly**/**2ysbOFE **|** bash **-**s 1.1**.**0 1.1**.**0 0.4**.**10

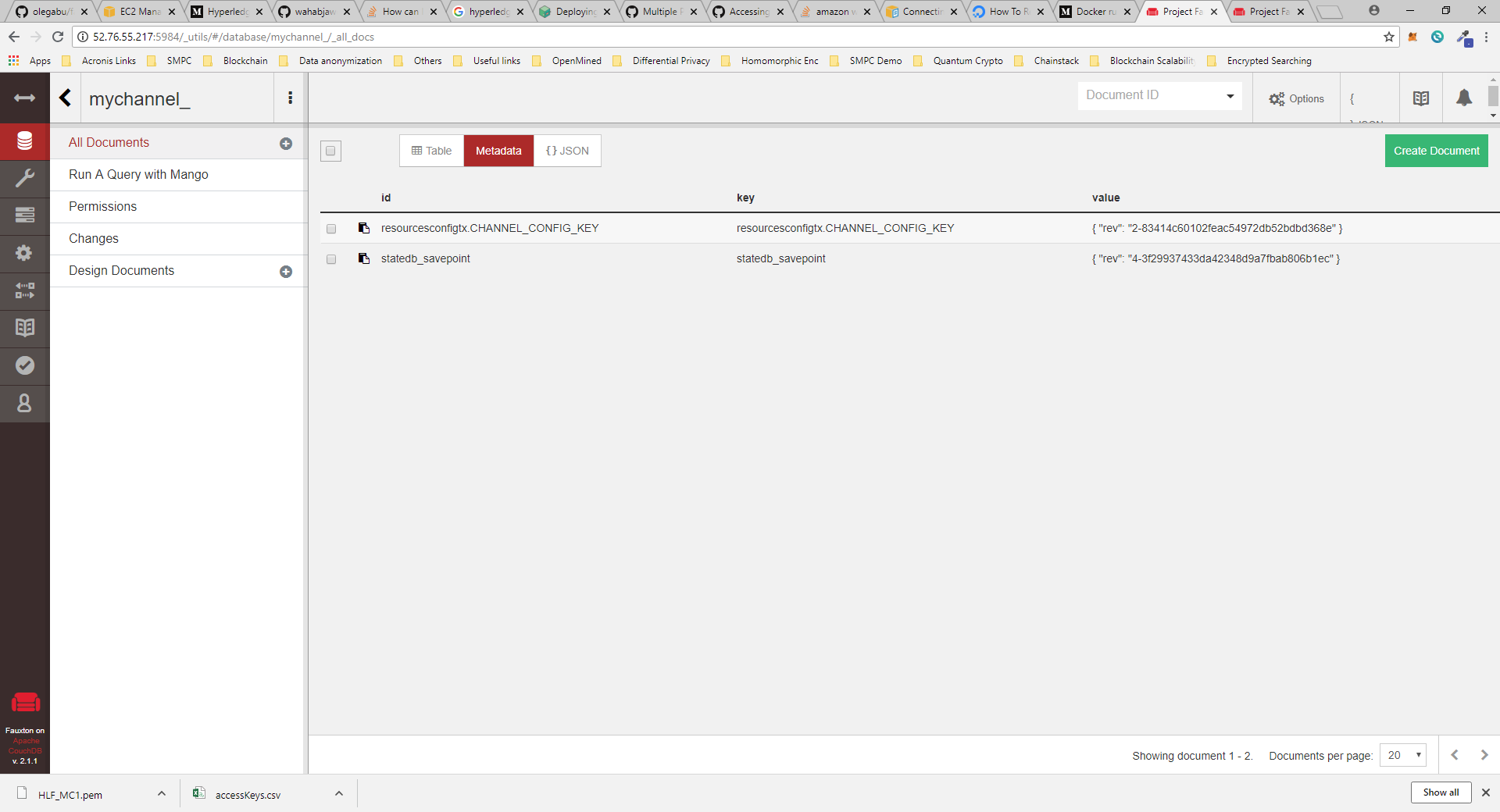
(Version 1.2.0 fails, don’t use for now)

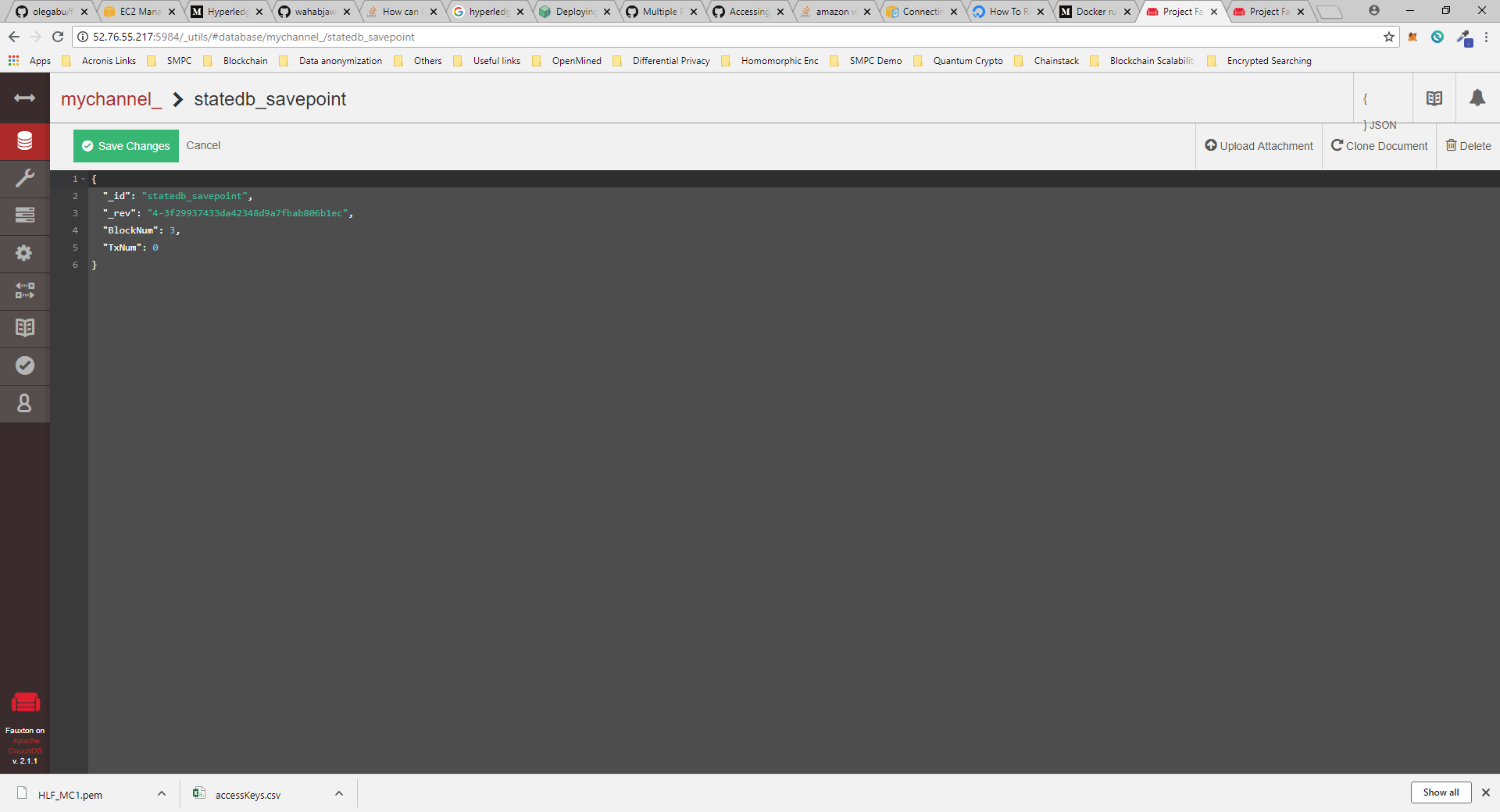
*(Base AMI has been created with the above steps and stored on AWS)*

1. Follow steps as listed in the main source link, starting with initializing a docker swarm and creating an Overlay network.

Notes:

1. The idea is to create a Docker swarm, add both the VMs to it as managers, create an Overlay network and use this for networking between components.
2. Using this link I am able to setup a simple Fabric network, create a channel, join Peers 0 & 1 to the channel, install, instantiate, invoke and query chaincode on that channel.
3. Interaction with the blockchain network is done via CLI docker container running on VM2.
4. The Docker images for each of the components (like CA, Peer, Orderer etc) are pulled from the repo and the containers are launched using ‘***docker run***’ on separate terminals.
5. Each of the containers generate logs for the components they run.
6. The crypto material under the ‘***crypto-config***’ and ‘***channel-artifacts***’ folders have to be shared between the VMs, otherwise the network will not communicate.
7. The CouchDB instances store the states of the 2 peers and can be explored via explorers opened out at specific ports; the statedb snapshots can be seen below:





Future Challenges:

1. Automating these steps
2. Customizing the network: adding or removing peers, sharing credentials; this may be easier with this config: <https://github.com/olegabu/fabric-starter>
3. Interfacing with some application, for instance: <https://hyperledger-fabric.readthedocs.io/en/release-1.2/write_first_app.html>
4. Mounting a multi-org network, with separate CAs and crypto materials;
5. Connecting Hyperledger Composer to the existing Fabric network