**IPFS Private Network**

No connection to the public network will be made, so all the data on this private network will only be accessible to known peers on the network.

**Pre Request :**

AWS EC2 ubuntu instance(Node1)

**Install Go**

1 **Download Go**

|  |
| --- |
| * curl -O <https://storage.googleapis.com/golang/go1.13.5.linux-amd64.tar.gz> * tar -xvf go1.13.5.linux-amd64.tar.gz * sudo mv go /usr/local |

2 **Set Path**

|  |
| --- |
| * sudo subl ~/.profile * export GOPATH=$HOME/work * export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin * source ~/.profile |

3 **Verify**

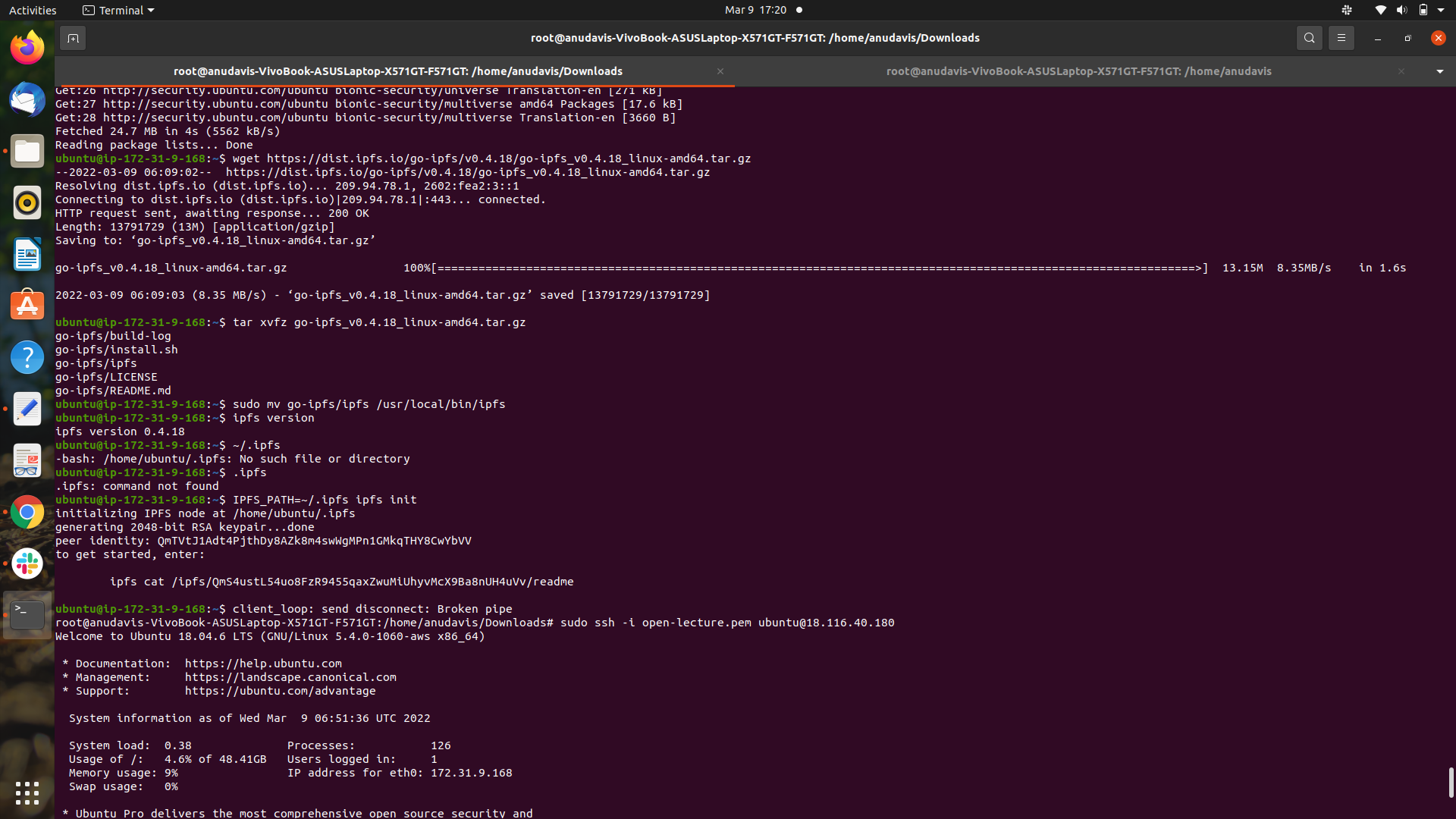
go version

**Installation IPFS:**

|  |
| --- |
| * sudo apt-get update * wget <https://dist.ipfs.io/go-ipfs/v0.4.18/go-ipfs_v0.4.18_linux-amd64.tar.gz> * tar xvfz go-ipfs\_v0.4.18\_linux-amd64.tar.gz * sudo mv go-ipfs/ipfs /usr/local/bin/ipfs * Ipfs version |

**Initialize nodes**

|  |
| --- |
| * IPFS\_PATH=~/.ipfs ipfs init |



# 

# **Create a private network**

* We have now installed two IPFS nodes, which could easily connect to the public IPFS network, but that’s not what we need. To create a private network, we will use a swarm key. This swarm key will be referenced by all the nodes in this private network.
* To generate the swarm key there are two options: use a bash script, or install a key generator.

## **Installation of a key generator**

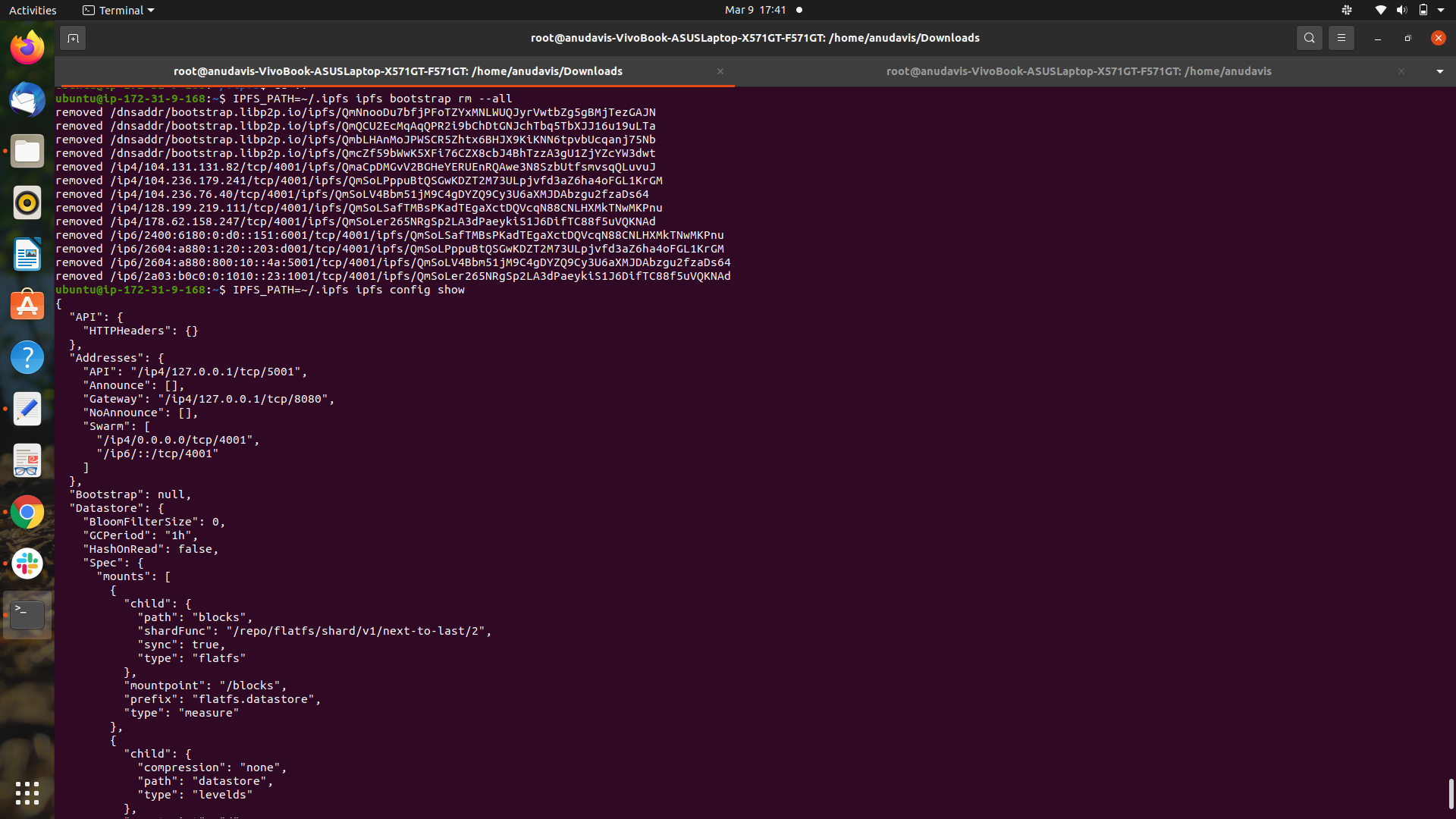
* The swarm key generator is executed on the bootstrap node only.
* To install the swarm key generator we use go get, which uses git. If you have not installed git yet on your bootstrap node.

|  |
| --- |
| * sudo apt-get install git * go get -u github.com/Kubuxu/go-ipfs-swarm-key-gen/ipfs-swarm-key-gen * ipfs-swarm-key-gen > ~/.ipfs/swarm.key |

**Remove the default bootstrap node**

* In order not to connect to the global IPFs network, you need to delete the node information of the default bootstrap.

|  |
| --- |
| * ipfs bootstrap rm --all * IPFS\_PATH=~/.ipfs ipfs config show |



* Get your peer id

|  |
| --- |
| * IPFS\_PATH=~/.ipfs ipfs config show | grep "PeerID" |

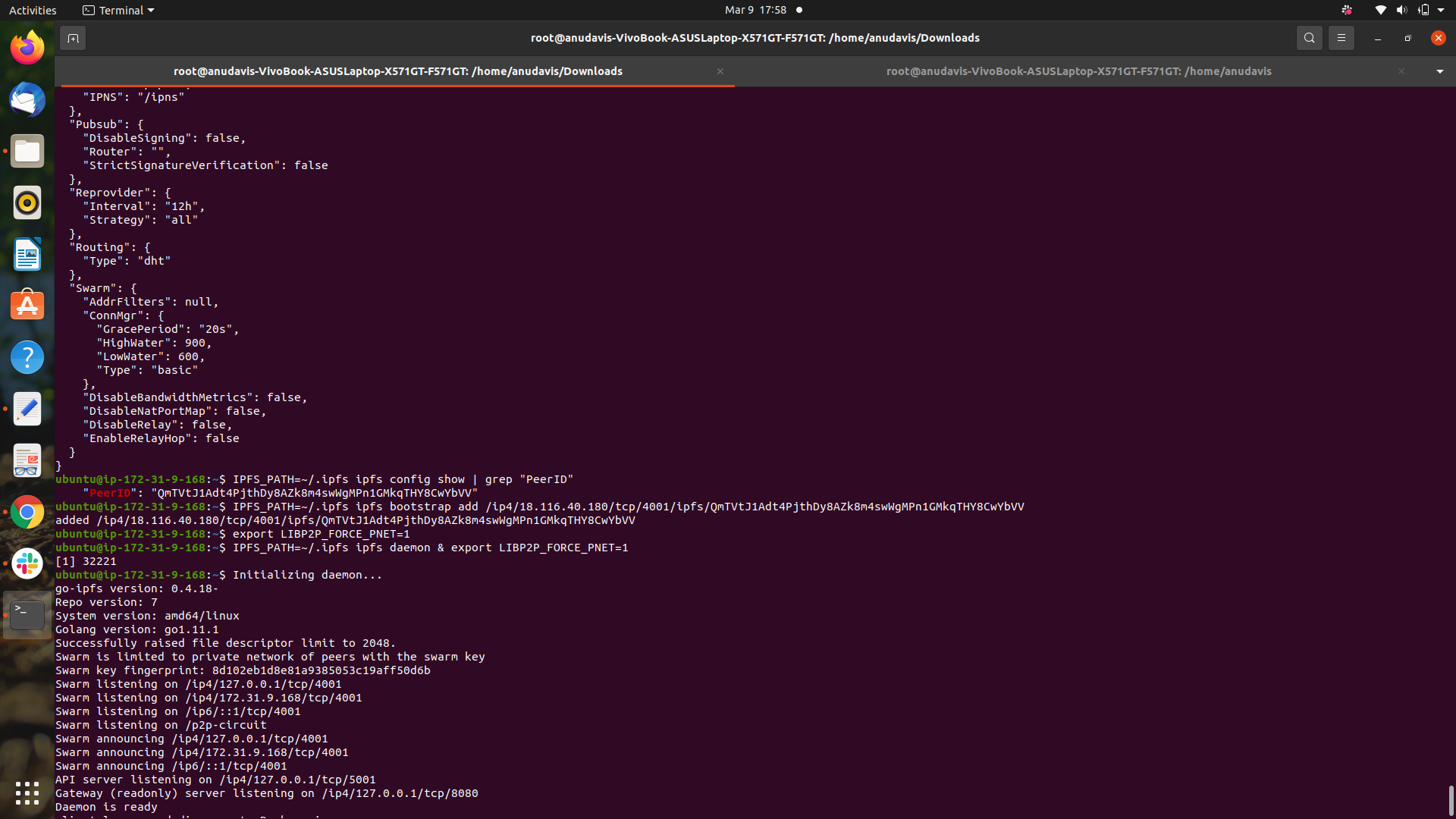
* Add bootstrap

|  |
| --- |
| * IPFS\_PATH=~/.ipfs ipfs bootstrap add /ip4/18.116.40.180/tcp/4001/ipfs/QmTVtJ1Adt4PjthDy8AZk8m4swWgMPn1GMkqTHY8CwYbVV |

**Start the network**

* The private network is installed, so we can test this network. We will use an environment variable to make sure that if there is a mistake in our configuration or the private network is not fully configured, the nodes don’t connect to the public IPFS network and the daemons just fail.

|  |
| --- |
| * export LIBP2P\_FORCE\_PNET=1 * IPFS\_PATH=~/.ipfs ipfs daemon & export LIBP2P\_FORCE\_PNET=1 |



**Reference:**

<https://developpaper.com/construction-of-ipfs-private-network-cluster/>

<https://medium.com/@s_van_laar/deploy-a-private-ipfs-network-on-ubuntu-in-5-steps-5aad95f7261b>