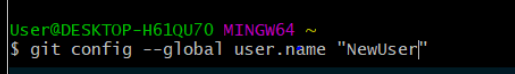
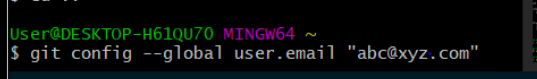
DEVOPS TRAINING

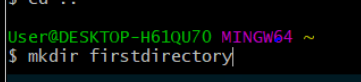
DEVOPS

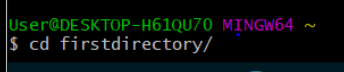
1. Configuration: Set name and email of the user.

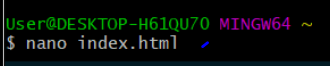


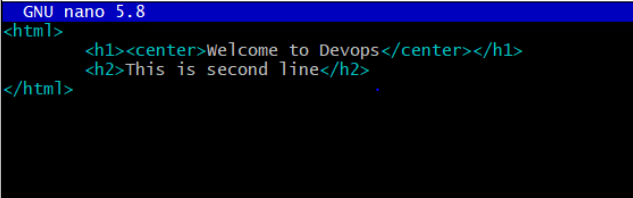


1. Make a directory and change dir



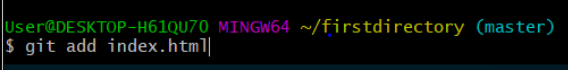


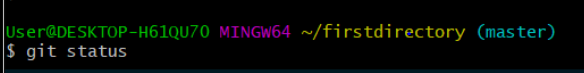
1. Write code(this case a small html). nano <filename>, ctrl+o for saving, ctrl+x for exiting



1. Initial git in the directory. git init



1. Ls for the files in the dir
2. Add file
3. Git status to know the changes staged, which files are being tracked by git.

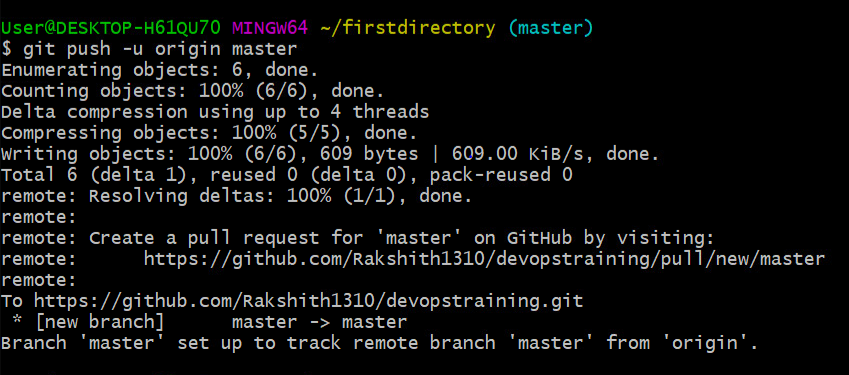


1. Git commit -m <message> to commit the changes
2. Adding files to github. 🡪 git remote add origin <path of the github repo>.

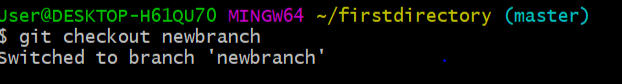
For this, github repo has to be created.



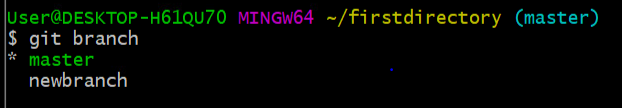
1. Pushing the files to the github



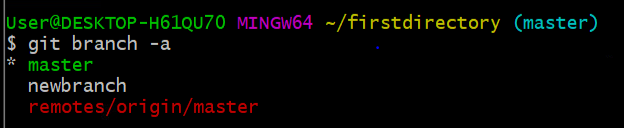
1. Creating new branch



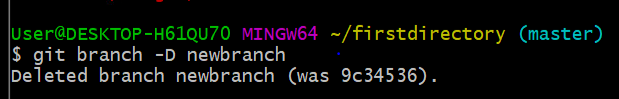
1. Viewing all branches



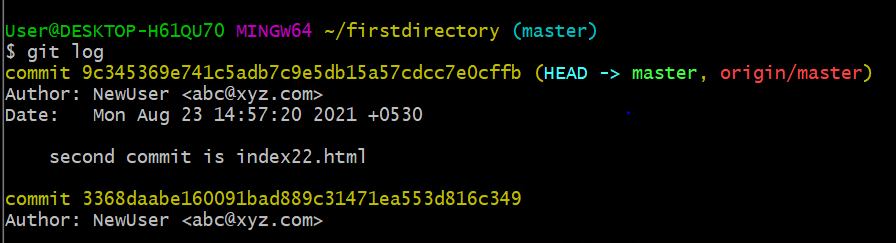
1. Viewing all branches including hidden ones



1. Deleting branches

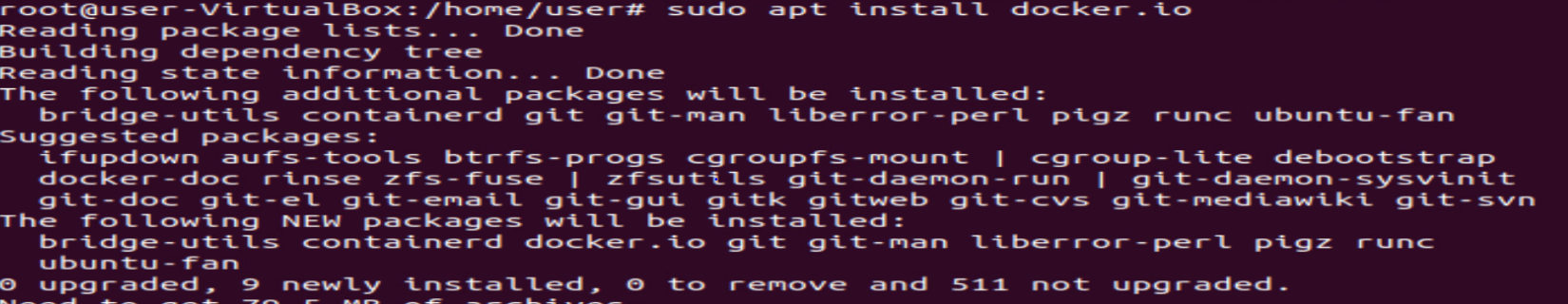


1. Checking logs

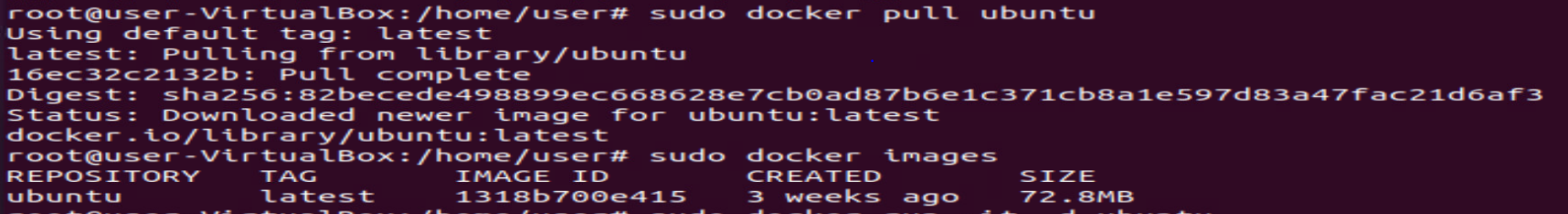


DOCKER

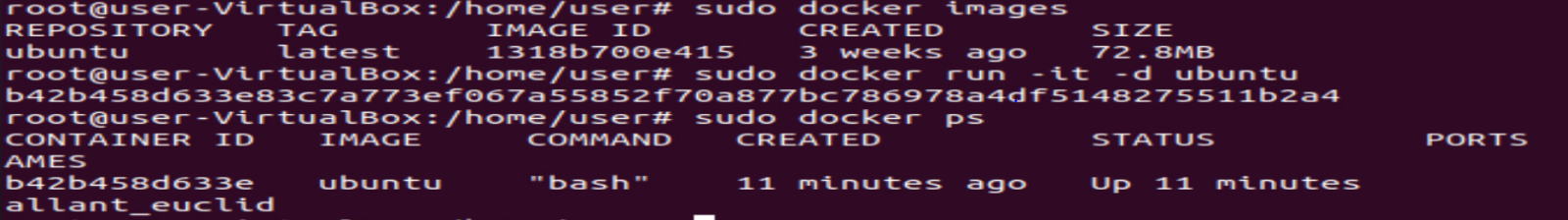
1. Installing docker

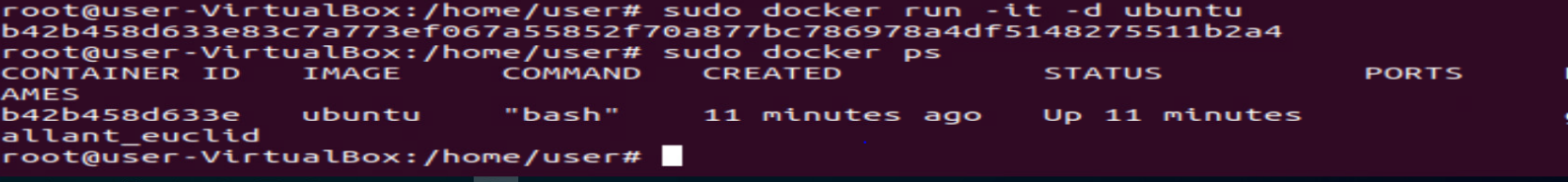


1. Taking ubuntu image/container



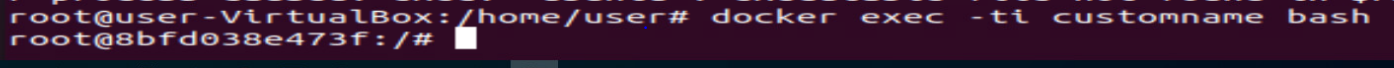
1. Checking docker images and viewing its Id, status..



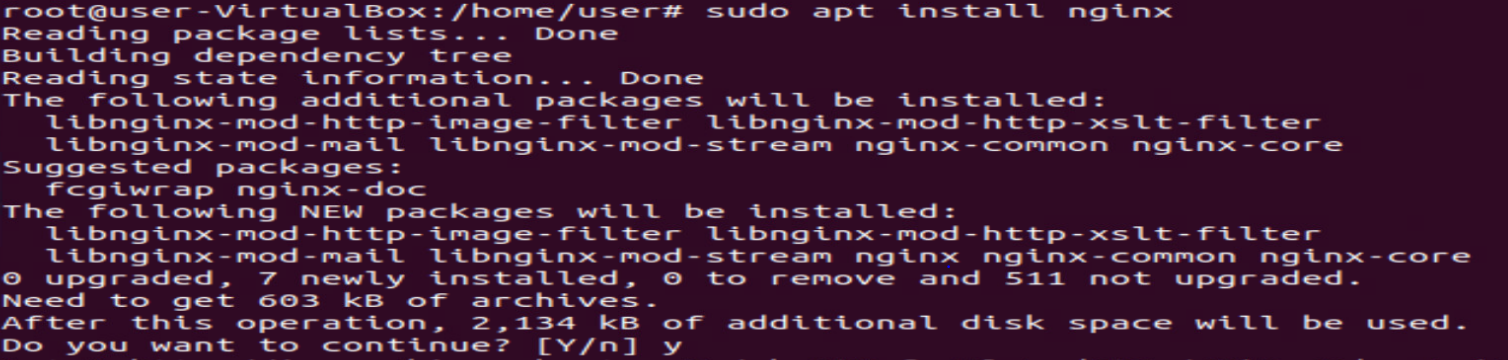
1. We can custom create image by 🡪 sudo docker run -it -d <name> ubuntu
2. Creating customized container



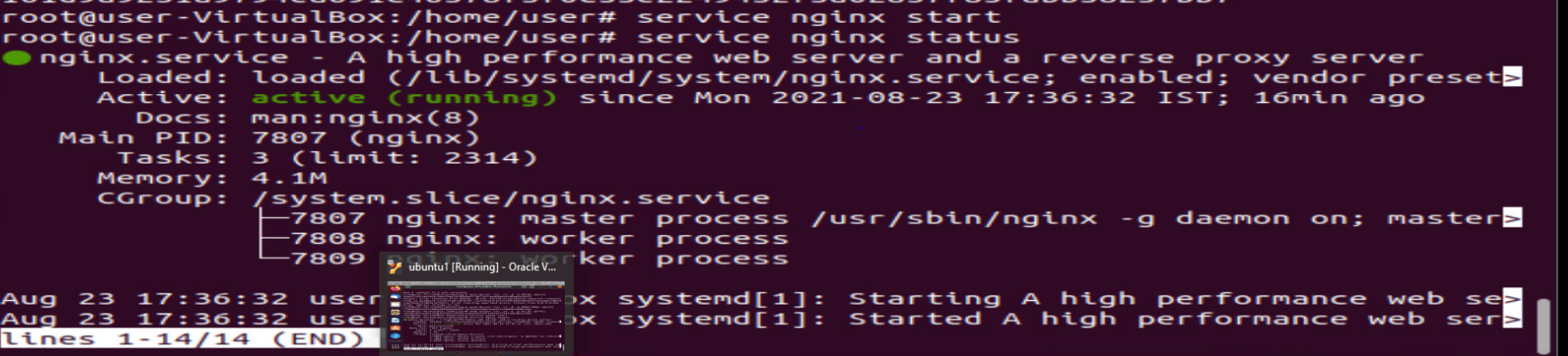
1. Entering into container 🡪 entering into customname(id=8bdf…)



1. Installing nginx, in root!!! Type exit if in another container



1. Starting nginx server and checking its status



AWS

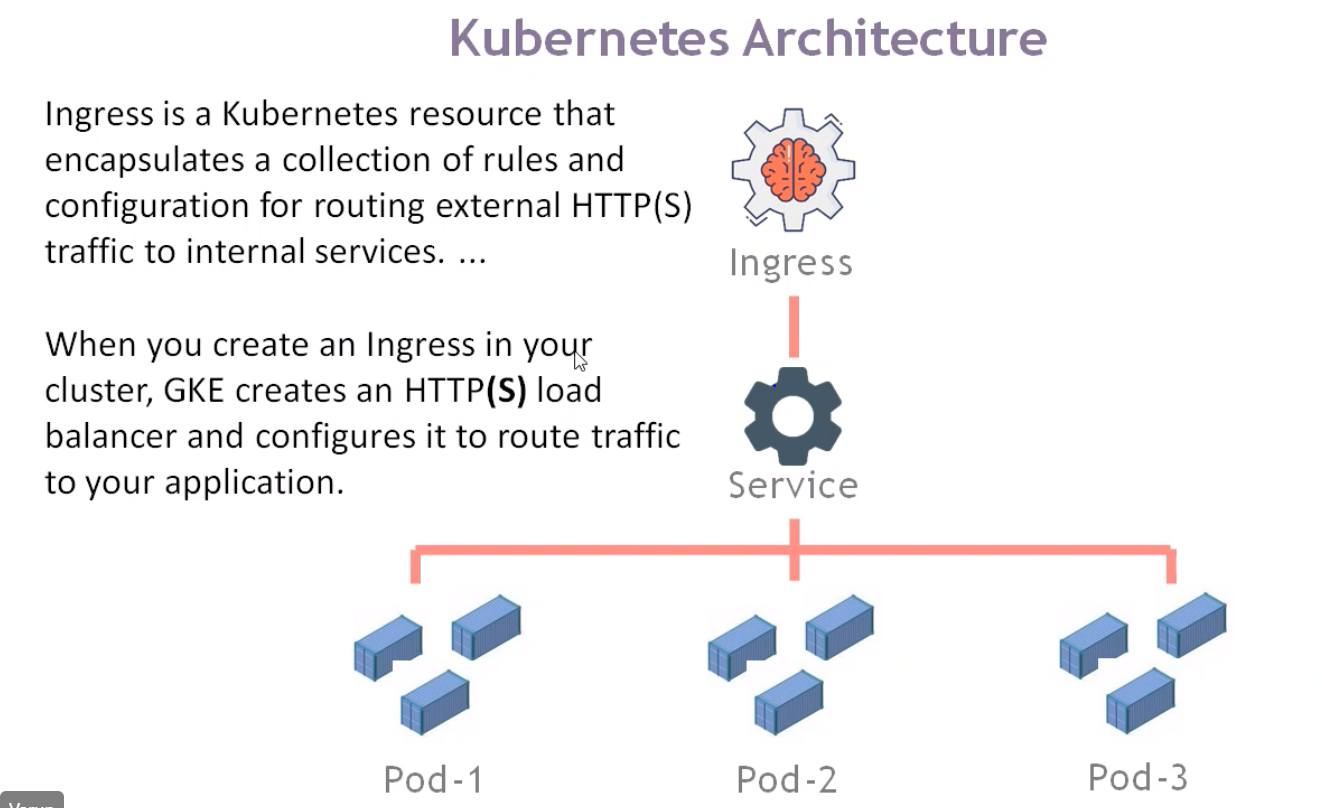
1. Create an aws account
2. Select ec2, select free one(OS)
3. Select instance type- t2, which is free tier
4. Configure, add storage 🡪 no changes to be made.
5. Type 🡪 all traffic, source 🡪 anywhere
6. While launch, create a new key pair (2nd option), download it.

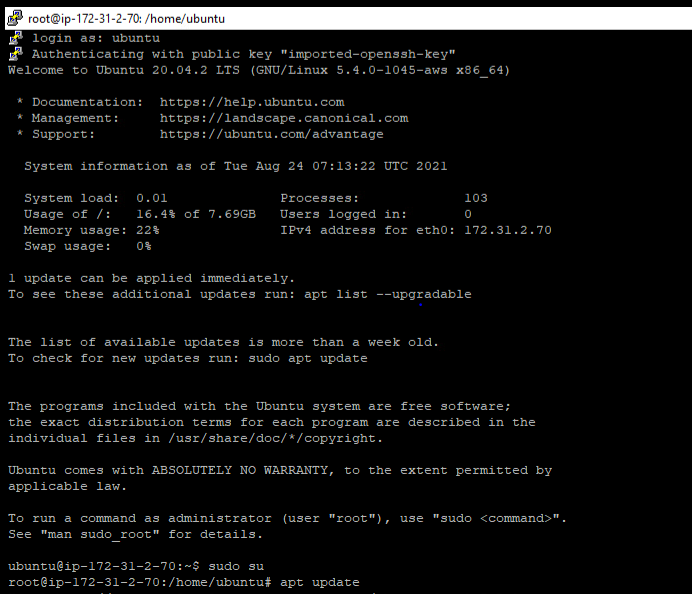
Why putty gen and putty app.

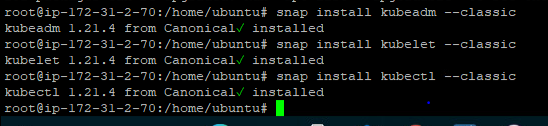
1. Putty gen 🡪 converts .pem file to ppk file
2. Putty app 🡪 launches the ppk file and connects to aws instance(not sure nodko).

KUBERNETES

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>



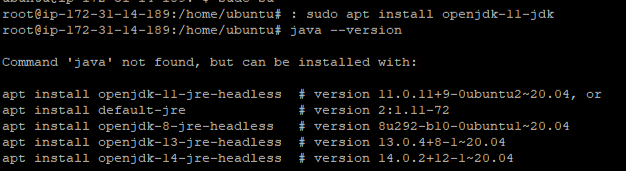
1. Create a dockervm, a master and two slave instances, and configure all of them with respect to putty gen and putty app.
2. Create instance, download key pair of instances (which would be .pem file). Copy its ipv4 public address, open puttyGen browse pem file and save its private key (with same name as .pem file)
3. Open putty app, paste IPv4 address here, search ssh 🡪 auth 🡪 browse for ppk file (which created during puttyGen) 🡪 open. This opens cmd prompt.
4. Type ubuntu 🡪 sudo su 🡪 apt update
5. Install docker and Kubernetes (kubeadm, kubectl, kubelet) in all instances i.e., master&slaves.



JENKINS

[Installation of Jenkins on ubuntu](https://www.jenkins.io/doc/book/installing/linux/" \l "debianubuntu)

1. Create a Jenkins instance in aws. Use puttyGen and putty app as usual.
2. Open putty app, paste IPv4 address here, search ssh 🡪 auth 🡪 browse for ppk file (which created during puttyGen) 🡪 open. This opens cmd prompt.
3. Install java on ubuntu.



1. Add Jenkins repositories





1. Unlock Jenkins by knowing the IPv4 address of Jenkins instance.
2. We will get a link in the browser, prefix “cat” and then run the link to obtain password, paste it in the browser to open the dashboard.