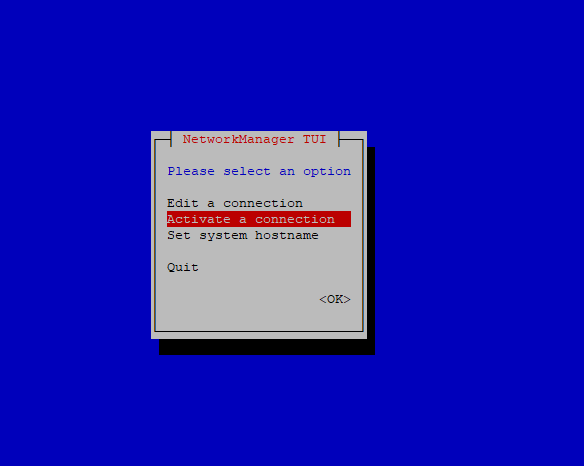
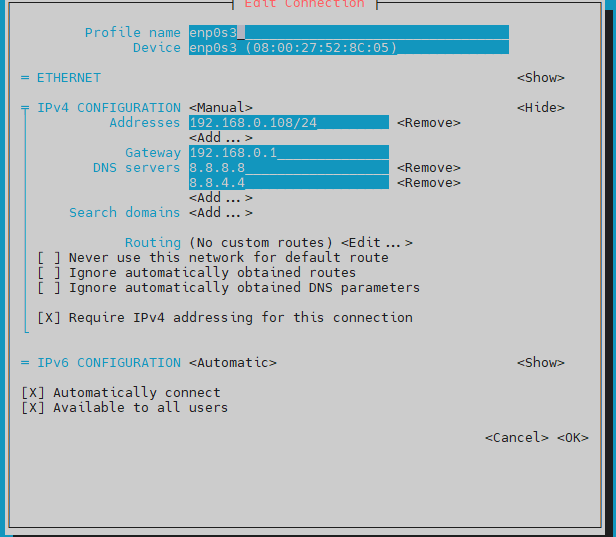
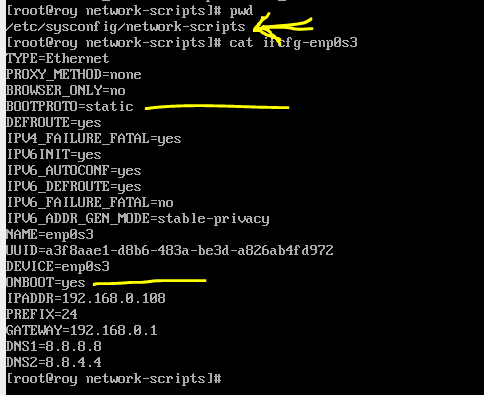
# 

# For CentOS 7 :-

# #**nmtui**







**/etc/init.d/network restart**

# systemctl restart network.service

or

# service network restart

or

# /etc/init.d/network restart

# Disable SELINUX:

#vi /etc/selinux/config

SELINUX=disabled

#sestatus

# In Ubuntu :

**cd /etc/netplan/02-static-ip.yaml**

**network:**

**version: 2**

**ethernets:**

**ens33:**

**dhcp4: no**

**addresses:**

**- 192.168.72.140/24**

**gateway4: 192.168.72.2**

**nameservers:**

**addresses:**

**- 8.8.8.8**

**- 8.8.4.4**

**# sudo netplan apply**

**#ping 8.8.8.8 // to test internet**

# Allow a port from Firewall:

firewall-cmd --permanent --add-port=9999/tcp

firewall-cmd --reload

# Where Are The Html Files Located In Linux?

There are two answers in this quiz. [HTML files](https://www.systranbox.com/how-to-host-a-html-website-on-a-linux-cpanel/) are typically sent to /var/www by default. /var/www/test, for example.

**Install Apache/httpd**

Use the following steps to install Apache:

1. Run the following command:
2. yum install httpd
3. Use the systemd systemctl tool to start the Apache service:
4. systemctl start httpd
5. Enable the service to start automatically on boot:
6. systemctl enable httpd.service
7. Open up port 80 for web traffic:
8. firewall-cmd --add-service=http –permanent
9. firewall-cmd --permanent --add-port=8181/tcp
10. Reload the firewall:
11. firewall-cmd –reload
12. firewall-cmd --list-all

Confirm successful installation by entering your server’s IP address in a browser to view the default Apache test page.

**Summary** :

Runnig on by default port : 80

Document root folder : /var/www/html/

**# To change port 80 :**

cd /etc/httpd/conf

vi httpd.conf

Search by "Listen"

**Nginx**

Install from local repo: -

cd /etc/yum.repos.d/

vi nginx.repo

[nginx]

name=nginx repo

baseurl=http://nginx.org/packages/mainline/centos/7/$basearch/

gpgcheck=0

enabled=1

yum repolist

yum install nginx -y

**Summary** :

Runnig on by default port : 80

**# To change port 80 :**

cd /etc/nginx/

vi nginx.conf

**Apache Tomcat**

# yum install java-1.8.0-openjdk-devel

#java -version

# useradd -m -U -d /opt/tomcat -s /bin/false tomcat

# yum install tomcat

# yum install tomcat-webapps tomcat-admin-webapps

# yum install tomcat-docs-webapp tomcat-javadoc

# systemctl start tomcat

# systemctl restart tomcat

# systemctl enable tomcat

# systemctl status tomcat

# firewall-cmd --permanent --add-port=8080/tcp

# firewall-cmd –reload

# systemctl restart tomcat

Need to change name and password :-

cd /usr/share/tomcat/conf/tomcat-users.xml

<user name="admin" password="admin" roles="admin,manager,admin-gui,admin-script,manager-gui,manager-script,manager-jmx,manager-status" />

readlink -f $(which java)



Now,

sudo vim /etc/systemd/system/tomcat.service

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=oneshot

RemainAfterExit=yes

User=tomcat

Group=tomcat

Environment="JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.372.b07-1.el7\_9.x86\_64/"

Environment="JAVA\_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"

Environment="CATALINA\_BASE=/usr/share/tomcat"

Environment="CATALINA\_HOME=/usr/share/tomcat"

Environment="CATALINA\_PID=/usr/share/tomcat/temp/tomcat.pid"

Environment="CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/usr/share/tomcat/bin/startup.sh

ExecStop=/usr/share/tomcat/bin/shutdown.sh

[Install]

WantedBy=multi-user.target

# systemctl restart tomcat

http://192.168.0.202:8080

Now you may put the user and password as admin

Install **HAProxy** (An open source LB)

Algorithm:

* Round Robin
* Weighted Roun Robin
* Leastconn

sudo yum update

sudo yum install haproxy

sudo nano /etc/haproxy/haproxy.cfg

cat /etc/haproxy/haproxy.cfg // *there are four section in this configuration*

global

log /dev/log local0

log /dev/log local1 notice

chroot /var/lib/haproxy

stats timeout 30s

user haproxy

group haproxy

daemon

defaults

log global

mode http

option httplog

option dontlognull

timeout connect 5000

timeout client 50000

timeout server 50000

frontend http\_front

bind \*:80

stats uri /haproxy?stats\_roy

default\_backend http\_back

backend http\_back

balance roundrobin

server web\_1\_roy 192.168.0.202:80 check

server web\_2\_roy 192.168.0.203:80 check

http://192.168.0.201/haproxy?stats\_roy

sudo systemctl restart haproxy

sudo systemctl enable haproxy

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd –reload

firewall-cmd --list-all

On both appserver, following command need to be ensured :

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

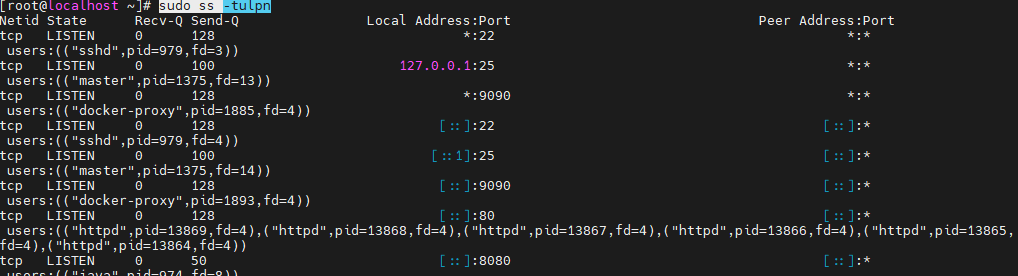
firewall-cmd --permanent --add-port=80/tcp

firewall-cmd --reload

firewall-cmd --list-all

**To see the ports used in the server**

sudo ss -tulpn



Remove any package

rpm -qa | grep java

yum remove <package\_name>

**GIT**

**git init //declare the repository on the directory**

**git add //send the files in the staging**

**git commit //include a version id which to be used for versioning**

**# git remote add origin https://github.com/shuvro86/testing\_git.git**

**# git branch -M main**

**# git push -u origin main**

**git branch //show branch**

**git checkout stage //switching to stage brnach**

git clone -b <branch> <remote\_repo> //clone specific branch

**Docker Network 👍**

Create Network

-----------------------

docker network create my\_net

Run two different containers from respective images

------------------------------------------------------------------------

docker run --name my\_nginx -p 5000:5000 -dit --network my\_net --rm nginx:alpine

docker run --name my\_node -p 6000:6000 -dit --network my\_net --rm node:alpine

SSH to containers to test connectivity among those containers

-----------------------------------------------------------------------------------

docker exec -it my\_node sh

ping my\_nginx

docker exec -it my\_nginx sh

ping my\_node

**RabbitMQ**

#######Pull code from Git repo################

git init

git pull <https://github.com/page-cloud/rabbitmq-message-broker.git>

##########Create Network###########

docker network create rabbits

##########Run RabbitMQ###########

docker run -d --rm --network=rabbits --hostname rabbit-1 -p 8080:15672 --name rabbit-1 rabbitmq:3-management

<http://192.168.0.201:8080/>

##########Build and run publisher from dockerfile##############

docker build -t my\_pub:v1 .

docker run -it --rm --network=rabbits -e RABBIT\_HOST=rabbit-1 -e RABBIT\_PORT=5672 -e RABBIT\_USERNAME=guest -e RABBIT\_PASSWORD=guest -p 8081:80 my\_pub:v1

#############Publish a message named HereIam#########

curl -X POST http://localhost:8081/publish/HereIam

##########Build and run consumer from dockerfile##############

docker build -t my\_con:v1 .

docker run -it --rm --network=rabbits -e RABBIT\_HOST=rabbit-1 -e RABBIT\_PORT=5672 -e RABBIT\_USERNAME=guest -e RABBIT\_PASSWORD=guest my\_con:v1

**Install Docker Compose in CentOS**

sudo curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

# docker-compose --version