Read docs,

* How to install CENTOS on VirtualBox
* How to login using putty.exe
* What is RPM

Redhat Package Manager

https://app.pluralsight.com/paths/certificate/linux-foundation-certified-system-admin

Todays, Topics and Assignments:

Module 1: LFCS; Linux Essentials

Lesson 4. Working the Command Line

EXCERSIZES:

1: Start a shell. Enter each of the following commands in turn.

■ date ? #todays date mm/dd hh:m:sec timezone yr

■ whoami #root

■ hostname #localhost.localdomain

■ uname #version name of the machine - Linux

■ uptime #current time, length of running time, user#, load average in 1,5,15 mins

2. a. Use the ls command to see if you have any files.

b. Create a new file using the cat command as follows:

$ cat > hello.txt Hello world!

This is a text file.

Press Enter at the end of the last line, then Ctrl+D to denote the end of the file.

( cat > hello

I am prashanth

Yes yes yes ctrl+d

c. Use ls again to verify that the new file exists.

d. Display the contents of the file.

Cat name of the file

e. Display the file again, but use the cursor keys to execute the same command again without having to retype it.

3. a. Create a second file. Call it secret-of-the-universe, and put in whatever content you deem appropriate.

b. Check its creation with ls.

c. Display the contents of this file. Minimize the typing needed to do this:

■ Scroll back through the command history to the command you used to create the file.

■ Change that command to display secret-of-the-universe instead of creating it.

4. After each of the following steps, use ls and cat to verify what has happened.

a. Copy secret-of-the-universe to a new file called answer.txt. Use Tab to avoid typing the existing file’s name in full.

b. Now copy hello.txt to answer.txt. What’s happened now?

c. Delete the original file, hello.txt.

d. Rename answer.txt to message.

e. Try asking rm to delete a file called missing. What happens?

f. Try copying secret-of-the-universe again, but don’t specify a filename to which to copy. What happens now?

Archiving Files

completed

Accessing command line help

completed

Understanding file permissions

Accessing root account

* Sudo lets you run one line of command as a super user while su changes the user as a super user
* Add file into sudoers by

**Sudo visudo** – to open the vi editor

**Prashanth ALL = (ALL) ALL** – to add user

Add users and give permissions available here:

<https://glenngillen.com/thoughts/adding-yourself-to-the-sudoers-file>

Accessing servers with SSH

**Ssh -l root ipaddress** - Login as root from ssh

But if we want to stop people accessing root;

Vi /etc/ssh/sshd\_config

VI opens

Change #permitrootlogin to **no** below loginGraceTime

**Systemctl restart sshd** (for system restart)

Logout and try to relogin using **ssh -l root ipaddress** and permission denied now

Try logging in using **ssh prashanth@ipaddress**

Then **su -** and give password to login. It shows a report about the previously failed login

In this link, today complete all these tasks

Today, complete the tasks from the above link,

Starting and stopping CentOS 7

Messaging:

Open two putty machines one for root and another with hostname

In root host type **write prashanth** and fill the context

**Ctrl+d** will quit the message

If you have to send message to a large number of users as a broadcast

**Cat >message<<END** and fill in context

**wall <message** to broadcast message

Shutdown:

**Halt poweroff reboot** are three commands used separately to shutdown a system

Also init 0 and telinit 0 can be used for the same function (see –help)

**Shutdown -h 10 “system shutting down in 10 minutes”** (creates auto shutdown with message)

**Shutdown -c** or **ctrl+c** (cancels the shut down)

Changing Runlevels:

Who -r = to determine the run level

Systemctl get-default = shows the run level; ex: multiuser target is run level 3, graphical target is level 5

systemctl list-units --type=target == is used for checking available targets

The Boot Process

Grub:

Enter boot screen to see 3 kernels and continue to default

Login and open **vi /etc/default/grub**

Change grub\_disable\_recovery = “false” which will disable the recovery mode and making it false means enabling it

Save and exit vi

**Grub2-mkconfig -o /boot/grub2/grub.cfg** == to update grub config

Type reboot

Now there are many recovery modes in the boot menu

Open the latest (second) recovery mode

Put in passwd

Check **who -r** and you can see runlevel1

Reset/Recover lost root passwords:

Boot – press e for edit

Go to linux 16 and to the end of the line

Erase **rhgb quiet** and replace with **rd.break enforcinf=0**

Ctrl+x to continue boot. Boot stops at RAM phase

Switch\_root:/# is the new host

Mount -o remount,rw /sysroot == remount the root file system as it is mounted to read only

Mount -o remount,rw /sysroot

Chroot /systroot == sets false root

Passwd

Enter new pass

Exit

Mount -o remount,ro /sysroot == change earlier rw permission to ro

Exit

Logout ---- this will continue the boot process

Now login as root and use the new pass

Restorecon /etc/shadow

Setenforce 1 === back in enforcing mode for selinux

**Managing Linux Processes**

Monitor Linux Performance

Using Sysstat to monitor Performance

Managing Software on CentOS 7

Introduction to linux user and group management

Creating and managing local users

Find your group using id -Gn

Sudo useradd -m user1

Give passwd

This creates new user

Managing local groups

Id -Gn = shows your current group

Sudo usermod -G sales, root = changes your id to root sales group

Using PAM to control user access

Task

Partitioning & file system

That comes between the storage space, the partition creates the difference

Journaling

Why xfs

Mount

Fdisk vs xfs in partitioning

Nic bonding

PARTED

Parted /dev/sdb

Mklabel msdos

Mkpart

Primary; xfs give start and end

Mkpart

Follow instructions -------------------- sdb1 and sdb2 created

VOLUME GROUP

Pvcreate /dev/sdc /dev/sdd /dev/sde

Vgcreate vg\_vm1 /dev/sdc /dev/sdd /dev/sde

Vgdisplay == you can see that 15GB is created as a VG

MOUNT

Mkdir sdb1\_xfs sdb2\_ext4

Mkfs.xfs /dev/sdb1

Mkfs.ext4 /dev/sdb2

Open vi /etc/fstab

Allocate UUID path xfs or ext4 defaults using info from blkid -o list

LFCS

Ip vs ifconfig

Ifconfig enp0s8 == see ip details

Ip r == shows root ip

Ip n == neighbor

Ip netns == creates namespaces to create addl. Kernel. Namespaces can be used for running on

Creating Network namespace

# namespaces mimic what mininet does automatically

Sudo ip netns add development

Ip netns



Configuring hostnames

/etc/hostname

Hostname name == gives a hostname

Hostnamectl status == gives the existing hostnames

Hostnamectl set-hostname name\_ofthe\_host == sets name for the host

Hostnamectl set-hostname “illegal characters” –pretty/static == sets pretty/static name

Hostname -f == see the hostname with address

DNS name resolution:

Dig command can be used to query DNS

Dig twitter.com +short == gives ip of twitter

Dig twitter.com == displays all info about twitter in sections

Dig twitter.com +noall +answer == keeps the answer sections and deletes all other sections

Netstat

Netstat -nr (or) route == shows your ip, gateway, genmask

Netstat -nt == lists tcp connections

Netstat -l == find all listening connections

Netstat -lt == find all tcp listeners

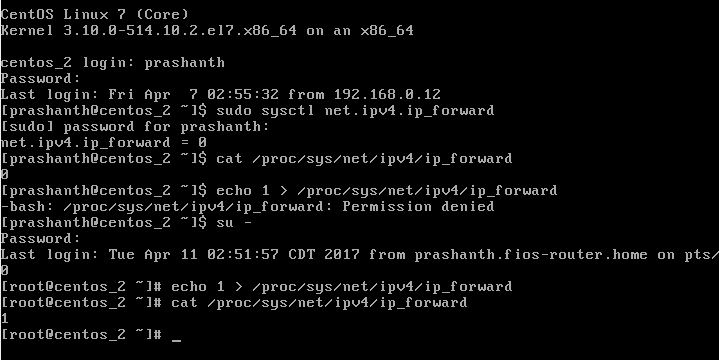
Netstat -lu = find all listening udp

Netstat -s == statistics of connections (applies for su and st)

Netstat -lp == which application listening to which port

Network Services

Enabling IP forwarding



Use echo 0 > /proc/sys/net/ipv4/ip\_forward == to disable

Firewalls

Enable firewall == systemctl enable/disable/status firewalld

VMWARE

Step 1 creating usb bootable pendrive

Step 2 install esxi 6.5

3 add iso to datastore

4 creating win vm

5 creating centos vm

6 install vcenter in win vm

7 add hosts

8 creating snapshots

9 creating template

Kickstart in servers

Ghost image –

Dual booting

Drs scheduling

Vcenter Server HA

Vcenter installation requires installation of .Net framework 3.5

SSID service set identifier

**Where is Vm’s log located**

# **AWS**

* Attach a role to a service

To a user using AWS a/c

To anybody using AWS resources

* The levels of a user to connect to an ec2 instance are 5. The iam user connects from internet gateway, route tables, nacl, subnet, security group and then to ec2 instance, The same order follows backwards for an ec2 instance to connect to an iam user.
* Proxy servers ensure low latency and high availability
* Stateless – has allow and deny policies. Stateful – doesn’t have an allow or deny option it just explicitly allow. Anything we create in security group it allows. But in nacl everything will be denied and allowed only when setup so. The allow and deny uses the last setup rule
* Power users, admin users and read only users are the three kinds of users in AWS
* Types of instances in AWS are 3. On demand instance – its more expensive, reserved instances – for 2-3 year project, spot instances – bidding ; but if the price goes above the acquired bid the instance is terminated, hence used for testing purposes.
* Create a snapshot when working on spot instances. Snapshot volumes are preserved in S3 bucket.

Lambda, SNS, VPN network and security

Standard, glacier, infreq, reduced redundancy,

VPC: is a logical isolation and definition within AWS where AWS resources can be launched.

IAM: is a resource and access control manager for the users. IAM enables to set permissions for allowing or denial of service.

EC2: consists of varying combinations of CPU, memory and storage that provides the desired mix of resources.

ELB: manage the incoming traffic across multiple instances to achieve fault tolerance. Classic and application load balancers are two types of them. Provides availability through fault tolerance, elastic and secure.

Route 53: 53 because of the tcp/udp port 53 it uses where requests are sent. DNS generally uses 53. The service redirects the domain to the site built in EC2. It provides various routing policies like weighted, latency, geolocation, failover.

S3: is a service by AWS where we can store documents, files etc. Websites can also be hosted on S3 which are image and file centric. Capability of versioning.

Partial, incremental and full backups can be stored including archives.

Bastion host: provides access to remote infrastructure through SSH. It is easier to log sessions and admin the sessions

Security group: acts like a virtual firewall for the instance. You add rules that control the incoming traffic.

NACL: subnet level firewall which is optional.

|  |  |
| --- | --- |
| **Security Group** | **Network ACL** |
| Operates at the instance level (first layer of defense) | Operates at the subnet level (second layer of defense) |
| Supports allow rules only | Supports allow rules and deny rules |
| Is stateful: Return traffic is automatically allowed, regardless of any rules | Is stateless: Return traffic must be explicitly allowed by rules |
| We evaluate all rules before deciding whether to allow traffic | We process rules in number order when deciding whether to allow traffic |
| Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on | Automatically applies to all instances in the subnets it's associated with (backup layer of defense, so you don't have to rely on someone specifying the security group) |

**Forward lookup zone & reverse lookup zone**

**A record , aaaa record, c record, mail exchange**

**User group creations, sudoers, file permissions, services to install, lvms, mounts, httpd, change default ports, shell scripts, cron jobs, netstat, top, vmstat, free, home folders, conf files, monitoring tools,**

**Increase swap memory**

**SDLC – methodologies and tools**

* Agile, rationally unified p , waterfall
* Tools

Most web applications come in java, .net, c etc, node js on server side and angular JS, rennet Js on client side.

Once infra is created we have to install and configure software based on applicatons or DB.

* SDLC CONSISTS OF REQUIREMENT GATERING and planning to achieve goals, time, budgeting and investment. This is the inception stage.
* Microservices

CI/CD

* Environment provisioning, config mgmt. and deployment are some of the important aspects of CD
* Sysadmin does the environment provisioning and config mgmt. and deployment are done by middleware admins
* CI is done by build engineers

All the above concepts fall under DevOps

Webserver/ app server / database server

Apache tomcat oracle

Nginx jboss/

Websphere mysql; sql, mongodb, Cassandra

Ih s IIS sql server

Version Control: To manage file changes we need a proper version control to track the change and stop changes. Tools give capabilities to compare between different versions and revisions. Tagging, branching are some of the important functions of VC tools. Each project needs a repository in VC. Repositories are nothing but directories.

Ex of naming convention: svn.jnitinc.com:443/vitalsigns/trunk or branch/ version #

Tag:

Branch: comes after the trunk. Its about managing particular product version.

Trunk: is the master.

Codefreeze

SVN server: centralized repository. Tortoise or eclipse are the clients. Updates/checkin/checkout

Git: is a distributed repository. Each user has his own repository. Pull/push/ pull request

CI

* Source code is the start and
* Source code being the first step, compiling is the second
* Execute test cases (Junit)
* deployable artefact/ .war, .jar, .dll) to end with.(nexus, maven, artefactory)

The following happens due to a build machine where we run the sequence of all instructions.

* It uses ant, maven scripts to perform builds
* PowerShell/ MS build and .net apps

Jenkins becomes the build machine wherever its installed

MySQL is light weight