Linux Commands

- **chown [-R] username.groupname filename:-** This commandis used to change the ownership of a file.
- **Chmod [options] [mode] filename:-** This command isused to change the permissions of the file.
- **Passwd:-** This command reset passwrd for user
- **userdel:-** This command is used to delete a useraccount
- **useradd:-**This command is used to add/create a user inRedHat based OS
- **Adduser:-** This command add/create a user in Debianbased OS.
- **addgroup:-** This command used to add/create a group.
- **Last:-** Show last logins on the system.
- **lsof -u testuser:** List files opened by specific user
- **usermod -G groupname username:-** Used to add user ingroup.
- **find:-** Used to search a file or directory's path.
- **free -m:** Shows amount of used and free memory
- **df -h:** Display disk space in human-readable form
- **uptime:-** Show how long system is running.
- **echo $?:-** It will return exit status of last command.
- **ps:-** Display your currently active proccesses.
- **top:** Display all running process.
- **cat:** Displays the file.
- **head:** It prints top 10 lines of a specific file.
- **tail:** It prints last 10 lines of a specific file.
- **Less:** Display file content page wise or line wise.
- **grep:** Find texts from any text input.
- **Ifconfig:** Displays the current network interaceconfiguration information.
- **file filepath:** Used to determine the type of file.
- **Sed 's/oldstring/replacestring/g' filename:-** standsfor stream editor, it is used to find and replace astring in file.
- **Hostname**: show system hostname.
- **Ssh user@host:- (secure shell)** It used to connecthost as user.
- **scp:- (secure copy)**It allows you to securely copyfiles/directories between two locations
- **rsync:-** It is used to synchronize files between twolocations.

- **systemctl start service:-** used to start the service

AWS SERVICES

Amazon Web Service is a most comprehensive & broadly used cloud platform. It is a flexible & secure cloud environment available today. It offers 200 plus fullyfeatured services to million of customers to lower their cost & innovate faster.
It offers 80 Avalability zones & 25 Regions.

- **IAM: (Identity & Access Management)** You can access and manage users &groups. Also you can choose permissions to allow and deny their access.
- **EC2: (Elastic Compute Cloud)** is a web-service which provides secure andresizable compute capacity in the cloud. That allows users to rent virtual computers.
- **EBS: (Elastic Block Storage)** It provides Extra storage for EC2. It tune theperformance of volume.
- **ELB: (Elastic Load Balance)** It automatically distributes your incommingtraffic among multiple targets/servers.
- **EFS: (Elastic File system)** It is a shared storage in AWS. It automaticallyscale your file system. It uses NFS protocol 2049 for sharing storage.
- **RDS: (Relational Database Service)** is a web-service designed to simplifyset-up operation & scaling of relational database.
- **S3: (Simple storage service)** You can store and retreive any amount of data atany time, from any where. It stores data as Objects within bucket & it is Globalservice.
- **VPC: (Virtual Private Cloud)** Amazon VPC enables you to launch AWS resources on to your own virtual network. You have complete control over yourvirtual network. You can add multi layers of security. It extend your network to cloud.you can use both IPV4 & IPV6 in your VPC
- **EC2 Autoscaling:** It enables you to automatically launch & terminate EC2instance based-on policies.
- **Elastic Beanstalk:** End-to-end web application management service. It

is usedfor deploying and scaling web applications.
You can simply upload your code and elastic beanstalk automatically handlesthe deployment, provisioning, Autoscaling, Load balancing,application to health monitoring.

- **CloudWatch monitoring:** It monitors AWS resources and applications. WithCloudwatch, you can collect and track metrics, collect and monitor log files, and set alarms.
- **CloudFront:-** Is a content delivery network service that securely delivers data,vedios, applications and API's to edge locations with high transfer speed.
- **CloudTrail:** Record all the API calls and save it to s3 bucket. It track youractivity and API usage.It provide event history of your account.
- **AWS Organization:** It helps you centrally manage and govorn multiple awsaccounts.you can grow all accounts to organisation unit (OU). Billing is centralised.

- **Amazon Inspector:** It is a security assesment service. It can scan vulnabitiesin your network and OS. It helps to improve security.
- **KMS:**Key Management Service creates and manage cryptographic keys. It is asecure service, it encrypt your datat to cloud by protecting your data.
- **AWS Trusted Advisor:** Trusted Advisor evaluates your account by using checks. These checks identify ways to optimize your AWS infrastructure, improve security and performance, reduce costs, and monitor service quotas.
- **AWS Config:** Record and evaluate configuration of your AWS resources.
- **AWS GuardDuty:** Amazon GuardDuty is a continuous security monitoring service. It is a thread detection service that continously monitors for maliciousactivities and unauthorized behaviour to protect your AWS environment
- **AWS WAF:** Web Applicartion Firewall helps you to protect your webapplications from common web exploits & attacks.
- **AWS WAF & AWS Shield:** It helps you to protect your AWS resource fromexploits and attacks.

AMI: (Amazon Machine Image) is a template with software configuration used tolaunch an instance.
Snapshot: Backup mechanism from volume is called snapshot.

GIT

GIT is the most widely used mordern Version Control System in the world today.GIT is mature & actively maintain opensource projects orginally developed by LINUS TORVALD in 2005.

Q: Why use git?

- It’s fast

- You dont need access to server.
- Everyone can use it.
- Amazingly good & mearging simultanously.

GIT COMMANDS:-

git init:- It is used to create a new blank repository. It is used to make an existingproject as a Git project.

git status:- It display the state of the repository and allows you to see the tracked &untracked files.

git add:- It adds a change in the working directory to the staging area. changes arenot actually recorded until you run **git** commit.

git commit:- It commits the staged snapshot to project history.

git config:- It is is used to set Git configuration values on a global or local projectlevel.

git log:- Show all previous commits/history.

git diff:- Show changes between two commits.

git branch:- It lets you create, list, rename, and delete branches in currentrepositories.

git push:- This command sync from local server to remote server changes goes to GitHub.

git pull:- This command fetches and merges changes on the remote server to yourlocal server (working directory).

git fetch:- It is similar to Pull but is doesnt merge i.e., it fetches (fetches=retrive) **git merge:-** It take the changes from staging area and merge (combine) it with repo**git checkout:-** Rollback your file to specific branch or switching to that specified branch.

git clone:- It copies an existing Git repository.

git revert:- Rollback your file after commiting & stores the history.

git reset:- Rollback your file without storing history or logs.

git remote:- This command let you create, view & delete connections to the remoterepository.

MAVEN

Maven is a build tool that automates Java code build process.

Maven phases:-

The representation of the default Maven Phases and its 8 steps are: Validate, Compile, Test, Package, Integration test, Verify, Install and Deploy.

1. Validate – Validate that the project is correct & all necessary information isavailable.

2.Compile:- Compile the source code of the project.

3. Test:- Test the compiled source code using a sutaible unit test framework.

4. Package:- Take the compiled code and package it in its distributable format, suchas a JAR.

6. Integration test:-

5. Verify:- Run the checks on the result of integration test to ensure quality criteriaare met.

6. Install:- Install the package into the local repository, for use as a dependency inother projects locally.

7. Deploy:- Done in the building environment, copies the final package to the remoterepository for sharing with other developers and projects

ANSIBLE MODULES

1. **Copy Module:-** Allows to push the file. Allows to copy file from source todestination.

2. **Template Module:-** It is similar to copy module, but it processes contentusing jinja2 extention.

3. **Lifeline Module:-** The lifeline module manages lines in a text file.
   - It ensures a particular line is in a file or replaces an existing line using aback-referenced regular expression.
   - It's primarily useful when you want to change just a single line in a file.

4. **User Module:-** User module is used to create new users in the targetmachines and also manage them.

5. **Package Module:-** The package module allows you to install, update, orremove software packages from your target system.

6. **Service Module:-** Service module is used to manage the services, Alsostart and enable the installed softwares.

7. **File module:-** File module is used to change the properties.
   - File modules also used for removing a file.
   - It is used for creating a link or creating a directory.
   - It is used for changing file permissions or ownership.
8. **Debug Module:-** The debug module prints statements during executionand can be useful for debugging variables or expressions

9. **Ping Module:-** Similar to ssh. Used to check the connection with our hostfile established or not.

10. **file module:-** Set attributes of files, symlinks or directories. Alternatively, remove files, symlinks or directories.

11. **Stat module:-** Retrieves facts for a file similar to the Linux/Unix ‘stat’ command.

12. **include_tasks module:-** Includes a file with a list of tasks to be executed in the current playbook.

13. **Apt_key module:-** Add or remove an *apt* key, optionally downloading it.

14. **Replace module:-** This module will replace all instances of a pattern within a file.

15. **Group module:-** Manage presence of groups on a host.

16. **add_host module:-** Use variables to create new hosts and groups ininventory for use in later plays of the same playbook.

17. **Selinux module:-** Configures the SELinux mode and policy. A reboot may be
required after usage.

18. **mount module:-** This module controls active and configured mountpoints in /etc/fstab.

19. **Yum Module:-** Yum module is used to install a service on centos.

20. **Archive Module:-** The archive module creates a compressed archive of one or more files. By default, it assumes the compression source exists onthe target.

21. **Apt Module:-** apt module is used to install services on ubuntu.

22. .**Setup module:-** The setup module is used when we want to see the information of all the hosts, their configuration, and detailed information.

23. **Include module:-** When we want to include another playbook in ourplaybook, then we use the Include module.

24. **include_vars module:-** Loads YAML/JSON variables dynamically from afile or directory, recursively, during task runtime.

25. **import_tasks module:-** Imports a list of tasks to be added to the currentplaybook for subsequent execution.

DOCKER

Q: Difference between VM & Container

VIRTUAL MACHINES	CONTAINER
• In VM's we have, hypervisor. Ex:- 1. Oracle VM Virtual Box. 2. Vmware server	• In Cotainerization, we have container run time environment. Ex:- 1. Docker 2. kubernetes
• VM's are portable but bulky • One VM is min 8Gb	• Containers are Lightweight.
• VM's need Operating System torun. • OS needs licensing	• Container doesnot need OS. • Containers uses host OS forcompute resource. They are just a process runningfron directory Isolated.
• It need resources for OS & requiremore storage.	• It uses less storage and lessresources.
• Time consumption to boot	• Less Time consumption.
• VM's are hardware Virtualization	• Containers are OS virtualization
• VM's need virtualization.	• Containers offers Isolation notvirtualization.
• VM's are expensive	·

Dockerfile Instruction :-

- **FROM =** Base Image
- **LABELS =** Adds metadata to an Image
- **RUN =** Execute commands in a new layer and commit the result.
- **ADD =** Adds files and folders into image
- **COPY =** Dump the file
- **CMD =** Runs binaries/commands on docker run.

- ENTRYPOINT = Allow you to comnfigure a container that will run as anexecutable
- VOLUME = Creates a mount point an d marks it as a holding externallymounted volume.
- EXPORT = Container listens on the specified network ports at runtime.
- ENV = Set the environment variable.
- USER = Set the username or UID.
- WORKDIR = Sets the working directory.

- ARG = Defines a variable that users can pass at build-time
- ONBUILD = Adds to the image a trigger instructions to be executed at a latertime.

KUBERNETES

Kubernetes was originally developed by Google in 2014.
Kubernetes is an opensource orchstration tool for managing containerized workloadand services
kubernetes support multiple run time environments. Kubernetes is portable,extensible, self-healing & it is reffered as k8s.

Each componenet of kubernetes:-

Kubectl:- is a command line configuration tool (CLI) for kubernetes used to interactwith Master node of kubernetes.

- Kubectl has a config file called kubeconfig, This file has the information aboutthe server and authentication information to access the API server.

Kubernetes has two main components :-

- Master Node
- Worker Node

Master Node:-

-- Master node is also called as controll plane
-- Master Node is a main node! which is reponsible for managing the entirekubernetes clusters. It handles the orchastration of worker nodes.
-- It has 4 components:-

1. API Server
2. etcd
3. Controller-manager
4. scheduler

1. API Server:-
   - Handle all the request & enable communication accross stack.
   - It is front-end of kubernetes control-plane.
   - Admins connect to it using kubectl CLI.
2. etcd:-
   - stores all the information.(how many container are running, their statuts , keys)
   - It should be backed up regularly.
   - Stores current state of everything in the clusterr
3. schedular:-
   - Watches the pods & assign pods to run on specific hosts
   - Decides on which worker node should the pod be running

- API stores the info of container in etcd, schedular will pickup that info &decides where pod should be placed in worker node.

4. Controller manager;-

- To reduce complexity, multiple srvices combine into one & run in a singleprocess
- The following controllers includes:-

  - Node controller:- Responsible for noticing & responding when the node goes down

  - Replication controller:- responsible for maintaining the correct number of pods for every replication controller in the system & keep monitoring

  - End point controller:- Populates the end point objects (that is joins services &pods together)

  - Service account & Tocken controller:- (Handle access management,) createdefault accounts & API access tocken for new namespace

- **Worker Node:-**

worker nodes are the nodes where the application actually running in kubernetescluster, It is also known as minion

These each worker nodes are controlled by the master node using kubelet processso, container platform must be running on each platform

Worker node has 3 components:-

1. Kubelet :
2. kube-proxy
3. Docker

1. kubelet :-

kubelet is an agent ,that runs on each node in the cluster. It makes sure thatcontainers are running in a Pod.

2. kube-proxy:-

--- kube-proxy is a network proxy that runs on each node in your cluster.
--- Kube-proxy maintains network rules on nodes.
--- Exposes your container to outside world.
--- These network rules allow network communication to your Pods from networksessions inside or outside of your cluster.

3. Docker:-

The first requirement of each node is Docker which helps in running the encapsulated application containers in a relatively isolated but lightweight operatingenvironment.

Objects of kubernetes:-

1. PODS
2. SERVICE
3. REPLICATION CONTROLLER
4. DEPLOYMENT

5. NAMESPACE

1. PODS:-

--Runs your apps isolated.

-- A pod represent proccesses running on ur cluster

-- But we always access pods, pods will handle containers.

-- Pod is basic execution unit of k8s application-- the smalest & simpliest unit in k8sobject model that u create or deploy.

Pods that runs a single container:-

-- The one-container-per-pod is most common use case in k8s.

-- pod as wrapped around single container.

-- K8s manages pods rather than the containers directly.

Pods that runs multi containers:-

-- Tightly coupled and need to share resources

-- One main container and others as side-cars or init containers

-- Each pod runs a single instance of a given application

-- Should use multi pods to scale horizontally

2. SERVICE

--> way to expose an application running on set of pods as a network service.

--> it is similar to loadbalncer

3. Replication controller:-

-->Pods maintained by replication controller are automatically replaced, if they areterminated or deleted , or failed.

--> if there are too many pods the rc deletes the extra pods

--> if there are too few rc starts the new more pods

--> if pod goes down, nodes will b migrated to healthy pod only when it is created byrc

4. Deployment:-

--> A deployment controller provides declarative updates for pods and replica sets.

--> Define desired state in a deployment . And deployment controller changes theactual state to desired state at a controlled rate

--> Deployment creates ReplicaSet to manage number of pods.

5. NAMESPACE:- set of resources

SETUP WITH KOPS

-- U need Domain for k8s DNS records

-- Create a linux & setup

- kops
- kubectl
- ssh keys
- awscli

-- login to AWS account & setup

- IAM user for awscli
- s3 bucket
- Route 53 Hosted zone

Kops commands to setup k8s cluster:-

- kops create cluster --name=saiteja.irrinki.xyz --state=s3://k8s-buckett --zone=eu-west-3a,eu-west-3b --node-count=2 –node-size=t2.micro --master- size=t2.micro --dns-zone=saiteja.irrinki.xyz --node-volume-size=8 --master-volume-size=8

(It will create configuration of kops)

- kops update cluster --name=saiteja.irrinki.xyz -- state=s3://k8s-buckett --yes --admin (It will create kopsdata in S3 buckett)(It start creating a cluster &

it takes 10 mins)

- kops validate cluster --name=saiteja.irrinki.xyz --state=s3://k8s-buckett (It shows ur cluster is ready)?& check inaws account.
- ls -a (it will create .kube cube directory, it will have config file ,&it is used by kubectl to access with API server).

JENKINS

Jenkins plugins:

s3publisher: This is a plugin to upload files to Amazon *S3* buckets.

Nexux Artifact Uploader: This plugin to upload the artifacts to Nexus Repository.

Copy Artifact: copies the artifact from another project.

SonarQuality scanner: Find bugs, vulnaribities.

SonarQube scanner: This plugin lets you centralize the configuration of SonarQube server connection details in Jenkins global configuration

Build pipeline: This plugin provides a build pipeline View of upstream anddownstream connected jobs that typically form a *build pipeline*.

Violation: This plugin detect the violations such as checstyle, pmd, cpd, findbugs.

http request: It is sanity test , it checks URL is up or not