* Hi, I am Krishna working as Consultant in Capgemini from the last 9 months. Total I have four years’ Experience on AWS and DevOps tools. In my previous project, I worked on AWS services such as (EC2, S3, VPC, Auto Scaling, ELB, IAM, RDS, SNS, Cloud watch and also, we used GitHub as a source code management tool, Jenkins as a CI automation server, Maven as the build tool. **Launching Amazon EC2 Cloud Instances using Amazon Web Services (Linux/ Ubuntu) and configuring launched instances with respect to specific applications.**
* **Defined AWS Security Groups which acted as virtual firewalls that controlled the traffic allowed reaching one or more AWS EC2 instances.**
* **Working on Multiple AWS instances, set the security groups, Elastic Load Balancer and AMIs, Auto scaling to design cost effective, fault tolerant and highly available systems.**
* **Configuring and Networking of Virtual Private Cloud (VPC).**
* **Creating S3 buckets and also managing policies for S3 buckets and Utilized S3 bucket and Glacier for storage and backup on AWS.**
* **Designed and deployed AWS solutions using E2C, S3, EBS, Elastic Load Configured S3 versioning and lifecycle policies to, backup files, and archive files in Glacier.**
* **Designed and deployed AWS solutions using E2C, S3, EBS, Elastic Load Balancer, Auto scaling groups, Opsworks.**
* **Hands-on experience in writing Jenkins to automate CI and CD.**
* **Responsible for Successful Release of an Application.**
* **Configured Jenkins Jobs install plugins for generating Project artifacts.**
* **Skills Profile Responsible for tagging and maintaining code on version control GIT.**
* **Responsible for creating Builds and Deployments.**
* **Automate the Build Process Using Jenkins jobs. Worked on creation of custom Docker container images, tagging and pushing the images.**

Day to Day activities:  
 1. Working on user access requests, tickets, and Service requests   
 3. Configuring AWS services based on client requirements.  
 4. Performing security checks

5.Support for pipeline failures  
 6. Pipeline maintenance and enhancements

7. Worked on Multi branch pipeline  
 8.Attending daily stand-ups and client calls.  
 past client J&J   
################################################################  
  
Pipeline flow is,  
Whenever developer checks in the code to Github, Jenkins webhook will trigger job to build the code using maven and code quality test is done using sonarqube and  
docker image for application is created using Docker file and the image is pushed to Docker hub registry and it is deployed on pods kubeAdm cluster.  
  
Stages in Pipeline:  
Setup - pull code from SCM  
Compile - mvn compile  
Static analysis - mvn sonar:sonar  
Unit testing - mvn test  
Artifact push - artifacts will be pushed to Jfrog artifactory  
Docker Image - creating docker image using docker file and pushing it to docker hub.  
Deploy - Deploy the app in pods along with middleware.  
Post-Deploy - Run checks to validate app services.

S3

Amazon S3 is easy-to-use object storage with a simple web service interface that you can use to store and retrieve any amount of data from anywhere on the web. which is Secure, durable, and highly-scalable cloud storage Amazon, S3 provides set of permissions, access controls, and encryption. We can create 100 bucket per account and Objects can range in size from 0 bytes up to 5TB.

Amazon S3 object can be addressed by a unique URL formed using the web services endpoint, the bucket name, and the object key. It provides

Backup and archive for on-premises or cloud data,Content, media, and software storage and distribution,Big data analytics,Static website hosting,Cloud-native mobile and Internet application hosting, Disaster recovery

STORAGE CLASSESS;

Amazon S3 Reduced Redundancy Storage (RRS) offers slightly lower durability (4 nines)than Standard or Standard-IA at a reduced cost. It is most appropriate for derived data thatcan be easily reproduced, such as image thumbnails.

STANDARD CLASS; Amazon S3 Standard offers high durability, high availability, low latency, and high performance object storage for general purpose use. Which is used for both frequently access , infrequently access. And long term storage.

Infrequent Access (Standard-IA) offers the same durability, low latency, and high throughput as Amazon S3 Standard, but is designed for long-lived, less frequently accessed data.it is best suited for infrequently accessed data that is stored for longer than 30 days.

Amazon Glacier;

offers secure, durable, and extremely low-cost cloud storage for data that does not require real-time access, such as archives and long-term backups. To keep costs low, Amazon Glacier is optimized for infrequently accessed data.

LIFE CYCLY RULE; Using Amazon S3 lifecycle configuration rules, it can significantly reduce your storage costs by automatically transitioning data from one storage class to another or even automatically deleting data after a period of time.

VERSIONING; Amazon S3 versioning helps protects your data against accidental or malicious deletion by Keeping Versioning allows you to preserve, retrieve, and restore every version of object stored in your S3 bucket. ping multiple versions of each object in the bucket identified by a unique version ID.

ENCRYPTION;

Client-Side Encryption; Client-side encryption refers to encrypting data on the client side of your application before Sending it to Amazon S3.

Multipart Upload

To better support uploading or copying of large objects, Amazon S3 provides the Multipart U multipart upload for objects larger than 5GB pload API. In general, you should use multipart upload for objects larger than 100 Mbytes.

Cross-Region Replication

Cross-region replication is a feature of S3 that allows you to asynchronously replicate all new objects in the source bucket in one region to a target bucket in another region. Cross-region replication is commonly used to reduce the latency required to access objects. cross-region replication will only replicate new objects. Existing objects will not be replicated.

Amazon Glacier

Amazon Glacier is Offers an extremely low-cost storage service that provides durable, secure, and flexible storage for data archiving and online backup.it is designed for infrequently accessed data where a retrieval time of three to five hours is acceptable.

EC2

w the steps provided below to recover an EC2 instance if you have lost the key:

1. Verify that the EC2Config service is running
2. Detach the root volume for the instance
3. Attach the volume to a temporary instance
4. Modify the configuration file
5. Restart the original instanc

AMI; The Amazon Machine Image (AMI) defines the initial software that will be on an instance when it is launched. —AWS publishes AMIs with versions of many different OSs, both Linux and Windows —An AMI can be created from an existing Amazon EC2 instance

Elastic IP—An elastic IP address is an address unique on the Internet that you reserve independently and associate with an Amazon EC2 instance. This IP address persists until the customer releases.

KEY-PAIRS;

AWS stores the public key, and the private key is kept by the customer Amazon EC2 uses public-key cryptography to encrypt and decrypt login information, Public Key cryptography uses a public key to encrypt a piece of data and an associated private key to decrypt the data. These two keys together are called a key pair. stored path is /.ssh/authorized\_keys.

SECURITY KEYS: AWS allows you to control traffic in and out of your instances through virtual firewalls called security groups. Security groups allow you to control traffic based on port, protocol, and source/destination.

CIDR block—An x.x.x.x/x style definition that defines a specific range of IP addresses.

STATELESS AND STASTE FULL

Stateless: This means any changes applied to an incoming rule will not be applied to the outgoing rule.

A placement group is a logical grouping of instances within a single Availability Zone.Placement groups enable applications to participate in a low-latency, 10 Gbps network.Placement groups are recommended for applications that benefit from low network latency,high network throughput, or both. Remember that this represents network connectivity between instances. To fully use this network performance for your placement group, choosean instance type that supports enhanced networking and 10 Gbps network perf.

**Reserved Instances**

the term commitment is the duration of the reservation and can be either one or three years.The longer the commitment, the bigger the discount. customers can save up to 75 percent over the on-demand hourly rate.

The price per hour for each instance type published on the AWS website represents the price for On-Demand Instances.

**Instance Stores**

An instance store (sometimes referred to as ephemeral storage) provides temporary blockLevel storage for your instance. This storage is located on disks that are physically attached to the host computer. Storage will be lost onces The underlying disk drive fails. The instance stops (the data will persist if an instance reboots). The instance terminates.

**Elastic Block Store Basics**

Amazon EBS provides persistent block-level storage volumes for EC2instances. Each EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Multiple EBS volumes can be attached to a single EC2 instance, although a volume can only be attached to a single instance at a time.

Provisioned IOPS SSD

Provisioned IOPS SSD volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads that are sensitive to storage performance and consistency inrandom access I/O throughput.Provisioned IOPS SSD volume can range in size from 4 GB to 16 TB. Large database workloads Critical business applications that require sustained IOPS performance

General-purpose SSD volumes offer cost-effective storage that is ideal for a broad range ofworkloads A general-purpose SSD volume can range in size from 1 GB to 16 TB and provides a baseline performance of three IOPS per gigabyte provisioned, capping at 10,000 IOP.

SNAPSHOTS;

ctivity- ip route show, ifconfig

Why EC2 Connect is not working? **EC2 Instance Connect doesn't support the OS distribution,** The EC2 Instance Connect package isn't installed on the instance. There are missing or incorrect AWS Identity and Access Management (IAM) policies or permissions.

AWS CloudTrail **enables auditing, security monitoring, and operational troubleshooting by tracking user activity and API usage**. You can identify wHow long are CloudTrail logs kept?

90 days

Although CloudTrail provides **90 days** of event history information for management events in the CloudTrail console without creating a trail, it is not a permanent record

ho or what took which action, what resources were acted upon, when the event occurred,

There are **three types of events** that can be logged in CloudTrail: management events, data events, and CloudTrail Insights events

What is a Read replica?

A read replica is **a copy of the primary instance that reflects changes to the primary in almost real time, in normal circumstances**. You can use a read replica to offload read requests or analytics traffic from the primary instance. Additionally, for disaster recovery, you can perform a regional migration.

Read Replica **helps in decreasing load on the primary DB by serving read-only traffic**. A Read Replica can be manually promoted as a standalone database instance.

To replace a lost key pair, you can use the AWS Systems Manager AWS Support-Reset Access Automation document. Or, you can **create an Amazon Machine Image (AMI) of the existing instance, launch a new instance, and then select a new key pair**.

IAM

What is IAM policy vs role?

IAM Roles vs. Policies. **IAM Roles manage who has access to your AWS resources, whereas IAM policies control their permissions**. A Role with no Policy attached to it won't have to access any AWS resources

An IAM role is an IAM entity that defines a set of permissions for making AWS service requests, while an IAM user has permanent long-term credentials and is used to interact with the AWS services directly.

LOAD BALANCER

Elastic Load Balancing automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Each target group routes requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specif

you can use the sticky session feature (also known as session affinity) to enable the load balancer to bind a user's session to a specific target.

Identity-based policies can be managed or inline. Resource-based policies are attached to a resource.

VPC

A VPC endpoint **allows you to privately connect your VPC to supported AWS services.**

A bastion host is **a dedicated server that lets authorized users access a private network from an external network such as the internet.**

AWS Direct Connect is **a network service that provides an alternative to using the Internet to utilize AWS cloud services.**

Gateway endpoints **provide reliable connectivity to Amazon S3 and DynamoDB without requiring an internet gateway or a NAT device for your VPC**

A network access control list (ACL) allows or denies specific inbound or outbound traffic at the subnet level.

NAT device to allow resources in private subnets to connect to the internet, other VPCs, or on-premises networks.

You launch a NAT instance in a public subnet to enable instances in the private subnet to initiate outbound IPv4 traffic to the internet or other AWS services

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC.

* Security groups - This acts as a firewall for the EC2 instances, controlling inbound and outbound traffic at the instance level.
* Network access control lists - It acts as a firewall for the subnets, controlling inbound and outbound traffic at the subnet level.

Flow logs - These capture the inbound and outbound traffic from the netThe AWS Resources owner is identical to an Administrator User. The Administrator User can build, change, delete, and inspect resources, as well as grant permissions to other AWS users.

* Administrator Access without the ability to control users and permissions is provided to a Power User. A Power User Access user cannot provide permissions to other users but has the ability to modify, remove, view, and create resources.work interfaces in your VPC.

AWS WAF or AWS Web Application Firewall protects your web applications from web exploitations. It helps you control the traffic flow to your applications. With WAF, you can also create custom rules that block common attack patterns. It can be used for three cases: allow all requests, prevent all requests, and count all requests for a new policy.

Connection Draining is an AWS service that allows us to serve current requests on the servers that are either being decommissioned or updated.

**ECS**

When the rolling update (ECS) deployment type is used for your service, when a new service deployment is started the Amazon ECS service scheduler replaces the currently running tasks with new tasks.

A task definition is required to run Docker containers in Amazon ECS

An Amazon ECS cluster is a logical grouping of tasks or services. Your tasks and services are run on infrastructure that is registered to a cluster.

Amazon ECS capacity providers are used to manage the infrastructure the tasks in your clusters use. Each cluster can have one or more capacity providers and an optional default capacity provider strategy. The capacity provider strategy determines how the tasks are spread across the cluster's capacity providers.

ROUTE53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking.

**Check the health of your resources**

Route 53 sends automated requests over the internet to a resource, such as a web server, to verify that it's reachable, available, and functional. You also can choose to receive notifications when a resource becomes unavailable and

**DNS resolver** When you open a browser and enter a domain name in the address bar, your query goes first to a DNS resolver. The resolver communicates with DNS name servers to get the IP address for the corresponding resource, such as a web serverchoose to route internet traffic away from unhealthy resources.

**DNS query**The device that initiated the request uses the IP address to communicate with the resource.

**alias record**

A type of record that you can create with Amazon Route 53 to route traffic to AWS resources such as Amazon CloudFront distributions and Amazon S3 buckets

**CNAME records**

A CNAME record can redirect DNS queries to any DNS record. For example, you can create a CNAME record that redirects exp.com to expo.com

Hosted zone

A hosted zone is a container that holds information about how you want to route traffic on the internet for a specific domain. For example, lms.simplilearn.com is a hosted zone.

**Simple routing policy** – Use to route internet traffic to a single resource that performs a given function for your domain,

* **Geolocation routing policy** – Use when you want to route internet traffic to your resources based on the location of your users.
* **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
* **Latency routing policy** – Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency
* **IP-based routing policy** – Use when you want to route traffic based on the location of your users, and have the IP addresses that the traffic originates from.
* **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
* **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

CLOUD FORMATION

You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and CloudFormation takes care of provisioning and configuring those resources for you.

TEMPLATE-

A template describes all your resources and their properties

AWSLAMBDA

Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, you can run code for virtually any type of application or backend service

To transfer terabytes of data outside and inside of the AWS environment, a small application called SnowBall is used.

Data transferring using SnowBall is done in the following ways:

1. A job is created.
2. The SnowBall application is connected.
3. The data is copied into the SnowBall application.
4. Data is then moved to the AWS S3.

RDS maintenance window lets you decide when DB instance modifications, database engine version upgrades, and software patching have to occur. The automatic scheduling is done only for patches that are related to security and durability

**RDS PORT NUMBER;**

|  |  |
| --- | --- |
| **Database Engine** | **Default Port Number** |
| Aurora/MySQL/MariaDB | 3306 |
| PostgreSQL | 5432 |
| Oracle | 1521 |
| SQL Server | 1433 |

**TERRAFORM**

The provider block configures the specified provider, in this case aws. A provider is a plugin that Terraform uses to create and manage your resources.

Use resource blocks to define components of your infrastructure. A resource might be a physical or virtual component such as an EC2 instance,R esource type,name. Resource blocks contain arguments which you use to configure the resource

A Terraform module is **a set of Terraform configuration files in a single directory**. Even a simple configuration consisting of a single directory with one or more .tf files is a module.

A data block **requests that Terraform read from a given data source ("aws\_ami") and export the result under the given local name** ("example"). The name is used to refer to this resource from elsewhere in the same Terraform module, but has no significance outside of the scope of a module.

When you run Terraform commands directly from such a directory, it is considered the root module.

**Terraform stores information about your infrastructure in a state file**. This state file keeps track of resources created by your configuration and maps them to real-world resources.

What are the two types of backends?

Backend Types. At a very high level, there are two types of backends: **Local — where the state file is stored on the local filesystem.** **Remote — where the state file is stored in some remote filesystem or database.**

**Plan - Preview changes before applying.**

**Apply - Provision reproducible infrastructure**.

inspect the complete state

use the `terraform show` command.

What is TF state file?

This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures. This state is stored by default in a local file named "**terraform.** **tfstate**"

 terraform taint command informs Terraform that a particular object has become degraded or damaged. Terraform represents this by marking the object as "tainted" in the Terraform state, and Terraform will propose to replace it in the next plan you create

How do you find Terraform syntax?

The **terraform validate command** is used to validate the syntax of the terraform files. Terraform performs a syntax check on all the terraform files in the directory, and will display an error if any of the files doesn't validate. This command does not check formatting (e.g. tabs vs spaces, newlines, comments etc.).

Command login – ssh -i [terraform.pem ubuntu@ip](mailto:terraform.pem@ip) address

Curl http;//checkip.amazon.com

A module is **a container for multiple resources that are used together**. Every Terraform configuration has at least one module, known as its root module.

The backend configuration specifies where a state file is stored, in other words where Terraform tracks created resources. For real infrastructures, the backend configuration should instead be on secure storage with high durability and backup capabilities, such as being on an S3 bucket.

* [alias, for using the same provider with different configurations for different resources](https://www.terraform.io/language/providers/configuration#alias-multiple-provider-configurations)

The terraform graph command is used to generate a visual representation of either a configuration or execution plan. The output is in the DOT format, which can be used by [GraphViz](http://www.graphviz.org/) to generate charts

The terraform import command [imports existing resources](https://www.terraform.io/cli/import) into Terraform.

The terraform output command is used to extract the value of an output variable from the state file.

The null\_resourceThe null\_resource resource **implements the standard resource lifecycle but takes no further action**. **helps us to execute any command remotely or locally to provisioning any resource or even to create a configuration file or perform some command or scripts to change some configuration**.

Module 5: Terraform Registry The Terraform Registry is a repository of modules written by the Terraform community. The registry can help you get started with Terraform more quickly

The terraform output command is used to extract the value of an output variable from the state file.

.2 Introducing Terraform Provisioners Provisioners are used to execute scripts on a local or remote machine as part of resource creation or destruction. local-exec provisioners allow us to invoke a local executable after the resource is created. One of the most used approaches of local-exec is to run ansible-playbooks on the created server after the resource is created

Remote-exec provisioners allow invoking scripts directly on the remote server

Beanstalk

**Rolling updates** occur when you change settings that require new Amazon EC2 instances to be provisioned for your environment

NANO-EDITOR COMMANDS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **File handling**   |  |  | | --- | --- | | Ctrl+S | Save current file | | Ctrl+O | Offer to write file ("Save as") | | Ctrl+R | Insert a file into current one | | Ctrl+X | Close buffer, exit from nano |   **Editing**   |  |  | | --- | --- | | Ctrl+K | Cut current line into cutbuffer | | Alt+6 | Copy current line into cutbuffer | | Ctrl+U | Paste contents of cutbuffer | | Alt+T | Cut until end of buffer | | Ctrl+] | Complete current word | | Alt+3 | Comment/uncomment line/region | | Alt+U | Undo last action | | Alt+E | Redo last undone action |   **Search and replace**   |  |  | | --- | --- | | Ctrl+Q | Start backward search | | Ctrl+W | Start forward search | | Alt+Q | Find next occurrence backward | | Alt+W | Find next occurrence forward | | Alt+R | Start a replacing session |   **Deletion**   |  |  | | --- | --- | | Ctrl+H | Delete character before cursor | | Ctrl+D | Delete character under cursor | | Alt+Bsp | Delete word to the left | | Ctrl+Del | Delete word to the right | | Alt+Del | Delete current line |   **Operations**   |  |  | | --- | --- | | Ctrl+T | Execute some command | | Ctrl+J | Justify paragraph or region | | Alt+J | Justify entire buffer | | Alt+B | Run a syntax check | | Alt+F | Run a formatter/fixer/arranger | | Alt+: | Start/stop recording of macro | | Alt+; | Replay macro | | **Moving around**   |  |  | | --- | --- | | Ctrl+B | One character backward | | Ctrl+F | One character forward | | Ctrl+**←** | One word backward | | Ctrl+**→** | One word forward | | Ctrl+A | To start of line | | Ctrl+E | To end of line | | Ctrl+P | One line up | | Ctrl+N | One line down | | Ctrl+**↑** | To previous block | | Ctrl+**↓** | To next block | | Ctrl+Y | One page up | | Ctrl+V | One page down | | Alt+\ | To top of buffer | | Alt+/ | To end of buffer |   **Special movement**   |  |  | | --- | --- | | Alt+G | Go to specified line | | Alt+] | Go to complementary bracket | | Alt+**↑** | Scroll viewport up | | Alt+**↓** | Scroll viewport down | | Alt+< | Switch to preceding buffer | | Alt+> | Switch to succeeding buffer |   **Information**   |  |  | | --- | --- | | Ctrl+C | Report cursor position | | Alt+D | Report line/word/character count | | Ctrl+G | Display help text |   **Various**   |  |  | | --- | --- | | Alt+A | Turn the mark on/off | | Tab | Indent marked region | | Shift+Tab | Unindent marked region | | Alt+V | Enter next keystroke verbatim | | Alt+N | Turn line numbers on/off | | Alt+P | Turn visible whitespace on/off | | Alt+X | Hide or unhide the help lines | | Ctrl+L | Refresh the screen | |

**LINUX COMMANDS**

**File Commands**

**List files** in the directory:

ls

**List all files** ([shows hidden files](https://phoenixnap.com/kb/show-hidden-files-linux)):

ls -a

[Show directory](https://phoenixnap.com/kb/pwd-linux) you are currently working in:

pwd

[Create a new directory](https://phoenixnap.com/kb/create-directory-linux-mkdir-command):

mkdir [directory]

[Remove a file](https://phoenixnap.com/kb/how-to-remove-files-directories-linux-command-line):

rm [file\_name]

**Remove a directory** recursively:

rm -r [directory\_name]

**Recursively remove a directory** without requiring confirmation:

rm -rf [directory\_name]

[Copy the contents of one file](https://phoenixnap.com/kb/how-to-copy-files-directories-linux) to another file:

cp [file\_name1] [file\_name2]

**Recursively copy the contents of one file** to a second file:

cp -r [directory\_name1] [directory\_name2]

**Rename** **[file\_name1]** to **[file\_name2]** with the command:

mv [file\_name1] [file\_name2]

[Create a symbolic link](https://phoenixnap.com/kb/symbolic-link-linux) to a file:

ln -s /path/to/[file\_name] [link\_name]

Create a **new file** using [touch](https://phoenixnap.com/kb/touch-command-in-linux):

touch [file\_name]

**Show the contents** of a file:

more [file\_name]

or use the [**cat** command](https://phoenixnap.com/kb/linux-cat-command):

cat [file\_name]

Append file contents to another file:

cat [file\_name1] >> [file\_name2]

Display the **first 10 lines** of a file with [head command](https://phoenixnap.com/kb/linux-head):

head [file\_name]

Show the **last 10 lines** of a file:

tail [file\_name]

**Encrypt** a file:

gpg -c [file\_name]

**Decrypt** a file:

gpg [file\_name.gpg]

Show the **number of words, lines, and bytes** in a file using [wc](https://phoenixnap.com/kb/wc-linux" \t "_blank):

wc

List number of lines/words/characters in each file in a directory with [the xargs command](https://phoenixnap.com/kb/xargs-command):

ls | xargs wc

[Cut a section of a file](https://phoenixnap.com/kb/linux-cut) and print the result to standard output:

cut -d[delimiter] [filename]

Cut a section of piped data and print the result to standard output:

[data] | cut -d[delimiter]

[Print all lines matching a pattern](https://phoenixnap.com/kb/awk-command-in-linux) in a file:

awk '[pattern] {print $0}' [filename]

[Overwrite a file](https://phoenixnap.com/kb/shred-linux) to prevent its recovery, then delete it:

shred -u [filename]

[Compare two files](https://phoenixnap.com/kb/linux-diff) and display differences:

diff [file1] [file2]

[Read and execute the file content](https://phoenixnap.com/kb/linux-source-command) in the current shell:

source [filename]

[Sort file contents](https://phoenixnap.com/kb/linux-sort) and print the result in standard output:

sort [options] filename

[Store the command output in a file](https://phoenixnap.com/kb/linux-tee) and skip the terminal output:

[command] | tee [filename] >/dev/null

**File Compression**

Archive an existing file:

tar cf [compressed\_file.tar] [file\_name]

[Extract an archived file](https://phoenixnap.com/kb/extract-tar-gz-files-linux-command-line#htoc-using-tar-utility):

tar xf [compressed\_file.tar]

Create a **gzip compressed tar file** by running:

tar czf [compressed\_file.tar.gz]

**Compress** a file with the **.gz** extension:

gzip [file\_name]

**File Transfer**

Copy a file to a server directory securely using the [Linux scp command](https://phoenixnap.com/kb/linux-scp-command):

scp [file\_name.txt] [server/tmp]

**Synchronize** the contents of a directory **with a backup directory** using the [rsync command](https://phoenixnap.com/kb/rsync-command-linux-examples" \t "_blank):

rsync -a [/your/directory] [/backup/]

**Users and Groups**

See details about the **active users**:

id

Show**last system logins**:

last

Display who is**currently logged into the system** with the [who command](https://phoenixnap.com/kb/linux-who-command):

who

Show which users are**logged in** and **their activity**:

w

**Add a new group** by typing:

groupadd [group\_name]

Add a**new user**:

adduser [user\_name]

Add a **user to a group**:

usermod -aG [group\_name] [user\_name]

Temporarily **elevate user privileges** to superuser or root using the [sudo command](https://phoenixnap.com/kb/linux-sudo-command" \t "_blank):

sudo [command\_to\_be\_executed\_as\_superuser]

**Delete** a user:

userdel [user\_name]

[Modify user information](https://phoenixnap.com/kb/usermod-linux) with:

usermod

[Change directory group](https://phoenixnap.com/kb/chgrp-command):

chgrp [group-name] [directory-name]

**Package Installation**

[List all installed packages](https://phoenixnap.com/kb/how-to-list-installed-packages-on-centos) with **yum**:

yum list installed

Find a package by a **related keyword**:

yum search [keyword]

Show**package information and summary**:

yum info [package\_name]

Install a package using the **YUM package manager**:

yum install [package\_name.rpm]

Install a package using the **DNF package manager**:

dnf install [package\_name.rpm]

Install a package[using the **APT package manager**](https://phoenixnap.com/kb/how-to-use-apt-get-commands):

apt install [package\_name]

**Install** an **.rpm** package from a local file:

rpm -i [package\_name.rpm]

**Remove** an **.rpm** package:

rpm -e [package\_name.rpm]

Install software from **source code**:

tar zxvf [source\_code.tar.gz]

cd [source\_code]

./configure

make

make install

**Process Related**

See a **snapshot of active processes**:

ps

Show **processes in a tree-like diagram**:

pstree

Display a **memory usage map** of processes:

pmap

See [all running processes](https://phoenixnap.com/kb/top-command-in-linux):

top

[Terminate a Linux process](https://phoenixnap.com/kb/how-to-kill-a-process-in-linux) under a**given ID**:

kill [process\_id]

Terminate a process under a **specific name**:

pkill [proc\_name]

Terminate all processes **labelled** **“proc”**:

killall [proc\_name]

**List and resume stopped jobs** in the background:

bg

Bring the most **recently suspended job to the** **foreground**:

fg

Bring a **particular job to the** **foreground**:

fg [job]

List **files opened by running processes** with [lsof command](https://phoenixnap.com/kb/lsof-command" \t "_blank):

lsof

[Catch a system error signal](https://phoenixnap.com/kb/bash-trap-command) in a shell script:

trap "[commands-to-execute-on-trapping]" [signal]

[Pause terminal or a Bash script](https://phoenixnap.com/kb/bash-wait-command) until a running process is completed:

wait

[Run a Linux process](https://phoenixnap.com/kb/linux-nohup) in the background:

nohup [command] &

**System Management and Information**

Show **system information**:

uname -r

See [kernel release information](https://phoenixnap.com/kb/check-linux-kernel-version):

uname -a

Display **how long the system has been running**, including load average:

uptime

See system **hostname**:

hostname

Show the**IP address** of the system:

hostname -i

List system **reboot history**:

last reboot

See [current time and date](https://phoenixnap.com/kb/linux-date-command):

date

Query and **change the system clock** with:

timedatectl

Show current **calendar** (month and day):

cal

[List logged in users](https://phoenixnap.com/kb/w-command-in-linux):

w

[See which **user you are using**](https://phoenixnap.com/kb/whoami-linux):

whoami

Show **information about a particular user**:

finger [username]

[View or limit](https://phoenixnap.com/kb/ulimit-linux-command) system resource amounts:

ulimit [flags] [limit]

[Schedule a system shutdown](https://phoenixnap.com/kb/linux-shutdown-command):

shutdown [hh:mm]

Shut Down the system immediately:

shutdown now

[Add a new kernel module](https://phoenixnap.com/kb/modprobe-command):

modprobe [module-name]

**Disk Usage**

You can use the df and du commands to [check disk space in Linux](https://phoenixnap.com/kb/linux-check-disk-space).

See **free and used space** on mounted systems:

df -h

Show **free inodes** on mounted filesystems:

df -i

Display **disk partitions, sizes, and types** with the command:

fdisk -l

See [**disk usage** for all files and directory](https://phoenixnap.com/kb/show-linux-directory-size):

du -ah

Show**disk usage of the directory** you are currently in:

du -sh

Display**target mount point** for all filesystem:

findmnt

[**Mount a device**](https://phoenixnap.com/kb/linux-mount-command):

mount [device\_path] [mount\_point]

**SSH Login**

**Connect to host** as user:

ssh user@host

Securely **connect to host via SSH** default port 22:

ssh host

Connect to host **using a particular port**:

ssh -p [port] user@host

Connect to host **via telnet default port 23**:

telnet host

**File Permission**

[Chown command in Linux](https://phoenixnap.com/kb/linux-chown-command-with-examples) changes file and directory ownership.

Assign **read, write, and execute permission** to everyone:

chmod 777 [file\_name]

Give **read, write, and execute permission to owner**, and r**ead and execute permission to group and others**:

chmod 755 [file\_name]

Assign **full permission to owner**, and **read and write permission to group and others**:

chmod 766 [file\_name]

Change the **ownership of a file**:

chown [user] [file\_name]

Change the **owner and group ownership of a file**:

chown [user]:[group] [file\_name]

**Network**

[List IP addresses](https://phoenixnap.com/kb/linux-ip-command-examples)and **network interfaces**:

ip addr show

Assign an **IP address to interface eth0**:

ip address add [IP\_address]

Display **IP addresses of all network interfaces** with:

ifconfig

See **active (listening) ports** with the [netstat command](https://phoenixnap.com/kb/netstat-command):

netstat -pnltu

Show **tcp** and **udp** **ports** and their programs:

netstat -nutlp

Display more **information about a domain**:

whois [domain]

Show **DNS information**about a domain using the [dig command](https://phoenixnap.com/kb/linux-dig-command-examples):

dig [domain]

Do a **reverse lookup** **on domain**:

dig -x host

Do **reverse lookup of an IP address**:

dig -x [ip\_address]

Perform an **IP lookup for a domain**:

host [domain]

Show the **local IP address**:

hostname -I

**Download a file** from a domain using the **[wget](https://phoenixnap.com/kb/wget-command-with-examples" \t "_blank)**[command](https://phoenixnap.com/kb/wget-command-with-examples" \t "_blank):

wget [file\_name]

Receive [information about an internet domain](https://phoenixnap.com/kb/nslookup-command):

nslookup [domain-name]

[Save a remote file to your system](https://phoenixnap.com/kb/curl-command) using the filename that corresponds to the filename on the server:

curl -O [file-url]

**Variables**

[Assign an integer value](https://phoenixnap.com/kb/bash-let) to a variable:

let "[variable]=[value]"

[Export a Bash variable](https://phoenixnap.com/kb/bash-export-variable):

export [variable-name]

[Declare a Bash variable](https://phoenixnap.com/kb/bash-declare):

declare [variable-name]= "[value]"

List the names of [all the shell variables and functions](https://phoenixnap.com/kb/linux-set):

set

[Display the value](https://phoenixnap.com/kb/echo-command-linux) of a variable:

echo $[variable-name]

**Shell Command Management**

[Create an alias](https://phoenixnap.com/kb/linux-alias-command) for a command:

alias [alias-name]='[command]'

[Set a custom interval](https://phoenixnap.com/kb/linux-watch-command) to run a user-defined command:

watch -n [interval-in-seconds] [command]

[Postpone the execution](https://phoenixnap.com/kb/linux-sleep) of a command:

sleep [time-interval] && [command]

Create a job to be executed at a certain time (**Ctrl+D** to exit prompt after you type in the command):

at [hh:mm]

[Display a built-in manual](https://phoenixnap.com/kb/linux-man) for a command:

man [command]

Print the history of the commands you used in the terminal:

history

**Linux Keyboard Shortcuts**

**Kill process** running in the terminal:

Ctrl + C

Stop**current process**:

Ctrl + Z

The process can be **resumed** in the **foreground** with **fg** or in the **background** with **bg**.

Cut **one word before the cursor** and add it to clipboard:

Ctrl + W

Cut **part of the line before the cursor** and add it to clipboard:

Ctrl + U

Cut **part of the line after the cursor** and add it to clipboard:

Ctrl + K

**Paste** from clipboard:

Ctrl + Y

**Recall last command** that matches the provided characters:

Ctrl + R

**Run** the previously recalled command:

Ctrl + O

**Exit command history** without running a command:

Ctrl + G

**Run the last command** again:

!!

**Log out** of current session:

exit

**Jenkins**

1.A build parameter **allows us to pass data into our Jenkins jobs**. Using build parameters, we can pass any data we want: git branch name, secret credentials, hostnames and ports, and so on. Any Jenkins job or pipeline can be parameterized.

1. [Continuous Integration](https://www.simplilearn.com/tutorials/devops-tutorial/continuous-integration) is a development practice where the codes can be integrated into a shared repository.

Continuous Delivery (CD) refers to the building, testing, and delivering improvements to the software code.

1. Continuous Deployment (CD) is the ultimate stage in the DevOps pipeline. It  refers to automatic release of any developer changes from the repository to the production stage.
2. You may move a job from one Jenkins installation to another just by copying the corresponding job directory.
3. You may make a copy of an already existing job by making a clone of the job directory with an uncommon name.
4. You may also just rename a current job by renaming a directory.
5. JENKINS\_HOME directory is the place where all the settings, logs, and configurations. are stored. It stores all this information in XML files
6. The workspace directory is **where Jenkins builds our project**:
7. **There is a way to cleanup workspace in Jenkins-work space plugin**
8. This build step **enables your Jenkins job to run a Command Workflows in Commander**.
9. Jenkins Pipeline (or simply Pipeline with a capital P) is a suite of plugins that supports implementing and integrating continuous delivery pipelines into Jenkins. This **allows you to automate the process of getting software from version control through to your users and customers**.
10. **Declarative pipelines break down stages into multiple steps, while in scripted pipelines there is no need for this**
11. **Schedule jobs in build triger section -use the cron expression five artisticks**
12. **Periodically polls the SCM to check whether changes were made (commits), and triggers the job if new commits where pushed.**

**DOCKER**

**Docker file—set instruction which is used to build the images and run as a container**

**Docker file commands- 1from- basic image, 2.Run- it creating run shell commands, 3.CMD-** **instruction sets the command to be executed when running the image.**

**4.** **LABEL** **instruction adds metadata to an image. A LABEL is a key-value pair**

**5.** **The MAINTAINER instruction sets the Author field of the generated images**.

6.**The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime**

**7.** The ENV instruction sets the environment variable <key> to the value <value>

8.The ADD instruction copies new files, directories or remote file URLs from <src> and adds them to the filesystem of the image at the path <dest>.

9. The COPY instruction copies new files or directories from <src> and adds them to the filesystem of the container at the path <dest>

10. An ENTRYPOINT allows you to configure a container that will run as an executable.

11. The VOLUME instruction creates a mount point with the specified name and marks it as holding externally mounted volumes from native host or other containers.

12. he WORKDIR instruction sets the working directory for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions that follow it in the Dockerfile.

13. The ARG instruction defines a variable that users can pass at build-time to the builder with the docker build command using the --build-arg <varname>=<value> flag.

14. The ONBUILD instruction adds to the image a trigger instruction to be executed at a later time, when the image is used as the base for another build.

15. The STOPSIGNAL instruction sets the system call signal that will be sent to the container to exit.

Docker-COMMANDS

-DOCKER ATTACH-docker attach [OPTIONS] CONTAINER

-DOCKER BUILD-The command builds Docker images from a Dockerfile and a “context”.

-DOCKER CHECKPOINT-heckpoint and Restore is an experimental feature that allows you to freeze a running container by checkpointing it, which turns its state into a collection of files on disk.

-DOCKER COMPOSE-Y to build and manage multiple services in Docker containers.

You can supply multiple -f configuration files. When you supply multiple files, Compose combines them into a single configuration.

-DOCKER CREATE- (or shorthand: docker create) command creates a new container from the specified image, without starting it.

-DOCKER DIFF-List the changed files and directories in a container᾿s filesystem since the container was created

-DOCKER EXEC-The docker exec command runs a new command in a running container.

DOCKER EXPORT-docker export will export the contents of the underlying directory, not the contents of the volume.

DOCKER INFO-This command displays system wide information regarding the Docker installation.

DOCKER INSPECT-Docker inspect provides detailed information on constructs controlled by Docker.

DOCKER KILL-Kill one or more running containers

DOCKER LOGS-The docker logs command batch-retrieves logs present at the time of execution.

DOCKER NETWORK- Manage networks. You can use subcommands to create, inspect, list, remove, prune, connect, and disconnect networks.

DOCKER PAUSE -The docker pause command suspends all processes in the specified containers.

DOCKER TAG -Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE

DOCKER WAIT-docker wait, which should block until the container exits.

**GIT**

The clone command in git is used when you want to download an existing git repository to your local computer.

**2 git pull**

When you want to take changes or updates done by other developer/team member on git repository, you have to use git pull.

In detail git pull is the command that fetches the content from a remote repository and integrates it with the local repository/branch. It is, in actuality, a combination of git fetch and git merge called in that order.

**3 git fetch**

Git "fetch" Downloads commits, objects and refs from another repository. It fetches branches and tags from one or more repositories.

The git stash command **takes your uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from your working copy.**

To "squash" in Git means **to combine multiple commits into one**.

**A conflict arises when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other**