

Linux

Q1: Ten basic Linux commands

A: Common commands are `ls`, `cd`, `pwd`, `mkdir`, `rm`, `cp`, `mv`, `touch`, `cat`, and `chmod`. Each performs basic file or directory operations. They're essential for system navigation and management.

Q2: What is shell?

A: A shell is a command interpreter that takes user input and passes it to the kernel for execution. Examples include Bash, Zsh, and Ksh. It acts as an interface between user and OS.

Q3: Architecture of Linux kernel

A: The Linux architecture includes User Space, Kernel Space, and Hardware. Kernel space handles process management, memory, device drivers, and system calls.

Q4: Use of touch command

A: The `touch` command creates an empty file or updates a file's access and modification timestamps.

Q5: What is timestamp?

A: A timestamp records the date and time when a file was created, modified, or accessed.

Q6: 19 directory

A: Possibly refers to `/tmp` or `/proc` (if typo). Clarify if you mean a specific directory — there's no "19 directory" in Linux by default.

Q7: Editors in Linux system

A: Common editors are `vi/vim`, `nano`, and `gedit`. They allow text editing directly in the terminal or GUI.

Q8: Vi/Vim editor modes

A: Vim has three main modes — Command mode (for operations), Insert mode (for typing text), and Visual mode (for selection).

Q9: How to create parent directory in Linux system

A: Use `mkdir -p /path/to/parent/child`. The `-p` flag creates missing parent directories automatically.

Q10: How to copy file in Linux system

A: Use `cp source destination`. Example: `cp file1.txt /home/user/`.

Q11: How to move file in Linux system

A: Use `mv source destination`. It can also rename files or move them to another directory.

Q12: How to rename files in Linux

A: The `mv` command is used to rename files. Example: `mv old.txt new.txt`.

Q13: Explain sudo user

A: A sudo user can run commands with root privileges temporarily using `sudo`. It's safer than logging in as root.

Q14: Explain root user

A: The root user is the superuser with unrestricted system access. It can modify any file or configuration.

Q15: Difference between root and sudo user

A: Root has permanent administrative rights; a sudo user temporarily executes commands as root using `sudo`.

Q16: Explain command mode in Vim editor

A: In command mode, you execute commands like `:w`, `:q`, or `dd`. It's used for saving, quitting, or editing text efficiently.

Q17: What are the different types of users in Linux system

A: Types are Root, System users, and Normal (local) users. Each has different permissions and roles.

Q18: What is local user

A: A local user is a standard user created for regular tasks. It's defined in `/etc/passwd`.

Q19: What is system user

A: System users are created by the OS or services like `mysql`, `nginx` for running background processes.

Q20: How to create a local user in Linux

A: Use `useradd username` or `adduser username`, then set a password with `passwd username`.

Q21: What are the different files affected when new user is added

A: Files like `/etc/passwd`, `/etc/shadow`, `/etc/group`, and `/etc/login.defs` are updated.

Q22: Can we create new user without using useradd or adduser command?

A: Yes, by manually editing `/etc/passwd`, `/etc/shadow`, and `/etc/group`, but it's not recommended.

Q23: How to assign password to user in Linux system

A: Use `passwd username`. It updates the encrypted password in `/etc/shadow`.

Q24: Explain /etc/shadow file

A: It stores encrypted user passwords and password aging details. Only root can read it.

Q25: Explain /etc/passwd file

A: It contains user account details like username, UID, GID, home directory, and shell.

Q26: Password aging policy

A: Password aging enforces rules like maximum/minimum password age and warning days. Managed via `chage` command.

Q27: What is umask

A: Umask defines default permission bits removed when a new file or directory is created.

Q28: Default umask of root and local user

A: Root default is 0022 (files 755), normal user default is 0002 (files 775).

Q29: How to change temporary and permanent umask of any user

A: Temporary: use `umask value` in terminal. Permanent: add it in `~/.bashrc` or `/etc/profile`.

Q30: What is link count

A: It's the number of directory entries (hard links) pointing to the same inode.

Q31: Difference between hardlink and softlink

A: Hard links point to inode; deleting original doesn't affect it. Soft links (symbolic) point to file path and break if original is deleted.

Q32: Explain archiving

A: Archiving combines multiple files into one archive (e.g., `.tar`) without compression.

Q33: Explain compression

A: Compression reduces file size using algorithms like `gzip` or `bzip2`.

Q34: Difference between gzip and tar file

A: `tar` archives multiple files; `gzip` compresses a single file. Together (`.tar.gz`) they create a compressed archive.

Q35: Fields of crontab

A: Crontab has 6 fields: Minute, Hour, Day of Month, Month, Day of Week, and Command.

Q36: Set a crontab for every Friday at 10 AM

A: `0 10 * * 5 command`.

Q37: Cat command uses

A: Displays, creates, or concatenates file contents. Example: `cat file.txt`.

Q38: What are redirectors in Linux

A: Redirectors (`>`, `>>`, `<`, `2>`) are used to redirect input/output streams.

Q39: Explain top command

A: Shows real-time system processes, CPU, and memory usage. Useful for monitoring performance.

Q40: Difference between jobs and ps command

A: `jobs` shows background processes of the current shell; `ps` lists all running processes in the system.

Q41: How to terminate jobs in Linux

A: Use `kill %jobnumber` or `kill PID`. `kill -9 PID` forces termination.

Q42: Difference between free memory and available memory

A: Free is unused memory; available includes free + cache + buffer that can be reused.

Q43: What is ISP

A: ISP (Internet Service Provider) provides internet connectivity services.

Q44: What is difference between public and private IP

A: Public IP is accessible over the internet; private IP is used within local networks.

Q45: What is the use of static IP

A: A static IP remains fixed; ideal for servers, hosting, or consistent remote access.

Q46: Explain OSI model

A: OSI has 7 layers — Physical, Data Link, Network, Transport, Session, Presentation, and Application — for structured network communication.

Q47: What is load average in top command and what it indicates

A: Load average shows system workload over 1, 5, and 15 minutes. Values above CPU count indicate high load.

Q48: What is kernel

A: The kernel is the core of the OS managing CPU, memory, and device interactions.

Q49: What is low level language

A: Low-level languages like Assembly or Machine Code are hardware-oriented and require deep system knowledge.

Q50: How to see Linux system configuration

A: Use commands like `uname -a`, `lscpu`, `lsblk`, `free -h`, or `df -h` for system details.

AWS

Q1: Difference between virtualization and AWS cloud

A: Virtualization creates multiple virtual machines on a single physical host using a hypervisor. AWS cloud provides on-demand virtualized resources plus managed services (compute, storage, networking, managed databases), billing, autoscaling, and global infrastructure.

Q2: Difference between region and availability zone

A: A region is a geographically distinct area (e.g., ap-south-1). An Availability Zone (AZ) is an isolated data center within a region; AZs provide redundancy and low-latency links inside a region.

Q3: Explain security group

A: A security group is an instance-level virtual firewall for EC2 controlling inbound/outbound traffic by rules. It is stateful — return traffic is automatically allowed.

Q4: Difference between NACL and security group

A: NACLs are subnet-level stateless filters with allow/deny rules applied in order; security groups are stateful instance-level firewalls. Security groups are preferred for per-instance rules; NACLs for coarse subnet control.

Q5: Scenario: lost your SSH key pair — how to get it back?

A: You cannot recover a lost private key from AWS; restore access by replacing the key: use EC2 Instance Connect, Systems Manager Session Manager, or detach the root volume, mount it on another instance to add a new public key to `~/.ssh/authorized_keys`, then reattach.

Q6: What is instance and different types of instance type

A: An EC2 instance is a virtual server in AWS. Types include General purpose (t3/t4), Compute optimized (c5), Memory optimized (r5), Storage optimized (i3), and GPU instances (g4), each tuned for workload profiles.

Q7: What is status check — explain in short?

A: Status checks are EC2 health checks: System status (hypervisor/network) and Instance status (OS-level). Failures indicate underlying hardware or guest OS issues and guide remediation.

Q8: What is AMI

A: AMI (Amazon Machine Image) is a template containing an OS, application server, and launch permissions used to create EC2 instances.

Q9: What is custom AMI

A: A custom AMI is an AMI you create from a configured instance (installed packages, configs) to launch identical instances later.

Q10: Difference between AMI and launch template

A: AMI is the machine image (OS + software); a Launch Template holds launch configuration (AMI ID, instance type, key pair, security groups, user data). Launch Templates reference AMIs and other options for launches.

Q11: What are different purchasing options

A: Options: On-Demand (pay-per-hour/second), Reserved Instances / Savings Plans (commit for discount), Spot Instances (unused capacity at discount), Dedicated Hosts/Instances for isolation.

Q12: Explain EBS volume and its types

A: EBS is block storage for EC2 instances; types include gp3/gp2 (general purpose SSD), io2/io1 (provisioned IOPS SSD), st1 (throughput HDD), sc1 (cold HDD) and standard (magnetic, legacy).

Q13: How to mount EBS volume in EC2 instance

A: Attach the EBS to the instance, then on the instance create a filesystem (`mkfs`), create mount point (`mkdir /data`), and `mount /dev/xvdf /data`; add `/etc/fstab` for persistence.

Q14: What is snapshot

A: An EBS snapshot is a point-in-time, incremental backup of a volume stored in S3; snapshots can create new volumes or AMIs.

Q15: Create a schedule to take backup every 31st of month

A: Use EventBridge cron: `cron(0 0 31 * ? *)` to run at 00:00 UTC on the 31st; attach this rule to a Lambda that creates snapshots. (Note: months without 31 days will skip.)

Q16: Lambda to take backup of volume every 31st of month (Python script)

A: Explanation: EventBridge cron triggers Lambda which calls EC2 `create_snapshot` for specified EBS volume IDs or volumes attached to specified instances; add tags and retention policy.

Code (Lambda, python3.9):

```
```bash
import os
import boto3
import datetime

ec2 = boto3.client('ec2')

Expected env var: VOLUME_IDS (comma-separated) OR
INSTANCE_IDS (comma-separated)
VOLUME_IDS = os.environ.get('VOLUME_IDS') # e.g.
"vol-0123,vol-0456"
INSTANCE_IDS = os.environ.get('INSTANCE_IDS') # optional
DESCRIPTION_PREFIX = os.environ.get('DESC_PREFIX',
'LambdaBackup')

def get_volumes_from_instances(instance_ids):
 vols = []
 if not instance_ids:
 return vols
 resp = ec2.describe_instances(InstanceIds=instance_ids)
 for r in resp['Reservations']:
 for i in r['Instances']:
 for dev in i.get('BlockDeviceMappings', []):
 if 'Ebs' in dev and 'VolumeId' in dev['Ebs']:
 vols.append(dev['Ebs']['VolumeId'])
 return vols
```

```

def lambda_handler(event, context):
 volumes = []
 if VOLUME_IDS:
 volumes = [v.strip() for v in VOLUME_IDS.split(',')]
 if v.strip():
 if INSTANCE_IDS:
 volumes += get_volumes_from_instances([x.strip() for
x in INSTANCE_IDS.split(',') if x.strip()])
 volumes = list(set(volumes))
 if not volumes:
 return {'status': 'no-volumes-found'}
 created = []
 for vol in volumes:
 desc = f"{DESCRIPTION_PREFIX} {vol}
{datetime.datetime.utcnow().isoformat()}"
 resp = ec2.create_snapshot(VolumeId=vol,
Description=desc)
 # Optionally tag snapshot
 ec2.create_tags(Resources=[resp['SnapshotId']],
Tags=[{'Key': 'CreatedBy', 'Value': 'LambdaBackup'}])
 created.append(resp['SnapshotId'])
 return {'created_snapshots': created}
 ...

```

**IAM:** Lambda needs `ec2:CreateSnapshot`, `ec2:DescribeInstances`, `ec2:CreateTags`, and `ec2:DescribeVolumes` permissions.

#### **Q17: What is NFS**

**A:** NFS (Network File System) is a distributed filesystem protocol allowing clients to mount and access remote directories over the network as local mounts.

#### **Q18: How to create and mount EFS in two different EC2 machine**

**A:** Create EFS filesystem, create mount targets in the VPC subnets, allow NFS (TCP 2049) in security groups, then on each EC2 install `amazon-efs-utils` and run `sudo mount -t efs fs-xxxx: /mnt/efs` or use the EFS mount helper.

#### **Q19: What is IP explain different types and its use**

**A:** IP (Internet Protocol) is the addressing scheme for networked hosts. Types: IPv4 (32-bit), IPv6 (128-bit), public/private addresses; used to route packets between hosts.

#### **Q20: Different classes of IP address**

**A:** IPv4 classes historically: A (1.0.0.0/8), B (128.0.0.0/16), C (192.0.0.0/24); today CIDR is used instead of strict classes. Private ranges: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16.

#### **Q21: What is Internet Gateway**

**A:** An Internet Gateway (IGW) enables communication between instances in a VPC and the Internet, providing NAT for IPv4 and a target for default route.

**Q22: What is NAT Gateway**

**A:** A NAT Gateway allows instances in private subnets to initiate outbound internet connections while preventing inbound connections from the internet.

**Q23: Difference between NAT and IGW**

**A:** IGW provides two-way internet access to public subnet instances; NAT Gateway offers outbound-only internet access for private subnet instances and preserves private IPs.

**Q24: What is Elastic IP**

**A:** An Elastic IP is a static public IPv4 address you can associate with an EC2 instance or network interface to maintain a persistent public IP.

**Q25: What are placement groups**

**A:** Placement groups determine EC2 placement strategy: cluster (low-latency, same AZ), spread (isolation across hardware), and partition (HDFS-style partitioning across racks).

**Q26: What is Load Balancer**

**A:** An AWS Load Balancer distributes incoming traffic across targets (EC2, IPs, Lambdas) to improve availability and scalability.

**Q27: What is OSI model — Explain layers**

**A:** OSI is 7-layer model: Physical (bits), Data Link (MAC), Network (IP/routing), Transport (TCP/UDP), Session (connections), Presentation (format/encoding), Application (HTTP, SMTP). It's a conceptual stack for network communication.

**Q28: Difference between ALB and NLB**

**A:** ALB (Application Load Balancer) operates at Layer 7 and supports path/host-based routing and HTTP features. NLB (Network Load Balancer) operates at Layer 4, handles millions of connections with ultra-low latency and preserves source IP.

**Q29: What is Auto Scaling Group**

**A:** An ASG manages a collection of instances and automatically scales capacity based on policies, schedules, or health checks to meet demand.

**Q30: What are target groups**

**A:** Target groups define the set of targets (instances, IPs, Lambdas) for a load balancer and include health check configuration for routing decisions.

**Q31: Scenario: servers are getting more requests than they regularly handle — how will it handle increased number of requests?**

**A:** Use Auto Scaling to add instances, use Load Balancer to distribute traffic, enable caching (CloudFront / ElastiCache), and scale database read replicas to reduce load.

**Q32: How servers can handle request in DDoS attack**

**A:** Use AWS Shield/Shield Advanced, AWS WAF, rate limiting, autoscaling carefully, CloudFront caching, and network-level protections to absorb/mitigate attack traffic and block malicious patterns.

**Q33: What is VPC peering (*interpreting "VPC wearing" as VPC peering*)**

**A:** VPC Peering connects two VPCs for private routing using private IPs without a gateway; it's non-transitive and good for simple VPC-to-VPC connectivity.



**Q34: Difference between VPC peering and Transit Gateway**

**A:** VPC Peering is point-to-point and non-transitive; Transit Gateway provides hub-and-spoke scalable routing across many VPCs and on-prem connections with centralized control.

**Q35: What is IAM service in AWS**

**A:** IAM (Identity and Access Management) controls authentication and authorization for AWS resources using users, groups, roles, and policies.

**Q36: What are roles**

**A:** IAM roles are identities with permissions assumed by trusted entities (EC2, Lambda, users) to grant temporary credentials without long-term keys.

**Q37: What are policies**

**A:** Policies are JSON documents that define allowed or denied actions on AWS resources; attachable to users, groups, or roles.

**Q38: Difference between IAM roles and IAM policy**

**A:** Roles are identities you assume to get permissions; policies define the permissions (what's allowed/denied) attached to identities.

**Q39: Scenario: User1 has only policy attached to access EC2, and for security User1 cannot give S3 access — how can User1 get access to S3 bucket to work with bucket data?**

**A:** Use an IAM role with S3 access and let User1 assume that role (via STS) or create a resource-based S3 bucket policy granting access to a role/user; or use a role-based EC2 instance profile to access S3 from EC2 without giving User1 direct S3 permissions.

**Q40: S3 Bucket storage classes**

**A:** S3 classes: Standard, Intelligent-Tiering, Standard-IA, One Zone-IA, Glacier Instant/Flexible/Deep Archive — offering tradeoffs of cost, access time, and redundancy.

**Q41: What is replication rule**

**A:** A replication rule automatically copies objects from one S3 bucket to another (same or different region) based on prefix/tag filters and can change storage class on destination.

**Q42: Create Lambda to replicate data from S3 bucket1 to bucket2 in different storage class**

**A:** Explanation: Trigger on S3 PUT, Lambda copies the object to destination bucket with `StorageClass` set (e.g., `STANDARD_IA`), and can add metadata/tags. Ensure proper IAM and event notifications.

**Code (Lambda — python3.9):**

```
```bash
import boto3
import urllib.parse
import os

s3 = boto3.client('s3')
DEST_BUCKET = os.environ.get('DEST_BUCKET') # destination
bucket name
DEST_STORAGE_CLASS = os.environ.get('DEST_STORAGE_CLASS',
'STANDARD_IA') # e.g., STANDARD_IA
```

```

def lambda_handler(event, context):
    for rec in event.get('Records', []):
        src_bucket = rec['s3']['bucket']['name']
        key = urllib.parse.unquote_plus(rec['s3']['object']
['key'])
        copy_source = {'Bucket': src_bucket, 'Key': key}
        dest_key = key # preserve key, or modify if needed
        # Copy object to destination with desired storage
class
        s3.copy_object(
            Bucket=DEST_BUCKET,
            Key=dest_key,
            CopySource=copy_source,
            StorageClass=DEST_STORAGE_CLASS,
            MetadataDirective='COPY'
        )
    return {'status': 'ok'}

```

IAM: Lambda needs `s3:GetObject` on source and `s3:PutObject` on destination, and `s3:GetBucketNotification` if required.

Q43: What is CloudWatch

A: CloudWatch collects and visualizes metrics, logs, and events from AWS resources and can trigger alarms and EventBridge rules for automation.

Q44: What is SNS

A: SNS (Simple Notification Service) is a pub/sub messaging service for sending notifications to subscribers via email, SMS, HTTP, SQS, or Lambda.

Q45: What is RDS

A: RDS (Relational Database Service) is a managed DB service for MySQL, PostgreSQL, Oracle, SQL Server, and Aurora that handles backups, patching, and failover.

Q46: What is schema in Database

A: A schema is the structure/blueprint of a database — tables, fields, relationships, types, constraints — defining how data is organized.

Q47: Difference between Relational and non-relational database

A: Relational DBs use structured schemas and SQL with ACID guarantees; non-relational (NoSQL) DBs are schema-flexible, scale horizontally, and prioritize performance over strict joins/transactions.

Q48: What is DNS service — explain different records

A: DNS maps names to IPs; records include **A** (IPv4), **AAAA** (IPv6), **CNAME** (alias), **MX** (mail exchangers), **TXT** (text), **SRV**(service), and **NS** (name servers).

Q49: What is CloudFront

A: CloudFront is AWS CDN that caches and delivers content globally from edge locations to reduce latency and offload origin servers.

Q50: Create a Lambda function to stop running instance in region ap-south-1

A: Explanation: EventBridge cron triggers Lambda (region ap-south-1) that finds running instances (optionally by tag) and calls `stop_instances`. Ensure IAM includes `ec2:StopInstances` and `ec2:DescribeInstances`.

Code (Lambda — python3.9):

```
```bash
import boto3
import os

REGION = os.environ.get('REGION', 'ap-south-1')
Optionally set TAG_KEY and TAG_VALUE to filter instances;
if not set, will stop all RUNNING instances.
TAG_KEY = os.environ.get('TAG_KEY') # e.g., "AutoStop"
TAG_VALUE = os.environ.get('TAG_VALUE') # e.g., "true"

ec2 = boto3.client('ec2', region_name=REGION)

def lambda_handler(event, context):
 filters = [{'Name': 'instance-state-name', 'Values':
['running']}]}
 if TAG_KEY and TAG_VALUE:
 filters.append({'Name': f'tag:{TAG_KEY}', 'Values':
[TAG_VALUE]})
 resp = ec2.describe_instances(Filters=filters)
 instance_ids = []
 for r in resp['Reservations']:
 for i in r['Instances']:
 instance_ids.append(i['InstanceId'])
 if not instance_ids:
 return {'status': 'no-running-instances'}
 ec2.stop_instances(InstanceIds=instance_ids)
 return {'stopping': instance_ids}
```
```

DevOps

1) Difference between Agile metrology and Waterfall methodology

Waterfall is a linear, sequential model where each phase must be completed before moving to the next. Agile is iterative and incremental, delivering working features in short sprints with continuous feedback. Agile adapts to change faster, while Waterfall expects requirements to be fixed.

2) What is GIT

Git is a distributed version control system that tracks code changes, enables collaboration, and provides features like branching, merging, and rollback. It allows developers to work in parallel and maintain code history efficiently.

3) Lifecycle of GIT

Typical Git workflow cycle includes: modify files in Working Directory → stage them to Index using `git add` → commit changes to Local Repo using `git commit` → push changes to Remote Repo → pull/merge others' changes → deploy or release.

4) Difference between Git and Github

Git is a command-line tool to manage version control locally. GitHub is a cloud platform that hosts Git repositories and adds collaboration features like PRs, issues, webhooks, and CI integrations.

5) Github commands basic

Common GitHub-related Git commands include `git clone` to copy remote repo, `git add` to stage, `git commit` to save changes, `git push` to upload, `git pull` to sync, `git branch` for branches, and `git merge` to combine branches.

6) What are the different metrology of authentication

Authentication methods include password-based, SSH key-based, token-based (API tokens), OAuth-based SSO, and directory-based authentication via LDAP/AD. Each method secures identity verification before granting resource access.

7) What key we use while establishing ssh with EC2

SSH connections to EC2 use asymmetric key pairs consisting of a public key stored on the EC2 instance and a private `.pem` key stored locally. The private key is used during SSH to authenticate securely without password.

8) Explain different branching strategy in git

Strategies include Git Flow (develop + release branches), Feature Branching (each feature isolated), Trunk-based Development (short-lived branches merged into main frequently), and Release Branching for controlled production releases.

9) Difference between monolithic and micro service based application

Monolithic applications bundle all business logic into a single tightly-coupled deployable unit.

Microservices break functionality into independent services communicating via APIs, allowing independent deployment, scaling, and fault isolation.

10) What is containerization

Containerization packages applications with dependencies into isolated, lightweight runtime environments called containers. This ensures consistent behavior across development, testing, and production environments.

11) Docker architecture

Docker uses a client-server model where the Docker Client communicates with the Docker Daemon. The daemon manages Images, Containers, Networks, and Volumes, and pulls/pushes images from/to a Registry such as Docker Hub.

12) Explain dockerfile

A Dockerfile is a declarative script containing instructions (FROM, RUN, COPY, EXPOSE, CMD etc.) to build a Docker image. It defines environment, dependencies, files, and commands required at container runtime.

13) Difference between dockerfile and docker image

Dockerfile is the source blueprint containing image build instructions. Docker Image is the output artifact built from that Dockerfile, used to start containers on any host.

14) Docker networking explain different network used in Docker

Docker networks include Bridge (default isolated network), Host (shares host network), None (no network), Overlay (multi-host networking for Swarm), and Macvlan (assigns MAC addresses to containers for LAN-level access).

15) What is docker hub

Docker Hub is a public cloud registry to store, share, and distribute container images. It supports versioning, automated builds, and access control for open-source or private repositories.

16) What is ECR (elastic container registry)

Amazon ECR is AWS's fully managed private container image registry integrated with IAM, ECS, and EKS. It provides secure, scalable storage and native lifecycle and vulnerability scanning capabilities.

17) What is HPA (Horizontal pod autoscaling)

HPA automatically adjusts the number of running pod replicas in a Deployment or ReplicaSet based on resource metrics like CPU, memory, or custom metrics from metrics-server.

18) Difference between HPA and VPA

HPA scales pods horizontally by increasing/decreasing pod count. VPA scales vertically by adjusting CPU/Memory requests/limits of existing pods for capacity optimization.

19) Explain kubernetes architecture

Kubernetes control plane includes API Server, Scheduler, Controller Manager, and etcd for cluster state. Worker nodes run Kubelet, Kube-proxy, and Pods to host application workloads and networking.

20) What are pods

Pods are the smallest deployable unit in Kubernetes that encapsulate one or more containers sharing the same network namespace and optional shared storage volumes.

21) Best practices while dealing with Pods

Set resource limits and requests, define probes (liveness/readiness), use sidecar containers when needed, avoid running many containers per pod, and use labels/affinity for scheduling control.

22) How to expose pods to external traffic

Pods can be exposed using a Service of type NodePort or LoadBalancer for external access, or using an Ingress resource with an Ingress Controller for HTTP/HTTPS routing.

23) What does a Kubernetes YAML manifest file define?

It defines the desired state of a Kubernetes object including apiVersion, kind, metadata, and spec fields, enabling declarative infrastructure and application management.

24) What is the purpose of a Kubernetes Namespace?

Namespaces logically isolate Kubernetes resources for multi-tenancy, environment grouping, or organizational segregation, enabling access control and quota enforcement at segment level.

25) What is replica set in k8s

ReplicaSet ensures a specified number of identical pod replicas are running at all times. If a pod crashes or is deleted, ReplicaSet automatically replaces it.

26) What are docker volumes

Docker volumes provide persistent storage independent of a container's lifecycle. They are managed by Docker and support data sharing across containers and hosts.

27) How to list all the docker volumes

Use `docker volume ls` to display all volumes defined on the host, and `docker volume inspect <name>` to view volume metadata and mount details.

28) How to build lightweight docker images

Use minimal base images like `alpine`, perform multi-stage builds, remove build-time dependencies, and avoid copying unnecessary files to reduce image size.

29) What is a YAML manifest file in Kubernetes?

It is a declarative configuration file describing Kubernetes resources and their desired state. Kubernetes reconciles actual state to match the declared state.

30) Difference between deployment and statefulset

Deployment manages stateless applications with rolling updates and identical replicas. StatefulSet manages stateful workloads with stable identities, ordered deployment, and persistent storage association.

31) Types of autoscaling in kubernetes

Horizontal Pod Autoscaling (pods count), Vertical Pod Autoscaling (resource tuning), and Cluster Autoscaler (adds/removes worker nodes based on pod scheduling needs).

32) What is an Ingress Controller?

An Ingress Controller is a Kubernetes component that processes Ingress rules to route external HTTP/HTTPS traffic to internal services. It provides SSL termination and host/path routing.

33) What is Infrastructure as Code (IaC) and why do we need it in cloud environments?

IaC is the practice of defining and provisioning infrastructure using code files instead of manual steps. It enables version control, repeatability, consistency, automation, and reduces human errors in cloud provisioning.

34) What is tfstate file in terraform

The `terraform.tfstate` file stores the current state of managed infrastructure and maps resources in configuration to real-world cloud resources. Terraform uses it to plan and apply incremental changes accurately.

35) Explain terraform life cycle

Terraform follows `init` to configure backend/plugins, `plan` to preview changes, `apply` to execute changes, and `destroy` to remove managed infrastructure. This ensures controlled and predictable provisioning.

36) Difference between terraform and cloud-formation

Terraform is a multi-cloud IaC tool using HCL language and works across AWS, Azure, and GCP. CloudFormation is AWS-native, tightly integrated with AWS services, and uses JSON/YAML for templates.

37) How to add state locking feature

Use a remote backend like AWS S3 for state storage combined with DynamoDB table for state locking. This prevents concurrent changes to the same state in collaborative environments.

38) What is blue green deployment

Blue-Green deployment maintains two identical environments — Blue (current) and Green (new). Traffic is switched to Green after validation, ensuring zero downtime and easy rollback.

39) What is canary deployment

Canary deployment releases a new version to a small subset of users first. Once verified, traffic is gradually shifted to the new version to reduce risk in case of failure.

40) What is maven

Maven is a build automation and dependency management tool for Java projects. It uses `pom.xml` to define dependencies, plugins, packaging, and lifecycle phases like compile, test, and deploy.

41) How to check code vulnerabilities

Use security scanners like Snyk, Trivy, SonarQube, OWASP Dependency Check or GitHub Security Scans to detect vulnerable libraries, insecure code patterns, or policy violations.

42) What are webhooks

Webhooks are HTTP callbacks triggered on specific events (e.g., git push) that notify or trigger external workflows like CI/CD pipelines, notifications, or automation jobs.

43) What is Jenkins?

Jenkins is an open-source automation server used for implementing CI/CD pipelines. It integrates with version control, build tools, containers, and deployment platforms to automate software delivery.

44) Explain stages of Jenkins pipeline

A pipeline typically includes stages like Checkout (source code), Build (compile/package), Test (unit/integration), Scan (quality/security), Deploy (to environments), and Post actions (cleanup/notifications).

45) What are Jenkins plugins, and how are they used?

Jenkins plugins extend functionality to integrate with tools like GitHub, Maven, Docker, Kubernetes, Slack, and Terraform. They enable new pipeline steps, UI features, and third-party integrations without custom code.

46) What's the difference between continuous integration, continuous delivery, and continuous deployment?

CI focuses on frequent code integration with automated testing. Continuous Delivery ensures code is always in a releasable state but deploys manually. Continuous Deployment automatically pushes every validated change to production.

47) Explain different stages in CI-CD setup?

Typical CI/CD stages include source checkout, build, test, artifact packaging, security scanning, environment deployment, post-deployment verification, and monitoring/rollback handling.

48) What is different between Scripted and Declarative Pipelines?

Scripted pipelines use full Groovy scripting and provide complete flexibility and complexity control. Declarative pipelines use a structured syntax with opinionated blocks, making them easier, clearer, and less error-prone.

49) What is Observability

Observability is the ability to understand internal system state using external telemetry — logs, metrics, and traces. It helps in debugging, performance tuning, and proactive incident response.

50) What is Datadog agent

Datadog Agent is a lightweight software installed on hosts or containers that collects metrics, logs, traces, and sends them to the Datadog platform for monitoring, alerting, and analytics.

Submitted by : Mukund Deotale (CDEC trainer)