INTERVIEW QUESTIONS& ANSWERS

**1. What is DevOps?** DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the development life cycle, improve collaboration, and deliver high-quality software continuously. It emphasizes automation, continuous integration, continuous delivery, and monitoring.

**2. How many types of OS in Linux?** There are multiple Linux distributions (distros), such as Ubuntu, CentOS, Red Hat Enterprise Linux (RHEL), Fedora, Debian, Arch Linux, and SUSE. These distros vary based on package management, use cases, and supported environments.

**3. Add secondary group (Linux):** Use the command:

usermod -aG <groupname> <username>

**4. What is a build tool?** A build tool automates the process of compiling source code, packaging applications, and managing dependencies. Examples: Maven (Java), Gradle, Make.

**5. Maven Lifecycle:**

* **validate**: check project structure
* **compile**: compile source code
* **test**: run unit tests
* **package**: package into JAR/WAR
* **verify**: verify integration tests
* **install**: install to local repo
* **deploy**: deploy to remote repo

**6. What is CI/CD?**

* **CI (Continuous Integration)**: regularly integrating code into a shared repo and testing it automatically
* **CD (Continuous Delivery/Deployment)**: automatically preparing/deploying code to staging or production

**7. Check running/stopped containers:**

docker ps # running

docker ps -a # all

Check Docker system info:

docker system info

**8. What is orchestration?** Orchestration automates container deployment, scaling, networking, and management across clusters. Kubernetes and Docker Swarm are orchestration tools.

**9. What is pull, fetch, and merge? (Git)**

* **pull**: fetch + merge
* **fetch**: download new commits without applying
* **merge**: integrate fetched changes into the current branch

**10. Kubernetes services:**

* ClusterIP (default)
* NodePort
* LoadBalancer
* ExternalName

**11. How to build a Docker image:**

docker build -t myapp:latest .

**12. What is Jenkins?** Jenkins is an open-source CI/CD tool that automates the building, testing, and deployment of applications.

**13. Master and Slave in Jenkins:** The master handles scheduling and monitoring, while slaves (agents) execute jobs to distribute the load.

**14. Parameterized Build in Jenkins:** Allows passing custom values (parameters) like branch name or version during build execution.

**15. Containerization vs Virtualization:**

* **Virtualization**: runs full OS per VM; uses hypervisors
* **Containerization**: shares host OS; runs isolated apps in containers (Docker)

**16. Default Jenkins plugins:**

* Git plugin
* Pipeline
* SSH
* Credentials plugin

**17. Build Triggers in Jenkins:** Conditions to automatically start builds:

* Poll SCM
* GitHub hook
* Upstream project

**18. Cron syntax in Jenkins:**

MIN HOUR DOM MON DOW

Example: H 2 \* \* \* (every day at 2 AM)

**19. Node Affinity in Kubernetes:** Schedule pods on specific nodes using labels and rules (preferred or required).

**20. What is system prune (Docker)?** Cleans unused containers, networks, images:

docker system prune

**21. How scheduler works (Kubernetes)?** The scheduler assigns pods to suitable nodes based on resource requirements, constraints, and affinities.

**22. Types of Kubernetes controllers:**

* Deployment
* ReplicaSet
* StatefulSet
* DaemonSet
* Job / CronJob

**23. Jenkins Pipeline:** Scripted (Groovy) or Declarative (structured YAML-like) way to define build steps:

declarative pipeline {

agent any

stages {

stage('Build') { steps { ... } }

stage('Test') { steps { ... } }

stage('Deploy'){ steps { ... } }

}

}

**24. Check if container is running on host:**

docker ps

Use:

hostname

inside the container, or inspect the container.

**25. Role and RoleBinding (Kubernetes RBAC):**

* **Role**: defines permissions within a namespace
* **RoleBinding**: assigns a Role to a user/service account

**26. Docker stages:**

* Pull base image
* Install dependencies
* Copy source code
* Build app
* Set environment
* Run command (ENTRYPOINT/CMD)

**27. Deploy pod to specific node:** Use nodeSelector or affinity rules to deploy to node3

spec:

nodeSelector:

kubernetes.io/hostname: node3

**28. Taints and Tolerations:**

* **Taint**: restricts pods from running on a node
* **Toleration**: allows pods to run on tainted nodes

**29. Run container in background:**

docker run -d myapp

**30. Send 5GB file to remote server:**

* Use scp:

scp largefile.zip user@remote:/path

**31. Connect to a server (SSH):**

ssh user@hostname\_or\_ip

**32. Check OS version (Linux):**

cat /etc/os-release

uname -a

**33. StatefulSet vs Deployment:**

* **StatefulSet**: stable identity, ordered deployment, for databases
* **Deployment**: stateless apps, auto scaling, rolling updates

**34. Run one app in 10 pods:** Create a Deployment with replicas: 10

spec:

replicas: 10

**35. Docker Compose:** Tool to run multi-container apps with a YAML file

docker-compose up -d

**36. What is Trivy?** A security scanner for Docker containers to detect vulnerabilities in OS packages and dependencies…