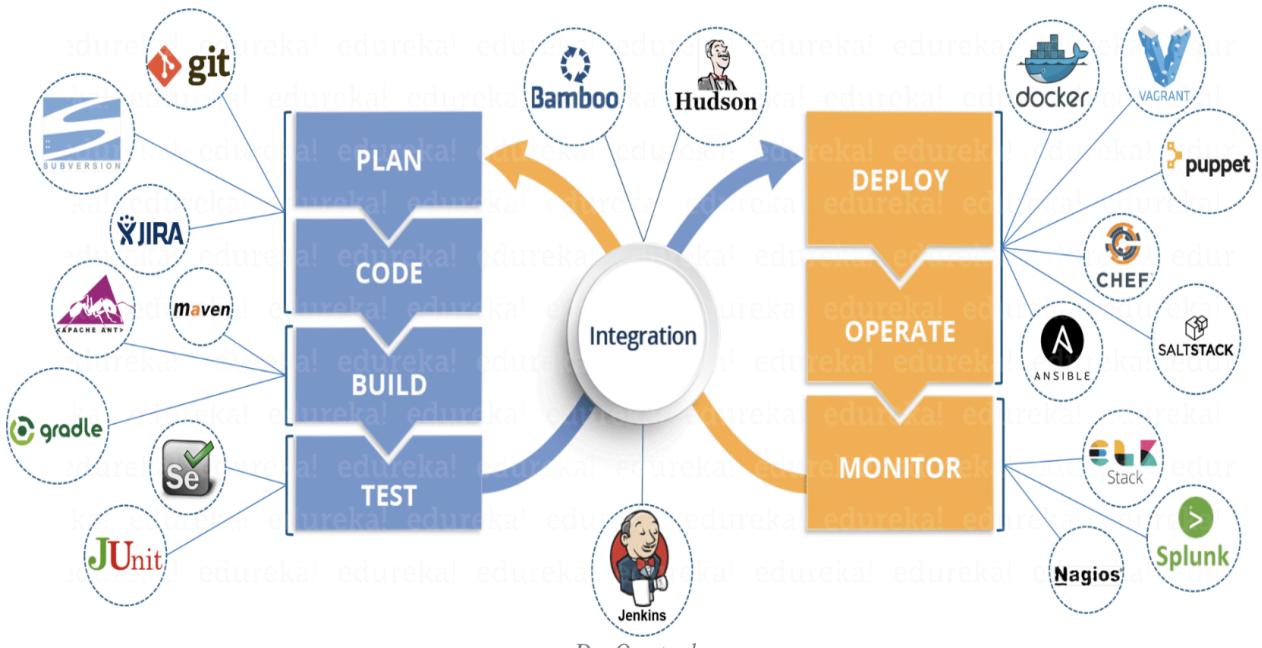
DEVOPS INTERVIEW QUESTIONS

1. What is Devops?
   (Development + Operations)
   Automation in all stages of Software Development life cycle.
   Increase's speed of delivering applications

   Benefits:
   Faster Deployment Time
   Collaborative Work environment
   Defect fixes at an early Stage
   Quick Releases

2. Can you name devops Tools and their uses



DevOps tools

3. What is the role of Configuration Management? Some examples?

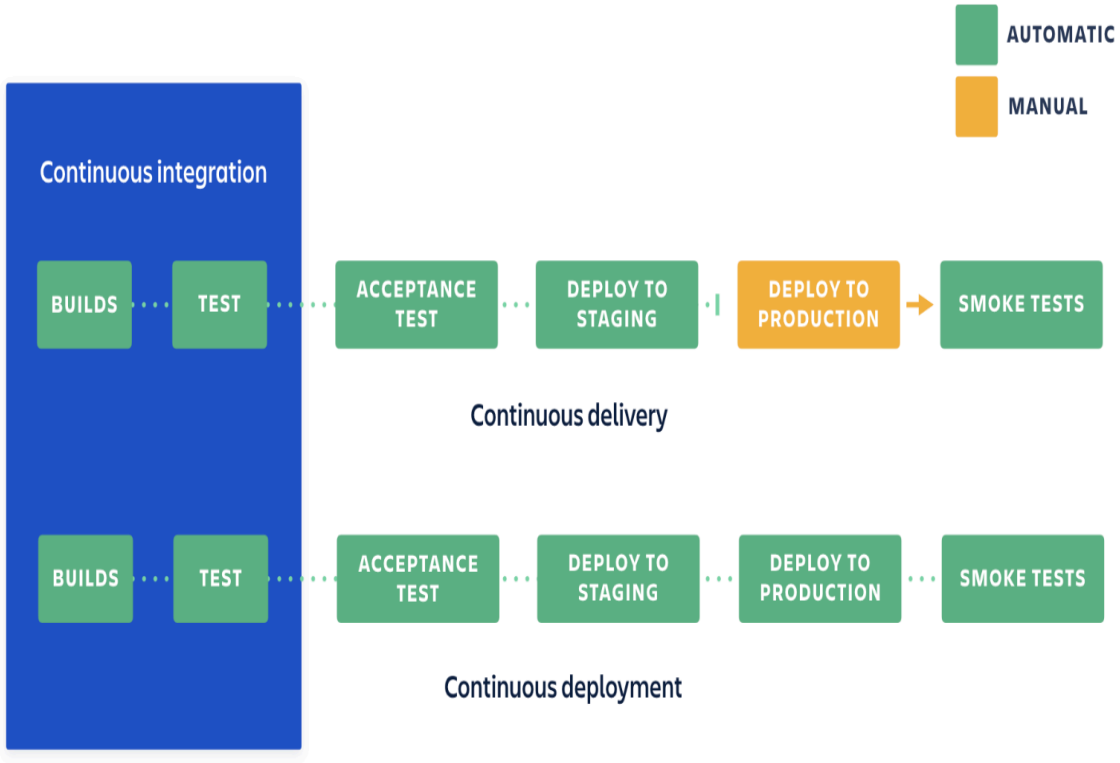
Process in which Servers/Sytems are automated,monitored and managed

Avoids any manual Installation and configuration and accidental changes

Example: Ansible,Terraform,salt

4. Difference between Continuous Integration and Continuous Delivery?

Continuous Integration	***Continuous Delivery & Deployment***
Automatically building and testing Code. Run Tests automatically for every new Code commits pushed	**Continuous deployment:** Automated Deployments till production without any manual Intervention(**One Click deployment)** **Continuous Delivery:** Automated Deployments till production with some manual interventions Depends upon company use cases
Uses: Less Bugs shipped to production	Quick Feature releases which helps companies

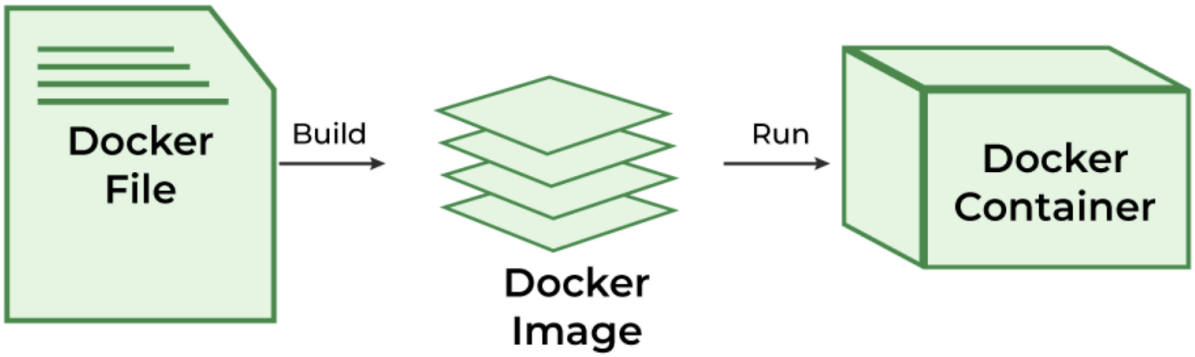


5. Can you name some of CI/CD Tools?
   Jenkins, Gitlab, Azure Devops Pipeline,ArgoCD,Circle ci etc

6. What is Docker?

   Docker Containers are standardized, executable components that combine application source code with the operating system (OS) libraries and dependencies required to run that code in any environment.

   [OS+Libraries] Packaged



7. **What is Docker File?**
   Contains Instruction for creating Docker Image

   Example

   FROM nginx
   COPY index.html /usr/share/nginx/html/index.html

   Cmd : docker build -t test:v1 .

8. What is Docker Image?
   Blue print for libraries and dependencies required inside container for application to run
   Used to create a container
   Can be stored in Repository (Docker hub,gitlab registry)

   # docker push vdr/app:v1

9. What is Docker container?
   When you run a image it becomes container

   # docker run –name devops -d app:v1
   #docker stop devops

10. What is Branching Strategy?

Release branching, which creates a branch for a potential new release

Feature branching, which creates a branch for specific features or tasks

11. What are Configuration Management Tools?

Configuration management tools help system administrators keep track of the current state of applications and services, so that they can more quickly identify systems that require attention,
Ex: Ansible,puppet,chef

Ansible is a popular DevOps automation tool used during the build, configuration, and management phases. By automating processes like testing and deployment, Ansible helps DevOps teams save time, reduce errors, and scale in pace with growing demand.

12. What is Kubernetes?

Kubernetes, also known as K8s, is an open source system for automating deployment, scaling, and management of containerized applications. It is a Container Orchestration tool for containers

13. What are Different Devops Deployment Types?

Blue/Green Deployment	Canary Deployment
Maintains 2 identical Environments. One for Production and one for new	Rolling/Go live for specific subset of users first before releasing them to

version	production. Router/Load Balancer for redirecting some requests to new server
Use Case: Financial systems which should not have downtime when switching on Major Releases	Use Case: 1. Strong focus on UX changes 2. Incremental Deployments
Advantages: No downtime Disadvantage: Cost	Advantages: New software release Less Cost

14. What is IAC(Infrastructure as Code)? Some examples?
    Infrastructure automation to create Environments. Helps to Version the infrastructure
    No manual changes needed. When any manual changes done it will be overwritten in next RUN

    Example: terraform (Cloud agnostic), AWS Cloud Formation, Microsoft azure resource manager,Redhat,Ansible. etc

15. What is State(file) in Terraform?
    It stores Information about your infrastructure in a State file. It is a source of Truth for Infrastructure

16 . What are the differences between Containerization and Virtualization?

Virtualization	Containerization
Developers can run multiple operating systems on the hardware of single physical server	Deploy multiple applications under the same operating system on single physical server

17. What is Regression Testing?

Regression testing assesses how an application behaves after a new change has been implemented.

18. What is Load Balancing?

Load balancing is the process of evenly distributing incoming network traffic across a group of backend servers.

19 What is Monitoring and Observability?

Monitoring	**Observability**
Tracking and Data Collection helps Devops Team better understand the current state of system	Use findings (data)from Monitoring to diagnose problems and actively resolve them through debugging. Proactive and take actions (Artificial Intelligence)
Ex: Prometheus,grafana	Ex: Prometheus,grafana,Newrelic

20) What is Artificial Intelligence?

Build Machines capable of performing tasks that typically require Human Intelligence

Pre trained AI services:

Usecases:

1. Chatbot & Virtual Assistant - customer support
2. Text to Image processing
3. Code Generation
4. Extract details from PDF
5. Self Driving cars
6. Extract data from Scanned documents
7. Image and Video Analysis

21) How do you monitor the performance of Application and Infrastructure?

Application → Prometheus,Grafana/Datadog/Dynatrace

Infrastructure - Prometheus,Grafana

22) How to ensure Security in Devops Process?

Enable SAST scanning - Static application security scanning - Analyse source code for security vulnerabilities

Opensource SAST tools: Sonar,Snyk

Container Scanning: Snyk

Enterprise: Gitlab premium, Fortify Sast

24) What is Helm?

Helps to manage Kubernetes applications. Helm charts help you define, install,upgrade applications

23) How do you rollback deployments ?

In case of application failure in Kubernetes we can use helm rollback command

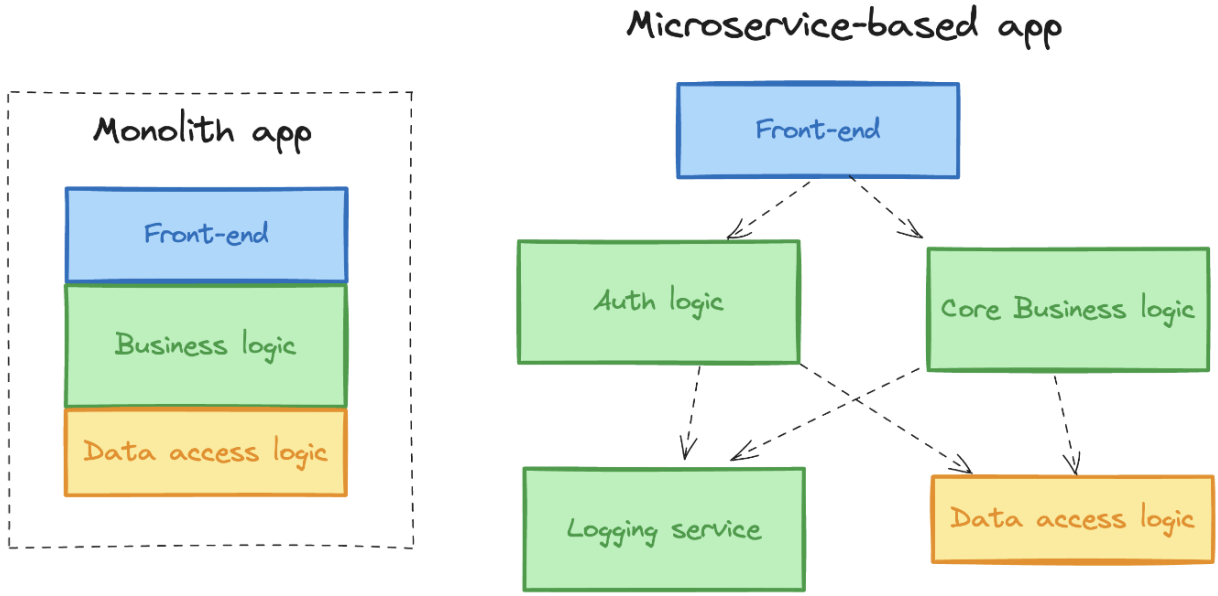
Kubectl: kubectl rollout undo deployment/app --to-revision=2
(OR)
Helm: helm rollback <releasename> <versionnumber>

24) What scripting Tools mainly used for devops?
Python and bash

25) What do you know about Cloud platform Providers?
IAAS - Infrastructure as a service
Cloud providers - AWS,GCP,Azure,Alibaba cloud,salesforce,Digital Ocean,IBM,Dell

26) What is difference between Microservice and Monolithic applications?

Monolithic	Microservice
Single (often massive) unit where all functions and services are interconnected and run as a single process.	A microservice is an architectural style that structures an application as a collection of small, loosely coupled, and independently deployable services



27) Difference between Horizontal scaling Vs Vertical Scaling

Horizontal Scaling	Vertical Scaling
1. Adding more machines or instances to your infrastructure.	Adding more resources (CPU, RAM, storage) to an existing machine.
2. Ex: Distributed systems and cloud environments.	Disadvantage: Limited by the maximum capacity of the hardware.

28) How to Handle secrets in Devops Pipeline?

i) Use secret Management tool HashiCorp Vault, AWS Secrets Manager, or Azure Key Vault to securely store and retrieve secrets.

ii) Storing secrets in environment variables managed by the CI/CD tool.

29) Difference between Gitops and Traditional CICD

GITOPS	CICD Traditional
GitOps is a practice that uses Git as the single source of truth for infrastructure and application management changes are automatically applied by reconciling the desired state in Git with the actual state in the environment GitOps emphasizes a declarative approach where the desired state is defined in Git	Configurations may be scattered across various tools and scripts. manual steps for deployment.

30) What are main components of Kubernetes Cluster?

MASTER NODE COMPONENTS	• **API Server**: The front-end for the Kubernetes control plane, handling all RESTful requests for the cluster. • **etcd**: A distributed key-value store that holds the cluster's configuration and state. • **Controller** Manager: Manages various controllers that regulate the state of the cluster. • **Scheduler**: Assigns workloads to different nodes based on resource availability and other constraints.
WORKER NODE COMPONENTS	• **Kubelet**: This is an agent that runs on each node, and it ensures that each container is running in a Pod. • **Kube-proxy**: A network proxy that maintains network rules and handles routing for services. • **Container Runtime**: This software runs containers, such as Docker, containerd, or CRI-O.

OTHERS	• **Pods**: These are the smallest deployable units in Kubernetes; they consist of one or more containers. • **Services**: Services define a logical set of Pods and a policy for accessing them, they're often used for load balancing. • **ConfigMaps and Secrets**: They manage configuration data and sensitive information, respectively. • **Ingress**: It manages external access to services, typically through HTTP/HTTPS. • **Namespaces**: They provide a mechanism for isolating groups of resources within a single cluster.

31) What is Autoscaling?

It is a cloud computing feature that automatically adjusts the number of resources allocated to a workload based on demand.

Cloud: AWS autoscaling
Kubernetes: HPA(Horizontal Pod autoscaling)

32) Difference between High availability and Disaster recovery in Cloud Environment?

High Availability	Disaster Recovery
Website/service is accessible even if one or servers is down	Ability to continue providing service even in the face of a regional network outage (when multiple sections of the world are rendered unreachable).
Auto-Scaling	Multi-Region Deployment,Failover Mechanisms:

33) What is VPC?

Creating and isolating resources within Virtual Private Clouds (VPCs), organizing them into subnets, and controlling traffic using security groups and network ACLs.

34) Difference between security group and network acl?

Security Group	Network ACL
Control inbound and outbound traffic at the instance level,	Operates in VPC subnet level
Supports only Allow Rules	Supports both allow and deny rules

35) What is Serverless Computing?

Serverless computing is a cloud computing model where the cloud provider automatically manages the infrastructure, allowing developers to focus solely on writing and deploying code. In this model, you don't have to manage servers or worry about scaling, as the cloud provider dynamically allocates resources as needed.

One of the great qualities of this model is that you pay only for the compute time your code actually uses, rather than for pre-allocated infrastructure

36) Name some ansible modules and uses?

apt or yum or package module: Used to install software or packages in Operating systems

Users: setup non-root users for login.

`service/systemd : To make service up and running`

`Lineinfile: replace some pattern in a file`

Ping: to check if ansible hosts are reachable

Setup: to gather information about Target hosts

ansible remotehost -m **setup** -u ec2-user

Command: To execute commands on remote Node

Script: To execute scripts

Copy: to copy files from remote source

Get_url: Download files using http remote url

File: creates files

Blockinfile: Insert/update/remove block of multiline text

Debug: For debugging variables/expression and print execution

```
---
- name: debug module
  hosts: all
  tasks:
  - name: Print a simple statement
      debug:
      msg: "Hello World! A custom message"

  - name: Get uptime information
    shell: /usr/bin/uptime
    register: result

  - name: Print return information from the previous task
    debug:
      var: result
      verbosity: 2
```

37) What is Devsecops?

Implement security in a DevOps pipeline (DevSecOps) and integrate security practices throughout the development and deployment process.

SAST → DAST (Dynamic security testing)→helm lint—> Container scanning(snyk)-> Deployment

38) What is the Kubernetes operator ?

Kubernetes offers limited initial functionality to ensure flexibility and scalability. K8s Operators are software extensions that make use of Kubernetes APIs to extend behavior.

Use Cases:

- The ability to deploy an application on demand;
- Making a backup of an application state or restarting an application from a given backup;
- Managing the update of an application with all its dependencies including new configuration settings and necessary database changes;
- Exposing a service to applications that do not support Kubernetes APIs.

39) What is Inode in linux ?

In Linux, an inode, or index node, is a data structure that stores metadata about files and directories in a file system

Each file and directory in Linux has an inode assigned to it when it's created. Inodes are stored in blocks, just like files are stored in units of a given size.

Check the inode number in a specific file

There are different ways to check the inode number. The following example shows the creation of a file named **mytestfile**. The command `stat` displays the file statistics, including the unique inode number:

```
[root@Rackspace-Server /]# touch mytestfile
[root@Rackspace-Server /]# stat mytestfile
File: mytestfile
Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: ca01h/51713d    Inode: 13          Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Context: unconfined_u:object_r:etc_runtime_t:s0
Access: 2021-03-26 15:51:27.036124392 -0500
Modify: 2021-03-26 15:51:27.036124392 -0500
Change: 2021-03-26 15:51:27.036124392 -0500
Birth: -
```

You can also check the inode number of **mytestfile** by listing the contents of the directory. You can run a combination of commands in the directory by using `ls` or `grep`, as shown in the following examples:

```
[root@Rackspace-Server /]# ls -lhi | grep mytestfile
    13 -rw-r--r--.   1 root root    0 Mar 26 15:51 mytestfile
```

```
[root@Rackspace-Server /]# ls -i mytestfile
13 mytestfile
```

40) What is /etc/fstab used for?

Used to mount file system

41) Command to find server Load and other information

top

```
Processes: 504 total, 4 running, 500 sleeping, 3493 threads
Load Avg: 5.65, 5.53, 6.12  CPU usage: 17.53% user, 8.15% sys, 74.31% idle   SharedLibs: 552M resident, 116M data, 45M linkedit.
MemRegions: 409617 total, 3767M resident, 155M private, 4030M shared. PhysMem: 15G used (1915M wired, 4856M compressor), 82M unused.
VM: 216T vsize, 4915M framework vsize, 636194(0) swapins, 1033066(0) swapouts. Networks: packets: 173000480/176G in, 49926487/22G out.
Disks: 110485213/1560G read, 56082443/813G written.

PID    COMMAND      %CPU TIME     #TH   #WQ  #PORTS MEM    PURG   CMPRS  PGRP  PPID  STATE    BOOSTS           %CPU_ME %CPU_OTHRS UID  FAULTS
4950   com.apple.Vi 45.0 68:09:58 24    2    86     7975M  0B     9643M- 4950  1     sleeping *6[3]            0.00000 0.00000    503  483648
398    WindowServer 31.1 26:58:23 21    5    4575+  1034M+ 37M-   355M-  398   1     sleeping *0[1]            0.03385 4.56256    88   902679
1163   Electron Hel 21.9 13:36:42 16    5    159    42M+   2336K  17M-   977   977   sleeping *0[1]            4.27266 0.00000    503  184388
1182   Electron Hel 17.5 10:59:31 22/2  1    156    77M+   0B     31M    977   977   running  *0[1]            0.00000 0.00000    503  122246
0      kernel_task  10.6 21:59:16 742/8 0    0      26M    0B     0B     0     0     running   0[0]            0.00000 0.00000    0    67388
457    coreaudiod   8.7  06:11:56 14    6    4067   38M    0B     26M    457   1     sleeping *0[1]            0.00000 0.00000    202  531599
67995  top          7.9  00:01.19 1/1   0    34+    7346K  0B     0B     67995 90156 running  *0[1]            0.00000 0.00000    0    5813+
```

42) What are iptables in linux?

It is a firewall to allow and deny access to certain services/ports

Example:
sudo iptables -A INPUT -s 203.0.113.51 -j REJECT
sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

43) What are Cgroups (control groups) in linux?

Used to limit resources(CPU,mem) that specific user can access in system

Commands:
cgcreate -g cpu,memory:/<cgroup_name>. #Creating a cgroup with the user or users you want to control.
cgset -r cpu.cfs_quota_us=50000 -r memory.limit_in_bytes=1G <cgroup_name>. # Limit CPU usage to 50% and memory usage to 1GB:
cgclassify -g cpu,memory:<cgroup_name> <username> #add a user to the cgroup.

44) Important Linux commands with explanation

find /path/to/directory -type f -mmin -30. # Find all files that are modified in last 30 minutes

Crontab -l # To schedule recurring tasks

find /path/to/start/directory -depth -type d -empty -exec rmdir {} \; # Delete all empty dirs

netstat -tulp # List open ports in linux server

grep <pattern> # Search pattern in Text files

ps aux --sort=-%cpu | head -n 10. # Top 10 process of CPU usage

45) What are Terraform providers?

They are drivers which help terraform to connect to Cloud /Onprem resources or services

```
# The default provider configuration; resources that begin with `aws_` will use
# it as the default, and it can be referenced as `aws`.
provider "aws" {
  region = "us-east-1"
}

# Additional provider configuration for west coast region; resources can
# reference this as `aws.west`.
provider "aws" {
  alias  = "west"
  region = "us-west-2"
}
```

46) Name some CICD Tools and uses

Jenkins,Gitlab,Circle ci,Azure devops pipeline, AWS code deploy ….

47) How to reduce costs in Cloud/On prem resources?

Script to shutdown instances during off working hours
Autoscaling using spot instances

48) What are types of provisioners in Terraform?

Remote exec: Run commands using Terraform on a remote server
Local exec: Run commands using Terraform on the local system

49) What databases familiar?

Mysql,postgres,Oracle

Command to Take mysql backup of a database

mysqldump -u [uname] -p db_name > db_backup.sql.

50) What are different types of Http requests?

GET: Fetch Data

POST: Post or submit data to server

PUT: Update a resource or data on the server

PATCH: Applies partial updates to a resource on the server

HEAD: To retrieve headers