*During the interview*

Normally before the end of interview, interviewer will ask if you have any questions. Then ask questions like.

1) What is the role which I would be playing if I could make this?

2) Is it entirely new project or an existing project to which I will be joining for further enhancements?

3) What tools are you using?

*\*Q1. Intro: Tell me about yourself \**

I am currently working as a Sr DevOps Engineer in xxx organization. I have total xx years of exp, out of which xx years into DevOps. As part of my DevOps role, I got to work on many tools

1) Version Control tool as Git and SCM (Source Code Mgmt) as GitHub Enterprise, Bitbucket or Azure Repos.

2) CI/CD Automation using Azure DevOps Pipelines and Jenkins.

3) I am involved in app modernization of our applications, as part of this worked with docker, writing dockerfiles/ generating docker image and deploying to Managed Kubernetes Service(s).

4) Worked with infrastructure as code provisioning tools using Terraform scripts and change management automation using Ansible.

5) I did work with both AWS and Azure Cloud Providers.

· AWS (VPC, EC2 Instances, S3, Lamba Functions, RDS – MySQL, ECR for Private Docker Registry, EKS for Kubernetes and IAM Role Management)

· Azure (Virtual Networks, Azure VM, App Service Plan, Azure Functions, Azure SQL DB, ACR for Private Docker Registry and AKS)

-- **More timing given**

6) I am pretty used to monitoring tools like CloudWatch Logs (in AWS) and Azure Monitor (in Azure), Prometheus and Grafana visualization for Kubernetes Cluster Monitoring

7) More Experience into shell scripting and entry-level with Python.

.*Q2. Tell Me about your project / client (SPECIFIC TO EACH CANDIDATE RESUME)*

In the transition from EC2 instances to a more scalable and containerized architecture, our team has come up on a comprehensive project to migrate Data Science and Machine Learning applications to Amazon EKS (Elastic Kubernetes Service).

* So we are maintaining the source code in Git VCS.
* Starting with the checkout of source code locally I start dockerizing the application by creating minimal images and removing all extra binaries from the container image ensuring image security. I then test it locally, all the docker images are pushed to Amazon ECR (Elastic Container Registry) with minimal access to users.
* As a part of the deployment process, I use terraform to create objects/ resources on AWS for each of the applications into different environments dev, staging and prod employing EKS blueprints.
* For deployment flexibility and consistency I also create the Helm charts which are customizable using a set of YAML configuration files that define the deployment, services, and other resources needed, where we maintain pre configured kubernetes resources to automate the deployment process of our application onto the EKS environment.
* For all the application level secrets I use AWS secret manager to store them with a secret rotation policy incorporated. For all the applications which needs access to ML models I mount these models/static files to Volume mounts or EFS(Elastic Files System).
* For all the environment-specific variables I maintain .env files on the fly categorized by environments(dev/staging/prod) which we store in the Jenkins server which are dynamically picked during the Jenkins build process.
* The CI/CD pipeline is orchestrated through Jenkins, automating the deployment process to Kubernetes.

Once deployed I check whether the pods are running, have completed successfully, or have encountered issues by determining the health checks within the pod with a mechanism of liveness and readiness probes. If I encounter any issues I finally go with troubleshooting to make the pods up and running successfully.

.*Q3. Tell Me about your second project / client (SPECIFIC TO EACH CANDIDATE RESUME)*

The main aim of the project is to migrate all the servers from on premises to Azure cloud.

Migration uses:

* Costing would be less for the firm.
* Maintenance is easy in the cloud when compared to on premises.

**Project:**

* As the Infrastructure Engineer within a 12-member team I am responsible for the comprehensive management and deployment of cloud resources for the entire company,
* My primary focus is on Azure, with occasional involvement in AWS.
* Utilizing Azure DevOps (ADO) pipelines and ARM templates in JSON format, our team efficiently deploys and maintains Virtual Machines (VMs), Resource Groups (RGs), and Storage Accounts, etc. Using pre-existing templates for various resources, we customize these templates according to the requirements and use them across different teams/environments to deploy various resources.
* Ticketing requests from different teams, coming through ServiceNow, form a crucial aspect of our workflow. Post-deployment, we continuously focus on managing permissions for Active Directory users and groups, restricting access levels based on organizational needs.
* As the company undergoes migration, my role extends to providing 24/7 on-call support during the weekend migration window, from Friday 7 PM to Sunday 11 AM.
* During this critical period, I troubleshoot any arising issues, ensuring a smooth transition. Additionally, my responsibilities include the installation of essential agents, such as the Microsoft Monitoring Agent (MMA) and OMS Linux agents, on VMs. I also use Automation scripts for deployment of these agents, contributing to monitoring capabilities and overall system reliability.

*Q4. What are your roles and responsibilities / daily routines in current project*

Currently we are 8 member team, involved into multiple projects deployments and infrastructure automation and maintenance. I am primarily involved in migrating applications which are running in Virtual Servers of On-Prem to Cloud and containerizing our existing applications as most of our applications are using microservices architecture shifted from monolithic, So writing dockerfiles for each microservice, modifying existing pipelines to build and push docker images to private registry, Scanning containers for vulnerabilities, deployment into non-prod and prod environments and maintenance of Kubernetes Cluster which is shared across multiple applications.

Troubleshoot issues we face as part of this journey and as a senior member of team I mentor junior team members and help them to expedite their tasks.

Maintenance of services what we have provisioned for our applications (Monitoring health checks and cleanup activities if server usage is more)

Taking care of User Permissions in Jenkins Server/ Azure Pipelines and Kubernetes RBAC (IAM)

Working on Automating / Changing Infrastructure Provisioning or software patching/installation using Terraform and Ansible.

Regular Kanban/Sprint Planning, Technical discussion for any architectural changes, Compliance issues, and documenting using confluence.

Defining Branching and Release Strategies and preparing plan for next releases and sending our release notes as per release cadence.

If we encounter any issues, having war rooms (sending up post mortem reports by doing Root Cause Analysis)

*\* Q5. Whats the Methodology which you are following in your project (Agile methodology)\**

We are following Agile methodology in our SDLC with scrum and kanban framework, 2 weeks and for each sprint when it comes to scrum board we follow (tasks like CI/CD, Infra Provisioning), during this we will interact with dev team / Product owners for the requirement gathering, once we have requirement, then we will story point the user stories with acceptance criteria, We are using Azure DevOps Boards /JIRA as a Project Management tool and we will be assigned with User Stories / Tasks. Once we are done with our defined work we would test them (Ex: Pipeline we would manually build applications and enable automatic trigger and monitor for few days until its stable) along with this, we parallelly define for the next sprint user stories and work on them based on team’s velocity.

We do have Kanban boards too when unknown (Issue Triaging and Fixing) or some process improvement backlogs are defined.

As part of SDLC Life Cycle we are deploying our application into multiple environments phases ( Dev-->Test-->UAT--> Staging -> PROD)

*\* Q6 Scenario 1 : How did you migrate applications to cloud?*

That’s an interesting one and it has long journey, We had adapted 3R Methodology (ReHost, Replatforming or ReArchitect) which basically says whether this on-prem application architecture has to be changed, or just host it like lift and shift)

We did some PoCs (Proof Of Concepts) to identity right service fitment in Cloud. As part of this we lifted and shifted applications to Managed Services like Lambda and Azure AppService and for Database we have chosen Database Migration Service provided by AWS and Azure Data Factory in Azure, as data is critical here wanted more secure and entire data is migrated correctly with encryption at rest and in transit with all firewall rules. At present we are modernizing our applications and deploying them to Kubernetes Cluster as part of Kubernetes Migration we did create 1 test cluster initially at US East Region and deployed using manifest files. Then later created 2 Prod Clusters at multiple regions to support our application across globe and to reduce latency, so handling using external Load Balancer.

As part of this journey Azure Migrate Service in Azure and AWS Migration Services helped us a lot.

*Q7) Scenario 2: what are some critical production issues you have faced and how you resolved them?*

1) We have some legacy applications, they are deployed in IIS Web Server of VMs, As this application is dependent on other apis within our organization to get environment variables and sometimes called api reaches rate limit and would throw issues and we reach max connections and application comes down, so we had to restart IIS each time to rest pool.

To eradicate that we have added caching layer (Redis Cache) to get key-values for 24hrs to limit restrictions to Central API to get data and invalidate it in next 24 hours, other approach put secrets in Keyvault and consume as alternative if Central api fails

Using Redis Cache in Public Cloud is costed heavily so we migrated our caching logic from Redis to Blob Storage which gave us almost most performance will less transaction and storage costs.

2) Using containerized apps, we faced issues with configmaps changes….if we change something in configmap for new release as this is immutable changes are not reflected, we need to reset pod to reflect changes

*Q8) Scenario 3: What was some automation you have done*

Very recently I have done a small utility, basically we get files from different clients for analysis in different formats and we process them through recommendation engine and sometimes the input flles we don’t receive them on time, so instead of manually checking each time to look in storage, I did a small utility which connects to storage and check for each client based on timestamp I defined and let me know if its received or not by sending message to Teams Channel…All these are achieved using Python Libraries like (boto3 in AWS or azure storage sdk in Azure)

*Q9) Scenario 4: what are the issues you have faced and how you resolved them?*

1) I recall recent issue which we faced, so when we run container in Kubernetes clusters, we face 2 kinds of issues one while deploying applications and pods wont be started, we can debug pods

using **kubectl describe pod <podname>** command to identify events and see event error(s) and troubleshoot

second kind of issues we faced are applications are deployed successfully but while running application pods would be exited, in recent times we had this issue where we run one of our service which actually read csv files and plot images using matplotlib library based on service bus message queues, and our pods are stopped after couple of messages being processed, when i troubleshooted this issue using Container Logs of Log Analytics and Identified Memory issue, pod is reaching memory limit we have set and stopping.. and the reason for this is in code we save figures based on plots and those images are not closed once plotted, due to which memory leak is happening and its hitting threshold...

Development Team had to quickly fix this based on my findings and deployed new tag to test cluster then to prod cluster as hotfix

2) And there were couple of issues with networking as well, some changes in network policies would affect our connection between services across same subscription within different VNETs and subnets...Used to monitor them using Azure Monitor and identify such issues and escalate it to network team.

3) When we are containerizing our applications and generating docker images, we followed multi stage docker build to reduce image size, as part of this we were left with lot of dangling images in Build Agent Server. So we wanted to prune those docker images to free up disk space. So written Cron Jobs with docker system prune and run it fortnightly to eradicate this problem.

Second and most common issue which we are facing each day, workloads on our build servers will be more as we have multiple teams working on with multiple services so running multiple jobs parallelly. To solve this issue we did bring up multiple slave nodes and use nodes slave wisely per service, As part of long-term we are going to deploy Jenkins into Kubernetes Cluster and auto-scale it based on CPU / RAM Threshold usage.

Finally issues with Pods not being started after deployment, so we need to check logs using kubectl or CloudWatch Container Logs and get more metrics from Prometheus and identify root cause.. most cases it turned out to be resource limit issues, so as best practice we try to set resources requests and limits on namespace and pod level too.

Apart from all these regular deployment issues when we setup new deployments, Permission Issues, Usage I read documentation and troubleshoot and make it fixed.

*Q10)* *Whats ur branching strategy and some git questions?*

We have used Feature /Task Branching strategy, having master and develop remote branches and set protection policies on those branches, create feature/hotfix/bugfix branches for each UserStory/Task and merge those commits to develop by raising PR.

We merge from develop to master for production deployments by creating release branches for each release. (Semantic Versioning x.x.x) => release/1.13.0

***Git fetch vs git pull***

When we do git pullit automatically merges the commits without reviewing them unless merge conflicts arises.

When we do git fetch => It doesn’t get files transferred, it just checks if there are new changes.

Git gathers any commits from the target branch that do not exist in your current branch and stores them in your local repo. However, it does not merge them with your current branch. To integrate the commits into your current branch, you must use merge afterwards.

***Git merge vs git rebase***

Merging preserves the entire history of your repository by creating new commit, while rebasing creates a linear history by moving your feature branch onto the tip of main

**How do your push ur changes to remote repo:**

Git add . => all files to be staged.

Git commit -m “message”

Git push origin <branch-name>

*Q11)* *How do we search and replace in bash scripting?*

sed => stream editor (to replace text) (sed ‘s/{beforetext}

Grep => search

df => disk free (disk space)

ps -ef => to list all processes

find . -name => Find files and folders based on pattern

*Q12) Have you configured VPC?*

Yes, We have configured VPC for each region by defining CIDR Range and created Public and Private Subnets based on availability zones

Route to those subnet using associations in route tables and talk to public internet using internet gateway where as for private subnets we have routetable configuration using NAT Gateway or VPC Endpoint.

*Q13) TCP vs UDP*

TCP is connected-oriented protocol which means it first establishes connection between devices then communicates by transferring data (three-way handshake)

UDP is connectionless IP Protocol which doesn’t establish connection first, and it doesn’t wait for the response back which makes it faster to deliver compared to TCP. UDP is widely used in Video/audio streaming (Ex: Zoom etc)

*Q14) Scenario 6: How can you read artifacts stored in S3 into EC2*

Create an AWS Identity and Access Management (IAM) profile role that grants access to Amazon S3.

Attach the IAM instance profile to the EC2 instance.

Validate permissions on your S3 bucket.

Validate access to S3 buckets using aws cli commands

*Q15) Please explain steps/workflow involved in your CI/CD Process?*

Checkout => During this stage we clone repo from SCM using git and get it to Jenkins workspace (GitHub Plugin)

And in Jenkinsfile We will have multiple stages in Pipeline

**Speaking about Traditional Way (Conventional):**

Build Application =>

Test Application => Running functional unit test cases

Sonar Qube Analysis => Generating Scan Reports and Check Quality Gate Status

Push to Artifactory

Deployment to Managed Services (Azure App Service / AWS BeanStalk)

Each Stage for each environment

DEV => Continuous Deployments

TEST

UAT => Once per each Sprint

STAGING / Pre-Prod

PRODUCTION => Approvals before deployment

**Containerization (Modern Way)**

Run Unit Tests

Build Docker Image=> we build dockers which run instructions of dockerfile and give new tag for each build

Push Docker Image => Once docker image is built, we tag it and push it to ECR (AWS) or ACR(Azure)

Sonar Qube Analysis =>

Deploy to Kubernetes => Here we are going to deploy to multiple clusters based on environments (1 non-prod cluster and 2 prod clusters), so for each environment we will have one stage (Kubernetes Deploy Plugin)

Each Stage for each environment

DEV => Continuous Deployments

TEST

UAT => Once per each Sprint

STAGING

PRODUCTION => Approvals before deployment

Email => Finally we trigger email if deployment is succeeded or failed

**Continuous Delivery vs Continuous Deployment:** Continuous Delivery takesmanual approval to deploy specific release which is planned to production. With continuous deployment, production happens automatically without explicit approval.

*Q16) Scenario 5: What all different plugins you installed and how jenkins pipeline is getting triggered automatically?*

1) GitHub Plugin to connect to Source Code Repos and Authenticate it using Jenkins Credentials (Username and Password)

2) Maven Integration to Build Maven Projects with Pom.xml file and Goals

3) Docker Pipeline for Build and Push Docker Images to Repositories

4) Kubernetes Deploy => To Kubernetes Deployment to EKS

5) Blue Ocean => For more intuitive Jenkins UI

Jenkins Build Pipeline will be triggered when there is a change in Source Code Automatically by configuring Webhooks by providing Jenkins Url and sending Push events

Or by doing Pooling Mechanism of SCM by setting up CRON Expression every 5 mins or so but which is not a good practice though

*Q17)* *Scenario 6: How will you write dockerfile for dotnet/Java/Python Application?*

For any Application to containerize, we follow these steps and we follow multi-stage builds

FROM command to use BASE IMAGE, for Java Application we can use OPENJDK or MAVEN

COPY Command => Which copies all files from source code

RUN Commands => Which generates Jar/War

ENTRYPOINT or CMD Commands at the end which executes when container executes

**What commands you use in docker**

1) Docker build -t <service-name>:<tag> -f “path of dockerfile” .

2) Docker push => to push docker images to Private/Public Registry

Q18) Scenario 7: What did you do using Ansible

1) I have written a simple playbook, which installs docker into multiple nodes using yum module as we use RHEL 7 Linux Servers as our Build Agents

2) I have also used it for Copy Activity, to copy sudoers file across nodes using copy inbuilt module.

*Q19)* *Scenario 8: How do you debug pods in Kubernetes?*

We use kubectl commands during deployment failures and run

**Kubectl describe pod <pod name>** ---- it gives complete information about pod infra

**kubectl exec -it <pod name>** -**-sh** – SSH to given pod interactively to see content

How you deploy application to kubernetes cluster

**Kubectl apply -f “<manifest-files>.yml” or kubectl create -f “manfest”**

*Q20) What are different kind of Agents we have in Azure Pipelines and which one you used ?*

We have Microsoft-Hosted Agents and Self Hosted Agent.

We configure agents using Agent Pool and can add Multiple Agents in Pool.

We are using Self Hosted Linux Agent, Configuring and running agent service in linux machine.

*Q21) How does Azure Knows which agent to run for pipelines?*

Azure Pipelines is written in YML Language, and in azure-pipelines.yml file we specify pool with name or pool with vmImage

Syntax goes like this

Pool

Name: <agent-pool-name>

Or

Pool

vmimage: <image-name*>*

*Q22) What different monitoring tools have you used?*

I have used CloudWatch and CloudTrial Logs in AWS

Log Analytics Workspace and App Insights in Azure

And Prometheus and Grafana Dashboard for Kubernetes metrics logs, By using Alert Manager, setting thresholds and raising incidents to ServiceNow

*Q23)* *What/why terraform and what is state file and what did you create using terraform?*

Terraform is an open-source infrastructure as codecloud**-**agnostic software tool that enables us to create and modify infrastructure. As its Cloud-agnostic we can create infra using various providers especially with major Cloud Service Providers (aws, azurerm and gcloud)

**Statefile**: Westore current state about our infrastructure which is provisioned into statefile and we store it remotely into Azure Blob Storage Container in Azure and S3 Object in AWS which will be configured as backend.

I have created EC2 along with Security Groups and Auto Scaling Group by getting existing subnetId

Creation of Azure App Service

Creation of EKS Cluster

**What commands you use in Terraform**

1) Terraform init => To initialize terraform backend and providers

2) Terraform plan => dry-run (It will generate a plan)

3) Terraform apply => By Executing this command resources will be provisioned in Provider we mention (AWS, Azurerm)

*Q24)* *How can we provision EC2 using terraform?*

Firsly we will define provider as aws and use secret key and access secret to authenticate in main.tf

Then we set variables , will define all required resources like vpc, subnetid, ami\_id, instancetype which are required for EC2

Finally will create EC2 Instance resource and Security Group using “aws\_instance” by passing variable file (variables can be defined in separate file Ex test.tfvars)

Then we can print output of terraform using output modules

Terraform Commands :

Terraform init (Initialization of terraform main.tf file)

Terraform plan (dry run)

Terraform apply –var-file=test.tfvars (which provision resources)

*Q25) What are different components in Kubernetes ?\*\*\*\*\*\**

Control Plane / Master Node Components:

Kube-apiserver => Front End for Kubernetes control plane

Etcd => Storage of cluster data

Kube-scheduler => Assign Nodes to Pods

Node Components:

Kubelet

Kube-Proxy

Pod is the smallest unit of Kubernetes which hosts (runs) docker containers

*Q26) What is daemonset and when do we use it in Kubernetes?*

A DaemonSet ensures that all Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created. It runs the pod as background service Ex: Prometheus should run in all nodes to collect metrics from each node so we run them as daemonset

*Q27) Scenario 7: How to configure RBAC in Kubernetes*

1) *Create Service Account =>* **identity for pods**

2) *Create Role which are specific to namespaces or ClusterRole where you have permissions for entire cluster*

3) *Create RoleBindings which will map ServiceAccount to role or ClusterRoleBindings for entire cluster*

*Q28) Scenario 8: How to expose Pods to outside world / public*

We can do that using Services, and we have 3 types of services which is mapped to pod using selector

1) NodePort => listens to specific port ranging from 30000 - 32000

2) ClusterIp => Within cluster to communicate between multiple pods and services (No Public Communication)

3) LoadBalancer => Which creates external IP address for each Service