Self intro:

Hi this is XXXXX working as DevOps Engineer for XXXX company
for last XX Years.

Currently i have been working for XXXX client where our team is managing
2 applications.

As a DevOps engineer i make sure the environment is clean and running.

I have worked on various Devops Tools like Git,Github,Jenkins,maven,Ansible,Docker,Terraform and Kubernetes for orchestration,

Aws cloud and physical Data Centers.

In aws i have worked on various services like ec2,s3,route53,application load balancer,VPC,cloud watch,cloudtrial,elastic container service and elastic container registry.

Roles and Responsibilities:

we worked along with development teams for next releases,where our team is responsible for UAT and PROD releases.

We have weekely release and also flex release which will happen for every 2 or 3 months.

If we have any monitoring alerts then i will be working on that alerts.

Managing jenkins pipelines and other infrastructure.

working on patching activities along with linux team for every quarter
to make sure the kernels are updated,our team is responsible for
pre patching and post patching activities.

We provide 24/7 support means pagerduty if high priority tickets then will
get a call on mobile and will look into that issue.

Prepatching and post patching activities (Answer only if asked):

Patching activities..

Tell that we support patching activities..

We do pre patching and post patching activities.

Pre patching means bringing down the traffic by disabling the health check, brining down all the services running in the servers.

We have ansible playbooks to do this job.

Post patching means brining up all the services running in the servers and enabling the traffic back.

Team (Answer only if asked)

We have a team of 6 members from offshore and we work 24/7.
From onshore we have our manager and scrum master.

Managerial Questions:

1)Tell me something which is not mentioned in your resume?

My family is not mentioned in my resume,basically my father is ____ Worker,he has been very hard and supported our family and he is my role model as well.I Have learned many things like love what we do and be dedicated toward your work.My mother is housemaker and i have learnt time management and the way she supported family.

If you have sister/brothers just tell there name and profession.

2) What is INCIDENT and PROBLEM management?

Any issue/error can be marked as incident,group of similar Incidents can be marked as a Problem.

3) Where would you like to see yourself after 2-3 years?

I would like to see myself at a position handling a team/project.

4) What if your project has not done on sprint duration,how will you manage the client?.

First thing is based on priority of work,i will try to complete that on time, but if still i was unable to complete then I will explain my manager/client about the process to finish the task, and the benefits of the particular process in improving the quality.

If i got blocked then will try to reach out team and get the enough information to clear the issue.

5) What is client has changed the requirement for the sprint?

If the Client changed the requirement then i will try to put extra time and finish my work.

and also when i get a requirement from client then will ask to share the email with requirement, so that there will be no sudden surprises in changing the requirement

Interview Questions:

DevOps:

1.What is DevOps?

Devops is a methodology which help to remove the gap between development and operations team.

Implementation of devops will help us to track the bugs at initial stage and also release the applciation faster.

2.What are the different stages of DevOps?

Plan
Code
Build
Test
Release
Operate
monitor and
Feedback.

3.What are the methodologies before DevOps?

Waterfall model and agile process.

4.Have you worked on Agile Process?

Yes,where we have scrum master who will be assigning set of task for sprint.

5.What is the Sprint duration?

2 weeks.

6.Difference between IAAS,PAAS and SAAS?

SaaS (Software as a service) platforms involve software that is available via third-party over the Internet.

Examples of popular SaaS providers include:
BigCommerce.
Google Workspace, Salesforce.
Dropbox.
MailChimp.
ZenDesk.
DocuSign.
Slack.

Hubspot.

PaaS (Platform as a service) focuses primarily on hardware and software tools available over the internet.
Examples of popular PaaS providers include:
AWS Elastic Beanstalk.
Heroku.
Windows Azure (mainly used as PaaS).
Force.com.
Google App Engine.
OpenShift.
Apache Stratos.
Adobe Magento Commerce Cloud.

IaaS(Infrastructre as a service) works primarily with cloud-based and pay-as-you-go services such as storage, networking and virtualization.
Examples of popular IaaS providers include:
AWS EC2.
Rackspace.
Google Compute Engine (GCE).
Digital Ocean.
Microsoft Azure.
Magento 1 Enterprise Edition.

Linux:

1.Tell 10 linux commands?

Netstat -na -- To check the running port numbers
Ps -ef -- TO check the running services
find / -name filename -- find a file
grep keyword filename -- Find a keyword from a file
mkdir -- to create the direcotry
top -- cpu utilization
sed -- to replace a text from file at run time
kill -9 -- to kill the process

touch filename -- to create zero byte size file

tar -xvf filename.tar -- TO bundle a directory

2.Command to check for the running ports?

netstat -na | grep port_number

3.Command to check the processes running?

ps -ef

4.Command to replace a word from file?

sed -i 's/old-text/new-text/g' input.txt

Examples for SED and AWK commands?

awk example : https://www.geeksforgeeks.org/awk-command-unixlinux-examples/

Sed Example : https://www.cyberciti.biz/faq/how-to-use-sed-to-find-and-replace-text-in-files-in-linux-unix-shell/

5.What is swap memory?

When the actual memory of RAM is full then swap memory is used to process the request.

6.Command to kill a process?

kill -9 PID

7.process to free disk space?

When we see memory alert then follow the below steps:

1)Login to the ec2 using ssh

2) df -h -- shows the filesystems memory and check which filesystem is occupied with space example /var

3) cd /var -- Change to var directory

4) du -sk -- TO check the each directory size in var

5) Delete the directory / clear the sapce which has consumed more space.

8.What is Inode?

An inode is an index node. It serves as a unique identifier for a specific piece of metadata on a given filesystem.

Each piece of metadata describes what we think of as a file.

9.We have disk space available but still unable to create a file,what might be the reason?

Basically an inode is used for each file on the filesystem.

So running out of inodes generally means we are unable to create file even if we have filesystem.

10.Command to find a file and delete the file using the same command?

```
        find / -name "FILE-TO-FIND" -exec rm -rf {} \;
```

Bash Scripting:

1. #! in bash scripting means = Shebang

2. Difference between bash and shell script?

Shell scripting is a method to automate tasks as a collection of commands.

Shell script can be execute on any shell.

Example of shell scripting:

```
#!/bin/sh
myString="GeeksforGeeks"
echo "myString: $myString"
```

Bash: Bourne after shell.

Bash scripting is a subset of shell scripting.

Example of bash scripting:

```
#!/bin/bash
myString="GeeksforGeeks"
echo "myString: $myString"
```

3.Write a bash script to check a directory in a location,and should dispaly the message if the directory is available?

```
#!/bin/bash
if [ -d "/path/to/dir" ]
then
   echo "Directory /path/to/dir exists."
else
   echo "Error: Directory /path/to/dir does not exists."
fi
```

4.Write a bash script to check if a service is running or not,if not running then script should start the service?

```
#!/bin/bash
service=replace_me_with_a_valid_service
if (( $(ps -ef | grep -v grep | grep $service | wc -l) > 0 ))
then
```

```
echo "$service is running!!!"
else
/etc/init.d/$service start
fi
```

5. How to know whether the previous command has been executed successfully or not?

With the help of exit status.
If exit status is 0 then it has been successfully executed.
If Exit status is 1 then it has not been successfully executed.

6. $? is the exit status of the most recently-executed command.

7. $# = Total number of command line arguments passed.

8. $* = It's a space separated string of all arguments. For example, if $1 is "hello" and $2 is "world", then $* is "hello world".

9. $@ = Stores all the arguments that were entered on the command line.

10. $_ = special variable

11. How to run bash script in the background?

Use nohup to run the script in background.
Example:

```
sudo nohup ./hello_world.sh
```

12. Which version of linux are you using?

We can say RHEL 7.9 or amazon linux
Note: The versions will get changed as per time.

ERROR CODE:

200, 301, 404, & OTHER NUMBERS: HTTP ERROR CODES

You've probably seen it before: 404 Page Not Found.

404? What's that?

It's an HTTP status code.

There are a surprising number of HTTP status codes aside from the two or three you may have run into in the wild. They exist to represent the status of requests that browsers make to load websites, and they're split into five groups.

1XX — Informational Responses

2XX — Success Responses

3XX — Redirection Responses

4XX — Client Error Responses

5XX — Server Error Responses

You can see these codes in action if by using developer tools. With Chrome, just press F12 (on Windows) or ⌥⌘I (on macOS), then select network at the top, and reload the page you're on.

We'll be highlighting some well-known and a few not-so-well-known codes.

1XX Codes

An informational response lets the browser know that the request to load a website or document was received and understood. Codes like these are issued while the request processing continues on the server, and it lets the browser know that it should wait for a final response.

100 — Continue

A status code of 100 indicates that (usually the first) part of a request has been received without any problems, and that the rest of the request should now be sent.

2XX Codes

A success response lets the browser know that the request was received, understood, and accepted — that's what separates them from the 1XX codes.

200 — OK

This is the code that browsers receive when every has gone according to plan.

201 — Created

This code indicates that a request was successful and as a result, a resource has been created (for example a new page).

204 — No Content

The 204 status code means that the request was received and understood, but that there is no need to send any data back.

205 — Reset Content

This code is a request from the server to the client to reset the document from which the original request was sent. For example, if a user fills out a form, and submits it, a status code of 205 means the server is asking the browser to clear the form.

206 — Partial Content

This is a response to a request for part of a document. This is used by advanced caching tools, when a browser requests only a small part of a page, and just that section is returned.

3XX Codes

These status codes tell the browser that it must take additional action to complete the request. Many of these status codes are used in URL redirection.

300 — Multiple Choices

The 300 status code indicates that a page or document has moved. The response will also include a list of new locations so the browser can pick a place to redirect to.

301 — Moved Permanently

This tells a browser that the resource it asked for has permanently moved to a new location. The response should also include the location. It also tells the browser which URL to use the next time it wants to fetch it.

304 — Not Modified

The 304 status code is sent in response to a request (for a document) that asked for the document only if it was newer than the one the client already had. Normally, when a document is cached, the date it was cached is stored. The next time the document is viewed, the client asks the server if the document has changed. If not, the client just reloads the document from the cache.

307 — Temporary Redirect

307 is the status code that is sent when a document is temporarily available at a different URL, which is also returned. There is very little difference between a 302 status code and a 307 status code. 307 was created as another, less ambiguous, version of the 302 status code.

4XX Codes

This type of status code is intended for situations in which an error seems to have been caused by the browser or user, like the infamous 404 error.

400 — Bad Request

A status code of 400 indicates that the server did not understand the request due to bad syntax.

401 — Unauthorized

A 401 status code indicates that before a resource can be accessed, the client must be authorised by the server.

402 — Payment Required

The 402 status code is not currently in use, being listed as "reserved for future use". It's interesting to think about how this will be used in the future, especially now that Chrome natively blocks some intrusive ads.

403 — Forbidden

A 403 status code indicates that the client cannot access the requested resource. That might mean that the wrong username and password were sent in the request, or that the permissions on the server do not allow what was being asked.

404 — Not Found

The best known of them all, the 404 status code indicates that the requested resource was not found at the URL given, and the server has no idea how long for.

408 — Request Timeout

A 408 status code means that the client did not produce a request quickly enough. A server is set to only wait a certain amount of time for responses from clients, and a 408 status code indicates that time has passed.

410 — Gone

A 410 status code is the 404's lesser known cousin. It indicates that a resource has permanently gone (a 404 status code gives no indication if a resource has gone permanently or temporarily), and no new address is known for it.

415 — Unsupported Media Type

A 415 status code is returned by a server to indicate that part of the request was in an unsupported format.

5XX Codes

Simply put, these codes are sent when the server failed to fulfil a request.

500 - Internal Server Error

A 500 status code (which developers see more often that they want) indicates that the server encountered something it didn't expect and was unable to complete the request.

503 — Service Unavailable

A 503 status code is most often seen on extremely busy servers, and it indicates that the server was unable to complete the request due to a server overload.

503 Isn't So Bad

If you ever see one of these errors on your own website, and you don't know what to do, take a look at this list. With this, you'll be able to let us (if we host your website) know what's actually going on when your website looks like it's broken.

Git & GitHub:

1.What will git init do ?
This will help us to initialize .git folder in local repository.

2.What does .git contains?
.git file contains the configuration of the repository,like hooks,config etc..

3.Why do we use git ?
Git is a version control tool which is used to track the changes of the project.

4.Which branching strategy are you using?
We use GITFLOW startegy.

> **Release branch** is used for PRODUCTION Environment.
> **Develop branch** is used for development/staging Environment.
> **Feature branch** is used for next updates.
> **Hotfixes** is used for fix the bugs of Production server.
> **Master** is used as a copy of RELEASE branch.

5.Git cherry-pick?
Cherry-pick is used to pick a specific commit of a branch to another branch.

6.Git stash?
For example if iam working on a specific branch and i don't want to save or commit the changes, in the mean time i got the request to work on other file from same branch then i can use gitstash.
This will help me to push the files to temporary stash memory.

7.Git rebase?
This is used to Merge two different branches.

8.Git merge?
This is used to merge two different branches.

9.Git fork?
This is used to copy a repo from other account to our account on Github.

10.Difference between git clone and git pull?
Git clone is used to download the complete repository from central to local.

Git pull is to pull the latest changes of code from central to local.

11.Difference Between Git merge and Git Rebase?

Git merge and Rebase are used to merge the two branches but git rebase will keep the commit history more clear than git merge.

12.Difference between Git pull and git fetch?

Git pull is to pull the latest changes of code from central to local.

Git fetch is used to pull the metadata configuration to local like branches,configuration,HEAD,Ref etc..

13.Is Git a distributed repository system or central repository system?

Git is a central repository system.

14.Difference between Distributed and Central Repository?

The main difference between centralized and distributed version control is that, in centralized version control, the versions are saved in the remote repository, while in distributed version control,versions can be saved in the remote repository as well as in local repositories of the local machines.

15.Which version of git are you using?

git version 2.34.1.windows.1

16.What are merge conflicts?

If we have two similar files in different branches with similar content,

when we try to merge these branches then there is a chance to see merge conflicts.

This happens because git will not apply the auto-merge strategy if the file is in same region.

17.How to resolve the merge conflicts?

By default git will Auto-Merge many chnages,but if we see merge conflicts then git will leave that to us to resolve.

In this case there is no automation way to resolve the conflicts.

we need to check the differences in both the commit and manually make changes and push them back to repo.

This will help us to resolve the merge conflicts.

Note: Mostly Merge conflicts will be taken care by Development Team as they are aware of the code changes.

18.How to merge multiple commits into a single commit.

We can use squash to merge multiple commits into single commit.

Apache Tomcat:

1.Default port number of Tomcat?

Tomcat runs on 8080 port number.

2.Default Deployment location of Tomcat?

Webapps will be the default deployment location.

3.Can we run multiple tomcat servers,If yes then please explain the process?

Yes we can run multiple tomcat servers by changing the startup and shutdown port numbers of one server.

4.How to create Users in tomcat?

We can create users in tomcat-users.xml file.

5.Tomcat is Developed in ___________ language?

Java language.

6.What are the dependencies for tomcat?

Java is the only prerequisite for tomcat.

7.Apache Tomcat is ________ server?

Application Server.

8.Difference between app and web servers?

App server is mainly used for to serve dynamic content.

Web servers are used to serve static content.

9.list of app servers available in the market?

Tomcat

Jboss

Weblogic

10.Which version of tomcat are you using?

Tomcat-9.0.65

11.File name of Apache Tomcat Logs?

catalina.out is the log file for tomcat.

12.How to change the port number of Tomcat?

Go to tomcat>conf folder

Edit server.xml

Search "Connector port"

Replace "8080" by your port number

Restart tomcat server.

Nginx and Apache:

1.Default port for apache?

Default for Apache is 80.

2.Default port for nginx?

Default port for Apache is 80.

3.Difference between apache and nginx?

Apache and Nginx both are web servers.

Nginx can be used as load balancer and we dont need any extra configurations.

Apache can be used as load balander but we need to do extra configurations.

Nginx is pretty fast than apache in serving static content.

4.Default deployment location for apache and nginx?

Apache:/var/www/html/

Nginx: /usr/share/nginx/html

5.Log File names of apache and nginx?

Apache: /var/log/httpd/access_log & /var/log/httpd/error_log

Nginx: /var/log/nginx/access_log & /var/log/nginx/error_log

6.How to change the default port of apache and nginx?

Apache: /etc/httpd/conf/httpd.conf

Nginx: /etc/nginx/nginx.conf

HA PROXY:

1.What is HA Proxy?

HA proxy is the high availability, open source load balancer, reverse based proxy which is used for maintaining the website to work smoothly without causing any overload.

2.What is reverse Proxy?

Reverse proxy server will be at the front of the servers such that when the users sent the request the reverse proxy server acts as load balancer and distributes the traffic to the servers. Due to this the users can't directly intercat with the servers.

3.Configuration file name of HA PROXY?

haproxy.cfg

4.How to configure HA Proxy?

We can configure the haproxy at etc/haproxy/haproxy.cfg

AWS:

1.List of aws services you have worked on?

I have worked on IAM,ec2,s3,applciation loadbalancer,Route53,VPC,CLoud watch,cloud trial, ecr and ecs.

2.What is region and availability zone in aws?

A region is the collection of avaialbility zones/data centres. Availabilty zone is an independent data centre.

3.what is edge location?

Edge locations are used for delivering the content with low latency and high transfer rate.

4.Name the global services of aws?

S3, IAM, cloudfront and route53

5.What is aws access key and secret key?

AWS access key and secret key are generate for connecting to the AWS Account through the CLI.

Ec2:

1.Ec2 stands for?

Elastic Cloud Compute.

2.How many types of ec2 are available?

On Demand,Reserved and Spot Instances.

3.Different families of ec2?

General purpose, storage optimized, compute optimized, memory optimized, accelerated computing.

4.What is the type of ec2 you are using?

On Demand (t2 large -- 2 core cpu and 8 gb memory)

5.Can we attach multiple ec2 to one ebs volume?

No We can't attached multiple ec2 to same ebs volumes,but we can attach multiple ebs to single ec2.

6.What if the pem key of ec2 is lost?

If the pem key is lost then we need to take the sanpshot of the ec2 and launch a new ec2 using the snapshot using a new PEM key.

7.Can we connect to ec2 if the pem is lost,if yes then explain the process?

No we can't connect to ec2 without the pem key.

Only way is possible to take snapshot and launch using new pem key or if we have copied the ssh keygen to the ec2 earlier then we can login without pem key.

8.How to take a snapshot of ec2?

Select the ec2,go to settings and click on snapshot.

9.What is a snapshot?

Snapshots are a point of time copy of the ebs.

Example i have launched one amazon linux ec2 and installed jenkins and took snapshot then the same snapshot can we used to launch new ec2 and by default it will be havings jenkins in it.

10.A snapshot of t2.micro has been taken,can we change the type of instance when using the snapshot to launch new ec2?

A snapshot of t2.micro can be changed to t2.large.

11.What is the 2/2 system checks in ec2?

1) System connectivity Checks

2) Instance Status Checks

12.What are the possible reasons for system 2/2 checks?

A) System connectivity Checks:

Loss of network connectivity

Loss of system power

Software issues on the physical host

Hardware issues on the physical host

B) Instance Status Checks:

Mis-configured networking or startup configuration

Exhausted memory

Corrupted file system

Incompatible kernel

13. What can be done if we see system checks as failure?

General scenario we can restart the instance and we can see the system checks will be normal as the instance will get hosted/launched on another hardware stack.

14.Difference between SG and NACL?

SG AND NACL both are firewalls which helps to restrict unwanted traffic.

SG works at instance level and by default everything is denied in SG.
These are stateful, which means any changes which are applied to an incoming rule is automatically applied to a rule which is outgoing.

NACL works at subnet level and we can add rules to allow and deny the source.
NACL works on rule number.
These are stateless, meaning any change applied to an incoming rule isn't automatically applied to an outgoing rule.

<u>AMI:</u>

1.What does AMI stand for ?

AMI stands for Amazon machine images

2.Difference between AMI and snapshot?

Snapshots are use to take the copy of certain time of ebs volume,where as AMI is used for ec2.

3.What is AMI?

AMI are a template that has the configurations which can be used to provision ec2.

4.What is AMI id ?

Thats the unique identifier number for each AMI.

<u>IAM:</u>

1.IAM stands for?

Identity access management.

2.How to create a user using IAM?

Got to IAM servvice,select create user and provide the details.

3.WHat are groups in IAM?

Groups can be used to keep multiple users in the same cluster.
Example if we have 5 users under development,then we can create a group with dev and add users in that group.

Apply set of rules/policies to that group and
each user within that group will have the same level of access.

4.What is role in IAM?

IAM role is an identity within your AWS account that has specific permissions.
It is similar to an IAM user, but is not associated with a specific person.

5.What is Policy in IAM?

Policy in IAM define permissions for an action regardless the method use to perform.
A policy is something that will be assigned to role/user/group.

6.Difference between role and policy?

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.

7.What are the different levels of access we can provide to users using IAM?

We can provide two types of access.
Console/GUI access
Programmatic/CLI Access.

8.What do you mean by identity in IAM?

Identity can be user,group,roles.

9.Types of Roles in Iam?

1) Aws service Role:
A service role is an IAM role that a service assumes to perform actions on your behalf.
2) AWS account:
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
3) Web identity:
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
4) SAML 2.0 federation (Security Assertion Markup Language):
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
5) Custom trust policy:

Create a custom trust policy to enable others to perform actions in this account.

10.Types of Policies?

1) Identity Based policies:

InLine Policy:

An inline policy is a policy created for a single IAM identity (a user, group, or role).

Inline policies maintain a strict one-to-one relationship between a policy and an identity.

They are deleted when you delete the identity.

Managed Policies:

1) Aws Managed Policy
2) Customer Managed Policy

AWS managed policies – Managed policies that are created and managed by AWS.

Customer managed policies – Managed policies that you create and manage in your AWS account. Customer managed policies provide more precise control over your policies than AWS managed policies.

2) Resource-based policies:

Are attached directly to resources and specify permissions for specific actions on the resource by some principals.

3) IAM permissions boundaries:

Define the maximum permissions for an IAM entity and are used as safeguards.

4) Access control lists(ACLs):

Are attached to resources and control cross-account permissions for principals from other accounts.

5) Organizations Service Control Policies(SCPs):

Specify the maximum level of permissions for an organization’s accounts. These policies are used to limit the permissions that can be assigned within member accounts.

6) Session policies:
Are advanced policies used during temporary sessions for roles or federated users.

EBS:

1.EBS stands for ?
Elastic Block storage
2.Can we attach multiple ebs to one ec2?
Yes we can attach multiple ebs to one ec2.
Once ebs is attached then we need to follow below steps.
lsblk -- to check the list of filesystem
then we need to create a mount point to that file system
df -h -- to check the filesystem.
3.What is an EBS snapshot?
EBS snapshot is a copy of certain time/backup of ebs volume.
4.Different types of EBS volumes available?
General Purpose SSD volumes.
Provisioned IOPS SSD volumes.
Throughput Optimized HDD and Cold HDD volumes.
Previous generation Magnetic volumes.

VPC:

1.VPC stands for?
Virtual Private Cloud
2.How to create VPC?
Got to the VPC service click on create VPC. Give the name and IPV4 CIDR. Click on create VPC.
Then create subnets in different availability zones. Create a internet gateway and attach it to the VPC.

A default route table is created when the VPC is created and edit the routes and subnet associations.

3.How many VPC can be created in one aws region?

Five - 5

4.What is CIDR ?

Classless inter domain range

5.How many Ips will be associated to 192.168.0.0/24?

256

6.How many ips will be used internally by aws for communication?

five

7.What are subnets?

Subnets are the rang of ipv4 address in the VPC.

8.What is the difference between public and private subnet?

Public subnet is assosciated with the internet gateway in the route table.

In private subnet it is not assosciated with the internet gateway in the route table.

9.What is IGW?

Internet gateway is attached to VPC,which helps for communication between VPC.

10.What is RT?

Route Table is used to route the traffic to targeted subnets.

11.What is NAT gateway/ NAT instance?

NAT gateway is a aws service which is used to connect with instances in private subnet.

NAT gateway service will be managed by aws.

Nat instances is like similar ec2 and also used connect with instances in private subnets.

NAT instance maintenace part will be on our head.

12.Nat will be deployed in which subnet?

Public

13.What is VPC Peering?

VPC peering is used to establish the connection between two VPC in the same region or different region or different aws account.

14.Explain the process of VPC peering?

Create two VPC in different regions with different ip address.

Create two instances in the both regions.

Create a peering connection and accept the conection in the other region.

Modify the route table by adding the private address in the routes.

15.What is VPC Transist Gateway?

VPC Transist Gateway service is used to connect with cloud and on premise networks.

16.What is the difference between Internet Gateway and NAT gateway?

IGW allows both inbound and outbound access to the internet,NAT gateway will only allow outbound access.

17.What is vpc flow log service ?

VPC flog log service is used to store the logs related to VPC.

18.Where can we store vpc logs using flowlogs service?

We can store logs in two ways.

1) Using Amazon s3.
2) Cloud watch log stream.

Load Balancer:

1.Full form of ELB?

Elastic Load balancer.

2.How many types of load balancer are available in aws?

We have 4 types of load balancer

1) Classic Load Balancer
2) Application Load balancer
3) Network Load balancer
4) gateway Load Balancer (Available in specific regions only)

3.How does Classic Load Balancer work?

Classic load balancer works on the HTTP, HTTPS, TCP.

If we have equal numbers of instances in both AZ then classic can be used.

4.How does Application Load Balancer work?

Application Load balancer works on HTTP/HTTPS protocol

It works on application layer that is layer 7.

We can configure different listeners.

It will work based on path prefix and rules configure.

Target group will be attached to the load balancer.

5.How does Network Load Balancer work?

Network Load balancer works on TCP/UDP protocol
It works on Transport layer that is layer 4.
We can configure different listeners.
It will work based on path prefix and rules configure.
Target group will be attached to the load balancer.
we can assign a elastic ip to NLB.

6.When can we use ALB and NLB?

ALB can be used if we are aware of the traffic hitting our application.
NLB will be used if we are not sure of the traffic hitting our applicaiton and if we are expecting sudden spikes in traffic then NLB
can easily handle such situations.

Autoscaling Groups:

1.ASG stands for ?

Autoscaling Groups.

2.Difference between launch template and launch configuration?

Launch COnfigurations cannot be editable,if we want to update then we need to replace the launch configuration.
launch Template can be edited and can have different versions of template.
We can pass parameters in Launch template.

3.Different types of scalings in ASG?

We have Horizontal and vertical scalings.

4.What is Horizontal Scaling?

Horizontal scaling means adding additional ec2 to our environment.

5.What is Vertical Scaling?

vertical scaling means adding more power to our existing ec2.

S3 Buckets:

1.Why do we use s3 Buckets?

S3 buckets are used to store fixed objects.

2.What are objects in s3?

Objects can be referred to any file uploaded in s3.

3.Default s3 bucket class?

Standard.

4.Different classes in s3 bucket?

Amazon S3 Standard
Amazon S3 Intelligent-Tiering
Amazon S3 Standard-Infrequent Access
Amazon S3 One Zone-Infrequent Access
Amazon S3 Glacier Instant Retrieval
Amazon S3 Glacier Flexible Retrieval
Amazon S3 Glacier Deep Archive

5.How to deploy static website in s3?

We need to enable the static website hosting option in s3 bucket.
Create one index.html and one error.html file and upload the same to s3 bucket.
Attach a policy to make the objects public in s3 bucket.
We can access the website using s3bucket url on browser.

6.What is versioning?

Versioning helps us to store file with multiple version.

7.what is bucket policy?

S3 bucket policies specify what actions are allowed or denied for which principles on the bucket that the bucket policy is attached to.

8.S3 object Encryptions?

Object encryption helps us to encrypt the data before saved to disk and decrypt when downloaded.
Encryption is a method which helps to secure the data.

9.What is object lock?

Object Lock can help prevent objects from being deleted or overwritten for a period of retention time.

10.Expalin S3 Cross region replciation?

Cross region repliacation helps to automatically upload the data into destination bucket without manual intervention.
This can be used as a backup for s3 bucket.

Route53:

1.What is Route53?

Route53 is a DOMAIN NAME SERVICE in aws.

2.Is Route53 a global or regional servie?

Global Service.

3.What is Alias in Route53?

Alias records helps us to route the traffic to selected aws resources.

4.What are nameservers in Route53?

Amazon Route 53 automatically creates a name server (NS) record that has the same name as your hosted zone.

It lists the four name servers that are the authoritative name servers for your hosted zone.

5.What are different records availabe in Route53?

A (address record)
AAAA (IPv6 address record)
CNAME (canonical name record)
CAA (certification authority authorization)
MX (mail exchange record)
NAPTR (name authority pointer record)
NS (name server record)
PTR (pointer record)

6.Different hostings policies available in Route53 and explain them?

Simple Routing Policy.
=====================

Simple routing policy is a simple round-robin policy and can be applied when there is a single
resource doing the function for the domain e.g. web server that serves content for the website.

Weighted Routing Policy.

Weighted routing policy helps route traffic to different resources in specified proportions
(weights) e.g., 75% to one server and 25% to the other during a pilot release

Latency-based Routing (LBR) Policy.
=================================

Latency-based Routing Policy helps respond to the DNS query based on which data center gives the user the lowest network latency.

Failover Routing Policy.

=========================

Failover routing policy allows active-passive failover configuration, in which one resource (primary)
takes all traffic when it's healthy and the other resource (secondary) takes all traffic when the first isn't healthy.

Geolocation Routing Policy.

==========================

Geolocation routing policy helps respond to DNS queries based on the geographic location of the users
i.e. location from which the DNS queries originate.

Geoproximity Routing Policy.

===========================

Geoproximity routing helps route traffic to the resources based on the geographic location of the users and the resources.

Multivalue Routing Policy.

=========================

Multivalue routing helps return multiple values, e.g. IP addresses for the web servers, in response to DNS queries.

Route 53 Traffic Flow.

======================

Route 53 Traffic Flow helps easily manage traffic globally through a variety of routing types combined with DNS Failover
in order to enable a variety of low-latency, fault-tolerant architectures.

Cloud Trial:

1.What is Cloudtrial?

Cloudtrail is a service that enables governance,compliance,Operational auditing and risk auditing of our aws account.
Cloudtrial is a service which helps us to track the activities
in aws account.
It records all the activities done by user,roles and aws services.

These recorded sessions are called as EVENTS.
CloudTrail is enabled on your AWS account when you create it.
View events in Event History, where you can view, search,
and download the past 90 days of activity in your AWS account.

2. What are the different type of Log events?

1) Management Events
Capture management operations performed on aws resources.
Free of cost for one management event by aws.
Management Events will be created by default.
Ex: who signed in at what time and other api calls
execute on resources.

2)Data Events:
Log the resource operations performed on or within a resource.
Ex: when user uploaded,deleted and downloaded files in s3.

3)Insight Events:
Identify unusual activity, errors, or user behavior in your account.
Ex:Unauthorized API calls indicate someone tried to perform an action
in your AWS account that they did not have permission to carry out.

<u>Cloudwatch:</u>

1.What is cloudwatch?

Cloudwatch is a monitoring tool used to monitor our aws resources,application.

2.What are default metrics and custom metrics?

Deafult metrics are cpu utilization,diskreads,diskwrites etc.
Custom metrics can be any metric which can be pushed with bash scripting,CLI or API.

3.How to configure a cloudwacth alert to a specific custom servcie?

We need to create a alarm and configure SNS to that alarm.

4.What is SNS?

Simple notification service which will helps the user to notify based on email,SMS.

4.What are the different metrics we can configure on cloudwatch?

CPU Utilization

Service running or not Metric

Disk write and not metric etc..

5.How to configure dashboard in cloudwatch?

Goto CLoudwatch and click on dashboard,select the metric to which you want to create dashboard.

Select the type of widget you want to configure and save.

6.What is cloud watch agent used for?

Cloud watch agents are responsible to push the metrics to cloudwatch dashboard.

Agents needs to be up and running on the ec2.

7.What is Metric?

Metrics are a measurement at a point in time for the system.

8.What is Alarm?

Cloudwatch alarm is used to monitor a single cloud watch
metric or the result of Match expression using cloud watch metrics.
Also, it sends out a notification based on the threshold we set for
each service in the cloud watch alarm.

9.What is cloudwatch log group?

A log stream is a sequence of log events that share the same source.

Each separate source of logs in CloudWatch Logs makes up a separate log stream.

10.What is cloudwatch Events?

CloudWatch Events help you to respond to state changes to your AWS resources.

Cloudformation:

1.Why do we use cloudformation?

Cloudformation Templates are used to provision infrastructure on aws environment.

2.What is IAC?

Infrastructre As Code.

3.Can you explain what a template is in the context of AWS CloudFormation?

A CF template can be written in JSON or YAML language.

A template describes all your resources and their properties.

4.What are stacks?

A stack is a collection of AWS resources that you can manage as a single unit.

5.Sample Cloudformation Template to provision ec2?

```
AWSTemplateFormatVersion: 2010-09-09
Description: Part 1 - Build a webapp stack with CloudFormation

Resources:
  WebAppInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0d5eff06f840b45e9 # ImageID valid only in us-east-1 region
      InstanceType: t2.micro
```

6.How to add parameters in CF template while creating Stack?

While Parameters are technically optional, they are essential to building flexible CloudFormation templates

Example:

```
Parameters:
  BucketNameParam:
    Description: Name of the bucket
    Type: String
    Default: PrimaryUploadBucket

Resources:
  MyS3Bucket:
    Type: "AWS::S3::Bucket"
    Properties:
      AccessControl: PublicRead
      BucketName: !Ref BucketNameParam
```

7.What is nested stack?

Nested stacks are stacks created as part of other stacks. You create a nested stack within another stack by using the AWS::CloudFormation::Stack resource.

8.What is reference?

When you are declaring a resource in a template and you need to specify another template resource by name,
you can use the Ref to refer to that other resource.

9.What is output in template?

The optional Outputs section declares output values that you can import into other stacks (to create cross-stack references),
return in response (to describe stack calls), or view on the AWS CloudFormation console.
For example, you can output the S3 bucket name for a stack to make the bucket easier to find.

10.What is mapping?

CloudFormation provides two elements known as Mappings and Conditionals.
Mappings allow you to create simple "Key:Value"
dictionaries or hashes for use in your resource declarations.
And Conditionals allow you to use some logic-based decisions in your resources to add or modify values.

AWS RDS:

1.What is RDS service?

RDS is a relational database service,which helps us to setup and maintain database in cloud.

2.Types of Databases and examples?

Relational database stored information in tables.
Often, these tables have shared information between them, causing a relationship to form between tables.
This is where a relational database gets its name from.
Example: MySql, Oracle, MariaDB, Postgres and MS-SQL

Non-Relational Database,sometimes called NoSQL (Not Only SQL), is any kind of database that doesn’t use the
tables, fields, and columns structured data concept from relational databases.
Example: MongoDB, BigTable, Redis, RavenDb, Cassandra, and CouchDb.

3.How can we migrate db from ec2 to RDS?

Migrating DB from EC2 to RDS:
=============================

1) Get the dump of your existing DB on EC2

```
mysqldump -u root -p database_name > file_name.sql
```

2) Migrate the DB dump that you have taken in step 1 to RDS

```
mysql -h <replace-rds-end-point-here> -P 3306 -u <user_name> -p database_name < ec2db.sql
```

3) Connect to your RDS DB instance

```
mysql -h <replace-rds-end-point-here> -P 3306 -u rdsuser -p
```

4) Switch to the database and verify the details.

```
USE rdsdb
SELECT * FROM table1;
```

4.What is multi AZ RDS?

Multi Az option can be enabled on rds instance and it will create a replica/standby db of original db in anaother availability zone.
The ebs volume attached for standby db will be in the created in the same region.
RDS access only via database CNAME. so we cannot access standy db for any reason via RDS.
Standby replica cannot be used as extra capacity.
synchronus replication means if some is written in the db then the same will be replciated in standby db.

This is helpful if for some reasons primary is not working as expected then aws will detect
and it will take 1-2 mins to create a new instance.(Fail Over)

Fail over means it will have some downtime.
Fault tolerance means there will be no downtime.

Points to remember:
==============
1) Multi AZ feature is not free and we need to pay twice the price.
2) Standy replcia cannot be accessed directly unless a failure occurs.
3) Failover is highly available, not faulty tolerant.
4) Backups taken from standby (Removes performance impacts)
5) Multi AZ will be created in same region.

5.What is RPO?

1) Time between the last backup and whenever the failure occured.
Ex: We have taken backup at 6 a.m and failure happened at 7 a.m
then RPO us for 1 hour and we will use 1 hr of data.
2) Amount of maximum data loss.
3) Business provides an RPO value.
4) influence technical solution and close.

6.What is RTO?

1) Time between the DR event and full recovery.
Ex: Disaster happened at 6 a.m and you took 30 mins to restore back the DB.
Then 30 mins we can say as RTO.
2) Influenced by process,staff,tech and documentation.

7.How to take backup of RDS?

We can create snapshot of the RDS instance.
if we have Multi Az then backup will be taken from standby db.

8.What is retention period in db?

Retention period means after taking backup it will be kept for 7 days if retention period is selected as 7.

9.What is Read replica and when in which scenario it will be used?

Read Replcias is a copy of the primary instance.
Use case: Let’s assume we have a primary RDS instance that serves both read and write traffic.
Due to the size of the instance and the amount of read-intensive traffic being directed to the database for queries. the performance of the instance is taking a hit.
To help resolve this, you can create a read replica. A snapshot will be taken of your database, and if you are using Multi-AZ, then this snapshot will be taken of your secondary database instance to ensure that there are no performance impacts during

the process. Once the snapshot is completed, a read replica instance is created from this data.

10.Difference between multi az and read replicas?

Multi Az is configured for fail over and read replicas
are used to improve the performance of the db.

11.What is the difference between Database Cluster and Database Instances?

Database Cluster

Its the name of the cluster that holds instances.

Database Instances

Its the name of each instance in the cluster.

12.Difference between RDS service and DB in ec2/vm ?

RDS service is provided by aws where we can spin up rds instance and start using th db.

DB in ec2/vm is standalone DB where we need to install and do the maintenance as well.

13.How can we connect to MS SQL RDS?

We can install sql workbench and enter the dbhost url and port number and click on test connection.

14.List Backup types supported by Amazon RDS?

There are two types of backups supported by Amazon RDS such as automated backups and database snapshots.

The automated backup enables point-in-time recovery of your DB instance automatically.

A DB snapshot is a manual process to backup the DB instance. It can be done as frequently as you wish.

15.What is RDS backup retention period?

That will be period to take backup of RDS.

If retention period is of 7 days then it will take vackup after 7 days.

Docker:

1.What is Docker?

Docker is for containerization where we can deploy different applications on different containers.

Docker containers are independent of Operating system and the boot up time for docker is very fast comparing to VM.

Docker containers can be easily ship from one env to anaohter env and the images can be stored in central Repo (DockerHub).

2.What is the difference between containerization and virtualization?

Virtualization enables you to run multiple operating systems on the hardware of a single physical server,

while containerization enables you to deploy multiple applications using the same operating system on a single virtual machine or server.

3.Explain the Architecture of Docker?

Docker follows Client-Server architecture, which includes the three main components that are Docker Client, Docker Host, and Docker Registry.

1. Docker Client

Docker client uses commands and REST APIs to communicate with the Docker Daemon (Server). When a client runs any docker command on the docker client terminal, the client terminal sends these docker commands to the Docker daemon. Docker daemon receives these commands from the docker client in the form of command and REST API's request.

Note: Docker Client has an ability to communicate with more than one docker daemon.

Docker Client uses Command Line Interface (CLI) to run the following commands -

docker build
docker pull
docker run

2. Docker Host

Docker Host is used to provide an environment to execute and run applications. It contains the docker daemon, images, containers, networks, and storage.

3. Docker Registry

Docker Registry manages and stores the Docker images.

There are two types of registries in the Docker -

Pubic Registry - Public Registry is also called as Docker hub.

Private Registry - It is used to share images within the enterprise.

4.What is Docker file?

DockerFile is a template which containes set of instructions which helps to build docker image.

5.What is Docker image?

A Docker image is a Template used to execute code in a Docker container.

6.What is docker container?

A docker container is a running state of image,which execute the code of docker image and help to run application.

7.Create sample dockerfile use case?

```
FROM alpine:3.12
MAINTAINER <Your_Name>
RUN mkdir /usr/local/tomcat/
WORKDIR /usr/local/tomcat
RUN apk --no-cache add curl && \
    apk add --update curl && \
          curl -O https://mirrors.sonic.net/apache/tomcat/tomcat-8/v8.5.61/bin/apache-tomcat-8.5.61.tar.gz
RUN tar xvfz apache*.tar.gz
RUN mv apache-tomcat-8.5.61/* /usr/local/tomcat/.
RUN rm -rf apache-*
COPY SampleWebApp.war /usr/local/tomcat/webapps
RUN apk update && apk add openjdk8
WORKDIR /usr/local/tomcat
EXPOSE 8080
CMD ["/usr/local/tomcat/bin/catalina.sh", "run"]
```

8.How to build the dockerfile?

docker build -t . (-t means tag '.' represent current location of docker file) OR
docker build -t /pathofDockerFile

9.Difference between CMD and Entry Point?

CMD and Entry Point both are executables which will help to execute the commands at the runtime of container.

We can have multiple CMD but only the last CMD will get execute.

CMD are Overridable and Entry point is not overridable.

CMD and ENTRY both can we used is same Docker File but that’s not the best practice.

EXAMPLE:

```
FROM ubuntu
RUN apt-get update
```

```
ENTRYPOINT ["echo","hello"]
CMD ["WORLD"]
```

When running the container it will show "Hello,WORLD"
Suppose we are passign argument like docker run -it -d <image_name> Sabair
Output: "hello Sabair"

10.Difference between COPY and ADD ?

COPY is used to copy file from dockerhost to docker image.
ADD can be used to download file from internet and also to extract the tar file.

11.Explain different modules of Dockerfile?

FROM: where to download the base image.-
MAINTAINER: Non executable instruction used to indicate the author of DockerFile.
ADD: It copy the file from source to destination and also extracts the file.
CMD: It specifies the intended command for the image.
ENTRYPOINT: when the container is up,entry point is the starting of execution.
ENV: This instruction can be used to set the env variables in the container.
EXPOSE: expose a specified port.
RUN: This instruction is used to execute a command on top of an existing layer.
USER: This is used to set the username.
VOLUME: Volume instruction is used to enable access to a location.
WORKDIR: To change the directory.
ONBUILD: It will add a trigger instruction.

12.MultiStage Dockefile?

Multi stage Docker file is used to reduce the size of image,better security and to spin out containers much faster.

13.Example for single stage and multi stage dockerfile?

Example for single stage dockerfile.
Clone the below repo and add the DockerFile.
https://github.com/betawins/multi-stage-example.git
Single stage Dockerfile:

=================

```
FROM openjdk:8-jdk-alpine
RUN mkdir -p /app/source
COPY . /app/source
WORKDIR /app/source
RUN ./mvnw clean package
EXPOSE 8080
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom", "-jar", "/app/source/target/multi-stage-example-0.0.1-SNAPSHOT.jar"]
```

Multi stage-Dockerfile:

================

```
#Build Image
FROM openjdk:8-jdk-alpine as builder
RUN mkdir -p /app/source
COPY . /app/source
WORKDIR /app/source
RUN ./mvnw clean package

#Run image
FROM openjdk:8-jdk-alpine
WORKDIR /app
COPY --from=builder /app/source/target/*.jar /app/app.jar
EXPOSE 8080
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom", "-jar", "/app/app.jar"]
```

14.Docker scan ?

Docker scan is used to scan the docker images for vulnerabilities.
By default we have Docker scout available in dockerhub registry, will helps us to scan every image which is pushed to dockerhub.

15.How to list the images in docker?

docker image ps

16.How to delete the images in docker?

docker image rm <image_name>

17.How to get the details of docker image?

docker inspect <Docker_image>

18.What is dangling images?

Dangling images are layers that have no relationship to any tagged images.
They no longer serve a purpose and consume disk space.

19.What is tag in docker images?

Tag in docker images specify the version of the image.

20.What is docker registry?

Docker registry is centralized place to store our images similar like github.

21.How to push docker image to docker registry?

We need to login to docker registry first by using
docker login
We need to tag image name with docker repo name
once login is succeeded,use
docker push <image_name>

22.How to run a container?

docker run -itd -p 8080:8080 -name test <image_name>
(-IT= Interactive Terminal -d= detach mode/foreground -p= port exposing -name= name of image).

23.command to check the running containers?

docker container ps

24.Command to Get the details of containers?

docker container inspect <COntainer_id>

25.Can we delete a pause container?

No, we need to stop the pause container and then delete the container by using
docker container stop <Container_ID>

26.How to check the list of containers running and exited?

docker container ps -a

27.What is -it and -d in docker run commands?

-IT = interactive Terminal

-D = Detach mode /foreground/background

28.What is port mapping in docker containers?

Port mapping is used to expose a port to public.

29.How to connect to the running container?

docker container exec -it <Container_id> /bin/bash

30.How to kill a running container?

docker kill my_container

31.How to check the logs of container?

docker container logs <container_id> --follow

32.How to commit a running container?

docker commit <container_id>

33.What is docker volumes and how many types of volumes are availabe?

Docker volumes are used to store docker container data.

This is used to store persistent data.

Even if our container is killed or deleted then the data will be stored in volume.

We have 3 types of volumes

1) Volume
2) tmpfs mount
3) BindMount

34.Command to run docker container with a volume attached?

docker run -d --name <container_name> -v <Volume_name>:/app <Image_name>

35.What is docker networks?

Docker networking is primarily used to establish communication between Docker containers and the outside world via the host machine where the Docker daemon is running.

36.Different types of network and there usage?

The Bridge Driver:

This is the default. Whenever you start Docker, a bridge network gets created and all newly started
containers will connect automatically to the default bridge network.

The Host Driver:

You can use the host network if you don't want to rely on Docker's networking but instead rely on the host machine networking.

The Overlay Driver:

The Overlay driver is for multi-host network communication, as with Docker Swarm or Kubernetes.

Overlay network allows us to communicate with containers running on different node machines.

37.What is docker compose?

Docker compose is a different tool used to provision multiple container at the same time.

If we have dependencies on two containers then we can easily manage them with the help of docker Compose.

38.Sample-two-tier:docker-compose file

```
version:  '3'   -> Depends upon docker-compose version
services:       -> All the services should be in this section
  db:          -> Name of service (name can be anything)
    image: mysql:5.7  -> Images used to run container
    volumes:          -> volumes to store contaienr data
      - db_data:/var/lib/mysql    -> mount point
    restart: always
    environment:    -> Environmental variables
      - MYSQL_ROOT_PASSWORD=somewordpress
      - MYSQL_DATABASE=wordpress
      - MYSQL_USER=wordpress
      - MYSQL_PASSWORD=wordpress
  wordpress:
    depends_on:   -> It will wait for db service to get started.
      -   db
    image: wordpress:latest
    ports:        -> port to expose to outerworld
      - "8000:80"
    restart: always
    environment:
      - WORDPRESS_DB_HOST=db:3306
      - WORDPRESS_DB_USER=wordpress
      - WORDPRESS_DB_PASSWORD=wordpress
      - WORDPRESS_DB_NAME=wordpress

volumes:
   db_data: { }
```

39.Docker-compose commands?

1) docker-compose up -> To run the docker-compose file

2) docker-compose up -d -> To run the docker-compose file in detach mode

3) docker-compose down -> To stop & delete the docker-compose services

4) docker-compose restart -> To restart docker-compose

5) docker-compose stop -> To stop docker-compose.

6) docker-compose start -> To start docker-compose

7) docker-compose ps -> list of containers

8) docker-compose pause -> pause the docker-compose

9) docker-compose unpause -> unpause the docker-compose

10)docker-compose top -> to view top performances

11)docker-compose up -d --scale db=5 ->To run 5 db container (db is nothing but service name)

Note: we can't scale the services using port number because of port conflicts.

12)docker-compose -f docker-compose2.yml up -d -> To use other docker-compose file

docker-compose create -> It will create container with only default network

13)docker-compose iamges -> List of imags used in docker compose

14)docker-compose kill -> kill the containers

15)docker-compose logs -> To check the logs of containers

16)docker-compose port webapp 80 -> To check if port 80 is bind with webapp service or not

17)docker-compose exec webapp ls -> Execute the command in webapp container

40.Docker Commands?

docker run hello-world -- run container of hello-world image

docker ps -- to check the list of running containers

docker ps -a --list of running and exited containers

docker images -- list of images

docker search tomcat -- search for tomcat image locally

```
docker pull tomcat   -- pull tomcat image from docker registry
docker run -it -p 1234:8080 tomcat  -- RUN container using tomcat image
docker build -t sabair .  -- Build docker image with name sabair
docker tag sabair_ubuntu sabair_ubuntu-PROD -- tag the image sabair_ubuntu with sabair_ubuntu-PROD
docker rm  <Container_ID> -- delete container
docker rmi <Image_ID>  -- delete image
docker exec -it 5267e21d140 /bin/bash -- Connect with image 5267 (container_id)
docker commit 5267e21d140 sabair_v2:latest  -- save the changes made to container sabair_v2
```

41.Export/Import Docker Image to file?

```
docker save image:tag > arch_name.tar
docker load -i arch_name.tar
```

42.Import/Export Docker Image to AWS ECR?

```
docker build -t sabair:v1 .
aws ecr get-login --no-include-email --region=ca-central-1
docker tag  sabair:v1  12345678.dkr.ecr.ca-central-1.amazonaws.com/myrepo:latest
docker push 12345678.dkr.ecr.ca-central-1.amazonaws.com/myrepo:lastest
docker pull 12345678.dkr.ecr.ca-central-1.amazonaws.com/myrepo:latest
```

43.Kill and Delete Containers and Images?

```
docker rm -f $(docker ps -aq)        # Delete all Containers
docker rmi -f $(docker images -q)    # Delete all Images
```

<u>Ansible:</u>

1.What is ansible used for?

Ansible is used for configuration management,if suppose we have 10 servers and need to
make changes in the 10 servers then ansible is something which can help us to do the changes in a single click.

Ansible is agentless arhitecture which works on port number 22 (SSH).

We need to configure inventory/hosts whihc will have the ip or dns of servers.

With ansible adhoc commands or playbooks we can execute the job and make our work done.

Ansible is an ecample of push based configuration management tool.

2.Explain the architecture of ansible?

Playbooks
Modules
Inventory
Plusgins

3.Is ansible push based or pull based?

Push based.

4.What is inventory in ansible?

Inventory can be also called as hosts file,which will contains the ip address of other servers.

5.How many types of inventory available in ansbile and explain them?

Static Inventory - If we have 10 servers then 10 ips will be configured in inventory.

Dynamic Inventory - If we are provisioning new ec2 servers and in this case dynamic inventory

will help us to capture the newly provisioned server ips.

For this we need to have python script which will execute and capture the ip of newly provisioned servers.

6.What is playbook in ansible?

Playbook will consists of different task/plays which will get executed on servers.

7.ansible is written in which language?

Python

8.Ansible playbooks can be written in which language?

YAML Language, YML Language

9.Tell 15 modules in ansible?

Yum
Service
Shell
lineinfile
dir
copy
handlers
notifiers
apt
service

debug
register
file
git
archive

10.What is adhoc command in ansbile?

Adhoc commands are single line commands to execute the task and server/node machines.

Example:

ansible all -m ping -- To check the connetivity between slaves

11.What are ansible roles?

Ansible roles are used to make small parts of a playbook.
meaning instead of having all the tasks within single playbook and make it clumpsy,
we create roles with small set of tasks.
Roles are also reusable,roles an be called with any playbook and start using them.

12.Direcotry structure of ansible roles?

```
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

13.What is ansible vault?

Ansible vault is used to store the files with encryption.
if we have password then we use the vault to encrypt the file.

14.What is handlers in ansible?

handlers are used if we have dependency on different plays/task.
Example task2 should get execute only if task1 is successfully executed then
in task 1 we configure "Notifier" which helps to send the alert to handler in task2 that the task is completed.
Note: Notifier name and Handler name should be same.

15.What is an Ansible galaxy?

Ansible Galaxy is essentially a large public repository of Ansible roles. Roles ship with READMEs detailing the
role's use and available variables. Galaxy contains a large number of roles that are constantly evolving and increasing.

16.what is ansible tower?

Ansible Tower is more for enterprise edition,where we can use GUI and easily manage and execute our playbooks
and inventories.

17.Write 5 Sample Ansible playbooks?

https://github.com/betawins/Ansible-playbook-NcodeIT

<u>Jenkins:</u>

1.What is jenkins?

Jenkins is a Opensource CICD tool,this is used to automate the jobs and easily deploy on environments.

2.What is CICD?

Continous Integration,Continous Deployment/Continous Delivery.

3.What is continous integration,continous Delivery and continous Deployment?

Continous Integration:
When a commit occurs on github repository then with the help of webhooks and based upon the event occure jenkins job will get triggered, which will execute different stages and build a war package, This package will be push to artifactory location.

This part comes under Continous Integration.
Continous Deployment Means when ever a package is created then it will automatically deploy on the end environment.
Continous Delivery makes sure that the environment is ready for deployment but it will actually not deploy the package.

4.Jenkins is written in which language?

Jenkins is written in JAVA.

5.Jenkins run on which port number?

8080

6.What is Jenkins file?

Jenkins File is a text file which consists of different stages.
Each stage will have a set of task to be executed.

7.What is Jenkins pipeline?

Jenkins Pipeline will be executed based on Jenkins File,where different stages will get executed.
We have two types of pipelines,
1) Declarative
2) Scripted

8.Types of jenkins job/projects?

Freestyle project.
Maven project.
Pipeline.
Multibranch pipeline.
External Job.
Multi-configuration project.
Github organization.

9.Name some jenkins plugins?

Git
Java
Maven
role based access plugin
Docker
tomcat
Sonarqube
Nexus Artifactory & Slack

10.What is workspace in jenkins?

Workspace is a place where all the jenkins jobs data will be stored.

11.DIfferent stages of Jenkins pipeline?

Git clone -- Helps to clone the source code.

Sonarqube -- To check the quality of code.

Maven -- TO build the source code and create package out of it.

Nexus -- TO push the artifacts to nexus repository.

Slack -- To send the build out put to slack channel.

Email -- To send the build output to email.

12.How many types of pipeline we have in jenkins?

We have two types of pipelines,

1) Declarative
2) Scripted

13.Difference between declarative and scripted pipeline?

Declarative Pipeline is configured within the Jenkinsfile and stored in Central Repository (GitHub)

Declarative Pipeline will start with pipelines and if there is any issue in pipeline then the moment pipeline is triggered it will throw the error.

Example syntax:

```
pipeline {
   agent any

   stages {
      stage('Build') {
         steps {
            echo 'Building..'
         }
      }
      stage('Test') {
         steps {
            echo 'Testing..'
         }
      }
      stage('Deploy') {
            steps {
                echo 'Deploying....'
            }
        }
    }
}
```

Scripted pipelines are configured inside the job.
Scripted pipelines will start with nodes and if there is any error in 100 the line then it will execute all the 99 lines and throw the error at the 100 line.
Example:

```
node {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Building..'
            }
        }
        stage('Test') {
            steps {
                echo 'Testing..'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying....'
            }
        }
    }
}
```

14.How to segregate user roles in Jenkins?

We can use role based strategy plugin and add the user read,view and other access.

15.How to take backup of jenkins server?

Jenkins backup can be taken in two ways either by using
Thinbackup Plugin or
by using bash script to takle the backup of workspace and dump that in some other location.

16.What is master and slave configuration jenkins?

Master and salve configuration is implemented for highly availability of jenkins server.

We can offload the work from master to slave and master can be used only to pass order on slave nodes.

Slave node are also responsible to execute some specific job.

For example if we have java and python applications then we can confgiure two slaves for JAVA and Python applications.

java slave is responsible to execute java related jobs and python slave for python related jobs.

17.What is label in slave?

Label are names to define our slave machines.

labels play very important role when dedicating a job to the particualr slave.

18.What is executors in jenkins?

If we have 3 executors then means we can run 3 jobs at the same time.

19.Write sample Jenkins Pipeline?

```
pipeline {
    agent {
        label "master"
    }
    tools {
        // Note: this should match with the tool name configured in your jenkins instance (JENKINS_URL/configureTools/)
        maven "MVN_HOME"

    }
          environment {
        // This can be nexus3 or nexus2
        NEXUS_VERSION = "nexus3"
        // This can be http or https
        NEXUS_PROTOCOL = "http"
        // Where your Nexus is running
        NEXUS_URL = "18.216.151.197:8081/"
        // Repository where we will upload the artifact
        NEXUS_REPOSITORY = "soanrqube"
        // Jenkins credential id to authenticate to Nexus OSS
        NEXUS_CREDENTIAL_ID = "nexus_keygen"
    }
    stages {
        stage("clone code") {
            steps {
                script {
                    // Let's clone the source
                    git 'https://github.com/betawins/sabear_simplecutomerapp.git';
                }
            }
        }
        stage("mvn build") {
            steps {
                script {
                    // If you are using Windows then you should use "bat" step
                    // Since unit testing is out of the scope we skip them
                    sh 'mvn -Dmaven.test.failure.ignore=true install'
                }
            }
        }
        stage("publish to nexus") {
            steps {
                script {
                    // Read POM xml file using 'readMavenPom' step , this step
'readMavenPom' is included in: https://plugins.jenkins.io/pipeline-utility-steps
                    pom = readMavenPom file: "pom.xml";
                    // Find built artifact under target folder
                    filesByGlob = findFiles(glob: "target/*.${pom.packaging}");
                    // Print some info from the artifact found
                    echo "${filesByGlob[0].name} ${filesByGlob[0].path}
${filesByGlob[0].directory} ${filesByGlob[0].length}
${filesByGlob[0].lastModified}"
                    // Extract the path from the File found
                    artifactPath = filesByGlob[0].path;
                    // Assign to a boolean response verifying If the artifact name exists
                    artifactExists = fileExists artifactPath;
                    if(artifactExists) {
                      echo "*** File: ${artifactPath}, group: ${pom.groupId}, packaging:
${pom.packaging}, version ${pom.version}";
                      nexusArtifactUploader(
                        nexusVersion: NEXUS_VERSION,
                        protocol: NEXUS_PROTOCOL,
                        nexusUrl: NEXUS_URL,
                            groupId: pom.groupId,
                        version: pom.version,
                        repository: NEXUS_REPOSITORY,
                        credentialsId: NEXUS_CREDENTIAL_ID,
                        artifacts: [
                          // Artifact generated such as .jar, .ear and .war files.
                          [artifactId: pom.artifactId,
                          classifier: ",
                          file: artifactPath,
                          type: pom.packaging],
                          // Lets upload the pom.xml file for additional information for
Transitive dependencies
                          [artifactId: pom.artifactId,
                          classifier: ",
                          file: "pom.xml",
                          type: "pom"]
                        ]
                      );
                    } else {
```

```
                        error "*** File: ${artifactPath}, could not be found";
                    }
                }
            }
        }
    }
}
```

20.What is jenkins parameterized job?

Parameterized jobs will helps us to pass the parameters in jenkins job.

For example if we want to deploy the same job in different environments then instead of creating different jobs

we can configure a parameter with environment and the same will be reflected in job and seggregatethe environments.

21.What are the global variables in jenkins?

Global environment variables are the variables that can be used in any and every Pipeline or Job built on Jenkins.

The global variables are set via the Jenkins console and via the groovy script of a pipeline.

Example:

BUILD_NUMBER - The current build number. For example "153"

BUILD_ID - The current build id. For example "2018-08-22_23-59-59"

BUILD_DISPLAY_NAME - The name of the current build. For example "#153".

JOB_NAME - Name of the project of this build. For example "foo"

BUILD_TAG - String of "jenkins-${JOB_NAME}-${BUILD_NUMBER}".

EXECUTOR_NUMBER - The unique number that identifies the current executor.

NODE_NAME - Name of the "slave" or "master". For example "linux".

NODE_LABELS - Whitespace-separated list of labels that the node is assigned.

WORKSPACE - Absolute path of the build as a workspace.

JENKINS_HOME - Absolute path on the master node for Jenkins to store data.

JENKINS_URL - URL of Jenkins. For example http://server:port/jenkins/

BUILD_URL - Full URL of this build. For example http://server:port/jenkins/job/foo/15/

JOB_URL - Full URL of this job. For example http://server:port/jenkins/job/foo/

22.What is Maven?

Maven is build tool used to build java applications.

With maven we can compile the source code and build the packages.

23.Maven lifecycles?

prepare-resources -- Resource copying can be customized in this phase.

validate -- Validates if the project is correct and if all necessary information is available.

compile -- code compilation is done in this phase.

Test -- Tests the compiled source code suitable for testing framework.

package -- This phase creates the JAR/WAR package as mentioned in the packaging in POM.xml.

install -- This phase installs the package in local/remote maven repository.

Deploy -- Copies the final package to the remote repository.

24.What is Pom.xml?

PROJECT OBJECT MODEL.

POM containes artifact id,versions,packaging,plugins and dependencies.

25.Maven Repositories?

Local

Central and

Remote Repositories.

26.What is sonarqube?

Sonarqube is a tool to check the quality of code.

27.What are sonar scanners?

Sonar Scanner will help to run project analysus and send the results to sonarqube server.

28.What is quality gate and quality profile in sonarqube?

Quality Profiles are a core component of SonarQube where you define sets of Rules that, when violated, raise issues on your codebase.

Quality Gates are the set of conditions a project must meet before it should be pushed to further environments.

Quality Gates considers all of the quality metrics for a project and assigns a passed or failed designation for that project.

29.Sonarqube runs on which port number?

9000

30.What is nexus?

Nexus is a central repository to store the artifacts (packages)

31.Nexus runs on which port number?

8081

33.What is snapshots and releases in nexus?

Snapshots are not the final version of artifact and can be replaced.
Release are final artifact version and this cannot be replaced.

<u>Terraform</u>:

1.Basic terraform commands?

```
Common commands:
    apply            Builds or changes infrastructure
    console          Interactive console for Terraform interpolations
    destroy          Destroy Terraform-managed infrastructure
    fmt              Rewrites config files to canonical format
    get              Download and install modules for the configuration
    graph            Create a visual graph of Terraform resources
    import           Import existing infrastructure into Terraform
    init             Initialize a new or existing Terraform configuration
    output           Read an output from a state file
    plan             Generate and show an execution plan
    providers        Prints a tree of the providers used in the configuration
    push             Upload this Terraform module to Terraform Enterprise to run
    refresh          Update local state file against real resources
    show             Inspect Terraform state or plan
    taint            Manually mark a resource for recreation
    untaint          Manually unmark a resource as tainted
    validate         Validates the Terraform files
    version          Prints the Terraform version
    workspace        Workspace management
All other commands:
    debug            Debug output management (experimental)
    force-unlock     Manually unlock the terraform state
    state            Advanced state management
```

2.What is a Provisioner Terraform?

Provisioners are used to execute scripts on a local or remote machine as part of resource creation or destruction.

3.What are terraform providers?

A provider is a Terraform plugin that allows users to manage an external API. Provider plugins like the AWS provider or the cloud-init
provider act as a translation layer that allows Terraform to communicate with many different cloud providers, databases, and services.

4.What happens if a resource is removed from terraform state file?

Items removed from the Terraform state are only no longer managed by Terraform. For example, if you remove an AWS instance from the state, the AWS instance will continue running, but terraform plan will no longer see that instance.

5.What is State File Locking?

State file locking is a Terraform mechanism that prevents operations on a specific state file from being performed by multiple
users at the same time. Once the lock from one user is released, any other user who has taken a lock on that state file can operate on it.

This aids in the prevention of state file corruption. The acquiring of a lock on a state file in the backend is a backend operation.

If acquiring a lock on the state file takes longer than expected, you will receive a status message as an output.

6.What is a Remote Backend in Terraform?

Terraform remote backend is used to store Terraform's state and can also run operations in Terraform Cloud.

Multiple terraform commands such as init, plan, apply, destroy (terraform version >= v0.11.12), get, output,
providers, state (sub-commands: list, mv, pull, push, rm, show), taint, untaint, validate, and many more are
available via remote backend. It is compatible with a single remote Terraform cloud workspace or multiple workspaces.

You can use terraform cloud's run environment to run remote operations such as terraform plan or terraform apply.

7.What is terraform taint and tainted resource?

The 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.

8.What are components of terraform?

Terraform has two important components: Terraform Core and Terraform Plugins.

9.Modules in terraform?

A Terraform module allows you to create logical abstraction on the top of some resource set.

In other words, a module allows you to group resources together and reuse this group later, possibly many times.

10.What is Import in Terraform?

Terraform is able to import existing infrastructure. This allows us take resources we've created by some other

means (i.e. via console) and bring it under Terraform management.

The terraform import command is used to import existing infrastructure.

11.What is Workspace in Terraform?

Workspace can be used if we have multiple project/environments using the same configuration.

KUBERNETES:

1.What is Kubernetes?

Kubernetes is also called as k8s.

k8s is an open source popular orchestration tool developed and used by google in prod env.

We have different orchestration technologies like dockerswarm,kubernets and mesos.

2.What are the different components in k8s?

We have two Kind of machines in k8s.

1) Master

2) Worker node

Again we have different components on master and worker nodes.

Master Components:

==============

1) Kube api server

2) etcd

3) Controller

4) Scheduler

worker node components:

====================

1) Docker container runtime

2) Kubelet agent

3) Kubeproxy

Note: kubectl is a command line tool to execute k8s commands.

Api server: Acts as frontend of k8s,Authenticate user,authorization and api server is the only gateway to communciate with our k8s cluster.

Etcd: Cluster brain,which stores the information related to cluster,node in the form of key value pair.

Scheduler: Responsible to schedule pods on the node machine.

Controller manager: controller is the brain behind orchestration,if nodes,container,endpoint goes down then controller will work behind to bring them up.

Container runtime: any container service to start,stop container (Docker).

kubelet: Kubelet run on all the machines,kubelet make sure that the containers are running as expected on node machines.

KubeProxy: It maintains network rules on your nodes and enables network communication to your Pods.

3.What is Pod?

pod is the smallest unit in k8s.

pod will have atleast one container.

4.What is Helper Container?

we can have multiple container in one pod.

one container with application and another container will be helper container.

Helper container: this will be deployed along with our application,helper container will help our application with any processing/functionality.

5.List out some Basic commands of Pods?

kubectl run nginx --image=nginx --> to run a pod with name nginx and image nginx

kubectl get pods --> to display the list of pods

kubectl describe pod nginx --> Detail information of the pod created.

kubectl get pods -o wide --> additional information of pod like node,ip etc

kubectl explain pods -->Detailed description of pods

kubectl get pods -w --> Continously watch the status of POD

kubectl delete pod pod_name --> To delete the pod

kubectl delete resourcetype resourcename

kubectl run nginx --dry-run=client --image=nginx --> To check how the command will execute.

kubectl delete pods --all --> Delete all pods

6.What is Labels?

Labels are key-value pairs which are attached to pods,
replication controller and services. They are used as identifying
attributes for objects such as pods and replication controller.
They can be added to an object at creation time and can be added or modified at the run time.

7.Basic commands to check/attach/delete labels?

kubectl describe pod <pod_name> --> to check the label attached

kubectl label pod firstpod env=test --> To attach label env=test

kubectl label --overwrite pod firstpod env=prod --> To update the label env

kubectl label pod firtpod env- --> To delete the label env

Kubectl label pods -all status=xyz --> To update the label to all pods

kubectl get pods --show-labels --> to check labels

8.Define the mandatory fields in YAML?

k8s definition filed contains four top level fileds.

apiversion

kind

metadata

spec

kind	version
POD	v1
service	v1
ReplicaSet	apps/v1
Deployment	apps/v1

9.How to run pod using YAML?

```
kubectl create -f <file_name>.yml --dry-run --> to check the output
kubectl create -f <file_name>.yml
```

10.Sample YAML file to create pod?

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: firstpod
  labels:
    env: prod
spec:
  containers:
  - name: nginx
    image: nginx
```

10.What is Namespaces in k8s?

These are used to group your applications.

They can be helpful when different teams or projects share a Kubernetes cluster.

11.Difference between create and apply?

kubectl supports three types of object management.

1) Imperative commands --> Which are not created from yaml

2) Imperative object configuration --> Kubectl create is example for imperative.

3) Declarative object configuration --> Kubectl apply is example for declarative.

Imperative :

You have to manage different resources like pods, service, replica sets, etc by your own.

Imperative object configuration will helps us to modify the objects and these changes are not stored in yaml.

The kubectl create command creates a resource from a file or from stdin. JSON and YAML formats are accepted.

If the resource already exists, kubectl create will error.

Declarative :

K8 will take care of all the resources, all you need have to specify what is your actual requirement.

Declarative object configuration will helps to modify the yaml file.

The kubectl apply command applies a configuration to a resource by file name or stdin. The resource name must be specified.

This resource will be created if it doesn’t exist yet. If the resource already exists, this command will not error. JSON and YAML formats are accepted.

12.What is init container in pods?

Init containers are used if you want any container to execute before the app container starts.

Init containers are exactly like regular containers, except:

Init containers always run to completion.

Each init container must complete successfully before the next one starts.

13.How to create init container?

```
apiVersion: v1
kind: Pod
metadata:
  name: firstpod
  labels:
    env: prod
    name: sabair
spec:
  containers:
  - name: firstcontainer
    image: nginx
    env:
      - name: myname
        value: sabair
   - name: secondcontainer
    image: nginx
  initContainers:
  - name: initcontainer
    image: nginx
    env:
      - name: myname
        value: sabair
      - name: City
        value: Hyderabad
    args: ["sleep", "30"]
```

14.Services in k8s?

Services are used to expose our pods to outside world.

For example we have deployed a apache webserver and want to access from our computer/outside network then it is possible with services only.

services are also a object like nodes,pods in k8s.

Default service will be ClusterIp in k8s.

if we dont specify type in spec then it will assume as ClusterIp only.

15.How many types of services are available in k8s?

Three types of services are available on k8s.

1) ClusterIp

2) Nodeport

3) Load balancer

CLusterIp:

Cluster ip is used to communicate within the cluster.

we may have some pods running frontent,backend and database.

And in micro services frontent will be talking to backend and databases.

we can create a service for backend and database and group the pods related to Microservice.

Each service will have a ip and we call it as ClusterIp which help to communciate with other services.

NodePort:

NodePort is used to access for pod outside the world,means via browser.

Port can be assigned between (30000 - 32767).

LoadBalancer:

LoadBalancer service can be used if we are using any cloud platform elb.

16.How services will work?

Services will check the request which is sent on port and redirect that based on label.

17.Replication Controller in k8s?

Replication controller or RC will ensure that these many pods are running on the cluster.

For any reason if the pod is down then RC will monitor that and spin up a new pod for us.

RC will get attached to the pods based on labels and selectors.

18.Why do we need Labels and selectors?

If suppose we have 100 of containers running then labels and selectors will help RS to filter the containers and apply monitoring based on labels and selectors.

19.Replica set in k8s?

RS and RC both purpose is same but they are not same.

RC is older technology and k8s recommend to use RS.

RS is just the updated version of RP.

20.Replication controller vs Replica set Difference?

RC and RS functionality is almost similar.

RC will work based on equality based selector.

Ex: env = prod

RC selects all resources with key equal to env and value equal to prod.

RS will work based on equality based selector and set based selector.

Ex: env = (prod, test)

RS selects all resources with key equal to env and value equal to prod or test.

21.What is Deployments in k8s?

Deployments are used to deploy,upgrade,undo the changes,pausing and again resuming the changes.

Deployments use rolling update means the new version will be slowing updated with the older version

and the users will be able to access our application without downtime.

Hierarchy for deployment:

pods --> replicasets --> Deployment

22.How many types of Deployment startegies are used in k8s?

There are two types of depoyment strategy in k8s.

1) Recreate

This will destroy the existing pods and create new.

We can see downtime in recreate deployment strategy

2) Rolling update

This is the default deployment strategy.

No downtime for rolling update strategy.

For example if we have 5 pods then 1 will be destroyed and a newver version will be created and 4 will be servering the traffic.

In this way it will destroy all the 5 and create new 5 pods slowly.

we can see the difference when we use kubectl describe deployment deployment_name command

In recreate it will scale down the RS to 0 and then Scale up to 5.

In Rolling it will scale down to 4 and and bring one new up and then vice versa.

23)what is Rollout,Rollback and versioning?

when we create deployment then a rollout is created and a Revision is created.

When we again update the deployment then again a rollout will be carried and New Revision is created.

These revisions are helpful when we want to rollback to previous version.

Note: Rollout will get triggered only when there are change in the container configurations only.

24)What is Maxunavailable and MaxSurge in Deployments?

Maxunavailable means the number of unaavailable pods at the time of update.

If we configure Maxunavailable as 1 then it will create one new and delete one old.

If 5 as value then create 5 and delete 5.

This will help us to complete the deployment faster.

Max Surge: This will imply the number of pods that will be there on top of total replicas being mentioned.

Example if the replicas in the deployment is mentioned as 3, when rolling update will kick in this property

will define how many extra pods will be created at that point of time.

25)What will be the Maxunavailable,MaxSurge,MinReady Seconds and strategy if not mentioned in deployment?

Strategy: RollingUpdate

MinReady Seconds: 0

Max surge and Max unavailable: 25%

26)How to check the history of deployment?

kubectl rollout history deployment <Deployment_name>

27)what is rollback in k8s and how to do the rollback?

Rollback will helps us to get back to old version of deployment.

How to do the rollback:

kubectl rollout undo deployment <deployment_name>

This will roll back to previous version.

28)what is Resource Request in k8s?

When we schedule a pod on any node without any resource request then it will automatically take the resource
available on the node.
Resource request will be RAM and CPU request.

29)Why do we use resource request ?

If suppose our pod needs 200MB of RAM then scheduler will check on which node the RAM is available
and then it will schedule the pod on that node only.

30)what is Resource Limits in k8s?

Resource limit will ensures that container is not using more than the specified RAM and CPU.

30)Namespaces in k8s?

In Kubernetes, namespaces provide a way to divide a cluster into virtual partitions or segments.
They act as a logical boundary that isolates and separates resources within a cluster.
In Kubernetes,A cluster can be divided into multiple namespaces, and each namespace can have its own set of resources.
This helps organize and manage applications and services running in the cluster.
By Default the pod what we have created will be created in "Default" namespace.

31)Some important commands related to namespaces?

kubectl get ns --> List of name spaces

kubectl create ns <name> --> To create namespace

kubectl apply -f po.yml --namespace test --> Create pod in ns

kubectl get pods -n test --> to list pods in sn test.

kubectl get pods --all-namespaces --> To view the pods running in all namespaces.

kubectl delete pod <pod_name> -n test --> To delete pod from namespace test.

kubectl api-resources --> To check what all can be created inside namespace.

kubectl delete ns <namesapce> --> To delete the name space

kubectl config set-context --current --namespace=test --> To make test as default

32)How many namespaces are available in k8s by default and what is there use?

when we create k8s cluster then by default we can see 4 name spaces.

1)default

This ns will get created when we try to create any object without mentioning namespace

2)kube-node-lease

node lease is a mechanism for workers nodes to master about there health status and they are ready to take workloads.

lease should be sent for every 60 secs,if lease is not renewed then master may be considered nodes as unhealthy or unresponsive.

3)kube-public

serves as a central location for storing public resources that need to be accessed by all users and service accounts within a Kubernetes cluster.

4)kube-system

All the pods,management related stuff to create clusters are stored in kube-system

33)What is Service DNS?

In k8s service DNS, also known as the service discovery DNS,

is a built-in mechanism that allows communication between services using their names instead of their IP addresses. Each service deployed in a Kubernetes cluster is assigned a DNS name that can be used by other services to access it.

34)what is Resource Quota in Namespace:

Resource quota in k8s helps us to enforce resource limits on namespace.

We can restrict out namesapce with certain cpu,memory and storage.

We have two types of resource quota.

1) Resource based Quota.

2) Compute based quota.

Resource/object Based Quota:

=====================

In resource based quota we are going to set how many objects can be created in namespaces.

Ex: pods,services,PVC,RS etc. which are supported by namespaces.

Compute based Quota:

====================

In compute based quota we can restrict the namespace with certial CPU limits.

35)what is LimitRange in k8s?

The LimitRange resource in Kubernetes allows you to define default and maximum resource limits for

containers running within a namespace.

It helps ensure resource fairness and prevent containers from consuming excessive resources.

36)What is max and min in Limit range?

Min is used to allocate min amount of cpu and memory to a container.

max is used to allocate max amount of cpu and memory to a container.

Min should be always lesser or equal to default request.

Max should be always greater or equal to default limit.

37)What is Max Limit/Request Ratio?

This will helps us to fix a ratio between max limit and max request.

If suppose the Max Limit/Request ratio is set as 2

and our max limit = 1000 and max request = 100 then

1000/100 = 10 : value we have set is MAX 2 then it will not create the pod.

38)What is Config Maps?

ConfigMaps in Kubernetes are used to store non-sensitive configuration data that can be consumed by pods or other Kubernetes objects. They provide a way to decouple

configuration from application code, making it easier to manage and update configurations without

redeploying the application

ConfigMaps can be created in different ways

--from-literal

--from-file

--from-env-file and

from directory.

39)How to create config maps?

kubectl create cm <Config_map_name> --from-literal=database_ip=192.168.0.1

40)What is Secrets in k8s?

Secrets in k8s is used to store small amount of secure data like passwords.

Small amount of data can be upto 1MB of size.

When ever we try to create any secret then it will be converted into base64 and stored.

Secrets Will be again of 3 types:

1) Docker-registry

2) Generic

3) tls

Most commonly used will be generic.

Secrets can be created in different ways same as configmaps like

--from-literal

--from-file

--from-env-file and

from directory.

40)Taint and Tolerations in k8s?

Taint and Tolerations are advance k8s pod scheduling technique.

Taint are attached to Node.

Toleration are attached to Pods.

Taint in general terms means a filter.

Tolerations with matching taint will be scheduled on tainted node.

Tolerations allow the scheduler to schedule pods with matching taints.

41)Node Selector in k8s?

Node selector helps us to select node where the pod can be scheduled.

It will work based on the label attached to node.

42)Node Affinity in k8s?

Node affinity is a set of rules used by the scheduler to determine where a pod can be placed.

Two types of Node Affinity is available:

1) PreferredDuringSchedulingIgnoredDuringExecution (Soft Scheduling):

Means we are asking scheduler to check if the values/label is matching then schedule pod on the node and

if node with same label is not available then it will schedule on any available node.

Ignored During Execution means if the pod is already scheduled on node and later on we have removed the label then

the pod will be ingored and continue running on the same node.

2) RequiredDuringSchedulingIgnoredDuringExecution (Hard Scheduling):

Means we are asking scheduler to check if the values/label is matching then schedule pod on the node.

If not matched then pod will not be scheduled in other nodes.

Ignored During Execution means if the pod is already scheduled on node and later on we have removed the label then

the pod will be ingored and continue running on the same node.

43)K8s Volumes?

Volumes in k8s are used to store persistent data.

We have two types of pods one is stateful,other can be stateless.

Stateful: In simple terms sateful applications can be storing any data.

Stateless: In simple terms stateless applications will not be storing any data.

44)How many types of volumes are available?

1) EmptyDIr:

============

We create volume inside a pod to store data related to container.

If the container is killed for any reason, a new container will be created in the same pod and

the same volume will be attached to container.

2) Hostpath:

===========

We create volume on hostpath, means volume will be created outside pod.

If pod get deleted and a new pod is created then it can access the volume available on the host.

3) Amazon Elastic Block Storage:

================================

If we have multinode k8s cluster,then in this case we need to keep our volume outside the cluster.

If our pod is created on other node then the volume should also be moved to that node.

4) Amazon Elastic File Storage:

================================

If we have multinode k8s cluster,then in this case we need to keep our volume outside the cluster.

If our pod is created on other node then the volume should also be moved to that node.

45)Daemon Set in k8s?

DaemonSet is a type of workload that ensures that a specific pod is running on each node in a cluster.

It is useful for scenarios where you need to run a single instance of a pod on every node, such as log collection or monitoring agents.

46)Liveness Probe in k8s?

Liveness can use with pod configuration to check the health status of pod.

If pod in not healthy for any reason then Liveness will restart the pod.

47)Readiness Probe in k8s?

Readiness can use with pod configuration to check the if the application running in the pod is ready to serve traffic.

If pod in not healthy for any reason then Readiness will remove the pod from endpoints and it will be in "not Ready" status.

Recent issues:

1)We were able to see lot of tickets from development teams that they are not able to see the logs of production server on splunk centralized server.
We checked if the splunk forwarders are running or not on that servers and found that these forwarders were down for some reason.

So we have created a bash script to check the status of splunk forwarder for every 5 mins and alert the team if these forwarders are down ,so that we can forecast the issue and make sure splunk forwarders are up and running always
2) In Jenkins we can able to see lot of issues with connectivity, this will happen when there is a change in the systems,i mean bcz of patching activities..

And also pipelines will get failed bcz of any missing dependencies or any kind of missing artifacts.

Recent challenge/achievment:

1) Used s3 bucket to store terraform statefiles and also used dymanod db locking to protect statefiles which not be used by multiple users at the same time.