**Self-Introduction**

* 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, Aws cloud and physical Data Centers.
* In aws i have worked on various services like ec2,s3,route53,application load balancer,VPC,cloud watch for monitoring.

**Roles and Responsibilities:**

* we worked along with development teams for next releases.
* If we have any monitoring alerts then i will be working on that alerts.

Managaing jenkins pipelines and other infrastructure.

* New on boarding of application to our environment.
* 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.

* Prepatching and post patching means if patching is schedule at 2:30 then 30 mins prior patching (2 p.m) we bring down all the services running on the servers, once patching is done then will bring up all the services.
* 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 bringin 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 benifits
* 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:**

**7.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

**8.Command to check for the running ports?**

netstat -na | grep port\_number

**9.Command to check the processes running?**

ps -ef

**10.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/

**11.What is swap memory?**

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

**12.Command to kill a process?**

kill -9 PID

**13.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.

**14.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.

**15.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.

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

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

**Bash Scripting:**

**#! in bash scripting means = Shebang**

**17.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"

**18.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

**19.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

**20.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.

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

$# = Total number of command line arguments passed.

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

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

$\_ = special variable

**21.How to run bash script in the background?**

Use nohup to run the script in background.

Example:

sudo nohup ./hello\_world.sh

**22.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.

Git & GitHub:

**23.What will git init do ?**

This will help us to initialize .git folder in local repository.

**24.What does .git contains?**

**.git** file contains the configuration of the repository,like hooks,config etc..

**25.Why do we use git ?**

Git is a version control tool which is used to track the changes of the project.

**26.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.

**27.Git cherry-pick?**

Cherry-pick is used to pick a specific commit of a branch to another branch.

**28.Git stash?**

For example if iam working on a specific branch and i dont 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.

**29.Git rebase?**

This is used to Merge two different branches.

**30.Git merge?**

This is used to merge two different branches.

**31.Git fork?**

This is used to copy a repo from other account to our account on Github.

**32.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.

**33.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.

**34.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..

**35.Is Git a distributed repository system or central repository system?**

Git is a central repository system.

**36.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.

**37.Which version of git are you using?**

git version 2.34.1.windows.1

**38.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.

**39.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.

**Apache Tomcat:**

**40.Default port number of Tomcat?**

Tomcat runs on 8080 port number.

**41.Default Deployment location of Tomcat?**

Webapps will be the default deployment lcoation.

**42.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.

**43.How to create Users in tomcat?**

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

**44.Tomcat is Developed in \_\_\_\_\_\_\_\_\_\_\_ language**?

Java language.

**45.What are the dependencies for tomcat?**

Java is the only prerequisite for tomcat.

**46.Apache Tomcat is \_\_\_\_\_\_\_\_ server?**

Application Server.

**47.Difference between app and web servers?**

App server is mainly used for to serve dynamic content.

Web servers are used to serve static content.

**48.list of app servers available in the market?**

Tomcat

Jboss

Weblogi

**49.Which version of tomcat are you using?**

Tomcat-9.0.65

**50.File name of Apache Tomcat Logs?**

catalina.out is the log file for tomcat.

**51.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:**

**52.Default port for apache?**

Default for Apache is 80.

**53.Default port for nginx?**

Default port for Apache is 80.

**54.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.

**55.Default deployment location for apache and nginx?**

Apache:/var/www/html/

Nginx: /usr/share/nginx/html

**56.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

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

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

Nginx: /etc/nginx/nginx.conf

**HA PROXY:**

**58.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.

**59.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.

**60.Configuration file name of HA PROXY?**

haproxy.cfg

**61.How to configure HA Proxy?**

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

**AWS:**

**62.List of aws services you have worked on?**

I have worked on IAM,ec2,s3,applciation loadbalancer,Route53,VPC,CLoud watch and cloudformation templates.

**63.What is region and availability zone in aws?**

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

**64.what is edge location?**

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

**65.Name the global services of aws?**

S3, IAM, cloudfront, route53

**66.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:**

**67.Ec2 stands for?**

Elastic Cloud Compute.

**68.How many types of ec2 are available?**

On Demand,Reserved and Spot Instances.

**69.Different families of ec2?**

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

**70.What is the type of ec2 you are using?**

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

**71.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.

**71.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.

**72.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.

**73.How to take a snapshot of ec2?**

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

**74.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 i will be havings jenkins in it.

**75.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't be chnaged to t2.large.

we can increase the ebs volumes but not decrease the volume.

**AMI:**

**76.What does AMI stand for ?**

AMI stands for Amazon machine images

**77.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.

**78.What is AMI?**

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

**79.What is AMI id ?**

Thats the unique identifier number for each AMI.

IAM:

**80.IAM stands for?**

Identity access management.

**81.How to create a user using IAM?**

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

**82.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.

**83.What is the role of IAM?**

IAM riles are attached to users to provide specific access across the aws acoount.

**84.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.

**85.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.

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

We can provide two types of access.

COnsole/GUI access

Programmetic/CLI Access.

**EBS:**

**87.EBS stands for ?**

Elastic Block storage

**88.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.

**89.What is an EBS snapshot?**

EBS snapshot is a copyof certain time/backup of ebs volume.

**90.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**:

**91.VPC stands for?**

Virtual Private Cloud

**92.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 differet availability zones. Creat 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 assosciations.

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

**5**

**94.What is CIDR ?**

Classless inter domain range

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

256

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

five

**97.What are subnets?**

Subnets are the rang of ipv4 address in the VPC.

**98.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.

**99.What is IGW?**

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

**100.What is RT?**

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

**101.What is NAT gateway/ NAT instance?**

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

NAt gatewat 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.

**102.Nat will be deployed in which subnet?**

Public

**103.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.

**104.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.

**Load Balancer**:

**105.Full form of ELB?**

Elastic Load balancer.

**106.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)

**107.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.

**108.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.

**109.How does Network Load Balancer work?**

Network Load balancer works on TCP/UDP protocol

It works on session 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.

**110.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:**

**111.ASG stands for ?**

Autoscaling Groups.

**112.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.

**113.Different types of scalings in ASG?**

We have Horizontal and vertical scalings.

**114.What is Horizontal Scaling?**

Horizontal scaling means adding additional ec2 to our environment.

**115.What is Vertical Scaling?**

vertical scaling means adding more power to our existing ec2.

**S3 Buckets:**

**116.Why do we use s3 Buckets?**

S3 buckets are used to store fixed objects.

**117.What are objects in s3?**

Objects can be referred to any file uploaded in s3.

**118.Default s3 bucket class?**

Standard.

**119.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

**120.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.

**121.What is versioning?**

Versioning helps us to store file with multiple version.

**122.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.

**123.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.

**124.What is object lock?**

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

**125.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:

**126.What is Route53?**

Route53 is a DOMAIN NAME SERVICE in aws.

**127.Is Route53 a global or regional servie?**

GLobal Service.

**128.What is Alias in Route53?**

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

**129.What are nameservers in Route53?**

**130.What is nameserver in Route 53?**

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.

**131.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)

**132.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.

**CLoudwatch:**

**133.What is cloudwatch?**

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

**134.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.

**135.How to configure a cloudwacth alert to a specific custom servcie?**

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

**136.What is SNS?**

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

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

CPU Utilization

Service running or not Metric

Disk write and out metric etc..

**138.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.

**139.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.

**Cloudformation:**

**140.Why do we use cloudformation?**

Cloudformation Templates are used to provision infrastructure on aws environment.

**141.What is IAC?**

Infrastructre As Code.

**142.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.

**143.What are stacks?**

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

**144.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

**145.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

**146.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.

**147.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.

**148.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.

**149.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.

**150.Sample CF template to create VPC,2subnets,Launch ec2,create Loadbalancer and attach the ec2 as target group to loadbalancer?**

**AWS RDS**:

**151.RDS means?**

Relational Database Service.

**152.What is the use of RDS?**

RDS is used to store the data related to applciation.

With the help of RDS service we can directly provision a ec2 with db and start using the service.

**153.What are the types of DB Storages?**

IOPS helps in creating report of IOPS value for every minute.

Latency helps in indicating a slower performance.

Throughput is the number of bytes transferred to and from the disk every second.

Queue Depth helps in indicating a slow performance.

**154.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.

**155.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.

**156.How can we connect to MS SQL RDS?**

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

**157.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.

**158.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:**

**159.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).

**160.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.

**161.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.

**162.What is Docker file?**

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

**163.What is Docker image?**

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

**164.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.

**165.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"]

**166.How to build the dockerfile?**

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

docker build -t /pathofDockerFile

**167.Difference between CMD and EntryPoint?**

CMD and ENtryPoint 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 Overriable and Entrypoint is not overridable.

CMD and ENTRY both can we used is same DockerFile but thats not the best practise.

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"

**168.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.

**169.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.

**170.How to list the images in docker?**

docker image ps

**171.How to delete the images in docker?**

docker image rm <image\_name>

**172.How to get the details of docker image?**

docker inspect <Docker\_image>

**173.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.

**174.What is tag in docker images?**

Tag in docker images specify the version of the image.

**175.What is docker registry?**

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

**176.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>

**177.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)

**178.command to check the running containers?**

docker container ps

**179.Command to Get the details of containers?**

docker container inspect <COntainer\_id>

**180.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>

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

docker container ps -a

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

-IT = interactive Terminal

-D = Detach mode /foreground/background

**183.What is port mapping in docker containers?**

Port mapping is used to expose a port to public.

**184.How to connect to the running container?**

docker container exec -it <Container\_id> /bin/bash

**185.How to kill a running container?**

docker kill my\_container

**186.How to check the logs of container?**

docker container logs <container\_id> --follow

**187.How to commit a running container?**

docker save -o <File\_NAME>.tar <Image\_Name>

**188.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

**189.Command to run docker container with a volume attached?**

docker run -d --name <container\_name> -v <Volume\_name>:/app <Image\_name>

**190.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.

**191.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.

**192.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 dockerswarm.

**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: { }

**193.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

**194.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

Export/Import Docker Image to file

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

docker save image:tag > arch\_name.tar

docker load -i arch\_name.tar

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

Kill and Delete Containers and Images

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

docker rm -f $(docker ps -aq) # Delete all Containers

docker rmi -f $(docker images -q) # Delete all Images

**Ansible**:

**195.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.

**196.Explain the architecture of ansible?**

Playbooks

Modules

Inventory

Plusgins

**197.Is ansible push based or pull based?**

Push based.

**198.What is inventory in ansible?**

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

**199.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.

**200.What is playbook in ansible?**

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

**201.ansible is written in which language?**

Python

**202.Ansible playbooks can be written in which language?**

YML Language.

**203.Tell 15 modules in ansble?**

Yum

Service

Shell

lineinfile

dir

copy

handlers

notifiers

apt

service

debug

register

file

git

archive

**204.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

**205.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.

**206.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

**207.What is ansible vault?**

Ansiable vault is used to store the files with encryption.

if we have password then we use the vault to encrypt the file.

**208.What is handlers in ansible?**

handlers are used if we have dependecy 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.

**209.What is ansible galaxy?**

**210.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.

**211.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.

**212.Wtire 5 Sample Ansible playbooks?**

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

**Jenkins:**

**213.What is jenkins?**

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

**214.What is CICD?**

Continous Integration,Continous Deployment/Continous Delivery.

**215.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.

**216.Jenkins is written in which language?**

Jenkins is written in JAVA.

**217.Jenkins run on which port number?**

8080

**218.What is Jenkins file?**

jenkinsFile is a text file which consists of different stages.

Each stage will have a set of task to be executed.

**219.What is Jenkins pipeline?**

jenkins Pipeline will be executed based on jenkinsFile,where differen stages will get executed.

We have two types of pipelines,

1) Declarative

2) Scripted

**220.Types of jenkins job/projects?**

Freestyle project.

Maven project.

Pipeline.

Multibranch pipeline.

External Job.

Multi-configuration project.

Github organization.

**221.Name some jenkins plugins?**

Git

Java

Maven

role based access

Docker

tomcat

sonarqube

Nexus Artifactory

Slack

**222.What is workspace in jenkins?**

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

**223.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.

**224.How many types of pipeline we have in jenkins?**

We have two types of pipelines,

1) Declarative

2) Scripted

**225.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....'

}

}

}

}

**226.How to seggregate user roles in jenkins?**

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

**227.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.

**228.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.

**229.What is label in slave?**

Label are names to define our slave machines.

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

**230.What is executors in jenkins?**

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

**231.Write sample jenkinsPipeline?**

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";

}

}

}

}

}

}

**232.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.

**233.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/

**234.What is Maven?**

Maven is build tool used to build java applications.

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

**235.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.

**236.What is Pom.xml?**

PROJECT OBJECT MODEL.

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

**237.Maven Repositories?**

Local

Central and

Remote Repositories.

**238.What is sonarqube?**

Sonarqube is a tool to check the qulaity of code.

**239.What are sonarscanners?**

**240.What does Sonar scanner do?**

SonarScanner is a separate client type application that in connection with the SonarQube server will run project analysis

and then send the results to the SonarQube server to process it.

**241.What is qualitygate 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.

**242.Sonarqube runs on which port number?**

9000

**243.What is nexus?**

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

**244.Nexus runs on which port number?**

8081

**245.What is snapshots and releases in nexus?**

Snaposhots are not the final version of artifact and can be replaced.

Release are final artifact version and this cannot be replaced.

**Terraform**:

**246.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

**247.What are terraform provisioners?**

**248.What is a Provisioner Terraform?**

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

**249.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.

**250.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.

**251.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.

**252.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.

**253.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.

**254.What are components of terraform?**

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

**255.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.

**256.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.

k8s:

====

**257.Statefullsets vs deployment?**

**258.Secret vs configmap?**

**259.Node vs pod?**

**260.What is helm?**

**261.Daemonset vs replicaset?**

**262.K8s architecture?**

**263.Services types?**

**264.Uses of taints and tolerations?**

**265.K8s basic commands?**

**266.What is Init container and uses?**

**267.Init container vs side car container?**

**268.Swarm or docker vs k8s?**

**269.Pv and pvcs?**

## ERROR CODES

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’s 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 data 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 authorized 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 Timed out**

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.

## KUBERNETES

**Statefulsets vs deployment?**

**Suitable Use Cases**

|  |  |
| --- | --- |
| **STATEFULSET** | **DEPLOYMENT** |
| Stateful workloads that require persistent storage on each cluster node | Stateless workloads such as web servers |
| Ideal for key-value stores or database systems | For balancing requests between replicas |
| Assigning ordered, unique identities | To automatically scale (creation/termination) pod replicas of a cluster |
| For gradual rollouts | For controlled rollout and rollbacks |
| Deploy new pod replicas incrementally | For deploying new pods with similar environment variables and ConfigMaps |

Secret vs configmap?

Node vs pod?

What is helm?

Daemonset vs replicaset?

K8s architecture?

Services types?

Uses of taints and tolerations?

K8s basic commands?

What is Init container and uses?

Init container vs side car container?

Swarm or docker vs k8s?

PV and PVCs?

# Kubernetes

### Why Kubernetes?

Monolithic Services Vs Micro Services

|  |  |
| --- | --- |
| **Monolithic Service** | **Micro Services** |
| Total application is running in single (one) server | The application divided into multiple parts, means every service is divided into different PODs |
| Example : If any changes happened in application, in one service we need to modify all applications (Application need to re-deploy if any services is change)🡪 it is monolithic.  Mostly this service is used in Banking sectors  like Account, Debit, Credit, Loans, Balance (it will be written in one language like java, .net, python) | Example : UBER  If there are 7 services in the Uber (Micro services)  We can maintain each application individual service / database / language/ Separate end point  One Big monolithic, we divide into individual services, it is called as micro services  If we want any update in one service of application / we can update easily for the specific service only without disturbing any other services. And there is less down time. |

### Automated rollouts and roll backs?

Kubernetes progressively rolls out changes to your application or its configuration

While monitoring application health to ensure

It doesn’t kill all your instances at the same time.

If something goes wrong, Kubernetes will roll back the changes for you.

Take advantage of growing ecosystem of deployment solutions.

### Storage orchestration:

Automatically mount the storage system of your choice, whether from local storage, a Public Cloud Providers such as AWS, GCP

### Secret and configuration management:

Deploy and update secrets & application configuration without rebuilding your image and without exposing secrets.

### Service discovery and loading balancing

No need to modify your application to use unfamiliar services discovery mechanism

Kubernetes gives pods their own IP address and single dns name for set of Pods and can load balance across them

### Self Healing

Restart containers that fail, replace, and reschedule the containers when the node dies, kills container that don’t respond to your user defined health check. And doesn’t advertise them to client until they are ready to serve

### Automatic bin packing

Automatically places containers based on their resource requirements and other constraints while not sacrificing availability.

Mix critical and best effort workloads in order to drive up utilisation and save even more resources.

### Horizontal scaling

Scale your application up and down with a simple command with UI (or) automatically based on CPU usage

### Designed for extensibility

Add features to your Kubernetes cluster without changing upstream source code

### Batch execution

In addition, services, Kubernetes can manage your batch and CI workloads, replacing containers that fail if desired.

### IPv4 / IPv6 dual-stack

Allocation of IPv4 and IPv6 addresses to pods and services

### What can Kubernetes do for you?

With modern web services

Users expect applications to be available 24/7

And developer expect to deploy new version of those application several times a day

Containerisation helps package software to serve these goals.

Enabling application to be released and update without downtime

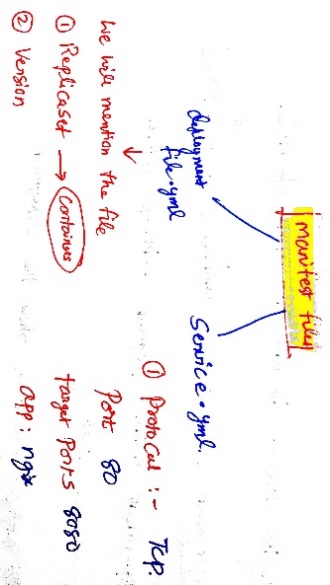
Kubernetes helps you make sure those containerised applications run where and when you want and helps them find the resources and tools they need to work

Kubernetes is production ready, open-source platform designed with google accumulated experience in container orchestration

Combined with best of breed ideas from the community.

# KUBERNETES BASIC COMMANDS

### Kubectl commands

* Kubectl get nodes 🡪 get the list of nodes
* Kubectl version 🡪 will get the version of Kubernetes
* Kubectl describe node 🡪 it will show details of node
* For create deployment – 2 ways [command / manifest file]
* Create deployment nginx-deployment --image nginx 🡪nginx deployment will create.
* Kubectl get deployment 🡪 it will show the deployments
* Kubectl get replicaset🡪 will get replica sets
* We can check the details of deployment 🡪 Kubectl describe pod nginx-deployment – 688cd6f (replicaset ID)
* Logs to check 🡪 Kubectl logs nginx -deployment-
* Kubectl help 🡪
* Kubectl get services 🡪 we can see the services
* Kubectl get pods -o wide 🡪 will get IP details of deployments
* To apply (execute manifest deployment file) 🡪kubectl apply -f nginx -service.yml

If deployment file.yml is executed with the help of service.yml file. If service.yml file is not there deployment file.yml will not be executed.

### Taints and Tolerations

Taints and tolerations work together to ensure that pods are not scheduled on to inappropriate nodes.

Taints are Kubernetes node property that enable nodes to repel certain pods

One or more taints are applied to a node

This mark that the node should not accept any pods, that do not tolerate the taints

### What is init container and uses?

Init container can contain utilities or custom code for setup that are not present in any app image

Ex: there is no need to make any image from any other image just to use a tool like SED/AWK / Python

### Side car container Kubernetes?

Side car containers are containers that are needed to run alongside the main container

The two containers share resources like pod storage and network interfaces

The side car containers can also share volumes with main containers, allowing the main container to address the data in the side car

### What is POD?

* Pod is a wrapper which can contains one or more containers
* A pod can have multiple containers
* One will be main container
* Which serves the application traffic
* The other containers are called as side car containers

### What is replica set?

* Replica set is the next generation replication controller that supports the new set – based label selector
* Its mainly used by development as a mechanism to orchestrate pod condition, deletion and updates

### Why do we use replica sets?

* The main purpose of replica set will be make sure that the desired number of pods running at any cost in case of any downtime happens, then the pods are serving the notes
* The main drawback is not the job description, replication control is updating the image.

### What is Kubernetes deployment?

* A Kubernetes deployment tells Kubernetes
* How to create or modify the instances of the pod that hold a containerised application
* Deployment can help to efficiently scale the number of replica pods, enable the rollout of updated code in a control manner (or)
* Rollback to an earlier deployment version of necessary

Add no. of pods which will be created while you are updating

No. of pods it can be delete during the updating

### Min ready seconds

Spec min ready seconds is an optional field that specifies the minimum number of seconds

Which a newly created pod should be ready without any of its containers crashing for it to be considered available.

### Kube API server:

The Kube API server express the Kubernetes API, it is where the Kubernetes resources are published into.

### What is Kubernetes?

Kubernetes is a container orchestration tool, that is used for automating the tasks of managing, monitoring, scaling and deploying containerized applications

It creates a group of containers

That can be logically discovered and managed for easy operation on containers

### What is difference between Kubernetes and docker swarm

Docker swarm is a default container orchestration tool that comes with docker

Docker swarm can only orchestrate simple docker containers

Kubernetes on the other hand, helps manage much more complex software application containers

Kubernetes offers supports for larger demand production environment

Docker swarm cannot do autoscaling, docker swarm doesn’t have GUI

Docker swarm can deploy rolling updates, but can’t deploy roll backs

Docker swarm requires the third-party tool like ELK stack for logging and monitoring

While Kubernetes has integrated tool for same

Docker swarm can share storage volumes with any container easily

While Kubernetes can only share storage volumes with containers in the same pod

### What are the different services within the Kubernetes?

We can push the internal to the pod or deployment we have need a services

### Cluster IP services

Can be able to reach from inside the cluster

**Node port:**

It some ephemeral ports

Default 30000 to 32767 (node port default range)

**External Name:**

It is used for to connect the outside of Kubernetes service.

**Load balancer:**

LB, service can be used for the production of the applications

### How to set a static IP for k8s LB?

Kubernetes master assigns a new IP address

We can set a static IP for Kubernetes load balancer by changing the DNS records, whenever Kubernetes master assigns a new IP Address

### Can we many claims out of a PV?

The mapping B/W PV and PVC is always one to one, even when you delete the claim.

Persistent volume still remains as we set persistent volume reclaim policy is set to retain and it will not be rescued by any other claims

### How do you deploy a feature with zero down time with Kubernetes?

In Kubernetes you can define the update strategy in deployment, you should put rolling update as a strategy to ensure no downtime

### What is the difference between replication controller and replica set?

Replication controllers are obsolete now the latest version of Kubernetes

The only difference between replication controller and replica sets is the selectors

Replication controllers doesn’t have selector in their spec

### Explain PVC?

PVC stands for persistent volume claim

Its storage requested by K8s for Pods

The user doesn’t require to know the underlying provisioning

This claim should be created in the same name space

Where the pod is created

### If you have a sensitive information in your cluster, how would you secure it?

We cane pass sensitive information in K8s using secrets

Secrets can be created through Yaml and text files of the organisations

Use secretes to pass sensitive information like username and password.

### If you delete a pod? what happens to information inside of LB?

Deployment will make sure that a new pod created to maintain the number of replicas.

Now talking about the information it depends on the type of volume, mount used.

If you want information to be retained, then you need to use PV.

### Is there any pattern to pods being assigned to node?

### Can you make sure a pod gets scheduled to particular node?

Generally when you create a pod it spawns automatically on any node but lets say if you want spawn a pod on particular node that can also be done through taints

### What is Kubestate metrics?

The kube state metrics is a simple service that listens to the Kubernetes API server and generates metrics about the state of the objects. It is focused on the health of the individual Kubernetes components, rather on the various objects inside, such as deployments nodes and pods.

## What is Prometheus?

It is software application used for event monitoring and altering

It records real time metrics in a time series database, built using a HTTP pull model

### Why do we need Prometheus?

It basically contributes the DevOps system by monitoring cloud native applications and infrastructure and also by watching the over 100 of micro services. Now it’s really working well for recording any purely numeric time series data and another really important benefit oof Prometheus both machine centric monitoring and monitoring of highly dynamic service-oriented architecture

### Advantages and disadvantages of Prometheus

**Advantages:**

It is reliable

Standalone and self-containing meaning it doesn’t depend on n/w storage or other remote services

It is meant to work when other pods of the infrastructure broken, needs to setup extensive infrastructure to use it.

**Disadvantages:**

Prometheus difficult to scale. So, when you have hundreds of servers you might want to have multiple Prometheus servers, that some where aggregate of all this metrics data configuring that and scaling Prometheus in that way can actually be very different because of this characteristic so while using a single node is less compress.

### Where does Prometheus store the data?

### How It collects and then aggregates and how can other systems access this?

It stores the metrics data on disk so it includes a local on disk time series data base but also optionally integrates with remote storage system. And the data is stored in a custom time series format and because of that you cannot write Prometheus data directly into a relational database.

For ex: once you have collected the metrics of Prometheus also lets you query the metrics data on targets through its server API.

Using Prometheus query language

### Metric types in Prometheus?

The Prometheus client libraries offers four core metric types.

These are currently only differentiated in the client libraries and in the wire protocol.

1. Counter
2. Gauge
3. Histogram
4. Summary

# Recent issues:

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

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.