1. **Introduction :**

Good Morning sir.

My name is Jayant Bundile , I have done my graduation after that I am currently working at FISSOLUTIONS company from last 3.2 year as a DEVOPS engineer .

As a DEVOPS Engineer , I am using this tools on my organization .

I am using shell scripting language

For CI / CD tools I am using **JENKINS .**

For containerization tool I am using **DOCKER & DOCKER COMPOSE .**

For configuration management tool I am using **ANSIBLE .**

For source code management I am using **GIT & GITHUB**

For build tools I am using **MAVEN** .

**E)** all my infrastructure is available on cloud so I am using AWS cloud .

**F)** for that AWS cloud I am using this service ( EC2 , IAM , AMI , EBS , EFS , S3 , VPC , CW , CT , CF , RDS , Auto scaling , R53 , SNS ) .

1. **Roles & Responsibility :**
2. when I am login my laptop 1st we check the mail & notification .
3. Then I am checking the jira ticketing tool for pending task .
4. Daily attend the status calls & also attend the scrum meeting .
5. Client take the meeting on every week and they suggest they need some changes on there application .
6. For that changes we set the branching strategy and provide the separate branches for the developer .
7. After that they perform the testing and then merge the branch .
8. After merge the branch we check branch is merge or not properly , if any merge error is showing then

we resolve this merge conflict with the discussion of developer and push the branch into master .

1. Verifying all the servers & services are up & in running condition .
2. Taking backups of instance & restore if require .
3. Provided the access for developer & QA team .
4. Performing monitoring on daily basis .
5. create Ci/cd pipeline , run the pipeline , create the user , create environmental variable & configure email , install plugins , manage plugins , manage granular access , to create the job , deploy the code on multiple environment , build the code via **Jenkins** .
6. write the playbook , deploy the application on multiple servers at a time , update the application version at a time via **Ansible** .
7. Create the branches , check the commit , generate the poll request , merge the code , maintain the branches , provide the branch access on another team member , clone the project via **GIT** .
8. Create the docker file , create the image , create the docker container , deploy the docker container , deploy the application on docker container via **Docker** with the help of Jenkins .
9. Set up SNS topic , create the subscribers , add subscribers .
10. Create the Bucker , upload the object , manage the bucket policy .
11. Compile the code , upload the package into Remote repository , create the package , execute the test via **Maven** .
12. **What is SDLC :** ( software development life cycle ) it is the process of deploying the software **OR** it is the way to create And develop high quality software .

It consists of multiple phases (Planning , Requirement , Design , Development , Testing , Deployment , Operation & Maintance ) PRDDTDo&m .

1. **At which file I can change the port number** : vi /etc/httpd/conf/httpd.conf
2. **Which language is used to backend & frontend** : backend : .net , java , ruby , python

Frontend : html , css , javascript

1. **What is cloud computing :** It is the delivery of computing services—including servers, storage, databases, networking, software and intelligence over the Internet .
2. **VPC , why use VPC , Types of VPC : VPC** it is a virtual private cloud , it helps to create a virtual isolated environment in the same cloud . **why use VPC :** your private network keep safe and secure from public subnet . it is fully isolated . **Types of VPC :** there are 2 types of VPC . **A) Default VPC :** DNS is enabled & AWS is managed the setting . **B) Non Default VPC :** DNS is not enabled & User has managed the setting .
3. **Explain your project architecture :** I am using 3 tier architecture . this is my tier 1 architecture also this is my client under that client my browser is open this is called as presentation layer . this is my tier 2 architecture , this is also called as middle layer , under that tier 2 architecture 2 layer is available 1) application layer & 2) web layer . this is my server , under that server my DB is available , I am using MYSQL DB . I have to compile this from maven & deploy on tomcat . the web layer is communicate with DB layer . tier 1 architecture send the **Request** to tier 2 architecture . tier 2 architecture send the **Request** to Server . Server send the **Response** to tier 2 architecture , & tier 2 architecture send the **Response** totier architecture . I am installing below software as per requirement ( GIT , maven , Tomcat , Jenkins , Ansible , MYSQL )
4. **Types of EC2 Instance :** A) General Purpose Instance . B) Compute optimized . C) Memory Optimized

D) GPU graphics . E) Storage optimized . ( GCMGS ) .

**10) How many ways to launch the EC2 Instance :** there are 4 ways to launch EC2 instance . A) On demand

B) Spot . C) Reserved . D) Dedicated Host .

**11) Difference Between EBS , EFS & S3 : EBS** : A) it is block storage B) it works on both Windows & Linux C) it is

Bootable D) it is limited storage E) it is region specific .

**EFS** : A) it is file storage . B) it works on Linux C) it is not bootable . D) it is unlimited storage E) it is region

specific .

**S3 :** A)itis object storage B) it works on windows C) it is not bootable . D) unlimited storage E) Non region S

**12) What is AMI , How to create AMI , How to share AMI :** AMI it is a template of your virtual machine , with the

help of this AMI you can create a new instance which is exact copy of your running instance available on cloud

**How to create AMI :** Select instance -> Action -> Select image & template -> create image -> select no reboot

* Create AMI -> close .

**How to share AMI :** choose AMI-> action -> edit AMI -> permission -> private -> under shared account -> add

account id -> share AMI -> save .

**13) What is ELB , Types of ELB , Difference Between ALB & NLB** : A) it is a device that distribute network or

application traffic across a number of servers . B) it distributes incoming traffic into available server .

**How does it works :** 1) it checks which server traffic is less & forward the request on that server . 2) in case 1 of the server is down it will move the request on healthy server which is configured . 3) the LB will also do health check on regular basis time interval & forward request only to the healthy server . 4) if any server is faulty then it gets replaced with healthy server .

**Types Of LB :** 1) Application LB 2) Network LB 3) Gateway LB 4) Classic LB.

**Difference Between Application & Network LB : Application LB :** 1) It works on 7th layer of OSI model . 2) it supports http & https protocol .

**Network LB :** 1) it works on 4th layer of OSI model . 2) it supports UDP & TCP protocol .

**HOW to create LB :** 1st create the target group , the add all the instance into that target group , then create LB .

**14) What is Listener :** it is process for check connection request , it is configured with protocol & port for front end & back end connection , it is listen incoming request & forward .

**15) Suppose I have index.html file can this access through http & https port if yes how :** 1st we check under security group http ( 80 ) & https ( 443 ) port is open or not , if the port is not open then we open this port . and then hit the url again , also we try to open this url into another browser , if this url is open on another browser then we clear the history and again refresh the url if not working then we add ssl certificate into setting option .

**16) What is auto scaling , Types of auto scaling & components of auto scaling :** as per our requirement increasing or decreasing the capacity of the resources is called auto scaling .

**Types :** there are 2 types **A) Horizontal :** adding more machine into your pool resources .

**B) Vertical :** adding more power into your existing machine .

**Components : A)** auto scaling group **B)** auto scaling policy **C)** launch configuration **D)** Health check

**17) Which configuration have you done in your private VPC :** 1st we create public subnet , with the help of this we go to private subnet & at the time of installing packages we create the NAT gateway and attaché to the VPC .

**18) Difference between NAT & NACL : NAT :** it is translate private IP into public IP so you can access private IP easily

**NACL :** it can control incoming & outgoing traffic at subnet level .

**Where do we kept the NAT gateway :** inside the public subnet **.**

**How to attach NAT gateway :** 1st create NAT gateway , go to VPC create NAT gateway , then create Route tables -> select private routing -> edit routes -> add route -> select destination = open to world , select target = NAT gateway

-> save changes .

**19) How to access Private VPC from public :** 1st create NAT gateway , go to VPC create NAT gateway , then create Route tables in public routing -> go to route -> edit route -> select destination = open to world , select target = IGW

-> save changes . then again go to route table -> go to subnet association -> edit subnet association -> select public subnet -> save association .

**19 A ) What is Security Group :** it is a 1st layer of fire wall of your EC2 instance to control incoming & outgoing traffic .

**Diff NACL & SG : NACL : A)** it can operate as subnet level . **B)** it can support allow & deny traffic . **C)** it is state less .

**SG : A)** it can operate at instance level . **B)** it can support only allow traffic . **C)** it is state full

**20) Where we use peering connection :** to connect 2 different VPC .

**21) Difference Between CW & CF : CW : )** it can monitor your AWS services & the application you run on AWS in real time .

**CF :** it can accelerates delivery of your websites, API, video content . The major issues now a days with site access over the internet is latency, hence there is another layer of service added is called CDN . it is used to resolve latency issue .

**22) Types Of Monitoring :** 2 types of monitoring . **A) Basic :** it is free . **B)Detailed :** monitoring for every 1 minute .

**23) where VPC logs are stored :** under logs group of cloud watch .

**24) From CW which services have you monitored :** S3 , SNS , ALB , EBS , Ec2 .

**25) What is S3 , What is public Bucket & Private Bucket , Which 1 is secure : S3 :** it is object storage service . the data store as it is . this is unable to mount . it is not bootable . cannot install OS . it can be access as a http & https protocol over the internet .

**Public Bucket :** you can access the internet . it is the paid . **Private Bucket :** you can not access the internet . it is free .Private bucket is secure .

**26) What is object storage :** object storage is a computer data storage architecture that manages data as object .

**27) can we host static & dynamic website on S3 :** you can host only static website .

**28) what is S3 cross region replication :** Mostly Use for backup . 1)It is new feature that automatically replicates data across AWS Region . 2) Note : when u perform Cross Region Replication then it is compulsory to enable versioning , else u unable to do perform cross region replication . 3) every object uploaded to an S3 bucket is automatic replicate to a destination bucket in different region that you choose . 4) if we make changes in the destination bucket then the source bucket is not update . 5) if we delete object in source bucket , then the object is not deleted on destination bucket .

**29) What is S3 Versioning :** is incremental in nature . the delete marker becomes the current object version . you can still delete the object by deleting the delete marker .

**30) S3 Life Cycle :** It is a set of rules that are applied to the objects in specific S3 buckets.

**31) if bucket is public then how you make it private :** Open bucket -> go to permission -> edit public block access -> click check box -> save & conform . Now your bucket is private .

**32) What is static hosting in S3 :** static website supports fixed content HTML based websites that display the same information to all visitors .

**33) For website hosting which service you can use it : Chargeable Services :** ipage.com & godaddy.com .

**Free services :** Amazon S3 ( I am using this ) & forge .

**34) can you access S3 bucket from private subnet :** No , for that case we use VPC end point .

**35) What is VPC / S3 end point :** it is used to access S3 bucket from private subnet .

**36) what kind of work is done in EC2 :** in Ec2 most of the people to deploy the Jenkins servers for installing some application , we need to setup the project , setup the recruit packages on that server , so team will be take care of this .

**37) What is port forwarding in SSH :** it is used to forward port between local & remote machine using SSH protocol .

**38) How to give access to IAM user :** when I am create the user by Using IAM then 2 option is available on that time A) Programmatic access & B ) AWS management console access , so we provide the access as per there manager requirement .

**39) Internet Gateway :** it is a 2 way communication . **NAT gateway :** it is 1 way communication .

**40) What is edge location :** Cloud Front delivers your content through a worldwide network of data centers called edge locations.

**41) What is invalidation :** it allows you to remove an object from the Cloud Front cache before it expires.

**42) How to create invalidation :** select files -> right click of the files -> go to CF -> click invalidate -> select file -> close

**43) SNS & SQS it is push based or pull based :** SNS it is **Push** based & SQS it is **Pull** Based .

**44) Which protocol are SNS supported :** HTTP & HTTPS .

**45) What is Topic :** It is logical access point that communicate between publisher & subscriber .

**46) Diff Between Centralized & Distributed VCS : Centralized VCS** it can saved the version into Remote repository .

**Distributed VCS :** it can saved the version into Remote repository as well as local repository .

**47) Which branching strategic have you used in your Project :** I am using the master branch because master branch having the latest code , once the code is release , we create the branch & keep it a side .

**48) What is merge conflict & how will you resolve : Merge Conflict :** combine the changes of remote repository & local repository .

**how will you resolve :** open conflict file -> edit changes -> then add below files into staging area -> commit -> push .

**49) GIT Reverse :** it is a tool of undoing committed changes . **GIT Reset :** it is a tool of undoing uncommitted changes

**GIT : )** it is a s/w or a tool . this is the tool of version control system with the help of this we create repository & control the version .

**Git Hub :** it is remote repository .

**Working of GIT :** it allows user to track code changes & manage the project of simple git command .

**50) GIT Fetch :** Download The Changes From Remote To Local Repository .

**GIT Pull :** Download The Changes & Merge into your Current Branch .

**51) What is GIT Stash :** if you want to record the current state of the working directory but do not go back to a clean working directory **.**

**How to remove it command :** git stash clear .

**52) How to revert the commit from local repo & without commit ID how will you do :** git rebase -i sha1~1

**53)Why use GIT :** git it is the most popular distributed version control system . when you can copy your whole code from centralized version control system to local repository , then this whole code comes with the history , in case your code is loss from centralized version control system then the whole code is available from local repository of git so you can move easily .

**54) How to access S3 bucket when user is in another account :** select bucket -> edit permission -> add user by email / ID -> click more -> enter user ID -> provide permission -> click apply and changes .

**55) git pull & git push command :** git pull origin master&git push origin dev

**56) Explain Git command :**

**Create git folder :** git init **Red :** Workspace **Green :** Staging

**Check file is available on which area :** git status **( W , S & L ) To find commit ID :** git log **latest commit :** git log -1

**To set name:** git config --global user.name “ yuvraj **To set mail id :** git config --global user.email “ mailid “

**Check url origin add or not :** git remote -v **Push origin into RR :** git push origin master

**Pull the code :** git pull origin master **Add both files into staging area :** git add \* / git add fn

**Download the code from RR to EC2 I :** git clone pest clone url **Add the folder into git :** git add aws/aws.css

**Delete the untract file :** git clean -fd **See the branches :** git branches

**Switch M to D & D to M :** git checkout dev **/** master **Push dev to origin :** git push origin dev

**Push both files at a time :** git push --all **Delete the branch :** git branch -d BN

**To clear stash :** git stash clear **Check number of tags :** git tag **details of tag :** git show TN

**Delete the tags :** git tag -d TN **Delete repository :** rm -rf RN **Delete all repository :** rm -rf \*

**Push 6 commit to M branch :** git push origin commit ID:bra **Or** git cherry-pick < commit ID >

**57) Git rebase :** this is like a merge from 1 branch to another branch .

**58) Diff between public & private subnet : Public Subnet :** you can access the internet & it is used to web server **Private Subnet** : you can not access the internet & it is used to backend server & data base server .

**59) How to create the job in Jenkins :** go to Jenkins dashboard -> click new item -> enter job name -> select pipeline job -> click ok -> click add build -> click add build steps -> select execute shell -> pest script -> apply & save -> click build now -> console output .

**60) How to add users in Jenkins :** go to Jenkins dashboard -> manage Jenkins -> manage user -> create user .

**61) why use spot instance :** in some time developer code is not working on , on demand instance & reserved instance , on that cases we launch the spot instance & run the code , test the code , execute the code on that instance after that we create the AMI of that EC2 instance & create Snap shot of that instance and save this on S3 bucket & terminate the instance .

**62) which command is used to check linux version :** uname -a

**63) What is CI / CD : A)** 1st developer write there code properly & push this code into git hub server .

**B)** then we integrate this code with Jenkins, this integration is done in between git hub server & Jenkins server , **OR** we clone the project repository into Jenkins server . **C)** then Jenkins pull the code into git hub server & push it into ansible server . **D)** then we create the playbook in ansible & with the help of this playbook we run the command into docker server & automatically 1 tomcat server is created & under that tomcat server automatic run the developer code this whole process called as CI / CD Process .

**64) What is Jenkins :** itis self contained open source automation server , it is the power full application that allow CI & CD on there project . it is the heart of pipeline .

**65) Advantages Of Jenkins : A)** it has lots of plugins is available . **B)** we can use community plugins . **C)** you can create your own plugins . **D)** you can do what ever in plugins . **E)** we can attach the slaves to Jenkins master .

**66) what is the use of multi branch pipeline :** 1 repository is available & multiple branches is available .

**67) Explain Jenkins architecture : A)** manage the build with the help of master slave architecture **. B)** master slave units communicate with each other by using TCP/IP protocol . **C)** Jenkins master it is the primary server of Jenkins .

**D)** Jenkins master is capable of directly executing build jobs .

**68) Working of Jenkins master :** it can forward the request to slave .

**69) Working of Jenkins slave :** buildthe job dispatched by the master & read the request to the master .

**70) Types of Jobs : A)** Freestyle **B)** pipeline **C)** multi – configuration **D)** multi branch pipeline .

**71) What is pipeline jobs :** it is the collection of events that are interlinked with another one in a sequence .

Build -> Deploy -> Test -> Release .

**72) Steps in pipeline jobs :** steps is like a single command which performs a single action , when a step is success it moves to the next steps . when a step is fail then to execute correctly the pipeline will fail .

**73) Suppose I have a project in git hub repo in that there are many branches is available I wont build this project which job do you prefer :** I will go to the dev branch , because dev branch it is a main branch & in dev branch multiple sprints are available so we use this sprints and get the output & need to deploy it .

**74) What are the stages in Jenkins pipeline :** Build -> Deploy -> Test -> Release

**75) How will you install java in Jenkins :** by using below command “ yum install java -y “

**76) How to create multi branch pipeline :** go to Jenkins dashboard -> select new item -> select multi branch pipeline -> ok -> click add -> go to source -> select git hub -> under credential field select Jenkins & create credential with your git hub username & password -> go to execute -> pest script -> apply & save -> click build now -> console output

**77) Work flow of Jenkins : A)** 1st we install the Jenkins on the instance , this is my Jenkins server . **B)** then we have to install some plugins ( git , maven , selenium , artifactory , deploy to container ) into Jenkins . **C)** once developer commit the code into git hub server then Jenkins pull that code & send to the maven for build . **D)** once build is done then Jenkins put the code & send to the selenium for testing . **E)** once testing is done then jenkins put that code & send to artifactory for archiving purpose as per requirement . **F)** also we can deploy through Jenkins .

**78) What is declarative pipeline :** it is a separate language of Jenkins . which is write on groovy . which has provided there own syntax . with the help of this syntax you can create your own pipeline script as a code . **I am using declarative pipeline .**

**79) What is Scripted Pipeline :** it is the traditional way of writing the code .

**80) What is role based strategy in Jenkins :** it is used to add a new role – based mechanism to manage user permission , support features & create global roles .

**81) How I provide the server access to the other developer without using IAM :** go to Jenkins dashboard -> configure global security -> metric based security -> select user -> provide access -> apply & save .

**82) Why do we use actually use of master slave :** the basic idea of master slave is high availability . the client nodes one who take care of everything .

**83) What is Jenkins cluster :** it is hosting for continuous integration & continuous delivery .

**84) What are the ways to install the Jenkins : A)** 1st we install the yum repository . **B)** then download the yum key .

**C)** then install java . **D)** install Jenkins .

**85) What is the use of maven :** it is a build tool , build tools means compile the code & create the package of that code also execute the test . with the help of this we create we compile , we test & we package .

**86) What is RDS , Types of RDS , Features of RDS : What Is database :** collection of data ( data has been divided into 2 components Rows & Column )

**Why called as relational data base service :** content of rows & column is directly connected with multiple rows & column that’s why this is called as Relational Database Service .

**Types Of RDS : A)** Aurora **B)** Postgre SQL **C)** MYSQL **D)** Maria DB **E)** ORACLE **F)** SQL Server .

**Features Of RDS : A)** Read Replica **B)** Multi AZ deployment **C)** Monitoring **D)** Patching **E)** backup **F)** pay as u go

**87) How to connect data base into instance :** by using below command “ mysql -h db url -u admin -p “

**88) How can I improve security in RDS :** RDS server is launched in private subnet with NACL as it will provide more security .

**89) Port Number : MYSQL = 3306 , RDP = 3389 , Network File system = 2049 , http = 80 , https = 443 , ssh = 22 ,**

**ftp = 20/21 , DNS = 53 .**

**90) What is hard link :** it is for backup **.** it is only for files . **Soft Link :** it is for shortcut , it is for files & folder .

**91) How many Internet gateway connect to 1 VPC :** 1

**92) What is user , group & roles in IAM : User :** it is long term credential , it is directly interact with AWS .

**Group :** it is a collection of users . it can provide the permission for a collection of users .

**Roles :** Basically this is nothing but service to service access . We assign a role to aws resource to access another resource.

**93) Types of loops :** there are 6 types of loops . For , Infinity , Loop Control , Nested , Unity , While .

**94) What is shell scripting : A)** it is command line interpreter , it translate the user command and convert into the language which is understood by the kernel . all command are listed in a script in the order of execution

**B)** the command should be given start using the # sign . **C)** the .sh at the end of the file called as its shell script 4) at the start of shell script its compulsory to add shebang ( #! ) .

**95) How to get instance backup :** by using AWS backup service or Create the AMI .

**96) What is IAM Policy : A) AWS manage policy** : it is standalone policy . **B) Customer manage policy :** it is standalone identity based policy . **C) Inline Policy :** it is strict 1:1 relationship between the entity & the policy .

**97) Difference Between Role & Policy :** IAM roles define the set of permissions for making AWS service request whereas IAM policies define the permissions that you will require.

**98) What is tenancy in EC2 :** tenancy it is define how EC2 instance are distributed across physical hardware . 3 option is available . **A)** Default **B)** Multiple **C)** Dedicated

**99) Why use elastic IP on NAT :** you do not loss the connection between your domain & IP .

**100) What is traffic management policy : A) Simple routing Policy :** traffic is route to a single resource .

**B) Failover :** it is used to create Active/Passive set-up . **C) Weighted :** it help to split traffic between 2 regions .

**D) Latency :** the region that will give the fastest response time. **E) Geo Location :** Use when you want to route traffic based on the location of your users . **F) Geo-proximity :** Route traffic to specific instances based on the location .

**101) What is elastic IP : How to create , How to connect : Elastic IP :** it is a static IP , when you connect the elastic IP on any instance then after restart the instance your elastic IP is not change .

**How to create elastic IP :** go to EC2 -> go to network security -> elastic IP -> allocate elastic IP -> enter key & value -> allocate .

**How to connect elastic IP :** select elastic IP -> action -> associate elastic IP -> select instance ID -> click associate .

**102) SDLC & devops it is different or same :** agile is a SDLC that provides a methodology for delivering technical product & devops it is a method to delivery same technology product those who need it .

**103) How many elastic IP I can connect on my instance :** 1 elastic IP I can connect .

**104) How many default elastic IP is available on 1 region :** 5

**105) How many security group we can add in 1 instance :** 5 security group .

**106) each subnet how many AZ is available :** min 2 & maximum 6

**107) Transit Gateway :** it connects your Amazon Virtual Private Clouds (VPCs) and on-premises networks through a central hub.

**108) What is DNS , Types of supported DNS , Use of R53 : DNS :** it is Domain name system. It helps resolve Domain name to IP address . This works on port number 53, hence this DNS related service which is provided by Amazon is called as Route 53.

**Types of Supported DNS : A)** **A record :** this is address record which maps domain name to ipv4 address. This is 32 bit . **B)** **AAAA records :** this is for DNS to IPV6 mapping, its 128 bit. **C) CNAME :** canonical name records, means maps alias to a hostname. **D) NS record :** name server record. **E) SOA record** . **F) MX record** : for mail exchange defines where to deliver email .

**Use of R53 :** Register New domain , manage domain , transfer existing domain .

**109) How many listener is available in ALB :** there are total 11 listener is available . **Listener configuration** , **listener rules** , **Rule action type** , **Rule condition type** , **Create a http listener** , **create an https listener** , **update listener rules** , **update an https listener** , **authenticate user** , **x-forward headers** , **delete a listener** .

**110) How many types of permission is available in linux : chmod777 , chmod754 , chmod755 .**

**111) which command is used to list out the number of lines in particular folder :** wc -l filename

**112) which command is used to check the word :** /

**113) CPU information :** cat /proc/cpuinfo **current cpu utilization :** TOP p

**114) Memory Information :** cat / proc/meminfo **current memory utilization :** TOP m

**115) which command is used to check your apache is running or not :** systemctl status httpd

**116) I wont to check which all ports are running in my OS :** netstat , lsof , nmap

**117) How to check disk storage in your machine :** fdisk , fdisk -l

**118) you have 100 machine & went to see you release information of that all machine :** ansible all --lists-hosts

**119) I wont to check particular group system information :** ansible dev --lists-hosts

**120) suppose I have 100 machine & I wont to launch httpd on that all 100 machine :**

ansible all -ba “yum install httpd -y “

**121) How can you set all 100 machine IP in your linux machine :** on that case we activate the host file , host file it is also called as inventory file . go to that file -> create 1 section on that file & enter all machine private IP address .

**Note :** host file is available in **ansible.cfg** folder

**122) which command is used to found all IP address in your network :** ifconfig

**123) how to change the permission :** by using chmod command

**124) use of grep command :** to show the pattern

**125) What is private key & public key :** it is a set of security credential that you use to prove your identity when connecting to an instance . **Public Key :** present to AWS , **Private Key :** present to user .

**126) I wont to schedule some task to be continuously running on a particular time on a day so how can I achieve that :** we use build periodically option on Jenkins & CRON tab option for shell scripting .

\*\*\*\*\* **1st \*** Minutes ( 0 – 59) , **2nd \*** Hour ( 0 – 23 ) , **3rd \*** day of the month ( 1 – 31 ) , **4th \*** Month ( 1 – 12 ) ,

**5th \*** week ( 0 – 6 ) 0 = Sunday & 6 = Saturday .

1. **Every 2 min daily :** H/2 \*\*\*\* **B) Every 15 min daily :** H/15 \*\*\*\* **C) Every Hour \*1\*\*\***

**D) if job run every 8:30 AM Monday to Friday :** 30 8 \*\* 1-5

**E) if job run every 8:30 AM :** 30 8 \*\*\* **F) if job run 7 AM Monday to Friday :** 0 7 \*\* 1-5

**G) if job run 7 PM Monday to Friday :** 0 19 \*\* 1-5

**127) what is the use of export command :** to create environmental variable

**128) how to generate ssh key :** by using ssh-keygen command

**129) which command is used to check ssh key is available or not :** ls -la ~/.ssh/

**130) which command is used to copy and access the another server** : ssh-copy-id ec2-user@serverBIP

**131) what is the default location of .ssh file :** /root

**132) what is important 2 files to actually copy the resources :** non host & authorized key

**133) which command is used to check file & folder :** du -sh **all file & folder :** du -sh\*

**134) what is the use of sed command :** to replace old value with new value sed -l ‘s/old/new/g’ filename

**135) use of awk command :** pattern scanning & processing ,

**136) check current CPU / Memory utilization : TOP P** for CPU utilization & **TOP M** for Memory utilization .

**137) Which command is use to check ping connectivity of target system :** ping , traceert & traceout

**138) if CPU utilization is crosses 80 % then what will happen :** we can configure Cloud Trail service so we get the logs & also we create 1 SNS topic if your CPU utilization is crosses 80% so that will get the alert .

**139) Difference between In node & Process ID : In node :** it is the unique name given by the operation system to each file . **Process ID :** it is the unique ID given to each process .

**140) What is sudo user :** run the command as root user .

**141) how to connect 2 EC2 instance without password . you can login to 1 how to access another : A)** 1st connect the 1 instance . **B)** then generate the ssh key by using below command “ ssh-keygen “ press 4 time enter .

**C)** copy the public key from server A to Server B by using below command

“ssh-copy-id ec2-user@serverBIP “ enter yes -> asking password , enter the Server B password 1st & last time

Server B is properly copy with server A **D)** connect server B into server Aby using below command

“ ssh ec2-user serverBIP “ Now you are connected properly .

**142) Physical Memory :** it is the RAM of linux machine .

**143) Swap memory :** it is the space of hard disk which is used when RAM is full .

**144) What is linux partition :** splits your hard drive into multiple section .

**145) Which web server you can use : in Redhat :** httpd , **in Ubuntu :** apache

**146) SQS :** it is simple queue service . it is totally related to coding . it is push based service . and it can be used only developer .

**147) if you loss your private key then how you login again :** 1st we create the AMI of my EC2 instance , then we create the Snap shot of my EBS volume , then launch new instance with the help of my created AMI and connect snap shot of that newly launch instance , create new private key and login .

**148) what is front end & back end architecture in AWS : Front end architecture :** it includes components like local network , web browsers & web application .

**Back end architecture :** it comprises hard ware storage & they are located on remote server .

**149) How to clone a single branch only :** by using below command  **“** git clone origin branch name “

**150) How I provide read write access of read replica :** select read replica of data base -> action -> promote -> continue .

**151) How to generate access key and secret key :** go to aws console -> go to your profile -> click security credential -> click access & secret key -> create a new access key -> download key file -> close .

**152) How to set the Cloud watch alarm : A)** before creating the alarm make sure your EC2 & Cloud watch region both are same because CW is a region specific service .

**B)** when you set the alarm in EC2 then at the time of creating EC2 instance enable the monitoring option in step 3 .

**C)** after launch EC2 instance copy instance ID , why because we set the filter on that instance ID .

**D)** then go to Cloud watch service -> go to all alarm -> create alarm -> select metric -> select EC2 -> select per instance metric -> select CPU utilization -> select metric -> static = average , period = 1 min , threshold type = static ,

CPU utilization = greater than 80 -> click next , notification = in alarm , create a new topic , email ID = enter email ID , create topic -> click SNS console -> conform the subscription .

**E)** again go to cloud watch -> select SNS topic -> select existing SNS topic -> next -> enter name & description -> next

Create a new status is insufficient data . **F)** go to dashboard -> create dashboard -> enter name -> create dashboard .

**G)** go to alarm -> select alarm -> action -> add to dash board -> check cloud watch -> now dash board is showing .

**153) How can you mount EFS in your EC2 :** create 2 security group & enable Network file system port . then create EFS . then launch EC2 instance . connect instance . create 1 folder mnt . go that folder cd /mnt

Go to EFS -> click attach copy NFS client common & as it is pest on instance & replace last efs into mnt .

**154) Explain EBS volume , and which 1 is faster :** it is elastic block storage . it is used with EC2 service .

There are 5 types volume . **A)** general purpose SSD **B)** provisioned IOPS **C)** throughput optimized **D)**cold HDD **E)** magnetic . Provisioned IOPS is faster as compared to all this .

**155) How will you encrypt the EBS volume with EC2 :** select unencrypted volume -> action -> create snap shot . select newly created snap shot -> action copy -> click encryption of check box -> select default -> click copy snap shot

**156) How to add extra EBS volume but not connect new EBS** select volume -> action -> modify -> add size -> modify

**157) I need to push 1 file or I need to change in 1 repository then how we push that :** git push origin master

**158) which branching strategy have you work :** we have to use master , dev , hotfix , release , feature .

**159) How to change CIDR range if VPC is created :** add a secondary IPv4 CIDR block to your VPC & create a new VPC with a different IPv4 CIDR block & migrate your resources .

**160) what type of instance you worked on EC2 : t2.xlarge ( 4 CPU & 16 RAM ) , t2.2xlarge ( 8 CPU & 32 RAM )**

**161) what type of instance you used on RDS :** db.m6.2xlarge ( 8 CPU & 32 RAM )

**162) How to create Data base :** 1st create the subnet group then create data base . if you select data backup option then automated backup is also created .

**163) Difference between multi AZ & Read replica : Multi AZ :** in multi AZ there are 2 subnet group is available . you have full read & write access of primary subnet group , & you don’t have read & write access of secondary subnet group , if in case primary is fail then only you can access secondary .

**Read Replica :** secondary subnet group is exact replica of primary subnet group . you have full read & write access of primary & only read access of read replica not write access .

**164) difference between t2.micro & t3.large : t2.micro :** 1 CPU & 1 RAM , **t3.large :** 2 CPU & 8 RAM

**165) which plugins have you install in Jenkins :** docker plugins , docker common plugins , docker pipeline , amazon EC2 , cloud bees , Branch API , git , maven , deploy to container .

**166) what is $0 in shell script :** file name of current script . **S1 :** 1st command line argument passed to shell script .

**167) at which file I can change the port number of tomcat :** server.xml

**168) what are the 3 main files in apache tomcat installation :** server.xml , catlina.out , startup.sh

**169) what are the continuous deployment part in Jenkins : A)** after done CI part we deploy this on tomcat server . 1st install the “ **deploy to container** “ plugins . **B)** then set the credential -> create credential -> add credential .

**C)** then go to Jenkins dashboard -> click your pipeline -> configure -> go to post build action -> select deploy war/ear to container . then select tomcat credential , enter tomcat url -> apply & save .

**D)** then click build now it should able to generate the war file once the war file is generated it should get deploy into your tomcat server . war file is deploy into web apps folder .

**170) How EC2 is integrate with Jenkins :** by using master – slave architecture .

**171) what is docker compose :** it is a tool for defining & running multi – container docker application .

**172) docker networking :** it is a passage through which all the isolated container communication .

**There are 5 types of docker networking drivers is available :** bridge , none , host , macvlan , overlay

**173) difference between VM & Container : VM :**  it is fully virtual machine . & **container :**  it is not a virtual machine but it is like a virtual machine because resources which you need who run the application .

**174) Difference between container & process :** container it is also a process but for more isolation then you can run the whole process .

**175) what is virtualization & containerization : virtualization :** it is work on software which is called as hypervisor& it is a technique to import guest OS into top of the host OS .

**Containerization :** it is the form of virtualization where I can run the application on isolated user space .

**176) Docker command :**

**Install Docker :** yum install docker -y **create container :** docker run -it testimage:1.0 bash

**start docker service :** service docker start **&** systemctl start docker **docker version :** docker --version

**pull container / launch container :** docker pull CN **which os is available in docker :** execute uname -a **check which container is running :** docker ps **just now :** dockerps -a

**start // stop container :** docker container start / stop CID **forcefully delete the image :** docker rmi -f CN

**stop all container :** docker container stop $(docker ps –q) **for shell access :** docker run -it CN:tag bash

**image is download or not :** docker images **image list :** docker images ls

**check how many process is running :** ps -ef **container logs :** docker logs -f CID **search image :** docker search

**container is running so how can I go to the C :** docker exec -it CID bash

**delete all the container :** docker system prune -a -f **create image :** docker run -t testimage:1.0 .

**which command is used to bind the port :** docker run -itdp 80:80 httpd

**kill C :** docker container kill CID **remove C :** docker container rm CID **Docker information :** docker info

**check running container memory utilization :** docker stats CID

**Which cmd is used to create volume :** docker volume create-my-vol

**177) Write a simple docker file :** it is a normal text file , under that file multiple components are available on that file

**Components of Docker : A) FROM :** to download the image into docker hub . **B) RUN :** execute the command

**C) MAINTAINER :** owner of the file . **D) COPY :** copy any file from local machine to docker container .

**E) ADD :** zip , unzip , extract the files . **F) EXPOSE :** to expose the port

**G) WORKDIRECTORY :** use to work on any container . **H) CMD :** if you run any command then use CMD .

**I) ENTRY POINT :** entry point & cmd both are working same but the priority of entry point is high .

**J) ENV :** to set environmental variable **K) ARG :** to pass the argument .

Docker fiel it is used for build .

**Syntax of docker file :** 1st create a fiel vi Dockerfile

FROM ubuntu:18.04

WORKDIR /tmp

RUN echo “hello all “ > my indexpage.html

ENV name yuvraj

COPY sample.war /data

ADD yuvraj.tar.gz /tmp & save the file :x this is my docker file

We create the **docker image** with the help of this **docker file** of below command “ docker run -t testimage:1.o . “

We create **docker container** with the help of **docker image** of below command “ docker run -it testimage:1.0 bash “

**I** = interactive & **t** = terminal

**178) which command is used stop the container :** docker container stop CID

**179) which command is used to stop all the container :** docker container stop $(docker ps -q)

**180) which command is used to found which os is available on Docker :** execute uname -a

**181) how many ways to create the docker image :** there are 3 ways to create the docker image .

**A)** interactively building images **B)** using docker file **C)** importing for a tar ball .

But I am create image with the help of docker file . “ docker run -t testimage:1.o . “

**182) Difference between ADD & Copy : ADD :** when you zip , unzip or extract the file then I am use ADD .

**COPY :** when you can copy any file from local machine to docker container then use COPY .

**183) TAG & COPY : TAG :** tag it is the naming convention of particular section of code .

**COPY :** when you can copy any file from local machine to docker container then use COPY .

**184) ENTRY POINT & CMD :** both are same but the priority of empty point is high as compared to CMD .

**185) ARG & ENV : ARG :** when you pass any argument then use ARG .

**ENV :** when you add any environmental variable then use ENV .

**186) How to create the docker volume :** with the help of below command “ docker volume create-my-vol “

**187) What is multi stage docker build :** it allows you to use multiple images & there you have only 1 docker file but can define multiple images .

**188) I have 1 container & I need to go inside the container : docker exec -it CID bash**

**189) Difference between continuous delivery & continuous deployment : Continuous delivery :**it is the automation of steps to safely get changes into production. & it focuses on the release and release strategy.

**Continuous Deployment :** it focuses on the actual deployment .

**190) Difference between WAR / JAR & EAR : WAR :** it only requires a java enterprise edition web profile complaint application server to run WAR file .

**JAR :** it only requires a java installation .

**EAR :** it requires a fully java platform , enterprise edition or Jakarta enterprise edition complaint application server , such as web sphere or JBOSS to run EAR file .

**191) how to check how many process / container is running :** by using this command ps -ef

**192) which command is used to check the running container image logs :** by using this command docker logs -f CID

**193) What is D Dos attack :** DDoS Attack means "Distributed Denial-of-Service (DDoS) Attack" and it is a cybercrime in which the attacker floods a server with internet traffic to prevent users from accessing connected online services and sites.

**194) What is dangling images in docker :** Dangling images are layers that have no relationship to any tagged images. They no longer serve a purpose and consume disk space.

**195) Why use Tomcat :** to deploy war file .

**196) What are the continuous Integration part in Jenkins : A)** 1st create the CI job , we create the freestyle job -> under source code management select git -> enter git url -> select credential -> select branch -> in build option verify **pom.xml** is available & in goal option = clean install package -> Apply & save .

**B)** now your pipeline is start & execution is start .

**C)** if your job execution is showing **blue** it means your job is executed successfully . &

If your job execution is showing **Red** it means your job is fail , then modify the job & save this job .

**D)** again execute your job & complete the execution then it is testing the test cases & pop up is coming success .

**E)** go to build & check both jar & war file is created . this whole process called as CI process .

**197) What is ansible :** it is open source IT automation engine provided by Red hate . it is configuration management tool . it is push based mechanism tool . it is automation tool , it is GUI but I am using CLI .

**198) Advantages of ansible :** easy configuration , simple architecture , agent less , it is free , it is light weight , secure

**199) path of configuration file in ansible :** etc /ansible

**200) use of ansible : A)** to install the application on all the server at a time .

**B)** create the folder & file at a time on all the server . **C)** maintain the server & node properly

**201) what is inventory file :** inventory file it is also called as host file . under host file you can add other node private IP address .

**202) How ansible work :** it is work on SSH connection . it does not require to open any network port . only need to do enable password less authentication .

**203) what is exact use of node :** it is used for managing the remote nodes which is connected by the ansible server .

**204) what is the use of ansible module :** a module it is a reusable , stand alone script that ansible run either locally or remotely .

**205) what is the use of loop in playbook :** it can use repeat any task in multiple times . create the multiple users using user module , installing multiple packages or changing the permission on several files using the module .

**206) Explain playbook which you created : Playbook :** playbook are normally managed & maintain in a version control system like git .

**Use of playbook :** if you can use multiple module , perform multiple task at a time then you can playbook .

Playbook format is yaml .

Enter playbook file name , vi playbook.yaml

--- #my 1st playbook

-hosts:dev

user:ansible

become:yes

connection:ssh

gather-facts:yes & save :x

For run the playbook use below command “ ansible-playbook playbook.yaml

**207) What is idem potency :** it is the proper syntax .

**208) What is the role in ansible :** role it can isolates these components like ( files , templates , tasks , handlers ) its easy to reuse & share with other people .

**209) in ansible what is -a :** argument

**210) what is ansible collection :** Ansible Collections are a distribution format for Red Hat Ansible Automation Platform content that can include playbooks, roles, modules, and plugins around specific topic areas. Ansible Collections represent the new standard of distributing, maintaining, and consuming automation.

**211) Ansible galaxy :** it is used to create the role .

**212) Use of role :** you have to define the different task , handlers , variables & loops under role .

**213) Ansible tower :** it is graphical user interface . but I am using CLI .

**214) Ansible cluster :** it is sharing load between hosts . Each instance should be able to act as an entry point for UI and API access .

**215) How to install ansible :**before install ansible 1st we install the epel repository ( rpm file ) & then install the ansible server .

yum install epel-release -y

then update the rpm “ yum update -y “

then install ansible “ yum install ansible -y “

**216) Ansible handlers :** it is same As task but trigger is available for event & notifier .

**217) What is static & dynamic inventory : Static inventory :** it is a plain text file containing a list of managed hosts or remote nodes whose numbers and IP addresses remain fairly constant .

**Dynamic Inventory :** it Is host file keeps changing as you add new hosts or decommission old ones .

**218) How many types to run the command in ansible :** there are 3 types to run the command in ansible .

**A) adhoc command :** this is only simple linux command .

**B) Modules :** some command is already available & you can use directly . **C) Playbook :** it is a script .

**219) How to maintain inventory file :** we have go to the host file , ( vi etc/ansible/hosts )

Some files are committed & some files are un committed so as per requirement we have committed & un committed the files .

**220) Use of inventory file :** create the groups & update the configuration directly or separately . also we create the

Section under that file and add multiple node private IP address on that file .

**221) what is the command to copy the file from ansible server to destination :** mv ansible-node1ansible-devnode1

**222) which command is used to copy the bucket from 1 region to another by using CLI :**

aws s3 cp s3://bucketname .--recursive

**223) what is web traffic :** incoming request & outgoing response this whole life cycle is called as web traffic .

**224) How to calculate load :** by using web traffic .

**225) if my instance memory is full then what will happen :** 1st we check the memory utilization & current memory utilization , if the memory utilization is high then we stop the instance for changing the instance type .

Before stop the instance we mark mail to all the engineer which has use this instance and mention the message .

“ the memory is full so we need to change the instance type “ and then change the instance type .

Select instance -> click action -> instance type -> change instance type -> close . and restart the instance .

**226) How to configure cross account :** Make sure you have the account ID for the Dev account. Sign in to the Prod account as a user with administrator privileges. In the IAM console, create a new role and name it Cross Account Sign in . Choose the wizard option for creating cross-account access between accounts that you own.

**227) if my CIDR range is 192.168.0.1/16 then what is my next IP :** 192 it is the class C IP address , so CIDR is not 16 .

CIDR is 24 & my next IP is 192.168.0.2/24

**228) EC2 session out troubleshooting :** check ssh port , 2/2 check is done or not , check the key permission , check usable key is correct or not , check user is correct or not .

**229) RDS session time out troubleshooting : A)** we check the RDS url is showing or not on console , if data base url is not showing then we wait for next 2 min and check . **B)** also we check in security group mysql port number 3306 is open or not and connect or not on same security group .

**230) which command is used to restart the RDS in linux :** reboot-db-instance

**231) which command is used to find how many file system are mounted in linux :** findmnt

**232) I wont to push only ( eg = 6 ) commit to master branch :** git cherry-pick < commit ID >

Git push origin commit id : branch

**233) what are the various types of build triggers in the Jenkins jobs : A)** windows batch command **B)** shell execute

**C)** ant **D)** gradle script **E)** top level maven **F)** run with timeout

**234) benefits of containerization :** portability , efficiency , faster delivery , improved security , faster application startup , flexibility .

**235) can you have multiple FROM in 1 docker file :** No

**236) which command is used to change the server name :** hostnamectl set-hostname Jenkins bash

**237) which command is used to extract the file :** tar-xvzf --- tar.gz

**238) which command is used to move the tar file into tmp folder :** mv\*.tar.gz /tmp

**239) my java is not run on any where so how can I do :** we add java – home environmental variable & save the file .

**240) How to create the user by using IAM :** go to IAM -> click user -> add user -> enter name -> select programmatic access or aws management console access -> select auto generated password -> click password reset -> permission -> tag -> review -> create user -> download file -> close .

**This user is create with the help of root user .**

Open IAM link on another browser -> enter account ID -> enter user name -> enter password -> change password .

**241) How to create the group :** go to IAM -> create group -> enter group name -> select user & assign policy -> create group .

**242) How to create the role :** go to IAM -> role -> Create role -> select trusted entity type -> select use case -> select permission policy -> enter role name -> create role .

**243) which command is used to check ansible master to ansible client communication is enable or not :**

ansible all -m ping

**244) which command is used to restart the Jenkins :** go to browser enter “ Jenkins server IP://8080/restart “

**245) how to deploy war file into tomcat : A)** 1st create the CI job then deploy this job into tomcat server by using Jenkins . **B)** install the **deploy to container** plugins in Jenkins then set the tomcat credential .

**C)** then go Jenkins dashboard -> select your job -> configure -> go to post build action -> select deploy war/ear to container .

**D)** then select path of war/ear file -> select container -> then select tomcat credential -> enter tomcat url -> apply & save .

**246) How to connect git hub server into 3rd party application :** by using personal access token

**247) How to create git repository :** login account -> go to profile -> your repository -> create new -> enter name -> create .

**248) How to integrate Jenkins with git hub server :** we install **git client** plugins

**249) Which branch is create pull request :** multi branch pipeline .

**250) if the branch is deleted into source code so how that is done :** in git hub setting we can enable that particular option to delete the source branch when merge is done .

**251) How to compile the source code :** by using **mvn compile** command

**252) How to create the package :** by using **mvn package** command

**253) How to create the branch :** go to connected instance -> create git folder -> create files -> add both files into staging area -> commit the files -> enter the command ( **git branch** )

**254) How to deploy Jenkins war files :** go to google -> search install Jenkins -> select war files select latest Jenkins war file -> right click generic java package -> copy link address -> go to instance -> **wget pest url** -> enter command

“ mv Jenkins.war webapps “ go to webapps folder & run startup.sh file .

**255) Where is the store Jenkins master file :** config.xml

**256) Where is the store tomcat master file :** server.xml

**257) What is the home directory of Jenkins :** /root/.jenkinsa directory where your Jenkins is install .

**258) How to get the Jenkins backup :** by using Jenkins folder , zip the Jenkins file and store in s3 , if Jenkins is corrupt then unzip the zip file from same location & start the tomcat server in Jenkins file .

**259) what is the use of pom.xml file :** to run the mvn install command .

**260) Which 1 file is important for maven installation :** setting.xml

**261) what is the use of setting.xml file :** set user name , set password , set custom repository , set repository path , proxy setting all this is done under setting.xml file .

**262) Life cycle of maven :** validate , compile , test , package , integration test , verify , install & deploy