Describe about yourself.

Say about ur education and experience

Client: light house,

Client: synicron

Roles and responses:

Say those what are in your cv.

**CI/CD FLOW:**

GIT/BITBUCKET

1) We use Atlassion bitbucket as a scm(source code management) which hosted on cloud.

2) Each repository consists a jenkins file to achive ci/cd.

Q)HOW JENKINS FETCH LATEST COMMITS

3)we enabled webhook(A webhook is an HTTP callback, an HTTP POST that occurs when something happens through a simple event-notification via HTTP POST.) in our bitbucket with jenkins url for every repository. Coming to our jenkins,for every job we saved the configuration with the option like.... trigger build when ever a change is pushed to bitbucket, if the repo url present in the jenkins job configuration then it will automatically trigger a build.

Q) JENKINS ARCHITECTURE

4) Here we are using docker dynamic slaves (for better resourse utilization)concept.for every build it will launch new docker slave in kubernetes .

Q) WHERE U SETUP CONFIGURATION

5) We define slave images in jenkins cofig under kubernetes section.

6) In kubernetes pod template we can mention which docker image can act as slave we installed all our packages like java maven npm gradle python docker kubectl in that docker image. if any issue occurs in slave image we can update our docker image.

**In our pipeline based on our requirements we included six stages.it may change depending on requirements.**

1.Src code checkout

2: genkins file validation ,

3: maven build

4 publish to artefact:(nexus)

5:deploy to kubernetes

6: notifications.

**PIPELINE STAGES:**

1. IDENTIFY PROJECT TYPE( EITHER MAVEN OR NODEJS)
2. CREATE APPROPRATE PROJECT SLAVE(DYNAMIC SLAVE)
3. CLONE THE CODE INTO JENKINS SLAVE
4. VALIDATE JENKINS FILE
5. BUILD CODE IF CODE IS OF MAVEN ,PIPELINE WILL BE USING MAVEN BUILD OR IF CODE IS NODEJS USE NPM INSTALL
6. PUBLISH BUILD IMAGES TO ARTIFACT(CENTRAL REPOSITARY) WITH APPROPRATE TAGGING) EG:IF IT IS FROM DEV BRANCH ARTIFACT WILL BE TAGGED AS APPLICATIONNAME.APPLICATIONID.ENVIRONMENT NAME.BUILD NUMBER
7. PUBLISH BUILD IMAGES TO DOCKER HUB WITH APPROPRATE TAGGING
8. PULL IMAGE FROM DOCKER REGISTERY EG: OUR PRIVATE REGISTERY
9. DEPLOY TO KUBERNETIS
10. NOTIFICATIONS

Q) Where you will maintain dependencies of java,

A) In pom.xml file

Q) Where you will maintain dependencies of node.

A) Package.json

Q) How will you debug when a build is failed.

A)

1) first I will see the Jenkins console log and identify the issue.

2) if its failing due to dependencies version or package not available I will update the requird package in artifactory by taking the package justification from the user.

3) if its failing due to test or docker steps I will inform developers to update the changes.

Q) How you tagged your application builds

A) We have configured our pipeline in such a way if code is from develop it needs to be tagged as Application name.application id. env.buildnumber.(ENV DEV OR QA)

Q) From where you will be getting application id.

1.We are maintaining an excel in wiki to maintain all application ids.

Jenkins ci/cd process.

4. whenever a change is pushed to bitbucket repo it will trigger build in jenkins.

5. initially pipeline will start with checking kubernetes pod template then it will launch a docker slave container in kubernetes then it will start executing remaining process

6. in first stage it will fetch code from bitbucket repo based on brach declaration in our pipeline.

7.then it will start build if

It is java app if it is node app

We are using mvn clean install to build we are using npm install to build

To download its dependencies from our for authorization and to download dependencies from private nexus repository we are defining settings.xml local nexus repository we are using .npmrc file

And settings-security.xml in settings.xml file we are in that we are declaring nexus repo url for depencies

Nexus repo to download dependencies in settings.xml

And master encoded password in settings-security.xml

10. then after build got success it will start building docker image with these artifacts and pushed to our docker private registry.

12 we are publishing artifacts to nexus repo if it is maven we are using small syntax in pipeline to publish or you can use mvn build (need to add distribution managemnet in pom.xml file) if it is node app we are using npm publish.

11.then finally docker image will deploy in our kubernetes environment we are using shell commands in our pipeline to deploy.

GIT FLOW STRATAGY

1)MASTER ----PRODUCTION (No data initially)

2)DEV -------DEV ENVIRONMENT

3)FEATURE ------INDIVIDUAL DEVELOPMENT

4)RELEASE ----QA

5)BUGFIX

6)HOTFIX

BRANCHING STRATGY DIAGRAM

**UNIVERSAL BRANCHING STRATAGY**

* DEFAULT MASTER—NO CODE IN IT
* EVERY DEVLOPER WILL BE DEVELOPING THEIR OWN LOGIC IN THEIR OWN FEATURE BRANCH AND RAISE A PULL REQST TO DEVELOP BRANCH
* CODE REVIEWER WILL APPROVE THE PULL REQUEST AND MERGE CODE IN DEV BRANCH

GIT COMMANDS

* git clone, git config, git add, git status, git commit, git push, git pull, git branch, git checkout, and git merge

GITFLOW STRATAGY COMMANDS.

1) Add your Git username and set your email

A) git config --global user.name "YOUR\_USERNAME"

git config --global user.email "your\_email\_address@example.com"

2.) Initialize a local directory for Git

A) git init

3. )Clone a repository

A) git clone <repository clone url>

4) once you clone your repository in local move to cloned directory and then follow this steps.

1. By default once you use git clone it will create master branch in your local and you will be using master branch

2. To check on which branch you are into git branch

3) To create a new branch into local

git checkout -b <newbranch name>

4. To change from one branch to another branch in your local

git checkout <desired branch name>

5. To add the changes to new branch/repository

Git add .

6. To commit/save changes

git commit -m "comments"

7. To push the code from local branch to remote branch

Git push origin <Desired remote branch name >

8. To pull the changes to remote repository

Git pull <remote branch name>

9. To View your remote repositories

git remote -v

10.View the changes you’ve made

git status

11. To view the differences between your local, unstaged changes and the repository versions that you cloned or pulled, type

git diff

12. To delete all local changes in the repository that have not been added to the staging area, and leave unstaged files/folders, type:

git checkout .

**Q)Process need to followed by developers**

1. By default you will be in master.

2. create a develop branch both in you local and remote

3. create a feature branch from your develop to do your own changes and push to your feature branch

4. once all the functionality is checked in feature branch then create a pull request from feature branch to develop branch.

4. once all the functionality is checked in devenvironment then create a release branch from develop environment by raising a pull request.

5. if any bug is idenfified create a bugfix branch from release/qa branches and fix the bug once the bug is fixed move your code from bugfix branch to release branch

6. once all the functionality is checked in release/testing then move code from release branch to master branch.

**KUBERNETIS :**

**USED FOR ORCHESTRISATION ALL CONTAINER INFO LIKE APP NAME,VOLUME DETAILS,RESOURCE SETAILS,PORT DETAILS SERVICE NAME DETAILS WILL BE SPECIFIED IN DEPLOY CONFIG FILE LIKE JSAM FILE/AMFL FILE.**

**NAMESPACE:TO KEEP ALL RELATED APPS TOGETHER IN KUBERNETS IS CALLED AS NAME SPACE**

**PROJECT IN OPENSHIFT**

**Q) HOW MANY NODES IN UR CLUSTER**

**A**

**OPENSHIFT**

debugging Notes.

2) delete docker images those are not in use.

A) docker rmi $(docker images | grep -v -e | awk {'print $3'})

3) To login in to Openshift

A) oc login <environment name>

Ex: oc login .url:8443

4) To Move to particular project

A) oc project <project Name>

Ex: oc project holdings-dev

5) To list all the pods in the project

A) oc get pod

6) To Delete the pods

A) oc delete pod name

7) To list all the objects related to particular application except PV(persistant Volume)

A) oc get all -l app=<App name>

oc get all -l app= holdings

8) To delete all the objects related to particular application except PV(persistant Volume)

A) oc get all -l app=<App namee

Ex: oc get all -l app= holdings

9) To force delete the pods when they are struck in terminating state

A) oc delete pods <pod name> --grace-period=0 --force

10) To list all the pods according to status

A) oc get pods -a | grep <status Name>

11) To delete all the pods according to status name

A) oc get pods -a | grep <status Name> | cut -d' ' -f 1 | xargs oc delete pod

12) To list all the node that are using OCP

A) oc get nodes

13) How to describe the node and see the information related to node

A) oc describe node .<node name>

Ex:oc describe node ip-172-30-147-234.saccap.int

14) How list nodes belong to a group/type

A) oc get nodes -l type=<Type Name>

oc get nodes -l type=ux

15) list the persistent volumes using nfs

A) oc get pv | grep nfs

16) How to describe PV

A) oc describe pv <pvname>(by describing the pv you can find the volume id that is mounted in AWs

Ex: oc describe pv pvc-ffaf94bc-e012-11e7-beb0-124556a47eb

Name: pvc-ffaf94bc-e012-11e7-beb0-124556a47eb8

Labels: failure-domain.beta.kubernetes.io/region=us-east-1

failure-domain.beta.kubernetes.io/zone=us-east-1b

**PERSISTANT VOLUMES (TAKEN FROM CLOUD)**

1. GENERALLY WE KEEP IN NODE IF NODE FAILS SO WE KEEP IN CLOUD.

**Q) HOW TO EXPAND EBASE VOLUMES**

A) THROUGH UI(ANSIBLE)

1)FIRST IDENTIFY VOLUME ID

2)SCALE DOWN THE POD

3) INCREASE VOLUME CAPACITY FROM AWS CONSOLE OR CLI

4) EDIT PERSISTANT VOLUME FILE IN KUBERNETIS

5) CHANGE CAPACITY FROM PREVIOUS VALUE TO NEW VALUE AND SAVE

6) SCALE UP THE POD

7) IDENTIFY ON WHICH NODE THE POD IS DEPLOYED

8) GO TO PARTICULAR NODE AND EXPAND THE PARTITION

9)IF FILE SYSTEM TYPE IS XFS USE

XFS \_GROWFS/<PARTITION NAME>

IF FILE SYSTEM TYPE IS EXT4

RESIZE2FS/<[ARTITION NAME>

SNAPSHOT CREATION AND VOLUME ID ALSO SAME ABOVE STEPS REPLACE VOLUME ID

**ELK**

**ELASTIC SEARCH:** CAN USE AS DOCUMENT D.B

**LOGISTASH :** IS CALLED AS SHIPPER

**KIBANA:** IS CALLED AS VISULIZATION

USED FOR MONITERING LOGS. EG: SOME SYSTEMS DON’T HAVE ACCESS TO SEE LOGS. SO TO SEE LOGS MOVE LOGS TO ONE CENTRAL DB NOTHING BUT ELASTIC SEARCH.

INSTALL YUM TO INSTALL ELASTIC SERACH---YUM INSTALL ELASTIC SEARCH

**Q) HOW MANY MASTERS AND SLAVES**

1 MASTER AND 2 SLAVES LOGISTASH AND KIBANA ARE SETUP IN ONE NODE.

**Q)How to debug ELK**

1. I will check which indexes are not loading.
2. Will check elastic search services are running or not.
3. Will check the resources of the elk cluser.(resources: cpu,memory,disk space)
4. Check if there is any changes in filebeat file on the particular host (etc/filebeat/filebeat.yml)
5. If there are no changes I will go through the filebeat logs on particular host(var/log/filebeat)
6. If every thing is good I will go to logstash and check the configuration .

If any issue is found I will fix it and restart logstash and filebeat services.

**Q) ELASTIC SEARCH CLUSTER HEALTH**

A) TO LIST CLUSTER HEALTH

MASTERNODE:9200/\_CLUSTER/HEALTH

**Q)LIST ALL THE INDICES**

**A) MASTERNODE-9200/-CAT/INDICES?V**

METRICBEATS: TO FETCH METRICS OF NODE/SYSTEM

LOGINBEATS: TO READ LOGFILE

FILEBEATS.YML: WE SPECIFY WHICH TYPE OF FILES TO FETCH.

**Q) IF WE WANT TO READ LOGS FROM ANY HOST**

**WE NEED FILE BEATS TO INSTALL**

/ETC/FILE BEATS/FILEBEATS.YML

WE ARE FETCHING LOGS FROM RESPECTIVE HOSTS AND SENDING TO LOGISTACH.

**Q) HOW ELASTIC SEARCH WORKS.**

READ LOGS FROM RESPECTIVE NODES BY USING FILE BEATS AND SHIP TO LOGISTASH SERVER AND APPLY FILTERS IN LOGISTASH FOR CREATING DIFFERENT TYPES OF INDEXES AND SEND TO ELASTIC SEARCH.

**Q) DELETE INDEXES 14 DAYS BEFORE ONE**

A) WE HAVE ONE PLUGIN CALLED CURATOR

Q) FROM WHERE U FETCH DEPENDENCIES

A) WE R MAINTAINING ALL OUR DEPENDENCIES IN OUR ARTICRAFT AND WE WILL TARGET OUR POM FILE TO FETCH THE DEPENDENCIES FROM OUR ARTIFACTORY

**Q)FOR NODE JS**

SAME AS ABOVE

**Q) HOW U TAG BUILDING**

APPLICATION NAME.APPLICATION ID.ENV.BUILD NUMBER

BY DEFAULT JENKINS WILL BE INCREASED BY 1

**Q) WHERE WILL U PUSH DOCKER IMAGES**

A)WE HAVE PRIVATE DOCKER REGISTERY

**Q) HOW U CREATE SLAVE IMAGES**

A)WE HAVE CREATED DOCKER AGENT IMAGES.WHILE CREATING DOCKER FILE WE HAVE MENTIONED TO INSTALL JAVA RELATED NODE REALTED PACKAGES WILL ACT AS AGENTS.

WE SEND DOCKER IMAGE FILES TO DOCKER REGISTERY.

WE PUSH/PULL DOCKER IMAGES FROM DOCKER REGISTERY

**Q)MAVEN WHICH IMAGE SPINS**

A)

**Q) HOW BIG IS UR CLUSTER**

A)3 MASTER AND 10 SLAVES FOR DEV AND QA

3 MASTERS AND 10 SLAVES FOR PRODUCTION

CONFIGURATION 16GB RAM 8CORE CPU

**Q)HOW WILL U FIND OUT LIST OF NODES IN KUBERNETIS CLUSTER.**

A) KUBECTL GET NODES

**Q) HOW U FIN WHICH PROD IS DISPLAYED ON WHICH NODE**

A)KUBECTL GET PODS –OWIDE DISPLAYS POD NAME POD ID NAMESPACE

**Q) HOW TO SEE ALL COMPONENTS RELATED TO A POD**

A) KUBECTL GET ALL –L POD=POD NAME(APPLICATION NAME)

**Q) HOW TO LIST ALL D SERVICES IN NAME SPACE**

A)KUBECTL GET SERVICES –NAMESPACE

**Q)HOW TO ANNOTATE ROUTES**

A)GENERALLY IN KUBERNATES WE DON’T HAVE ROUTES, IN OPENSHIFT WE CAN ALLOCATE ROUTES

HAVE BIT KNOWLEDGE ON OPENSHIFT

1. OC ANNOTATE ROUTE ROUTE NAME –TIME=’10S’ OR ‘20S’

**Q) HOW MANY PODS RUNNING IN A NODE**

A)KUBECTL DESCRIBE NODE NODE NAME

Q) HOW U FIND LIST OF VOLUMES BY USING LSBLK(

A)ON ONE NODE HOW MANY PARTITIONS)

\* ONCE WE IDENTIFY PARTITIO NS AND CHECK FILE SYSTEM TYPE IF XFS THEN XFS\_GROWFS

Linux basic commands.

To find the disk scace

df –h

to find which folder is consuming huge diskspace.

Du –sh \*

Gzip foldername.

List all the files.

Ls –al

Tough in project

Some times my kubernetis worker nodes will be running in ½ status at that time it will take some time to debug the issue by raising a case to aws.

If it is taking huge time I ll take approval frm my manager and restart the node

If pod is scaled down and scaled up it may deploy on any of d node that is available in cluster.

How to delete pod

At that time I ll force delete d pod by using pod name once pod is deleted if we scale up by using deploy config it will create one more pod with new pod id.

Issues

Disk will fill at times we expand disk

Some times routes may not work due to time out then I ll increase the time out.

Kubectl annotate route routename –time= ‘10s’ or ‘20s’