Advantages

Increase deployment frequency

Lower failure rate of new releases

Shortened lead time between fixes

Faster mean time to recovery in the event of new release crashing

Demands of DevOPs

Team collabration

Agility

Code deployed frequently

Continuous delivery

Reliability

process

code

build - continuous Integrations - Jenkins

test - selenium

release - Contnuous Deplyment - Chef

deploy

Operate

Monitor

DevOps - practise that bring developers, testers and Monitoring team to collaborate and work together.

How is devops different from agile?

agile is a methodology - it defines certain parameters - only to development, doesnt deal with test and monitoring

DevOps - covers from starting to productions

What is the need for DevOps?

Incremented deployment frequencies - changes are faster and updations are faster.

Monitoring - requirements are tested continuously

Docker?

MicroServices - small small applications that provide different services.

In git how do u revert a commit?

git revert <commit#>

How do u find the list of files that have been changed?

git diff-tree

How do you revert last N commits?

git reset --soft Head~<last no of commits> &&

combines last N commits to 1

Continuous Integrations

jenkins plugins must be installed

plugins jenkins should be connected

Git plugin

SSH plugin

Build-Pipeline plugin

Email-ext Plugin

HTML Publisher plugin

Multi slave config plugin

Parameterized trigger

What are containers?

Virtualizations - Splitting resources into seperate boxes.

Containerization - It depends on runtime.

Provide a run time environment.

container engine is needed.

run a small application on ubuntu but u have redhat OS.

Each container has it own run time environment.

microservice - break an application into small services. with service having its own set of dependencies.

container runs only a container service

https://www.edureka.co/blog/interview-questions/top-devops-interview-questions-2016/

Git

Stores files, history, config

Hidden Repository (.git)

Contains older versions of working code in the working directory.

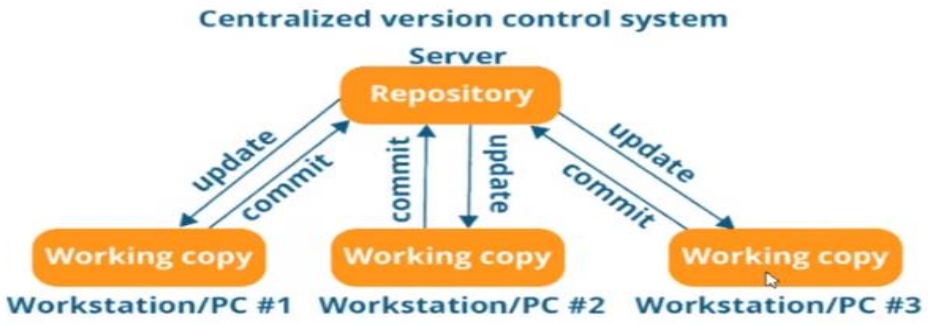
3 stages

1. Working directory
2. Staging area (committed changes but not pushed to the remote)
3. Git repository

Remote Repository (github or server)

Master Branch (repository where you are committing)

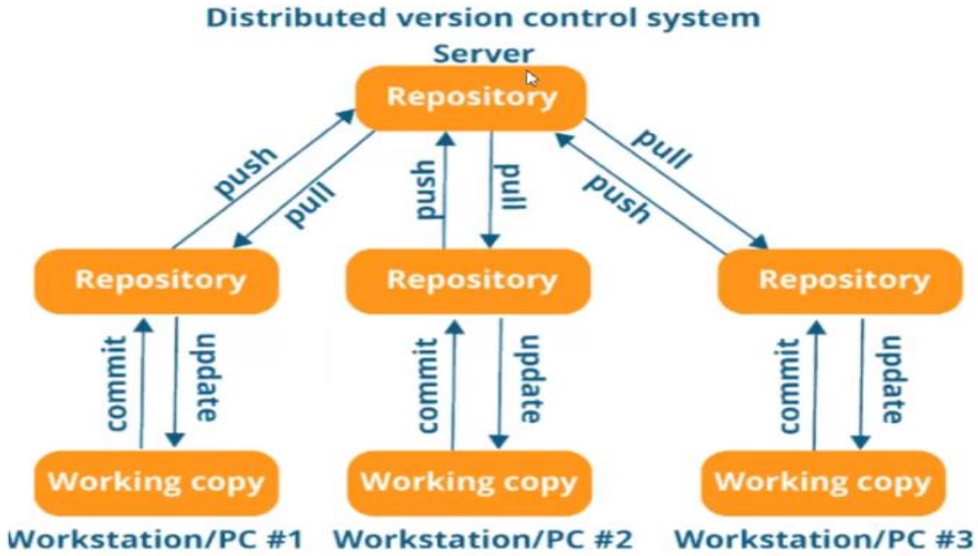
Centralized Version Control System



Drawbacks:

Only one can work on the centralized repository at a time.

Distributed Version Control System



Git is Distributed Version Control System

Configuring a User

git config

List all possible configurations

Important ones

git config –global user.name “Venkataramana, Jeevan”

git config –global user.email “jeevan.venkataramana@gmail.com”

git config –list (list all details configured)

git help

git help add – complete details about add

Initialize the repository

git init

Status of the Repository

Return information about last commit and changes done after that.

git status

red marked files are not added into the staging area

Add Changes

Adds the changes before committing

git add .

add only a certain file

git add <filename>

Commit Changes

Commit changes in to the staging directory

git commit -m “<status here>”

Logs or Commit history

History of commits

git log

commits done specific to a user

git log –author=”<name>”

commits of a certain file

git log -- <filename>

Difference in text file

git diff

difference between the staged repository and working repository

git diff –staged

Delete a file

git rm <document name>

Adding Remote Repository (GITHUB)

git remote add origin “<copy link here>”

Pulling Remote Repository from Github

git pull origin <branch name or master>

Push Changes to Remote Repository

git push origin <branch name or master>

Branches

Master branch – can be the main code

Branches – different teams can create their own branches and commit to their local branch.

Creating Branch

git branch <new branch name>

master branch code will be copied to new branch. (not a duplicate copy but a new pointer link to memory)

changes done on 1 branch will not reflected on master branch unless merged.

Switching to different branches

git checkout <branch name>

when you switch only files in that branch will be visible in that folder.

List all branches

git branch

Merging branch with master branch

Checkout to master branch `git checkout master`

Merge the branch `git merge <branch name>`

It will automatically be committed.

Un Staging

`git reset HEAD <filename>` -- after green

Changes done but not staged. You have to revert back to code that was there after previous commit.

`git checkout -- <filename>` -- when in red

Revert a Commit in GIT

Revert to last commit

`git reset HEAD~`

Revert with commit id

`git revert <commit id>` -- reverts the changes done on that commit

Fork an existing project on GITHUB

Copy some ones repository

Fork on GUI

Clone into local Machine

git clone <url>

Instructions to push

git init

git add .

git status

git commit -m “<memo>”

git remote add origin <repo url>

git push origin <master or specific branch name>

Instructions to pull

git pull origin master

git push origin master

Creating Alias

When a long command is used again and again

Suppose the below command has to be shortened

git log --all --graph --decorate –oneline ➔ git <shortened name>

git config --global alias.history “log --all --graph --decorate –oneline”

git history

Excluding files from working directory

Create a .gitignore file

nano .gitignore

write the file names which needs to be excluded into it

ignoring text files: *.txt

ignoring files in a directory: <directory name>/

https://www.edureka.co/blog/interview-questions/kubernetes-interview-questions/

1. Kubernetes vs Docker

Advantages of kubernetes:

1. Auto Scaling
2. GUI
3. Updates can be roll back.

Advantages of Docker

1. Auto load balancing
2. Can share storage volumes with any container.
3. Scales faster than kubernetes.

2. Why we need kubernetes for Docker?

   Docker builds containers and these containers communicate each other through kubernetes.

3. What is the difference between deploying applications on hosts and containers?

   Hosts – all applications share the same host os and libraries
   Containers – depends on the application and can run on any OS with a docker engine.

4. What is container orchestration?

   Consider a application need 5 micro services. Now these micro services are put in individual container and they wont be able to contact each other. Kubernetes helps communication between containers.

5. Important features of kubernetes?

   Automatic Scheduling
   Self healing
   Automated roll backs and roll outs
   Horizontal scaling and load balancing.

6. What do you know about clusters in kubernetes?

   Kubernetes is all about setting a desired state and the cluster manages the container operations so that the desired state is met.

7. What is Heapster?

   Heapster is a cluster wide aggregator of data provided by kubelet running on each node.

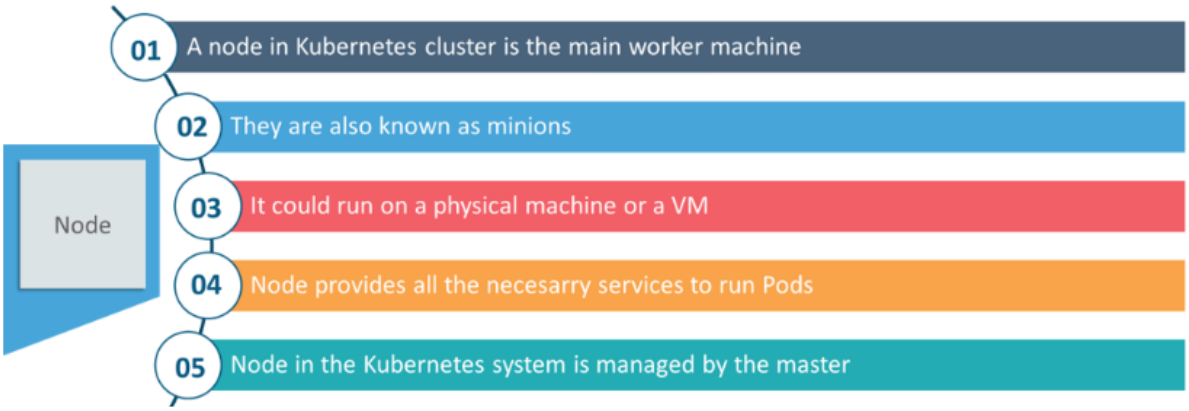
8. What is Minikube?

   Tool that makes kubernetes to run locally. Runs a single node kubernetes cluster on the VM.

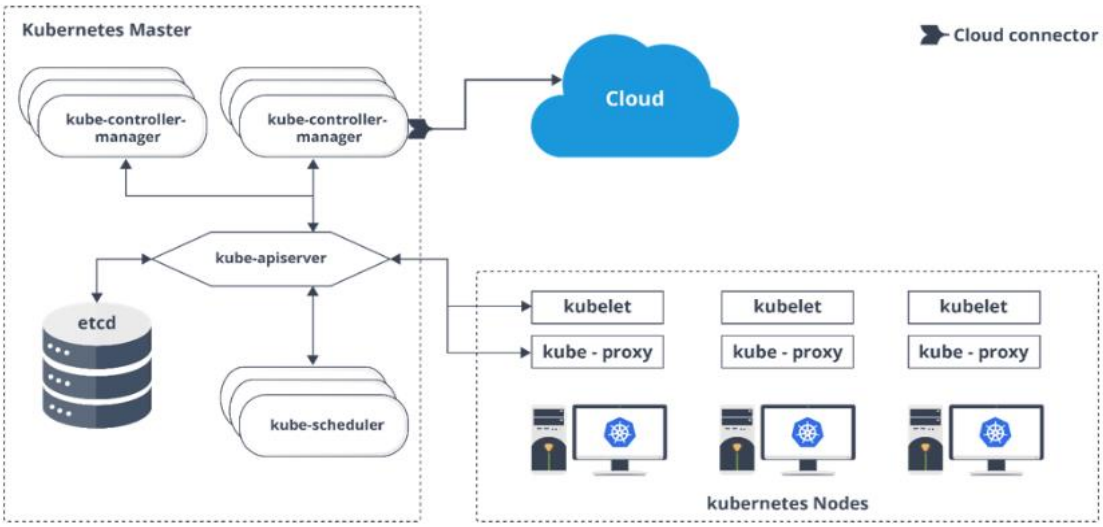
9. What is Kubectl?

   Provide command line tools for configuring kubernetes

10. What is a node?



11. What are the different components of kubernetes architecture?



12. What is kube proxy

    Runs on each and every node and can do simple TCP UDP packet forwarding across backends.

13. Working of master node in kubernetes?

    1. Kubernetes master controls the nodes and inside the nodes the containers are present.

    2. Inside, These individual containers are contained inside pods and inside each pod, you can have a various number of containers based upon the configuration and requirements.

    3. If the pods have to be deployed, then they can either be deployed using user interface or command line interface.

    4. Then, these pods are scheduled on the nodes and based on the resource requirements, the pods are allocated to these nodes.

    5. The kube-apiserver makes sure that there is communication established between the Kubernetes node and the master components.

14. What is the function of kube api server?

    Establish communication between kubernetes master and nodes.

15. What is Kube Scheduler?

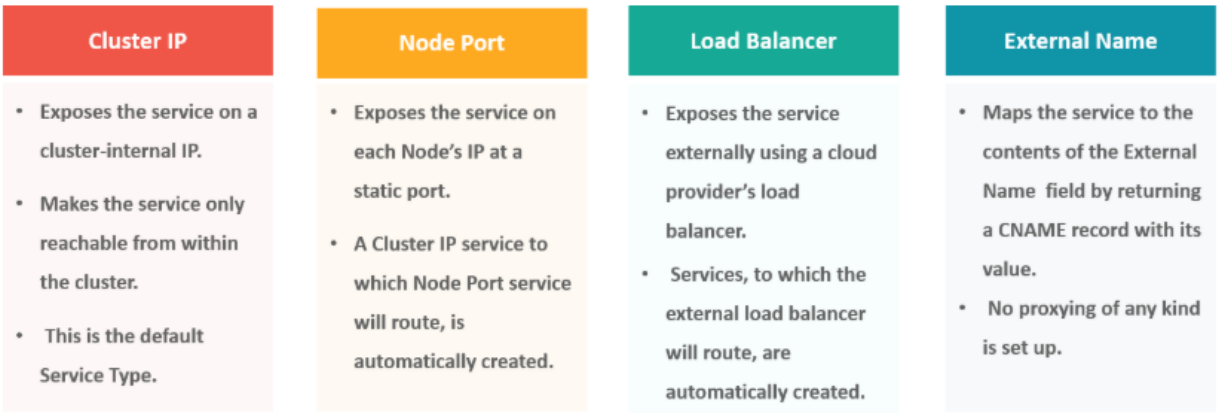
Kube scheduler is responsible for distribution and management of workload on the worker nodes.

16. What do understand by load balancing in kubernetes?

    Internal load balancer: Automatically balances load and allocated the pods with required configuration.

    External load balancer: Directs the traffic from external load to backend pods.

17. What are the different services of kubernetes?



Python Scripting

1. What is the difference between programming and scripting language?

   Scripting used to automate and no need of compilation.

2. Passing command line arguments to python scripts?

   Import sys
   sys.argv -→ will be a list with first element the file name

3. Changing directories

Import os
Os.getcwd() → returns pwd

4. Run on shell by scripts?

   Import subprocess

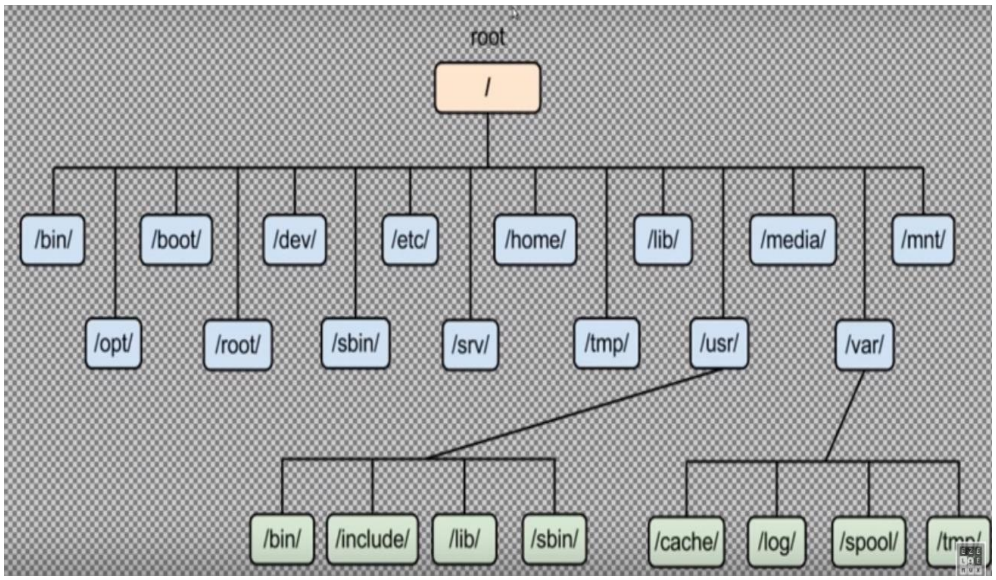
   subprocess.call("ls") → will return the exit code

   subprocess.call(["ls","-l"]) → when more arguments present

   subprocess.check_output(['ls','-l']) → returns the output as a string

   Popen → handled a process open class. Executed the program as a child process

   Linux File Structure



   /bin → all softwares, commands.

   /boot → Kernel needed boot loader

   /etc → all configuration files

/home → all drives present in the system

/lib → libraries needed for the system

/tmp → temporary files are stored

/var → logs and temporary files, caches backup