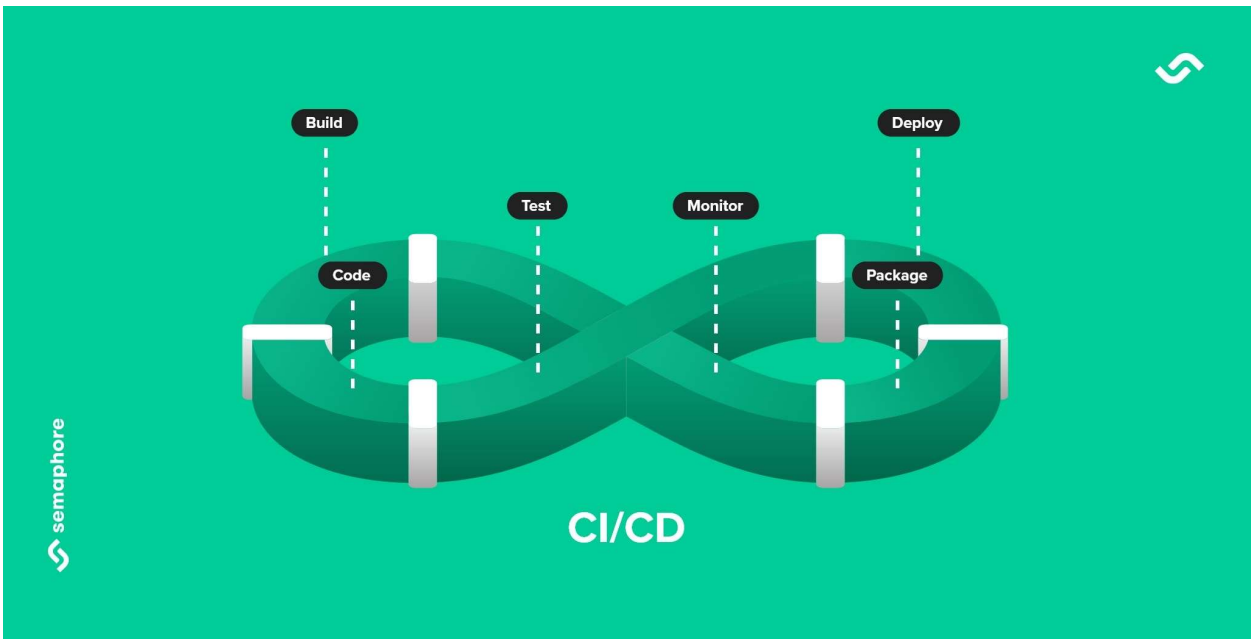
Assignment 2 :- Cloud Computing – CS 623-01

Netid :- hr8687

Name :- Rishab Parekh

Question 1:- Search and explain CI\CD ?

Answer:-



1. CI/CD is a technique in which we develop software and keep on adding updates at any time in an sustainable way.

2. The term CI/CD stands for Continuous Integration (CI) and Continuous Delivery (CD) or Continuous Deployment :-

a) **Continuous Integration (CI) :-**

i) Here developers merge the code changes to the main branch any number of times a day.

ii) Where every code merge invokes to build the code automatically and run a set of test cases or sequences.

iii) Also, the result received after running the test cases is fast, so developers can continuously stay focused in their work.

iv) The important role of Continuous Integration (CI) is to create a deployable artifact. This artifact is tested after running through different automated test is checked whether it is safe artifact with the updated version of code which can be deployed further.

b) **Continuous Delivery / Continuous Deployment:-**

i) Here, the code changes are continuous deployed whereas the deployments needs to be triggered manually.

ii) Whereas, if the process of moving code changes from the source repo to production is automated completely then it is called as **Continuous deployment.**

3. There are few principles of CI/CD which overall explains the purpose of using the CI/CD:-

a) The system that we have designed should be loosely coupled. So, that if we want any one module to be deployed we don't need to deploy the whole software or the system again only that part is build and deployed.

b) A set of test suite should be maintained and a practice of test driven environment should be maintained so once you submit your code that is tested against those test cases without any need to run them by yourself manually and we can get the results immediately in 10 – 20 min depending on the size of the module.

c) CI/CD process helps us to make deployment easy for anyone who is new to this. At the push of a button the deployment can be started easily.

d) For using CI/ CD you need to automate all process of software delivery and run it in CI/CD pipeline which includes the process of building the code and running the test suites against those code changes.

4. There are also various stages in the process of **CI/CD:-**

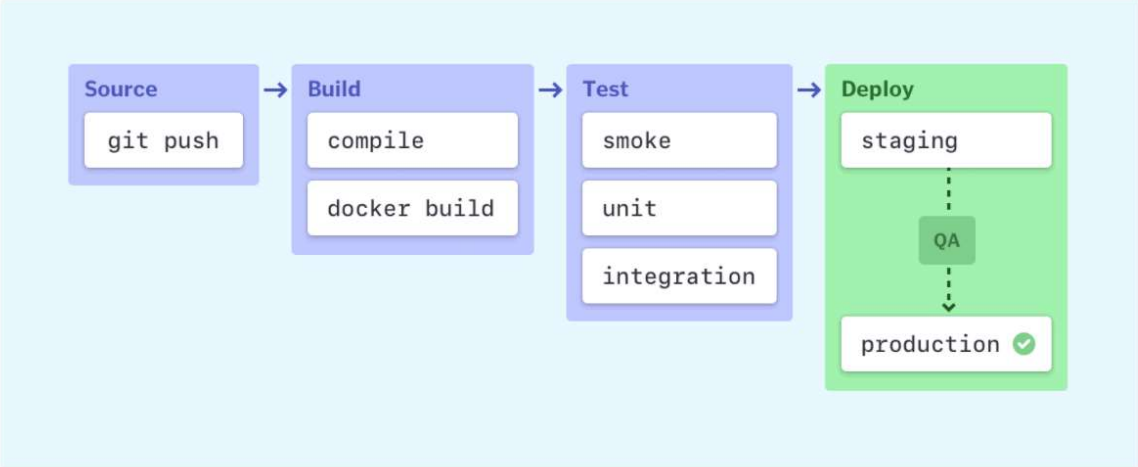
a) **Source Stage:-** Here we make the changes in code and once it is pushed a trigger is invoked in the CI/CD pipeline

b) **Build Stage:-** Once the trigger is invoked the code is then build suppose if a new dependency is added in the config file that is installed.

c) **Test Stage:-** Once the code is built it is run against the predefined test cases which should work according to the behavior of the system decided. Failure or success will return a log file of the test cases that are run.

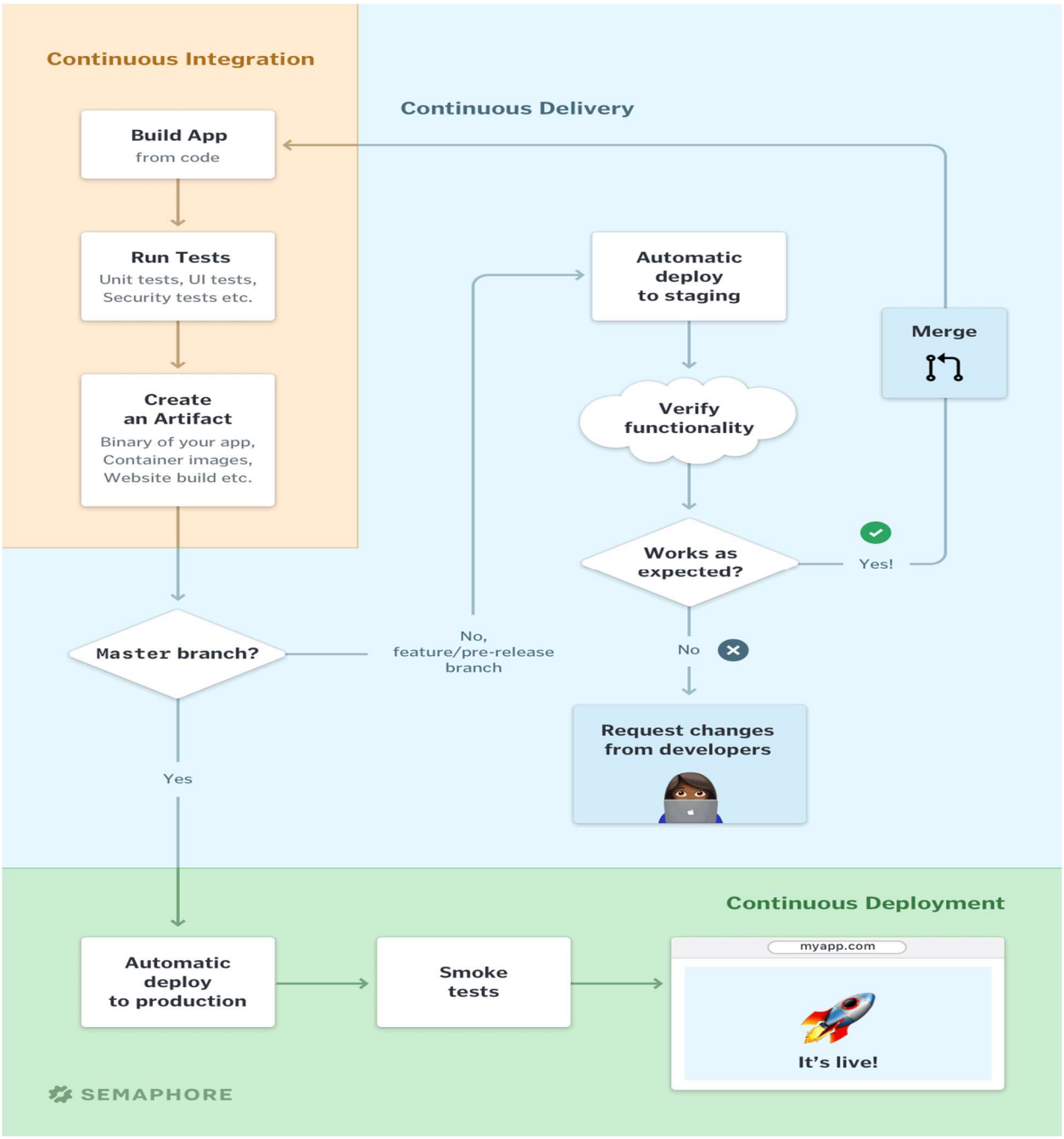
d) **Deploy stage:-** Once all the test cases are passed we move forward to deploy the artifact that we have created on to the production.

5. Below is the diagram of CI/CD pipeline process :-



Stages of a CI/CD pipeline

5. Below is the process of the CI/CD where step that we use is shown:-



Question 2:- What are the best tools for CI\CD and which one do you think is the best and why?

Answer:-

1. There are few tools which are among the best and most used. They are as follows:-
   a. **Jenkins :-** i) It is an open source tool where building of the code and continuous integration takes place.
   ii) It supports OS such as Windows, MacOS and other Unix-like OS. It provides support to various aspects by providing more than 100 plugins.

iii) It is user friendly, easily installed, high support for various plugins, provides notification on build status.

b. **CircleCi :-** i) It supports rapid software development and hosting of the changes.

   ii) It can be integrated with GitHub, BitBucket and GitHub Enterprises which makes it easier to build the code immediately by using any of above version control tools.

   iii) It supports the parallelization of the tasks.

   iv) Easy support for debugging the issue at any stage that we face. And, also supports n number of build deployments.

c. **TeamCity :-** i) It provides support for reusing settings and configuration of the child project from the main project.

   ii) Easy to interact with the software and easy to extend the server.

   iii) Provides high level of security by means of various authentication methods.

d. **Bamboo :-** i) It automates the process of software release by creating a CI/CD pipeline which covers functional testing, labelling the versions and activating new artifacts in the production.
   ii) It also maintains a previous image of the deployment so if the production fails for any reason we can revert back to previous version again.
   iii) It has integration with various version control tools such as GitHub, SVN, BitBucket and many more which makes it easy to build the project.

e. **GitLab :-** i) It is suite of tools for taking care of SDLC. It's parent product is web-based Git Repository which worldwide used.
   ii) It makes a report of analytics of our deployment. Also, since Git repo is a worldwide version control tool it is easy to push the code and commit the changes and build the latest version of code again.
   iii) Also, it has great support for Docker which enables to create the image of our project and directly deploy them.

2. There are many other CI/CD tools such as Buddy, TravisCI, Codeship, GoCD, Wercker, Semaphore, Nevercode,Spinnaker, Buildbot, etc.
3. Among all the above mentioned the best CI/CD the best tool and widely used tools is **Jenkins.**
4. **Reason** :- i) . Being an open source tool it is widely used by many businesses as well as students and small community.
   ii) Since it is used by many people the community is bigger and the sources of any issue that you face while deployment can be found anywhere or if the issue is posted it can be helped by many people.

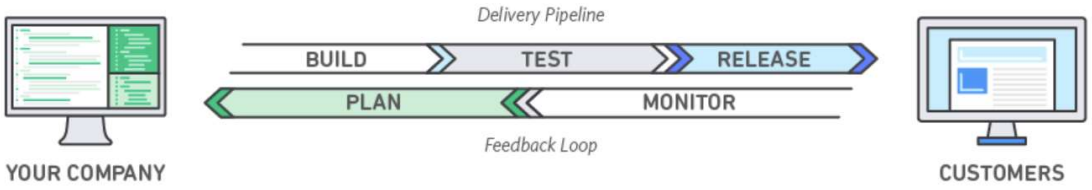
iii) Also, it supports many Operating Systems such as Windows, Linux, OSX. Also, the flexibility that it provides makes the complex deployment process easier.
iv) It provides support to many number of version control tools as specially Git which is widely used nowadays.
v) Also, since it has large community there are many user made plugins which helps out to ease the process of deployments.
vi) Also, the support for monitoring the test environment and providing the results as well the logs enable to find the issues if there are any in the deployment process and also helps us to make a report about the development process.

These are the reasons which **Jenkins the best CI/CD tool**.

Question 3:- Search and explain DevOps

Answer:-

1. DevOps is a combination of cultural philosophies, practices and tools which helps organization delivering the software and its services at a very high speed or velocity: which keeps on evolving and providing upgrades on the products using tradilational SDLC and management processes.
2. Below is the diagram of how DevOps works:



3. DevOps working Process :- a. In DevOps the development team and the Operations team are merged as one team and this team works across the SDLC of the software.
   b. Here, the full process to development, testing and deployment is taken care of by this full team.
   c. Sometimes, QA and the Security team are also tightly coupled to it to deliver the software at a faster rate.
   d. This team uses the method of automation i.e. they automate the few process of Software Development and Delivery which makes it possible to deliver software at a faster pace.
4. Benefits of DevOps services and tools are speed, rapid delivery, reliability, Scalability, Improved Collaboration and Security.
5. With the inclusion of DevOps we start following the best practices such as Continuous Integration using CI/CD tools, Continuous Delivery, Microservices

architecture, we follow the practices of IaaS, Ease of configuration management, Monitoring and logging, Communication and Collaboration.

Question 4:- What are the best tools for DevOps and which one do you think is the best and why

Answer:-

1. The concept of DevOps itself contributes to many aspects such as Delivery, Integration, Collaboration, Version Control, Deployment and many other aspects. So, there is not a single service provider which provides all these tools together. So, to cover all these aspects there are various best tools which you can use while the process of development and delivery which are as follows :-
   a. **Version Control Tool :-** i) There are various version control tools such as Github, BitBucket, SVN and many more. Among these the most widely used version control tool is **GitHub** as it is easy to use and also the integration with many CI/CD tools makes it easy for the DevOps process.
   b. **Build Tool:-** There are many tools for the building the application once the code changes are made which are Maven, Ant, Gradle and many more. For as of now we use Maven for building as it is easy to integrate with CI/CD tools and also it provides support to lots of dependencies. Also, Maven can help to build and manage projects. So, **Maven** is the best build tool. But, nowadays Gradle is coming in greater demand as well.
   c. **Continuous Integration Tool:-** As mentioned above **Jenkins** is the best integration tool as it supports version control tools such as GitHub and BitBucket which are widely used. And, also the ease at which it helps to deploy complex applications.
   d. **Configuration Management Tool :-** Ansible is the open source CM tool which is widely used for controlling the large and complex systems. These components include servers, networks, storage and applications.
   e. **Container Platforms :-** There are most widely used 2 tools for containerization which are **Docker and Kubernetes.** If the system is less complex with very few Microservices we prefer using **Docker** by if it has complex architecture then we use **Kubernetes**. So, both are preferred.
   f. **Communication and Collaboration Tool :-** Communicating with the team is also an important aspect which is under DevOps. And, in order to maintain that continuous communication and collaboration we have **Slack** which is widely used and preferred which provides many features such as video calling, notes taking, etc.
   g. **Cloud Tools :-** There are many cloud providers such as AWS, Google Cloud Platform, Azure. The mostly widely used cloud tool is AWS which provides many services such as storage, networking and other services.

h. **Monitoring, Alerting and Incident Response Tool :-** While in the process of deployment and testing we need to take care of maintain the report and logs so the tool that we can use and is widely used is **Splunk Cloud.**
i. **Testing Tool :- Selenium** is the mostly widely used tool which has the its own Selenium IDE for running and creating the test cases.
j. **IT Ticketing Tool :-** Mostly widely used tool is **ServiceNow** which helps to create the tickets if any defects are found for the designated team maybe operation, dev or testing team.

So, these are the best tools which are used in the DevOps .