BSc (Hons) in IT Specialized in CS

Year 2, 2022

Tutorial 01

SNP **DevOps**

01. What are the different phases in DevOps?

DevOps is mainly classified into 6 phases. Its phases are in a particular cycle. However, all the phases are not separated by boundaries, and no phase begins even if the previous one has ended completely. Now, let's study the DevOps cycle phase in more detail.

02. Why has DevOps gained popularity over the past few years?

Nowadays, DevOps are in great demand in the current industry, and many businesses are eagerly wanting to invest in DevOps talent. Some of the huge multi-national companies such as Facebook and Netflix are investing their money and time in DevOps for automation and pacing up application deployment as every large industry wants to see some automation in the coming years. It helps the organizations to grow and expand their businesses to generate large revenues. Its popularity continues to grow in demand as tech competition increases as most companies start adopting DevOps practices; then, it becomes even more important for the competitors to invest in similar or better development practices, increasing demand.

DevOps implementation has given provable results in businesses which contend higher efficiency, with its new technology standards; tech workers can implement codes faster than ever before, and with lesser errors. As now, more consumers and businesses rely on cloud software as it requires fast deployments to meet the consumer needs without interrupting services; this increases user adoption of cloud software like DevOps over the years.

03. What is the difference between Ansible and Puppet?

Today, DevOps professionals have to manage and control a huge number of servers hosting, so for this, they need exponential growth in computing as well as new technology such as virtualization and cloud computing. Thus Puppet and Ansible are the tools that are used for managing a large number of servers.

These are also called Remote Execution and Configuration Management tools, and it allows the admin to perform or execute the commands on many servers simultaneously. Its main feature is generally to maintain and configure thousands of servers at a single time. Apart from this, Ansible and Puppet has major differences right from the moment and can be differentiated concerning many mechanisms as shown below:

04. What are the benefits of using the Version Control System (VCS)?

- With the Version Control System (VCS), all the workers are allowed to access the file freely at any time. It also allows merging all the changes that are made in a common version.
- It is designed to help multiple people by collaboratively edit text files, which makes sharing comparatively easy between multiple computers.
- It is important for documents that require a lot of redrafting and revision as they provide an audit trail for redrafting and updating final versions.
- It permits all the team members to have access to the complete history of the project so that in case of any breakdown in the central server, we can use any teammate's storehouse.
- All the previous versions and variants are smartly packed up inside the VCS. Any version is requested at any time to get information about the previous complete projects.

05. What is the purpose of configuration management in DevOps?

Configuration management helps in automating tasks that are otherwise time-consuming and tedious and enhances an organization's agility. It brings consistency and improves the process of a product/service by streamlining design, documentation, control, and implementation of changes during various phases of the project.

06. What is the difference between a centralized and distributed version control system (VCS)?

In a centralized repository system, the repository is located in a central location, and clients access this system when they need something. In such a version control system, the repository is always updated with the latest changes as the changes are directly committed to the central system; therefore, all the clients always have access to the latest code. CVS and SVN are examples of centralized VCS.

In a distributed VCS, everyone in the team has their repository, which is a mirror of the central repository. It provides flexibility, as one can work offline. Only when the changes have to be committed to the central system, you need to be online. This makes distributed VCS faster. Git and Mercurial are distributed VCS.

07. Explain the differences between git pull and Git fetch?

git fetch	**git pull**
the command to use this feature is: git fetch <remote>	command to use: git pull <remote> <branch>
Fetches the changes from the remote repository but doesn't merge them with the local repository.	Fetches the changes of the branch from the remote repository and merges them with the local repository pull = fetch + merge
Done at latter stages, since no merging is involved, there are no conflicts to be resolved.	There are chances of merge conflicts if two or more people are working on different copies of the same code/file.
The local repository is unchanged, but the central repository is updated.	The changes from the central repository are updatec in the local repository.
Developers can see the changes made by others before they push their changes for integration.	Developers can first bring the latest files to their loca and then start updating the same.

08. What is a merge conflict in Git, and how can it be resolved?

Merge conflicts occur when changes are made to a single file by multiple people at the same time. Due to this, Git won't be able to tell which of the multiple versions is the correct version. To resolve the conflicts, we should create a new Git repo, add a file, create a branch, make the edits and commit the changes. The next step is to merge the new branch into the master. Once this is done, Git clearly shows the differences in the different versions of the file and where the edits need to be made to remove the conflicts.