

DevOps Engineer

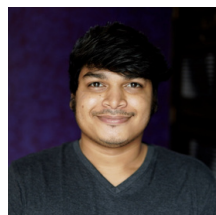


Interview



Questions & Answers Part 1

By Sandip Das



Q1: What is DevOps? explain in simple words

Answer

When get asked this question interview seeks your clear understanding of overall DevOps concepts, keep it simple, concise, and to the points (of course be creative and answer it your own way) e.g.

DevOps is the combination of cultural philosophies, practices, and tools that helps companies/organizations deliver applications and services faster & efficiently.

In a Simple sense "**DevOps**" represents "**Dev**" and "**Ops**" which stands for "**Development Team**" & "**Operations Team**", **DevOps** promotes communications and collaborations between Development and Operations teams, shares many responsibilities, and combine their workflows. This reduces inefficiencies and saves time e.g. reduced handover periods between developers and operations, writing code that takes into account the environment in which it is run.

Q2: What are the popular DevOps Practices? mention some popular related Tools for the same

Answer

Continuous Integration

Continuous integration is a software development practice where developers regularly merge their code changes into a shared central repository, after which automated builds and tests are run, this way we can find the bugs quicker, improve software quality, and save time .e.g. Jenkins, CircleCI, GitLabCI, Bamboo, AWS CodeBuild

Continuous Delivery

Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for release to production, as soon as CI is done, this takes actions and changes get deployed to test, staging, production (as configured), this way Developers will always have deployable artifact ready all the time. e.g.

Microservices

The microservices architecture is a design approach to build a single application as a set of small services. Each service runs in its own process and communicates with other services through a well-defined interface using a lightweight mechanism, typically an HTTP-based application programming interface (API) e.g. API Gateway + AWS Lambda, Load Balancer + AWS ECS, Kong, Kubernetes etc

Infrastructure as Code

Infrastructure as code is a practice in which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration. With the available Cloud API-driven model, it is possible to provision or configures resources via code instead of manually doing it. This helps to make reuse code, maintain consistency, save time and easily create duplicate infrastructure e.g. test, staging, production. e.g. Terraform, Pulumi, AWS CloudFormation, CDK etc

Monitoring and Logging

Organizations monitor metrics and logs to see how application and infrastructure performance impacts the experience of their product's end-user. This is used to find out root causes of problems if any issue happens, find out some insightful data by analyzing the logs, this also helps generate the alerts based on any incident. e.g. Cloudwatch, PagerDuty Dynatrace, Datadog, New Relic, Prometheus & Grafana, Nagios etc

Communication and Collaboration

Increased communication and collaboration in an organization is one of the key cultural aspects of DevOps. The use of DevOps tooling and automation of the software delivery process establishes collaboration by physically bringing together the workflows and responsibilities of development and operations. e.g. Microsoft teams, Jira, Slack etc

Q3: What is Source/Version Control Management (SCM/VCM)? Why needed with examples?

Answer

Source or Version Contoll Manageent

Source or version control refers to tracking and managing changes to code. This ensures that developers are always working on the right version of source code.

Why it needed?

- 1) it's allowing multiple developers to make changes to a specific branch or version and figure out who made what changes, it improve collaboration and accelerate the bug fixes, new changes thus improving release velocity.
- 2) Maintain a single source of truth (Code never lies)

Popular Source/Version Controll Tools

Git

Subversion

Mercurial

Perforce Helix Core

IBM Rational ClearCase

Q4: What is Configuration Management ? What are the popular configuration management tools ?

Answer

Configuration Management

Configuration management is a process for maintaining computer systems, servers, and software in a desired, consistent state. It's a way to make sure that a system performs as it's expected to as changes are made over time.

Popular Configuration Managemet Tools

Ansible

Chef

Puppet

Salt

Q5: What is Ansible?

Answer

Ansible

Ansible is a DevOps tool written in Python. It delivers simple IT automation that ends repetitive tasks and frees up DevOps teams for more strategic work.

It automates configuration management, cloud provisioning, application deployment, intra-service orchestration, and many other IT needs.

When Ansible is used as a configuration management tool, it is used to store the current state of our systems and help us to maintain that state, it makes changes and deployments faster, removing the potential for human error while making system management predictable and scalable.

We can run Ansible commands directly as Ad-hoc commands or write Ansible Playbook in YAML files and put all the business logic there

Q6: What is Chef?

Answer

Chef

Chef in DevOps is an automation tool written in Ruby & Erlang. It is a Configuration Management (CM) tool that lets you automate processes and tasks across numerous servers and other devices of an organization in simple steps. Such a framework is highly beneficial to a company.

Chef uses "recipes" written in Ruby to keep your infrastructure running up-to-date and compliant. The recipes describe a series of resources that should be in a particular state. Chef can run in client/server mode or in a standalone configuration named chef-solo. It has good integration with the major cloud providers to automatically provision and configure new machines.

Q7: What is SALT or SaltStack ?

Answer

Salt or Saltstack

It is a Python-based, open-source software/Tool for event-driven IT automation, remote task execution, and configuration management.

Salt uses a client-server topology (with the Salt master as server and Salt minions as clients). Configurations are kept in Salt state files, which describe everything required to keep a system in the desired state.

Q8: What are different phases in DevOps methodology

Answer

Phases of DevOps Metnodology

Planning ==> Development ==> Continuous Integration (CI) ==> Continuous Delivery / Deployment (CD) ==>
Operations ==> Monitoring

Q9: Agile and DevOps same or different ?

Answer

Agile

Agile is a software development methodology that focuses on incremental, iterative, and rapid releases of software features by involving the customer by means of feedback. **This methodology removes the gap between the requirement understanding of the clients and the developers**

DevOps

DevOps is a practice/culture that allows the collaboration of the development team and the operations team to come together for successful product development, it involves practices like continuous development, continuous integration, continuous testing, continuous deployment, and continuous monitoring.

DevOps tries to reduce the gap between the developers and the operations team

Verdict

We can call DevOps an extended Agile practice

Q10: Differentiate between Continuous Deployment and Continuous Delivery

Answer

Continuous Delivery vs. Continuous Deployment

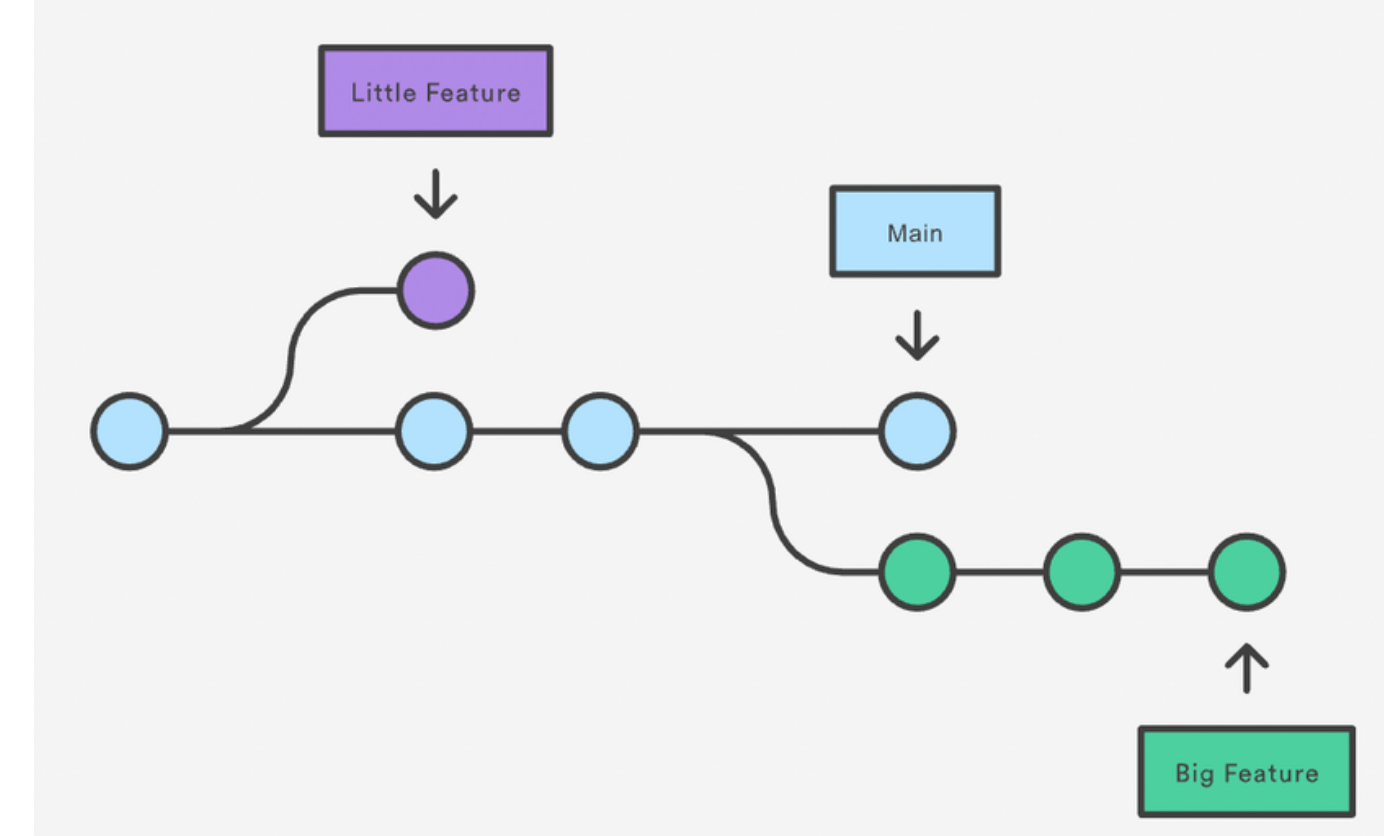
With continuous delivery, every code change is built, tested, and then pushed to a non-production testing or staging environment. There can be multiple, parallel test stages before a production deployment. The difference between continuous delivery and continuous deployment is the presence of a manual approval to update to production. With continuous deployment, production deployment happens automatically without explicit approval.

Q11: What is Git Branching Strategy in DevOps?

Answer

Git Branching

Branching is a feature available in most modern version control systems. Branching facilitates the development of bug fixes, the addition of new capabilities and the integration of new versions after they have been tested in isolation



Branching Strategy

A branching strategy helps define how the dev team functions and how each feature, improvement, or bug fix is handled by the developers. It also reduces the complexity of the delivery pipeline by allowing developers to focus on developments and deployments only on the relevant branches—without affecting the entire product. Below is the most popular git-flow:

Primary Branches: Main/Master, Develop

Feature Branches: this branches names start with **feature-**^{*}, mainly used for new features

Issue fixes branches: These branches start with **hotfix-***, mainly deal with fixes for the master branch, i.e. pulled from the master branch, then fixes pushed, then merged back with the master branch

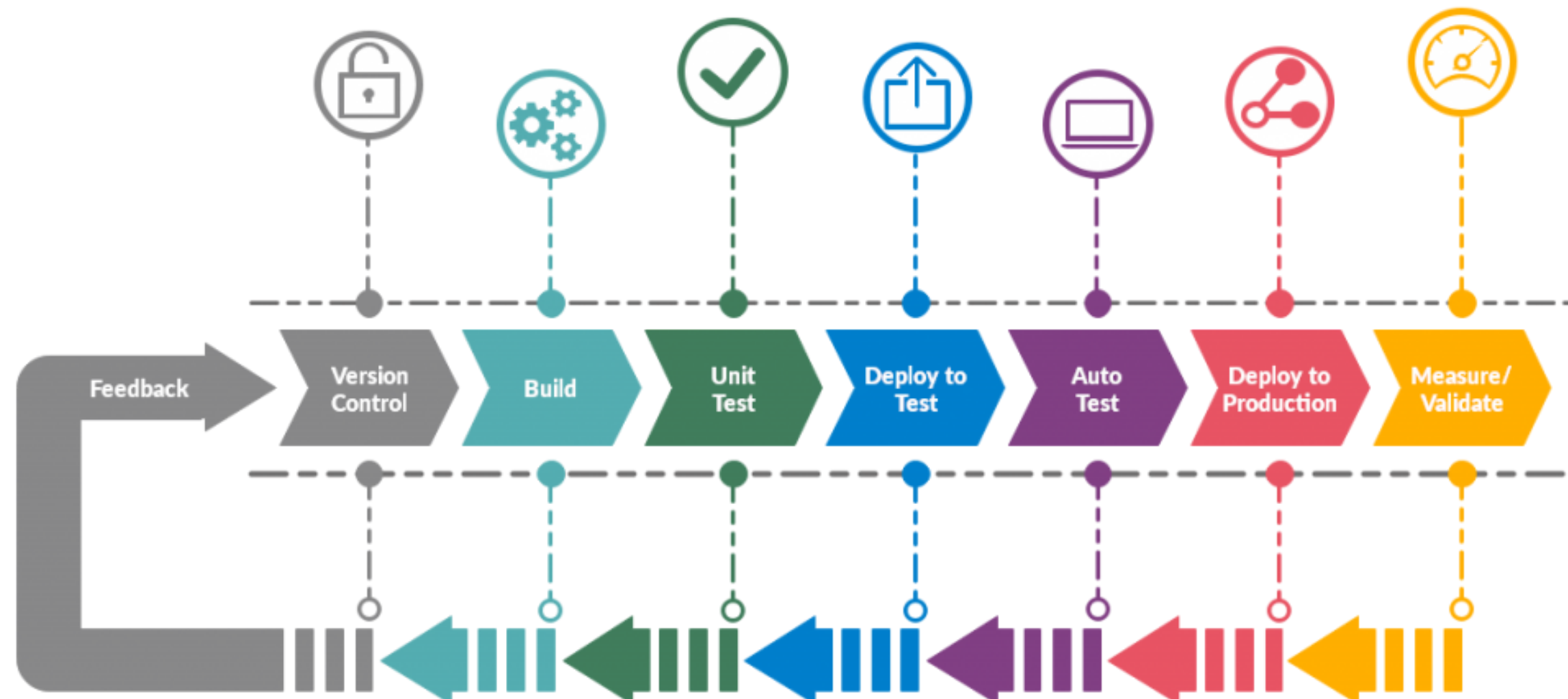
Release branch: this branch's name starts with the **release-***, this branch is mainly used for production releases or distribution purposes.

Q12: What is DevOps Pipeline?

Answer

DevOps Pipeline

A pipeline, in general, is a set of automated tasks/processes defined and followed by the software engineering team. DevOps pipeline is a pipeline which allows the DevOps engineers and the software developers to efficiently and reliably compile, build and deploy the software code to the production environments in a hassle free manner.



Q13: What is Autoscaling?

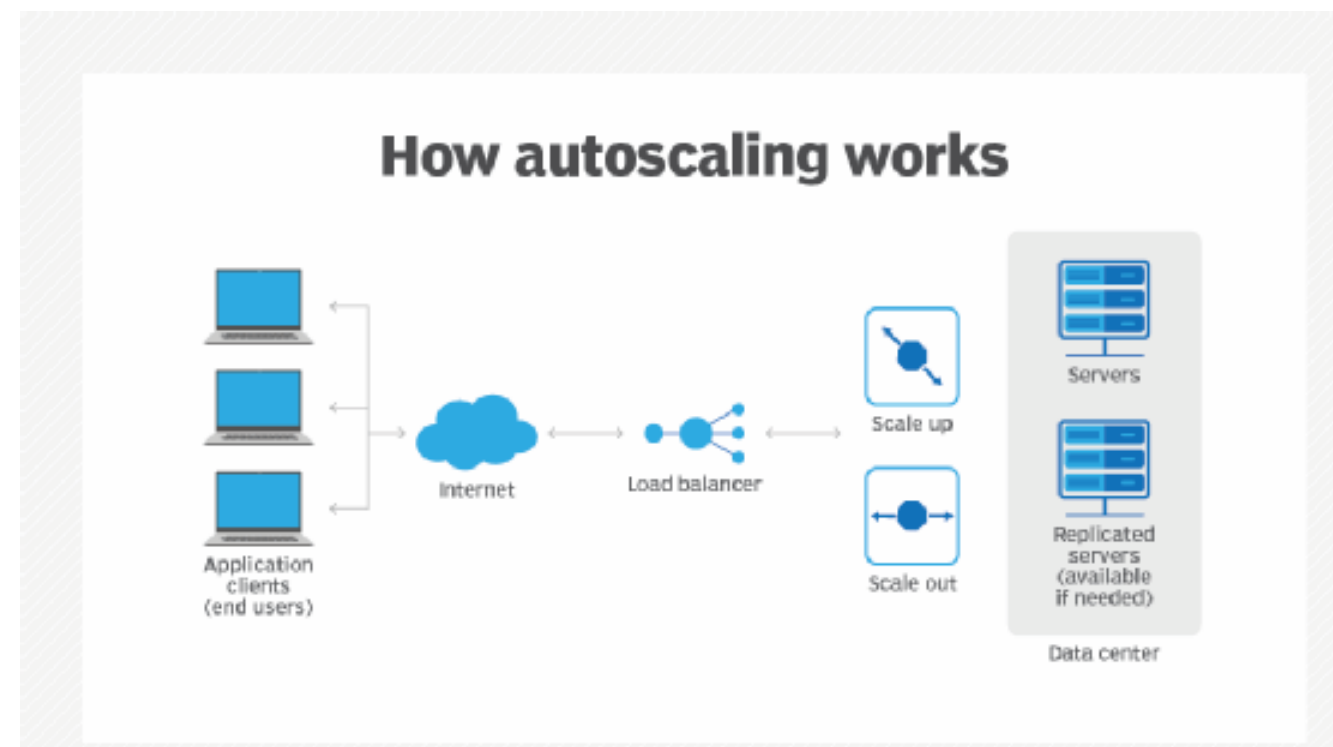
Answer

Autoscaling

Autoscaling is a cloud computing technique, it monitors your applications and automatically adjusts capacity (either increase or decreases) to maintain steady, predictable performance at the lowest possible cost based on the configured metrics.

There is two types of Autoscaling

- 1) **Vertical Autoscaling:** It's upgrading the main computing resources to achieve scalability e.g. upgrading CPU or RAM or Network Bandwidth
- 2) **Horizontal Autoscaling:** It's creating the same copy or replica and distributing load between all the newly created computing resources



Q14: What is Blue/Green Deployment?

Answer

BlueGreen Deployment

A blue-green pattern is a type of continuous deployment, application release pattern which focuses on gradually transferring the user traffic from a previously working version of the software or service to an almost identical new release - both versions running on production.

The blue environment would indicate the old version of the application whereas the green environment would be the new version.

The production traffic would be moved gradually from blue to green environment and once it is fully transferred, the blue environment is kept on hold just in case of rollback necessity.

Q15: What is Docker , Docker Image, Docker Container?

Answer

Docker

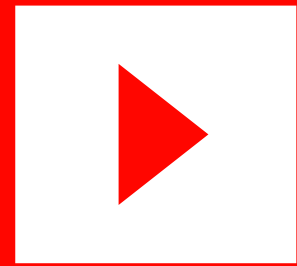
Docker is a containerization technology that packages your application and all its dependencies together in the form of Containers to ensure that your application works seamlessly in any environment.

Docker Image

Docker image is the source of Docker container. Or in other words, Docker images are used to create containers.

Docker Container

Docker Container is the running instance of Docker Image.



SUBSCRIBE

[http://](http://Learn.sandipdas.in)  Learn.sandipdas.in

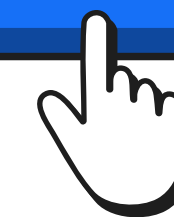
Contact Me



contact@sandipdas.in



FOLLOW



Support My Work



 Via **Patreon**: <https://www.patreon.com/learnwithsandip>

 Via **Buy Me a Coffee**: <https://www.buymeacoffee.com/LearnWSandip>

