DOCKER DEVOPS

Phaniraj

WHAT IS DEVOPS?

- DevOps is a collaboration between Development and IT Operations to make software production and Deployment in an automated & repeatable way.
- The full form of 'DevOps' is a combination of 'Development' and 'Oper

DevOps Explanation

WHY IS DEVOPS NEEDED?

- Before DevOps, the development and operation team worked in complete isolation.
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members spend a large amount of their time testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production.
- Coding & operation teams have separate timelines and are not synch, causing further delays.

HOW IS DEVOPS DIFFERENT FROM TRADITIONAL IT

- Old Process:
  - After placing an order for new servers, the Development team works on testing. The Operations team works on extensive paperwork as required in enterprises to deploy the infrastructure.
  - Projections about failover, redundancy, data center locations, and storage requirements are skewed as no inputs are available from developers who have deep knowledge of the application.
  - The operations team has no clue about the progress of the Development team. The operations team develops a monitoring plan as per their understanding.
  - Before going go-live, the load testing crashes the application, and the release is delayed.

THE NEW WAY OF DEPLOYMENT AND OPERATIONS

- After placing an order for new servers Development and Operations team work together on the paperwork to set up the new servers. This results in better visibility of infrastructure requirements.
- Projections about failover, redundancy, disaster recovery, data center locations, and storage requirements are pretty accurate due to the inputs from the developers.
- In DevOps, the Operations team is completely aware of the developers' progress. Operations teams interact with developers and jointly develop a monitoring plan that caters to IT and business needs. They also use advanced Application Performance Monitoring (APM) Tools.
- Before going go-live, the load testing makes the application a bit slow. The development team quickly fixes the bottlenecks, and the application is released on time.

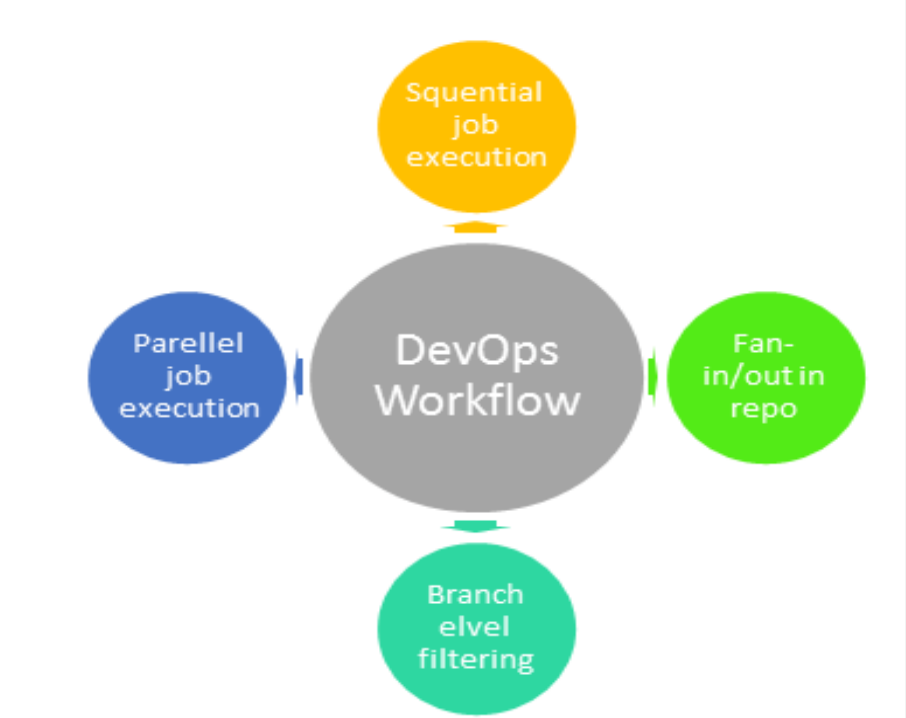
WHY DEVOPS IS USED?

- DevOps allows Agile Development Teams to implement Continuous Integration and Continuous Delivery, which helps them launch products faster into the market.
- Predictability
- Reproducibility
- Maintainability

WHEN NOT TO ADOPT DEVOPS?

- It should not be used in mission-critical applications like banks, power and other sensitive data sites.
- Such applications need strict access controls on the production environment, a detailed change management policy, and access control policy to the data centers.

DEVOPS WORKFLOW



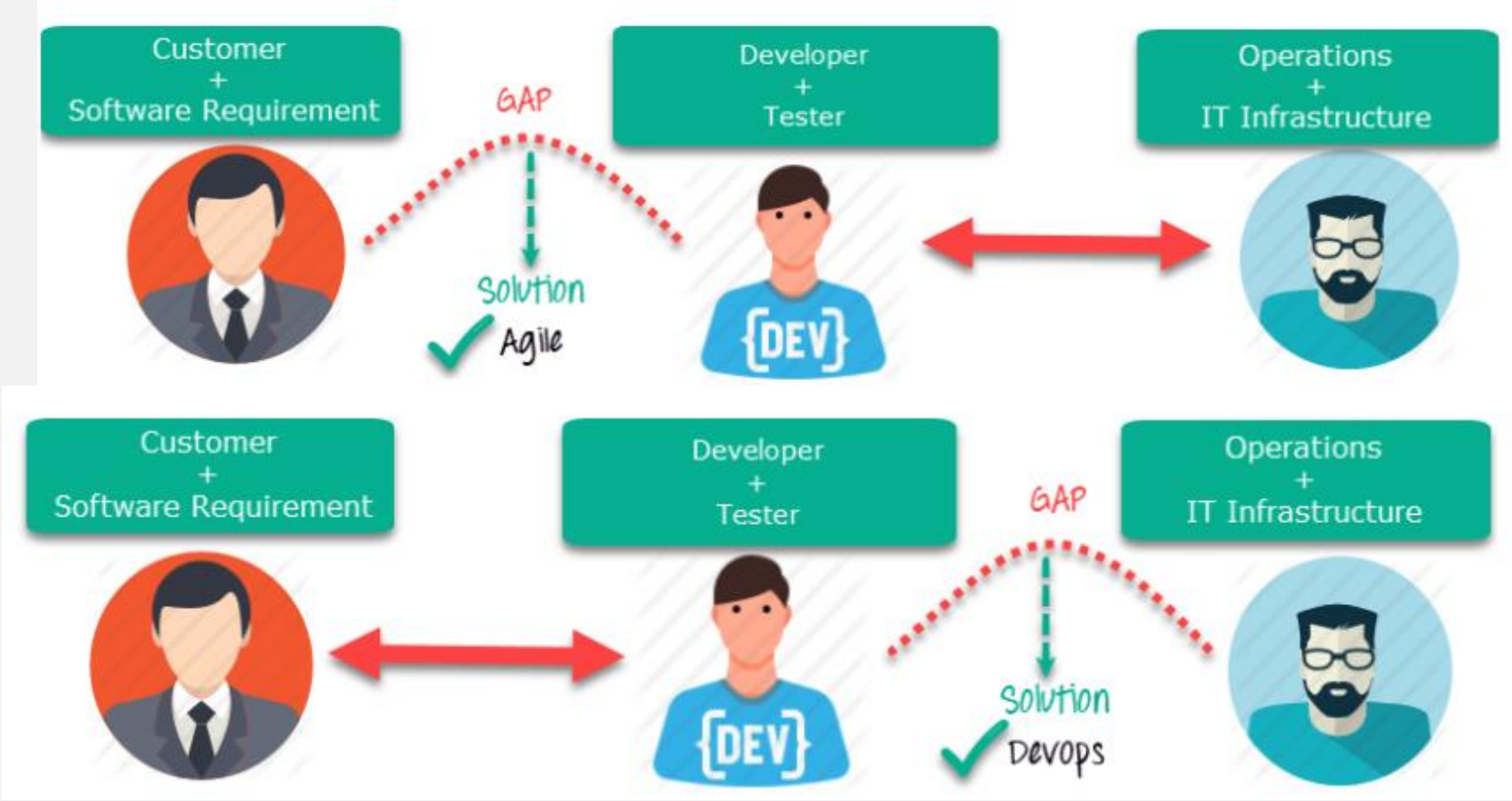
DevOps WorkFlow

DEVOPS LIFE CYCLE

MAJOR DEV OPS TOOLS

- Version Control Tool: Git (GitLab, GitHub, Bitbucket)
- Build Tool: Maven
- Continuous Integration Tool: Jenkins
- Configuration Management Tool: Chef
- Configuration Management Tool: Puppet
- Project Management: Jira and Confluence

DEVOPS VS AGILE



AGILE

- Emphasize breaking down barriers between developers and management.
- Addresses gaps between customer requirements and development teams.
- Focuses more on functional and non-functional readiness
- Agile development pertains mainly to the company's way development is thought out.
- Agile development emphasises training all team members to have varieties of similar and equal skills. So that, when something goes wrong, any team member can get assistance from any member in the absence of the team leader.
- Agile development manages on "sprints". It means that the timetable is much shorter (less than a month), and several features are to be produced and released in that period.

DEVOPS

- DevOps is about software deployment and operation teams.
- Addresses the gap between the development and Operation team
- It focuses on operational and business readiness.
- DevOps emphasises deploying software in the most reliable and safest ways that aren’t always the fastest.
- DevOps likes to divide and conquer, spreading the skill set between the development and operation teams. It also maintains consistent communication.
- DevOps strives for consolidated deadlines and benchmarks with significant releases rather than smaller and more frequent ones.

DEVOPS PRINCIPLES

- 1. Customer-Centric Action: The DevOps team must constantly take customer-centric action to invest in products and services.
- 2. End-To-End Responsibility: The DevOps team needs to provide performance support until they become end-of-life. This enhances the level of responsibility and the quality of the products engineered.
- 3. Continuous Improvement: DevOps culture focuses on continuous improvement to minimize waste, and it continuously speeds up the improvement of products or services offered.
- 4. Automate everything: Automation is a vital principle of the DevOps process, and this is not only for software development but also for the entire infrastructure landscape.
- 5. Work as one team: In the DevOps culture, the designer, developer, and tester are already defined, and all they need to do is work as one team with complete collaboration.
- 6. Monitor and test everything: Monitor and test everything: The DevOps team needs robust monitoring and testing procedures.

WHO IS DEVOPS ENGINEER?

- An IT professional who works with software developers, system operators, and other production IT staff to administer code releases.
- DevOps should have hard and soft skills
- Should have basics of Software development languages.

ROLES, RESPONSIBILITIES, AND SKILLS OF A DEVOPS ENGINEER

- Able to perform system troubleshooting and problem-solving across platform and application domains.
- Manage project effectively through open, standards-based platforms
- Increase project visibility thought traceability
- Improve quality and reduce development cost with collaboration
- Analyse, design and evaluate automation scripts & systems
- Ensuring critical resolution of system issues by using the best cloud security solutions services
- DevOps engineers should have the soft skill of problem-solver and quick-learner

DEVOPS AUTOMATION TOOLS

- Infrastructure Automation
- Configuration Management
- Deployment Automation
- Performance Management
- Log Management
- Monitoring

INFRASTRUCTURE AUTOMATION

- Amazon Web Services (AWS): Being a cloud service, you do not need to be physically present in the data center.
- Azure

CONFIGURATION MANAGEMENT

- Chef: It is a valuable DevOps tool for achieving speed, scale, and consistency. It can be used to ease out complex tasks and perform configuration management. The DevOps team can avoid making changes across ten thousand servers with this tool. Instead, they need to make changes in one place, automatically reflected in other servers.

DEPLOYMENT AUTOMATION

- Jenkins: This tool facilitates continuous integration and testing. It helps to integrate project changes more efficiently by quickly finding issues as soon as a built is deployed.

WHAT IS THE FUTURE OF DEVOPS?

- Organizations are shifting in their needs to weeks and months instead of years.
- We will see soon that DevOps engineers have more access and control of the end-user than any other person in the enterprise.
- DevOps is becoming a valued skill for IT people. For example, a survey conducted by Linux hiring found that 25% of respondents' job seekers are DevOps experts.
- DevOps and continuous delivery are here to stay. Therefore companies need to change as they have no choice but to evolve. However, the mainstreaming of DevOps will take 5 to 10 years.

WHAT IS DOCKER?

- Docker is a tool that allows developers, sys-admins etc. to easily deploy their applications in a sandbox (called containers) to run on the host operating system i.e. Linux.
- It allows users **to package an application with all of its dependencies into a standardized unit** for software development.
- Containerization is a technology that's been around for a long time, but it's seen new life with Docker.

CONTAINERS

- Containers offer a logical packaging mechanism in which applications can be abstracted from the environment in which they actually run.
- This gives developers the ability to create predictable environments that are isolated from the rest of the applications and can be run anywhere.

SETTING UP YOUR COMPUTER

- Download Docker Desktop for Windows from the following link:
  - https://docs.docker.com/desktop/windows/install/
- WSL 2 backend
  - Download and install the Linux kernel update package.

INSTALLATION SETUP

- Double-click Docker Desktop Installer.exe to run the installer.
- When prompted, ensure the Enable Hyper-V Windows Features or the Install required Windows components for WSL 2 option is selected on the Configuration page.
- Follow the instructions on the installation wizard to authorize the installer and proceed with the install.
- When the installation is successful, click Close to complete the installation process.
- If your admin account is different to your user account, you must add the user to the docker-users group. Run Computer Management as an administrator and navigate to Local Users and Groups > Groups > docker-users. Right-click to add the user to the group. Log out and log back in for the changes to take effect.

DOCKER COMMANDS

- docker images: see a list of all images on your system.
- docker run imageName: Runs the image.
- docker ps: Shows you all containers that are currently running.
- Docker rm containerid

RUN MYSQL IN DOCKER

- Start Mysql container
  - To start the container, type the command below in your terminal:
  - docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=1234 -e MYSQL_DATABASE=test -e MYSQL_USER=sa -e MYSQL_PASSWORD=1234 --publish 6603:3306 -d mysql:5.7
- Create a Database:
  - docker exec -it mysql-db bash
  - mysql -h localhost -u root –p
  - Create a User for out-of-container access as the root access is not allowed.(Optional)
  - Now create the database using DDL command