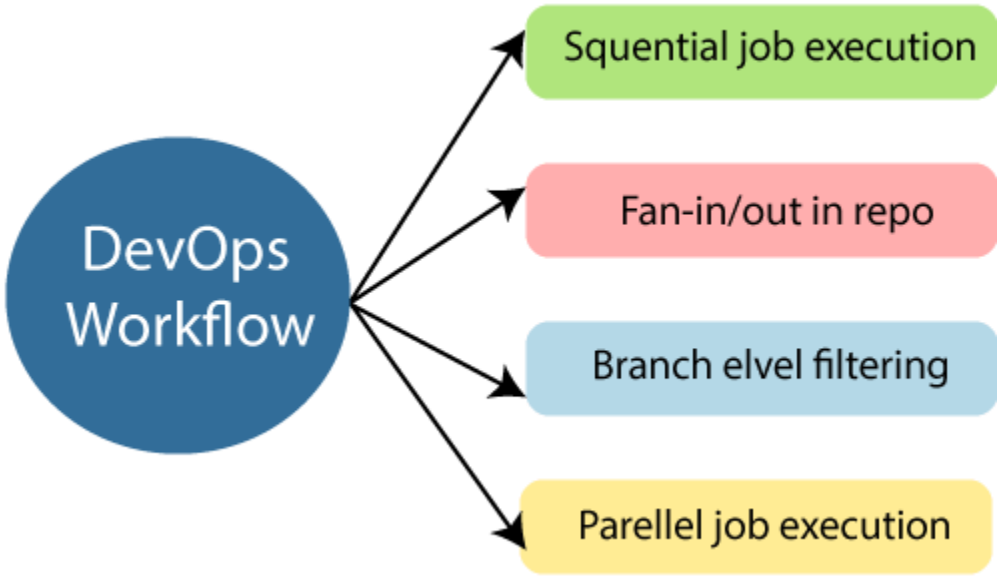
DevOps Workflow

DevOps workflow provides a visual overview of the sequence in which input is provided. Also, it tells about which one action is performed, and output is generated for an operations process.



DevOps workflow allows the ability to separate and arrange the jobs which are top requested by the users. Also, it gives the ability to mirror their ideal process in the configuration jobs.

DevOps Principles

The main principles of DevOps are Continuous delivery, automation, and fast reaction to the feedback.

1. **End to End Responsibility:** DevOps team need to provide performance support until they become the end of life. It enhances the responsibility and the quality of the products engineered.
2. **Continuous Improvement:** DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.

3. **Automate Everything:** Automation is an essential principle of the DevOps process. This is for software development and also for the entire infrastructure landscape.
4. **Custom Centric Action:** DevOps team must take customer-centric for that they should continuously invest in products and services.
5. **Monitor and test everything:** The DevOps team needs to have robust monitoring and testing procedures.
6. **Work as one team:** In the DevOps culture role of the designers, developers, and testers are already defined. All they needed to do is work as one team with complete collaboration.

These principles are achieved through several DevOps practices, which include frequent deployments, QA automation, continuous delivery, validating ideas as early as possible, and in-team collaboration.

DevOps Practices

Some identified DevOps practices are:

- Self-service configuration
- Continuous build
- Continuous integration
- Continuous delivery
- Incremental testing
- Automated provisioning
- Automated release management

DevOps Automation

Automation is the crucial need for DevOps practices, and automate everything is the fundamental principle of DevOps. Automation kick starts from the code generation on the developers machine, until the code is pushed to the code and after that to monitor the application and system in the production.

Automating infrastructure set up and configurations, and software deployment is the key highlight of DevOps practice. DevOps practice id is dependent on automation to make deliveries over a few hours and make frequent deliveries across platforms.

Automation in DevOps boosts speed, consistency, higher accuracy, reliability, and increases the number of deliveries. Automation in DevOps encapsulates everything right from the building, deploying, and monitoring.

DevOps Automation Tools

In large DevOps team that maintain extensive massive IT infrastructure can be classified into six categories, such as:

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

Below are few tools in each of these categories let see in brief, such as:

Infrastructure Automation

Amazon Web Services (AWS): Being a cloud service, you don't need to be physically present in the data center, they are easy to scale on-demand, and there are no up-front hardware costs. It can be configured to provide more servers based on traffic automatically.

Configuration Management

Chef: Chef is a handy DevOps tool for achieving speed, scale, and consistency. It can be used to ease out of complex tasks and perform configuration management. With the help of this tool, the DevOps team can avoid making changes across ten thousand servers. Rather, they need to make changes in one place, which is automatically reflected in other servers.

Deployment Automation

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

Performance Management

App Dynamic: It offers real-time performance monitoring. The data collected by this tool help developers to debug when issues occur.

Log Management

Splunk: This DevOps tool solves issues such as storing, aggregating, and analyzing all logs in one place.

Monitoring

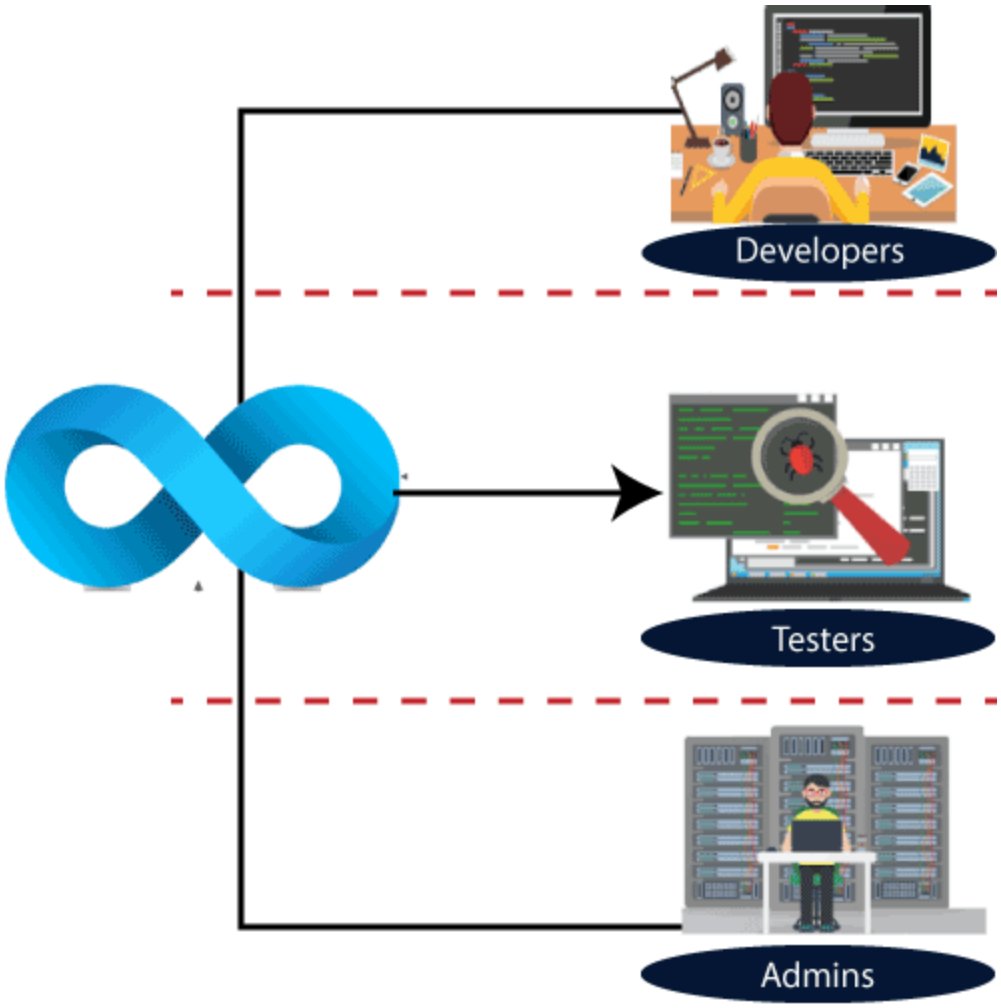
Nagios: It notified people when infrastructure and related service go down. Nagios is a tool for this purpose, which helps the DevOps team to find and correct problems.

DevOps Engineers

DevOps Engineer is an IT professional who works with system operators, software developers, and other production IT staff to administer code releases.

DevOps engineer understands the software development lifecycle and various automation tools for developing digital pipelines.

DevOps have hard as well as soft skills to communicate and collaborate with development, testing, and operations teams.



DevOps engineers need to code occasionally from scratch, and they must have the basics of software development languages.

The DevOps engineer will work with development team staff to tackle the coding and scripting needed to connect elements of code, like libraries or software development kits.

A bachelor's degree in computer science or related fields is generally required for DevOps engineers. Many companies prefer those who have a master's degree and at least three to five years of work experience in this field. HTTP, HTML, CSS, SSL, XML, Linux, Java, Amazon Web Services (AWS), NoSQL technologies, DNS, and web app development.

DevOps Engineer Roles and Responsibilities

DevOps engineers work full time. They are responsible for the production and continuing maintenance of a software application platform.

Below are some roles, responsibilities, and skills which are expected from DevOps engineers, such as:

- Manage projects effectively through an open standard based platform.
- Increases project visibility through traceability.
- Improve quality and reduce the development cost with collaboration.
- DevOps should have the soft skill of problem solver and a quick learner.
- Analyze, design, and evaluate automation scripts and systems.
- Able to perform system troubleshooting and problem-solving across the platform and application domains.
- Ensuring the critical resolution of system issues by using the best cloud security solution services.