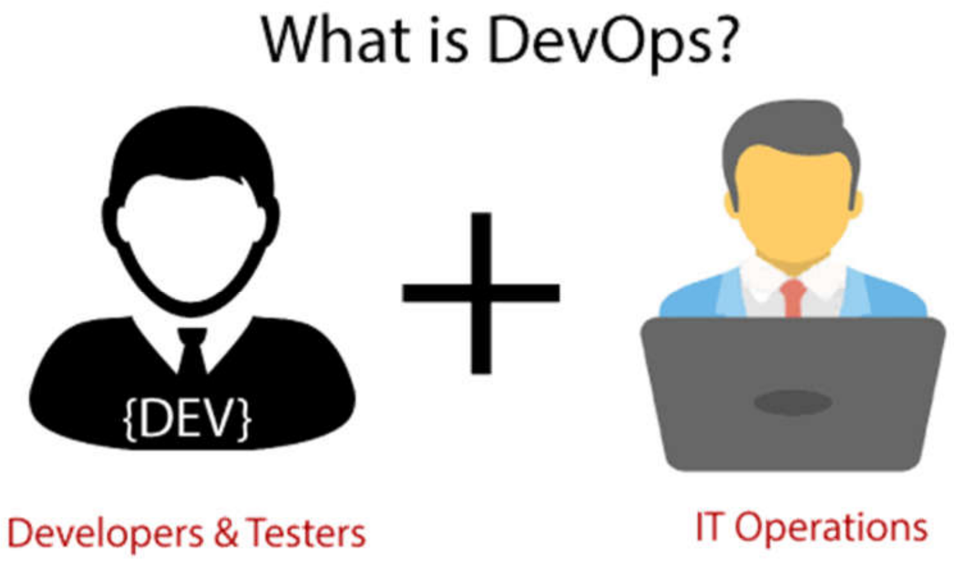
The DevOps is the combination of two words, one is **Development** and other is **Operations**. It is a culture to promote the development and operation process collectively.

The DevOps tutorial will help you to learn DevOps basics and provide depth knowledge of various DevOps tools such as **Git, Ansible, Docker, Puppet, Jenkins, Chef, Nagios**, and **Kubernetes**.

What is DevOps?

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to **testing, deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

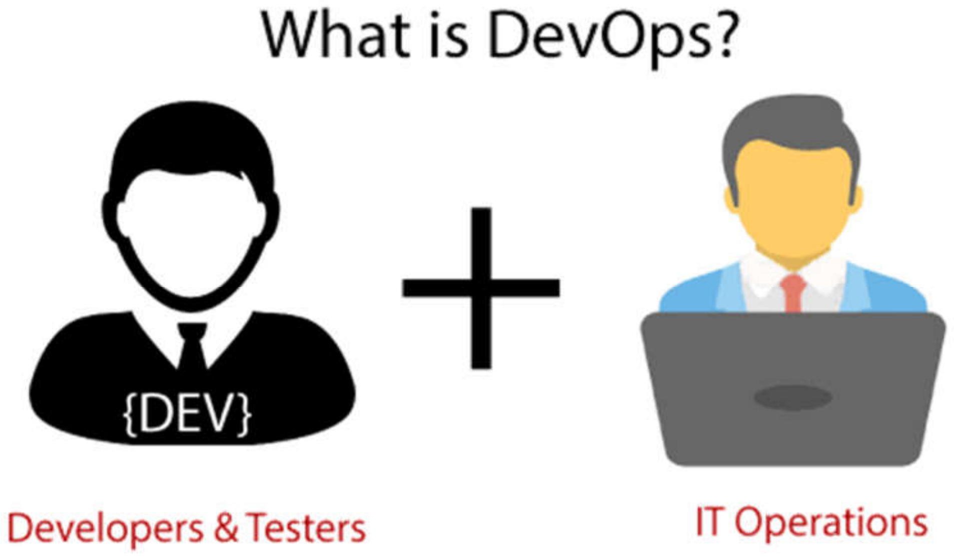


DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way. The DevOps is the combination of two words, one is **Development** and other is **Operations**. It is a culture to promote the development and operation process collectively.

The DevOps tutorial will help you to learn DevOps basics and provide depth knowledge of various DevOps tools such as **Git, Ansible, Docker, Puppet, Jenkins, Chef, Nagios**, and **Kubernetes**.

What is DevOps?

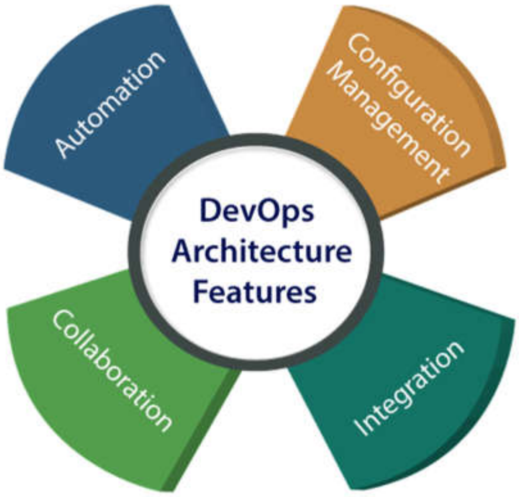
The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to **testing, deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.



DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way.

DevOps Architecture Features

Here are some key features of DevOps architecture, such as:



1) Automation

Automation can reduce time consumption, especially during the testing and deployment phase. The productivity increases, and releases are made quicker by automation. This will lead in catching bugs quickly so that it can be fixed easily. For contiguous delivery, each code is defined through automated tests, cloud-based services, and builds. This promotes production using automated deploys.

2) Collaboration

The Development and Operations team collaborates as a DevOps team, which improves the cultural model as the teams become more productive with their productivity, which strengthens accountability and ownership. The teams share their responsibilities and work closely in sync, which in turn makes the deployment to production faster.

3) Integration

Applications need to be integrated with other components in the environment. The integration phase is where the existing code is combined with new functionality and then tested. Continuous integration and testing enable continuous development. The frequency in the releases and micro-services leads to significant operational challenges. To overcome such problems, continuous integration and delivery are implemented to deliver in a **quicker, safer**, and **reliable manner**.

4) Configuration management

It ensures the application to interact with only those resources that are concerned with the environment in which it runs. The configuration files are not created where the external configuration to the application is separated from the source code. The configuration file can be written during deployment, or they can be loaded at the run time, depending on the environment in which it is running.

DevOps Advantages and Disadvantages

Here are some advantages and disadvantages that DevOps can have for business, such as:

Advantages

- DevOps is an excellent approach for quick development and deployment of applications.
- It responds faster to the market changes to improve business growth.

- DevOps escalate business profit by decreasing software delivery time and transportation costs.
- DevOps clears the descriptive process, which gives clarity on product development and delivery.
- It improves customer experience and satisfaction.
- DevOps simplifies collaboration and places all tools in the cloud for customers to access.
- DevOps means collective responsibility, which leads to better team engagement and productivity.

Disadvantages

- DevOps professional or expert's developers are less available.
- Developing with DevOps is so expensive.
- Adopting new DevOps technology into the industries is hard to manage in short time.
- Lack of DevOps knowledge can be a problem in the continuous integration of automation projects.

Prerequisite

To learn DevOps, you should have basic knowledge of **Linux**, and at least one **Scripting language**.

-=-

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

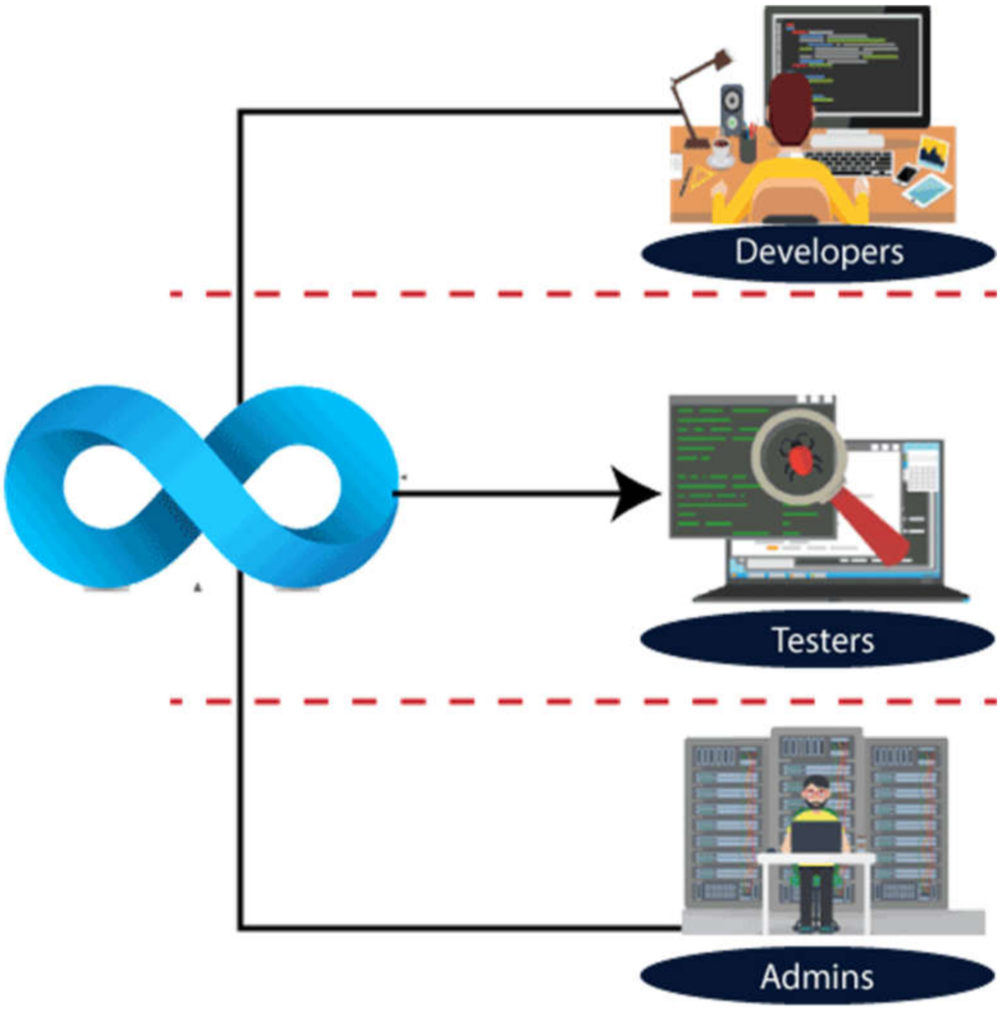
Nagios: It notified people when infrastructure and related service go down. Nagios is a tool for this purpose, which helps the DevOps team to find and correct problems.

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.

DevOps Engineers Salary

The DevOps Engineers salary estimates are based on two reports of salaries, wages, bonuses, and hourly pay.

Here is a list of DevOps engineers salary according to the most recent DevOps engineer salary report, such as:

Salary	Area	Experiences	Company
₹4,20,000	A DevOps engineer in the **Bengaluru** area reported making **₹4,20,000** per year.	0-year experience	Private
₹3,00,000	A DevOps engineer in the **Bengaluru** area reported making **₹3,00,000** per year.	1-2 year experience	Public
₹5,00,000	A DevOps engineer in the **Hyderabad** area reported making **₹5,00,000** per year.	3-4 year experiences	Public
₹3,00,000	A DevOps engineer in the **Hyderabad** area reported making **₹3,00,000** per year.	3-4 year experience	Private
₹6,00,000	An Azure DevOps engineer in the **Chennai** area reported making **₹6,00,000** per year.	1-2 year experience	Public
₹4,80,000	A DevOps engineer in the **Pune** area reported making **₹4,80,000** per year.	3-4 year experience	Public

₹11,06,561	A DevOps engineer in the **New Delhi** area reported making **₹11,06,561** per year.	3-4 year experience	Private

DevOps Pipeline

A pipeline in software engineering team is a set of automated processes which allows DevOps professionals and developer to reliably and efficiently compile, build, and deploy their code to their production compute platforms.

The most common components of a pipeline in DevOps are build automation or continuous integration, test automation, and deployment automation.

A pipeline consists of a set of tools which are classified into the following categories such as:

- Source control
- Build tools
- Containerization
- Configuration management
- Monitoring

Continuous Integration Pipeline

Continuous integration (CI) is a practice in which developers can check their code into a version-controlled repository several times per day. Automated build pipelines are triggered by these checks which allows fast and easy to locate error detection.

Some significant benefits of CI are:

- Small changes are easy to integrate into large codebases.
- More comfortable for other team members to see what you have been working.
- Fewer integration issues allowing rapid code delivery.
- Bugs are identified early, making them easier to fix, resulting in less debugging work.

Continuous Delivery Pipeline

Continuous delivery (CD) is the process that allows operation engineers and developers to deliver bug fixes, features, and configuration change into production reliably, quickly, and sustainably. Continuous delivery offers the benefits of code delivery pipelines, which are carried out that can be performed on demand.

Some significant benefits of the CD are:

- Faster bug fixes and features delivery.
- CD allows the team to work on features and bug fixes in small batches, which means user feedback received much quicker. It reduces the overall time and cost of the project.

DevOps Methodology

We have a demonstrated methodology that takes an approach to cloud adoption. It accounts for all the factors required for successful approval such as people, process, and technology, resulting in a focus on the following critical consideration:

- **The Teams:** Mission or project and cloud management.
- **Connectivity:** Public, on-premise, and hybrid cloud network access.
- **Automation:** Infrastructure as code, scripting the orchestration and deployment of resources.
- **On-boarding Process:** How the project gets started in the cloud.
- **Project Environment:** TEST, DEV, PROD (identical deployment, testing, and production).
- **Shared Services:** Common capabilities provided by the enterprise.
- **Naming Conventions:** Vital aspect to track resource utilization and billing.
- **Defining Standards Role across the Teams:** Permissions to access resources by job function.

Azure DevOps

Azure DevOps is also known as Microsoft visual studio team services (VSTS). It is a set of collaborative development tools built for the cloud. VSTS was commonly used as a standalone term, and Azure DevOps is a platform which is made up of a few different products, such as:

- Azure Test Plans
- Azure Boards

- Azure Repos
- Azure Pipeline
- Azure Artifacts

Azure DevOps is everything that needs to turn an idea into a working piece software. You can plan a project with azure tools.

The azure pipeline is the CI component of azure DevOps. The azure pipeline is Microsoft's cloud-native continuous integration server, which allows teams to continuously build, test, and deploy all from the cloud. An azure pipeline can connect to any number of source code repositories such as Azure Repos, GitHub, Tests, to grab code and artifacts for application delivery.

Azure DevOps Server

Azure DevOps Server is a Microsoft product that provides version control, requirements management, reporting, lab management, project management, testing, automated builds, and release management capabilities. It covers the entire application of lifecycle and enables DevOps capabilities.

Azure DevOps can be used as a back-end to the numerous integrated development environments, but it is modified for Microsoft visual studio and eclipse on all platforms.

Azure DevOps Services

Microsoft announced the release of the software as a service offering of visual studio on the Microsoft Azure platform at the time Microsoft called it a visual studio online.

Microsoft offers visual studio, basic, and stakeholder subscriber access levels for the Azure DevOps services. The basic plan is free of cost for up to five users. Users with a visual studio subscription can be added to a project with no additional charge.

AWS DevOps

AWS is the best cloud service provider, and DevOps is the implementation of the software development lifecycle.

Here are some reasons which make AWS DevOps a highly popular combination, such as:

- AWS CloudFormation
- AWS EC2

- AWS CloudWatch
- AWS CodePipeline

Let's see all of these by one in brief such as:

AWS CloudFormation

DevOps team is required to create and release cloud instances and services more frequently in comparison to development teams. Templates of AWS resources such as EC2 instances, ECS containers, and S3 storage buckets let you set up the entire stack without having to bring everything together.

AWS EC2

You can run containers inside EC2 instances. Hence you can leverage the AWS security and management features.

AWS CloudWatch

This monitoring tool tracks every resource that AWS has to offer. It makes it easy to use third-party tools for monitoring such as sumo logic etc.

AWS CodePipeline

Code Pipeline is an essential feature from AWS, which highly simplifies the way you manage your CI/CD toolset. It integrates with tools such as **Jenkins, GitHub**, and CodeDeploy that enable you to visually control the flow of app updates from build to production.

DevOps vs Agile

DevOps and Agile are the two software development methodologies with similar aims, getting the end-product as quickly and efficiently as possible. While many organizations are hoping to employ these practices, there is often some confusion between both methodologies.

What does each methodology enclose? Where do they overlap? Can they work together, or should we choose one over the other?

Before move further, take a glance at DevOps and Agile.

What is DevOps?

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to **testing, deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way.

DevOps helps to increase organization speed to deliver applications and services. It also allows organizations to serve their customers better and compete more strongly in the market.

DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.

DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, **quality**, and **speed** of the application delivery has improved to a great extent.

DevOps is nothing but a practice or methodology of making "**Developers**" and "**Operations**" folks work together. DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices in the context of a system-oriented approach.

What is Agile?

The Agile involves continuous iteration of development and testing in the **SDLC** process. Both development and testing activities are concurrent, unlike the waterfall model. This software development method emphasizes on incremental, iterative, and evolutionary development.

It breaks the product into small pieces and integrates them for final testing. It can be implemented in many ways, such as **Kanban, XP, Scrum**, etc.

The Agile software development focus on the four core values, such as:

- Working software over comprehensive documentation.
- Responded to change over following a plan.
- Customer collaboration over contract negotiation.
- Individual and team interaction over the process and tools.

Below are some essential differences between the DevOps and Agile:

Parameter	DevOps	Agile
Definition	DevOps is a practice of bringing development and operation teams together.	Agile refers to the continuous iterative approach, which focuses on collaboration, customer feedback, small, and rapid releases.
Purpose	DevOps purpose is to manage end to end engineering processes.	The agile purpose is to manage complex projects.
Task	It focuses on constant testing and delivery.	It focuses on constant changes.
Team size	It has a large team size as it involves all the stack holders.	It has a small team size. As smaller is the team, the fewer people work on it so that they can move faster.
Team skillset	The DevOps divides and spreads the skill set	The Agile development emphasizes training all

	between development and the operation team.	team members to have a wide variety of similar and equal skills.
Implementation	DevOps is focused on collaboration, so it does not have any commonly accepted framework.	Agile can implement within a range of tactical frameworks such as **safe, scrum**, and **sprint**.
Duration	The ideal goal is to deliver the code to production daily or every few hours.	Agile development is managed in units of sprints. So this time is much less than a month for each sprint.
Target areas	End to End business solution and fast delivery.	Software development.
Feedback	Feedback comes from the internal team.	In Agile, feedback is coming from the customer.
Shift left principle	It supports both variations left and right.	It supports only shift left.
Focus	DevOps focuses on operational and business readiness.	Agile focuses on functional and non-functional readiness.
Importance	In DevOps, developing, testing, and implementation all are equally important.	Developing software is inherent to Agile.
Quality	DevOps contributes to creating better quality with automation and early bug removal. Developers need to follow Coding and best Architectural practices to maintain quality standards.	The Agile produces better applications suites with the desired requirements. It can quickly adapt according to the changes made on time during the project life.

Tools	**Puppet, Chef, AWS, Ansible**, and team City OpenStack are popular DevOps tools.	**Bugzilla, Kanboard, JIRA** are some popular Agile tools.
Automation	Automation is the primary goal of DevOps. It works on the principle of maximizing efficiency when deploying software.	Agile does not emphasize on the automation.
Communication	DevOps communication involves specs and design documents. It is essential for the operational team to fully understand the software release and its network implications for the enough running the deployment process.	Scrum is the most common method of implementing Agile software development. Scrum meeting is carried out daily.
Documentation	In the DevOps, the process documentation is foremost because it will send the software to an operational team for deployment. Automation minimizes the impact of insufficient documentation. However, in the development of sophisticated software, it's difficult to transfer all the knowledge required.	The agile method gives priority to the working system over complete documentation. It is ideal when you are flexible and responsive. However, it can harm when you are trying to turn things over to another team for deployment.