



Share on your Social Media



DevOps Tutorial

Published On: August 16, 2024

DevOps Tutorial

To overcome these obstacles, DevOps promotes integration, cooperation, and communication between the development and operations teams. With this comprehensive DevOps tutorial, you will understand how it works for the organization and how you can upskill for a bright career.

[Download DevOps Tutorial PDF](#)

Introduction to DevOps

DevOps strives to accomplish important goals, including accelerating delivery, improving reliability, and stimulating creativity by fostering a cultural shift and making use of cutting-edge tools and methods. We cover the following concepts in this DevOps tutorial:

- Overview of DevOps

Featured Articles

Want to know more about becoming an expert in IT?

[Click Here to Get Started](#)

100% Placement Assurance

AUTHORISED CERTIFICATION PARTNER

IBI

Related Courses at SLA

Related Posts



Tableau Developer Salary in Chennai

Published On: October 12, 2024

Introduction A Tableau Developer designs, develops, and maintains dashboards and visualizations using Tableau software. Key...

Quick Enquiry

- DevOps Model
- DevOps Lifecycle
- DevOps Tools
- DevOps vs Agile

Overview of DevOps

Software Development and Operations are the two words that make up the term DevOps. As a result, the full application lifecycle, from development to testing, deployment, and operations, can be managed by a single team.

System administrators, software developers, and quality assurance (QA) engineers can become less disconnected with the aid of DevOps.

The entire process of gathering data, configuring the infrastructure, developing, testing, deploying, continuously monitoring, and obtaining end-user feedback falls on the shoulders of DevOps engineers.

Features of DevOps

DevOps has the following features:

- Faster development and delivery
- Improved collaboration
- Enhanced quality of products
- Increased automation in all stages.
- Improved scalability as per the project needs
- Enhanced security
- Improved resource utilization
- Enhanced client satisfaction

DevOps Interview Questions and Answers

DevOps Model

DevOps seeks to increase the standard and reliability of software releases while reducing the duration of the software development lifecycle.



VMware Tutorial for Cloud Computing Aspirants

Published On: October 12, 2024

VMware Tutorial for Cloud Computing Aspirants VMware software allows you to run a virtual machine...



VBA Macros Tutorial for Beginners

Published On: October 10, 2024

VBA Macros Tutorial for Beginners VBA macros are programs that automate repetitive operations in Microsoft...



VB.Net Tutorial for Beginners

Published On: October 10, 2024

VB.Net Tutorial for Beginners Visual Basic is an easy-to-learn programming language that is type-safe. Get...

Delivery Pipeline

The various phases that software goes through before being launched into production are represented by the pipeline. Usually, these phases could consist of:

Build: The process of compiling and packaging software code into a deployable unit.

Test: This phase involves thoroughly testing the software to make sure it performs as planned and finds any issues.

Release: The phase in which the program is made available to end users in production.

Feedback Loop

The loop shows how knowledge and insights gained from the production environment are incorporated back into the pipeline's early phases.

Future software releases and the software development process itself can benefit from this input.

How Does DevOps Operate?

DevOps will eliminate the "isolated" environments between the operations and development teams. These two teams will frequently collaborate throughout the whole application lifetime, from development and testing to deployment and operations, and they will acquire various skills that are not only related to one job.

- **Traditional:** These teams utilized protocols to automate manual, labor-intensive, slow operations in the past.
- **DevOps:** They use a technology stack and tools that make app development and operation fast, reliable, and easy.
- These tools not only help engineers accomplish tasks, such as providing infrastructure or delivering code,

autonomously that would have otherwise required support from other teams, but they also accelerate a team's velocity.

Under different DevOps models, teams in charge of quality assurance and security may also collaborate more closely with development and operations throughout the lifecycle of an application.

When security is a primary concern for every member of a DevOps team, the term "DevSecOps" is employed.

[Download DevOps Syllabus PDF](#)

DevOps Lifecycle

Bringing together developers, testers, system administrators, and quality assurance personnel under one roof is the main goal of DevOps. Thus, collectively, these individuals are referred to as DevOps engineers.

- **Code:** Developers write code on any platform as the initial stage in the DevOps life cycle.
- **Build:** Depending on the language they are using, developers construct a version of their software in any extension.
- **Test:** Any DevOps automation tool such as must be used to automate the testing process for DevOps to be successful. Mineral Selenium
- **Release:** A procedure used to coordinate, organize, schedule, and oversee the construction in various settings following testing but before deployment
- **Deploy:** This stage prepares all of the application's code files and artifacts and deploys/runs them on the server.
- **Operate:** Following deployment, clients use the program in real-world situations.
- **Monitor:** This stage provides vital information that promotes maximum performance and service uptime.

- **Plan:** Information from the monitoring stage is gathered during the planning stage, and modifications are implemented for improved performance based on input.

DevOps Lifecycle Stages

There are various DevOps life cycle stages that support the standard software development life cycle (SDLC):

- Continuous Development
- Continuous Integration
- Continuous Testing
- Continuous Monitoring
- Continuous Feedback
- Continuous Deployment
- Continuous Operation

In essence, these phases are what make up the DevOps objective.

Continuous Development

In this stage of DevOps, software is built constantly, whereas in the waterfall model, our software product is divided into many portions or sub-parts to shorten the development cycles.

Tools for this stage: We can use GIT to maintain various code versions as we code and build at this point. One dependable tool that we can use to build/package the code into an executable file is Maven.

Continuous Integration

At this point, new functionality that our code supports is regularly integrated with the current code.

The present code must “continuously” be integrated with the most recent version as long as continuous development is ongoing, and the modified code must guarantee that there are no defects in the current environment for it to function properly.

Tools for this stage: Jenkins is a tool utilized for continuous integration. Here, we may generate the build and deploy it on the test or production server using the most recent code that we can grab from the GIT source.

Continuous Testing

Using a variety of automated technologies, our generated software is continually tested to find defects during the continuous testing phase.

Tools for this stage: There are a variety of automated tools available for use in quality assurance and testing.

One popular tool for automation testing is Selenium, which enables QAs to test codes concurrently to make sure that there are no errors, inconsistencies, or software defects.

Continuous Monitoring

It is an essential step in the DevOps life cycle since it gives us vital data that enables us to guarantee optimal performance and service availability.

Reliable monitoring technologies provide the operations team with the information they need to identify and address application faults and defects.

Tools for this stage: Numerous tools are used to monitor the application, including Sensu, Splunk, ELK Stack, and Nagios.

- They assist us in monitoring the health and functionality of our servers and apps.
- During the continuous development phase, the development team receives any significant issues found by these tools and addresses them.

Continuous Feedback

The results of the software's activities are analyzed to continuously improve the application

development. This is accomplished by positioning the crucial stage of ongoing input between the construction of the present software program and its operations.

Continuous feedback is the key component of DevOps because it eliminates the needless processes needed to take a software application out of development, use it to identify problems, and then create an improved version. It eliminates any potential app efficiency and lowers the number of potential users.

Continuous Deployment

The code is pushed to the production servers at this phase. Furthermore, it is imperative to guarantee that the code is applied accurately across all servers.

Continuous deployment of new code is achieved through the use of configuration management technologies, which are critical to the frequent and efficient completion of tasks.

Tools for this stage: Here are several well-known tools that are utilized in this stage: SaltStack, Chef, Puppet, Ansible, and Puppet. Tools for containerization are also crucial during the deployment stage. Docker and Vagrant are two well-liked tools for this purpose.

Consistency in the development, staging, testing, and production environments is facilitated by these tools. They also aid in gently scaling up and down instances.

Continuous Operations

The foundation of all DevOps operations is continuous release process automation, which enables the company to continuously shorten its time to market.

The conversation makes it abundantly evident that

maintaining consistency is essential to DevOps to eliminate stages that frequently impede development.

We can enhance the number of potential clients and improve the efficiency of any software product by implementing DevOps.

Devops Engineer Salary

DevOps Tools

Here are some of the most widely used DevOps tools, along with a brief description:

Ansible

Ansible is an open-source IT engine that automates the provisioning of cloud infrastructure, the orchestration of intraservices, and the deployment of applications. It facilitates the scalability of automation and increases efficiency for DevOps teams.

Features of Ansible

- It gets rid of tedious work.
- Applications for open source deployment are simple to use.
- It aids in keeping the software development process simple.
- It expedites the development process and handles intricate deployments.

Chef

The cloud-based Chef software is available as open source. This technology develops fundamental building blocks, like recipes and cookbooks, using Ruby encoding.

The Chef is a tool for automating infrastructure management that helps to eliminate tedious and manual duties. Infrastructure management and automation depend on various building elements,

each of which has a convention inside Chef.

Features of Chef

- It continues to be highly available.
- Multiple cloud environments can be managed using it.
- It creates a domain-specific language using the widely used Ruby language.
- The chef doesn't assume anything about the node's present state. To determine the machine's current condition, it employs its mechanism.

Docker

Distributed programs may be developed, deployed, and run across several machines thanks to a high-end DevOps tool called Docker.

It also makes it easier to quickly assemble apps from their component pieces and is suitable for container management in general.

Features of Docker

- It increases productivity.
- It permits covert storage within the swarm.
- It speeds up and improves the comfort of system configuration.
- It offers containers, which are utilized to run the program in a segregated setting.
- It directs incoming requests for open ports on nodes that are available to an active container.
- Because of this functionality, the connection is maintained even when the node is not actively performing any tasks.

Git

Git is a distributed version management system that is freely available to all users and is open-source. It is made to work quickly and effectively on little to large jobs.

It was created to help programmers coordinate their work. You can collaborate and keep track of your team members in the same workspace thanks to version control.

For the DevOps tool, it serves as a vital distributed version control system.

Features of Git

- Git has great scalability.
- The pull request is supported.
- The utility is open source and free.
- It makes the releasing cycle faster.
- It makes dispersed development possible.
- It works quickly and securely to finish the job at hand.

Jenkins

Jenkins is a DevOps tool that monitors the fulfillment of recurring tasks. Jenkins is a program that makes continuous integration easier.

Jenkins will be installed on the server that is used for the central build. Early problem detection makes it easier to integrate project alterations more successfully.

Features of Jenkins

- Jenkins expands the automation's scope.
- Through a web interface, setup and configuration are simple.
- It can increase concurrency by dividing up the jobs among several machines.
- Both continuous delivery and continuous integration are supported.
- With its 400 plugins, you may theoretically develop and test any project.
- It features an integrated GUI tool for simple upgrades and requires little upkeep.

Nagios

Nagios can identify the faults and use server, network, infrastructure, and log monitoring technologies to fix them.

Features of Nagios

- It offers thorough server and desktop operating system monitoring.
- It offers comprehensive Java Management Extension monitoring.
- Identification of bottlenecks and optimization of bandwidth are facilitated by the network analyzer.
- Monitoring elements including services, apps, operating systems, and network protocols is beneficial.

Puppet

Puppet enables the rapid and frequent supply and distribution of technological improvements. It enables complete infrastructure management as code without expanding the team.

Features of Puppet

- Context-aware reporting in real time.
- Manage and model the environment as a whole.
- Outlined and constantly implemented infrastructure.
- Conflict detection and resolution in the desired state.
- It facilitates the developer's speedy delivery of excellent software.
- It removes the need for manual labor in the software delivery procedure.
- It examines and documents packages that are operating throughout the infrastructure.

Selenium

A framework for portable software testing of web applications is called Selenium. It offers a user-friendly interface for creating automated tests.

Features of Selenium

- The utility is open source and free.
- It facilitates cross-platform testing on both iOS and Android.
- Developing a keyword-driven WebDriver framework is a simple task.
- It generates reliable browser-based tests and suites for regression automation.

[Devops Training](#)

DevOps vs. Agile

Agile and DevOps are two software development approaches that share the goal of producing the final product as fast and effectively as feasible.

DevOps: Software Development and Operations are the two words that make up the term DevOps. All that DevOps is a process or approach to getting “developers” and “operations” people to collaborate.

Agile: Continuous testing and development iterations are part of the SDLC process with Agile. The contemporaneous nature of development and testing sets it apart from the waterfall paradigm.

It disassembles the product into manageable chunks and combines them for last-minute testing. It can be applied in a variety of ways, including Scrum, XP, and Kanban.

Agile software development places a strong emphasis on the following four basic values:

- Functional software is preferred above extensive documentation.
- Rather than following the plan, they adjusted to change.
- Customer cooperation in place of contract drafting.
- Individual and group communication

regarding the procedure and resources.

The following are some key distinctions between Agile and DevOps:

Factors	DevOps	Agile
Definition	Integration of development and operations teams is known as DevOps.	The term "agile" describes the iterative, continuous strategy that prioritizes customer feedback, small, frequent releases, and teamwork.
Purpose	The goal of DevOps is to oversee complete technical processes.	The management of complex projects is the aim of agile.
Size	Since all of the stack holders are involved, the team size is high.	The team is not very large. To proceed more quickly, fewer people work on a smaller team.
		Agile development

Skills	The operation and development teams share and split the skill set thanks to DevOps.	places a strong emphasis on equipping each team member with a diverse range of comparable and equal skills.
Implementation	Since DevOps emphasizes teamwork, it lacks a well recognized structure.	Agile can be used with a variety of tactical frameworks, including sprint, scrum, and safe.
Delivery Time	Delivering the code to production every day or every few hours is the best scenario.	Sprints are used as management units in agile development. Thus, each sprint takes far less time than a month in this case.
Significance	Development, testing, and implementation are all equally crucial in DevOps.	Agile development is ingrained in the process.

Tools	Popular DevOps tools include Chef, AWS, Ansible, Puppet, and Team City OpenStack.	JIRA, Kanboard, and Bugzilla are a few well-known Agile tools.
Automation	Automating tasks is DevOps' main objective. It operates under the tenet of optimizing software deployment efficiency.	Automation is not emphasized in agile.

Conclusion

Through this DevOps tutorial, we have covered the fundamental concepts of DevOps. Thrive in your career by enrolling in our DevOps tutorial.

Devops Training in Chennai

Share on your Social Media



Navigation

- About Us
- Blog Posts
- Careers
- Contact
- Placement Training

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.

Landmark: Karnataka Bank Building

Phone: [+91 86818 84318](tel:+918681884318)

Email: enquiry@softlogicsys.in

Map: [Google Maps Link](#)

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai – 600 130.

Landmark: Adj. to AGS Cinemas

Phone: [+91 89256 88858](tel:+918925688858)

Email: info@softlogicsys.in

Map: [Google Maps Link](#)

Courses

Python

Software Testing

Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

Corporate Training

Hire With Us

Job Seekers

SLA's Recently Placed Students

Reviews

Sitemap

Important Links

Disclaimer

Privacy Policy

Terms and Conditions

Social Media Links



Review Sources

Google

Trustpilot

Glassdoor

Mouthshut

Sulekha

Justdial

Ambitionbox

Indeed

Software Suggest

Sitejabber