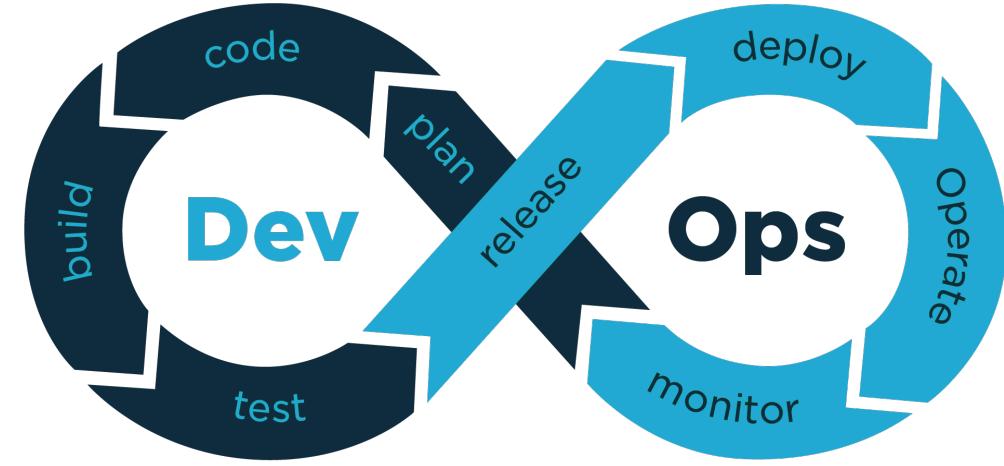


# DEVOPS

C O M P 3 1 0 4

DEVOPS



# More on what is DevOps ?

DevOps is a set of practices that works to automate and integrate the processes between software development and IT teams, so they can build, test, and release software faster and more reliably.

The term DevOps was formed by combining the words “development” and “operations” and signifies a cultural shift that bridges the gap between development and operation teams, which historically functioned in siloes.

## Where Did DevOps Come From?

**Enterprise systems management (ESM).** Many of the people involved in the initial definition of DevOps were system administrators. These operations experts brought key ESM best practices to DevOps, including configuration management, system monitoring, automated provisioning, and the toolchain approach.

**Agile development.** “DevOps can be interpreted as an outgrowth of Agile—agile software development prescribes close collaboration of customers, product management, developers, and (sometimes) QA to fill in the gaps and rapidly iterate towards a better product Agile principles beyond the boundaries of the code to the entire delivered service.”

[Web Reference Link](#)

# What problems led to the creation of DevOps?

- Set expectations and priorities and the fundamental beliefs that guide them.
- Collaborate both within and between teams on problem solving.
- Automate common and repetitive processes to free up time for higher-level work.
- Integrate feedback into the work, measuring everything that is moved into production.
- Share the data with everyone involved to foster a more effective culture of working well together across different skills and specialized knowledge.

# WHO IS A DEVOPS ENGINEER?

- DevOps Engineer is somebody who understands the Software Development Lifecycle and has the outright understanding of various automation tools for developing digital pipelines (CI/ CD pipelines).
- DevOps Engineer works with developers and the IT staff to oversee the code releases.
- They are either developers who get interested in deployment and network operations or sysadmins who have a passion for scripting and coding and move into the development side where they can improve the planning of test and deployment.

## Job Roles and Responsibilities for a DevOps Engineer

- **DevOps Evangelist** – The principal officer (leader) responsible for implementing DevOps
- **Release Manager** – The one releasing new features & ensuring post-release product stability
- **Automation Expert** – The guy responsible for achieving automation & orchestration of tools
- **Software Developer/ Tester** – The one who develops the code and tests it
- **Quality Assurance** – The one who ensures the quality of the product confirms to its requirement
- **Security Engineer** – The one always monitoring the product's security & health

# DEVOPS: TOOLS & TECHNIQUE

## **Planning Tools**

Collaboration and communication are essential tenets of the DevOps culture.

In accordance with Agile software development practices, the DevOps environment requires tools that empower sprint planning for iterations, release management features, and issue tracking.

### **DevOps planning tool to consider: JIRA**

JIRA by Atlassian is an Agile-friendly planning tool.

JIRA supports sprint planning with out-of-the-box or customizable Kanban and Scrum workflows.

JIRA software integrates with complex enterprise platforms and is well-suited for planning DevOps workflows that require continuous feedback loops, collaboration with multiple project teams, and open communication.

**Other DevOps planning tools:** Redmine, Trac, Rally.

# DEVOPS: TOOLS & TECHNIQUE

- **Building Tools**

A “build” is a version of software code (code build). It can also be used as a verb to describe the process of writing source code for a specific stage of development before the next step in the cycle. Developers use build tools to manage this work.

***DevOps build tool to consider: Apache Maven***

Apache Maven is a tool for Java-based development.

It provides a uniform system and a set of plugins for code builds, with project dependencies and development guidelines that support DevOps technical practices such as build automation.

Other DevOps build tools			
Apache Ant	Gulp	NAnt	Travis CI
Broccoli	Hudson	Packer	UrbanCode Build
Buildbot	Jam	QuickBuild	Visual Build
BuildMaster	LuntBuild	Rake	Visual Studio
FinalBuilder	Make	sbt	
Buildr	Meister	SSH	
CMake	Microsoft Build Engine (MSBuild)	TeamCity	

# DEVOPS: TOOLS & TECHNIQUE

## Integration Tools

- The practice of automating code builds and tests, merging them, and checking them into shared repositories in real time for continuous integration and delivery is a core DevOps convention.
- Integration tools in the DevOps toolchain that optimize code builds, empower automation, and help keep systems working are the key to high-velocity deployment.

### DevOps CI/CD tool to consider: Jenkins

- ❖ Jenkins is an open source automation server that provides a plugin architecture to support continuous integration and delivery.
- ❖ It integrates with a variety of software tools in the CI/CD toolchain and distributes work across multiple platforms.

Other DevOps CI/CD tools			
Apache ActiveMQ	Continua CI	Hudson	Travis CI
Bamboo	Continuum	Shippable	
Circle CI	CruiseControl	Snap CI	
Codeship	Gump	Solano CI	

# DEVOPS: TOOLS & TECHNIQUE

- Manual work in progress hinders continuous integration and delivery, a core DevOps technical practice.
- Using tools to automate functional and non-functional software tests and configuration management frees up the resources required to achieve high-velocity deployments.
- Version control is another core DevOps technical practice.
- These systems manage and track the changes to software code as revisions are released. Also known as **source control management (SCM)**, version control keeps track of code modifications in a database and helps teams manage continuous integration.

Provisioning, version control, and configuration management tools			
Ansible	Consul	ISPW	Rudder
AWS CloudFormation	Crucible	Kallithea	Rundeck
Bcfg2	Fisheye	Mercurial	Salt
BladeLogic	Foreman	Packer	Stash
CFEngine	Gerrit	PalletOps	Subversion
Chef	Git	Perforce	Terraform
Cobbler	GitLab	Puppet	

DevOps software testing tools			
Appium	HP UFT	Mocha	Selenium
Bosun	IBM Rational Quality Manager	Parasoft Environment Manager	SonarQube
Codacy	Jasmine	Parasoft SOAtest	SpecFlow
Cucumber	JMeter	Protractor	TestComplete
FitNesse	JUnit	Pytest	TestNG
Gatling	Karma	QUnit	Tosca
Gauntlet	Locust	Sahi	

# DEVOPS: TOOLS & TECHNIQUE

## Deployment Tools

- Another core DevOps technical practice is rapid, continuous deployment of every CI build.
- Enabling this practice requires managing change, automating builds and tests, and keeping everything in one stable environment for release.
- There is a variety of tools to help DevOps practitioners orchestrate continuous deployment and automate release processes in standardized environments.

## DevOps deployment tool to consider:

### Capistrano

- ❖ Capistrano is a remote server automation and deployment tool primarily used for deploying web applications.
- ❖ It is written in Ruby and can extend to support other coding languages or frameworks with special requirements.

Other DevOps deployment tools			
Automic	Deploybot	JuJu	RapidDeploy
AWS CodeDeploy	Deployer	Nomad	Rundeck
BMC Release	ElectricFlow	Octopus Deploy	SmartFrog
CA Nolio	Go	Otto	XL Deploy
Containership	Google Deployment Manager	Rancher	

# DEVOPS: TOOLS & TECHNIQUE

## Adopting DevOps Architecture

- The trend in DevOps architectural requirements is the use of continuous deployable microservices.
- **Microservices** architecture, in combination with DevOps practices, shortens the time it takes to deploy changes in code or configuration.
- The small, reusable building blocks of code in microservices ensures that the application under development is not affected by the increase in the velocity of deployments.
- **Cloud-based microservice** architecture uses containers to isolate, package, and deploy code at scale.

DevOps container tools			
Apache Mesos	Kubernetes	Packer	Swarm
CloudSlang	Linux Containers (LXC)	Rancher	Tectonic
Containership	Nomad	rkt ("rocket")	
Docker	OpenVZ	Solaris Containers	

# WHAT IS DEVOPS LIFE CYCLE?

Various phases such as continuous development, continuous integration, continuous testing, continuous deployment, and continuous monitoring constitute DevOps Life cycle.

## Continuous Development –

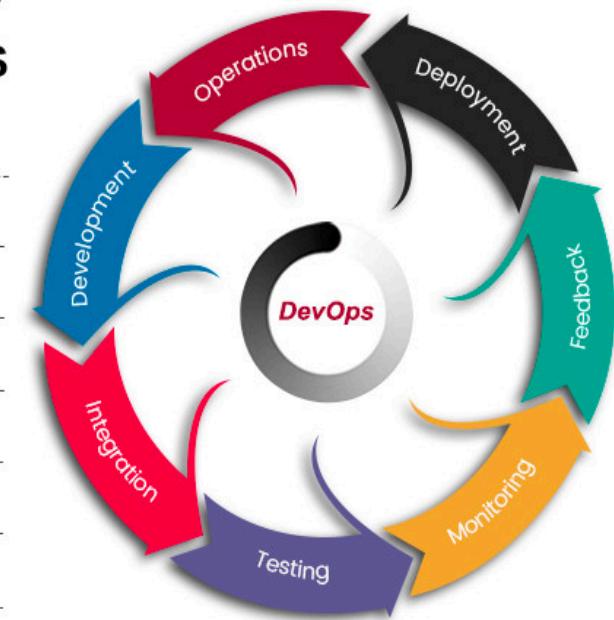
- This phase involves '**planning**' and '**coding**' of the software.
- The vision of the project is decided during the planning phase and the developers begin developing the code for the application.
- There are no DevOps tools that are required for planning, but there are several tools for maintaining the code.
- The code can be written in any language, but it is maintained by using Version Control tools.

- *Git, SVN, Mercurial, CVS, and JIRA.*

- *Ant, Maven, Gradle can be used in this phase for building/ packaging the code into an executable file that can be forwarded to any of the next phases.*

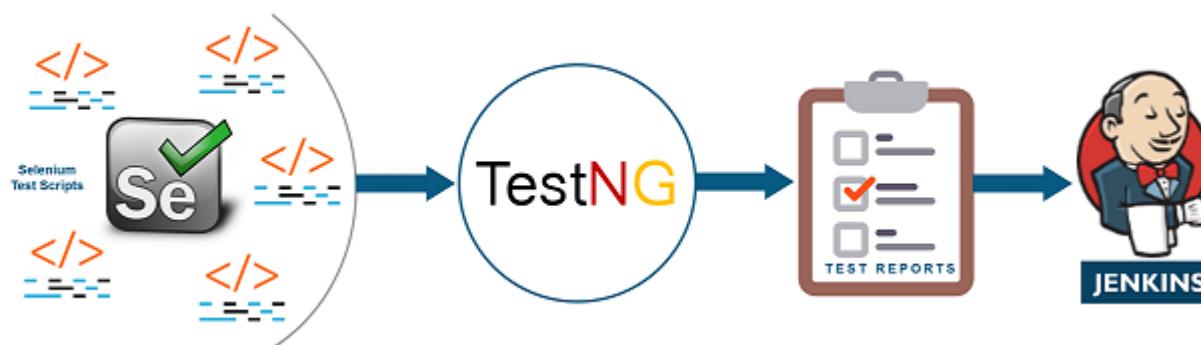
## DEVOPS LIFE CYCLE PHASES

- ✓ Development
- ✓ Integration
- ✓ Testing
- ✓ Monitoring
- ✓ Feedback
- ✓ Deployment
- ✓ Operations



## Continuous Testing –

- This is the stage where the developed software is continuously tested for bugs. For Continuous testing, automation testing tools like **Selenium**, **TestNG**, **JUnit**, etc are used.
- These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there are no flaws in the functionality. In this phase, **Docker Containers** can be used for simulating the test environment.
- Selenium does the automation testing, and the reports are generated by TestNG. This entire testing phase can be automated with the help of a **Continuous Integration tool called Jenkins**.
- Now you can build this code using ant or maven. Once the code is built, it is tested for **User Acceptance Testing (UAT)**. This entire process can be automated using Jenkins.
- Besides that, report generation is a big plus.



## **Continuous Deployment –**

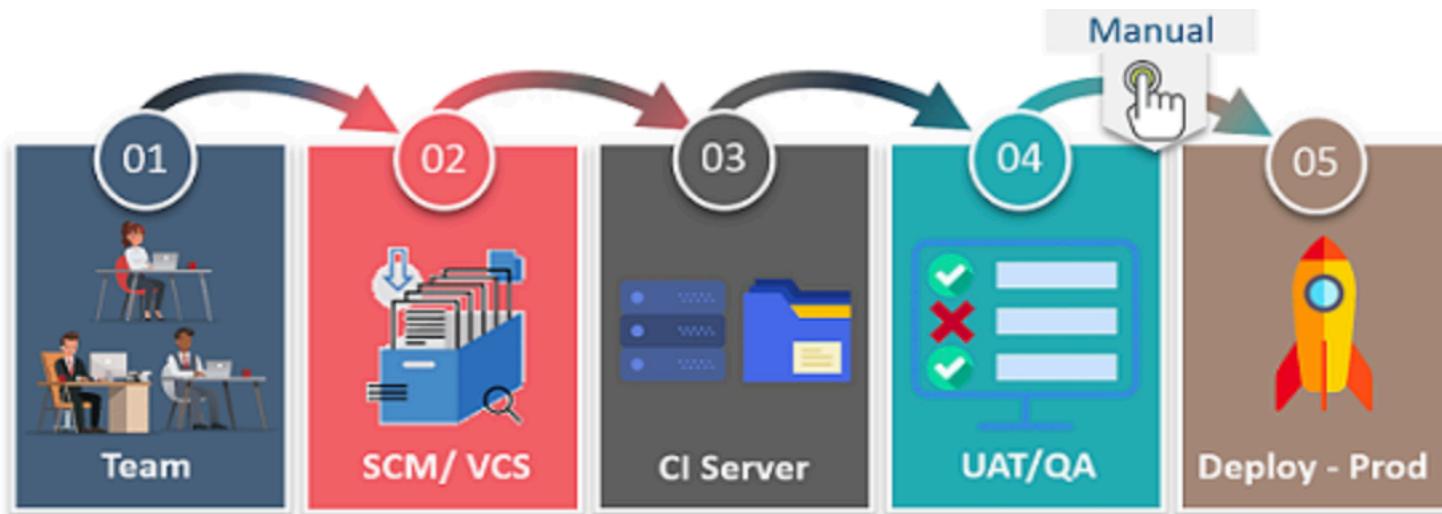
- This is the stage where the code is deployed to the production servers.
- It is also important to ensure that the code is correctly deployed on all the servers.
- Tools help in achieving Continuous Deployment (CD) are as follow.

### **Configuration management**

- Configuration Management is the act of establishing and maintaining consistency in an application's functional requirements and performance.
- Some popular tools are **Puppet, Chef, SaltStack, and Ansible**.

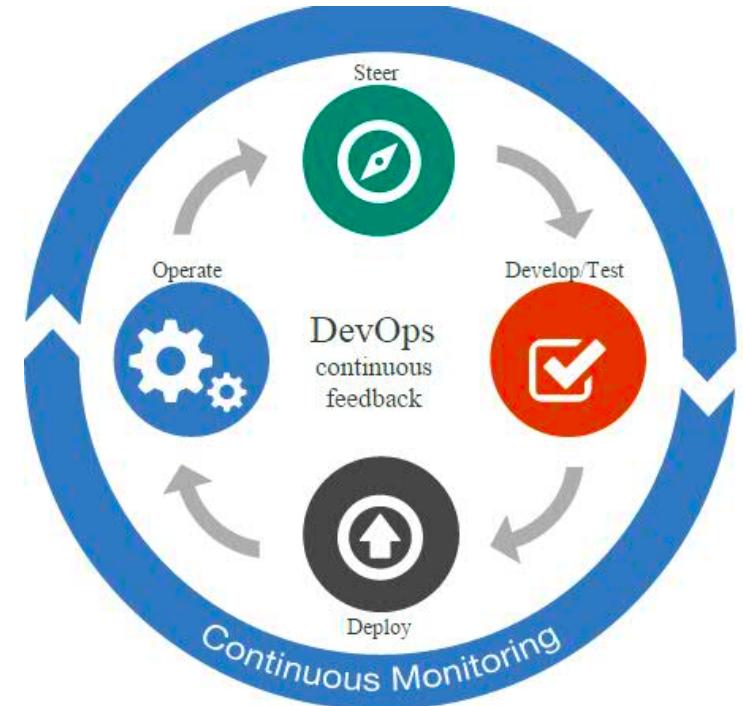
### **Containerization tools.**

- Containerization tools help in maintaining consistency across the environments where the application is developed, tested and deployed.



## **Continuous Monitoring –**

- This is a very crucial stage of the DevOps life cycle where you continuously monitor the performance of your application.
- The system errors such as low memory, server not reachable, etc. are resolved in this phase.
- It maintains the security and availability of the services, detect network issues, and are resolved in this phase.
- The popular tools used for this are **Splunk, ELK Stack, Nagios, NewRelic and Sensu**. These tools help you monitor the application's performance and the servers closely and also enable you to check the health of the system proactively.



# DEVOPS CORE PRINCIPLE

## PRINCIPLE 1 - Customer-Centric Action

- It is imperative nowadays to have short feedback loops with real customers and end-users, and that all activity in building IT products and services centres around these clients.
- To be able to meet these customers' requirements, DevOps organizations require the guts to act as lean start-ups that innovate continuously, pivot when an individual strategy is not (or no longer) working, and constantly invests in products and services that will receive a maximum level of customer delight.

## PRINCIPLE 2 - Create with the End in Mind

- Organizations need to let go of waterfall and process-oriented models where each unit or individual works only for a particular role/function, without overseeing the complete picture.
- They need to act as *product companies* that explicitly focus on building working products sold to real customers, and all employees need to share the engineering mindset that is required actually to envision and realize those products.

# DEVOPS CORE PRINCIPLE

## PRINCIPLE 3 - End-To-End Responsibility

- Where traditional organizations develop IT solutions and then hand them over to Operations to deploy and maintain these solutions, in a DevOps environment teams are vertically organized such that they are fully accountable from *concept to grave*.
- IT products or services created and delivered by these teams remain under the responsibility of these stable groups.
- These teams also provide performance support, until they become end-of-life, which greatly enhances the level of responsibility felt and the quality of the products engineered

## PRINCIPLE 4 - Cross-Functional Autonomous Teams

- In product organizations with vertical, fully responsible teams, these teams need to be entirely independent throughout the whole lifecycle.
- That requires a balanced set of skills and also highlights the need for team members with *T-shaped* all-round profiles instead of old-school IT specialists who are only knowledgeable or skilled in for example testing, requirements analysis or coding.

# DEVOPS CORE PRINCIPLE

## PRINCIPLE 5 - Continuous Improvement

- End-to-end responsibility also means that organizations need to adapt continuously in the light of changing circumstances (*e.g. customer needs, changes in legislation, new technology becomes available*).
- In a DevOps culture, a strong focus is put on continuous improvement to minimize waste, optimize for speed, costs, and ease of delivery, and to continuously improve the products/services offered.
- Experimentation is therefore an important activity to embed and develop a way of learning from failures is essential.

## PRINCIPLE 6 - Automate Everything You Can

- To adopt a continuous improvement culture with high cycle rates and to create an IT organization that receives instant feedback from end users or customers, many organizations have quite some waste to eliminate.
- Think of automation of not only the software development process (continuous delivery, including continuous integration and continuous deployment) but also of the whole infrastructure landscape by building next-gen container-based cloud platforms that allow infrastructure to be versioned and treated as code as well.
- Automation is synonymous with the drive to renew the way in which the team delivers its services.

# CARRIER OPPORTUNITY IN DEVOPS

- Career in DevOps is the **high demand** for job roles in the IT industry.
- DevOps professionals help business to collaborate the development and Operation team task for faster software delivery.
- The DevOps professionals are skilled with several tools and technologies for DevOps implementations in the organization.
- There are job roles such as, **DevOps engineer, security engineers, testers, DevOps architect, Release manager and DevOps evangelist** that are common in most of the organizations.
- These are **high paying jobs** and the demand is increasing for the new **cloud-based DevOps technology** experience professionals.
- Most of the organizations are implementing DevOps to match with the industry standard methodology that increases DevOps career opportunities.
- DevOps engineers are getting a good salary in the IT world.
  - The salary for **beginners** having less than 3 years experience is started around 60K USD per year to **65k USD** per year.
  - The **mid-level** professionals are getting around **70K USD** per year to 90K USD per year depending upon the skill.
  - The **senior-level** professionals are getting around **95K USD** per year to 120K USD per year. The freelancers are getting good salaries as well based on their skills and project.

# CONCLUSION

The career in DevOps is really progressive and great future. It has been observed that DevOps brings the revolution in the entire industry of information technology where large applications have been developed and enhanced. The emerging IT industry has to take the development team and operations team together to bridge the gap between them and benefit the large organizations.

It is also said that “**DevOps is culture**” followed in IT. DevOps make sure in learning, present and explain the things to client and peers. It is difficult to learn DevOps tool and start a career with as compared to other technologies. Once the career started in DevOps then the sky is the limit. The one can achieve what they like to achieve like a good salary, job designation, explore around the globe, travel, and exposure in work at the organization level.



[Talener Group Development Operations](#)

[Engineer](#)

137 salaries

[Development Operations Engineer Job available](#)

\$139,875 per year

MOTION  
RECRUITMENT

[Motion Recruitment Partners](#)

[Development Operations Engineer](#)

260 salaries

[Development Operations Engineer Job available](#)

\$140,019 per year



[Autodesk Development Operations](#)

[Engineer](#)

6 salaries

[Development Operations Engineer Job available](#)

\$138,887 per year

# THANK YOU