



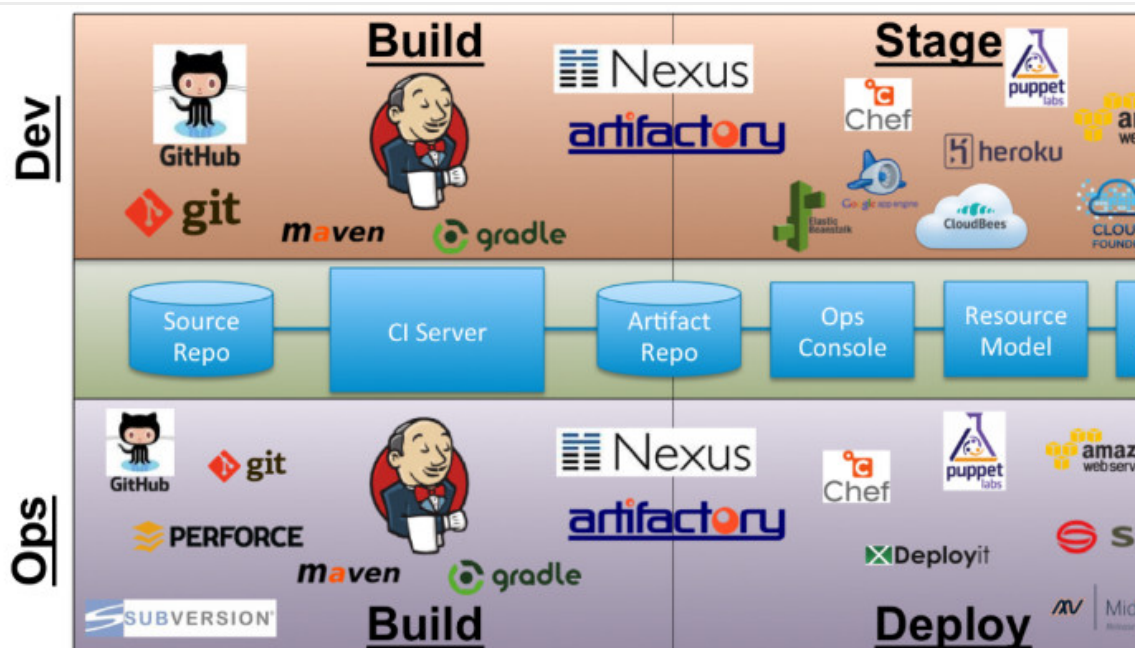
Home



My Network



Jobs



A beginners Guide to Learning DevOps

Published on June 20, 2016



Prayag Sangode

Senior Consultant at Atos

21 articles

[+ Follow](#)

DevOps is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.



DevOps is an evolution of agile software development, which demands frequent releases and iterative feedback so that applications move steadily toward the business goal, nurtured by collaboration



How DevOps Emerged?

Developer and Operations are two different roles.

Developer develops an application. He is responsible for planning the application, including designing it, architecting it, developing it etc. At some point he needs infrastructure. So he goes to systems guys who immediately point out that he should have come to them earlier and now his infrastructure will probably be late. To developer this delay is unacceptable. How long does getting a bunch of servers take ? To infrastructure guys the developer is being unreasonable. What does he understand ? They all go to Operations guy who now has to find capacity to support this application. Why didn't they come to him while the application was being planned ? And on and on it goes.

Role of infrastructure and operations was merged:

To overcome these difficulties slowly the role of infrastructure and operations was merged. Also Developers started involving Operations early in projects. Operations in turn started providing tools like continuous integration and continuous Delivery to ease the developer load. Further automation lead to virtualization and more coöperation with containers. More trust developed between Developers and Operations and the roles started overlapping and thus the role of DevOps was formed. So beyond a co-operation between the development and operations team, a series of common tools are found at any organization that implements the DevOps culture.

What you should learn or in short which tools you should learn ?

1. Source Code Management
2. Scripting
3. Continuous Integration Tools (CI)



Home

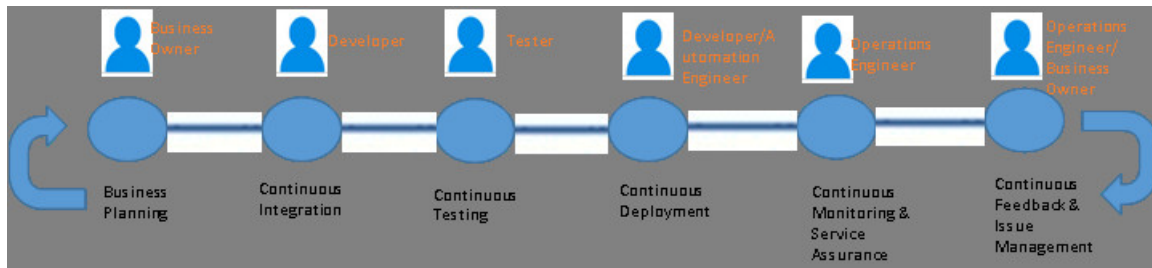
My Network

Jobs

6. Provisioning

7. Monitoring

8. Log Aggregation



1. SCM

Source control management system (SCM) is software that provides coordination and services between members of a software development team. At the most basic level, it provides file management and version control so that team members don't write over each other's changes, and only the newest versions of files are identified for use in the workspace. But that's only the beginning. SCMs also give developers the ability to work concurrently on files (in branches that may or may not converge), to merge changes with other developers' changes, to track and audit changes that were requested and made, to track bug-fix status and to perform releases. In some cases, SCMs may include other components to assist in managing a software process throughout the entire lifecycle. The difference between source control management systems and application lifecycle management (ALM) systems is really a matter of semantics and reflects the completeness of the tools provided in the system.

Examples :

i. GIT & GitHub

"Git is a free and open source distributed version control system



"GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features."

Git Hosting Service Providers :

GitHub, GitLab Assembla, Beanstalk, Bitbucket, CloudForge, Codebase, Fog Creek Kiln, planio, Perforce, RhodeCode, Unfuddle

ii. Subversion

iii. Mercurial

iv. CVS

v. ClearCase

vi. Perforce a.k.a Helix

2.Scripting :

For automation scripting one must learn scripting using **Bash, Ruby, Python**

3. Continuous Integration Tools :

A technique designed and named by Grady Booch that continually merges source code updates from all developers on a team into a shared mainline. This continual merging prevents a developer's local copy of a software project from drifting too far afield as new code is added by others, avoiding catastrophic merge conflicts. In practice, CI involves a centralized server that continually pulls in all new source code changes as developers commit them and builds the software application from scratch, notifying the team of any failures in the process. If a failure is seen, the development team is expected to refocus and fix the build before making any additional code changes.



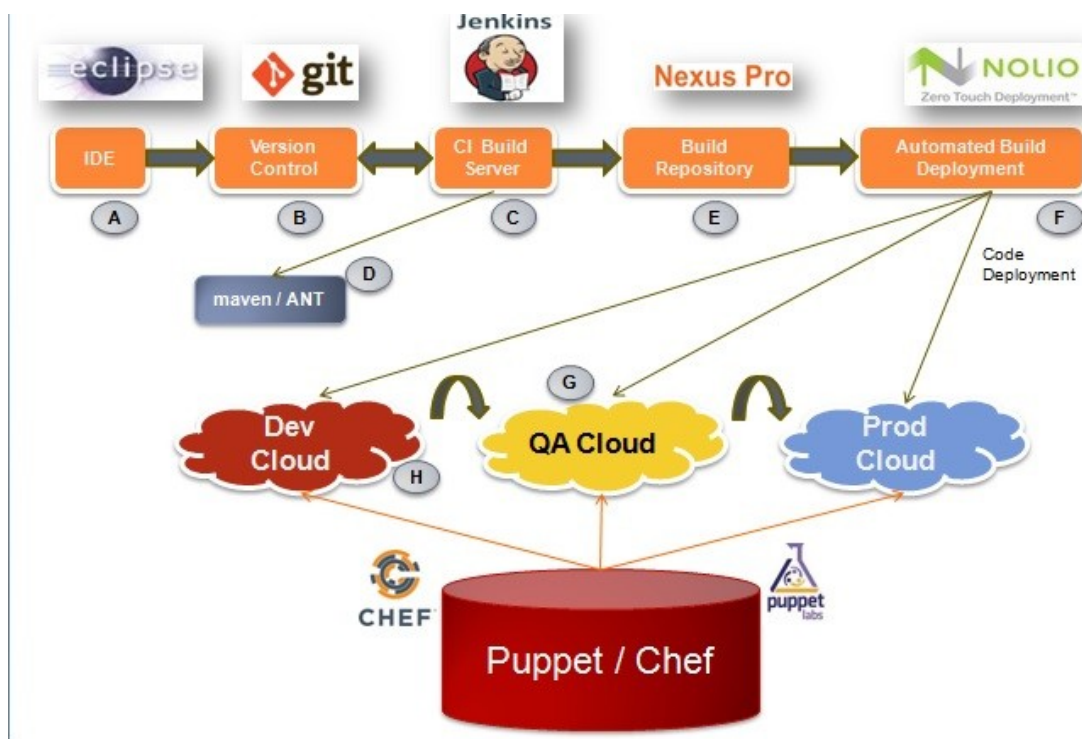
Home

My Network

Jobs

repository, ensures that the entire team is alerted any time the central code repository contains broken code. This severely limits the chance for catastrophic merge issues and loss of work built upon a broken codebase. In mature operations, the CI server may also automatically deploy the tested application to a quality assurance (QA) or staging environment, ensuring the Agile dream of a consistent working version of software.

Examples : Jenkins, Oracle's Hudson, TeamCity, Bamboo, CircleCI, Travis CI, Codeship, Semaphore CI, Drone.io



4. Virtualization:

Virtualization is especially useful in complex environments, i.e. when you want to have similar environments for Development, QA/Testing, and Production.

So if you want to set up and test anything in advance before running your apps in production blindly, then you need to configure suitable



Home

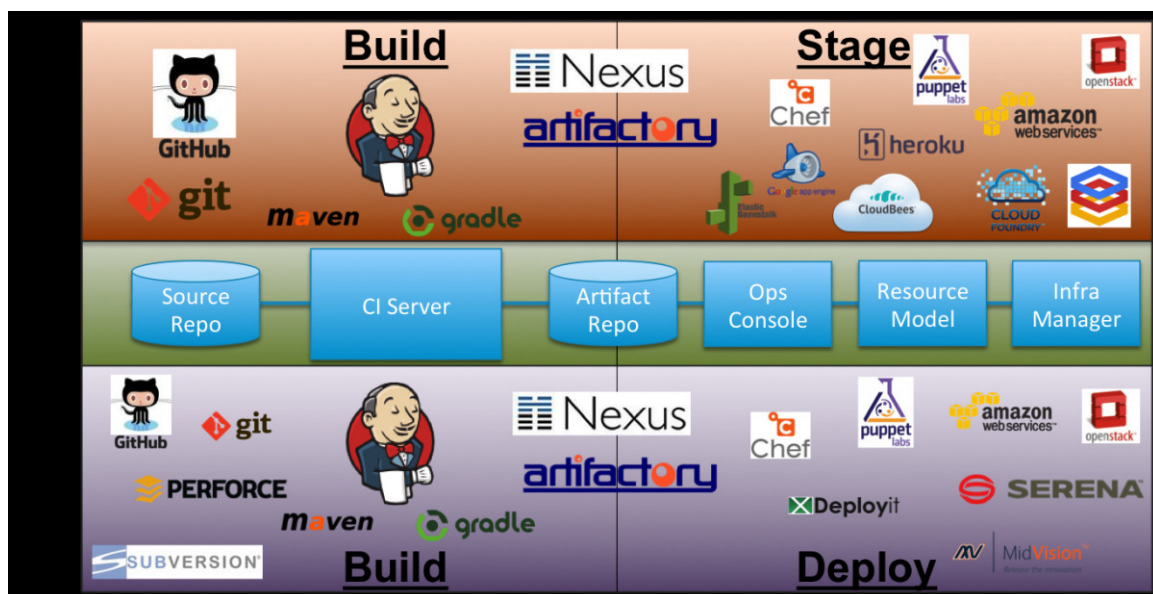
My Network

Jobs

“infrastructure as code” references to using tools to automate infrastructure configuration with all the benefits that brings.

Service virtualization allows developers and testers to access the exact test environments they need on demand. Organizations are able to accelerate the software development lifecycle and reduce risks with simulated test environments that promote earlier, faster, and more complete testing.

Examples : Docker & Vagrant



5. Cloud:

For DevOps you will mostly find hosting in clouds

AWS provides a set of flexible services designed to enable companies to more rapidly and reliably build and deliver products using AWS and DevOps practices. These services simplify provisioning and managing infrastructure, deploying application code, automating software release



With **Microsoft Azure** cloud platform, you can also use a host of popular open source DevOps tools such as Puppet, Chef, Terraform, etc

Open Shift, based on top of Docker containers and the Kubernetes container cluster manager, OpenShift 3 adds developer and operational centric tools to enable rapid application development, easy deployment and scaling, and long-term lifecycle maintenance for small and large teams and applications.

Walmart's OneOps is a Application Lifecycle Management tool that abstract most of the complex task of packaging, deployment, provisioning in all environments and is meant to give developers control over their own deployments.

Spinnaker is an open source, multi-cloud continuous delivery platform for releasing software changes with high velocity and confidence.

6. Provisioning :

Provisioning means automatically constructing software or server for a user to use without any manual steps. Provisioning is also used as a means Change Management.

Examples : Various provisioning softwares available out there however for DevOps some are favorite like Chef and Ansible.

7. Monitoring :

Nagios, Zabbix, AWS CloudWatch and StackDriver for Google.

8. Log Aggregation :



Home

My Network

Jobs

solution on all the server that not only channels all the logs to a central place but is also capable of providing good analysis of the log files.

Example : Graylog, Splunk, Logstash

List of DevOps Tools for DevOps Engineers

1. Chef
2. Jenkins
3. Puppet Enterprise
4. Bluepill
5. MCollective
6. Capistrano
7. Supervisor
8. Bcfg2
9. Graphite
10. RabbitMQ
11. Ant
12. CFEngine
13. Apache Maven
14. Ansible

[Report this](#)

Published by



Prayag Sangode

Senior Consultant at Atos
Published • 5y

21 articles

[+ Follow](#)



Like



Comment



Share



1,693 • 74 comments

Reactions





Home

My Network

Jobs

74 Comments

Most relevant ▼



Add a comment...

**Kishore Sharma Sundaram** • 2nd

3y ...

Application Development Senior Analyst at Accenture

Good info Thanks

Like | Reply

**Anu Krithiga S** • 2nd

4y ...

Growing Artist and Flutist too.. Software Engineer by Profession 🤓

I am new to Dev Ops, after reading this blog got a very clear idea about what it is exactly. Found this blog very helpful.

Like | Reply

Load more comments

**Prayag Sangode**

Senior Consultant at Atos

[+ Follow](#)

More from Prayag Sangode



Installing kind kubernetes cluster on Linux

A Beginner's guide on Installing Kubernetes

www.hustbread.com

Install kind Kubernetes cluster on Linux

Prayag Sangode on LinkedIn





Home

My Network

Jobs

Install Kubernetes locally on Linux -using k0s

Prayag Sangode on LinkedIn



Install k3d on Linux - Kubernetes Installation Guide

Prayag Sangode on LinkedIn



Install Kubernetes using k3s on CentOS7

Prayag Sangode on LinkedIn

[See all 21 articles](#)

[About](#)[Accessibility](#)[Talent Solutions](#)[Questions?](#)[Select Language](#)[Community
Guidelines](#)[Careers](#)[Marketing Solutions](#)[Visit our Help Center.](#)[English \(English\)](#)[Privacy & Terms](#) [Ad Choices](#)[Advertising](#)[Manage your account
and privacy](#)[Sales Solutions](#)[Mobile](#)[Small Business](#)[Go to your Settings.](#)[Safety Center](#)

LinkedIn Corporation © 2021