

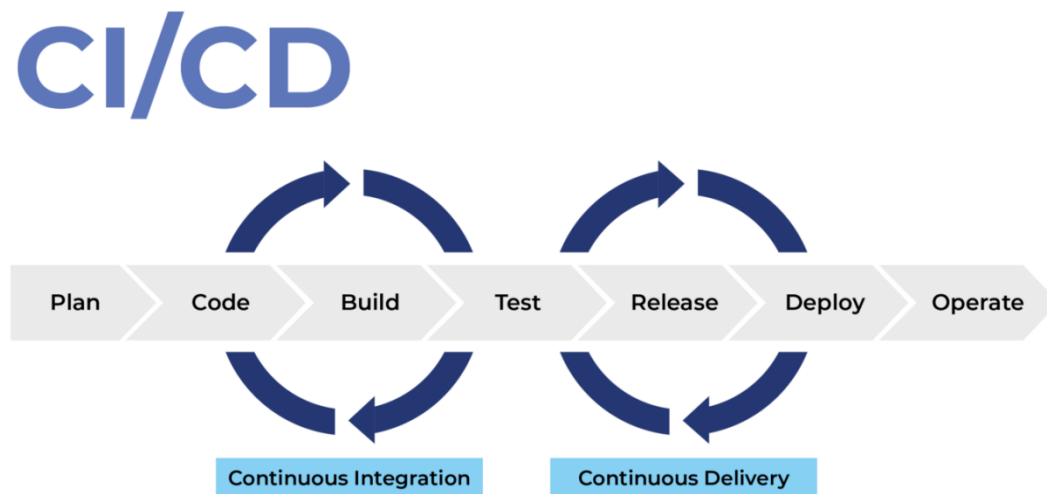
DevOps (Development + Operations)

What is DevOps?

DevOps is a term that has been trending across the IT field for at least half a decade. There are many official definitions as to what it actually means, but for our convenience we will take our own and simpler form of explanation.

In a typical software development flow there are two teams, a development team that codes the software and an operations team that hosts the project for the client and takes care of post-development stage of the project. Since the development team does not know much about post-development handling and similarly since the operations team did not know about the development stage of the software there was a lot of setbacks between teams both during production and during update releases. To overcome these set back by providing quality code and lesser time DevOps was introduced.

Where either a programmer also knows about the operations part and knows to work on it or either an operation team member knows to code the software. This might be a bit tricky but hold on it just gets easier.



Application Release Process:

Development & Operations are two main components in the whole application release process. Whenever an application is being developed, we always follow the same process for delivering the application to end-users. No matter the method used for developing the application, it can be waterfall or agile or any model at its core to create an application and deliver it to the end-user for them to make use of.

A typical software release looks like:

Idea -> Release -> Code -> Test -> Build & Package -> Deploy -> Operate and Monitor

After the initial launch of the application we do multiple updates to the application and to keep track we version the changes.

Challenges DevOps solve:

- Miscommunication and lack of collaboration between developers and operations.
- Conflict of Interest.
- Security, There is another term DevSecOps Which is DevOps+Security.
- Application Testing
- Manual Work

DevOps is anything that creates the process of releasing software fast and with high-quality

DevOps became a role with the tools

SCM, CI/CD, Container Orchestration, Continuous Monitoring, Package Management, Treat Infrastructure as Code, Cloud.

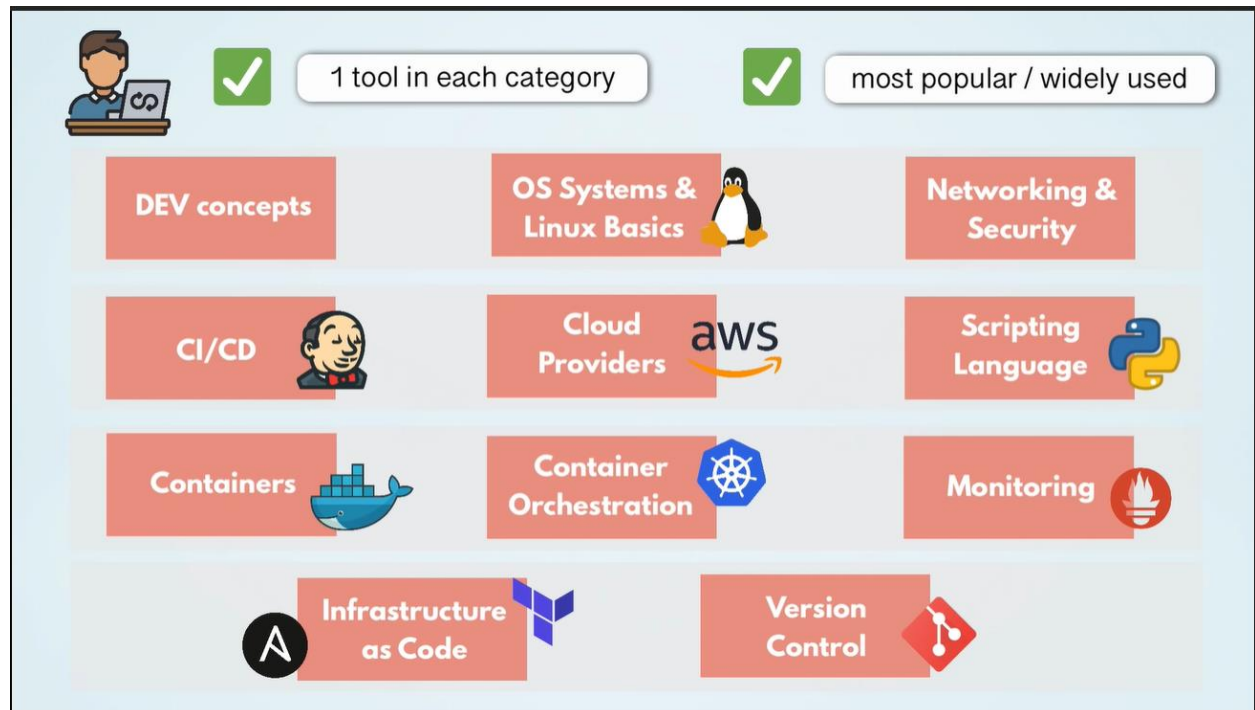
How to become a DevOps Engineer?

Concepts of Software Development - A developer develops the application and a code repository that helps them work on the code. As a DevOps engineer we don't code the application but instead we need to understand the concept of how developers work, which Git workflow they are using, how the application is configured to talk to other services and databases, concepts of Automated testing and so on.

OS & Linux Basics - Now that application needs to be deployed to the server, so that eventually the users can access it, to do this we need some kind of infrastructure for servers and such servers need to be created and configured to run the application. As a DevOps engineer we may be responsible for preparing the infrastructure to run the application, since most servers being Linux servers we need knowledge of Linux Basics, Shell Commands, comfortable with using CLI, basics of Linux File System and understanding the basics of how to administer a server, how to SSH into the server and so On.

Networking and Security - As a DevOps engineer we need to know certain basics of Networking and Security like: Configure the Firewalls to secure the application but also to open some ports to make application accessible from outside, Understanding of HTTP/HTTPS and Load Balancers, as well as understand IP address ports and how DNS works.

As a DevOps engineer we need to learn the right tools to keep up with the requirements and business needs. Some of the preferred DevOps tools are:



DevOps v/s IT Operations

As DevOps Engineer we don't need to have advanced super OS or Networking skills and be able to administer servers from start to finish. There are own professions like Network and System administrators, Security Engineers and so on that really specialize in one particular area. Our job is to understand the concepts and know all this to the extent that we are able to prepare the server to run our application but not to completely take over managing the servers and whole infrastructure.

Containers:

As containers have started becoming the hotspot for deploying applications. As a DevOps engineer we need to understand the concepts of Virtualization and Containers and also be able to manage Containerized applications on a server.

As an application has been updated the next task is to how we can make the new update release to a server, and the major task is how to do this continuously in an efficient fast and in automated way.

When a feature or a bug fix is done:

1. We need to run tests
2. Package the application as an artifact (JAR, WAR, Zip, etc..) so that it can be deployed. This is where build tools and package manager tools come in. (E.g.: Maven, Gradle)
3. Build Docker Images (Containerize Application)
4. Save the image generated to a repository. (E.g.: Nexus, DockerHub).

But we don't want to keep doing these steps manually rather we need an automated pipeline to do these repetitive process for us. So we make use of build automation tools and one such build automation tool is Jenkins and to keep this going we will need to connect this pipeline to our repository to get the code. This flow represents the core of CI/CD pipeline.

Cloud Providers:

Many companies rely on Cloud to store data. This helps reduce the cost and also provide a wide range of options to work with. This is Infrastructure as a service. Many of the features offered by cloud are Provider specific. As a DevOps engineer we need to learn about those cloud services (AWS Preferred), and again not everything about the cloud and it's provider only the services we actually need to deploy and run our specific application.

Container Orchestration:

To manage larger applications that come with many docker images we need a service to manage those containers and to do so we make use of Container Orchestration tool called Kubernetes. As a DevOps engineer we need to understand how Kubernetes works and be able to administer and manage the cluster as well as deploy applications in it.

Monitoring:

Now that we have many containers being run on our Kubernetes cluster running on many different servers, how shall we track performance of our individual applications?. As a DevOps engineer one of our responsibilities is to set-up monitoring for our running application and underlying kubernetes cluster and servers on which the clusters are running so we use monitoring tools like prometheus, Nagios, etc.

Infrastructure as a Code:

Let's say we have a production environment to run our code, but to make sure the code works properly we need to run tests and development of our code in the same environment multiple times, creating and maintaining such infrastructure for just one environment takes a lot

of time and is very error prone so we will not want to repeat the same three times. So we will automate this process, by creating an infrastructure, configuring it to run and deploy our app. Both of these steps can be done by using a combination of two types of infrastructure as code tools: 1.Infrastructure Provisioning Tool(Eg: TerraForm) 2.Configuration Management Tool(Eg: Ansible, Chef, Puppet, etc.). As a DevOps engineer we need to know at least one of these tools to make our work more efficient as well as make our environment more transparent so we can know exactly it's state making it easy to replicate and easy to recover.

Scripting Language:

In addition to being close to work with developers and system administrators to also automate some of their tasks we would most probably need to write scripts, maybe small applications to automate tasks like backups, system monitoring, cron jobs, and network management and so on. In order to do those we need to know a scripting language, this could be OS specific scripting language (E.g.: Bash/Powershell) or something more demanded more powerful and flexible language (E.g.: Python/Ruby/GO).

So as we learnt what we need and what it is to be a DevOps engineer, the last step to become one is to select the right combination of tools that when put together can work flawlessly. Preferred combinations are:

