

DevOps Introduction

Thursday, April 7, 2022 10:45 AM

What is DevOps?

- ❖ A compound of development (Dev) and operations (Ops), DevOps is the union of people, process, and technology to continually provide value to customers.
- ❖ DevOps enables formerly siloed roles—development, IT operations, quality engineering, and security—to coordinate and collaborate to produce better, more reliable products. By adopting a DevOps culture along with DevOps practices and tools, teams gain the ability to better respond to customer needs, increase confidence in the applications they build, and achieve business goals faster.

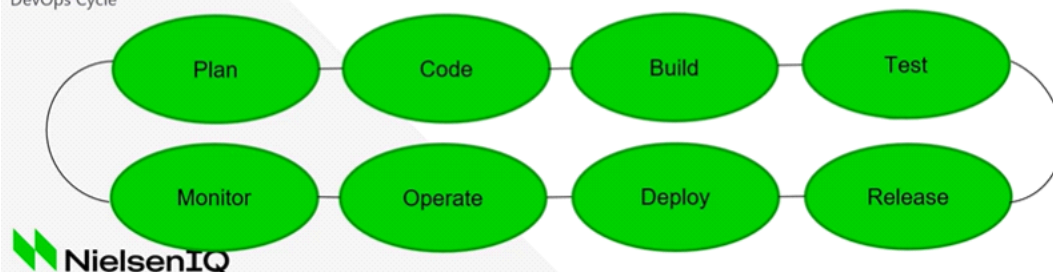
- Development team does all the coding stuff , operations team maintain operations of the application, infrastructure of the application.
- Initially before dev ops , development team does the coding of the application and push that application to the operation team. Development team doesn't consider the infrastructure stuff. There is no collaboration between development and operation team.
- Main aim is to establish a collaboration between development and operation team and return the customer satisfiable product.

Introduction to DevOps

Why do we need DevOps?

- ❖ Accelerating time to market
- ❖ Adapting to the market and competition
- ❖ Maintaining system stability and reliability
- ❖ Improving the mean time to recovery

DevOps Cycle



- DevOps collaboration will be only worked with the agile.
- As part of infrastructure , if we not written code properly. By sending these codes to Dev ops pipeline they find the errors using static analysis and vulnerability analysis. Operation team helps us to establish reliable infrastructure. Mean time to recovery relies on operations team.
- In planning we use software's like JIRA. And coding we do in Jenkins pipeline rather than locally. In building stage we make the application into different file format like jar files. And in testing we do static code test to make sure there is no useless code in it. And test for vulnerability.
- Once we deploy we could not do testing anymore from the coder side. We need to check how customer is using application or monitor did customer is facing any issues.
- We use Monitoring tools like logic monitor or data dog.
- In dev ops the common modules will be shared between teams.

Introduction to DevOps

- ❖ What is DevOps? - DevOps in simple terms is to bring together Dev and Operations teams while following the Agile Methodologies.
- ❖ Why do we need DevOps? - Close the gap between dev, test and operations teams. Improve performance, stability, reliability and scalability with continuous feedback.
- ❖ Common DevOps framework - CALMS - Culture, Automation, Lean, Measurement, Sharing

DevOps Tools

- ❖ Jenkins - Continuous Delivery and Continuous Integration
- ❖ Docker
- ❖ Terraform
- ❖ Bitbucket

What is Jenkins and why do we need it?

- ❖ Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.
- ❖ Faster development, Better software quality, Easily customizable, Easy to audit previous runs, Community support, Continuous Integration, testing and continuous delivery

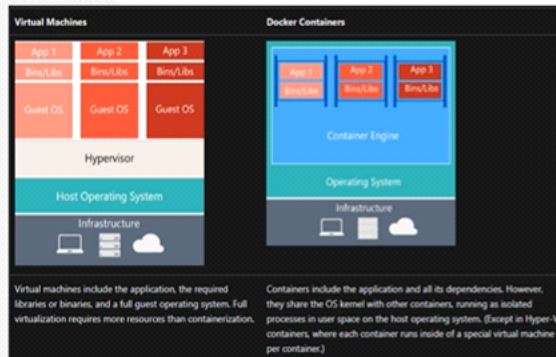
Common stages setup in Jenkins pipelines

- ❖ Commit, build, test, stage, deploy

What is Docker?

- ❖ Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers.

Difference between Docker and Virtual Machine



- In NIQ, all the docker containers will be scanned for vulnerability.

DevOps Tools

Bitbucket

- ❖ Git-based source code repository hosting service.
- ❖ Version Control system

Why do we need version control system?

- ❖ Long term change history
- ❖ Branching and merging
- ❖ Traceability

- Bit Bucket is a part of Atlassian services., it is similar to GIT HUB, we can have codes of our previous versions. If our code is verified and accepted it will be sent to master root.

Terraform

- ❖ HashiCorp Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share. You can then use a consistent workflow to provision and manage all of your infrastructure throughout its lifecycle. Terraform can manage low-level components like compute, storage, and networking resources, as well as high-level components like DNS entries and SaaS features.
- ❖ Terraform workflow consists of three stages – write, plan, apply
- ❖ Write – Define infrastructure in configuration files
- ❖ Plan – Review the changes Terraform will make to your infrastructure
- ❖ Apply – Terraform provisions your infrastructure and updates your state files.

Why do we use terraform?

- ❖ Manage Infrastructure
- ❖ Track your infrastructure
- ❖ Automate Changes
- ❖ Standardize configurations.

Terraform template is used to create different infrastructure
Every time we create a terraform we declare who is the provider.

Docker Introduction

Thursday, April 7, 2022 12:00 PM

Docker is process of containerizing and packaging the application including OS so we can send it to any computer and execute it anywhere. It is similar to executable file. Container have everything to run the application. It contains the OS , packages and libraries. Docker CLI manages the docker containers. If we create a application in form of docker container in Linux , we can run that docker container in any Operating system without depending of installing of libraries externally.

Introduction to Docker

Docker is a Linux-based, open-source containerization platform that developers use to build, run, and package applications for deployment using containers. Unlike virtual machines, Docker containers offer:

- ❖ OS-level abstraction with optimum resource utilization
- ❖ Interoperability
- ❖ Efficient build and test
- ❖ Faster application execution

Components of Docker Architecture

- ❖ Images
- ❖ Containers
- ❖ Registries
- ❖ Docker Engine

❖ Dockerfile instructions

FROM, LABEL, RUN, CMD, EXPOSE, ENV, ADD/COPY, ENTRYPOINT, VOLUME, USER, WORKDIR, ONBUILD

Demo – Creating a docker image

Demo – simple optimization of docker image

- All the images, containers that we create will be stored in azure container registry but not bit bucket.
- From => import the dependencies or OR or any files

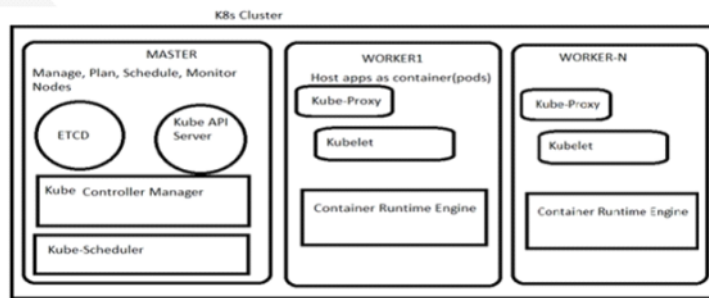
Kubernetes Introduction

Thursday, April 7, 2022 2:54 PM

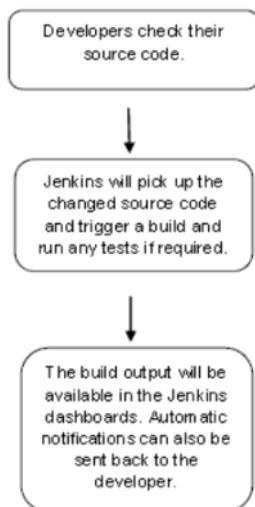
Introduction to Kubernetes

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

K8s Architecture



- Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place.



- There are two primary reasons for it. First, to make your applications scalable. If the workload on the application goes up, you're able to create more containers, or else increase the number of nodes within the Kubernetes cluster. Second, to increase the robustness of the infrastructure.

The screenshot shows the 'Node pools' interface for a Kubernetes service named 'muenpairfl2-anw'. The interface includes a search bar, a list of actions (Add node pool, Refresh, Upgrade Kubernetes, Update image, Scale node pool, Delete, Troubleshoot), and a table of node pools. The table has columns for Provisioning state, Power state, Node count, Mode, Kubernetes version, Node size, and Operating system. There are three node pools listed, all with a 'Succeeded' provisioning state and 'Running' power state.

Provisioning state	Power state	Node count	Mode	Kubernetes version	Node size	Operating system
Succeeded	Running	3/3 ready	User	1.16.9	Standard_D32s...	Linux
Succeeded	Running	4/4 ready	User	1.16.9	Standard_D16s...	Linux
Succeeded	Running	2/2 ready	System	1.16.9	Standard_D8s_v3	Linux

- These Thread pools are managed by kube scheduler.
- In Kubernetes we use Yaml files. And even we can use docker images using Yaml files under Kubernetes.
- In Kubernetes cluster we can execute the services parallelly or even non parallel.

DevSecOps means thinking about application and infrastructure security from the start. It also means automating some security gates to keep the DevOps workflow from slowing down and selecting the right tools to continuously integrate security

Tools Used in NIQ for security

- ❖ CI/CD security scan – Prisma Cloud, Wiz
- ❖ SIEM (Security Information and Event Management) – Splunk
- ❖ Web Security WAF
- ❖ Cloud Firewalls (Azure Firewall)

DevOps Udemy Course

Monday, October 17, 2022

10:50 PM

- infrastructure , cloud computing and coding are the main components of the devops
- Ansible is used for configuration management
- Docker is used for containerization
- Container Orchestration by using kubernetes
- Cloud automation using terraform
- Names of the Tools in Devops
- So from a long time, IT industry was divided into two zones. You can say, development and operations, and in development. You have developers, testers, build and release anybody who is involved in creating and testing softwares and in the Ops side system administrators, cloud engineers. The list goes on and on. So Operation Side is involved in delivering the software or its features to the end users and over the period of time because of Agile and many of other methodologies, development has become very fast, but the operation is still slow. So the software and its feature are not getting delivered as fast as it's getting developed. And we know business comes from the end users. So here devops solves the issue by integrating many applications from the development side , and make some automated tasks which faster the deployment. And which in turn reduce the work on the operations team.
- So DevOps really solve this problem of delivery, which I usually say DevOps is like logistics for the software delivery as you logistics to deliver or distribute your product in the retail market everywhere. So anybody in the industry who involved in developing or delivering the software will really need DevOps because, hey, everyone wants to deliver their product quickly and efficiently.
- Software Development Process
 - Requirement Analysis
 - Product Features
 - Usage
 - Users
 - User requirement
 - Market State
 - Planning
 - What do we want
 - Defines the cost and resource and risk taken to build the application
 - Design Architects : These are the road map for the developers
 - Architects design the software application on how it should it look like and the feature modeling by taking inputs from the previous phase and give the design document to the developers, so developers develop based on that design.
 - Development by Developers
 - Write software code based on the design model
 - Software Testing: Software will be pushed to production only if all the issues are resolved that are found by the testers in the development.
 - Deployment: Operation team works in deploying the product for the user. Software is pushed to the production environment. So the end user can use the product. It is the duty of system admin and operation team to run the application without downtime.
 - Maintainance: System health , performance , uptime.
 - Balance the regular changes that are happening in production and uptime.
- Total overview of the SDLC
 - 1) Requirement Analysis
 - 2) Planning
 - 3) Design
 - 4) Development
 - 5) Testing
 - 6) Deployment and maintenance
- Architects design the software , developers develop the software , testers test the software and operations team deploy and maintain the software , this is part of SDLC.

- Models in SDLC
 - Waterfall
 - Spiral
 - BIG BANG
 - Agile
- Model selection is based on the cost , risk and time taken to reach the destination