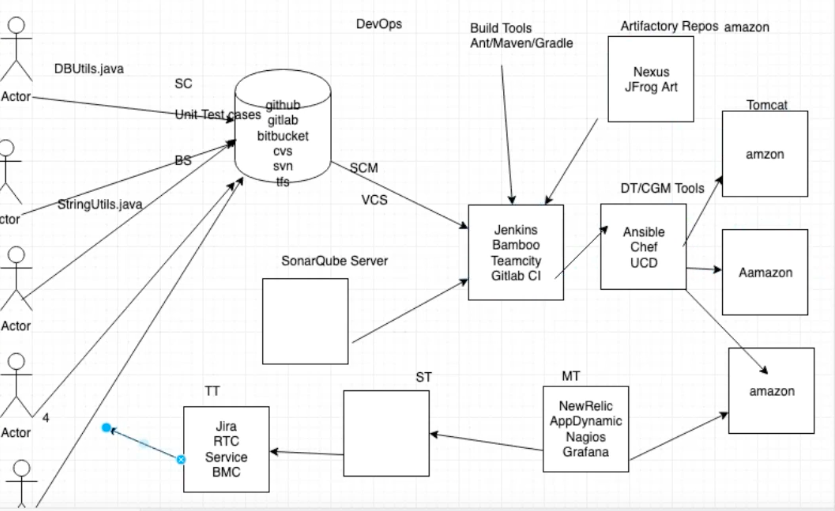
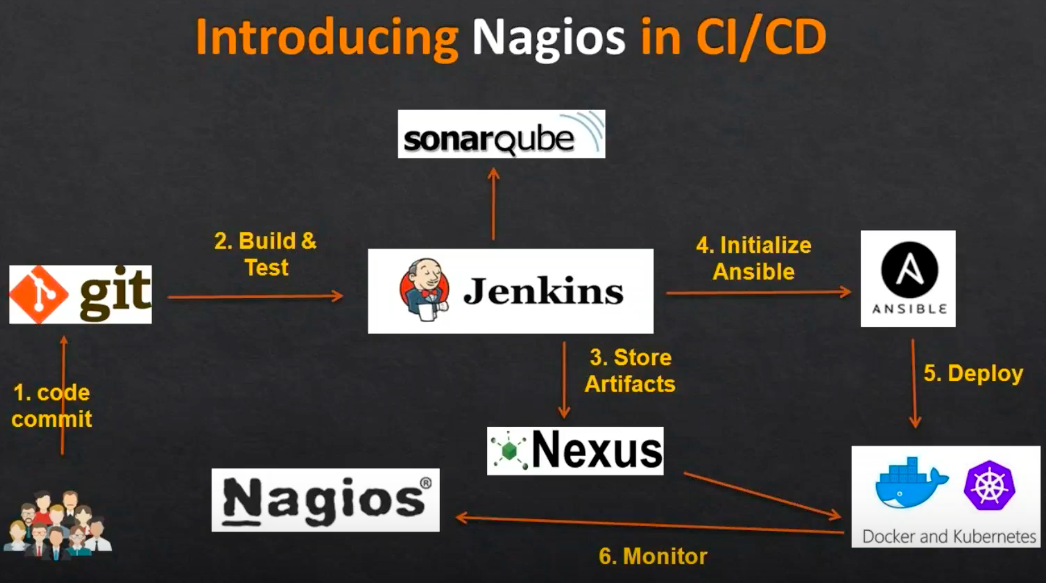
11/21/2019

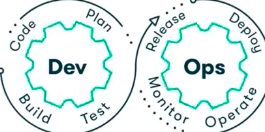
Welcome to DevOps!

<https://www.youtube.com/watch?v=XDh8awTpC_E>





Interview Questions: <https://www.edureka.co/blog/interview-questions/top-devops-interview-questions-2016/>

The short history of DevOps with Damon Edwards. 

DevOps Principles:

1. Incremental - divide a project into small junks
2. Iterative - calculating a desired result by means of a repeated cycle of operations
3. Automated
4. Continuous
5. Collaboratives
6. Self-Service
7. Holistic

DevOps Life Cycle:

1. Developers   
   develop the code and this source code is managed by Version Control System tools like Git etc.
2. GitHub   
   Developers send this code to the Git repository, and any changes made in the code is
3. Jenkins - Continuous Integration and Delivery  
   Jenkins pulls this code from the repository using the Git plugin and build it using tools like Ant or Maven.
4. Configuration Management   
   Configuration management tools like Puppet and Ansible deploys & provisions testing environment and then Jenkins releases this code on the test environment on which testing is done using tools like selenium.
5. Continuous Deployment  
   Once the code is tested, Jenkins send it for deployment on the production server (even production server is provisioned & maintained by tools like puppet).
6. Continuous Monitoring  
   After deployment It is continuously monitored by tools like Nagios.
7. Containerization  
   Docker containers provides testing environment to test the build features.

Other Tools:

* Code Review is done by tool like SonarQube
* Build Packages - with all the dependencies requited to run the app
* Back up files / artifactory are manage by tool like Nexus, JFrog
* Single Deployment uses Tomcat
* Support Team - works 24/7 to monitor, track any outages or downtime
* Ticketing Tool - used to create a ticket to notify

Technical Benefits:

* Continuous software delivery
* Less complex problem to fix
* Faster resolution of problems

Business Benefits:

* Faster delivery of features
* More stable operating environments
* Mote time available to add value (rather than fix / maintain)

Definitions:

DevOps is a software development approach that involves Continuous Development, Continuous Testing, Continuous Integration, Continuous Deployment and Continuous Monitoring of the software throughout its development life cycle.

It bridges the gap between developers and operation by improving communication, collaboration and automation. Hence, it is a culture that the organizations trying to implement where as everyone is going to work together for a common goal.

DevOps is a set of practices that combines software development and IT operations which aims to shorten the software development life cycle and provide continuous delivery with high software quality.

Problems:

Needs:

* + Faster Delivery
  + Higher Quality
  + Lesser Spending
  + Reduced Outage

Issues:

* Developer & Operation teams were working in a silo
* Each team was blaming others

Solution:

DevOps:

* Continuous Integration
* Communication
* Collaboration
* Automation

Agile Practices:

* Continuous Integration
* Nightly Builds – given to QA to testing
* Automated Unit Tests
* Automated Functional and UI Test
* Unified Release Packages
* Continuous Deployment

Agile Methods:

* Scrum – 2 weeks /10 days - sprint/iteration
* Kanban
* Lean Development
* Test-driven development

Key Areas of DevOps in relation to application dev and infrastructure:

Development sites:

* Building the code
* Covering the code
* Testing the code
* Packaging
* Deployment

Infrastructure sites:

* Provisioning
* Configuration
* Orchestration
* Deployment

Velocity:

Velocity, is the amount of work done during a sprint.

Sprint:

Sprint is one timeboxed iteration of a continuous development cycle.

Within a Sprint, planned amount of work has to be completed by the team and made ready for review.

Code Review:

* SonarQube Server
* Artifacts/packages is the same thing

Artifactory Repository:

* Nexus
* jFrog Art

Supporting Team:

Ticketing Tools:

* Jira
* RTC
* BMC

Career Opportunities:

* Continuous Integration
* Continuous Delivery
* Automation
* Continuous Monitoring
* Cloud Computing
* Configuration Management

DevOps Curriculum:

* Version Control System - git
* Build Management - Apache Maven
* Continuous Integration - Jenkins
* Configuration Management - Ansible
* Virtualization - Docker
* Cloud Computing - AWS
* Scripting - python
* Linux Essential