

Icesi DevOps Club 2019/02:

Location: Universidad Icesi

Meeting Hours: Every Thursday from 18:00 to 20:00

Duration: 12 sessions

Release: 1

Github: www.github.com/Icesi-DevOps

Primary Objective:

- Use the appropriate set of engineering tools and techniques for the solution of engineering problems. Problems will belong to the context of cloud computing services deployment

Secondary Objectives

- Know about tools and techniques that match with engineering software processes
- Use engineering tools and techniques in the creation of solutions
- Use advanced technical vocabulary for the public presentation of architectures and solutions

Club Results:

- First iteration of DevOps career plan (matrix of skills and levels)
- Portfolio of problems with detailed requirements

Members Results

- Members are able to explain the purpose of a set of cloud computing resources
- Members are able to design cloud architectures that solved cloud computing services deployment problems
- Members are able to explain to others with the appropriate technical vocabulary how the tools and techniques were integrated in order to get the solution

Methodology

Each session will be dedicated to understand, discuss and solve an engineering problem. Problems will belong to the context of cloud computing services deployment. An expanded explanation of a club session is as follows:

- Teams are randomly created
- Problem is presented, deadline is also established (this might be from one week to two weeks)
- Problem is discussed. Requirements are collected by the teams

- Teams must determine the processes to follow in order to achieve the desired goals. Processes include but are not limited to: versioning, software methodology, software architecture, infrastructure architecture, integration strategy, testing (unit, end to end, performance, acceptance), deployment, monitoring, security, cloud provider selection, minimum viable product. Constraints will be defined by the nature of the problem
- Teams present the selected processes. This selection is publicly discussed
- Teams are able to adopt the chosen processes by other teams
- Teams work on the solution. Teams match the defined processes with tools and techniques
- Solutions are publicly presented and discussed

Among with each session the topics listed in the core topics session will be covered using an active learning strategy

Problem Cases

1. Create an application that allows to the members of the DevOps club to store in a database a list of skills and levels per member. The matrix of skills and levels will be expanded with time according with emerging technologies and the club maturity. It must be possible to perform queries according to name, skill level, technology. The information in the matrix can be updated by a request of all of the members of the club. The application will allow to active and deactivate users. An authorization method is needed. Public access is available for read only queries.

Possible matched core topics: 1 to 9

2. Deploy the created solution in a cloud environment using virtual machines. The application must provide high availability in terms of the application and the database. All tests must pass in order to create a valid artifact. Performance test are needed. Monitoring and alerts must be in place also. The deployment and rollback must be done automatically. Only privileged users are able to get into production instances. Failovers must be automatically tested. All infrastructure must be able to be deleted instantly or re-created in another region. The cost of the solution must be included.

Possible match for core topics: 1 to 11

3. Deploy the solution above in a container environment.

Possible match for core topics: 1 to 11

4. Deploy the solution above in a serverless environment.

Possible match for core topics: 1 to 11

5. Create a frontend for the application. The frontend must be done using a JS framework. Use an authorization method. The UI must allow to perform the queries accepted by the backend.

Possible match for core topics: 1 to 11

6. The same requirements for the backend also apply in terms of the deployment of the solution.

Possible match for core topics: 1 to 11

7. Develop optimization improvements related with resources consumption, billing, solution complexity for any or both of frontend and backend.
8. New feature: Everytime an event is generated by the app, a notification must be sent to a specific communication channel. The notifications must be sent to Slack and Teams communication channels.
9. New feature: Create a CLI that allows a member of the team to query the backend endpoint (URIs). Authorization using username/password and MFA is needed. Only admin users can activate/deactivate members.

Core Topics

1. Agile Methodologies
 - a. Concept of Agile
 - b. Agile Manifesto
 - c. Scrum
 - i. Roles
 - ii. Project, Features, User Stories and Tasks
 - iii. Velocity (Personal Software Process)
2. Linux
 - a. Shells (zsh, fish)
 - b. Plugins
 - c. Tools
3. Git
 - a. Basics
 - b. Git flow
 - c. Tools (GitHub, GitLab, Git Editors, Git Plugins)
4. Infrastructure as Code (IaaS)
 - a. Declarative
 - b. Imperative
5. Microservices

- a. REST, GraphQL
 - b. Microframeworks, Solution template
 - c. OPEN API 3.0
- 6. Databases
 - a. SQL
 - b. NOSQL
 - c. New SQL
- 7. Tests
 - a. TDD, BDD
 - b. Performance tests
 - c. End to End
 - d. Acceptance
- 8. Microservices intercommunication
 - a. Service Discovery
 - b. Message Queues
 - c. Streaming
 - d. Message Topics
 - e. GRPC
- 9. CI/CD Tools
 - a. Jenkins
 - b. Concourse
 - c. Travis
 - d. Drone.io
 - e. CircleCI
 - f. GoCD
- 10. Solutions architectures across cloud providers
 - a. VMs
 - b. Containers
 - c. Serverless
- 11. Deployment
 - a. Blue green deployment
 - b. Red black deployment
 - c. Canary deployment

References

1. Agile Methodologies
 - a. [What is scrum?](#)
 - b. [How Does Scrum Fit With Agile?](#)
 - c. [Kanban or Scrum?](#)
2. Linux
 - a. [Bashscripting Guide](#)
 - b. Commands: [Awesome Shell](#)
3. Git
 - a. [Branching strategies](#) ([GitFlow](#), [Trunk Based Development](#))
 - b. [Commands](#)
4. Software Architecture
 - a. [Microservices in a Nutshell](#), Microservices [Definition by Martin Fowler](#)
 - b. [What is DDD?](#)
 - c. [Domain Driven Design \(DDD\) — A Technical Perspective On Software Platform Strategy](#)
 - d. [MDA](#)
 - e. [DDD versus MDA](#)
5. Tests
 - a. [Test Drive Development](#)
 - i. [Diseño ágil con TDD](#)
 - b. Behaviour Driven Development
 - i. [Main Concepts](#)
 - ii. [How to start?](#)
 - iii. [By Martin Fowler](#)
 - c. Performance and Load Test
 - i. [My boss wants me to load test our application! - JMETER](#) ([Alternative 1](#), [Alternative 2](#))
6. Microservices intercommunication
7. CI/CD Tools
8. General Resources
 - a. [DevOps Roadmap 2019](#)
 - b. [9 open source devops tools](#)
 - c. [Book State of devOps Tools](#), downloaded from [devops.com](#)
 - d. [High Performance Computing Containers \(HPC\)](#)
 - i. [Kubernetes, HPC and MPI](#)
 - ii. [Bright Computing - High Performance Computing \(HPC\) and Big Data on demand with Cluster-as-a-Service \(CaaS\)](#)
 - iii. High Throughput Computing (**HTC**)
 - iv. [HTC and HPC](#)
 - e. [Creating A Highly Scalable Website Monitoring Service in half an hour using AWS Lambda](#)