Web Development Engineer

DevOps

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

- Continuous Integration: is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run
- Continuos Delivery: is a DevOps software development practice where code changes are automatically built, tested, and prepared for a release to production.

Blue/Green DevOps

A blue/green deployment is a change management strategy for releasing software code. Blue/green deployments, which may also be referred to as A/B deployments require two identical hardware environments that are configured exactly the same way. While one environment is active and serving end users, the other environment remains idle.

Blue/green deployments are often used for consumer-facing applications and applications with critical uptime requirements. New code is released to the inactive environment, where it is thoroughly tested. Once the code has been vetted, the team makes the idle environment active, typically by adjusting a router configuration to redirect application program traffic. The process reverses when the next software iteration is ready for release.

-Check Out on Aws

Infrastructure as code

Using AWS CloudFormation you can write a description of the resources that you want to create on your AWS account, and then ask AWS CloudFormation to make this description into reality.

Benefits of Infrastructure as code

- Visibility: An infrastructure as code template serves as a very clear reference of what resources are on your account, and what their settings are. You don't have to navigate to the web console to check the parameters.
- Stability: If you accidentally change the wrong setting or delete the wrong resource in the web console you can break things. Infrastructure as code helps solve this, especially when it is combined with version control, such as Git.
- Scalability: With infrastructure as code you can write it once and then reuse it many times. This means that one well written template can be used as the basis for multiple services, in multiple regions around the world, making it much easier to horizontally scale.
- Security: Once again infrastructure as code gives you a unified template for how to deploy your architecture. If you create one well secured architecture you can reuse it multiple times, and know that each deployed version is following the same settings.

- Transactional: CloudFormation not only creates resources on your AWS account but also waits for them to stabilize while they start. It verifies that provisioning was successful, and if there is a failure it can gracefully roll the infrastructure back to a past known good state.

AWS CLI

The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI introduces a new set of simple file commands for efficient file transfers to and from Amazon S3.

- AWS CLI AND CLI TOOLS

Web Dev

- Static Website
- S3
- Web Hosting
- And Much More

Mobile Dev

- Aws Amplify
- Aws Mobile
- Aws Mobile Hub

Game Dev

- Aws Mobile For Startups and Gaming

Databases

Data Structures

- Basics
- Edx
- Cloud Oriented DS

Very Important Reading

- Important Reading

Non-Relational

- MongoDB on AWS
- Read More on NoSQL Databases.

Relational

- Link