Getting Started with DevOps on AWS

Sanchit Jain
12th March, Saturday
10:30 AM to 12:00 PM

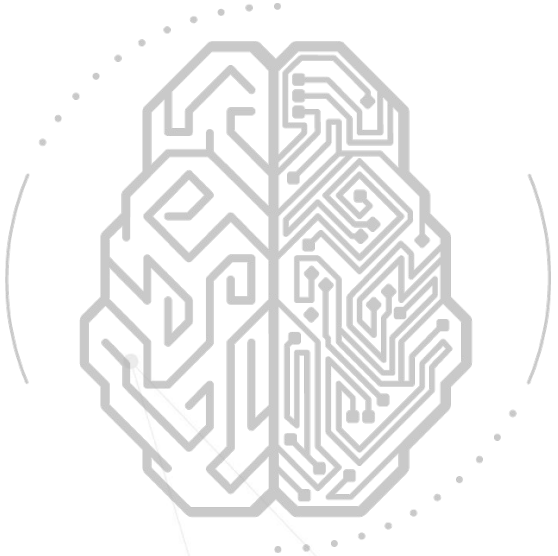
Speakers

Sanchit Jain

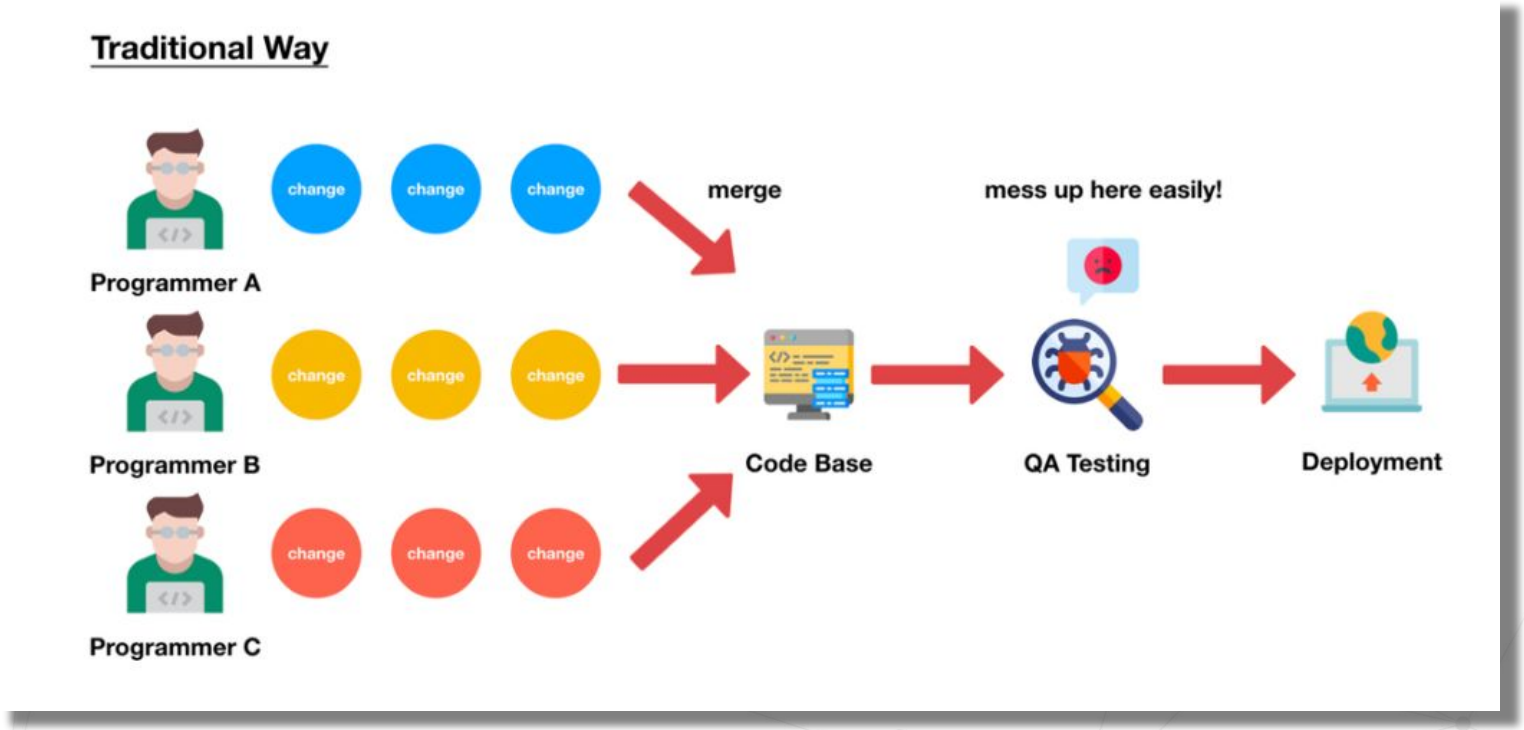
Lead Architect - AWS at Quantiphi
AWS APN Ambassador

FOLLOW ME

Overview of DevOps



Traditional Way



Life Without DevOps

Meme Credits: https://dev.to/presto412/hyperledger-fabric-transitioning-from-development-to-production-4dch

What Drives the Need for DevOps?

Lack of Automated and Secured Workflow Management Systems in Business organization

Building And Maintaining Servers

Time consuming and unproductive

No Environment Management

Differences in development and production environments

Slow Deployments

Costly error prone manual process and efforts

No Shared Ownership

Lack of feedback and proper metric leads

No Proper Configuration Management

Discrepancies in managing configurations

Deployments Are A Blocker

manual management of application configuration and versions

Production Downtime

Due to lack of improper deployment instructions / checklist

Hacking & Security Threat

Fixing directly in production, instead of a proper hotfix process

Impact on business due to challenges & Problems

Ineffective utilisation of resources

Misuse of working capital

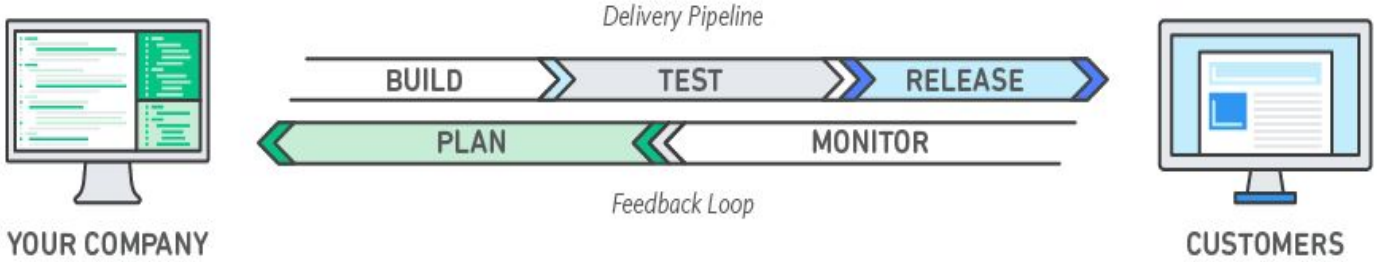
Increased in time-to-market

Laboursome process

Introduction to DevOps

What Is DevOps?

- DevOps is a combination of cultural philosophies, practices and tools that increase an organization's ability to deliver applications and services at high velocity; evolving and improving products at a faster pace than organizations that use traditional software development and infrastructure management processes.
- It combines software development (Dev) and information-technology operations (Ops) that results in shortening the systems development life cycle and providing continuous delivery with high software quality and exceptional performance.
- In order for DevOps to impact businesses, it needs to be embraced and adopted by everyone responsible.



CALMS Framework

CALMS is a framework that assesses a company's ability to adopt DevOps processes, as well as a way of measuring success during a DevOps transformation

Culture

Automation

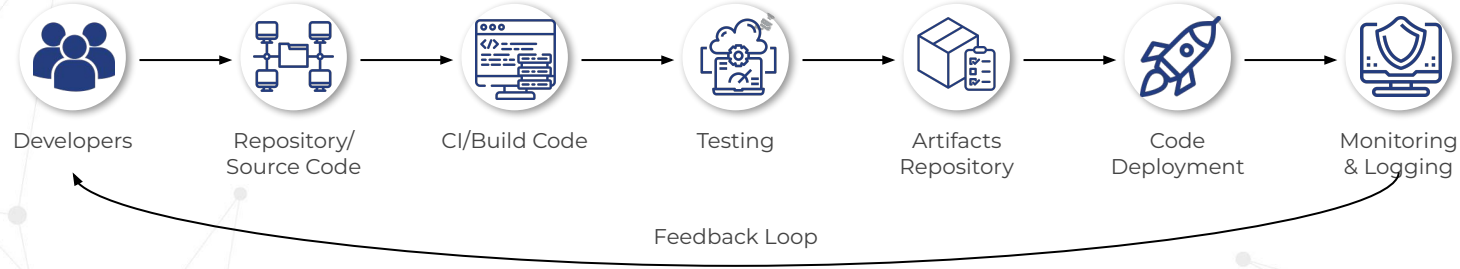
Lean

Measurement

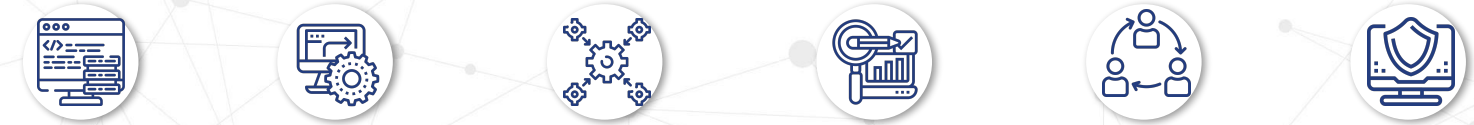
Sharing

Introduction to DevOps

The DevOps Process



DevOps Practices



Impact of DevOps

High Speed Operations

Rapid Delivery

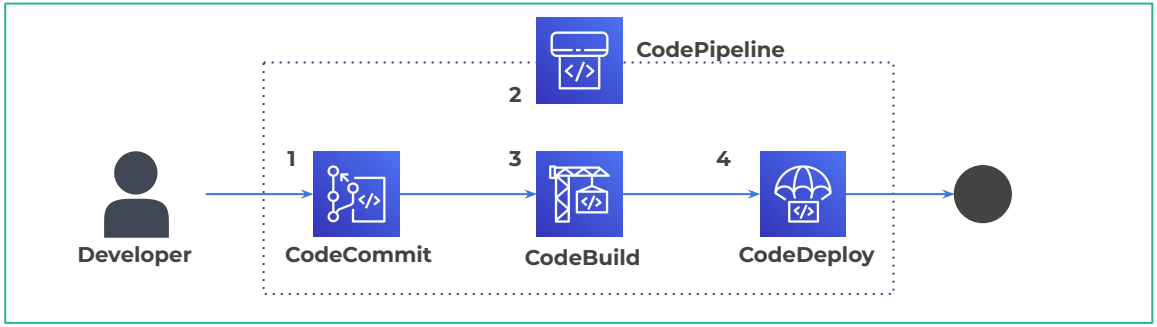
Increased Reliability

High Scalability

Enhanced Security

Better Collaboration

DevOps Cycle



Process:

1. The Developer pushes the changes to the repository in CodeCommit and the flow is triggered
2. CodePipeline listens to push event and gets the source code. It launches CodeBuild
3. CodeBuild builds the code and pushes it to CodeDeploy.
4. CodeDeploy deploys the application code in to a designated destination

Other AWS Services Used for DevOps

Amazon EC2
Virtual Servers to deploy Code

Amazon ECS
Container Management Service

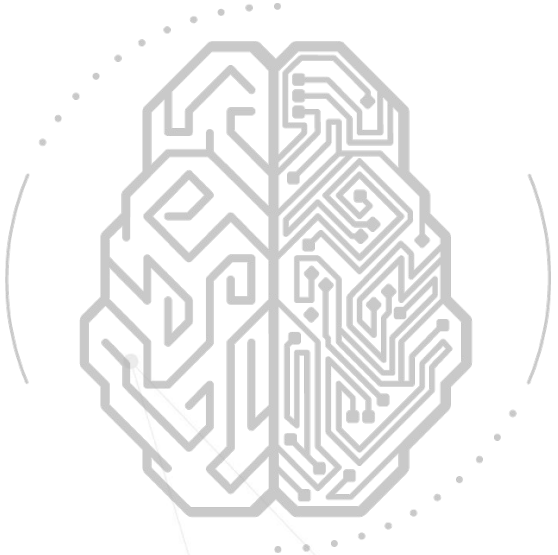
Amazon Lambda
Runs code without managing servers

Amazon CloudFormation
Manage a collection of related AWS services

Amazon OpsWorks
To configure and operate applications

Amazon Config
AWS configuration history and change notifications

AWS DevOps Services



Devops Technology Stack

CONTINUOUS INTEGRATION CONTINUOUS DELIVERY


AWS CodeCommit


AWS CodeBuild


AWS CodeDeploy


AWS CodePipeline

INFRASTRUCTURE & AUTOMATION


AWS CloudFormation


AWS OpsWorks


AWS System Manager


AWS CodeDeploy

MONITORING & SECURITY


Amazon CloudWatch


Amazon CloudTrail


Amazon X-Ray


Amazon Config


Amazon Inspector


AWS Trusted Advisor


AWS System Manager


AWS KMS

PLATFORM AS SERVICE


AWS Lambda

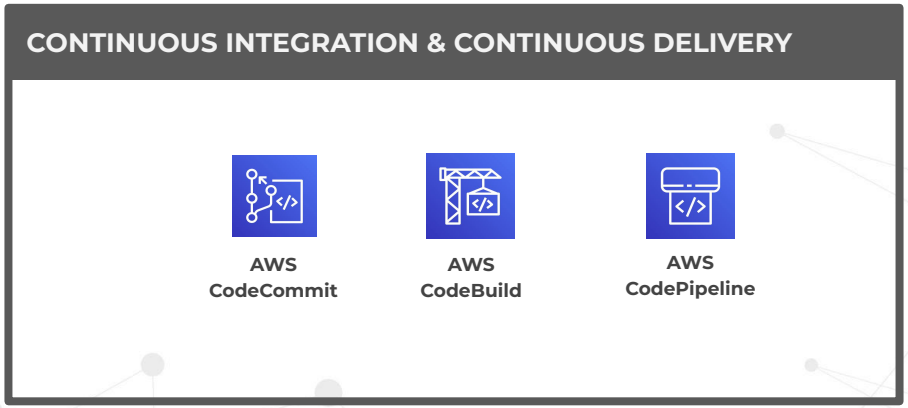

AWS Elastic Beanstalk


AWS Elastic Container Service


Amazon ECS for Kubernetes


AWS Fargate

AWS DevOps Services



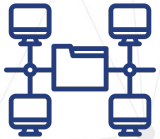
Amazon CodeCommit

Securely host highly scalable private Git repositories. Collaborate on code.

AWS CodeCommit Features

Encryption

Access Control using IAM



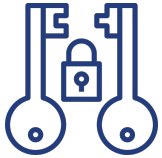
Unlimited Repositories

Easy Access and Integration

Notifications and Custom Scripts

AWS CodeCommit Benefits

Fully managed



Secure

High Availability and Durability

Collaborate on code

Faster development lifecycle

Use your existing tools

AWS CodeBuild

Build and test code with continuous scaling

Source Code

AWS CodeBuild

Continuous integration and delivery workflows

Security Monitoring and permissions
- Set granular controls over access
- Monitoring and Alerts

Build and test your code
- Preconfigured build environments
- Customize build environments

Configurable settings
- Specify build commands
- Select compute type
- Choose source integrations

Key Benefits

Fully managed build service

AWS CodeBuild eliminates the need to set up, patch, update, and manage your own build servers and software

Continuous scaling

AWS CodeBuild scales up and down automatically to meet your build volume

Pay as you go

With AWS CodeBuild, you are charged based on the number of minutes it takes to complete your build

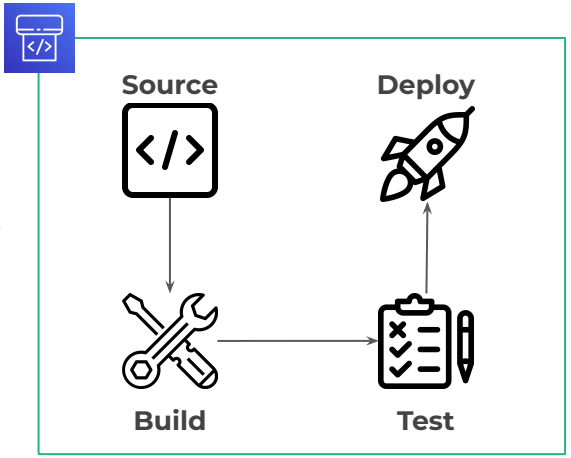
Extensible

You can bring your own build tools and programming runtimes to use with AWS CodeBuild

AWS CodePipeline

Automate continuous delivery pipelines for fast and reliable updates

Application Lifecycle



AWS CodePipeline

- Workflow modeling
- AWS integrations
- Pre-built & Custom plugins
- Receive Notifications
- Declarative templates
- Access control

Key Benefits

Serverless Architecture - Zero administration

With AWS CodePipeline, you can immediately begin to model your software release process. There are no servers to provision or set up

Rapid delivery

AWS CodePipeline automates your software release process, allowing you to rapidly release new features to your users

Configurable workflow

AWS CodePipeline allows you to model the different stages of your software release process

Easy to integrate

AWS CodePipeline can easily be extended to adapt to your specific needs. You can use pre-built plugins or your own custom plugins in any step of your release process

AWS DevOps Services

INFRASTRUCTURE & AUTOMATION

AWS
CloudFormation

AWS
CodeDeploy

AWS CloudFormation

Model and provision all your cloud infrastructure resources

Cloud Infrastructure Resources

AWS CloudFormation

Management
- Cross account & cross-region management
- Dependency management
- Preview changes to your environment
- Automates the provisioning and updating of your infrastructure in a safe and controlled manner

Extensibility
- Model, provision, and manage third party application resources
- Familiarity with programming languages
- Authoring with JSON/YAML

Key Benefits

Model it all

AWS CloudFormation allows you to model your entire infrastructure and application resources with either a text file or programming languages

Automate & deploy

AWS CloudFormation provisions your application resources in a safe, repeatable manner, allowing you to build and rebuild your infrastructure and applications, without having to perform manual actions

Codification

You can author infrastructure with any code editor, check it into a version control system, and review the files with team members before deploying into production.

Why use AWS CloudFormation?

Support for Different Resources:	AWS CF assists a smorgasbord of resources, which lets you make a highly reliable, available, and scalable or upgradeable AWS infrastructure to cater to your specific application requirements.
Easy to use:	With CloudFormation, it's easy to classify and station a suite of the secure platform's resources. It allows you to describe any contingencies or special yardsticks to pass in during runtime. You may employ any of the several CloudFormation sample templates verbatim or as a beginning point.
Flexible and Declarative:	To erect the infrastructure you require, you list out the platform's resources, interconnections, and configuration values you want the template to have and then permit CloudFormation to take care of the rest with some basic clicks in the console
Customization through parameters:	You may use parameters for customizing specific template aspects at run time. For instance, you could pass the Amazon EC2 instance types, Amazon EBS volume size, RDS database size, server port numbers, and database to the platform when creating a stack.
Drag and drop UI to visualize and edit:	AWS CloudFormation Designer offers a template diagram vhth icons denoting the Amazon platform's resources and arrow signs indicating relationships. You could create and modify templates using the interface and then alter template details with the help of the inbuilt JSON text editor.

How AWS CloudFormation work?

Code your infrastructure from scratch with the CloudFormation template language, in either YAML or JSON format, or start from many available sample templates

Check out your template code locally, or upload it into an S3 bucket

Use AWS CloudFormation via the browser console, command line tools or APIs to create a stack based on your template code

AWS CloudFormation provisions and configures the stacks and resources you specified on your template

Templates

- A template in AWS CloudFormation is a formatted text file in JSON or YAML language that describes your AWS infrastructure
- To create, view and modify templates you can use AWS CloudFormation Designer or any text editor tool
- Templates in AWS CloudFormation consists of 9 main objects

Format version

Description

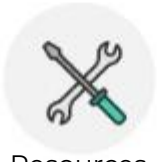
Metadata

Parameters

Mappings

Conditions

Transform



Resources

Outputs

Template Structure

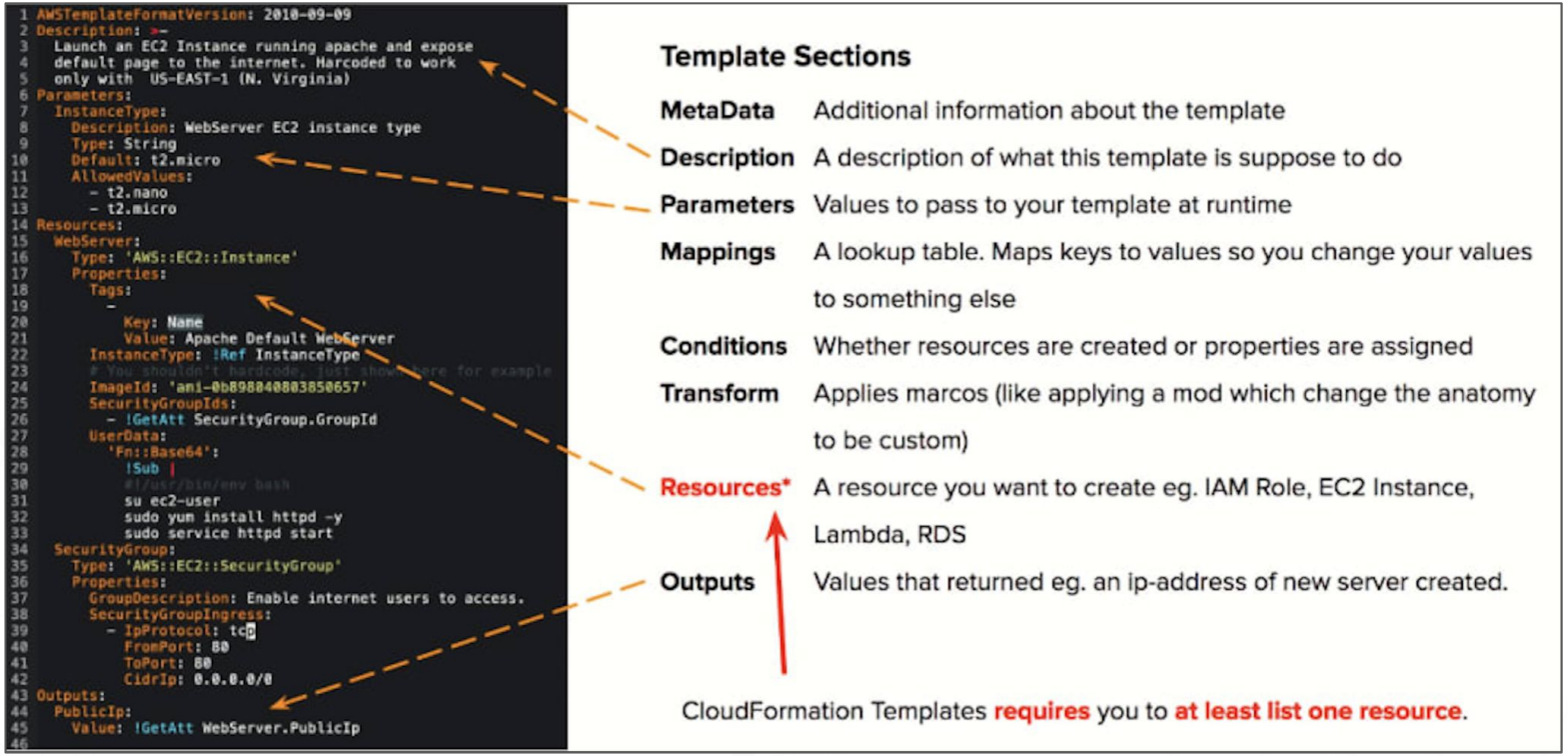
JSON

```
{
  "AWSTemplateFormatVersion" : "version date",

  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  },

  "Parameters" : {
    set of parameters
  },

  "Mappings" : {
    set of mappings
  },

  "Conditions" : {
    set of conditions
  },

  "Transform" : {
    set of transforms
  },

  "Resources" : {
    set of resources
  },

  "Outputs" : {
    set of outputs
  }
}
```

YAML

```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Template Structure



AWS CodeDeploy

Automate code deployments to maintain application uptime

Software and Applications

AWS CodeDeploy

Instance deployments
- Repeatable deployments
- Automatic scaling
- On-premises deployments

Control
- Monitoring and control
- Deployment groups
- Deployment history
- Review defined events

Easy to adopt
- Language and architecture agnostic
- Tool chain integration

Updates, Tracking and Rollback
- Rolling and Blue/Green updates
- Deployment health tracking
- Stop and rollback

Key Benefits

Centralized control

AWS CodeDeploy allows you to easily launch and track the status of your application deployments through the AWS Management Console or the AWS CLI

Automated deployments

AWS CodeDeploy fully automates your software deployments, allowing you to deploy reliably and rapidly

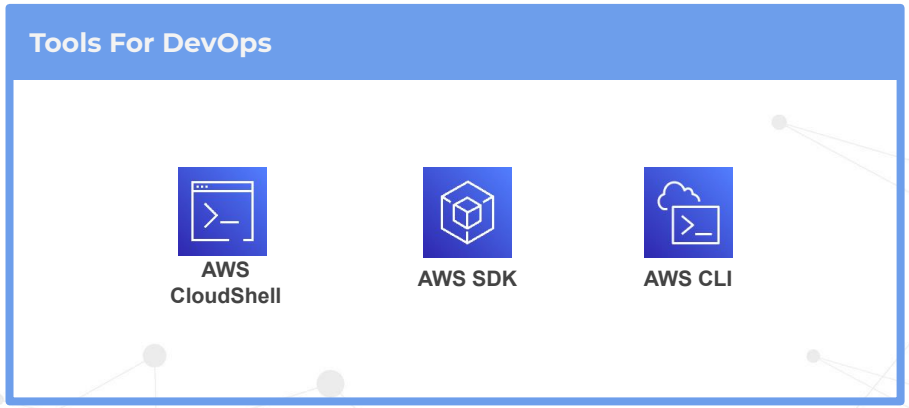
Minimize downtime

AWS CodeDeploy helps maximize your application availability during the software deployment process

Easy to adopt

AWS CodeDeploy is platform and language agnostic and works with any application

AWS DevOps Services



AWS SDK Overview

Highly secure, reliable, and scalable way to run containers

SDKs

SDKs take the complexity out of coding by providing language-specific APIs for AWS services

JavaScript

Python

PHP

.NET

Ruby

Java

Go

Node.js

C++

AWS SDK Example

Highly secure, reliable, and scalable way to run containers

```
pip install boto3
```

```
import boto3

# Let's use Amazon S3
s3 = boto3.resource('s3')
```

Steps:

1. Install boto3
2. Import and indicate which service to use
3. Use the object to create/configure.

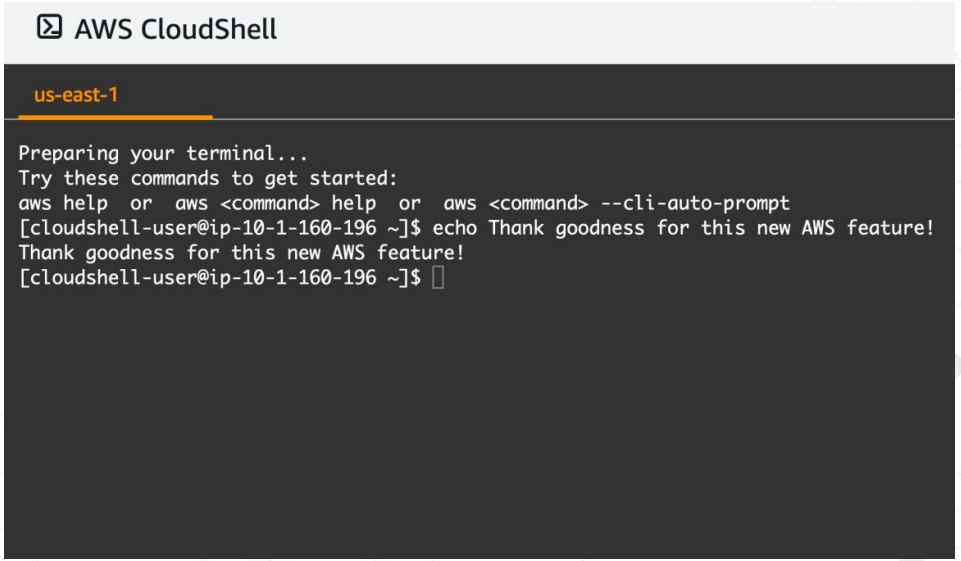
Boto3 Api Reference

```
# Print out bucket names
for bucket in s3.buckets.all():
    print(bucket.name)
```

AWS Cloudshell Overview

Common development and operations tools right from your browser and at no additional cost

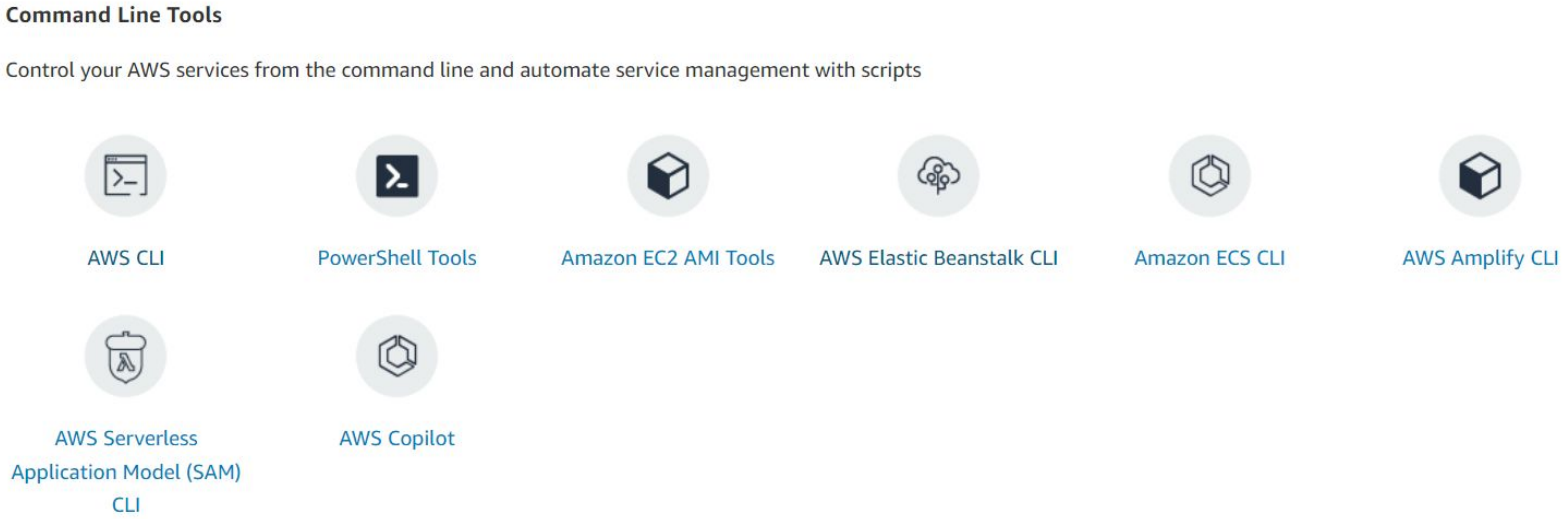
- AWS CloudShell is a browser-based shell that makes it easy to securely manage, explore, and interact with your AWS resources.
- CloudShell is pre-authenticated with console credentials
- Common development and operations tools are pre-installed, so no local installation or configuration is required.



AWS CLI Overview

unified tool to manage your AWS services

- AWS CLI is a unified tool to manage your AWS services.
- We can control multiple AWS services from the command line and automate them through scripts.



AWS CLI Example

Highly secure, reliable, and scalable way to run containers

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

```
$ aws --version
aws-cli/2.4.5 Python/3.8.8
```

```
$ aws s3 ls s3://mybucket
        LastWriteTime            Length Name
        -------------            ------ ----
                                  PRE myfolder/
2013-09-03 10:00:00            1234 myfile.txt
```

Steps:

1. Install aws cli
2. Check the installation (using aws –version)
3. Run aws cli commands

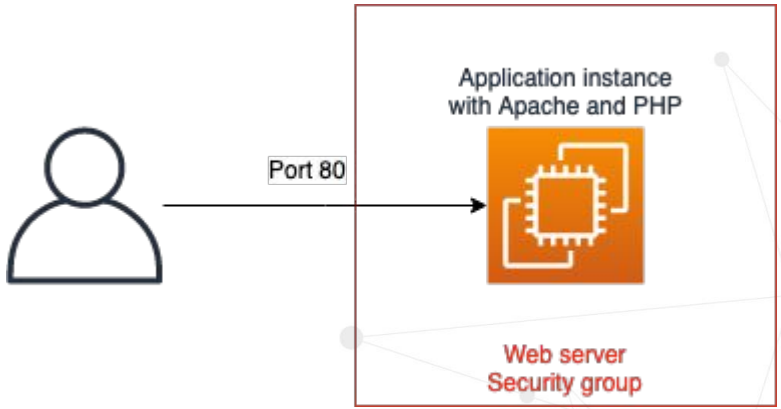
AWS CLI Reference

DEMO

Demo

- In this lab you will deploy an Apache Web server with a simple PHP application via UserData property.
  - First, you will bootstrap EC2 instance to install web server and content.
  - Then you will create an EC2 Security Group and allow access on port 80 to the instance.
  - Finally, you will view the content served by the web server.

The following diagram provides a high-level overview of the architecture you will implement.



THANK YOU