

# Getting Started with DevOps on AWS

Sanchit Jain  
26th March, Saturday  
10:30 AM to 12:00 PM



# Speakers



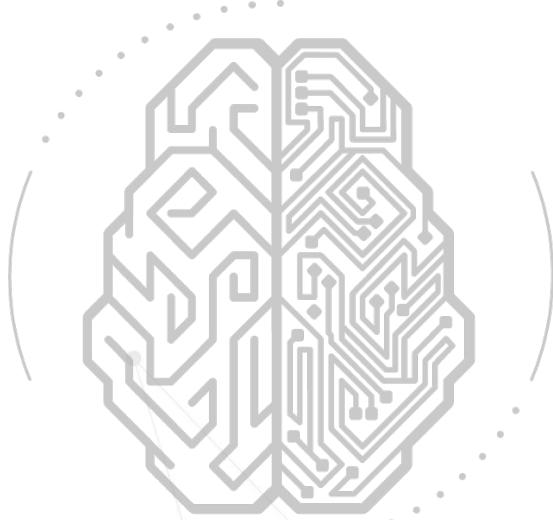
**Sanchit Jain**

Lead Architect - AWS at Quantiphi  
AWS APN Ambassador

FOLLOW ME



# ReCap



# What Drives the Need for DevOps?

Lack of Automated and Secured Workflow Management Systems in Business organization



**Impact on business  
due to challenges  
& Problems**



Ineffective utilisation  
of resources



Misuse of  
working capital



Increased in  
time-to-market

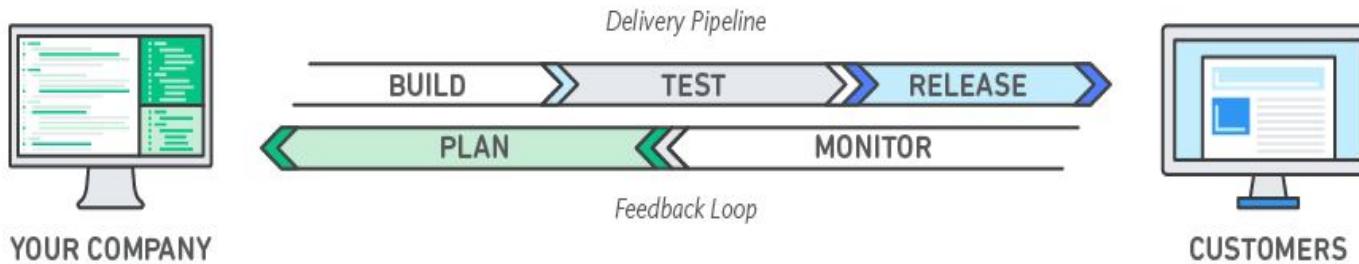


Laboursome  
process

# 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



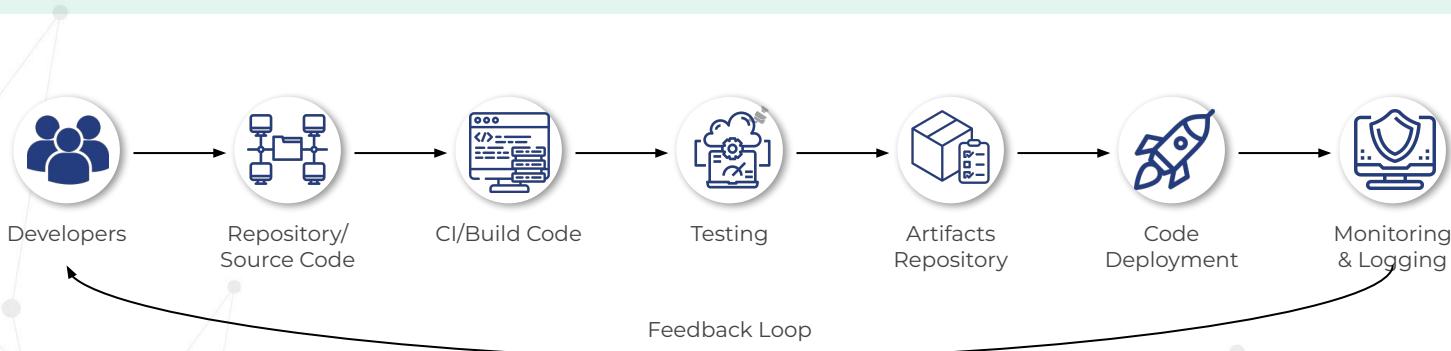
Measurement



Sharing

# Introduction to DevOps

## The DevOps Process



## Impact of DevOps



High Speed Operations



Rapid Delivery



Increased Reliability



High Scalability

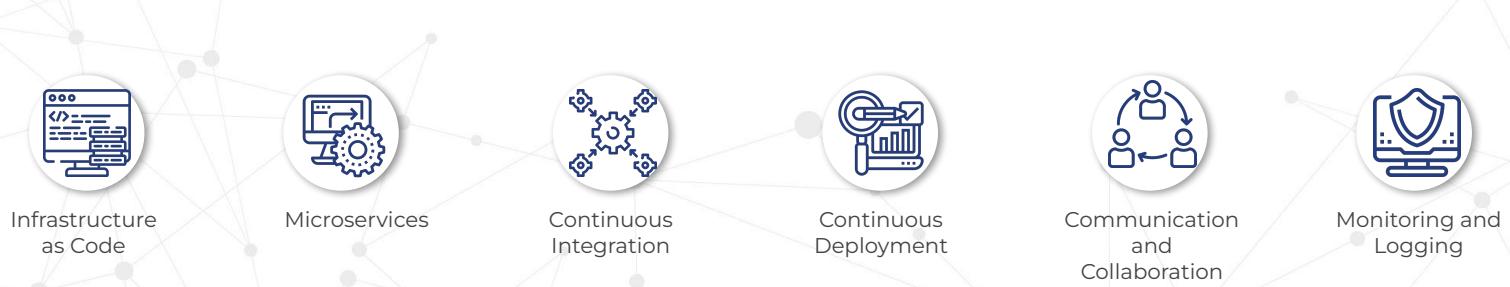


Enhanced Security



Better Collaboration

## DevOps Practices



# 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

## CONTINUOUS INTEGRATION & CONTINUOUS DELIVERY



AWS  
CodeCommit



AWS  
CodeBuild



AWS  
CodePipeline

# AWS DevOps Services

## INFRASTRUCTURE & AUTOMATION

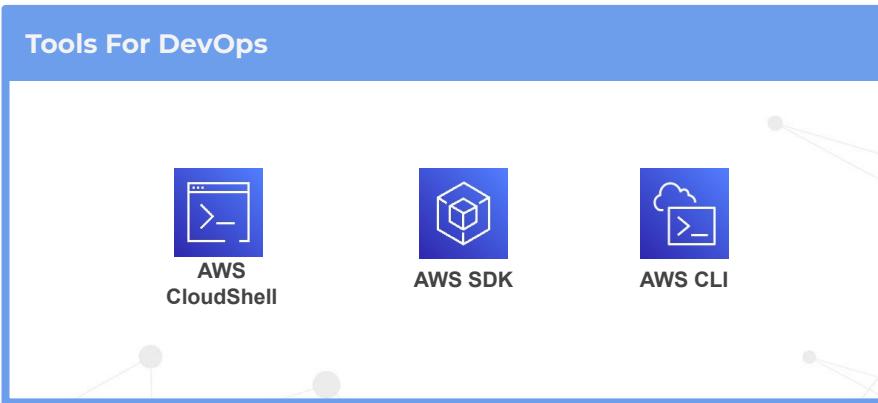


AWS  
CloudFormation



AWS  
CodeDeploy

# AWS DevOps Services



# 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

## MONITORING & SECURITY



Amazon  
CloudWatch



Amazon  
CloudTrail



Amazon  
Config



AWS  
Xray



AWS Secrets  
Manager

# Amazon CloudWatch

Observability of your AWS resources and applications on AWS and on-premises



AWS resources and applications



Amazon CloudWatch

## Key Benefits



Observability on a single platform across applications and infrastructure



Easiest way to collect metrics in AWS and on-premises



Improve operational performance and resource optimization



Get operational visibility and insight



Derive actionable insights from logs

### Collect

- Easily collect and store logs
- Built-in metrics
- Custom Metrics
- Collect and aggregate container metrics and logs

### Act

- Auto Scaling
- Automate response to operational changes with CloudWatch Events
- Alarm and automate actions on EKS, ECS, and k8s clusters

### Monitor

- Unified operational view with dashboards
- High resolution alarms
  - Logs and metrics correlation
- Application Insights for .NET and SQL Server applications
- Container monitoring insights
- Anomaly Detection
  - ServiceLens
  - Synthetics

### Analyze

- Granular data and extended retention
- Custom operations on metrics
- Analyze container metrics, logs, and traces
- Contributor Insights

# AWS CloudTrail

Track user activity and API usage



AWS resources and applications



AWS CloudTrail

Multi-region configuration

Monitoring and tracking

- Always on
- Logging Event history
- Receive Notifications

Log file integrity validation & encryption

Insights

- Data events provide insights into the resource operations
- Management events provide insights into the management operations
- Identify unusual activity in your AWS accounts

## Key Benefits

### Simplified compliance



With AWS CloudTrail, simplify your compliance audits by automatically recording and storing event logs for actions made within your AWS account

### Visibility into user and resource activity



AWS CloudTrail increases visibility into your user and resource activity by recording AWS Management Console actions and API calls

### Security analysis and troubleshooting



Discover and troubleshoot security and operational issues by capturing a comprehensive history of changes that occurred in your AWS account within a specified period of time

### Security automation



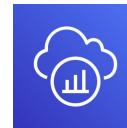
AWS CloudTrail allows you track and automatically respond to account activity threatening the security of your AWS resources

# AWS XRay

Analyze and debug production, distributed applications



AWS resources and applications



AWS Xray

## Key Benefits



Review request behavior



Discover application issues



Improve application performance



Designed for a variety of applications

Analyze and Debug Application

Analyze Application Performance

End to End tracing

# AWS Config

Record and evaluate configurations of your AWS resources



Configurations of your AWS resources



**AWS Config**

## Key Benefits

Continuously monitor and record configuration changes of your AWS resources



Continuously audit and assess the overall compliance of your AWS resource configurations



Track the relationships among resources and review resource dependencies



Simplify troubleshooting by capturing history of AWS resource configuration changes



Enterprise-wide compliance monitoring



Support for third-party resources

### Audit

- Multi-account, multi-region data aggregation
- Cloud governance dashboard
- Conformance packs to manage compliance of your AWS resource configuration at scale

### Evaluate

- Configuration snapshots
- Configuration history of AWS resources
- Configuration history of software
- Resource relationships tracking

### Integrations

- Configurable and customizable rules
- Partner solutions that integrate with AWS Config
- Integrations with other services
- Publish the configuration of third-party resources into AWS Config

# AWS Secrets Manager

Easily rotate, manage, and retrieve secrets through their lifecycle



Configurations of your AWS resources



## AWS Secrets Manager

Protect secrets to access applications

Easily rotate, manage, and retrieve credentials

Fine-grained permissions and audit secret rotation

## Key Benefits



Rotate secrets safely



Manage access with fine-grained policies



Secure and audit secrets centrally



Easily replicate secrets to multiple regions

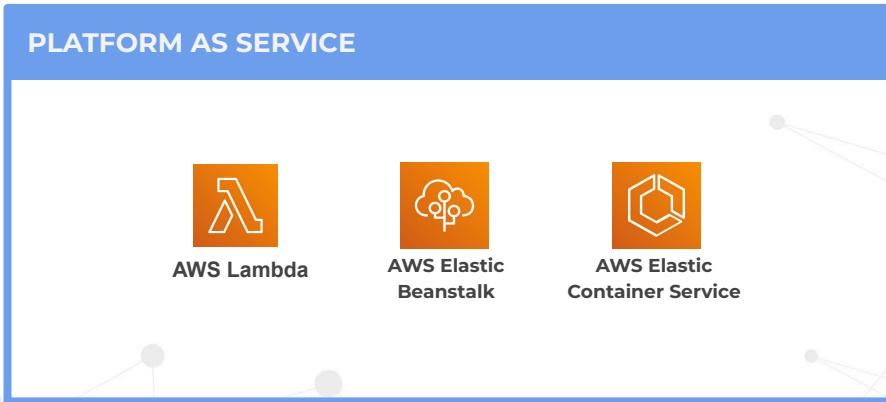


Pay as you go

# Secret Manager vs SSM Parameter Store

Features	Secrets Manager	SSM Parameter Store
Free to use	No	<b>Yes</b>
Built in Password generator	<b>Yes</b>	No
Cross-account access	<b>Yes</b>	No
Automated Secret rotation	<b>Yes</b>	No
Encryption using KMS	<b>Yes</b>	<b>Yes</b>
Integration with RDS,Redshift	<b>Yes</b>	No

# AWS DevOps Services



# AWS Lambda

Run code without thinking about servers



Provide your own code



AWS Lambda

## Administration

- Completely automated administration
- Built-in fault tolerance
- Automatic scaling
- Orchestrate multiple functions
- Fine grained control over performance

## Compatibility

- Extend other AWS services with custom logic
- Connect to relational databases

## Integrations

- Build custom back-end services
- Integrated security model
- Flexible resource model

## Key Benefits

### Serverless Architecture - Zero administration

AWS Lambda is serverless, so there is no infrastructure to manage.



### Continuous Scaling

AWS Lambda automatically scales your application by running code in response to each trigger



### Pay for what you consume

With AWS Lambda, you are charged for every 100ms your code executes and the number of times your code is triggered



### Consistent Performance

With AWS Lambda, you can optimize your code execution time by choosing the right memory size for your function



# AWS Elastic Beanstalk

An easy-to-use service for deploying and scaling web applications and services



Web Applications and Services



AWS Elastic Beanstalk

Wide Selection of Application Platforms & Deployment Options

Monitoring Application Health, Logging, and Tracing

## Operations

- Management and Updates
- Customization of AWS Resources
- Scaling

## Key Benefits

### Fast and simple to begin



Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring

### Continuous Scaling



Elastic Beanstalk automatically scales your application up and down based on your application's specific need using easily adjustable Auto Scaling settings

### Enhance Developer productivity



Elastic Beanstalk provisions and operates the infrastructure and manages the application stack for you, so you don't have to spend the time or develop the expertise

### Complete resource control



Elastic Beanstalk lets you retain full control over the AWS resources powering your application.

# Amazon Elastic Container Service

Highly secure, reliable, and scalable way to run containers



Sensitive and Mission Critical Applications



## Amazon Elastic Container Service

### Key Benefits

#### Serverless Architecture - Zero administration

ECS supports Fargate to provide serverless compute for containers



#### Performance at scale



You can rapidly launch thousands of containers using ECS with no additional complexity

#### Optimized for cost



Realize up to 90% discounts compared to on-demand prices for running stateless and fault tolerant applications

#### Secure and Reliable



ECS launches your containers in your own Amazon VPC, allowing you to use your VPC security groups and network ACLs

### AWS Fargate Support

#### Networking and Security

- Service Discovery
- Service Mesh
- Task Networking
- Load Balancing
- Security

### Hybrid Deployments

#### Development

- Docker Support
- Windows Containers Compatibility
- Local Development
- Repository Support

#### Monitoring and Logging

#### Management

- Task Definitions
- Programmatic Control
- Container Deployments
- Blue/Green Deployments
- Container Auto-Recovery
- Capacity Providers

#### Scheduling

- Task Scheduling
- Service Scheduling
  - Daemon Scheduling
  - Task Placement

# Demo - AWS Secret Manager



# Demo - AWS System Manager Parameter Store



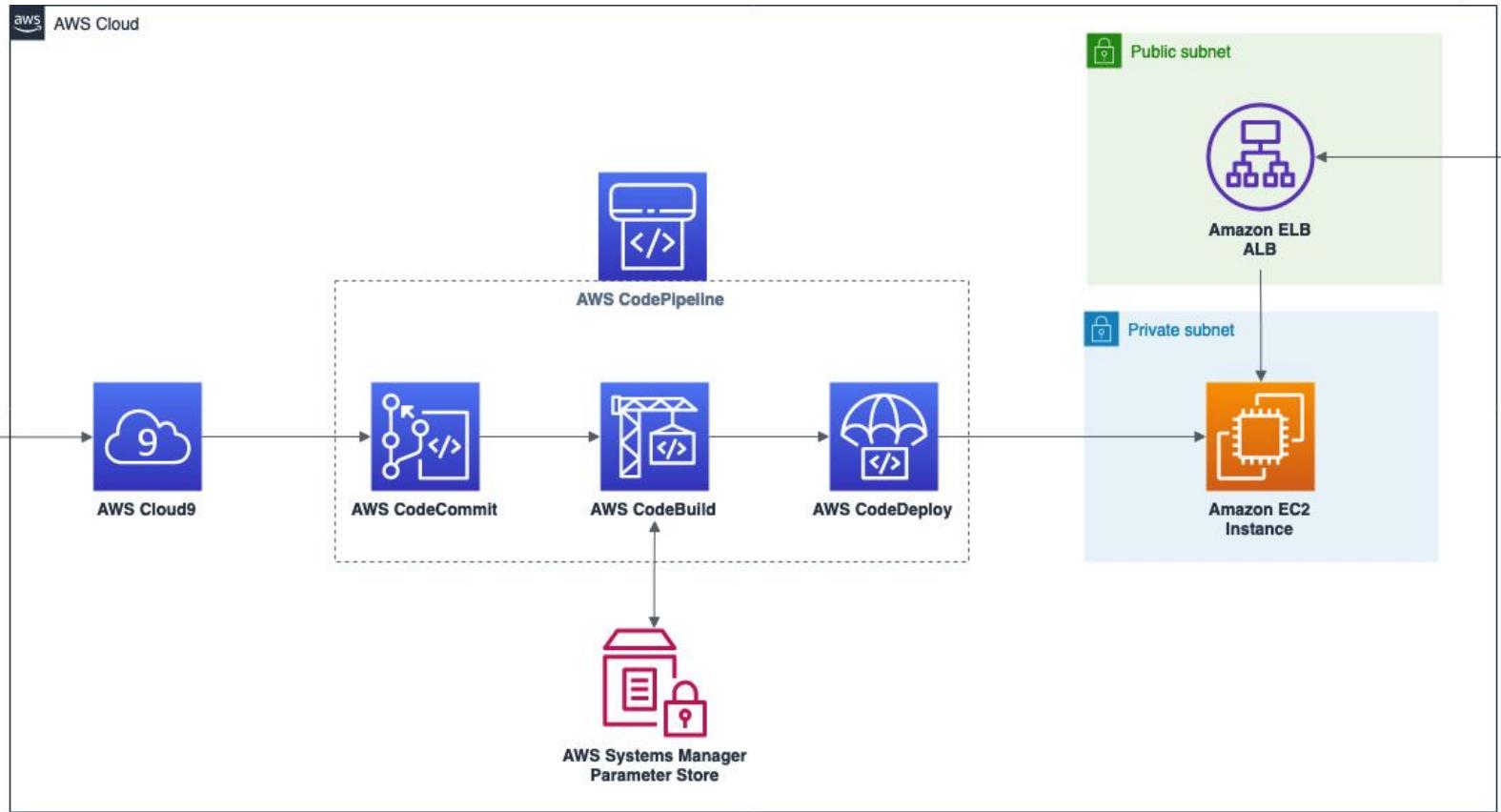
# Demo - AWS Xray, CloudWatch



# Demo - CloudNative CI/CD Pipeline



# AWS CI/CD simple Node.js app



# Key Pairs

- Open the EC2 console at <https://console.aws.amazon.com/ec2>
- In the navigation pane choose KeyPair, and create or import a Key Pair named devops-demo

# Cloudformation

- Open the CloudFormation console at <https://console.aws.amazon.com/cloudformation>
- On the Welcome page, click on Create stack button
- On the Step 1 - Specify template: Choose Upload a template file, click on Choose file button and select the template.yaml located inside deploy directory
- On the Step 2 - Specify stack details: Enter the Stack name as 'cicd-demo'
- On the Step 3 - Configure stack options: Just click on Next button
- On the Step 4 - Review: Enable the checkbox I acknowledge that AWS CloudFormation might create IAM resources with custom names., and click on Create Stack button
- Wait for the stack get into status CREATE\_COMPLETE
- Under the Outputs tab, take a note of ELB value

# Configure Ec2

- CodeDeploy Agent Install

```
sudo yum install -y ruby
```

```
sudo wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install
```

```
sudo chmod +x ./install
```

```
sudo ./install auto
```

```
sudo service codedeploy-agent status
```

- Node.JS Install

```
sudo yum install -y gcc-c++ make
```

```
curl -sL https://rpm.nodesource.com/setup_14.x | sudo -E bash -
```

```
sudo yum install -y nodejs
```

```
npm -v
```

# CodeBuild

- Open the CodeBuild console at <https://console.aws.amazon.com/codebuild>
- Click on Create build project button
- Enter the Project name as 'cicd-demo'
- On Source define Github as the source provider and setup the OAuth between Github and AWS CodeBuild
- On Environment choose Ubuntu for Operational System, Standard for Runtime and aws/codebuild/standard:4.0 as the Image version. After, select Existing service role and search for CodeBuild-cicd-demo
- Uncheck the checkbox Allow AWS CodeBuild to modify this service role so it can be used with this build project
- Click on Create build project button

# CodeDeploy

- Open the CodeDeploy console at <https://console.aws.amazon.com/codedeploy>
- Click on Create application button
- Enter the Application name as 'cicd-demo', select EC2/On-premises for Compute platform and then click on Create button
- Once your application was created, under Deployment groups tab click on Create deployment group button
- Enter the deployment group name as 'cicd-demo'
- Select the Service Role as CodeDeploy-cicd-demo
- On Environment configuration select Amazon EC2 instances, enter for Key 'Name' and for Value 'cicd-demo'
- On Load Balancer configuration select cicd-techtalk-tg
- Click on Create deployment group button

# CodePipeline

- Open the CodePipeline console at <https://console.aws.amazon.com/codepipeline>
- Click on Create pipeline button
- On the Step 1 - Choose pipeline setting: Enter the Pipeline name as 'cicd-demo'
- On the Step 2 - Add source stage:
  - Select Github as the source provider and setup the OAuth between Github and AWS CodeBuild. Provide right Github repo
  - Select cicd-demo for Repository name
  - Select master for Branch name
- On the Step 3 - Add build stage: Select AWS Codebuild for Build provider, and 'cicd-demo' for Project name
- On the Step 4 - Add deploy stage: Select AWS CodeDeploy for Deploy provider, cicd-demo for Application and Deployment group
- On the Step 5 - Review: Click on Create pipeline button

# Test and fix your application

- Access the application opening your browser at the ELB DNS obtained in the step CloudFormation
- Test it using a celebrity picture of your preference and you will receive an error because it cannot find the configuration file
- Fix the configuration file name Using Cloud9, by opening the file app/app.js and changing on the first line from PLEASE-FIX-ME to config
- The pipeline will be started and you can follow its execution on CodePipeline console. It should finish successfully!
- Try again the application!

# CodeCommit Way!

- Open the CodeCommit console at <https://console.aws.amazon.com/codecommit>
- Click on Create repository button, enter the Repository name as 'cicd-demo' and then click on Create button

## Grant access for CodeCommit on IAM

- Open the IAM console at <https://console.aws.amazon.com/iam>
- In the navigation pane select Users, and then click on demo username
- Under Security Credentials tab, click on Generate button of section HTTPS Git credentials for AWS CodeCommit
- Click on Download credentials button and save the generated csv file

## Push code into code-commit

- `git init`
- `git add .`
- `git commit -m "Repo Init"`
- `git remote add origin https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/cicd-demo`
- `git push -u origin master`



A network graph consisting of numerous small, semi-transparent gray circular nodes scattered across the frame. These nodes are interconnected by a web of thin, light-gray lines, creating a sense of a complex, distributed system or community.

**THANK YOU**