AWS Cloud Practitioner Certification Bootcamp

Week - 5

Session 5 - Cloudformation, DevOps, CloudWatch, Cloudtrail

12th Feb, Saturday
7:00 PM to 8:30 PM IST

aws USER GROUP
BANGLADESH

AWS
User Groups

Speakers

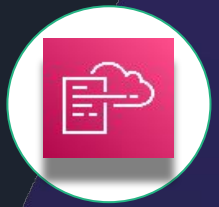
Sanchit Jain

Lead Architect - AWS at Quantiphi

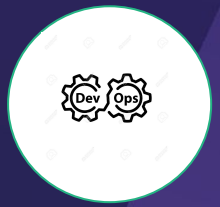
AWS APN Ambassador

Agenda

Re-cap of
Last session

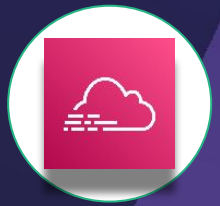


AWS Cloudformation



DevOps on AWS

AWS CloudWatch



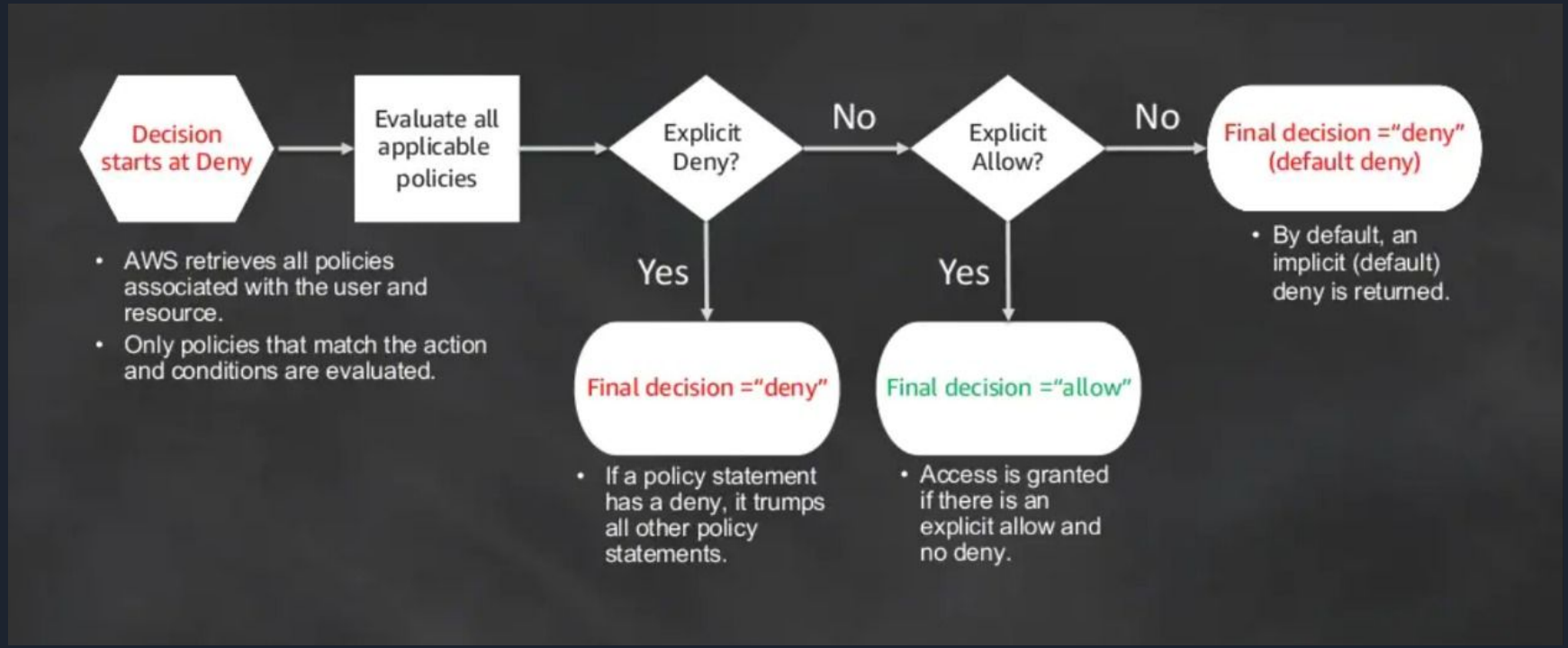
AWS CloudTrail

Re-cap of Last session

AWS IAM - Users and Groups

- Root account created by default, shouldn’t be used or shared as it as Administrative privileges.
- An **IAM user** is an entity that you create in AWS to represent the person or application that uses it to interact with AWS. A user in AWS consists of a name and credentials.
- Users performing same tasks with the same resources can be added to a Group.

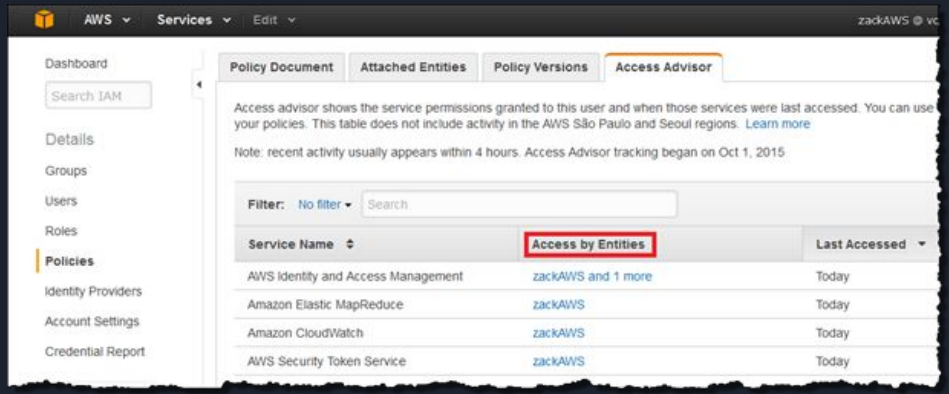
IAM policy Evaluation Logic



AWS IAM Security Tools

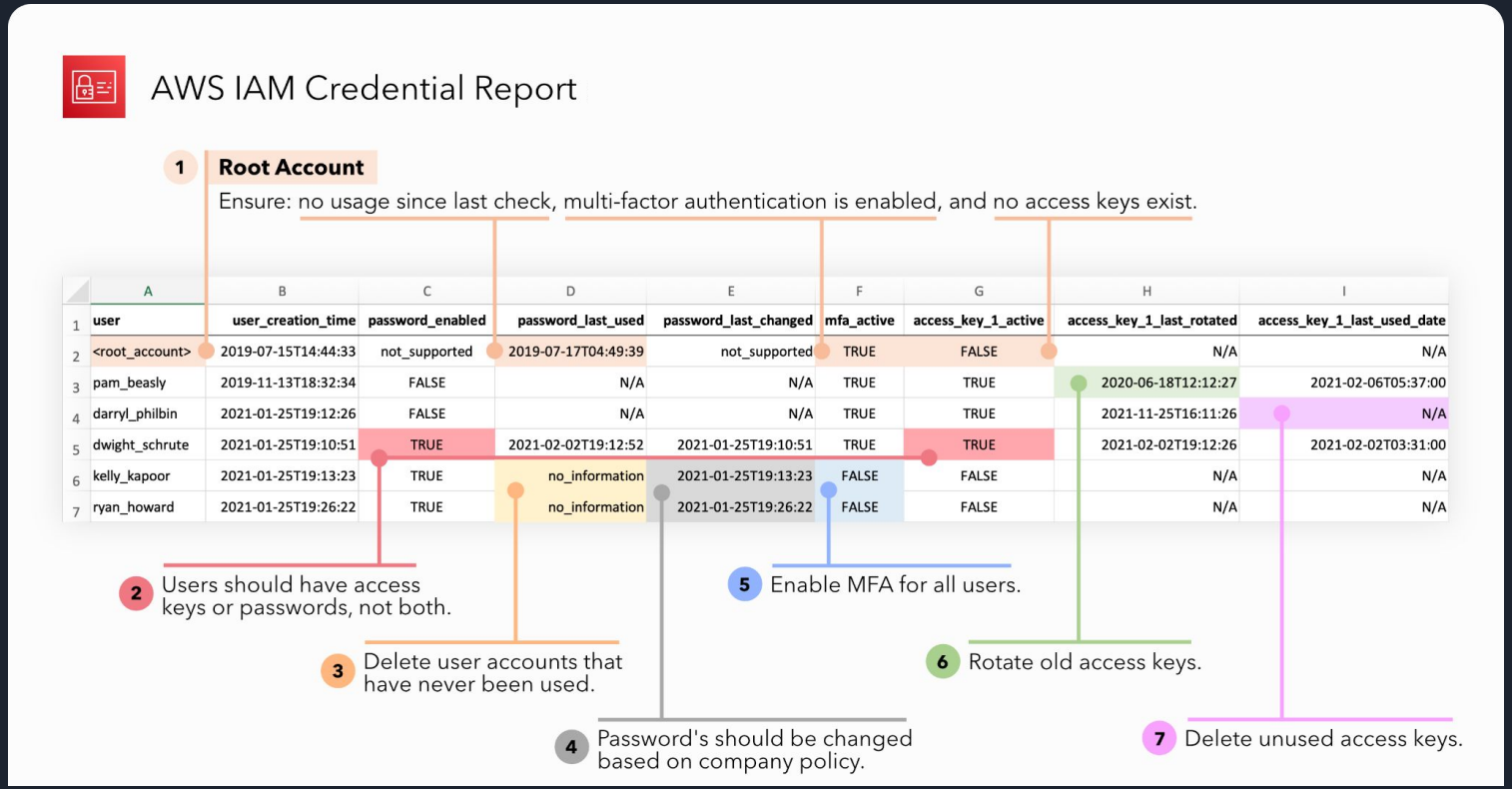
IAM Access Advisor (user-level)

- Access advisor shows the service permissions granted to a user and when those services were last accessed.
- You can use this information to revise your policies.



IAM Credentials Report (account-level)

- a report that lists all your account's users and the status of their various Credentials



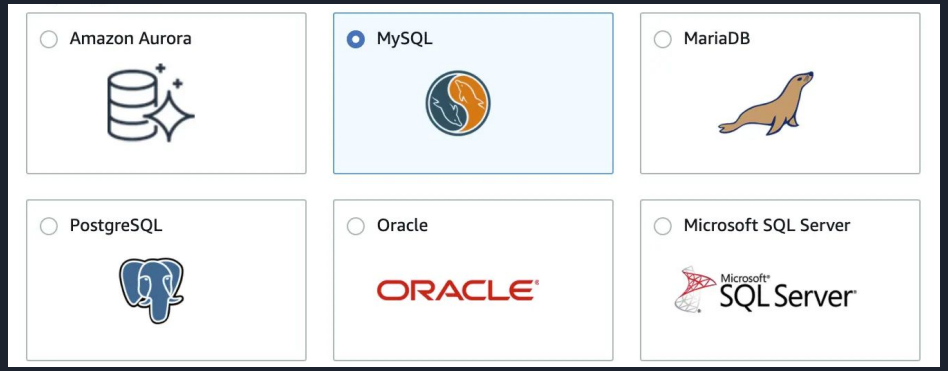
	A	B	C	D	E	F	G	H	I
1	user	user_creation_time	password_enabled	password_last_used	password_last_changed	mfa_active	access_key_1_active	access_key_1_last_rotated	access_key_1_last_used_date
2	<root_account>	2019-07-15T14:44:33	not_supported	2019-07-17T04:49:39	not_supported	TRUE	FALSE	N/A	N/A
3	pam_beasly	2019-11-13T18:32:34	FALSE	N/A	N/A	TRUE	TRUE	2020-06-18T12:12:27	2021-02-06T05:37:00
4	darryl_philbin	2021-01-25T19:12:26	FALSE	N/A	N/A	TRUE	TRUE	2021-11-25T16:11:26	N/A
5	dwight_schrute	2021-01-25T19:10:51	TRUE	2021-02-02T19:12:52	2021-01-25T19:10:51	TRUE	TRUE	2021-02-02T19:12:26	2021-02-02T03:31:00
6	kelly_kapoor	2021-01-25T19:13:23	TRUE	no_information	2021-01-25T19:13:23	FALSE	FALSE	N/A	N/A
7	ryan_howard	2021-01-25T19:26:22	TRUE	no_information	2021-01-25T19:26:22	FALSE	FALSE	N/A	N/A

IAM: Guidelines & Best Practices

- Lock away your AWS account root user access keys
- Create individual IAM users
- Assign users to groups and assign permissions to groups
- Grant least privilege
- Configure a strong password policy for your users
- Use and enforce the use of Multi Factor Authentication (MFA)
- Create and use Roles for giving permissions to AWS services
- Never share IAM users & Access Keys
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Audit permissions of your account with the IAM Credentials Report

Amazon RDS - Instances

- No special handling required. Just point to the new Instance
- A *DB instance class* determines the computation and memory capacity of a DB instance.
- Amazon RDS currently supports the following *DB engines*:



- Each DB engine has a set of parameters in a *DB parameter group* that control the behavior of the databases that it manages.

Amazon Aurora Databases

- Fully managed relational database engine compatible with **MySQL and PostgreSQL**. Includes high-performance distributed storage across AZs.
- Aurora maintains 6 copies of data across 3 AZs for Backup and Recovery.
- Has a master node and Read Replicas(upto 15). Replicas can Auto Scale based on Requests
- Aurora Endpoints
  - Writer endpoint - Points to master node
  - Reader endpoint - Common endpoint for all read-replica(s)
  - Custom endpoints - User defined, for offloading specific heavy analytical tasks

Amazon Aurora Serverless

- Configuring capacity correctly in advance isn't always possible with the provisioned model. It can also result in higher costs if you overprovision and have capacity that you don't use.

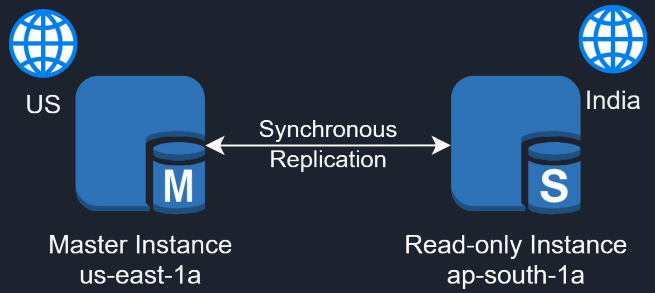
- Checkout Serverless - Create a database endpoint without specifying the DB instance class or size !?

- Auto-Multi-AZ Failover, Always encrypted database

INFINITE POWER!

Amazon Aurora Global

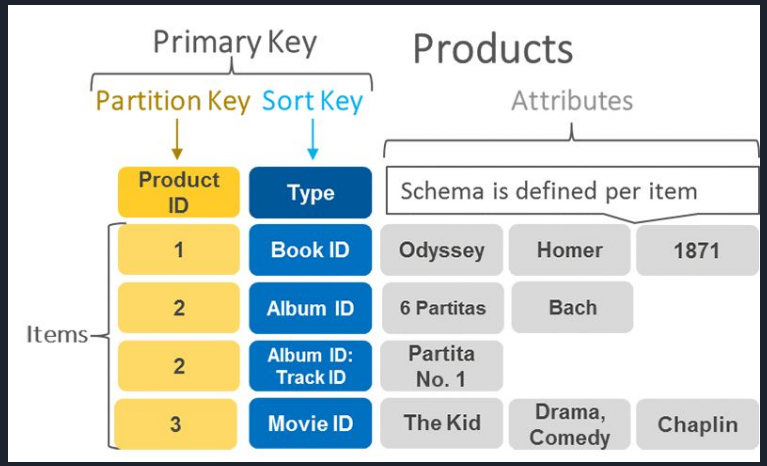
- Enables low latency global reads and fast recovery from the rare outage that affect an entire AWS Region.



- The secondary cluster is read-only, so it can support up to 16 read-only Aurora Replica instances. Note you cannot auto-scale the read-replicas here

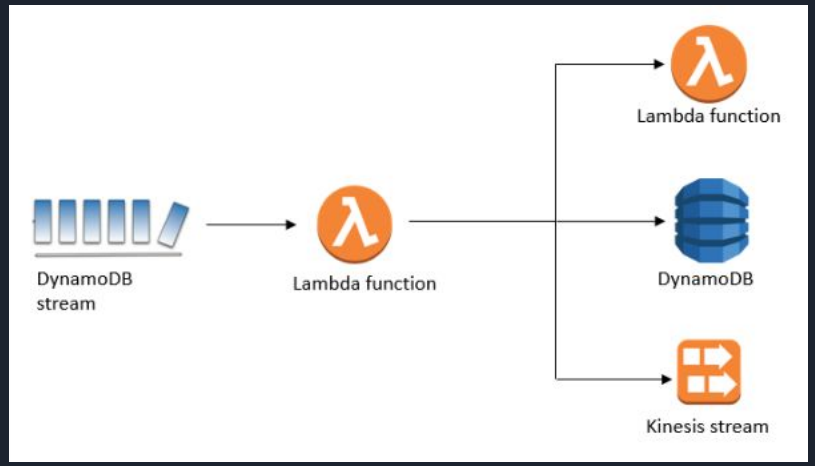
DynamoDB - Introduction

- Fully managed, fast NoSQL key-value database service
- No hardware provisioning, setup and configuration, replication, software patching, or cluster scaling required
- Allows to delete expired items from tables automatically to help reduce storage usage using TTL



DynamoDB Table Structure

Image Ref: AWS Blogs

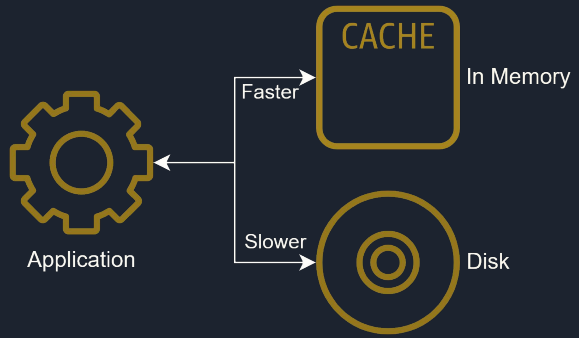


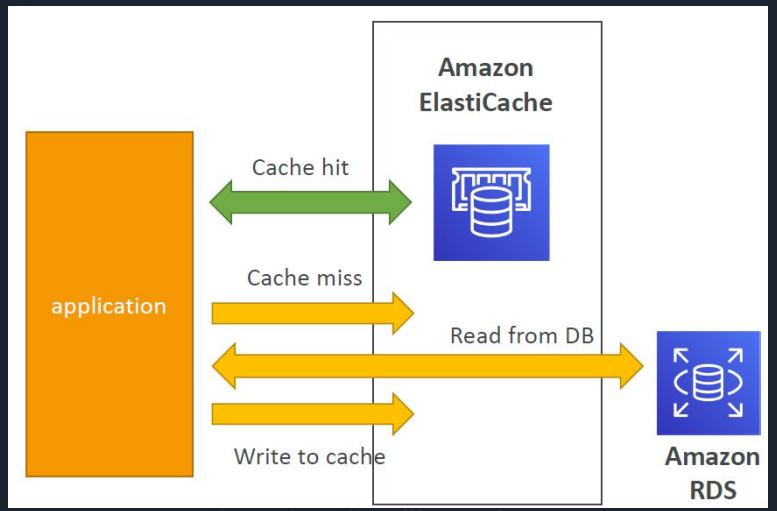
Streams Illustration

Image Ref: AWS Blogs

Elasticache - Introduction

- Provides a high-performance, scalable, and cost-effective caching solution and removes the complexity associated with deploying & managing a distributed cache environment.





Lazy Loading Illustration

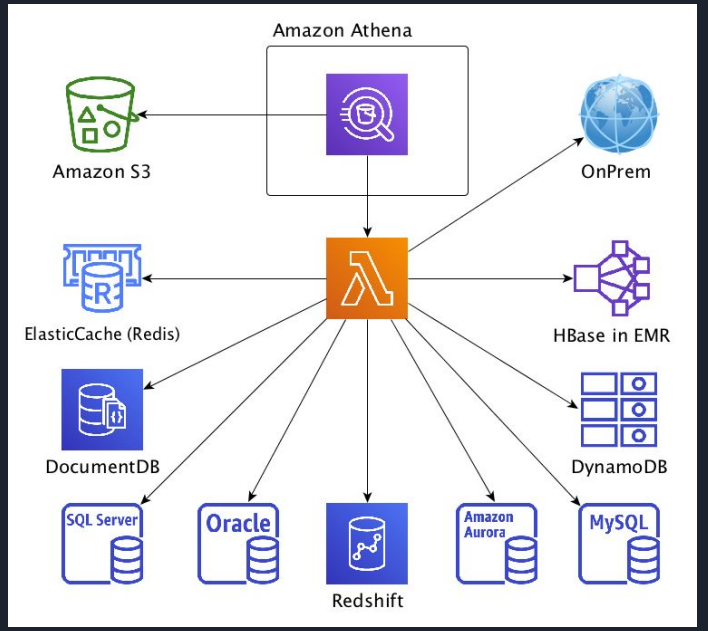
Image Ref. DataCumulus

Redshift - Introduction

- Fully managed, petabyte-scale data warehouse service.
- An Amazon Redshift cluster is a set of nodes, which consists of a leader node and one or more compute nodes.
- Based on PostgreSQL but not used for OLTP workloads but rather used for OLAP (warehouse).
- Columnar storage, Massively Parallel Query Execution(MPP)
- COPY command helps to load data from S3 and similarly UNLOAD helps to offload data into S3.

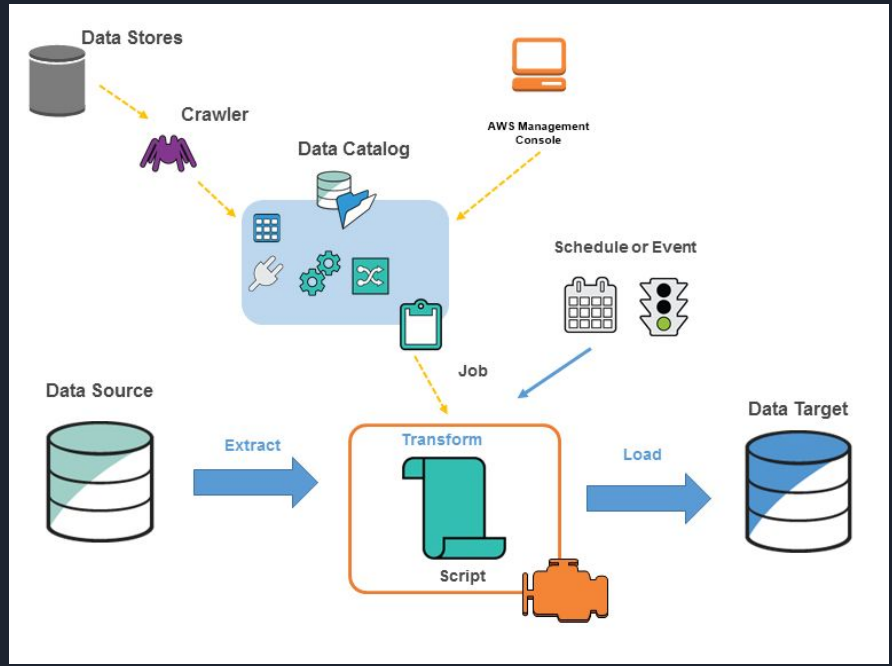
Amazon Athena

- Serverless query service to perform analytics against S3 objects
- Uses standard SQL language to query the files
- Supports CSV,JSON,ORC,Avro,and Parquet(builtonPresto)
- Pricing: $5.00 per TB of data scanned
- Use compressed or columnar data for cost-savings (less scan)
- Use cases: Business intelligence / analytics, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc...
- Exam Tip: analyze data in S3 using serverless SQL, use Athena



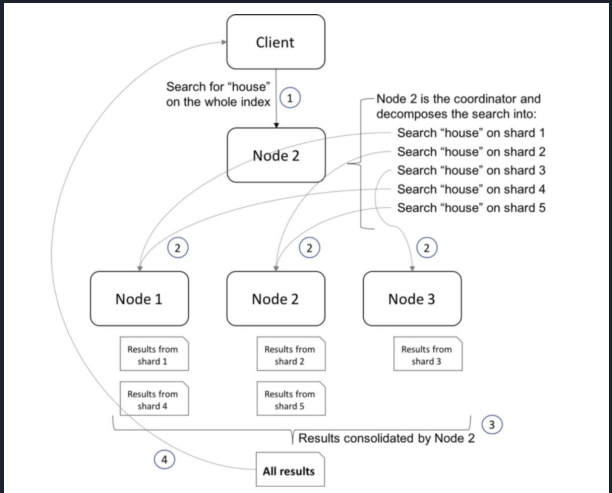
AWS Glue

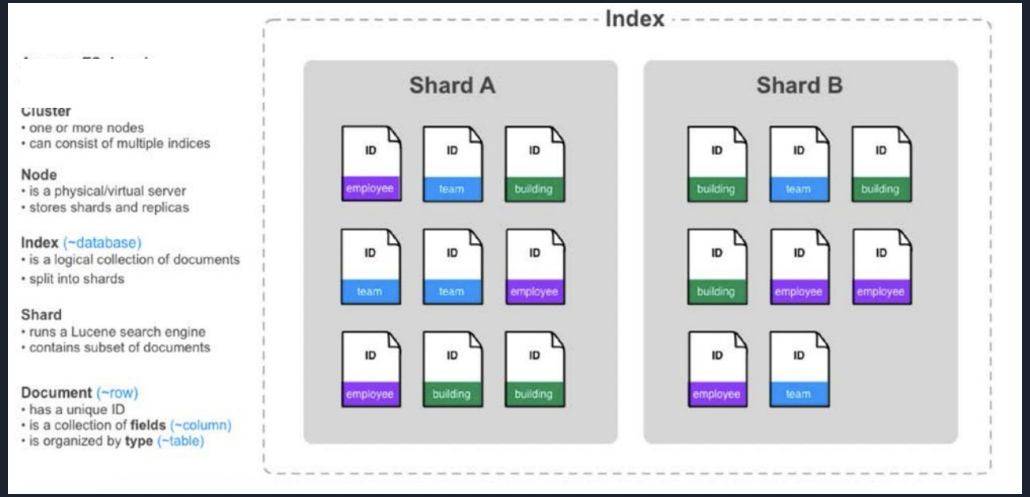
- Fully managed serverless ETL service
- Serverless; runs on a fully managed, auto-scaling Spark environment.
- Glue Components
    - Databases - Tables and Connections
    - Data Catalog - Crawler and Classifiers
    - ETL - Jobs and Triggers



OpenSearch

- Fully open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and clickstream analysis
- Data visualization using OpenSearch Dashboards (the successor to Kibana)
- Also works on pattern matching(like)/partial matches of data quite fast.

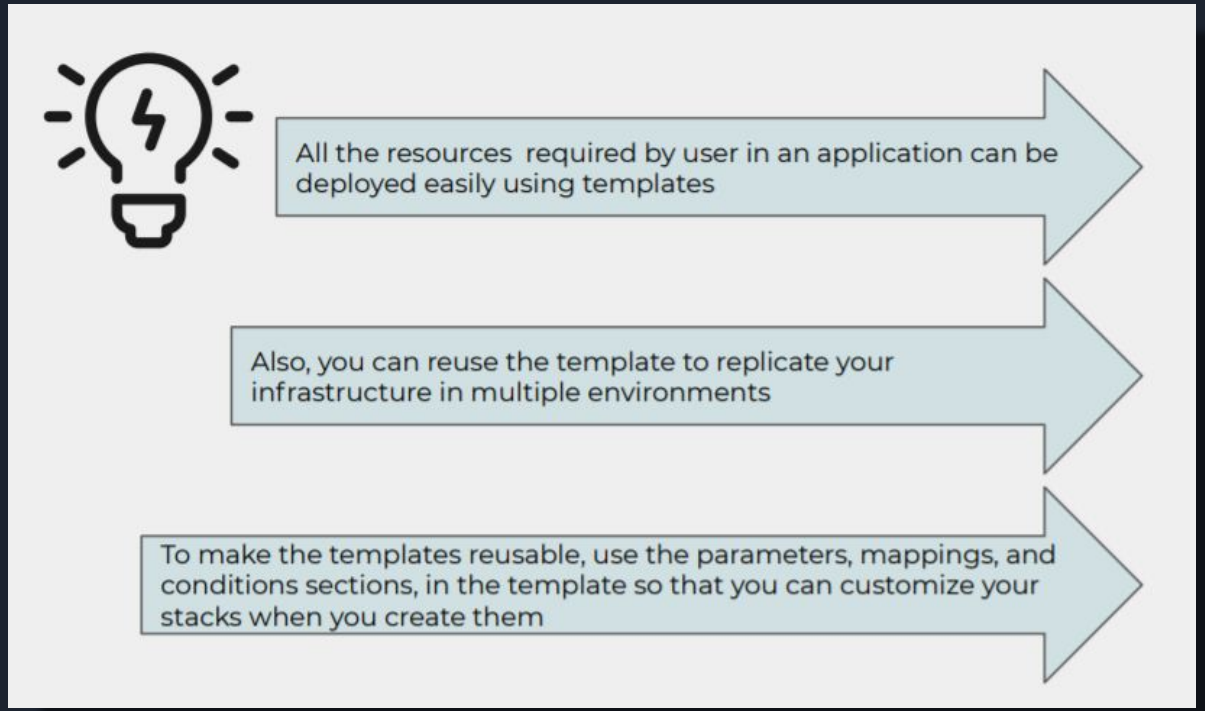




AWS Cloudformation

AWS
User Groups

CloudFormation Overview



CloudFormation Overview

- Infrastructure as a code(IAAC)
- One Click Creation and Deletion
- Declarative in nature
- Code can be version control
- Supports Yaml/Json

Image Credits:
https://memegenerator.net/instance/72724467/why-not-both-json-or-yaml-we-love-cloudformation

Template Structure

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Launch an EC2 Instance running apache and expose
  default page to the internet. Harcoded to work
  only with  US-EAST-1 (N. Virginia)
Parameters:
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.nano
      - t2.micro
Resources:
  WebServer:
    Type: 'AWS::EC2::Instance'
    Properties:
      Tags:
        -
          Key: Name
          Value: Apache Default WebServer
      InstanceType: !Ref InstanceType
      # You shouldn't hardcode, just shown here for example
      ImageId: 'ami-0b898040803850657'
      SecurityGroupIds:
        - !GetAtt SecurityGroup.GroupId
      UserData:
        'Fn::Base64':
          !Sub |
          #!/usr/bin/env bash
          su ec2-user
          sudo yum install httpd -y
          sudo service httpd start
  SecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Enable internet users to access.
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
Outputs:
  PublicIp:
    Value: !GetAtt WebServer.PublicIp
```

Template Sections

MetaData	Additional information about the template
Description	A description of what this template is suppose to do
Parameters	Values to pass to your template at runtime
Mappings	A lookup table. Maps keys to values so you change your values to something else
Conditions	Whether resources are created or properties are assigned
Transform	Applies marcos (like applying a mod which change the anatomy to be custom)
Resources*	A resource you want to create eg. IAM Role, EC2 Instance, Lambda, RDS
Outputs	Values that returned eg. an ip-address of new server created.

CloudFormation Templates **requires** you to **at least list one resource.**

Conditions

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Parameters": {
      "EnvType": {
         "Description": "Environment type.",
         "Default": "test",
         "Type": "String",
         "AllowedValues": ["prod","test"],
         "ConstraintDescription": "must specify prod or test."
   }},
   "Conditions": {
      "CreateProdResources": {
         "Fn::Equals": [
            {
               "Ref": "EnvType"
            },
            "prod"
   ]}},
   "Resources": {
      "EC2Instance": {
         "Type": "AWS::EC2::Instance",
         "Properties": {
            "ImageId": "ami-0ff8a91507f77f867"
         }
      }}}
```

Mappings

```
{
  ...
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
      "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
      "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" }
      }},
  "Resources" : {
    "myEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "32"]},
        "InstanceType" : "m1.small"
      } }
  }
}
```

Functions

- Fn::FindInMap
- Fn::GetAtt
- Fn::GetAZs
- Fn::ImportValue
- Fn::Join
- Fn::Select
- Fn::Split
- Fn::Sub
- Fn::Transform
- Ref

```
Fn::FindInMap: [ MapName, TopLevelKey, SecondLevelKey ]
```

```
Fn::GetAtt: [ logicalNameOfResource, attributeName ]
```

```
Fn::ImportValue: sharedValueToImport
```

```
Fn::Join: [ delimiter, [ comma-delimited list of values ] ]
```

```
Fn::Select: [ index, listOfObjects ]
```

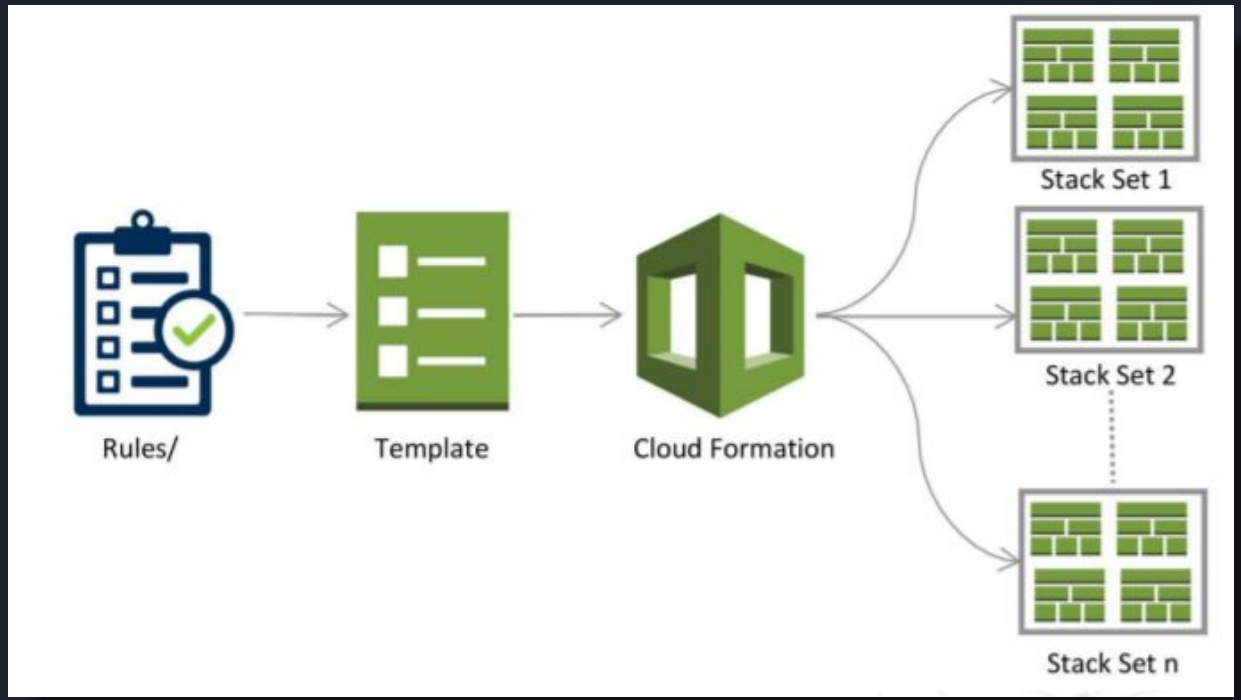
```
Fn::Split: [ delimiter, source string ]
```

```
Ref: logicalName
```

```
Fn::Sub:
  - String
  - Var1Name: Var1Value
    Var2Name: Var2Value
```

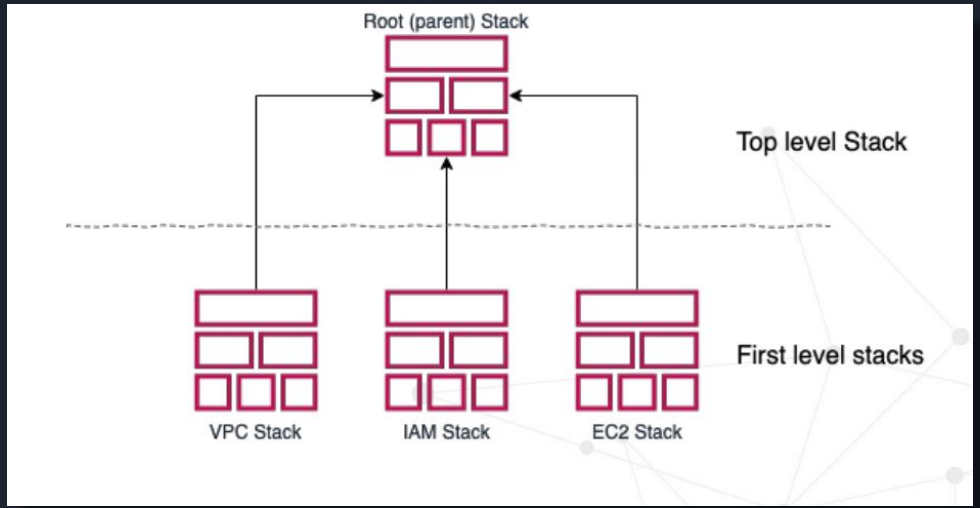
CloudFormation Stacks

- A collection of AWS resources is called a Stack and it can be managed in a single unit



Nested Stacks

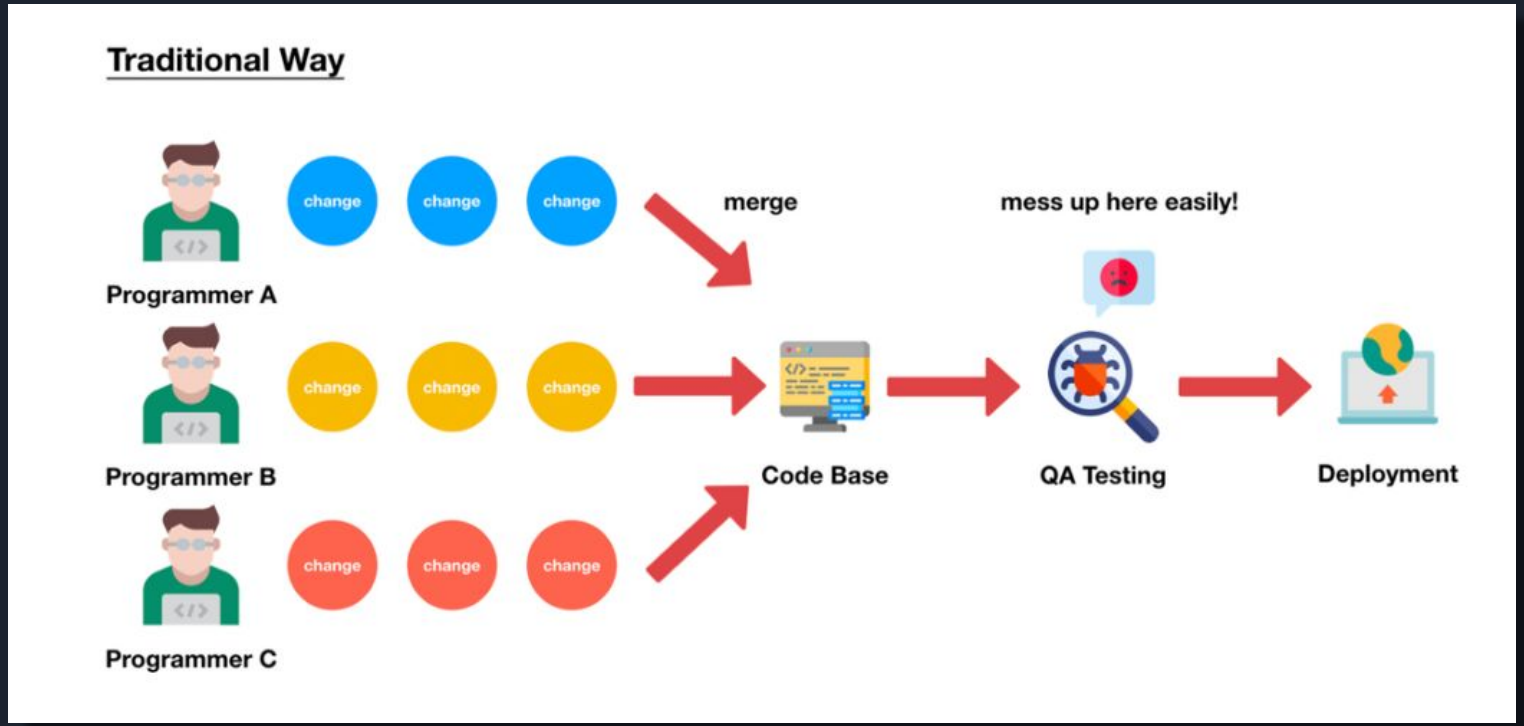
- Nested stacks are stacks created as part of other stacks. We create a nested stack within another stack by using the AWS::CloudFormation::Stack resource
- Nested stacks can themselves contain other nested stacks, resulting in a hierarchy of stacks, as in the diagram below. The root stack is the top-level stack to which all the nested stacks ultimately belong.



DevOps on AWS

AWS
User Groups

Traditional Way



Challenges

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

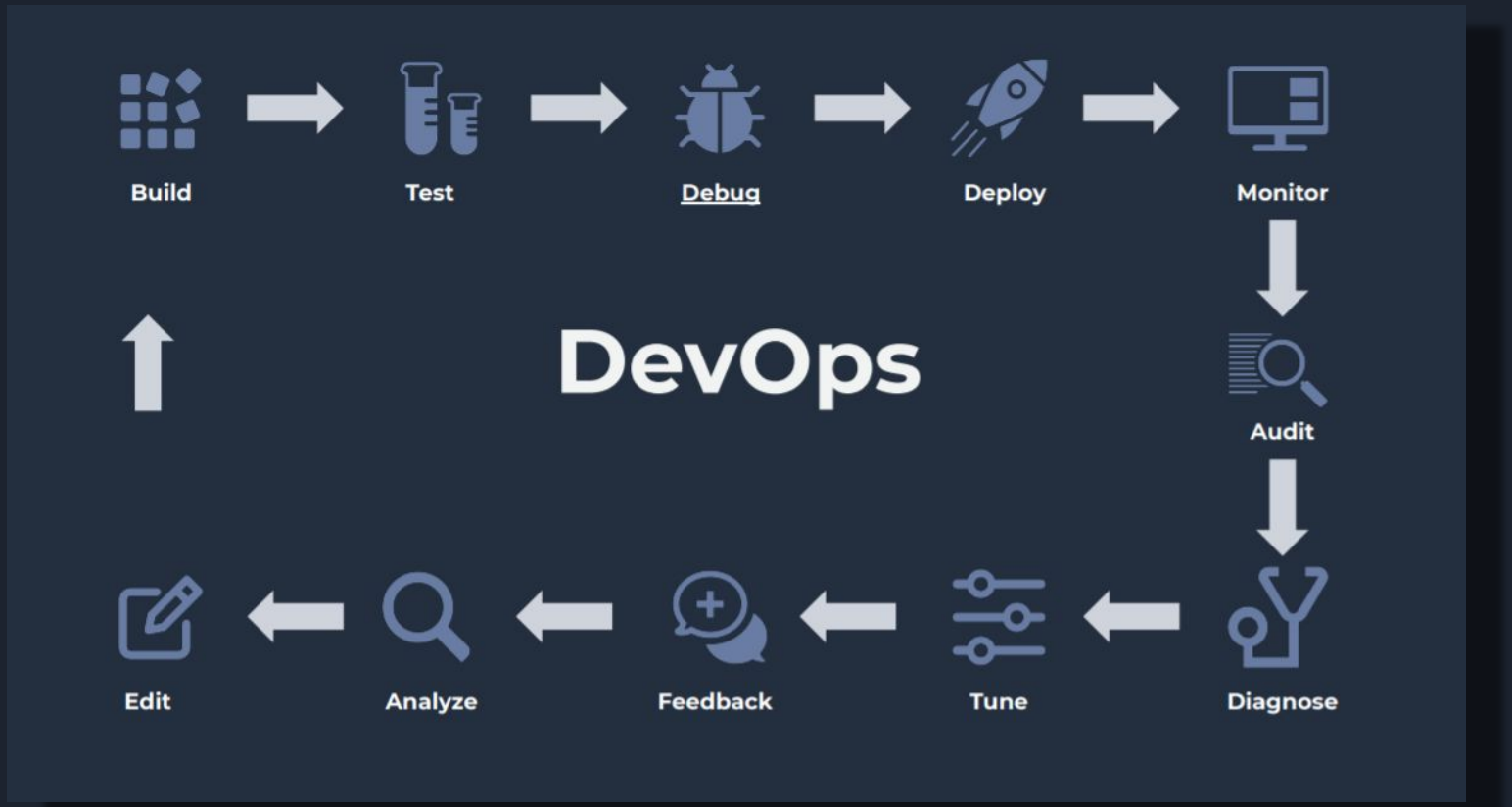
Hacking & Security Threat

Fixing directly in production, instead of a proper hotfix process

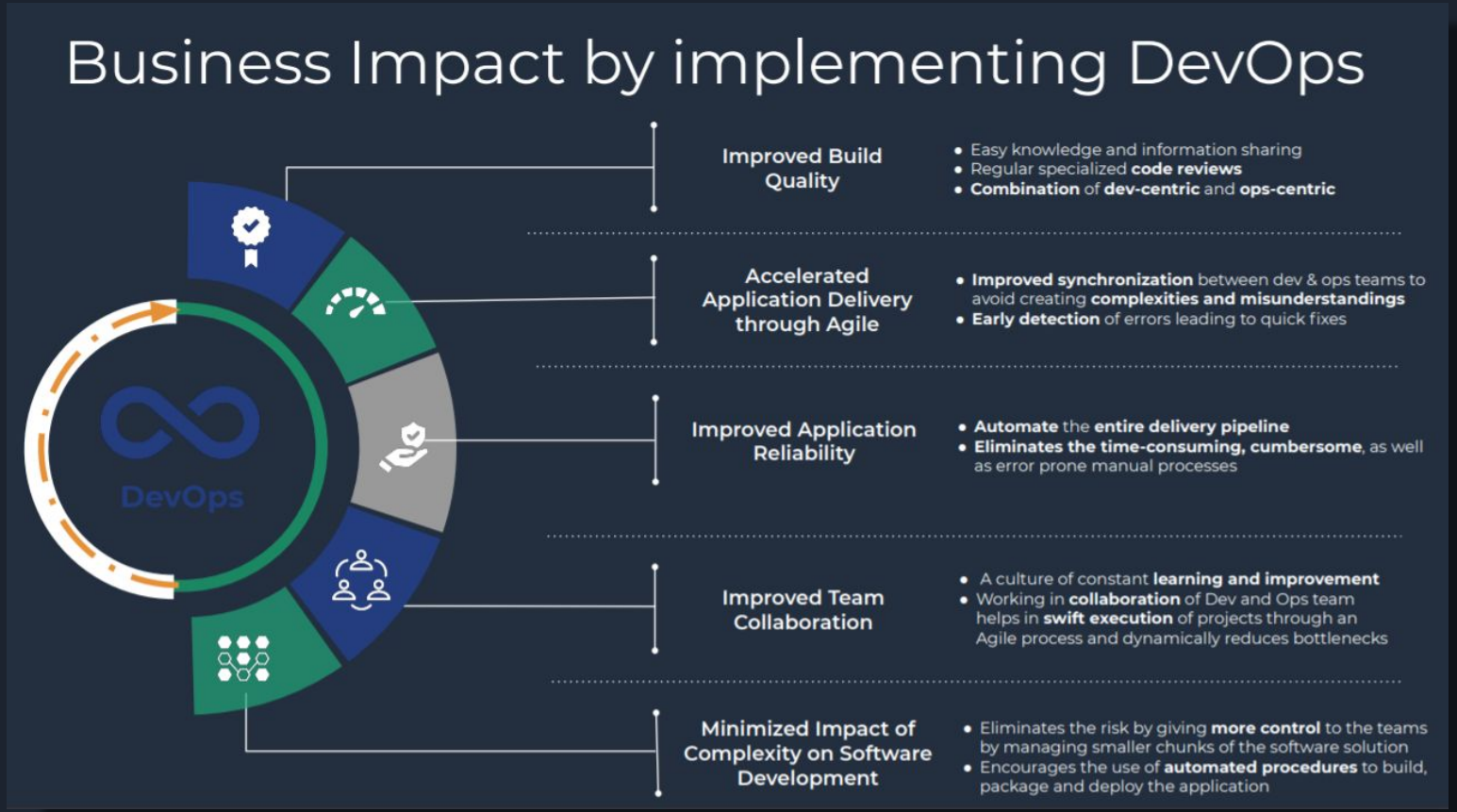
Life Without Devops

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

Devops Cycle



Devops Overview

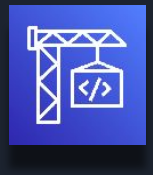


CICD Overview

- Continuous Integration
- Continuous Deployment
- Benefits of CICD
  - Small and frequent code changes
  - Faster Release Rate
  - Easier Rollbacks
  - Increase Team Transparency and Accountability

AWS CICD Tools

AWS CodeCommit

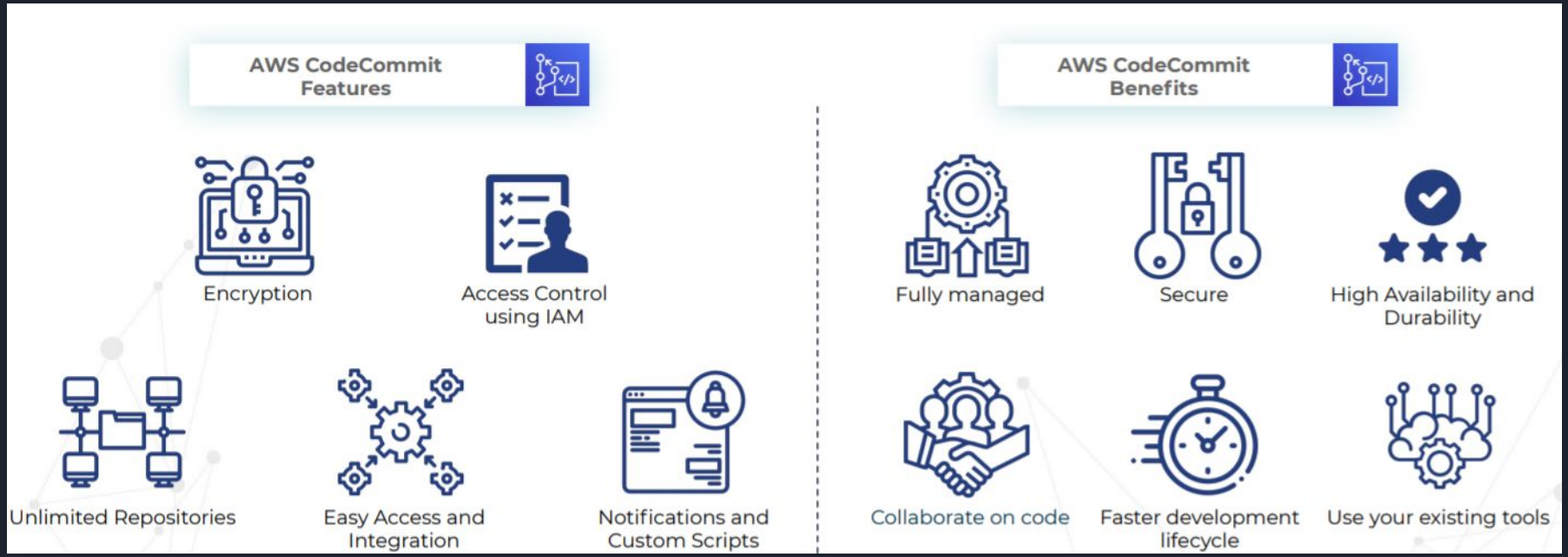


AWS CodeBuild

AWS CodeDeploy

AWS CodePipeline

AWS CodeCommit Overview



CodeBuild Overview

- Fully Managed Build service(alternative to tools like jenkins)
- Serverless
- Secure
- Continuous scaling(no servers to manage)
- Pay only for the build time.
- Supports custom Docker images
- Build instructions defined in buildspec.yml file

buildspec.yml structure

```yaml
version: 0.2
phases:
  install:
    commands:
      - echo Entered the install phase...
      - apt-get update -y
      - apt-get install -y maven
    finally:
      - echo This always runs even if the update or install command fails

  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin
        $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com

  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
      - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
```

buildspecs.yaml structure

```yaml
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG

artifacts:
files:
  - target/messageUtil-1.0.jar

cache:
paths:
  - '/root/.m2/**/*'
```

link: buildspec.yml syntax

CodeDeploy Overview

- Automates application deployment to EC2,Lambda,ECS
- Also supports deployment On-Premise servers
- Secure,Highly scalable,Fully managed
- Deploy instructions defined in appspec.yml file
- Automated Rollback

Note: CodeDeploy does not provision resources

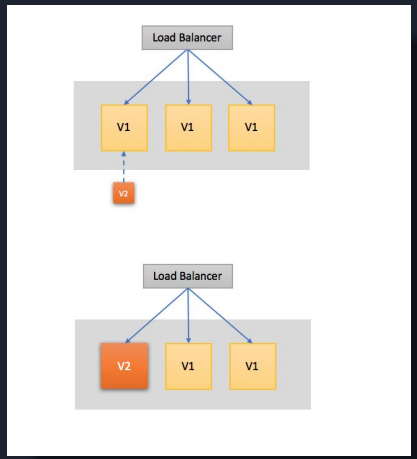
Working of CodeDeploy

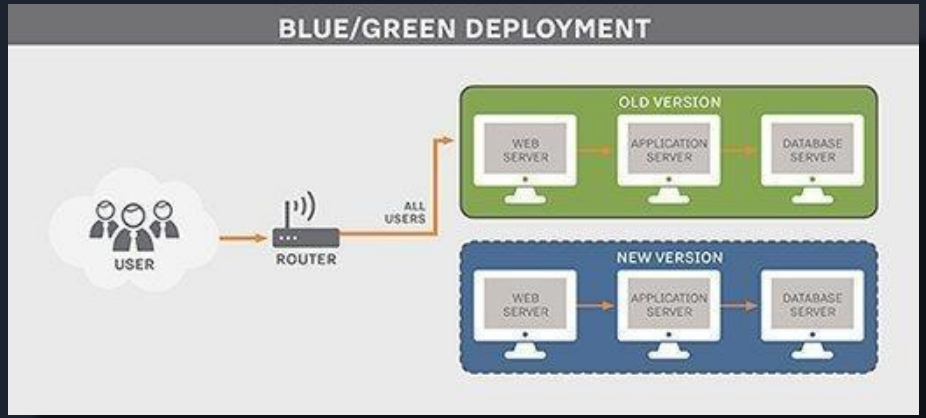
Note: Install Codedeploy agent on EC2 or On Premise machines
***

- Code Deploy sends appspec.yml file to each machine
- Machines deploys the application based on instruction written in appspec.yml file.
- Code Deploy agent reports success/failure of deployment.

Deployment Type

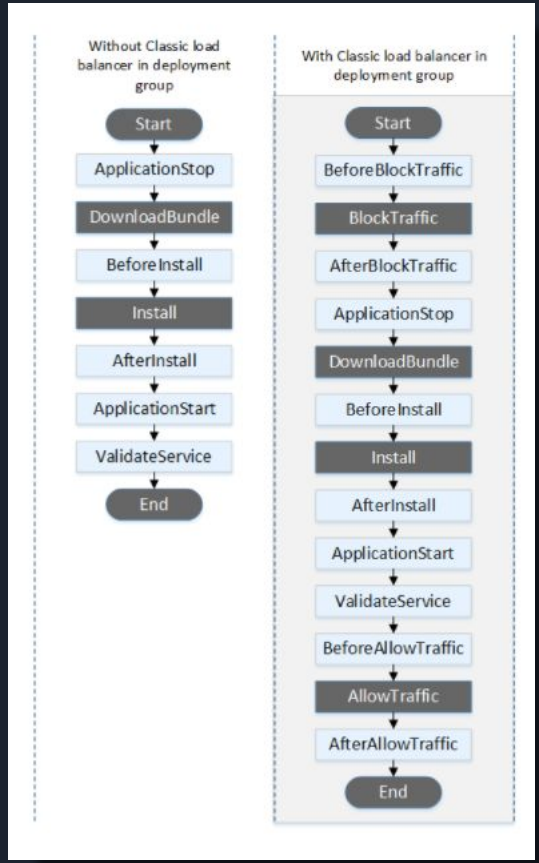
- In place
- Blue/Green



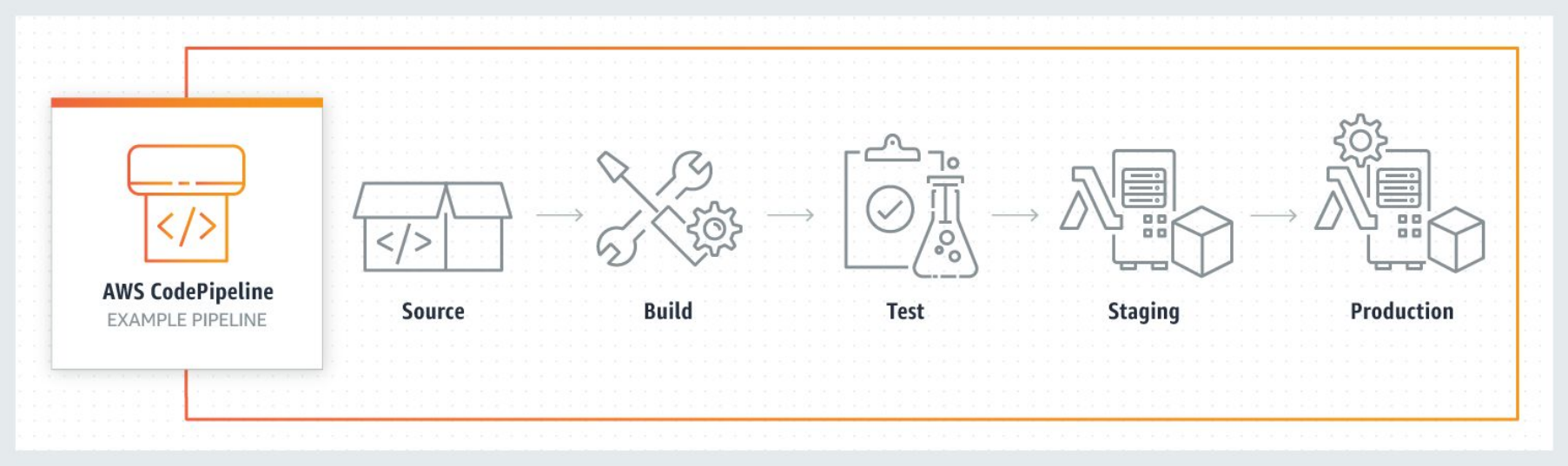


appspec.yml structure

```yaml
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
    - location: scripts/create_test_db.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```



CodePipeline Overview



https://aws.amazon.com/codepipeline/

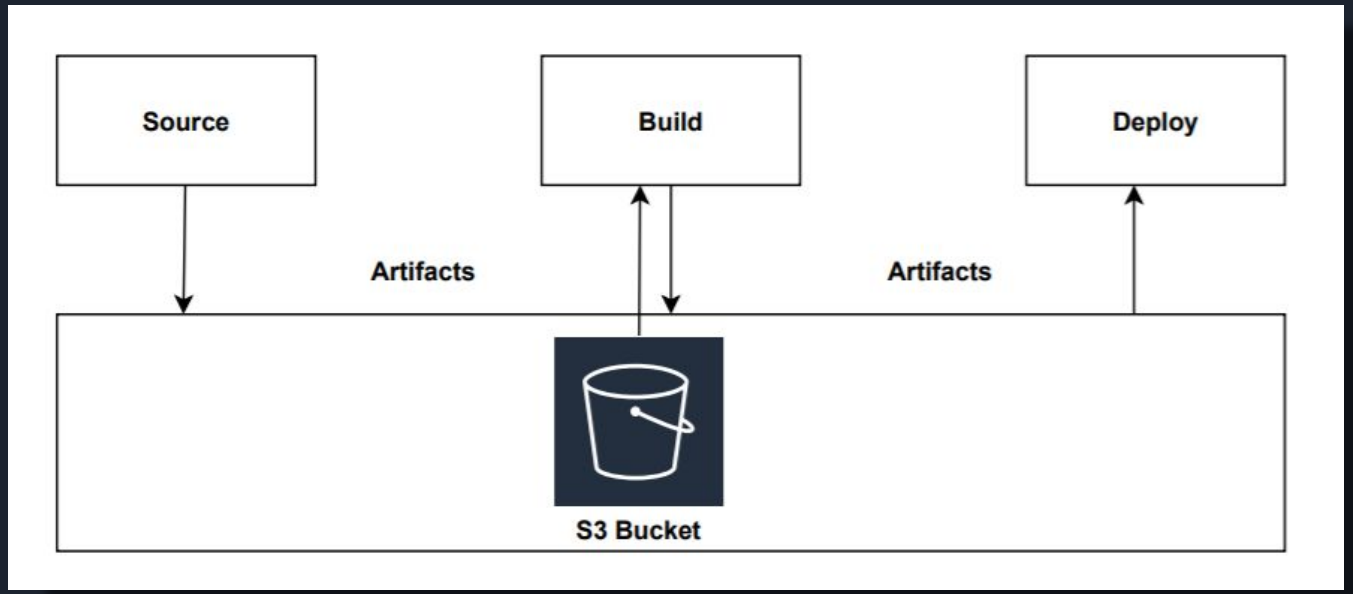
CodePipeline Overview

- Orchestrate entire CI/CD pipeline
- Visual workflow
- Multiple Stages
- Manual Approval
- Supports cross account and cross region deployment
- Pricing: 1 $ per pipeline

https://docs.aws.amazon.com/codepipeline/latest/userguide/welcome-introducing.html

CodePipeline Working

- Each pipeline stage creates **artifacts**
- These artifacts are stored in s3 bucket.
- Each stage takes artifacts stored in s3 as input and dumps as output for next stage.



AWS CloudWatch

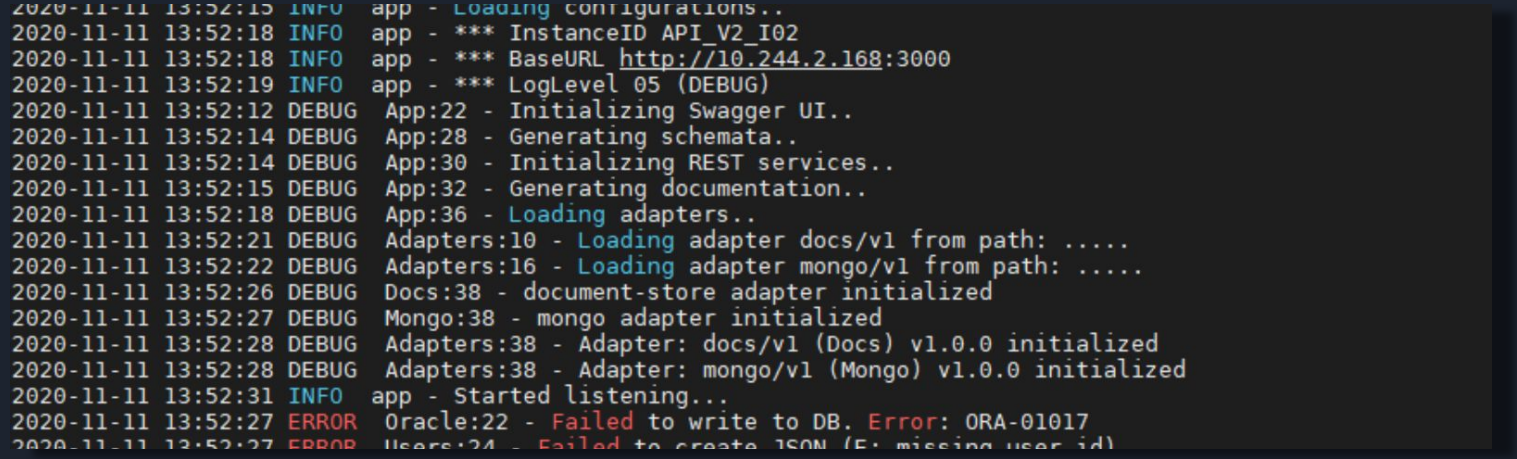
AWS
User Groups

Cloudwatch Logs

Image Credits: https://www.pinterest.com/pin/743094007243620223/

Cloudwatch Logs Overview

- Logging is essential to understand the behaviour of the application and to debug the issues which may impact business.
- Cloudwatch logs is used to collect log files
    - Logs can be used to review any event within a system, including failures, state transformations.

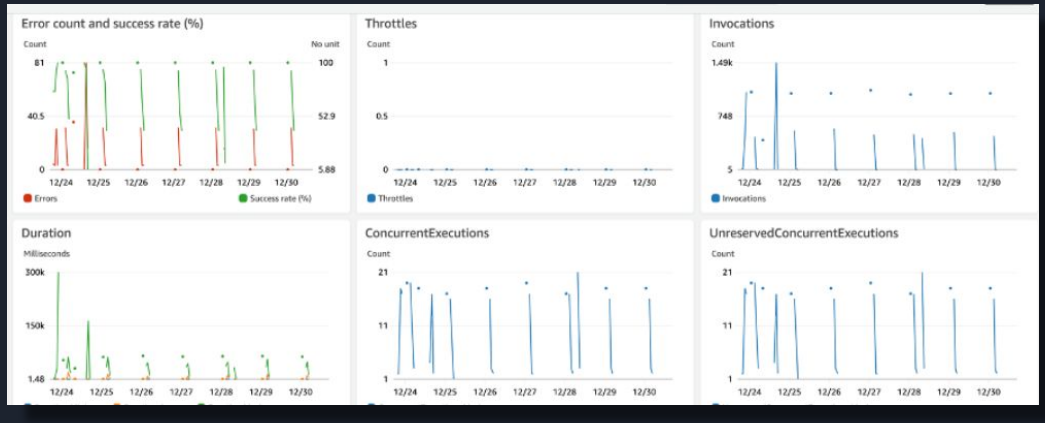


Cloudwatch Logs Overview

- In the production environment, we can’t debug issues without proper log files as they become the only source of information
- Each services has one log group
- Logs can be queried
- Application specific Alarms can be created
- Cloudwatch agents for application logs
- Logs can be encrypted using KMS

Cloudwatch Metrics Overview

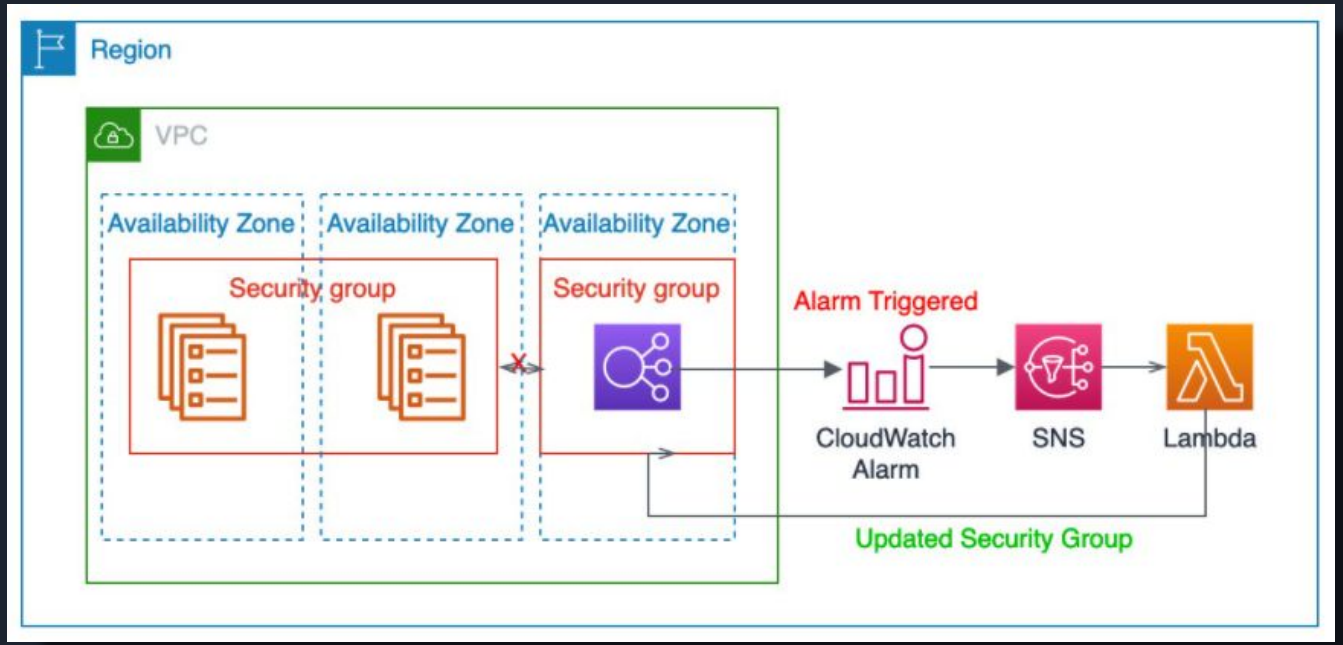
- Time-ordered set of data points published to CloudWatch
- Monitoring Types:
  - Basic Monitoring
    - Collects data every 5 minutes
  - Detailed Monitoring
    - Collects data every 1 minutes



Cloudwatch Alarm Overview

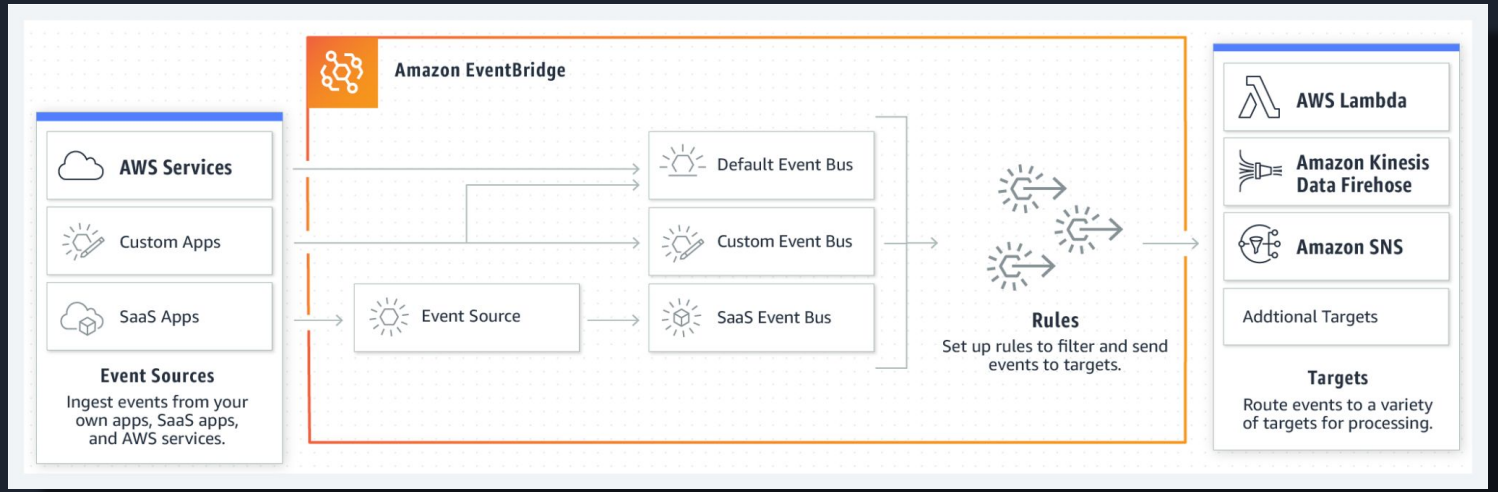
- Alarm State:
  - In alarm
  - Insufficient data
  - OK
- Actions
  - SNS
  - EC2 Action
  - Auto scaling action

Automation using cloudwatch Alarm



EventBridge Overview

- Monitors the real time changes in AWS Services
- Using Rules, AWS services can act automatically when changes occur in other AWS Services
- Pay for what you use



AWS Cloudtrail

AWS
User Groups

AWS CloudTrail Overview

```
"userIdentity": {
        "accessKeyId": "AKEXAMPLE123EJVA",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Bob",
        "principalId": "AIEXAMPLE987ZKLALD3HS",
        "type": "IAMUser",
        "userName": "Bob"
            }
```

What was the API call?
What resources were acted up on?

- API call and the service the API call belongs to

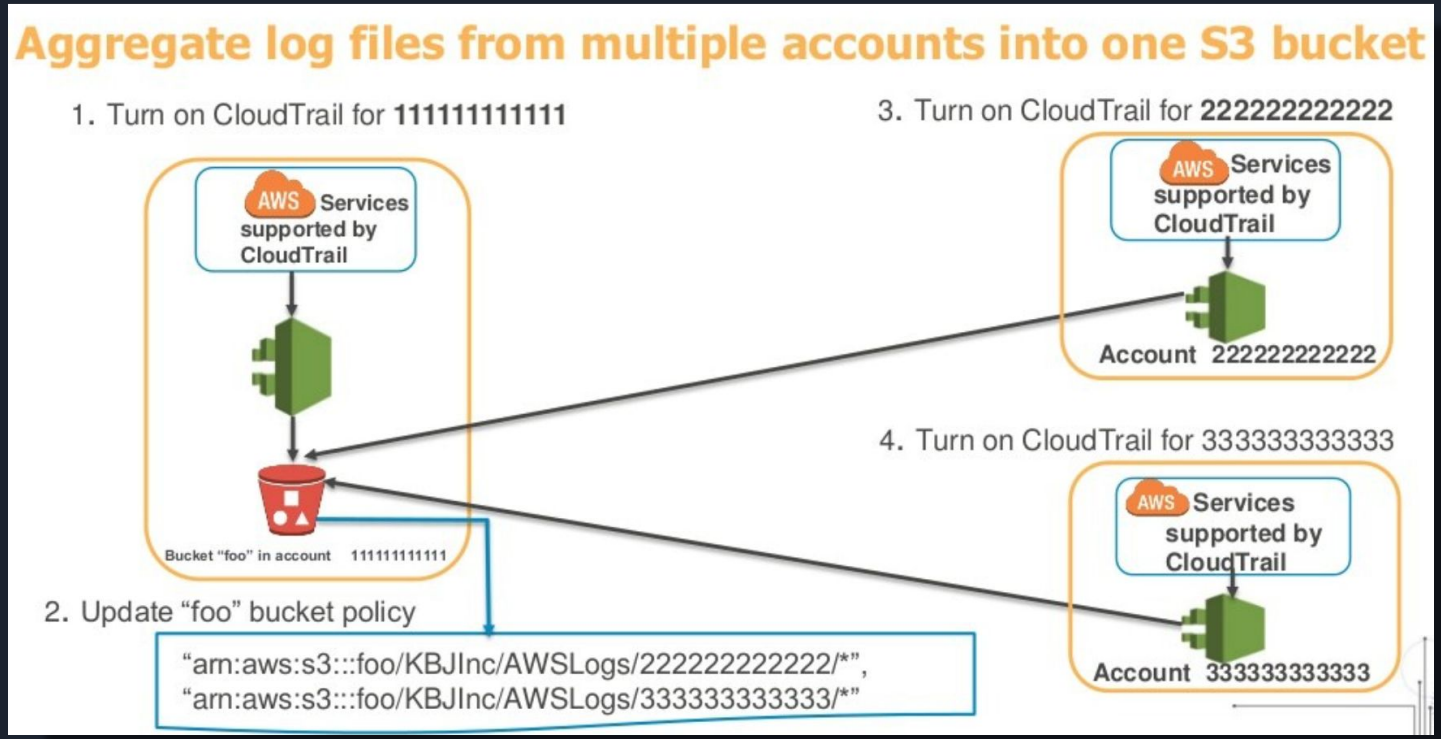
"eventName": "RunInstances"
"eventSource": "EC2"

Where was the API call made from?

- Apparent IP address of the requester making the API call.
- Records the apparent IP address of the requester when making API calls from AWS Management Console.
- AWS region to which the API call was made. Global services (Examples: IAM/STS) will be recorded as us-east-1.

"sourceIPAddress": "54.234.127.135",
"awsRegion": "us-east-1",

AWS CloudTrail Overview



Thank you!

aws USER GROUP
BANGLADESH

AWS
User Groups