

Agenda

Snowball

OpsWork

Elastic
Transcoder

AWS
WorksSpaces

AWS AppSync

AWS Glue

AWS CICD

AWS ECS

AWS EKS

Step Functions
& SWF

EMR

CloudFormation

TrustedAdvisor



What is Snowball?

- A petabyte-scale data transport solution that uses secured **physical devices** to transfer large amounts of data into and out of the AWS Cloud.

Why to use it?

- Simple, fast, secure (256-bit encryption), and highly scalable.
- Saves cost! As little as 1/5th the cost of transferring data via high-speed Internet.
- Addresses concerns of high network costs, long transfer times, and security.
- No need to write any code.
- No hardware purchase necessary.
- Common use cases: Migration of analytics data, genomics data, video libraries, image repositories, backups, and to archive part of data center shutdowns, tape replacement or application migration projects.



Features:

- Rain and dust Resistant
- Tamper-resistant case
- Sizes: 50TB and 80TB
- 10G Network
- All data encrypted end to end



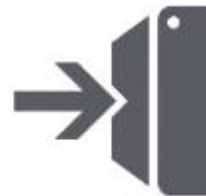
1. Create an import job
in AWS Management Console



2. Receive the Snowball
appliance



3. Connect the AWS Snowball
device to local network



4. Transfer the data



5. Return the device
To AWS



6. Monitor the import status

AWS Snowball Edge



What is Snowball Edge?

- Similar to Snowball but has on-board storage and compute power for selective AWS capabilities
- In addition to transferring data to AWS, Snowball Edge can undertake local processing and edge computing workloads.
- 100 TB of storage capacity, twice the storage space of the original Snowball device.

Differences between Snowball and Snowball Edge

Use Case	Snowball	Snowball Edge
Import data into Amazon S3	✓	✓
Export from Amazon S3	✓	✓
Durable local storage		✓
Local compute with AWS Lambda		✓
Amazon EC2 compute instances		✓
Use in a cluster of devices		✓
Use with AWS Greengrass (IoT)		✓
Transfer files through NFS with a GUI		✓





What is Snowmobile?

- Exabyte-scale data transfer service used to move extremely large amounts of data to AWS.
- You can transfer up to 100PB per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck.

AWS OpsWorks



- AWS OpsWorks is a configuration management service that helps you configure and operate applications in a cloud enterprise by using Puppet or Chef.
- Help development and operations teams manage applications and infrastructure.
- **AWS OpsWorks for Puppet Enterprise**
 - Lets you create AWS-managed Puppet master servers.
 - A Puppet master server manages nodes in your infrastructure, stores facts about those nodes, and serves as a central repository for your Puppet modules.



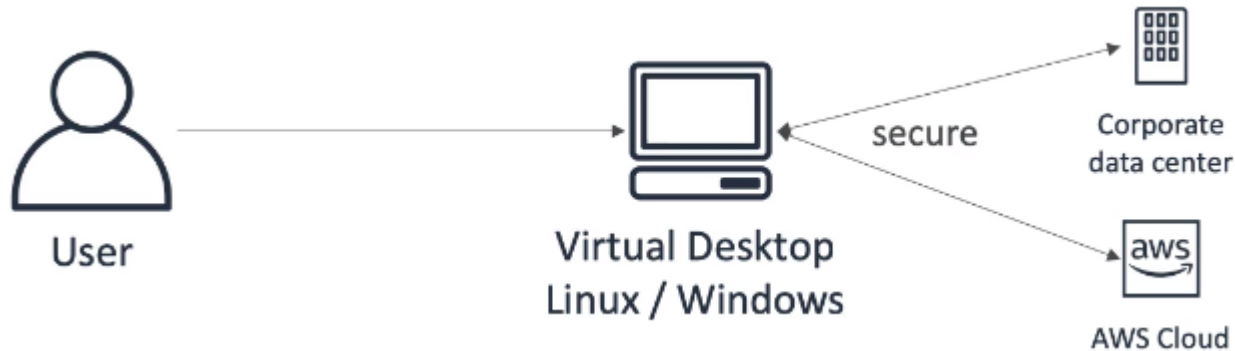
- **AWS OpsWorks for Chef Automate**
 - Lets you create AWS-managed Chef servers that include Chef Automate premium features, and use the Chef DK and other Chef tooling to manage them.
 - Chef Automate is an included server software package that provides automated workflow for continuous deployment and compliance checks.
 - AWS OpsWorks for Chef Automate installs and manages both the Chef server and Chef Automate by using a single Amazon Elastic Compute Cloud instance.
- **AWS OpsWorks Stacks**
 - Provides a simple and flexible way to create and manage stacks and lets you deploy and monitor applications in your stacks.

Elastic Transcoder

- Convert media files (video + music) stored in S3 into various formats for tablets, PC, Smartphone, TV ,etc..
- Feature: bit rate optimization, thumbnail, watermarks, captions, DRM, progressive download, encryption.
- 4 Components:
 - Jobs : what does the work of the transcoder.
 - Pipeline : Queue that manages the transcoding job.
 - Presets: Templates for converting media from one format to another.
 - Notifications: SNS for example.
 - Pay for what you use, fully managed, scales automatically.

AWS WorkSpaces

- Managed, secure, cloud desktop
- Great to eliminate management of on-premises VDI
- Secure, Encrypted, Network Isolation
- Integrated with Microsoft Active Directory



AWS AppSync



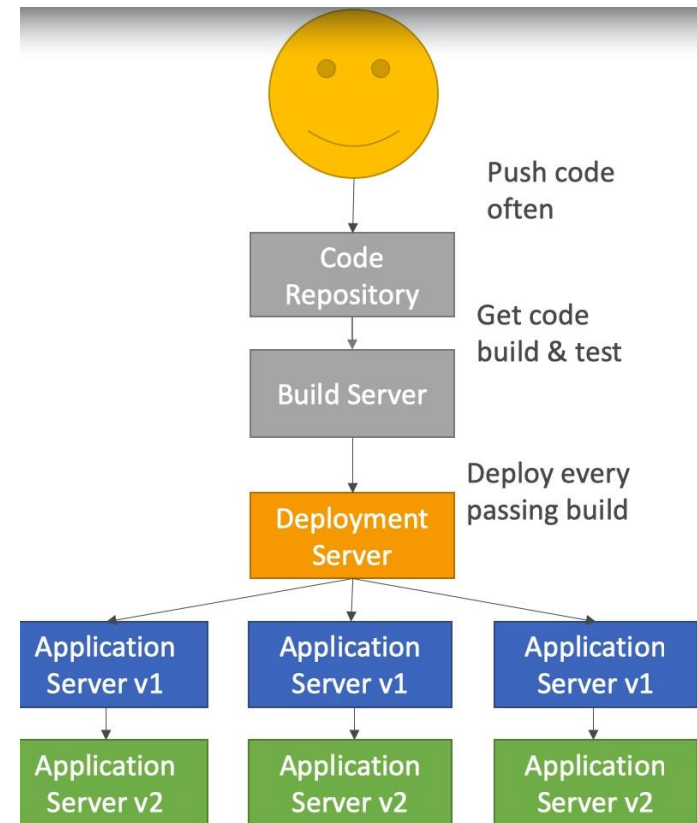
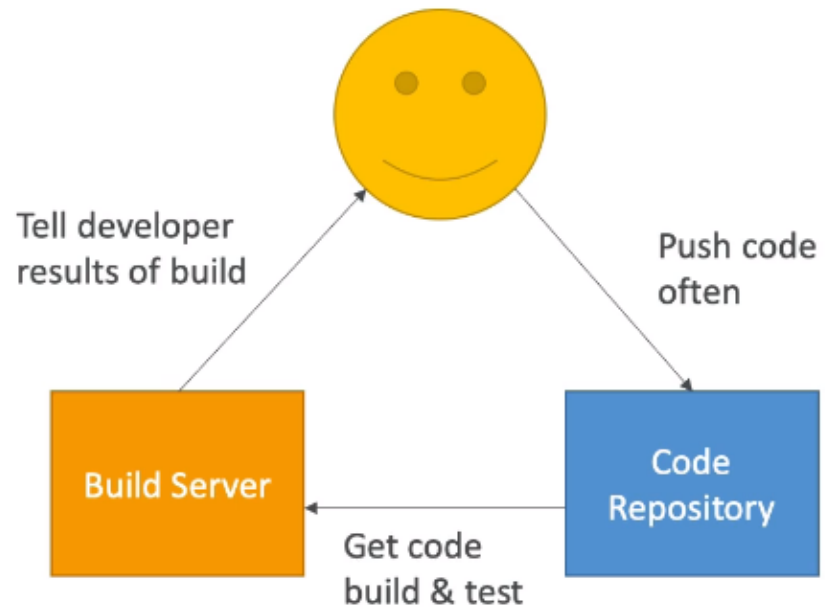
- Store and sync data across mobile and web apps in real-time.
- Makes use of GraphQL (mobile technology from Facebook)
- Client code can be generated automatically
- Integrations with DynamoDB/Lambda
- Real-time subscriptions
- Offline data synchronization (replaces Cognito Sync)
- Fine Grained Security

AWS Glue

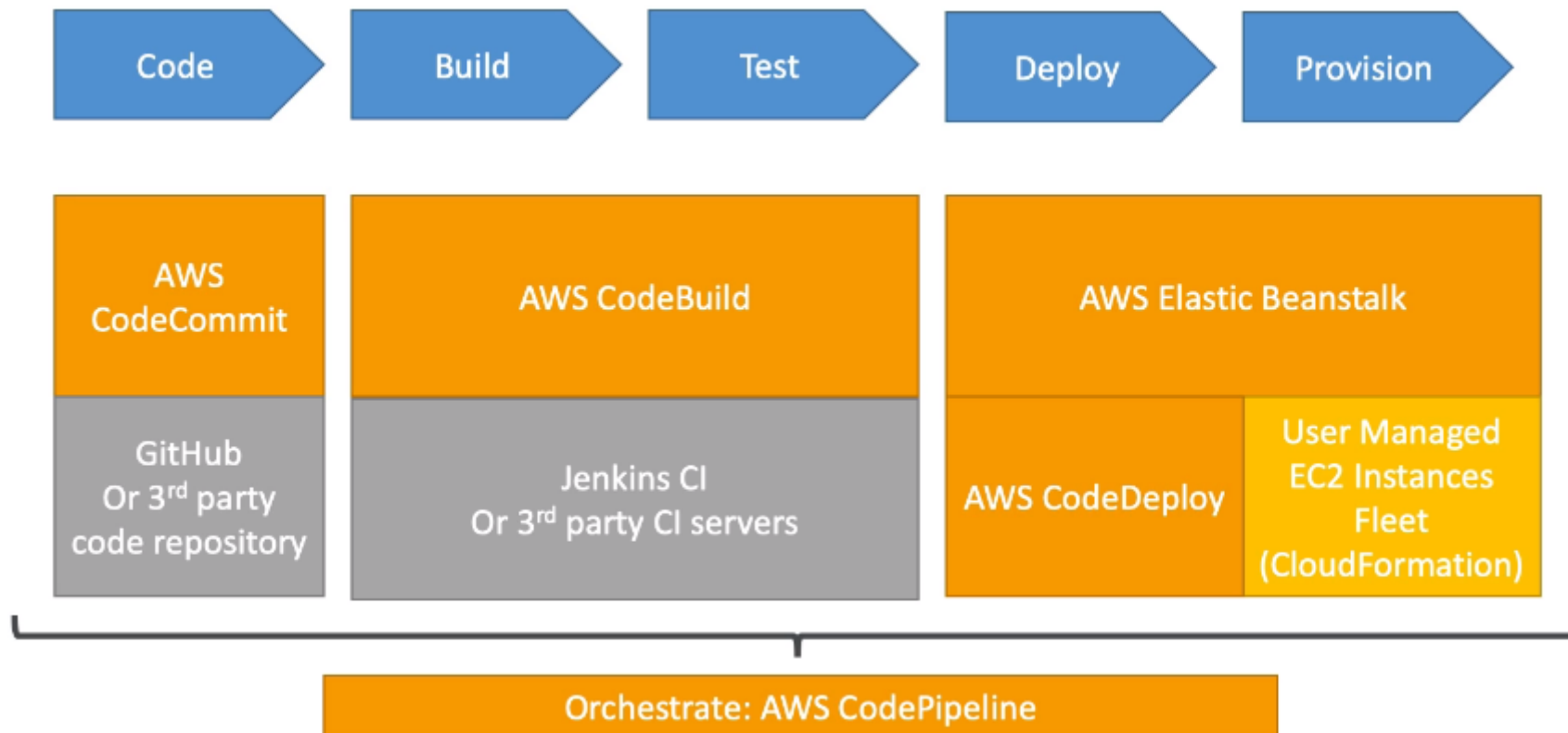
- Fully managed ETL (Extract, Transform & Load) service
- Automate time consuming steps of data preparation for analytics
- Serverless, pay as you go, fully managed, provisions Apache Spark
- Crawls data sources and identifies data formats (schema inference)
- Automated Code Generation
- Sources: Aurora, RDS, RedShift & S3
- Sinks: S3, RedShift, etc..
- Glue Data Catalog: Metadata (definition & schema) of the source tables.

AWS CICD

- Continuous Integration / Continuous Deployment



Technology Stack for CICD



AWS ECS – Elastic Container Service

- ECS is a container orchestration service.
- ECS helps you run Docker containers on EC2 machines.
- ECS is complicated and made of:
 - ECS Core: Running ECS on user-provisioned EC2 instances.
 - Fargate: Running ECS tasks on AWS provisioned compute (serverless)
 - EKS: Running ECS on AWS provisioned Kubernetes (on EC2)
 - ECR: Docker Container Registry hosted by AWS
 - ECS & Docker are very popular for microservices.
 - IAM security and Roles are at ECS tasks level.

Fargate

- When you launch an ECS cluster, we have to create our EC2 instances.
- If we need to scale, we need to add EC2 instances and manage infrastructure.
- With help of Fargate, it's all serverless means you don't provision EC2 instances and just create tasks and AWS will run containers for us.
- To scale you increase the task number not EC2.

Amazon EKS



- Stands for Elastic Kubernetes Service
- Way to launch managed Kubernetes clusters on AWS
- Kubernetes is an open-source system for automatic deployment, scaling and management of containerized (usually Docker) application.
- Similar like ECS but different API
- EKS supports EC2 if you want to deploy worker nodes or Fargate to deploy serverless containers.
- Use Case: If your company is already using Kubernetes on-premises or in another cloud, and wants to migrate to AWS using Kubernetes.

AWS Step Functions

- Build Serverless visual workflow to orchestrate your Lambda functions.
- Represents flow as a JSON state machine.
- Features: sequence, parallel, conditions, timeouts, error handling..
- Integrate with EC2, ECS, API Gateway, On Premise Servers
- Maximum execution time of 1 year
- Possibility to implement human approval feature
- Use Cases: Order Fulfilment, Data Processing

Visual Workflow in Step Functions



■ Success ■ Failed ■ Cancelled ■ In Progress



■ Success ■ Failed ■ Cancelled ■ In Progress



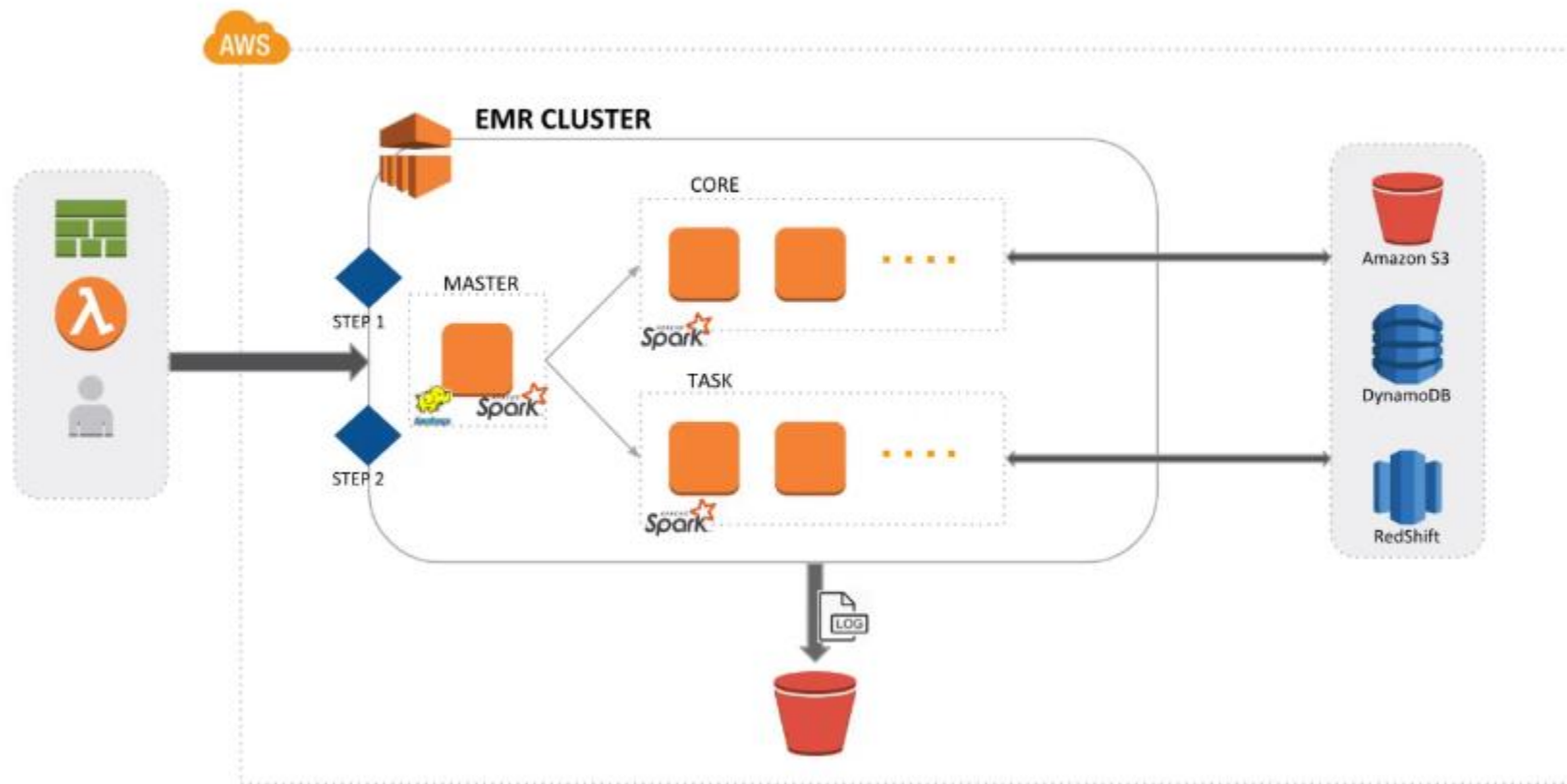
AWS SWF – Simple Workflow Service

- Coordinate work amongst applications
- Code runs on EC2 (not serverless)
- 1year max. runtime
- Concept of “activity step” and “decision step”
- Has built-in human intervention step
- Example: order fulfilment from web to warehouse to delivery
- Step functions is recommended to be used for new applications, except:
- If you need external signals to intervene in the processes.
- If you need child processes that returns values to parent processes.

AWS EMR

- Stands for Elastic MapReduce
- Helps creating Hadoop clusters (Big Data) to analyze and process vast amount of data.
- These clusters can be of hundred of EC2 instances.
- Supports Apache Spark, Hbase, Flink, Presto..
- Takes care of all the provisioning and configuration
- Auto-Scaling and integrated with Spot Instances
- Use Cases: Machine Learning, Web Indexing, Big Data

How it works?



CloudFormation



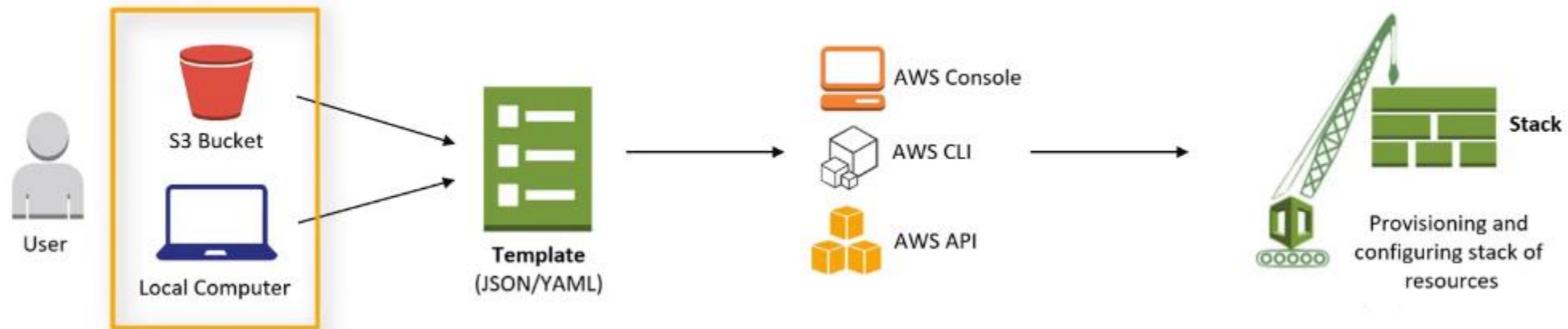
What is CloudFormation?

- Provisions, configures and manages the *stack* of AWS resources (infrastructure) based on the user-given *template*.
- Infrastructure-As-A-Code service.

Why to use it?

- Simplifies infrastructure management.
- Provides ability to view the design of the resources before provisioning.
- Quickly replicates the infrastructure.
- Controls and tracks infrastructure changes (versioning).

How does it work?

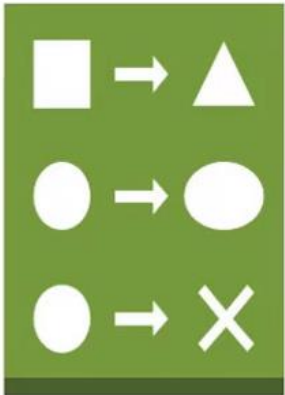


Stack



- Collection of AWS resources that are created according to the template.
- CloudFormation manages resources by creating, updating, or deleting the stacks.
- Can be as simple as one resource to an entire web application.
- Must all be created or deleted successfully for the stack to be created or deleted.
- If a resource cannot be created, CloudFormation rolls the stack back and automatically deletes any resources that were created.
- If a resource cannot be deleted, any remaining resources are retained until the stack can be successfully deleted.

Stack Updates



- Upon update, CloudFormation compares the changes submitted with the current state of the stack and **up only the changed resources**.
- Anyone with stack update permissions can update all of the resources in the stack.
- The progress of a stack update can be monitored by viewing the stack's *events*.
- Types of updates:
 - Direct Update:
 - Changes deployed immediately.
 - Used for quick deployments.
 - Via Change Sets:
 - Changes can be previewed before deciding whether to apply the changes or not.
 - Used if ensuring only the intentional changes are deployed is important.
- Updates can be cancelled if the stack is still in UPDATE_IN_PROGRESS state.
- Cancelling a stack rolls it back to the stack configuration that existed prior to initiating the stack update.
- Unintentional updates or deletion of stack can be prevented during a stack update by using a *stack policy*.



- Contains all the configuration details of the AWS resources to be provisioned via the stack.
- JSON or YAML.
- Template Designer can be used to visualize how the resources will be created and modified.
- Typical template has following sections (sections in **bold** are important part of template)
 - Format Version – has version date.
 - Description – describes the template via a simple string.
 - Metadata – contains additional data about the template.
 - **Resources (Mandatory) – set of resources to be created.**
 - **Parameters – set of custom values.**
 - **Mappings – contains key:value pairs where keys map to values.**
 - **Conditions – define circumstances in which the resources will be created.**
 - **Transform – specifies macros to process the template.**
 - **Outputs – are the values that can be returned after the stack creation or imported into other stacks as input.**

Deletion Policy

- Used to specify whether the resource should be deleted, preserved, or backed-up when the stack is deleted.
- Is specified as an attribute in the template.
- If a resource has no DeletionPolicy attribute, CloudFormation deletes the resource by default.
- Options:
 - Delete
 - Retain
 - Snapshot

```
"Resources" : {  
    "myS3Bucket" : {  
        "Type" : "AWS::S3::Bucket",  
        "DeletionPolicy" : "Retain"  
    }  
}
```


It helps you provision your resources by following best practices. Trusted Advisor inspects your AWS environment and finds opportunities to save money, improve system performance and reliability, or help close security gaps.



Cost Optimization

Under utilized EC2 instances

Idle Elastic Load Balancers

Unassociated Elastic IPS

...
...
...



Performance

Highly utilized EC2 instances

Rules in Ec2 security group

Over utilized EBS Volumes

...
...
...



Security

Security Groups
unrestricted access

IAM Password
Policy

...
...
...



Fault Tolerance

EC2 instance
distribution across
AZs in a region

AWS RDS Multi AZ

...
...
...



Service Limits

Service limits on
AWS VPC, EBS,
IAM, S3 etc



3 ✓ 1 ⚠ 2 ❗