AWS CloudFormation

- A service that gives developers and businesses an easy way to create a collection of related AWS resources and provision them in an **orderly** and predictable fashion.
- **Features**:
    - allows you to model your entire infrastructure in a text file called a **template**. (JSON or YAML)
    - to design visually, you can use AWS CloudFormation Designer
    - template = desired resources + configurations + dependencies, launched as a **stack**.
    - Can use template to create, update and delete an entire stack as a single unit.
    - CloudFormation automates the provisioning and updating of your infrastructure in a safe and controlled manner.
    - can use **Rollback Triggers** to specify the CloudWatch alarm that CloudFormation should monitor during the stack creation and update process.
    - If any of the alarms are breached, CloudFormation rolls back the entire stack operation to a previously deployed state.
    - **CloudFormation Change Sets** allow you to preview how proposed changes to a stack might impact your running resources.
    - **CloudFormation StackSets** let you provision a common set of AWS resources across multiple accounts and regions, with a single CloudFormation template.
    - **CloudFormation registry** helps you discover and provision private and public extensions such as resources, modules, and hooks in your AWS CloudFormation templates.
- **CloudFormation Vs Elastic Beanstalk**

Elastic Beanstalk	**CloudFormation**
Platform as a Service (PaaS)	Infrastructure as Code (IaC)
suitable for developers who want to focus on writing code and let AWS handle the infrastructure management	suitable for infrastructure and operations teams who want to manage and automate their infrastructure as code.
simplifies application deployment and management	provides more control over resource provisioning and configuration
built in support for app deployment, scaling, monitoring	you need to define these features in a template
supports a limited set of languages and platforms	more flexible and versatile.
provide security features, such as IAM roles and policies	IAM roles and policies + additional integrations with AWS Config and KMS for monitoring and encryption.
supports auto-scaling based on CloudWatch metrics	need to configure auto-scaling groups and policies in your templates.

    - use both Elastic Beanstalk and CloudFormation together.
    - Elastic Beanstalk uses CloudFormation templates under the hood to create and manage resources.
- **CloudFormation Four concepts : template, stack, ChangeSet, Stackset**
- **Template:**
    - a JSON or YAML declarative code file
    - include several major sections. The **Resources** section is the only required section.
        - Resources (Required)
        - Format Version (optional)
        - Description (optional)
        - Metadata (optional)
        - Parameters (optional)
        - Rules (optional)
        - Mappings (optional)
        - Conditions (optional)
        - Transform (optional)

        - Outputs (optional)
    - 5 types of elements:
        1. Template parameters (input values supplied at stack creation time) (optional)
        2. Output values (e.g., the complete URL to a web application) (optional)
        3. data tables used to look up static configuration values (e.g., AMI names) (optional)
        4. The list of AWS resources and their configuration values
        5. A template file format version number
- **Limitations**
    - there is a limit on the number of stacks you can create per region, and the size of a template cannot exceed 460,800 bytes.
    - a maximum of 200 parameters in an AWS CloudFormation template.
- **Things to know:**
    - CloudFormation resources name = logical name (given in template) + stack name
    - By default, the "automatic rollback on error" feature is enabled.
    - If one of the resources in the stack cannot be completed, CloudFormation reverts complete stack.
    - a *WaitCondition* resource can be used to wait stack creation until signal is received from external app.
    - CloudFormation allows you to define deletion policies for resources in the template
      e.g. You can specify that snapshots be created for Amazon EBS volumes or Amazon RDS database instances before they are deleted.
      can also specify that a resource should be preserved and not deleted when the stack is deleted.
    - CloudFormation supports creating VPCs, subnets, gateways, route tables and network ACLs as well as creating resources such as elastic IPs, Amazon EC2 Instances, EC2 security groups, auto scaling groups, elastic load balancers, Amazon RDS database instances and Amazon RDS security groups in a VPC.
- **Development :**
    - Working with templates: Defining parameter in a template

JSON
```
 "Parameters" : {
  "InstanceTypeParameter" : {
    "Type" : "String",
    "Default" : "t2.micro",
    "AllowedValues" : ["t2.micro", "m1.small",
"m1.large"],
    "Description" : "Enter t2.micro, m1.small, or
m1.large. Default is t2.micro."
  }
}
```

YAML
```
Parameters:
  InstanceTypeParameter:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
      - m1.large
    Description: Enter t2.micro, m1.small, or m1.large.
Default is t2.micro.
```

    - Referencing a parameter within a template -use the **Ref** intrinsic function to reference a parameter
    - Parameters must be declared and referenced from within the same template.

JSON
```
"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "InstanceType" : { "Ref" : "InstanceTypeParameter"
},
    "ImageId" : "ami-0ff8a91507f77f867"
  }
}
```

YAML
```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType:
      Ref: InstanceTypeParameter
    ImageId: ami-0ff8a91507f77f867
```

- **Integrations with other AWS services**
    - For *Dynamic references*, currently supports the following dynamic reference patterns:

    - ssm, for plaintext values stored in AWS Systems Manager Parameter Store.
    - ssm-secure, for secure strings stored in AWS Systems Manager Parameter Store.
    - secretsmanager, for entire secrets or secret values stored in AWS Secrets Manager.
-