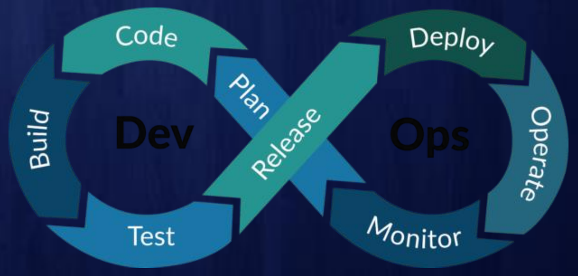
AWS SAA + SysOps + Developer + DevOps Course #Day-13

We will start at 8 AM,
Stay tuned



RAKESH TANINKI
LEARN TO UNLEARN

Recap:

- AWS Cloud Watch
  - Metrics
  - Alarms
  - Logs
  - Events (basics)
  - Demo
- AWS Cloud Trail
  - Event history
  - Organizational trail
  - Demo

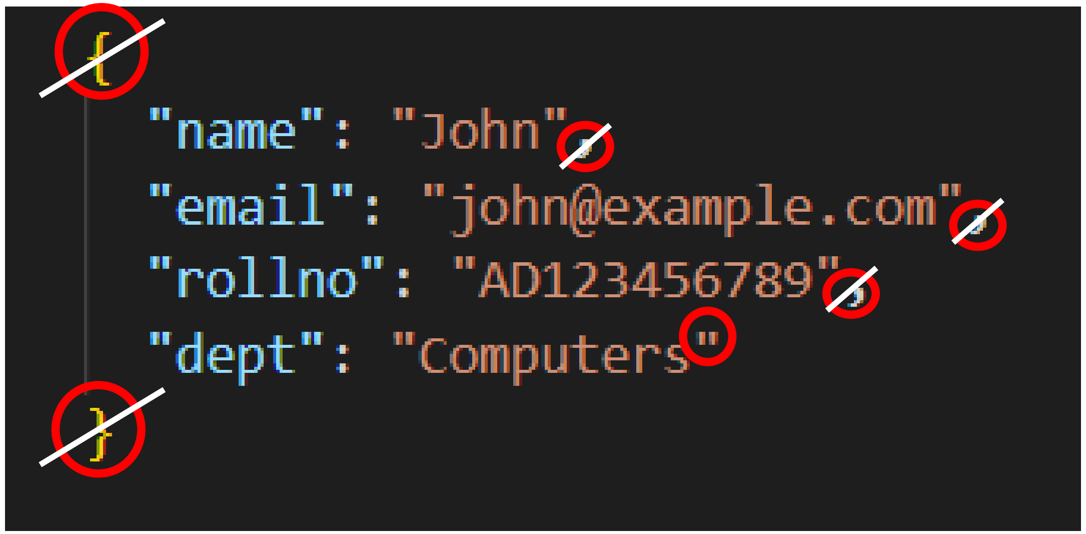
Today’s topics:

- YAML – Introduction
- Cloud Formation
  - Basics
  - Templates and Stacks
  - Template sections
  - Resources
  - Demo
- S3 Static website hosting
  - Demo

YAML – Ain’t Markup Language

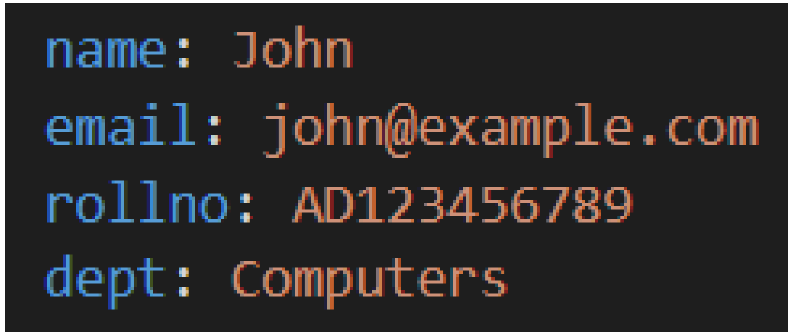
YAML – Yet Another Markup Language

- **YAML** is a **human-readable** and **data-serialization** language
- Commonly used for **configuration files –** Ansible, Cloud formation, Terraform
- AWS Services – Cloud Formation, Code Build file, SAM, etc.
- Easier than **JSON** and **XML** with minimal syntax
- .yml or .yaml extensions



```
{
    "name": "John",
    "email": "john@example.com",
    "rollno": "AD123456789",
    "dept": "Computers"
}
```

JSON Object

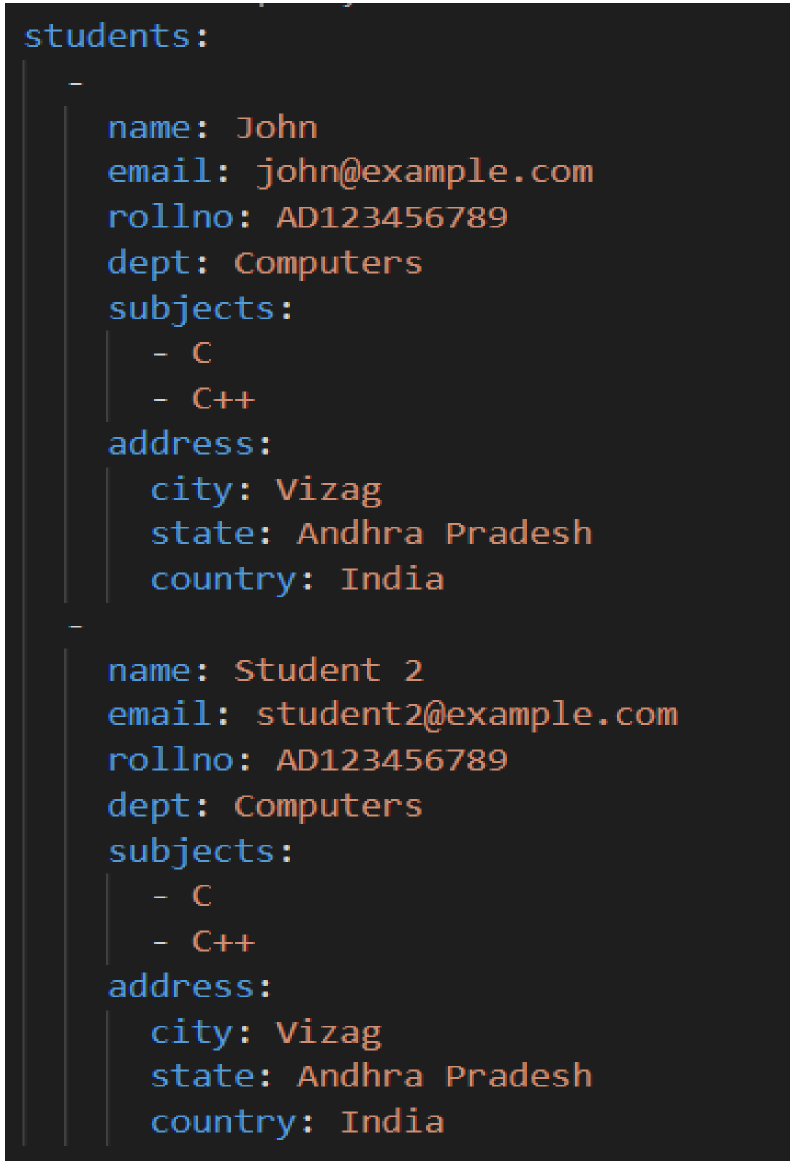


```
name: John
email: john@example.com
rollno: AD123456789
dept: Computers
```

YAML Object

YAML – Syntax

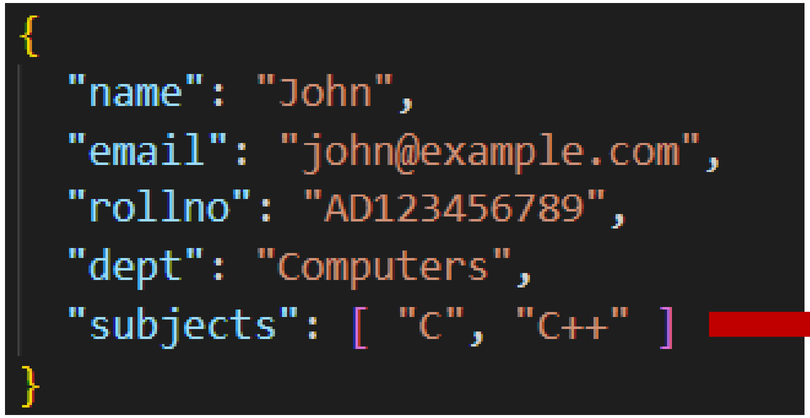
- **Key Value** pairs
- **Indentation –** similar to Python
- **Data Types:**
  - Number
  - String
  - Float
  - null
  - Boolean
  - List
  - Nested JSON type



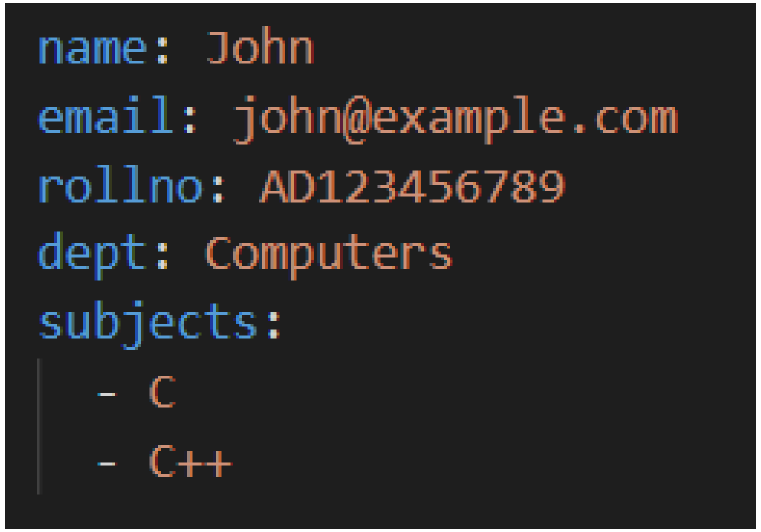
```yaml
students:
  -
    name: John
    email: john@example.com
    rollno: AD123456789
    dept: Computers
    subjects:
      - C
      - C++
    address:
      city: Vizag
      state: Andhra Pradesh
      country: India
  -
    name: Student 2
    email: student2@example.com
    rollno: AD123456789
    dept: Computers
    subjects:
      - C
      - C++
    address:
      city: Vizag
      state: Andhra Pradesh
      country: India
```

YAML – List / Array

- An **array** is a group of similar values with a single name.
- In **YAML**, Array represents a single key mapped to multiple values. Each value starts with a hyphen - symbol followed by space.

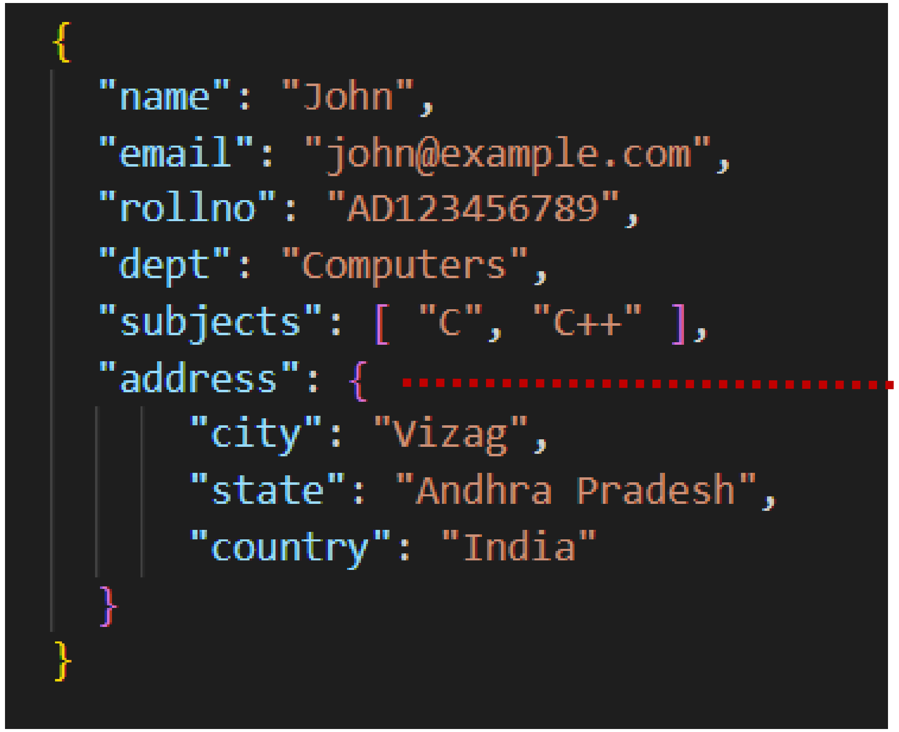


```json
{
  "name": "John",
  "email": "john@example.com",
  "rollno": "AD123456789",
  "dept": "Computers",
  "subjects": [ "C", "C++" ]
}
```

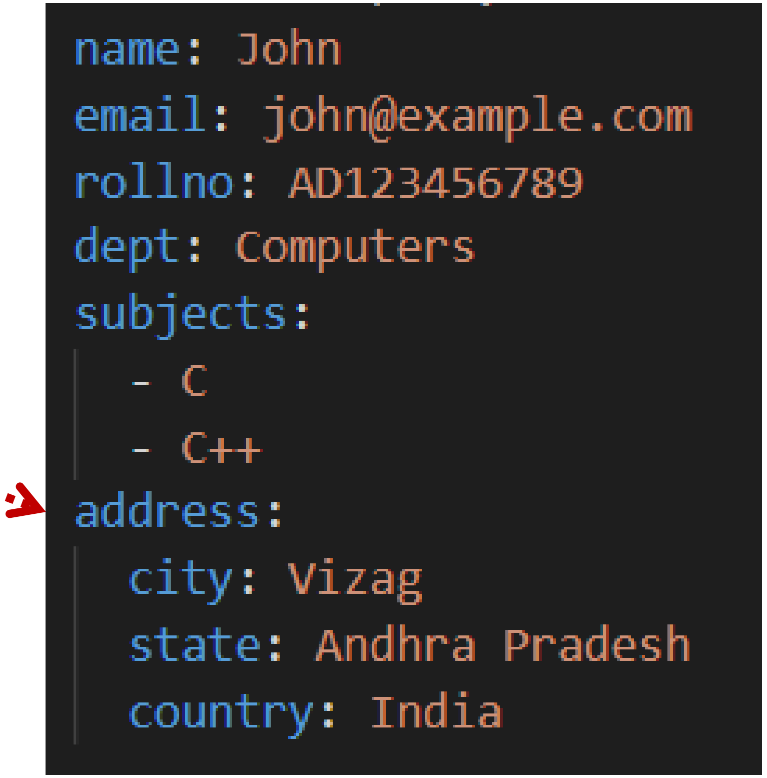


```yaml
name: John
email: john@example.com
rollno: AD123456789
dept: Computers
subjects:
  - C
  - C++
```

YAML – Nested Object

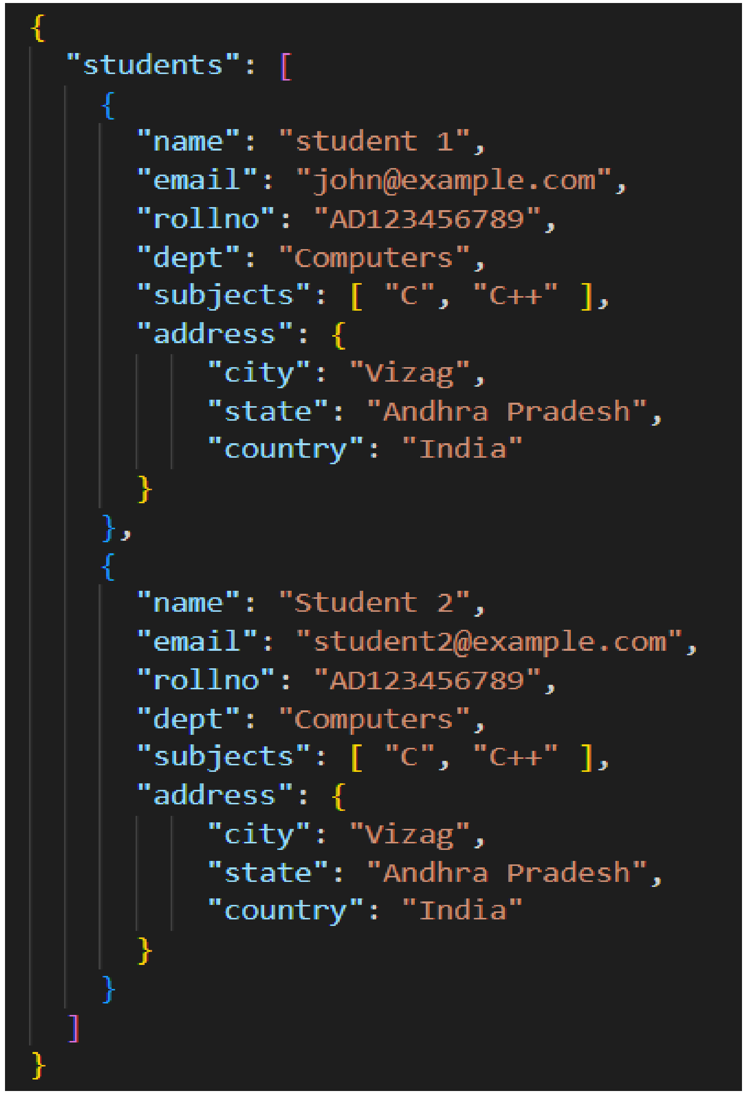


```json
{
  "name": "John",
  "email": "john@example.com",
  "rollno": "AD123456789",
  "dept": "Computers",
  "subjects": [ "C", "C++" ],
  "address": {
      "city": "Vizag",
      "state": "Andhra Pradesh",
      "country": "India"
  }
}
```

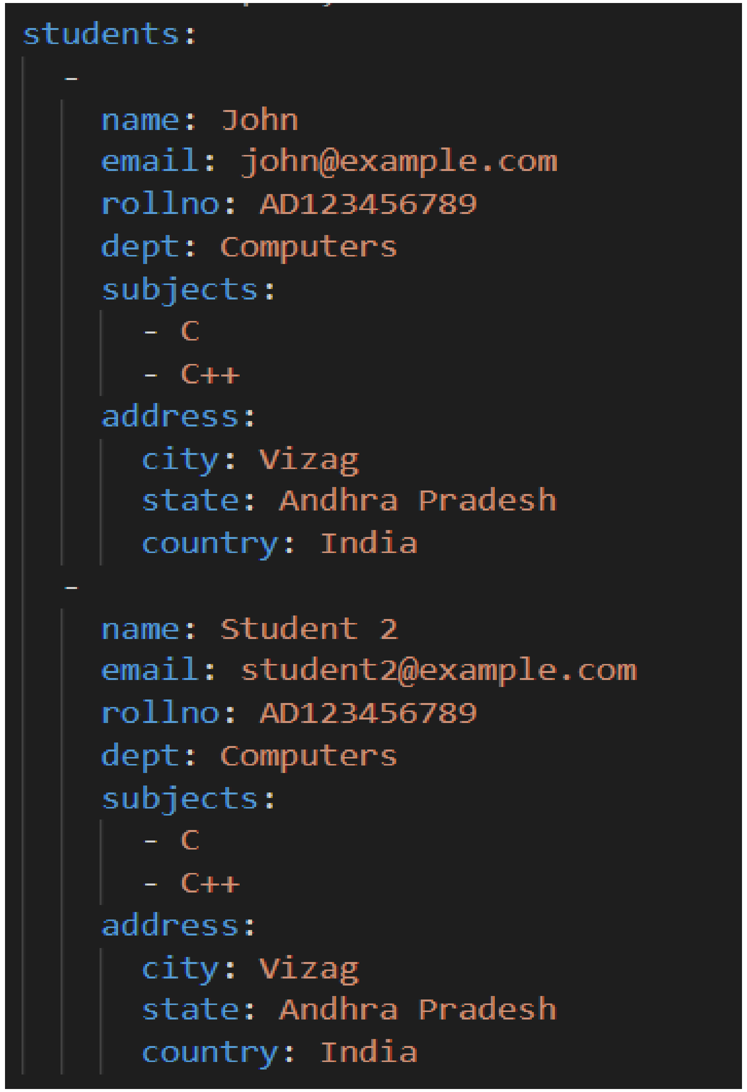


```yaml
name: John
email: john@example.com
rollno: AD123456789
dept: Computers
subjects:
  - C
  - C++
address:
  city: Vizag
  state: Andhra Pradesh
  country: India
```

YAML – Array of objects



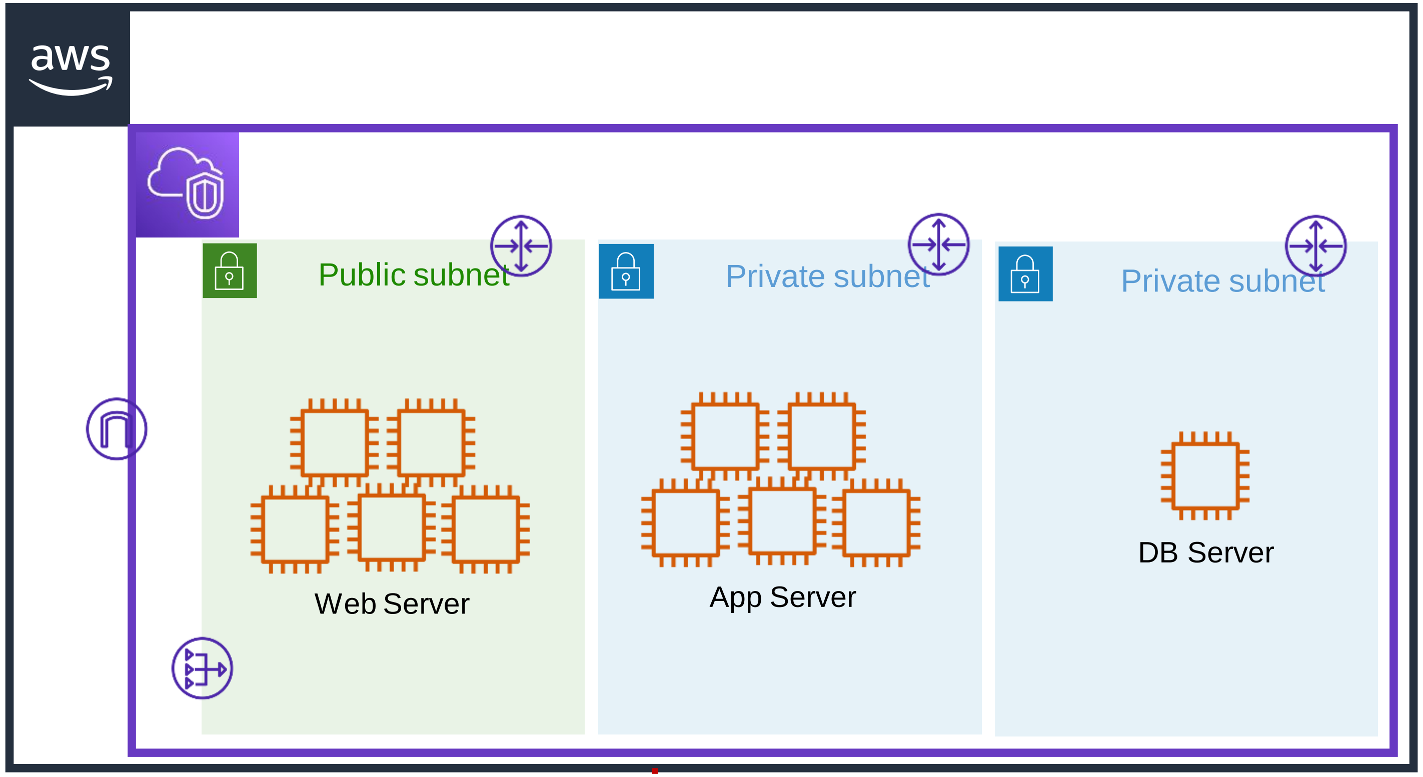
```json
{
  "students": [
    {
      "name": "student 1",
      "email": "john@example.com",
      "rollno": "AD123456789",
      "dept": "Computers",
      "subjects": [ "C", "C++" ],
      "address": {
          "city": "Vizag",
          "state": "Andhra Pradesh",
          "country": "India"
      }
    },
    {
      "name": "Student 2",
      "email": "student2@example.com",
      "rollno": "AD123456789",
      "dept": "Computers",
      "subjects": [ "C", "C++" ],
      "address": {
          "city": "Vizag",
          "state": "Andhra Pradesh",
          "country": "India"
      }
    }
  ]
}
```



```yaml
students:
  -
    name: John
    email: john@example.com
    rollno: AD123456789
    dept: Computers
    subjects:
      - C
      - C++
    address:
      city: Vizag
      state: Andhra Pradesh
      country: India
  -
    name: Student 2
    email: student2@example.com
    rollno: AD123456789
    dept: Computers
    subjects:
      - C
      - C++
    address:
      city: Vizag
      state: Andhra Pradesh
      country: India
```

AWS Cloud Formation

AWS Cloud Formation – Example architecture



Difficult and **time taking** process to setup every time manually

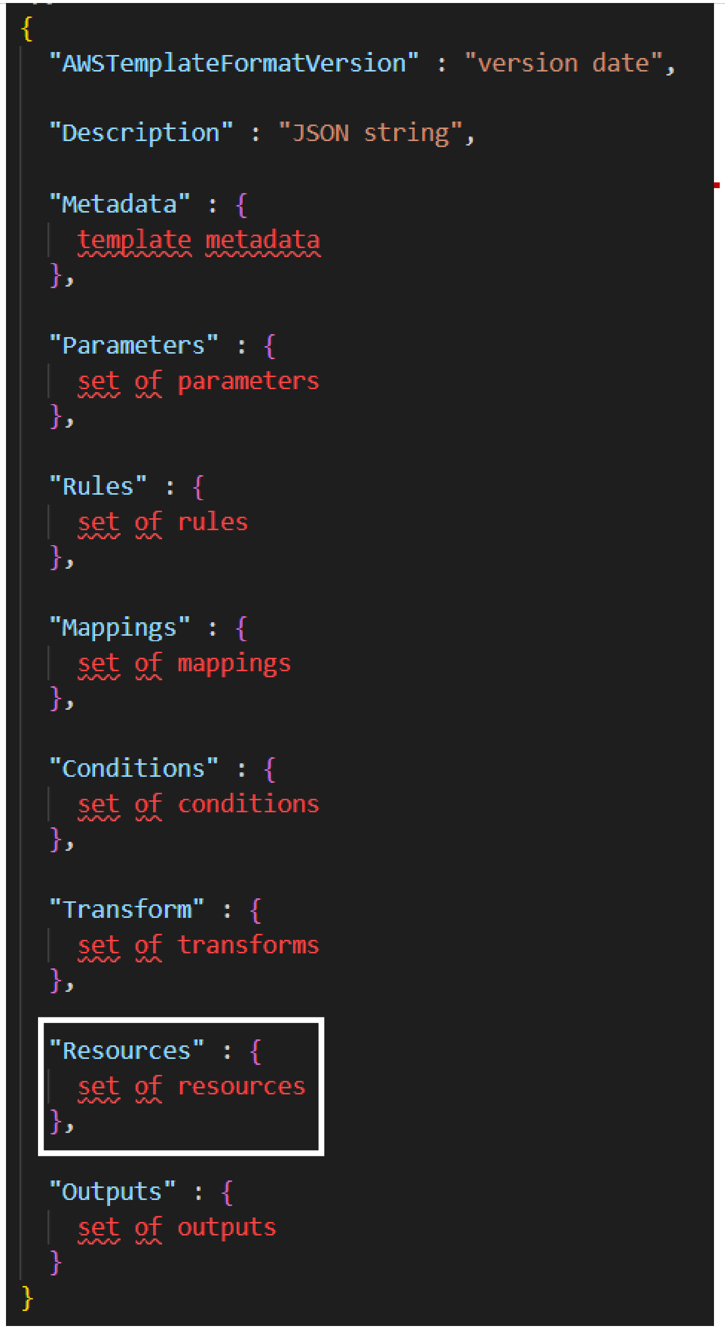
Components:

1. VPC
2. Internet Gateway
3. Attach internet gateway
4. NAT gateway creation and attachment
5. Subnets
6. Routers
7. EC2s
8. Security Groups
9. etc

AWS Cloud Formation – Basics

- **Cloud Formation – CFN**
- **CFN –** is a template to provision AWS resources automatically
- Quickly **replicate** infrastructure between environments
- Quickly terminate / **tear down** configured infrastructure
- **Control** and **track changes** to your infrastructure
- It supports both **JSON** and **YAML** formats

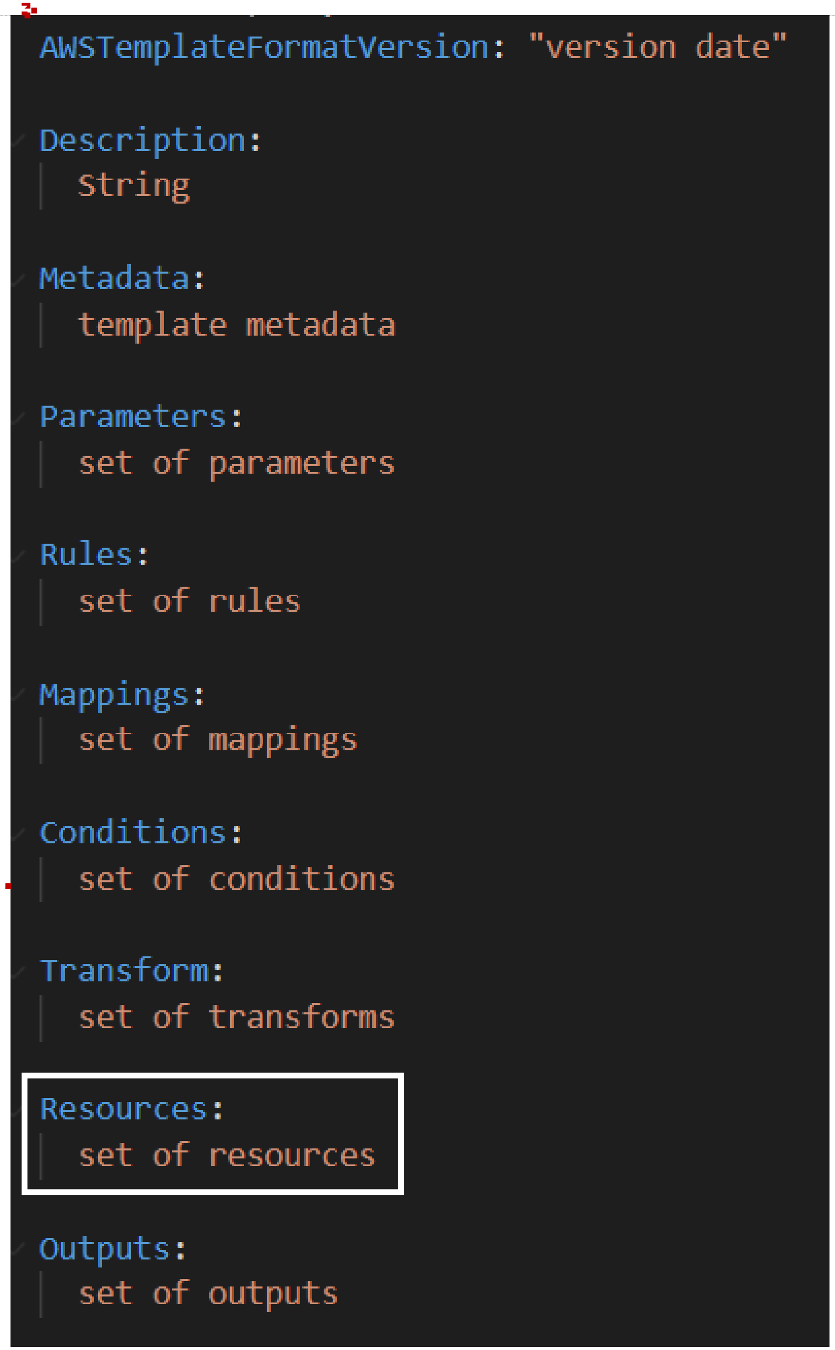
AWS Cloud Formation – Blue Print (Anatomy)



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

  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  },

  "Parameters" : {
    set of parameters
  },

  "Rules" : {
    set of rules
  },

  "Mappings" : {
    set of mappings
  },

  "Conditions" : {
    set of conditions
  },

  "Transform" : {
    set of transforms
  },

  "Resources" : {
    set of resources
  },

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

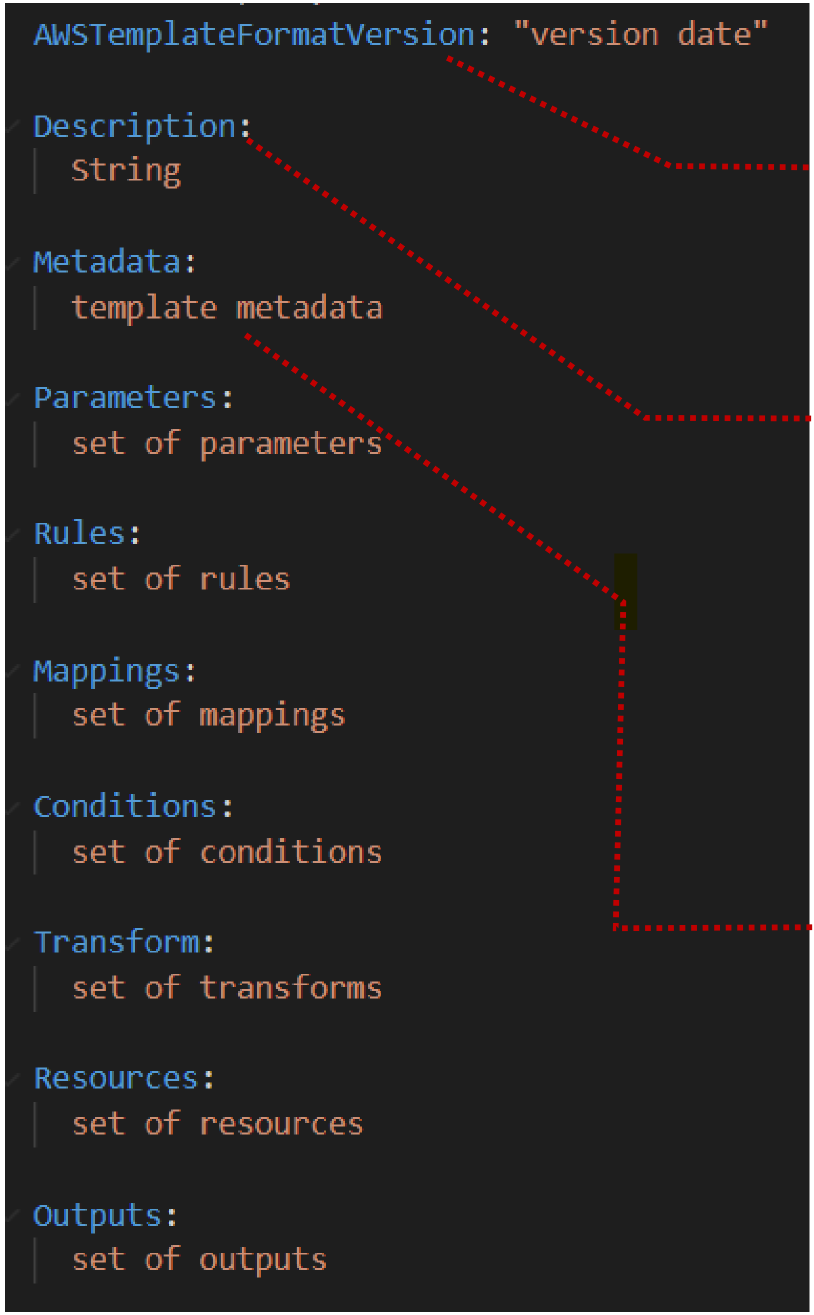
JSON

YAML



```
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

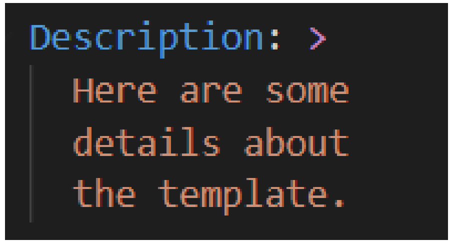
AWS Cloud Formation – Sections



```
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

Template version (optional)
Ex: 2010-09-09

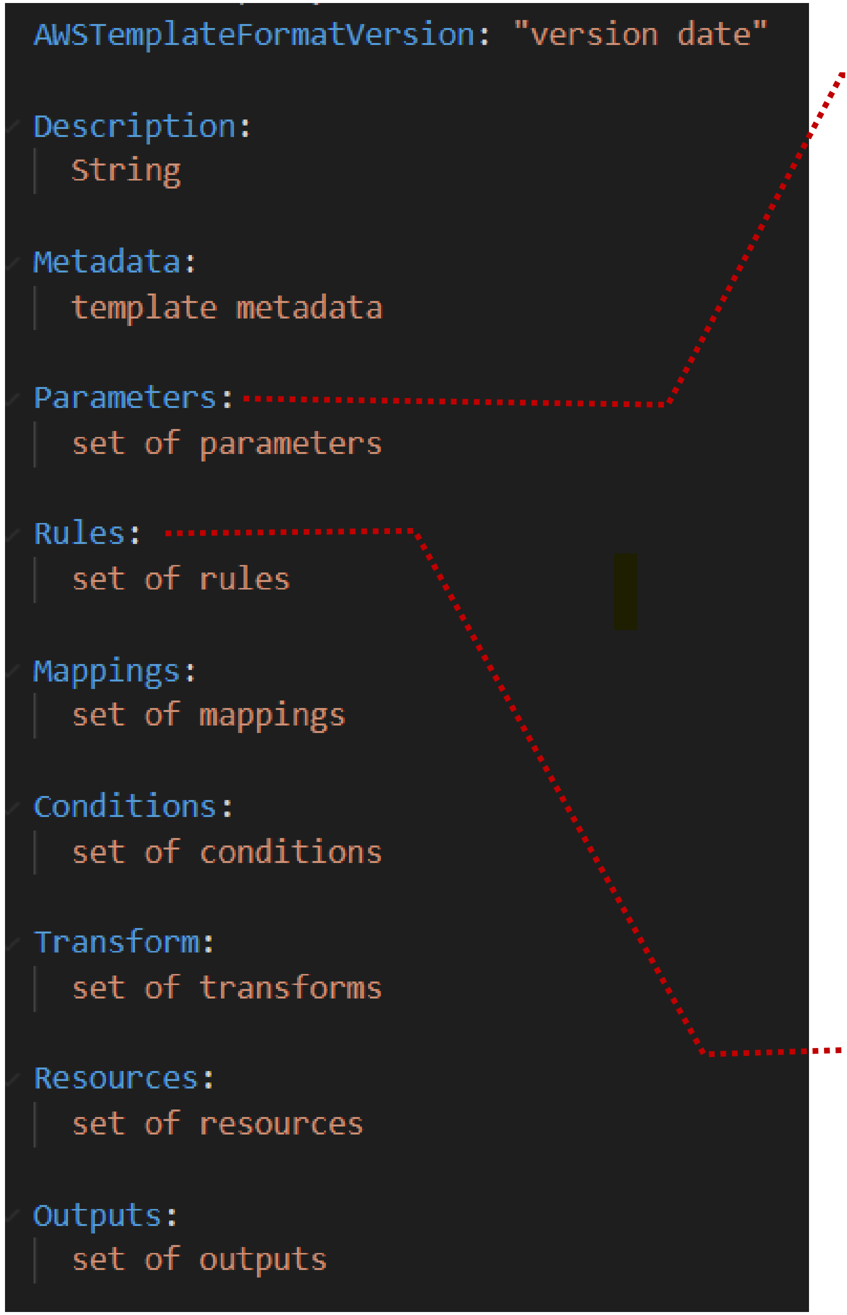
- **Optional section**
- **A text string that describes the template.**
- **must follow the template format version section.**



```
Description: >
  Here are some
  details about
  the template.
```

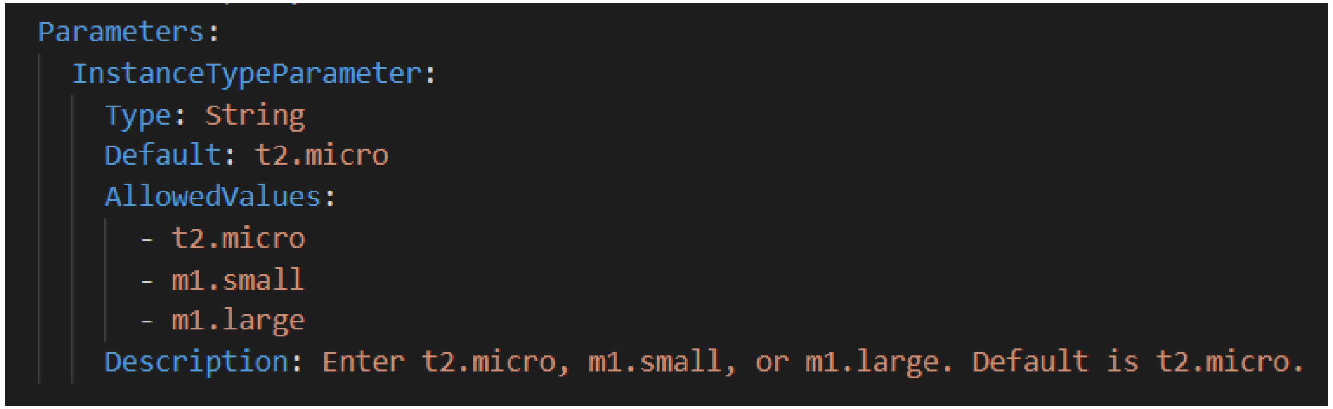
- **Optional section**
- **Objects that provide additional information about the template**
- **Formats the UI for parameters section**

AWS Cloud Formation – Sections



```
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

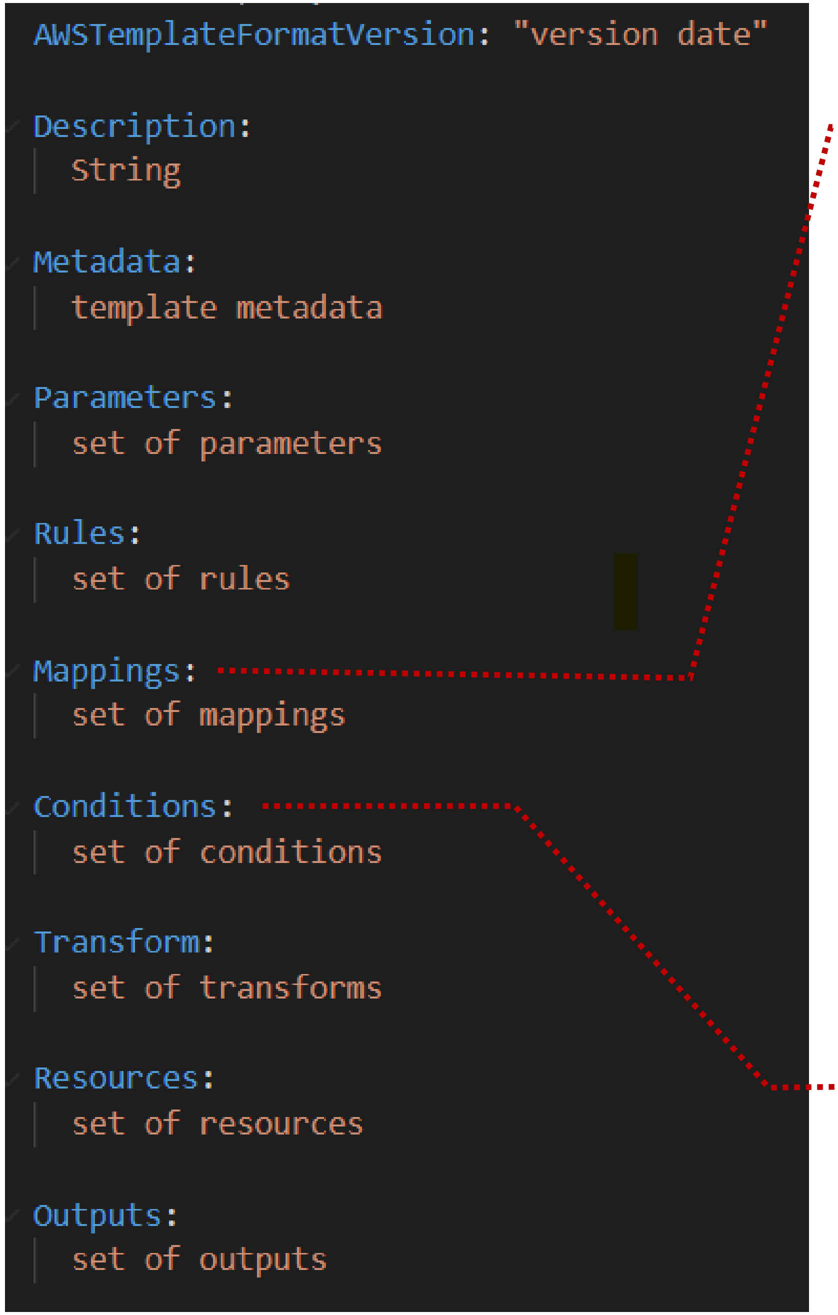
- Optional section
- Values to pass to your template at runtime (when you create or update a stack)



```
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.
```

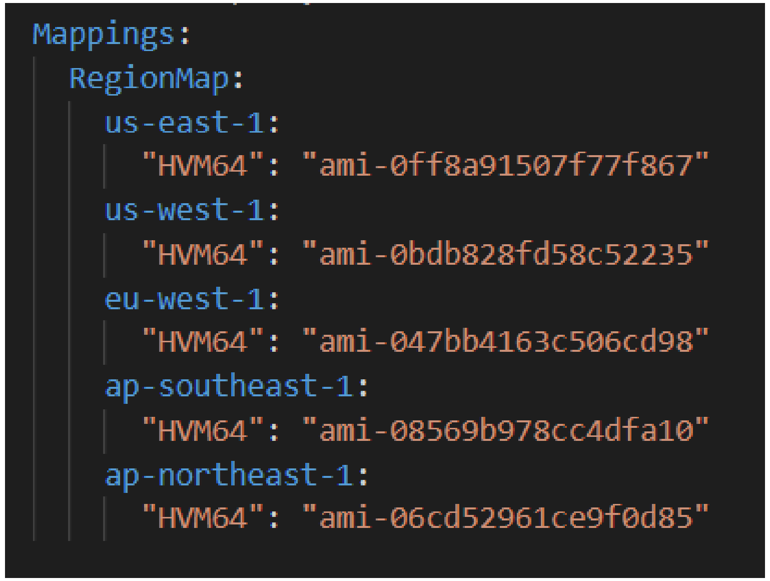
- Optional section
- validates a parameter or a combination of parameters passed to a template during a stack creation or stack update

AWS Cloud Formation – Sections



```
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

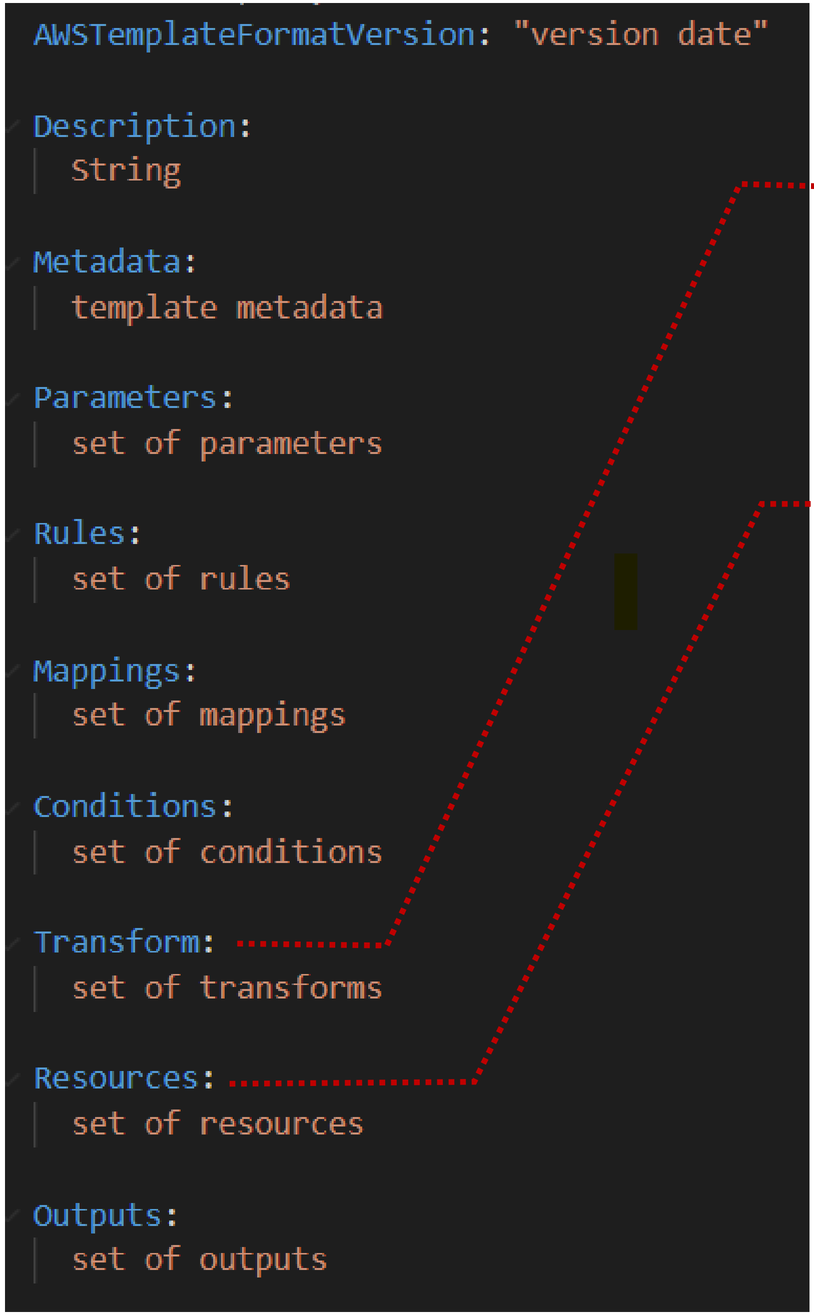
- **Optional section**
- **matches a key to a corresponding set of named values**



```
Mappings:
  RegionMap:
    us-east-1:
      "HVM64": "ami-0ff8a91507f77f867"
    us-west-1:
      "HVM64": "ami-0bdb828fd58c52235"
    eu-west-1:
      "HVM64": "ami-047bb4163c506cd98"
    ap-southeast-1:
      "HVM64": "ami-08569b978cc4dfa10"
    ap-northeast-1:
      "HVM64": "ami-06cd52961ce9f0d85"
```

- **Optional section**
- **contains statements that define the circumstances under which entities are created or configured**

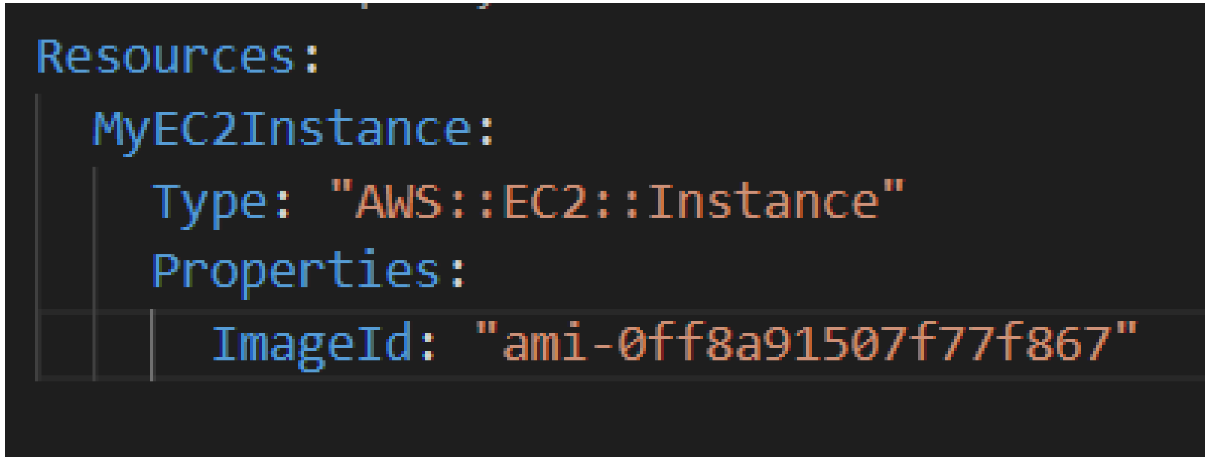
AWS Cloud Formation – Sections



```yaml
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

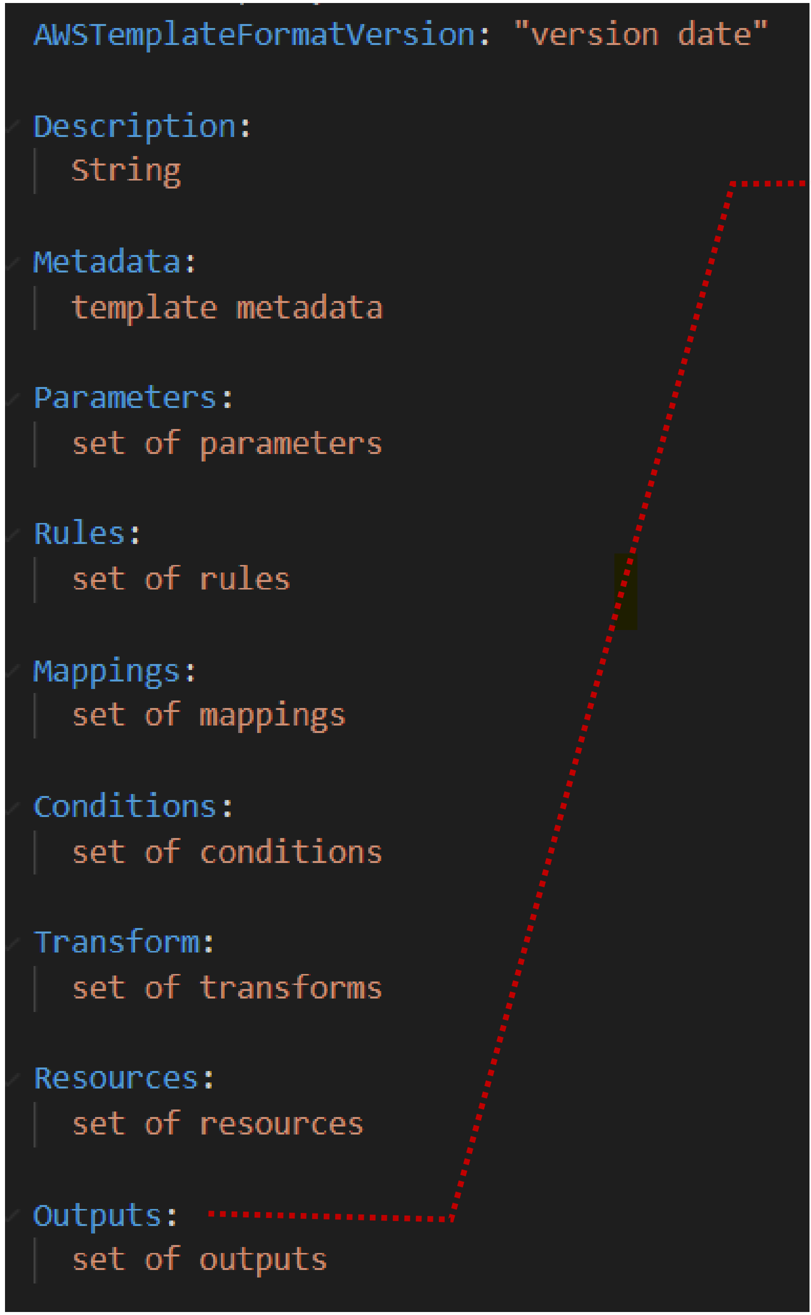
- Optional section
- specifies one or more macros that AWS CFN uses to process your template

- Mandatory section
- specifies one or more macros that AWS CFN uses to process your template



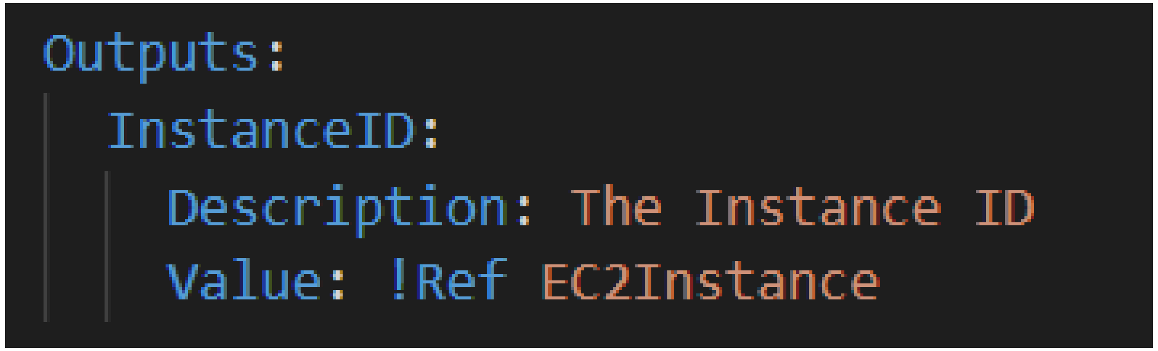
```yaml
Resources:
  MyEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: "ami-0ff8a91507f77f867"
```

AWS Cloud Formation – Sections



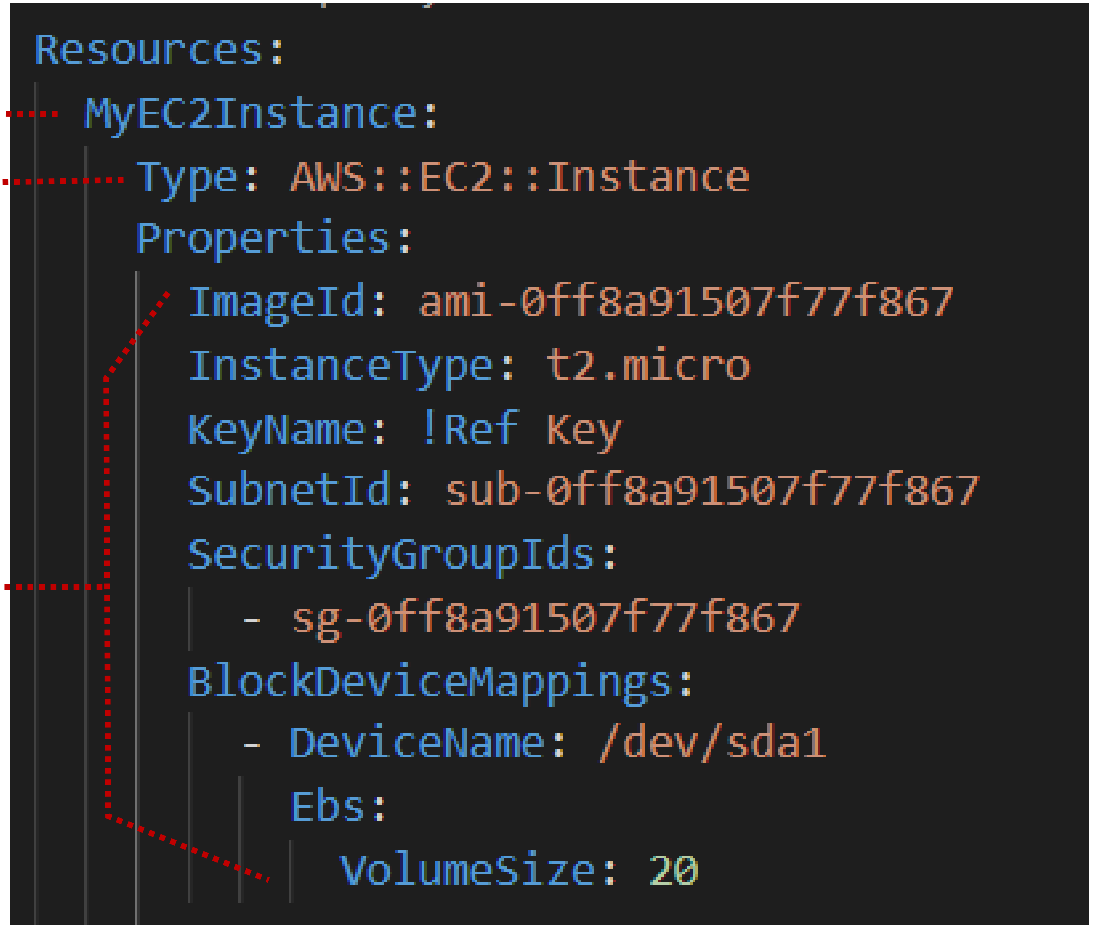
```
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Rules:
  set of rules

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

- **Optional section**
- **Displays the outputs created with the template**



```
Outputs:
  InstanceID:
    Description: The Instance ID
    Value: !Ref EC2Instance
```

AWS Cloud Formation – Resources (Ex: EC2)



```yaml
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: !Ref Key
      SubnetId: sub-0ff8a91507f77f867
      SecurityGroupIds:
        - sg-0ff8a91507f77f867
      BlockDeviceMappings:
        - DeviceName: /dev/sda1
          Ebs:
            VolumeSize: 20
```

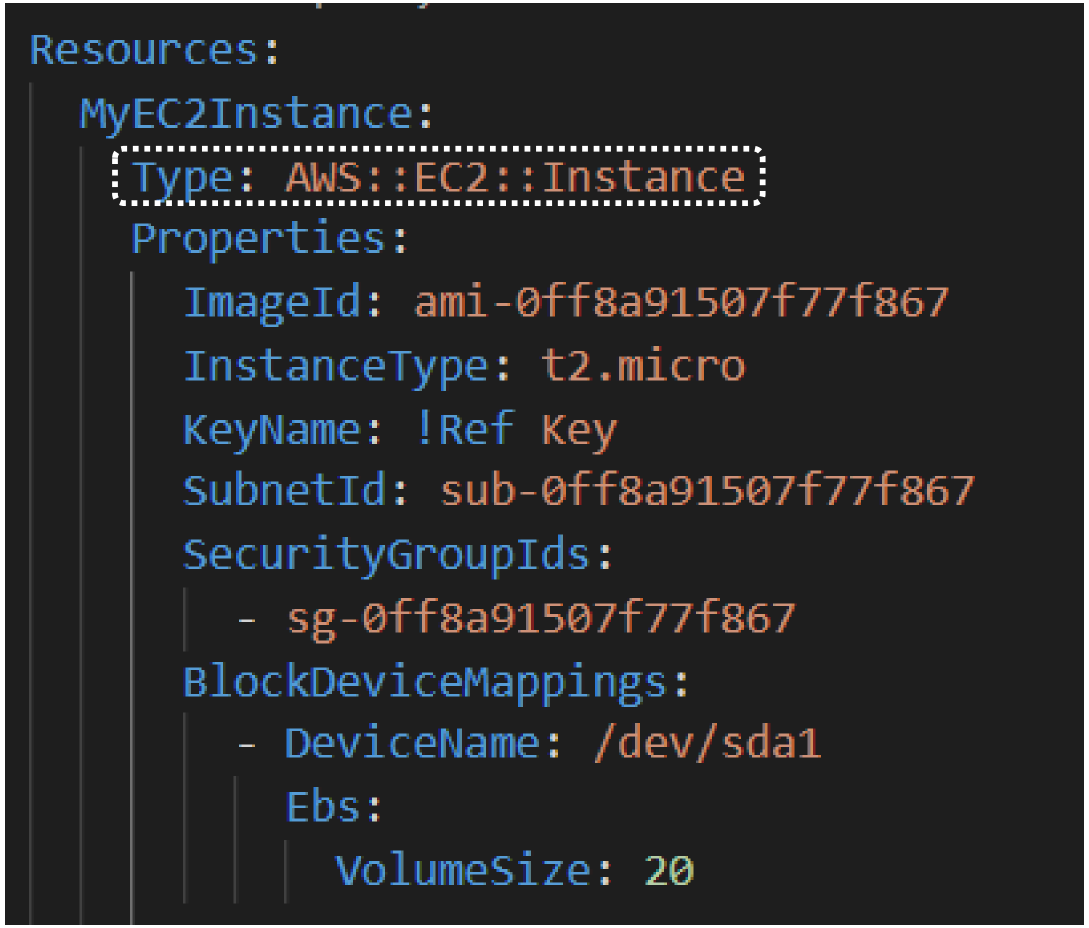
Logical ID

Resource type

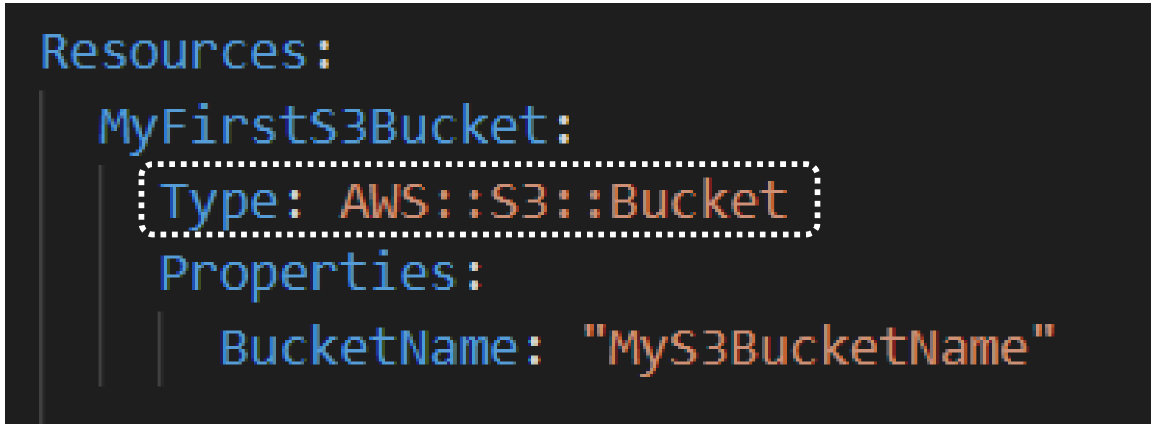
AWS::IAM::User
AWS::EC2::VPC
AWS::EC2::Subnet

Properties of a resource

AWS Cloud Formation – Resources (Ex: S3)



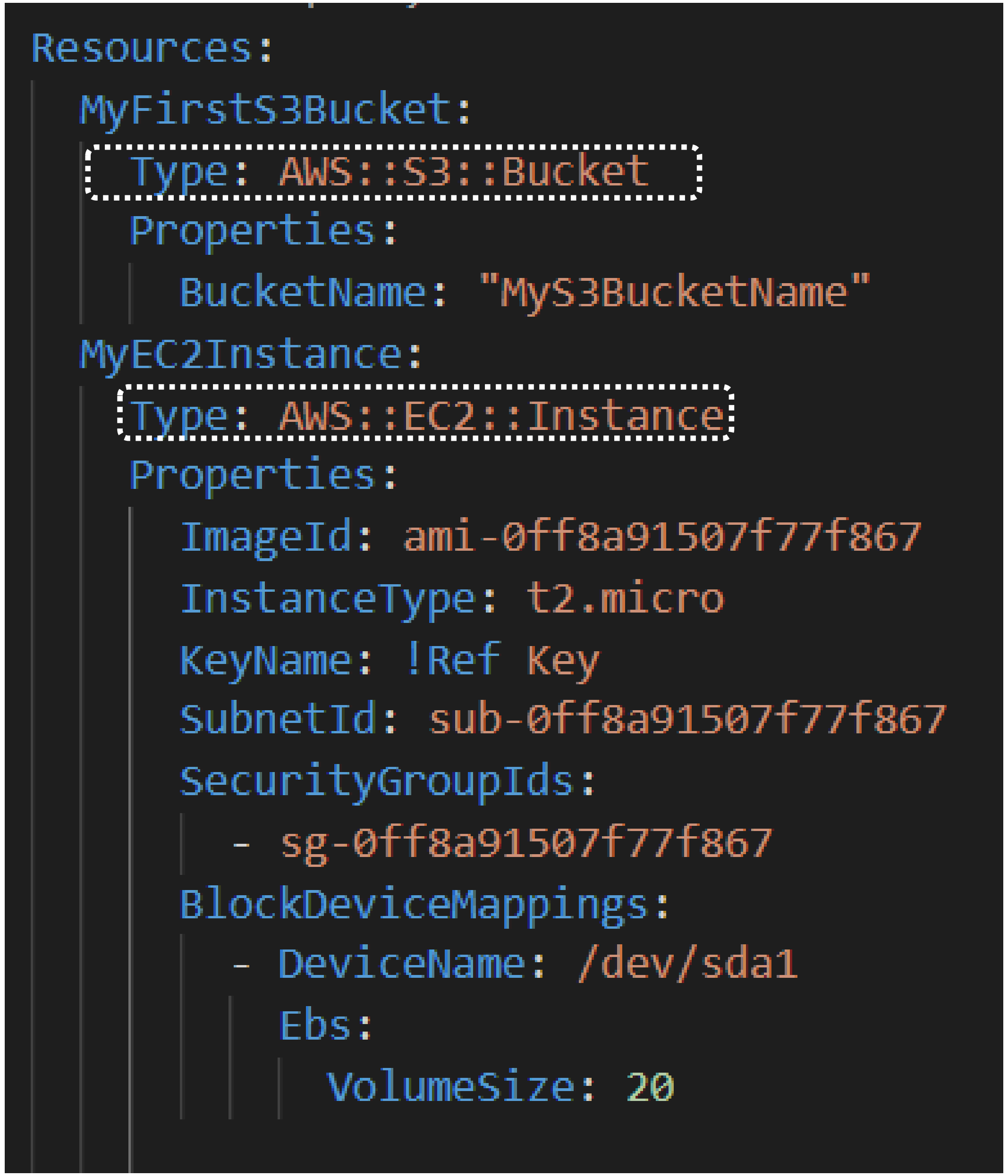
```yaml
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: !Ref Key
      SubnetId: sub-0ff8a91507f77f867
      SecurityGroupIds:
        - sg-0ff8a91507f77f867
      BlockDeviceMappings:
        - DeviceName: /dev/sda1
          Ebs:
            VolumeSize: 20
```



```yaml
Resources:
  MyFirstS3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: "MyS3BucketName"
```

AWS Cloud Formation – Resources (Ex: S3 and EC2)

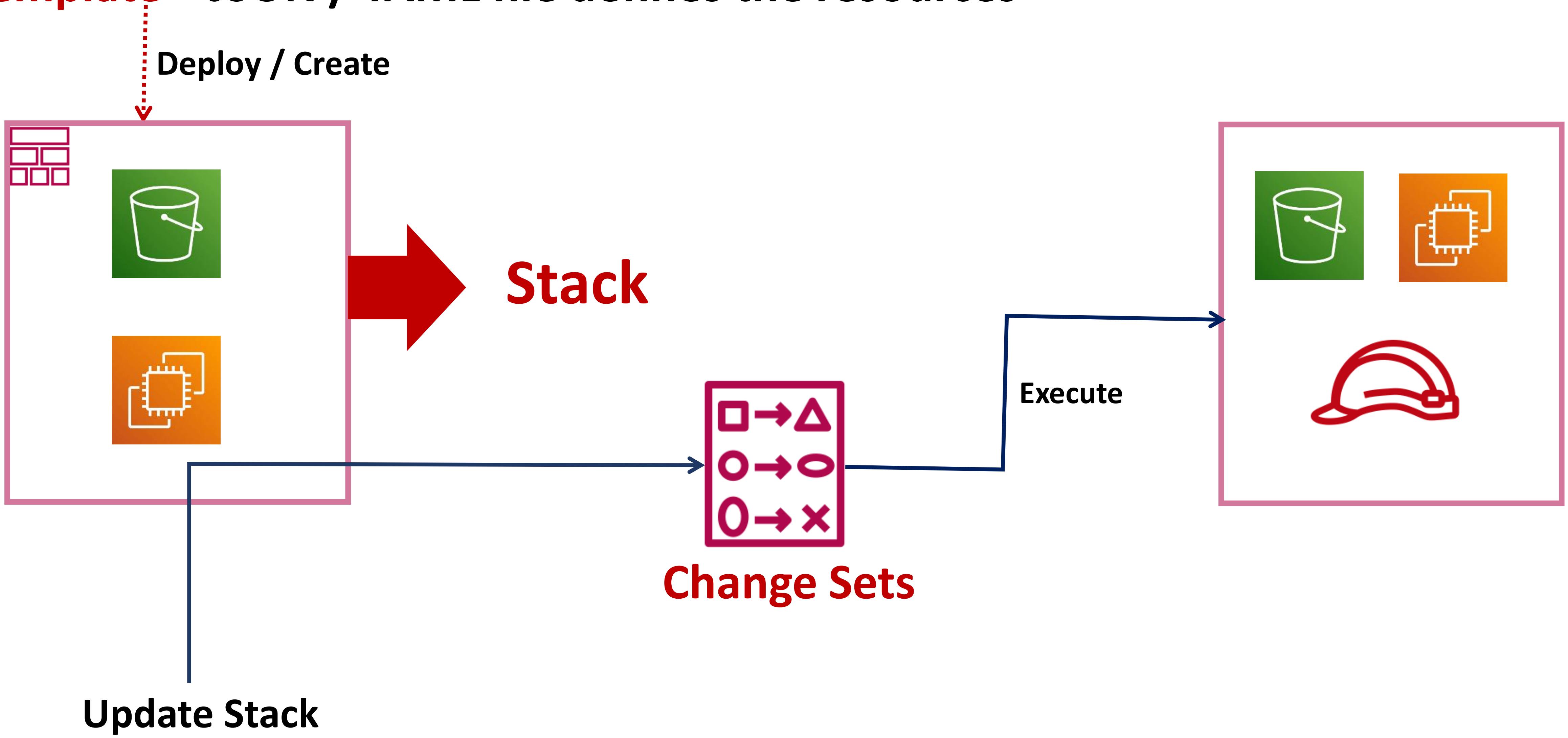
Indentation is important



```yaml
Resources:
  MyFirstS3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: "MyS3BucketName"
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: !Ref Key
      SubnetId: sub-0ff8a91507f77f867
      SecurityGroupIds:
        - sg-0ff8a91507f77f867
      BlockDeviceMappings:
        - DeviceName: /dev/sda1
          Ebs:
            VolumeSize: 20
```

AWS Cloud Formation – Terminologies

Template – JSON / YAML file defines the resources



Deploy / Create

Stack

Change Sets

Execute

Update Stack

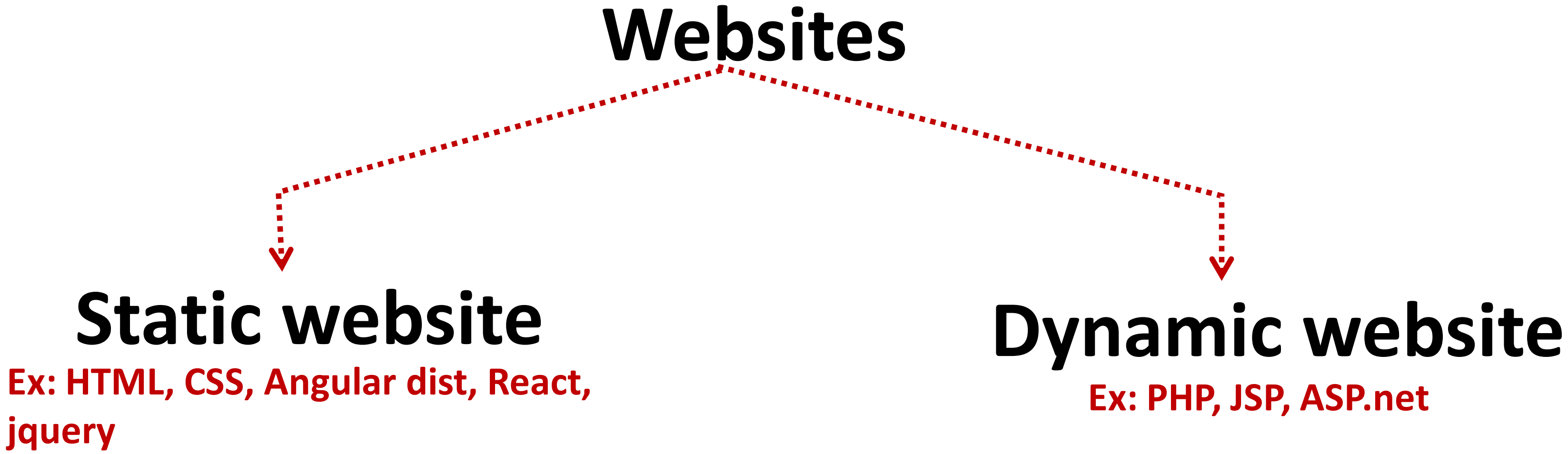
AWS Cloud Formation

Basic Demo

AWS S3 Static Website Hosting

AWS S3 – Static Website Hosting

- S3 can be used to host static websites



Websites

Static website

Ex: HTML, CSS, Angular dist, React, jquery

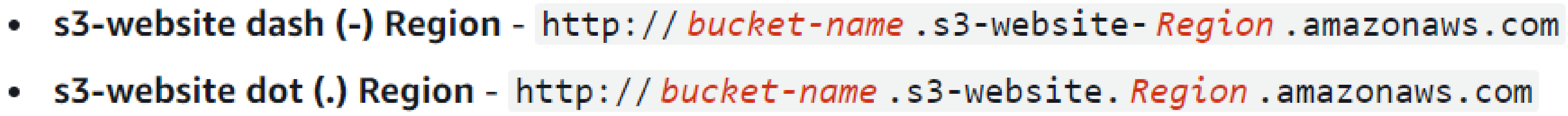
Dynamic website

Ex: PHP, JSP, ASP.net

- Dynamic websites are also supported if micro services architecture is followed

AWS S3 – Static Website Hosting

- **Website endpoints (**depends on the region**)**



  - s3-website dash (-) Region - `http://bucket-name.s3-website-Region.amazonaws.com`
  - s3-website dot (.) Region - `http://bucket-name.s3-website.Region.amazonaws.com`

- Only **HTTP** is supported by default
- **Custom domain** can be mapped with **AWS Route 53 or** 3rd party domain providers

AWS S3 Static Website Hosting Demo

Thank you, will meet in tomorrow’s session

ANY
QUESTIONS?