

Cloud Computing Architecture

The CloudFormation Template



The CloudFormation Template

This presentation:

- The Sections of CloudFormation Template



Images licensed under creative commons.

Understanding the Template

- May not need to be able to speak/write from scratch
- Need to understand it
- Garbage in/Garbage Out
- Well documented

Anatomy of an AWS CloudFormation Template



```
"Description" : "JSON  
string",  
"Metadata" : {  
    template metadata },  
"Parameters" : {  
    set of parameters },  
"Mappings" : {  
    set of mappings },  
"Conditions" : {  
    set of conditions },  
"Resources" : {  
    set of resources },  
"Outputs" : {  
    set of outputs }
```

JSON example

Description:

Text string that describes the template.

- Literal string between 0 and 1024 bytes long
- Cannot use a parameter or function to specify it

```
"Description" : "This template builds  
a VPC with one public and one private  
subnet",
```

Anatomy of an AWS CloudFormation Template



```
"Description" : "JSON  
string",  
"Metadata" : {  
    template metadata },  
"Parameters" : {  
    set of parameters },  
"Mappings" : {  
    set of mappings },  
"Conditions" : {  
    set of conditions },  
"Resources" : {  
    set of resources },  
"Outputs" : {  
    set of outputs }
```

JSON example

Metadata:

JSON objects that provide additional details about the template.

- Includes settings or configuration information that some AWS CloudFormation features need to retrieve
- Can be specified at the template or resource level

```
"Metadata" : {  
    "Instances" : {"Description" : "<Information  
        about the instances>"},  
    "Databases" : {"Description" : "<Information  
        about the databases>"}}
```

Anatomy of an AWS CloudFormation Template



```
"Description" : "JSON  
string",  
"Metadata" : {  
    template metadata },  
"Parameters" : {  
    set of parameters },  
"Mappings" : {  
    set of mappings },  
"Conditions" : {  
    set of conditions },  
"Resources" : {  
    set of resources },  
"Outputs" : {  
    set of outputs }
```

JSON example

Resources:

Resources (and their properties) that will be included in the stack.

Properties:

- Each resource must be declared separately (except multiple instances of the same resource)
- Resource declaration has the resource's attributes

```
"Resources" : {  
    "Logical ID" : {  
        "Type" : "Resource type",  
        "Properties" : {  
            Set of properties } } }
```

Anatomy of an AWS CloudFormation Template: Resources



```
"Resources" : {  
    "MyInstance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
            "UserData" : {  
                "Fn::Base64" : {  
                    "Fn::Join" : [ "", [ "Queue=", { "Ref" : "MyQueue" } ] ]  
                } },  
            "AvailabilityZone" : "us-east-1a",  
            "ImageId" : "ami-20b65349" }  
        },  
    "MyQueue" : {  
        "Type" : "AWS::SQS::Queue",  
        "Properties" : { } } }
```

JSON example

Resource Attribute: DependsOn

The "DependsOn" attribute specifies that the creation of a specific resource follows another. You can use the DependsOn attribute with any resource.

```
"Resources" : {  
    "AppServerInstance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
            "ImageId" : {  
                "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]  
            }  
        },  
        "DependsOn" : "myDB"  
    },  
    "myDB" : {  
        "Type" : "AWS::RDS::DBInstance",  
        "Properties" : {  
            ...  
        }  
    }  
}
```



Creates the Amazon EC2 instance **only after** the RDS database instance has been created.

JSON example

When a DependsOn Attribute Is Required

The following resources depend on a VPC gateway attachment when they have an associated public IP address and are in a VPC:

- Auto Scaling groups
- Amazon EC2 instances
- Elastic Load Balancing load balancers
- Elastic IP addresses
- Amazon RDS database instances
- Amazon VPC routes that include the internet gateway

What Is Going to Break?



If you share your template, what could **potentially break**?

Things that are specific to your environment, such as:

- Amazon EC2 key pairs
- Security group names
- Subnet ID
- AWS EBS snapshot IDs

```
"Resources" : {  
    "Ec2Instance" : {  
        "Type" :  
        "AWS::EC2::Instance",  
        "Properties" : {  
            "KeyName" : "MyKeyPair",  
            "ImageId" : "ami-75g0061f",  
            "InstanceType" : "m1.medium"  
            ...  
        } } },
```

How can you fix this?

Parameters, Mappings, and Conditions

Anatomy of an AWS CloudFormation Template



```
"Description" : "JSON  
string",  
"Metadata" : {  
    template metadata },  
"Parameters" : {  
    set of parameters },  
"Mappings" : {  
    set of mappings },  
"Conditions" : {  
    set of conditions },  
"Resources" : {  
    set of resources },  
"Outputs" : {  
    set of outputs }
```

JSON example

Parameters:

Values you can pass in to your template at runtime.

- Allow stacks to be customized at launch of a template
- Can specify allowed and default values for each parameter

AWS CloudFormation Template: Parameters Example



```
"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." } }
```

```
"Resources" : {  
    "Ec2Instance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
            "InstanceType" : { "Ref" : "InstanceTypeParameter" },  
            "ImageId" : "ami-2f726546"  
        }  
    } }
```

JSON example

Anatomy of an AWS CloudFormation Template



```
"Description" : "JSON  
string",  
  
"Metadata" : {  
    template metadata },  
  
"Parameters" : {  
    set of parameters },  
  
"Mappings" : {  
    set of mappings },  
  
"Conditions" : {  
    set of conditions },  
  
"Resources" : {  
    set of resources },  
  
"Outputs" : {  
    set of outputs }
```

JSON example

Mappings:

Keys and associated values that specify conditional parameter values.

```
"Mappings" : {  
    "RegionMap" : {  
        "us-east-1" : { "64" : "ami-6411e20d"},  
        "eu-west-1" : { "64" : "ami-37c2f643"},  
        "ap-southeast-1" : { "64" : "ami-66f28c34"}  
    }  
}
```

AWS CloudFormation Template: Mappings Example



```
"Mappings" : {  
    "RegionAndInstanceTypeToAMIID" : {  
        "us-east-1" : {  
            "m1.small" : "ami-1ccae774",  
            "t2.micro" : "ami-1ecae776"  
        },  
        "us-west-2" : {  
            "m1.small" : "ami-ff527ecf",  
            "t2.micro" : "ami-e7527ed7"  
        }  
    }  
}
```

Specify multiple mapping levels

JSON example

Anatomy of an AWS CloudFormation Template



```
"Description" : "JSON  
string",  
"Metadata" : {  
    template metadata },  
"Parameters" : {  
    set of parameters },  
"Mappings" : {  
    set of mappings },  
"Conditions" : {  
    set of conditions },  
"Resources" : {  
    set of resources },  
"Outputs" : {  
    set of outputs }
```

JSON example

Conditions:

Control whether certain resources are created or certain properties are assigned a value during stack creation or update.

Anatomy of an AWS CloudFormation Template: Conditions

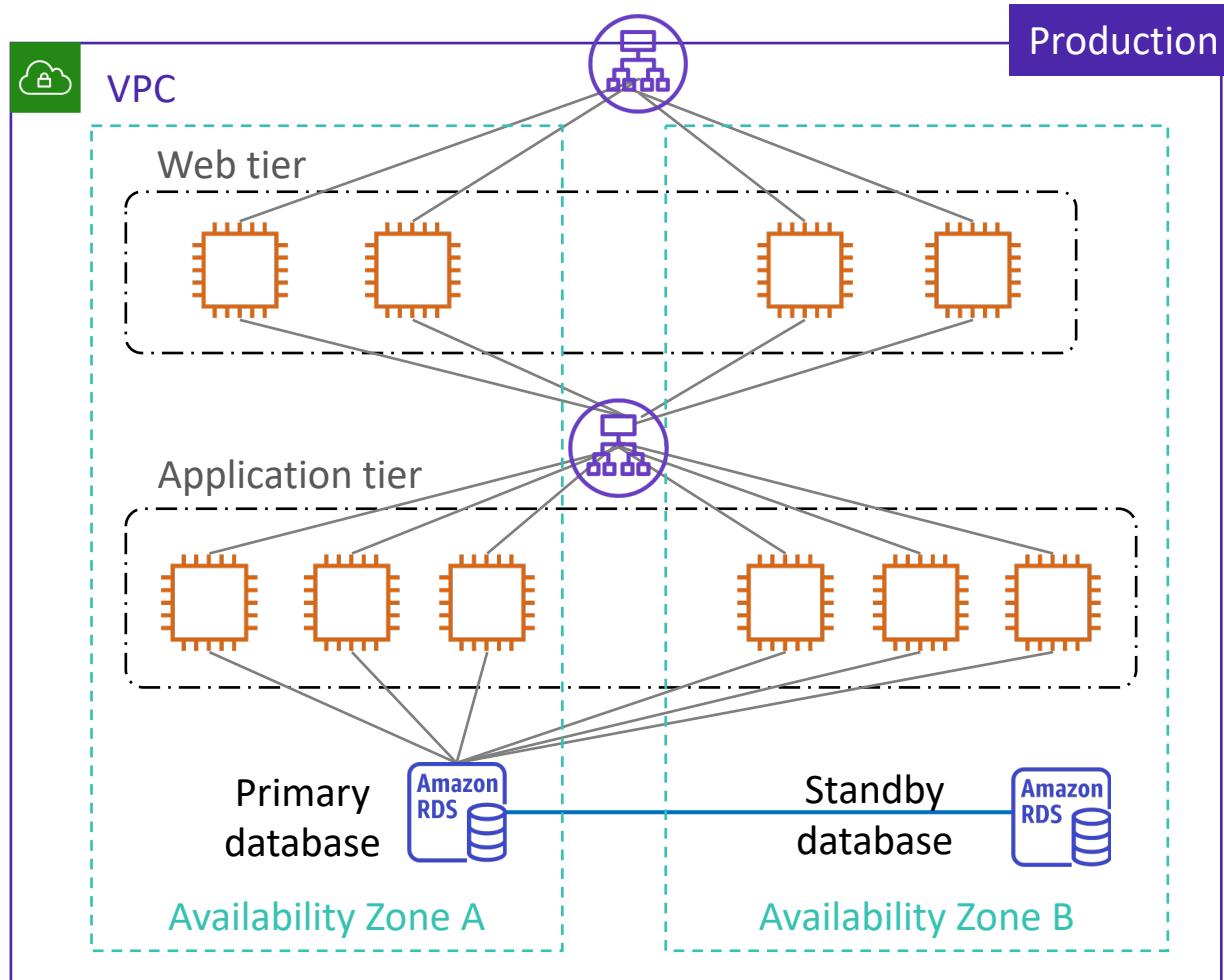


```
"Parameters" : {  
    "EnvType" : {  
        "Description" : "Specifies if this is a Dev, QA or Prod environment",  
        "Type" : "String",  
        "Default" : "Dev",  
        "AllowedValues" : ["Dev", "QA", "Prod"]  
    },  
    ...  
},  
  
"Conditions" : {  
    "CreateProdResources" : {"Fn::Equals" : [{"Ref" : "EnvType"}, "Prod"]}  
},  
...
```

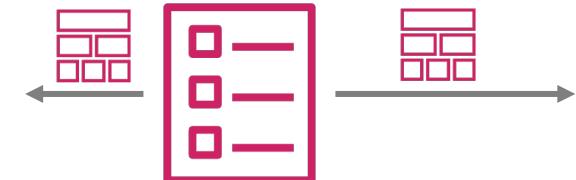
JSON example

CreateProdResources condition evaluates to true if EnvType is Prod

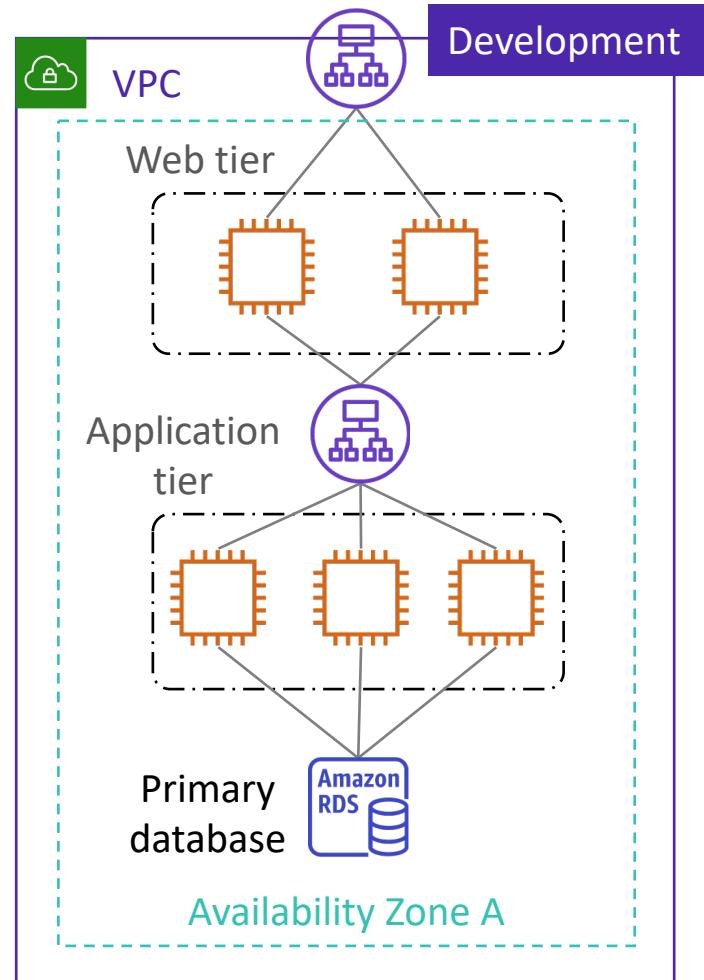
AWS CloudFormation



Need to update a configuration?
Edit the template details, then update the stacks.



A single AWS CloudFormation template with defined **conditions** can deploy a different application infrastructure to different environments.



Anatomy of a CloudFormation Template



```
"Description" : "JSON  
string",  
"Metadata" : {  
    template metadata },  
"Parameters" : {  
    set of parameters },  
"Mappings" : {  
    set of mappings },  
"Conditions" : {  
    set of conditions },  
"Resources" : {  
    set of resources },  
"Outputs" : {  
    set of outputs }
```

Outputs:

Values returned whenever you view your stack's properties.

Properties:

Declares output values that you want to view from the CloudFormation console or that you want to return in response to describe-stack calls.

```
"Outputs" : {  
    "<Logical ID>" : {  
        "Description" : "<Information about the value>",  
        "Value" : "<Value to return>" } }
```

Lecture References

References

Recommend Viewing

Swinburne Lecture – High Level Overview

AWS Academy – Deeper dive

ACA Module 10