

Using Parameters and Outputs to Customize a Stack



Overview



Reuse templates for multiple slightly different stacks

Add parameters to a template

Integrate stacks into existing environments

Use outputs to access results from a running stack



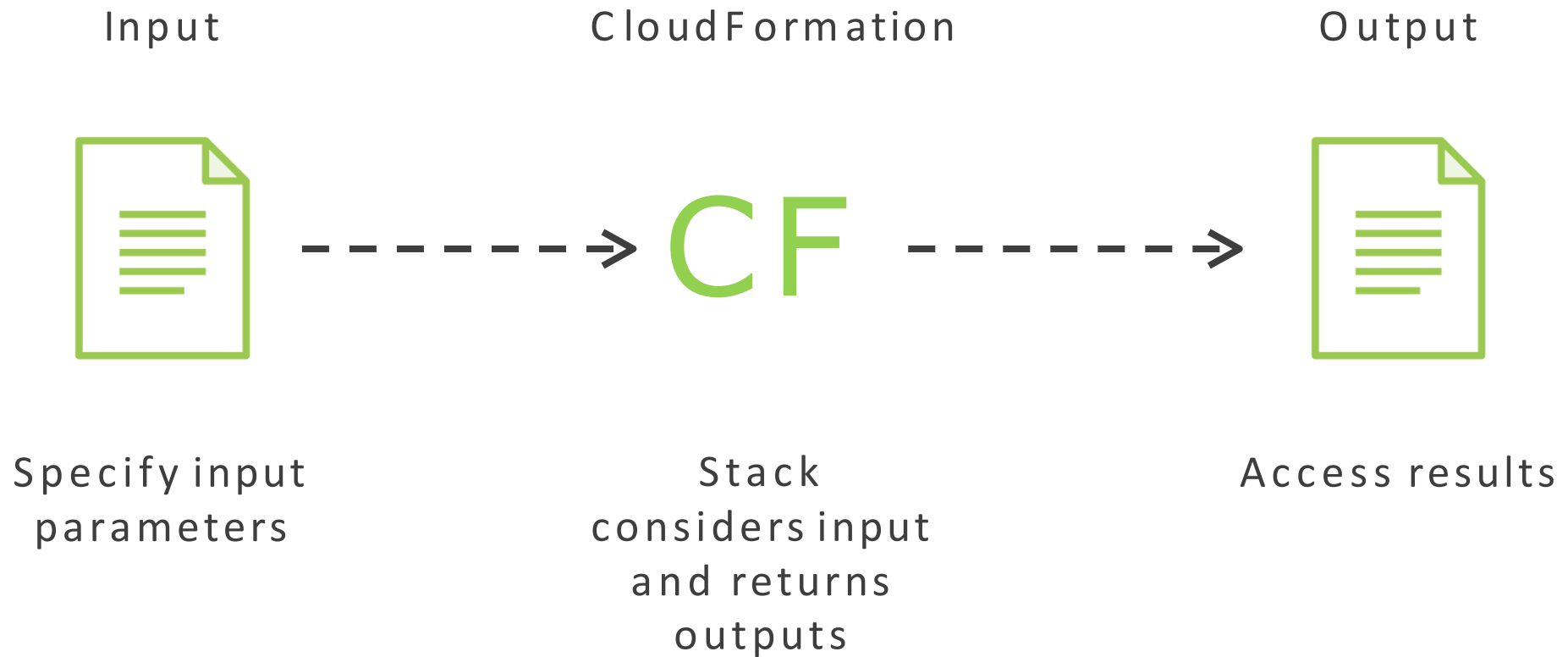
Already done

- CloudFormation template including a SSH bastion host environment

To do

- Customize stack for slightly different use cases

Input and Output



Parameters to Specify Input

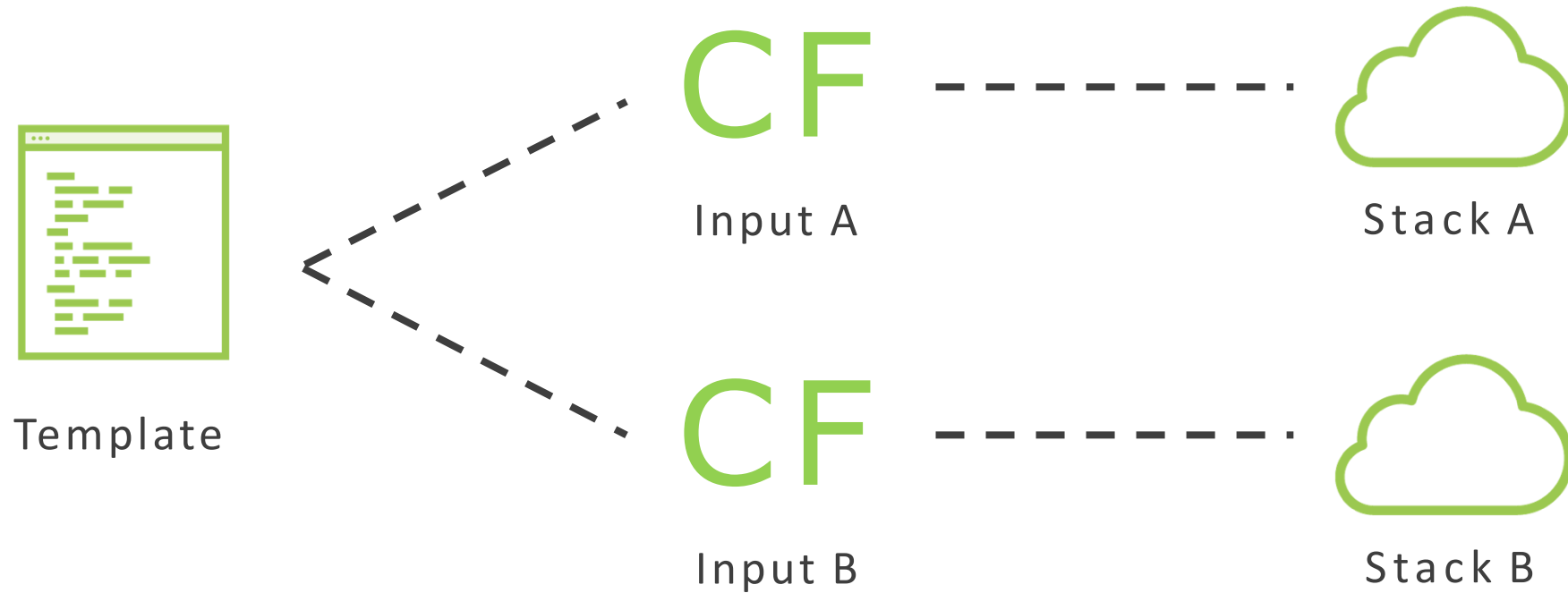


Adam wants to reuse his template for multiple SSH bastion host setups

Small changes are needed depending for different setups

- Key Pair
- Instance Type

Reuse Template



Parameters Section

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "SSH Bastion Host",  
  "Parameters": {...},  
  "Resources": {...}  
}
```


Defining a Parameter

```
"Parameters": {  
    "KeyPair": {  
        "Description": "A SSH key pair.",  
        "Type": "AWS::EC2::KeyPair::KeyName"  
    }  
}
```

Defining a Parameter

```
"Parameters": {  
  "KeyPair": {  
    "Description": "A SSH key pair.",  
    "Type": "AWS::EC2::KeyPair::KeyName"  
  }  
}
```

```
"InstanceType": {  
    "Description": "The instance type.",  
    "Type": "String"  
}
```

Description

Describe your parameters to simplify filling them out later

Defining a Parameter

```
"Parameters": {  
  "KeyPair": {  
    "Description": "A SSH key pair.",  
    "Type": "AWS::EC2::KeyPair::KeyName"  
  }  
}
```

String : "string"

Number : 1

List<Number> : [1,2,3]

CommaDelimitedList : "1,2,3"

Parameter Type

General parameter types

AWS::EC2::Image::Id : "ami-b43503a4"

AWS::EC2::Instance::Id : "i-47cad982"

AWS::EC2::SecurityGroup::Id : "sg-6246b70c"

AWS::Route53::HostedZone::Id : "Z2VHZWU704GYI1"

Parameter Type

AWS parameter types

Referencing Parameters

```
"Parameters": {  
    "KeyPair": {...}  
},  
"Resources": {  
    "EC2Instance": {  
        "Properties": {  
            "KeyName": {"Ref": "KeyPair"}  
        }  
    }  
}
```

Referencing Parameters

```
"Parameters": {  
    "KeyPair": {...}  
},  
"Resources": {  
    "EC2Instance": {  
        "Properties": {  
            "KeyName": {"Ref": "KeyPair"}  
        }  
    }  
}
```


AllowedValues : ["a", "b", "c"]

AllowedPattern : "[a-z0-9]*"

MaxLength / MinLength : 1

MaxValue / MinValue : 2

Input Validation

Software development 101: validate your inputs



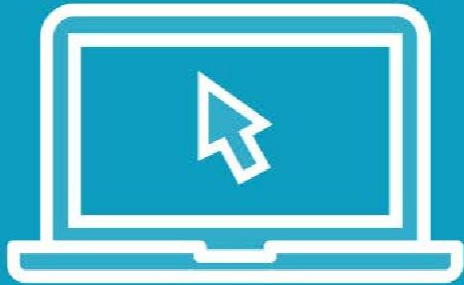
Validating Instance Type

```
"InstanceType": {  
  "Description": "The instance type.",  
  "Type": "String",  
  "AllowedValues": ["t2.nano", "t2.micro", "t2.small"]  
}
```

Default Value

```
"InstanceType": {  
  "Description": "The instance type.",  
  "Type": "String",  
  "Default": "t2.nano"  
}
```

Demo



Walk -through template

- Input Parameters
- Validation

Create stack out of template by using
Command Line Interface (CLI)

Specify input parameters for stack

Outputs to Access Results from Stack



Adam wants to integrate SSH bastion server stacks into existing environment

He needs to access the results from a stack to be able to

- Update DNS record with a Public IP
- Add a Private IP to firewall configuration

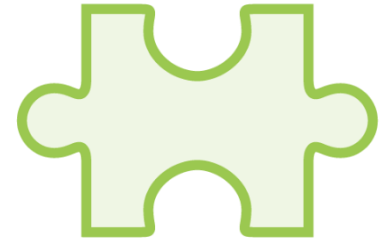
Access Results of Stack



Template
defines outputs



Stack returns
outputs after
stack was created



Outputs can be used
for integration with
other environments

Outputs Section

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "SSH Bastion Host",  
  "Parameters": {...},  
  "Resources": {...},  
  "Outputs": {...}  
}
```


Defining an Output

```
"Outputs": {  
  "SSHBastionHostPublicIp": {  
    "Description": "Public IP address ...",  
    "Value": {"Ref": "ElasticIP"}  
  }  
}
```

Defining an Output

```
"Outputs": {  
  "SSHBastionHostPublicIp": {  
    "Description": "Public IP address ...",  
    "Value": {"Ref": "ElasticIP"}  
  }  
}
```

Defining an Output

```
"Outputs": {  
  "SSHBastionHostPublicIp": {  
    "Description": "Public IP address ...",  
    "Value": {"Ref": "ElasticIP"}  
  }  
}
```

```
"SSHBastionHostPublicIp": {  
    "Description": "Public IP address ...",  
    "Value": {"Ref": "ElasticIP"}  
}
```

Default Resource Return Value

Accessible via built-in Ref function

Return Value depends on the ResourceType



```
"SSHBastionHostPrivateIp": {  
  "Description": "Private IP address ...",  
  "Value": {"Fn::GetAtt": ["EC2Instance", "PrivateIp"]}  
}
```

Specific Resource Return Value

Accessible via built-in Fn::GetAtt function

Different return values depending on the Resource Type



Returning Public IP

```
"Resources": {  
    "ElasticIP": {...}  
}  
"Outputs": {  
    "SSHBastionHostPublicIp": {  
        "Description": "Public IP address ...",  
        "Value": {"Ref": "ElasticIP"}  
    }  
}
```

Returning Public IP

```
"Resources": {  
    "ElasticIP": {...}  
}  
  
"Outputs": {  
    "SSHBastionHostPublicIp": {  
        "Description": "Public IP address ...",  
        "Value": {"Ref": "ElasticIP"}  
    }  
}
```

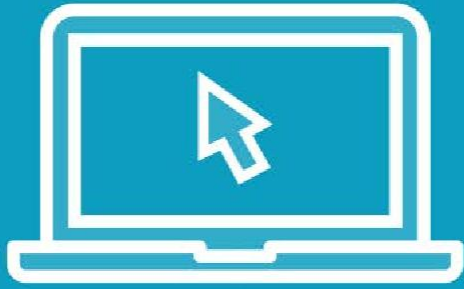
Returning Private IP

```
"Resources": {  
    "EC2": {...}  
}  
"Outputs": {  
    "SSHBastionHostPrivateIp": {  
        "Description": "Private IP address...",  
        "Value": {"Fn::GetAtt": ["EC2", "PrivateIp"]}  
    }  
}
```


Returning Private IP

```
"Resources": {  
    "EC2": {...}  
}  
  
"Outputs": {  
    "SSHBastionHostPrivateIp": {  
        "Description": "Private IP address...",  
        "Value": {"Fn::GetAtt": ["EC2", "PrivateIp"]}  
    }  
}
```

Demo



Walk -through template

- Defining Outputs
- Accessing return values from resources

Get outputs from a running stack via CLI

Summary



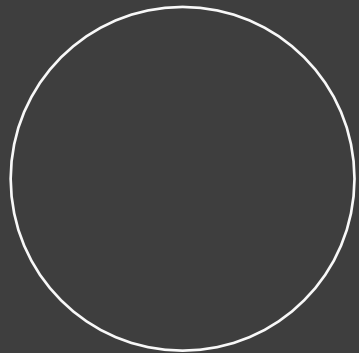
Inputs

- Allows you to reuse a template in slightly different use cases
- Template defines possible Parameters
- Parameters have to be specified when creating/updating a stack

Outputs

- Allows you to access results from a stack
- Template defines outputs
- Outputs can be accessed after stack was created/updated successfully

Using Parameters and Outputs to Customize a Stack



Overview



Reuse templates for multiple slightly different stacks

Add parameters to a template

Integrate stacks into existing environments

Use outputs to access results from a running stack



Already done

- CloudFormation template including a SSH bastion host environment

To do

- Customize stack for slightly different use cases



CF



Parameters to Specify Input



Adam wants to reuse his template for multiple SSH bastion host setups

Small changes are needed depending for different setups

- Key Pair
- Instance Type



CF



CF



Parameters Section

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "SSH Bastion Host",  
  "Parameters": {...},  
  "Resources": {...}  
}
```

Defining a Parameter

```
"Parameters": {  
  "KeyPair": {  
    "Description": "A SSH key pair.",  
    "Type": "AWS::EC2::KeyPair::KeyName"  
  }  
}
```

Defining a Parameter

```
"Parameters": {  
  "KeyPair": {  
    "Description": "A SSH key pair.",  
    "Type": "AWS::EC2::KeyPair::KeyName"  
  }  
}
```


Defining a Parameter

```
"Parameters": {  
  "KeyPair": {  
    "Description": "A SSH key pair.",  
    "Type": "AWS::EC2::KeyPair::KeyName"  
  }  
}
```


Referencing Parameters

```
"Parameters": {  
    "KeyPair": {...}  
},  
"Resources": {  
    "EC2Instance": {  
        "Properties": {  
            "KeyName": {"Ref": "KeyPair"}  
        }  
    }  
}
```

Referencing Parameters

```
"Parameters": {  
    "KeyPair": {...}  
},  
"Resources": {  
    "EC2Instance": {  
        "Properties": {  
            "KeyName": {"Ref": "KeyPair"}  
        }  
    }  
}
```

AllowedValues ["a", "b", "c"]

AllowedPattern "[a-z0-9]*"

MaxLength MinLength 1

MaxValue MinValue 2



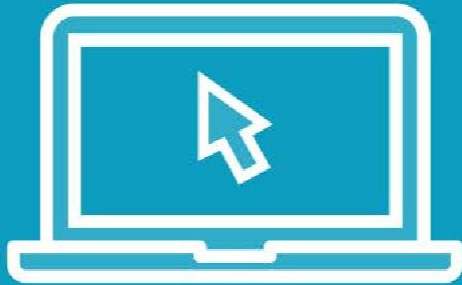
Validating Instance Type

```
"InstanceType": {  
  "Description": "The instance type.",  
  "Type": "String",  
  "AllowedValues": ["t2.nano", "t2.micro", "t2.small"]  
}
```

Default Value

```
"InstanceType": {  
  "Description": "The instance type.",  
  "Type": "String",  
  "Default": "t2.nano"  
}
```

Demo



Walk -through template

- Input Parameters
- Validation

Create stack out of template by using
Command Line Interface (CLI)

Specify input parameters for stack

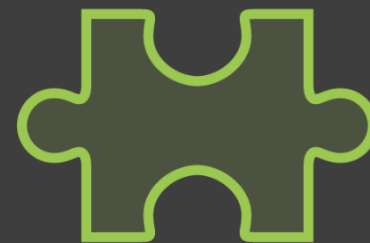
Outputs to Access Results from Stack



Adam wants to integrate SSH bastion server stacks into existing environment

He needs to access the results from a stack to be able to

- Update DNS record with a Public IP
- Add a Private IP to firewall configuration



Outputs Section

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "SSH Bastion Host",  
  "Parameters": {...},  
  "Resources": {...},  
  "Outputs": {...}  
}
```

Defining an Output

```
"Outputs": {  
  "SSHBastionHostPublicIp": {  
    "Description": "Public IP address ...",  
    "Value": {"Ref": "ElasticIP"}  
  }  
}
```

Defining an Output

```
"Outputs": {  
  "SSHBastionHostPublicIp": {  
    "Description": "Public IP address ...",  
    "Value": {"Ref": "ElasticIP"}  
  }  
}
```

Defining an Output

```
"Outputs": {  
  "SSHBastionHostPublicIp": {  
    "Description": "Public IP address ...",  
    "Value": {"Ref": "ElasticIP"}  
  }  
}
```

Ref ElasticIP



`Fn::GetAtt` "EC2Instance", "PrivateIp"



Returning Public IP

```
"Resources": {  
    "ElasticIP": {...}  
}  
"Outputs": {  
    "SSHBastionHostPublicIp": {  
        "Description": "Public IP address ...",  
        "Value": {"Ref": "ElasticIP"}  
    }  
}
```

Returning Public IP

```
"Resources": {  
    "ElasticIP": {...}  
}  
"Outputs": {  
    "SSHBastionHostPublicIp": {  
        "Description": "Public IP address ...",  
        "Value": {"Ref": "ElasticIP"}  
    }  
}
```

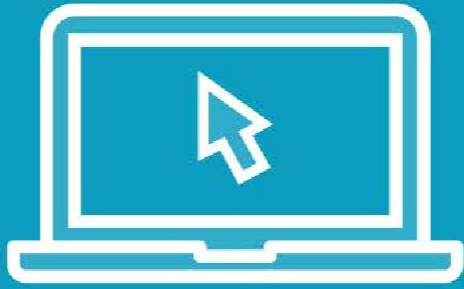
Returning Private IP

```
"Resources": {  
    "EC2": {...}  
}  
"Outputs": {  
    "SSHBastionHostPrivateIp": {  
        "Description": "Private IP address...",  
        "Value": {"Fn::GetAtt": ["EC2", "PrivateIp"]}  
    }  
}
```

Returning Private IP

```
"Resources": {  
    "EC2": {...}  
}  
"Outputs": {  
    "SSHBastionHostPrivateIp": {  
        "Description": "Private IP address...",  
        "Value": {"Fn::GetAtt": ["EC2", "PrivateIp"]}  
    }  
}
```

Demo



Walk -through template

- Defining Outputs
- Accessing return values from resources

Get outputs from a running stack via CLI

Summary



Inputs

- Allows you to reuse a template in slightly different use cases
- Template defines possible Parameters
- Parameters have to be specified when creating/updating a stack

Outputs

- Allows you to access results from a stack
- Template defines outputs
- Outputs can be accessed after stack was created/updated successfully