

Mappings

In this lab, we are working with AWS CloudFormation to create and manage stacks in different regions and environments. The process involves creating mappings for Amazon Machine Image (AMI) IDs and different environments (development and production). Here is a summary of the steps and the end goals:

Creating a Map for AMI IDs

1. **Define Mappings:** We define mappings in our CloudFormation template for AMI IDs in two regions of our choice.
2. **Deploy Stack in Singapore:** We create a CloudFormation stack in the Singapore region using the template.
3. **Deploy Stack in Sydney:** We create another CloudFormation stack in the Sydney region using the same template.
4. **Verification:** Verify that the stacks and instances are created and running in both regions.
5. **Cleanup:** Delete the stacks from both regions once done.

Creating a Map for Environment

1. **Define Environment Mappings:** We define mappings for different environments (development and production) with corresponding instance types.
2. **Deploy Stack for Development Environment:** Create a stack in CloudFormation, choosing the development environment, and verify that the instance type is as expected (t2.micro).
3. **Deploy Stack for Production Environment:** Create a stack in the Sydney region, choosing the production environment, and verify that the instance type is as expected (t2.small).
4. **Verification:** Ensure that the stacks and instances are created and running as specified.
5. **Cleanup:** Delete the stacks from both regions once done.

End Goal

The end goal is to understand and practice the use of mappings in AWS CloudFormation templates to manage resources across different regions and environments effectively. This exercise demonstrates how to automate the deployment of resources using templates and how to handle environment-specific configurations.

To begin with the Lab

Creating a Map for AMI

1. Now in this lab we will create Mapping for AMI IDs. We will choose two regions of our choice and we will mention AMI ID from both areas in our code or say template. Then we will create our stack in both regions to verify things.

- So, you can see the code below and in this, you will know that we have a mappings section, where we have given two different regions, and we will create our stack in both these regions mentioned below in the code.
- Also, you will get these templates from GitHub.

```

1  AwSTemplateFormatVersion: 2010-09-09
2
3  < Parameters:
4    MyKeyName:
5      Type: AWS::EC2::KeyPair::KeyName
6      Description: Select the keypair
7
8  < Mappings:
9    MyRegionMap:
10   ap-southeast-1:
11     HVM64: ami-0e97ea97a2f374e3d
12   ap-southeast-2:
13     HVM64: ami-030a5acd7c996ef60
14
15 < Resources:
16   MyInstance:
17     Type: AWS::EC2::Instance
18     Properties:
19       ImageId: !FindInMap
20         - MyRegionMap
21         - !Ref 'AWS::Region'
22         - HVM64
23       InstanceType: t2.micro
24       KeyName: !Ref MyKeyName
25

```

- Now we will first go to the Singapore region and create our stack in cloud formation. Choose the same options as shown below and upload your template.

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Choose an existing template
Upload or choose an existing template.

Use a sample template
Choose from our sample template library.

Build from Application Composer
Create a template using a visual builder.

Specify template Info
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL
Provide an Amazon S3 URL to your template.

Upload a template file
Upload your template directly to the console.

Sync from Git - new
Sync a template from your Git repository.

Upload a template file

- Then you need to give your stack a name and choose your key pair, as we have defined a parameter section.
- After just move to the review page and create your stack.

Specify stack details

Provide a stack name

Stack name
Stack-Mappings

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 14/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

MyKeyName
Select the keypair
CFKeyPair

Cancel Previous Next

- Below you can see that in the events tab we have the process and we can see that our stack has been created.

CloudFormation > Stacks > Stack-Mappings

Stack-Mappings

Stack info Events Resources Outputs Parameters Template Change sets Git sync - new

Events (7)

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-19 13:16:32 UTC+0530	Stack-Mappings	CREATE_COMPLETE	-	-
2024-07-19 13:16:31 UTC+0530	MyInstance	CREATE_COMPLETE	-	-
2024-07-19 13:16:21 UTC+0530	Stack-Mappings	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated
2024-07-19 13:16:21 UTC+0530	MyInstance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated
2024-07-19 13:16:10 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	Resource creation initiated
2024-07-19 13:16:08 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	-
2024-07-19 13:16:06 UTC+0530	Stack-Mappings	CREATE_IN_PROGRESS	-	User Initiated

- Also, if you go to EC2 you will see our instance has been running.

The screenshot shows the AWS CloudWatch Instances console. At the top, there's a search bar and a filter dropdown set to 'All states'. Below the header is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and a timestamp. One row is selected, showing 'i-07f4d66e55e34cb18' as the instance ID, 'Running' as the state, 't2.micro' as the instance type, and 'Initializing' as the status check. The availability zone is listed as 'ap-southeast-1a'. At the bottom of the table, there's a link to 'View alarms'.

i-07f4d66e55e34cb18

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary [Info](#)

Instance details [Info](#)

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-0e97ea97a2f374e3d	disabled
Platform details	AMI name	Termination protection
Linux/UNIX	al2023-ami-2023.5.20240708.0-kernel-6.1-x86_64	Disabled
Stop protection	Launch time	AMI location
Disabled	Fri Jul 19 2024 13:16:10 GMT+0530 (India Standard Time) (2 minutes)	amazon/al2023-ami-2023.5.20240708.0-kernel-6.1-x86_64
Instance auto-recovery	Lifecycle	Stop-hibernate behavior
Default	normal	Disabled
AMI Launch index	Key pair assigned at launch	State transition reason
0	CFKeyPair	-

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

9. Now we will go to the Sydney region and create our stack there. As you can see below, we are in the Sydney region, and we will create our stack here this time.

The screenshot shows the AWS CloudFormation console. On the left, there's a sidebar with options like 'Stacks', 'StackSets', 'Exports', 'Application Composer', and 'Registry'. The main area is titled 'CloudFormation > Stacks' and shows a table with one row: 'No stacks'. There's a 'Create stack' button at the bottom of the table.

10. So, now upload your template. Then move to the next page and give your stack a name then choose your key pair. Just remember in the new region you need to create a key pair before you create your stack or you can create one at the same time.

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Choose an existing template
Upload or choose an existing template.

Use a sample template
Choose from our sample template library.

Build from Application Composer
Create a template using a visual builder.

Specify template [Info](#)
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL
Provide an Amazon S3 URL to your template.

Upload a template file
Upload your template directly to the console.

Sync from Git - new
Sync a template from your Git repository.

Upload a template file

Specify stack details

Provide a stack name

Stack name
Stack-Mappings-Sydney

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 21/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

MyKeyName
Select the keypair
new

Cancel Previous Next

11. Below you can see that our stack has been created successfully.

CloudFormation > Stacks > Stack-Mappings-Sydney

Stacks (1)

Events (7)

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-19 13:27:55 UTC+0530	Stack-Mappings-Sydney	CREATE_COMPLETE	-	-
2024-07-19 13:27:54 UTC+0530	MyInstance	CREATE_COMPLETE	-	-
2024-07-19 13:27:45 UTC+0530	Stack-Mappings-Sydney	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated
2024-07-19 13:27:44 UTC+0530	MyInstance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated
2024-07-19 13:27:33 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	Resource creation Initiated
2024-07-19 13:27:32 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	-
2024-07-19 13:27:29 UTC+0530	Stack-Mappings-Sydney	CREATE_IN_PROGRESS	-	User Initiated

12. And we also have an instance running.

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
i-03fc578e69fb32987	i-03fc578e69fb32987	Running	t2.micro	Initializing	View alarms +	ap-southeast-2a	ec2-3-107-26-104.ap-southeast-2.compute.amazonaws.com	3.107.26.104

i-03fc578e69fb32987

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID i-03fc578e69fb32987	Public IPv4 address 3.107.26.104 open address	Private IPv4 addresses 172.31.14.193
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-107-26-104.ap-southeast-2.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-14-193.ap-southeast-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-14-193.ap-southeast-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.
Auto-assigned IP address 5.107.26.104 [Public IP]	VPC ID vpc-009af1c4df3b5a9da	Learn more

13. Once you are done delete your stacks from both the regions.

Creating a Map for Environment

1. Now we are going to add mapping for the environment and here we will create two types of environments, one for development and the other one for production.
2. Based on the environment we have given our instance types.

```

1  AWS::TemplateFormatVersion: 2010-09-09
2
3  Parameters:
4      MyKeyName:
5          Type: AWS::EC2::KeyPair::KeyName
6          Description: Select the keypair
7      EnvironmentName:
8          Description: Select the environment
9          Type: String
10         Default: dev
11         AllowedValues:
12             - dev
13             - prod
14
15
16  Mappings:
17      MyRegionMap:
18          ap-southeast-1:
19              HVM64: ami-0e97ea97a2f374e3d
20          ap-southeast-2:
21              HVM64: ami-030a5acd7c996ef60
22
23      MyEnvironmentMap:
24          dev:
25              instanceType: t2.micro
26          prod:
27              instanceType: t2.small
28
29  Resources:
30      MyInstance:
31          Type: AWS::EC2::Instance
32          Properties:
33              ImageId: !FindInMap
34                  - MyRegionMap
35                  - !Ref 'AWS::Region'
36                  - HVM64
37              InstanceType: !FindInMap
38                  - MyEnvironmentMap
39                  - !Ref EnvironmentName
40                  - instanceType
41              KeyName: !Ref MyKeyName
42

```

3. Now go to cloud formation and create your stack. Choose the same options as shown below, upload your template and move to next page.

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Choose an existing template
Upload or choose an existing template.

Use a sample template
Choose from our sample template library.

Build from Application Composer
Create a template using a visual builder.

Specify template Info

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL
Provide an Amazon S3 URL to your template.

Upload a template file
Upload your template directly to the console.

Sync from Git - new
Sync a template from your Git repository.

Upload a template file

Choose file

- Below you can see that we have given the stack name and then in the parameters you can see that we have our environment name section from here, we can choose any one of them dev or prod. Then choose your key pair and create your stack.

Specify stack details

Provide a stack name

Stack name

Stack-Mappings-Env

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 18/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

EnvironmentName

Select the environment

dev

MyKeyPair

Select the keypair

CFKeyPair

Cancel

Previous

Next

- Now you can see that our stack has been created.

CloudFormation > Stacks > Stack-Mappings-Env

Stack-Mappings-Env

Stacks Delete Update Stack actions Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Git sync - new

Events (7)

Search events

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-19 13:36:56 UTC+0530	Stack-Mappings-Env	CREATE_COMPLETE	-	-
2024-07-19 13:36:55 UTC+0530	MyInstance	CREATE_COMPLETE	-	-
2024-07-19 13:36:45 UTC+0530	Stack-Mappings-Env	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated
2024-07-19 13:36:45 UTC+0530	MyInstance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated
2024-07-19 13:36:34 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	Resource creation Initiated
2024-07-19 13:36:32 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	-
2024-07-19 13:36:30 UTC+0530	Stack-Mappings-Env	CREATE_IN_PROGRESS	-	User Initiated

- Also, you will notice that we have chosen the dev environment so our instance launched in t2.micro type.

The screenshot shows the AWS CloudWatch Metrics console. A single metric named "AWS Lambda Function Invocations" is displayed. The value is 1, and it is shown for the last hour. The metric is represented by a blue line graph with a single data point at the end.

- Now go to the Sydney region and create our stack again but this time we will choose the prod environment. Below you can see that we chose prod environment.

The screenshot shows the AWS CloudWatch Metrics console. A single metric named "AWS Lambda Function Invocations" is displayed. The value is 1, and it is shown for the last hour. The metric is represented by a blue line graph with a single data point at the end.

- And you can see that our stack has been created successfully.

Stack-Mappings-Env																																														
Stack info	Events	Resources	Outputs	Parameters	Template	Change sets																																								
Stacks Delete Update Stack actions ▾ Create stack ▾																																														
Events (7)																																														
<table border="1"> <thead> <tr> <th>Timestamp</th> <th>Logical ID</th> <th>Status</th> <th>Detailed status</th> <th>Status reason</th> </tr> </thead> <tbody> <tr> <td>2024-07-19 13:41:23 UTC+0530</td> <td>Stack-Mappings-Env</td> <td>CREATE_COMPLETE</td> <td>-</td> <td>-</td> </tr> <tr> <td>2024-07-19 13:41:22 UTC+0530</td> <td>MyInstance</td> <td>CREATE_COMPLETE</td> <td>-</td> <td>-</td> </tr> <tr> <td>2024-07-19 13:41:02 UTC+0530</td> <td>Stack-Mappings-Env</td> <td>CREATE_IN_PROGRESS</td> <td>CONFIGURATION_COMPLETE</td> <td>Eventual consistency check initiated</td> </tr> <tr> <td>2024-07-19 13:41:02 UTC+0530</td> <td>MyInstance</td> <td>CREATE_IN_PROGRESS</td> <td>CONFIGURATION_COMPLETE</td> <td>Eventual consistency check initiated</td> </tr> <tr> <td>2024-07-19 13:40:51 UTC+0530</td> <td>MyInstance</td> <td>CREATE_IN_PROGRESS</td> <td>-</td> <td>Resource creation initiated</td> </tr> <tr> <td>2024-07-19 13:40:49 UTC+0530</td> <td>MyInstance</td> <td>CREATE_IN_PROGRESS</td> <td>-</td> <td>-</td> </tr> <tr> <td>2024-07-19 13:40:46 UTC+0530</td> <td>Stack-Mappings-Env</td> <td>CREATE_IN_PROGRESS</td> <td>-</td> <td>User Initiated</td> </tr> </tbody> </table>							Timestamp	Logical ID	Status	Detailed status	Status reason	2024-07-19 13:41:23 UTC+0530	Stack-Mappings-Env	CREATE_COMPLETE	-	-	2024-07-19 13:41:22 UTC+0530	MyInstance	CREATE_COMPLETE	-	-	2024-07-19 13:41:02 UTC+0530	Stack-Mappings-Env	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	2024-07-19 13:41:02 UTC+0530	MyInstance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	2024-07-19 13:40:51 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	Resource creation initiated	2024-07-19 13:40:49 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	-	2024-07-19 13:40:46 UTC+0530	Stack-Mappings-Env	CREATE_IN_PROGRESS	-	User Initiated
Timestamp	Logical ID	Status	Detailed status	Status reason																																										
2024-07-19 13:41:23 UTC+0530	Stack-Mappings-Env	CREATE_COMPLETE	-	-																																										
2024-07-19 13:41:22 UTC+0530	MyInstance	CREATE_COMPLETE	-	-																																										
2024-07-19 13:41:02 UTC+0530	Stack-Mappings-Env	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated																																										
2024-07-19 13:41:02 UTC+0530	MyInstance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated																																										
2024-07-19 13:40:51 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	Resource creation initiated																																										
2024-07-19 13:40:49 UTC+0530	MyInstance	CREATE_IN_PROGRESS	-	-																																										
2024-07-19 13:40:46 UTC+0530	Stack-Mappings-Env	CREATE_IN_PROGRESS	-	User Initiated																																										

9. Now go to EC2, as we had chosen the prod environment we can see that our instance is of t2.small type.

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a filter button for 'Instance state = running'. Below the header, a table lists one instance: i-09fb19cb578a75f8f, which is running and has an instance type of t2.small. The instance has a public IPv4 address of 3.27.162.20 and a private IPv4 address of 172.31.15.129. It is associated with the VPC ID vpc-009af1c4df3b5a9da. The instance summary section provides detailed information about the instance's configuration, including its public and private IP addresses, instance state, instance type, and VPC ID.

10. Once you are done just delete your stacks from both of the regions.