



Amazon CloudFormation

Amazon CloudFormation is a service provided by Amazon Web Services (AWS) that allows you to automate the deployment of infrastructure resources in a consistent and repeatable manner. It uses templates written in JSON or YAML format to describe the desired state of your infrastructure, including things like EC2 instances, databases, networking configurations, and more.

With CloudFormation, you define your infrastructure as code, which allows you to version control your infrastructure changes, track modifications, and easily replicate environments across different regions or AWS accounts. CloudFormation manages the provisioning and configuration of resources, handling dependencies and ensuring that resources are created or updated in the correct order.

You can create, update, and delete entire stacks of resources using CloudFormation, making it a powerful tool for managing AWS infrastructure at scale. It also integrates with other AWS services, such as AWS CloudTrail for logging and AWS Identity and Access Management (IAM) for controlling access to resources.



Use cases of CloudFormation:

CloudFormation is widely used across various industries and organizations for automating the deployment and management of infrastructure resources on AWS. Some common use cases include:

1. **Application Deployment:** CloudFormation allows you to define all the resources needed for your application, such as EC2 instances, load balancers, databases, and storage, in a template. This enables you to consistently and reliably deploy your application across different environments, such as development, staging, and production.
2. **Microservices Architecture:** In a microservices architecture, applications are composed of multiple independent services. CloudFormation can manage the infrastructure for each service separately, making it easier to scale, update, and maintain the individual components of your application.
3. **Infrastructure as Code (IaC):** CloudFormation enables you to treat your infrastructure as code, allowing you to version control your infrastructure changes, track modifications, and collaborate more effectively with your team members. This helps ensure that your infrastructure configurations are reproducible and can be easily audited and reviewed.
4. **Disaster Recovery:** CloudFormation templates can be used to create backup infrastructure in a separate AWS region, enabling you to quickly recover your applications and data in the event of a disaster or outage in your primary region.
5. **DevOps Automation:** CloudFormation integrates with other DevOps tools and services, such as AWS CodePipeline and AWS CodeDeploy, to automate the deployment pipeline. This allows you to automatically trigger CloudFormation stacks based on code changes, perform automated testing, and deploy applications to production environments.

6. **Compliance and Governance:** CloudFormation can help enforce compliance and governance policies by defining security configurations, IAM roles, and resource tagging standards in your templates. This ensures that your infrastructure meets regulatory requirements and security best practices.
7. **Cost Optimization:** CloudFormation enables you to define cost-saving measures, such as auto-scaling policies, instance types, and resource tagging strategies, in your templates. This helps you optimize your infrastructure costs by right-sizing resources and minimizing waste.

Lab Exercises:

1. **Create Stack:**
 - Navigate to CloudFormation, upload a template, and create a stack.
 - Monitor the creation process and check the resources created in EC2.
2. **Update Stack:**
 - Update an existing stack by modifying the template and uploading it.
 - Track updates and ensure new configurations are applied.
3. **Create Change Set – Key Name:**
 - Use Change Sets to update configurations such as key pairs.
 - Execute changes and monitor the updates.
4. **Create Change Set – Instance Type:**
 - Change instance types using Change Sets.
 - Execute and monitor updates in EC2.

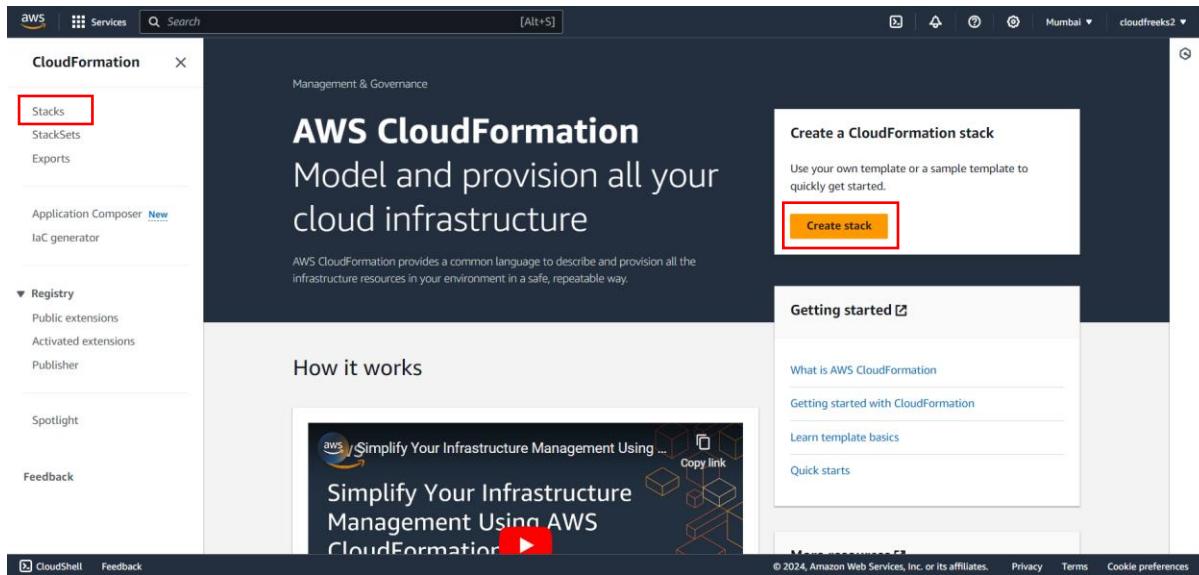
End Goal

Amazon CloudFormation simplifies the management of your AWS resources. By using templates, you can automatically set up and update your infrastructure (like servers and databases) without manually configuring each part. This helps ensure everything is done correctly and consistently, saves time, and makes it easier to manage large-scale deployments.

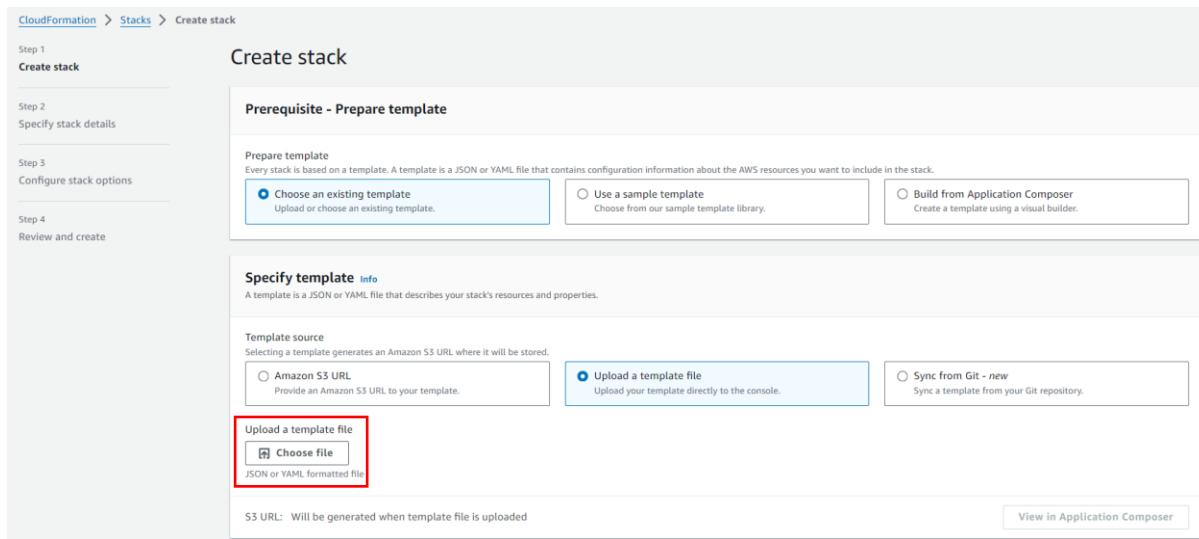
 **To begin with the Lab:**

 **Create Stack**

1. Login to AWS Console and navigate to CloudFormation. Then from its dashboard you need to click on Create stack or you can choose to go to Stacks.



2. Then you have to click on Choose an existing template and then click on Upload a template file.



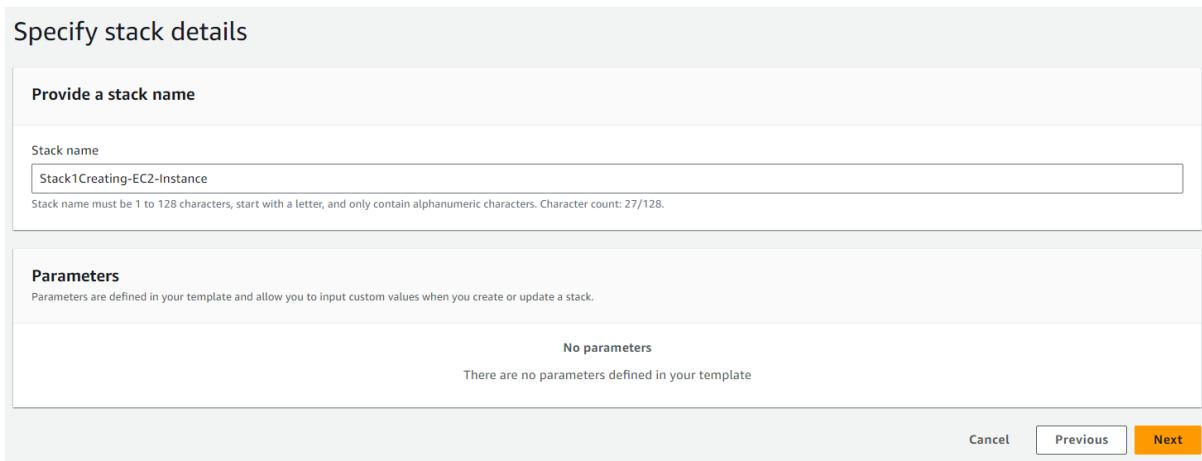
3. Now for the template file below you can see the code which we have written inside our template. Also, you will get all the templates on the GitHub.
4. Now download the templates unzip them and then you must change the **Image ID**, you need to go to EC2 click on launch instances and then click on browse for AMIs then choose Amazon Linux 2023 AMI ID and paste it here in the template.
5. After that save your template and then upload it to your stack.

```

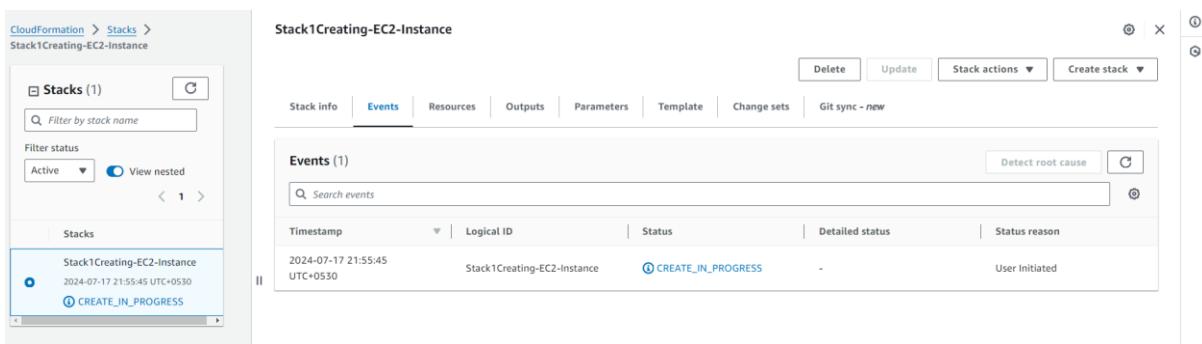
1 AWSTemplateFormatVersion: 2010-09-09
2 ▼ Resources:
3 ▼ EC2Instance:
4   Type: 'AWS::EC2::Instance'
5 ▼ Properties:
6   ImageId: ami-0ec0e125bb6c6e8ec
7   InstanceType: t2.micro
8

```

- Once you have uploaded your template file then click on next. Now you need to give your stack a name and then click on next.



- After that leave step 3 and move to step 4 which is your review page and create your stack.
- Below you can see that your stack is creating.



- Also, if you go to events then you can track the progress of your template. Below you can see that our template has been created.

							Delete	Update	Stack actions ▾	Create stack ▾
Stack info	Events	Resources	Outputs	Parameters	Template	Change sets	Git sync - new			
Events (7)										
<div style="display: flex; justify-content: space-between;"> Search events Detect root cause C </div>										
Timestamp	Logical ID	Status	Detailed status	Status reason						
2024-07-17 21:56:21 UTC+0530	Stack1Creating-EC2-Instance	CREATE_COMPLETE	-	-						
2024-07-17 21:56:20 UTC+0530	EC2Instance	CREATE_COMPLETE	-	-						
2024-07-17 21:56:00 UTC+0530	Stack1Creating-EC2-Instance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated						
2024-07-17 21:56:00 UTC+0530	EC2Instance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated						
2024-07-17 21:55:49 UTC+0530	EC2Instance	CREATE_IN_PROGRESS	-	Resource creation Initiated						
2024-07-17 21:55:47 UTC+0530	EC2Instance	CREATE_IN_PROGRESS	-	-						
2024-07-17 21:55:45 UTC+0530	Stack1Creating-EC2-Instance	CREATE_IN_PROGRESS	-	User Initiated						

10. Now if you navigate to EC2 you will see that an instance has been launched.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (CloudShell, Feedback). The main content area is titled "Instances (1/1) Info". It shows a table with one row for the instance i-09818e49f322f330b. The instance is listed as "Running" (status), t2.micro (instance type), and "Initializing" (status check). Below the table, there's a detailed view for the instance i-09818e49f322f330b, showing details like Public IPv4 address (3.111.157.151), Instance state (Running), and Instance type (t2.micro).

11. Also, if you go to Tags there you will see some tags which has been initiated by cloud formation.

The screenshot shows the "Tags" section of the EC2 instance details page for the instance i-09818e49f322f330b. At the top, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags (which is selected). The Tags section contains three entries:

Key	Value
aws:cloudformation:stack-name	Stack1Creating-EC2-Instance
aws:cloudformation:logical-id	EC2Instance
aws:cloudformation:stack-id	arn:aws:cloudformation:ap-south-1:878893308172:stack/Stack1Creating-EC2-Instance/38312e20-4459-11ef-8b33-06e41e853a01

Update Stack

1. Now we are going to see another feature of cloud formation which is update stack.
2. By using this feature, we can update our current stack. For now, we will first create a key pair in EC2 and then we will update our template.
3. So, navigate to EC2, and from the left pane go towards the key pair and then click to create a key pair. Below you can see that I have created a basic key pair with .pem format.

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type | [Info](#)
 RSA ED25519

Private key file format
 .pem
For use with OpenSSH
 .ppk
For use with PuTTY

Tags - *optional*
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Create key pair](#)

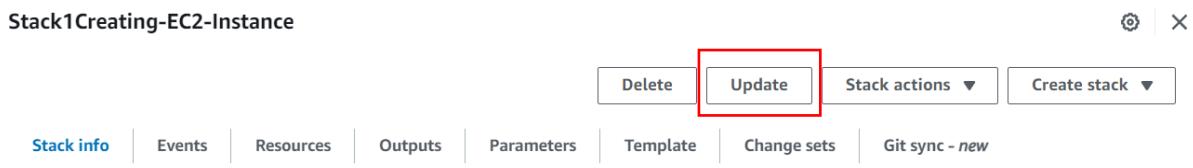
4. Below you can see the code that we are going to use in our template it is similar to our previous code we just added the Key name in the properties and gave the key pair name.

```

1 AWSTemplateFormatVersion: 2010-09-09
2 ▼ Resources:
3 ▼ EC2Instance:
4   Type: 'AWS::EC2::Instance'
5   Properties:
6     ImageId: ami-0ec0e125bb6c6e8ec
7     InstanceType: t2.micro
8     KeyName: CFkeyPair
9

```

- Now come back to cloud formation and in your stack, you will see a feature to update your stack. Click on it.



- Now you must choose Replace existing template then choose to upload your template from your local machine.

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.

Use existing template
Proceed with the template you are already using for this stack.

Replace existing template
Replace your existing template with a new template.

Edit in Application Composer
Edit your template in a visual builder.

Specify template
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

Upload a template file

Upload a template file

03-02-cfn-ec2-instance-update-Add-Key.yaml

JSON or YAML formatted file

S3 URL: <https://s3.ap-south-1.amazonaws.com/cf-templates-kmi81w4ukk76-ap-south-1/2024-07-17T164300.467Zpqj-03-02-cfn-ec2-instance-update-Add-Key.yaml>

- After that just move to the review page, here and scroll down to the bottom, then you will see that the action is to modify, and the replacement is true. This means that our template is genuine, and it will work fine.
- Now just click on Submit.

Change set preview

Changes (1)					
<input type="text"/> Search changes					
Action	Logical ID	Physical ID	Resource type	Replacement	Module
Modify	EC2Instance	i-09818e49f322f330b	AWS::EC2::Instance	True	-

[View change set](#)
Cancel
Previous
Submit

- Now in the events you can see that the update is completed the deletion is in progress for our previous template and the new template has been updated successfully.

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

Events (15)					
<input type="text"/> Search events					
Timestamp	Logical ID	Status	Detailed status	Status reason	
2024-07-17 22:31:52 UTC+0530	Stack1Creating-EC2-Instance	✓ UPDATE_COMPLETE	-	-	
2024-07-17 22:31:52 UTC+0530	EC2Instance	✓ DELETE_COMPLETE	-	-	
2024-07-17 22:31:13 UTC+0530	EC2Instance	ⓘ DELETE_IN_PROGRESS	-	-	
2024-07-17 22:31:12 UTC+0530	Stack1Creating-EC2-Instance	ⓘ UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-	-	

- Also, if you go to EC2, here you will see that your previous instance has been terminated and the new instance has the key pair assigned to it.

Instances (1/2) [info](#)

<input type="text"/> Find Instance by attribute or tag (case-sensitive)					
<input type="button" value="All states"/> < 1 > ↻					
Name	Instance ID	Instance state	Instance type	Status check	Actions
<input type="checkbox"/>	i-09818e49f322f330b	Terminated	t2.micro	-	Launch instances
<input checked="" type="checkbox"/>	i-06fb198eb4996eb2b	Running	t2.micro	2/2 checks passed	Actions

i-06fb198eb4996eb2b

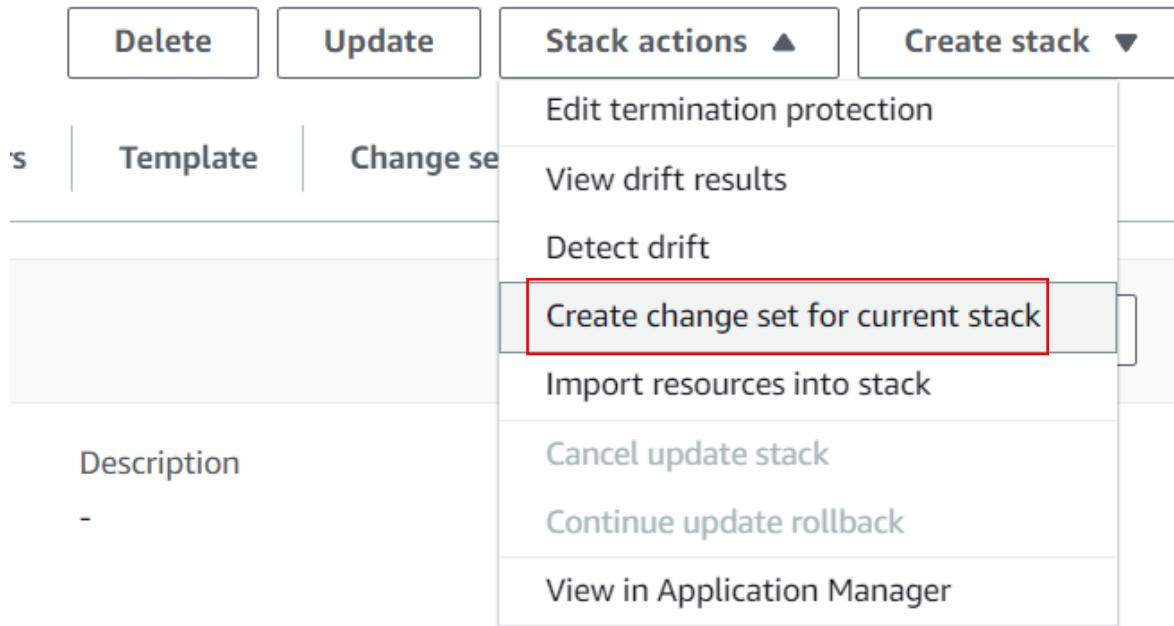
AMI Launch index 0	Key pair assigned at launch CFkeyPair	State transition reason -
Credit specification standard	Kernel ID -	State transition message -
Usage operation RunInstances	RAM disk ID -	Owner 878893308172

Create Change Set – Key Name

1. Now we are going to look at another feature called create change set. So, in the code below you can see that everything is same we just change the name of our key pair.

```
1 AWSTemplateFormatVersion: 2010-09-09
2 
3 Resources:
4   EC2Instance:
5     Type: 'AWS::EC2::Instance'
6     Properties:
7       ImageId: ami-0ec0e125bb6c6e8ec
8       InstanceType: t2.micro
9       KeyName: CFkeyPair2
```

2. So, now in your cloud formation you need to expand stack actions and choose to create change set for current stack.



3. And again, we need to replace it with the existing template and then upload our template.
4. After that just move to the review page and click on submit.

Create change set for Stack1Creating-EC2-Instance

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.

Use existing template

Proceed with the template you are already using for this stack.

Replace existing template

Replace your existing template with a new template.

Edit in Application Composer

Edit your template in a visual builder.

Specify template

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

Upload a template file

5. Now you will see that your change set has been created and now you have the ability to execute this change set.
6. If you want to execute then click on execute or else, you can delete this change set.
7. Now just click on execute change set.

[CloudFormation](#) > [Stacks](#) > [Stack1Creating-EC2-Instance](#) > Change sets: Stack1Crea-holeis473le-eng11c6si9o

Stack1Crea-holeis473le-eng11c6si9o

[Delete change set](#) [Execute change set](#)

Overview	
Change set ID arn:aws:cloudformation:ap-south-1:878893308172:changeSet/Stack1Crea-holeis473le-eng11c6si9o/5a551dee-ee2b-4680-9a7e-6b08bbe10086	Status CREATE_COMPLETE
Description	Status reason
Created time 2024-07-17 22:49:09 UTC+0530	Execution status AVAILABLE

[Changes](#) [Input](#) [Template](#) [JSON changes](#) [Hook invocations](#)

Changes (1)

Preview how proposed changes to a stack will impact running resources. Click on "View details" to preview the impact on property values for a resource.

Changes (1)							
Action	Logical ID	Resource type	Replacement	Property-level changes	Policy action	Physical ID	Module
Modify	EC2Instance	AWS::EC2::Instance	True	View details	ReplaceAndDelete	j-06fb198eb4996eb2b	-

8. Below you can see that your change set has failed because we don't have the keypair created in your EC2. So, first, we need to create a key pair. Also you can see that it has roll back all the changes if occurred any.

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-17 22:53:20 UTC+0530	Stack1Creating-EC2-Instance	✖ UPDATE_ROLLBACK_COMPLETE	-	-
2024-07-17 22:53:20 UTC+0530	EC2Instance	☑ DELETE_COMPLETE	-	-
2024-07-17 22:53:19 UTC+0530	Stack1Creating-EC2-Instance	✖ UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	-	-
2024-07-17 22:53:19 UTC+0530	EC2Instance	☑ UPDATE_COMPLETE	-	-
2024-07-17 22:53:17 UTC+0530	Stack1Creating-EC2-Instance	✖ UPDATE_ROLLBACK_IN_PROGRESS	-	The following resource(s) failed to update: [EC2Instance].
2024-07-17 22:53:16 UTC+0530	EC2Instance	✖ UPDATE_FAILED	-	Resource handler returned message: "The key pair 'CFkeyPair2' does not exist (Service: Ec2, Status Code: 400, Request ID: a02208ec-d115-4953-9b3a-e05288937f9e)" (RequestToken: 25a5d29f-75e3-f35b-ecf1-60d4dff93b03, HandlerErrorCode: InvalidRequest)

9. Once your keypair has been created then just repeat the process. Once you have executed the change set go to EC2 you will see that the new instance is creating and the old one is shutting down.

Instances (3) Info		Connect	Instance state	Actions	Launch instances	
<input type="text"/>	Find Instance by attribute or tag (case-sensitive)	All states				
<input type="checkbox"/>	Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>		i-09818e49f522f330b	Terminated	t2.micro	-	View alarms +
<input type="checkbox"/>		i-0e4aff42df9d47bbe	Running	t2.micro	Initializing	View alarms +
<input type="checkbox"/>		i-06fb198eb4996eb2b	Shutting-d...	t2.micro	-	View alarms +

10. Also, in the events section of your cloud formation you will see all the updates related to your stack.

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-17 22:57:54 UTC+0530	Stack1Creating-EC2-Instance	☑ UPDATE_COMPLETE	-	-
2024-07-17 22:57:54 UTC+0530	EC2Instance	☑ DELETE_COMPLETE	-	-
2024-07-17 22:57:20 UTC+0530	EC2Instance	⌚ DELETE_IN_PROGRESS	-	-
2024-07-17 22:57:19 UTC+0530	Stack1Creating-EC2-Instance	⌚ UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-	-
2024-07-17 22:57:18 UTC+0530	EC2Instance	☑ UPDATE_COMPLETE	-	-
2024-07-17 22:56:47 UTC+0530	EC2Instance	⌚ UPDATE_IN_PROGRESS	-	Resource creation initiated
2024-07-17 22:56:45 UTC+0530	EC2Instance	⌚ UPDATE_IN_PROGRESS	-	Requested update requires the creation of a new physical resource; hence creating one.
2024-07-17 22:56:42 UTC+0530	Stack1Creating-EC2-Instance	⌚ UPDATE_IN_PROGRESS	-	User Initiated

Create Change Set – Instance Type

- Now this time we will use the change set feature again and this time we will change our instance type.
- Below in the code you can see that we have changed our instance type to t2.small.

```

1  AwSTemplateFormatVersion: 2010-09-09
2  ^ Resources:
3  |  EC2Instance:
4  |  |  Type: 'AWS::EC2::Instance'
5  |  |  Properties:
6  |  |  |  ImageId: ami-0ec0e125bb6c6e8ec
7  |  |  |  InstanceType: t2.small
8  |  |  |  KeyName: CFkeyPair2
9

```

3. Again, from your cloud formation choose an option to create a change set and then choose to replace the existing template, and then upload the new one.
4. After that move to review page and click on Submit.

Create change set for Stack1Creating-EC2-Instance

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.

Use existing template
Proceed with the template you are already using for this stack.

Replace existing template
Replace your existing template with a new template.

Edit in Application Composer
Edit your template in a visual builder.

Specify template
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

Upload a template file

Upload a template file

5. Now we will click on Execute change set.

CloudFormation > Stacks > Stack1Creating-EC2-Instance > Change sets: Stack1Crea-u4y29pxggr-qjsum07i3lr

Stack1Crea-u4y29pxggr-qjsum07i3lr

Overview

Change set ID arn:aws:cloudformation:ap-south-1:878893308172:changeSet/Stack1Crea-u4y29pxggr-qjsum07i3lr/9a106619-6f4c-429c-bba0-c52081c99dbe	Status CREATE_COMPLETE
Description -	Status reason -
Created time 2024-07-17 23:02:55 UTC+0530	Execution status AVAILABLE

Changes | Input | Template | JSON changes | Hook invocations

Changes (1)

Preview how proposed changes to a stack will impact running resources. Click on "View details" to preview the impact on property values for a resource.

Action	Logical ID	Resource type	Replacement	Property-level changes	Policy action	Physical ID	Module
Modify	EC2Instance	AWS::EC2::Instance	Conditional	View details	-	j-0e4aff42df9d47bbe	-

6. In the events you can see that our stack has been updated.

Events (36)

Search events

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-17 23:05:52 UTC+0530	Stack1Creating-EC2-Instance	UPDATE_COMPLETE	-	-
2024-07-17 23:05:51 UTC+0530	Stack1Creating-EC2-Instance	UPDATE_COMPLETE_CLEAN_UP_IN_PROGRESS	-	-
2024-07-17 23:05:51 UTC+0530	EC2Instance	UPDATE_COMPLETE	-	-

7. Also in the EC2 you can see that we have an instance of t2.small type creating.

Instances (3) Info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Available
	i-09818e49f322f330b	Terminated	t2.micro	-	View alarms +	ap-south-1
	i-0e4aff42df9d47bbe	Running	t2.small	Initializing	View alarms +	ap-south-1
	i-06fb198eb4996eb2b	Terminated	t2.micro	-	View alarms +	ap-south-1

8. So, once you are done you just need to delete your stack and it will terminate your instances.