

# S3 Bucket Using CloudFormation

## Step 1: Create a Template

1.1 Open a new file in a notepad

1.2 Write the following code in the notepad for the S3 bucket template:

**Resources:**

**S3Bucket:**

**Type:** 'AWS::S3::Bucket'

**DeletionPolicy:** Retain

**Properties:**

**BucketName:** docfilename

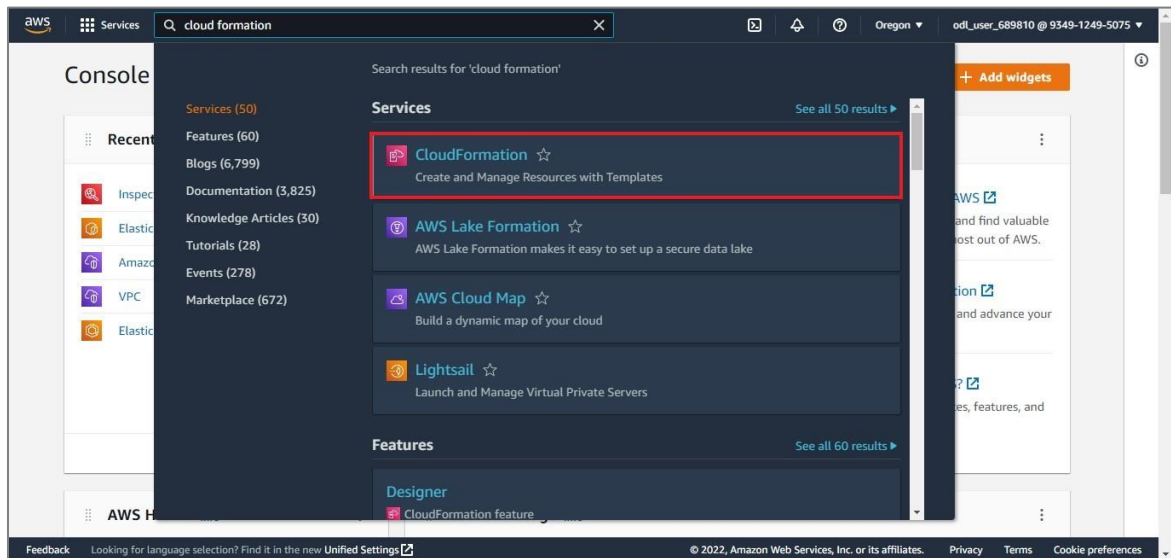
A screenshot of a Notepad window titled "simp.yaml - Notepad". The window contains the following CloudFormation template code:

```
Resources:
  S3Bucket:
    Type: 'AWS::S3::Bucket'
    DeletionPolicy: Retain
    Properties:
      BucketName: docfilename
```

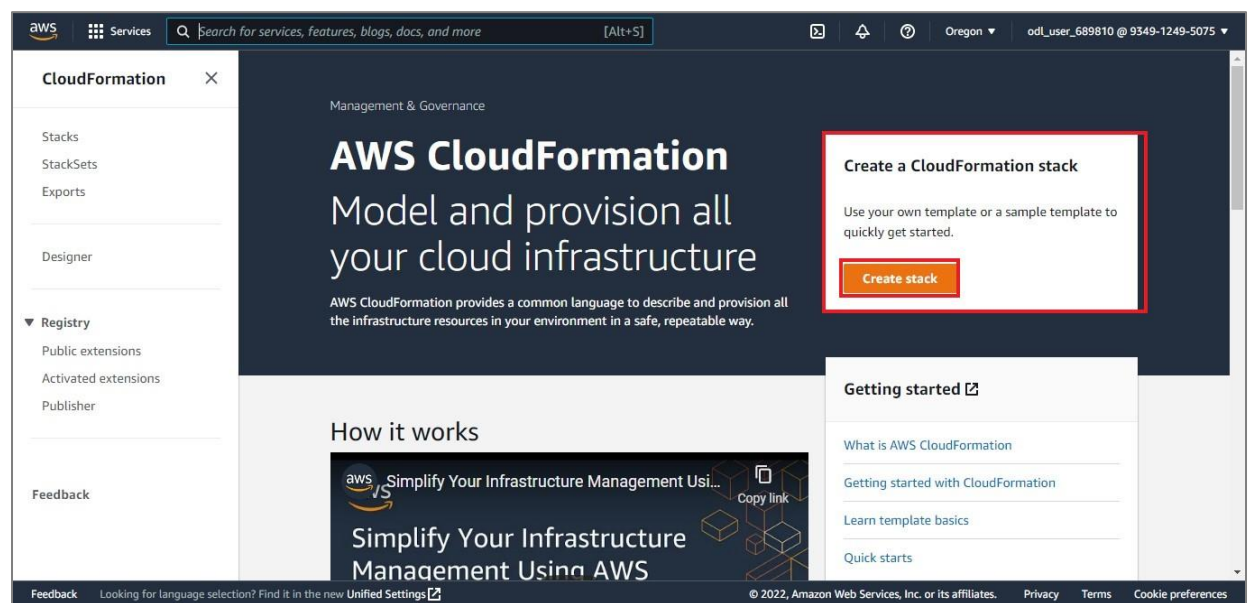
Save the file with a “.yaml” extension on your local system.

## Step 2: Creating an S3 bucket stack using CloudFormation

2.1 Go to the AWS Management Console, and search for **Cloud Formation**:



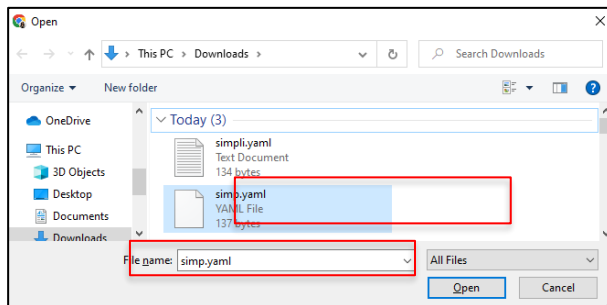
2.2 In the Cloud Formation Management Console, click on the **Create Stack**:



2.3 In **Create stack** console do the following:

- Choose **Upload a Template** in **Specify template** section.
- Click on **Choose file** and upload the template created in step 1, then click on **Next**:

The screenshot shows the AWS CloudFormation console's 'Create stack' wizard. The left sidebar indicates the current step is 'Step 1: Specify template'. The main content area is titled 'Create stack' and contains two sections: 'Prerequisite - Prepare template' and 'Specify template'. In the 'Specify template' section, the 'Template source' is set to 'Upload a template file', which is highlighted with a red box. Below this, the 'Upload a template file' button is also highlighted with a red box. The 'Choose file' button is highlighted with a red box, and the text 'No file chosen' is displayed next to it. The 'S3 URL' field is empty, and the 'View in Designer' button is visible at the bottom right. The 'Next' button is highlighted in orange at the bottom right of the wizard.



Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

### 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.


☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

### 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**  
Choose file  **simp.yaml**  
JSON or YAML formatted file

S3 URL: <https://s3-external-1.amazonaws.com/cf-templates-1astxgz28h8ca-us-east-1/20222341hz-simp.yaml> [View in Designer](#)

[Cancel](#) [Next](#)

2.4 Enter an arbitrary name for the stack and click on **Next**:

aws Services  [Alt+S]

CloudFormation > Stacks > Create stack

Step 1  
Specify template

Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review

### Specify stack details

**Stack name**

Stack name  
**Simpli-first-bucket**  
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

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

**No parameters**  
There are no parameters defined in your template

[Cancel](#) [Previous](#) [Next](#)

In the **Configure stack options**, keep all settings as default and click on **Next**

2.5 Open the **Duplicate** tab, in the dashboard search and select the **IAM**

2.6 Select the **Roles** and click on the Create role button

**Identity and Access Management (IAM)**

Unable to load search  
Dashboard

Access management  
User groups  
Users  
**Roles**

**IAM > Roles**

**Roles (19)** [Info](#) [Refresh](#) [Delete](#) [Create role](#)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-linked Role)	158 days ago

2.7 In the **Select trusted entity** section, specify the following values:

- Trusted entity type: AWS service

- Use case: CloudFormation

Step 2  
Add permissions

Step 3  
Name, review, and create

### Trusted entity type

☒ **AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

### Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

#### Common use cases

☐ **EC2**  
Allows EC2 instances to call AWS services on your behalf.

☐ **Lambda**  
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

CloudFormation

☒ **CloudFormation**  
Allows CloudFormation to create and manage AWS stacks and resources on your behalf.

Cancel Next

2.8 Click on the **Next** button

2.9 In the **Permissions policies**, search and select **AmazonS3FullAccess**

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

### Add permissions

**Permissions policies** (Selected 1/762)  
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter 1 match

"S3fullaccess" X Clear filters

<input checked="" type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.

► **Set permissions boundary - optional**  
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous **Next**

2.10 Enter the **role name** and click on the **Create role** button

2.11 Go back to the **Stack**, in **Configure stack options** page, select the **IAM role**

**Configure stack options**

**Tags**  
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Key Value Remove

Add tag

**Permissions**  
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

**IAM role - optional**  
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

iamRoleName role1 Remove

**Stack failure options**

2.12 In **Stack failure options**, select **Preserve successfully provisioned resources** and then click on the Next button

**Permissions**  
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

**IAM role - optional**  
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

iamRoleName arn:aws:iam::205204165784:role/role1 Remove

**Stack failure options**

**Behavior on provisioning failure**  
Specify the roll back behavior for a stack failure. [Learn more](#)

☐ Roll back all stack resources  
Roll back the stack to the last known stable state.

☒ **Preserve successfully provisioned resources**  
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

2.13 Review all the settings under the **Review** tab, click on **Create Stack**:

CloudFormation > Stacks > Create stack

**Review Simpli-first-bucket**

Step 1: Specify template Edit

**Template**

Template URL  
https://s3.ap-south-1.amazonaws.com/cf-templates-19h19ml41e133-ap-south-1/2022192A73-simpli-53.yaml

Stack description  
-

Estimate cost [\[Link\]](#)

<b>Permissions</b>	
IAM role name role1	IAM role ARN arn:aws:iam::205204165784:role/role1
<b>Stack failure options</b>	
Rollback on failure Disabled	

The newly created S3 bucket will be displayed in the list:

CloudFormation > Stacks > Simpli-first-bucket

Stacks (1)

Filter by stack name

Active View nested

Simpli-first-bucket  
2022-07-11 09:54:19 UTC+05:30  
CREATE\_COMPLETE

**Simpli-first-bucket**

Stack info Events Resources Outputs Parameters Template Change sets

Events (5)

Search events

Timestamp	Logical ID	Status	Status reason
2022-07-11 09:54:44 UTC+05:30	Simpli-first-bucket	CREATE_COMPLETE	-
2022-07-11 09:54:43 UTC+05:30	S3Bucket	CREATE_COMPLETE	-
2022-07-11 09:54:23 UTC+05:30	S3Bucket	CREATE_IN_PROGRESS	Resource creation initiated
2022-07-11 09:54:22 UTC+05:30	S3Bucket	CREATE_IN_PROGRESS	-
2022-07-11 09:54:19 UTC+05:30	Simpli-first-bucket	CREATE_IN_PROGRESS	User initiated

Hence the S3 bucket is created using CloudFormation.