

Grouping, Ordering and Labelling Parameters

The process involves enhancing a CloudFormation template by adding a Metadata section for grouping, ordering, and customizing parameter labels. Use `AWS::CloudFormation::Interface` to define `ParameterGroups` and specify parameter order and display labels. Update `ParameterLabels` to assign custom display names. Save the edited template, upload it to AWS CloudFormation, and create a stack with a unique name (e.g., `DatabaseStack`). Provide parameter values and submit the stack, observing grouped and labeled parameters in the creation console. Verify the stack's successful creation to confirm the template's functionality. Finally, delete the stack to clean up resources and avoid unnecessary costs.

Activity

1. Find the template files in our GitHub repository under the same name as the heading for easy access and edits. Find and Save the attached template locally, open it in VS Code for edits.
2. Review the template and Add the Metadata section at the top of the template before the Parameters section. Defined `AWS::CloudFormation::Interface` as the metadata key for grouping and ordering parameters. Created a `ParameterGroups` key under metadata and listed groups in the desired order. Added a `Label` attribute to each group with a `Default` value for its display name. Specified group members in a `Parameters` key under each group, listing parameter names. Defined `ParameterLabels` under metadata to customize parameter display labels. Customized labels for specific parameters using key-value pairs and `Default` attributes.

```
2  Description: Sample database stack for the Metadata and Mappings section
3  Metadata:
4    - AWS::CloudFormation::Interface:
5      - ParameterGroup:
6        - Label:
7          - default: Database Instance Settings
8        - Parameters:
9          - DbClass
10         - MultiAZ
11         - AllocatedStorage
12         - MasterUsername
13         - MasterUserPassword
14       - Label:
15         - default: Network Settings
16       - Parameters:
17         - VpcId
18         - DbSubnets
19         - SecurityGroupPorts
20     - ParameterLabels:
21       - DbClass:
22         - default: Database Instance Class
23       - AllocatedStorage:
24         - default: Allocated Storage Size
25     Parameters:
```

3. Change the Image Id, Subnet id and Vpc id according to your region. Save the updated template.
4. Upload the template to CloudFormation and create a stack.

Create stack

Prerequisite - Prepare template

You can also create a template by scanning your existing resources in the [laC generator](#).

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 Infrastructure 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

Sync a template from your Git repository.

Upload a template file

Choose file

database-stack-template.yaml

JSON or YAML formatted file

5. Provide a unique stack name e.g., ‘DataBaseStack’ and fill in the parameters as shown. Click on ‘NEXT’ to continue on the review page and submit.

Provide a stack name

Stack name

DataBaseStack

Stack name must be 1 to 128 characters, start with a letter, and can contain only letters, numbers, hyphens, and underscores.

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

Database Instance Settings

Database Instance Class

RDS instance class

db.t4g.micro

MultiAZ

Enable Multi-AZ?

false

Allocated Storage Size

Database storage size in GB

8

MasterUsername

Master username for the db instance

MasterUserPassword

Master user password for the db instance

6. Observed grouped, ordered, and custom-labeled parameters on the stack creation console.

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

Parameters (8)

Search

< 1 >

Key	Value	Resolved value
AllocatedStorage	8	-
DbClass	db.t4g.micro	-
DbSubnets	subnet-0619479663b4084af,subnet-08c772de46d053876,subnet-0a1bbb2feaa84bb95	-
MasterUsername	****	-
MasterUserPassword	****	-
MultiAZ	false	-
SecurityGroupPorts	80,3306	-
VpcId	vpc-0b0c8badfef0024a4	-

7. Verified the functionality and correctness of the template by successfully creating the stack.
8. Deleted the stack to clean up resources.