

# Implementing Terraform Templates with CoreStack

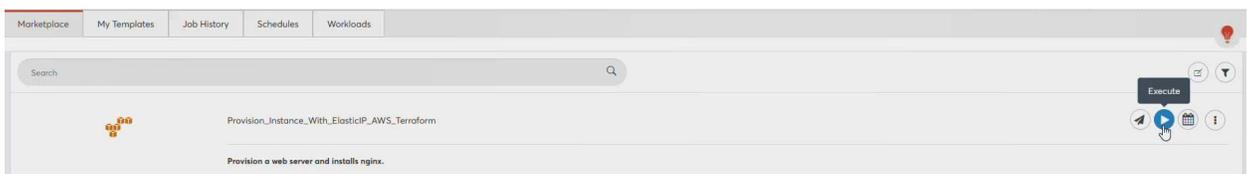
## Requirements

- CoreStack<sup>R</sup> 2.7.1 or later
- Role access: Admin
- Terraform scripts: You can use local file or use the GitHub link of the templates

## Executing Terraform from Templates - Marketplace

---

1. Log into CoreStack, Navigate to Templates
2. Within the Templates section, you can navigate to the Terraform Templates added to the Marketplace.
3. Once you've identified a Terraform template, hover over the  icon available on the right-hand side. CoreStack now provides with option to "Plan and Execute" and "Execute" options.



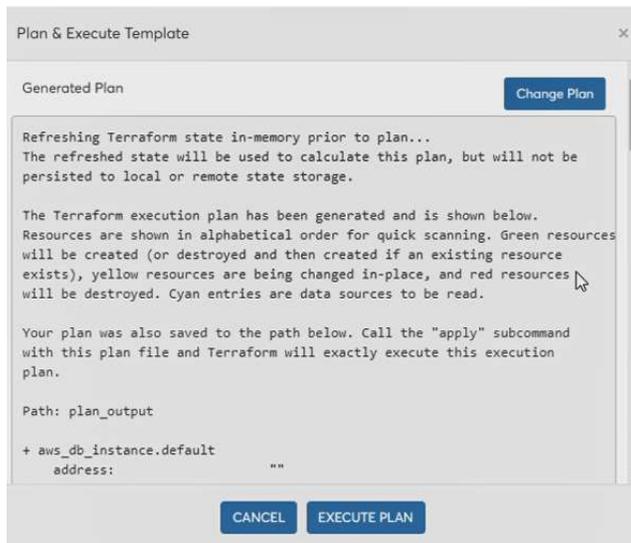
Users can use Plan and Execute option, to enter the details of the template and review the details before executing the template itself.

The User additionally has an option to Change Plan if they would like to edit any details during the preview of the plan.

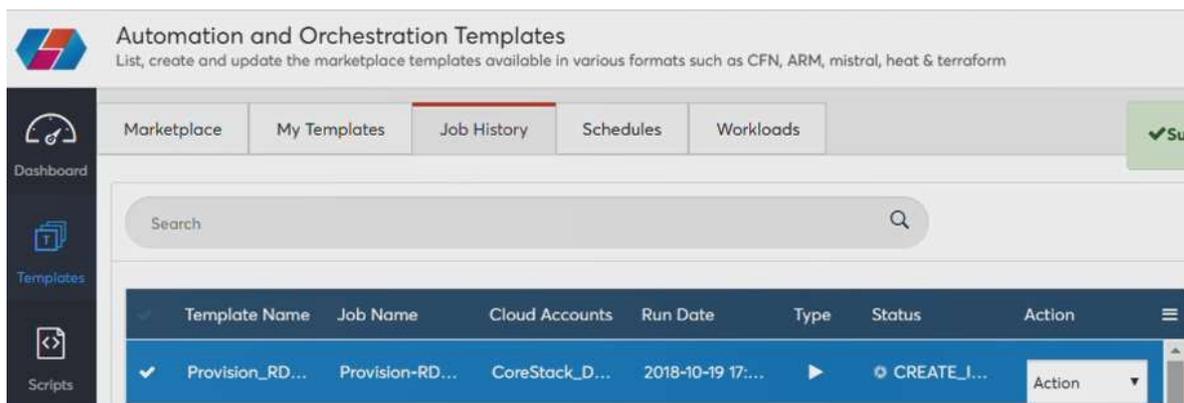
If the user would like to directly execute a Terraform template, they can use the Execute button.

## Plan and Execute a Template

1. Click on Plan and execute
2. In the dialog box, enter details as required. (Refer to the appendix)
3. After all the required information is entered, click on “Generate Plan” button.
4. Plan and Execute dialog box shows the plan. User can scroll to review the complete plan.



5. If the user needs to make any edits, they can click on “Change Plan” button.
6. If the user is satisfied with the plan, they can execute the template by clicking on “Execute Plan” button
7. After clicking on “Execute Plan” user is redirected to the Templates, Job History tab where the template is queued for execution.

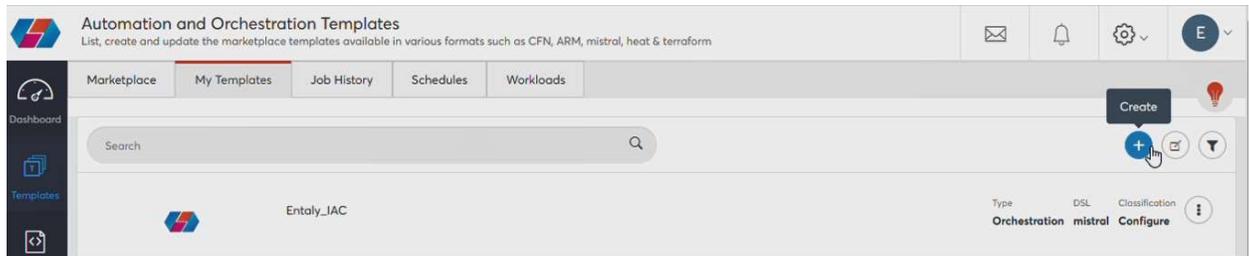


8. Highlighting on the Job will display the Input and Task details on the right-hand side. Once the template is executed, Output section is added with the template execution details.

## Create Terraform Templates

---

1. Log into CoreStack and navigate to Templates, My Templates tab
2. Click on “Create” Template button



3. Enter details of the template. (Refer to Appendix)
4. Once all the details are entered, click on Save.
5. The template will be listed on the top, in My Templates tab

## Execute a Template

---

1. Click on Execute
2. Enter information in the dialog box, with valid parameters.
3. After all the information is entered. Click on Run Now
4. The template will be queued for execution and will be listed on the top of Templates, Job History tab
5. Once the template is executed, user can select the template and review the details on the right-hand side.

# Appendix

## Plan and Execute Template

Plan & Execute Template

**AWS Service Accounts**

Service Account: saba-testing

Regions: ap-south-1

Subnet name1 \*:

Subnet1cidr \*: 10.0.6.0/24

Sg name \*: rds\_sg

Identifier \*: mydb-rds

Engine \*: postgres

**CANCEL** **GENERATE PLAN**

Inputs in Template vary according to the required parameters in template.

Fields	Descriptions
Service Account	Select the Account available to you
Subnet Name 1	Enter a name you would want to use for Subnet 1
Subnet1 CIDR	Enter the CIDR information for Subnet 1
Subnet Name 2	Enter a name you would want to use for Subnet 2
Sg Name	Enter a name for the Security Group
Engine	Select the Execution Engine type like Terraform (Includes CFN, Mistral etc.)
CIDR blocks	Enter the CIDR blocks information
Az 1	This field will have the Availability zone 1 information
Az2	This field will have the Availability zone 2 information
Storage	Enter the Size of the Storage in GB
Security group name	Enter the Security Group name with which the SG will be created in Cloud
Username	Enter the Username that will be used to authenticate
Instance Class	Enter the name of the class
DB Name	Enter the Data base name
Subnetcidr	Enter the Network IP that must be created
Db subnet group name	Enter the Data Base Subnet name

Fields	Descriptions
Vpc id	Enter already existing VPC id in which all the resource will be created
Password	Enter the password for all the instance
Region	Select the Region in which the resource must be created

## Create Template

Fields	Descriptions
Name	Enter the name of the template
Engine Type	Select Terraform
Type	Select type of template
Services	Select the type of account (AWS or Azure)
Classifications	Select the classification the template will list
Scope	
Input Resources	
Template Content	User can either choose 'File' or 'Git' to upload the template. Should the user choose 'File', then they have to navigate to the folder the template is available If the user chooses 'Git', then they need to provide the Git URL from which CoreStack will retrieve the templates

Fields	Descriptions
Yes, I have dependency files	If the user has module files, they can select this option and enter the details for the module files
Template Metadata	Here, user can add the description of the template being created
Operation Level	
Tags	User can add Tags and associated value as part of It's governance needs

### Execute Template

Execute Template
✕

**AzureRM Service Accounts**

Service Account

Resourcegroup

Locations

Vnet cidr \*

Client id \*

Mysql address prefix \*

Portal address prefix \*

Tenant id \*