**Design Document**

1. **NAPI AWS CloudFormation Design**

This document aims at defining the design startegy for the creating,managing ,provisioning and updating collection of AWS resources required for NAPI using AWS CloudFormation.This design strategy takes care of full deployment as well as partial deployments of AWS resources.Any environment specific changes can be accomodated by customizing the respective templates for the environment specific deployments.

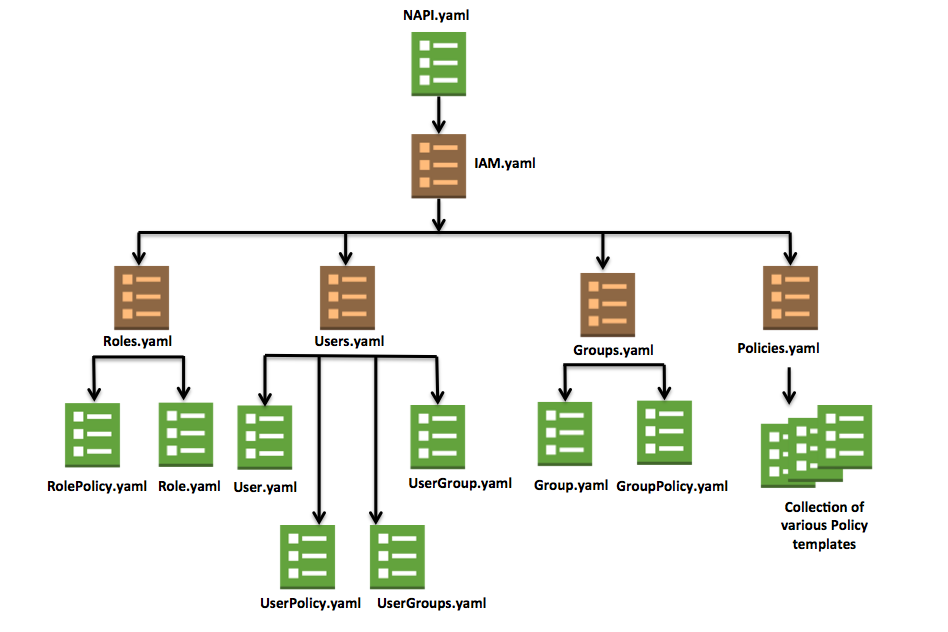
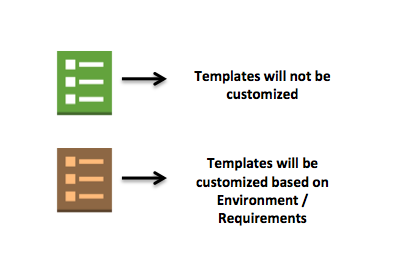
1. **Design Objectives:**

The design aims at following objectives for creating generic templates .

1. To achieve reusability by seperating out common components and creating dedicated templates.
2. To have a modular structure for the AWS CloudFormation templates in order to break down the complex templates into more manageable emplates.
3. To achieve abstraction by creating complex static templates and have a layer of templates that developer can customize.
4. To create a parent-child based hierarchial structure in oder to pass the parameters from one stack to another by using Cross-Stack references
5. **NAPI.yaml**

This master template will consist of individual stack that hold the path to the child template for individual AWS Resources . As this file is generic and will not require any customization ie it will not be allowed for any modification.

Following diagram gives the details about the template ditribution and the hierarchy of templates in the order of implementation.This makes the design modular as the templates are segregated into manageable templates.

****

1. **CloudFormation Template for IAM**
   1. **IAM.yaml**

This master template will consist of individual stack that hold the path to the child template about list of users,roles,groups and policies to be created .This template will invoke all the nested templates and will avoid creation of single template for all the resources.Any IAM resource creation will be handle by respective templates for Users,Roles,Groups and Policies which will be referd to childstacks. This file might need modification if the environ ment requirement of groups/users/roles changes.The list of attributes and properties can be found in the Design excel created for reference.Thus this shows how the design objective of having parent-child hierarchical ditribution is met using IAM.yaml as parent template and users,roles,groups and policies willbe the child templates .

**Design -**

****

1. **CloudFormation Template Design for attching the Policy to the User.**
   1. **Policy**

Separate templates can be defined to create policies.And an export feature is used to export the value of the policy created.This value then can be imported under the ManagedPolicyArns property of Group.yaml/User.yaml/Role.yaml in order to attach the policies to group/user/role.This method is called Cross-stack refernce and this helps to fulfill the degin objective to maintain reusability among stacks.

* 1. **Policies.yaml**

This template will be the master template for the Respective policy templates and will be refered in the TemplateURL of this stack.On deployment it will create IAM managed policies for the groups/users/roles as per the requirement.As parameters are the being passed in this template to customize the resource the deign objective to have absatraction is fulfilled as this emplate will not consist of details for policies.

1. **CloudFormation Template Design for IAM Group .**
   1. **Group.yaml**

This template will allow the developer to create custom group and Groupname can be used a parameter to create a the particular group. As this file is generic and will not require any customization , it will not be allowed for any modification.

* 1. **GroupPolicy.yaml**

This template will allow the developer to create custom groups ,policy to be added and Groupname and Policyname can be used a parameter to create a the particular group.This will utilze the import value of policyname in property ManagedpolicyArns. As this file is generic and will not require any customization , it will not be allowed for any modification.

* 1. **Groups.yaml**

In order to create a new group,Group Template should be invoked each time wih the unique Groupname and Users to be added as parameter .This template will be the master template for the CreateRole Template and will be refered in the TemplateURL of this stack. As parameters are the being passed in this template to customize the resource the deign objective to have abstraction is fulfilled as this emplate will not consist of lower level details for group or policy.

1. **CloudFormation Template Design for IAM User**
   1. **User.yaml**

This template defines the attributes required for creation of a single user, Parameter attribute is used for passing the unique username.Everytime this stack is invoked with a new value for the parameter (Username), new user with Access key and secret access key will be created.As this file is generic and will not require any customization , it will not be allowed for any modification.

* 1. **UserPolicy.yaml**

This template will allow the developer to create user with a policy ,Username and Policyname can be used a parameter to create a the particular group.This will utilze the import value of policyname in property ManagedpolicyArns. As this file is generic and will not require any customization , it will not be allowed for any modification.

* 1. **UserGroup.yaml**

This template will allow the developer to create user to be added to a group , Groupname and Username can be used a parameter to create a the particular group. As this file is generic and will not require any customization ie it will not be allowed for any modification.

* 1. **UserGroups.yaml**

This template will allow the developer to create user and add multiple groups to user,Groupnames and Username can be used a parameter to create a the particular user with list of groups. As this file is generic and will not require any customization , it will not be allowed for any modification.

* 1. **Users.yaml**

In order to create a new user,User.yaml or new user with policy UserPolicy.Yaml or add user to exisiting Group UserGroup.yaml should be invoked each time wih the unique parameters .This template will be the master template for the User Template and will be referd in the TemplateURL of this stack.

1. **CloudFormation Template Design for IAM Role .**
   1. **Role.yaml**

This template will allow the developer to create custom roles Service and rolename is used a parameter to create to the particular role.Here the service indicates the service which will assume the STS Role and Rolename is a custom name for the role. As this file is generic and will not require any customization , it will not be allowed for any modification.

* 1. **RolePolicy.yaml**

This template will allow the developer to create custom Role with a policy and Rolename and Policyname can be used a parameter to create a the particular group.This will utilze the import value of policyname in property ManagedpolicyArns. As this file is generic and will not require any customization , it will not be allowed for any modification.

* 1. **Roles.yaml**

In order to create a new role,Role.yaml or to attach a policy RolePolicy.yaml Template should be invoked each time wih the unique parameters .This template will be the master template for the CreateRole Template and will be referd in the TemplateURL of this stack.

1. **Design Use Case**
   1. **Create a User**

Following are the steps to be followed for creation of a new user:

Step 1: The developer opens the Users.yaml template.

Step 2:Create a new a Resource with a unique name under the “Resources” section with the Type: "AWS::CloudFormation::Stack"

Step 3: Enter the path to template the User.yaml as “TemplateURL” in the “Properties”.

Step 4:Pass the value to the “Parametes” required for the User.yaml , which is the name of the user to be created ie Username.

The snippet below gives an example of the resource to be created with the Username as User1.

UserReource:

Type : "AWS::CloudFormation::Stack"

Properties:

TemplateURL: User/User.yaml

Parameters:

Username: User1