**AWS AppStream 2.0 Build Guide**

Table of Contents

[Overview 3](#_Toc536705463)

[Deploy AppStream 2.0 Environment 4](#_Toc536705464)

[Deploy a standard landing zone for AppStream 2.0 service 4](#_Toc536705465)

[Deploy network.json Cloudformation template 5](#_Toc536705466)

[Deploy 0-createbucketloggingbucket Cloudformation template 7](#_Toc536705467)

[Deploy 1-awsconfigservice Cloudformation template 8](#_Toc536705468)

[Deploy 2-cloudtrailwatcher.json Cloudformation template 9](#_Toc536705469)

[Deploy 2b-multiacc-cloudtrailsecondtrail.json Cloudformation template 11](#_Toc536705470)

[Deploy 3-configrules.json Cloudformation template 12](#_Toc536705471)

[Deploy 4-iamgroupspolicies.json Cloudformation template 14](#_Toc536705472)

[Deploy AppStream 2.0 Service 15](#_Toc536705473)

[Deploy 0-directory.json Cloudformation template 16](#_Toc536705474)

[Deploy 1-ImageBuilder.json Cloudformation template 18](#_Toc536705475)

[Deploy 2-Stack.json Cloudformation template 19](#_Toc536705476)

[Deploy 3-Fleet.json Cloudformation template 20](#_Toc536705477)

[Okta and AppStream 2.0 Integration 22](#_Toc536705478)

[Generate metadata file for the AppStream 2.0 stack 22](#_Toc536705479)

[Create SAML Provider 23](#_Toc536705480)

[Create Policy and Role 23](#_Toc536705481)

[AppStream 2.0 Image Builder 27](#_Toc536705482)

# Overview

This document provides the instructions for deploying AppStream 2.0 service with Okta SAML integration (Okta installation is not included as part of this build guide).

The AppStream 2.0 service deployment is divided into two parts:

* Deploy a landing zone for AppStream 2.0 service.
* Deploy AppStream 2.0 service

# Deploy AppStream 2.0 Environment

## Deploy a standard landing zone for AppStream 2.0 service

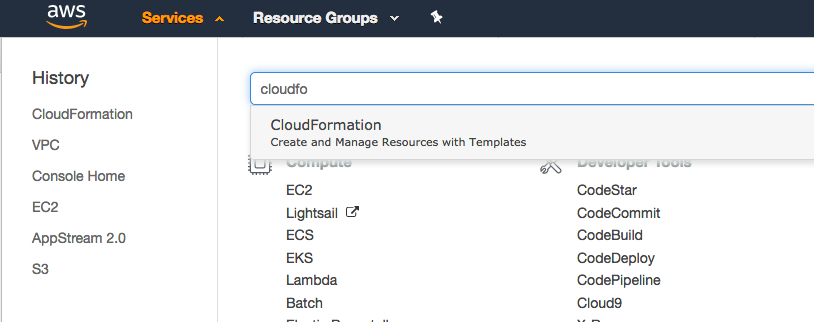
This deployment consists of 7 scripts:

|  |  |  |
| --- | --- | --- |
| Name of the Script | Description | Deployment order |
| Network.json | Deploy VPC, subnets, NACL rules, default security groups, route table, S3 endpoint and DHCP sets. | 1 |
| 0-createbucketloggingbucket | Create an S3 bucket for logging S3 access logs. | 2 |
| 1-awsconfigservice.json | Provision AWS Config Service to log to a dedicated bucket at scheduled intervals. The S3 bucket is created as part of the template | 3 |
| 2-cloudtrailwatcher.json | This template will configure CloudTrail to log all AWS API calls to CloudWatch Logs and an S3 Bucket with optional KMS encryption. CloudWatch Logs has a configurable retention policy and a predefined set of Log Filters which will generate alarms for these events to an SNS Topic which will then send emails to a nominated email address. The S3 Bucket has a configurable retention policy, after which logs will be moved into Glacier. | 4 |
| 2b-multiacc-cloudtrailsecondtrail.json\* | This template will configure a second CloudTrail trail to log AWS API calls to an S3 bucket outside this account, eg for entralized security logs. The S3 Bucket has a configurable retention policy, after which logs will be moved into infrequently accessed. | 5 |
| 3-configrules.json | AWS Config rules to verify AWS best practices are being followed. | 6 |
| 4-iamgroupspolicies.json | Creates default user groups and policies. | 7 |

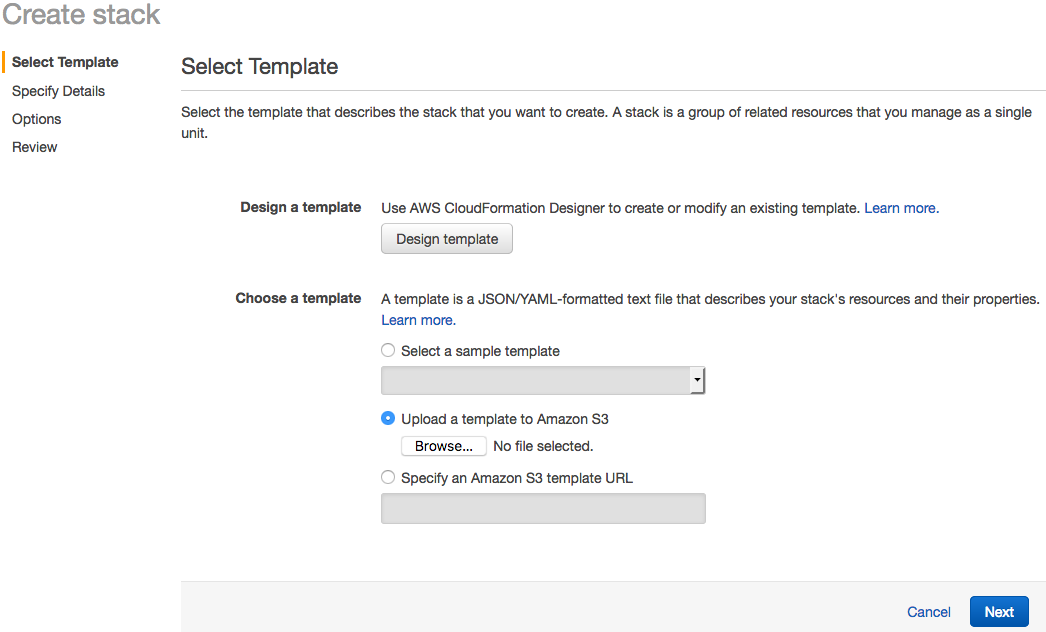
\*This script requires a dedicated security logging aws account. Skip this script if there isn’t a security logging account available in the environment.

### Deploy network.json Cloudformation template

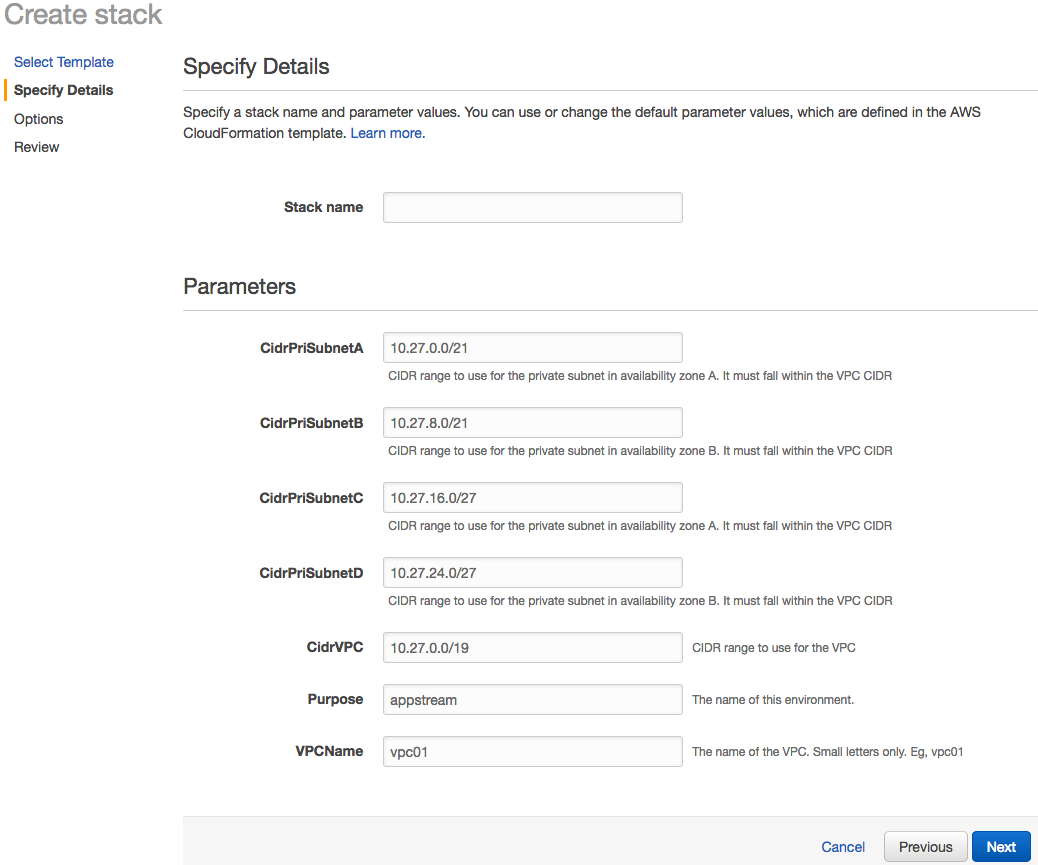
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



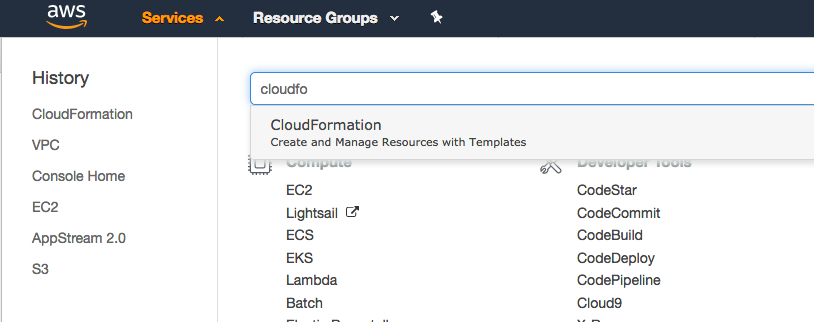
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



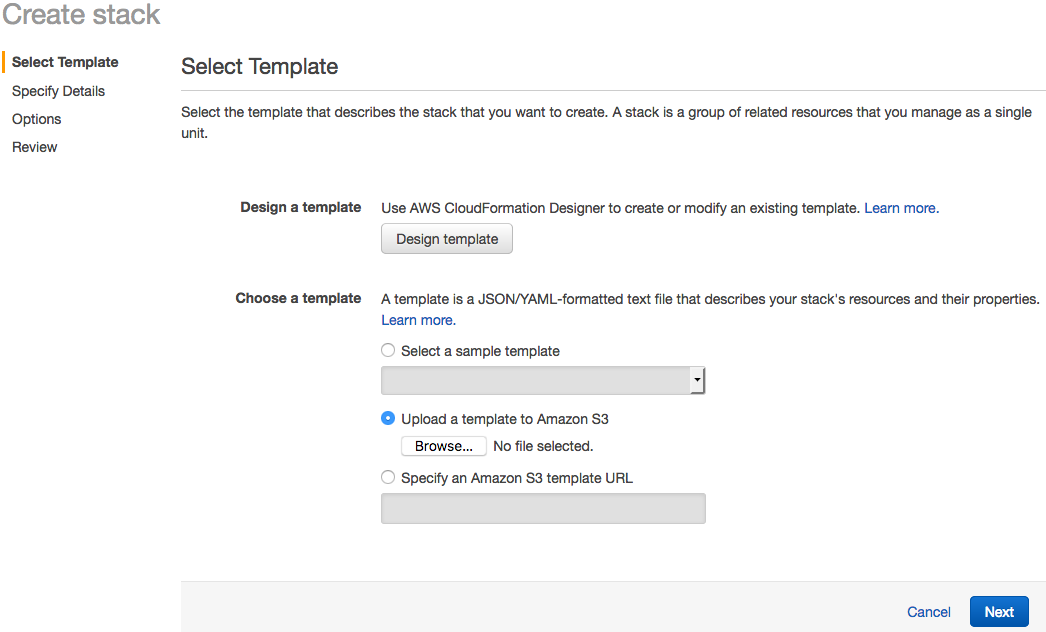
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 0-createbucketloggingbucket Cloudformation template

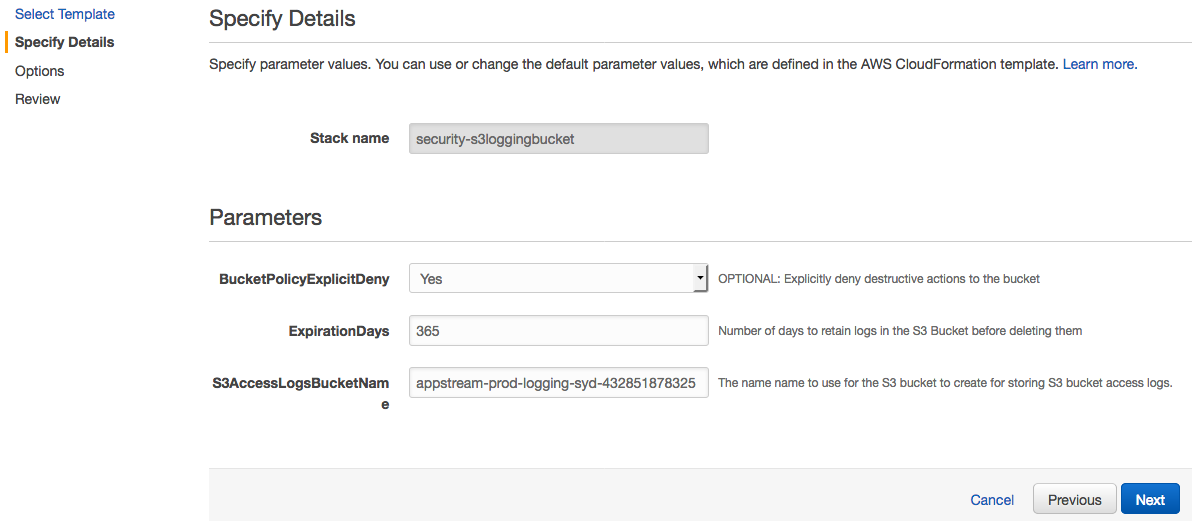
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



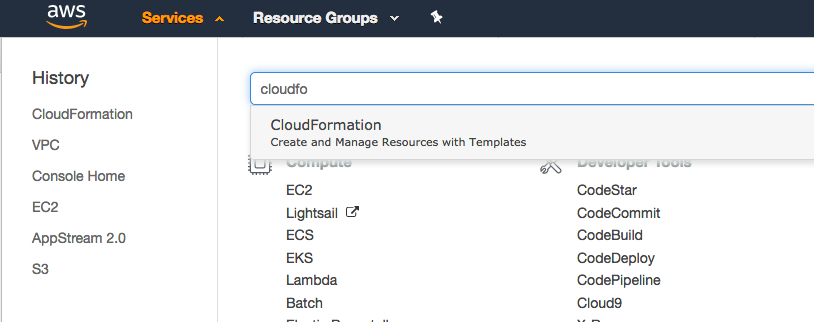
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



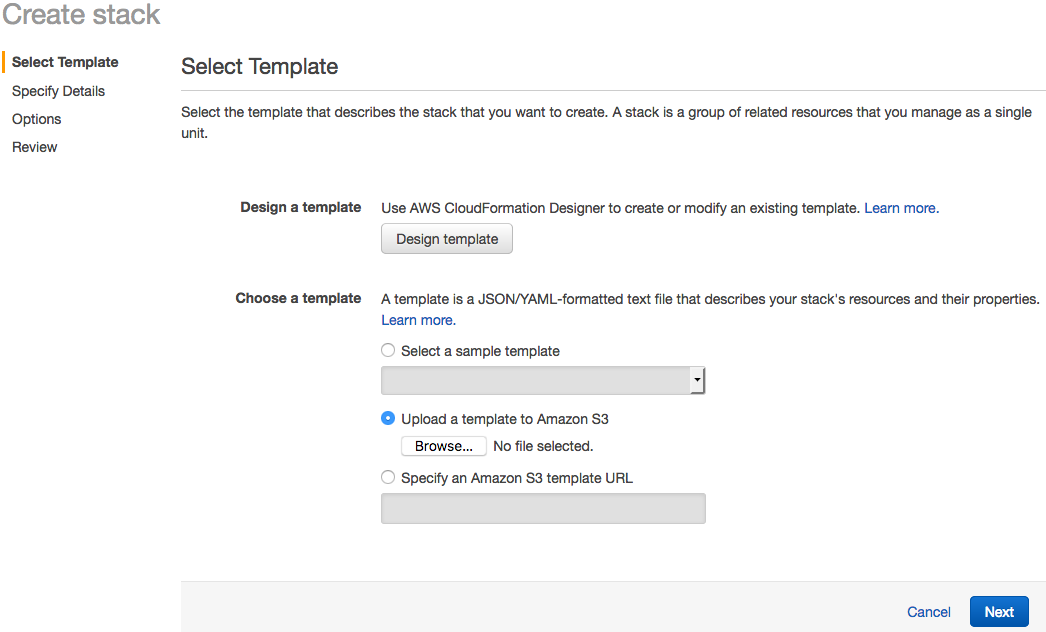
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 1-awsconfigservice Cloudformation template

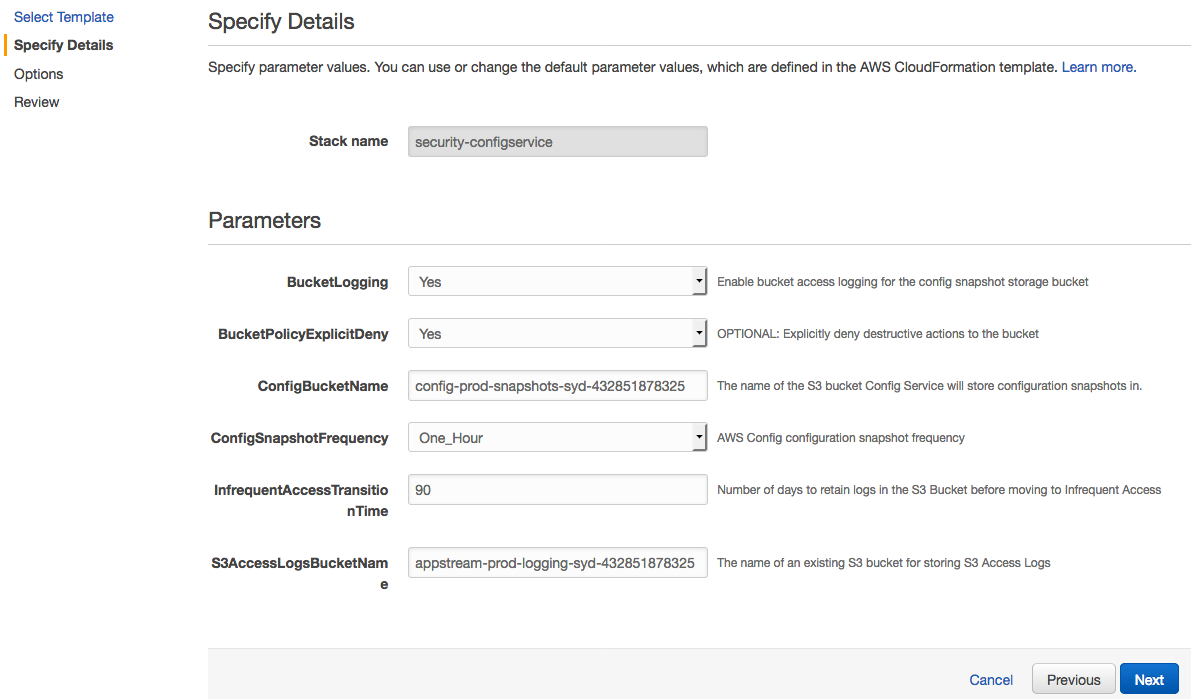
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



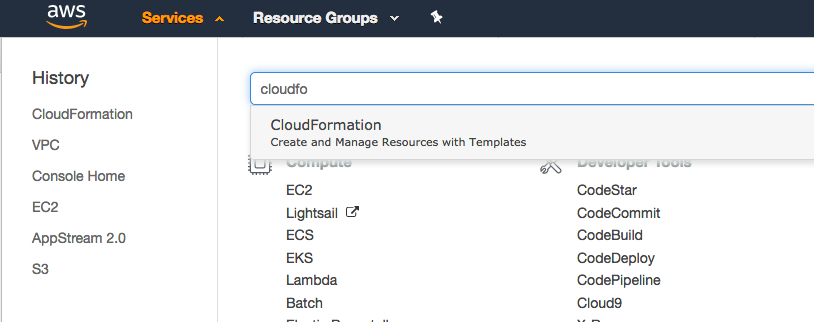
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



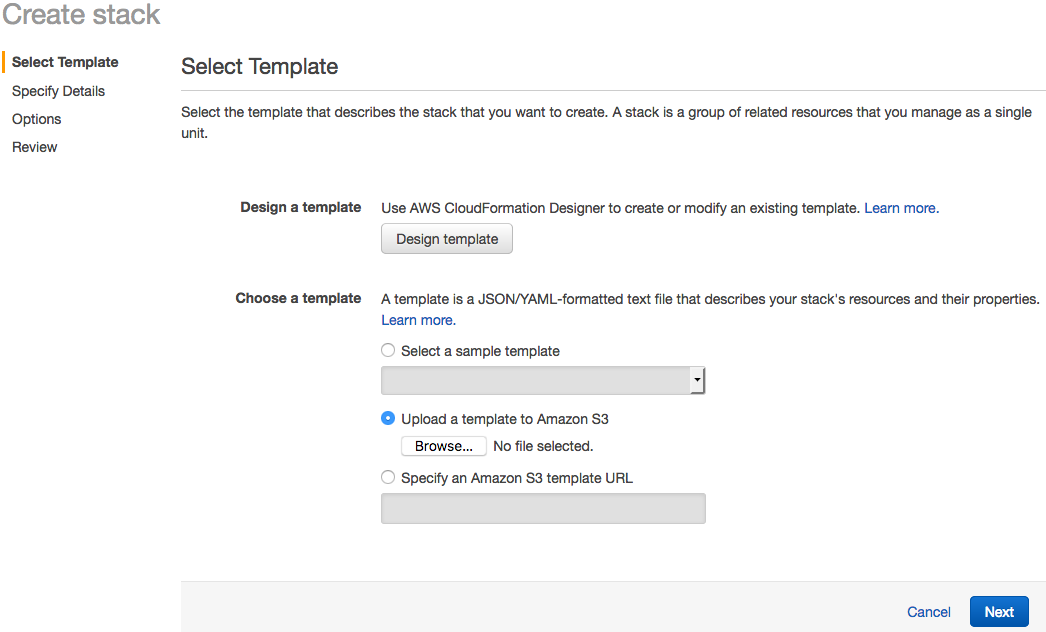
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 2-cloudtrailwatcher.json Cloudformation template

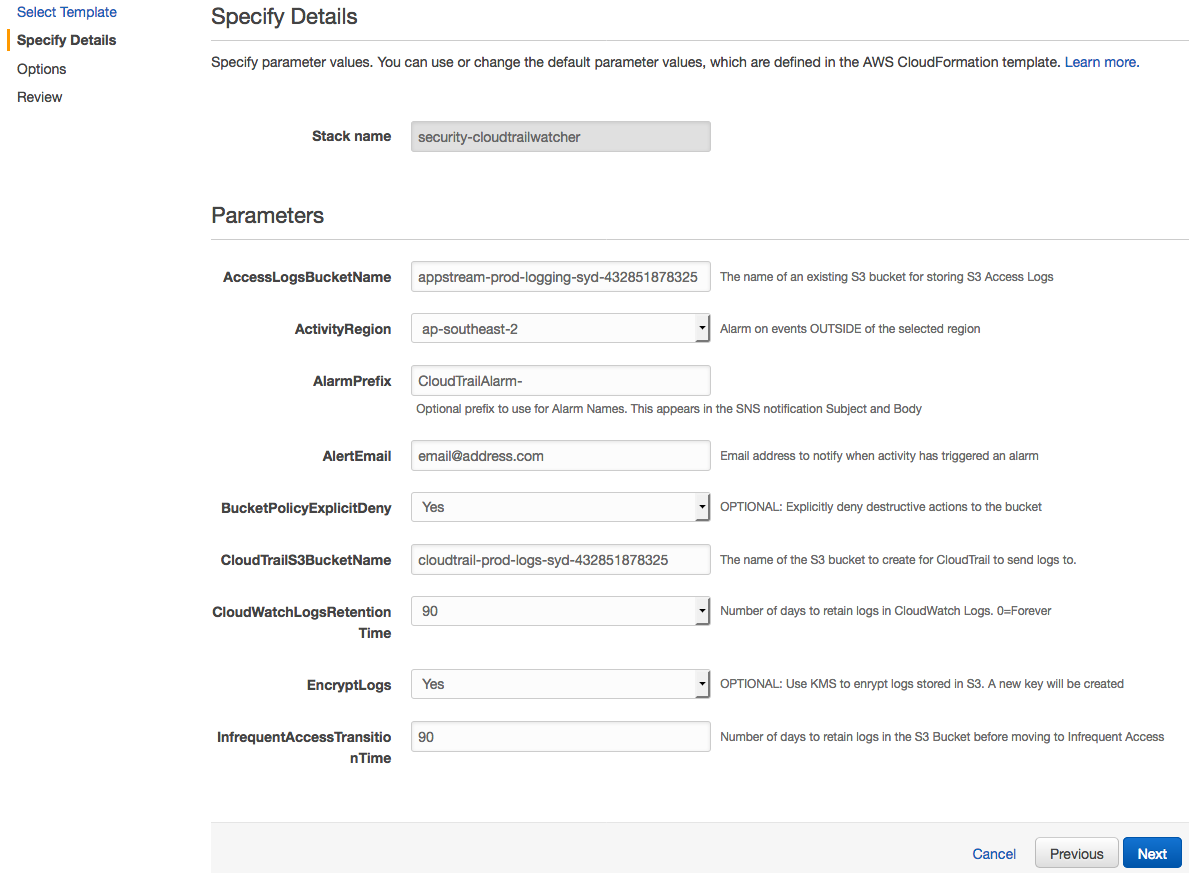
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



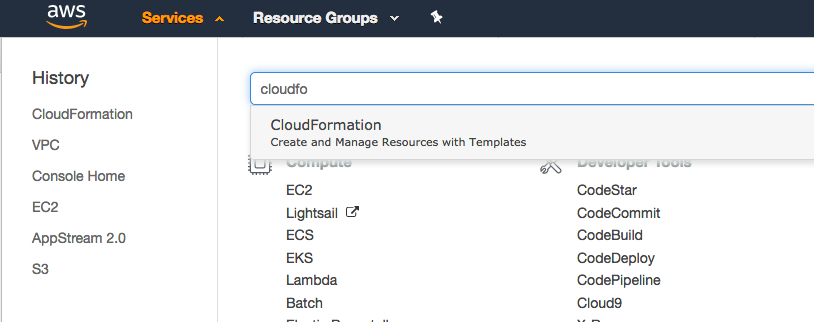
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



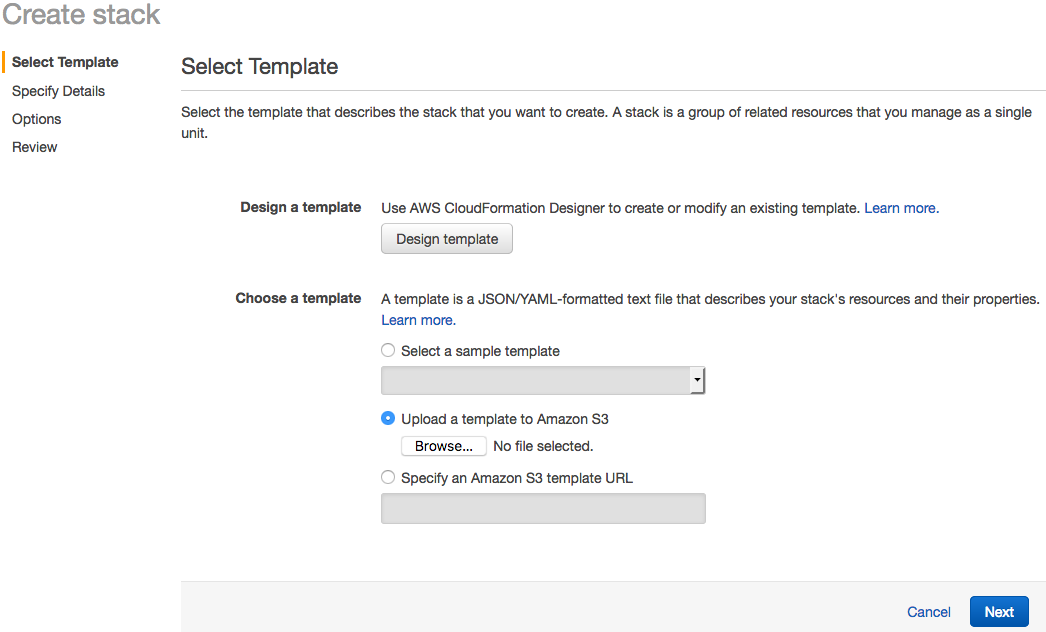
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 2b-multiacc-cloudtrailsecondtrail.json Cloudformation template

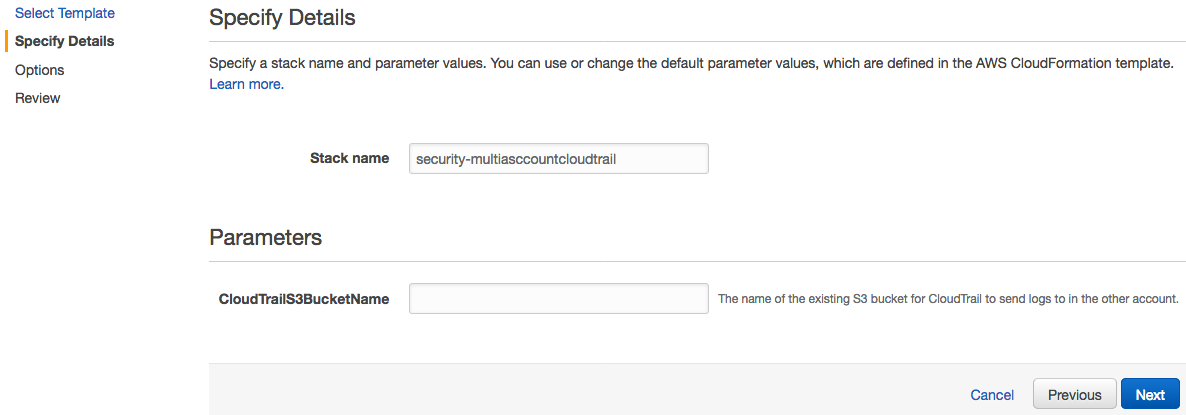
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



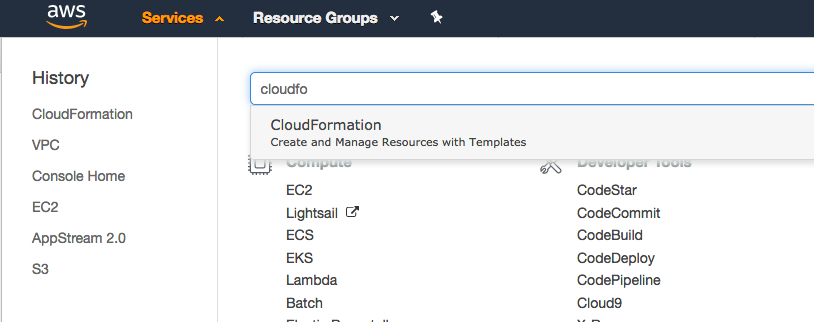
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



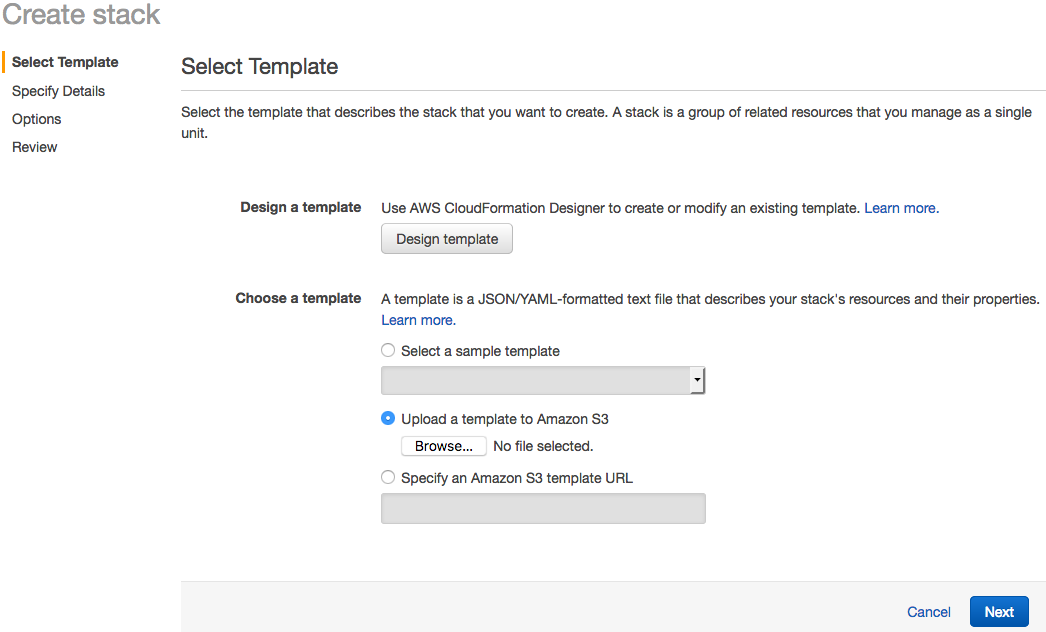
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 3-configrules.json Cloudformation template

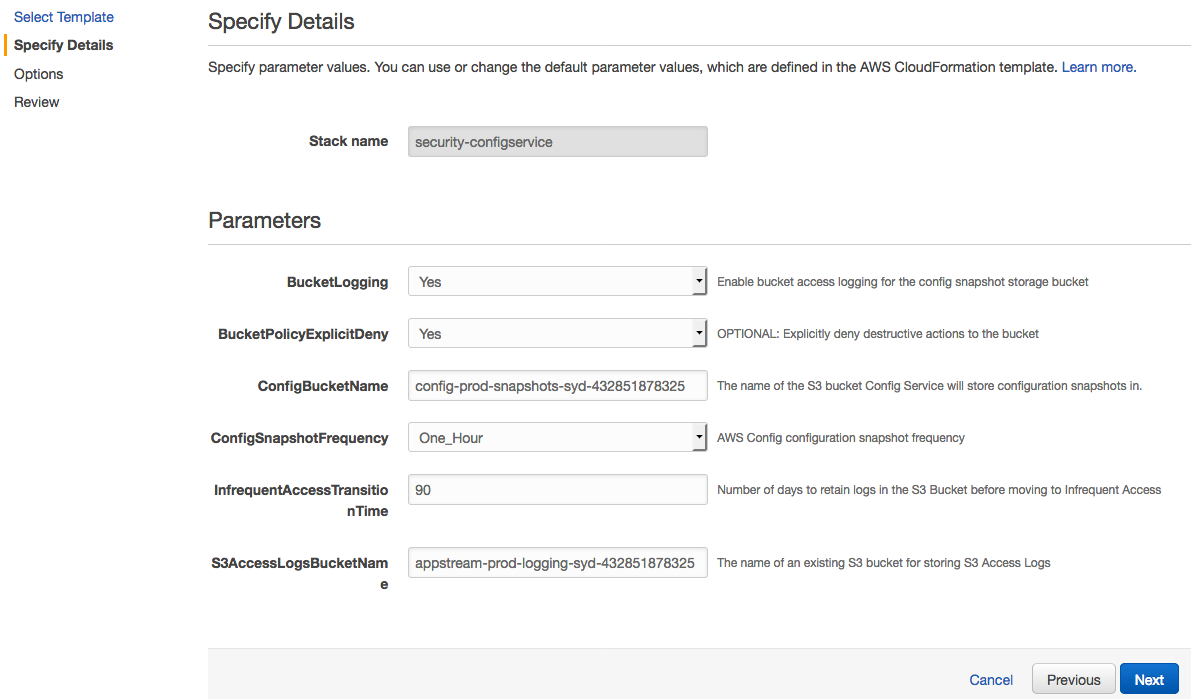
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



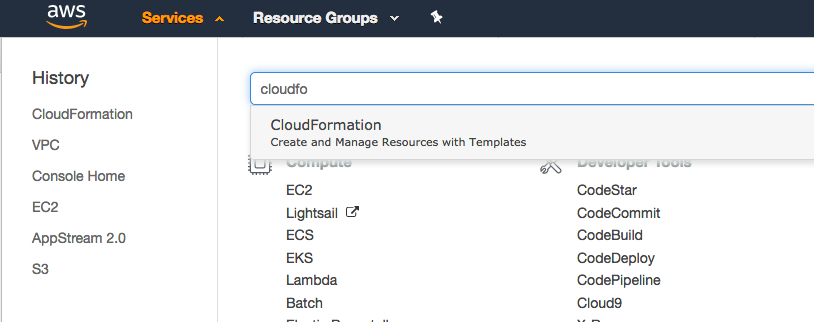
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



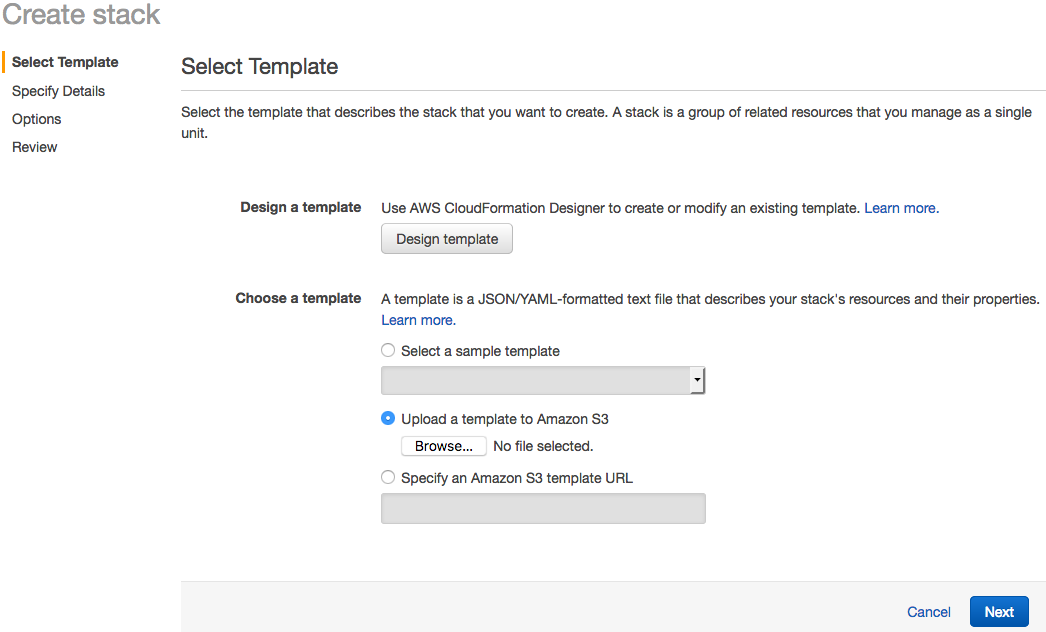
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 4-iamgroupspolicies.json Cloudformation template

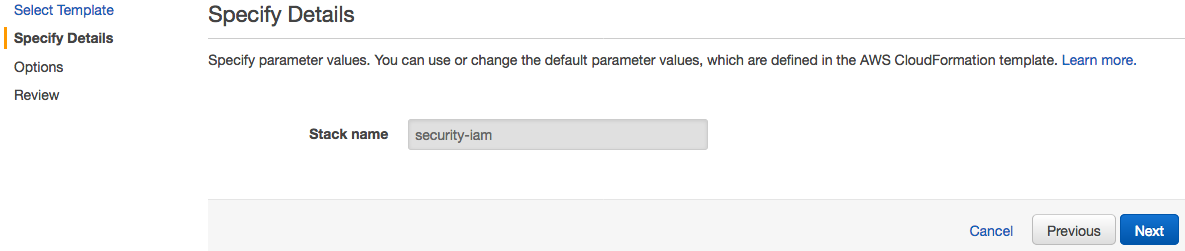
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



1. Click Next.
2. Tick the checkbox at the bottom of the page. Click Next.
3. Click Create.

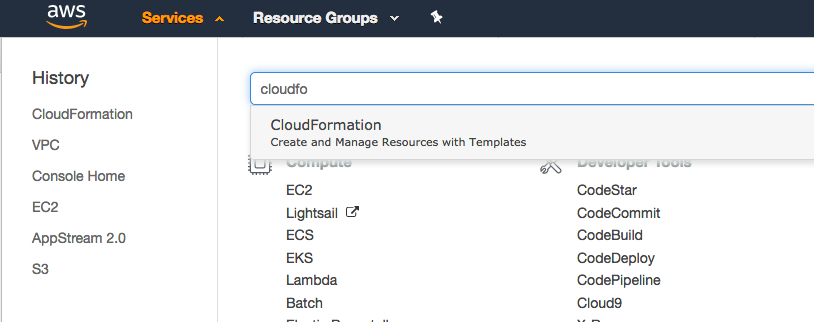
## Deploy AppStream 2.0 Service

This deployment consists of 4 scripts:

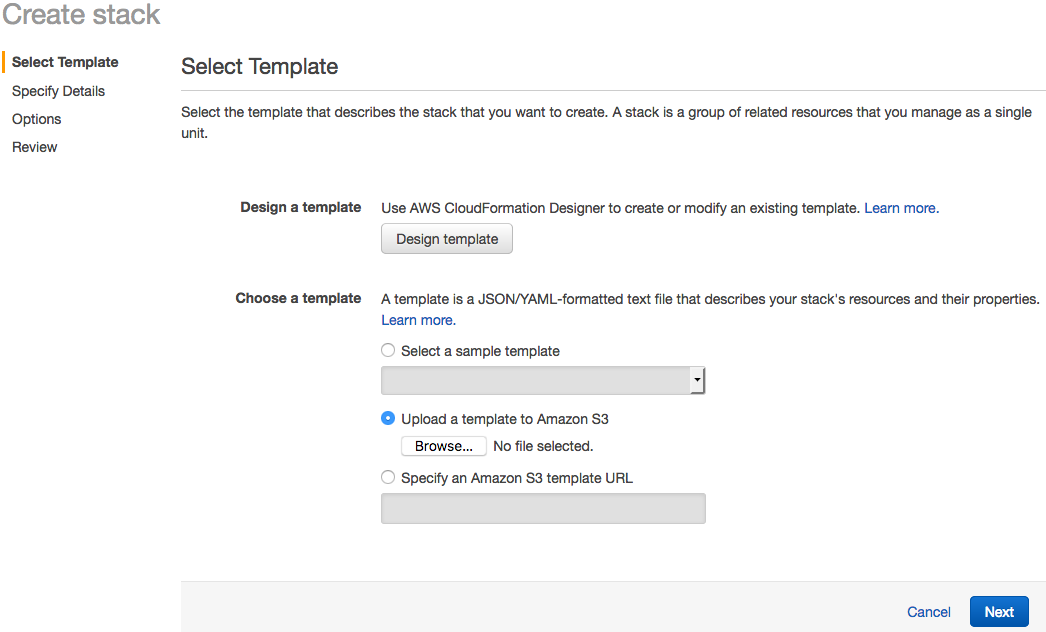
|  |  |  |
| --- | --- | --- |
| Name of the Script | Description | Deployment order |
| 0-directory.json | Create AppStream Directory Configuration. | 1 |
| 1-ImageBuilder.json | Create an AppStream Image Builder. | 2 |
| 2-Stack.json | Create an AppStream stack. | 3 |
| 3-Fleet.json | Create an AppStream fleet. | 4 |

### Deploy 0-directory.json Cloudformation template

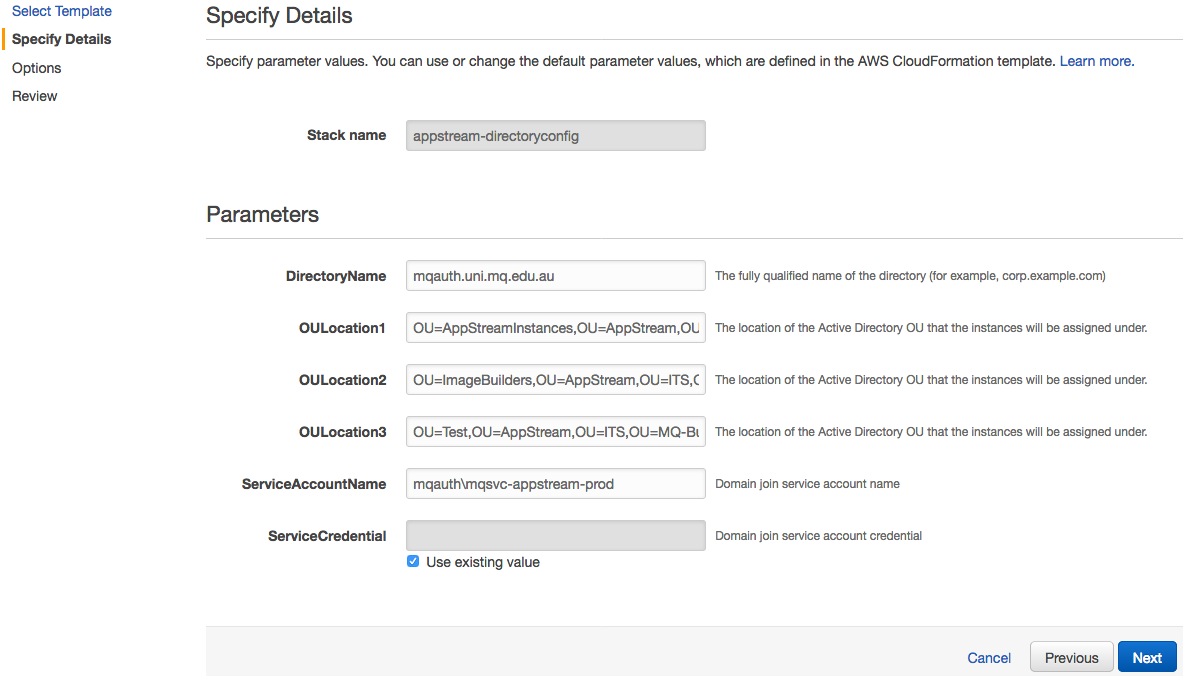
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



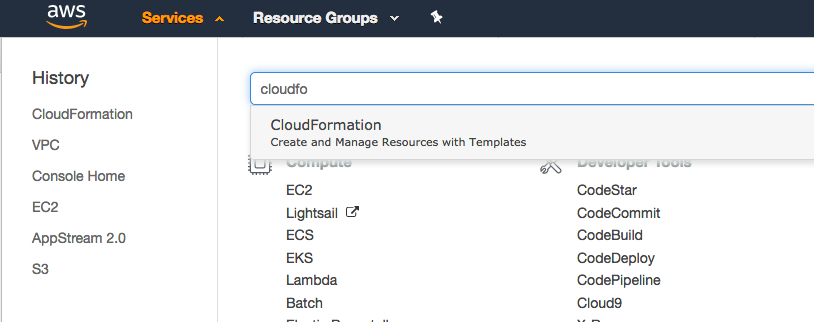
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



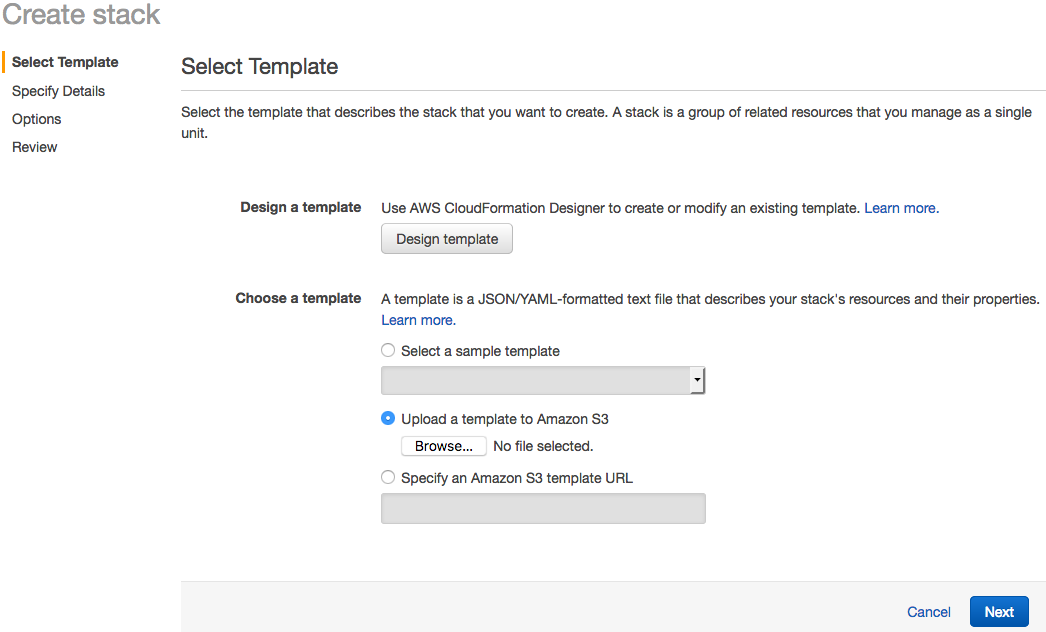
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 1-ImageBuilder.json Cloudformation template

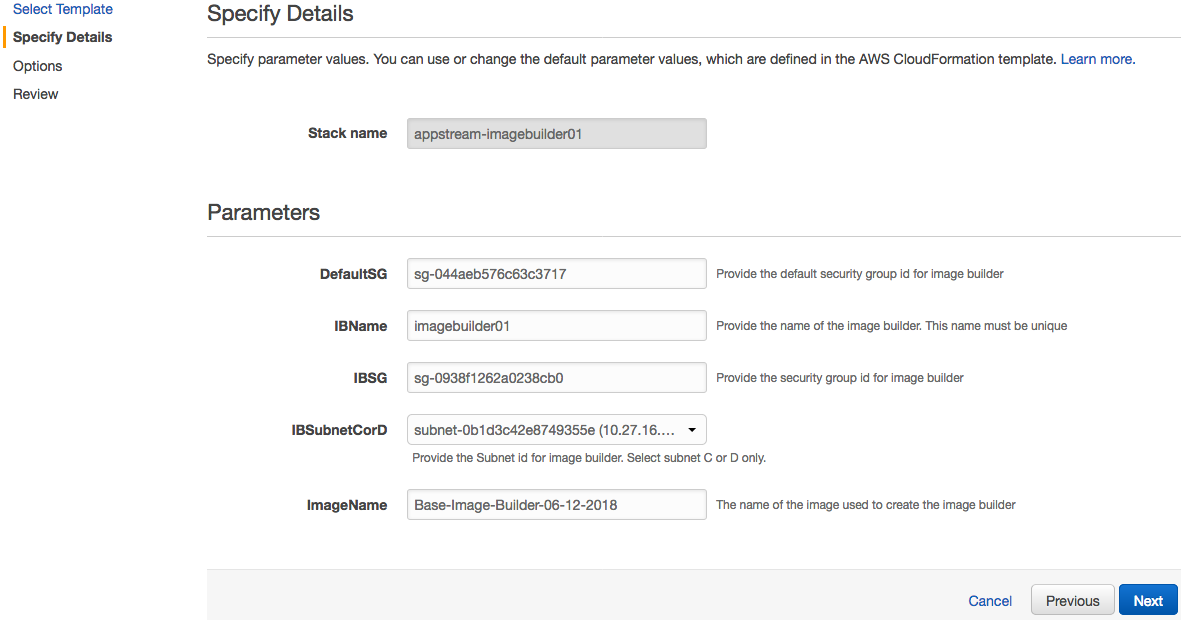
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



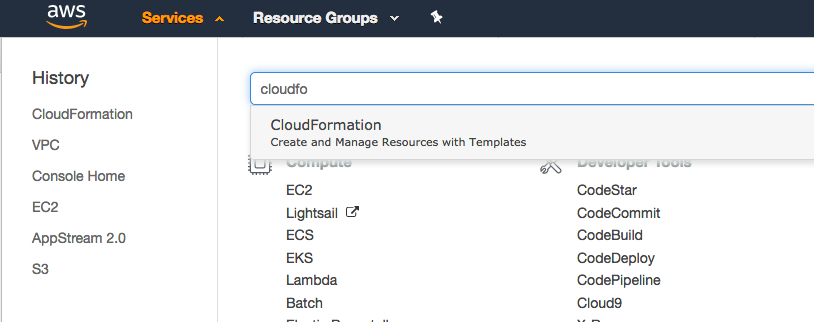
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



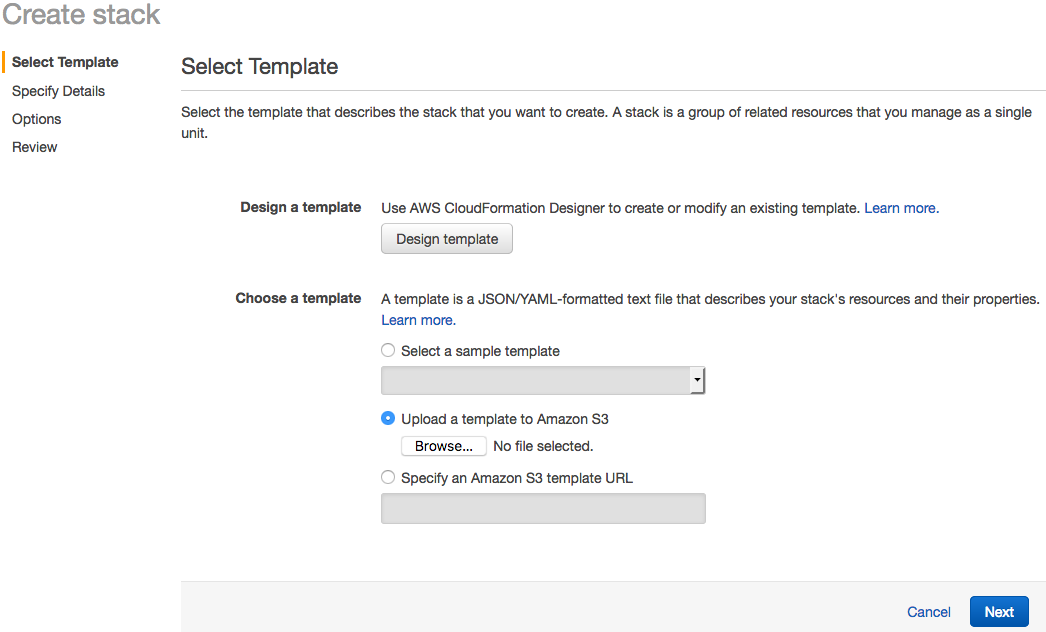
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 2-Stack.json Cloudformation template

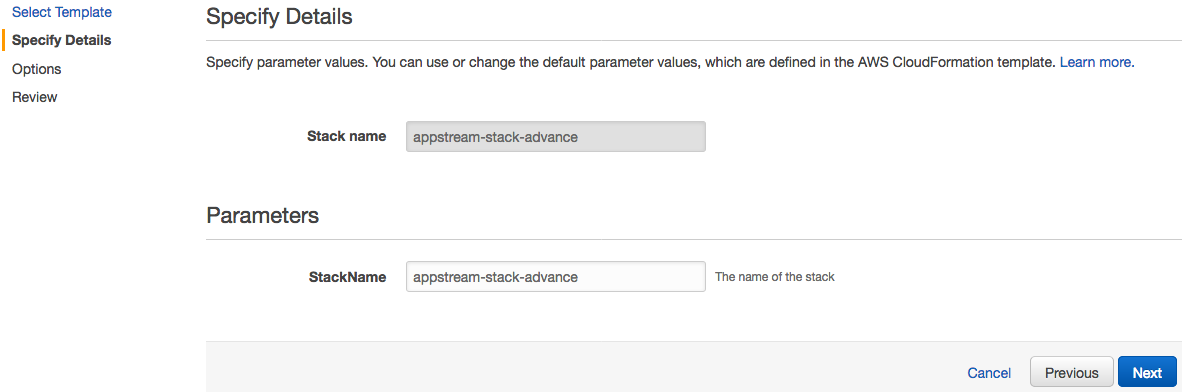
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



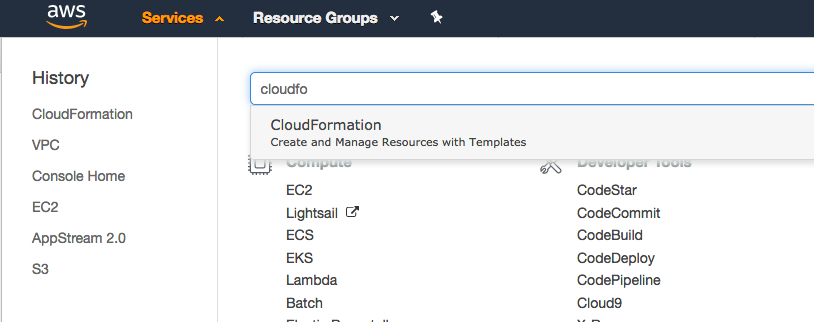
1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



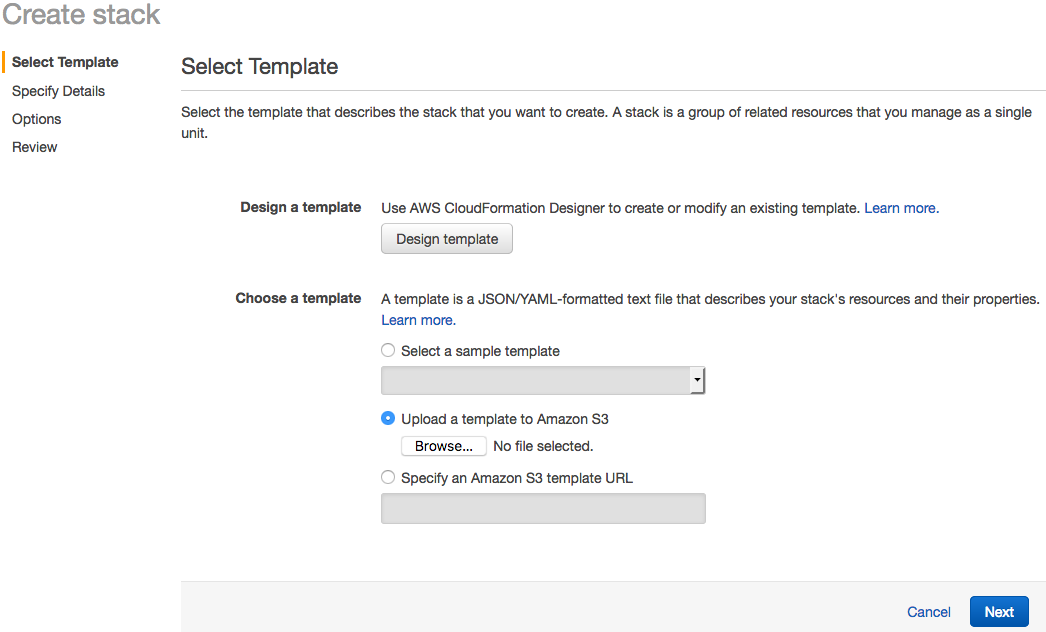
1. Click Next.
2. Click Next.
3. Click Create. Ensure the stack is deployed successfully before the next template deployment.

### Deploy 3-Fleet.json Cloudformation template

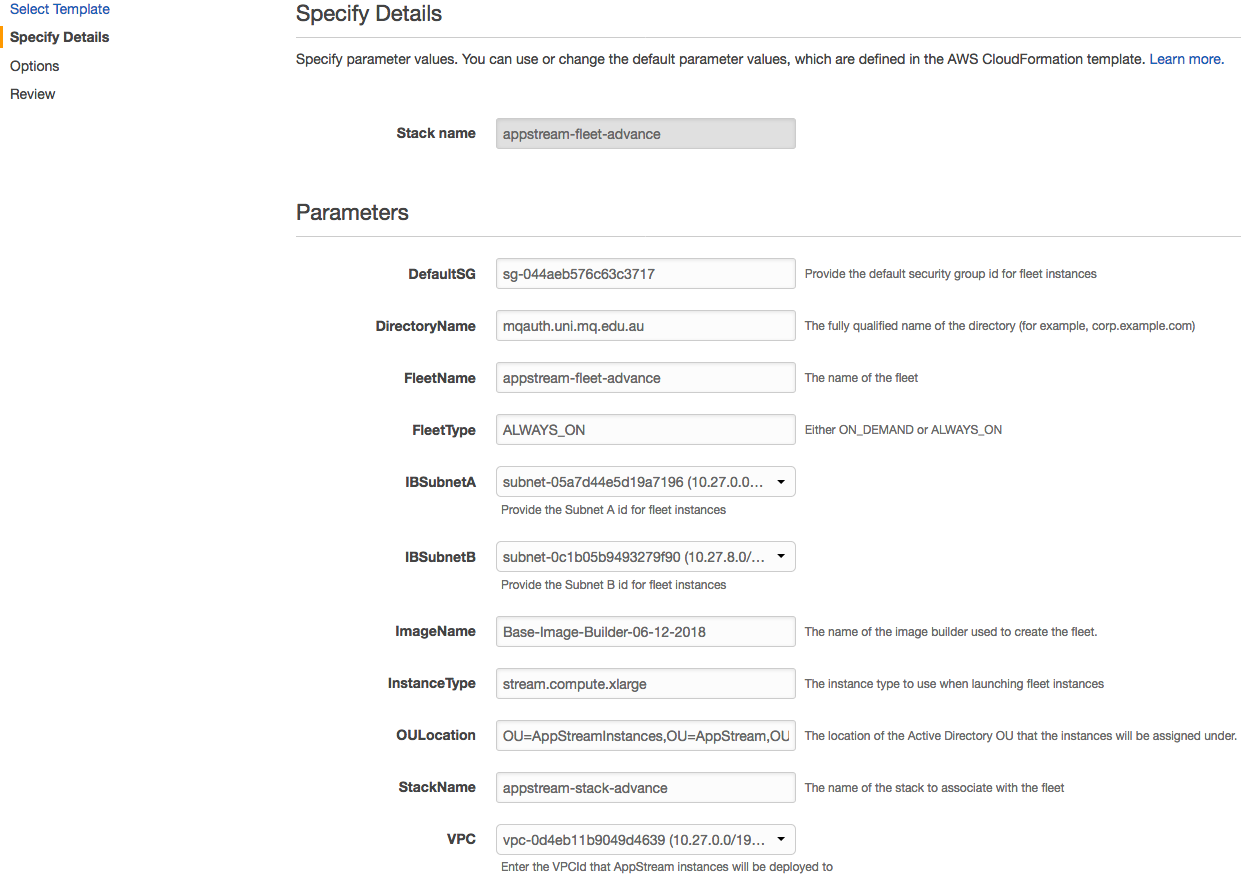
1. Log into AWS Console.
2. Click on Services drop down and select CloudFormation.



1. In CloudFormation console, select “Create Stack”
2. Select Upload a template to Amazon S3. Click on Browse to upload the CloudFormation script for execution.



1. Click on “Next”.
2. Provide a unique stack name and values for the Parameters.



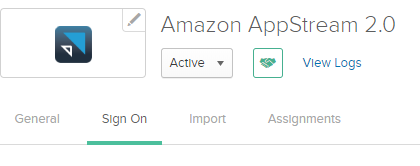
1. Click Next.
2. Click Next.
3. Click Create.

# Okta and AppStream 2.0 Integration

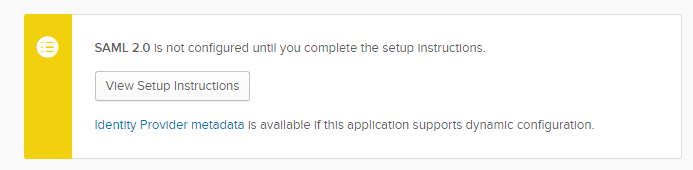
After the CloudFormation templates are successfully deployed, follow the below instructions to complete AppStream 2.0 and Okta integration. The following instruction assumes the AppStream 2.0 is already added to Okta as an user application.

## Generate metadata file for the AppStream 2.0 stack

1. Log into Okta console as an Admin.
2. Find the AppStream 2.0 application in Okta and click on Sign On tab.

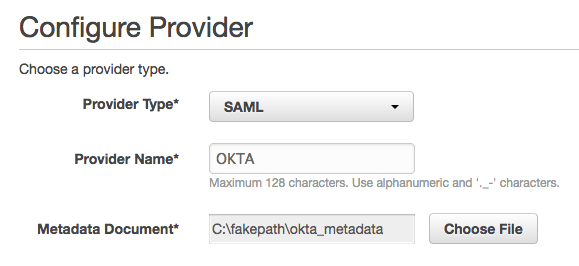
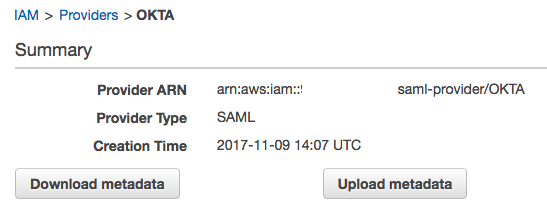


1. Click “Identity Provider metadata”. Save the metadata file locally (you will upload this to AWS console).



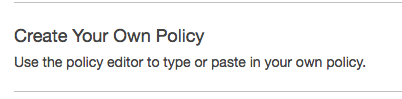
1. Copy and save the metadata.xml.

## Create SAML Provider

1. Open AWS Console and Click on IAM service.
2. Click “Identity Providers” and click “Create Provider”
3. Choose Provider Type: SAML. Give Provider a Name. Upload the metadata.xml file.
4. 
5. Click Create. Note your Provider ARN.
6. 

## Create Policy and Role

1. Open AWS Console and Click on IAM service.
2. Click “Policies” and “Create Policy”
3. Click Create Your Own Policy



1. Give your policy a recognizable Name, Description and past the policy details as provided. This will give users access to stack “appstream-stack-test” in AWS account “12345678910” as an example.

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "appstream:Stream",

"Resource": "arn:aws:appstream:ap-southeast-2:12345678910:stack/appstream-stack-test",

"Condition": {

"StringEquals": {

"appstream:userId": "${saml:sub}",

"saml:sub\_type": "persistent"

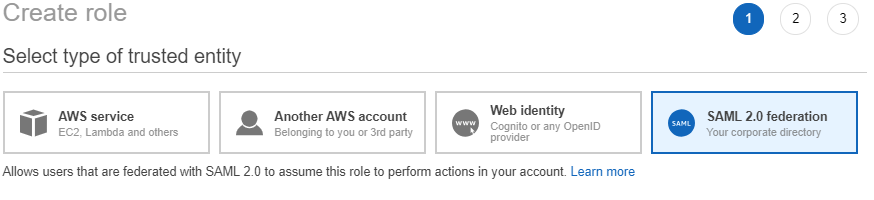
}

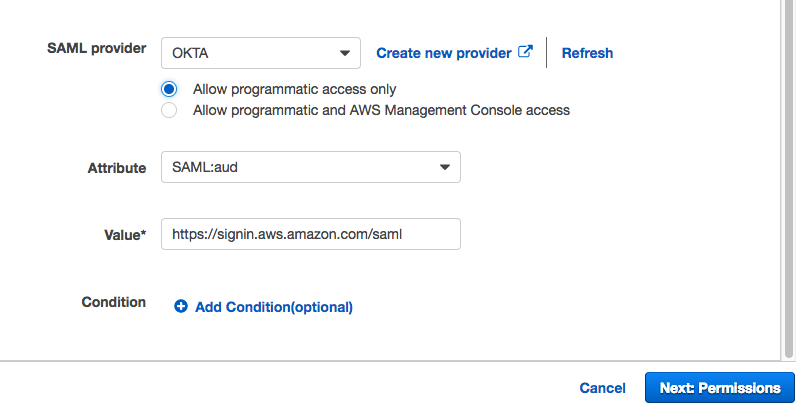
}

}

]

}

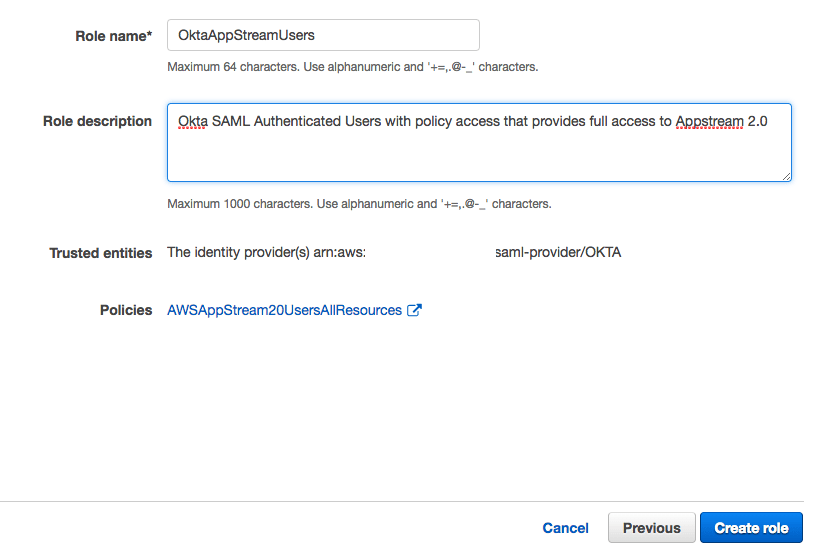
1. Once the policy is created, note down the ARN
2. Go back to the IAM console and click on “Roles”, click “Create Role”.
3. Click SAML 2.0 Federation.
4. 
5. Select SAML Provider created previously. Tick “Allow programmatic access only”. Type in Attribute “SAML:aud” with value <https://signin.aws.amazon.com/saml>. Click Next:Permissions.

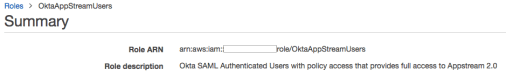


1. Select the Previsouly created AppStream policy. Click Next.

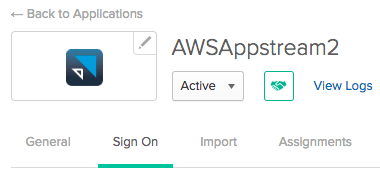
https://i0.wp.com/www.jsconsulting.services/wp-content/uploads/2017/11/word-image-429.png?w=960

https://i0.wp.com/www.jsconsulting.services/wp-content/uploads/2017/11/word-image-430.png?w=960

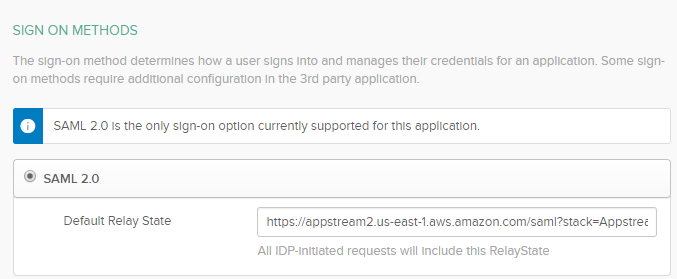
1. Click Create Role.
2. 
3. Click onto the Role Name and take note of the ARN.



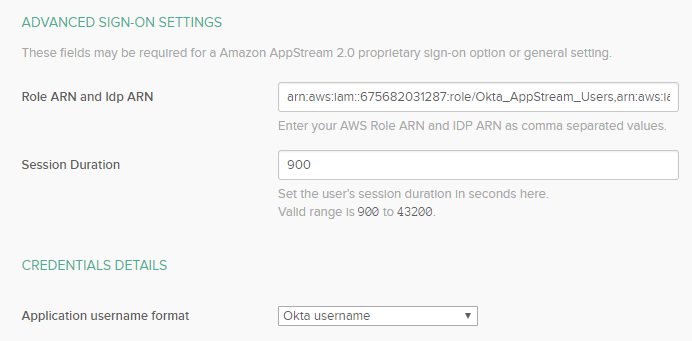
1. Take your ARN from both steps previous steps and combine them separated with a comma. i.e. roleARN,providerARN
2. For example if your Role ARN is:arn:aws:iam::123456789012:role/OktaAppStreamUsers and your IDP ARN is arn:aws:iam::123456789012:saml-provider/OKTA, enter (no white spaces):arn:aws:iam::123456789012:role/OktaAppStreamUsers,arn:aws:iam::123456789012:saml-provider/OKTA”
3. In Okta console under your Application. Click the Sign On tab.



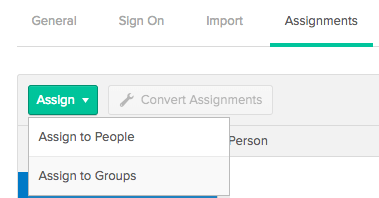
1. Click Edit.
2. Provide the Default Relay State for your AppStream sessions.
3. AppStream infrastructure is based in **Sydney ap-southeast-2** AppStream stack is called **AppStream.** Account id is **123456789012**
4. Our Relay State URL example
5. https://appstream2.ap-southeast-2.aws.amazon.com/saml?stack=Appstream&accountId=123456789012



1. Provide the Role ARN and Idp ARN



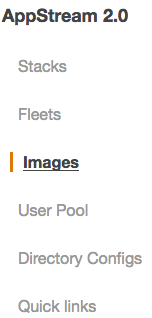
1. Click Save.
2. Assign this AppStream application nto your Okta users/AD security groups.



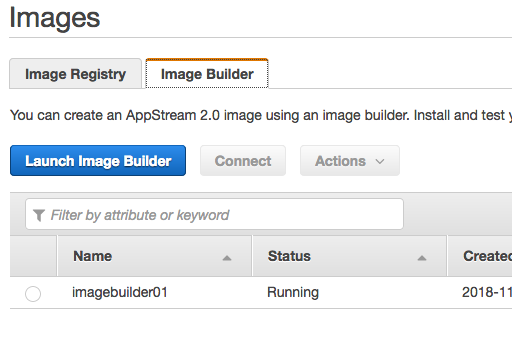
# AppStream 2.0 Image Builder

As part of the CloudFormation deployment, an AppStream 2.0 Image Builder is deployed and waiting to applications to be installed and configured.

1. Log into AWS Console.
2. Click on Services drop down and select AppStream2.0.
3. Click on Images.



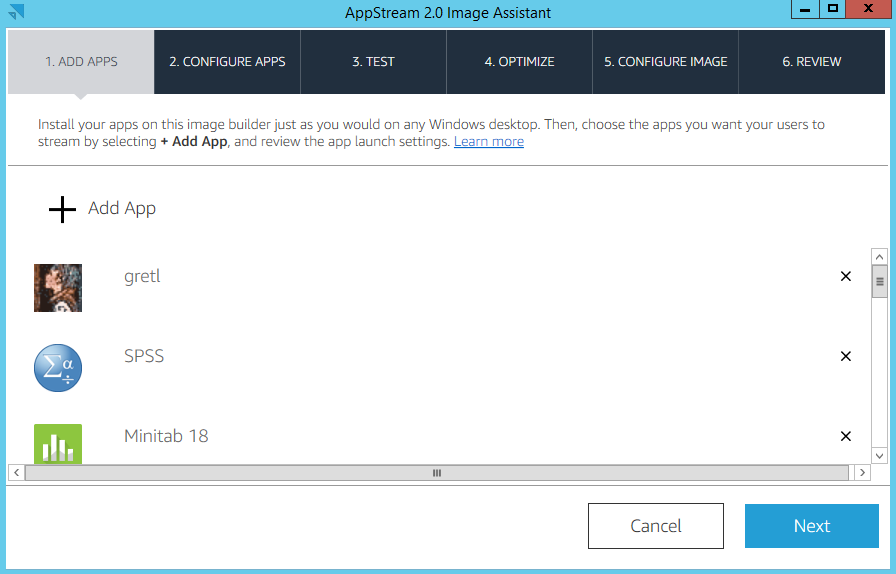
1. Select “Image Builder” tab.



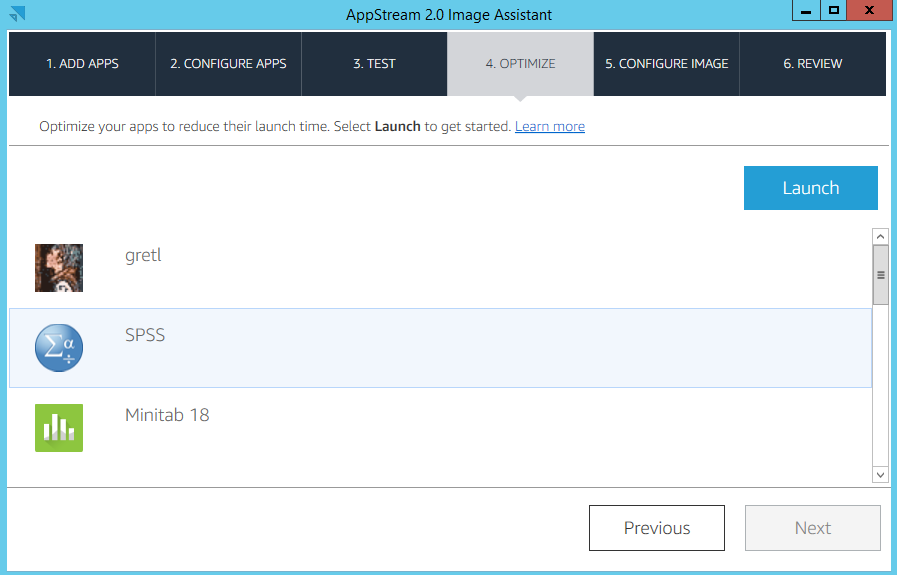
1. Select the Image Builder and click on “Connect” button to connect to the image builder instance.
2. Install and configure applications, including proxy settings and copy admin user profile to the default user profile.
3. Click on Image Assistant application on the desktop/



1. Add the application executable to Image Assistant.



1. Click Next.
2. Click Next.
3. Click Next to skip the Test.
4. Click on Launch to start the application optimization process.



1. Wait until the applications are launch then click on Next.
2. Provide a Name and Description of the image. The image name must be unique.
3. Click Next.
4. Click Finish. This will end the remote session.
5. Wait for the snapshot to complete. The snapshotting process will take 20-30 minutes. Once the process is complete, you will see the image under “Image Registry” tab.

