AWS Canvas Connector Installation Instructions

September 2022

Introduction

This slide deck will walk you through installing and configuring the AWS Canvas Connector. The instructions are broken down into three sections:

- Installing the Canvas Live Events Connector
- Installing the Canvas API Amazon AppFlow Connector
- Installing the AWS Glue and Glue Databrew resources

Download Template Files

Before you begin, you will need to download the configuration files by cloning Unicon's GitHub **AWS.CanvasLiveEventsConnector** repository:

https://github.com/Unicon/AWS.CanvasLiveEventsConnector

to your local environment.

Please see the README for further information on the project goals and components.

Installing the AWS Canvas Live Events Connector

Step 1 - Download the CloudFormation template

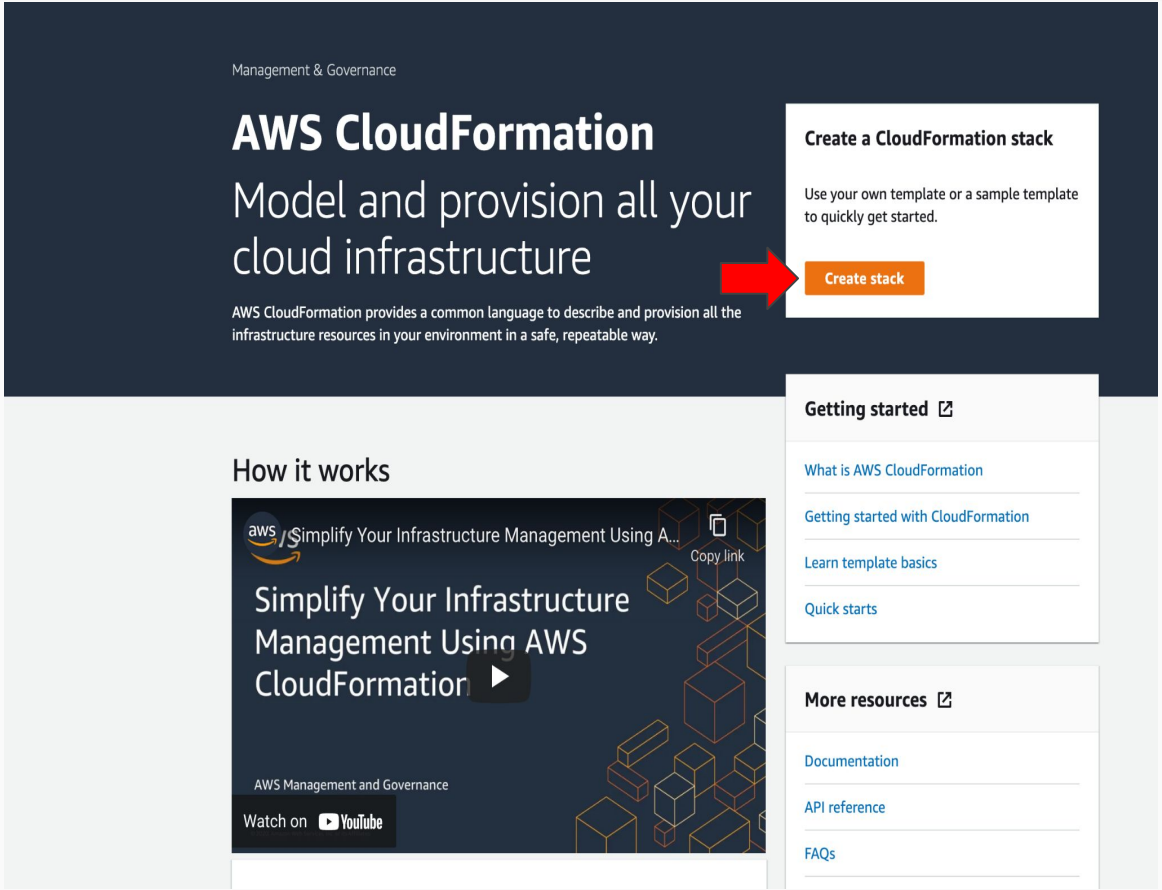
1. Use the AWS Canvas Live Events Connector CloudFormation template: **canvas-live-events.cfn.yaml**
2. If you wish to make any changes to the CloudFormation template (e.g., change resource names to match your naming conventions) do that at this time.
3. Log in to the AWS Management Console. Note, the user you are logging in with must have sufficient privileges to create CloudFormation stacks and the AWS resources needed for the Canvas Live Events Connector (list of resources below).
4. Navigate to CloudFormation

The CloudFormation stack will create the following resources:

- One Lambda function (CanvasLiveEventsLambda)
- One IAM role (CanvasLiveEventsLambdaRole)
- One Amazon S3 bucket (${Organization}-${Project}-canvas-events)
- One SQS queue (canvas-live-events-${Project}-${Organization})

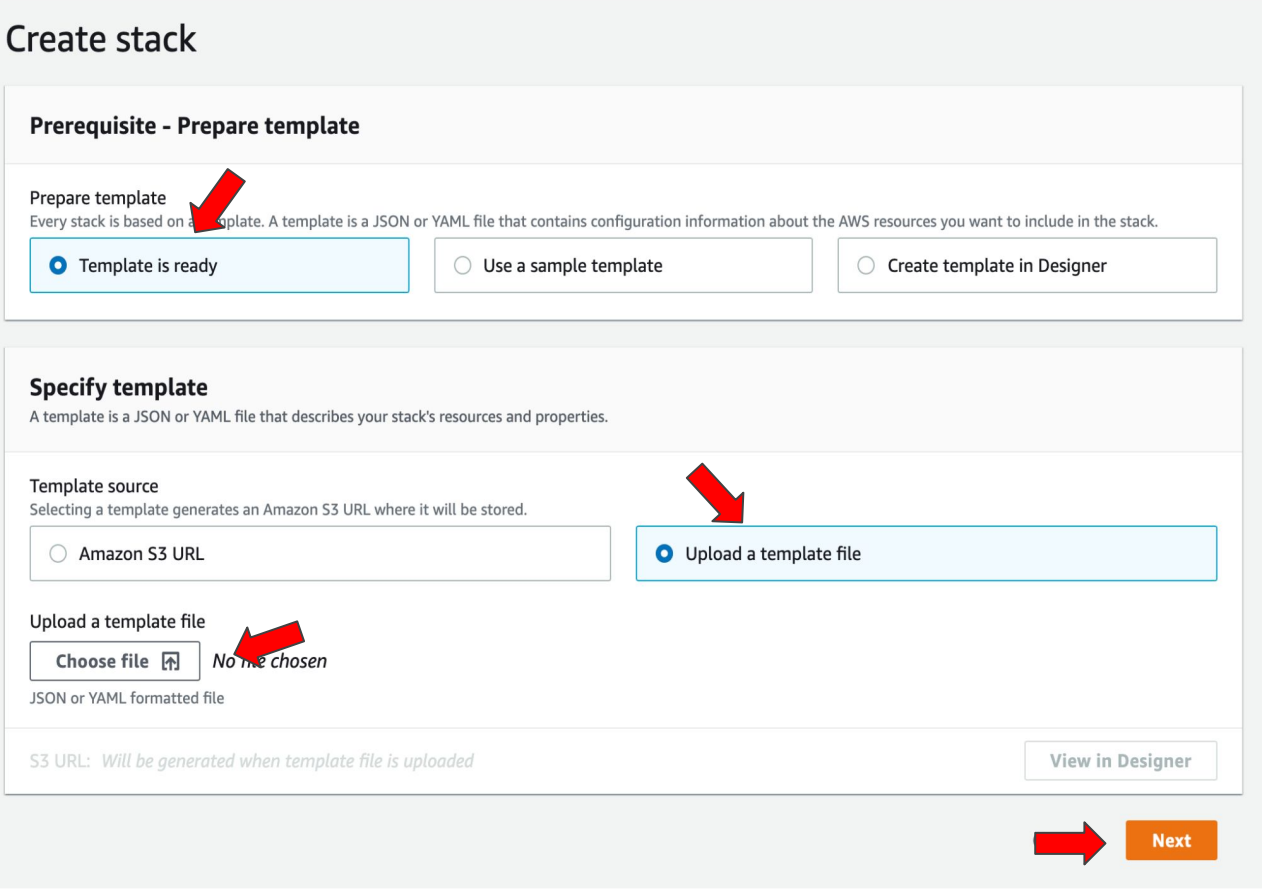
Step 2 - Create Stack

1. Click on the Create stack button



Step 3 - Create stack

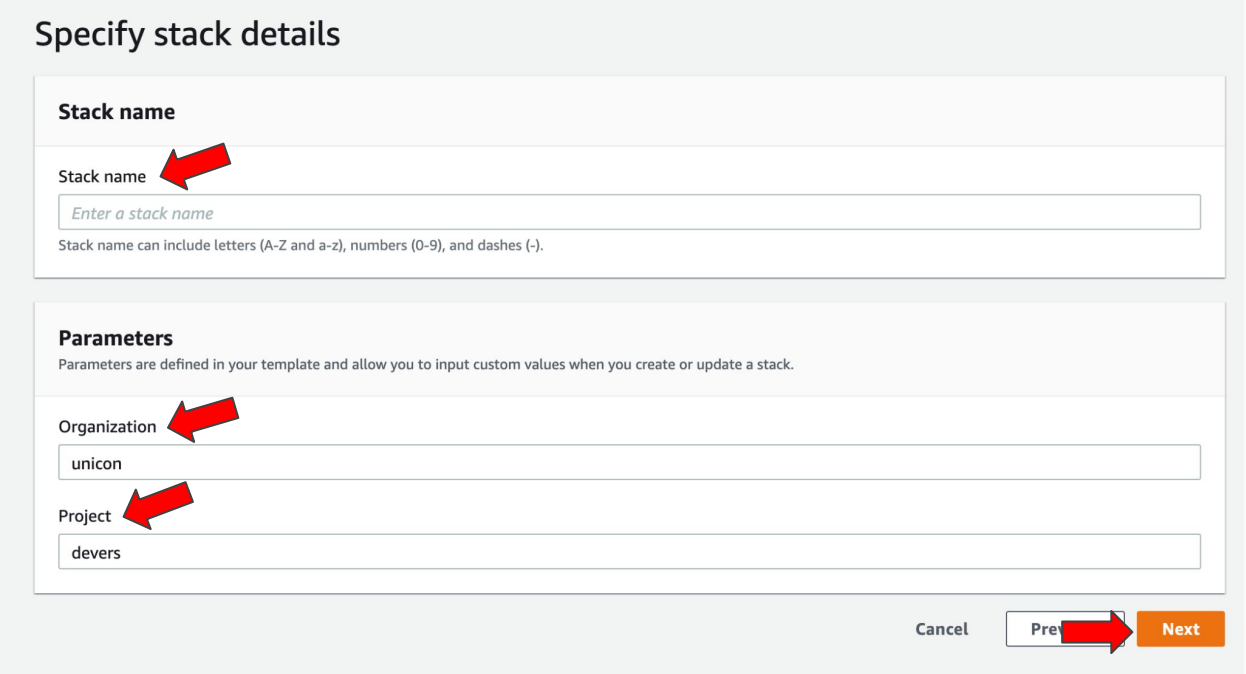
1. Select “Template is ready”
2. Select “Upload a template file”
3. Click Choose file and then choose the yaml file that you downloaded at Step 1
4. Click Next



Step 4 - Specify stack details

The Organization and Project parameters are used as substitution variables in AWS resource names (e.g., Amazon S3 bucket - calpoly-qa-canvas-events). Organization should be a short abbreviation for the institution. Project should be a short abbreviation for the initiative or environment. All AWS resources are also tagged with the Project value.

1. Enter a Stack name (e.g., CanvasLiveEventsStack)
2. Enter a value for the Organization parameter (e.g., calpoly)
3. Enter a value for the Project parameter (e.g., qa)
4. Click Next

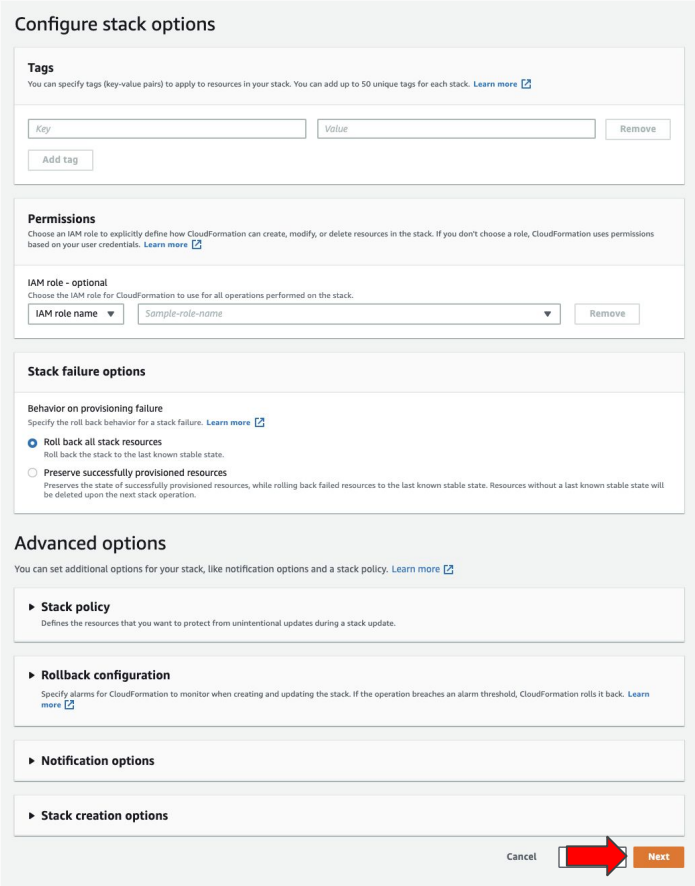


Step 5 - Configure stack options

No changes are required on this screen. However, if you have your own AWS best practices or conventions feel free to apply them now.

If you do not select IAM role to use, CloudFormation will create one for you.

1. Click Next

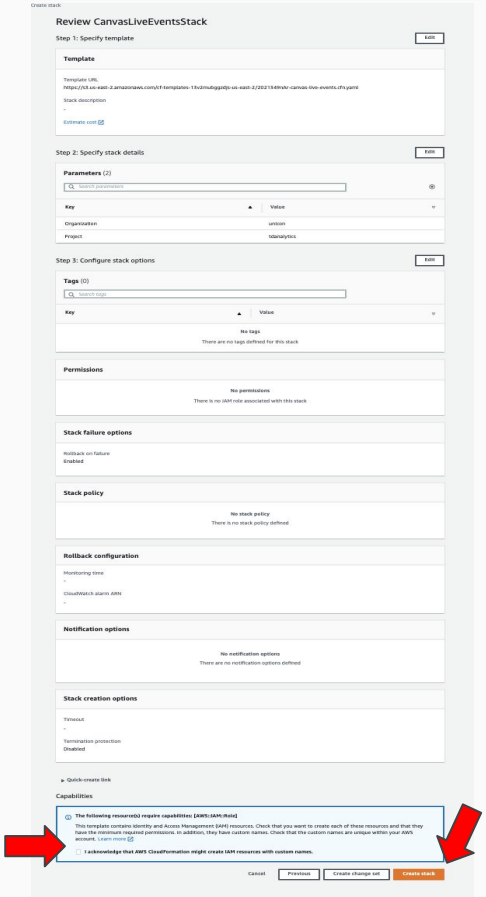


Step 6 - Review stack

Review the stack configuration.

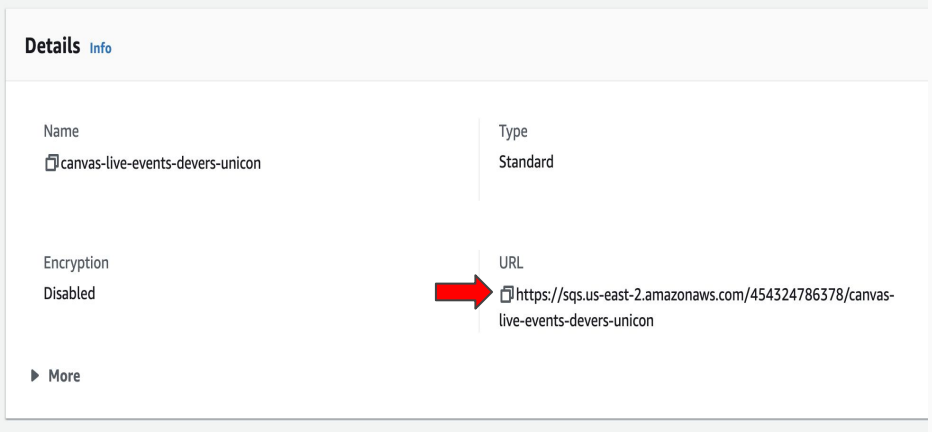
1. Click the checkbox (I acknowledge that AWS CloudFormation might create IAM resources with custom names.) in the blue box
2. Click Create stack

This will initiate the process of creating all of the required resources. This process can take a couple of minutes.



Step 7 - Setup AWS Canvas Live Events

1. In the AWS Management Console, navigate to SQS. Click on the link for the queue that was just created and copy the URL value. You will need this URL when configuring the Live Events in Canvas.
2. Open this link https://community.canvaslms.com/t5/Admin-Guide/How-do-I-subhttps://community.canvaslms.com/t5/Admin-Guide/How-do-I-subscribe-to-Live-Events-using-Canvas-Data-Services/ta-p/227scribe-to-Live-Events-using-Canvas-Data-Services/ta-p/227 and follow the instructions to configure Live Events.
   a. For Delivery Method select SQS
   b. For URL, enter the value you copied in step 1
   c. For Authentication select None (by default the queue is locked down to only Instructure and your AWS account); if you prefer to use AWS Credentials you can
   d. For Message Type select Caliper 1.1
   e. For Application Type select Data Streaming
   f. Check the Subscription box (this will send all messages, if you would prefer only certain messages, select those now)
3. Click Save & Exit



After completing these steps event data will be flowing into Amazon S3. To verify, navigate to S3 in the AWS management console, click on the bucket that ends with -canvas-events. You should see json files partitioned by date. If you don't see any events, log in and log out of Canvas, wait a minute or so and check S3 again.

Installing the Canvas API Amazon AppFlow Connector

Step 1 - Download the CloudFormation template

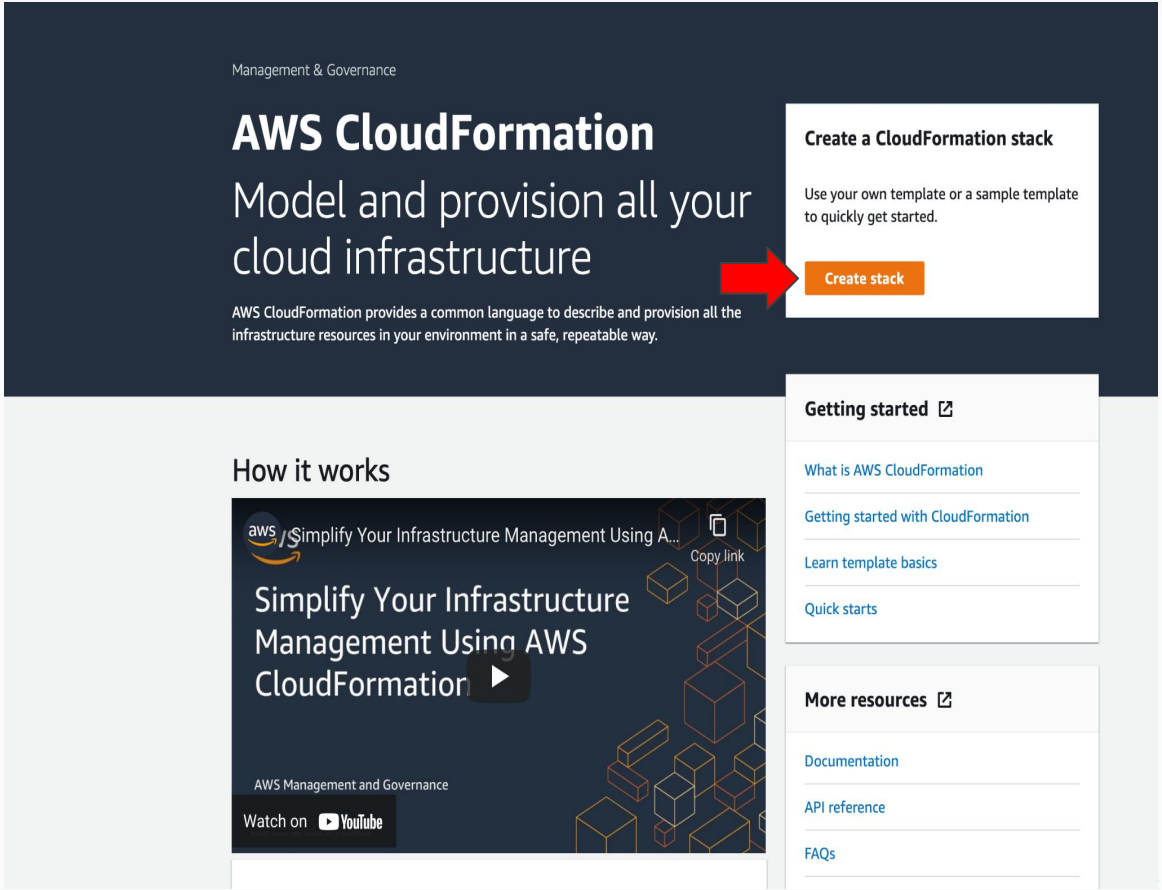
1. Use the Canvas API Connector CloudFormation template: **canvas-api-lambda.cfn.yaml**
2. If you wish to make any changes to the CloudFormation template (e.g., change resource names to match your naming conventions) do that at this time.
3. Log in to the AWS Management Console. Note, the user you are logging in with must have sufficient privileges to create CloudFormation stacks and the AWS resources needed for the Canvas API Connector (list of resources below).
4. Navigate to CloudFormation

The CloudFormation stack will create the following resources:

- Three Amazon S3 buckets
  - ${Organization}-${Project}-canvas-assignments
  - ${Organization}-${Project}-canvas-courses
  - ${Organization}-${Project}-canvas-users
- One Lambda function (CanvasApiAppFlowLambda)
- One IAM role

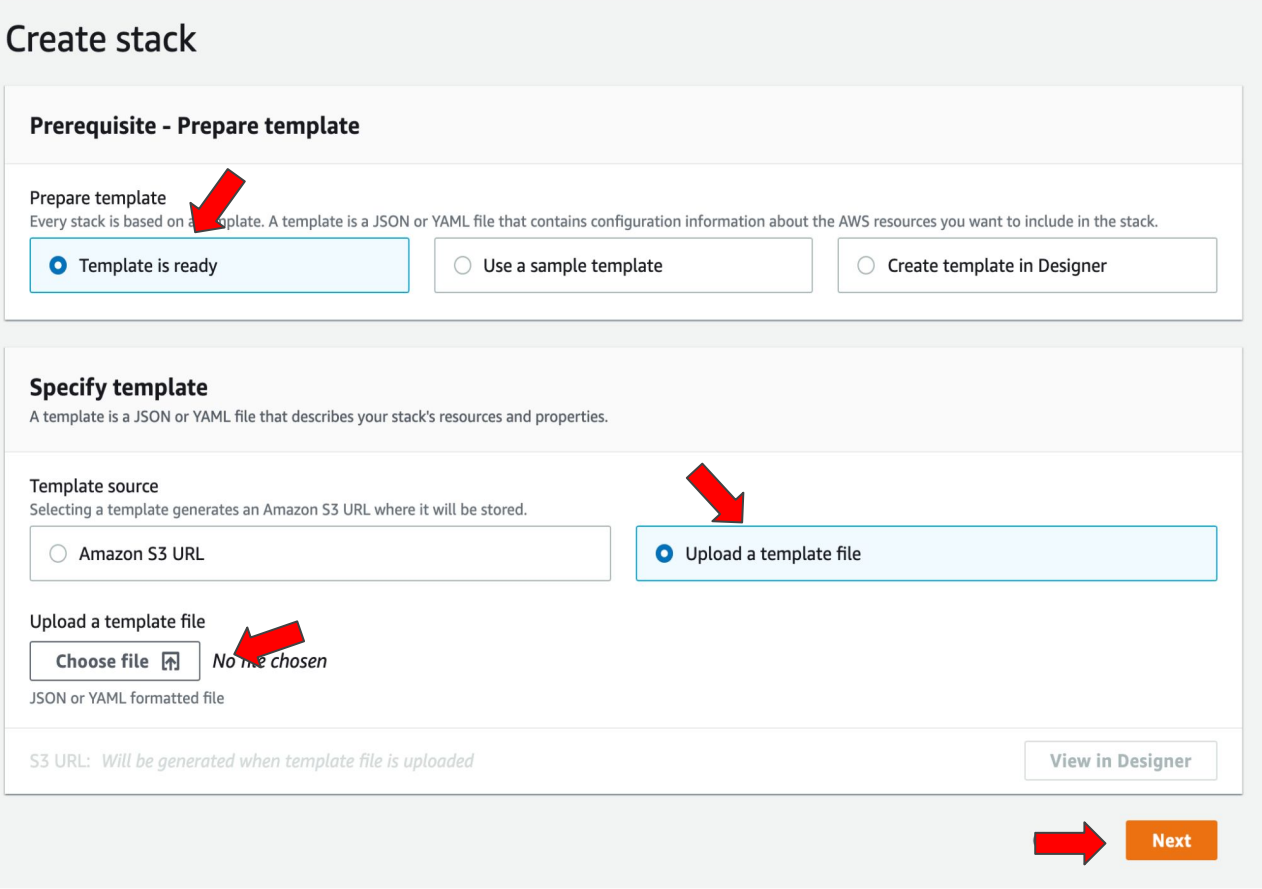
Step 2 - Create Stack

1. Click on the Create stack button



Step 3 - Create stack

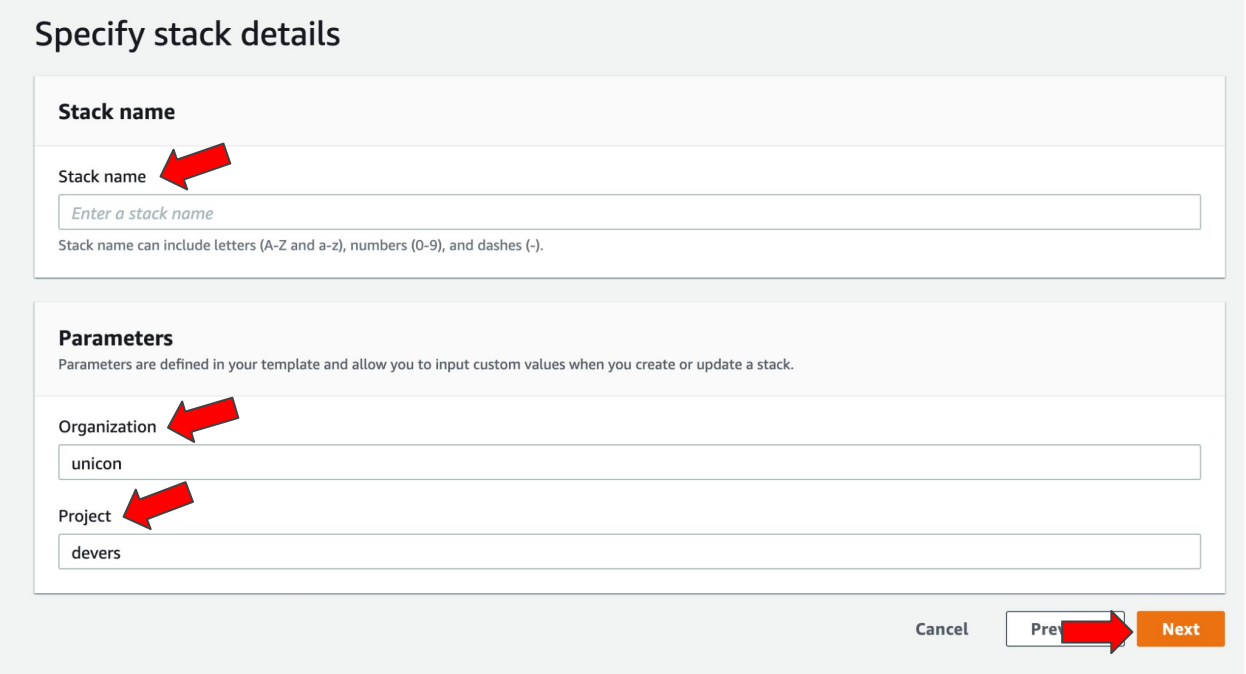
1. Select “Template is ready”
2. Select “Upload a template file”
3. Click Choose file and then choose the yaml file that you downloaded at Step 1
4. Click Next



Step 4 - Specify stack details

The Organization and Project parameters are used as substitution variables in AWS resource names (e.g., Amazon S3 bucket - calpoly-qa-canvas-events). Organization should be a short abbreviation for the institution. Project should be a short abbreviation for the initiative or environment. All AWS resources are also tagged with the Project value.

1. Enter a Stack name (e.g., CanvasAppFlowStack)
2. Enter a value for the Organization parameter (e.g., calpoly)
3. Enter a value for the Project parameter (e.g., qa)
4. Click Next

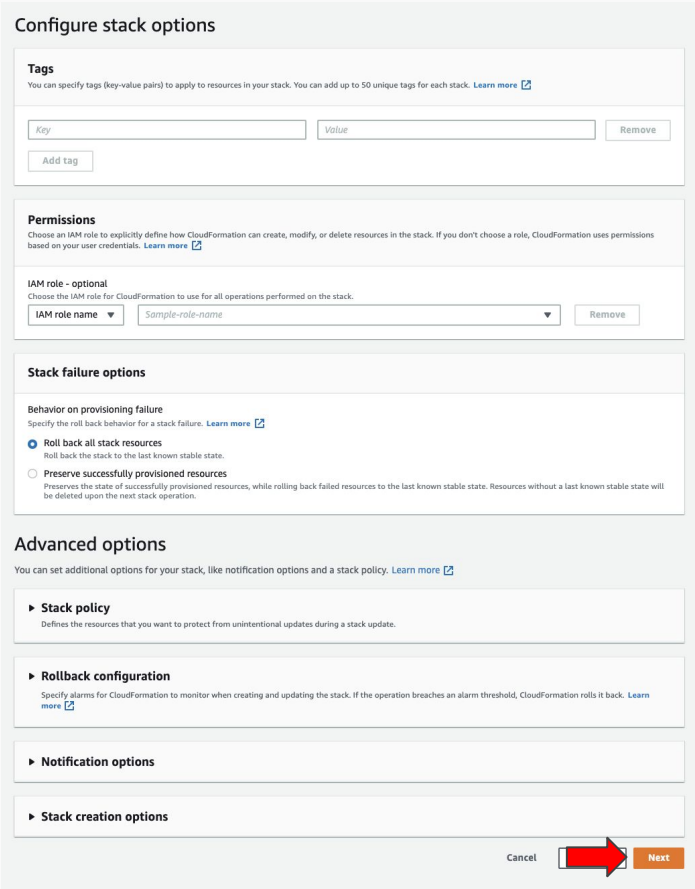


Step 5 - Configure stack options

No changes are required on this screen. However, if you have your own AWS best practices or conventions feel free to apply them now.

If you do not select IAM role to use, CloudFormation will create one for you.

1. Click Next

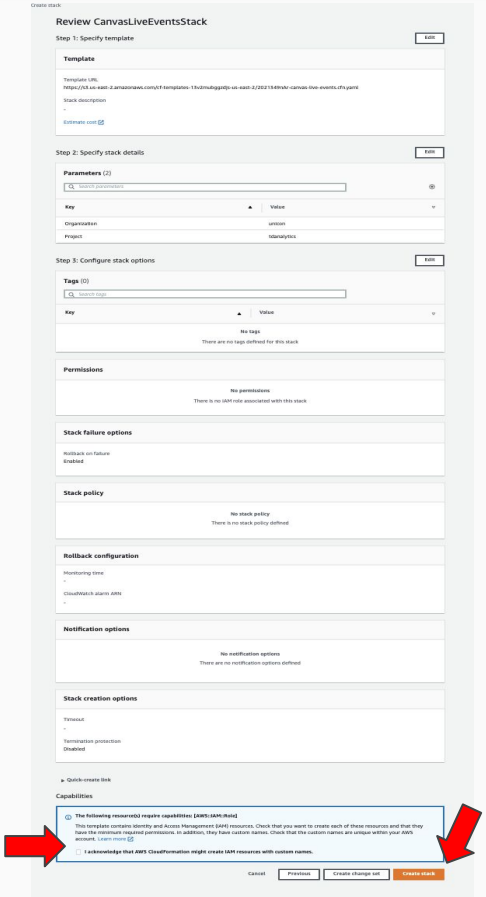


Step 6 - Review stack

Review the stack configuration.

1. Click the checkbox (I acknowledge that AWS CloudFormation might create IAM resources with custom names.) in the blue box
2. Click Create stack

This will initiate the process of creating all of the required resources. This process can take a couple of minutes.



Canvas Access Token

You will need a long lived Canvas API Access token in order to use the Canvas API with Amazon AppFlow. The easiest way to get that token is to follow the steps listed here: https://community.canvaslms.com/t5/Admin-Guide/How-do-I-manage-API-access-tokens-as-an-admin/ta-p/89

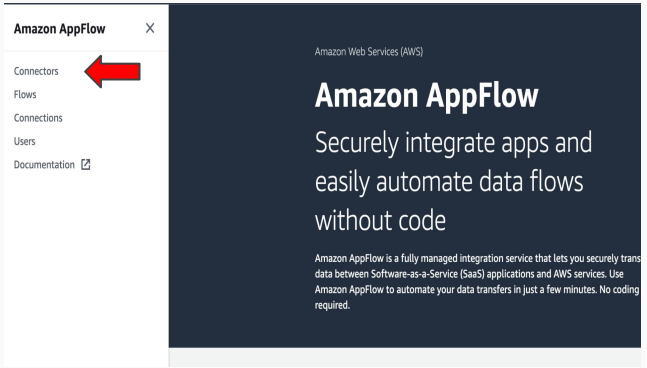
A couple of caveats:

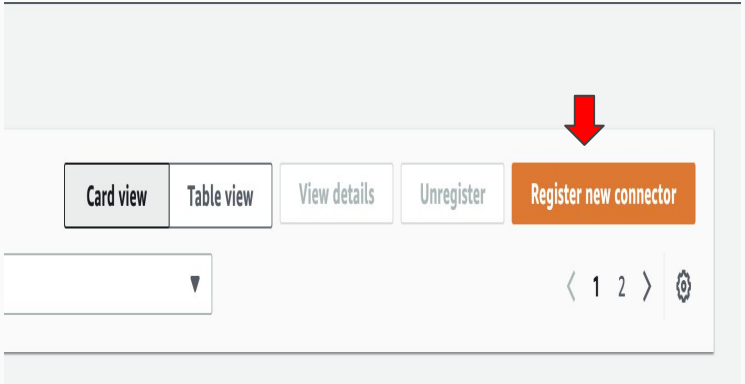
- If you want full access to the data for a Canvas account the user that you are generating the token for must have Canvas Admin privileges
- It is likely that we will change this in the future to use a more traditional OAuth2 approach

Step 1 - Create the Amazon AppFlow Connector

Unfortunately, due to the fact that Amazon AppFlow is a newer service, not everything is available via CloudFormation.

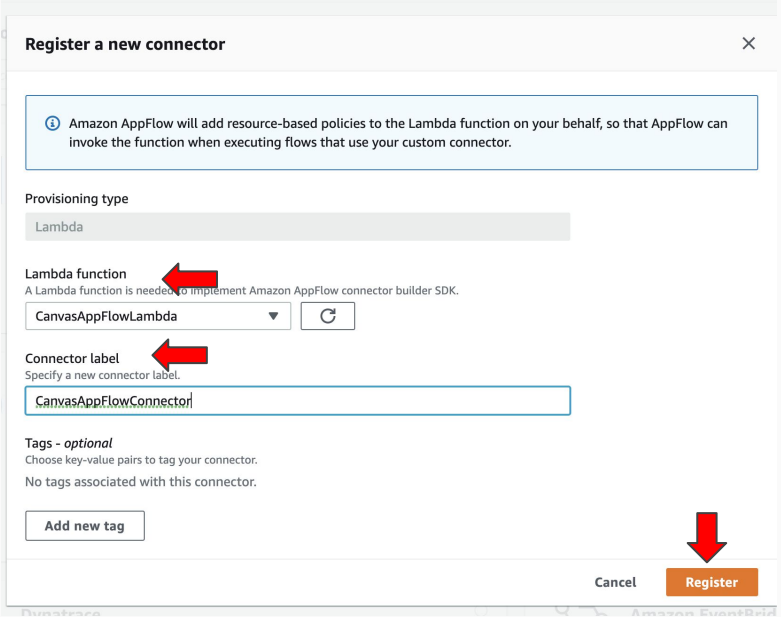
1. In the AWS Console, navigate to Amazon AppFlow
2. Click on Connectors in the left navigation pane
3. Click on Register new connector





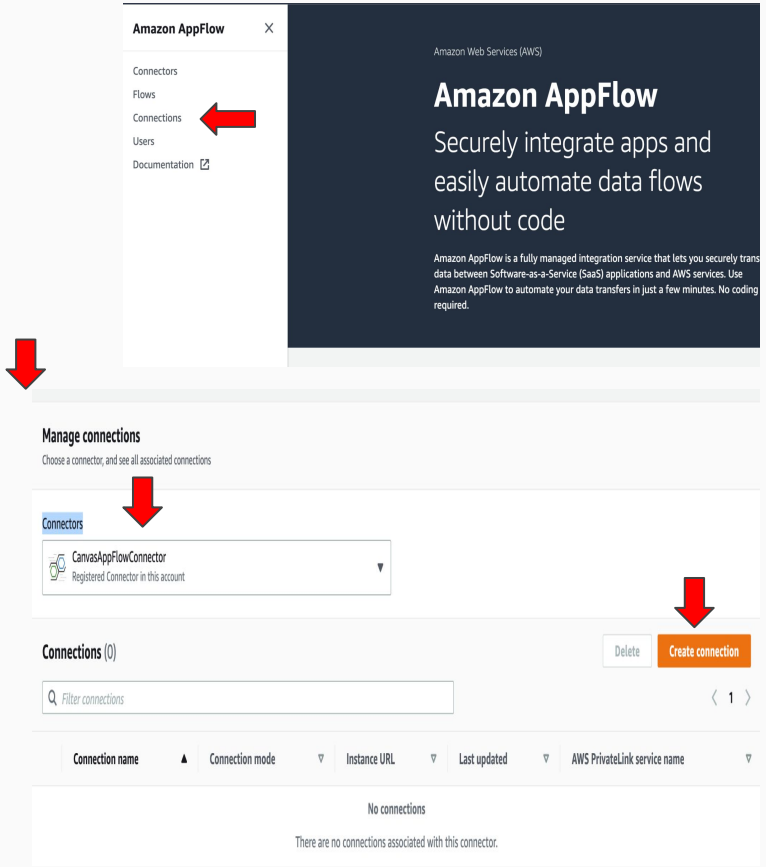
Step 1 cont. - Create AppFlow Connector

1. Select Lambda function: CanvasAppFlowLambda
2. Give the Connector a name (e.g., CanvasAppFlowConnector)
3. Optionally add any tags that you want to the Connector
4. Click the Register button



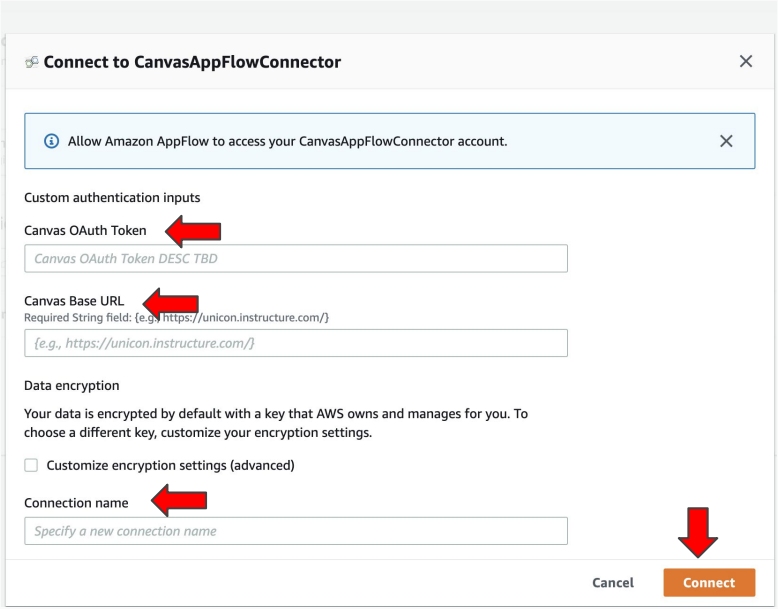
Step 2 - Create AppFlow Connection

1. In the AWS Console, navigate to Amazon AppFlow
2. Click on Connections in the left navigation pane
3. Click on Register new connector
4. From the Connectors dropdown, select CanvasAppFlowConnector (or whatever you named the connector in step 1)
5. Click Create connection



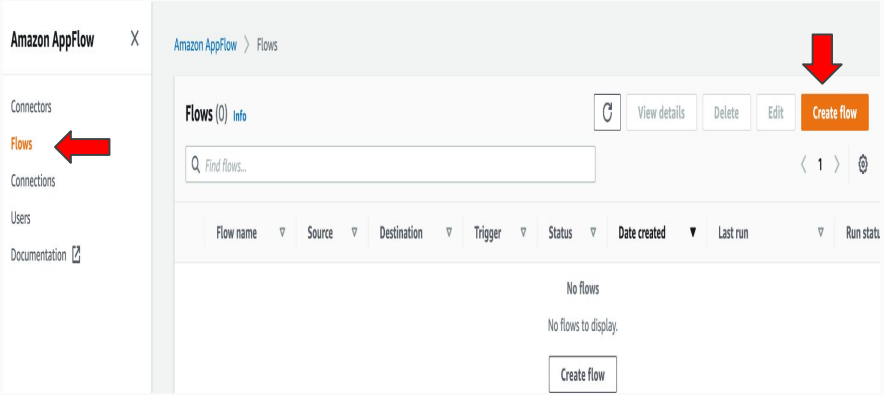
Step 2 cont. - Create the Amazon AppFlow Connection

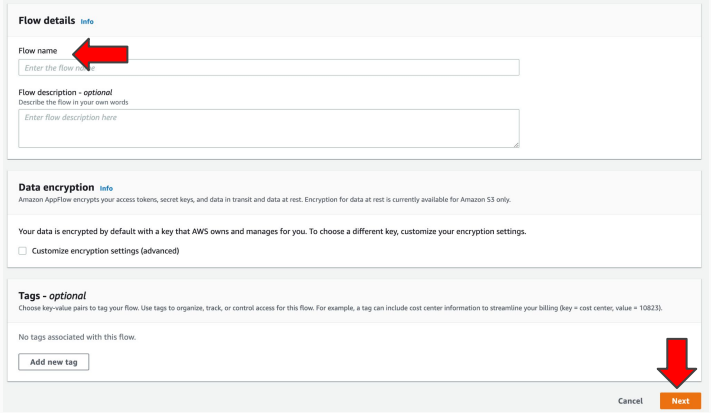
1. In the Canvas OAuth Token field enter the Access token that you created in Canvas
2. In the Canvas Base URL field enter the URL of your Canvas instance in http(s)://domain/ format (e.g., https://unicon.instructure.com/)
3. You can safely ignore Data encryption (configuration is encrypted by default) if you choose.
4. Enter a Connection name (e.g., CanvasAppFlowConnection)
5. Click the Connect button



Step 3 - Create User Flow

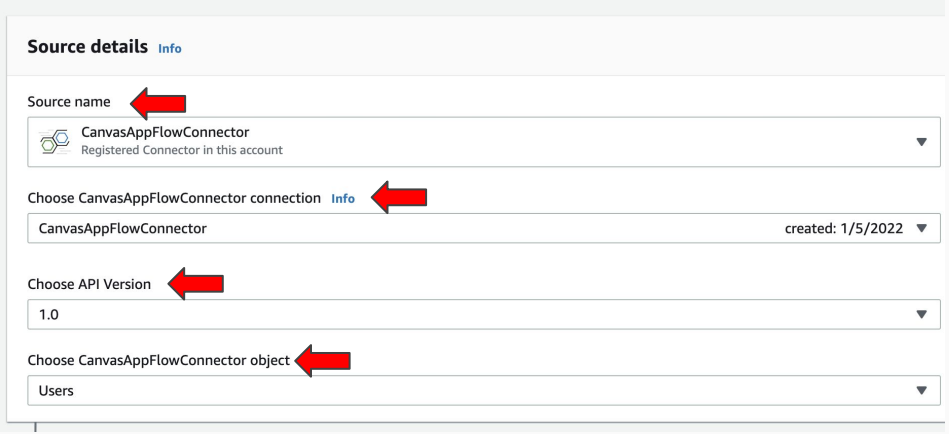
1. Click Flow in the left navigation pane
2. Click the Create flow button
3. Enter Flow name (e.g., CanvasUserFlow) - note the flow name is important, if you don't use CanvasUserFlow remember the value you enter as you will need to supply it to Cloudformation later
4. Data encryption can safely be ignored
5. Optionally add any tags
6. Click the Next button





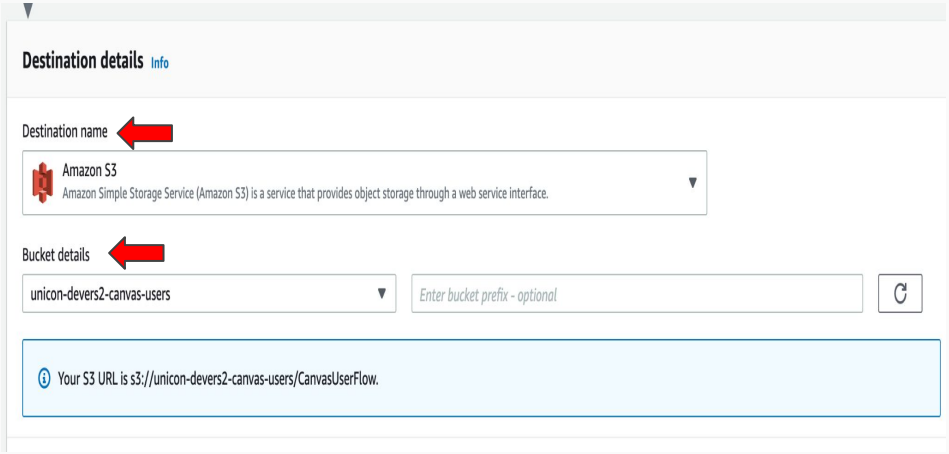
Step 3 - Create User Flow

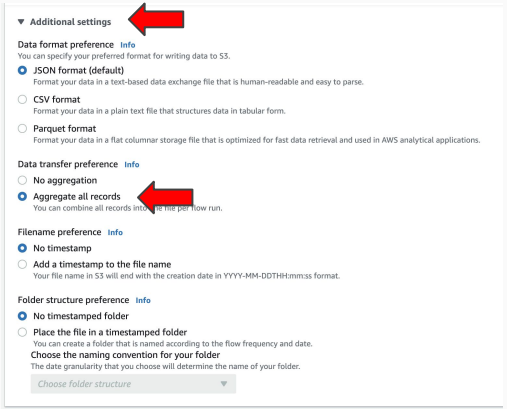
1. For Source name, select your Canvas Connector (e.g., CanvasAppFlowConnector)
2. Select the appropriate connection (e.g., CanvasAppFlowConnection)
3. Choose API Version 1.0
4. Choose Users object



Step 3 - Create User Flow

1. For Destination name, choose Amazon S3
2. For Bucket details, select the Canvas users bucket (e.g., {org}-{project}-canvas-users); leave bucket prefix empty
3. Expand Advanced settings
4. Leave all of the defaults except Aggregate all records
5. Scroll to the bottom of the page
6. Leave Run on demand as the Flow trigger (you can change this later)
7. Click the Next button

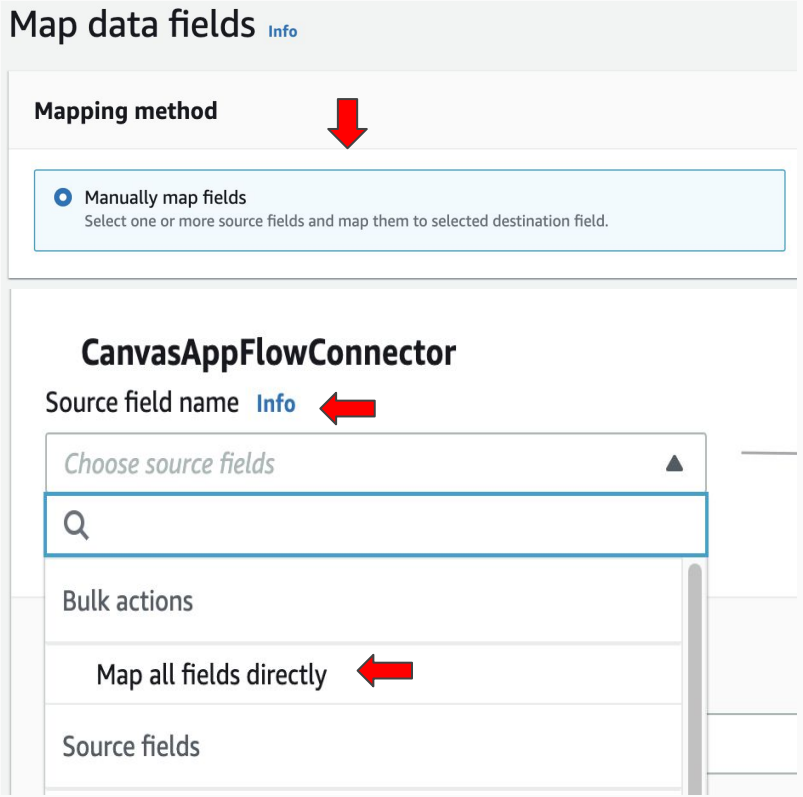




Step 3 - Create User Flow

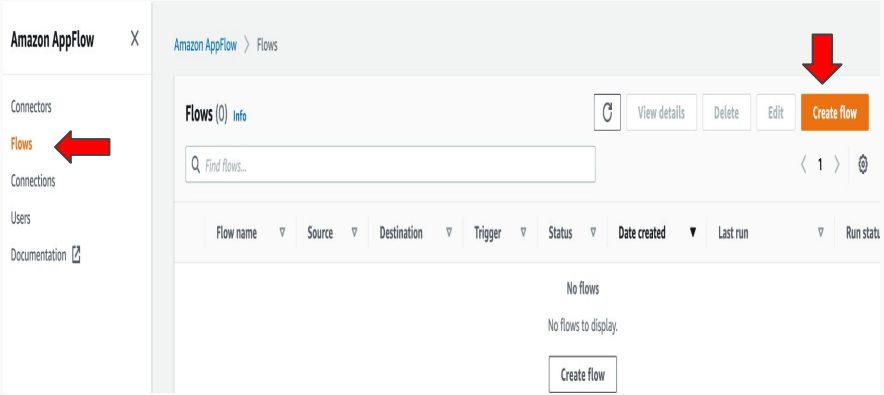
1. Select Manually map fields as the Mapping method
2. For Source field name, choose Map all fields directly
3. Scroll to the bottom of the page
4. Click the Next button
5. On the Add filters page, just click the Next button
6. On the Review and create page, scroll to the bottom and click the Create Flow button

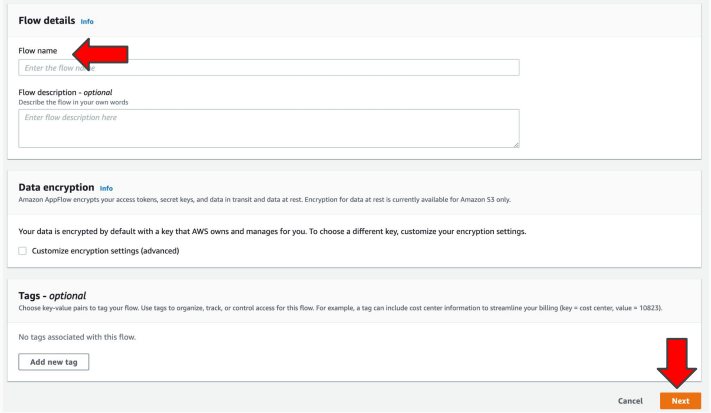
At this point, the Canvas User API flow is ready to run.



Step 4 - Create Course Flow

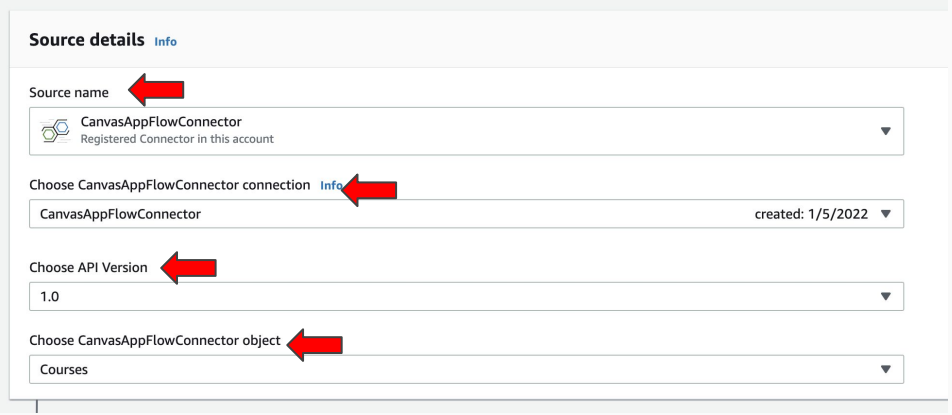
1. Click Flow in the left navigation pane
2. Click the Create flow button
3. Enter Flow name (e.g., CanvasCourseFlow) - note the flow name is important, if you don't use CanvasCourseFlow remember the value you enter as you will need to supply it to Cloudformation later
4. Data encryption can safely be ignored
5. Optionally add any tags
6. Click the Next button





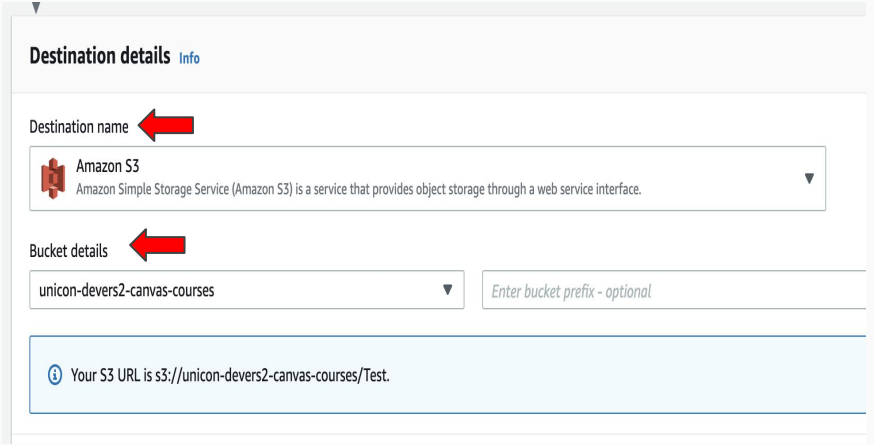
Step 4 - Create Course Flow

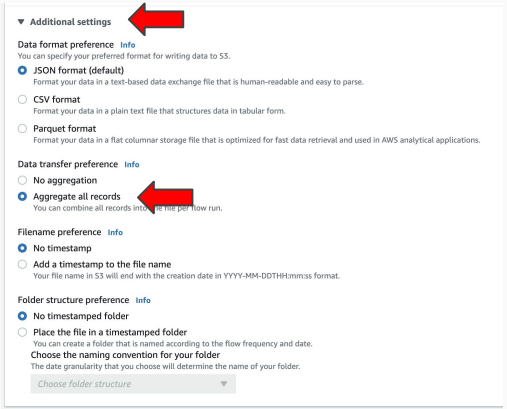
1. For Source name, select your Canvas Connector (e.g., CanvasAppFlowConnector)
2. Select the appropriate connection (e.g., CanvasAppFlowConnection)
3. Choose API Version 1.0
4. Choose Courses object



Step 4 - Create Course Flow

1. For Destination name, choose Amazon S3
2. For Bucket details, select the Canvas courses bucket (e.g., {org}-{project}-canvas-courses); leave bucket prefix empty
3. Expand Advanced settings
4. Leave all of the defaults except Aggregate all records
5. Scroll to the bottom of the page
6. Leave Run on demand as the Flow trigger (you can change this later)
7. Click the Next button

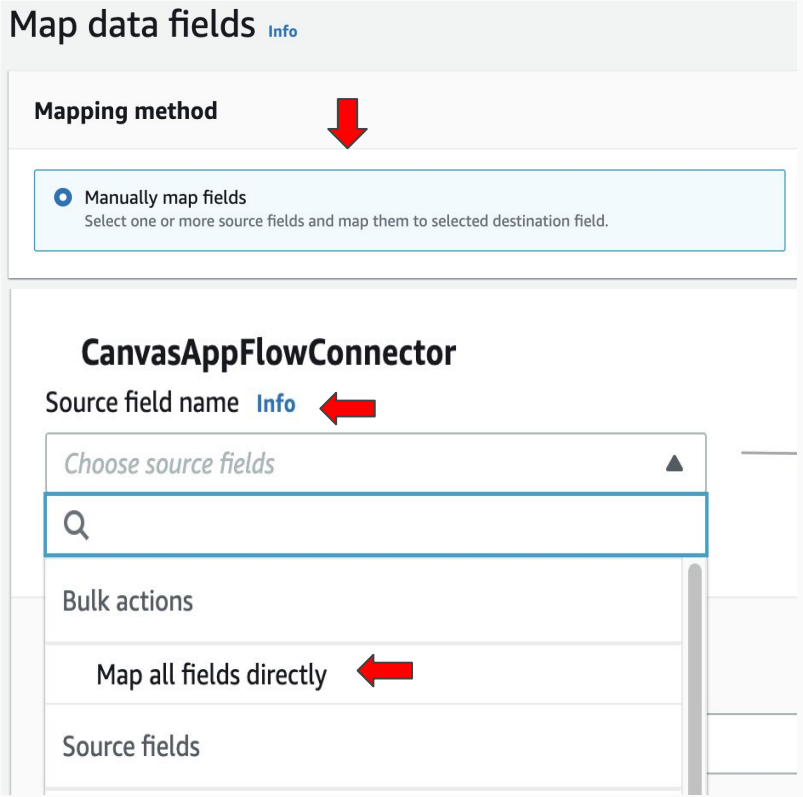




Step 4 - Create Course Flow

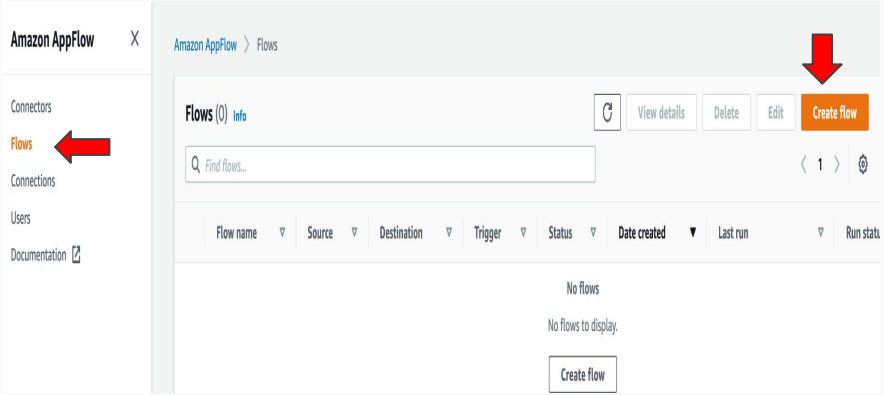
1. Select Manually map fields as the Mapping method
2. For Source field name, choose Map all fields directly
3. Scroll to the bottom of the page
4. Click the Next button
5. On the Add filters page, just click the Next button
6. On the Review and create page, scroll to the bottom and click the Create flow button

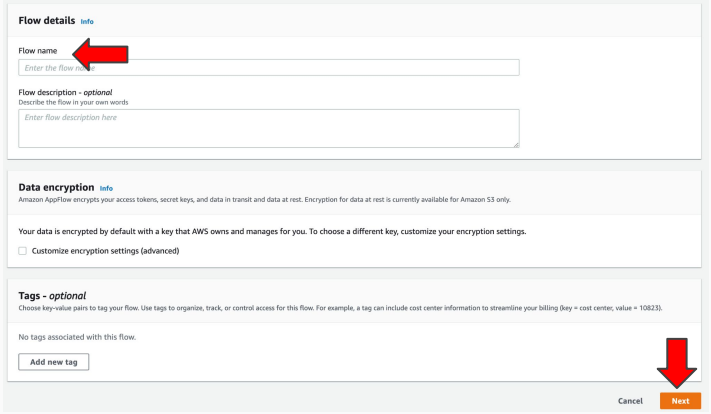
At this point, the Canvas Course API flow is ready to run.



Step 5 - Create Assignment Flow

1. Click Flow in the left navigation pane
2. Click the Create flow button
3. Enter Flow name (e.g., CanvasAssignmentFlow) - note the flow name is important, if you don't use CanvasAssignmentFlow remember the value you enter as you will need to supply it to Cloudformation later
4. Data encryption can safely be ignored
5. Optionally add any tags
6. Click the Next button



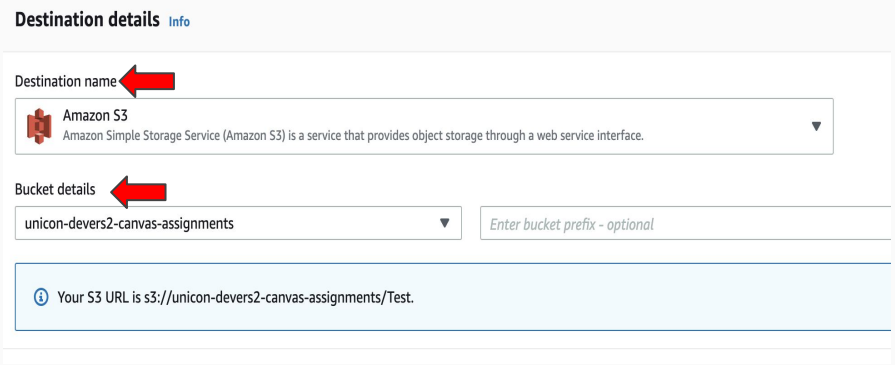


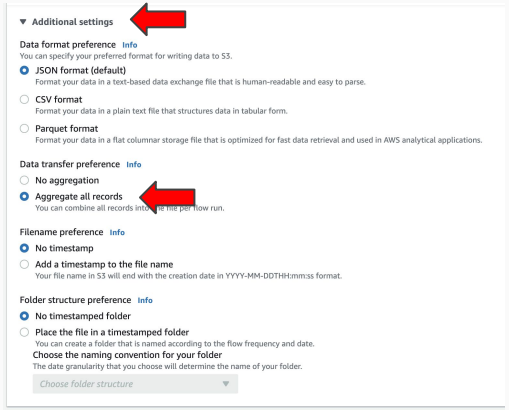
Step 5 - Create Assignment Flow

1. For Source name, select your Canvas Connector (e.g., CanvasAppFlowConnector)
2. Select the appropriate connection (e.g., CanvasAppFlowConnection)
3. Choose API Version 1.0
4. Choose Assignment object

Step 5 - Create Assignment Flow

1. For Destination name, choose Amazon S3
2. For Bucket details, select the Canvas assignments bucket (e.g., {org}-{project}-canvas-assignments); leave bucket prefix empty
3. Expand Advanced settings
4. Leave all of the defaults except Aggregate all records
5. Scroll to the bottom of the page
6. Leave Run on demand as the Flow trigger (you can change this later)
7. Click the Next button

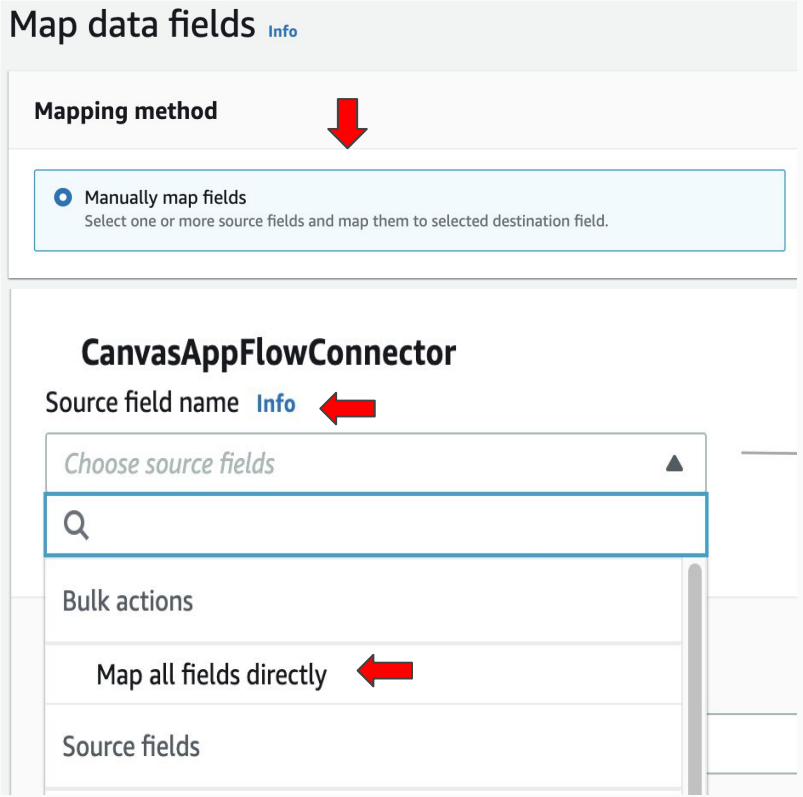




Step 5 - Create Assignment Flow

1. Select Manually map fields as the Mapping method
2. For Source field name, choose Map all fields directly
3. Scroll to the bottom of the page
4. Click the Next button
5. On the Add filters page, just click the Next button
6. On the Review and create page, scroll to the bottom and click the Create flow button

At this point, the Canvas Assignment API flow is ready to run.



Installing the Glue Components

Step 1 - Download the CloudFormation template

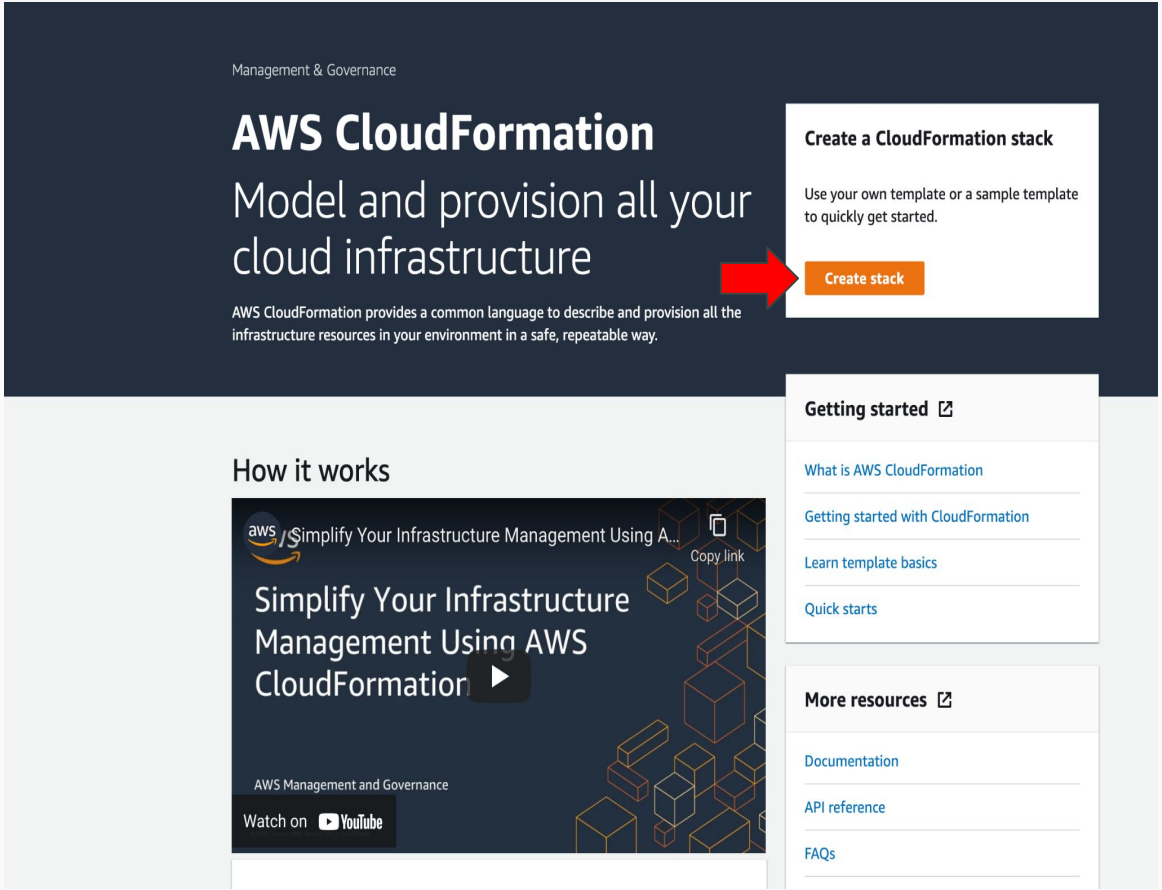
1. Use the Canvas Glue CloudFormation template: **canvas-data-product-glue.cfn.yaml**
2. If you wish to make any changes to the CloudFormation template (e.g., change resource names to match your naming conventions) do that at this time.
3. Log in to the AWS Management Console. Note, the user you are logging in with must have sufficient privileges to create CloudFormation stacks and the AWS resources needed for the AWS Canvas Live Events Connector (list of resources below).
4. Navigate to CloudFormation

The CloudFormation stack will create the following resources:

- One Glue database
- One Glue table
- One Glue crawler
- One IAM role

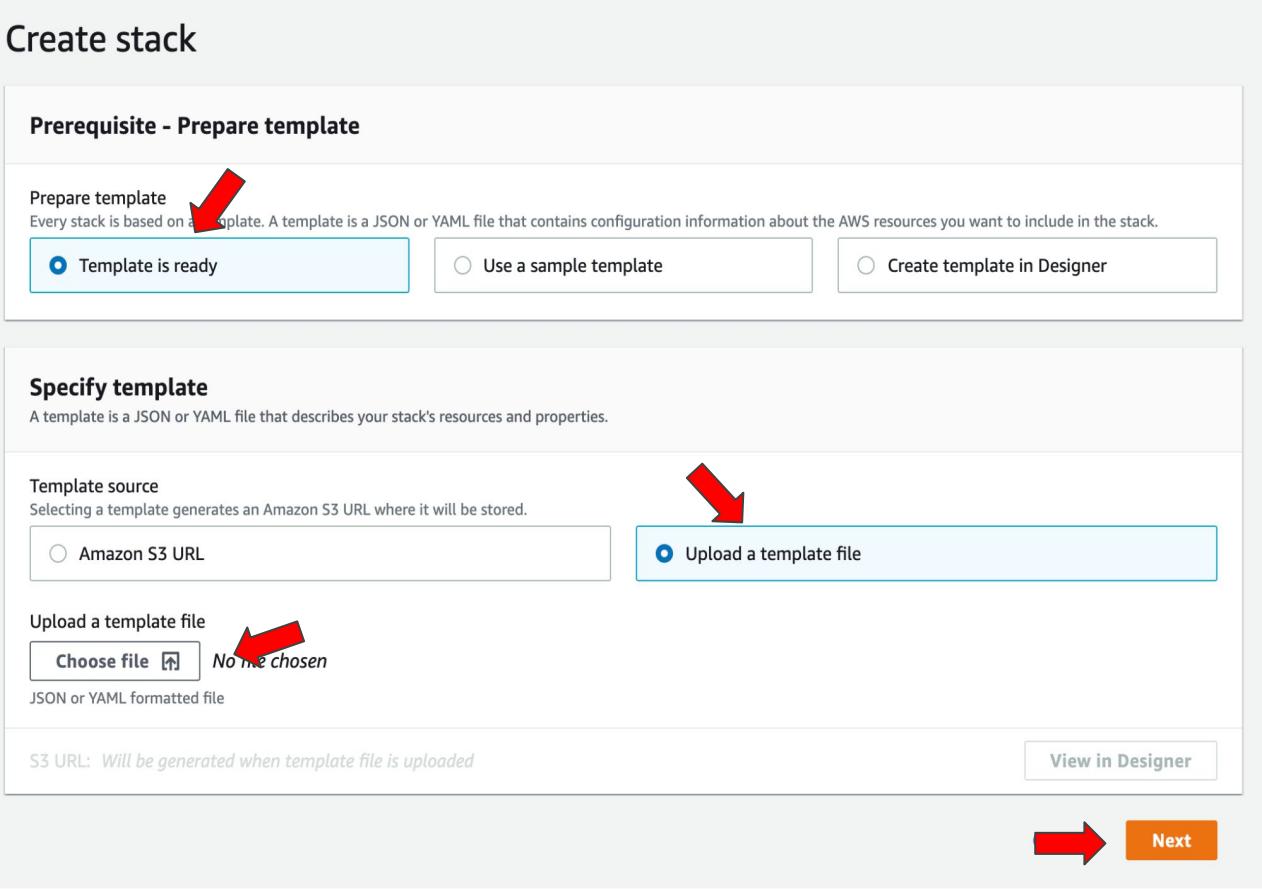
Step 2 - Create Stack

1. Click on the Create stack button



Step 3 - Create stack

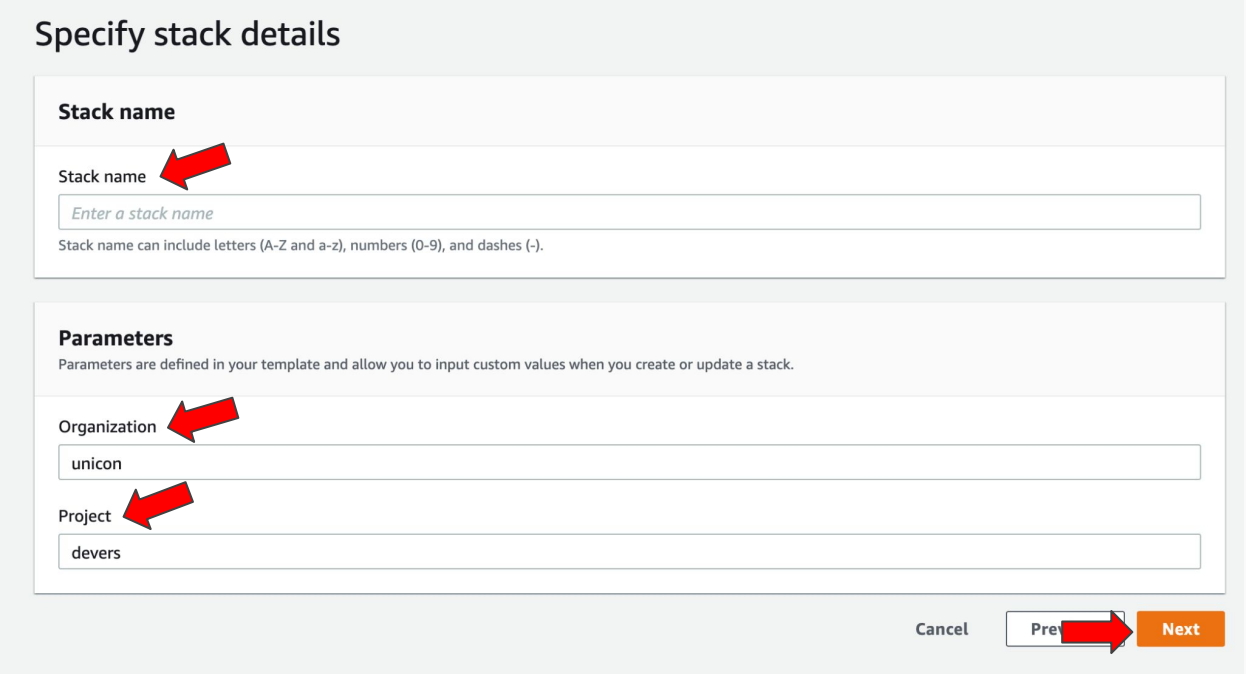
1. Select “Template is ready”
2. Select “Upload a template file”
3. Click Choose file and then choose the yaml file that you downloaded at Step 1
4. Click Next



Step 4 - Specify stack details

The Organization and Project parameters are used as substitution variables in AWS resource names (e.g., Amazon S3 bucket - calpoly-qa-canvas-events). Organization should be a short abbreviation for the institution. Project should be a short abbreviation for the initiative or environment. All AWS resources are also tagged with the Project value.

1. Enter a Stack name (e.g., CanvasGlueStack)
2. Enter a value for the Organization parameter (e.g., calpoly)
3. Enter a value for the Project parameter (e.g., qa)
4. Click Next

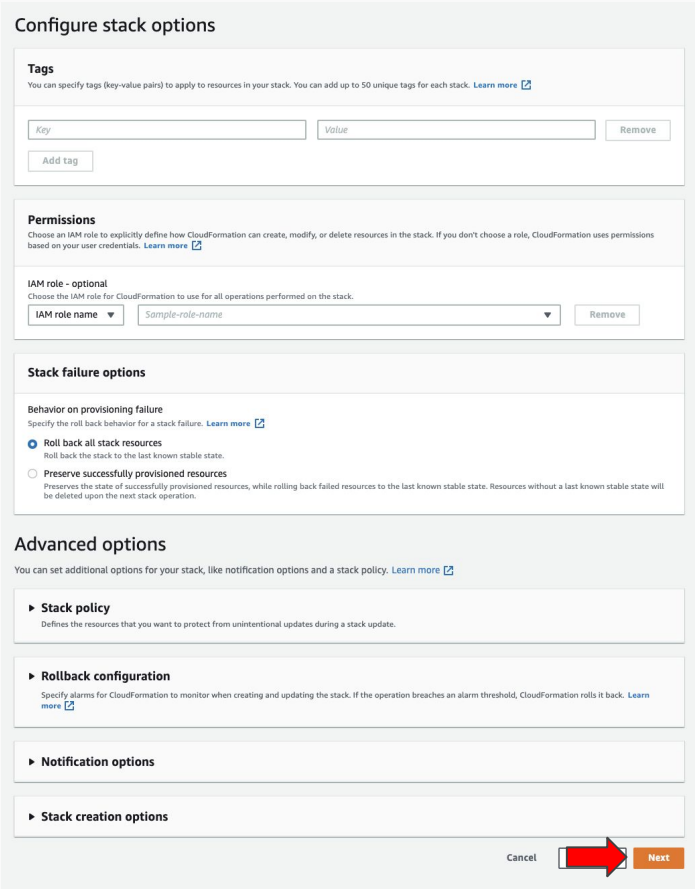


Step 5 - Configure stack options

No changes are required on this screen. However, if you have your own AWS best practices or conventions feel free to apply them now.

If you do not select IAM role to use, CloudFormation will create one for you.

1. Click Next

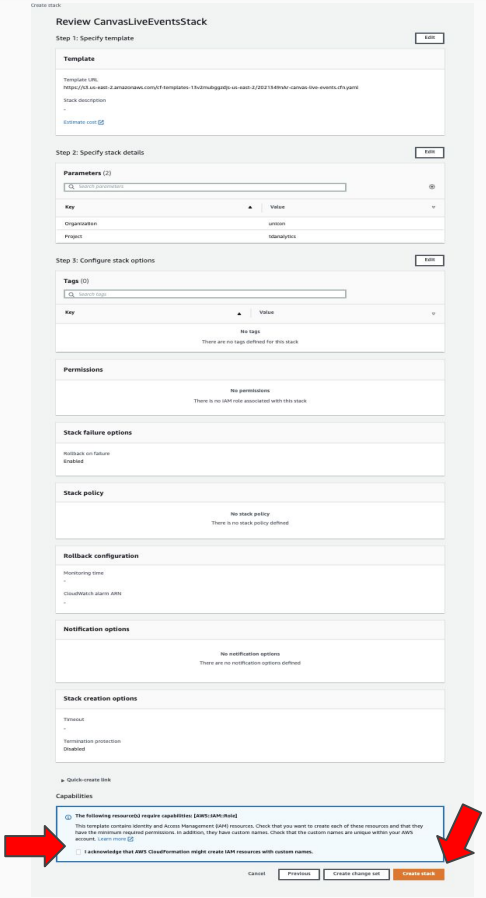


Step 6 - Review stack

Review the stack configuration.

1. Click the checkbox (I acknowledge that AWS CloudFormation might create IAM resources with custom names.) in the blue box
2. Click Create stack

This will initiate the process of creating all of the required resources. This process can take a couple of minutes.



Installing the AWS Glue DataBrew Components

Step 1 - Download the CloudFormation template

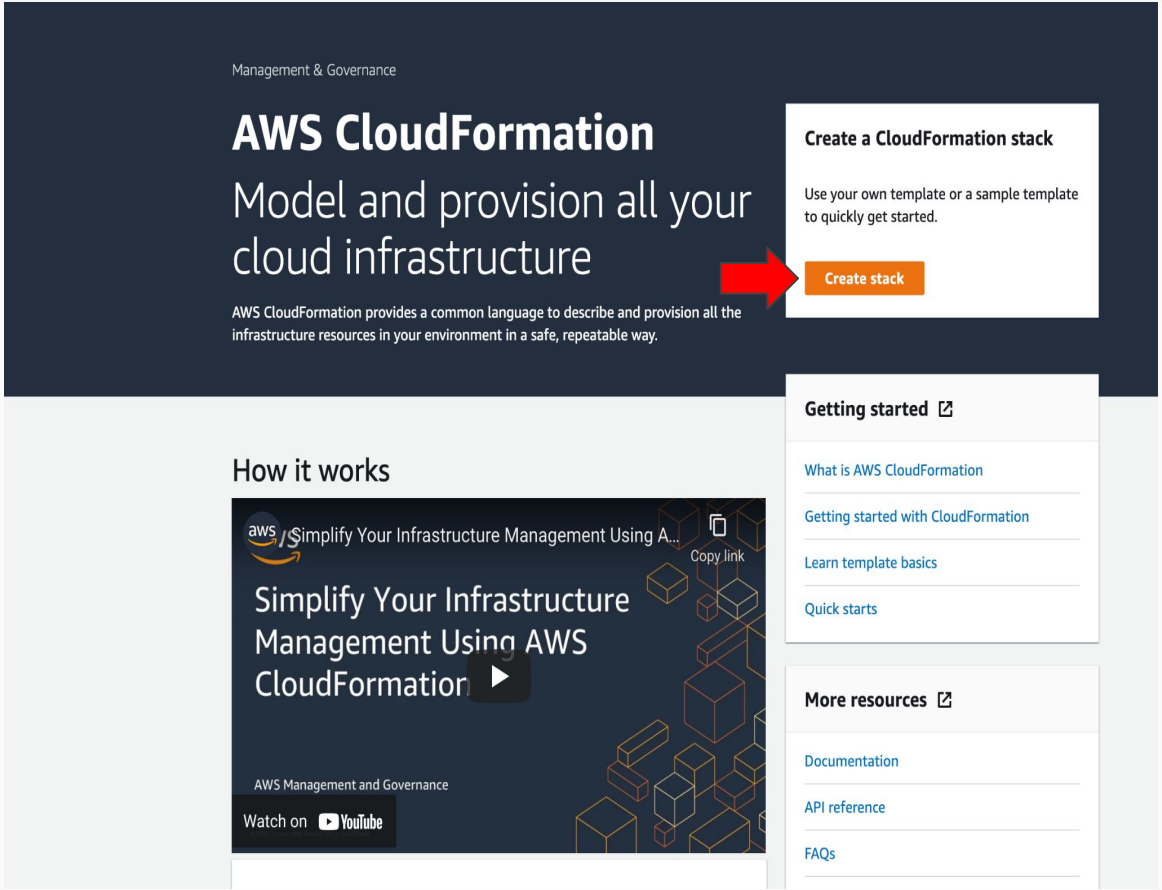
1. Use the AWS Canvas Live Events Connector CloudFormation template: **canvas-data-product-databrew.cfn.yaml**
2. If you wish to make any changes to the CloudFormation template (e.g., change resource names to match your naming conventions) do that at this time.
3. Log in to the AWS Management Console. Note, the user you are logging in with must have sufficient privileges to create CloudFormation stacks and the AWS resources needed for the Canvas Live Events Connector (list of resources below).
4. Navigate to CloudFormation

The CloudFormation stack will create the following resources:

- One Amazon S3 bucket
- One AWS Glue DataBrew project
- Four AWS Glue DataBrew datasets
- One IAM Role

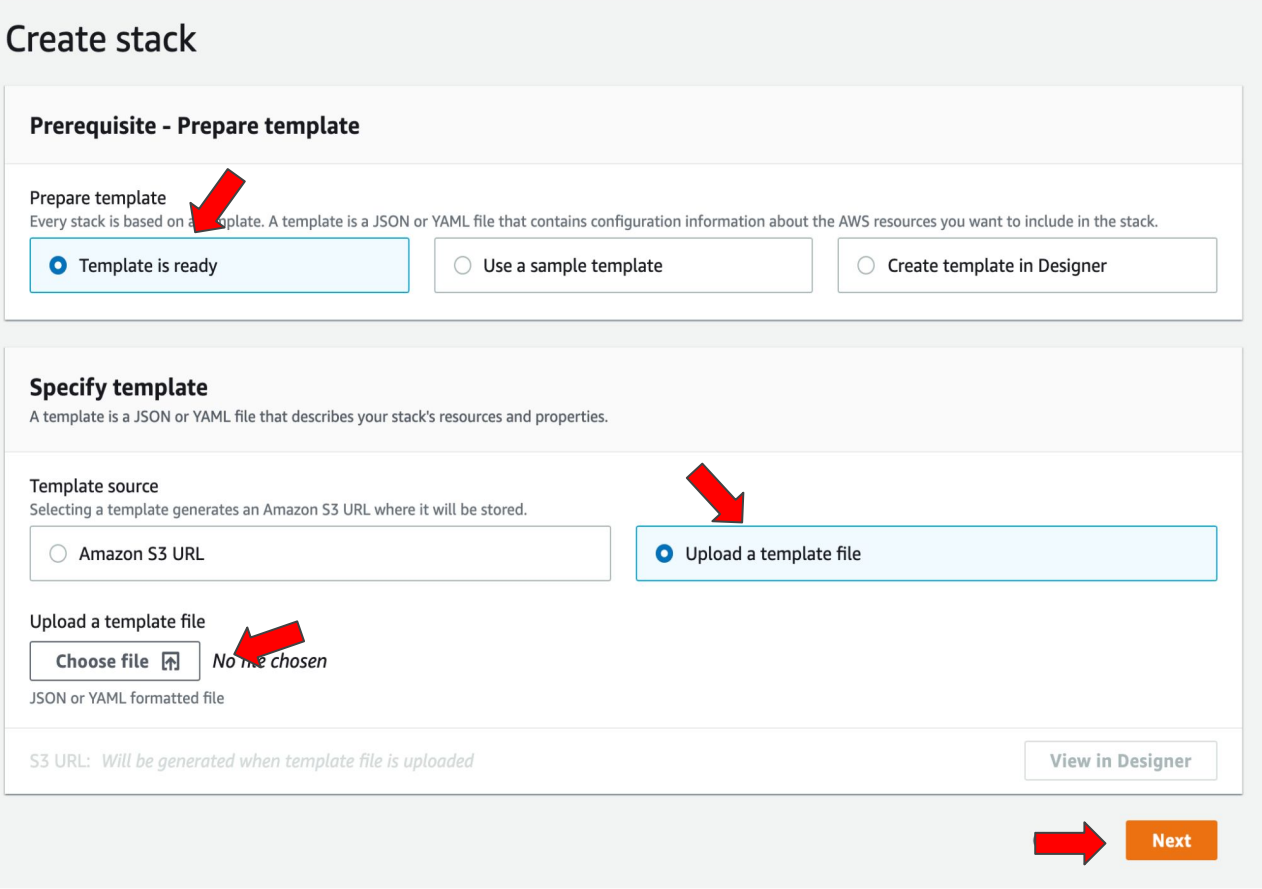
Step 2 - Create Stack

1. Click on the Create stack button



Step 3 - Create stack

1. Select “Template is ready”
2. Select “Upload a template file”
3. Click Choose file and then choose the yaml file that you downloaded at Step 1
4. Click Next

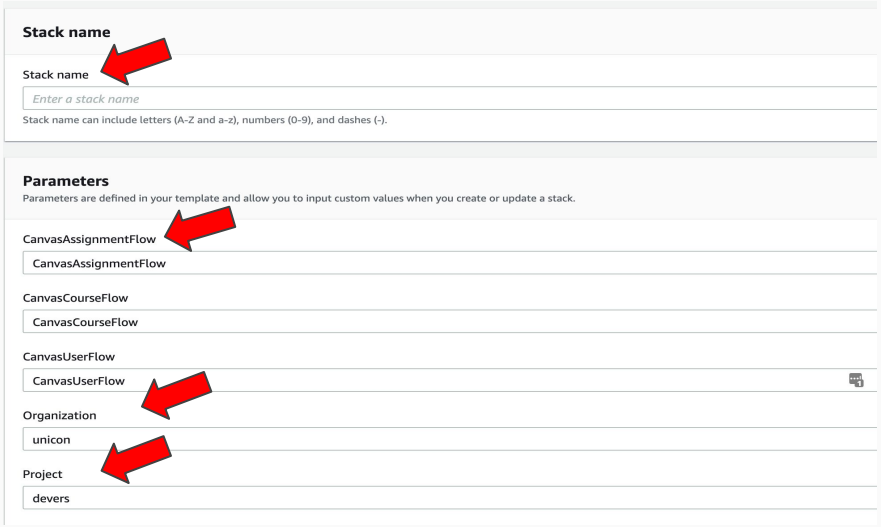


Step 4 - Specify stack details

The Organization and Project parameters are used as substitution variables in AWS resource names (e.g., Amazon S3 bucket - calpoly-qa-canvas-events). Organization should be a short abbreviation for the institution. Project should be a short abbreviation for the initiative or environment. All AWS resources are also tagged with the Project value.

The three *Flow parameters must match the names that you used when creating the Amazon AppFlows.

1. Enter a Stack name (e.g., CanvasDatabrewStack)
2. Leave the default values for the three Flow parameters (unless you used different names when creating the Amazon AppFlows - **they must match**)
3. Enter a value for the Organization parameter (e.g., calpoly)
4. Enter a value for the Project parameter (e.g., qa)
5. Click Next

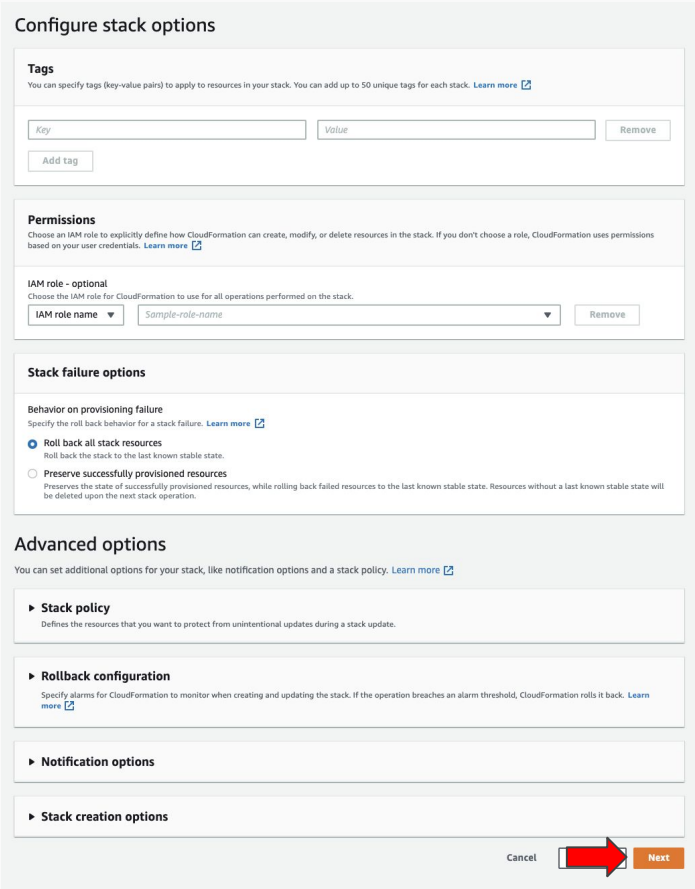


Step 5 - Configure stack options

No changes are required on this screen. However, if you have your own AWS best practices or conventions feel free to apply them now.

If you do not select IAM role to use, CloudFormation will create one for you.

1. Click Next

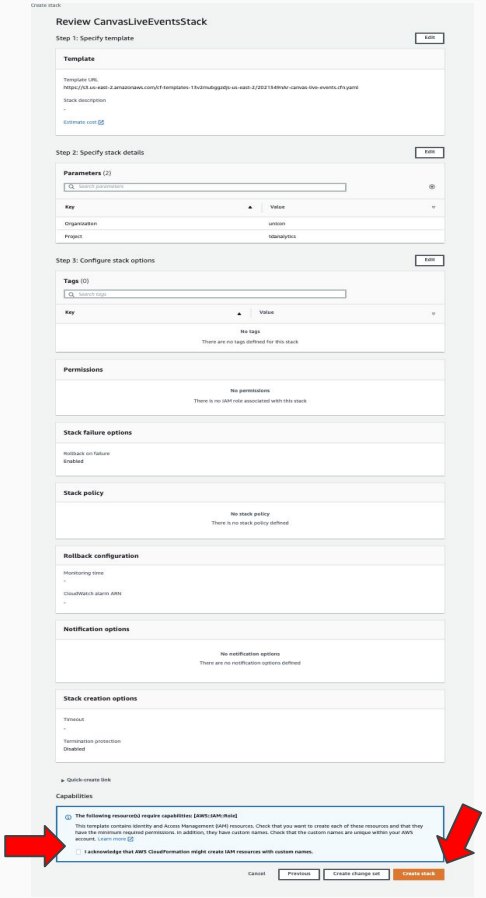


Step 6 - Review stack

Review the stack configuration.

1. Click the checkbox (I acknowledge that AWS CloudFormation might create IAM resources with custom names.) in the blue box
2. Click Create stack

This will initiate the process of creating all of the required resources. This process can take a couple of minutes.



You’re done! At this point all of the solution components are in place. Feel free to update the Glue Crawler, AppFlows or DataBrew job to run on a schedule.

Rolling Back

With the exception of the Amazon AppFlow components, the easiest way to rollback any parts of the solution is to delete the Cloudformation stack. This is one of the reasons we broke up the Cloudformation templates - so that you can manage each part of the solution independently.

With Amazon AppFlow, you will need to manually delete any of the components to remove them.