Project in AWS
Practice Lab

Creating and Assuming an Administrator AWS IAM Role

Andra-Diana Popescu

2025

ABOUT THIS LAB

In this Hands-on Lab we are going to work through creating a brand-new IAM Role within your AWS Sandbox account. This IAM Role will be granted Administrator Access permissions within the same account.

LEARNING OBJECTIVES

- Create IAM Role
- Assume the IAM Role
- Create & Deploy CloudFormation Template of IAM Role

AWS Documentation about IAM and CloudFormation:

https://aws.amazon.com/iam/faqs/

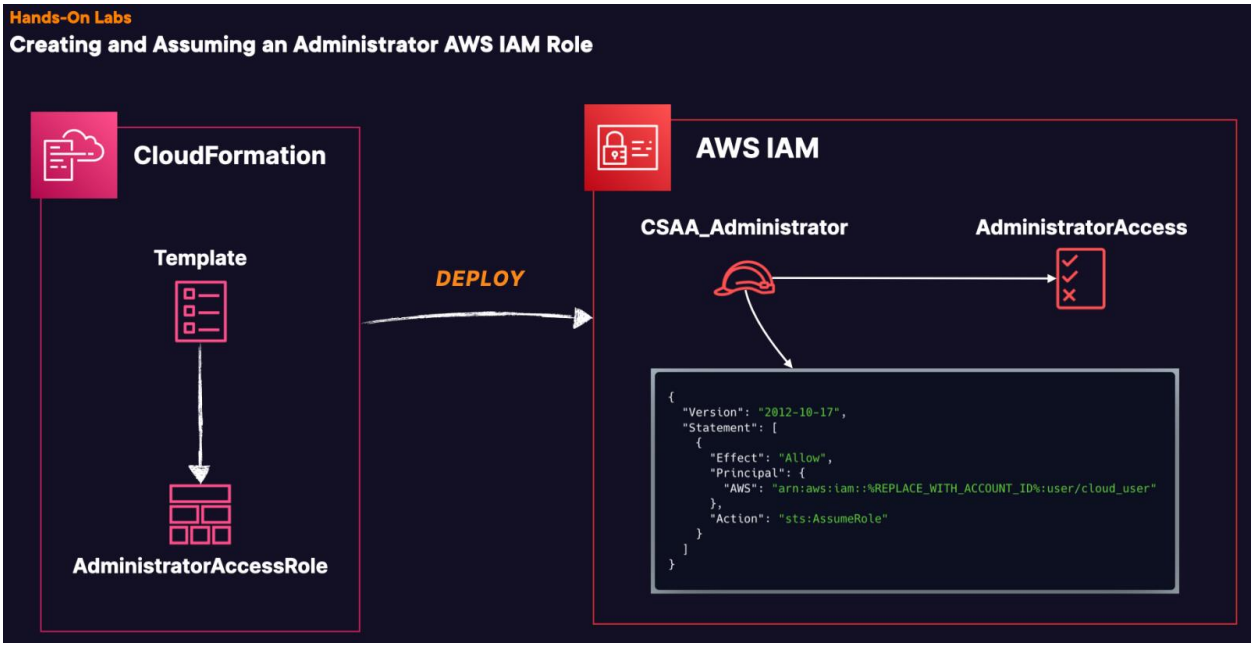
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html#policy-eval-denyallow

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html

Source: https://learn.acloud.guru/course/certified-solutions-architect-associate/

Table of Contents

Lab Diagram

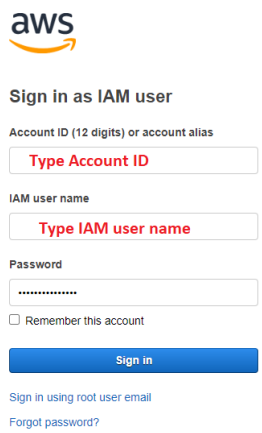


We have the AWS account in **us-east-1** Region. In this lab, we're going to create an IAM role, we're going to test that it works, and then we're going to create a CloudFormation template to automatically deploy that role in the future.

After we get through testing, we'll start by creating a template and deploying it to a new stack called ***AdministratorAccessRole***. Once deployed, what it's going to do is deploy a new IAM role (***CSAA_Administrator***) for us. That role is going to have the ***AdministratorAccess*** AWS managed policy attached. This role will have a custom trust policy in place where we are going to allow the cloud user to assume the role for testing.

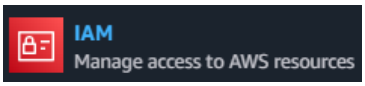
After all this, you can save this CloudFormation template for future use cases and future assignments or scenarios.

Log in to your AWS account



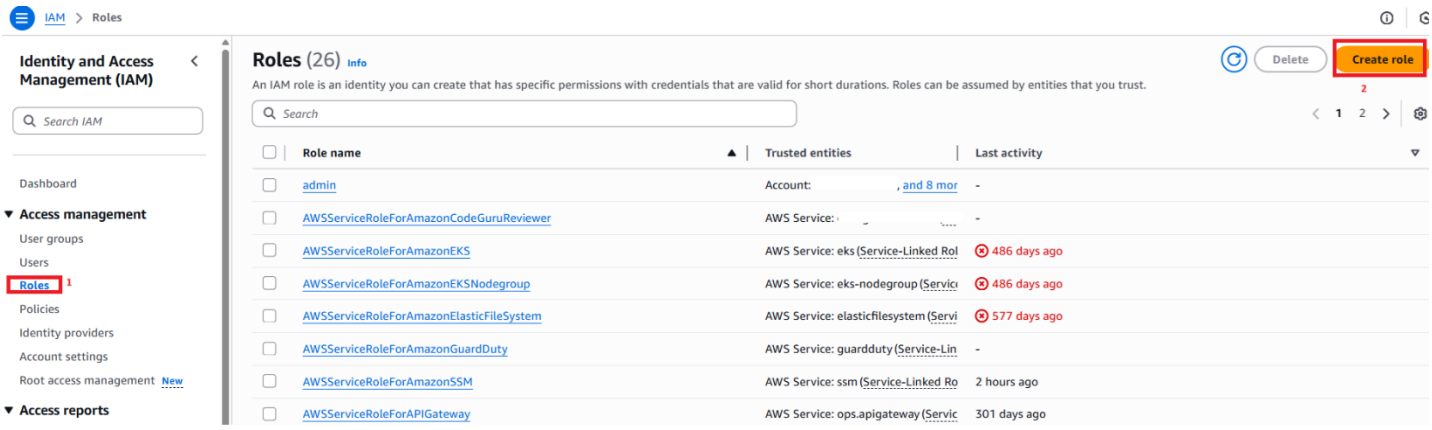
1. Create IAM Role

1. Once you are logged in to the AWS Management Console, navigate to **AWS Identity and Access Management (IAM)**.



2. Select **Roles** from the menu on the left.

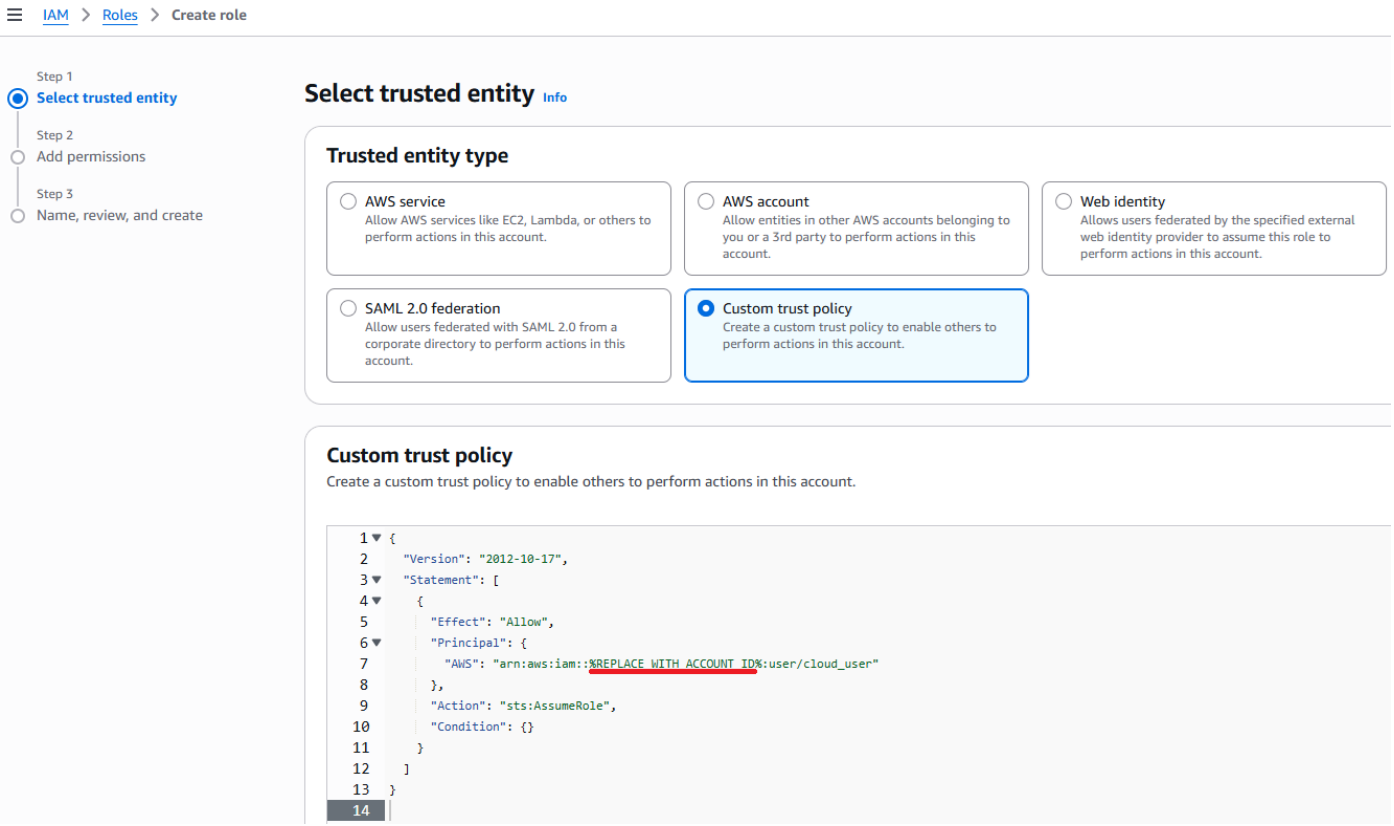
3. Click the orange **Create role** button.



4. Select **Custom trust policy** within ***Trusted entity type***.

5. Copy and paste the below JSON code. Be sure to replace the **%REPLACE_WITH_ACCOUNT_ID%** with your Account ID.

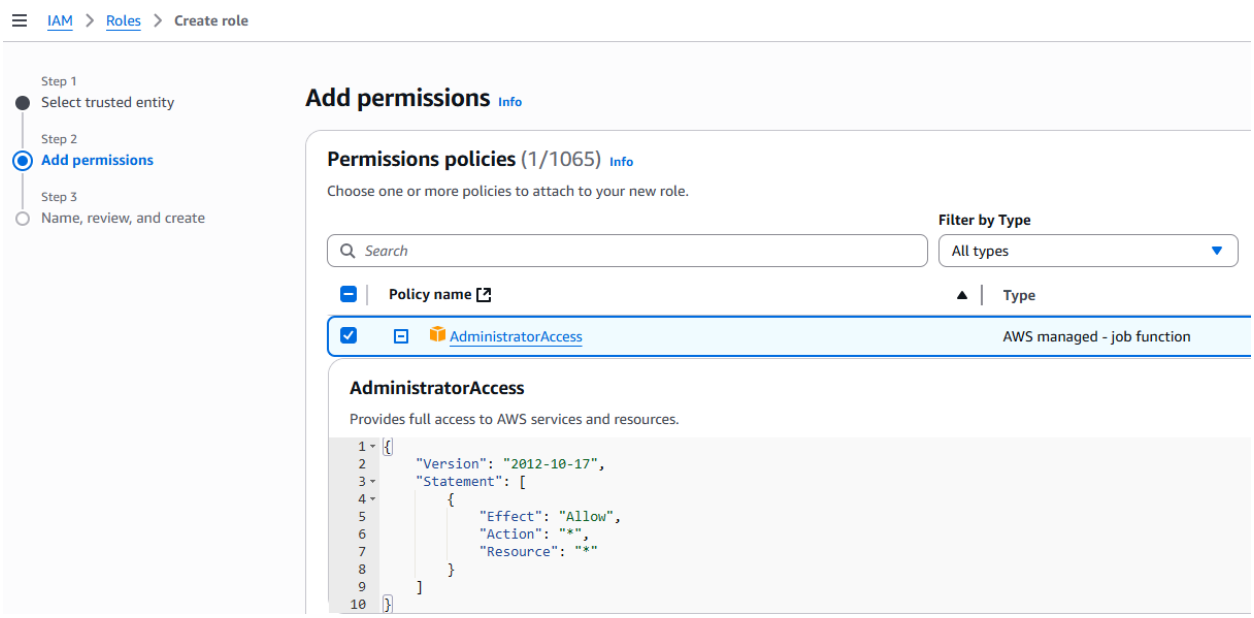
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::%REPLACE_WITH_ACCOUNT_ID%:user/cloud_user"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```



6. Select **Next**.

7. Under ***Add permissions*** select **AdministratorAccess** from the list of AWS-managed IAM policies.

8. Select **Next**.

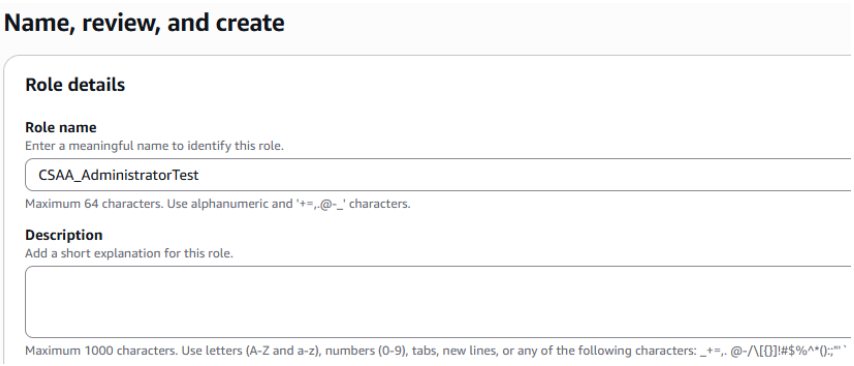


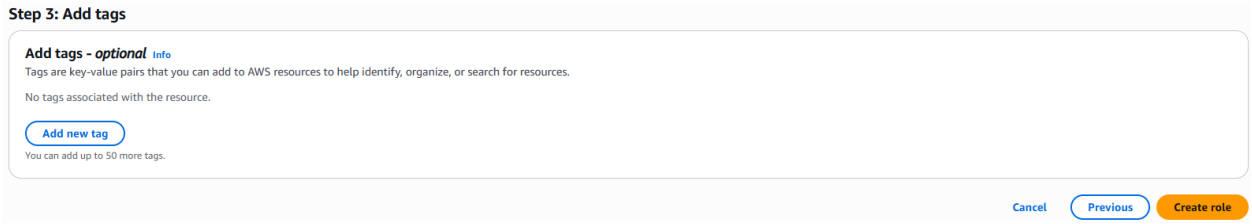
Note: You're going to see it has access to everything, this is obvious a restricted environment. Even though we're granting permissions for every resource and every action, this is going to be limited by 2 different things: ***Permissions boundaries*** (that are set on our IAM user, which allows the maximum amount of permissions that we can assign) and ***Service Control Policies*** (at the organizational level that you cannot avoid, which also limit our service usage). So, this is not truly opening everything up for this particular account, but if you did this in a brand new account, you would have full rights to every resource.

9. Under ***Role details***, for ***Role name*** enter: **CSAA_AdministratorTest**

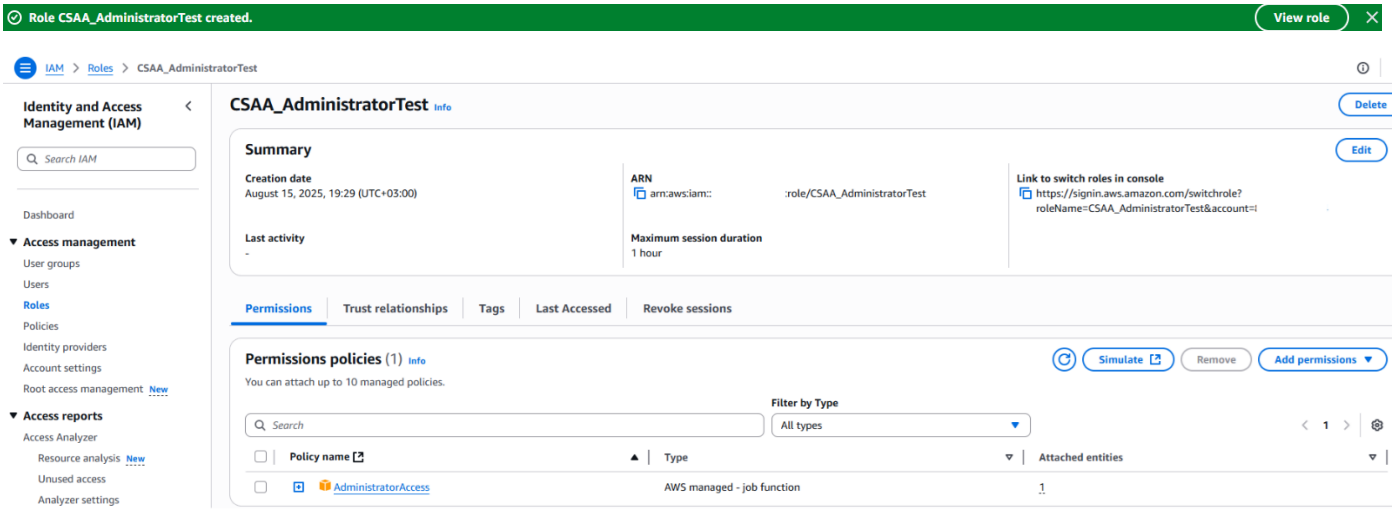
10. Under ***Role details***, for ***Description***, optionally enter your own description.

11. Review the details, then click **Create role**.





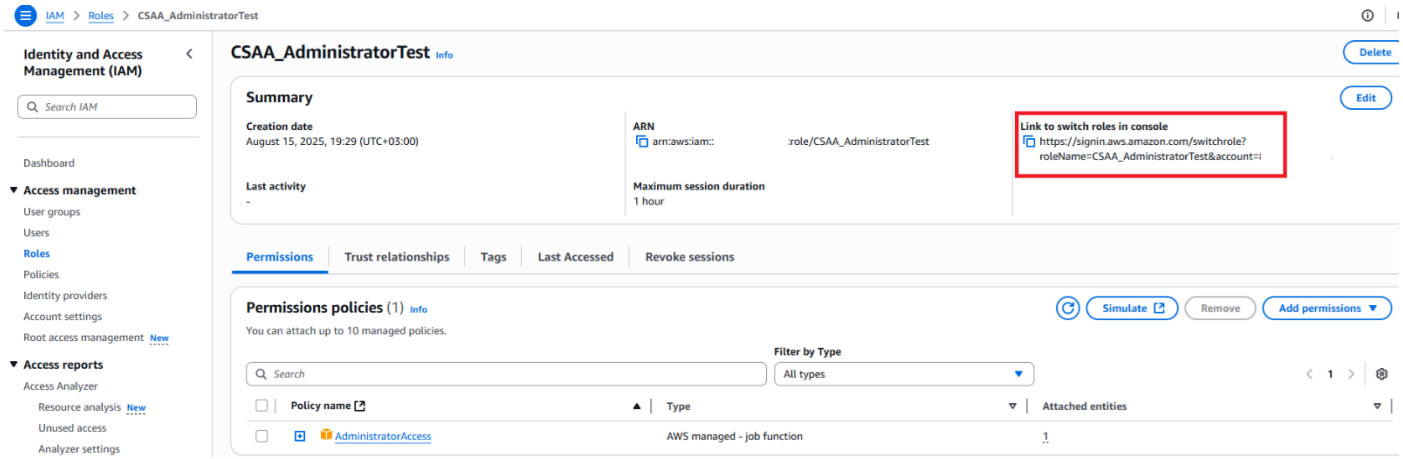
12. Click **View role**.



2. Assume the IAM Role

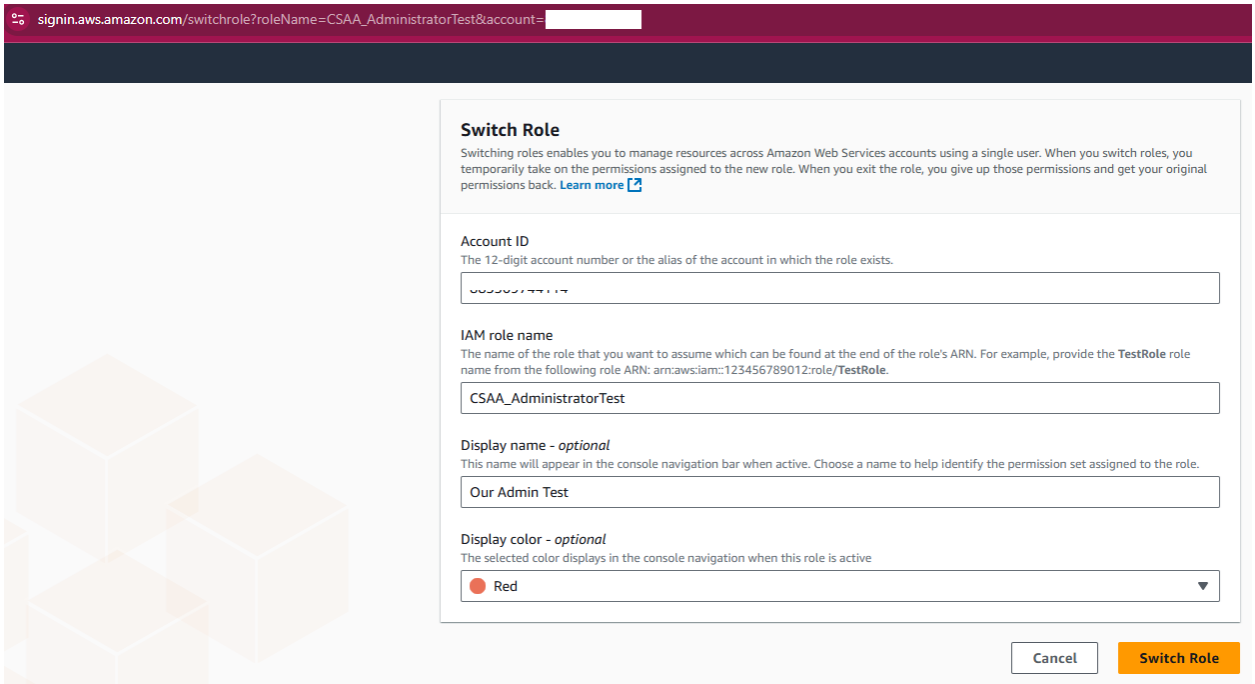
After creating the new IAM role, we need to assume it to test everything out.

1. If you don't already have your new role open from the previous **View role** step, find your new ***CSAA_AdministratorTest*** IAM role in the IAM roles list and select it.
2. Under the ***Summary*** section, find and copy the **Link to switch roles in console URL**.

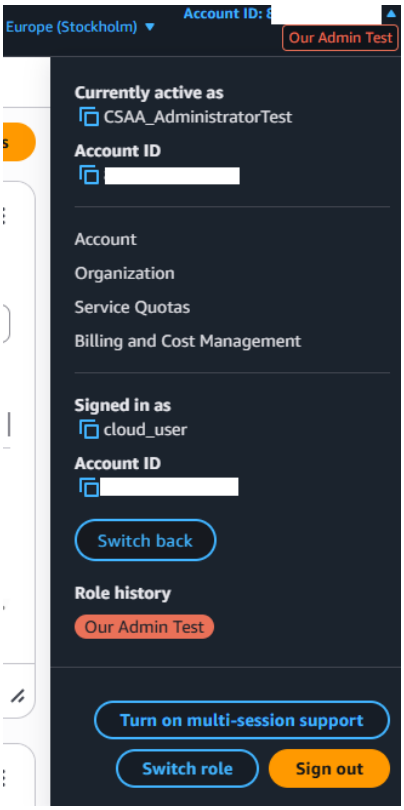


3. Open a new tab and navigate to the URL that was copied. The fields should be populated for you with the Account ID and Role ARN.

4. Optionally provide a ***Display name*** and choose a ***color***, then select **Switch Role**.



5. You should now be assuming the IAM role within the same AWS Account.

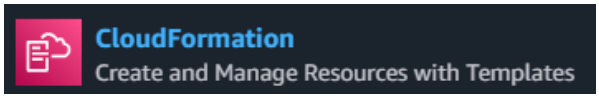


Note: In theory, we could perform all administrator actions for all services that are not restricted by permission boundaries or service control policies.

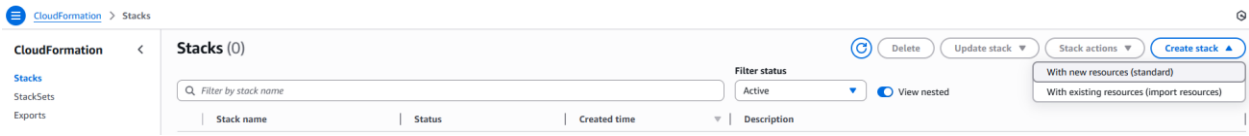
3. Create & Deploy CloudFormation Template of IAM Role

Now that our IAM role is verified to be working, let's codify the final version into a CloudFormation template!

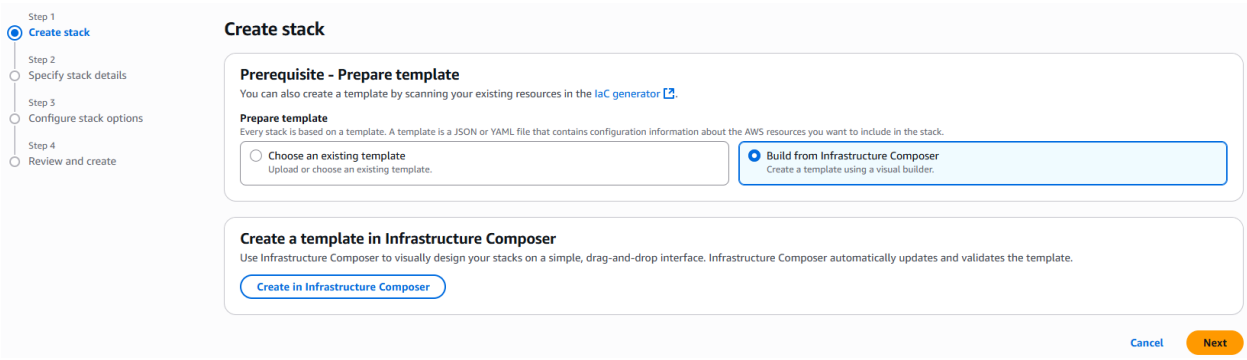
1. In a new tab, navigate to **CloudFormation**.



2. Under ***Stacks***, find and select **Create stack**.
3. Select **With new resources (standard)** from the dropdown menu.



4. Choose **Build from Application Composer** under ***Prerequisite – Prepare template***.
5. After that, click the button **Create in Application Composer**.



6. On the top portion of ***Application Composer***, click the **Template** button.
7. Select your desired template language (JSON or YAML) from the ***Choose template language*** toggle.
8. Paste the template code below for the language you chose into the console.
    a. **YAML Template Code**:

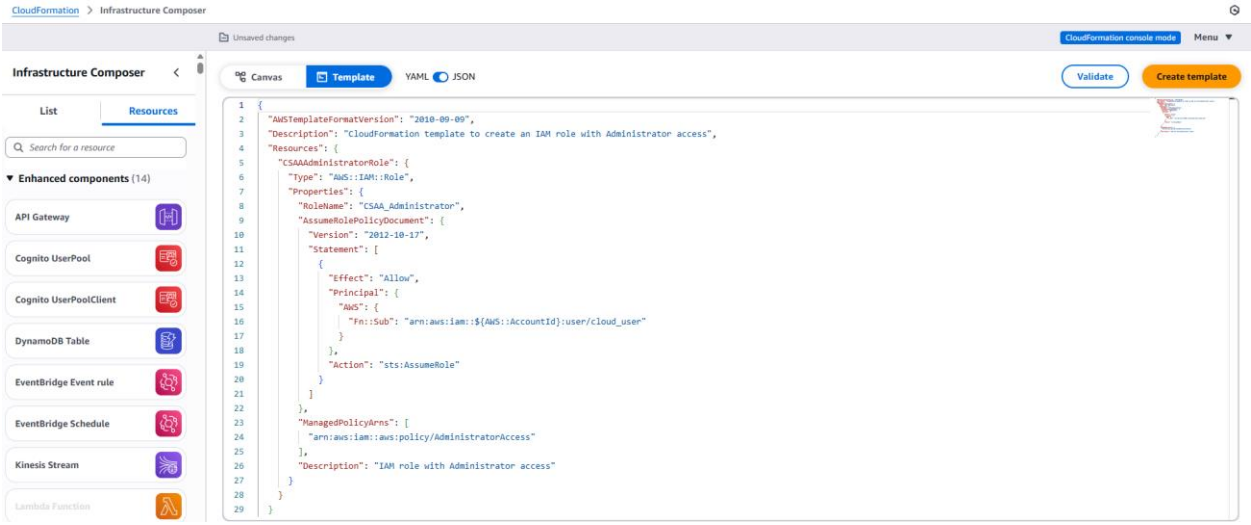
```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'CloudFormation template to create an IAM role with Administrator access'

Resources:
  CSAAAdministratorRole:
```

```
    Type: AWS::IAM::Role
    Properties:
      RoleName: CSAA_Administrator
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              AWS: !Sub 'arn:aws:iam::${AWS::AccountId}:user/cloud_user'
            Action: 'sts:AssumeRole'
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/AdministratorAccess'
      Description: 'IAM role with Administrator access'
```

b. **JSON Template Code**:

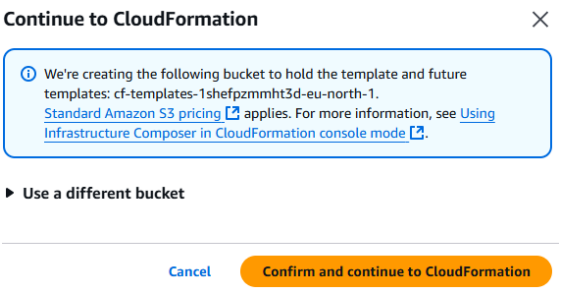
```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "CloudFormation template to create an IAM role with Administrator
access",
  "Resources": {
    "CSAAAdministratorRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": "CSAA_Administrator",
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "AWS": {
                  "Fn::Sub": "arn:aws:iam::${AWS::AccountId}:user/cloud_user"
                }
              },
              "Action": "sts:AssumeRole"
            }
          ]
        },
        "ManagedPolicyArns": [
          "arn:aws:iam::aws:policy/AdministratorAccess"
        ],
        "Description": "IAM role with Administrator access"
      }
    }
  }
}
```

9. Click **Create Template**.

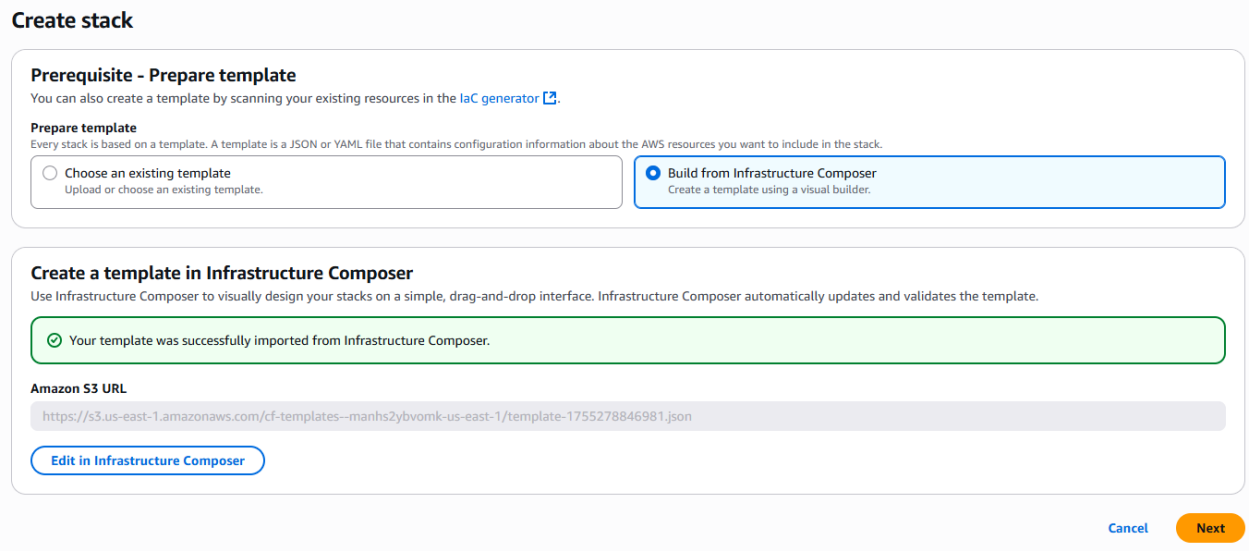


Note: You will notice the pseudo parameter (***${AWS::AccountId}***), this allows you to easily reference certain values within the template. This is allowing us to automatically reference the account ID for any account that this template is deployed in. This will automatically infer the value and insert it into the template.

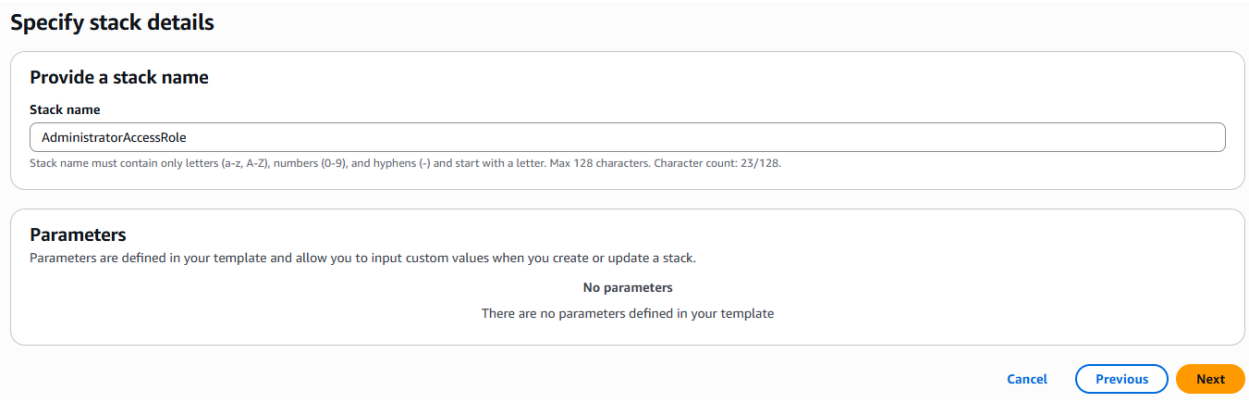
10. Accept the **Transfer bucket name** by clicking **Confirm and continue to CloudFormation**.



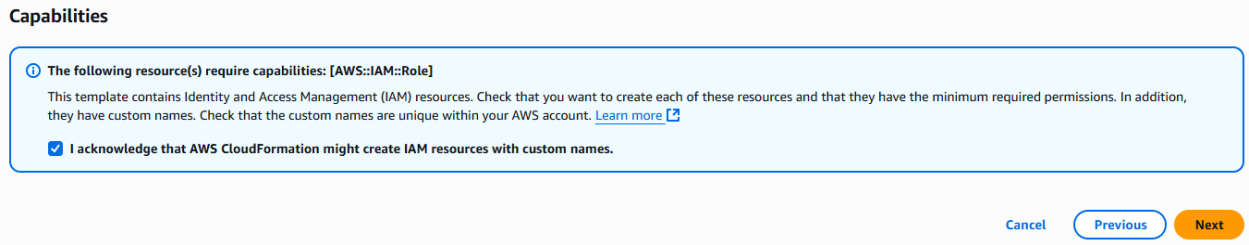
11. Find and select **Next**.



12. Enter ***AdministratorAccessRole*** as the stack name.
13. Select **Next**.
14. Optionally provide tags for the template if desired, and then select **Next**.



15. On the bottom of the review screen find and select the checkbox to ***acknowledge that AWS CloudFormation might create IAM resources with custom names***.



16. Select **Submit**.
17. Your template should deploy your new IAM role for future use! Click the refresh button until stack status shows **CREATE_COMPLETE**.

AdministratorAccessRole

Delete | Update stack | Stack actions | Create stack

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

Table view | Timeline view

Events (5)

View root cause

Search events

Timestamp	Logical ID	Status	Detailed status	Status reason
2025-08-15 20:31:08 UTC+0300	AdministratorAccessRole	CREATE_COMPLETE	-	-
2025-08-15 20:31:08 UTC+0300	CSAAAdministratorRole	CREATE_COMPLETE	-	-
2025-08-15 20:30:50 UTC+0300	CSAAAdministratorRole	CREATE_IN_PROGRESS	-	Resource creation Initiated
2025-08-15 20:30:49 UTC+0300	CSAAAdministratorRole	CREATE_IN_PROGRESS	-	-
2025-08-15 20:30:47 UTC+0300	AdministratorAccessRole	CREATE_IN_PROGRESS	-	User Initiated