

**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

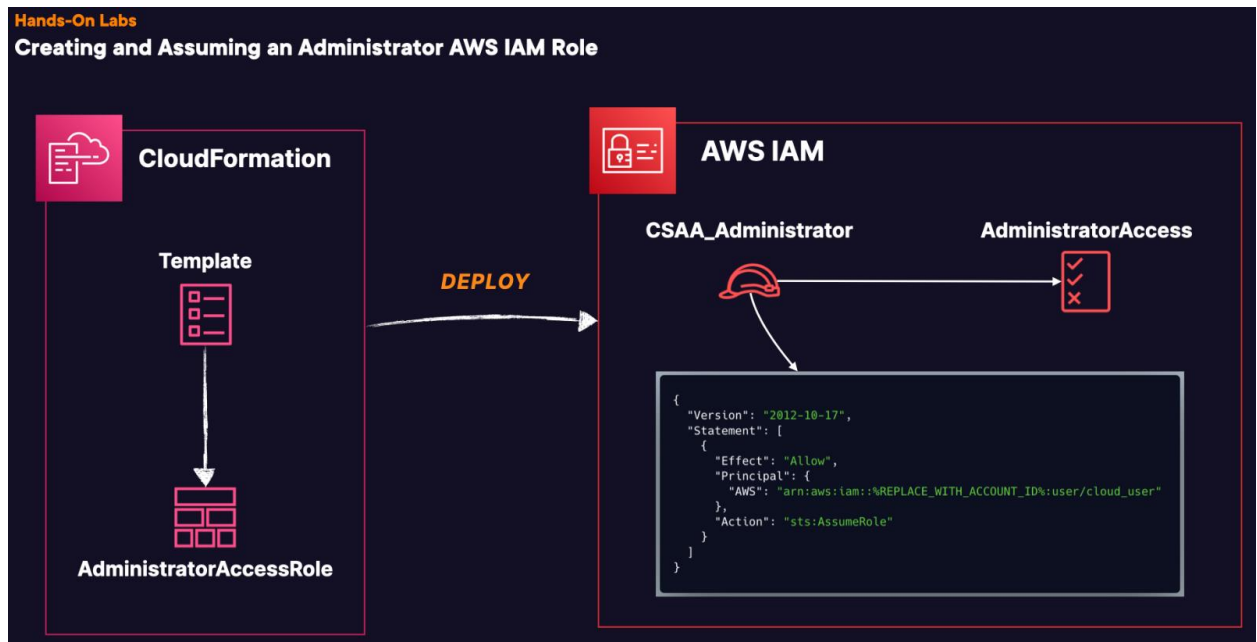
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html>

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

Table of Contents

Lab Diagram	4
Log in to your AWS account	5
1. Create IAM Role.....	5
2. Assume the IAM Role	8
3. Create & Deploy CloudFormation Template of IAM Role	10

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



Sign in as IAM user

Account ID (12 digits) or account alias

Type Account ID

IAM user name

Type IAM user name

Password

☐ Remember this account

Sign in

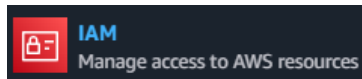
Sign in using root user email

[Forgot password?](#)



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.

Roles (26) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
admin	Account: , and 8 mor	-
AWSServiceRoleForAmazonCodeGuruReviewer	AWS Service:	-
AWSServiceRoleForAmazonEKS	AWS Service: eks (Service-Linked Rol	486 days ago
AWSServiceRoleForAmazonEKSNodegroup	AWS Service: eks-nodegroup (Servi	486 days ago
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Servi	577 days ago
AWSServiceRoleForAmazonGuardDuty	AWS Service: guardduty (Service-Lin	-
AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Ro	2 hours ago
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Servic	301 days ago

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": {}
    }
  ]
}
```

☰ IAM > Roles > Create role

- Step 1
☒ **Select trusted entity**
- Step 2
☐ Add permissions
- Step 3
☐ Name, review, and create

Select trusted entity [Info](#)

Trusted entity type

- ☐ **AWS service**
 Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ☐ **AWS account**
 Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ☐ **Web identity**
 Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ☐ **SAML 2.0 federation**
 Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ☒ **Custom trust policy**
 Create a custom trust policy to enable others to perform actions in this account.

Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "AWS": "arn:aws:iam::%REPLACE_WITH_ACCOUNT_ID%:user/cloud_user"
8       },
9       "Action": "sts:AssumeRole",
10      "Condition": {}
11    }
12  ]
13 }
14
```

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

IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Add permissions Info

Permissions policies (1/1065) Info

Choose one or more policies to attach to your new role.

Filter by Type
All types

<input checked="" type="checkbox"/>	Policy name <small>Info</small>	Type
<input checked="" type="checkbox"/>	AdministratorAccess	AWS managed - job function

AdministratorAccess

Provides full access to AWS services and resources.

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "*",
7       "Resource": "*"
8     }
9   ]
10 }

```

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**.

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+', '@', '-', '_' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+@, @-/\[\]!#\$%^&*()~`~`

Step 3: Add tags

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

[Cancel](#)

[Previous](#)

[Create role](#)

12. Click **View role**.

Role CSAA_AdministratorTest created. [View role](#)

Identity and Access Management (IAM)

Dashboard

▼ Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

▼ Access reports

- Access Analyzer
- Resource analysis [New](#)
- Unused access
- Analyzer settings

CSAA_AdministratorTest [Info](#)

Summary

Creation date: August 15, 2025, 19:29 (UTC+03:00)

ARN: [arn:aws:iam:::role/CSAA_AdministratorTest](#)

Link to switch roles in console: https://signin.aws.amazon.com/switchrole?roleName=CSAA_AdministratorTest&account=...

Last activity: -

Maximum session duration: 1 hour

Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AdministratorAccess	AWS managed - job function	1

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**.

Identity and Access Management (IAM)

Dashboard

▼ Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

▼ Access reports

- Access Analyzer
- Resource analysis [New](#)
- Unused access
- Analyzer settings

CSAA_AdministratorTest [Info](#)

Summary

Creation date: August 15, 2025, 19:29 (UTC+03:00)

ARN: [arn:aws:iam:::role/CSAA_AdministratorTest](#)

Link to switch roles in console: https://signin.aws.amazon.com/switchrole?roleName=CSAA_AdministratorTest&account=...

Last activity: -

Maximum session duration: 1 hour

Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

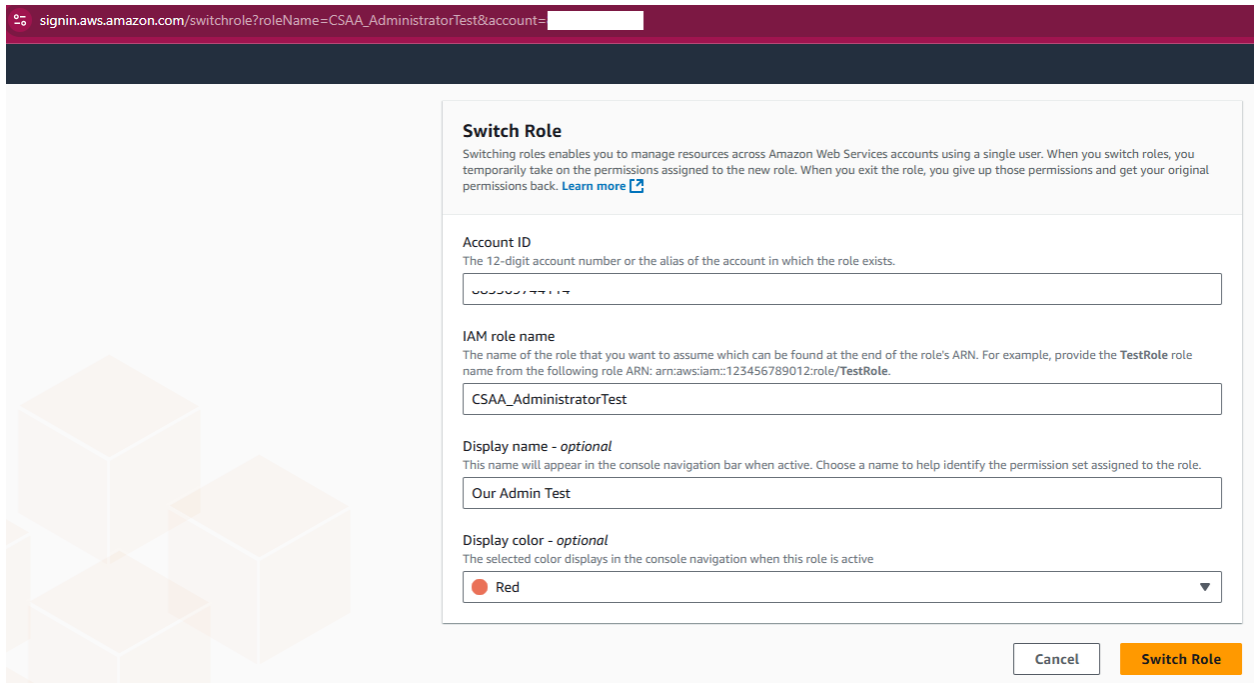
Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AdministratorAccess	AWS managed - job function	1

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**.



signin.aws.amazon.com/switchrole?roleName=CSAA_AdministratorTest&account=

Switch Role

Switching roles enables you to manage resources across Amazon Web Services accounts using a single user. When you switch roles, you temporarily take on the permissions assigned to the new role. When you exit the role, you give up those permissions and get your original permissions back. [Learn more](#)

Account ID
The 12-digit account number or the alias of the account in which the role exists.

IAM role name
The name of the role that you want to assume which can be found at the end of the role's ARN. For example, provide the **TestRole** role name from the following role ARN: `arn:aws:iam::123456789012:role/TestRole`.

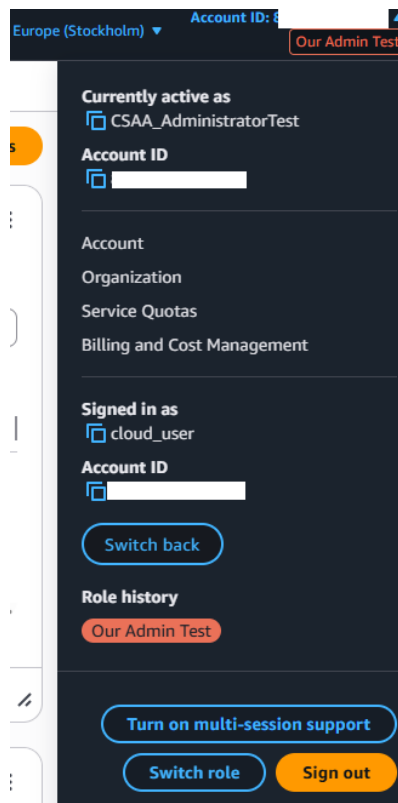
Display name - optional
This name will appear in the console navigation bar when active. Choose a name to help identify the permission set assigned to the role.

Display color - optional
The selected color displays in the console navigation when this role is active

☐ Red

Cancel Switch Role

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

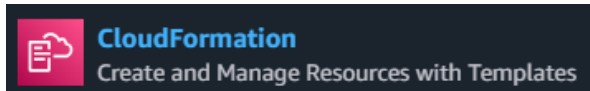


Note: In theory, we could perform all administrator actions for all services that are not restricted by permission boundaries or service control policies.

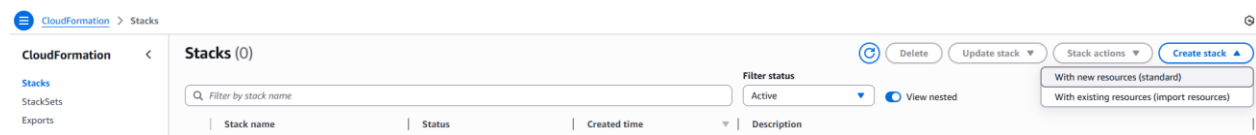
3. Create & Deploy CloudFormation Template of IAM Role

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

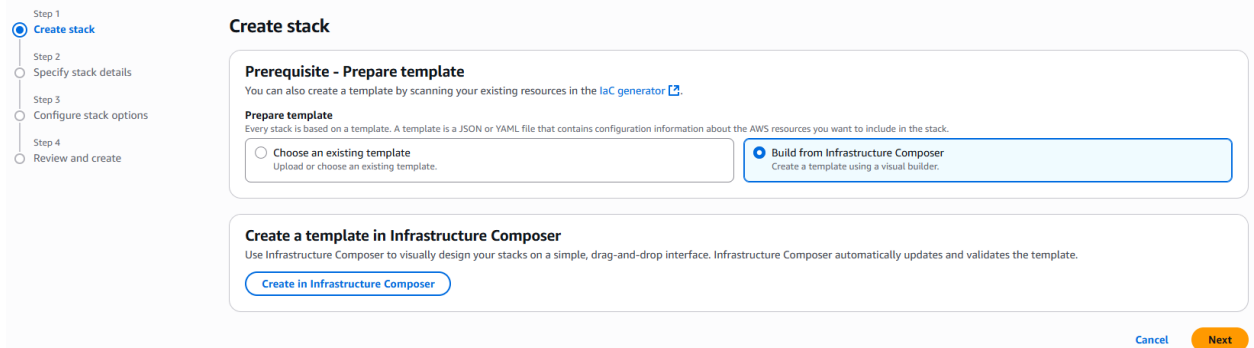
1. In a new tab, navigate to **CloudFormation**.



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:

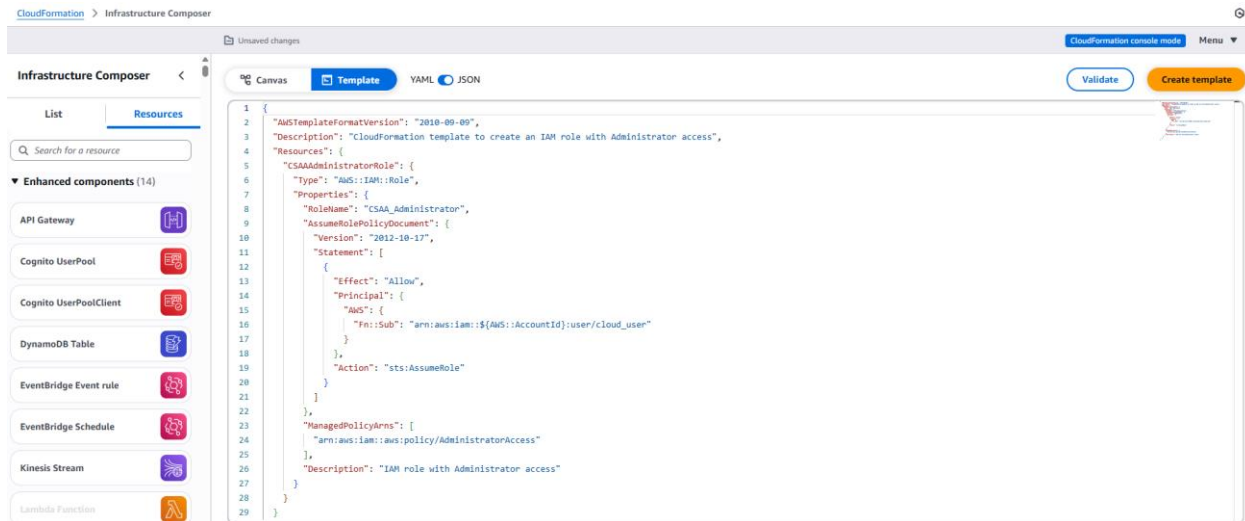
- 'arn:aws:iam::aws:policy/AdministratorAccess'

Description: 'IAM role with Administrator access'

b. JSON Template Code:

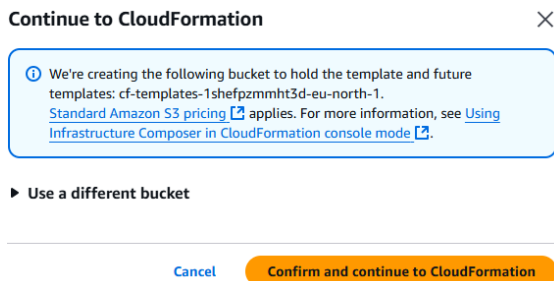
```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "CloudFormation template to create an IAM role with Administrator
access",
  "Resources": {
    "CSAAAdministratorRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": "CSAA_Administrator",
        "AssumeRolePolicyDocument": {
          "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.

Create stack

Prerequisite - Prepare template

You can also create a template by scanning your existing resources in the [IaC generator](#).

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☐ Choose an existing template

Upload or choose an existing template.

☒ Build from Infrastructure Composer
Create a template using a visual builder.

Create a template in Infrastructure Composer

Use Infrastructure Composer to visually design your stacks on a simple, drag-and-drop interface. Infrastructure Composer automatically updates and validates the template.

✓ Your template was successfully imported from Infrastructure Composer.

Amazon S3 URL

<https://s3.us-east-1.amazonaws.com/cf-templates--manhs2ybvomk-us-east-1/template-1755278846981.json>

[Edit in Infrastructure Composer](#)

[Cancel](#)

[Next](#)

12. Enter *AdministratorAccessRole* as the stack name.

13. Select **Next**.

14. Optionally provide tags for the template if desired, and then select **Next**.

Specify stack details

Provide a stack name

Stack name

AdministratorAccessRole

Stack name must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 23/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters

There are no parameters defined in your template

[Cancel](#)

[Previous](#)

[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*.

Capabilities

① The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

[Cancel](#)

[Previous](#)

[Next](#)

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**.

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_PROGRES S	-	Resource creation Initiated
2025-08-15 20:30:49 UTC+0300	CSAAAdministratorRole	ⓘ CREATE_IN_PROGRES S	-	-
2025-08-15 20:30:47 UTC+0300	AdministratorAccessRole 	ⓘ CREATE_IN_PROGRES S	-	User Initiated