Cloudockit's Optimal Setup - Enterprise - AWS

Before you start, let us help you navigate all of the steps to Cloudockit Optimal Setup. Simply book a call with one of our experts.

Book a call

Introduction

The purpose of this document is to provide the detailed steps to install and configure Cloudockit Desktop in an optimal way so you can get going as quickly as possible with your automated documentation generation for your AWS environment.

Cloudockit desktop can be installed in many ways. On a workstation, on a server, on a virtual machine.

Based on our experience we have identified that the optimal way is to create an EC2 instance and install Cloudockit desktop to automate your document generation.

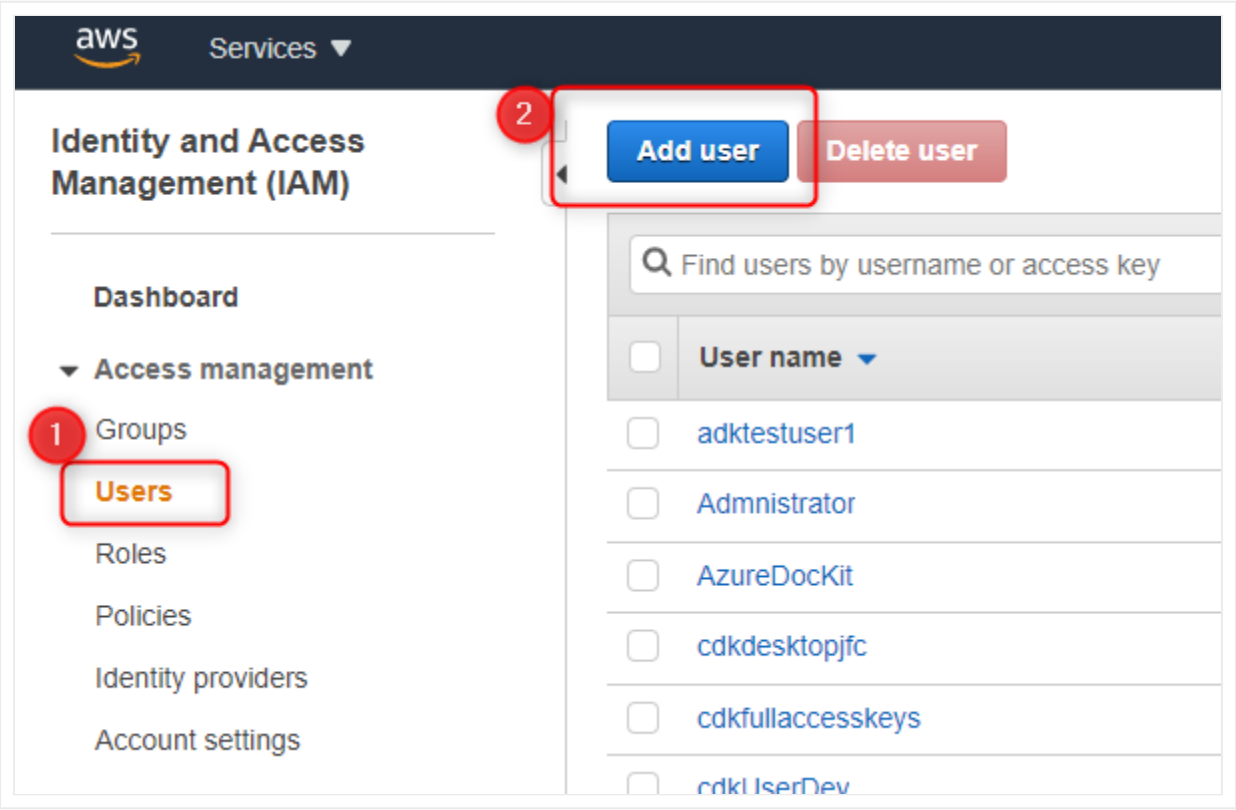
Step 1 – Create the IAM User

Your chosen IAM User will be used to list all of the accounts in the organization to enable Cloudockit Desktop to loop through the accounts and assume roles in each of them.

Create the User

Sign in to the AWS Console and open the IAM console: Amazon IAM

In the navigation panel, choose **Users**, press **Add** user.



Set User details and Select AWS access type

Enter a name

In Access Type, check **Programmatic access**

Click: Permissions

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* CDKOptimalMultiAccountScan

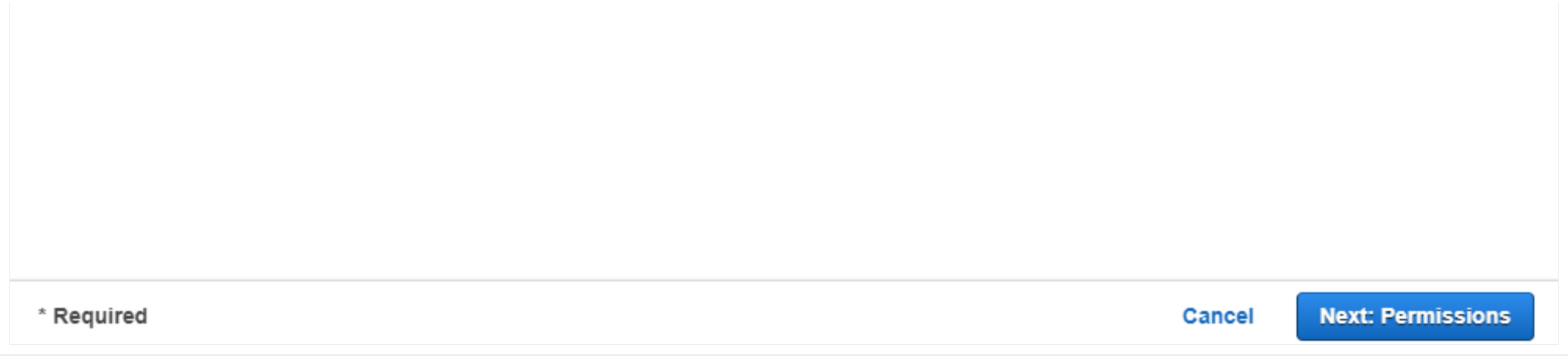
Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* ☑ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

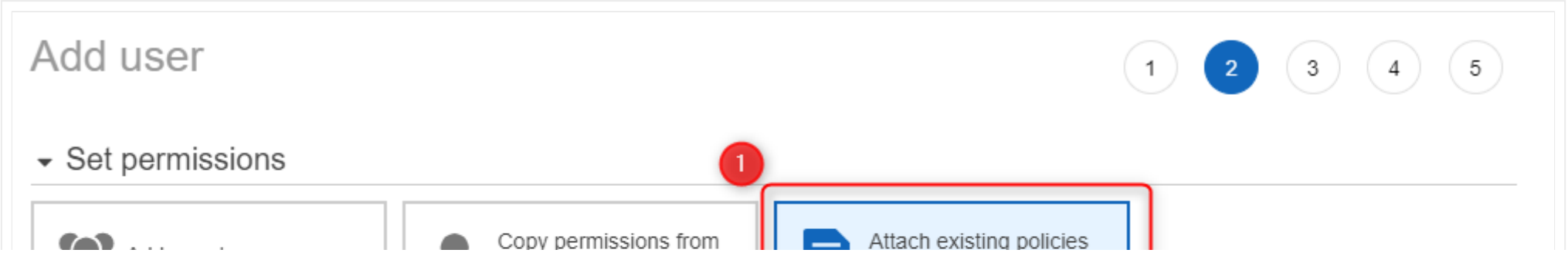
☐ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

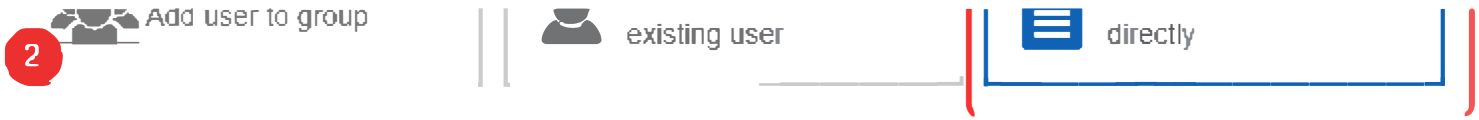


Create the Policy

Select **Attach existing policies directly.**

Click: Create Policy





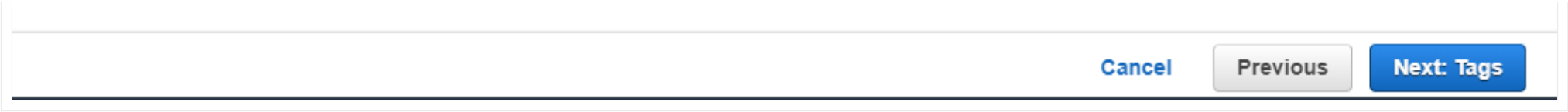
Create policy

Filter policies | Search

Showing 646 results

Policy name		Used as
accessToUmaknowAccount	Customer managed	Permissions policy (1)
adkpascaltest1	Customer managed	*None*
AdministratorAccess	Job function	Permissions policy (23)
AlexaForBusinessDeviceSetup	AWS managed	Boundary (1)
AlexaForBusinessFullAccess	AWS managed	*None*
AlexaForBusinessGatewayExecution	AWS managed	*None*
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	*None*
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	*None*

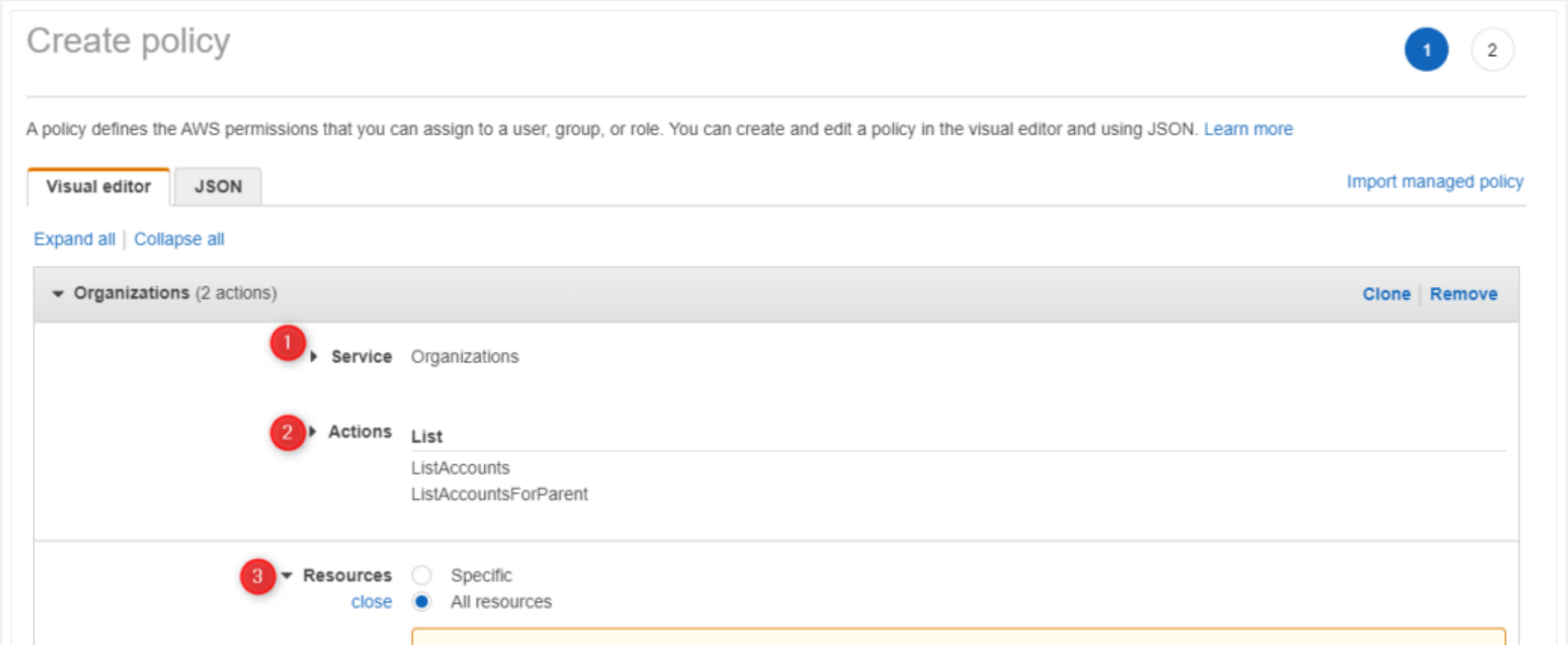
▸ Set permissions boundary

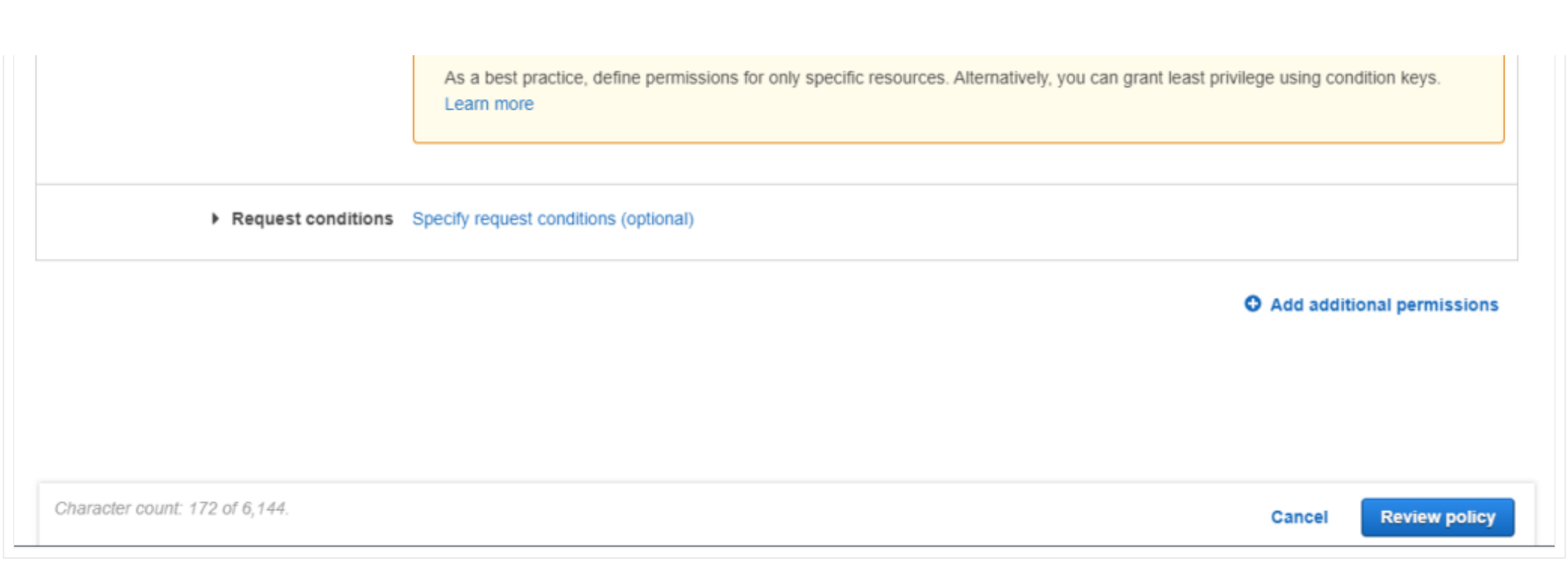


Make the following selections:

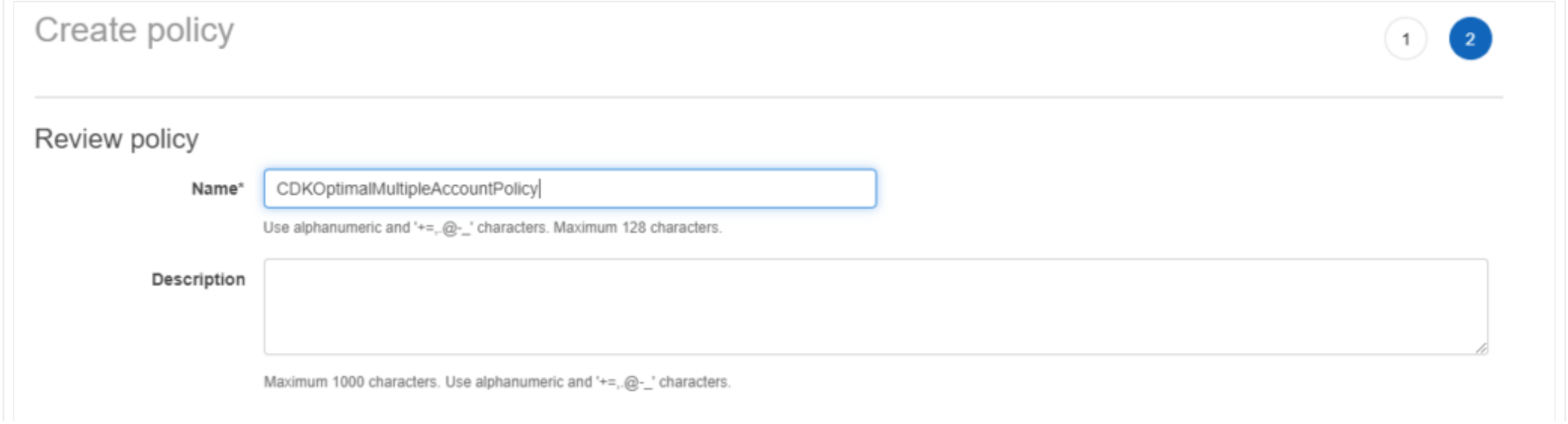
- In Service, select **Organizations**
- In action, select **Access Level** / **List Accounts and ListAccountsForParent**
- In Resources, select **All resources**

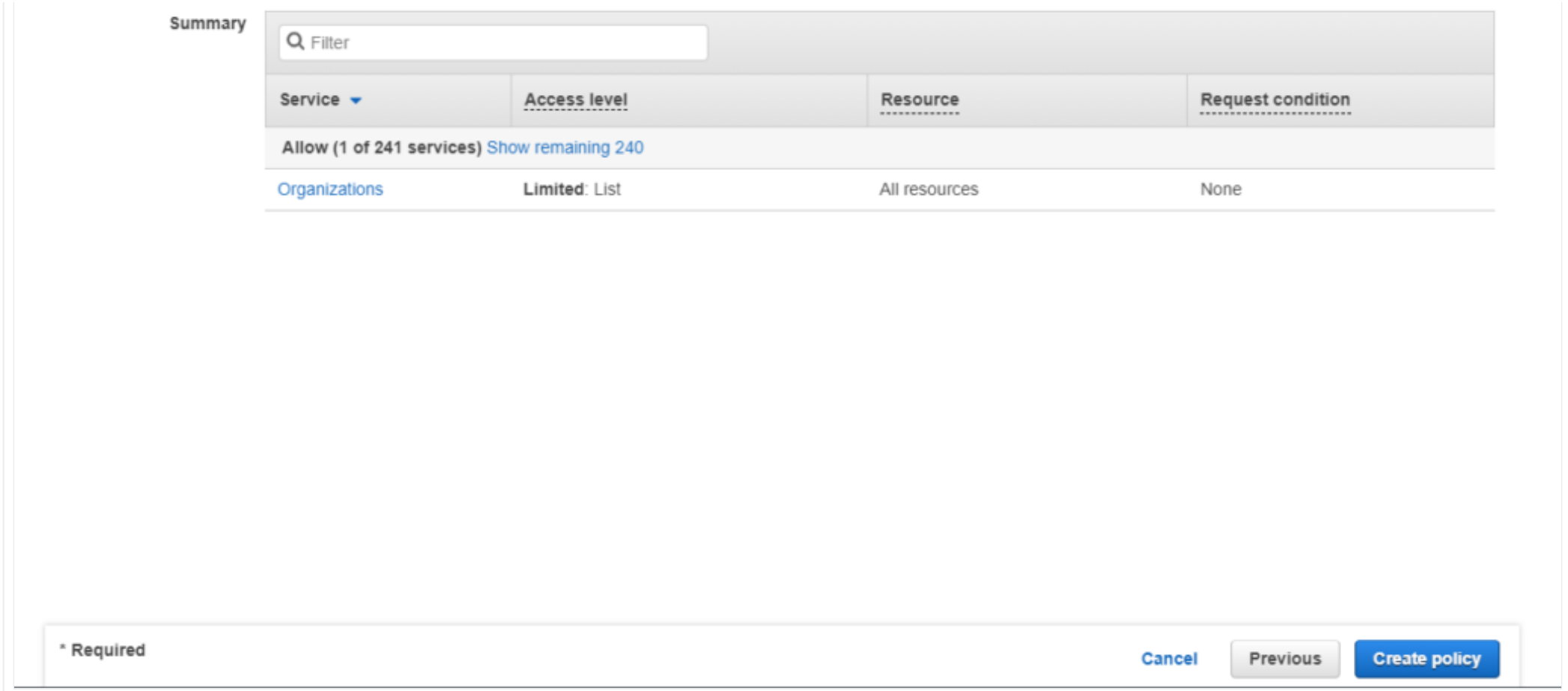
Click: Review Policy





Give your Policy a unique name and press **Create Policy**.



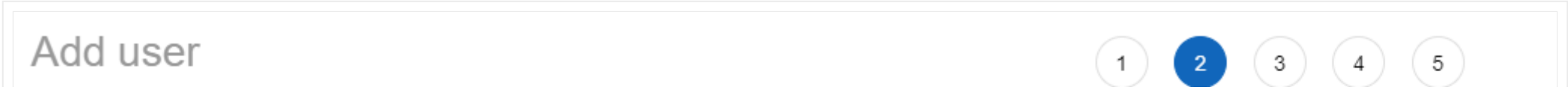


Close the opened tab to create the new policy.

Go back to the user creation screen and refresh the list.

Select the newly created policy

Click: Tags



Set permissions

Add user to group | Copy permissions from existing user | Attach existing policies directly

1

Create policy

Filter policies | cdk | Showing 5 results

	Policy name	Type	Used as
☐	CDK-NoPermissionCostAndUsageReport	Customer managed	*None*
☐	cdkCostExplorerServiceMissingPermissions	Customer managed	Permissions policy (1)
☐	CDKOptimalMultipleAccountPolicy	Customer managed	*None*
☐	CDKOptimalSetupPolicy	Customer managed	*None*
☐	kinesis-analytics-service-cdkKinesisAnalytic-us-east-1	Customer managed	Permissions policy (1)

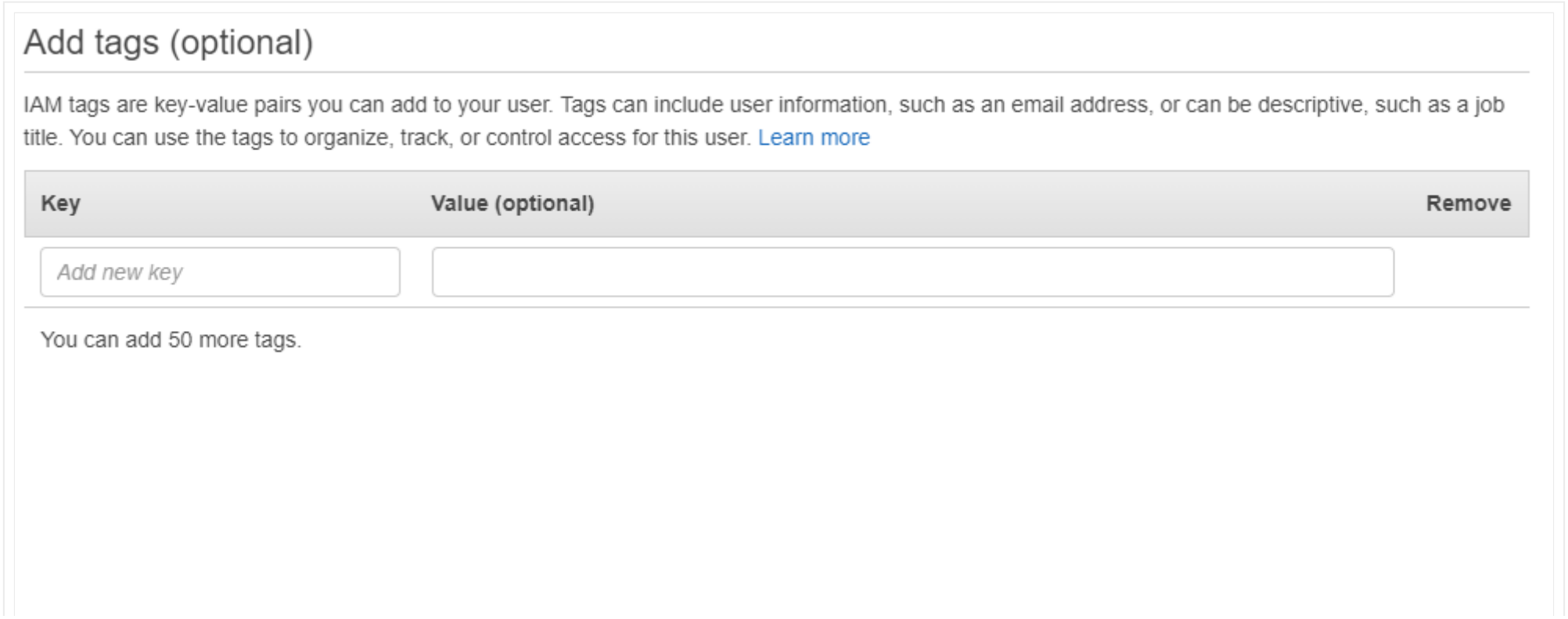
2

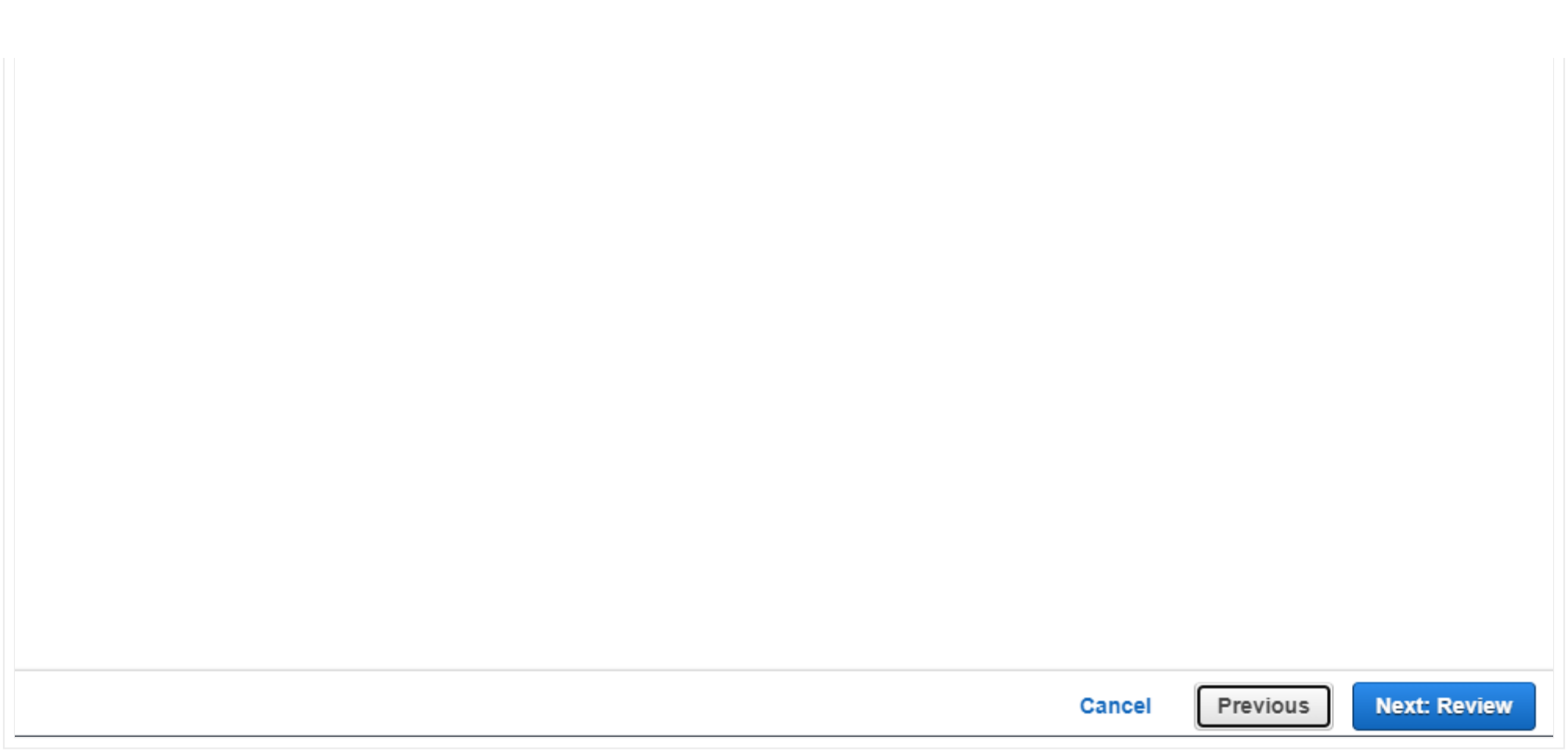
Set permissions boundary

Add Tags

Add tags based on your organization's policies.

Click: Review





Review

Review the parameters and create a user.

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	CDKOptimalMultiAccountScan
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

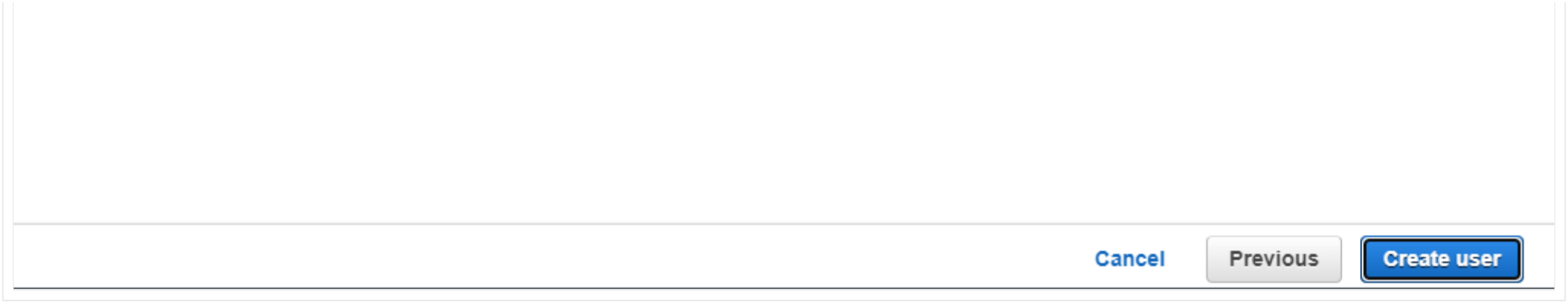
Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	CDKOptimalMultipleAccountPolicy

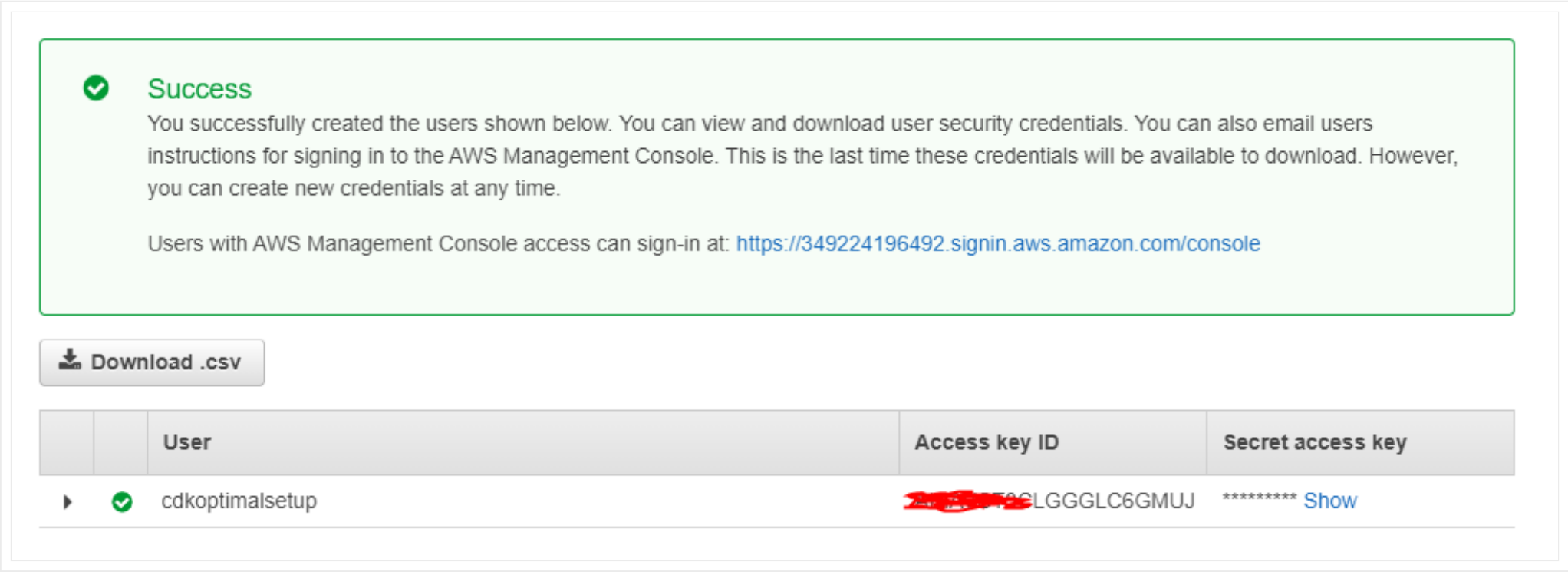
Tags

No tags were added.



Save the Access key ID as well as the Secret access key in a safe place.

You will need them to authenticate the account for Cloudockit.



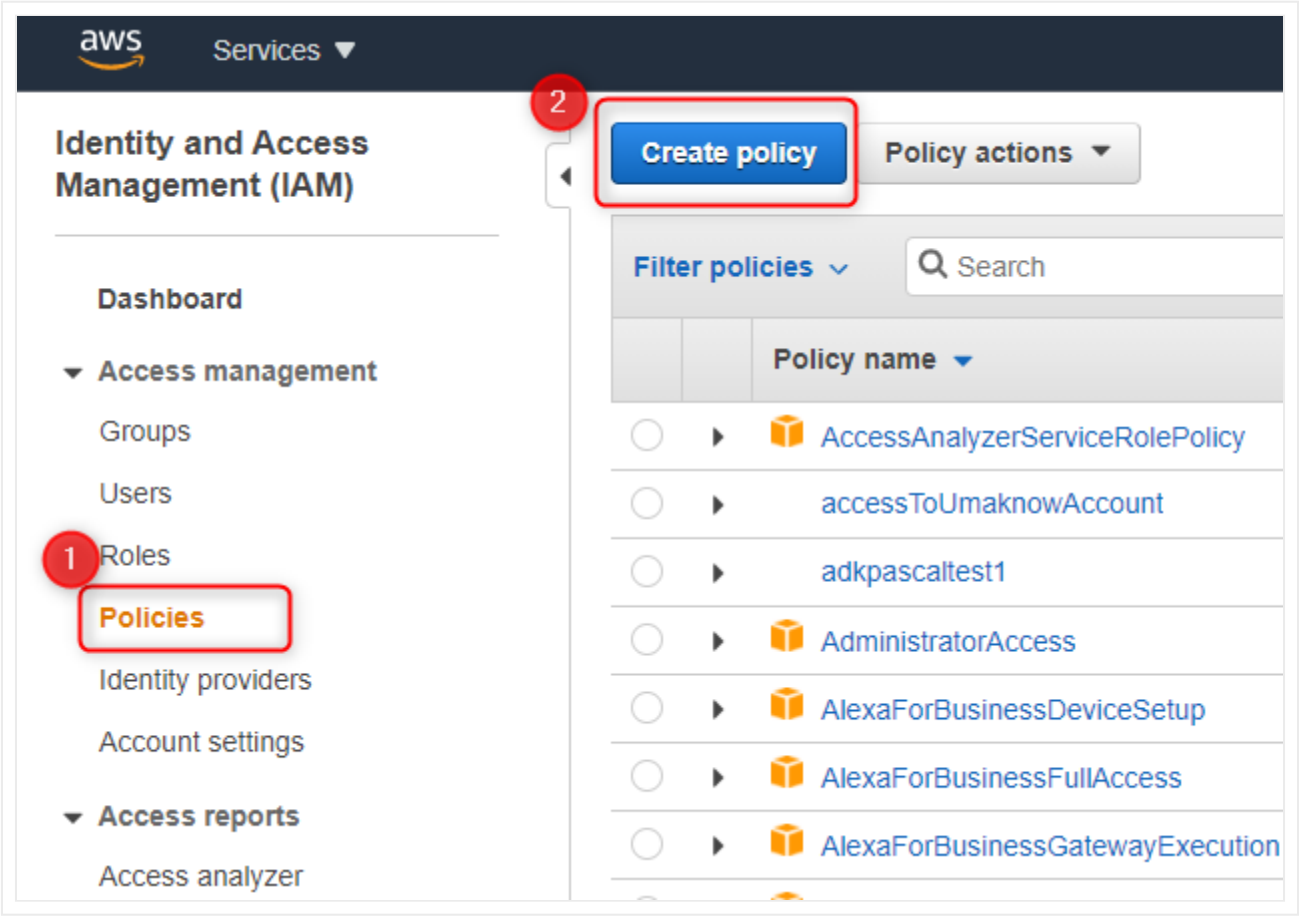
Step 2 – EC2 Instance Role and Policies

Let's create a policy and the role required for the EC2 instance to access the accounts and generate documentation.

Create a Policy

Connect to the AWS Console and select IAM.

Select **Policies** and press **Create Policy.**

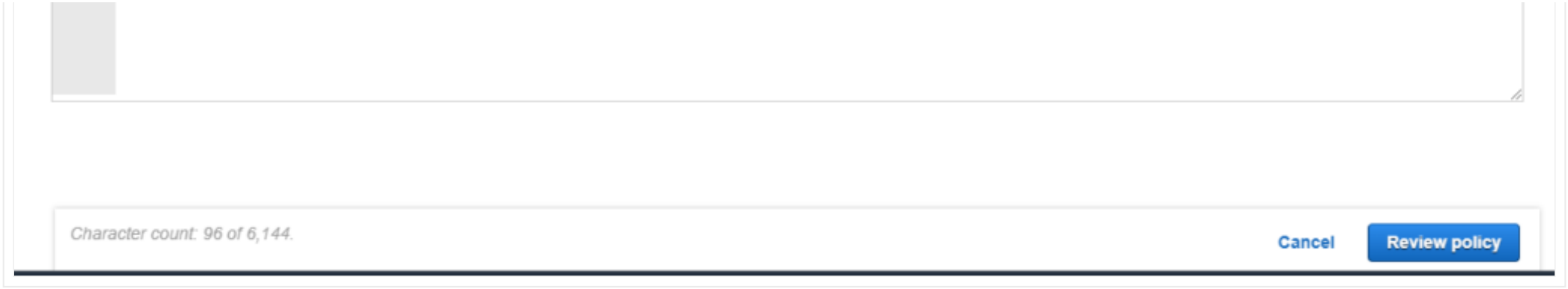


Select the JSON tab and paste this JSON into the window.

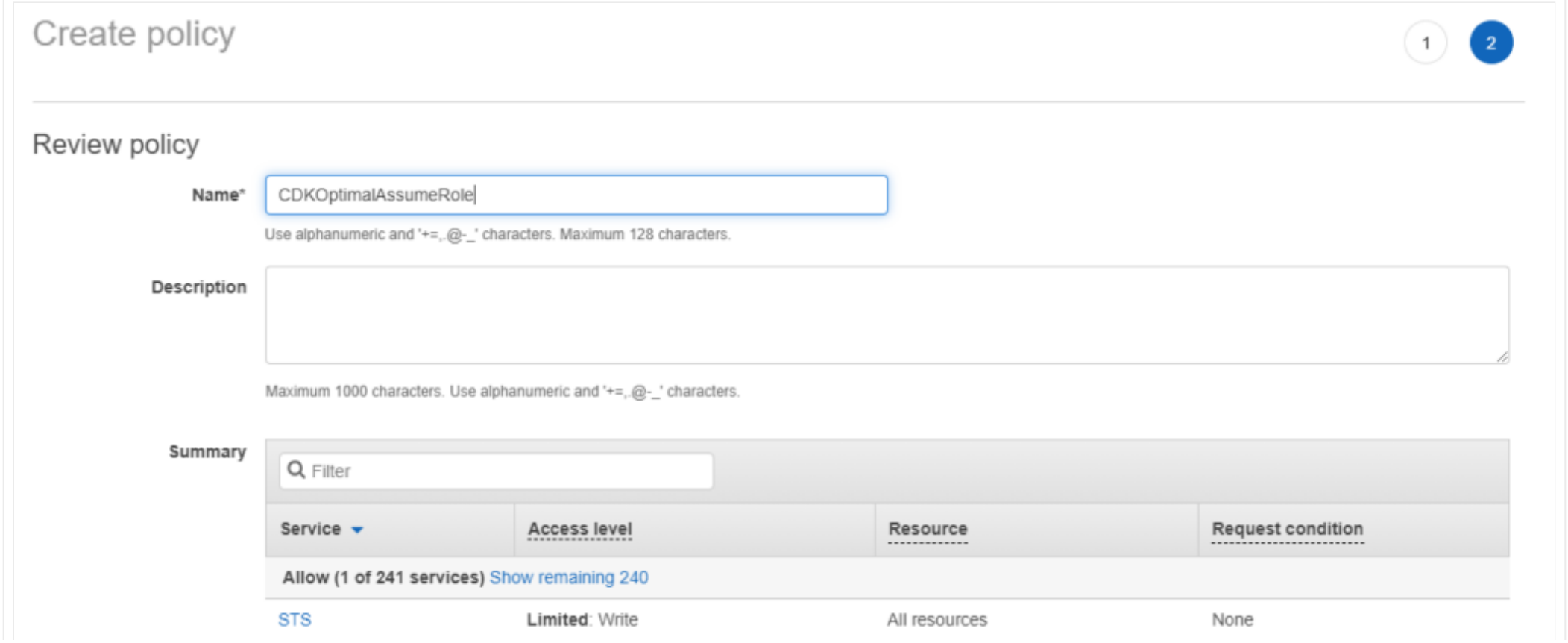
Click: Review Policy

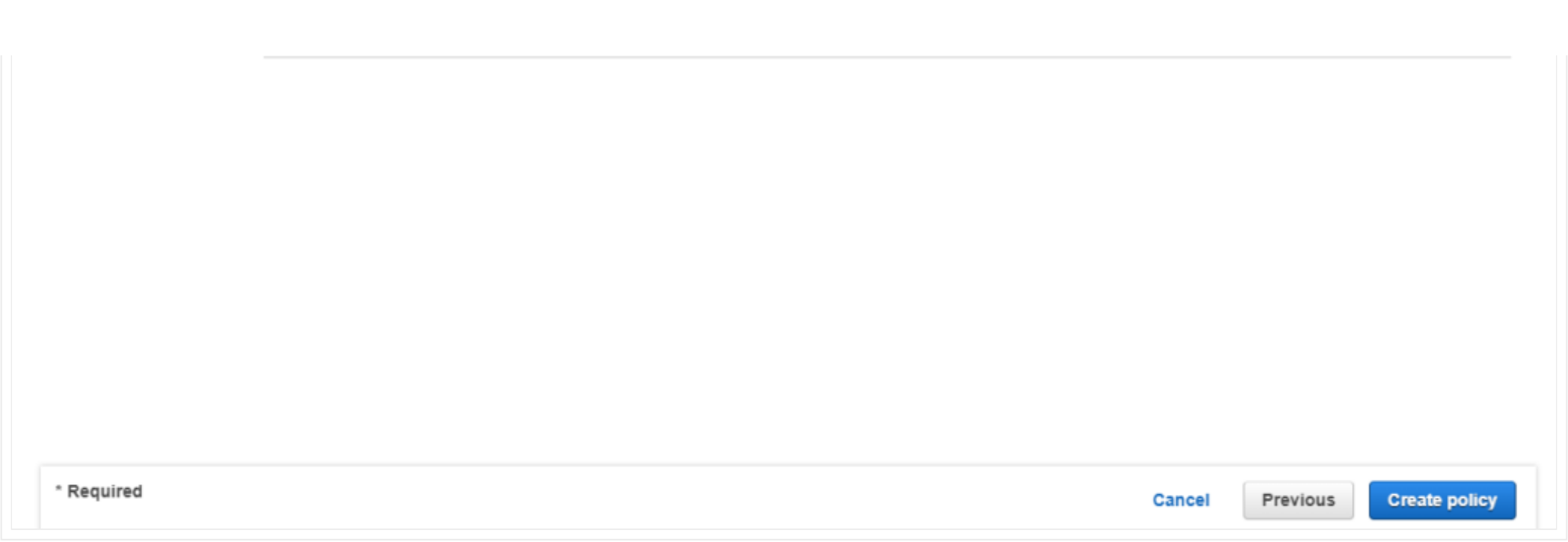
```
{
    "Version": "2008-10-17",
```

```
    "Statement": {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "*"
        }
}
```



Give the policy a unique name and press **Create policy.**





Add Tags

Add tags based on your organization's policies.

Click: Review

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. Learn more

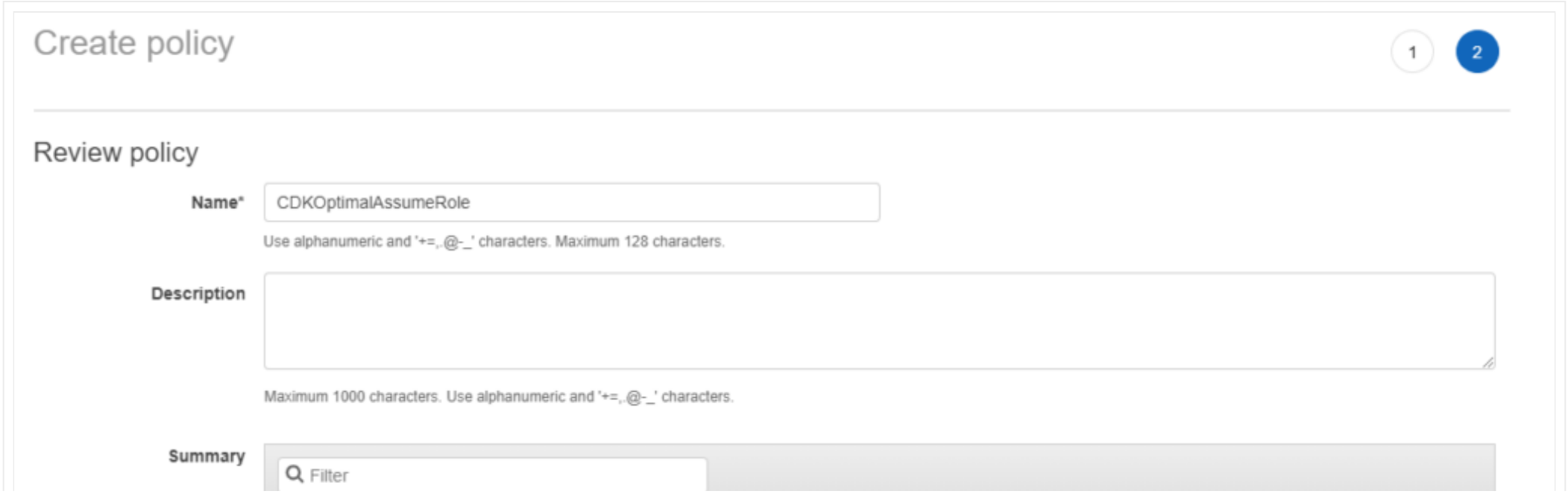
Key	Value (optional)	Remove
Add new key		

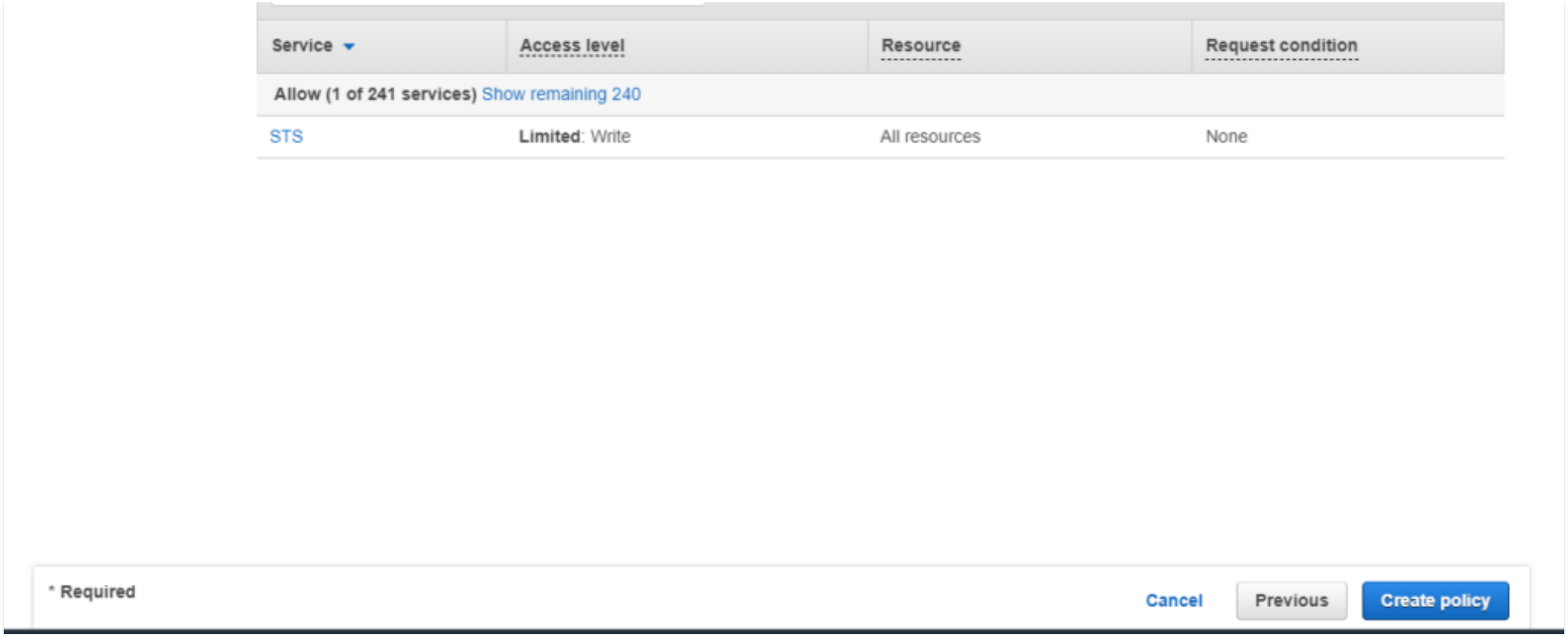
You can add 50 more tags.

Review

Give the policy a unique name and review to make sure everything is in order.

Click: Create Role

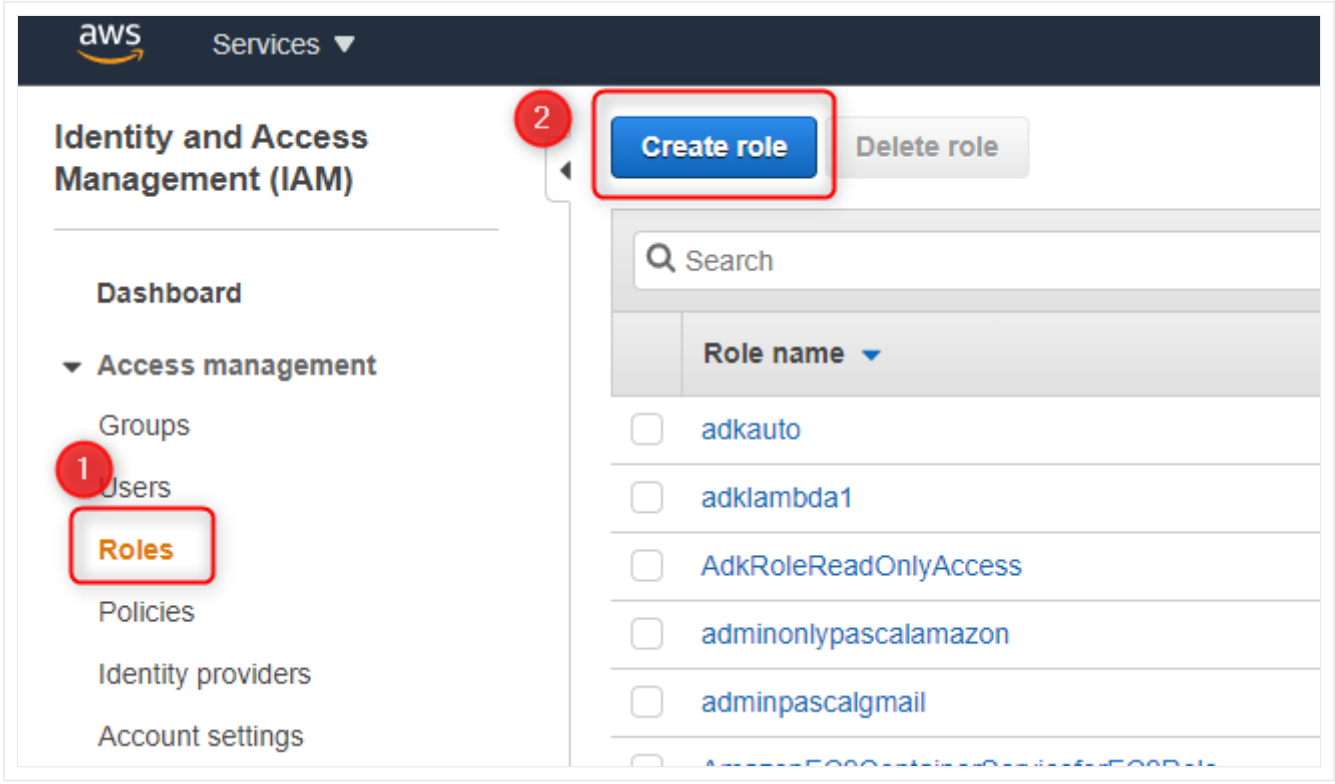




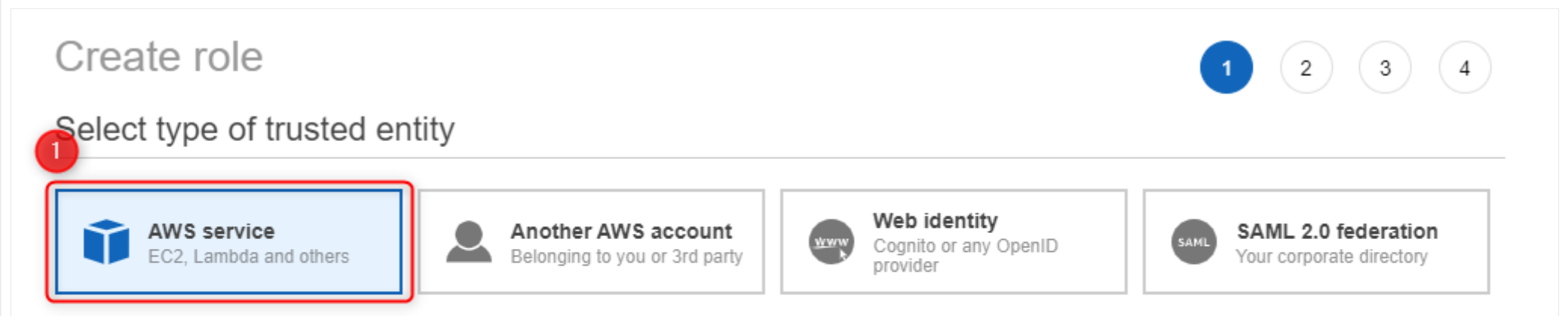
Create an EC2 Role for Cross-Account Documentation

Connect to the AWS Console and select IAM.

Select **Roles** and press **Create role.**



Under Select type of trusted entity, select AWS Service.



Allows AWS services to perform actions on your behalf. Learn more

Choose a use case

2

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CloudWatch Events	EKS	KMS	Rekognition
AWS Backup	CodeBuild	EMR	Kinesis	RoboMaker
AWS Chatbot	CodeDeploy	ElastiCache	Lake Formation	S3
AWS Marketplace	CodeGuru	Elastic Beanstalk	Lambda	SMS
AWS Support	CodeStar Notifications	Elastic Container Service	Lex	SNS
Amplify	Comprehend	Elastic Transcoder	License Manager	SWF
AppStream 2.0	Config	ElasticLoadBalancing	Machine Learning	SageMaker
AppSync	Connect	Forecast	Macie	Security Hub
Application Auto Scaling	DMS	GameLift	Managed Blockchain	Service Catalog
Application Discovery Service	Data Lifecycle Manager	Global Accelerator	MediaConvert	Step Functions
Batch	Data Pipeline	Glue	Migration Hub	Storage Gateway
	DataSync	Greengrass	OpsWorks	Systems Manager

3

* Required

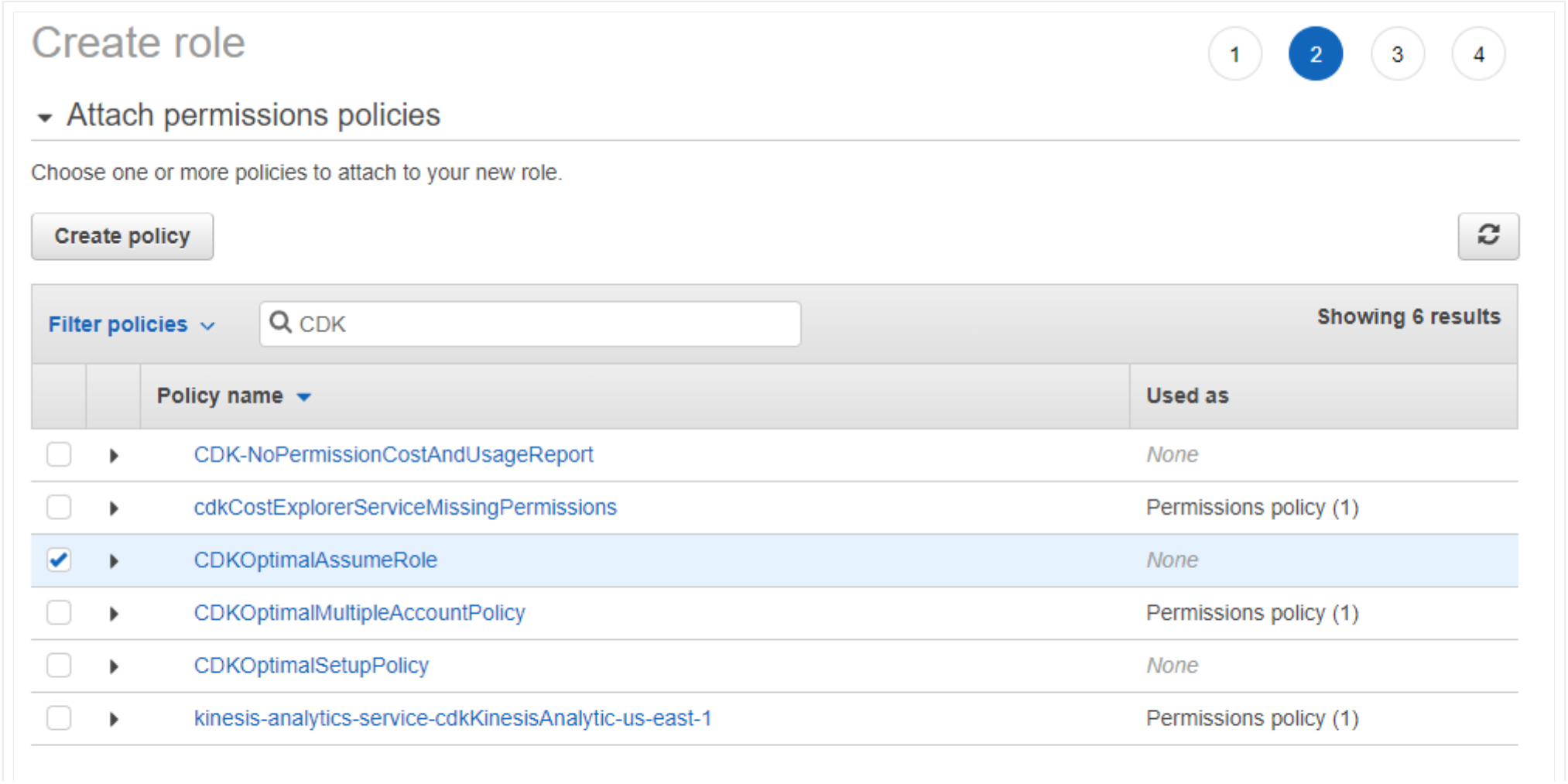
Cancel

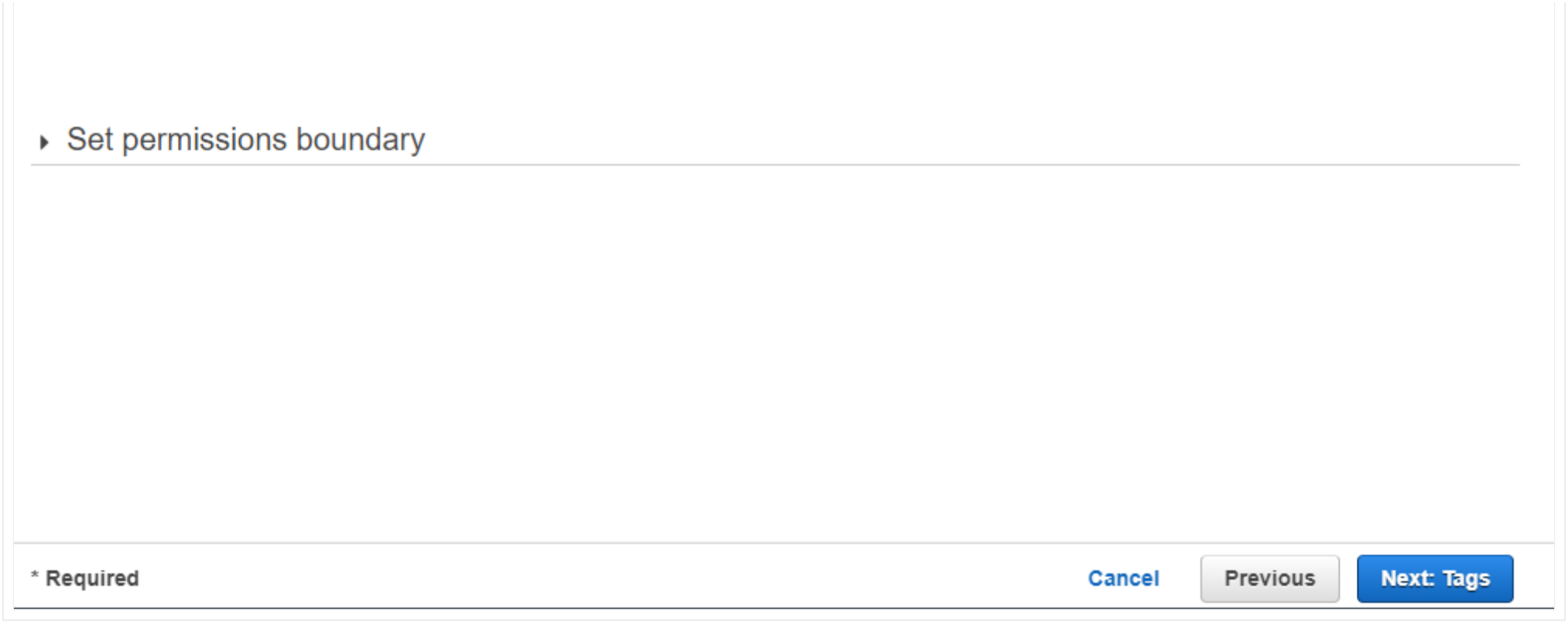
Next: Permissions

From the Attach permissions policies screen select the following policies:

- ReadOnlyAccess
- CDKOptimalAssumeRole (Name of the policy you have created in the previous setup)

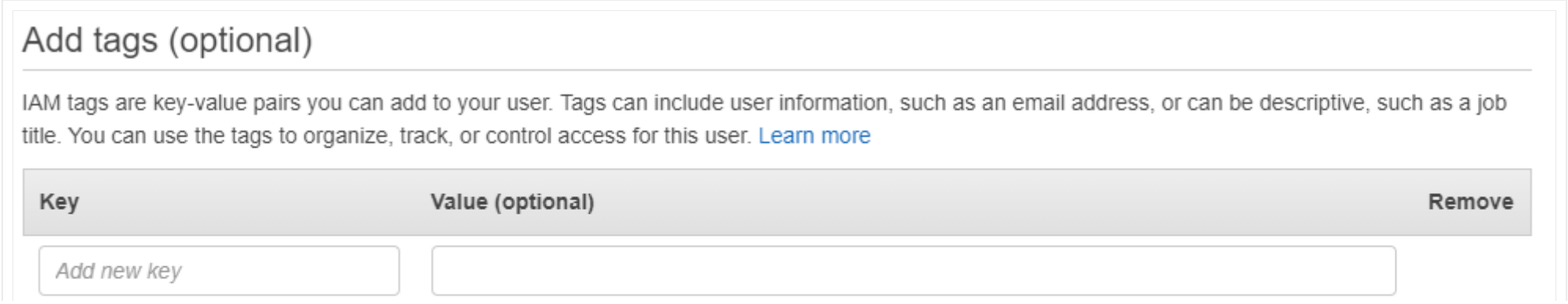
Click: Tags





Add tags based on your organization'ss policies.

Click: Review

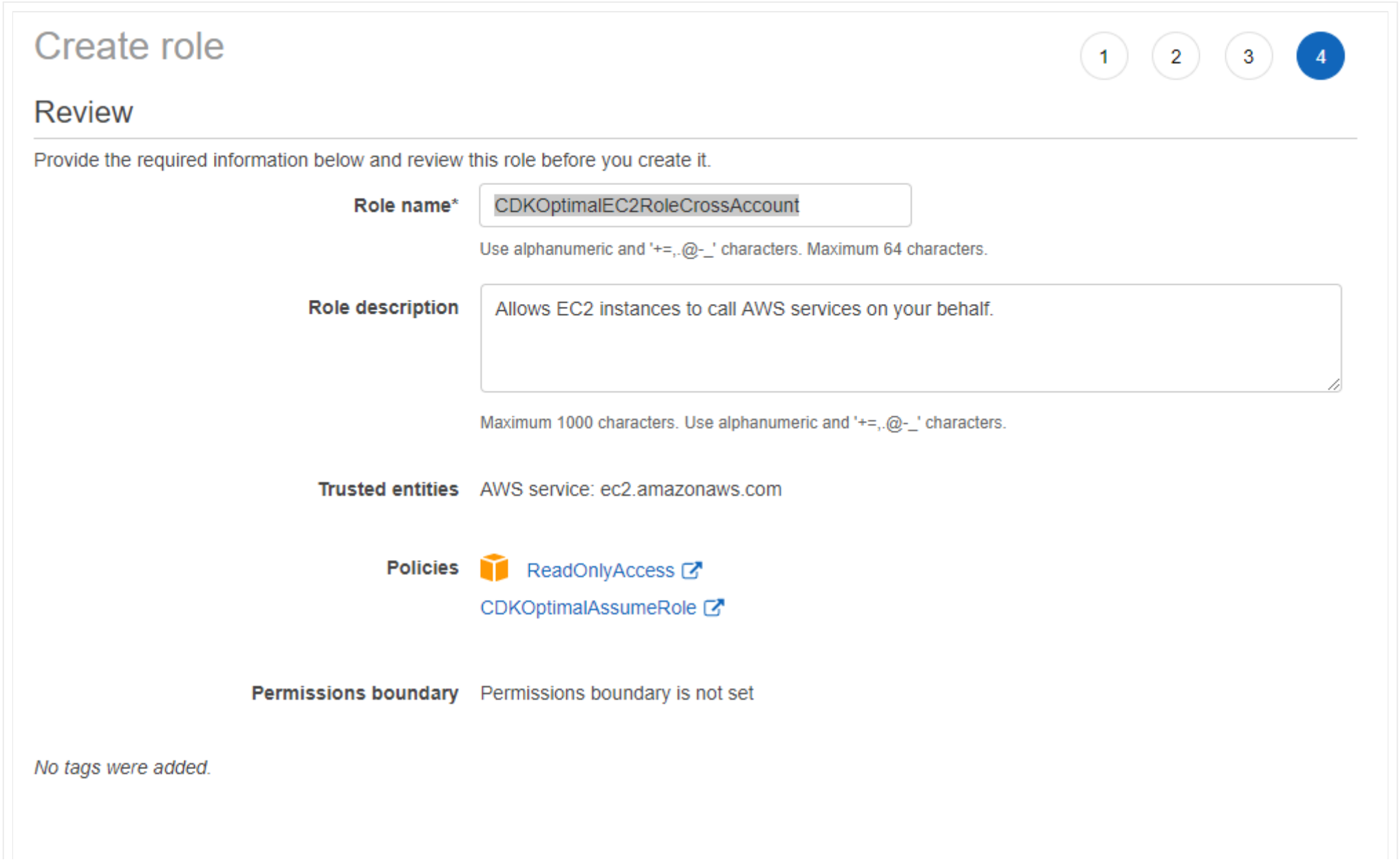


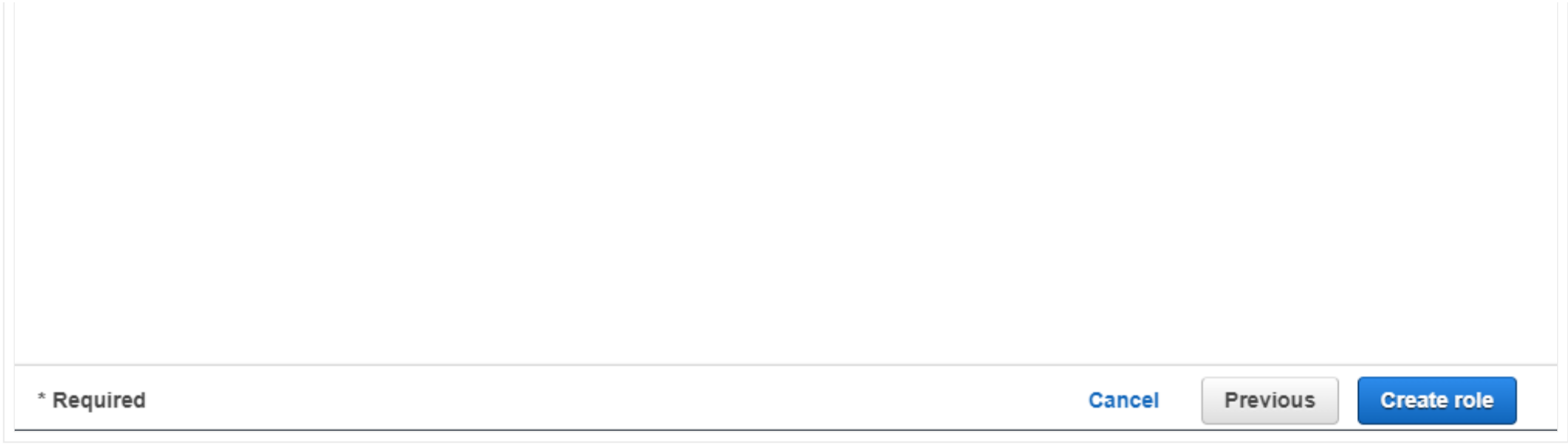
You can add 50 more tags.

Cancel | Previous | Next: Review

Enter a unique name for your role.

Review the parameters and press **Create Role**.

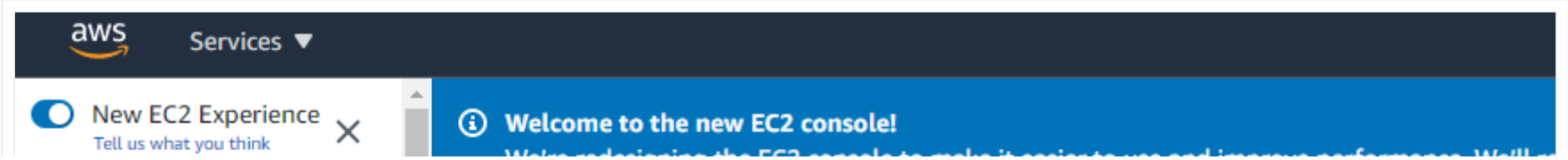


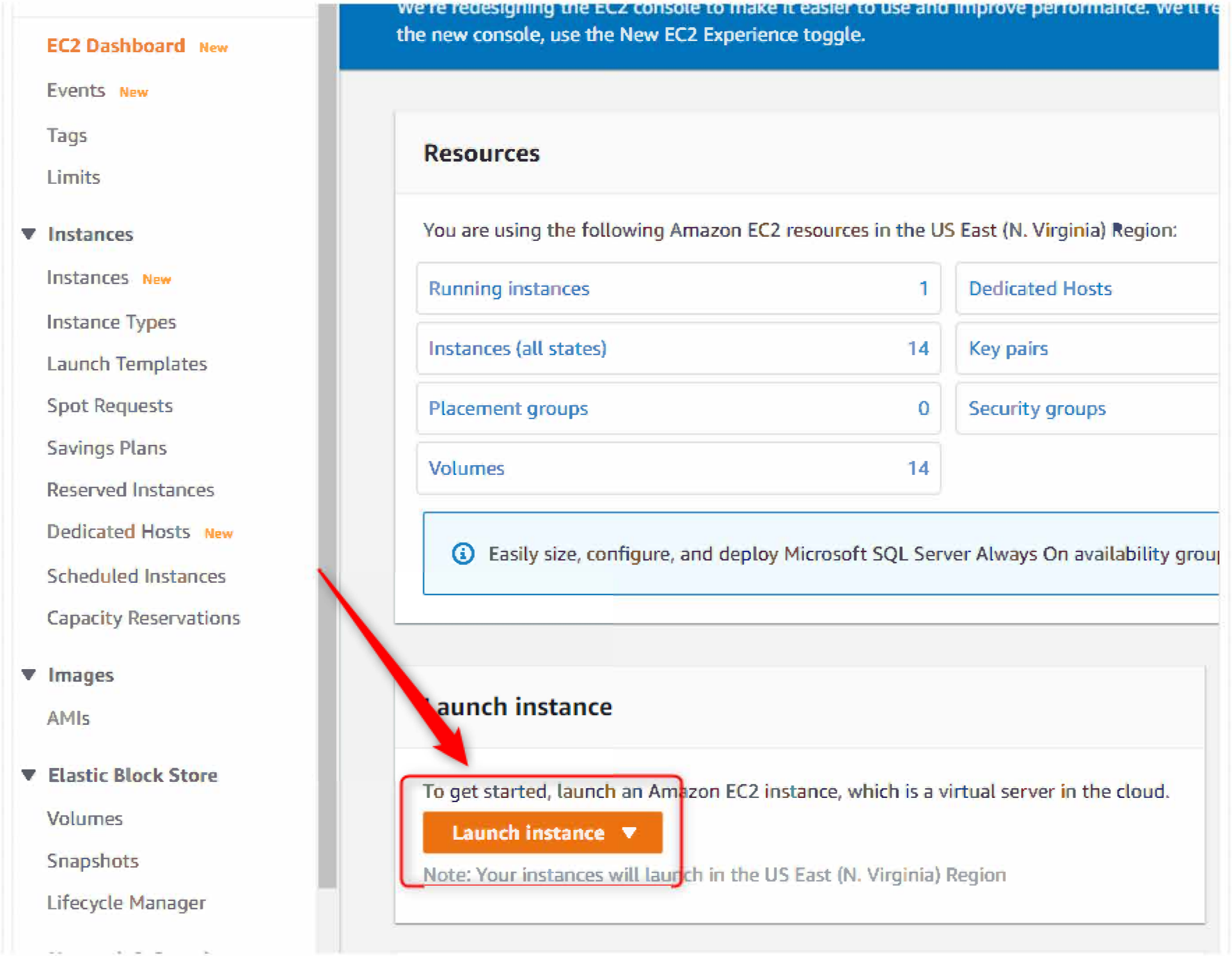


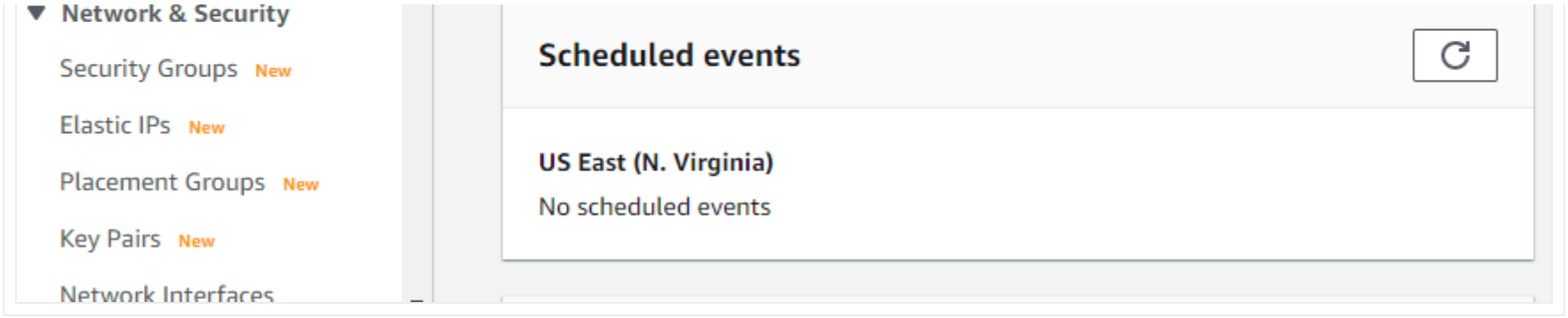
Step 3 – Creation of the EC2 Instance

Connect to the AWS Console and go to the EC2.

From the EC2 Dashboard page Press the **Launch instance** button.

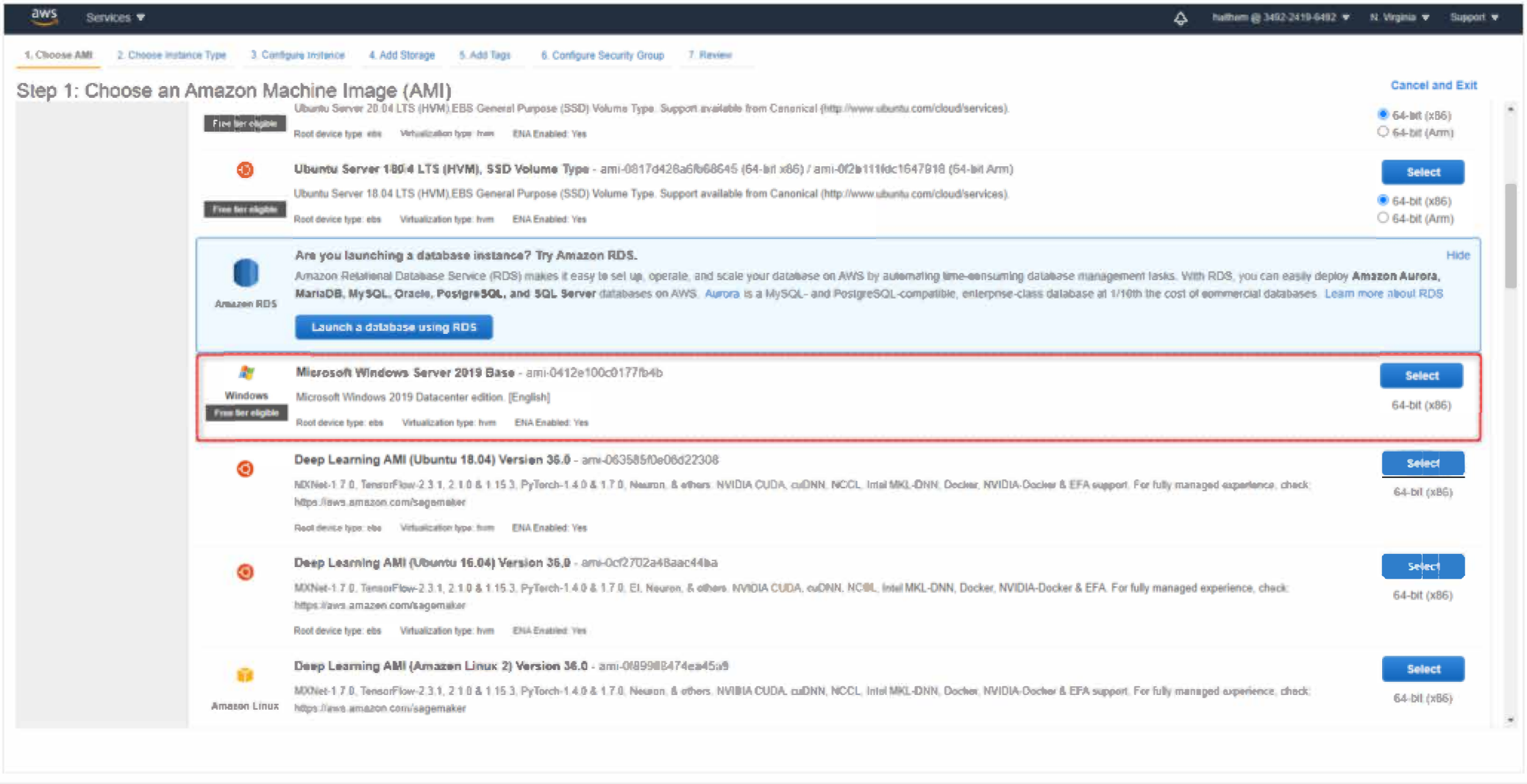



EC2 Dashboard New
Events New
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Scheduled Instances
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
the new console, use the New EC2 Experience toggle.
Resources
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:
Running instances 1
Dedicated Hosts
Instances (all states) 14
Key pairs
Placement groups 0
Security groups
Volumes 14
Easily size, configure, and deploy Microsoft SQL Server Always On availability grou
aunch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Note: Your instances will launch in the US East (N. Virginia) Region



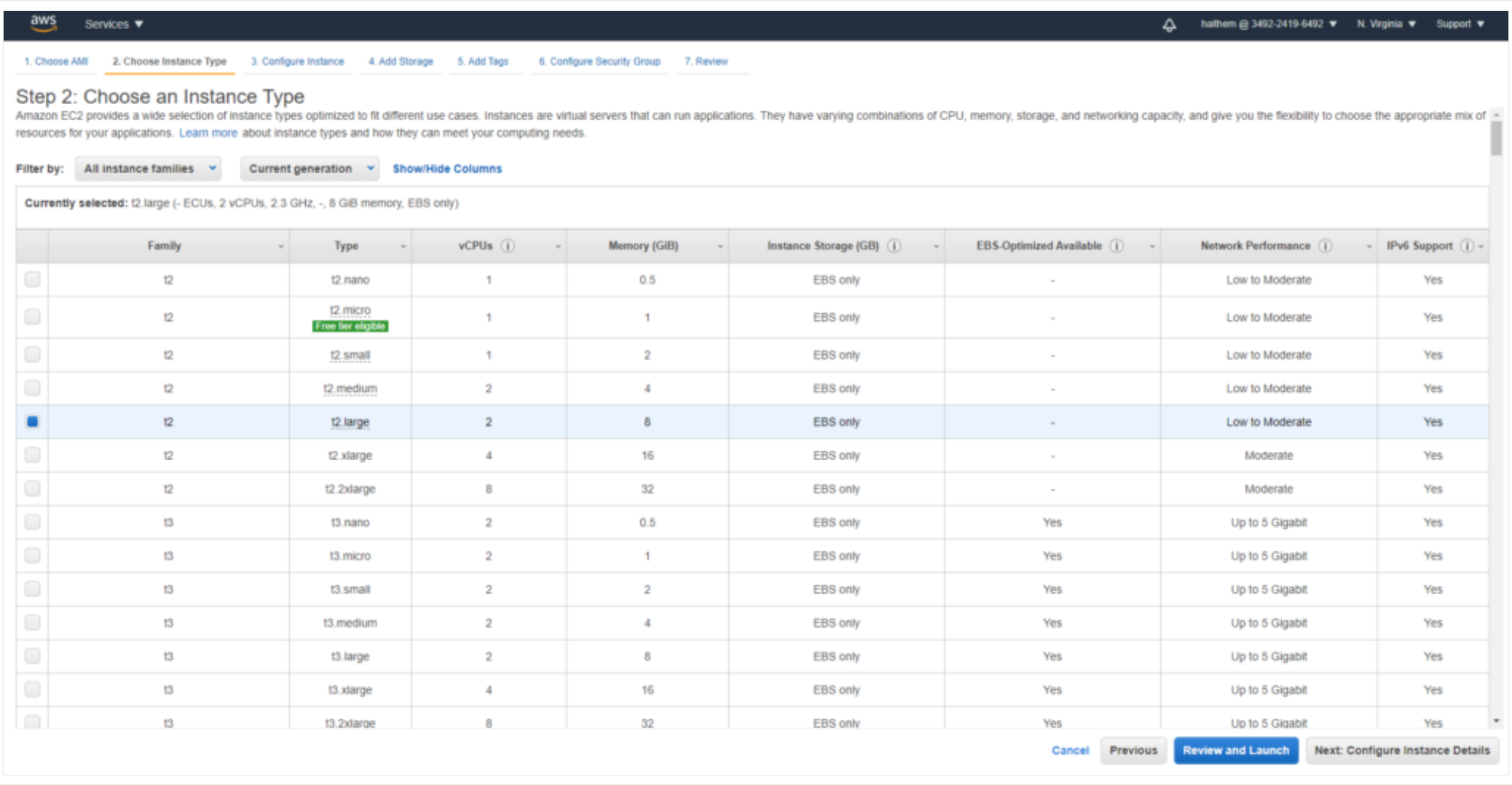
Choose an Amazon Machine Image (AMI)

Select Microsoft Windows Server 2019 Base. (Linux OS is not supported).


aws
Services
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
Free tier eligible
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)
64-bit (Arm)
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0817d428a6fb68645 (64-bit x86) / ami-0f2b111fdc1647918 (64-bit Arm)
Select
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Free tier eligible
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)
64-bit (Arm)
Amazon RDS
Are you launching a database instance? Try Amazon RDS.
Hide
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy Amazon Aurora, MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server databases on AWS. Aurora is a MySQL- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. Learn more about RDS
Launch a database using RDS
Windows
Microsoft Windows Server 2019 Base - ami-0412e100c0177fb4b
Select
Free tier eligible
Microsoft Windows 2019 Datacenter edition. [English]
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)
Deep Learning AMI (Ubuntu 18.04) Version 36.0 - ami-063585f0e06d22308
Select
MXNet-1.7.0, TensorFlow-2.3.1, 2.1.0 & 1.15.3, PyTorch-1.4.0 & 1.7.0, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: https://aws.amazon.com/sagemaker
64-bit (x86)
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Deep Learning AMI (Ubuntu 16.04) Version 36.0 - ami-0cf2702a4Baac44ba
Select
MXNet-1.7.0, TensorFlow-2.3.1, 2.1.0 & 1.15.3, PyTorch-1.4.0 & 1.7.0, EI, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA. For fully managed experience, check: https://aws.amazon.com/sagemaker
64-bit (x86)
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Deep Learning AMI (Amazon Linux 2) Version 36.0 - ami-0f899f8474ea45a9
Select
Amazon Linux
MXNet-1.7.0, TensorFlow-2.3.1, 2.1.0 & 1.15.3, PyTorch-1.4.0 & 1.7.0, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: https://aws.amazon.com/sagemaker
64-bit (x86)

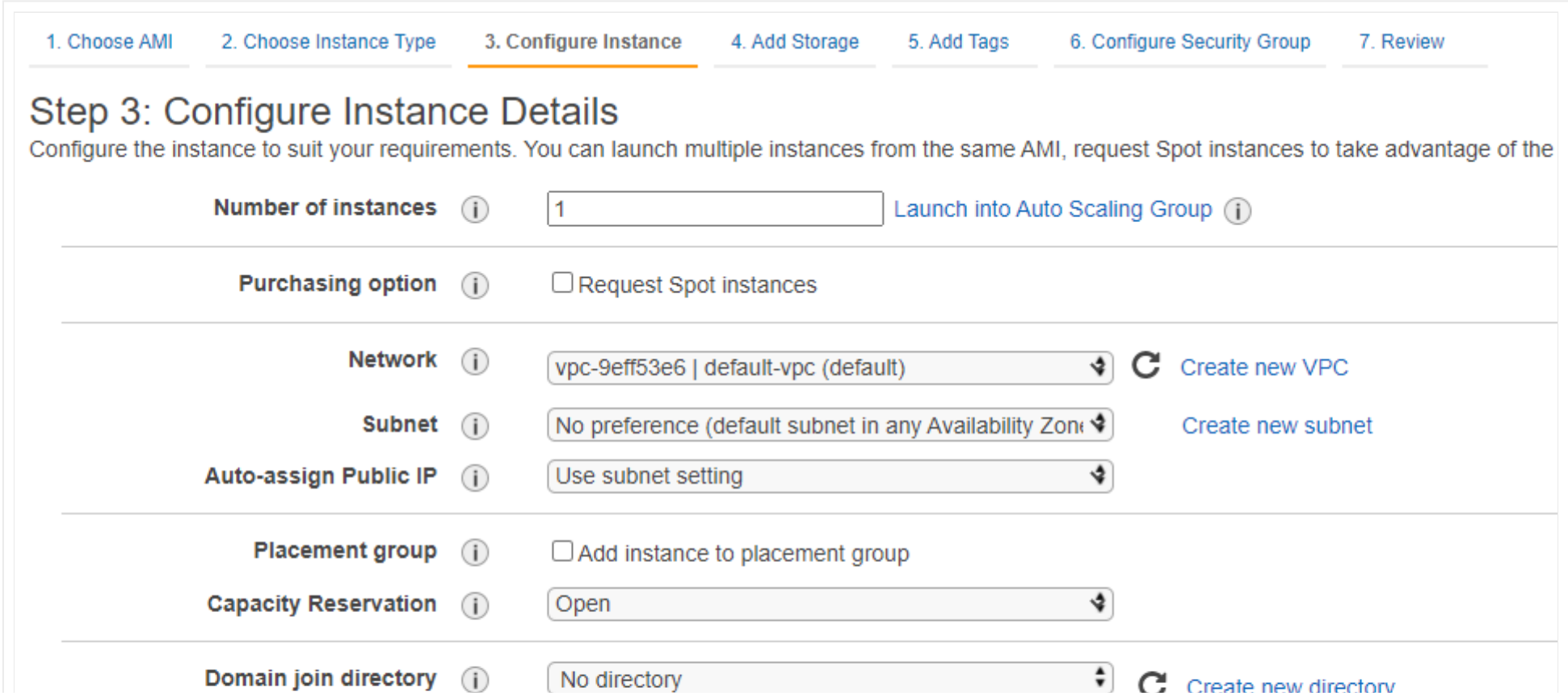
Choose an Instance Type

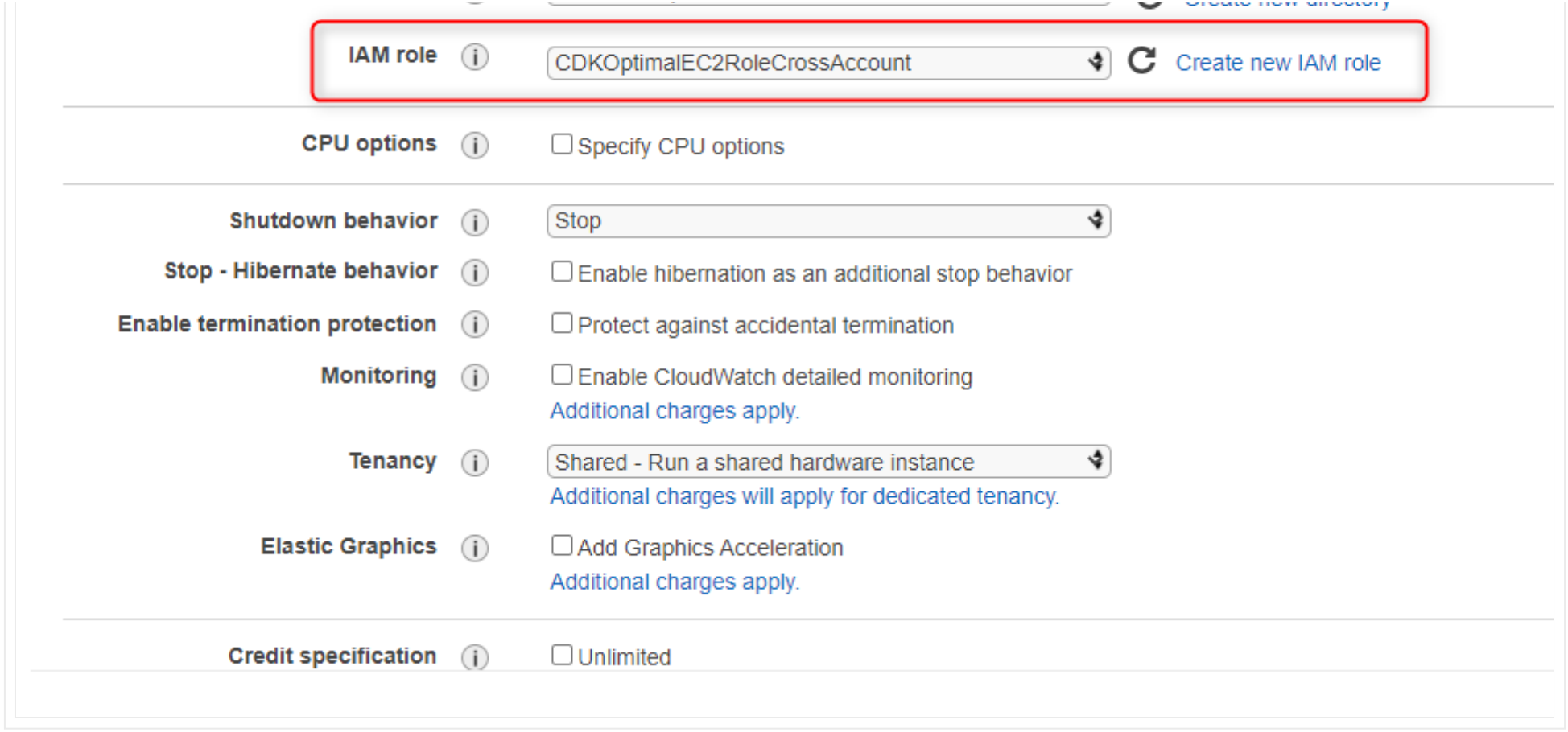
We have identified that Cloudockit Desktop performs at its best with 4 CPUs and 16 GiB of memory. You can however choose the type that you prefer.



Configure Instance Details

Configure the instance based on your organization's best practices and make sure to select the CrossAccount IAM role created in the previous step.





Add Storage

You can leave the default parameters.

Add Tags

Add the tags based on your organization’s tagging policy.

Configure Security Group

Create or assign a security group based on your organization’s security policies.

Review Instance Launch

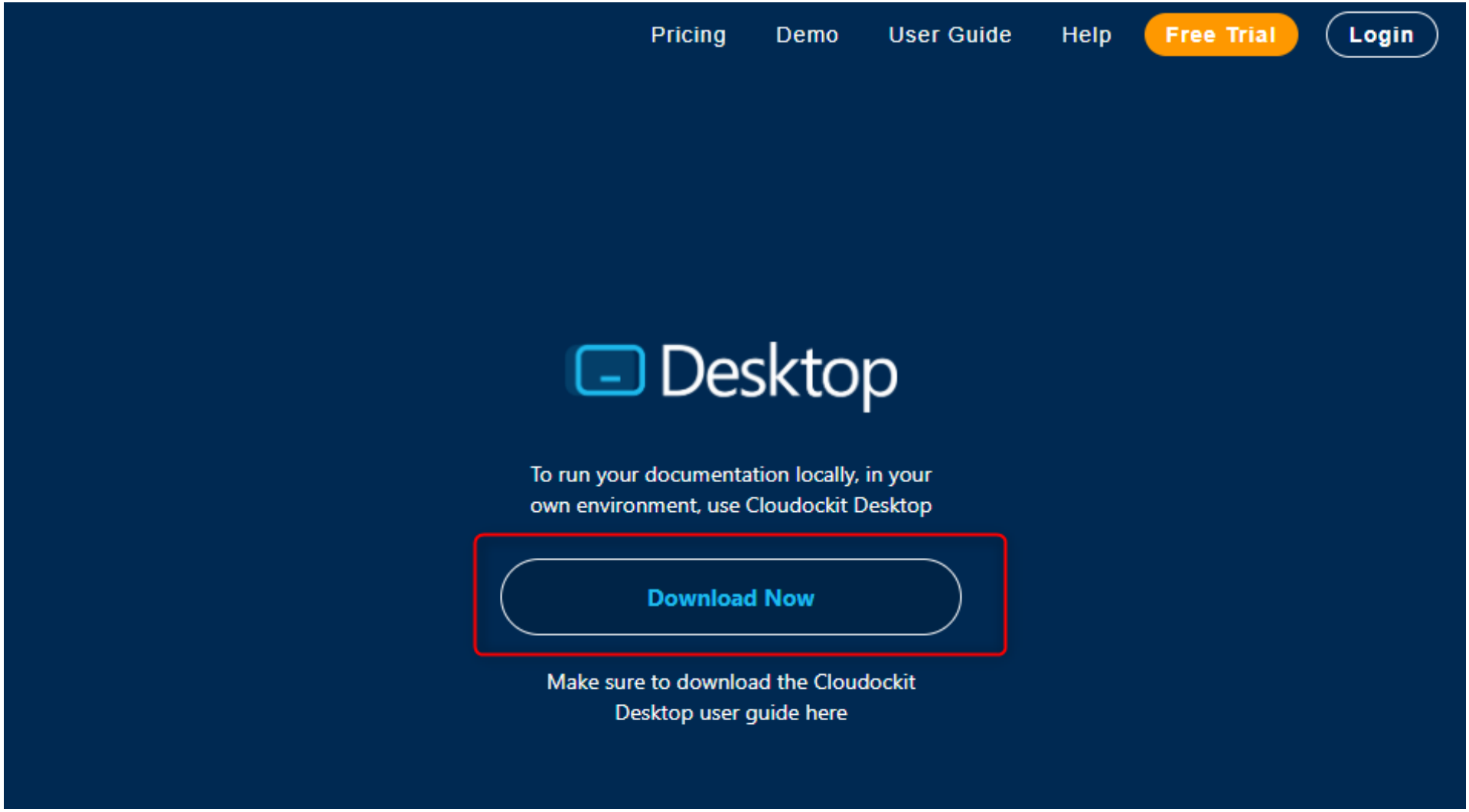
Review the parameters that have been set and press launch to create the instance.

Step 4 – Installing Cloudockit Desktop

Downloading the Document

You can get the Cloudockit installation file from our **website**.

Press the Download Now button to get the MSI.



Launch your instance and copy the MSI file on the instance.

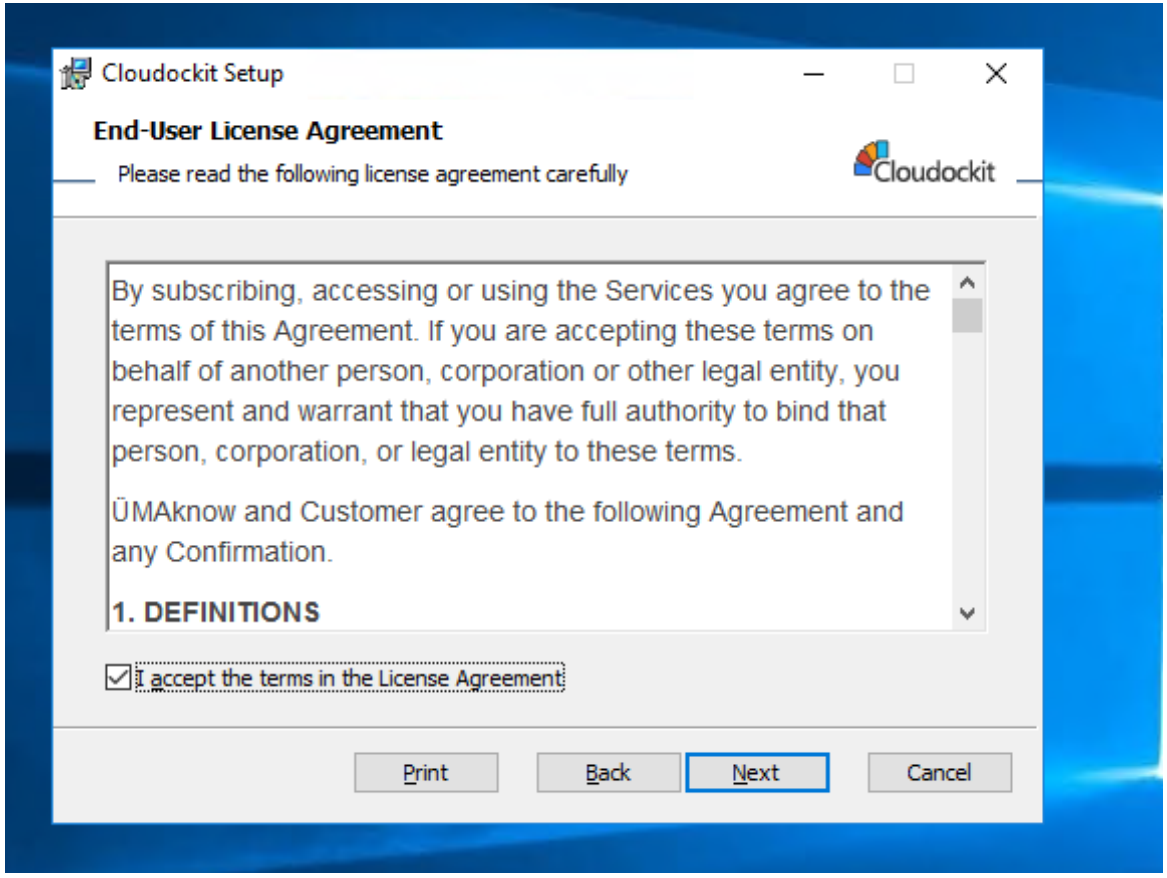
Double click the Cloudockit icon to start the installation.

Click: Next

Carefully read the terms in the license agreement.

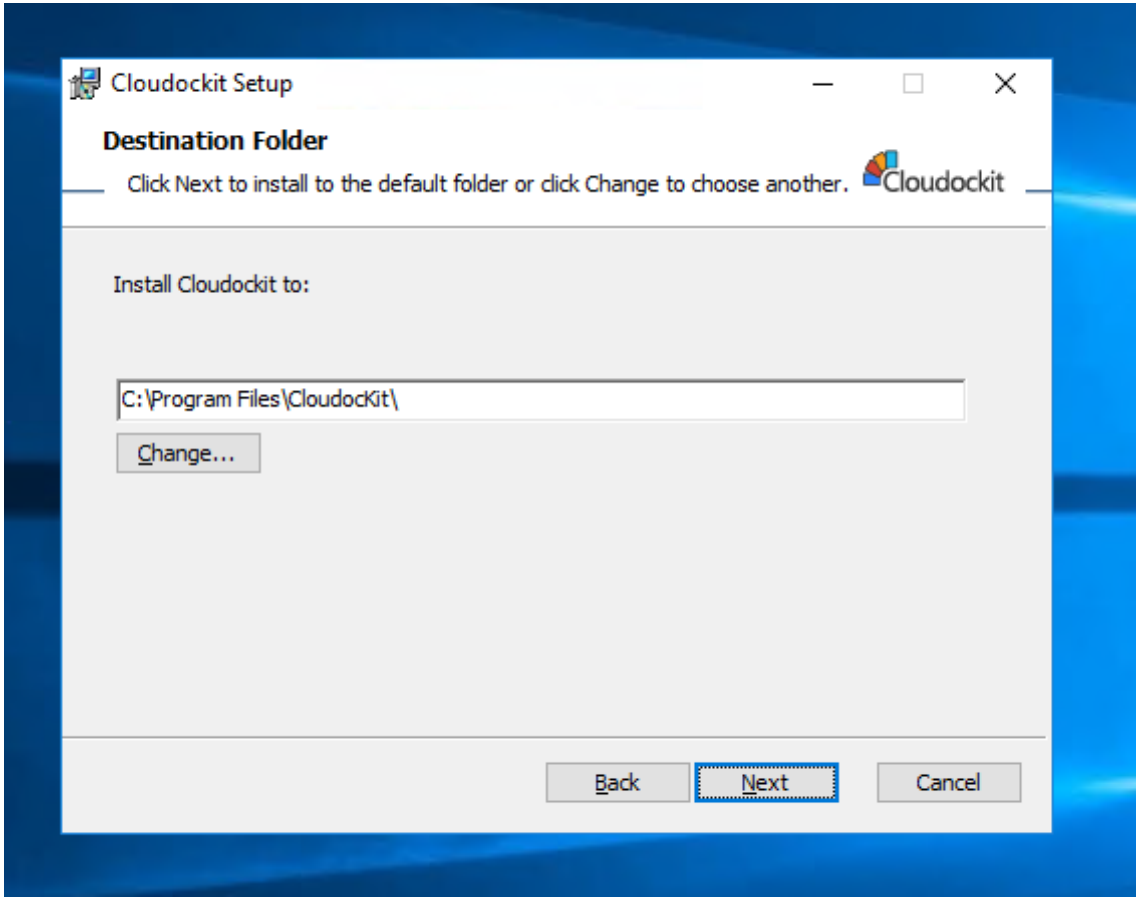
Check the box “I accept the terms in the License Agreement”.

Click: Next

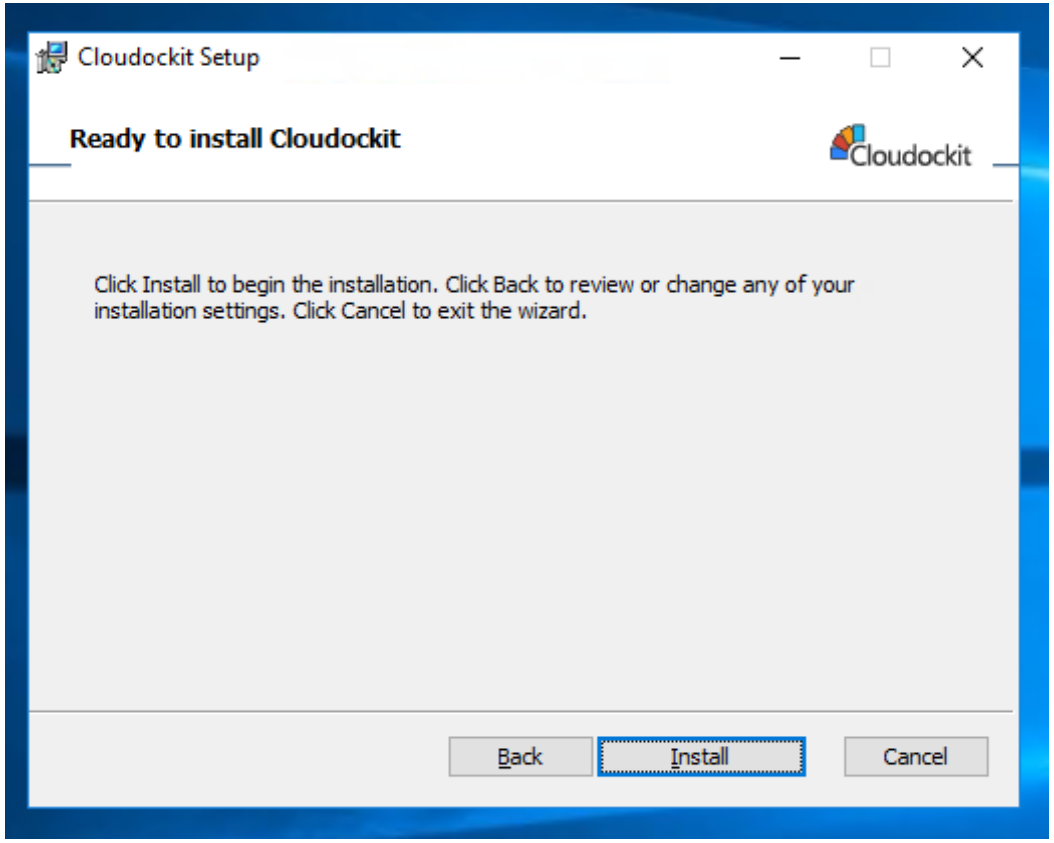


Select the path where you want to install Cloudockit Desktop.

Click: Next

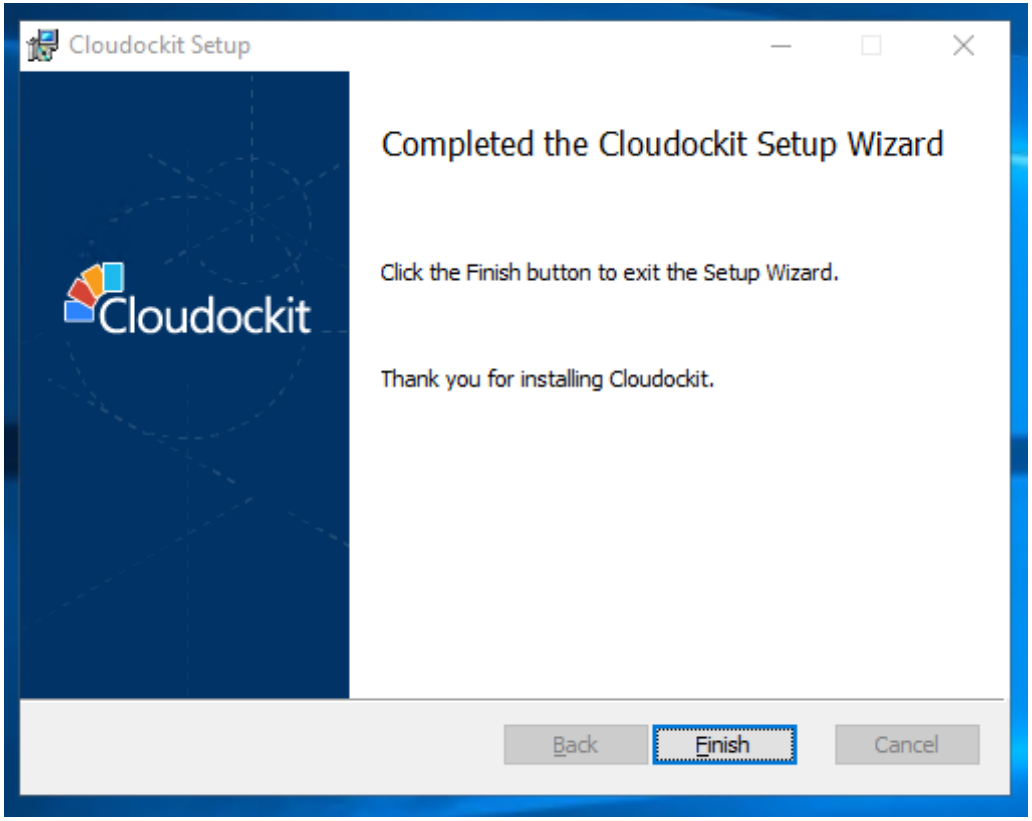


Click: Instal



Once the installation is complete.

Click: Finish

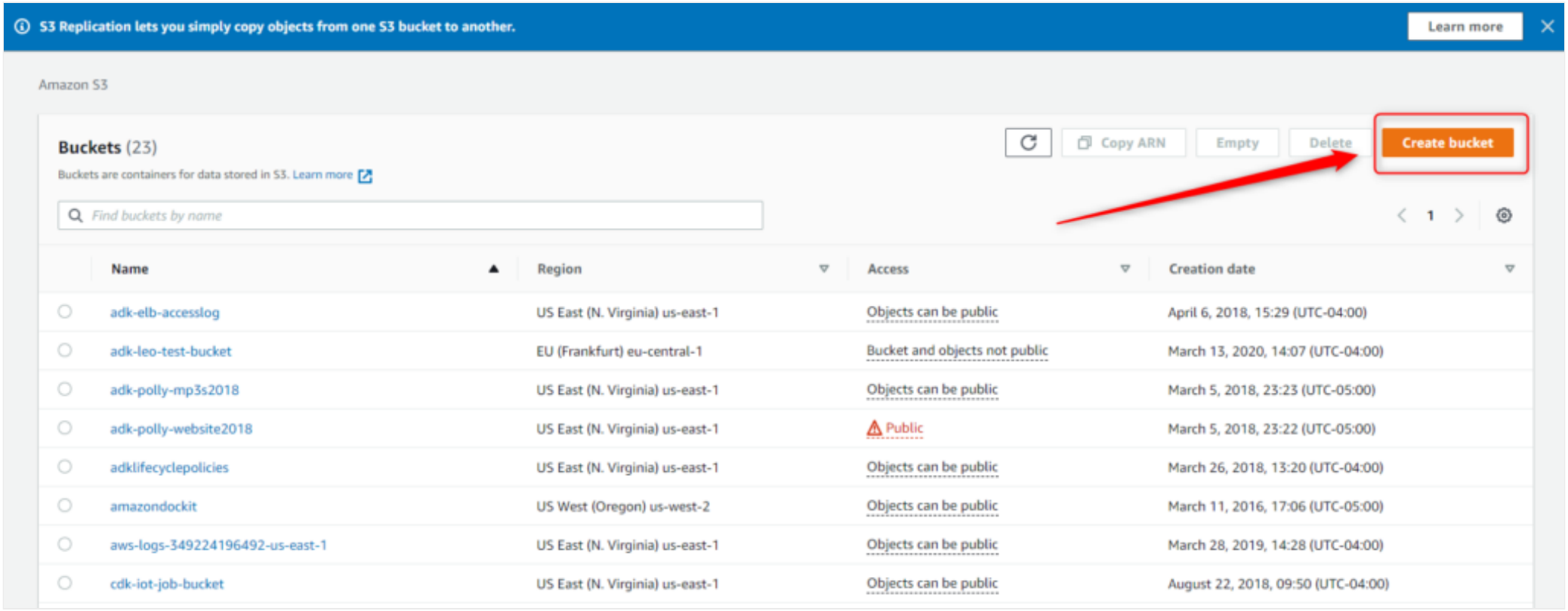


Step 5 – Creation of the Bucket

The bucket will allow you to save the documentation and be available to employees in your organization.

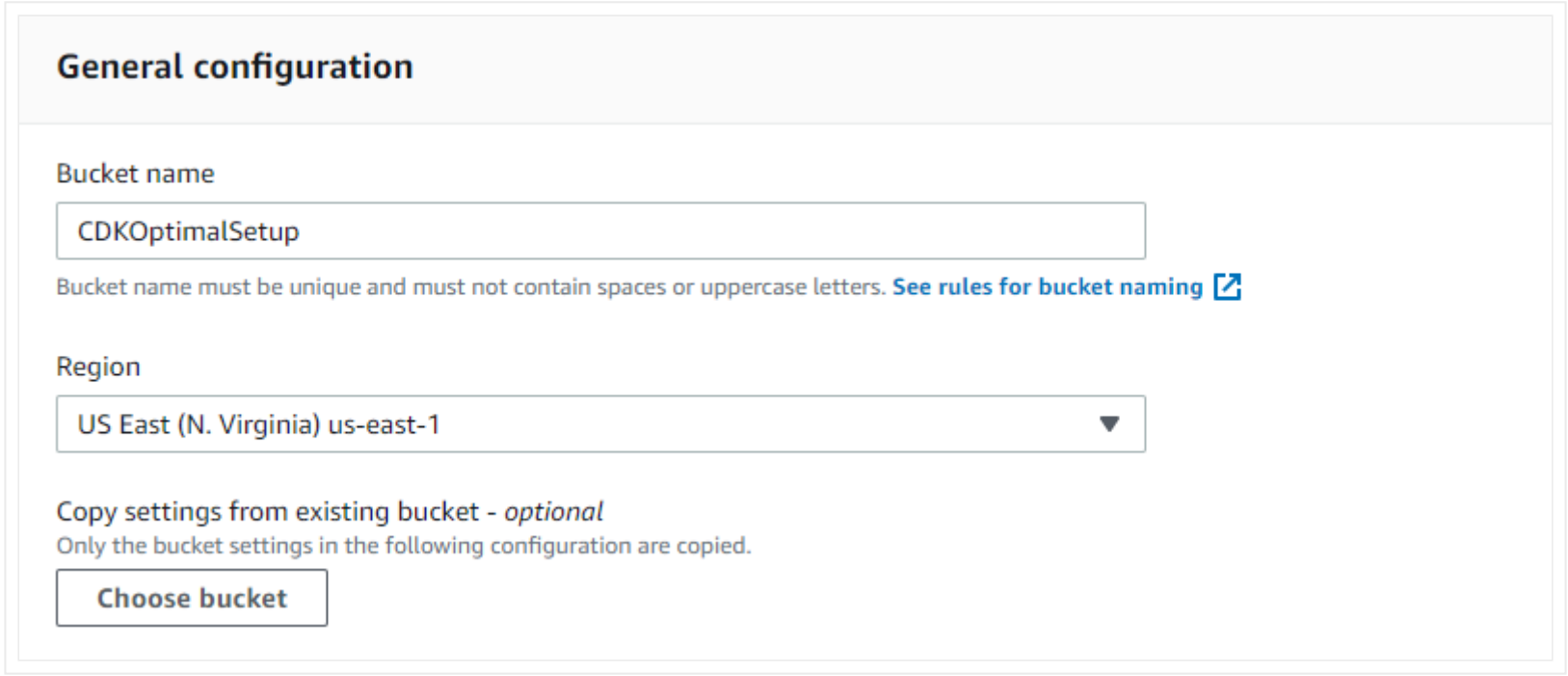
From the AWS Console, select **S3**

Press **Create bucket** in the upper right corner.



General Configuration

Name your bucket and select the Region of your choice.



Bucket Settings for Block Public Access

Define the public access based on your organization's best practices.

Bucket settings for Block Public Access

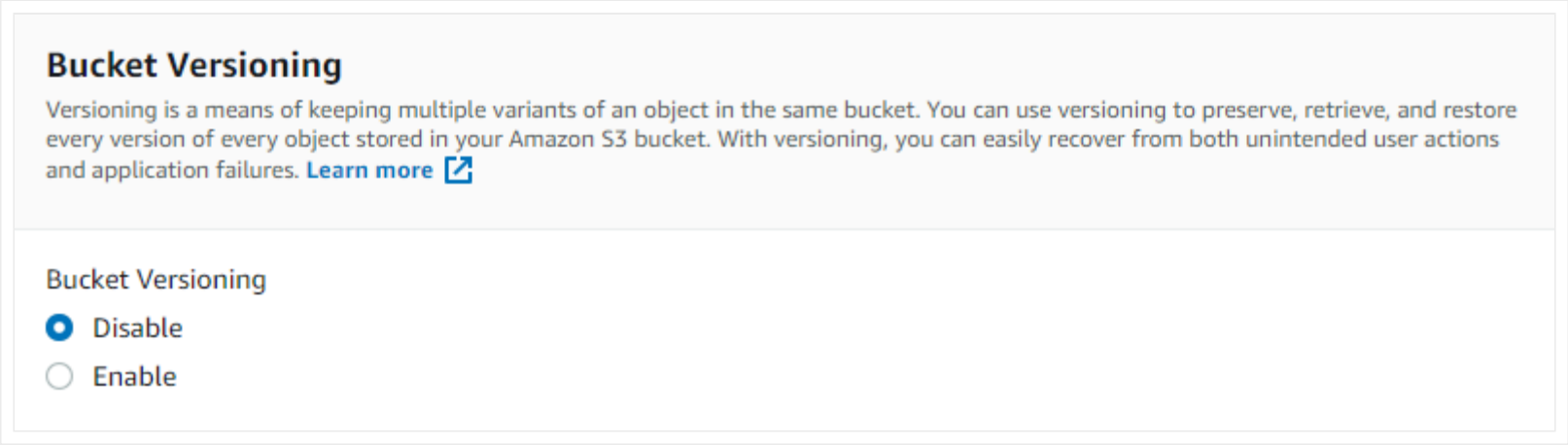
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

- [ ] **Block *all* public access**
  Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
  - [ ] **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
    S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - [ ] **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
    S3 will ignore all ACLs that grant public access to buckets and objects.
  - [ ] **Block public access to buckets and objects granted through *new* public bucket or access point policies**
    S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - [ ] **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
    S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

- [ ] I acknowledge that the current settings might result in this bucket and the objects within becoming public.

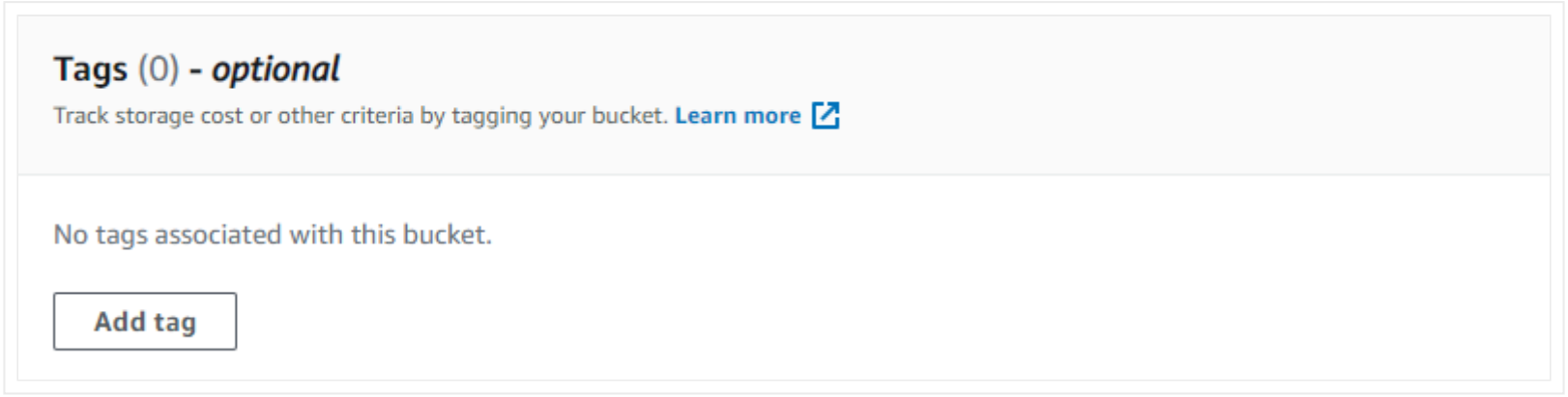
Bucket Versioning



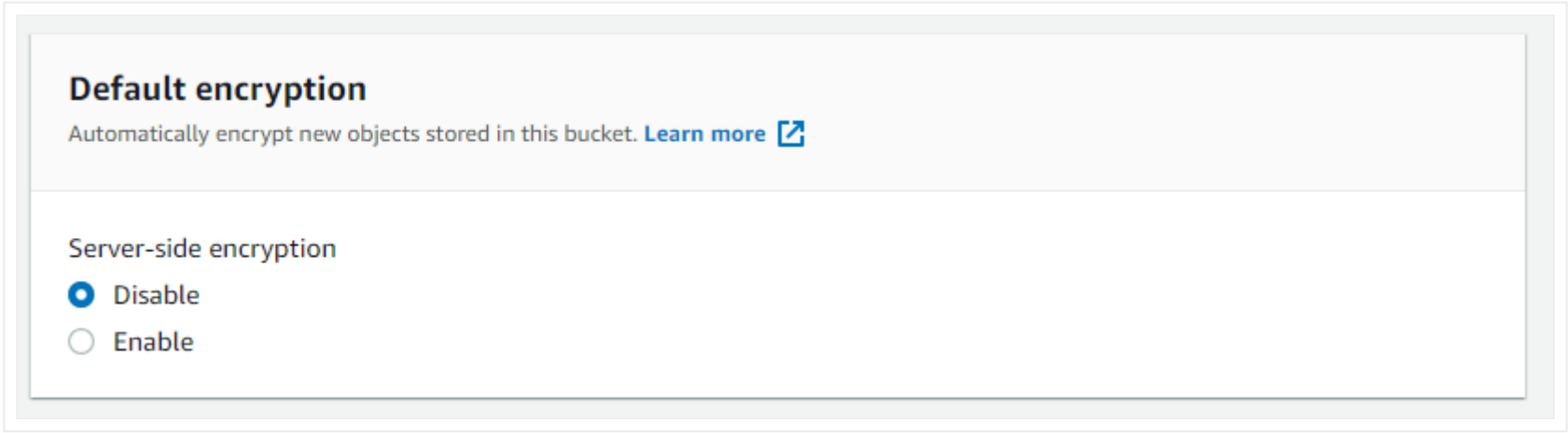
You can enable or disable bucket versioning based on your preferences.

Tags

Add tags based on your organization practices.



Default Encryption



Ensure to copy the Bucket ARN in a secure place. You will need it later.

Click: Finish

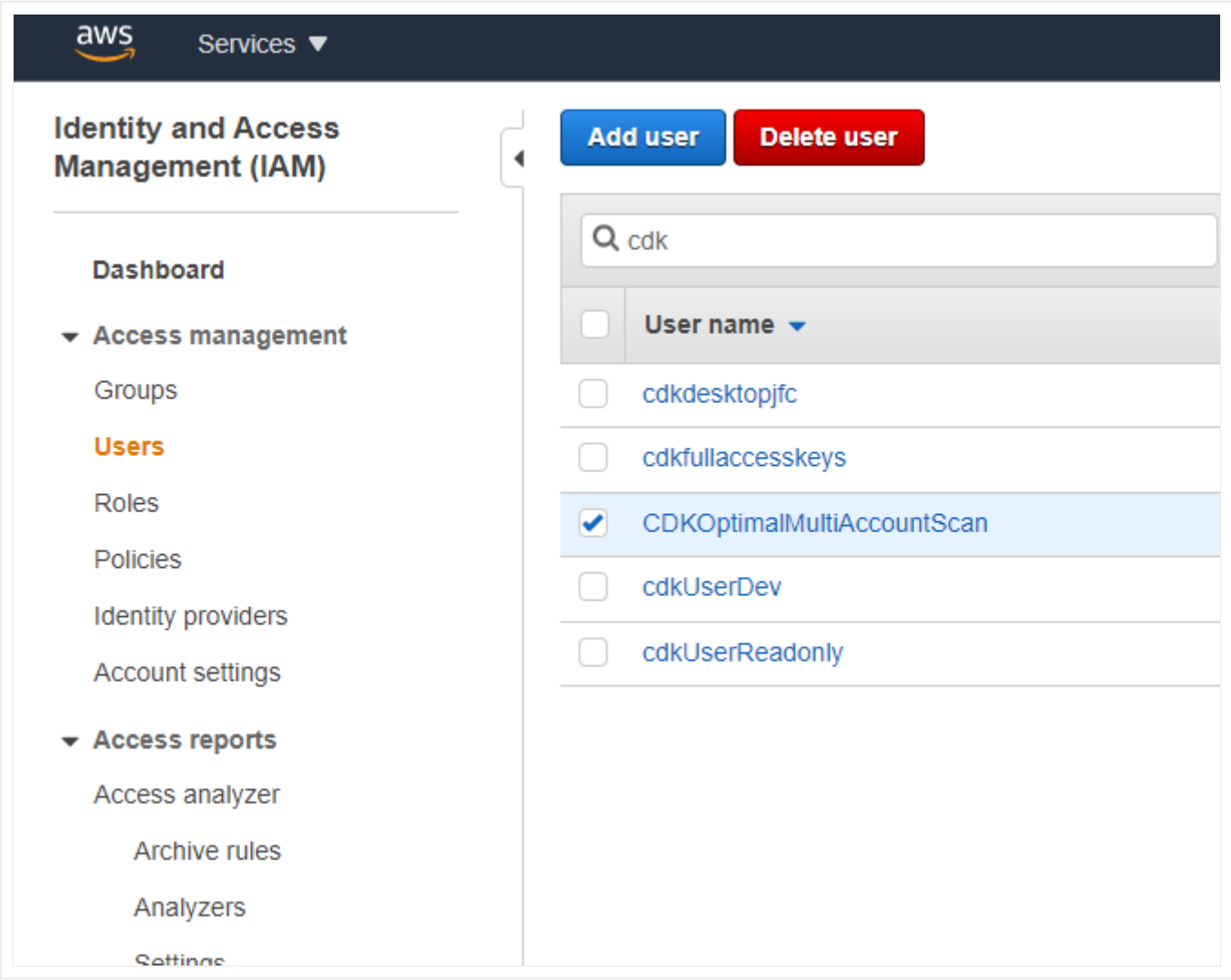
S3 Bucket Policy

You must now give your IAM user the policy to allow the S3 bucket to drop the files that Cloudockit will create.

Sign in to the AWS Console and open the IAM console at https://console.aws.amazon.com/iam/.

In the navigation panel, choose **Users** and search for the user you created.

Select the username

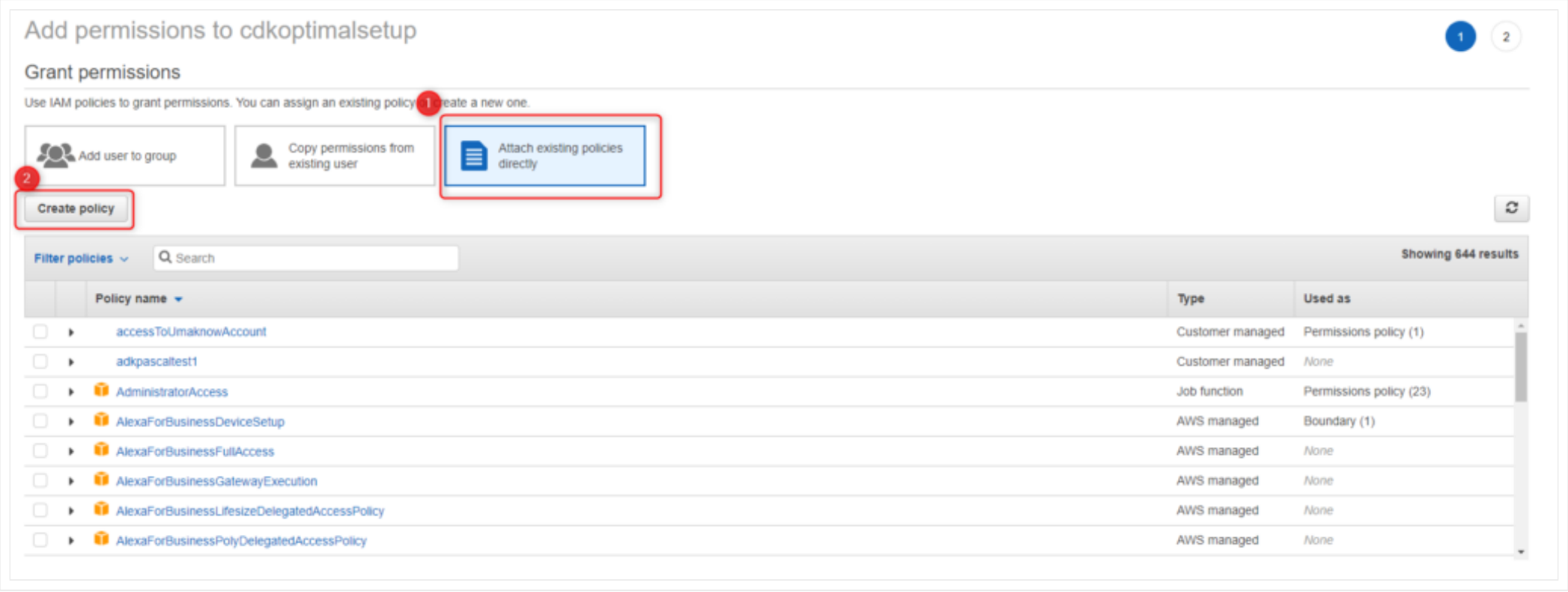


Click: Add Permissions

Add Permissions to Users

Select Attach existing policies directly.

Click: Create Policy



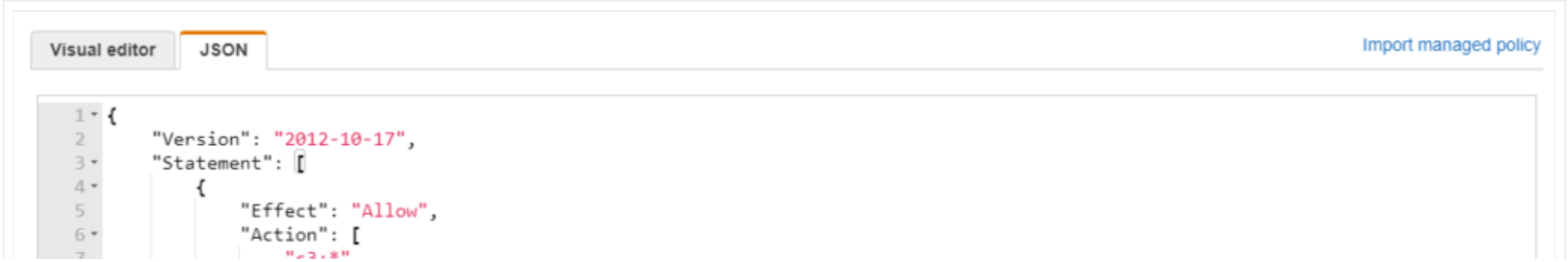
Create Policy

Select JSON tab and paste the following code in the window.

```
{
     "Version": "2008-10-17",
         "Statement": [
         {
              "Effect": "Allow",
              "Action":[ "s3:*" ],
              "Resource":[ "arn:aws:s3:::cdkoptimalsetup" ]
          }
     ]
}
```

Under the Resource value, make sure you replace arn:aws:s3:::cdkoptimalsetup with the Bucket ARN saved in the previous step.

Click: Review Policy



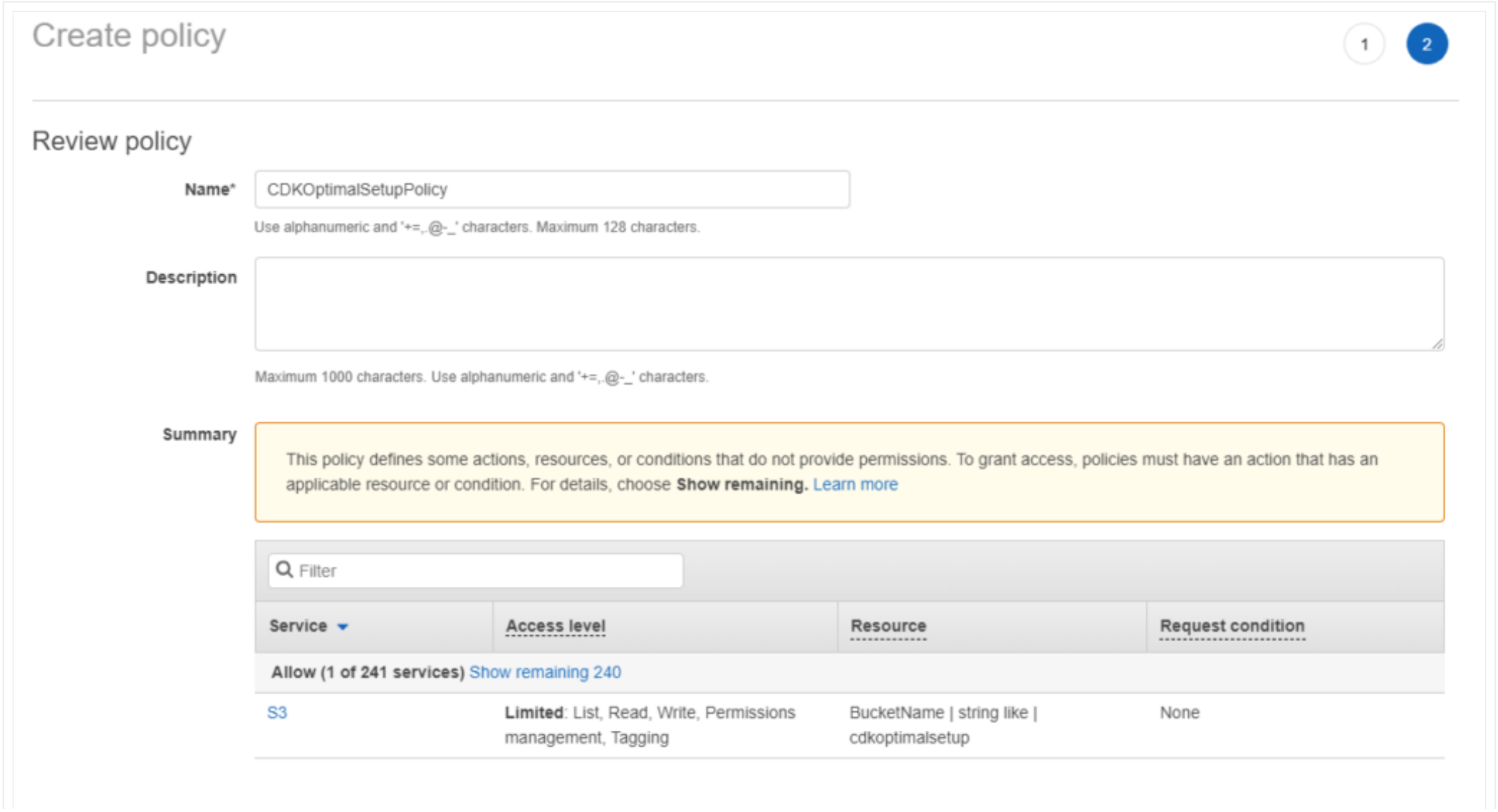
```
            ],
            "Resource": [
                "arn:aws:s3:::cdkoptimalsetup"
            ]
        }
    ]
}
```

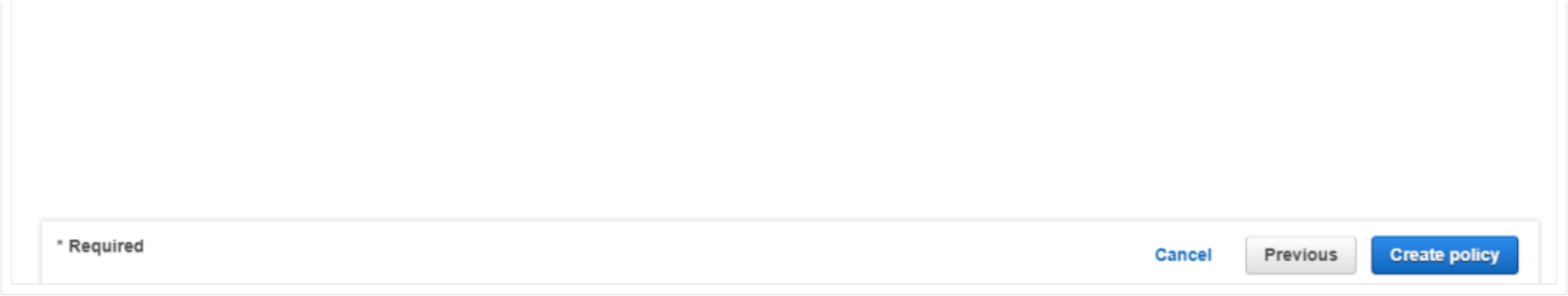
Character count: 119 of 6,144.

Cancel | Review policy

Review Policy

Give your policy a unique name, a description and click **Create Policy**.





Storage Account

From the Storage Account, click on **Permissions** and then **Bucket Policy**. Ensure you have the following statement: replace the IAM User Arn and Resource.

```
{
    "Version": "2008-10-17",
    "Id": "Policy1335892530063",
    "Statement": [
        {
            "Sid": "Stmt1335892526597",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::xxxx:user/xxxxxxx"
```

```
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::yourS3Bucket/*"
        }
    ]
}
```

Step 6 – Policies

Now that we have a user created, an EC2 Instance created as well as a storage account, it is time to apply the policies.

Policies were already given to the user account in the previous step, therefore it has read access at the account level.

AWS Billing

To read billing information from AWS, the credentials used to generate the documentation must have “**aws-portal:ViewBilling**” policy.

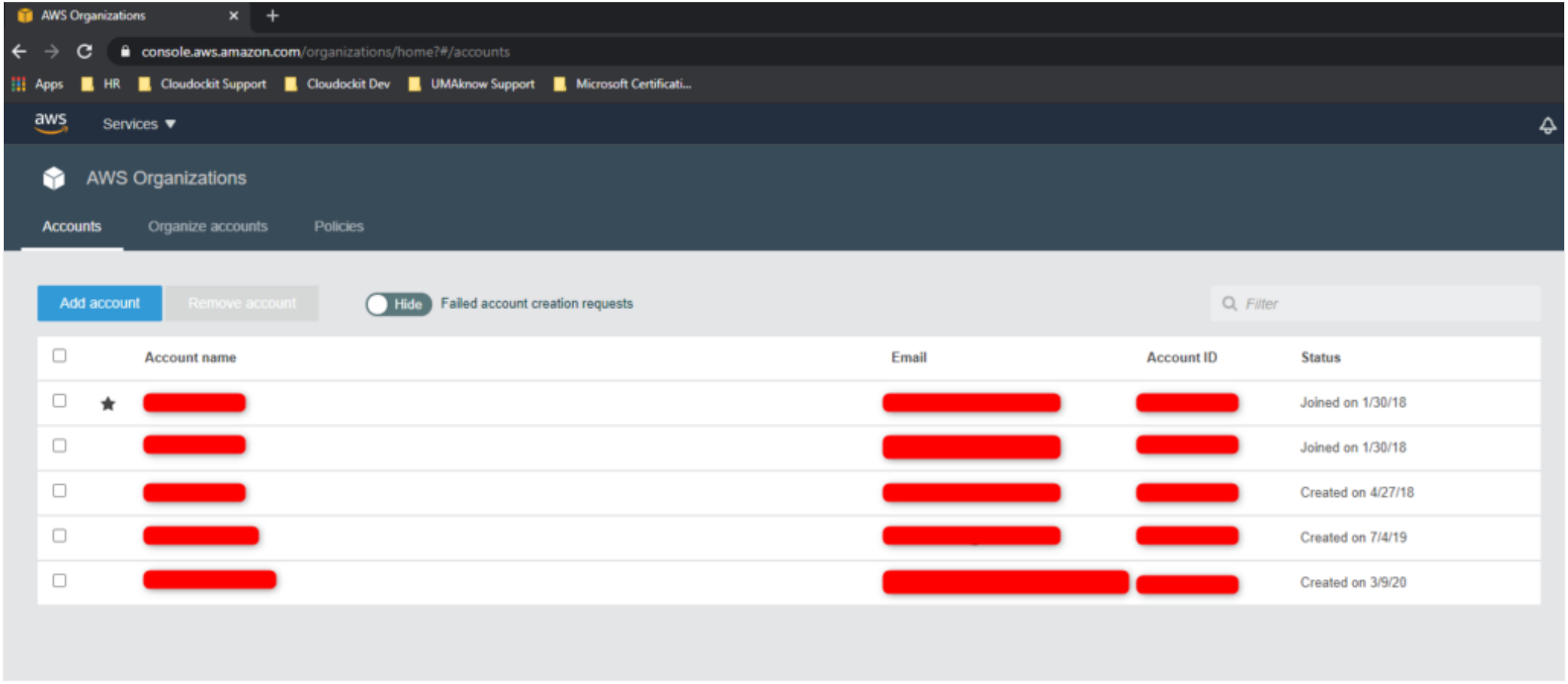
AWS Trusted Advisor

To read information from AWS Trusted Advisor, the credentials used to generate the documentation must have the following policy.

```
{
     "Version": "2008-10-17",
     "Statement": [
         {
              "Effect": "Allow",
              "Action":[
                       "ce:Get*",
                       "ce:List*",
                       "ce:Describe*"
              ],
              "Resource":[ "*" ]
         }
     ]
}
```

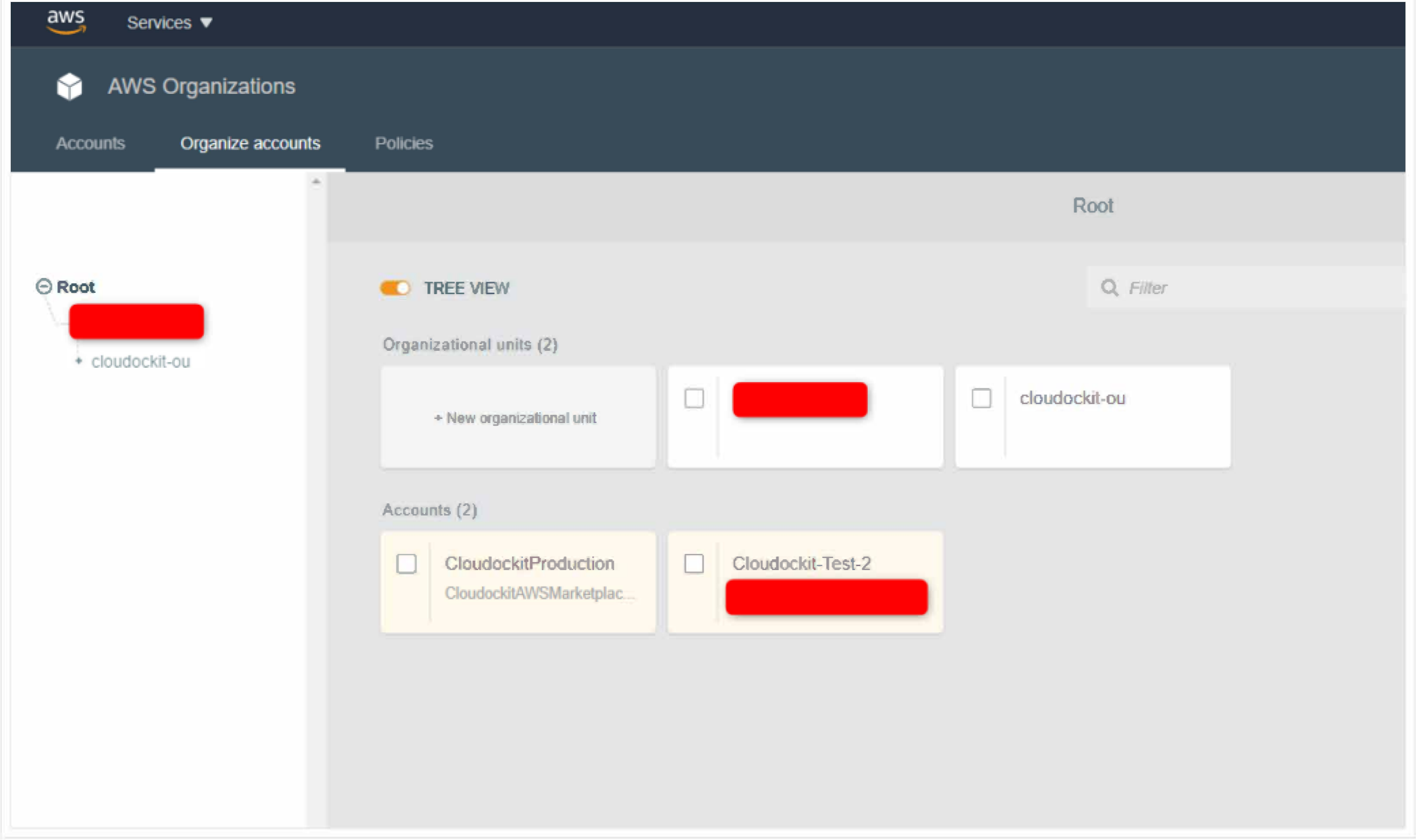
AWS Organizations Accounts

In the generated documents, if you want to view details of the accounts from your organization, you will need to choose an AWS master account when logging on Cloudockit website.



AWS Organizational Units and Accounts

If you want to view details of the organizational units and accounts of your organization in the generated documents, you will need to choose an AWS master account when logging on Cloudockit website.


aws
Services
AWS Organizations
Accounts
Organize accounts
Policies
Root
Root
cloudockit-ou
TREE VIEW
Filter
Organizational units (2)
+ New organizational unit
cloudockit-ou
Accounts (2)
CloudockitProduction
CloudockitAWSMarketplac...
Cloudockit-Test-2

AWS Member Account

When you choose an AWS member account, the generated documents will display the information of your organization and minimum information about your account (e.g.: Id, ARN).

Step 7 – AWS Cross-Account Roles

In each AWS Account you want to scan, you need to create a role named CloudockitScanRole (or any name that you prefer).

Here are the steps to create this role:

From IAM console, click on Roles and then Create role.

- Select Another AWS Account. Enter the Account ID where you are installing the EC2 instance that will run Cloudockit Desktop.
- Click Next and select the *ReadOnlyAccess*
- Click Next: Review
- Enter the name: CDKOptimalScanRole (Or which ever name you gave it)
- Click on create role
- Repeat those steps in all AWS Accounts

Step 8 – Launch Cloudockit Desktop and Schedule a Document Generation

Connect to the EC2 Instance created.

Launch Cloudockit Desktop.

Activating Cloudockit

Click on the desktop shortcut of Cloudockit to launch the application.

You will need to enter your product key to activate Cloudockit Desktop.

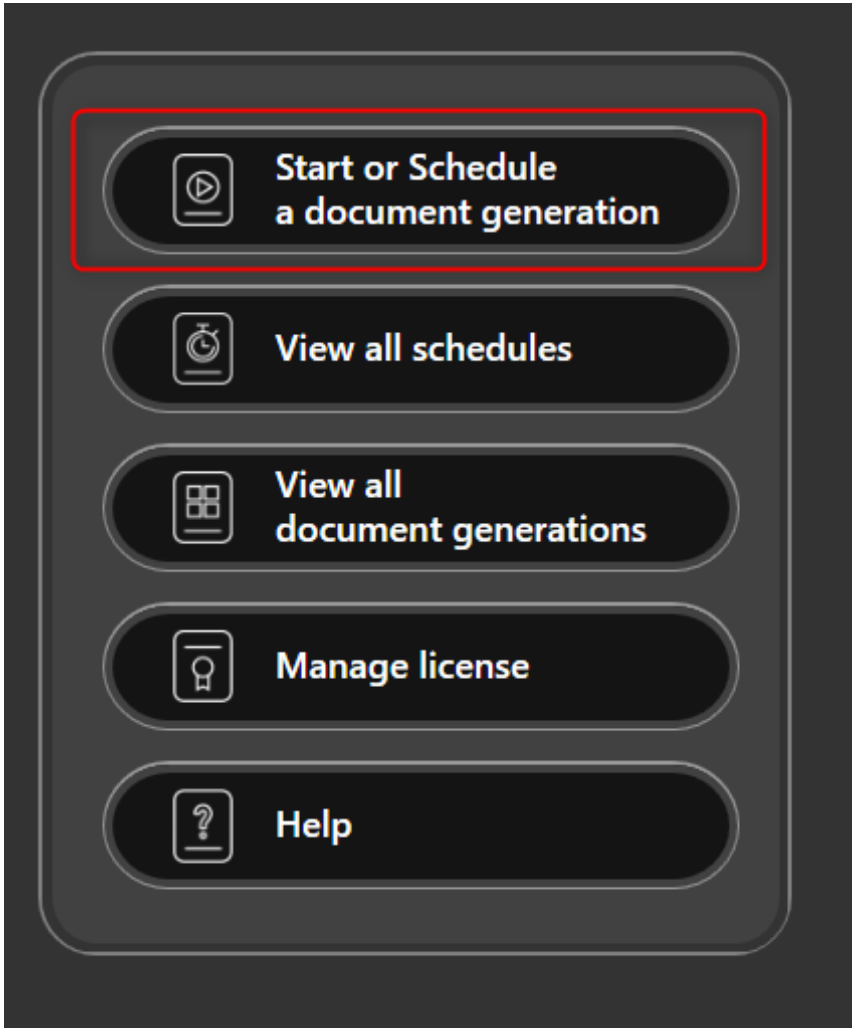
If you haven’t purchased a product key, please visit https://www.cloudockit.com/pricing/

You will see a message confirming that the activation was done successfully.

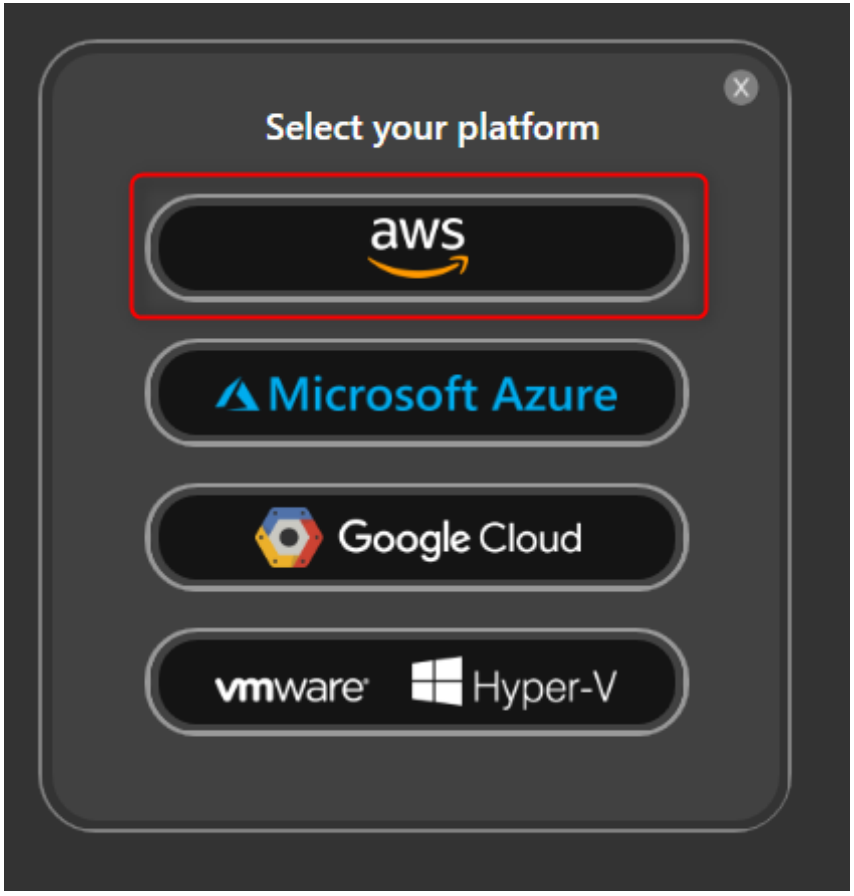
Click: OK

Connecting to an AWS Platform

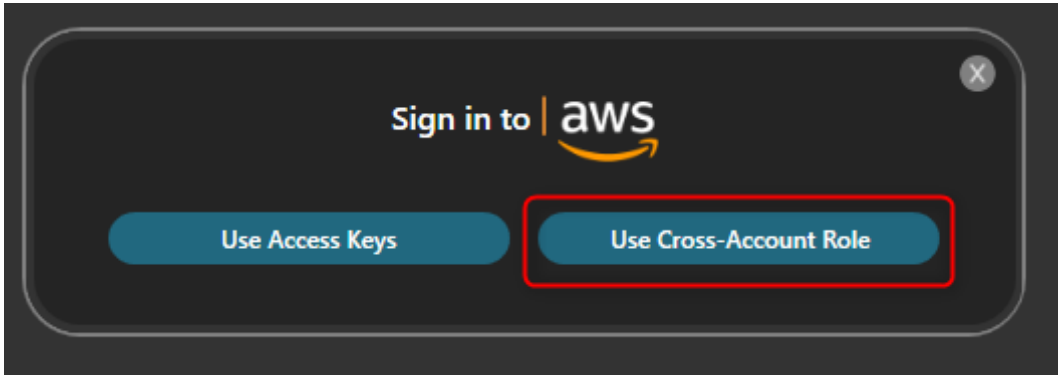
Select Start or Schedule a Document Generation



Select Cross-Account Role.

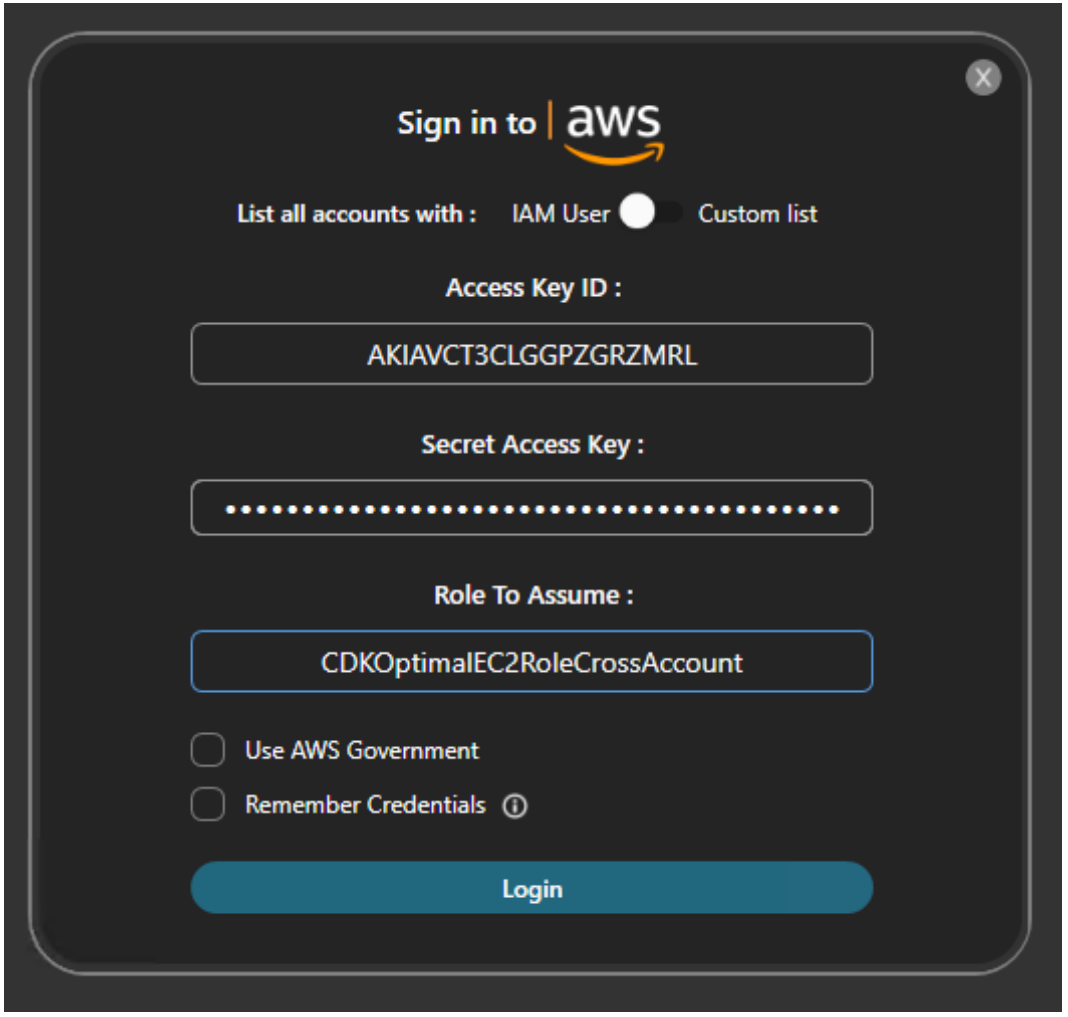


Select AWS from the list of platforms.



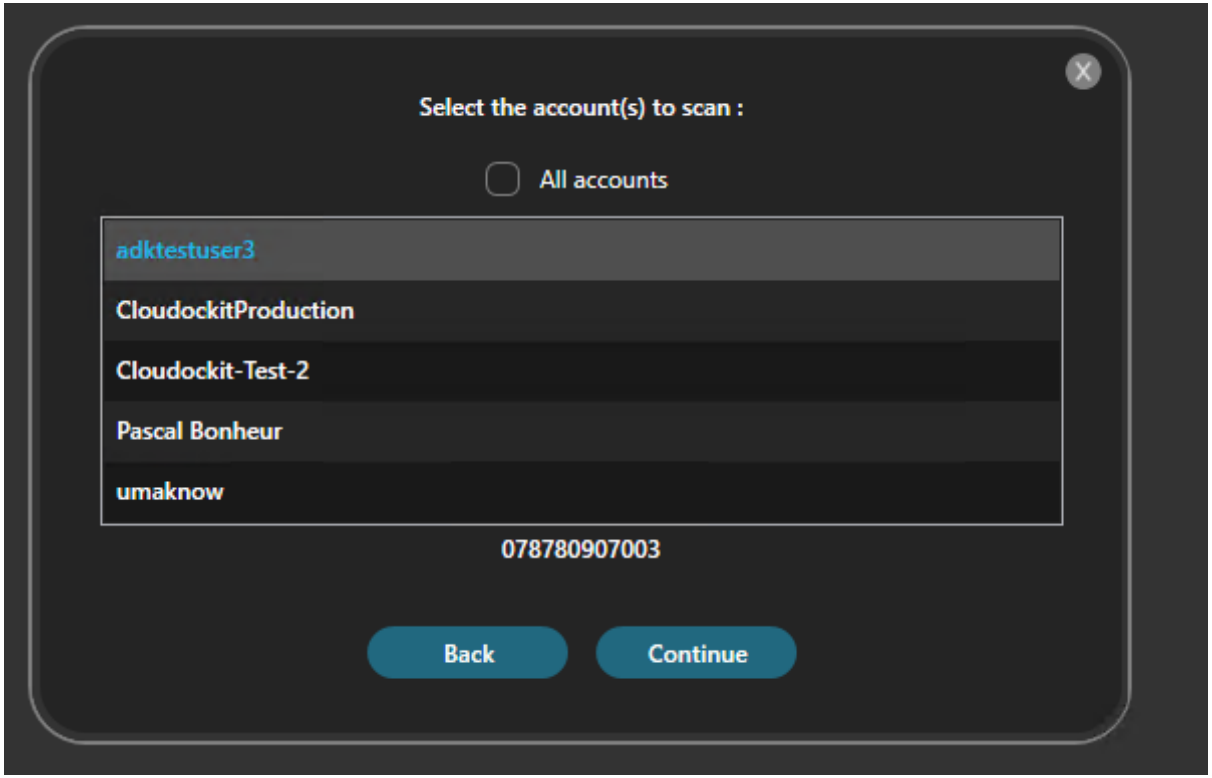
Enter the Access Key ID, Secret Access Key, and Role to Assume.

Click: Login



You now need to select the accounts you want to document.

Click: Continue



Schedule a Document Generation

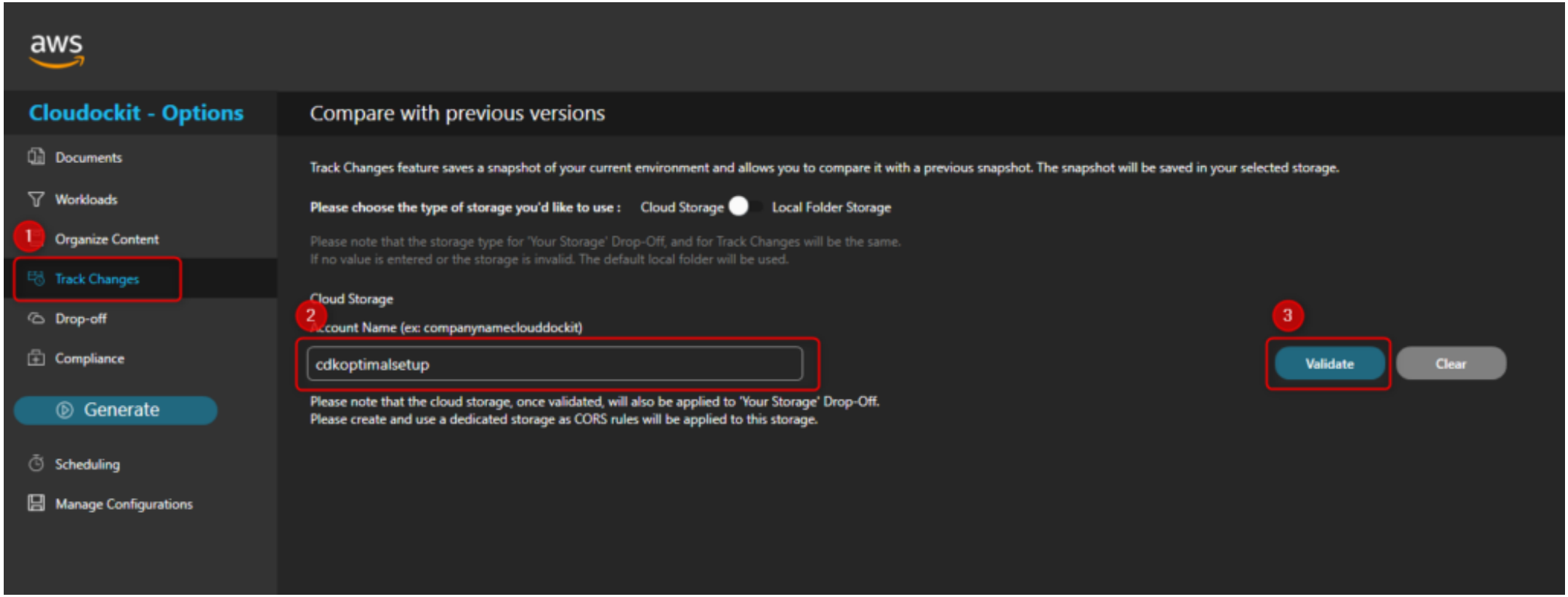
Now that you are logged in, it is time to define what information you want to generate using Cloudockit.

Set the desired parameters under Documents, Workloads, and Organize Content.

Track Changes

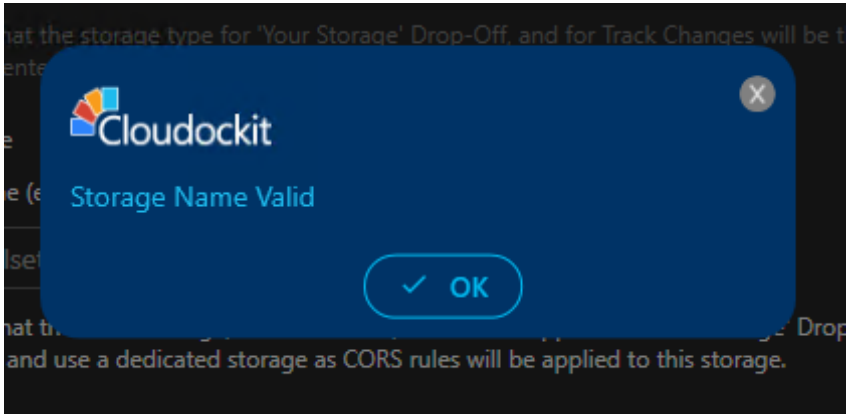
Use the storage account created previously for track changes. This will allow you to see the differences that have occurred between a previous document generation and the one running right now.

Select **Track Changes** from the left menu.



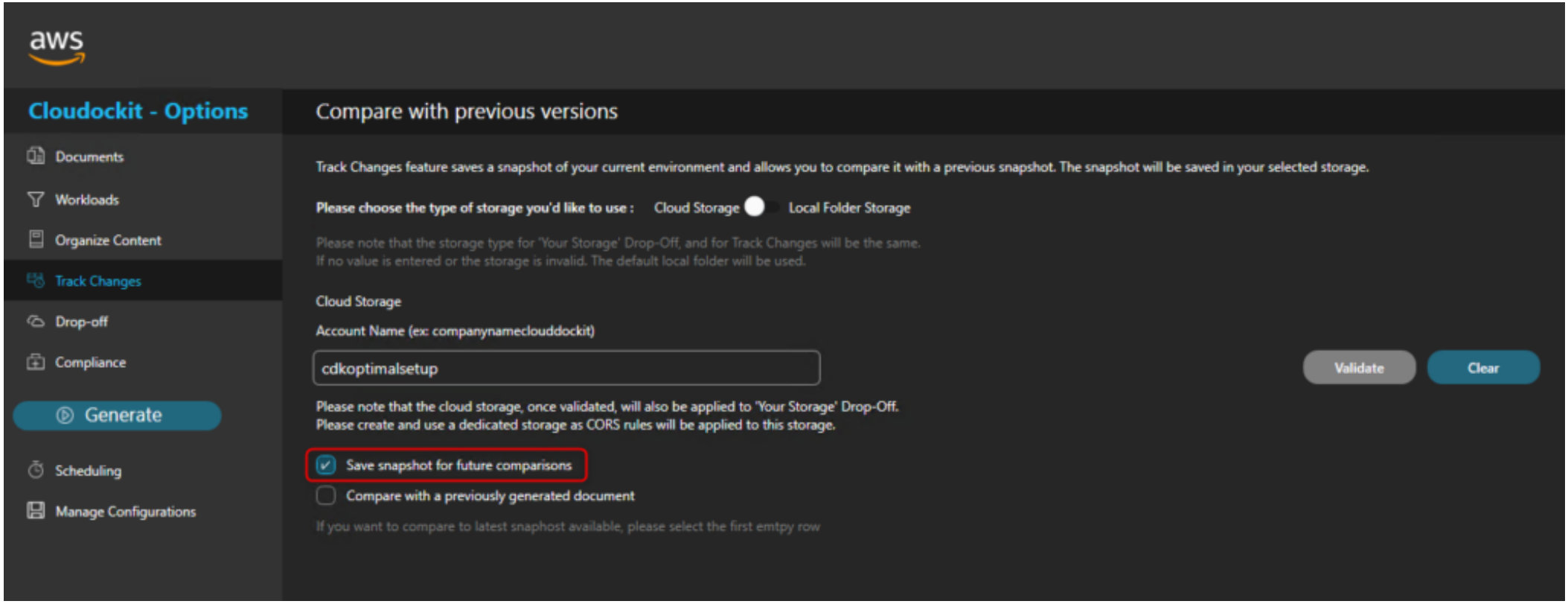
Enter the name of the bucket in the Account Name box and press **validate**.

A confirmation message will display that the bucket is valid.



Check the box **Save a snapshot for comparison.**

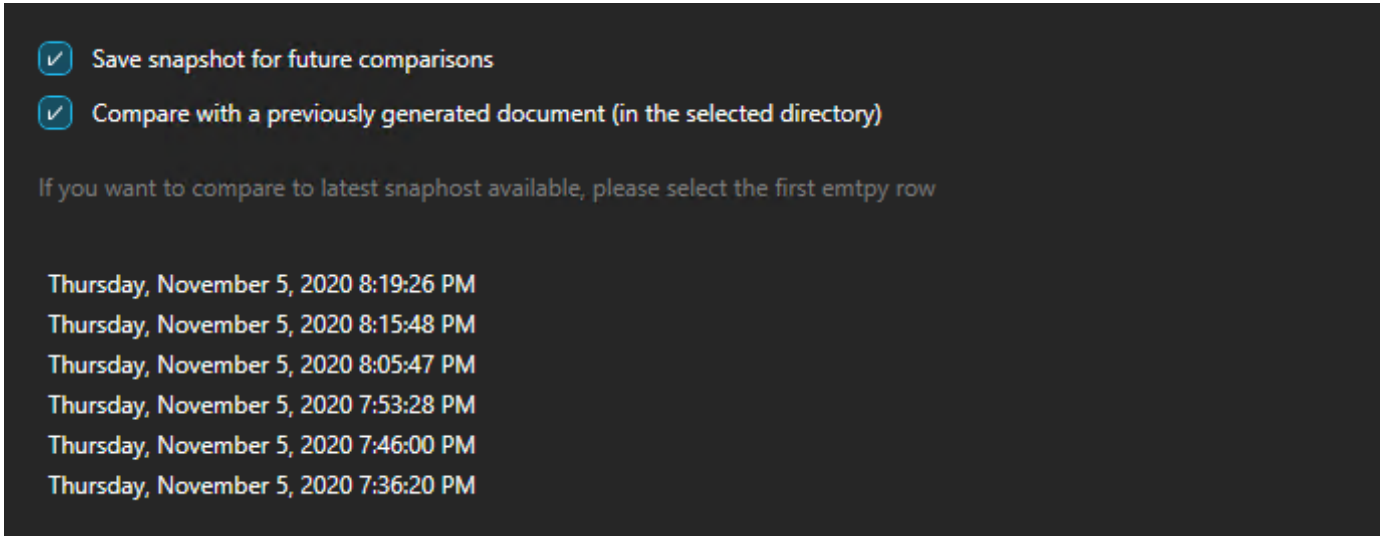
This will save a JSON file in the storage account every time a document generation runs.



Check the box **Compare with a previously generated document.**

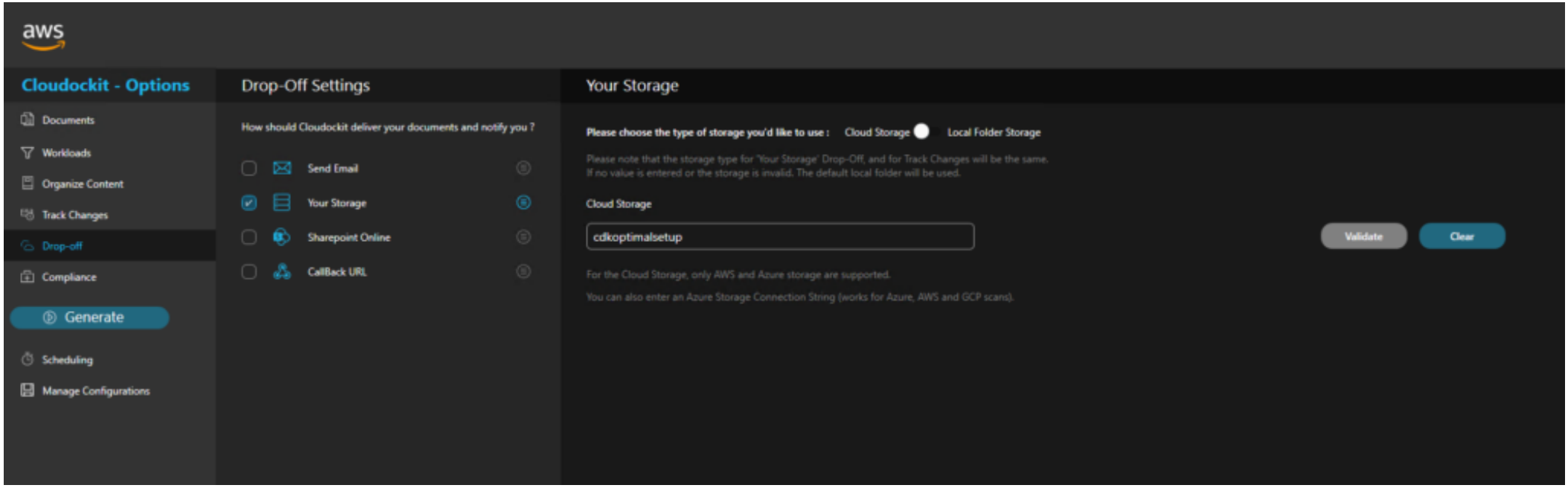
Select the first empty row that appears below.

This will always select the most recent file in the storage account to compare.



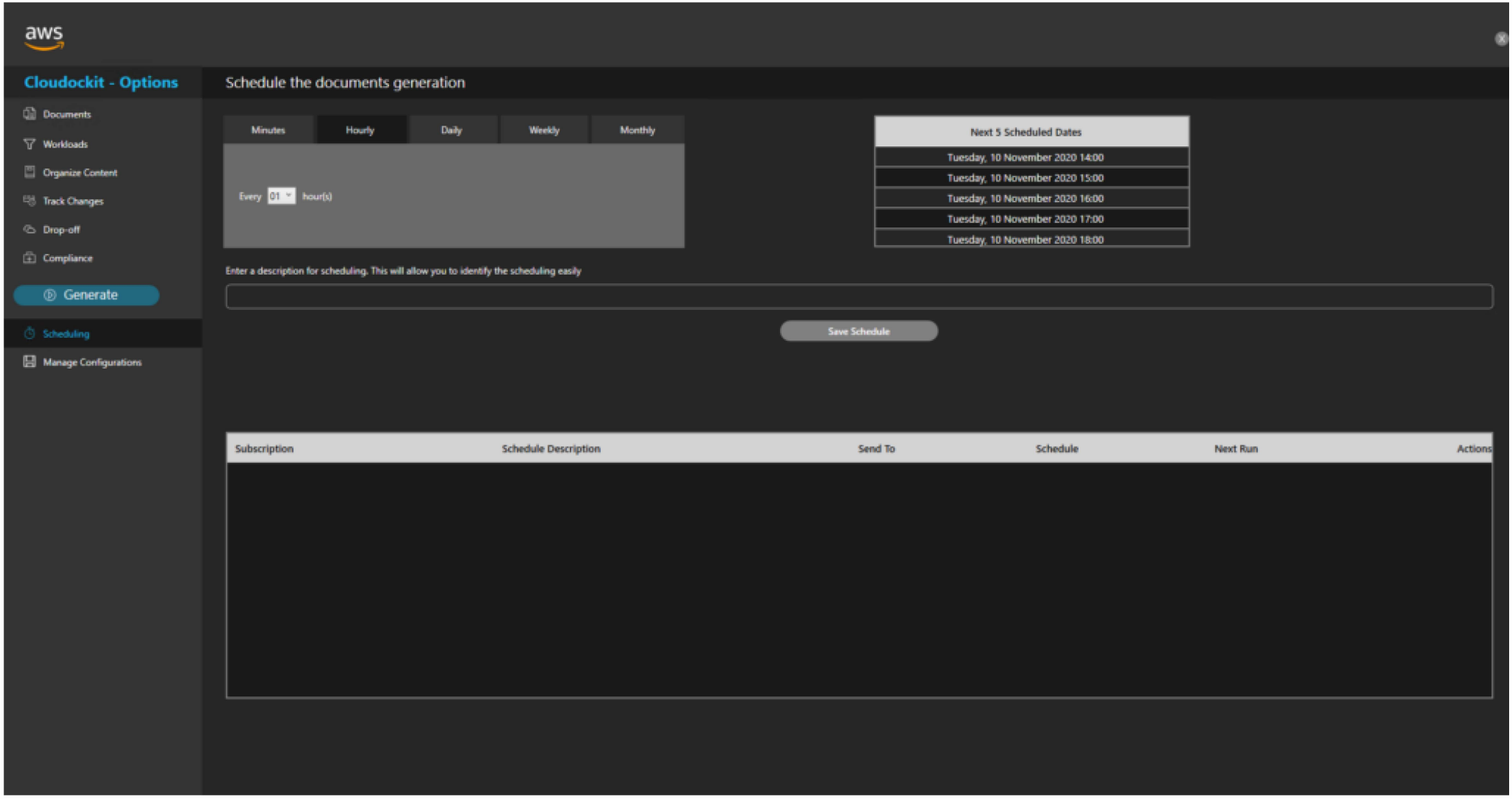
Drop-Off

In the Drop-Off settings, the same bucket as defined in the Track changes section is selected.



Scheduling

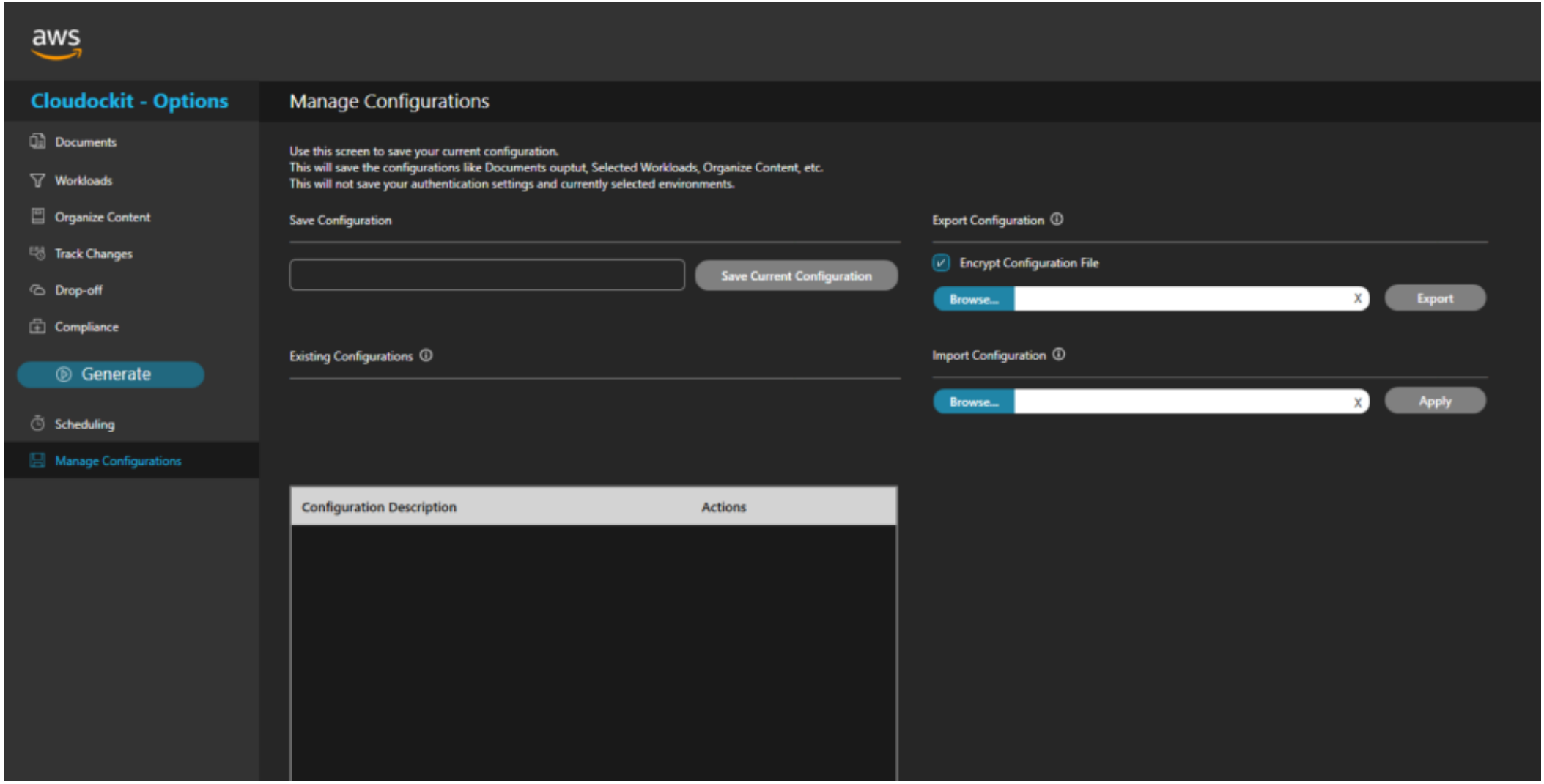
Define the desired schedule for your documentation to run and save your schedule.


aws
Cloudockit - Options
Documents
Workloads
Organize Content
Track Changes
Drop-off
Compliance
Generate
Scheduling
Manage Configurations
Schedule the documents generation
Minutes
Hourly
Daily
Weekly
Monthly
Every 01 hour(s)
Next 5 Scheduled Dates
Tuesday, 10 November 2020 14:00
Tuesday, 10 November 2020 15:00
Tuesday, 10 November 2020 16:00
Tuesday, 10 November 2020 17:00
Tuesday, 10 November 2020 18:00
Enter a description for scheduling. This will allow you to identify the scheduling easily
Save Schedule
Subscription
Schedule Description
Send To
Schedule
Next Run
Actions

Configuration

Enter a unique name in the parameters you have set and press **Save Current Configuration**.

Your configuration is saved, you can load or edit it in the future.



Step 9 – Validate that Documents are Successfully Generated

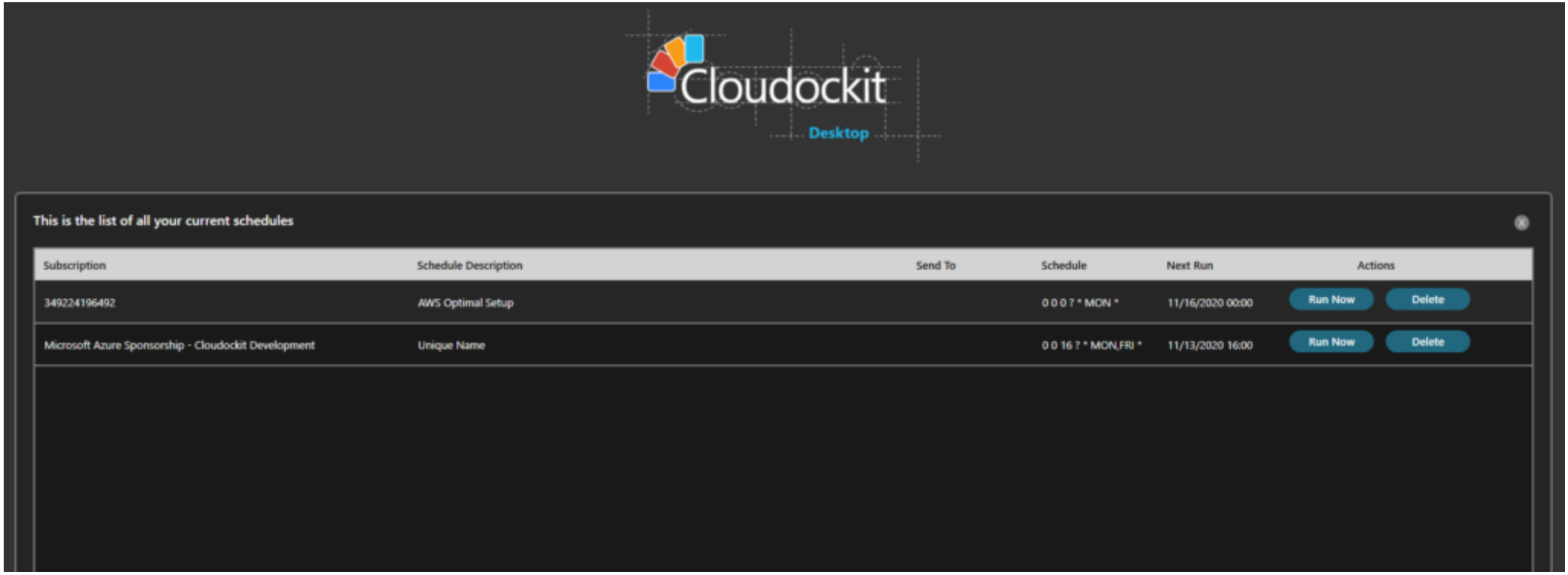
Once your scheduled document generation is complete, let’s validate that it has been scheduled properly.

From the main menu, select **View All schedules.**

You will see in the list the scheduled documentation you configured.

You can press run now to generate a manual document generation or wait for the schedule to run its course.

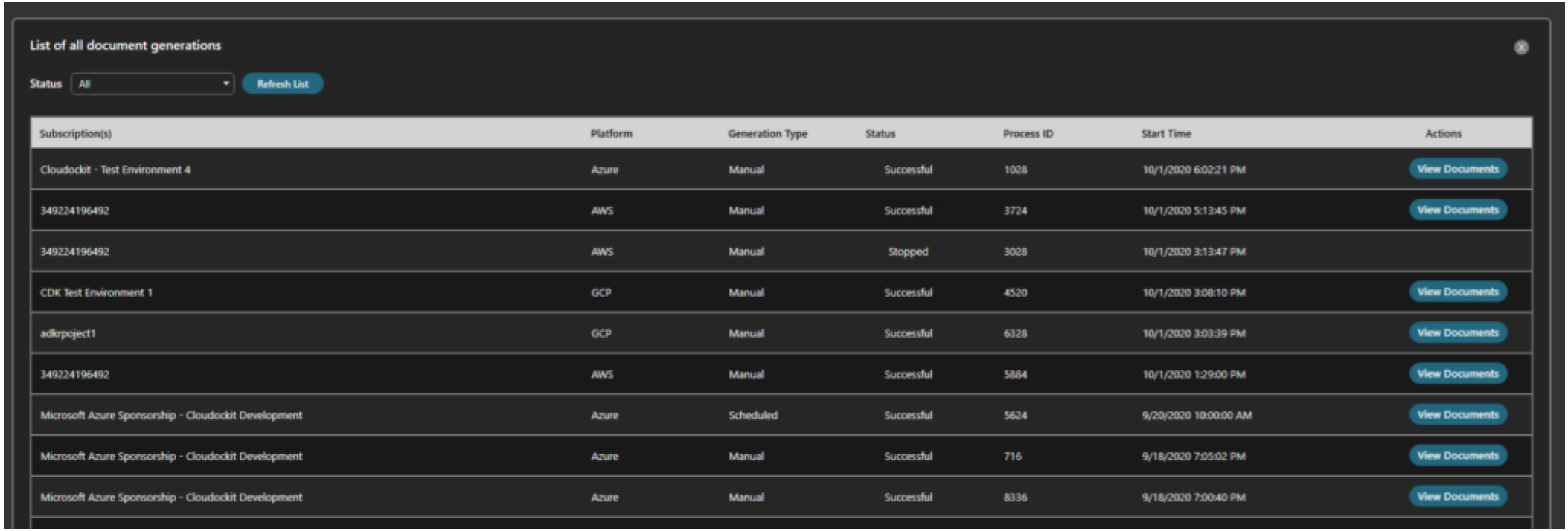
Once your document will be completed, you will be able to access it from the Storage Account or from the desktop application.



Click **View all document generations** from the main menu.

You have the list of all generated documents.

You can access the documents from the View Documents button on the right.



LATEST ON OUR BLOG

Great new Cloudockit Features Including Compliance Rules, AWS Backups, Google Cloud Billing, and More.

Great new features for Azure and AWS users plus download Cloudockit Desktop easily