Introduction to Research on AWS with Jupyter Notebooks Hands-On Lab

Getting Started AWS Core Services

Workshop Overview

This lab uses Jupyter Notebooks and AWS Python APIs to showcase several AWS Core services. We will launch a Juypter Notebook in Amazon Sagemaker, then deploy a simple webserver for demonstration purposes.

Identity & Access Management (IAM) Overview

AWS Identity and Access Management (IAM) is a core service that enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users, groups, and roles, then specify fine-grained permissions to allow or deny their access to specific AWS resources.

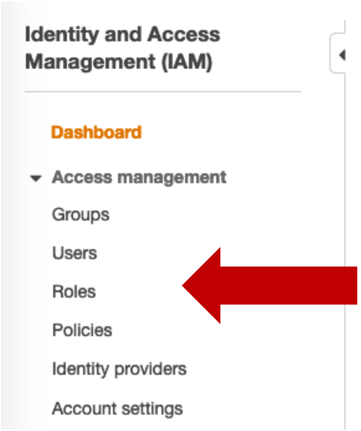
Create an IAM role

To use AWS APIs within your Amazon Sagemaker Notebook, you will need to first create an IAM role. IAM Roles can be assumed by AWS services, IAM users, or applications. They are assigned temporary rather than permanent credentials whenever assumed. Using roles for privileged permissions sets can help improve your security posture since credential exposure is minimized.

To accomplish this lab, you will create a role with full Admin rights. This can be a risk, and should not be used in production

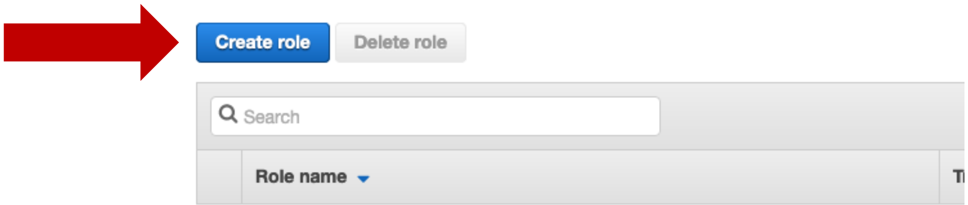
1. Sign into the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/iam.

2. Select **Roles** from the sidebar or go to https://console.aws.amazon.com/iam/home#/roles

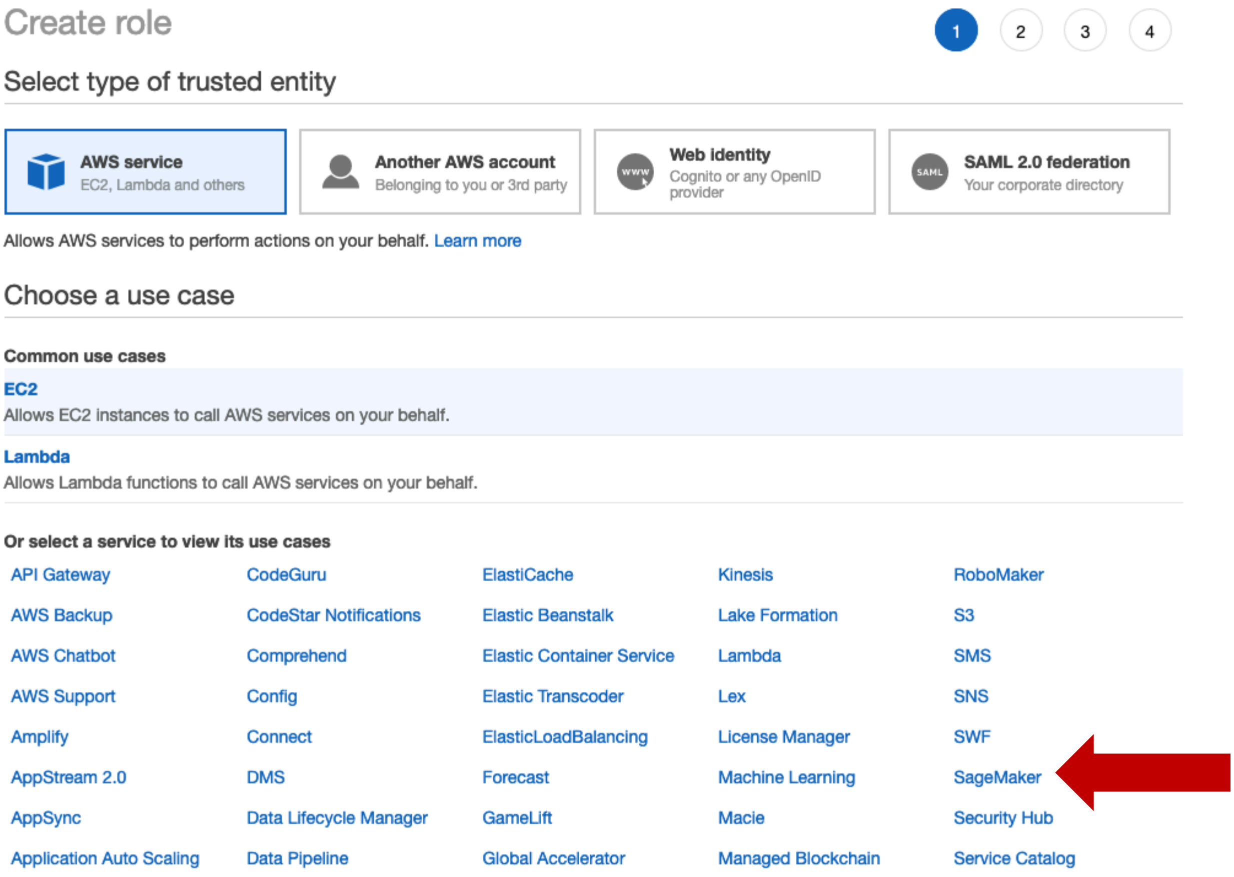


Note: the AWS console might look slightly different as we are constantly working to improve the user experience for our customers.

3. Click **Create Role**



4. On the **Select type of trusted identity** page, you decide *who* or *what* will be able to assume this role. For this lab, we will create a role that allows a *Sagemaker instance* call AWS APIs. Therefore, we will stay on the **AWS service** tab and select **Sagemaker**. Go to **Next: Permissions**.



Select type of trusted entity

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

Choose a use case

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.

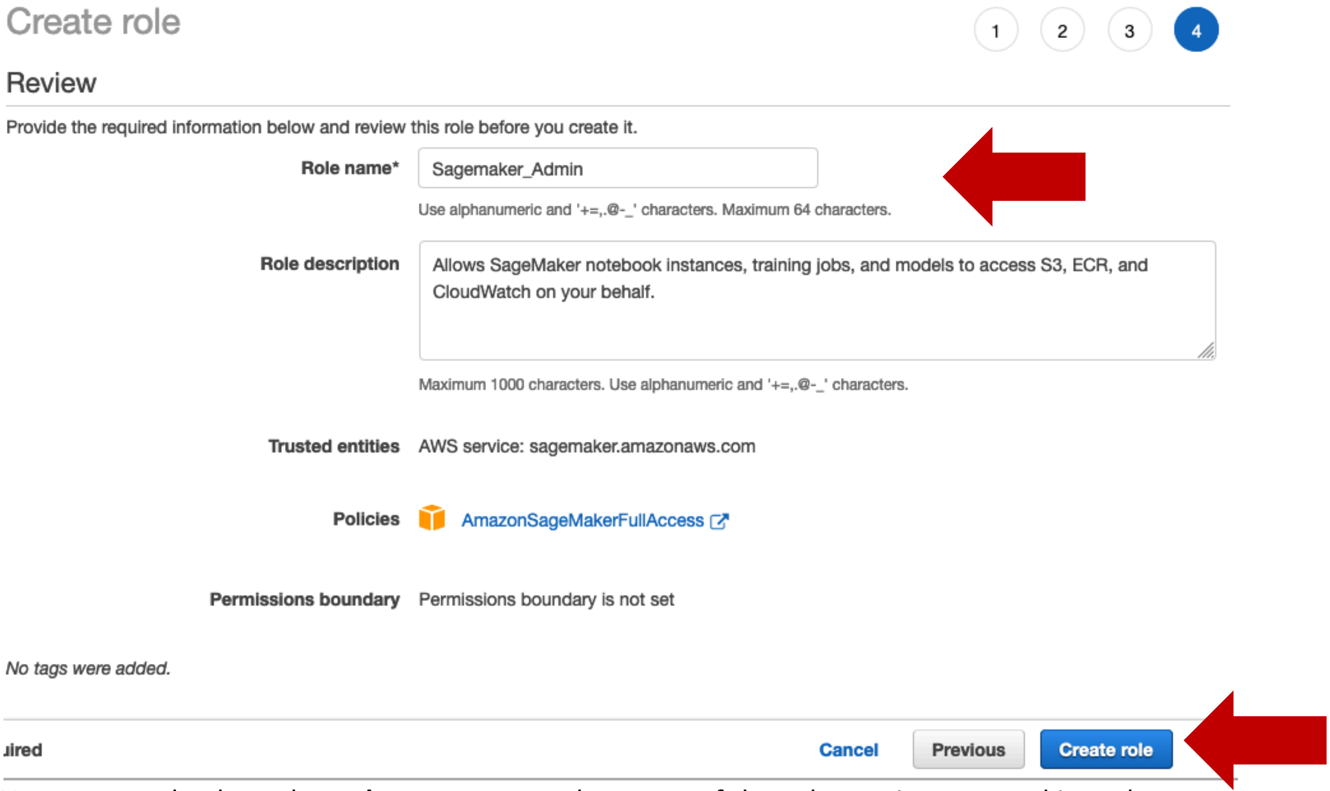
Or select a service to view its use cases

API Gateway
AWS Backup
AWS Chatbot
AWS Support
Amplify
AppStream 2.0
AppSync
Application Auto Scaling
CodeGuru
CodeStar Notifications
Comprehend
Config
Connect
DMS
Data Lifecycle Manager
Data Pipeline
ElastiCache
Elastic Beanstalk
Elastic Container Service
Elastic Transcoder
ElasticLoadBalancing
Forecast
GameLift
Global Accelerator
Kinesis
Lake Formation
Lambda
Lex
License Manager
Machine Learning
Macie
Managed Blockchain
RoboMaker
S3
SMS
SNS
SWF
SageMaker
Security Hub
Service Catalog

5. Go to **Next: Tags**.
There is no need to add Tags for this example. Tags are useful to group related resources, using a key:value metadata scheme, such as "project=myproject1".

6. Go to **Next: Review**

7. Give your role a descriptive name, such as **Sagemaker_Admin** and edit the **Role description** field to be a helpful summary of what this role is. When you're done, click on **Create Role**.



Provide the required information below and review this role before you create it.

Role name* Sagemaker_Admin

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

Role description Allows SageMaker notebook instances, training jobs, and models to access S3, ECR, and CloudWatch on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: sagemaker.amazonaws.com

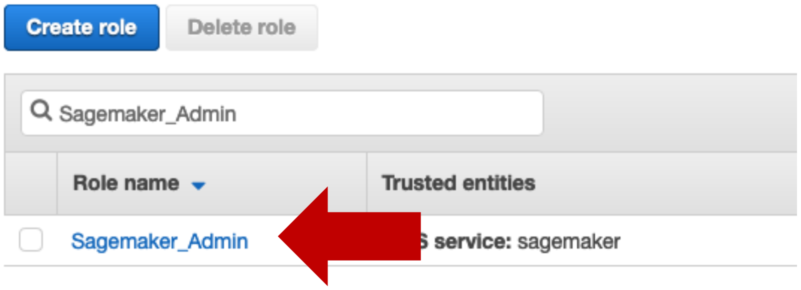
Policies AmazonSageMakerFullAccess

Permissions boundary Permissions boundary is not set

No tags were added.

ıired

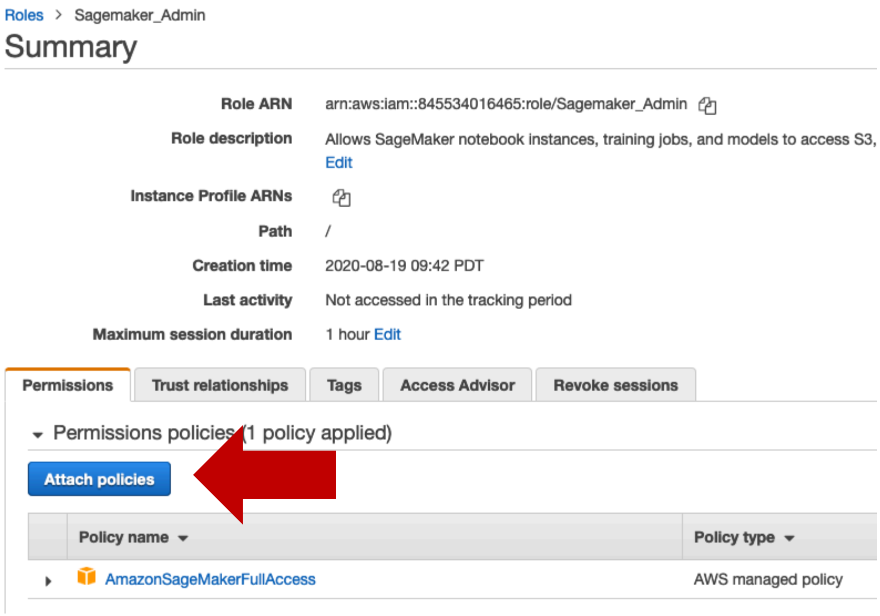
8. You are now back on the **Roles** page. Enter the name of the role you just created into the search bar and click on the role name.



Sagemaker_Admin

	Role name	Trusted entities
□	Sagemaker_Admin	S service: sagemaker

9. You are now on the **Summary** page of the role you just created. Here you can view and edit attributes of the role. Click on **Attach Policies**



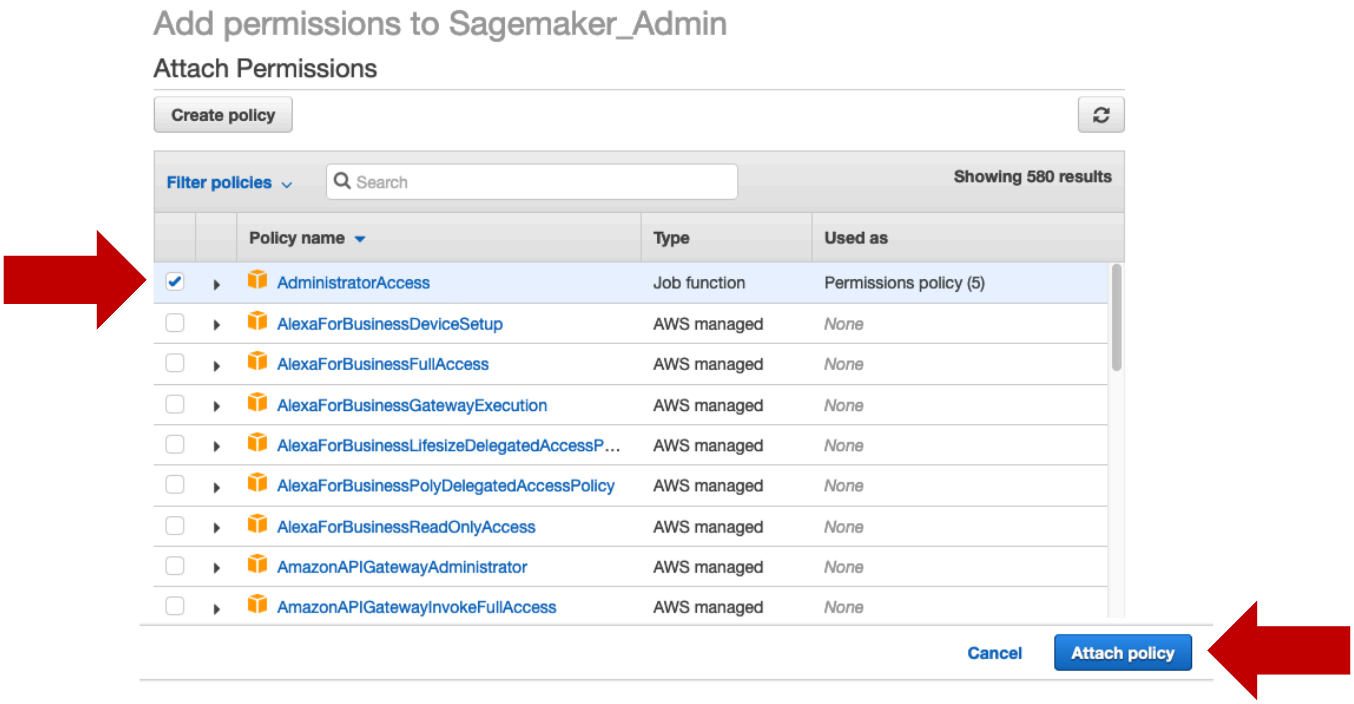
Roles > Sagemaker_Admin

Summary

Role ARN	arn:aws:iam::845534016465:role/Sagemaker_Admin
Role description	Allows SageMaker notebook instances, training jobs, and models to access S3, Edit
Instance Profile ARNs	
Path	/
Creation time	2020-08-19 09:42 PDT
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Policy name	Policy type
AmazonSageMakerFullAccess	AWS managed policy

10. Click the checkbox next to **AdministratorAccess**, then Click on **Attach policy**.



Add permissions to Sagemaker_Admin

Attach Permissions

	Policy name	Type	Used as
☑	AdministratorAccess	Job function	Permissions policy (5)
□	AlexaForBusinessDeviceSetup	AWS managed	None
□	AlexaForBusinessFullAccess	AWS managed	None
□	AlexaForBusinessGatewayExecution	AWS managed	None
□	AlexaForBusinessLifesizeDelegatedAccessP...	AWS managed	None
□	AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None
□	AlexaForBusinessReadOnlyAccess	AWS managed	None
□	AmazonAPIGatewayAdministrator	AWS managed	None
□	AmazonAPIGatewayInvokeFullAccess	AWS managed	None

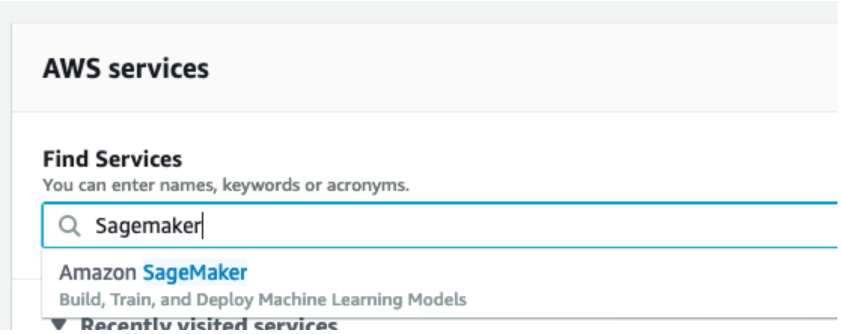
Congratulations! You've just created an IAM role which will allow Sagemaker Notebook instances in your account to assume this role and call AWS APIs.

Launch an Amazon Sagemaker Notebook

For this lab, we will be using Amazon Sagemaker built-in Notebook instances. Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning (ML) models quickly. SageMaker removes the heavy lifting from each step of the machine learning process to make it easier to develop high quality models. An Amazon SageMaker notebook instance is a machine learning (ML) compute instance running the Jupyter Notebook App. Amazon SageMaker manages creating the instance and related resources automatically.

Upon logging into your AWS Console, you should ALWAYS check which region you are operating in. This can be found in the top right of your Console window.

1. Sign into the AWS Management Console and On the AWS Console home page, type *Sagemaker* into the service search bar and select it.



AWS services

Find Services

You can enter names, keywords or acronyms.

Sagemaker

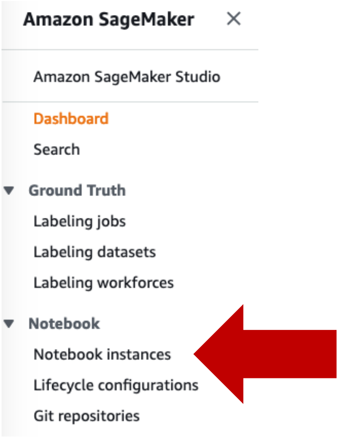
Amazon SageMaker

Build, Train, and Deploy Machine Learning Models

▼ Recently visited services

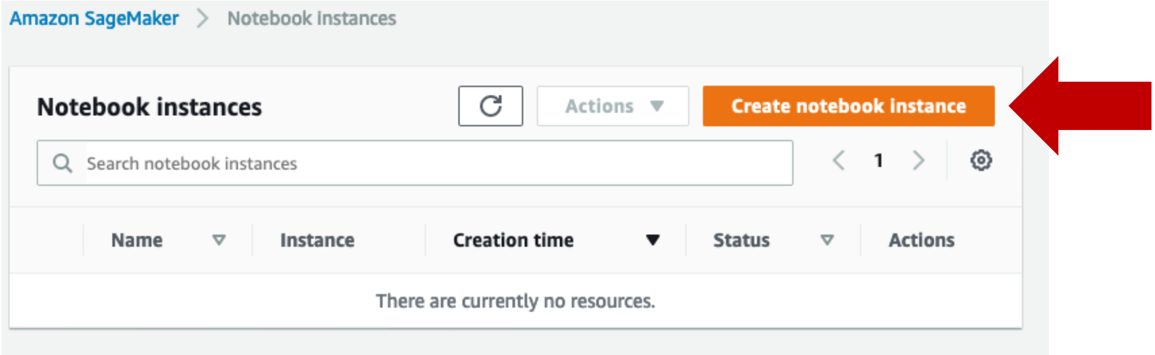
2. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

3. Choose **Notebook Instances** from the left sidebar:



Note: Sagemaker also provides a full-blown IDE named Sagemaker Studio. For this lab, we will use just the simpler Notebook instances, yet the examples can be run from Sagemaker Studio as well.

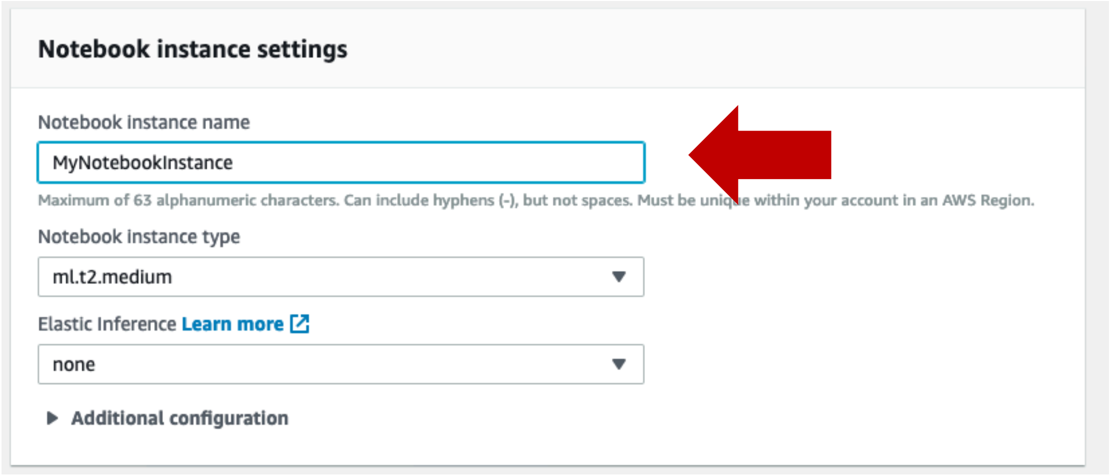
4. Click **Create notebook instance**



Name	Instance	Creation time	Status	Actions

There are currently no resources.

5. Give your notebook an instance a name, such as **MyNotebookInstance**



Notebook instance settings

Notebook instance name: MyNotebookInstance

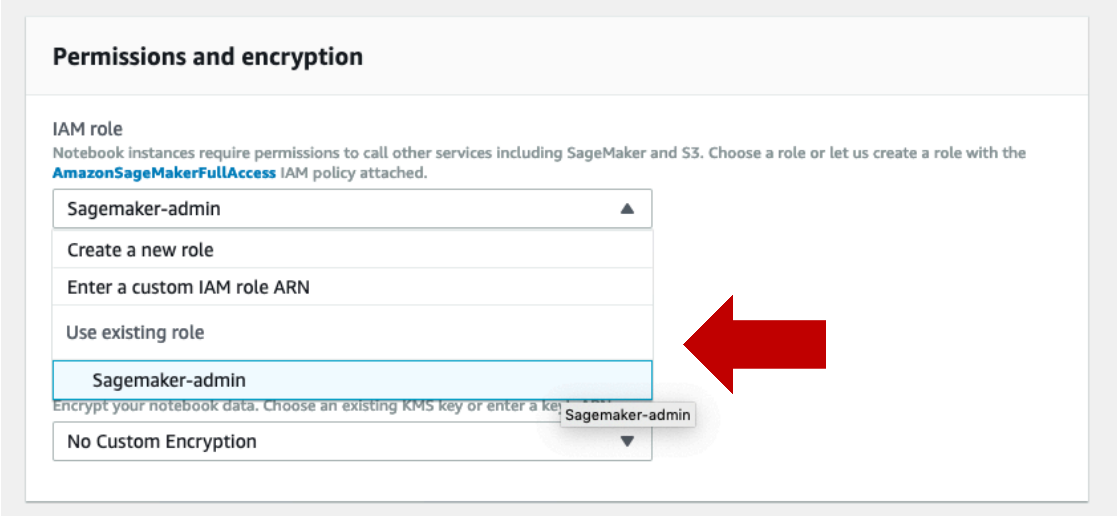
Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type: ml.t2.medium

Elastic Inference Learn more: none

If you wanted a more powerful instance, you can select different instances from the "Notebook instance type" dropdown.

6. Under **IAM Role**, choose the IAM role you created earlier and click **Create Notebook Instance**



Permissions and encryption

IAM role

Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the **AmazonSageMakerFullAccess** IAM policy attached.

Sagemaker-admin ▲

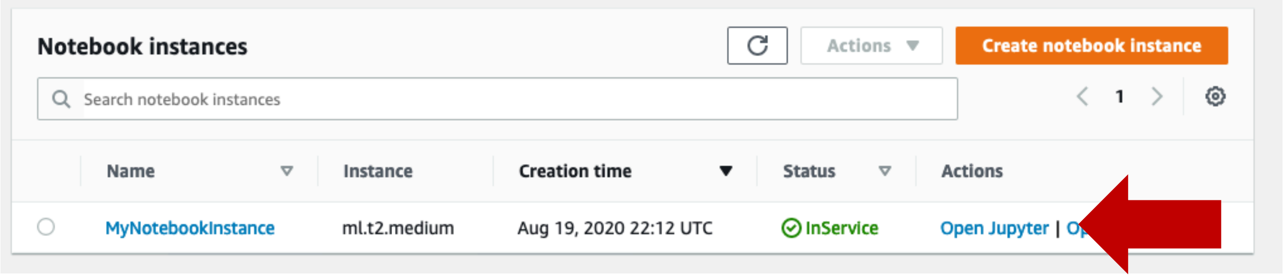
- Create a new role
- Enter a custom IAM role ARN
- Use existing role
  - Sagemaker-admin

Encrypt your notebook data. Choose an existing KMS key or enter a key

Sagemaker-admin

No Custom Encryption ▼

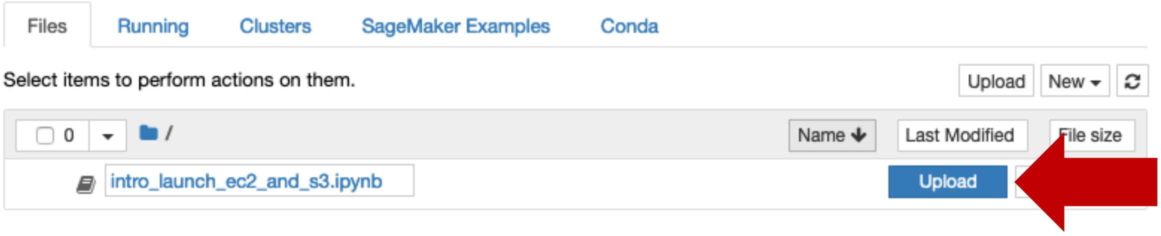
7. When the notebook instance is depicted in the **InService** state, click on **Open Jupyter**



Notebook instances

	Name	Instance	Creation time	Status	Actions
○	MyNotebookInstance	ml.t2.medium	Aug 19, 2020 22:12 UTC	✓ InService	Open Jupyter | Op

8. Upload the Jupyter Notebook that you downloaded earlier and launch it.

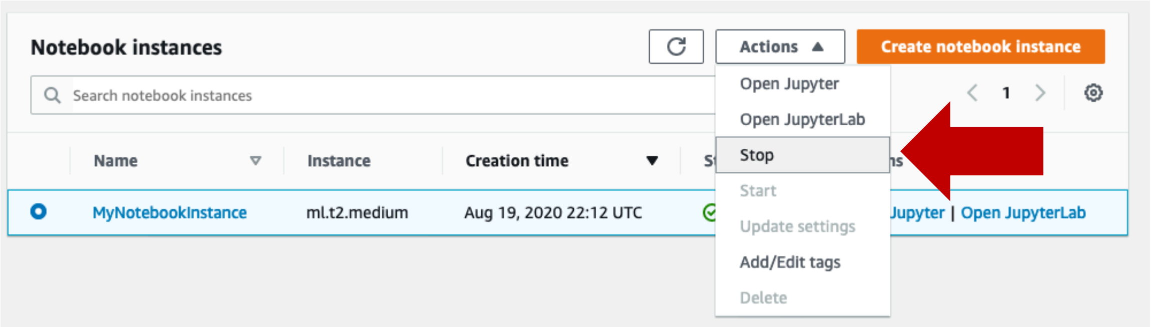


Select items to perform actions on them.

intro_launch_ec2_and_s3.ipynb

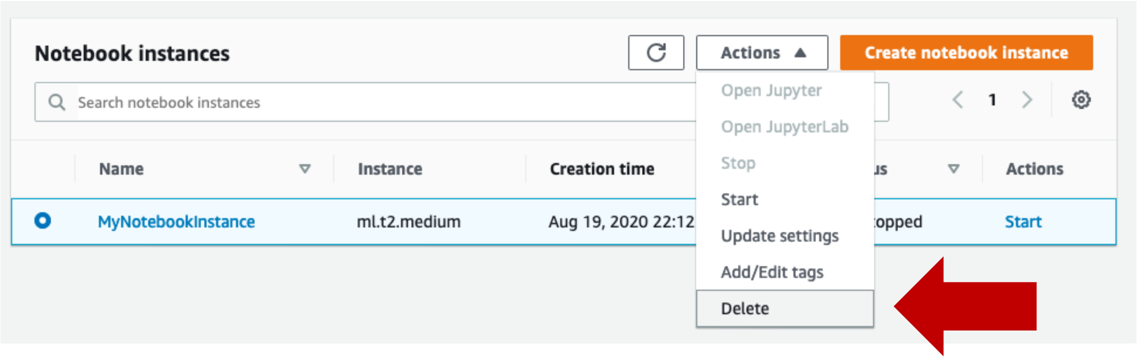
Delete the Sagemaker notebook instance

1. In the Amazon Sagemaker Notebook Instances console, select the notebook instance you intend to delete. Click the **Actions** button, and select **Stop**.



	Name	Instance	Creation time	S	ns
●	MyNotebookInstance	ml.t2.medium	Aug 19, 2020 22:12 UTC		Jupyter | Open JupyterLab

2. Once the notebook instance has stopped, click the **Actions** button again, and select **Delete**. Click the **Delete** button to confirm.

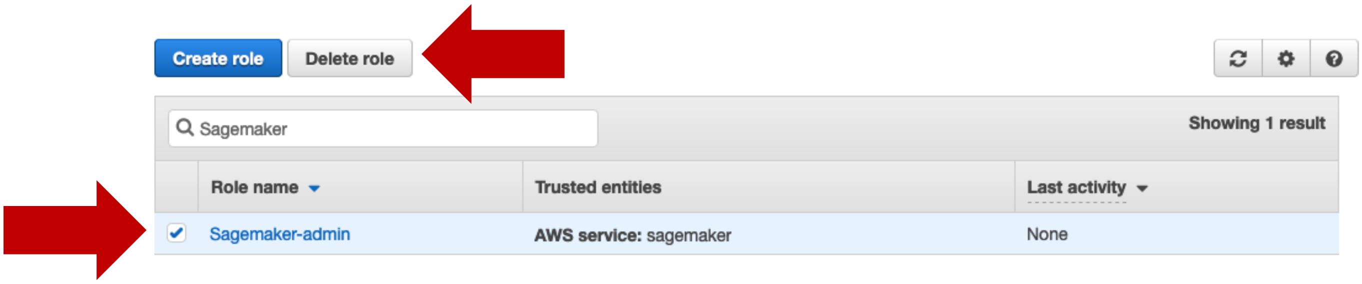


	Name	Instance	Creation time	us	Actions
●	MyNotebookInstance	ml.t2.medium	Aug 19, 2020 22:12	opped	Start

Well done, your notebook instance is now deleted!

Delete an IAM role

1. Now we need to remove our IAM Role. Navigate to the IAM Roles console, and search for your role, click the checkbox next to it and click the **Delete role** button.



	Role name	Trusted entities	Last activity
☑	Sagemaker-admin	AWS service: sagemaker	None

Congratulations you have deleted the IAM role!