



INTRODUCTION TO AMAZON WEB SERVICES

MAKEATHON QUICK START GUIDE

November 2025

TABLE OF CONTENTS

1.LOGIN

2.ROLES & PERMISSIONS

3.ACCESS KEYS

4.SAGEMAKER AI PLATFORM

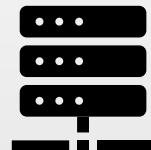
5.S3 STORAGE

JUST TAKE YOUR PICK

Cloud infrastructure for equal opportunities

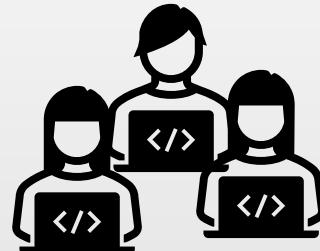


FastAPI



jupyter

You can use any
technology you like!



Team



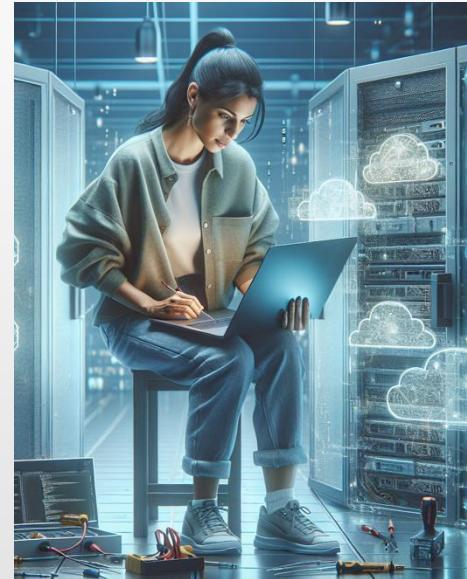
HINTS



→ Do your research!



→ Ask us for help!



→ We try to make things possible!



LOGIN



LOGIN

<https://eu-north-1.signin.aws.amazon.com/>

IAM user sign in [\(i\)](#)

Account ID or alias [\(Don't have?\)](#)
608495930675

Remember this account

IAM username

Password
 [Having trouble?](#)

Show Password [Having trouble?](#)

Sign In

[Sign in using root user email](#)

[Create a new AWS account](#)

By continuing, you agree to [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

Account ID

The same for all Groups

e.g. 608495930675

IAM username

Your Group-User Name

e.g. teamX.userY

$X \in [1, 10]$, $Y \in [1, 4]$

Password

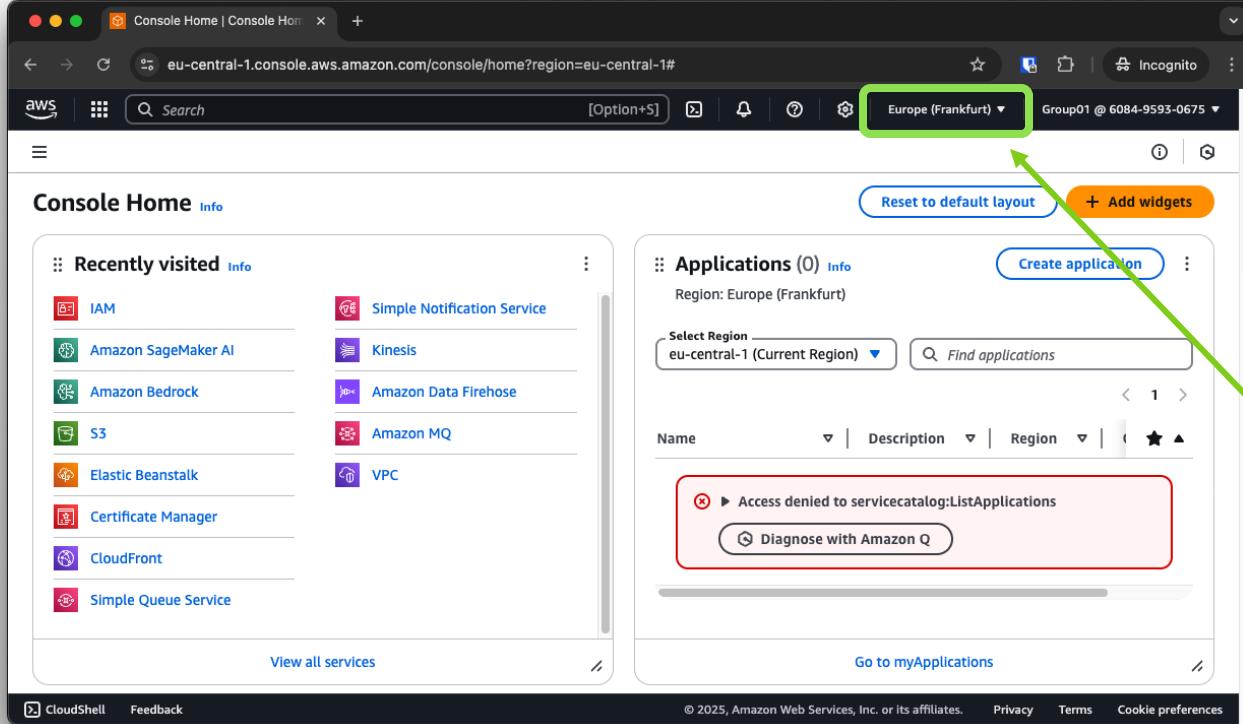
Your Password

e.g. 9lo/k\$ga1

Passwords will be handed to the groups individually



REGION



Make sure you
always select the
correct **region**.

Otherwise, the
services will **not**
work.

REGION:
eu-central-1

Europe (Frankfurt)



ROLES & GROUPS



IAM

Identity and Access Management

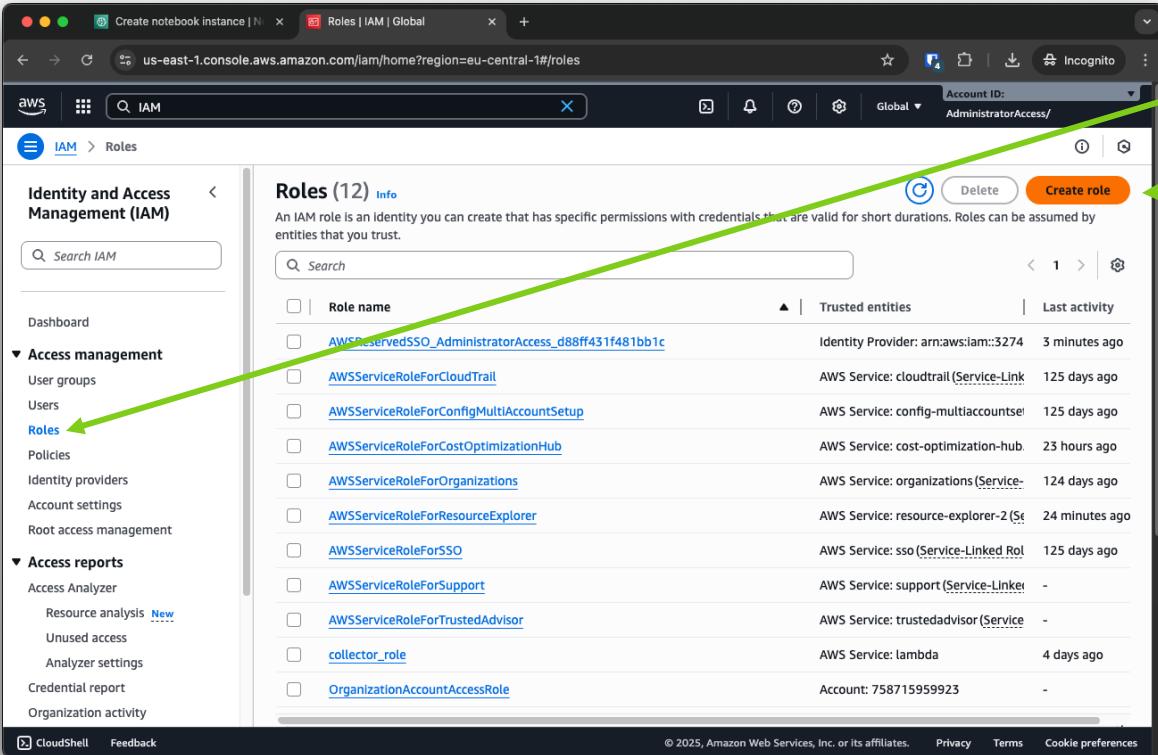
The screenshot shows the AWS IAM console homepage. On the left, there's a navigation sidebar with sections like Identity and Management, Access management, and Access reporting. The main area has two sections: 'Services' and 'Features'. Under 'Services', the 'IAM' service is highlighted with a red box and a green arrow pointing to it from the search bar above. The 'IAM' service entry includes a description: 'Manage access to AWS resources'. Below it are 'Top features' links for Groups, Users, Roles, Policies, and Access Analyzer. Under 'Features', there are entries for 'IAM Access analyzer for S3' (S3 feature) and 'Groups' (IAM feature). At the bottom, there's a 'Were these results helpful?' poll with 'Yes' and 'No' buttons. A modal window is open on the right, showing a list of roles with their last activity times, such as 'aws:iam::3274' (3 minutes ago), 'Service-Link' (125 days ago), and 'Multiaccounts' (125 days ago). The modal has a 'CREATE ROLE' button at the top.

1. Search for "IAM" in the search bar.
2. Select **IAM**



IAM

Create a role



The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Access management' expanded, showing 'Roles' highlighted with a green arrow. The main area displays a table of 12 existing roles, each with a checkbox, a role name, its trusted entity, and the last activity. At the top right of the table, there's a 'Create role' button, also highlighted with a green arrow.

Role name	Trusted entities	Last activity
AWSReservedSSO_AdministratorAccess_d88ff431f481bb1c	Identity Provider: arn:aws:iam::3274	3 minutes ago
AWSServiceRoleForCloudTrail	AWS Service: cloudtrail (Service-Link)	125 days ago
AWSServiceRoleForConfigMultiAccountSetup	AWS Service: config-multiaccounts	125 days ago
AWSServiceRoleForCostOptimizationHub	AWS Service: cost-optimization-hub	23 hours ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Link)	124 days ago
AWSServiceRoleForResourceExplorer	AWS Service: resource-explorer-2 (Service-Link)	24 minutes ago
AWSServiceRoleForSSO	AWS Service: sso (Service-Linked Role)	125 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Link)	-
collector_role	AWS Service: lambda	4 days ago
OrganizationAccountAccessRole	Account: 758715959923	-

3. Go to "Roles".

4. Create a new role.



IAM

Create a role

Screenshot of the AWS IAM "Create role" wizard Step 1: Select trusted entity. The "Trusted entity type" section shows "AWS service" selected (highlighted with a blue box and arrow). The "Use case" section shows "SageMaker" selected in the dropdown (highlighted with a blue box and arrow).

5. Create a role for an AWS Service

6. Select "SageMaker" as a Service.

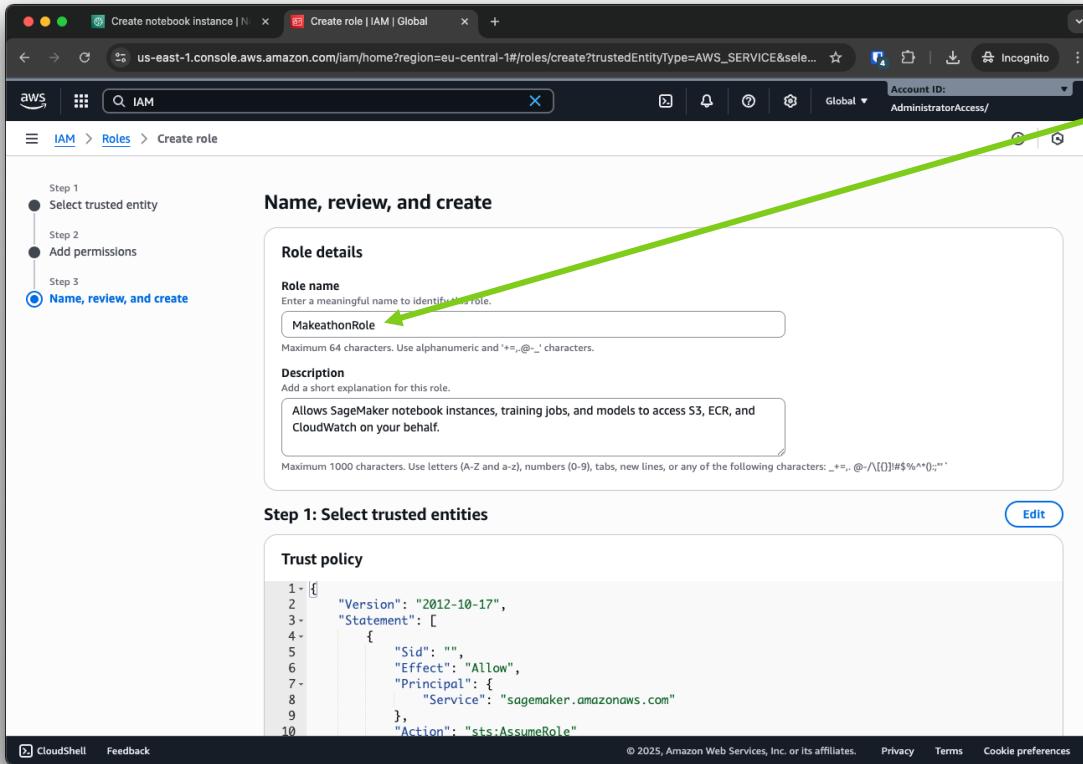
7. Click "Next".

8. Do not change anything on "Step 2" and just click "Next".



IAM

Create a role



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

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
MakeathonRole *

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

Description
Add a short explanation for this role.
Allows SageMaker notebook instances, training jobs, and models to access S3, ECR, and CloudWatch on your behalf.

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

Step 1: Select trusted entities

Trust policy

```
1 - {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "",
6       "Effect": "Allow",
7       "Principal": {
8         "Service": "sagemaker.amazonaws.com"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }
```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

9. Give the new role a meaningful name.

10. Click “Create role”.



IAM

Create a role

The screenshot shows the AWS IAM 'Create a role' page for a role named 'MakeathonRole'. The 'Permissions' tab is selected. A green arrow points from the text '11. Select the role you just created.' to the role name 'MakeathonRole' in the summary section. Another green arrow points from the text '12. Click on "Add Permissions" and select "Attach policies"' to the 'Add permissions' button in the 'Permissions policies' section, which has a dropdown menu open showing 'Attach policies' and 'Create inline policy'.

MakeathonRole Info

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

Summary

Creation date November 10, 2025, 13:37 (UTC+01:00)

Last activity -

ARN arn:aws:iam::327426388178:role/MakeathonRole

Maximum session duration 1 hour

Permissions Edit

Permissions policies (1) Info

You can attach up to 10 managed policies.

Filter by Type All types

Search Policy name Policy type Attached entities

Policy name Type Attached entities

AmazonSageMakerFullAccess AWS managed

Add permissions ▾

Attach policies Create inline policy

Permissions boundary (not set)

Generate policy based on CloudTrail events

<https://us-east-1.console.aws.amazon.com/iam/home?region=eu-central-1#roles/details/MakeathonRole/attach-policies>

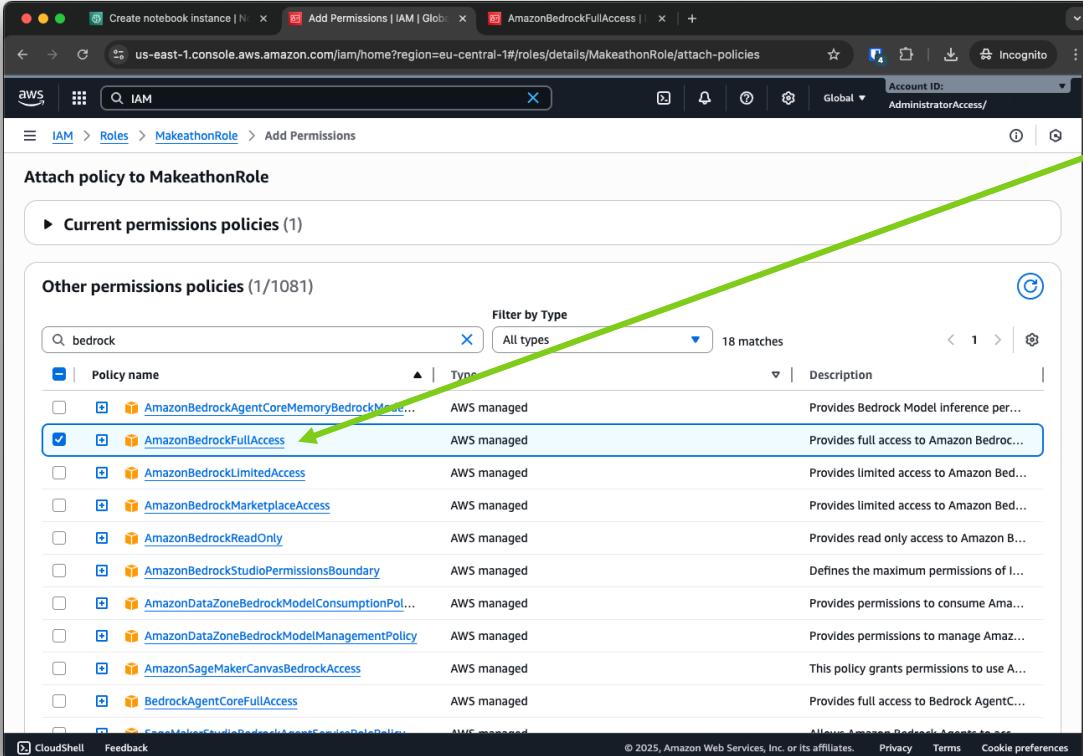
11. Select the role you just created.

12. Click on "Add Permissions" and select "Attach policies".



IAM

Create a role



The screenshot shows the AWS IAM 'Add Permissions' interface. The URL is <https://us-east-1.console.aws.amazon.com/iam/home?region=eu-central-1#/roles/details/MakeathonRole/attach-policies>. The 'Account ID' is listed as 'AdministratorAccess/'. The search bar at the top has 'bedrock' typed into it. A green arrow points from the text '13. Add the permission "Amazon Bedrock Full Access"' to the 'AmazonBedrockFullAccess' policy entry in the list.

Attach policy to MakeathonRole

▶ Current permissions policies (1)

Other permissions policies (1/1081)

Filter by Type: All types | 18 matches

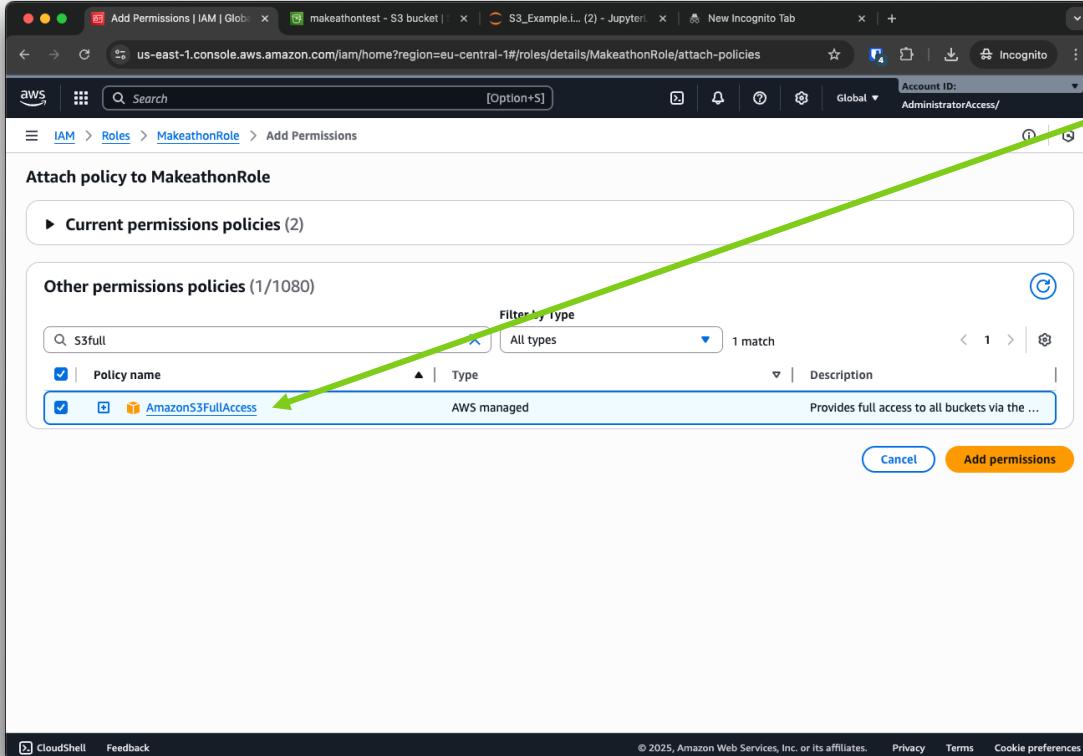
Policy name	Type	Description
AmazonBedrockAgentCoreMemoryBedrockModel...	AWS managed	Provides Bedrock Model inference per...
AmazonBedrockFullAccess	AWS managed	Provides full access to Amazon Bedroc...
AmazonBedrockLimitedAccess	AWS managed	Provides limited access to Amazon Bed...
AmazonBedrockMarketplaceAccess	AWS managed	Provides limited access to Amazon Bed...
AmazonBedrockReadOnly	AWS managed	Provides read only access to Amazon B...
AmazonBedrockStudioPermissionsBoundary	AWS managed	Defines the maximum permissions of ...
AmazonDataZoneBedrockModelConsumptionPol...	AWS managed	Provides permissions to consume Ama...
AmazonDataZoneBedrockModelManagementPolicy	AWS managed	Provides permissions to manage Amaz...
AmazonSageMakerCanvasBedrockAccess	AWS managed	This policy grants permissions to use A...
BedrockAgentCoreFullAccess	AWS managed	Provides full access to Bedrock AgentC...
CloudWatchCloudTrailDelivery...	AWS managed	Allows CloudTrail to deliver logs to ...

13. Add the permission
“Amazon Bedrock Full Access”



IAM

Create a role



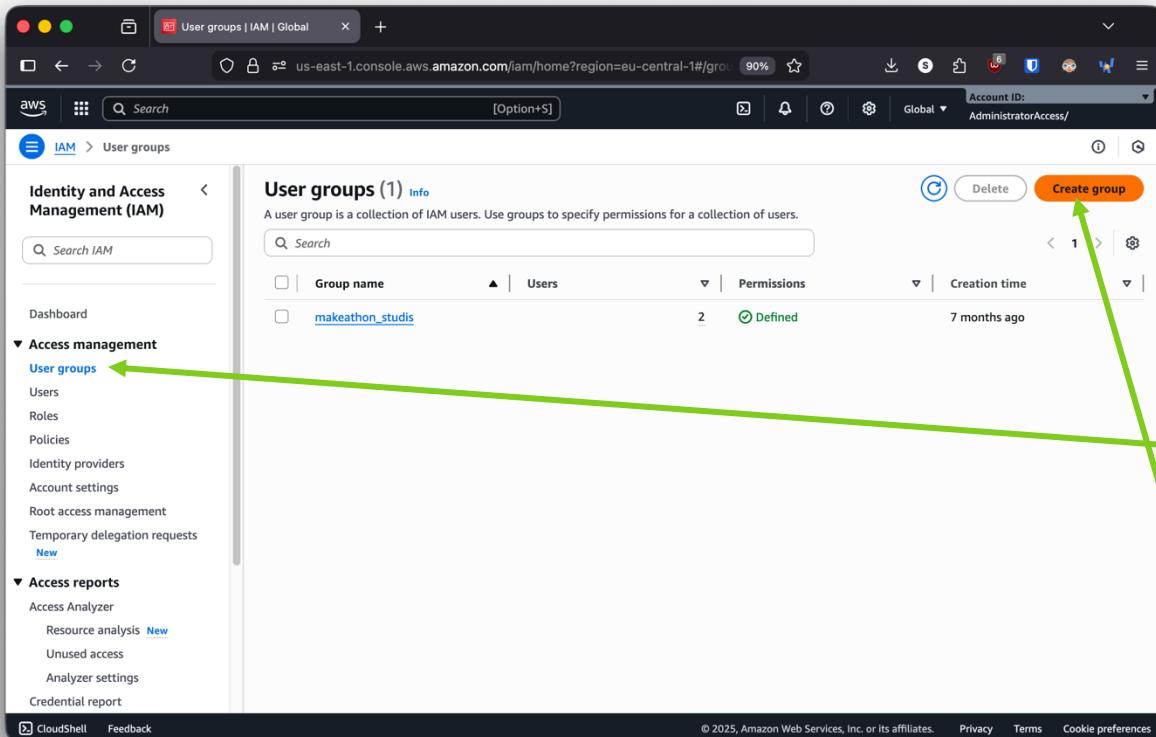
The screenshot shows the AWS IAM 'Add Permissions' interface for a role named 'MakeathonRole'. The 'Current permissions policies' section contains two items: 'AdministratorAccess' and 'AWS Lambda - Lambda execution role'. The 'Other permissions policies' section lists 1,080 policies, with one matching result for 'S3full'. A green arrow points from the text '14. Add the permission "Amazon S3 Full Access"' to the 'AmazonS3FullAccess' policy entry, which is highlighted with a blue border. The policy description indicates it provides full access to all buckets via the AWS Management Console.

14. Add the permission
“Amazon S3 Full Access”



IAM

Create a group



The screenshot shows the AWS IAM User groups page. On the left, there's a navigation sidebar with sections like Identity and Access Management (IAM), Access management, and Access reports. Under Access management, 'User groups' is highlighted with a green arrow. The main content area shows a table titled 'User groups (1)'. The table has columns for Group name, Users, Permissions, and Creation time. One row is listed: 'makeathon_studs' with 2 users, 'Defined' permissions, and created '7 months ago'. At the top right of the table, there are 'Delete' and 'Create group' buttons, with the 'Create group' button highlighted by a green arrow.

Group name	Users	Permissions	Creation time
makeathon_studs	2	Defined	7 months ago

If you want to share roles between users, you can create a **group** and assign the role to a group.

1. Go to “User Groups”
2. Click “Create Group”



IAM

Create a group

The screenshot shows the 'Create user group' wizard in the AWS IAM console. The left sidebar shows 'Access management' selected under 'User groups'. The main area has three tabs: 'Name the group', 'Add users to the group - Optional (1/2)', and 'Attach permissions policies - Optional (1/1111)'. A green arrow points from the first step to the second, another from the second to the third, and a final one from the third back to the first.

Name the group

User group name: PowerUsers

Add users to the group - Optional (1/2)

User name	Group	Last activity	Creation time
Group00	1	212 days ago	7 months ago
<input checked="" type="checkbox"/> haris	1	219 days ago	7 months ago

Attach permissions policies - Optional (1/1111)

Filter by Type: All types

Policy name	Type	Used as	Description
<input checked="" type="checkbox"/> AmazonSS3FullAccess	AWS managed	Permissions policy (1)	Provides full access to all buckets via the AWS Management Console.

Cancel Create user group

1. Give the group a proper name.

2. Add members to the group (can be done later).

3. Assign Permissions to the Group.

e.g.

"Amazon Bedrock Full Access"

"Amazon S3 Full Access"

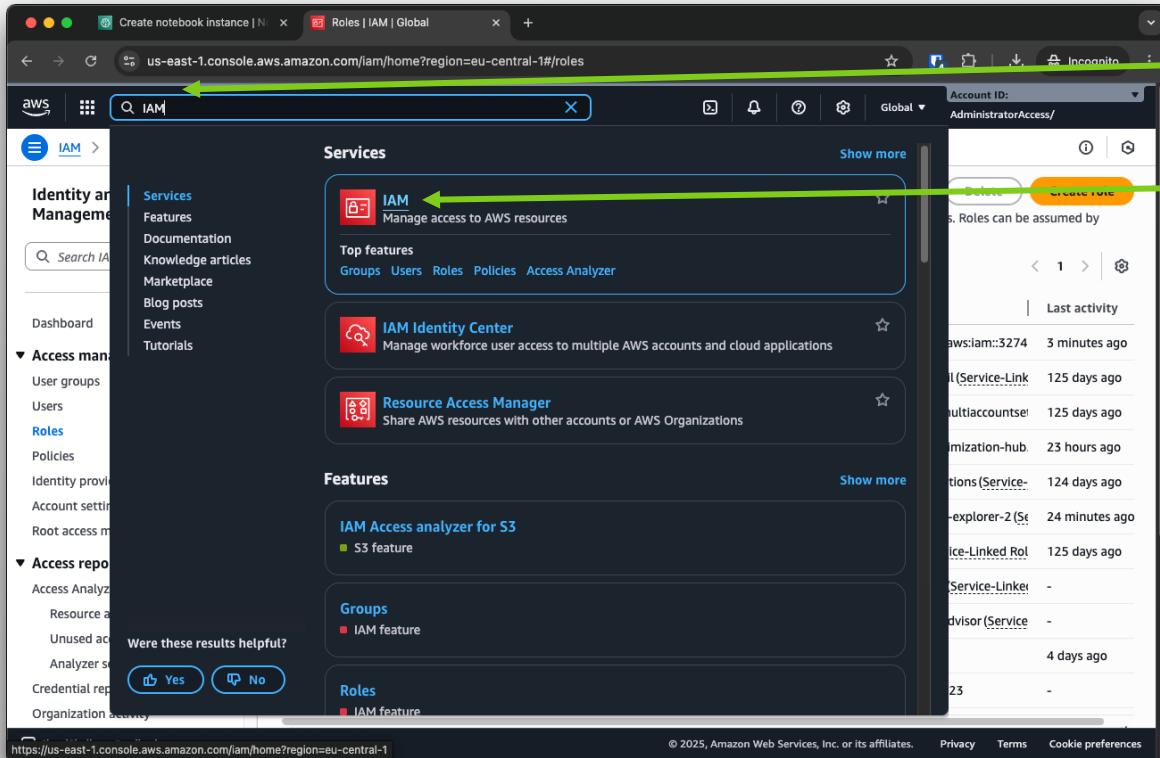


ACCESS KEYS



ACCESS KEYS

Use AWS locally



The screenshot shows the AWS IAM console interface. At the top, there is a search bar with the text "IAM". A green arrow points from the text "1. Search for 'IAM'" to this search bar. Below the search bar, the main content area has a heading "Services". Under "Services", there is a section titled "Top features" which includes links for Groups, Users, Roles, Policies, and Access Analyzer. A green arrow points from the text "2. Select IAM" to the "Roles" link. To the right of the services list, there is a sidebar with a list of recent activity items, such as "aws:iam::3274" and "Service-Link".

1. Search for "IAM" in the search bar.

2. Select IAM



ACCESS KEYS

Create a user

The screenshot shows the AWS IAM 'Users' page. On the left, a sidebar menu is open under 'Access management', with 'Users' selected. The main area displays a table titled 'Users (3) Info' with three entries. Each entry includes a checkbox, a user name (all starting with '/'), a path ('..'), a group ('1'), last activity dates ('4 days ago', '2 days ago', '5 days ago'), MFA status ('-'), and password age ('-'). At the top right of the table, there are 'Create user' and 'Delete' buttons. The URL in the browser is 'us-east-1.console.aws.amazon.com/iam/home?region=eu-central-1#/users'.

User name	Path	Group	Last activity	MFA	Password age
/	..	1	4 days ago	-	-
/	..	1	2 days ago	-	-
/	..	1	5 days ago	-	-

3. Go to **Users**

4. Create a new user or select an existing one.



ACCESS KEYS

Create a user

Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create

Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (3/1404)
Choose one or more policies to attach to your new user.

Filter by Type
S3 All types 17 matches

Policy name	Type	Attached entities
AmazonDMSRedshiftS3Role	AWS managed	0
AmazonS3FullAccess	AWS managed	2
AmazonS3ObjectLambdaE...	AWS managed	0
AmazonS3OutpostsFullAcc...	AWS managed	0
AmazonS3OutpostsReadO...	AWS managed	0

5. Attach policies directly

You need to attach these three policies

AmazonS3FullAccess
AmazonBedrockFullAccess
AmazonSageMakerFullAccess

6. Create User



ACCESS KEYS

Create access keys

The screenshot shows the AWS IAM 'hannes' user details page. The 'Security credentials' tab is selected. A green arrow points from the text '7. Click on "Security Credentials"' to the 'Create access key' button. Another green arrow points from the text '8. Scroll down to Access Keys.' to the 'Access keys (0)' section. A third green arrow points from the text '9. Click on Create access key.' to the 'Create access key' button in the 'Access keys (0)' section.

hannes | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=eu-central-1#users/details/hannes?section=security_credentials

Account ID: AdministratorAccess/

IAM > Users > hannes

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management

Access reports

- Access Analyzer
 - Resource analysis [New](#)
 - Unused access
 - Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies

CloudShell Feedback

Summary

ARN arn:aws:iam::327426388178:user/hannes

Console access Disabled

Created November 10, 2025, 15:11 (UTC+01:00)

Last console sign-in -

Access key 1 [Create access key](#)

Permissions Groups Tags Security credentials Last Accessed

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

[Create access key](#)

API keys for Amazon Bedrock (0)

Use API keys for Amazon Bedrock to integrate into your library of choice and make API requests programmatically. You can have a maximum of two long-term API keys (active, inactive, or expired) at a time. [Learn more](#)

API key name	Created	Expires	Status

Actions [Generate API Key](#)

7. Click on
“Security
Credentials”

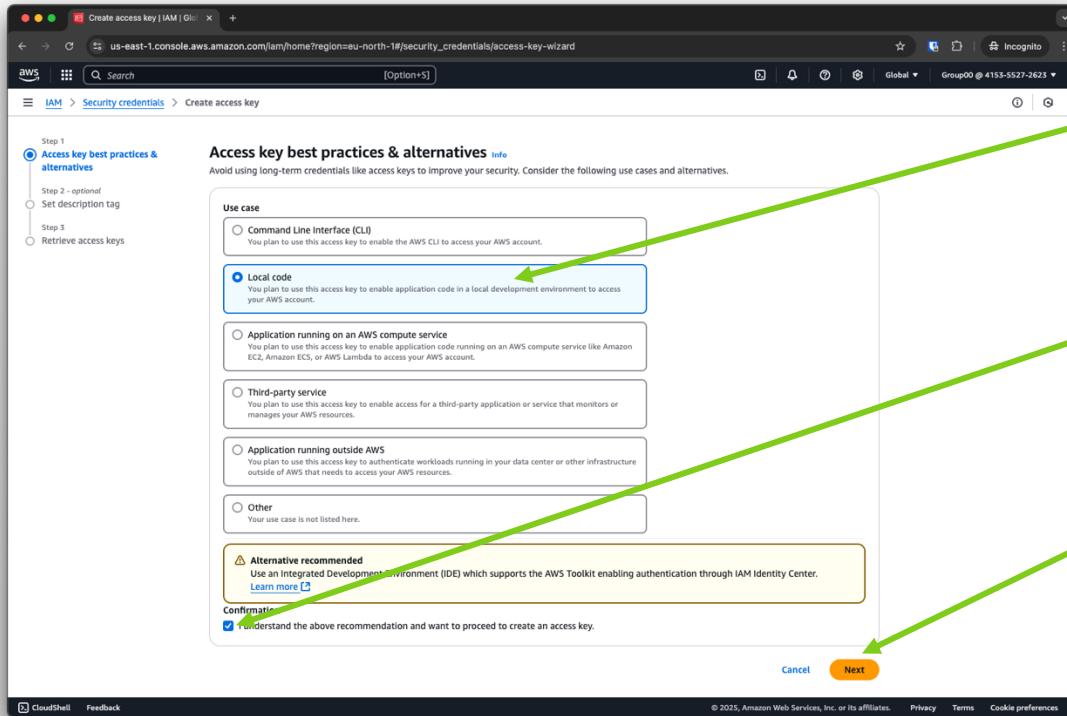
8. Scroll down to
Access Keys.

9. Click on **Create
access key**.



ACCESS KEYS

Create access keys



10. Select **Local code** as use case

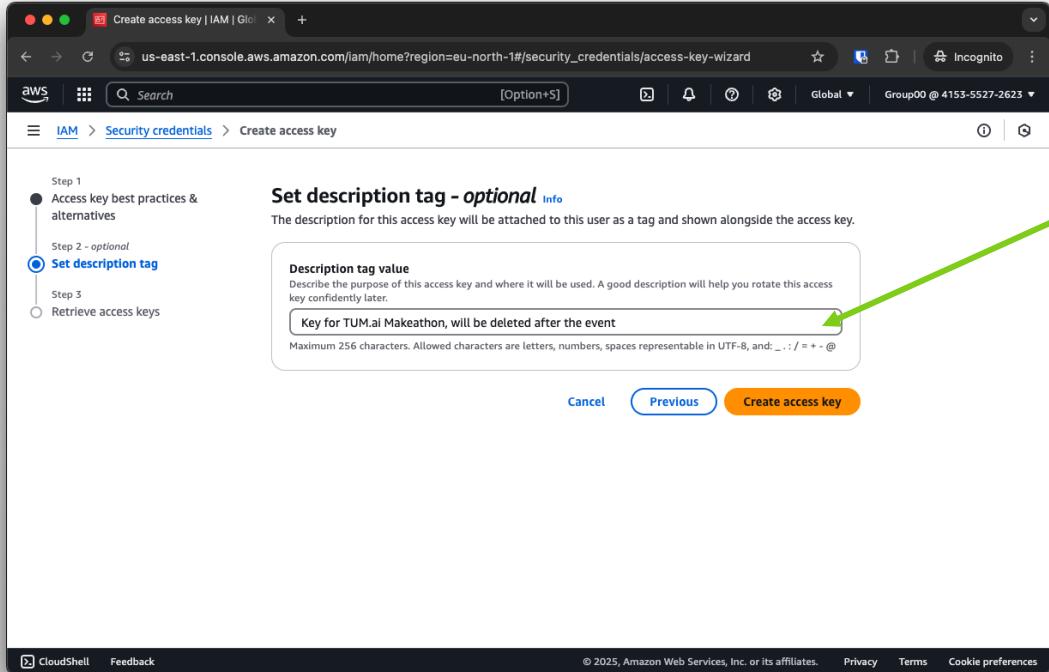
11. Confirm the recommended alternatives

12. Click on **Next**



ACCESS KEYS

Create access keys



The screenshot shows the 'Create access key' wizard on the AWS IAM console. The user is on Step 2, titled 'Set description tag - optional'. A green arrow points from the text '13. You can enter a meaningful description here.' to the 'Description tag value' input field. The input field contains the text 'Key for TUM.ai Makeathon, will be deleted after the event'. The interface includes a sidebar with steps: Step 1 (Access key best practices & alternatives), Step 2 - optional (Set description tag, which is selected), and Step 3 (Retrieve access keys). At the bottom are 'Cancel', 'Previous', and 'Create access key' buttons.

13. You can enter a meaningful description here.



ACCESS KEYS

Store access keys

The screenshot shows the 'Create access key' step in the AWS IAM console. A green banner at the top states: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this, a section titled 'Retrieve access keys' shows the 'Access key' fields. The 'Access key' field contains the value 'AKIAUYPBCGDJETL7QFWR'. To its right is a 'Secret access key' field, which is partially obscured by a series of asterisks ('*****') and has a 'Show' link next to it. On the left, a sidebar lists three steps: 'Access key best practices & alternatives' (Step 1), 'Set description tag' (Step 2 - optional), and 'Retrieve access keys' (Step 3, currently selected). At the bottom are 'Download .csv file' and 'Done' buttons.

14. If you see this everything worked well.

15. Store your **Access Key ID** and the **Access Key Secret** locally (→ .env file)

ATTENTION!
If you leave this page, you cannot show the secret again.
If you lose your access key, you will have to create a new one!

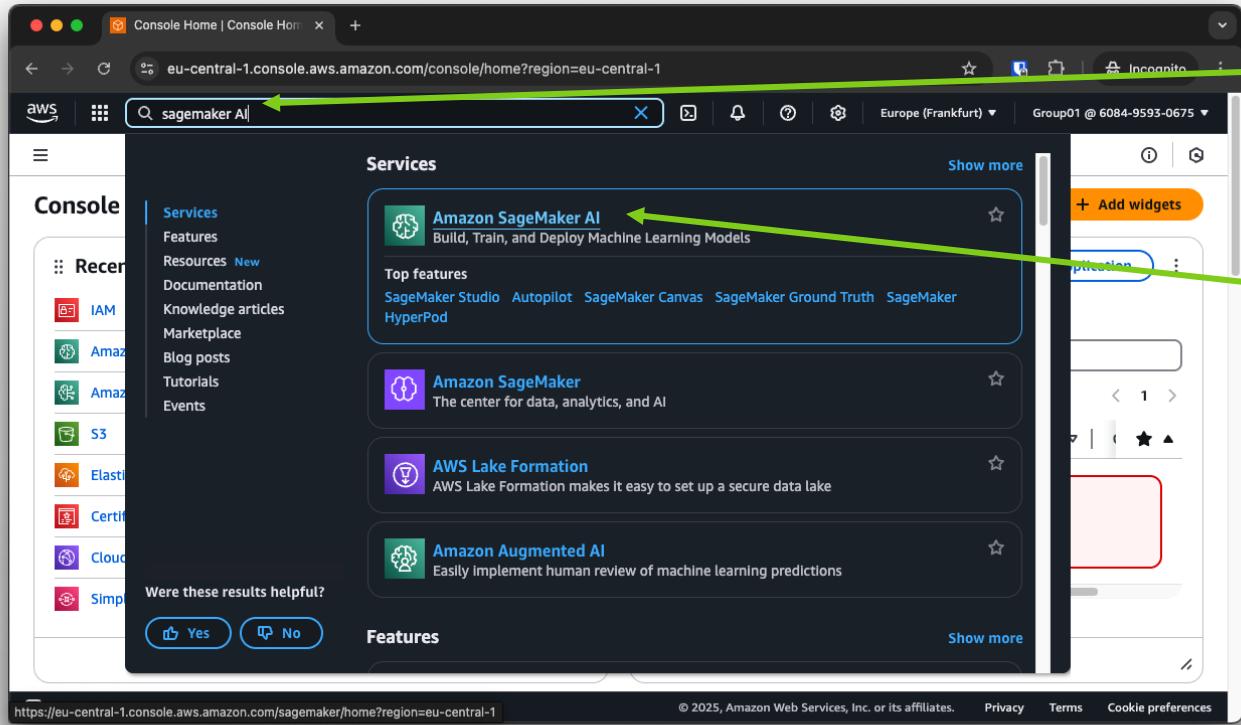


SAGEMAKER



SAGEMAKER

Easy Collaboration



1. Search for "Sagemaker" in the search bar.
2. Select **Amazon SageMaker AI**



SAGEMAKER

Easy Collaboration

The screenshot shows the Amazon SageMaker AI interface. On the left, there's a sidebar with sections like 'Amazon SageMaker AI', 'Getting started', 'Dashboard', 'What's new', 'Applications and IDEs' (with 'Notebooks' highlighted), and 'Admin configurations'. The main content area is titled 'Notebooks and Git repos' and features a 'Notebook instances' tab. It includes a search bar, a table header with columns 'Name', 'Instance', 'Creation time', 'Status', and 'Actions', and a message 'There are currently no resources.' A prominent orange button labeled 'Create notebook instance' is located at the top right of the table area. A green arrow points from the 'Notebooks' link in the sidebar to this button.

3. Click on
Notebooks.

4. You can create
you own virtual
machine running
JupyterLab



SAGEMAKER

Easy Collaboration

The screenshot shows the 'Create notebook instance' wizard in the Amazon SageMaker console. The 'Notebook instance settings' section is visible, containing the following fields:

- Notebook instance name:** mynotebook01 (highlighted by a green arrow)
- Notebook instance type:** ml.t3.medium (highlighted by a green arrow)
- Platform identifier:** Amazon Linux 2, Jupyter Lab 4
- Additional configuration:** A dropdown menu for creating a new role or selecting an existing one.
- Create a new role:** MakeathonRole (highlighted by a blue selection bar and a green arrow)
- Enter a custom IAM role ARN:** MakeathonRole
- Use existing role:** MakeathonRole (highlighted by a green arrow)
- Create role using the role creation wizard:** A button with a blue outline.
- Root access - optional:**
 - Enable - Give users root access to the notebook
 - Disable - Don't give users root access to the notebook

At the bottom of the wizard, there are links for CloudShell and Feedback, and standard AWS navigation links for Privacy, Terms, and Cookie preferences.

5. Choose a valid instance name.

6. Select a suitable kind of virtual machine. Try to stay within the Free Tier (see next slide).

7. Choose the role you created before



SAGEMAKER

Easy Collaboration

<https://aws.amazon.com/sagemaker/ai/pricing/>

The screenshot shows the AWS SageMaker Pricing page. At the top, there are tabs for Studio Classic, JupyterLab, Code Editor, RStudio, Notebook Instances, Processing, TensorBoard, Data Wrangler, Feature Store, and Training. Below these are sub-tabs for MLflow, Real-Time Inference, Asynchronous Inference, Batch Transform, Serverless Inference, JumpStart, Profiler, and HyperPod. A section titled "On-demand pricing" displays the "Amazon SageMaker JupyterLab" instance type. It includes a table with columns for Standard Instances, vCPU, Memory, and Price per Hour. The table lists several instance types with their respective specifications and hourly rates. A yellow arrow points to the "Price per Hour" column header.

Standard Instances	vCPU	Memory	Price per Hour
m1.t3.medium	2	4 GiB	\$0.058
m1.t3.large	2	8 GiB	\$0.115
m1.t3.xlarge	4	16 GiB	\$0.23
m1.t3.2xlarge	8	32 GiB	\$0.461
m1.m7t.large	2	8 GiB	\$0.145
m1.m7t.xlarge	4	16 GiB	\$0.29

There are VM types of every need.
But remember
great power comes with a high price! Do not run out of budget!



If you need **GPU** support, check out the specs of the available machines and their availability in eu-central-1.

<https://docs.aws.amazon.com/sagemaker/latest/dg/notebooks-available-instance-types.html>



SAGEMAKER

Easy Collaboration

The screenshot shows the Amazon SageMaker AI Notebook Instances console. On the left, there's a sidebar with sections like 'Getting started', 'Dashboard' (which is highlighted in green), 'What's new', 'Applications and IDEs' (Studio, Canvas, RStudio, TensorBoard, Profiler, Notebooks, Partner AI Apps), and 'Admin configurations' (Domains, Role manager, Images, Lifecycle configurations). The main content area is titled 'mynotebook01' and displays 'Notebook instance settings'. It includes fields for Name (mynotebook01), Status (inService), Notebook instance type (ml.t3.medium), Platform identifier (Amazon Linux 2, Jupyter Lab 4 (notebook-al2-v3)), ARN (arn:aws:sagemaker:eu-central-1:327426388178:notebook-instance/mynotebook01), Creation time (Nov 10, 2025 12:48 UTC), Volume Size (5GB EBS), Last updated (Nov 10, 2025 12:50 UTC), and Lifecycle configuration (-). Below this are sections for 'Git repositories' (empty) and 'Permissions and encryption' (IAM role ARN: arn:aws:iam::327426388178:role/Makeathon, Root access: Enabled, Encryption key: None). At the top of the main content area, there are three buttons: 'Delete', 'Stop' (highlighted with a green arrow), 'Open Jupyter', and 'Open JupyterLab'.

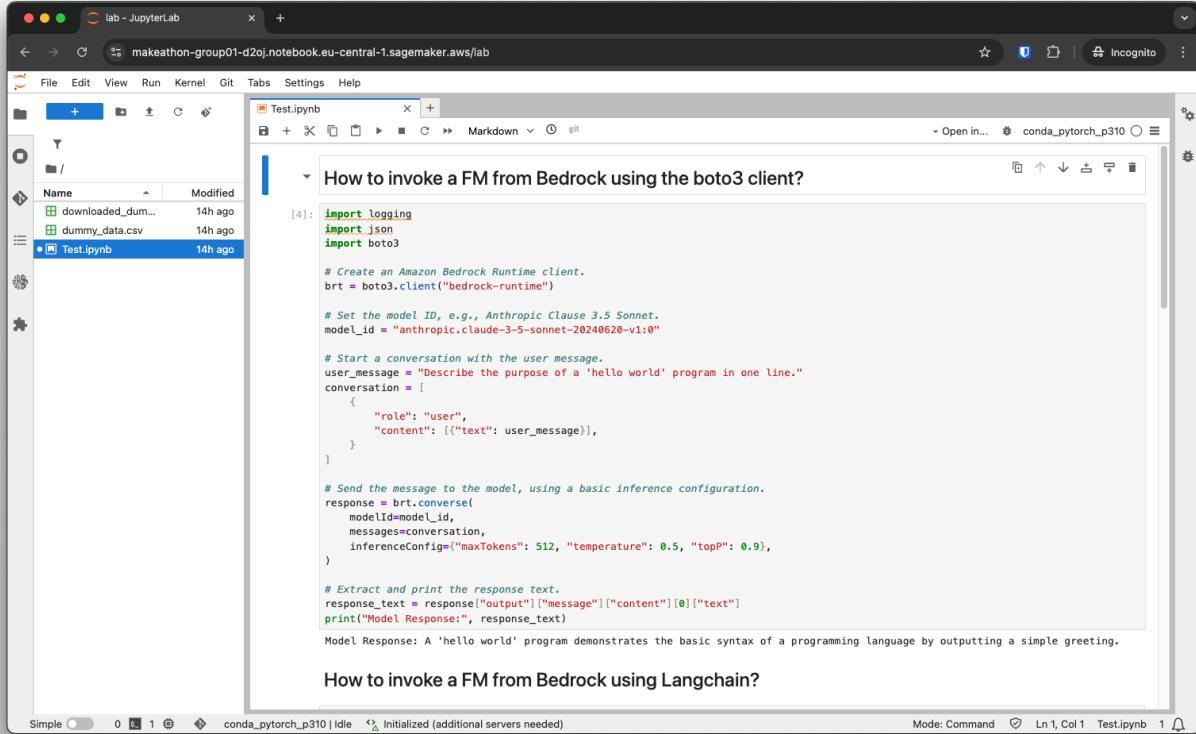
8. You can **Start / Stop** your Notebook Instance here.

9. You can access the **JupyterLab** instance here.



SAGEMAKER

Easy Collaboration



The screenshot shows a JupyterLab interface with a browser window titled "lab - JupyterLab". The browser displays a notebook file named "Test.ipynb". The notebook contains a code cell with the following Python script:

```
import logging
import json
import boto3

# Create an Amazon Bedrock Runtime client.
brt = boto3.client("bedrock-runtime")

# Set the model ID, e.g., Anthropic Claude 3.5 Sonnet.
model_id = "anthropic.claude-3-5-sonnet-20240620-v1:0"

# Start a conversation with the user message.
user_message = "Describe the purpose of a 'hello world' program in one line."
conversation = [
    {
        "role": "user",
        "content": [{"text": user_message}],
    }
]

# Send the message to the model, using a basic inference configuration.
response = brt.converse(
    modelId=model_id,
    messages=conversation,
    inferenceConfig={"maxTokens": 512, "temperature": 0.5, "topP": 0.9},
)

# Extract and print the response text.
response_text = response["output"]["message"][0]["text"]
print("Model Response:", response_text)
```

The output of the code cell is:

```
Model Response: A 'hello world' program demonstrates the basic syntax of a programming language by outputting a simple greeting.
```

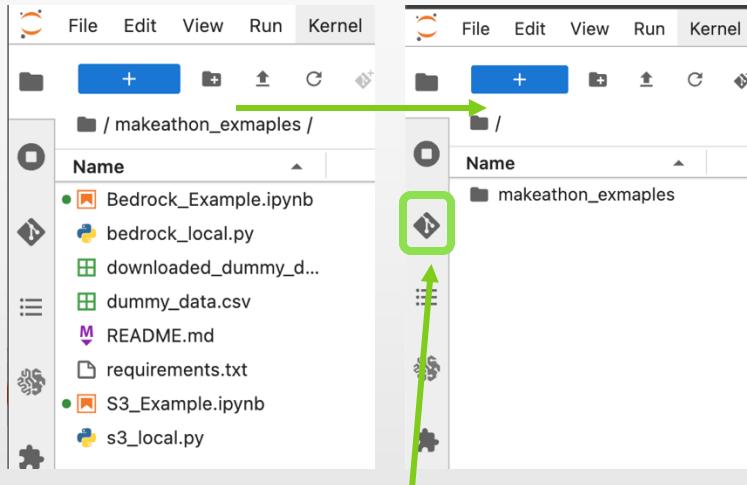
Below the code cell, there is a section title:

How to invoke a FM from Bedrock using Langchain?

10. Start Coding!

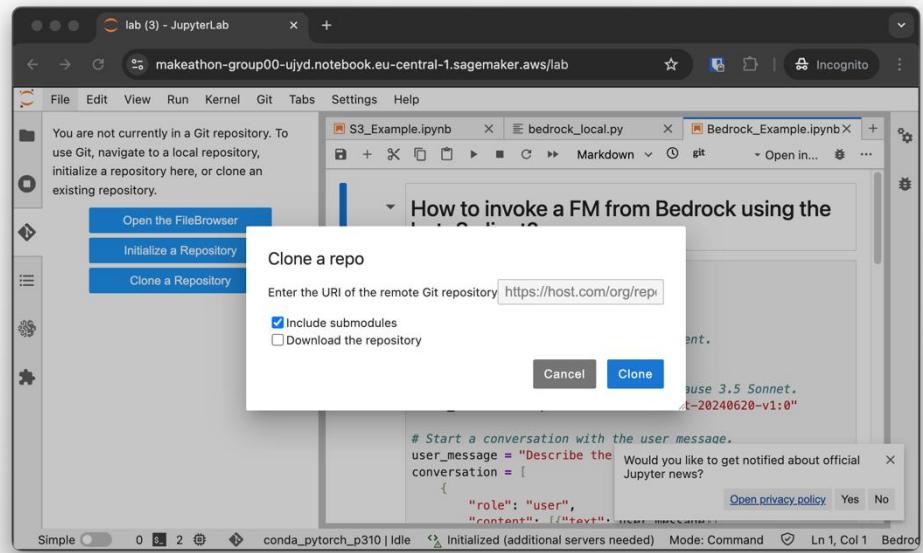


VERSION CONTROL



Make sure you
are in the root
folder

Click on the Git
Symbol



EXAMPLE FILES



AWS S3

S3_Example.ipynb x Bedrock_Example.ipynb x +

How to upload files and download files from S3?

```
[1]: import pandas as pd
import boto3

[2]: bucket_name = 'makeathonstest'
prefix = "data/"

[3]: # 1. Create dummy data to save to s3 and read from it
data = [
    'name': ['Alice', 'Bob', 'Charlie'],
    'age': [25, 30, 35],
    'email': ['alice@example.com', 'bob@example.com', 'charlie@example.com']
]
df = pd.DataFrame(data)

[4]: # 2. Store file locally
local_file_path = "dummy_data.csv"
df.to_csv(local_file_path, index=False)
print(f"CSV saved locally at: {local_file_path}")

CSV saved locally at: dummy_data.csv

[5]: # 3. Upload to S3
s3_key = prefix + "dummy_data.csv"
s3 = boto3.client('s3')
s3.upload_file(Filename=local_file_path, Bucket=bucket_name, Key=s3_key)
print(f"CSV uploaded to s3://(bucket_name)/(s3_key)")

CSV uploaded to s3://makeathonstest/data/dummy_data.csv

[ ]: # Alternatively store data directly to S3
full_s3_path = f"s3://{bucket_name}/{prefix}another_file.csv"
df.to_csv(full_s3_path, index=False)

[ ]: # List the content of your folder (prefix)
s3 = boto3.client('s3')
```



AWS Bedrock

S3_Example.ipynb x Bedrock_Example.ipynb x +

How to invoke a FM from Bedrock using the boto3 client?

```
[ ]: import logging
import json
import boto3

# Create an Amazon Bedrock Runtime client.
brt = boto3.client("bedrock-runtime")

# Set the model ID, e.g., Anthropic Claude 3.5 Sonnet.
model_id = "anthropic.claude-3-5-sonnet-20240620-v1:0"

# Start a conversation with the user message.
user_message = "Describe the purpose of a 'hello world' program in one line."
conversation = [
    {
        "role": "user",
        "content": [{"text": user_message}],
    }
]

# Send the message to the model, using a basic inference configuration.
response = brt.invoke(
    modelId=model_id,
    messages=conversation,
    inferenceConfig={"maxTokens": 512, "temperature": 0.5, "topP": 0.9},
)

# Extract and print the response text.
response_text = response["output"]["message"]["content"][0]["text"]
print("Model Response:", response_text)
```

How to invoke a FM from Bedrock using Langchain?



S3 STORAGE



S3 BUCKET

A lot of storage

The screenshot shows the AWS IAM console interface. In the top search bar, the text "s3" is typed, highlighted by a green arrow. Below the search bar, the "Services" section is expanded, showing several options. The "S3" service card is highlighted with a green arrow. The card includes the text "Scalable Storage in the Cloud" and "Top features: Buckets, Storage Lens dashboards, Batch Operations, S3 Express One Zone, S3 Access Grants". Other services listed include "S3 Glacier", "AWS Snow Family", and "Storage Gateway". At the bottom of the service list, there are buttons for "Were these results helpful?" ("Yes" or "No") and "Features" (with a "Show more" link). A red box highlights the "Access denied" message at the bottom right of the page.

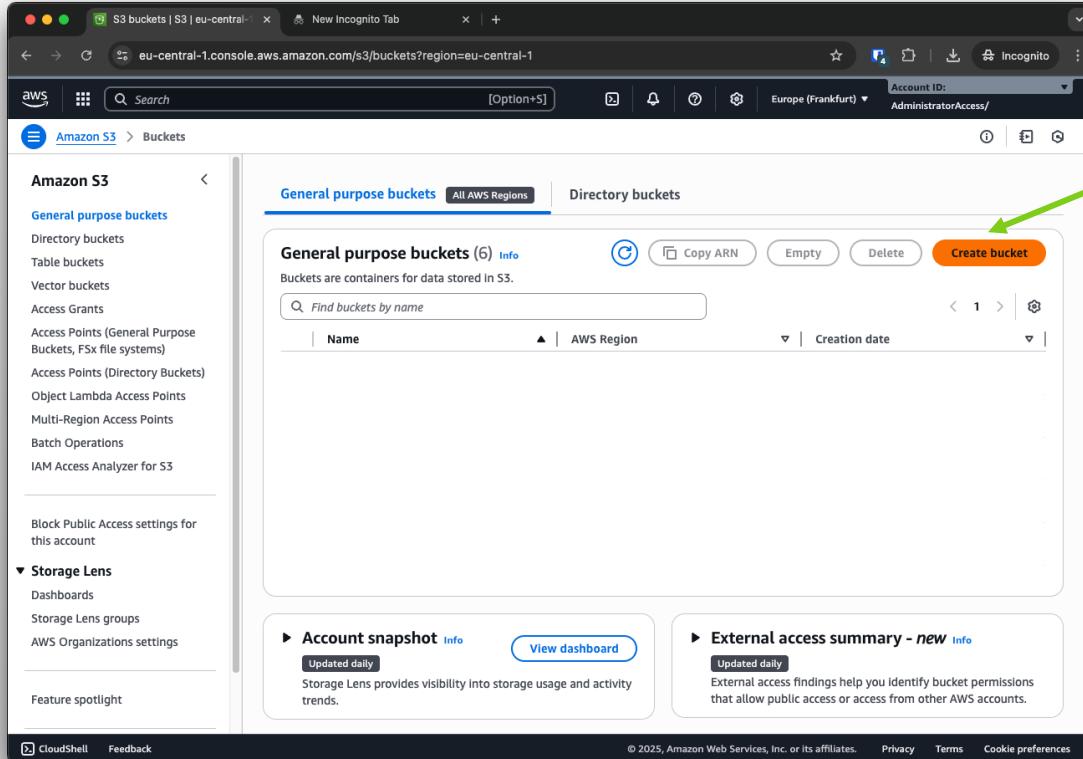
1. Search for “S3” in the search bar.

2. Click on **S3**.



S3 BUCKET

Create a bucket



The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various navigation options like 'General purpose buckets', 'Directory buckets', and 'Storage Lens'. The main area is titled 'General purpose buckets (6)' and contains a table with columns for Name, AWS Region, and Creation date. At the top right of this section is a prominent orange 'Create bucket' button. A large green arrow points from the text '3. Create a new bucket for your project.' to this button.

3. Create a new bucket for your project.



S3 BUCKET

Create a bucket

The screenshot shows the 'Create bucket' wizard on the AWS S3 console. The 'General configuration' section is visible, featuring a 'Bucket name' input field containing 'makeathonstest'. A green arrow points from the text 'makeathonstest' to the input field. Below the input field, a note states: 'Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn more](#)'.

Below the general configuration, there's an 'Object Ownership' section. It contains two radio button options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The 'ACLs disabled' option is described as: 'All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.' The 'ACLs enabled' option is described as: 'Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.'

At the bottom of the page, there are links for 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

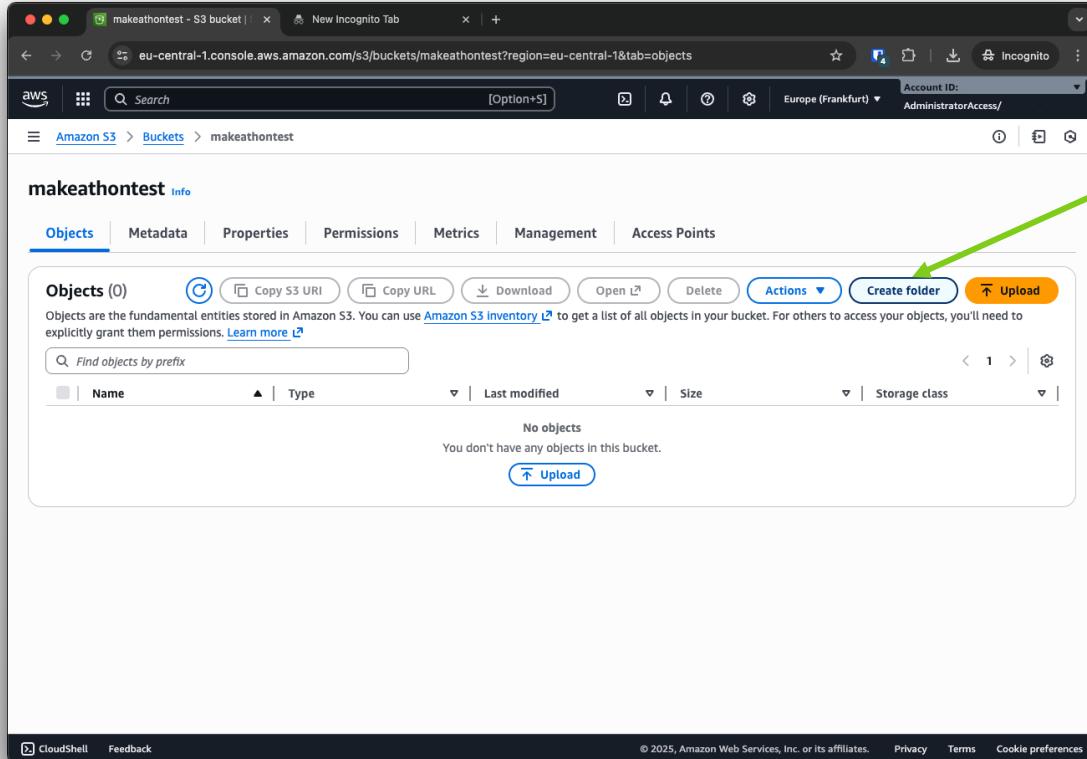
4. Select a **unique name** for your bucket.

(When you scroll down you can enable **versioning** if you want.)



S3 BUCKET

Create a folder



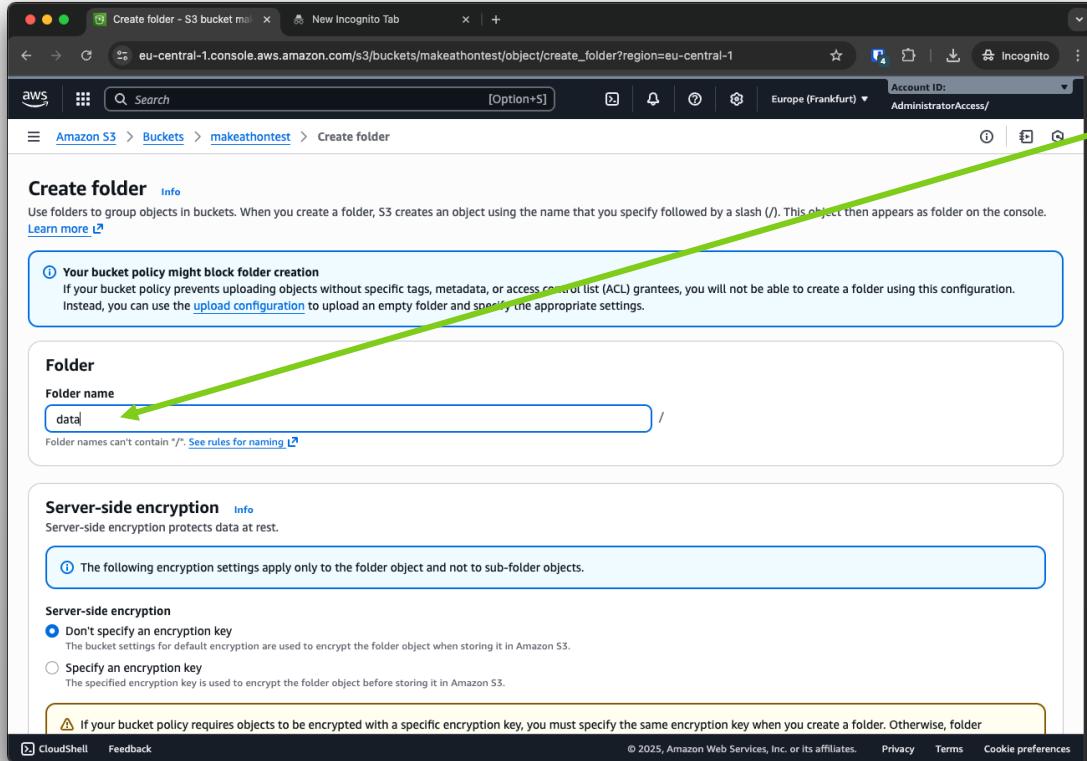
The screenshot shows the AWS S3 console interface for a bucket named 'makeathon'. The top navigation bar includes tabs for Objects, Metadata, Properties, Permissions, Metrics, Management, and Access Points. The 'Actions' tab is currently selected. A prominent orange button labeled 'Create folder' is located in the top right corner of the main content area. Below it, there is a large button labeled 'Upload' with a downward arrow icon. The main content area displays a message stating 'No objects' and 'You don't have any objects in this bucket.' At the bottom of the page, there are links for CloudShell, Feedback, and various AWS service terms like Privacy, Terms, and Cookie preferences.

5. Create a
“folder” in your
bucket. This is also
called **prefix**.



S3 BUCKET

Create a folder



The screenshot shows the 'Create folder' page in the AWS S3 console. At the top, there's a warning message about bucket policies blocking folder creation. Below it, a 'Folder' section has a 'Folder name' input field containing 'data'. A green arrow points from this field to a note about bucket policies. Further down, there's a 'Server-side encryption' section with two options: 'Don't specify an encryption key' (selected) and 'Specify an encryption key'. A note at the bottom states that if a bucket policy requires encryption, a specific key must be used.

Your bucket policy might block folder creation
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the [upload configuration](#) to upload an empty folder and specify the appropriate settings.

Folder

Folder name

data /

Folder names can't contain *?/. See rules for naming.

Server-side encryption

The following encryption settings apply only to the folder object and not to sub-folder objects.

Don't specify an encryption key
The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

Specify an encryption key
The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

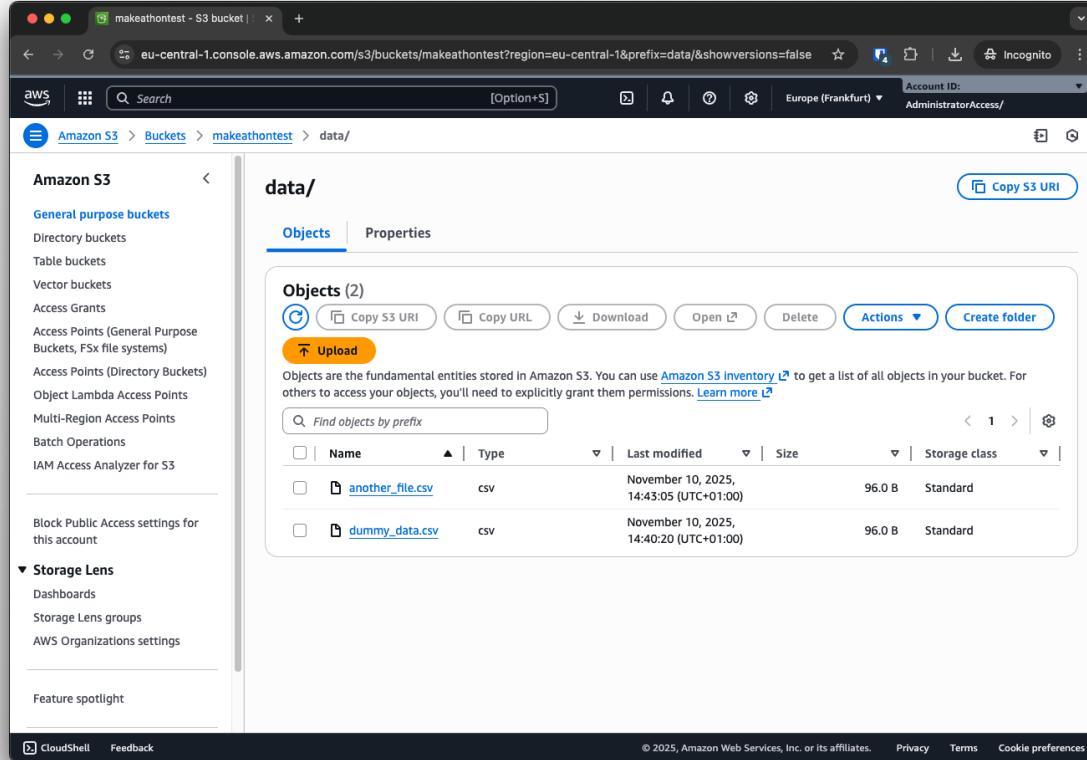
If your bucket policy requires objects to be encrypted with a specific encryption key, you must specify the same encryption key when you create a folder. Otherwise, folder

6. Pick a valid name for your prefix.



S3 BUCKET

Access Files



The screenshot shows the Amazon S3 console interface. The left sidebar has sections for General purpose buckets, Directory buckets, Table buckets, Vector buckets, Access Grants, Access Points (General Purpose Buckets, FSx file systems), Access Points (Directory Buckets), Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens (Dashboards, Storage Lens groups, AWS Organizations settings), and Feature spotlight. The main area shows a 'data' folder with two objects: 'another_file.csv' and 'dummy_data.csv'. Both are CSV files, 96.0 B in size, last modified on November 10, 2025, at 14:43:05 (UTC+01:00) and 14:40:20 (UTC+01:00) respectively, and are stored in Standard storage class. There are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, and Create folder.

Name	Type	Last modified	Size	Storage class
another_file.csv	csv	November 10, 2025, 14:43:05 (UTC+01:00)	96.0 B	Standard
dummy_data.csv	csv	November 10, 2025, 14:40:20 (UTC+01:00)	96.0 B	Standard

You can access your files here!

You can create folders, upload or download files.

Or use the SDK to access your files via code.

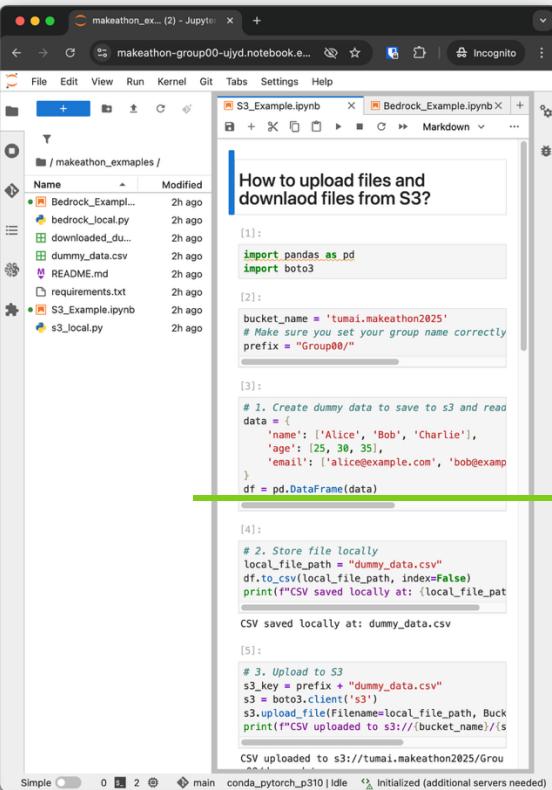


USE S3 FOR PERMANENT STORAGE

SageMaker storage is limited and may be volatile.

Make sure to use **version control** (git) for storing your code.

Store big files in S3.



The screenshot shows a Jupyter Notebook interface with two tabs: 'makeathon_ex... (2)' and 'Bedrock_Example.ipynb'. The current tab displays a code cell with the following content:

```
How to upload files and download files from S3?

[1]:
import pandas as pd
import boto3

[2]:
bucket_name = 'tumai.makeathon2025'
# Make sure you set your group name correctly
prefix = "Group00/"

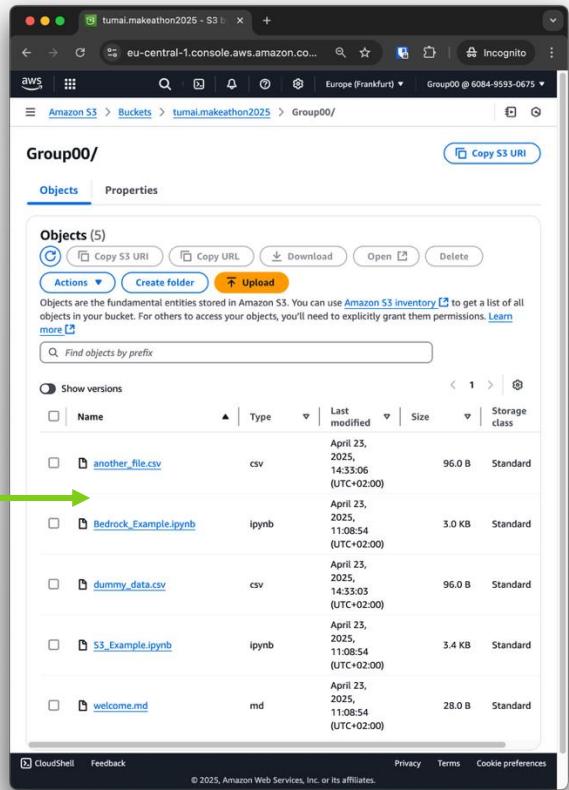
[3]:
# 1. Create dummy data to save to s3 and read
data = [
    {'name': ['Alice', 'Bob', 'Charlie'],
     'age': [25, 30, 35],
     'email': ['alice@example.com', 'bob@example.com']},
    df = pd.DataFrame(data)

[4]:
# 2. Store file locally
local_file_path = "dummy_data.csv"
df.to_csv(local_file_path, index=False)
print(f"CSV saved locally at: {local_file_path}")

CSV saved locally at: dummy_data.csv

[5]:
# 3. Upload to S3
s3_key = prefix + "dummy_data.csv"
s3 = boto3.client('s3')
s3.upload_file(Filename=local_file_path, Bucket=bucket_name, Key=s3_key)
print(f"CSV uploaded to s3:///{bucket_name}/{s3_key}")
```

A green arrow points from the bottom of this screenshot to the corresponding objects in the AWS S3 console.



The screenshot shows the AWS S3 console in a browser window. The URL is 'eu-central-1.console.aws.amazon.com'. The left sidebar shows 'Amazon S3 > Buckets > tumai.makeathon2025 > Group00/'. The main area is titled 'Group00/' and shows a list of objects:

Name	Type	Last modified	Size	Storage class
another_file.csv	csv	April 23, 2025, 14:33:06 (UTC+02:00)	96.0 B	Standard
Bedrock_Example.ipynb	ipynb	April 23, 2025, 11:08:54 (UTC+02:00)	3.0 KB	Standard
dummy_data.csv	csv	April 23, 2025, 14:33:03 (UTC+02:00)	96.0 B	Standard
S3_Example.ipynb	ipynb	April 23, 2025, 11:08:54 (UTC+02:00)	3.4 KB	Standard
welcome.md	md	April 23, 2025, 11:08:54 (UTC+02:00)	28.0 B	Standard

At the bottom, there are links for 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

