



Event ends in 2 days 15 hours 9 minutes.



[Event dashboard](#) > [Prerequisites](#) > **Self paced (Setup)**

Self paced (Setup)

 Please note the following recommendations:

- We recommend you run this workshop in the **us-east-1 (N. Virginia)** or **us-west-2 (Oregon)** region.
- If you are running the workshop in SageMaker Studio, here are the recommended kernel configurations:
 - Image: Data Science 3.0 (or greater)
 - Instance Type: ml.t3.medium

Downloading code samples from the GitHub repo

Clone the following repository, which includes examples showcased in the workshop:

```
1 git clone https://github.com/aws-samples/amazon-bedrock-workshop
```

IAM Policy for Amazon Bedrock

This workshop requires an IAM role that has access to Amazon Bedrock. Here is an example IAM policy to grant access to Bedrock APIs:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "bedrock:*",  
      "Resource": "*" }  
    ]  
}
```

```
    "Action": "bedrock:*",  
    "Resource": "*"    
  }  
]  
}
```

**Important**

If you are running this workshop in SageMaker Studio, add this policy to the SageMaker role.

If you are running into your own environment, ensure your user/role has these permissions.

Boto3 Setup

To execute this workshop, you need [Python](#) installed.



Python version must be greater than 3.8.

Next, install the boto3 and botocore [AWS SDK for Python \(Boto3\)](#) libraries. Both libraries contain required dependencies related with Bedrock APIs.

To install them, you can run the commands in the section "Prerequisites" from `bedrock_basics.ipynb` file:

```
1 pip install --no-build-isolation --force-reinstall \  
2     "boto3>=1.28.57" \  
3     "awscli>=1.29.57" \  
4     "botocore>=1.31.57"
```



boto3 version must be 1.28.57 or greater and botocore version must be 1.31.57 or greater.

Installing Supporting libraries

It's also necessary to install some supporting libraries like [FAISS](#) or some open source libraries like [LangChain](#). LangChain is a framework for

developing applications powered by language models.

Each notebook in the workshop installs the right versions of the libraries it requires. Make you sure you run those steps in every notebook to install the right version of the libraries. An example of that code is shown below:

```
1 pip install --quiet langchain==0.0.304
```

AWS Credentials

In the code repository, the file named [bedrock.py](#) in the `utils` folder instantiates boto3 to simplify the execution of this workshop. In this step, you will run this file.

If you are running this workshop on your local computer in Microsoft VS Code, PyCharm, etc., you should uncomment and fill in the variables in the following snippet to configure your credentials (to be able to run call AWS APIs):

```
1 import json
2 import os
3 import sys
4
5 import boto3
6
7 module_path = ".."
8 sys.path.append(os.path.abspath(module_path))
9 from utils import bedrock, print_wv
10
11
12 # ---- △ Un-comment and edit the below lines as needed for your AWS
13
14 # os.environ["AWS_DEFAULT_REGION"] = "<REGION_NAME>" # E.g. "us-east-1"
15 # os.environ["AWS_PROFILE"] = "<YOUR_PROFILE>"
16 # os.environ["BEDROCK_ASSUME_ROLE"] = "<YOUR_ROLE_ARN>" # E.g. "arn:aws:iam::123456789012:role/BedrockRole"
17
18
19 boto3_bedrock = bedrock.get_bedrock_client(
20     assumed_role=os.environ.get("BEDROCK_ASSUME_ROLE", None),
21     region=os.environ.get("AWS_DEFAULT_REGION", None),
22
23
```

```
runtime=False  
)
```

Explanation of Bedrock API

If you were able to run the previous steps successfully, execute the next cell block. It will list the available foundation models in your account.

```
1 boto3_bedrock.list_foundation_models()
```

Now let's call the Bedrock API and use the Amazon Titan model:

```
1 # If you'd like to try your own prompt, edit this parameter!  
2 prompt_data = """Command: Write me a blog about making strong business  
3  
4 Blog:  
5 """
```

```
1 body = json.dumps({"inputText": prompt_data})  
2 modelId = "amazon.titan-tg1-large"  
3 accept = "application/json"  
4 contentType = "application/json"  
5  
6 response = bedrock_runtime.invoke_model(  
7     body=body, modelId=modelId, accept=accept, contentType=contentType  
8 )  
9 response_body = json.loads(response.get("body").read())  
10  
11 print(response_body.get("results")[0].get("outputText"))
```

The notebook [00_prerequisites/bedrock_basics.ipynb](#) contains the steps for running run all necessary requirements for this workshop.

Now, try to execute other samples, making calls to other foundation models

Conclusion



Congratulations! In this section, you installed all prerequisites and understood Bedrock API.

Now, you can start the workshop.

[Previous](#)[Next](#)