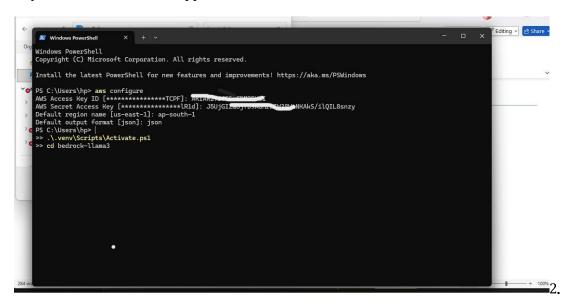
Deploying and Using AWS Bedrock

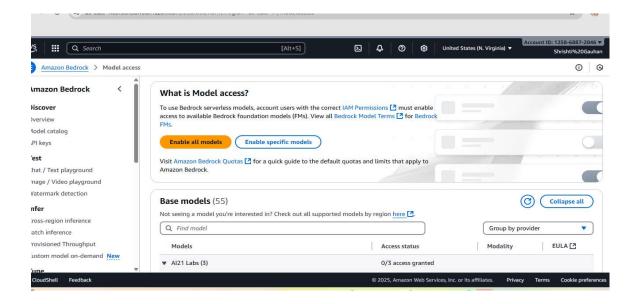
1. Prerequisites

- An AWS account with administrative access.
- Install AWS CLI: pip install awscli aws configure
- Provide AWS Access Key ID, Secret Key, and region (e.g., us-east-1).
- Python 3.9+ or another supported runtime.



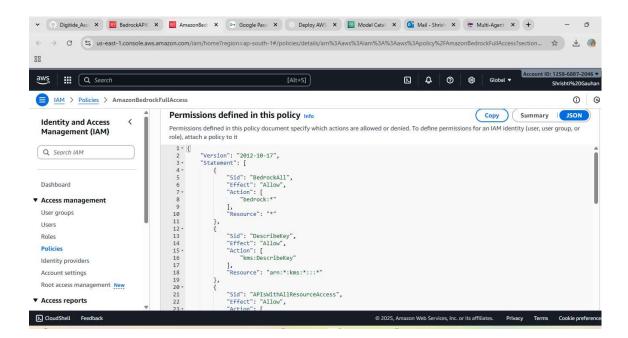
2. Enable Bedrock in Your Account

- 1. Go to AWS Bedrock Console.
- 2. Enable Model: AWS gives you a list of available foundation models (FMs) like:
 - o **Anthropic Claude** (chat & reasoning)
 - o **Meta Llama** (open-weight LLMs)
 - o **Mistral** (lightweight, fast models)
 - o **AI21 Jurassic** (text generation)
 - o **Amazon Titan** (text & embeddings)
- You can choose which models you want to use.
- Request Access → For some models, AWS requires you to click "Request model access".



3. Set IAM Permissions

Create a policy with permissions for Bedrock and attach to IAM user/role. Example policy:



4. Integrate with Python

```
📢 File Edit Selection View Go Run …

◆ bedrock_demo.py 
◆ import boto3 Untitled-1 
◆ app.py

                                                                                                                                                                               > □ ... □ ... | ... | ... | ... | ... | ... | ... | ... | ...
      ∨ BEDROCK
                                                                                                                                                                                                         > .venv
        bedrock_demo.py
                                               client = boto3.client("bedrock-runtime", region_name="ap-south-1")
MODEL_ID = "meta.llama3-8b-instruct-v1:0"
                                                                                                                                                                                                 Welcome to
                                                                                                                                                                                                      Copilot
                                                st.title(" 	☐ Chatbot with Meta Llama 3 (AWS Bedrock)")
                                                                                                                                                                                                   Let's get started
                                               if "messages" not in st.session_state
    st.session_state.messages = []
                                                for m in st.session_state.messages:
                                                     with st.chat_message(m["role"]
st.markdown(m["content"])
R
                                                                                                                                                                                                            ■ 0 ≥
                                                if prompt := st.chat_input("Ask me anything..."):
    st.session_state.messages.append({"role": "user", "content": prompt})
    with st.chat_message("user"):

    Build workspace

      > OUTLINE

✓ TIMELINE app.py

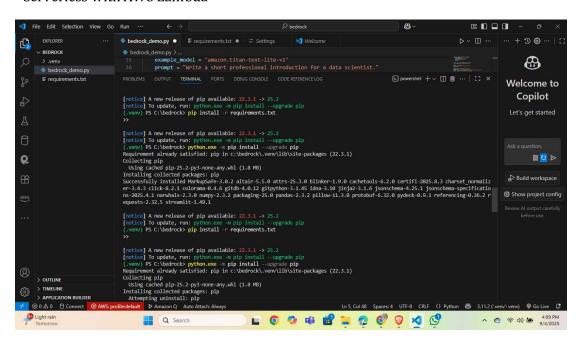
                                                                                                                                                                                                 🔀 Show project config
                                                        st.markdown(prompt)
                                                     <|begin_of_text|><|start_header_id|>user<|end_header_id|>
                                                     {prompt}
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

→ APPLICATION BUILDER

          Walkthrough of Applicati..
                                                    body = json.dumps({"prompt": formatted_prompt, "max_gen_len": 256, "temperature": 0.7, "top_p"
response = client.invoke_model(modelId=MODEL_ID, body=body)
          [No SAM templates foun...
                                                     result = json.loads(response["body"].read())["generation"]
8
                                                     st.session_state.messages.append({"role": "assistant", "content": result})
                                     Q Search
                                                                                                                                   ♥ ★ ♥
```

5. Deploy in Your Application

- Streamlit app (chatbot / RAG system)
- SageMaker pipelines (enterprise AI)
- Serverless with AWS Lambda



6. Result

