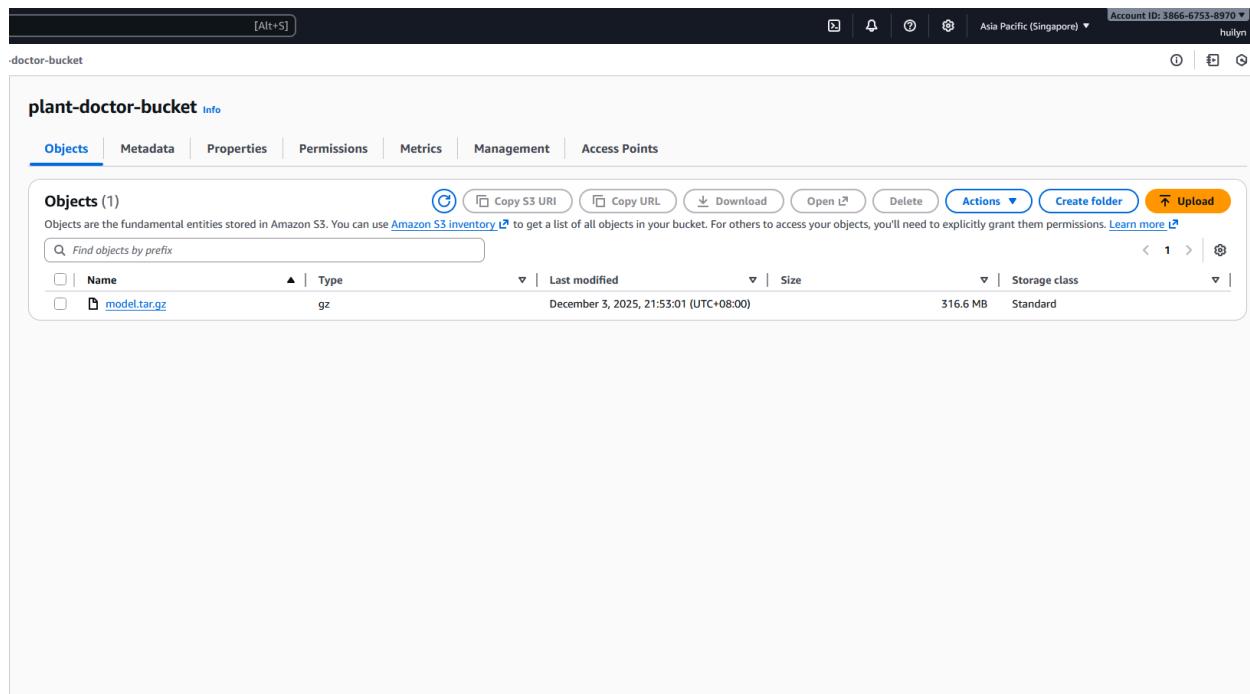


# Set Up Amazon SageMaker Notebook

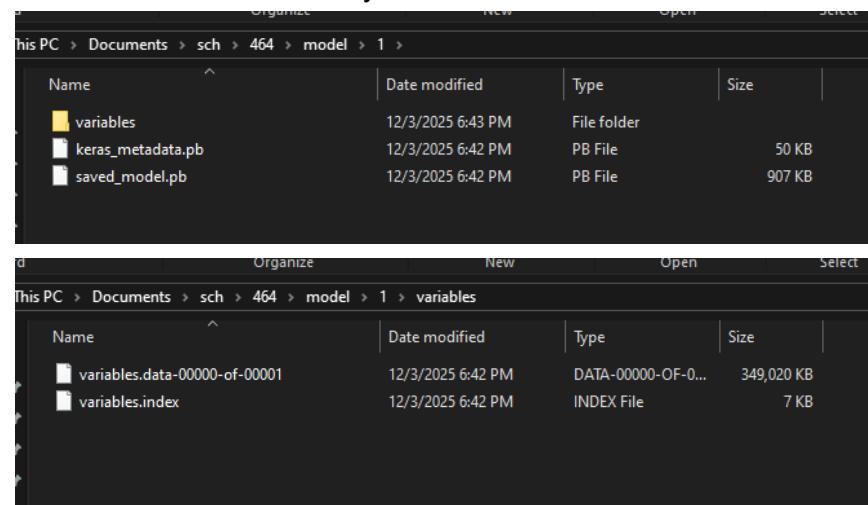
## Step 1: Create an S3 Bucket (if not already existing)

This step helps you make an S3 bucket that will house the model file and weights. Sagemaker will read from this S3 bucket.



The screenshot shows the AWS S3 console with the bucket name 'plant-doctor-bucket'. The 'Objects' tab is selected, displaying one object: 'model.tar.gz'. The file is a gzip-compressed archive, last modified on December 3, 2025, at 21:53:01 (UTC+08:00), and has a size of 316.6 MB. The storage class is Standard.

- We use plant-doctor-bucket.
- Upload [model.tar.gz](#) (from the plant-doctor-service repo) inside
  - This file should have this layout



The top screenshot shows the contents of the 'variables' folder:

| Name              | Date modified     | Type        | Size   |
|-------------------|-------------------|-------------|--------|
| variables         | 12/3/2025 6:43 PM | File folder |        |
| keras_metadata.pb | 12/3/2025 6:42 PM | PB File     | 50 KB  |
| saved_model.pb    | 12/3/2025 6:42 PM | PB File     | 907 KB |

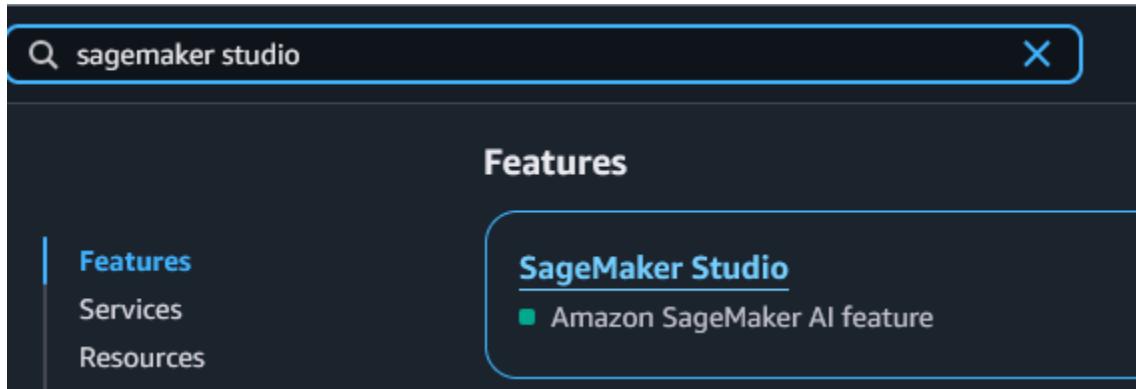
The bottom screenshot shows the contents of the 'variables\data-00000-of-00001' folder:

| Name                          | Date modified     | Type               | Size       |
|-------------------------------|-------------------|--------------------|------------|
| variables.data-00000-of-00001 | 12/3/2025 6:42 PM | DATA-00000-OF-0... | 349,020 KB |
| variables.index               | 12/3/2025 6:42 PM | INDEX File         | 7 KB       |

## Step 2: Launch SageMaker Notebook Instance

This step will tell you how to deploy the model so that the Plant-Doctor-Service backend can call it

Search Sagemaker Studio



Go to Notebooks from left sidebar > Create notebook instance

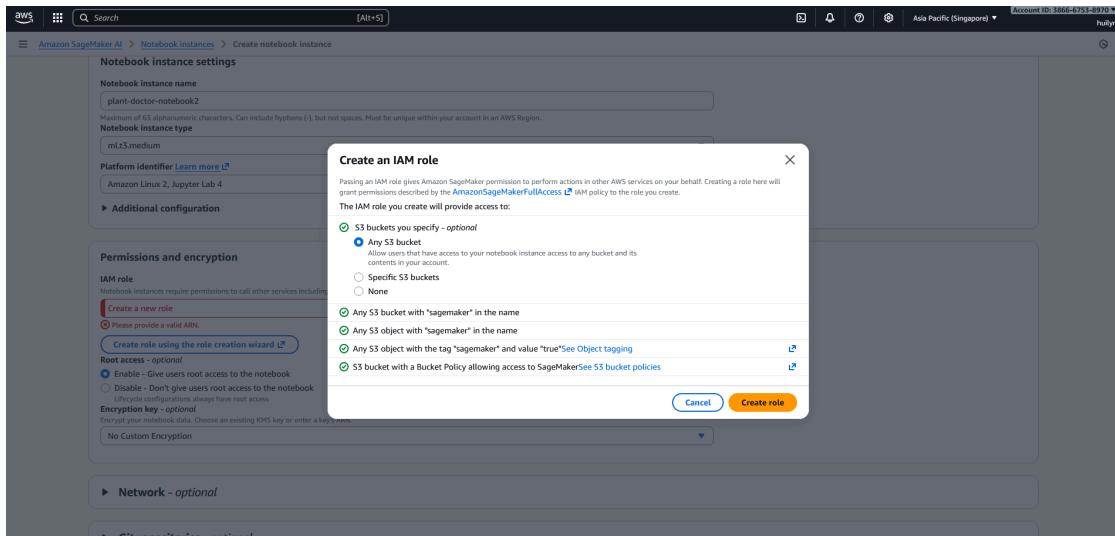
A screenshot of the Amazon SageMaker AI console. The left sidebar shows various navigation options like "Dashboard", "Environment configuration", "Applications and IDEs", "Model training &amp; customization", "Training plans", "Deployments &amp; inference", "Model governance", and "AWS Marketplace resources". The main content area is titled "Notebooks and Git repos". It features a "Try the new JupyterLab in SageMaker Studio" section with a "Get Started" button. Below this, there are tabs for "Notebook instances" and "Git repositories", with "Notebook instances" currently selected. A table titled "Notebook instances info" lists one instance: "plant-doctor-notebook" (ml.t3.medium, creation time 12/3/2025, 6:50:05 PM, status inService). There is a "Create notebook instance" button at the top right of the table.

Can use these notebook instance settings.

The screenshot shows the 'Create notebook instance' configuration page. It includes fields for 'Notebook instance name' (set to 'plant-doctor-notebook'), 'Notebook instance type' (set to 'ml.t3.medium'), 'Platform identifier' (set to 'Amazon Linux 2, Jupyter Lab 4'), and an 'Additional configuration' section. Below these, the 'Permissions and encryption' section is expanded, showing an 'IAM role' dropdown containing 'AmazonSageMaker-ExecutionRole-20251203T184975'. Underneath, there are options for 'Root access - optional' (with 'Enable' selected) and 'Encryption key - optional' (set to 'No Custom Encryption').

For IAM role, select “Create a new role” and click Any S3 bucket and create role

This screenshot shows the first step of the IAM role creation wizard. It has three tabs: 'Create a new role' (selected), 'Enter a custom role name' (disabled), and 'Use existing role'. The 'Create a new role' tab displays the role name 'AmazonSageMaker-ExecutionRole-20251203T184975'. A note at the bottom states: 'Lifecycle configurations always have root access'.



Once created, click “Open Jupyter”

**Notebooks and Git repos**

▼ Try the new JupyterLab in SageMaker Studio

Try the new JupyterLab in SageMaker Studio

- Launch notebooks in seconds and start coding instantly
- Use the similar underlying compute and storage as your notebook instances to enable more features at the same cost
- Seamlessly perform comprehensive ML and analytics workflows, all in one notebook
- Leverage GenAI-powered coding assistance from Amazon Q Developer and JupyterAI to accelerate development
- Collaborate with your peers in real-time on the same notebook for seamless ideation

► How to access JupyterLab in Studio?

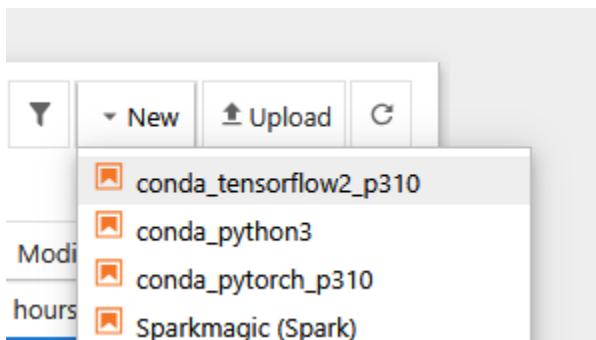
Get Started

**Notebook instances** | **Git repositories**

**Notebook instances** Info

| Name                  | Instance     | Creation time         | Status    | Actions                        |
|-----------------------|--------------|-----------------------|-----------|--------------------------------|
| plant-doctor-notebook | ml.t3.medium | 12/3/2025, 6:50:05 PM | InService | Open Jupyter   Open JupyterLab |

Create a new tensorflow2 ipynb in jupyter and name it something like plant-doctor



Paste into the first cell and run:

```

%%writefile inference.py
import tensorflow as tf
import json
import numpy as np

def model_fn(model_dir):
    return tf.saved_model.load(model_dir)

def input_fn(request_body, content_type):
    data = json.loads(request_body)
    return tf.convert_to_tensor(data["instances"], dtype=tf.float32)

def predict_fn(input_data, model):
    infer = model.signatures["serving_default"]
    output = infer(input_data)
    return {k: v.numpy().tolist() for k, v in output.items()}

def output_fn(prediction, accept):
    return json.dumps(prediction)

```

This makes a [inference.py](#) in your jupyter which helps run the plant-doctor model

Paste into the second file

NOTE: replace plant-doctor-bucket with your s3 bucket name. Ours is plant-doctor-bucket, but change the name based on what u called ur s3

```

import sagemaker
from sagemaker.tensorflow import TensorFlowModel
from sagemaker import get_execution_role

# ✓ Step 1: Use Singapore region
region = "ap-southeast-1"
sagemaker_session = sagemaker.Session()

# ✓ Step 2: Correct S3 bucket and model path
model_s3_path = "s3://plant-doctor-bucket/model.tar.gz"

# ✓ Step 3: Use your execution role (must exist in ap-southeast-1)
role = get_execution_role()

# ✓ Step 4: Create and deploy the TensorFlow model
model = TensorFlowModel(
    model_data=model_s3_path,
    role=role,
    framework_version="2.11",  # Match your TensorFlow version

```

```
sagemaker_session=sagemaker_session,  
)  
  
predictor = model.deploy(  
    initial_instance_count=1,  
    instance_type="ml.t2.medium", # Cheapest instance for testing  
)  
  
print("✅ Deployment complete!")  
print("Endpoint name:", predictor.endpoint_name)
```

The final file should look like this

The screenshot shows a Jupyter Notebook interface with the following content:

```
[3]: %%writefile inference.py
import tensorflow as tf
import json
import numpy as np

def model_fn(model_dir):
    return tf.saved_model.load(model_dir)

def input_fn(request_body, content_type):
    data = json.loads(request_body)
    return tf.convert_to_tensor(data["instances"], dtype=tf.float32)

def predict_fn(input_data, model):
    infer = model.signatures["serving_default"]
    output = infer(input_data)
    return {k: v.numpy().tolist() for k, v in output.items()}

def output_fn(prediction, accept):
    return json.dumps(prediction)

Writing inference.py
```

```
[7]: import sagemaker
from sagemaker.tensorflow import TensorFlowModel
from sagemaker import get_execution_role

# Step 1: Use Singapore region
region = "ap-southeast-1"
sagemaker_session = sagemaker.Session()

# Step 2: Correct S3 bucket and model path (from your screenshot)
model_s3_path = "s3://plant-doctor-bucket/model.tar.gz"

# Step 3: Use your execution role (must exist in ap-southeast-1)
role = get_execution_role()

# Step 4: Create and deploy the TensorFlow model
model = TensorFlowModel(
    model_data=model_s3_path,
    role=role,
    framework_version="2.11", # Match your TensorFlow version
    sagemaker_session=sagemaker_session,
)

predictor = model.deploy(
    initial_instance_count=1,
    instance_type="ml.t2.medium", # Cheapest instance for testing
)

print("Deployment complete!")
print("Endpoint name:", predictor.endpoint_name)
```

----! Deployment complete!  
Endpoint name: tensorflow-inference-2025-12-03-13-53-42-096

[ ]:

## ADDITIONAL NOTES!

For plant-doctor-service task, do check if you have the following env values  
 AWS\_REGION=ap-southeast-1  
 SAGEMAKER\_ENDPOINT={Endpoint generated from plant-doctor}  
*[ie, tensorflow-inference-2025-12-03-13-53-42-096]*

aws Search [Alt+S] Account ID: 3866-6753-8970 hulyin

Amazon Elastic Container Service > Task definitions > plant-doctor-task > Revision 2 > Containers

### plant-doctor-task:2

Last updated December 4, 2025, 14:58 (UTC+8:00) Deploy Actions Create new revision

**Overview** Info ARN amavses:ap-southeast-1:386667538970:task-definition/plant-doctor-task:2 Status ACTIVE Time created December 3, 2025, 22:53 (UTC+8:00) App environment Fargate Task role plant-doctor-role Task execution role ecsTaskExecutionRole Operating system/Architecture Linux/X86\_64 Network mode awsvpc Fault injection Turned off

**Containers** JSON Task placement Volumes (0) Requires attributes Tags

**Task size**

Task CPU 1,024 units (1 vCPU)

Task CPU maximum allocation for containers

Task memory 3,072 MiB (3 GB)

Task memory maximum allocation for container memory reservation

**Container: plant-doctor-service** Info Details JSON Essential container

Image 386667538970.dkr.ecr.ap-southeast-1.amazonaws.com/plant-doctor- Private registry Turned off Secrets Manager ARN or name -

### ▼ Container: plant-doctor-service

**Details** | **JSON**

**Image** 386667538970.dkr.ecr.ap-southeast-1.amazonaws.com/plant-doctor-service@sha256:a5fd08d0c5de93360ea8b0b5565764143dff4e2549ac3350b7b5a69376f5e847

**CPU** 0

**Private registry** Turned off

**Memory hard/soft limit** -/-

**Environment and secrets** Network settings Security and permissions Lifecycle and dependencies M

#### Environment variables (2)

| Key                | Type  | Value  |
|--------------------|-------|--|
| AWS_REGION         | value | ap-southeast-1                               |
| SAGEMAKER_ENDPOINT | value | tensorflow-inference-2025-12-03-13-53-42-096 |

#### Environment files (S3 ARN)

-