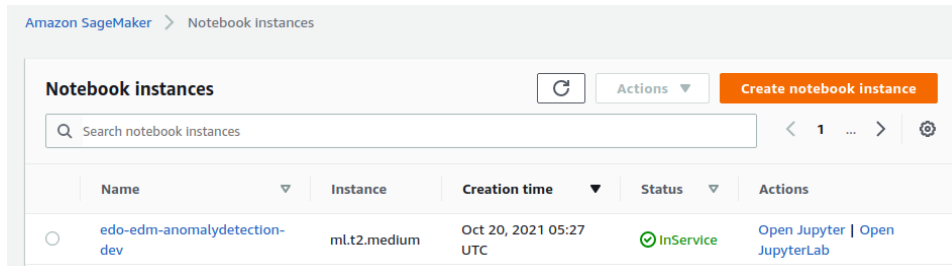


Step 1: Run the terraform script – which will create the notebook instance and ECR repository

Step 2: Go to the notebook instance which got created after executing the terraform script (edo-edm-anomalydetection-qa)

Sample screenshot from dev

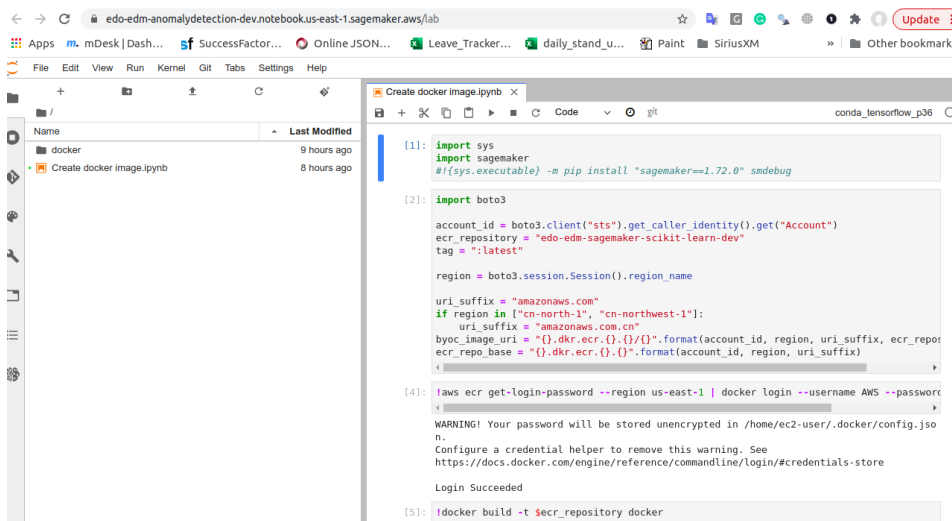


The screenshot shows the Amazon SageMaker console's 'Notebook Instances' page. At the top, there's a search bar and a 'Create notebook instance' button. Below is a table with one instance listed: 'edo-edm-anomalydetection-dev'. The instance is in 'InService' status and has links to 'Open Jupyter' and 'Open JupyterLab'.

Name	Instance	Creation time	Status	Actions
edo-edm-anomalydetection-dev	ml.t2.medium	Oct 20, 2021 05:27 UTC	InService	Open Jupyter Open JupyterLab

Step 3: Launch jupyterlab by clicking on 'Open JupyterLab'

Step 4: Upload content of "Docker image creation - qa.zip" with the same folder structure as shown in the screenshot.



The screenshot shows the JupyterLab interface. On the left is a file browser with a 'docker' folder and a 'Create docker image.ipynb' file. The main area displays the code from the notebook, which is a Python script for creating a Docker image and pushing it to ECR.

```
[1]: import sys
import sagemaker
# [sys.executable] -m pip install "sagemaker==1.72.0" smdebug

[2]: import boto3

account_id = boto3.client("sts").get_caller_identity().get("Account")
ecr_repository = "edo-edm-sagemaker-sci-kit-learn-dev"
tag = ":latest"

region = boto3.session.Session().region_name

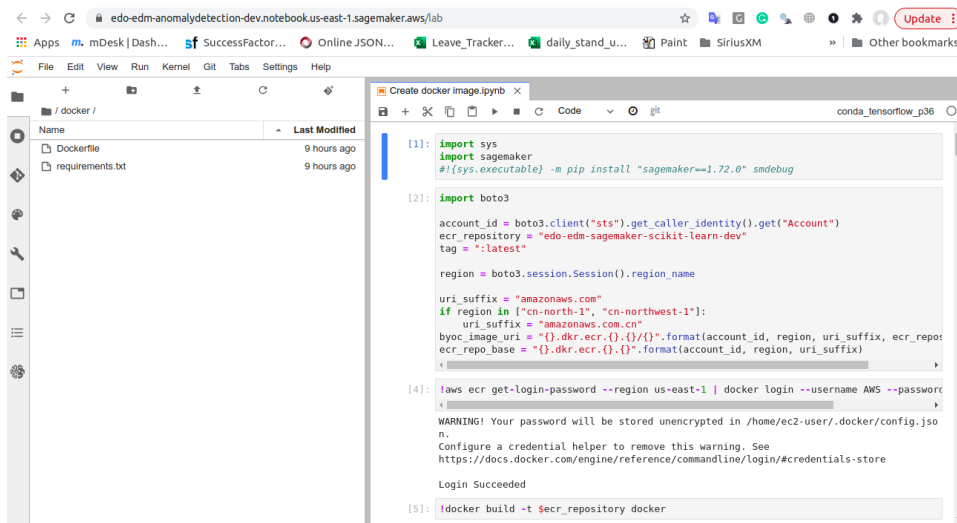
uri_suffix = "amazonaws.com"
if region in ["cn-north-1", "cn-northwest-1"]:
    uri_suffix = "amazonaws.com.cn"
byoc_image_uri = "{}.dkr.ecr.{}".format(account_id, region, uri_suffix)
ecr_repo_base = "{}.dkr.ecr.{}".format(account_id, region, uri_suffix)

[4]: aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin

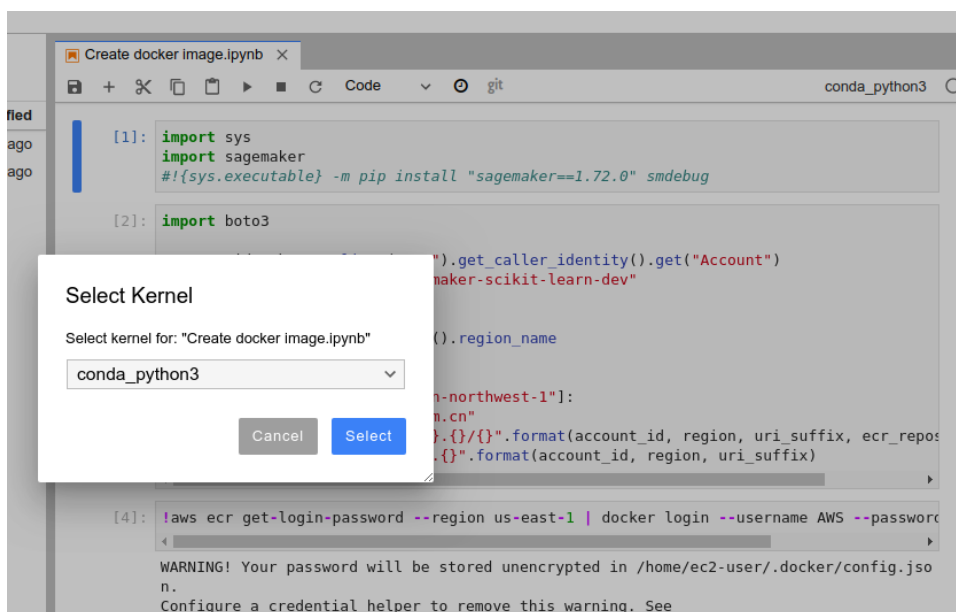
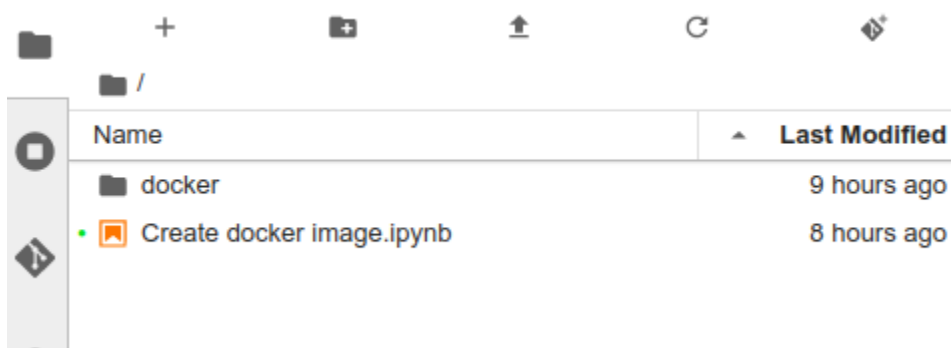
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

[5]: docker build -t $ecr_repository docker
```



Step 5: Open the jupyter notebook by double clicking on “Create docker image.ipynb” and select the kernel “conda_python3”



Step 6: Run through the notebook, which will create the image and push it to ECR – this step will take around 20 minutes