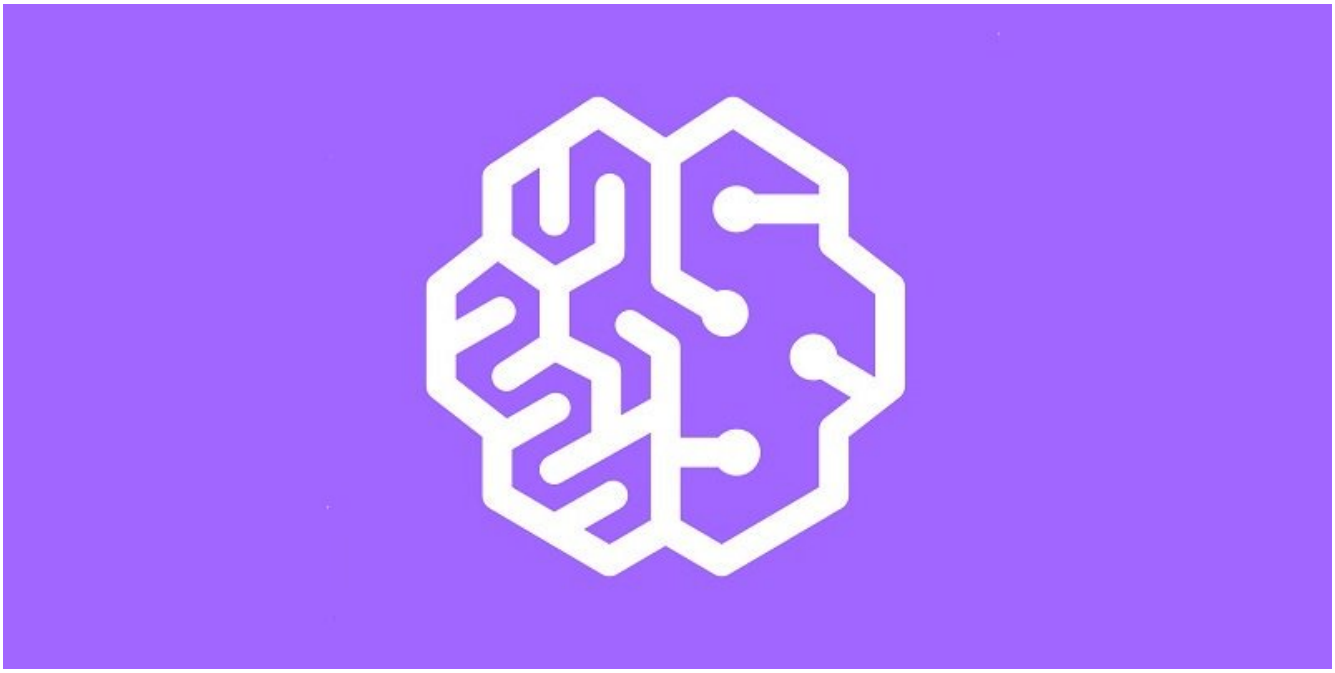


Getting Started

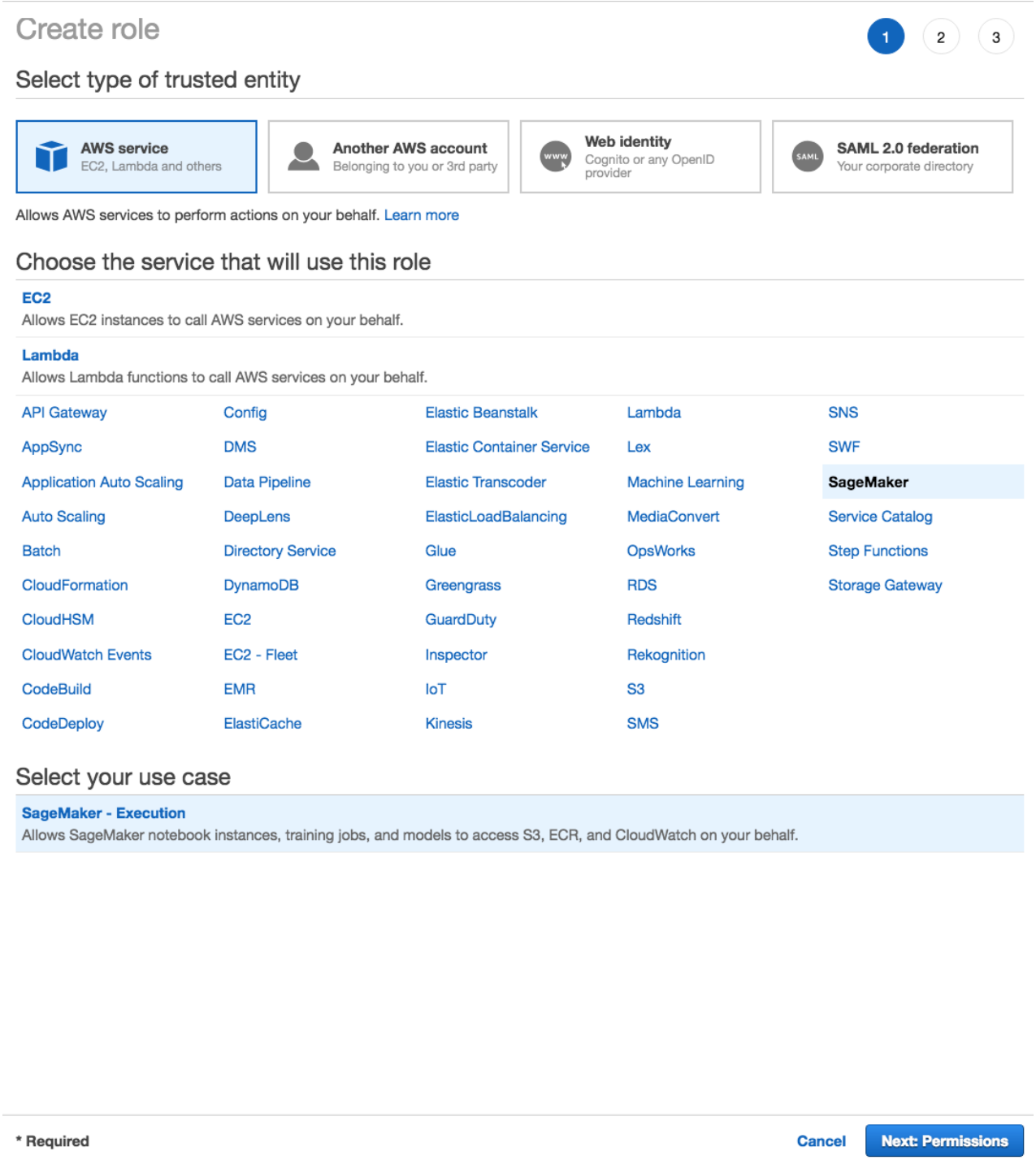
SageMaker notebook instances provide a managed Jupyter environment for data scientists to build, experiment, and deploy models from programmatically using Jupyter notebooks. For this lab you'll be working from your own SageMaker notebook instance using detailed Jupyter notebooks to train and deploy your models.

Prepare Notebook Instance IAM Role

Before we launch our SageMaker notebook instance and begin with the labs we need to create an IAM role for our SageMaker notebook instance to allow it to access AWS services on our behalf.

Create IAM Role

1. Navigate to the AWS Management Console for IAM.
2. On the lefthand side of the window click the **Roles** tab.
3. Click the **Create role** button.
4. Under **Choose the service that will use this role** click **SageMaker** then click the **Next: Permissions** button.
5. Click the **Next: Review** button.
6. In the Role name field enter **SageMakerLabRole**. **Note:** If you are in a shared account you may get the error **'A role named "SageMakerLabRole" already exists'**. This means a peer working in the shared account has already created the role on your behalf. If this is the case hit the **Cancel** button and proceed to the **Preparing Your SageMaker Notebook Instance** section.
7. Click the **Create role** button to create the IAM role.



Add ECR Policy to IAM Role

The notebook instance role created in the last section has all the permissions required for our notebook instance to access the SageMaker service, however we need to add additional permissions for our SciKit Bring-Your-Own Model lab module where we will be creating an ECR repository and pushing a Docker image to it. For the purposes of this lab we will add the **AmazonEC2ContainerRegistryFullAccess** managed policy to the role to grant these permissions. To do this follow the instructions below:

1. Navigate to the Roles tab of the IAM console.
2. In the **Search** field enter **SageMakerLabRole** to filter down to the role we created in the previous section.

3. Under the **Role name** column click the role name to navigate to the role summary page.
4. Click the **Attach policy** button.
5. In the **Search** field enter **AmazonEC2ContainerRegistryFullAccess** to filter down to the managed permissions policy we wish to add to the role.
6. Click the checkbox for the permissions policy on the lefthand side of the row then click the **Attach policy** button to add the policy to the role.

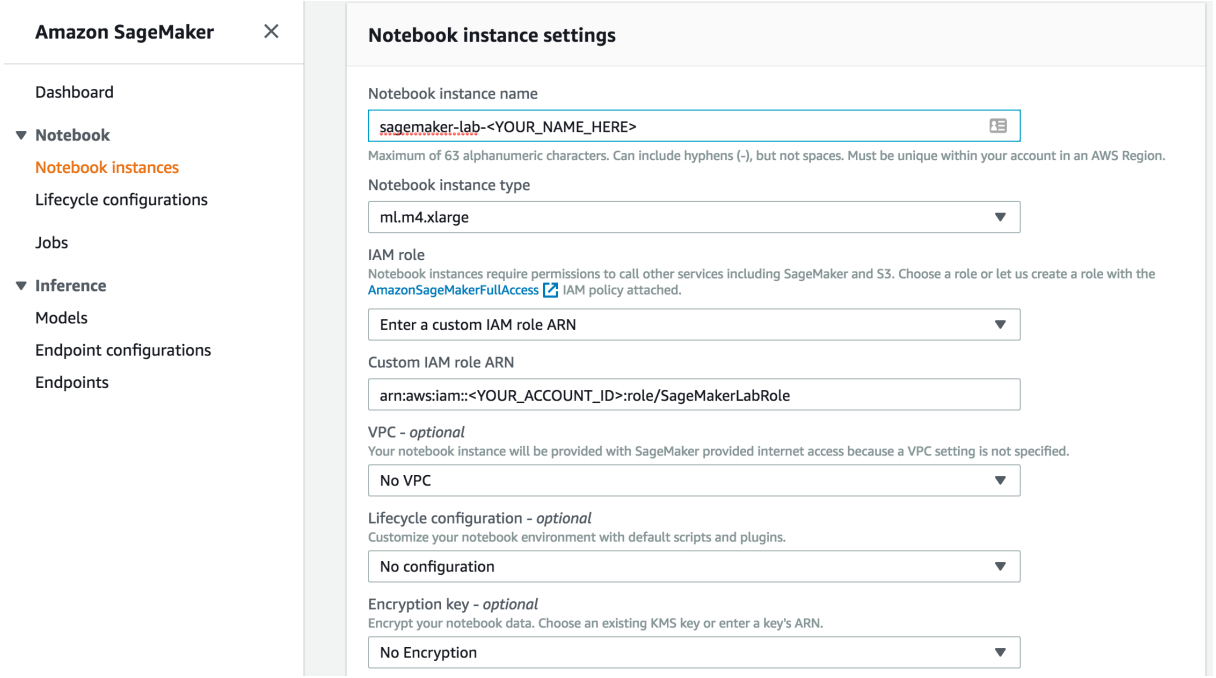
Preparing Your SageMaker Notebook Instance

Our next steps are to launch a SageMaker notebook instance with the created role then download our lab module Jupyter notebooks to the instance.

Create SageMaker Notebook Instance

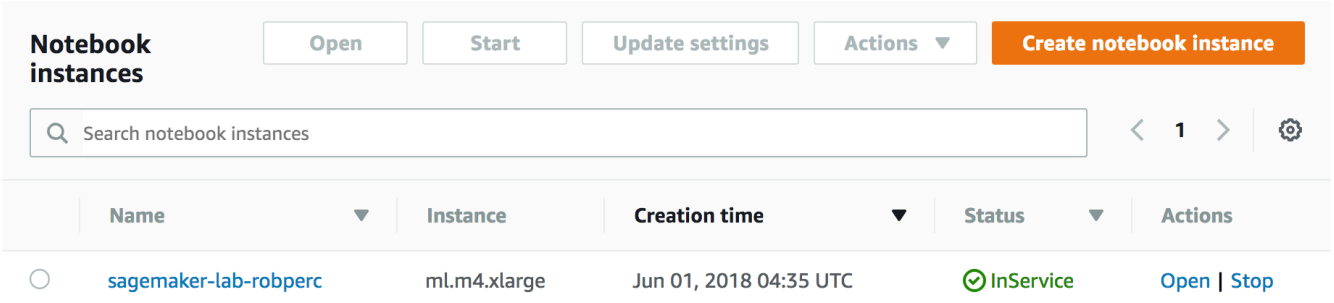
1. Open the AWS Management Console for SageMaker.
2. Verify your region by looking in the top right hand corner of the AWS Mangement Console. Choose the **US West (Oregon)** region.
3. If this is your first time visiting the AWS Management Console for SageMaker, you will see a Getting Started page, otherwise you will be redirected to the SageMaker dashboard overview page. Either way, click **Create notebook instance**. This will lead you to an instance configuration page where you will want to set the following settings:

    1. Set **Notebook instance name** to **sagemaker-lab-YOURUSERNAME**.
    2. Set **Notebook instance type** to **ml.m4.xlarge**.
    3. Set **IAM role** to **Enter a custom IAM role ARN**. This will make an extra setting titled **Custom IAM role ARN** appear.
    4. Set **Custom IAM role ARN** to **arn:aws:iam::YOUR_ACCOUNT_ID:role/SageMakerLabRole**, substituting in your AWS account ID where appropriate. If you need help figuring out your AWS account ID click here.
    5. Leave the remaining settings as their default values.



4. Scroll to the bottom of the page and click **Create notebook instance**.

This will begin the creation process for your notebook instance and redirect you to the notebook instances dashboard. Under the **Status** column your instance will show as **Pending** while it is being created. Once the **Status** moves into **InService** it can be opened by clicking **Open** under the **Actions** column.



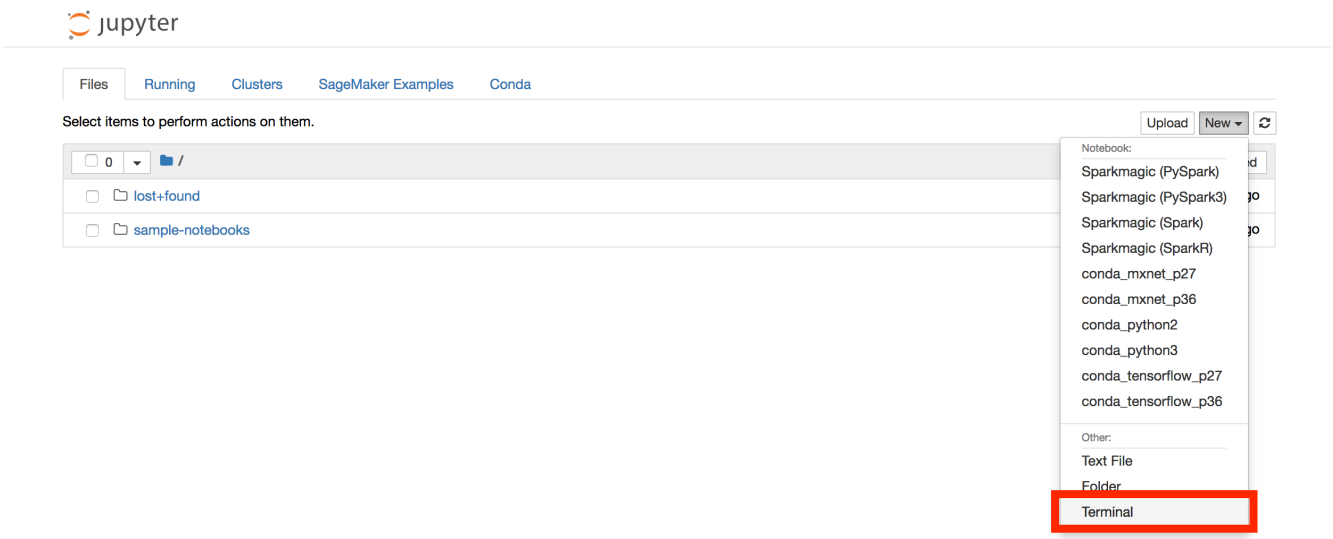
Download Lab Module Jupyter Notebooks to Instance

SageMaker notebook instances come preloaded with sample notebooks to help customers get started with learning more about machine learning and the platform. For this workshop we'll be working with custom prepared notebooks not included in this set of sample notebooks. To download these to your notebook instance use the following instructions:

1. From the Jupyter console opened in the previous section click the **New** button in the upper righthand side of the window. From the resulting dropdown click **Terminal**. This will open a new window with a bash terminal session.
2. From the bash terminal session run the following code to download the Jupyter notebooks for the lab modules to your instance.

```
cd SageMaker
git clone https://github.com/robperc/sagemaker-workshop.git
exit
```

3. Close the bash terminal session window and return to your original Jupyter console window.
4. Click the **sagemaker-workshop** folder to find the folders containing the Jupyter notebooks for the lab modules.



Lab Modules

There are two lab modules included in this workshop. The first, **deepar-retail-forecasting**, shows how to use SageMaker's built-in DeepAR forecasting algorithm to forecast product sales using a synthetic clothing retail dataset. The second, **scikit-byo-model** shows how leverage SageMaker for model training and hosting when bringing your own model framework, in this case SciKit-Learn. Use the Jupyter notebooks in the respective folders of each lab module to proceed.

Next Steps

After this lab you should have a basic understanding of how to take a model from training to production using a built-in model algorithm as well as the bring-your-own model approach. For more hands-on experience with SageMaker check out the example Jupyter notebooks in the **sample-notebooks** folder of your SageMaker notebook instance or in the amazon-sagemaker-examples GitHub repository.