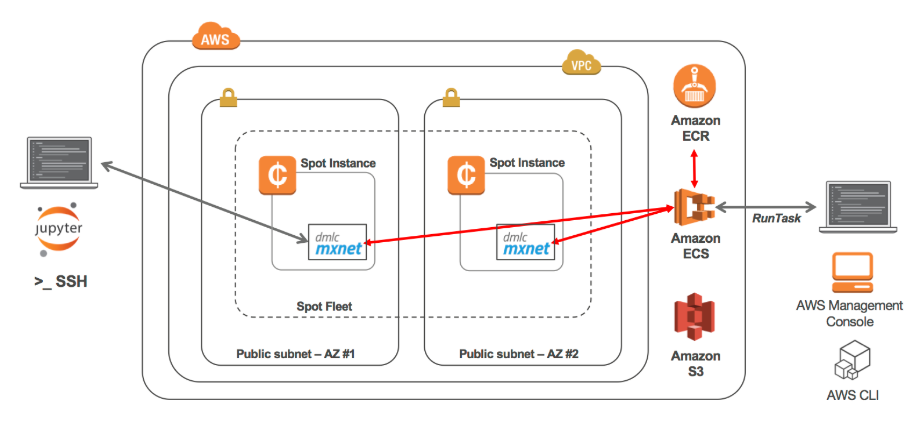
## Capstone Project

**Objective:** To deploy a deep learning environment on AWS using ECS, EC2, ECR, IAM

**Background:**

* This workshop will walk you through the deployment of a deep learning library called MXNet on AWS using Docker containers.
* There are just not enough cat pictures on social media these days, to the point where it would be amazing to have a social network dedicated to devoted cat lovers around the world. The problem is, how do you make sure images uploaded to this niche network are cat related? Image classification to the rescue
* Implement MXNet to recognize a variety of images, so you can specifically identify ones of our favorite feline friend
* Here is the overall architecture of what you will be building throughout this workshop.

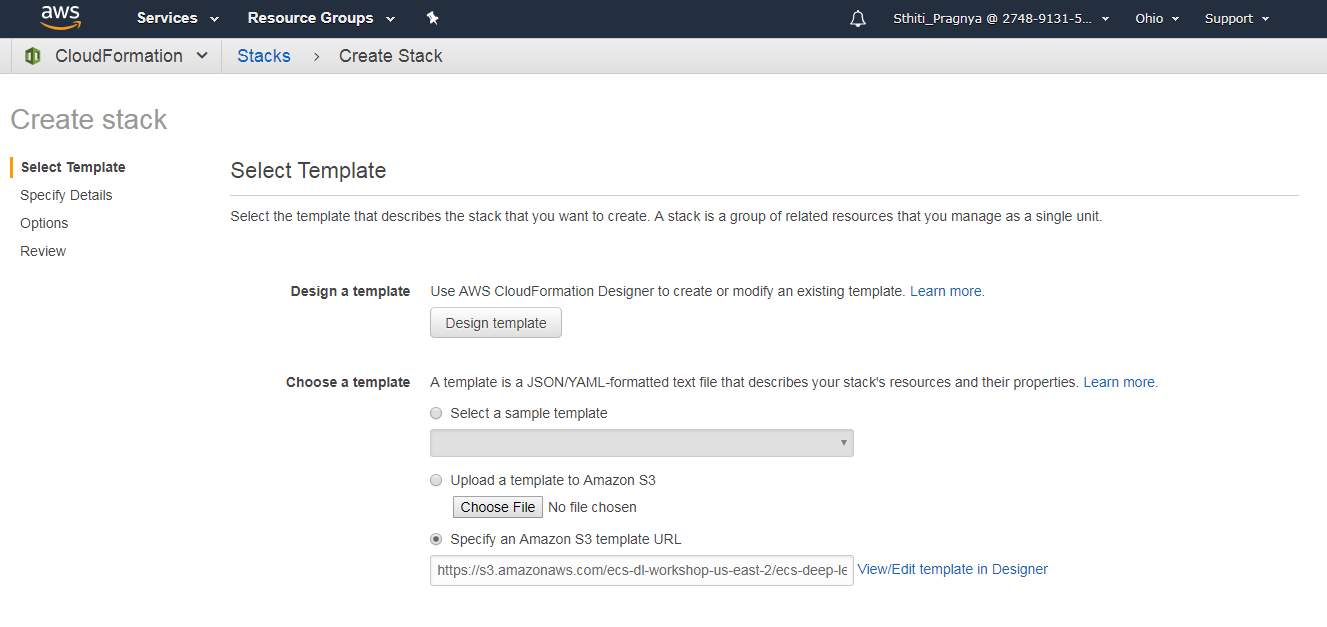


**Implementation**: In this exerciseyou will do the below tasks:

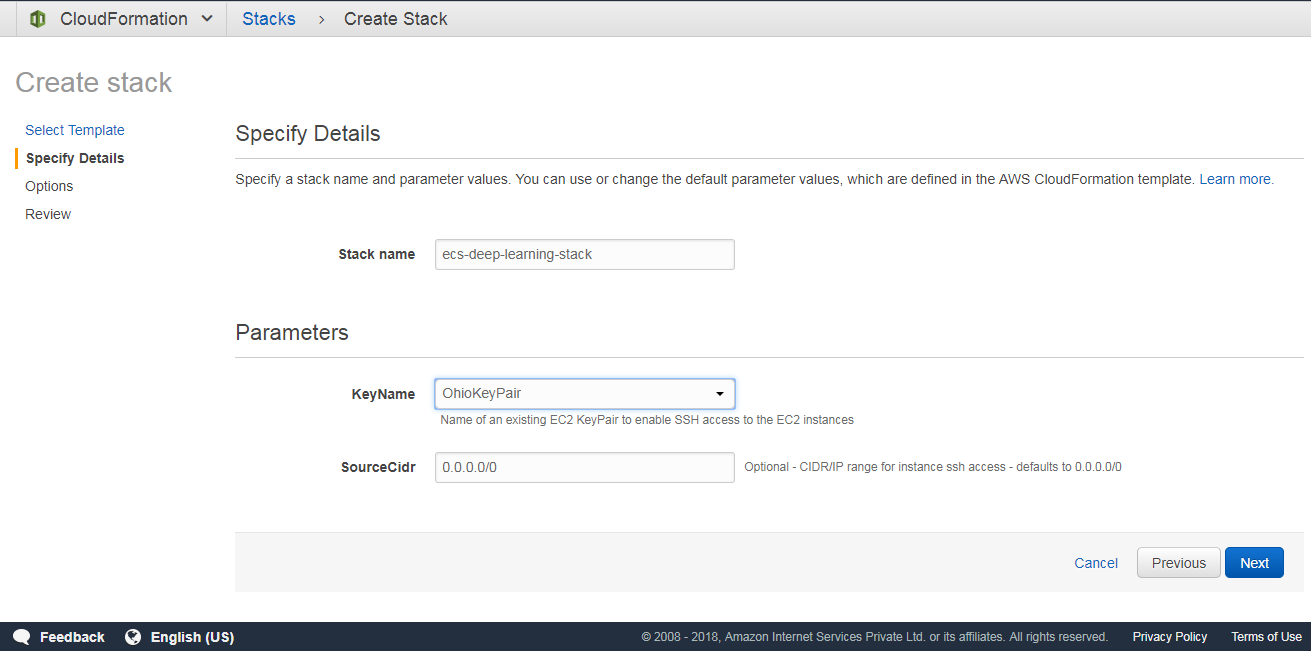
* **Task 1:** Setup the workshop environment on AWS
  + Click on one of these CloudFormation templates that matches the region you created your keypair in to launch your stack

|  |  |
| --- | --- |
| Ohio region(us-east-2) | <https://console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/new?stackName=ecs-deep-learning-stack&templateURL=https://s3.amazonaws.com/ecs-dl-workshop-us-east-2/ecs-deep-learning-workshop.yaml> |
| Oregon(us-west-2) | <https://console.aws.amazon.com/cloudformation/home?region=us-west-2#/stacks/new?stackName=ecs-deep-learning-stack&templateURL=https://s3.amazonaws.com/ecs-dl-workshop-us-west-2/ecs-deep-learning-workshop.yaml> |

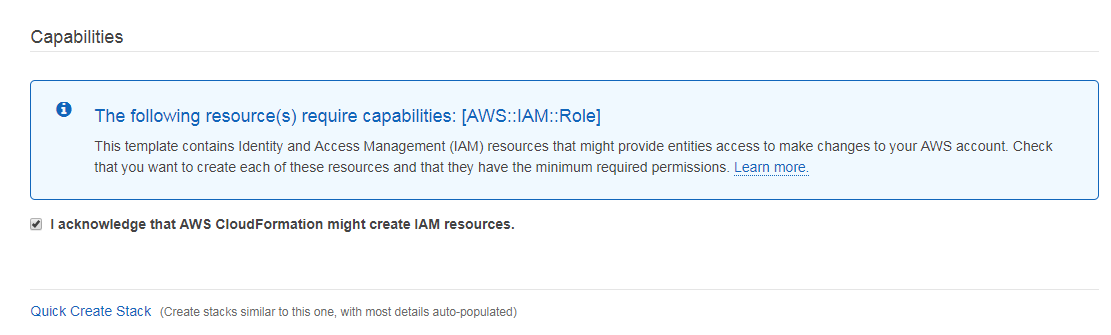
* + The template will automatically bring you to the CloudFormation Dashboard and start the stack creation process in the specified region. Click "Next"



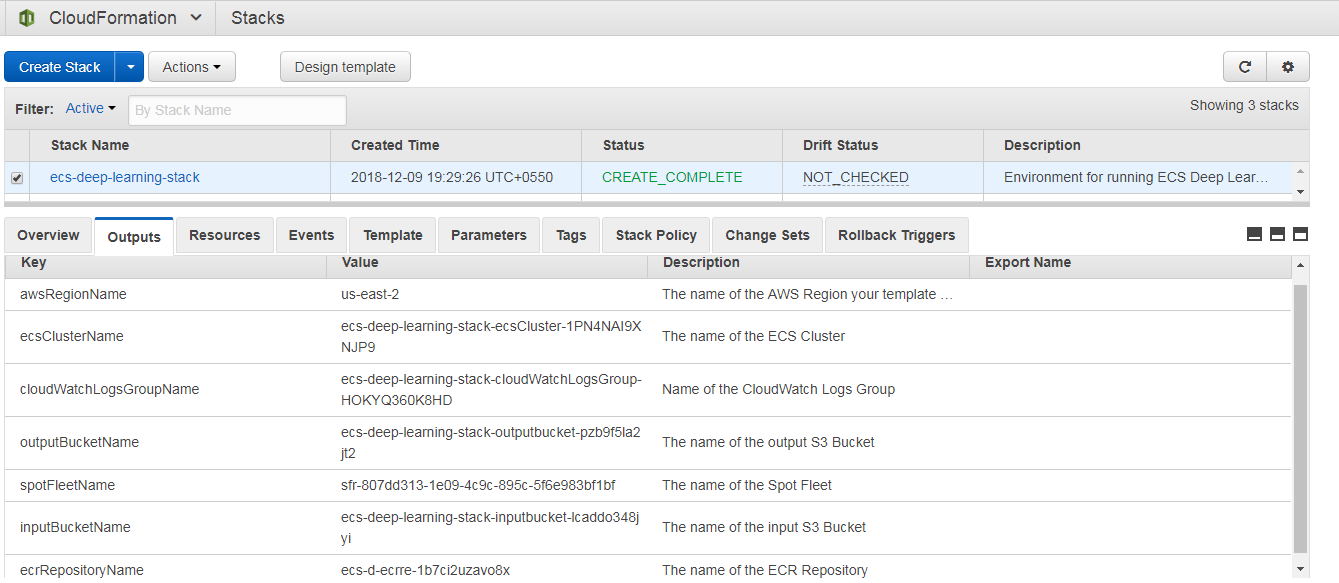
* + Make sure to choose an existing key pair in the region



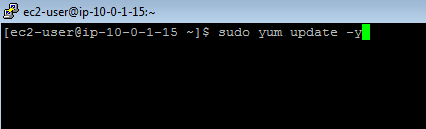
* + On the Review page, under Capabilities check the box next to "I acknowledge that AWS CloudFormation might create IAM resources." and click Create



* + In the Outputs tab, take note of the ecrRepository and spotFleetName values

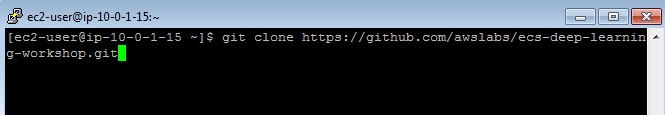


* **Task 2:** Build an MXNet Docker Image
  + Go to the EC2 Dashboard and connect to the EC2 instance created by the CloudFormation stack

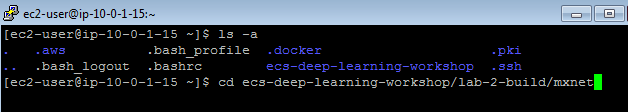


* + Clone the workshop github repository so you can easily access the Dockerfile:

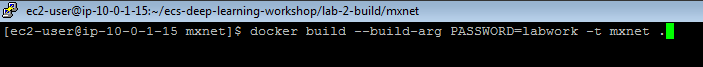
**git clone** [***http://infygit.ad.infosys.com/Sthiti\_Pragnya/awscapestoneproject2.git***](http://infygit.ad.infosys.com/Sthiti_Pragnya/awscapestoneproject2.git)



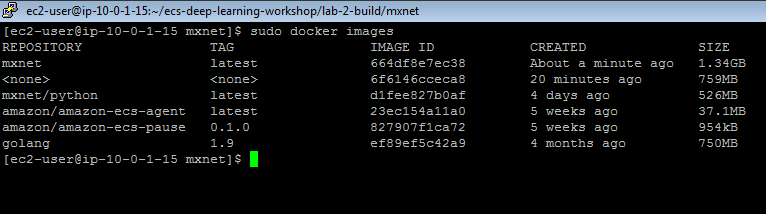
* + Navigate to the **lab-2-build/mxnet/** folder to use as your working directory



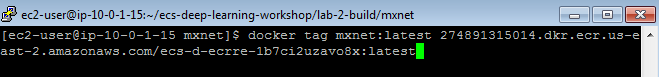
* + Build the Docker image using the provided Dockerfile



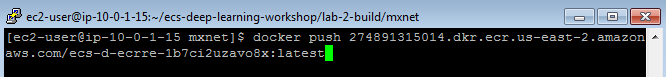
* + Run docker images to check for the images generated from the Dockerfile



* + Tag the docker image mxnet latest

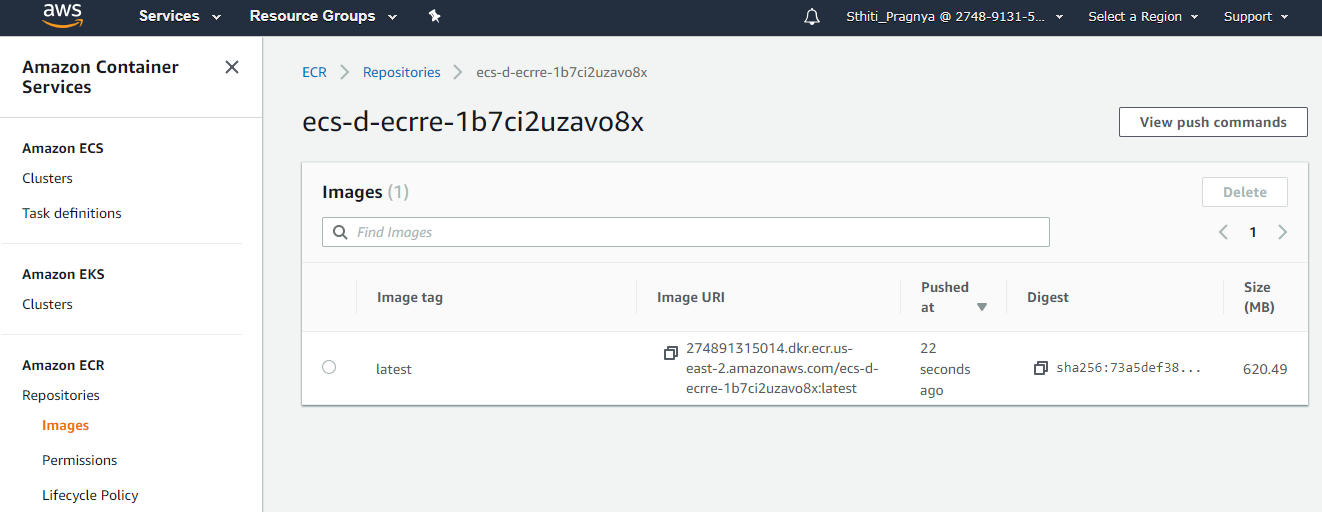


* + Run the command to push image into the ECR repository already created as a part of the CloudFormation stack

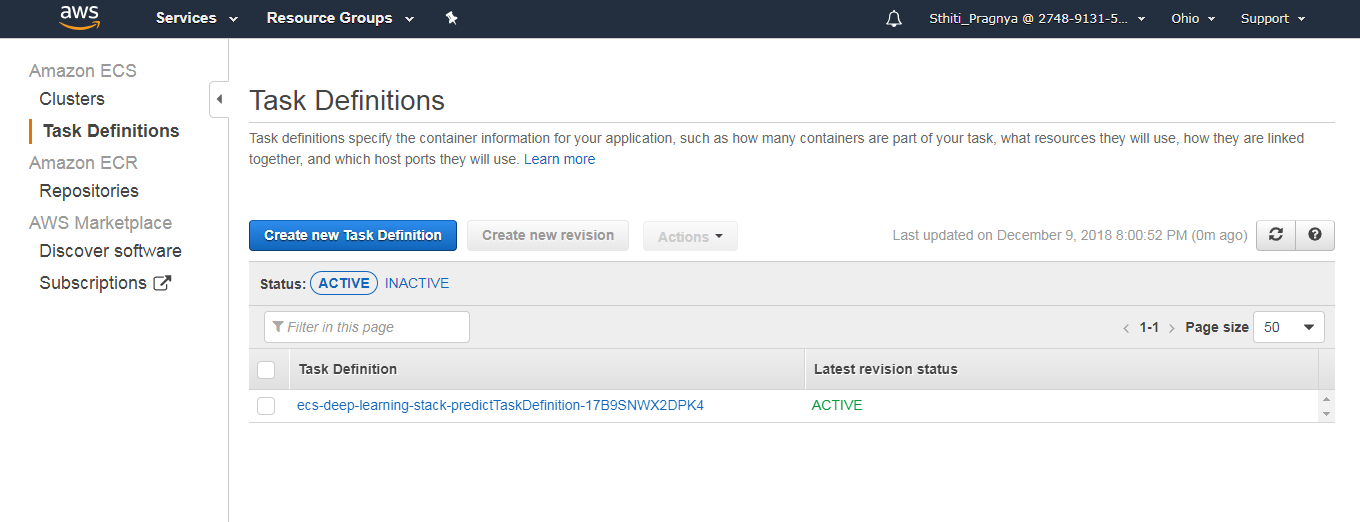


**Note:** Check the build, push, tag commands available for the ECR repository created as a part of CloudFormation stack and make necessary changes to the tag and push commands above

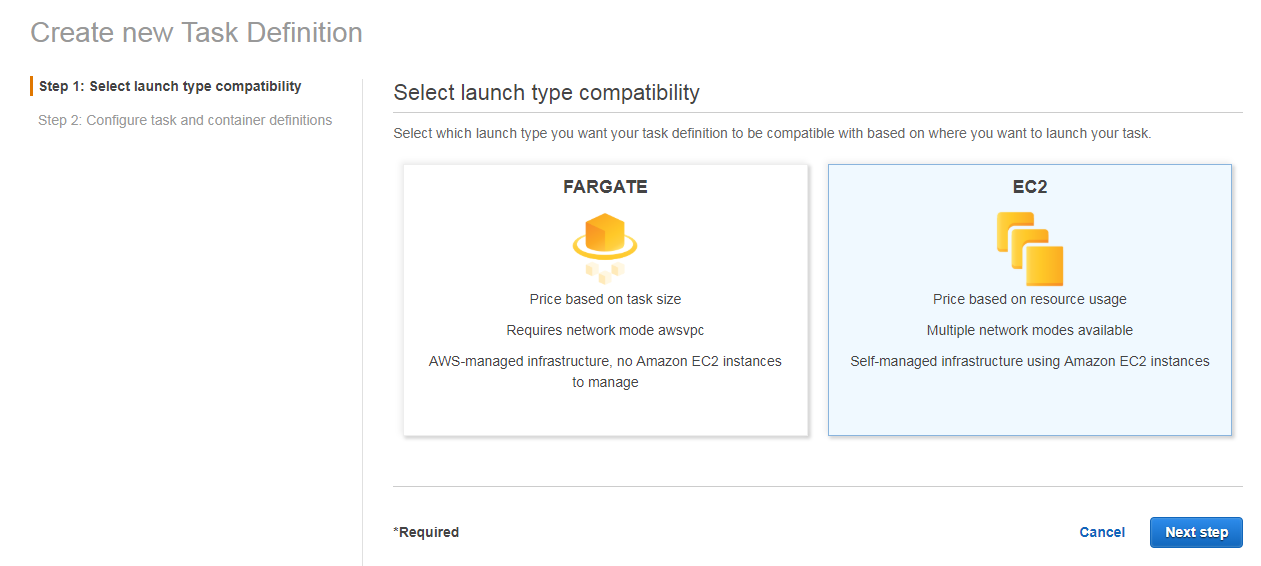
* + Check for the image in the ECR repo



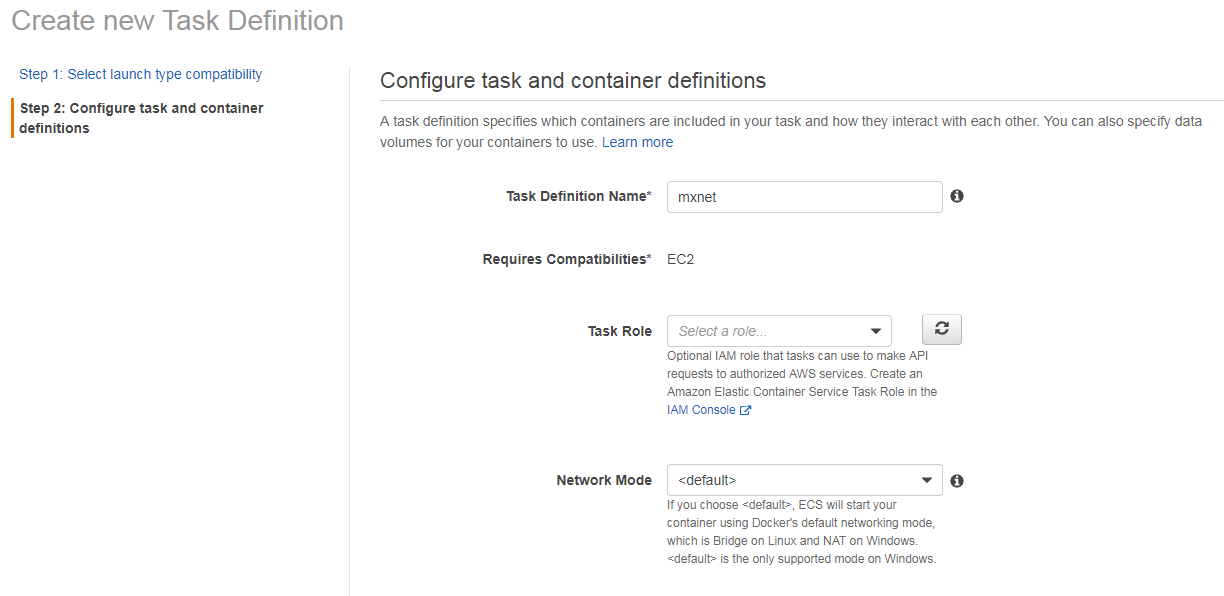
* **Task 3:** Deploy the MXNet Container with ECS
  + In the EC2 Container Service dashboard, click on Task Definitions and click Create new Task Definition



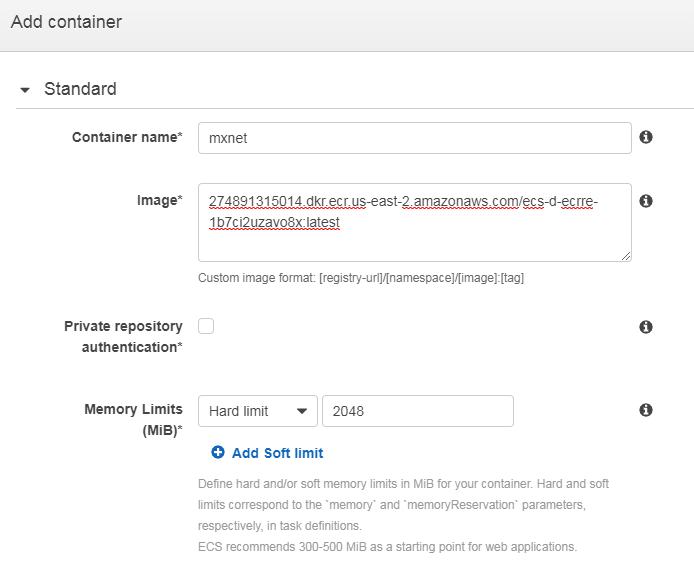
* + Choose EC2 as shown below

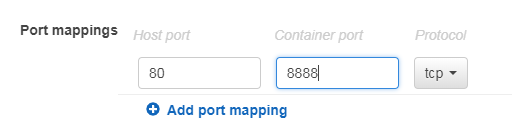


* + Provide name to the Task Definition



* + Click on Add Container and provide values as shown below and save it. Provide the image id from the ECR repository

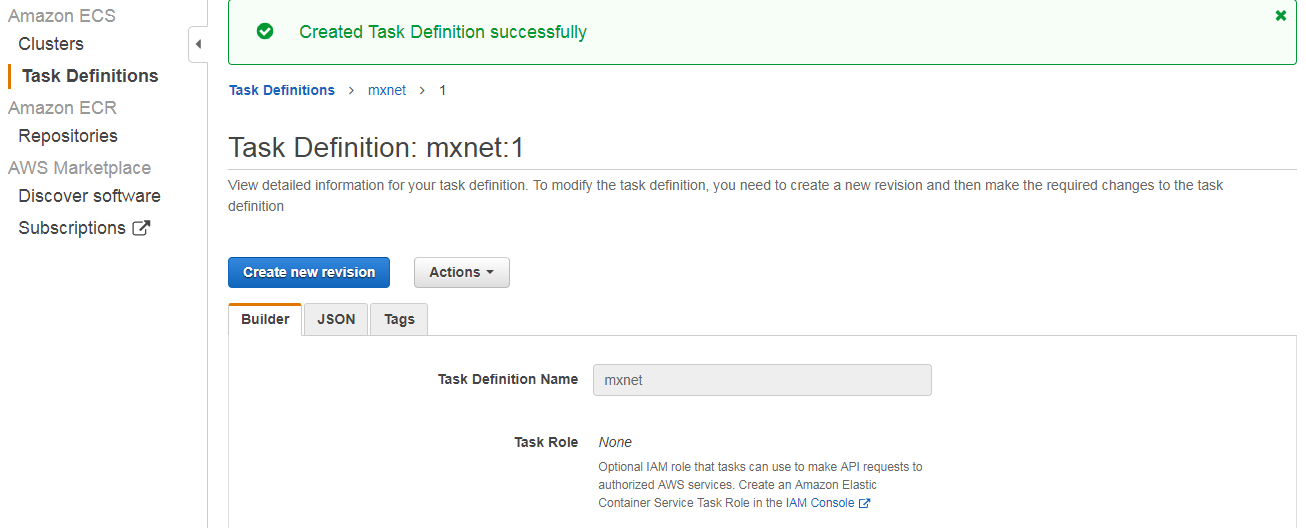




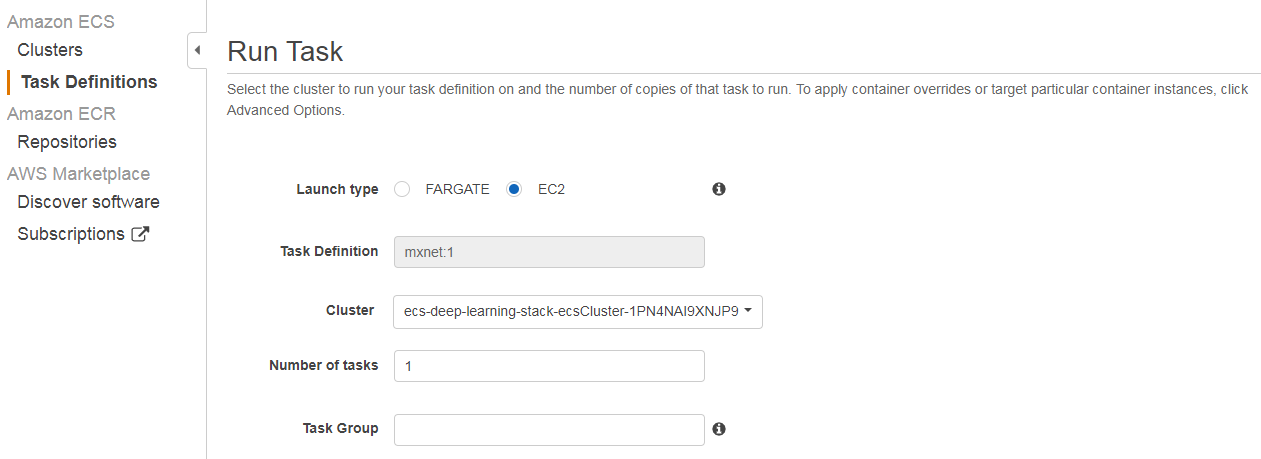
* + Click Create



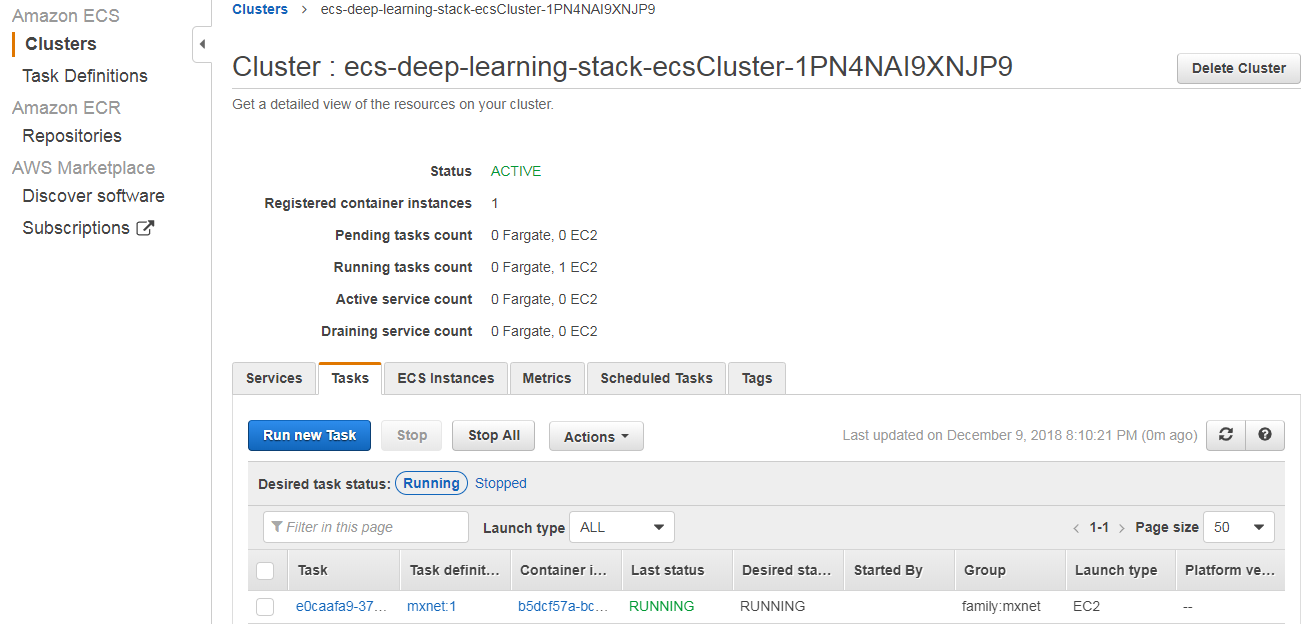
* + Click on Actions->Run Task



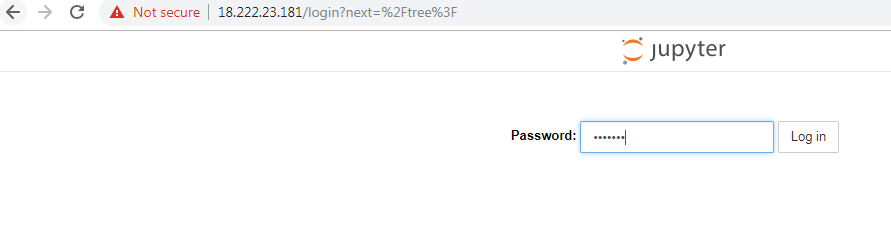
* + Provide values for the Run Task as below



* + Check for the task running in the EC2 instance



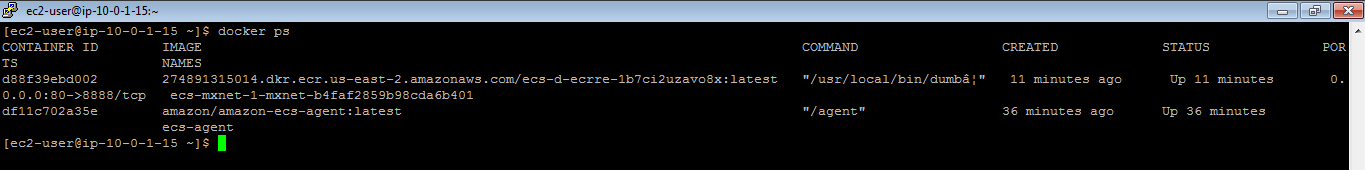
* + On a browser tab and load the public DNS of EC2 to test Jupyter loads properly



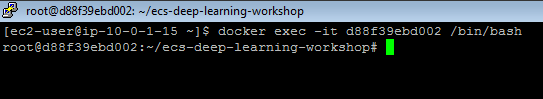
* + You should be prompted for the password you passed in earlier as a build-arg. Enter the password and you should be able to log in



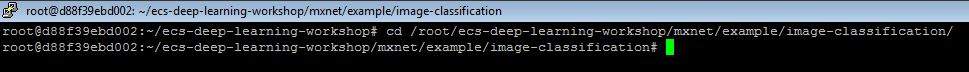
* **Task 4:** Image Classification with MXNet
  + SSH into the instance. Find the container to connect to by running the below command



* + You'll find two containers that are running. One for our mxnet container, and one for the amazon-ecs-agent
  + Note down the CONTAINER\_ID of the mxnet image so we can open a bash shell like this:



* + Navigate to /root/ecs-deep-learning-workshop/mxnet/example/image-classification/

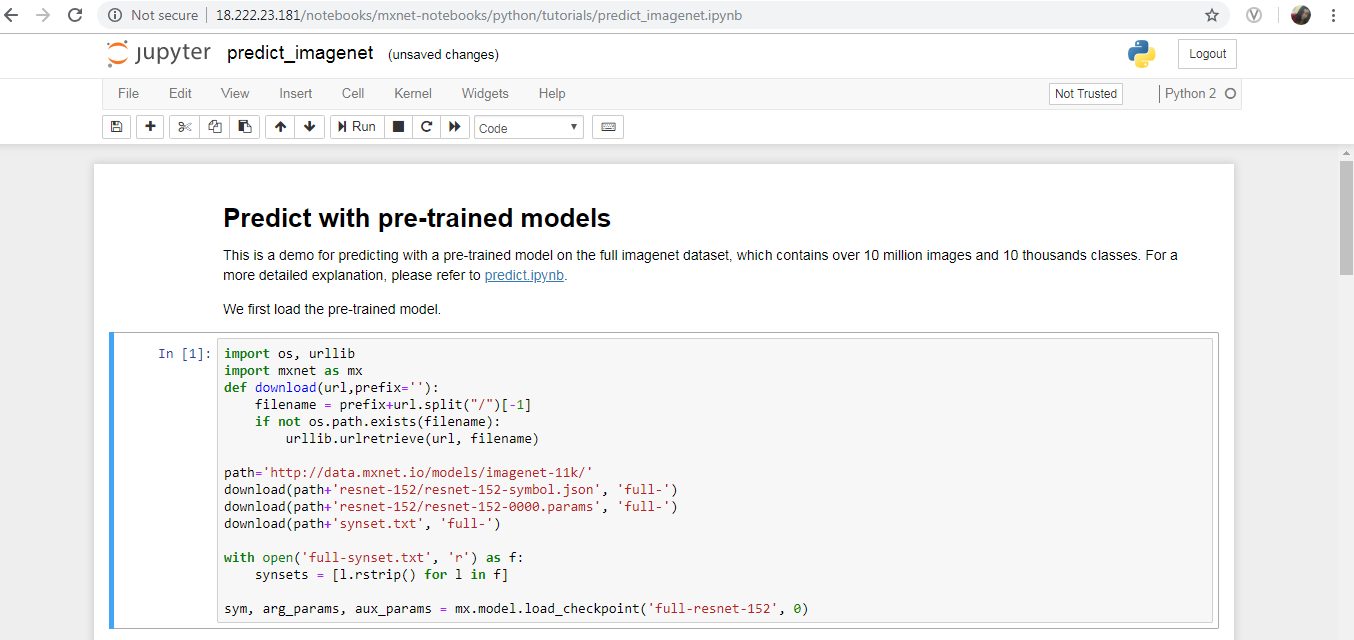


* + Run train\_mnist.py

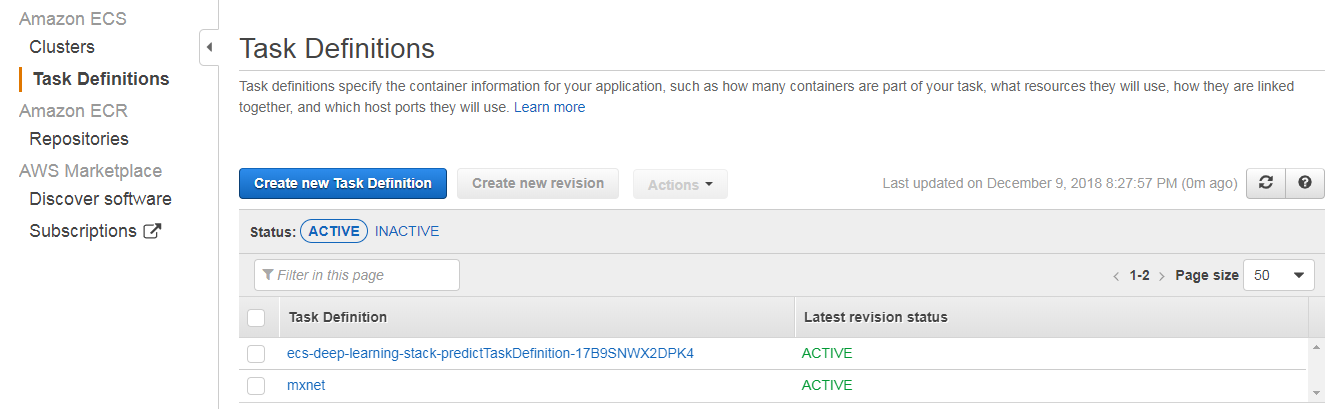


* + Open a web browser and visit this URL to access the Jupyter notebook for the demo:

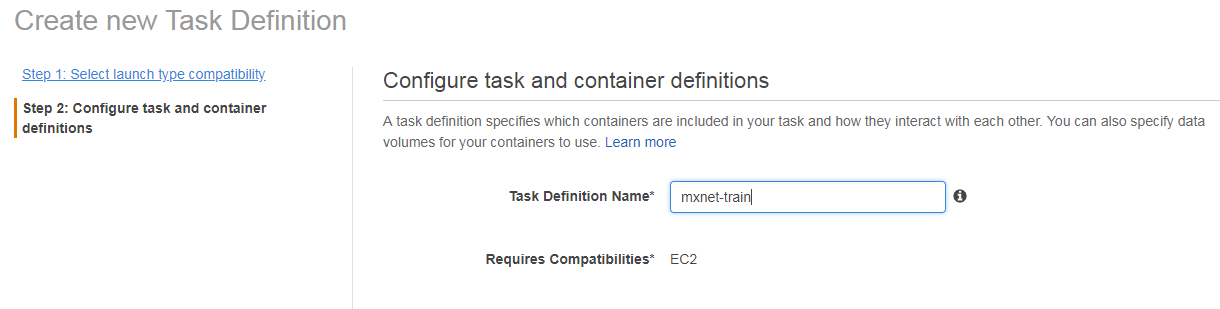
**http://EC2\_PUBLIC\_DNS\_NAME/notebooks/mxnet-notebooks/python/tutorials/predict\_imagenet.ipynb**



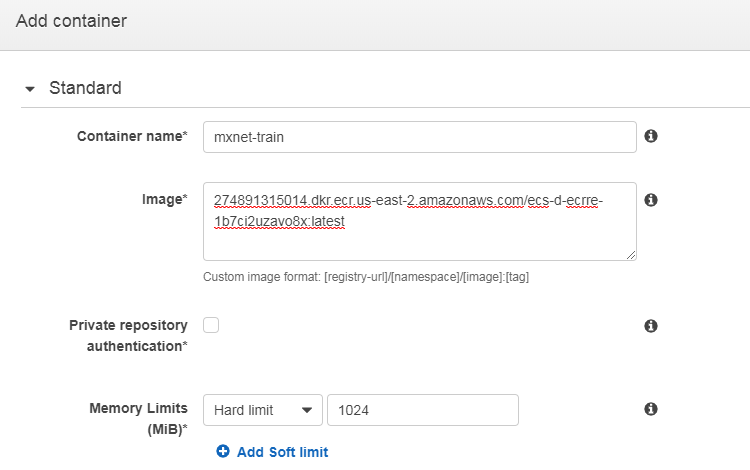
* **Task 5:** Wrap Image Classfication in an ECS Task
  + On EC2 Container Service dashboard, click on Task Definitions and click Create new Task Definition



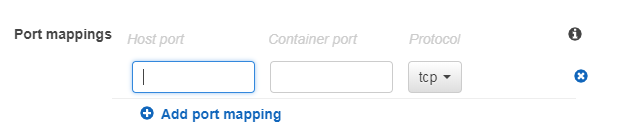
* + Name your task definition, e.g. "mxnet-train"



* + Click on Add Container and provide values as mentioned below. For image URI, pick the same one available in the ECR repository



* + Leave the port mapping section blank



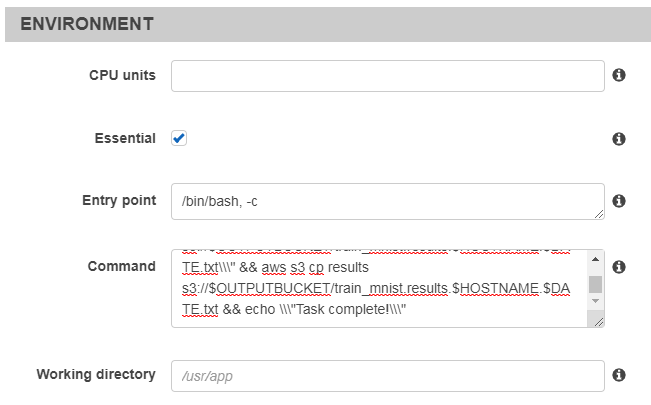
* + Go to the Environment section and provide the values as below

**Entry point:**

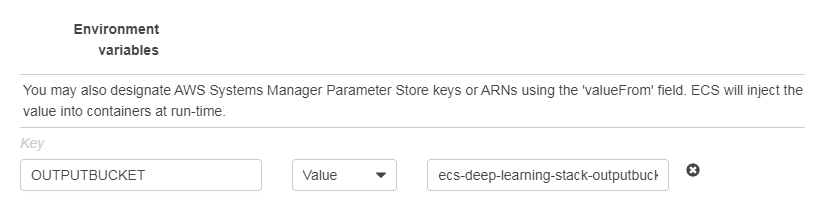
/bin/bash, -c

**Command:**

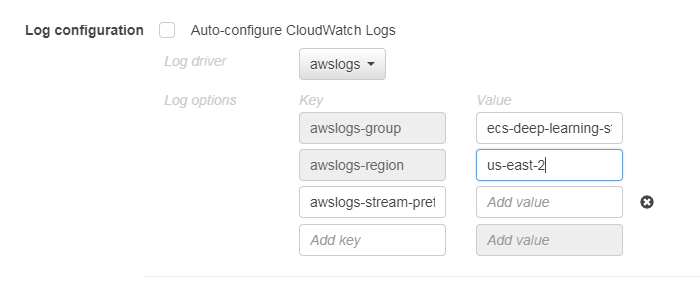
DATE=`date -Iseconds` && echo \\\"running train\_mnist.py\\\" && cd /root/ecs-deep-learning-workshop/mxnet/example/image-classification/ && python train\_mnist.py --lr-factor 1|& tee results && echo \\\"results being written to s3://$OUTPUTBUCKET/train\_mnist.results.$HOSTNAME.$DATE.txt\\\" && aws s3 cp results s3://$OUTPUTBUCKET/train\_mnist.results.$HOSTNAME.$DATE.txt && echo \\\"Task complete!\\\"



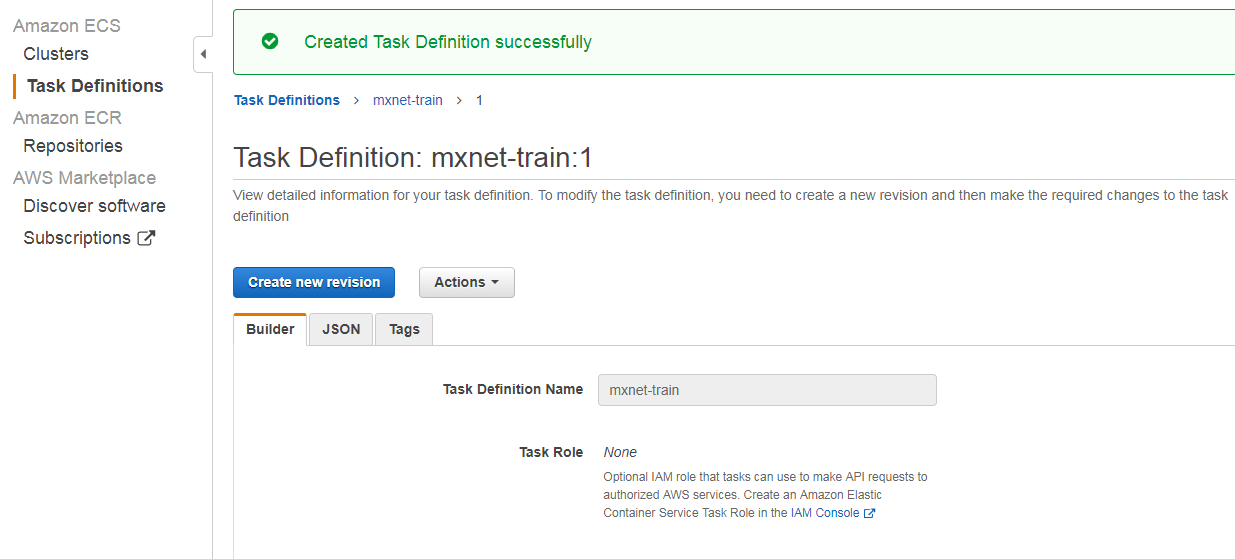
* + For the Env variables, set the key to be "OUTPUTBUCKET" and the value to be the S3 output bucket created by CloudFormation



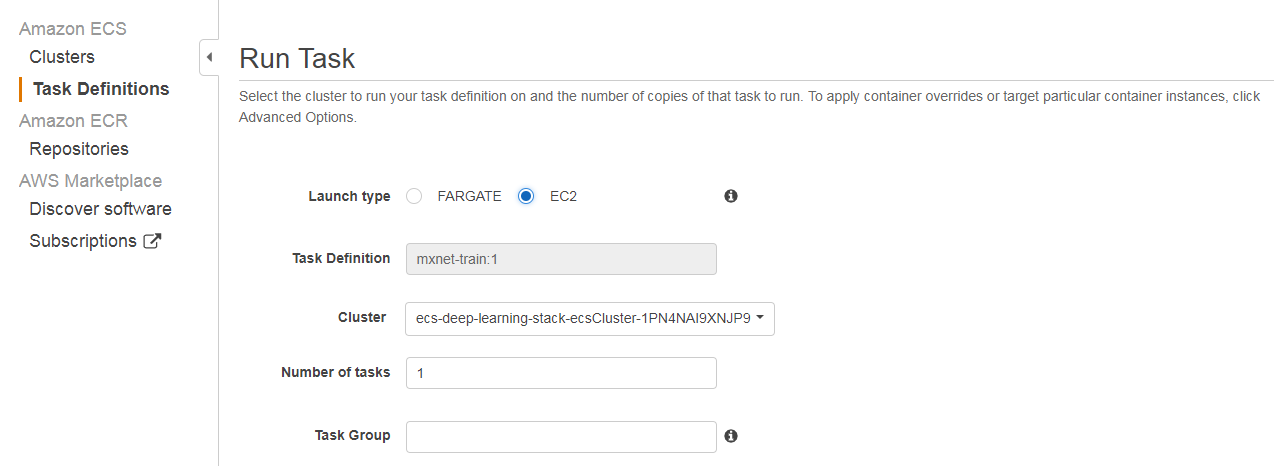
* + Scroll down to the Log configuration, select awslogs. For Log options, set the awslogs-group to be the value of cloudWatchLogsGroupName from the CloudFormation stack outputs tab
  + Type in the region you're currently using, e.g. Ohio would be us-east-2, Oregon would be us-west-2
  + Leave the awslogs-stream-prefix blank



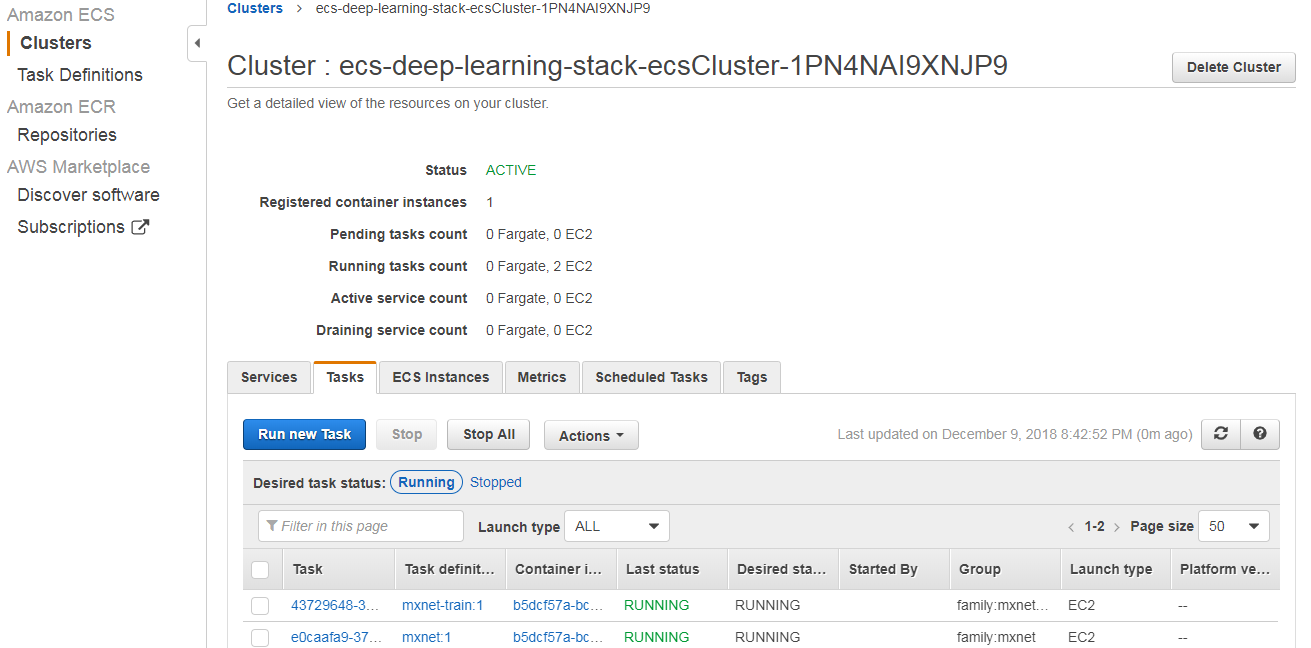
* + Click Add to save this configuration and add it to the task defintion. Click Create to complete the task definition creation step.

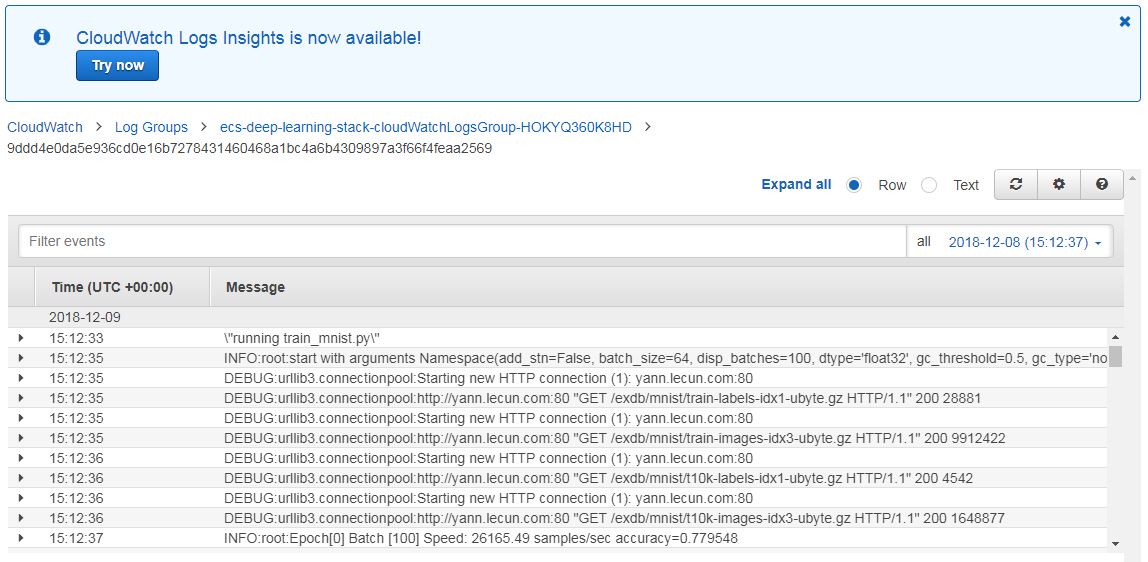


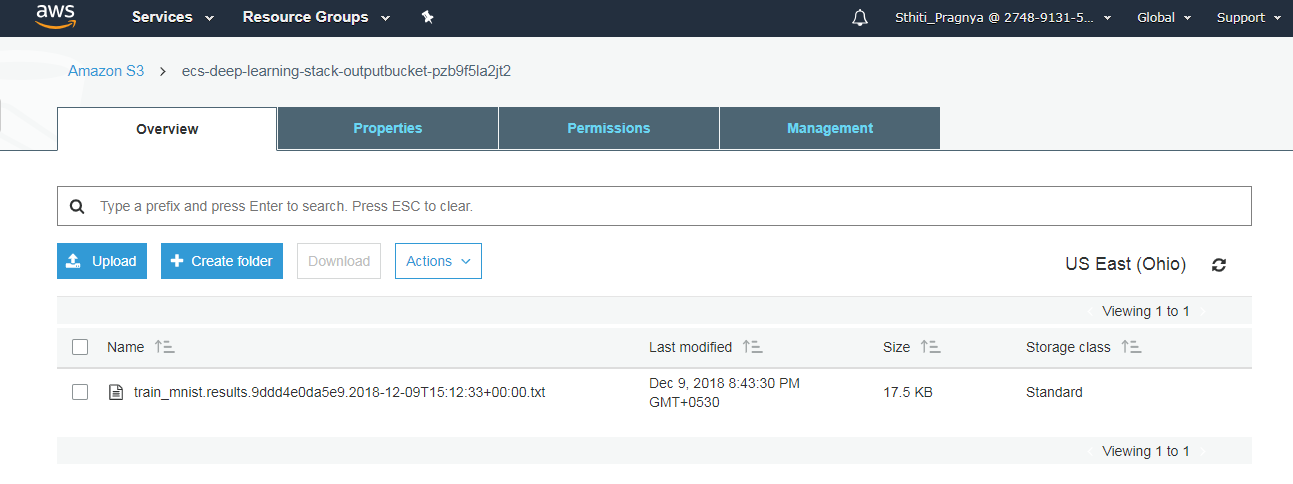
* + Go to Actions-> Run Task and provide values as below



* + Check for the task running in the EC2 instance



* + The task outputs logs to CloudWatch Logs as well as S3.
  + Open the CloudWatch dashboard, and click on Logs. Click on the log group, and then click on the log stream that was created. 
  + Once the task has completed and stopped, check your S3 output bucket, and you should see a log file has been written. Download the log file and check the content.



**Summary:** In this workshop, you interact directly with the MXNet containers using SSH or Jupyter notebooks.