# 

# RoboCar Rally Instructions

March 2019

# **Start your engines!**

You have been selected to be a vital member in building a race team that will help grow a global race league. As a team, you must build the fastest autonomous vehicle to complete timed laps without going out of bounds (cutting inside the track).

# **Your Pit Crew:**

Your team of 5 should define roles and responsibilities. These roles will allow you to complete many of the steps in parallel.

* **Crew Chief** - Motivates team members to ensure they stay focused doing their job and removes any blockers by interfacing with the event facilitators and observers.
* **AWS Architect** - Has access to the AWS console to ensure services such as IAM, S3, and SageMaker are implemented and running properly.
* **Master Mechanics**
  + Hardware Engineer: Keeps track of troubleshooting Robocar issues such as wrong cabling, servo failures, battery life, etc. It is very important to ensure that Robocar batteries are constantly charged.
  + Software Engineer: Manages the Robocar directories that were flashed on the SD card. Understands each directory and becomes familiar with the structure and intent of the files. Handles model deployment and script execution.
* **Stunt Driver** - Controls the RoboCar while training the model, ensures clean driving data is collected, and is responsible for operating the RoboCar on the track.

# **Goal:**

To field a competitive race car, you will need to complete these 3 high-level steps:

### **Collect Training Data**

Utilize the Robocar Software to collect training data. This training data will be stored on the local Raspberry Pi and needs to be transferred to S3.

### **Train a Model with SageMaker**

These instructions walk you through training on the notebook itself which is okay for smaller datasets and development environments. As a follow-up after the RoboCar Rally event, the race facilitators can provide you documentation on how to train on SageMaker for larger datasets in a production environment. Production training instructions include creating an image consisting of a custom Docker container and storing it in ECR. This image is then referenced by the training instance cluster that runs.

### **Deploy a Model with SageMaker**

Copy the model trained by SageMaker back to the vehicle and have the vehicle drive autonomously around the track 3 times as fast as possible without going out of bounds.

# **Important Notes:**

## **ABC Always Be Charging (Say this out loud as team!)**

The green battery that powers the motor of the car is known to run down quickly and can take hours to charge. Therefore, it is essential to keep the vehicle charging while it is not in use. Note: the only time you would turn the RC Robocar on would be during initial calibration and for the driver to quickly practice. For the remainder of the hackathon, you should only turn it on when you are ready to collect training data or testing a model. Otherwise, turn it off and charge!

## **Work in Parallel**

Focus on delivering the results required of your role. This is a fast workshop so you will need to complete your assigned role tasks in order to keep your team progressing. Most of the work that can be done in parallel can be accomplished with multiple SSH sessions to the vehicle.

## **Copy/Paste Code Errors**

Avoid the mistake of doing a copy/paste of code without modifying variables.

## **Quality Training Data**

Drive slowly to capture a smooth and accurate driving style. It is OK to go near the edges of the lanes, but avoid going over the lanes, or on the wrong side of the cones. If you do, we will provide you with a tool to clean your training data by removing those images.

Start Your Engines!

## **Step 1: Getting Started (Entire Pit Crew)**

The RoboCar is fully built and is ready to start driving and collect training data. But before you do that, you should familiarize yourself with the Donkey Code Base. This will help your teams move faster when they encounter challenges.

Read the Get Driving section of the documentation <http://docs.donkeycar.com/guide/get_driving/>

Read about the different utilities built into the Donkey Project

<http://docs.donkeycar.com/utility/donkey/>

## **Step 2: Connect to and Calibrate the vehicle (Master Mechanic, AWS Architect)**

Power on the Raspberry Pi using the USB attached battery. It will automatically connect to the guest wifi. Utilize your favorite SSH client, such as Putty, to connect to your vehicle IP address. Your IP address can be found on the whiteboard. Use these credentials:

User: pi

Password: asdfasdf

Using credentials from your AWS account, [configure the AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html#cli-quick-configuration) with an AWS IAM and access key and secret access key on the Raspberry Pi.

*Note: Use us-east-1 (N. Virginia) as the default region.*

To calibrate your car’s throttle and steering, follow these instructions: <http://docs.donkeycar.com/guide/calibrate/>

*Note: Reversing is out of scope for the Hackathon and does not need to be configured.*

## **Step 3: Start the vehicle control webserver (Master Mechanic, Stunt Driver)**

Issue the following commands to start the webserver on the Raspberry Pi. The webserver provides the interface to manual control the vehicle, as well as to start training data recording sessions. The webserver will listen on port 8887 and will be the main form of control over the vehicle.

cd ~/d2

python manage.py drive

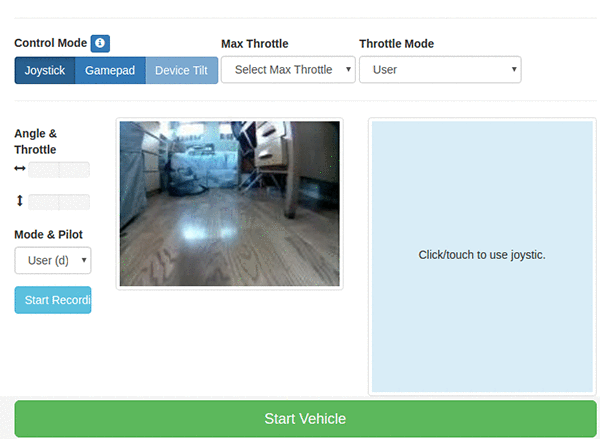
## **Step 4: Start driving (Stunt Driver)**

Using your tablet or mobile phone, connect to the IP address and webserver port (8887) to view the vehicle control page. For example, enter [172.20.10.3:8887](http://172.20.10.3:8887). Plug in the green NI-MH battery to the Electronic Speed Controller (ESC) of the vehicle. Slide the power switch to on and you should hear the car come alive (beep).

Start with setting the Max Throttle to 30% as these vehicles are very fast and will not be controllable to generate quality training data for new drivers. Using the tilt functionality on a mobile device will move/turn the vehicle in a corresponding direction. This will provide the car with steering and throttle input. This input combined with the image from the front facing camera forms the labeled training data.

Click Start Vehicle to gain control over the vehicle.

Practice driving to get the feel for the vehicle. When you have a handle on the controls then head over to the race track to collect clean training data. Take turns with other teams so there is one Robocar on recording data on the track at a time to prevent unnecessary noise.



## **Step 5: Data collection via driving (Master Mechanic and Stunt Driver)**

The car is programmed to collect training data that will later be used to create a model based on your driving. (Imitation Learning) To start the collection process, click Start Recording. As you apply throttle to the vehicle it the program will grab a 160x120 frame from the camera, and the values of the throttle and steering at that same moment.

* + JSON metadata 127 bytes
    - {"cam/image\_array": "9216\_cam-image\_array\_.jpg", "user/angle": 0.11428571428571437, "user/throttle": 0.45, "user/mode": "user"}
  + Image 160x120 5KB



## **Step 6: Mastering the Art of Driving (Stunt Driver)**

Good results will be achieved by slow consistent and smooth turn that keep the vehicle on course. Avoid the urge to start off driving fast while collecting training data in order to avoid teaching the model bad and risky driving behaviors. [If needed, go back and recalibrate if you are having trouble keeping the car driving straight.](http://docs.donkeycar.com/guide/calibrate/)

Watch this video to see the proper training technique:

<https://s3.amazonaws.com/robocarrallyevent/driving101.MOV>

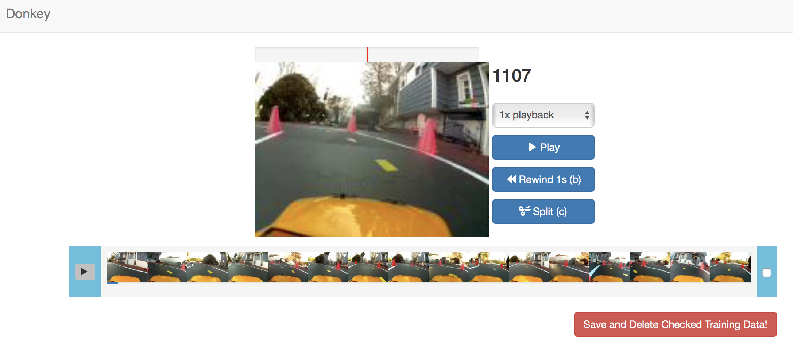
## **Step 7: Validate and Clean Training Data Prior to Training (Master Mechanic, Stunt Driver)**

The Donkey Car project will collect training data locally to the vehicle. This training data may not be the cleanest dataset, so we recommend first going through the data using a built-in tool called tubclean to pick out and drop the images after each data collection session.

Your driving may not have been perfect, and could possibly benefit from some “light” data preparation in order to eliminate crashes, bad turns, white shoe soles, etc. To do that, the Donkey software provides a means of “cleaning the tub” (tub is the term used for data collected during a recorded training session).

cd ~/d2

donkey tubclean ./data



A web server will deploy and be listening on your car’s IP address and port 8886. You can interact with it by heading to a browser on your laptop and choosing the tub of test data you would like to clean.

From the Robocar console, you can enter ‘Ctrl + c’ to exit when you are finished removing the bad frames from the training data. Once you have cleaned all your training data, you can move on to transferring it to S3 to make it readily available to create a model.

## **Step 8: Upload Training Data to S3 (Master Mechanic)**

In this step you will be using the AWS CLI to upload the data from stored in the tub directory from your recorded driving to an S3 bucket. Make sure you update the path relative to your vehicle and S3 bucket name.

Create the S3 bucket:

aws s3 mb s3://bucket\_name

Zip local data and copy to S3 bucket

cd ~/d2/data/

tar -cvzf *tub\_name*.tar.gz ~/d2/data/*tub\_name*

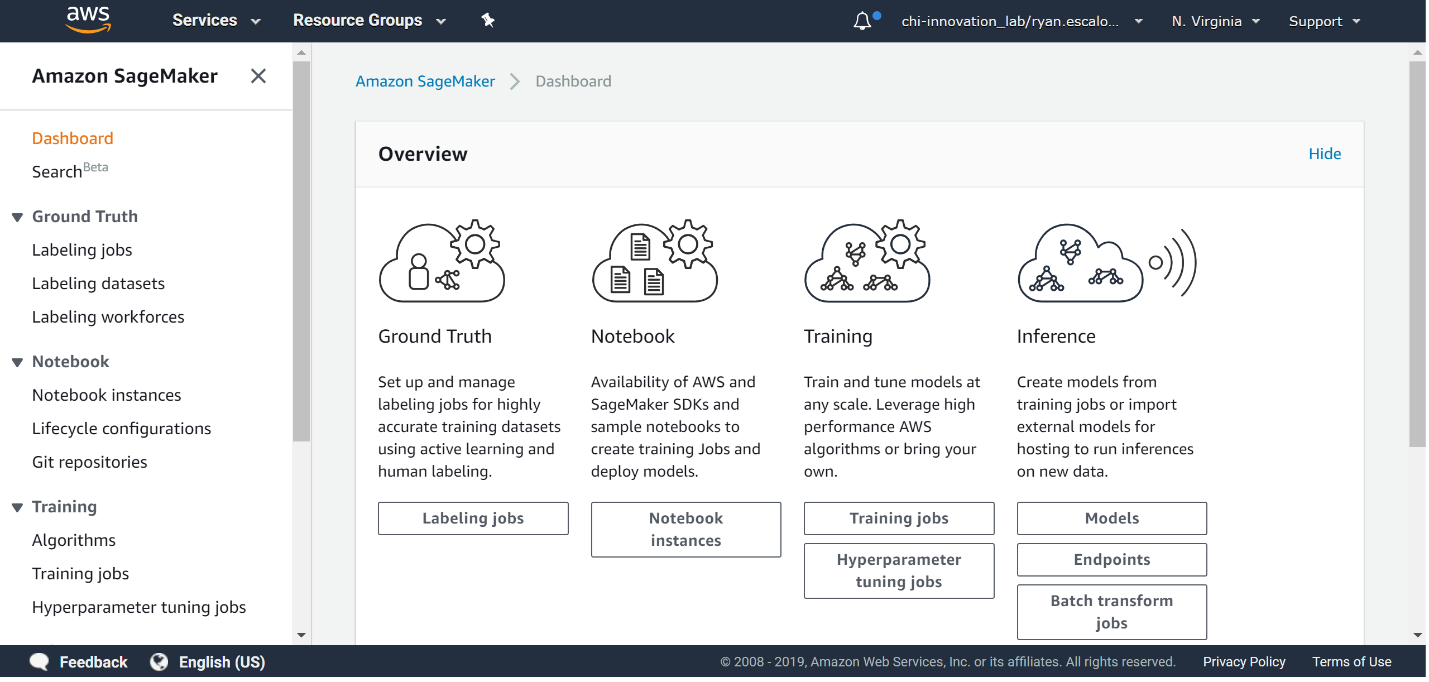
Copy zipped data from the vehicle to the new S3 Bucket

aws s3 cp ~/d2/data/*tub\_name*.tar.gz s3://*bucket\_name*/*tub\_name*.tar.gz

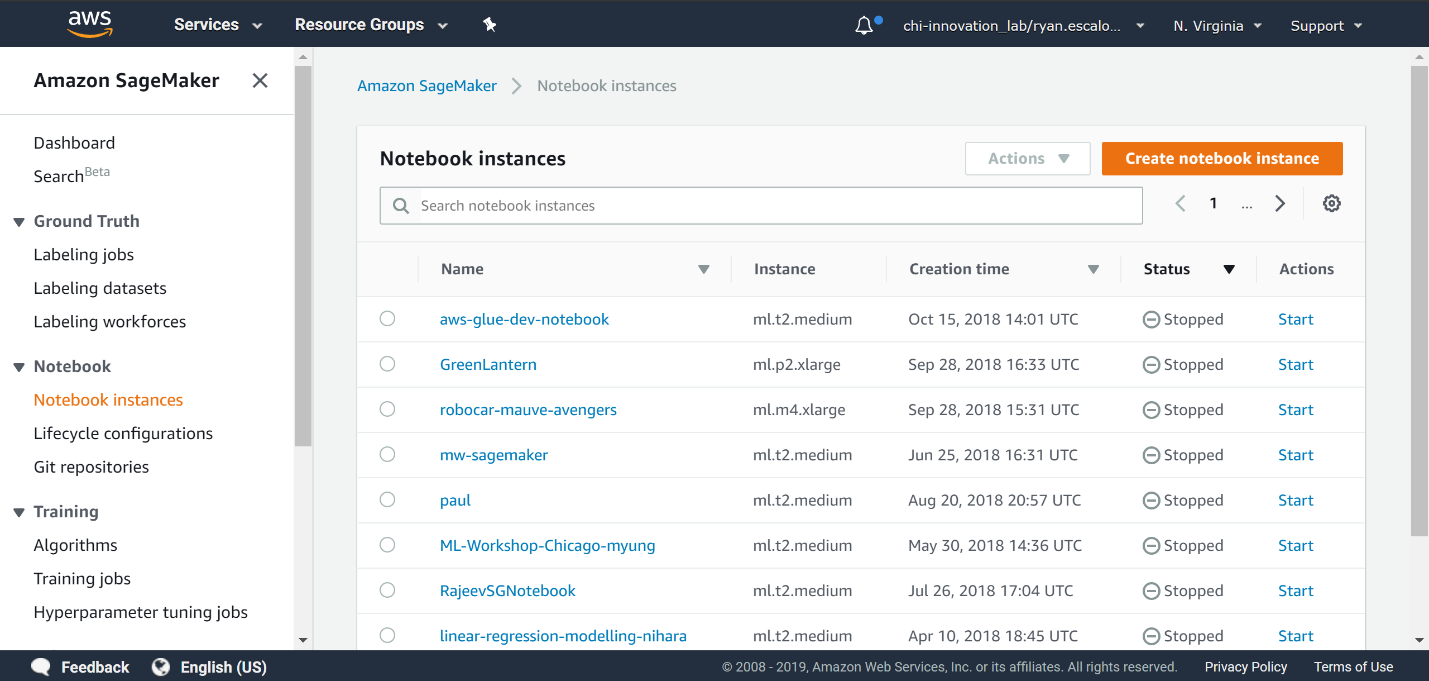
## **Step 9: Create a Jupyter Notebook with SageMaker (AWS Architect)**

Open SageMaker from the AWS Console.

Click “Notebook instances” from the SageMaker dashboard:



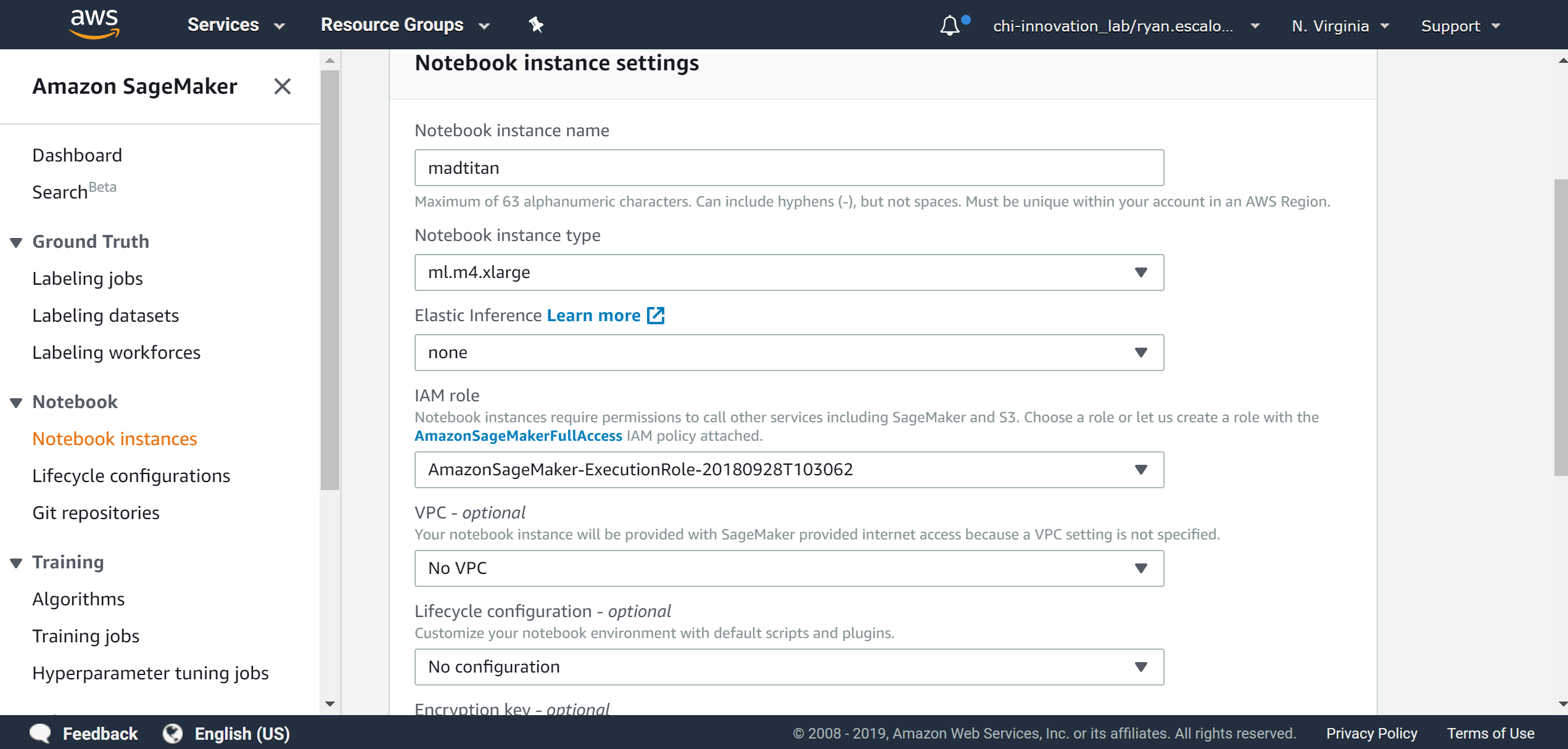
Then click “Create notebook instance”



Give your notebook instance a name, and select an instance type appropriate to the size of the training data. For our use case, since we are training on notebook and have a rather small set of training data, we could use a General Purpose based notebook instance type ml.t3.2xlarge to train the model.

Ensure the AmazonSageMaker-ExecutionRole you are using has access to your S3 bucket in the same region. Create a new IAM role if necessary.

*Note: While we are using a Standard CPU-based instance for our Jupyter notebook development right now, in a production environment we would use an Accelerated GPU-based p3 instance for our model training.*

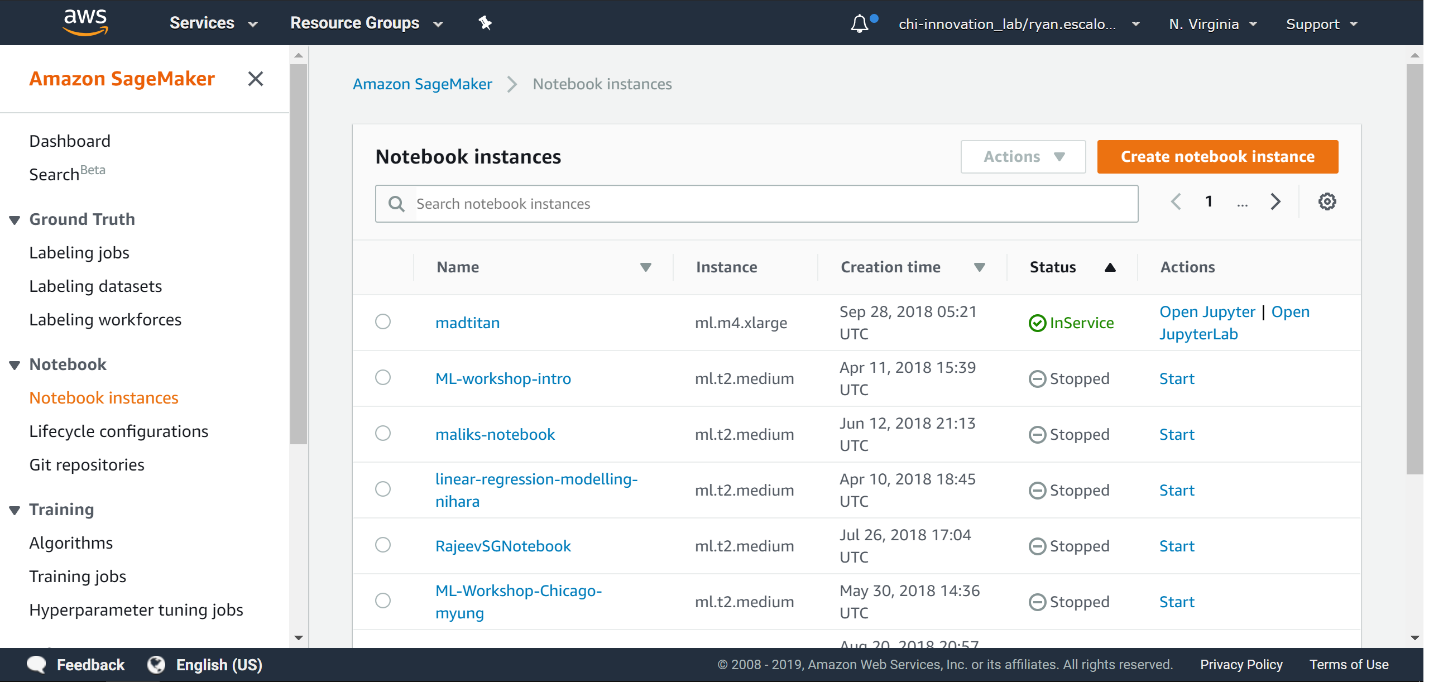


Once you click create, AWS will spin up your notebook for you. This one-time process will take a few moments.

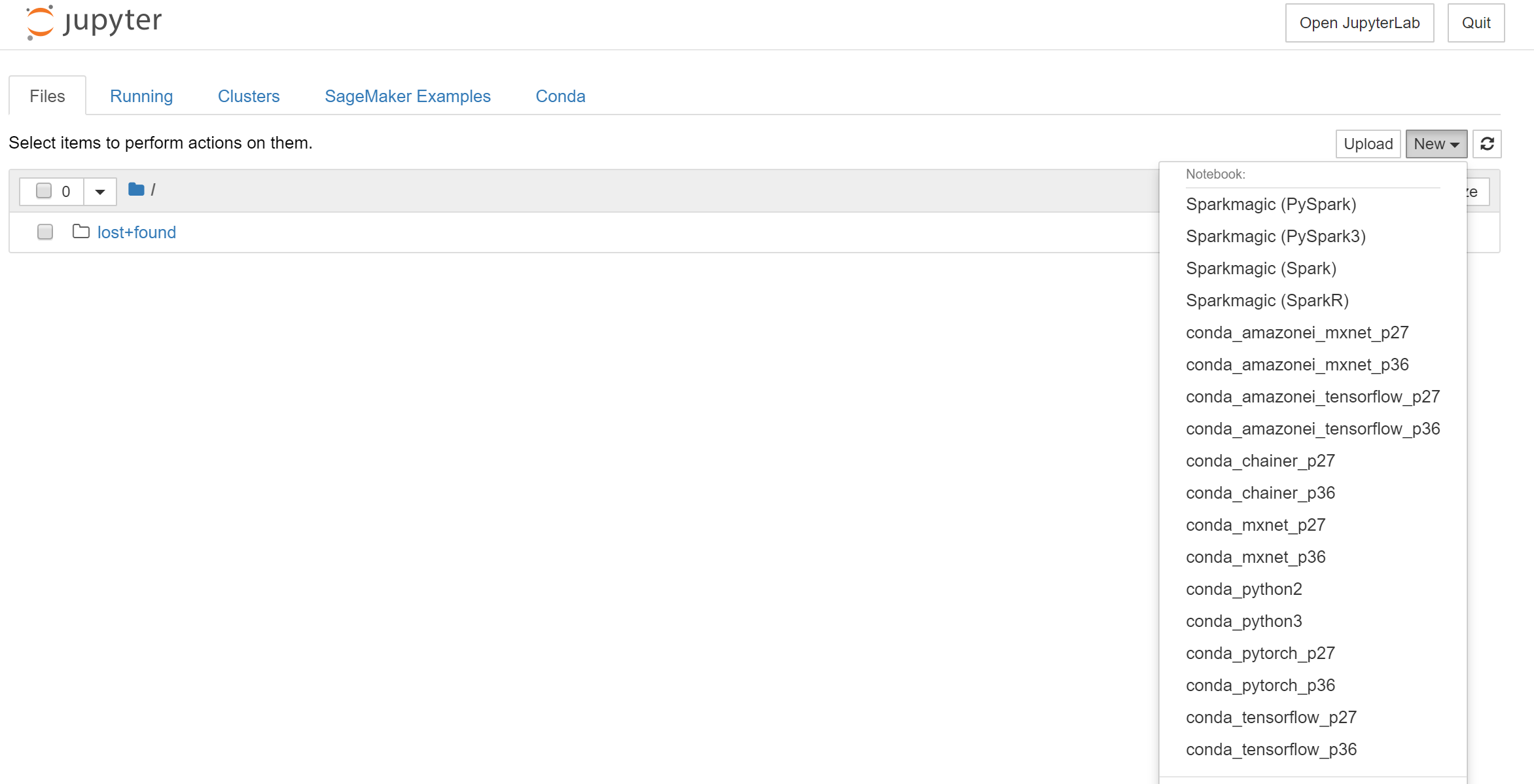
## **Step 10: Jupyter Notebook and Terminal Configuration (AWS Architect)**

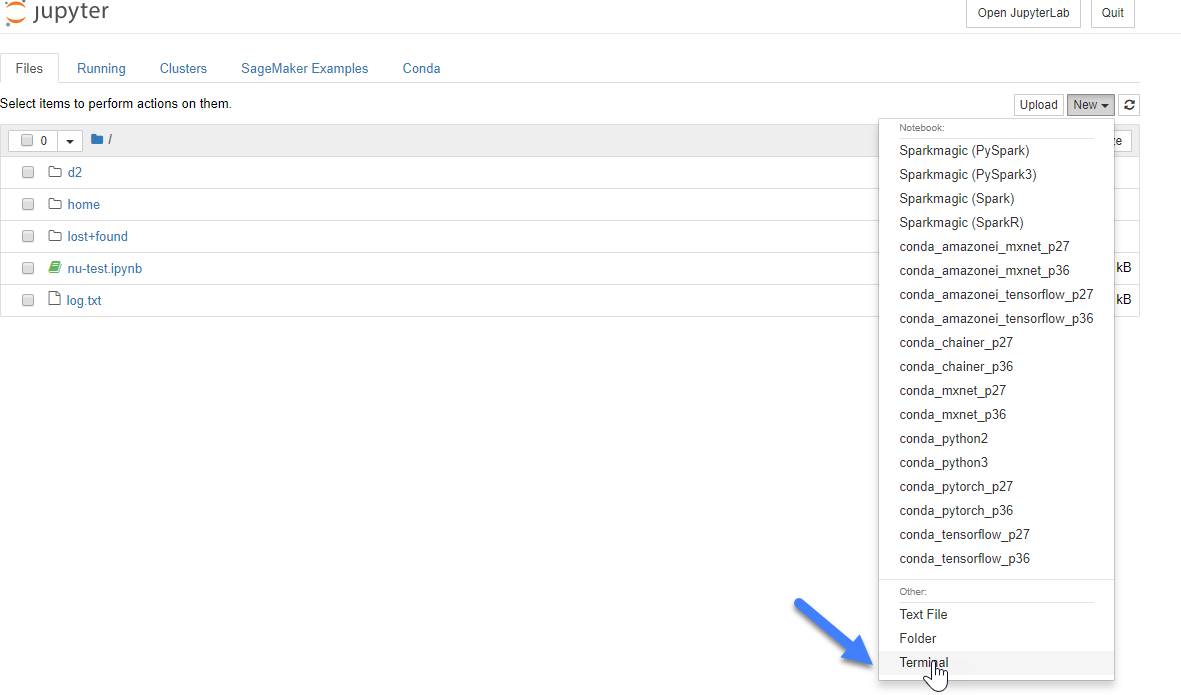
You will now setup the environment, execute a few cells in the notebook, and then run the train.py script to create the model. *Note: You will be jumping back and forth from the terminal and Notebook as part of a learning experience to get hands on with both.*

Once the notebook status says “InService” click “Open Jupyter” to jump directly into your hosted Jupyter notebook.



**Step 10a:** From your Jupyter notebook instance, select the ‘New’ dropdown and choose ‘Terminal’





From the terminal prompt you will run the following 4 commands:

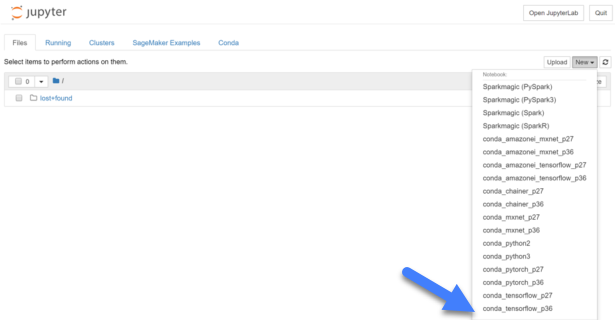
sudo usermod -G root ec2-user

id ec2-user

cd ~/SageMaker/donkey

pip install donkeycar==2.2.4

**Step 10b:** Switch back to your Jupyter notebook instance tab. Select the ‘New’ dropdown and choose ‘conda\_tensorflow\_p36’



Copy and run each cell individually and in order from the following notebook on github *(Read each cell carefully to ensure you are replacing the inputs where necessary, such as S3 bucket name)*:

<https://nbviewer.jupyter.org/github/tescal2/donkey/blob/master/University%20Training.ipynb>

**Step 10c:** Now you will train the data from the Notebook instance terminal and output your trained model in models directory. Before you train, however, you need to update the config.py file with your calibration settings that you set on the Robocar.

cd /home/ec2-user/SageMaker/d2

vi config.py

Next, proceed with training by replacing your tub name and giving your model a name. Run the following as a SINGLE command:

python /home/ec2-user/SageMaker/d2/manage.py train --tub home/pi/d2/data/tub\_name/ --model /home/ec2-user/SageMaker/d2/models/model\_name

**Step 10d:** Copy the model from Notebook instance, using the terminal, to S3. Ensure you update the model name below to choose your model.

aws s3 cp d2/models/nu-test s3://your\_bucket\_name/model\_name

*Note: Once these steps are complete, you will have a trained model in S3 ready to be pulled down onto the Robocar.*

## **Step 11: Load and Drive the Model on the Robocar (Master Mechanic)**

SSH back into your Robocar and head over to the following directory:

cd ~/d2/models

Download and extract the model that you just trained onto the Robocar by executing the following commands and updating it with your model path:

aws s3 cp s3://*path\_to\_your\_model*/model .

Note: If you are going to train another model later on SageMaker, then ensure that you rename your current car-model.pkl file for version control because the next model that you unzip will have the same name and overwrite it in the directory.

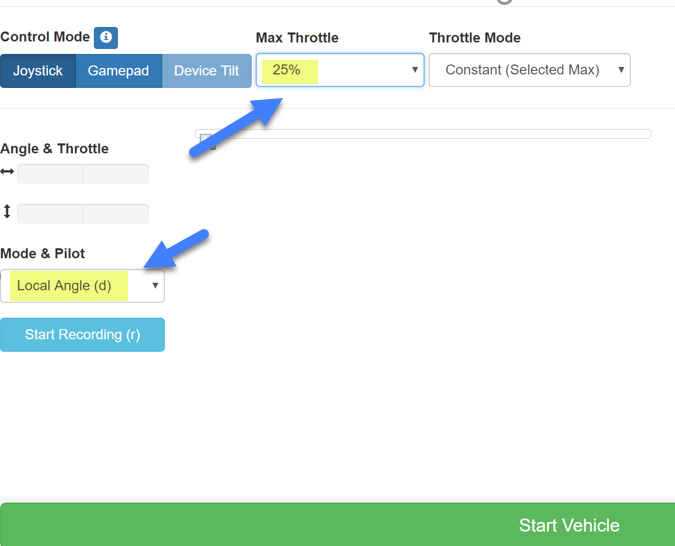
Now you can power on your Robocar again, change back to the d2 directory and pass your model to python drive script using the path to your model.

cd ..

python manage.py drive --model ~/d2/models/*car-model.pkl*

## **Step 12: Test the model (Stunt Driver)**

Return the vehicle to the track. From Mode & Pilot on the Web Controller, make sure to choose Local Angle mode and from Throttle mode select Constant, and from Max Throttle start with a low value like 25%. Local Angle Mode will instruct the model that it is responsible for predicting steering angles. Start Vehicle, hold the letter “i” down on the keyboard for a few seconds, and it should start driving autonomously using your newly trained SageMaker model. Adjust the Max Throttle to represent a speed that keeps the vehicle safely on the course. If you would prefer to manually control the throttle, you may do so by selecting User from the Throttle mode selection.



## **Step 16: Iterate and Win!**

Continue collecting quality training data to create new models. Run through the steps again expect for step 10a as your notebook instance is all setup. Test, get ready to compete with your team’s best model and most importantly --- HAVE FUN!