* Find the max curvature of track. This should dictate steering steps – shouldn’t exceed that
* Aggressive driving
* Each EC2 can run one model. We can speed up raining by adding more workers

Get started:

* Go/deepracerspot
* <https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-get-started.html>

|  |  |
| --- | --- |
| **One time setup for a sandbox** | |
| **SSO + RSA deep racer login** | <https://idag2.jpmorganchase.com/adfs/ls/idpinitiatedsignon.aspx?logintorp=ASB> |
| **cloning repo** | open AWS CloudShell  clone the repo: git clone <https://github.com/EsaLaine/deepracer-templates>  navigate into directory: cd deepracer-templates  see the files: ls |
| **create base resources** | ./create-base-resources.sh BASE-STACK-NAME YOUR-IP  **BASE-STACK-NAME**: name of this cloudformation stack you’re creating- could be **your name-base**  **YOUR-IP:** IPV4 of your machine you are using to access EC2 instance (<https://whatismyip.host> |
| **Create image builder** | ./create-image-builder.sh BASE-STACK-NAME IMAGE-STACK-NAME  **BASE-STACK-NAME:** same name you selected in the previous section. If you forgot the name, you can check in CloudFormation for stacks that have been created before  **IMAGE-STACK-NAME:** name of this cloudformation stack (recommended to be "your name-image". i.e. “tyler-image")  Run Pipeline: after this CloudFormation script completes, navigate to EC2 Image Builder> Image Pipelines > select your pipeline > hit Actions > run pipeline |
| **Steps applicable for each model training** | |
| **Update custom files** | The repo we cloned above in Cloud Shell has a sub directory “custom-files” which has configuration files that is used to customize your training. You will need to update these before you can start training your model.   * From the deepracer-templates directory, navigate to the custom-files: cd custom-files   In here, you will see 5 files that will be uploaded to the EC2 instance that control your training parameters   * custom-files/hyperparameters.json: Defines hyperparameters you can adjust for training your models such as learning rate, batch size, epochs, etc. * custom-files/model\_metadata.json: Defines model sensors, action space, training algorithm, and action space type (discrete/continuous).   custom-files/reward\_function.py: Python reward function you want your model to use to train  • See [Input Parameters of the AWS DeepRacer Reward Function](https://docs.aws.amazon.com/deepracer/latest/developerguide/deepracer-reward-function-input.html)  For below 2 files, please see [Deepracer-for-Cloud Reference](https://aws-deepracer-community.github.io/deepracer-for-cloud/reference.html) for documentation on variables used   * custom-files/run.env: Parameters for this model's training. You can set the race type (time trail,object avoidance, head to head), track, model name, and much much more.   + If you are training a new model, you will want to change the model's name each time. Below are some key variables you will need to customize   DR WORLD\_NAME=2022 march\_open  DR\_LOCAL\_S3\_MODEL\_PREFIX=<give a unique name for each new model you train>   * custom-files/system.env: Parameters for drfc training. You can define the sagemaker/robomaker/coach image versions to use, number of workers, etc. |
| **Create EC2 (Standard or Spot) instance** | ./create-standard-instance.sh BASE-STACK-NAME TRAINING-STACK-NAME TIME-TO-LIVE  -OR  ./create-spot-instance.sh BASE-STACK-NAME TRAINING-STACK-NAME TIME-TO-LIVE  **BASE-STACK-NAME:** same name you selected in the previous section. If you forgot the name, you can check in CloudFormation for stacks that have been created before  **TRAINING-STACK-NAME:** name you want this training to be called. (recommended to be "your name first-training. i.e. "tyler-first-training")  **TIME-TO-LIVE** number of minutes to keep the instance running before it is automatically terminated  CONGRATS! you trained your first machine learning model on an ec2 instance. This will start training on a EC2 instance which has cost savings over training in console. Training on spot instances is cheapest, even cheaper than standard instances.  You will be able to see your running instance in AWS Console > EC2 Instances. Be sure to stop/terminate the instance when you are done training. You can also get shell access to your EC2 instance by selecting your instance and using "Connect" button.  PS: Sometimes we have observed that instances do not start properly. In such cases you can go on EC2 Shell and manually stop/start training. See Deepracer-for-Cloud Reference for settings and commands that you can use on EC2 Shell command line.  About Spot Instances:   * Benefits of spot training: https://aws.amazon.com/ec2/spot/ * When you run this command, if there is not enough spot availability you may receive an error. If this does happen, try running the command after work hours (or weekends). If it still does not work you may have to fall back to using a standard instance. |
| **Viewing model train on track** | Once EC2 instance is created, in cloud shell console your EC2 instance public IP address is printed which can be used to view your mode's training video   * Paste the instance's ip, port 8080 into your browser and wait 5-8 minutes after running "create... instance.sh to see the car training on the track http://125.125.125.125:8080 * To reduce costs of watching the stream of you car, you can specify the quality and size of the window at the end of the url:   http://125.125.125.125.8080/stream\_viewer?  topic/racecar/deepracer/kvs\_stream&quality=10&width=400&height-300 |
| **Stopping training** | If you do not terminate/stop your instance, the instance will continue to incur charge per hour that it runs for irrespective of if you are continue to run training or training error out to due to parameters misconfiguration. Sometimes you may be done training earlier or due to some other issue do not want to continue running your EC2 instance.  From AWS Console navigate to EC2 service > Instance > select your instance> instance state > terminate.  Spot instances cannot be terminated from AWS console (with above method). Both Standard and Spot instance can be terminated by running below command on EC2 instance Shell (command line).  sudo shutdown now |
| **Uploading model to console** | * Navigate to the S3 service > select the bucket that was created in your base resources stack (can navigate to CloudFormation > base stack > Resources > Bucket) and locate the name of the model you ran. Open the upload directory, and hit 'copy S3 URI * Navigate to DeepRacer console> your models> import model> paste the S3 URI and name your model   (For example, first box is 's3://tyler-bucket/upload/" and second box is "tyler-model-1")   * You can now submit this model to races or evaluate in console |
| **Increment training/cloning model** | go to custom-files/nun env and change the following variables:   * DR\_LOCAL\_S3\_MODEL\_PREFIX=name of the new model you want to train * DR\_LOCAL\_S3\_PRETRAINED=True * DR\_LOCAL\_S3\_PRETRAINED\_PREFIX=name of the model you want to clone or train on top of |
| **Accessing logs for log analysis** | * Logs are automatically uploaded every 2 minutes of training to your model's S3 bucket. * Navigate to S3 service > select the bucket that was created in your resources stack (can find on CloudFormation > base stack > Resources > Bucket) and locate the name of the model you ran * In this folder, you can download the logs and use the guru log analysis tool to analyze logs locally (<https://github.com/dmh23/deep_racer_guru>) |
| **Adding users on team to access instance** | ./add-access.sh BASE-STACK-NAME ACCESS-STACK-NAME IP-TO-ADD  **BASE-STACK-NAME:** same name you selected in the previous section. If you forgot the name, you can check in CloudFormation for stacks that have been created before  **ACCESS-STACK-NAME:** the name of this cloudformation stack (recommended to be "your name-access. i.e. “tyler-access”)  **IP-TO-ADD:** IPV4 of the machine you are adding to be able to access the ec2 instance (<https://whatismyip.host/>).  NOT the same IP as create-base-resources.sh |
| **Troubleshooting** | If your instance doesn't start training, you can check logs in two locations:   * S3 bucket created in base resources, logs directory   EX: S3://base-tyler/model-name/logs   * In the boot cloud temp logs folder |