# AWS

## AWS CLI – command line interface

### Run on windows to create EC2 with spark-ec2 installed

aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HVHA]:

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*mfU3]:

Default region name [us-west-1]:

Default output format [json]:

<https://github.com/amplab/spark-ec2/blob/v4/ami-list/us-west-1/hvm>

### Authorise SSH access to EC2

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/authorizing-access-to-an-instance.html

### For ec2 cannot find the needed ami in aws ec2 dashboard, but can launch programmatically

aws ec2 describe-images --image-ids ami-72320f37

aws ec2 run-instances --image-id ami-72320f37 --instance-type m4.large --key-name AWSKeyPair2 --security-groups sec-grp-ssh-jupyter-spark

aws ec2 request-spot-instances --spot-price "0.1" --launch-specification file://C:/Dev/AWS/spot.json

aws ec2 describe-instances --query "Reservations[\*].Instances[\*].[InstanceId, ImageId, PublicDnsName]"

aws ec2 describe-instances --query "Reservations[\*].Instances[\*].[InstanceId,ImageId,PublicDnsName]" --filters "Name=image-id,Values=ami-72320f37"

## Putty SSH to AWS from Windows

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

PuttyGen to convert downloaded pem file to pkk

Putty GUI or

putty -ssh -i C:\Dev\AWS\AWSKeyPair2.ppk ec2-user@ec2-54-183-144-206.us-west-1.compute.amazonaws.com

<http://techexposures.com/how-to-copy-files-to-aws-ec2-server-from-windows-pc-command-prompt/>

pscp -i C:\Dev\AWS\AWSKeyPair2.ppk C:\Dev\AWS\AWSKeyPair2.pem ec2-user@ec2-54-183-144-206.us-west-1.compute.amazonaws.com:/home/ec2-user/AWSKeyPair2.pem

# AWS ML

## AWS AI

https://aws.amazon.com/amazon-ai/

## AWS CloudFormation for Deep Learning

https://aws.amazon.com/blogs/compute/distributed-deep-learning-made-easy/

https://github.com/awslabs/deeplearning-cfn

## AWS AMI for Deep Learning

### CS231N

<http://cs231n.github.io>

cs231n\_caffe\_torch7\_keras\_lasagne\_v2 (AMI ID: ami-125b2c72)

ubuntu@<public dns>

<http://cvit.iiit.ac.in/summerschool/>

Deep learning libraries: ubuntu 14.04 with torch, theano, keras, tensorflow, caffe, cuda, cudnn. based on Andrej Karpathy's cs231n image (ami-c23978a2)

### AWS with Jupyter

<http://efavdb.com/deep-learning-with-jupyter-on-aws/>

use empty password='' otherwise not working

#!/bin/bash

CERTIFICATE\_DIR="/home/ubuntu/certificate"

JUPYTER\_CONFIG\_DIR="/home/ubuntu/.jupyter"

THEANO\_CONFIG\_DIR="/home/ubuntu"

if [ ! -d "$CERTIFICATE\_DIR" ]; then

mkdir $CERTIFICATE\_DIR

openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout "$CERTIFICATE\_DIR/mykey.key" -out "$CERTIFICATE\_DIR/mycert.pem" -batch

chown -R ubuntu $CERTIFICATE\_DIR

fi

if [ ! -f "$JUPYTER\_CONFIG\_DIR/jupyter\_notebook\_config.py" ]; then

# generate default config file

#jupyter notebook --generate-config

mkdir $JUPYTER\_CONFIG\_DIR

# append notebook server settings

cat <<EOF >> "$JUPYTER\_CONFIG\_DIR/jupyter\_notebook\_config.py"

# Set options for certfile, ip, password, and toggle off browser auto-opening

c.NotebookApp.certfile = u'$CERTIFICATE\_DIR/mycert.pem'

c.NotebookApp.keyfile = u'$CERTIFICATE\_DIR/mykey.key'

# Set ip to '\*' to bind on all interfaces (ips) for the public server

c.NotebookApp.ip = '\*'

#c.NotebookApp.password = u''

c.NotebookApp.open\_browser = False

# It is a good idea to set a known, fixed port for server access

c.NotebookApp.port = 8888

EOF

chown -R ubuntu $JUPYTER\_CONFIG\_DIR

fi

<http://jupyter-notebook.readthedocs.io/en/latest/public_server.html>

<https://letsencrypt.org/getting-started/>

another link missing for being able to view jupyter notebook on ipad

$$$

Spot instances greyed out for Market AMIs

<https://www.paulwakeford.info/2016/01/07/aws-marketplace-and-spot-instances/>

<https://aws.amazon.com/marketplace/pp/B06VSPXKDX>

# TensorFlow

git clone <https://github.com/leriomaggio/deep-learning-keras-tensorflow>

<https://github.com/anishathalye/neural-style>

ModelZoo

<https://github.com/tensorflow/models>

## TensorBoard

<https://www.tensorflow.org/get_started/graph_viz>

<https://www.tensorflow.org/get_started/summaries_and_tensorboard>

# Lasagne

## Neural Art Style Transfer

<https://github.com/Lasagne/Recipes/blob/master/examples/styletransfer/Art%20Style%20Transfer.ipynb>

## KFKD

<http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/>

<https://www.kaggle.com/c/facial-keypoints-detection/details/deep-learning-tutorial>

# Frameworks

## dist-Keras Keras+Spark

<https://db-blog.web.cern.ch/blog/joeri-hermans/2017-01-distributed-deep-learning-apache-spark-and-keras>

## elephas Keras+Spark

<https://github.com/maxpumperla/elephas>

## PySpark install and Jupyter notebook configuration

<https://www.dataquest.io/blog/pyspark-installation-guide/>

# NVIDIA

nvidia-smi -l 1

# Setup instructions

AMI ML

<http://christopher5106.github.io/big/data/2016/01/27/two-AMI-to-create-the-fastest-cluster-with-gpu-at-the-minimal-engineering-cost-with-EC2-NVIDIA-Spark-and-BIDMach.html>

<http://markus.com/install-theano-on-aws/>

SPARK-EC2

<http://christopher5106.github.io/big/data/2016/01/27/two-AMI-to-create-the-fastest-cluster-with-gpu-at-the-minimal-engineering-cost-with-EC2-NVIDIA-Spark-and-BIDMach.html>

<https://github.com/amplab/spark-ec2.git>

<https://github.com/christopher5106/spark-ec2>

<http://ampcamp.berkeley.edu/3/exercises/launching-a-bdas-cluster-on-ec2.html>

# IPAD

Connect vis SSH on iPad

<http://www.packetnerd.com/?p=131>

# Courses

CS231N Convolutional NN for Visual Recognition

CS294 Deep Reinforcement Learning

CS224D Deep Learning for Natural Language Processing

<http://cvit.iiit.ac.in/summerschool/resources.html>

Deep RL

<http://efavdb.com/battleship/>

<http://karpathy.github.io/2016/05/31/rl/>

DeepLearning Course By Google

<https://www.udacity.com/course/deep-learning--ud730>

# Lerio Maggio Tutorial

should be versions:

numpy: 1.11.1

scipy: 0.18.0

matplotlib: 1.5.2

iPython: 5.1.0

scikit-learn: 0.18

keras: 2.0.2

Theano: 0.9.0

Tensorflow: 1.0.1

pip install --update pip

conda create -n tensorflow

source activate tensorflow

pip install --upgrade \

numpy==1.11.1 \

scipy==0.18.0 \

matplotlib==1.5.2 \

ipython==5.1.0 \

scikit-learn==0.18 \

keras==2.0.2 \

theano==0.9.0

https://www.tensorflow.org/install/install\_linux#InstallingAnaconda

https://www.tensorflow.org/install/install\_linux#the\_url\_of\_the\_tensorflow\_python\_package

pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.1.0-cp27-none-linux\_x86\_64.whl

or

pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\_gpu-1.1.0-cp27-none-linux\_x86\_64.whl

pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\_gpu-1.0.1-cp27-none-linux\_x86\_64.whl