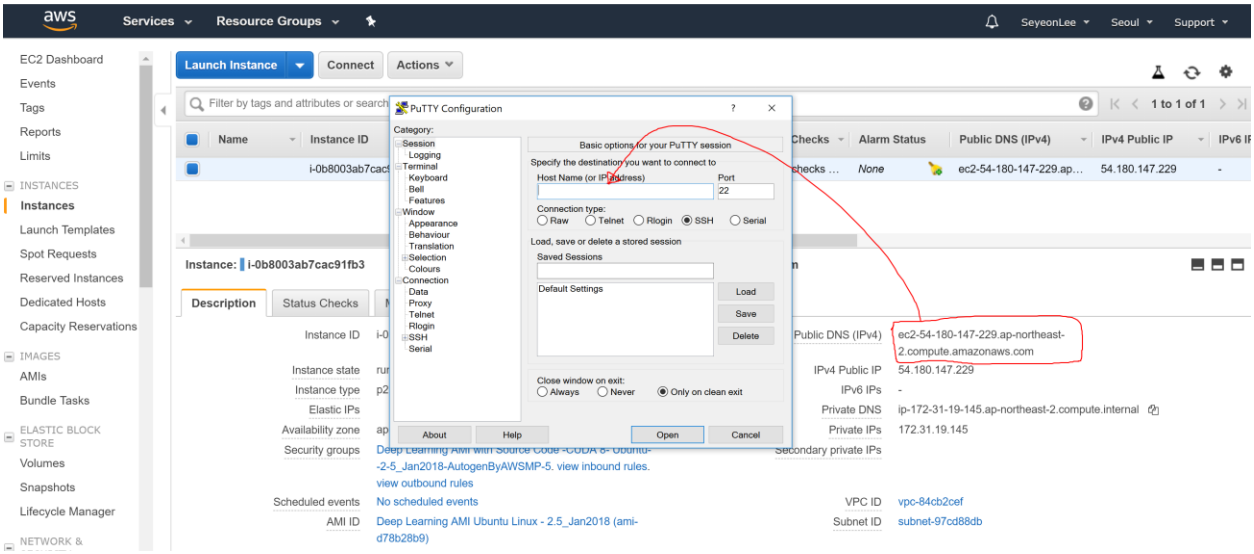
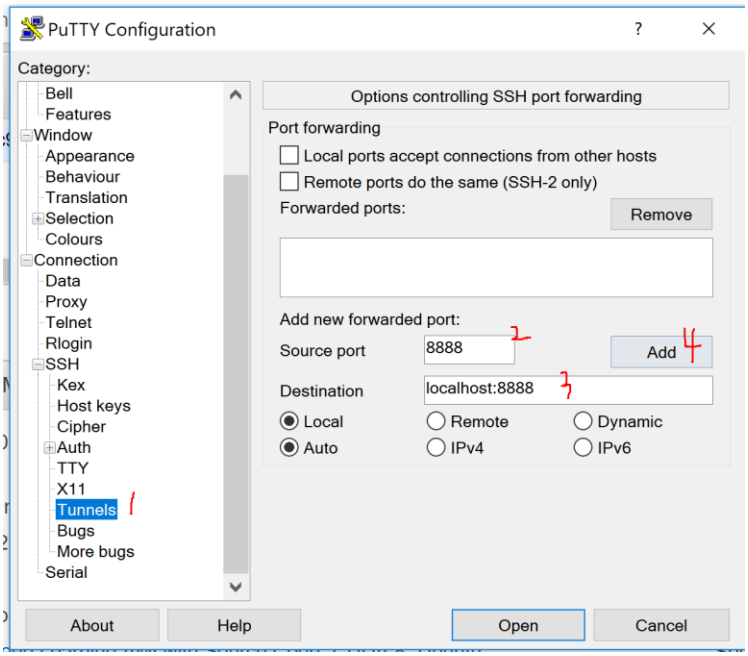
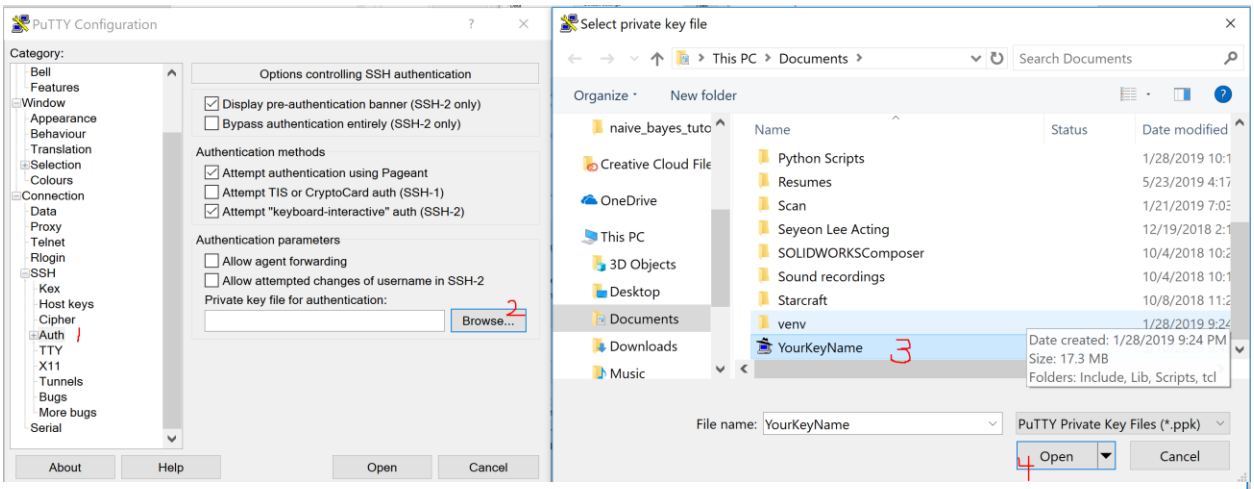
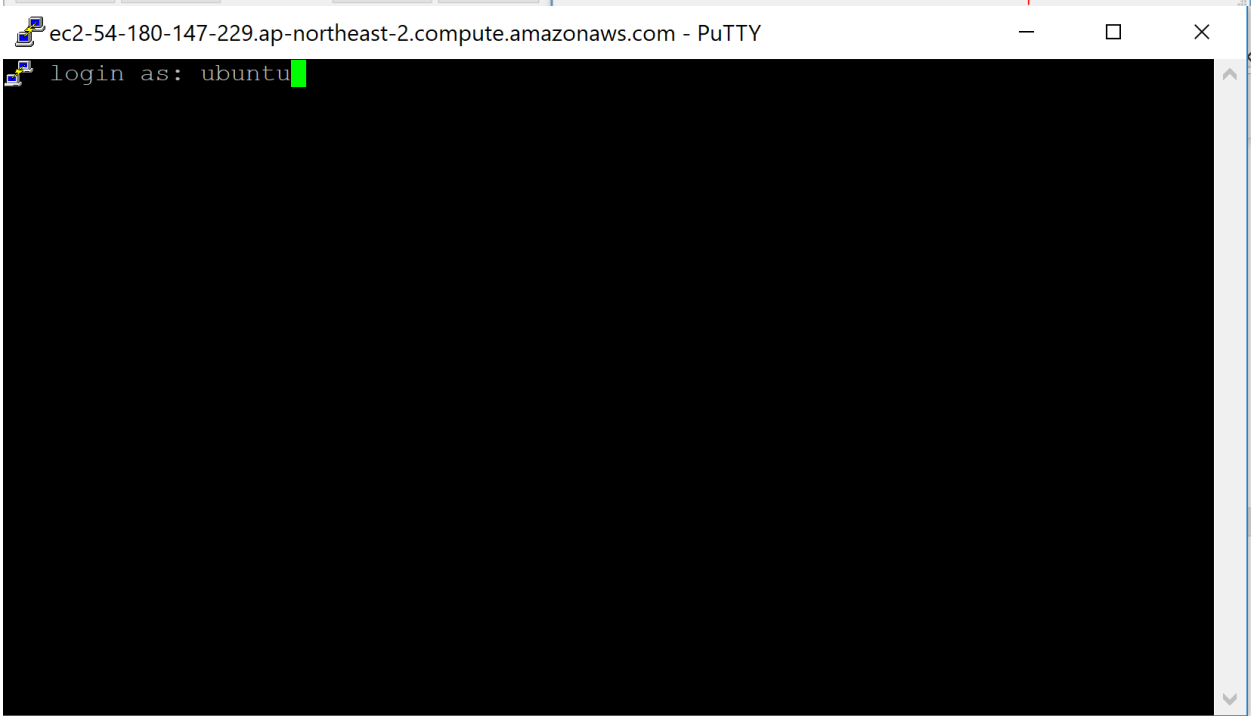

aws
Services
Resource Groups
SeyeonLee
Seoul
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
Launch Instance
Connect
Actions
Filter by tags and attributes or search
1 to 1 of 1
Name
Instance ID
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
i-0b8003ab7cac
None
ec2-54-180-147-229.ap...
54.180.147.229
PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Help
Open
Cancel
Instance: i-0b8003ab7cac91fb3
Description
Status Checks
Instance ID
Instance state
Instance type
Elastic IPs
Availability zone
Security groups
Scheduled events
AMI ID
No scheduled events
Deep Learning AMI Ubuntu Linux - 2.5_Jan2018 (ami-d78b28b9)
Public DNS (IPv4)
ec2-54-180-147-229.ap-northeast-2.compute.amazonaws.com
IPv4 Public IP
54.180.147.229
IPv6 IPs
Private DNS
ip-172-31-19-145.ap-northeast-2.compute.internal
Private IPs
172.31.19.145
Secondary private IPs
VPC ID
vpc-84cb2cef
Subnet ID
subnet-97cd88db


PuTTY Configuration
Category:
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Host keys
Cipher
Auth
TTY
X11
Tunnels
Bugs
More bugs
Serial
Options controlling SSH port forwarding
Port forwarding
Local ports accept connections from other hosts
Remote ports do the same (SSH-2 only)
Forwarded ports:
Remove
Add new forwarded port:
Source port
8888
Add
Destination
localhost:8888
Local
Remote
Dynamic
Auto
IPv4
IPv6
About
Help
Open
Cancel


PuTTY Configuration
Category:
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Host keys
Cipher
Auth
TTY
X11
Tunnels
Bugs
More bugs
Serial
Options controlling SSH authentication
Display pre-authentication banner (SSH-2 only)
Bypass authentication entirely (SSH-2 only)
Authentication methods
Attempt authentication using Pageant
Attempt TIS or CryptoCard auth (SSH-1)
Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
Allow agent forwarding
Allow attempted changes of username in SSH-2
Private key file for authentication:
Browse...
1
2
About
Help
Open
Cancel
Select private key file
This PC › Documents ›
Search Documents
Organize
New folder
naive_bayes_tuto
Creative Cloud File
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Name
Status
Date modified
Python Scripts
Resumes
Scan
Seyeon Lee Acting
SOLIDWORKSComposer
Sound recordings
Starcraft
venv
YourKeyName
3
Date created: 1/28/2019 9:24 PM
Size: 17.3 MB
Folders: Include, Lib, Scripts, tcl
File name: YourKeyName
PuTTY Private Key Files (*.ppk)
Open
Cancel
4


ec2-54-180-147-229.ap-northeast-2.compute.amazonaws.com - PuTTY
login as: ubuntu

Use `chmod 600 YourKeyName.pem` to make the key file accesible only by you.
Type `ssh -i YourKeyName.pem ubuntu@X.X.X.X`, where:

- `X.X.X.X` is the IPv4 Public IP found in AWS, and
- `YourKeyName.pem` is the name of your .pem file.

Note that if you've used a different AMI or specified a username, `ubuntu` will be replaced with the username, such as `ec2-user` for some Amazon AMI's. You would then instead enter `ssh -i YourKeyName.pem ec2-user@X.X.X.X`

Configure Jupyter notebook settings

In your instance, in order to create a config file for your Jupyter notebook settings, type: `jupyter notebook --generate-config`.
Then, to change the IP address config setting for notebooks (this is just a fancy one-line command to perform an exact string match replacement; you could do the same thing manually using vi/vim/nano/etc.), type: `sed -ie "s/#c.NotebookApp.ip = 'localhost'/#c.NotebookApp.ip = '*'/g" ~/.jupyter/jupyter_notebook_config.py`

Test the Instance

Make sure everything is working properly by verifying that the instance can run a Keras notebook.

On the EC2 instance

- Clone a GitHub repository
- `git clone https://github.com/udacity/aind2-cnn.git`
- Enter the repo directory
- `cd aind2-cnn`
- Install the requirements
- `sudo python3 -m pip install -r requirements/requirements.txt`
- Start Jupyter notebook
  `jupyter notebook --ip=0.0.0.0 --no-browser`

Now you should be able to access your jupyter notebook using the url

http://localhost:8888

jupyter

Password or token: Log in

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

```
jupyter notebook list
```

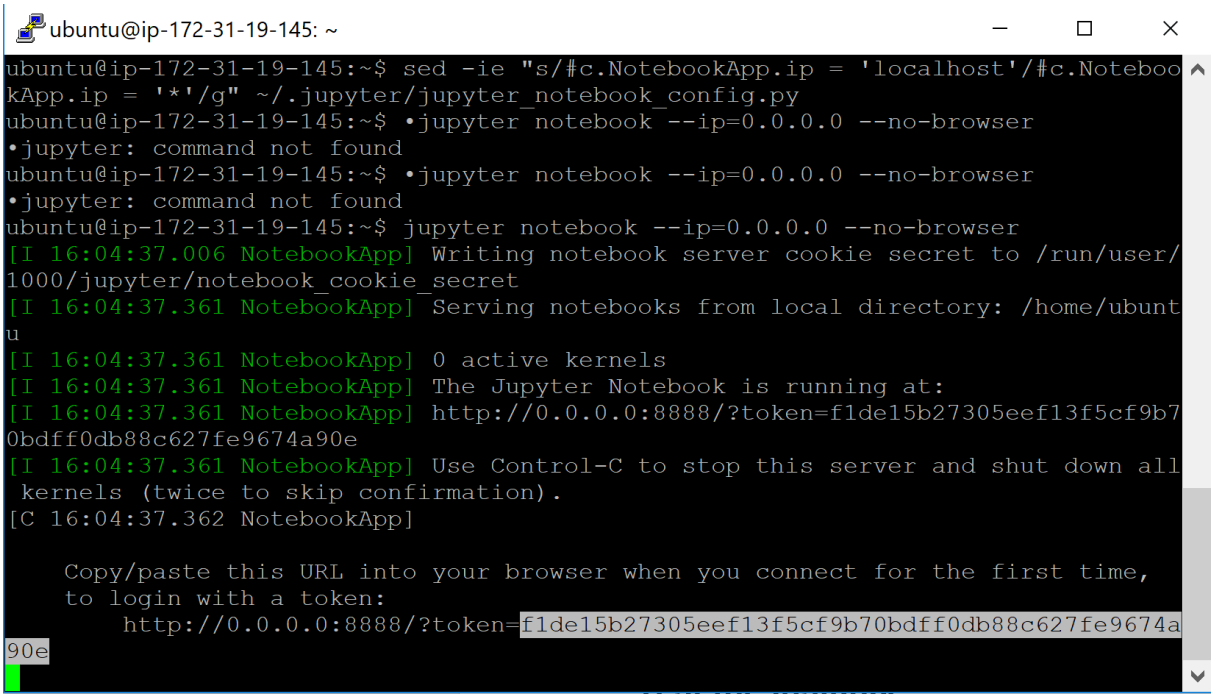
will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.



Copy the token and paste it yonder -> DONE