Linux Workshop

•••

Kriengkrai J. <kk@jira.org>
Super AI Engineer 2020

Workshop Outline (180 minutes)

Topics

1. Linux VM and Basic Commands
2. Python Environment (Miniconda)
3. Visual Studio Code - Remote SSH
4. Jupyter Notebook / Lab (Web service)

Tasks	Minutes
Linux VM and Basic Commands	30
Python Environment (Miniconda)	20
Visual Studio Code: Extensions, Coding, Terminal	20
Break #1 - Q&A	15
Object Detection with YOLO11	20
API service	20
Break #2 - Q&A	15
Run Jupyter Notebook in VS Code	15
Jupyter Notebook / Lab (Web Service)	15
Q&A	10

Linux VM and Basic Commands

•••

With Huawei Cloud

Related Huawei Cloud Services

- Elastic Cloud Server (VM - CPU / GPU / RAM)

  https://www.huaweicloud.com/intl/en-us/product/ecs.html

- Elastic Volume Service (Disk storage)

  https://www.huaweicloud.com/intl/en-us/product/evs.html

- Elastic IP (Network / Internet)

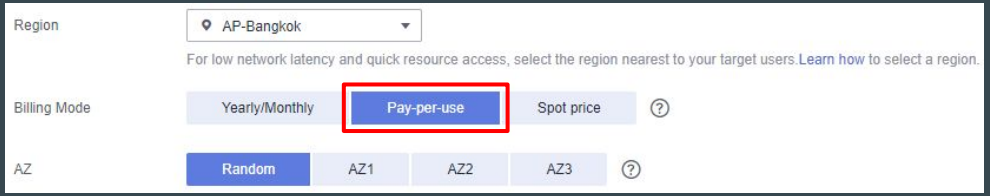
  https://www.huaweicloud.com/intl/en-us/product/eip.html

Linux VM Components

** Pay-per-use **

1. Region
   - Bangkok
   - Singapore
   - Hong Kong
2. CPU / RAM
3. GPU:
   - NVIDIA A100, V100
   - NVIDIA T4 x2, x4
4. Disk
5. Network

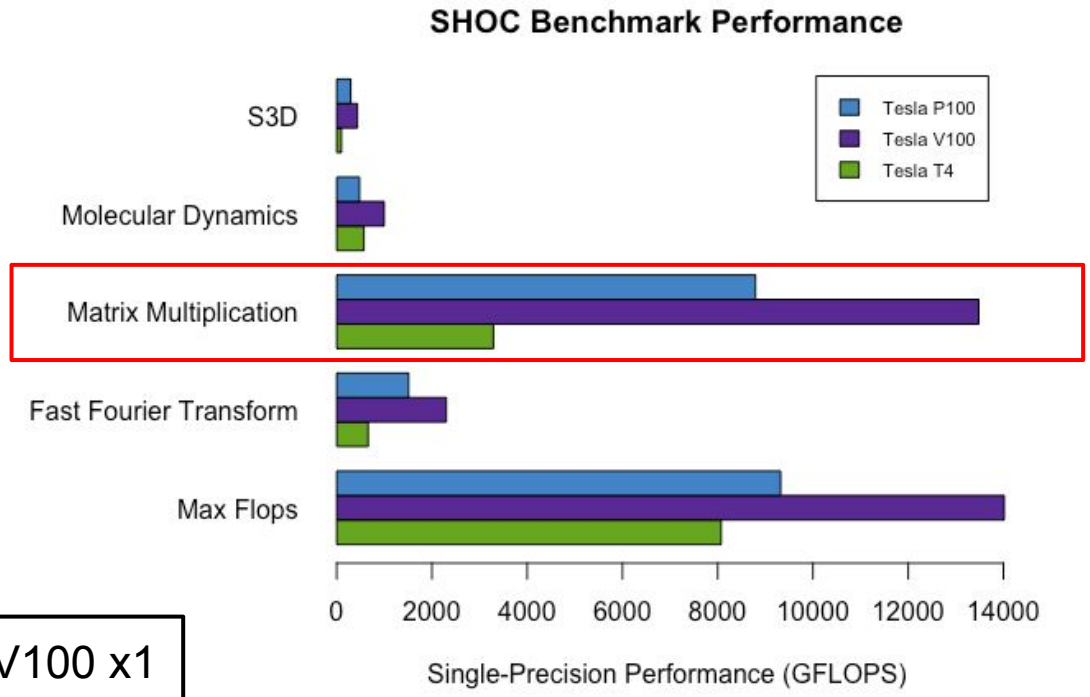
$$$$ USD / Hour $$$$



Huawei Elastic Cloud Server

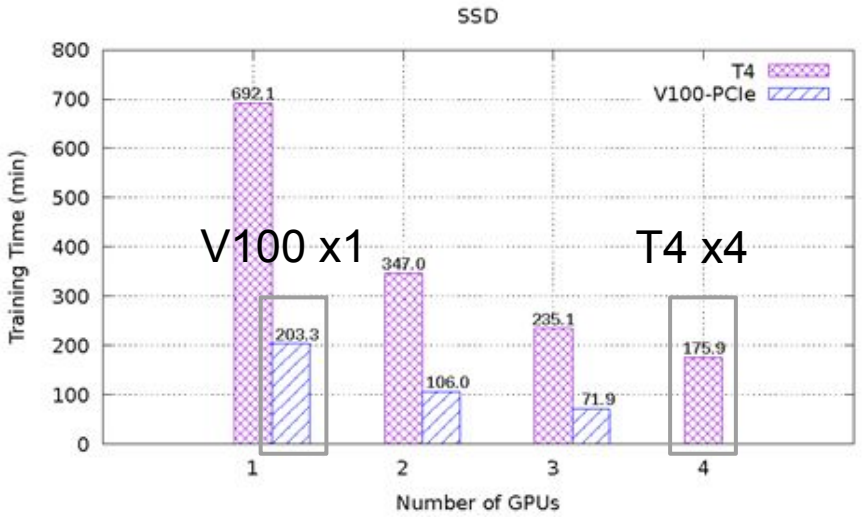
VM (with GPU)	CPU/RAM	Bandwidth / ping	Region (ping) / USD/hour				Usage
			AP-Hong Kong	AP-Bangkok	AP-Singapore	CN North-	
			117 ms	75 ms	83.5 ms	324.5 ms / 532 ms	
g5.8xlarge.4	32vCPUs/128GB	5 Mbps	5.89	5.06	5.02	3.19	USD/hr
(GPU: 1xV100 16GB)			989.52	850.08	843.36	535.92	(24hr/1wk)
			618.45	531.30	527.10	334.95	(15hr/1wk)
			494.76	425.04	421.68	267.96	(12hr/1wk)
		10 Mbps	6.09	5.17	5.14	3.38	USD/hr
			1,023.12	868.56	863.52	567.84	(24hr/1wk)
			639.45	542.85	539.70	354.90	(15hr/1wk)
			511.56	434.28	431.76	283.92	(12hr/1wk)
pi2.8xlarge.4	32vCPUs/128GB	5 Mbps	4.71	4.07	4.12	4.24	USD/hr
(GPU: 4xT4, 4x16GB)			791.28	683.76	692.16	712.32	(24hr/1wk)
			494.55	427.35	432.60	445.20	(15hr/1wk)
			395.64	341.88	346.08	356.16	(12hr/1wk)
		10 Mbps	4.91	4.18	4.24	4.43	USD/hr
			824.88	702.24	712.32	744.24	(24hr/1wk)
			515.55	438.90	445.20	465.15	(15hr/1wk)
			412.44	351.12	356.16	372.12	(12hr/1wk)

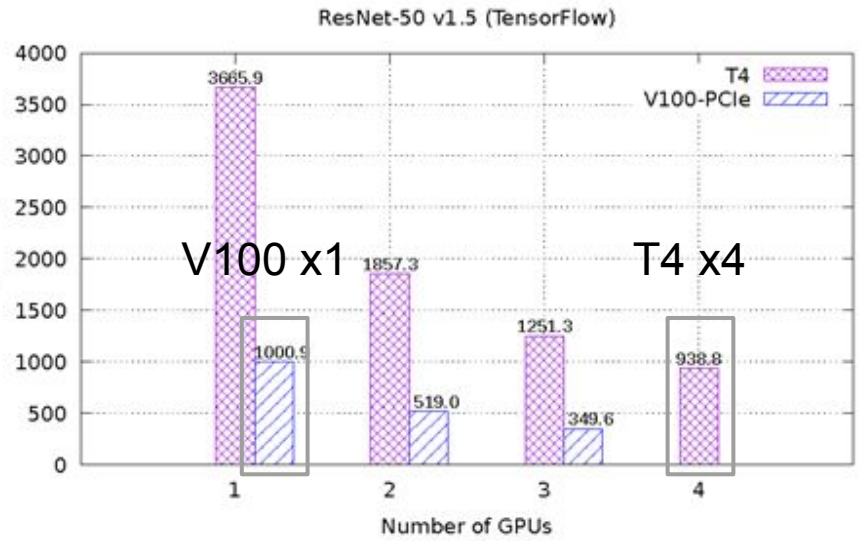
GPU: NVIDIA T4 vs V100



T4 x4 ≈ V100 x1

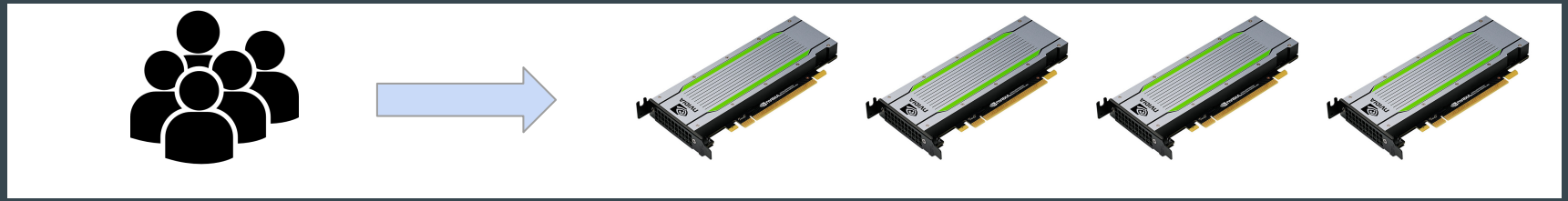
GPU: NVIDIA T4 x4 vs V100





T4 x4 ≈ V100 x1

GPU: Sharing



```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.67       Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:21:01.0 Off |                    0 |
| N/A   54C    P0    27W /  70W |   1745MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:21:02.0 Off |                    0 |
| N/A   67C    P0    73W /  70W |  11865MiB / 15079MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla T4            Off  | 00000000:21:03.0 Off |                    0 |
| N/A   44C    P0    32W /  70W |   1587MiB / 15079MiB |     40%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla T4            Off  | 00000000:21:04.0 Off |                    0 |
| N/A   45C    P0    54W /  70W |   1587MiB / 15079MiB |     61%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0     31569      C   python                                      1735MiB |
|    1     17904      C   python                                     11855MiB |
|    2     31637      C   python                                      1577MiB |
|    3     31707      C   python                                      1577MiB |
+-----------------------------------------------------------------------------+
```

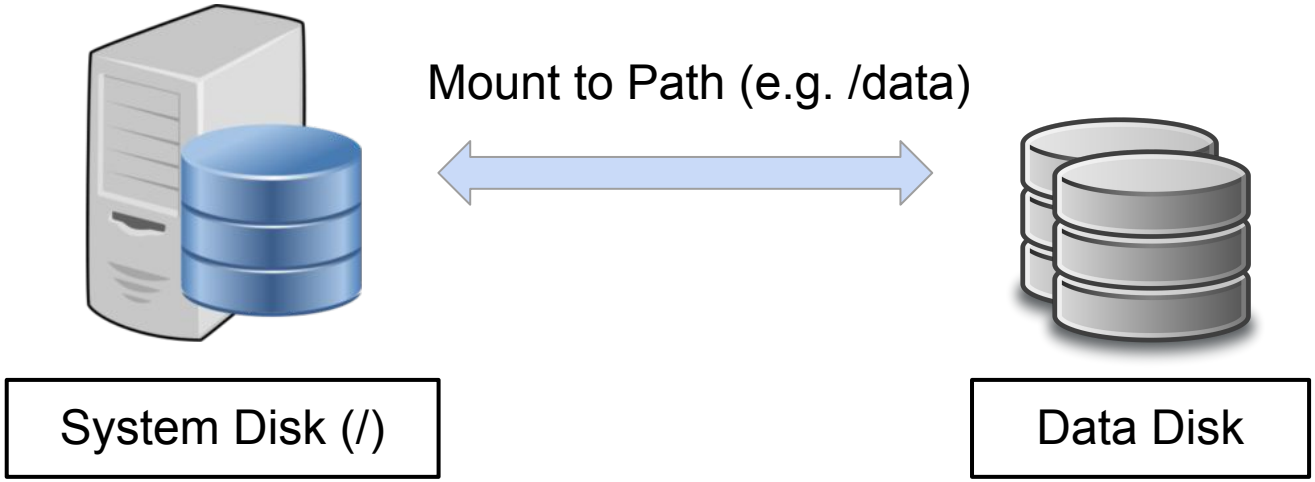
```
import os

os.environ["CUDA_VISIBLE_DEVICES"] = "0"
```

GPU ID: 0,1,2,3 (in order by nvidia-smi)

0,3 = GPU#0 + GPU#3

Disk Types

- System Disk - MAX 1 TB
- Data Disk - MAX 32 TB



Disk Spec

Disk Specifications

Ultra-high I/O	High I/O	Common I/O
33,000 IOPS	5,000 IOPS	2,200 IOPS
350 MB/s	150 MB/s	50 MB/s
1 ms	1 ms - 3 ms	5 ms - 10 ms
$0.0003 USD/GB-hour	**$0.0001 USD**/GB-hour	**$0.0001 USD**/GB-hour

Disk Size — 1,000 + GB Select Data Source

Selected Specifications Ultra-high I/O | 1000GB IOPS limit: 33,000. Throughput: 350 MB/s. Learn how they were calculated

1 TB = $0.3/Hour = $7.2/Day

Network

EIP

Auto assign | Use existing | Not required

EIP Type

Dynamic BGP

Greater than or equal to 99.95% service availability rate

Billed By

Bandwidth
For heavy/stable traffic

Traffic
For light/sharply fluctuating tra...

Shared bandwidth
For staggered peak hours

Billed based on total traffic irrespective of usage duration; configurable maximum bandwidth size.

Bandwidth Size

5 | 10 | 20 | 50 | 100 | Custom | 300

The bandwidth can be from 1 to 300 Mbit/s.

Free Anti-DDoS protection

Sample Budget Management

Region

- AP-Bangkok

GPU

- pi2.8xlarge.4 = T4 x 4

Disk

- System Disk
- Data Disk

Network

- Traffic (Max 300 Mbps)

AP-Bangkok	Disk[IOPS]	USD per hour/hour	24x7 168	15x7 105	12x7 84
pi2.8xlarge.4 (sys)	40GB[1440]	4.002	672.34	420.21	336.17
	500GB[4200]	4.048	680.06	425.04	340.03
	1TB[5000]	4.1004	688.87	430.54	344.43
	500GB[26500]	4.133	694.34	433.97	347.17
	1TB[33000]	4.2744	718.10	448.81	359.05
pi2.8xlarge.4	500GB[4200]	4.052	680.74	425.46	340.37
(FREE sys 40GB + data)	1TB[5000]	4.1044	689.54	430.96	344.77
	500GB[26500]	4.137	695.02	434.39	347.51
	1TB[33000]	4.2784	718.77	449.23	359.39
EIP (Bandwidth)	5 Mbps	0.065	10.92	6.83	5.46
	10 Mbps	0.175	29.40	18.38	14.70
	20 Mbps	0.395	66.36	41.48	33.18
	50 Mbps	1.055	177.24	110.78	88.62
	100 Mbps	2.155	362.04	226.28	181.02

Billing

Bills | Bill Details

Billing Cycle Jan 2021

Export

Estimated Total Discounts: $0.00000000 USD	$27.18 USD
HUAWEI CLOUD Charges	$27.18 USD

Details

HUAWEI CLOUD Charges	$27.18 USD
Elastic Volume Service	$1.33 USD
Elastic Cloud Server	$25.85 USD
Virtual Private Cloud	$0.00 USD

Access to a Linux VM

...

Task 1.1) Access to a Linux VM

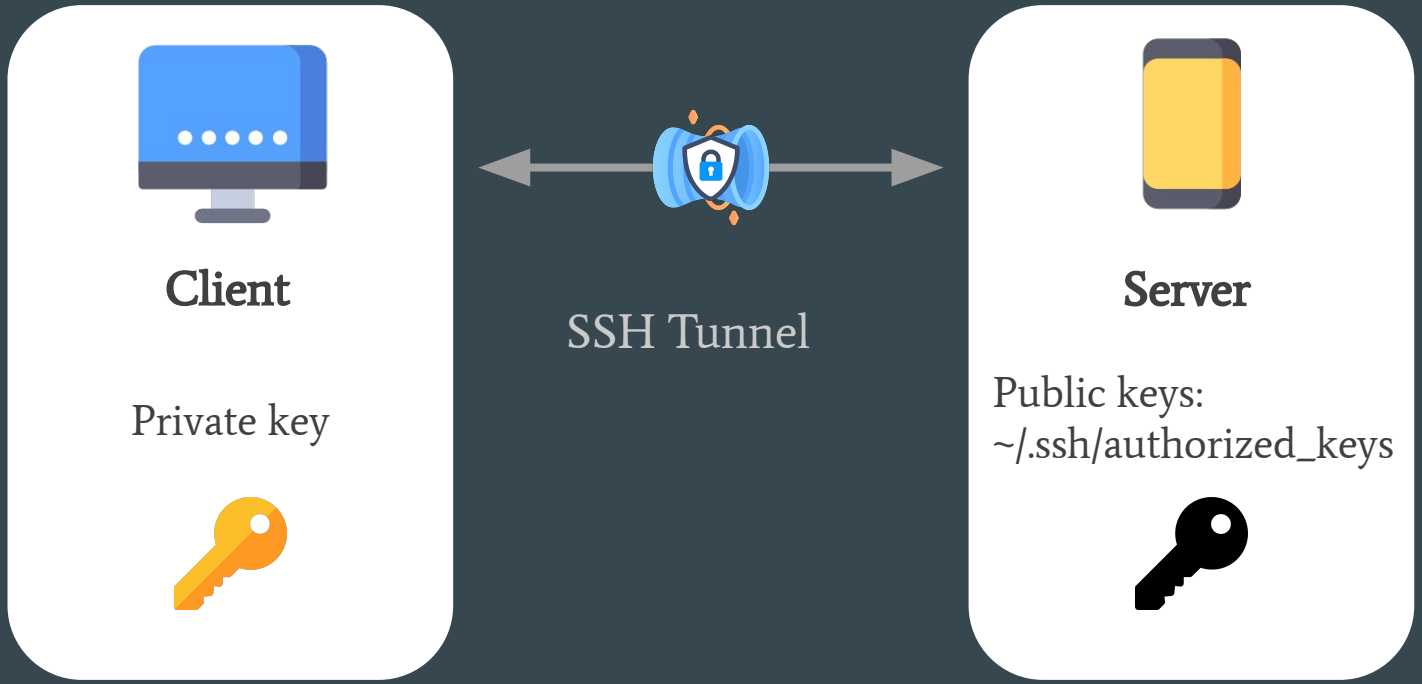
Host	vm1.lnode.com	vm2.lnode.com	vm3.lnode.com	vm4.lnode.com
Super AI ID	500003 - 500397	500407 - 500952	500988 - 502138	502197 - 510290

Username: u50xxxx Password: SuperAI@5

SSH to the Linux VM :

```
ssh    u50xxx@vm?.lnode.com
```

Task 1.2) Setup SSH Key Access (Concept)



Task 1.2) Setup SSH Key Access (1/2)

1. Generate an SSH key pair (with or without passphrase)

```
ssh-keygen -t ed25519 -f mykey    # output file name (mykey, mykey.pub)
```

2. Copy the public key to the server

```
sftp  u50xxx@vm?.lnode.com
mkdir .ssh
chmod 700 .ssh                    # 700 = (user)rwx / (group)--- / (others)---
cd .ssh
put mykey.pub authorized_keys
chmod 600 authorized_keys         # 600 = (user)rw- / (group)--- / (others)---
quit
```

3. Access with the private key

```
ssh -i mykey u50xxx@vm?.lnode.com
```

Setup SSH Key Access (2/2)

4. Setup SSH Config

   Windows: C:\users\<myuser>\.ssh\config
   Mac / Linux: ~/.ssh/config

```
Host  vm
    HostName          vm?.lnode.com
    User              u50xxx
    IdentityFile      ~/.ssh/mykey          # Path to private key
```

5. Access with the Host in SSH Config

```
ssh  vm
sftp vm
```

Passwordless System

1. Setup SSH server config
   Files: /etc/ssh/sshd_config, /etc/ssh/sshd_config.d/*

```
PasswordAuthentication  no
```

Restart ssh service

```
systemctl restart ssh          # for Ubuntu
```

2. Setup SUDOers not to use password (File: /etc/sudoers)

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) NOPASSWD: ALL
```

3. Add super users to supplementary group “sudo”

```
sudo usermod -a -G sudo <username>
```

Basic Linux Commands

...

Task 1.3) Basic Linux Commands

Network

- **ssh, sftp** - Secure-Shell (SSH)
- **ping** - host test
- **wget, gdown** - download files

File System (Storage)

- **ls** - list files, directories
- **rm** - remove files, directories
- **cat** - display text files, merge files
- **pwd, cd** - current, change directory
- **mkdir, rmdir** - make, remove directory
- **du, df** - disk usage, disk free
- **cp, mv** - copy, move-rename
- **ln** - create link

. = current dir, .. = prev dir, / = root, ~ = home

System

- **apt** - package management
- **passwd** - change password
- **sudo, su** - superuser do, switch user
- **env, export** - environment variables
- **date, timedatectl** - set date-time, timezone
- **reboot, shutdown**

Misc

- **find** - search for files, directories
- **chown, chgrp** - change owner, group
- **chmod** - change mode (permission)
- **nano, vi** - text editor
- **ps, kill, pkill, killall** - process management
- **top, htop** - system monitor
- **screen, tmux** - terminal multiplexer

Python Environment

•••

with Miniconda

Python Environment

System-wide Environment → sudo **apt install** python

/usr/bin/python

User Environment → venv, virtualenv, **Miniconda**

/home/*user*/*

Miniconda

A Free minimal installer for conda

Linux installers

Python version	Name	Size
Python 3.9	Miniconda3 Linux 64-bit	63.6 MiB
	Miniconda3 Linux-aarch64 64-bit	62.6 MiB
	Miniconda3 Linux-ppc64le 64-bit	60.6 MiB
	Miniconda3 Linux-s390x 64-bit	57.1 MiB
Python 3.8	Miniconda3 Linux 64-bit	98.8 MiB
	Miniconda3 Linux-aarch64 64-bit	94.8 MiB
	Miniconda3 Linux-ppc64le 64-bit	93.3 MiB
	Miniconda3 Linux-s390x 64-bit	89.0 MiB
Python 3.7	Miniconda3 Linux 64-bit	84.9 MiB
	Miniconda3 Linux-aarch64 64-bit	89.2 MiB
	Miniconda3 Linux-ppc64le 64-bit	88.1 MiB
	Miniconda3 Linux-s390x 64-bit	84.1 MiB

https://www.anaconda.com/docs/getting-started/miniconda/main

Tasks

2.1) Install Miniconda (Linux x64 Installer)

2.2) Manage Python/Conda Packages (pip/conda)

2.3) Work with multiple Python environments

2.4) Remove Env - Uninstall Miniconda

Task 2.1) Install Miniconda (Linux x64 Installer)

```
# Download
wget   https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

# Run
bash   Miniconda3-latest-Linux-x86_64.sh
```

Do you accept the license terms? >>> yes

Miniconda3 will now be installed into this location: [*/home/u50xxxx/miniconda3*] >>> <ENTER>

Do you wish to update your shell profile to automatically initialize conda? >>> yes

(or update login script by: `miniconda3/bin/` **`conda init`**)

```
# relogin or reload .bashrc
source  .bashrc
```

Task 2.2) Manage Python/Conda Packages (pip/conda)

List installed Python/ Software Packages:

```
pip/conda list
    (Channel:pypi = Python Package Index)
```

Install Python/ Software Packages:

```
pip/conda install <package>
    [==version ">=version" "<version"]
```

Uninstall Python/ Software Packages:

```
pip/conda uninstall <package>
```

Try to install packages:

- gdown
- numpy==2.0.0

TensorFlow and PyTorch Installation

- **TensorFlow**

GPU

Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow-2.19.0	3.9-3.12	Clang 18.1.8	Bazel 6.5.0	9.3	12.5
tensorflow-2.18.0	3.9-3.12	Clang 17.0.6	Bazel 6.5.0	9.3	12.5
tensorflow-2.17.0	3.9-3.12	Clang 17.0.6	Bazel 6.5.0	8.9	12.3
tensorflow-2.16.1	3.9-3.12	Clang 17.0.6	Bazel 6.5.0	8.9	12.3
tensorflow-2.15.0	3.9-3.11	Clang 16.0.0	Bazel 6.1.0	8.9	12.2
tensorflow-2.14.0	3.9-3.11	Clang 16.0.0	Bazel 6.1.0	8.7	11.8
tensorflow-2.13.0	3.8-3.11	Clang 16.0.0	Bazel 5.3.0	8.6	11.8
tensorflow-2.12.0	3.8-3.11	GCC 9.3.1	Bazel 5.3.0	8.6	11.8
tensorflow-2.11.0	3.7-3.10	GCC 9.3.1	Bazel 5.3.0	8.1	11.2
tensorflow-2.10.0	3.7-3.10	GCC 9.3.1	Bazel 5.1.1	8.1	11.2

https://www.tensorflow.org/install/pip

CUDA 12.5 ⇒ TensorFlow 2.18.0

CUDA 12.3 ⇒ TensorFlow 2.16.1

CUDA 11.2 ⇒ TensorFlow 2.10.0

- **PyTorch**

https://pytorch.org/get-started/locally/

Task 2.3) Work with multiple Python environments

base [Python 3.12]

- TensorFlow [Latest]
- numpy [Latest]

py2 [Python 2.7]

- TensorFlow 2.1
- numpy 1.16.6

Create Environment:

```
conda create -n py2 python=2.7
```

Activate / Deactivate Environment:

```
conda activate  py2
conda deactivate
```

CONDA Cheat Sheet:
https://conda.io/projects/conda/en/latest/user-guide/cheatsheet.html

Task 2.3) Work with multiple Python environments (cont.)

base [Python 3.12]

- numpy [Latest]

*/home/u50xxxx/miniconda3/***bin/python**

py2 [Python 2.7]

- numpy 1.16.6

*/home/u50xxxx/miniconda3/***envs/py2/bin/python**

#1: Try with different path to python

```
/home/u50xxxx/miniconda3/bin/python -c "import numpy; print(numpy.version.version)"
/home/u50xxxx/miniconda3/envs/py2/bin/python -c "import numpy; print(numpy.version.version)"
```

#2: Run with conda

```
conda run -n base python -c "import numpy; print(numpy.version.version)"
conda run -n py2 python -c "import numpy; print(numpy.version.version)"
```

Task 2.4) Remove Env - Uninstall Miniconda

- Remove Environment:

```
conda deactivate
conda remove -n py2 --all
#or conda env remove -n py2
```

- Uninstall Miniconda

```
conda deactivate
conda init --reverse                    # remove conda from login script
rm  -rf  ~/miniconda3 ~/.conda          # remove files
```

~ = $HOME = /home/*u50xxxx*

Visual Studio Code

...

Visual Studio Code

https://code.visualstudio.com/

Tasks

3.1) Install Extensions
- Remote - SSH
- Python

3.2) Coding and Terminal

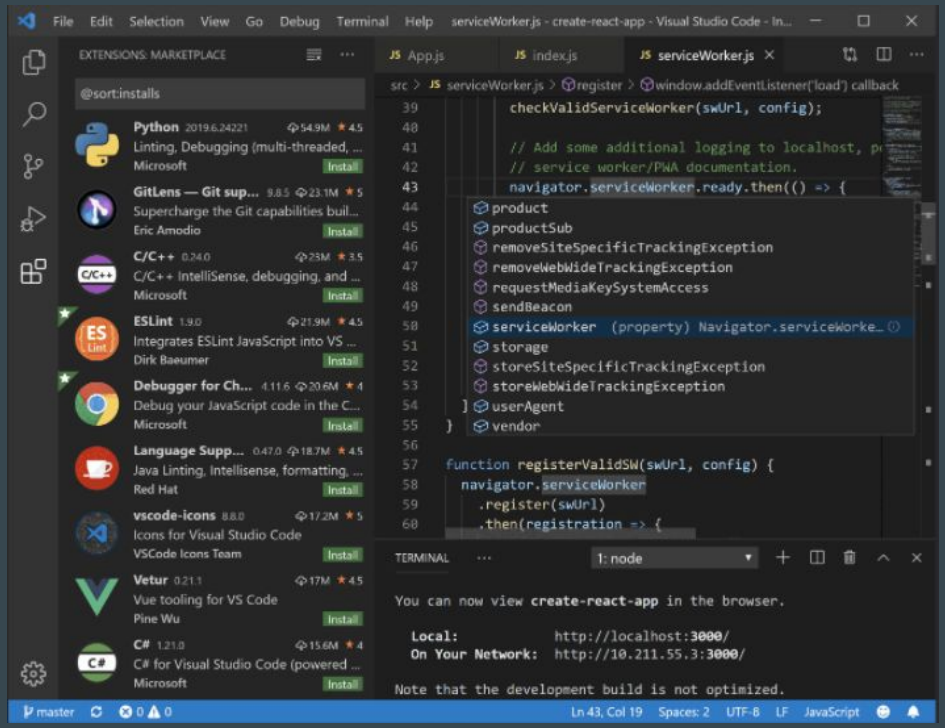
3.3) Object Detection with YOLO

3.4) File Transfer

3.5) Object Detection API (FastAPI)

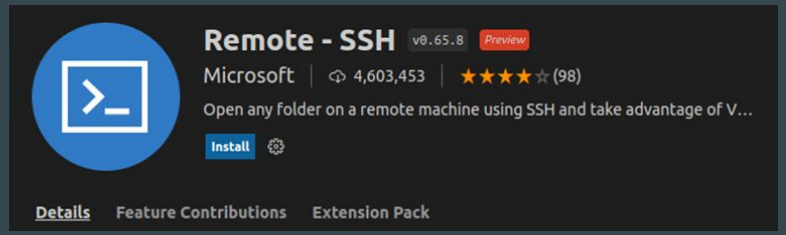
3.6) Port Forwarding

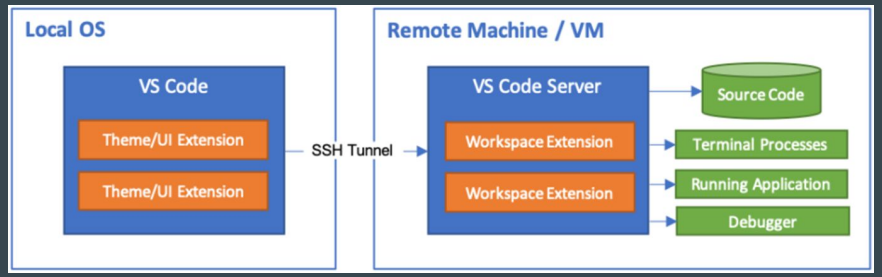
3.7) Jupyter Notebook (ipynb) support



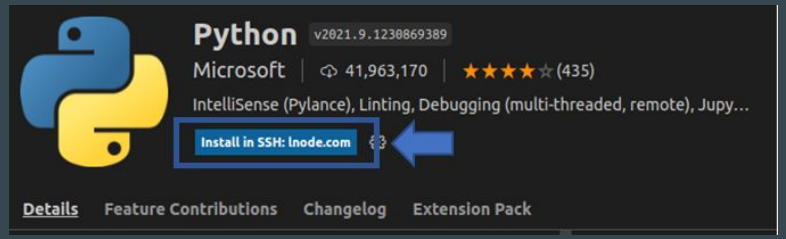
Task 3.1) Install Extensions

- **Remote - SSH**



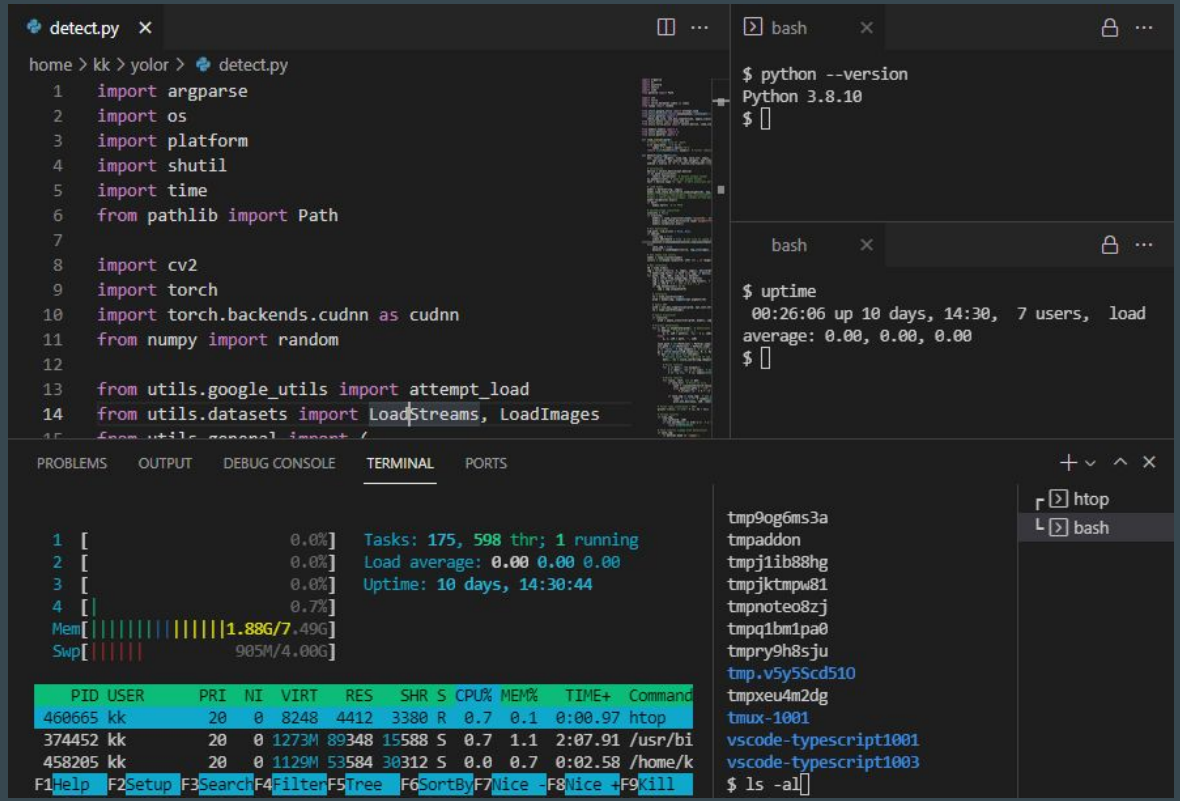


- **Python**



Task 3.2) Coding and Terminal

- **Connect to Host**
- **Explorer**
- **Write / Run Code**
- **Open / Close Terminal**
- **Move / Split Screens**



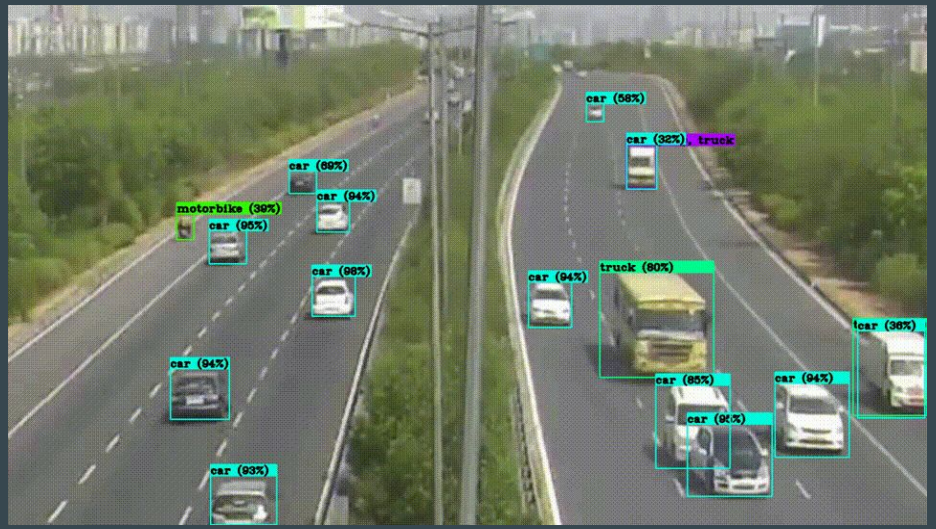
Linux Workshop

•••

Break #1 & QA

Task 3.3) Object Detection with YOLO

Object Detection

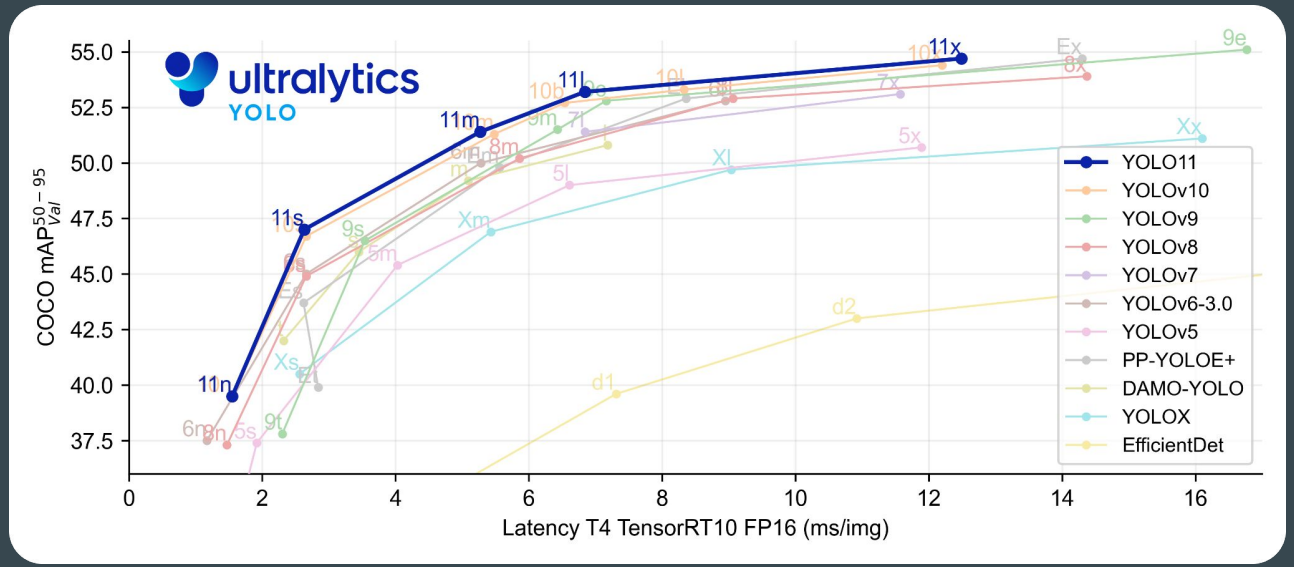


Object Tracking

Object Localization ⇒ Image Classification ⇒ Object Tracking

YOLO Family

Ultralytics YOLO11 vs others



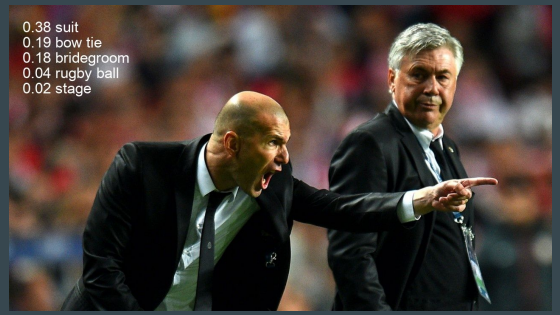
YOLO11 - https://docs.ultralytics.com/models/yolo11
YOLOv12 - https://github.com/sunsmarterjie/yolov12

YOLO Family

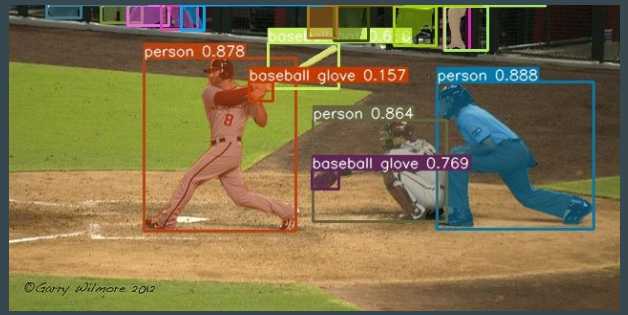
YOLOv12
YOLO11
YOLOv1-10
YOLO9000
PP-YOLO
YOLOR
YOLOX
…

More YOLO Tasks

Pose Estimation

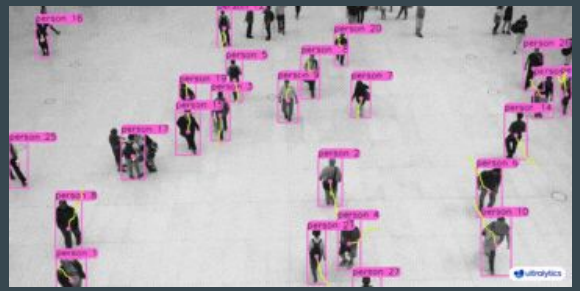


Classification



Instance Segmentation

Oriented Bounding Boxes



Track

and more..

YOLO11 - Installation

1. Install Ultralytics YOLO

```
pip install ultralytics
```

2. Try CLI Inference with pre-trained weight (COCO dataset - 80 classes)

```
# weight in: /data/yolo11n.pt, sample images: /data/*.jpg

yolo predict model=/data/yolo11n.pt source=/data/*.jpg
```

YOLO11 - Python

3. Inference with Python

```
import os
import glob
from ultralytics import YOLO

model = YOLO('/data/yolo11n.pt')

results = model(glob.glob('/data/*.jpg'))

for result in results:
    filename = f'result_{os.path.basename(result.path)}'
    result.save(filename)
```

Ref: https://docs.ultralytics.com/modes/predict/

Task 3.4) File Transfer

Try with your image files !

Put your result in:
https://drive.google.com/drive/folders/1hnpkJpOtByBJkASNntwIm_i0npU-wNFu

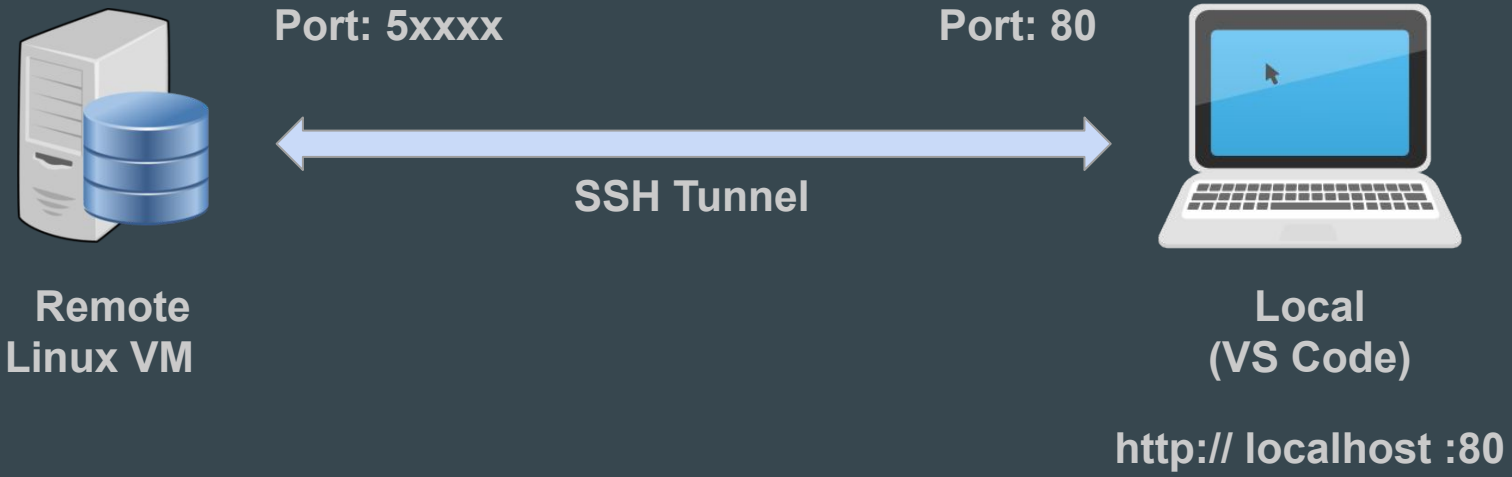
Task 3.5) Object Detection API (with FastAPI)

Web Browser
- API Client

API Server (HTTP)

IP:	0.0.0.0 (Public / Internet) 127.0.0.1 (Private / Local host)
Port:	*50000+superai id(4 digits)* valid port: 1024 - 65535)
Services:	**1) Main page** Method: GET Path: / Return: HTML (web) **2) Object Detection API** Method: POST Path: /detect/ Body: file (binary) Return: file (binary)

Task 3.6) Port Forwarding



Linux Workshop

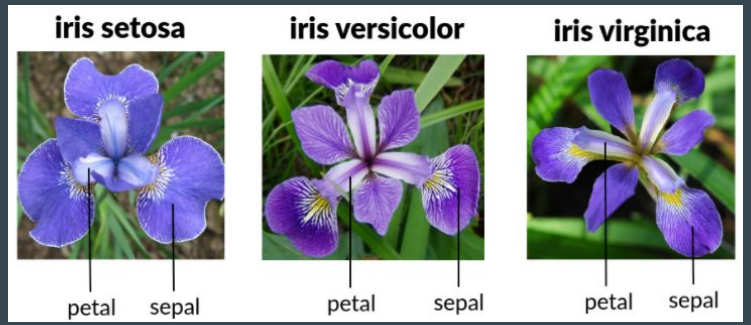
•••

Break #2 & QA

Task 3.7) Jupyter Notebook (ipynb) support

```
pip install jupyter
```

- Open / Create a new file with extension “.ipynb”
- Write Markdown and Code cells
- Run Code cells
- Try iris classification with scikit-learn

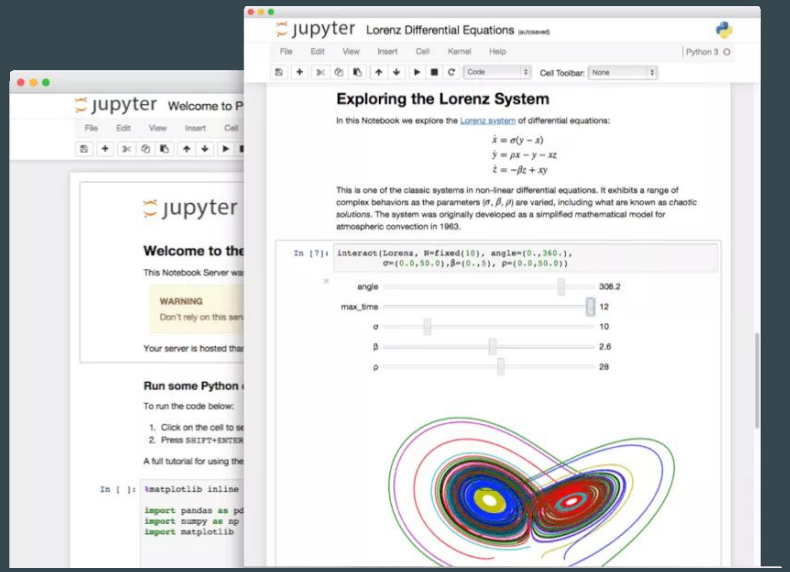


Jupyter Notebook / Lab

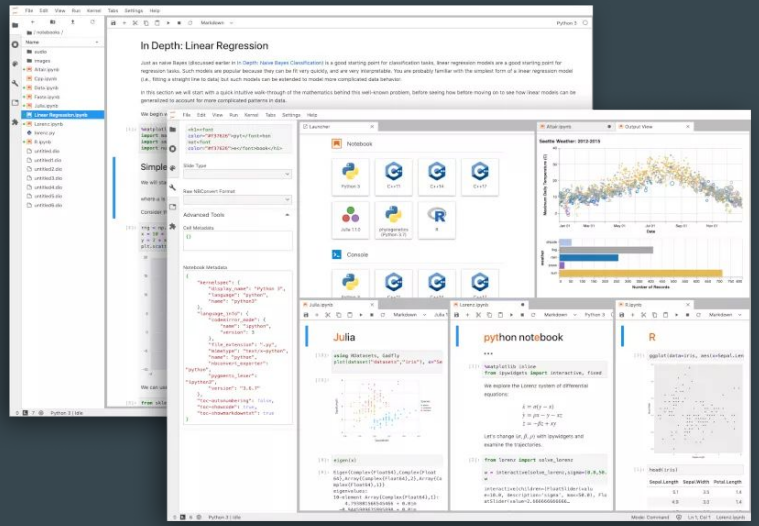
...

Project Jupyter

https://jupyter.org/



Jupyter Notebook



JupyterLab

Jupyter Notebook

4.1) Install

```
pip install jupyter
```

4.2) Setup

```
jupyter notebook --generate-config
```

⬇

~/.jupyter/jupyter_notebook_config.py

```
c.NotebookApp.ip = ''
c.NotebookApp.open_browser = False
c.NotebookApp.quit_button = False
c.NotebookApp.port = 50000 #+superai id(4 digits)
                          # valid port: 1024 - 65535)
```

4.3) Set Password

```
jupyter notebook password
```

Jupyter Notebook

4.4) Run

```
jupyter notebook
```

JupyterLab

4.5) Install and Run JupyterLab

```
pip install jupyterlab
```

```
jupyter lab
```

Jupyter Notebook

4.6) Run a notebook in background

```
jupyter nbconvert file.ipynb --to notebook --execute
```

Linux Workshop

•••

Q&A Session