



Teaching with JupyterHub - lessons learned



Prof. Martin Christen

FHNW – University of Applied Sciences and Arts Northwestern Switzerland
School of Architecture, Civil Engineering and Geomatics

Institute Geomatics

martin.christen@fhnw.ch



MartinChristen
GitHub: MartinChristen
GitLab: martin.christen

GitHub

<https://github.com/martinchristen/EuroSciPy-2018-JupyterHub>

<https://github.com/jupyterhub>

What am I teaching ? (Python related courses only)

Bachelor Programme

Introduction to Python (1st Semester)

Programming I (2nd Semester)

Programming II (3rd Semester)

Open (Geo-)Data Seminar (5th Semester)

Masters Programme

Geoprocessing (14 Lessons)

Where? Our New Campus (4500 People, 250M €, built in 4 years)



What am I teaching ? (Python related courses only)

Bachelor Programme

Introduction to Python (1st Semester)

Programming I (2nd Semester)

Programming II (3rd Semester)

Open (Geo-)Data Seminar (5th Semester)



Masters Programme

Geoprocessing (14 Lessons)

This course is about processing big geo data
Aerial images/Satellite images are large! (Petabyte range)

Problem #1

How can students store multiple 1-10 TB datasets?

Problem #2

Installing all required modules takes forever

About JupyterHub

Jupyterhub:
Jupyter for multiple users

- The Hub launches a proxy
- The proxy forwards all requests to the Hub by default
- The Hub handles **user login** and **spawns** single-user servers on demand
- The Hub configures the proxy to forward URL prefixes to the single-user notebook servers

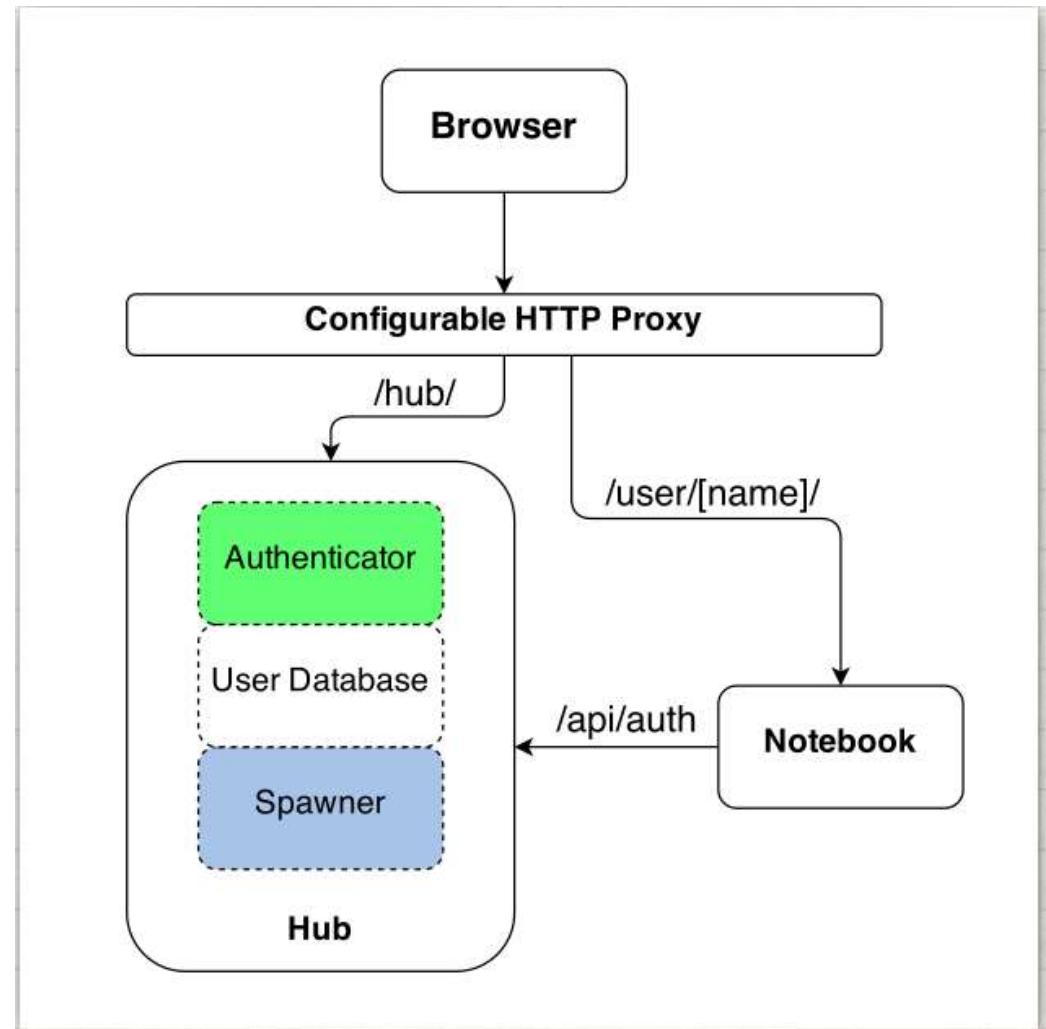


Image from <https://jupyterhub.readthedocs.io>

Authenticators – User Login

- **Local User**
- **OAuthenticator**: GitHub/BitBucket/MediaWiki/CILogon/Google OAuth/...
- Dummy Authenticator (For testing only! Not for production use.)
- **LDAP** Authenticator
- **Kerberos** Authenticator
- **REMOTE_USER** Authenticator (For when intermediate login infrastructure such as Apache offloads authentication and forwards REMOTE_USER header.)
- **JSONWebToken** Authenticator (For Authenticating/Authorising users based on a JSONWebToken forwarded in an Authorization: bearer <token> header.)
- **HashAuthenticator** (Each user comes with a predefined password which is a hash of their username. A separate authenticator may be used for admins.)
- **CASAuthenticator** for CAS Single Sign-on SSO

Spawners

- **BatchSpawner** for spawning remote servers using batch systems (Torque, PBS, Slurm, etc)
- **DockerSpawner**, which actually has two different spawners in it:
 - dockerspawner.DockerSpawner, for spawning identical Docker containers for each user
 - dockerspawner.SystemUserSpawner, for spawning Docker containers with an environment and home directory for each user
- **ImageSpawner** for allowing users to choose which Docker image to spawn.
- **KubeSpawner** for use with Kubernetes.
- **MarathonSpawner** for spawning instances on Marathon.
- **UCRSpawner** for spawning GPU instances on Marathon.
- **RemoteSpawner**
- **SimpleSpawner**, for testing purposes.
- **SystemdSpawner**, spawns notebooks with the isolation / security benefits of containers without the complexity of image management.
- **SudoSpawner** uses sudo. SystemdSpawner is recommended instead of SudoSpawner, if it is available to you.
- **WrapSpawner** allows spawner class and options to be chosen at runtime. Includes ProfilesSpawner for doing this from a spawner options form. Formerly part of BatchSpawner.

Installation

Installation is easy if you read the docs:

<https://jupyterhub.readthedocs.io/en/stable/installation-guide.html>

- You need Linux
- You can use Docker
- You can install Jupyterhub without Docker

Cluster? GPU?

<https://zero-to-jupyterhub-with-kubernetes.readthedocs.io/en/latest/index.html>

<https://github.com/gifford-lab/jupyterhub>

<https://ritazh.com/running-jupyter-notebook-and-tensorboard-on-gpu-on-azure-using-kubernetes-e4c43948e9a8>

<http://mathalope.co.uk/2017/09/25/how-to-setup-tensorflow-jupyter-notebook-on-intel-nervana-ai-cluster-colfax-for-deep-learning/>

...and much more...

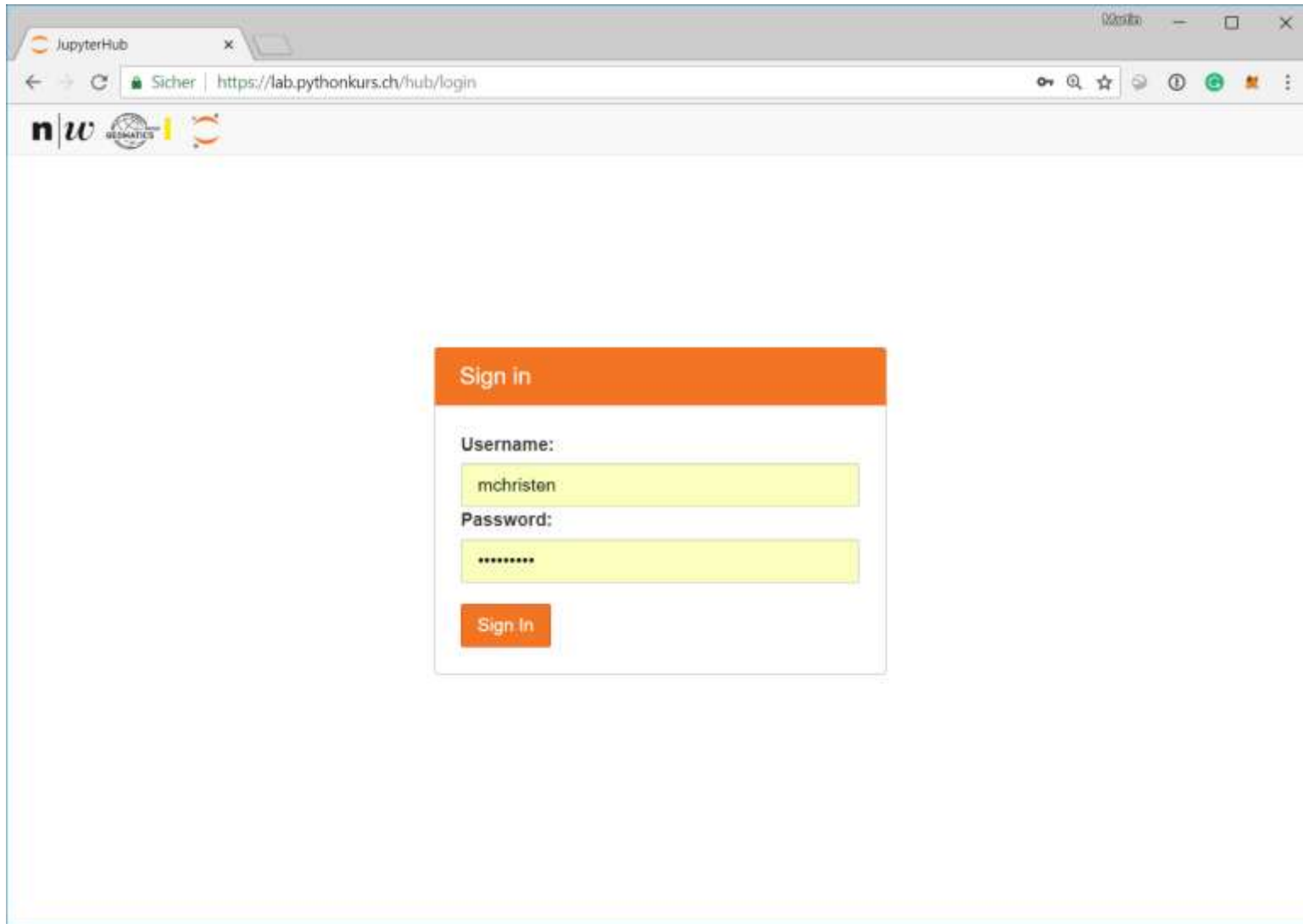
Configuration

/opt/anaconda3/bin/jupyterhub_config.py

```
# Custom Logo
c.JupyterHub.logo_file = "/path/to/logo.png"

# for Jupyter Lab
c.Spawner.cmd = ["jupyter-labhub"]
c.Spawner.default_url = '/lab'
c.Spawner.environment = {'JUPYTER_ENABLE_LAB': 'yes' }
```

Login to JupyterHub



The screenshot shows a web browser window with the title 'JupyterHub'. The address bar displays 'Sicher | https://lab.pythonkurs.ch/hub/login'. The page features the 'n|w' logo and a 'Sign in' button in the top left. The main content area contains a 'Sign in' form with an orange header. The form includes a 'Username:' label, a text input field containing 'mchristen', a 'Password:' label, a password input field with masked characters, and a 'Sign In' button.

JupyterHub

Sicher | https://lab.pythonkurs.ch/hub/login

n|w

Sign in

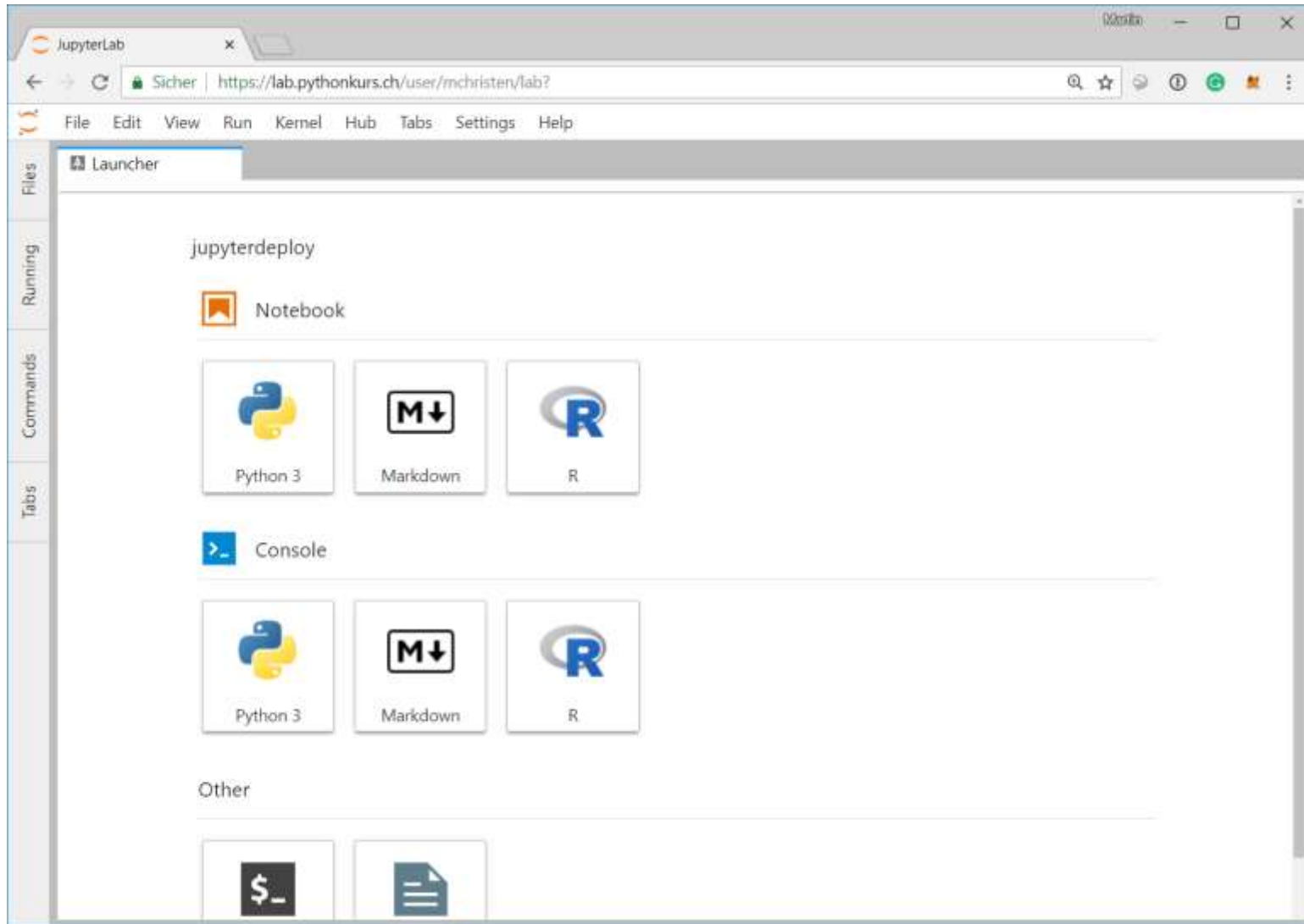
Username:

mchristen

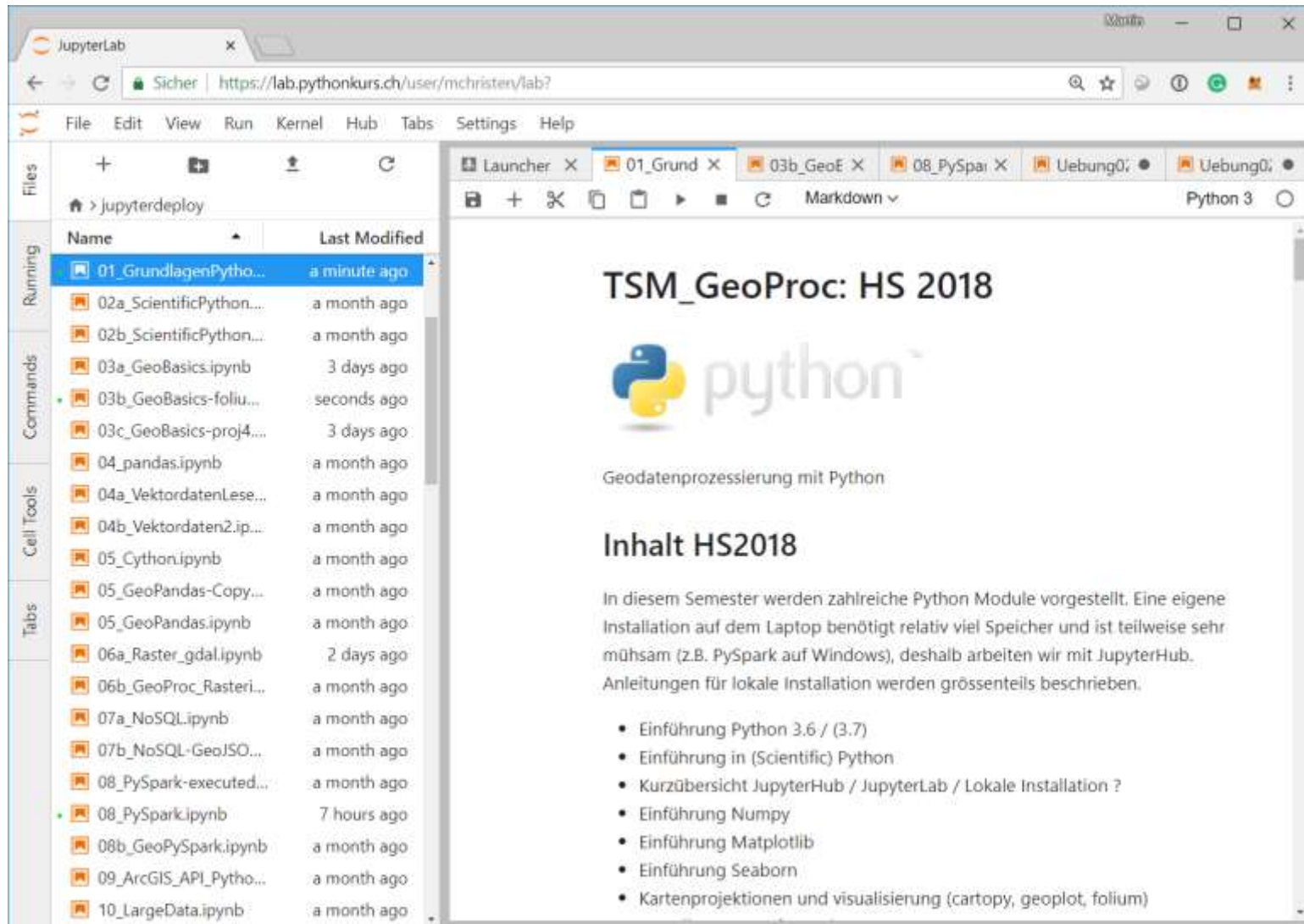
Password:

Sign In

Starting Jupyter Lab for the user



Login to JupyterHub



JupyterLab

File Edit View Run Kernel Hub Tabs Settings Help

Files

Running

Name	Last Modified
01_GrundlagenPytho...	a minute ago
02a_ScientificPython...	a month ago
02b_ScientificPython...	a month ago
03a_GeoBasics.ipynb	3 days ago
03b_GeoBasics-foliu...	seconds ago
03c_GeoBasics-proj4...	3 days ago
04_pandas.ipynb	a month ago
04a_VektordatenLese...	a month ago
04b_Vektordaten2.ip...	a month ago
05_Cython.ipynb	a month ago
05_GeoPandas-Copy...	a month ago
05_GeoPandas.ipynb	a month ago
06a_Raster_gdal.ipynb	2 days ago
06b_GeoProc_Rasteri...	a month ago
07a_NoSQL.ipynb	a month ago
07b_NoSQL-GeoISO...	a month ago
08_PySpark-executed...	a month ago
08_PySpark.ipynb	7 hours ago
08b_GeoPySpark.ipynb	a month ago
09_ArcGIS_API_Pytho...	a month ago
10_LargeData.ipynb	a month ago

Commands


Cell Tools

Tabs

Launcher X 01_Grund X 03b_GeoE X 08_PySpar X Uebung0; Uebung0;

Markdown Python 3

TSM_GeoProc: HS 2018



Geodatenprozessierung mit Python

Inhalt HS2018

In diesem Semester werden zahlreiche Python Module vorgestellt. Eine eigene Installation auf dem Laptop benötigt relativ viel Speicher und ist teilweise sehr mühsam (z.B. PySpark auf Windows), deshalb arbeiten wir mit JupyterHub. Anleitungen für lokale Installation werden grössenteils beschrieben.

- Einführung Python 3.6 / (3.7)
- Einführung in (Scientific) Python
- Kurzübersicht JupyterHub / JupyterLab / Lokale Installation ?
- Einführung Numpy
- Einführung Matplotlib
- Einführung Seaborn
- Kartenprojektionen und visualisierung (cartopy, geoplot, folium)

genusers.py - creating User/password for a specific course

The image shows two Notepad++ windows. The left window displays the 'gen_users.py' script, which is a Python program designed to generate user accounts. It includes imports for 'crypt', 'os', 'string', and 'random', and uses 'collect' for password generation. The script has a main loop that prompts the user for a host name and a number of users to generate, then iterates through the 'userlist_fm.txt' file to create each user. The right window shows the 'userlist_fm.txt' file, which contains a list of 29 entries, each consisting of a username (e.g., 'fm0', 'fm1') and a password (e.g., 'fqvrzk', 'zrjzks').

```

1  #!/opt/anaconda3/bin/python
2
3  #Python Script
4
5  import crypt
6  import os
7  import string
8  import random
9  from collections import Counter
10
11  if os.getuid() != 0:
12      print("This script must be run as root")
13      exit(0)
14
15  def createUser(username, password):
16      encPass = crypt.crypt(password, '$6$' + string.ascii_letters + string.digits + string.punctuation)
17      os.system('useradd -s /bin/bash ' + username)
18      os.system('echo ' + username + ':' + encPass | chpasswd')
19
20  def idgenerator():
21      chars = string.ascii_letters + string.digits + string.punctuation
22      return ''.join(random.choice(chars) for _ in range(8))
23
24  s = input("Host name: ")
25
26  try:
27      num = int(input("Number of users to generate: "))
28  except:
29      print("Error: Invalid input")
30      exit(0)
31
32  batch = idgenerator()
33
34  userlist_fm.txt

```

userlist_fm.txt

```

1  fm0, fqvrzk
2  fm1, zrjzks
3  fm2, ujprku
4  fm3, eyibym
5  fm4, dywnpt
6  fm5, msjrjt
7  fm6, tvvyau
8  fm7, xmrenu
9  fm8, jzeplo
10 fm9, nbjmei
11 fm10, tkgvjq
12 fm11, xpxmvz
13 fm12, jmnadq
14 fm13, hwavgk
15 fm14, yezkyi
16 fm15, addxhe
17 fm16, nighqb
18 fm17, pnppog
19 fm18, pztigl
20 fm19, ccljhh
21 fm20, qosqpg
22 fm21, xogvoy
23 fm22, iwndkb
24 fm23, varkiv
25 fm24, ehetsm
26 fm25, ckknnf
27 fm26, sfxpec
28 fm27, ivuzpd
29 fm28, taecel

```

Python file | leng Normal text file length : 380 lines : 31 | Ln : 31 Col : 1 Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

deploy_coursename.py

During the course:

- **Every week** there is a new (or serveral new) Jupyter Notebooks available and an Exercise Notebook.
- The notebooks are deployed to all students 72h before the next lesson
- Also included is a general solution for the exercise
- I created a simple deploy script which copies new files to all course members (using the course id created in genusers.py).
- Also directories with all contents can be copied (for small datasets, pdf, ...)

deploy_coursename.py

```

C:\Users\martin.christen\Documents\GitHub\EuroSciPy-2018-JupyterHub\scripts\deploy_geoproc.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
deploy_geoproc.py
1  #!/opt/anaconda3/bin/python
2
3  from sys import argv
4  from shutil import copyfile
5  from os import chmod
6  import os
7
8  DEPLOYDIR = "/home/"
9  COURSESHORTCUT = "fm"
10 USERCOUNT = 24
11
12 if len(argv) == 1 or len(argv)>3:
13     print("wrong usage!")
14     print("deploy.py file")
15     print("deploy.py dir -d")
16     exit(0)
17
18
19 directory = False
20
21 if len(argv) == 3 and argv[2]=="-d":
22     print("copy directory...")
23     directory = True
24
25 filename = argv[1]
26 coursename = COURSESHORTCUT
27 users = USERCOUNT
28
29 for i in range(users):
30     username = coursename+str(i)
31     dest = DEPLOYDIR + username + "/" + filename
32     print("copy " + filename + " to " + dest)
33     #copyfile(filename, dest)
34     if directory:

```

Python file length: 1'191 lines: 46 Ln: 10 Col: 15 Sel: 0 | 0 Unix (LF) UTF-8 INS

Large Data Files

Data is just located globally at /data/ and all users have read access.

This way multi TB datasets can be read by all without problems

Example: `/data/landsat.tif`

Previously I tried to use symbolic links but it didn't really work well

Cluster for JupyterHub



JupyterHub

124 CPU Cores

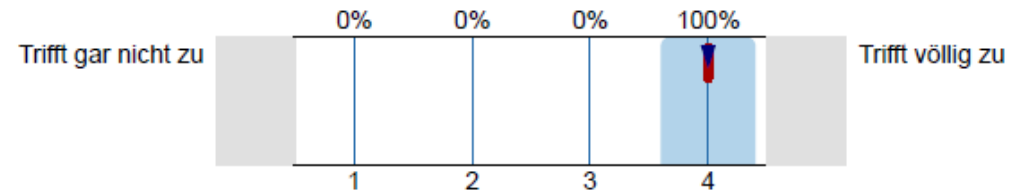
112 GB RAM

120 TB HD (~1 GB/s read/write)

currently only 4 GPUs, to be increased soon

Students feedback

- 2.7) Ihr/sein Unterricht ist klar strukturiert (roter Faden) und der Unterrichtsstoff wurde verständlich vermittelt.



«His teaching is clearly structured (central theme) and the subject matter was communicated in an understandable way.»

«Working with Jupyter(hub) was great and made everything much easier»

«Using Jupyterhub saved me much time»

«Programming using Jupyter(hub) was much fun. I wish other modules had something similar»

Conclusion & Outlook

- JupyterHub solves my major problems:
 - Doing exercises with really large datasets (1-10 TB)
 - No installation of modules / No time wasted in first lesson
 - Students don't need space on their laptop for anaconda/modules/data (if they don't want to install it locally)
 - I don't have to write installation instructions for each OS / Python Version
 - GPUs can be used even if students don't have a high-end graphics card.
 - Students like it

In future:

- Add more GPUs (GPU cluster in 2nd Rack)
- Also use JupyterHub for Research Projects

GeoPython 2019



Muttenz Switzerland
Basel

June 24-26

- Python in General
- GIS/Mapping
- Geography / Geophysics / Geodesy / Geomatics
- Earth Sciences / Environmental Sciences
- Geovisualization
- Smart Cities
- Spatial Data / Geodata
- Geospatial Webservices
- Big Data
- Data Processing
- (Spatial) Databases
- Computer Vision
- Remote Sensing
- Image Processing
- (Geo-)Machine Learning / (Geo-)Deep Learning

<http://2019.geopython.net>



@GeoPythonConf



Questions

