

How to use Jupyter and Tensorflow on Izar

This is a short guide to describe how to use jupyter notebook on Izar through ssh port forwarding.

Installing IPython and Jupyter

The key to successfully installing the tools is using python virtual environments.

1. Load the compiler you want to use and python using modules

```
$ module load gcc
$ module load python
```

```
$ module load intel
$ module load python
```

2. Create a virtual environment

In the following, we use GCC but the same procedure applies to Intel compiler. In your home folder, create a virtual environment:

```
$ virtualenv -p python3 --system-site-packages opt/venv-gcc

Running virtualenv with interpreter /ssoft/spack/arvine/v1/opt/spack/linux-rhel7-skylake_avx512/gcc-8.4.0/python-3.7.7-drpdlwdb03lmtkcbckq227ypnzno4ek3/bin/python3
Already using interpreter /ssoft/spack/arvine/v1/opt/spack/linux-rhel7-skylake_avx512/gcc-8.4.0/python-3.7.7-drpdlwdb03lmtkcbckq227ypnzno4ek3/bin/python3
Using base prefix '/ssoft/spack/arvine/v1/opt/spack/linux-rhel7-skylake_avx512/gcc-8.4.0/python-3.7.7-drpdlwdb03lmtkcbckq227ypnzno4ek3'
New python executable in /home/user/opt/venv-gcc/bin/python3
Also creating executable in /home/user/opt/venv-gcc/bin/python
Installing setuptools, pip, wheel...
done.
```

3. Activate virtual environment

```
$ source opt/venv-gcc/bin/activate

(venv-gcc) [user@izar ~]$
```

4. Install Jupyter

```
(venv-gcc) [user@izar ~]$ pip install jupyter
Collecting jupyter
..
..
```

5. Set passwordless access to Izar by using ssh key and have the following in your ~/.ssh/config file on your personal computer ([How to generate an SSH key](#)). It is assumed that you are inside the EPFL network (or connected through VPN).

```
Host izar
  Hostname izar.epfl.ch
  User [username]

Host i*.izar
  ProxyCommand ssh -W $(echo "%h" | sed -e 's/.izar//'):%p izar
```

6. Run jupyter notebook on Izar.

The script below is a template that allows you to start jupyter notebook on a compute node of Izar. You can copy it in a file called `launch_jupyter.sh` for example. It has to be placed in your home (or you have to modify it accordingly). Note that all the modules that jupyter may need have to be loaded here. As an example, here, we loaded the modules that allow us to use TensorFlow in the notebook.

launch_jupyter.sh

```
#!/bin/bash -l
#SBATCH --job-name=ipython-trial
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --exclusive
#SBATCH --time=01:00:00
#SBATCH --output jupyter-log-%J.out

module load gcc mvapich2 py-tensorflow

source opt/venv-gcc/bin/activate

ipnport=$(shuf -i8000-9999 -n1)

jupyter-notebook --no-browser --port=${ipnport} --ip=$(hostname -i)
```

Launch your job as usual:

```
sbatch launch_jupyter.sh
```

Once the job is running analyze the output jupyter-log-[SLURM_ID].out. Then, look for a line like the following:

```
Or copy and paste one of these URLs:
    http://10.91.27.63:8504/?token=4b17ae5cfa505b5470dc84bb5240ab43ae714aa9480a163c
```

It has the form:

```
http://<IP ADDRESS>:<PORT NUMBER>/?token=<TOKEN>
```

7. On your local machine do the following with the information provided by the above step

```
ssh -L <PORT NUMBER>:<IP ADDRESS>:<PORT NUMBER> -l <USERNAME> izar.epfl.ch -f -N

## Or if you created the ssh config from step 5
ssh -L <PORT NUMBER>:<IP ADDRESS>:<PORT NUMBER> izar -f -N
```

For our example, this gives:

```
ssh -L 8504:10.91.27.63:8504 -l user izar.epfl.ch -f -N

## with config
ssh -L 8504:10.91.27.63:8504 izar -f -N
```

8. Now you should be able to access to Izar compute node through the web browser by pasting the following address

```
http://localhost:<PORT NUMBER>/?token=<TOKEN>
```

For our example, this gives:

```
http://localhost:8504/?token=4b17ae5cfa505b5470dc84bb5240ab43ae714aa9480a163c
```

Here is a screenshot of a jupyter session running a simple TensorFlow example. The computations are run on the cluster node, but you access the results directly from your browser.

```

In [11]: izarp tensorflow as tf
In [12]: !cat /etc/passwd
Thu Oct 29 05:19:32 2020
-----
| NVIDIA-SMI 450.26.05   Driver Version: 450.26.05   CUDA Version: 11.3   |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
| GPU Name      Persistence-M| Bus-Id      Disp.# | Voltage Mode    | Part-ID      | Serial          | GPU-Memory   |
| Fan Temp  Perf  Pwr  Usage% | Memory-Usage |                               | GPU-Part     | GPU-Memory   | Part-ID        |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
| 0 Tesla V100-A112  On       | 00000000:00:04:0 | Off | On/Default      | 1048576352   |                | 1048576352   | 1048576352   | 1048576352   |
| 1 Tesla V100-A112  On       | 00000000:00:04:1 | Off | On/Default      | 1048576352   |                | 1048576352   | 1048576352   | 1048576352   |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
| Processes: |
| GPU  PID  Type  Process name                               | GPU Memory |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
| No running processes found
-----

In [13]: !pip install tensorboard
Collecting tensorboard
In [12]: izarp tensorflow as tf
izarp database
In [17]: !clear ipynb logs previous runs
rm -rf *.ipynb
In [14]: main = tf.keras.callbacks.main
@train, @test, @ eval, @ test) = main.load_data()
@ train, @ test = @ train / 250, @ test / 250
def create_model():
    return tf.keras.Sequential([
        tf.keras.layers.Flatten(input_shape=(28, 28)),
        tf.keras.layers.Dense(128, activation='relu'),
        tf.keras.layers.Dense(128, activation='relu'),
        tf.keras.layers.Dense(10, activation='softmax')
    ])
In [15]: model = create_model()
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1)
model.fit(x=train,
          y=train_labels,
          validation_data=(test, test_labels),
          callbacks=[tensorboard_callback])

```

If you wish to use tensorboard you must open a second connection as follow

```

ssh -L <PORT NUMBER>localhost:<PORT NUMBER> -o ProxyCommand="ssh -W %h:%p izar.epfl.ch" -l <USERNAME> <NODE ALLOCATED>

## OR if you created the config from step 5
ssh -L <PORT NUMBER>localhost:<PORT NUMBER> <NODE ALLOCATED>.izar

```

For example

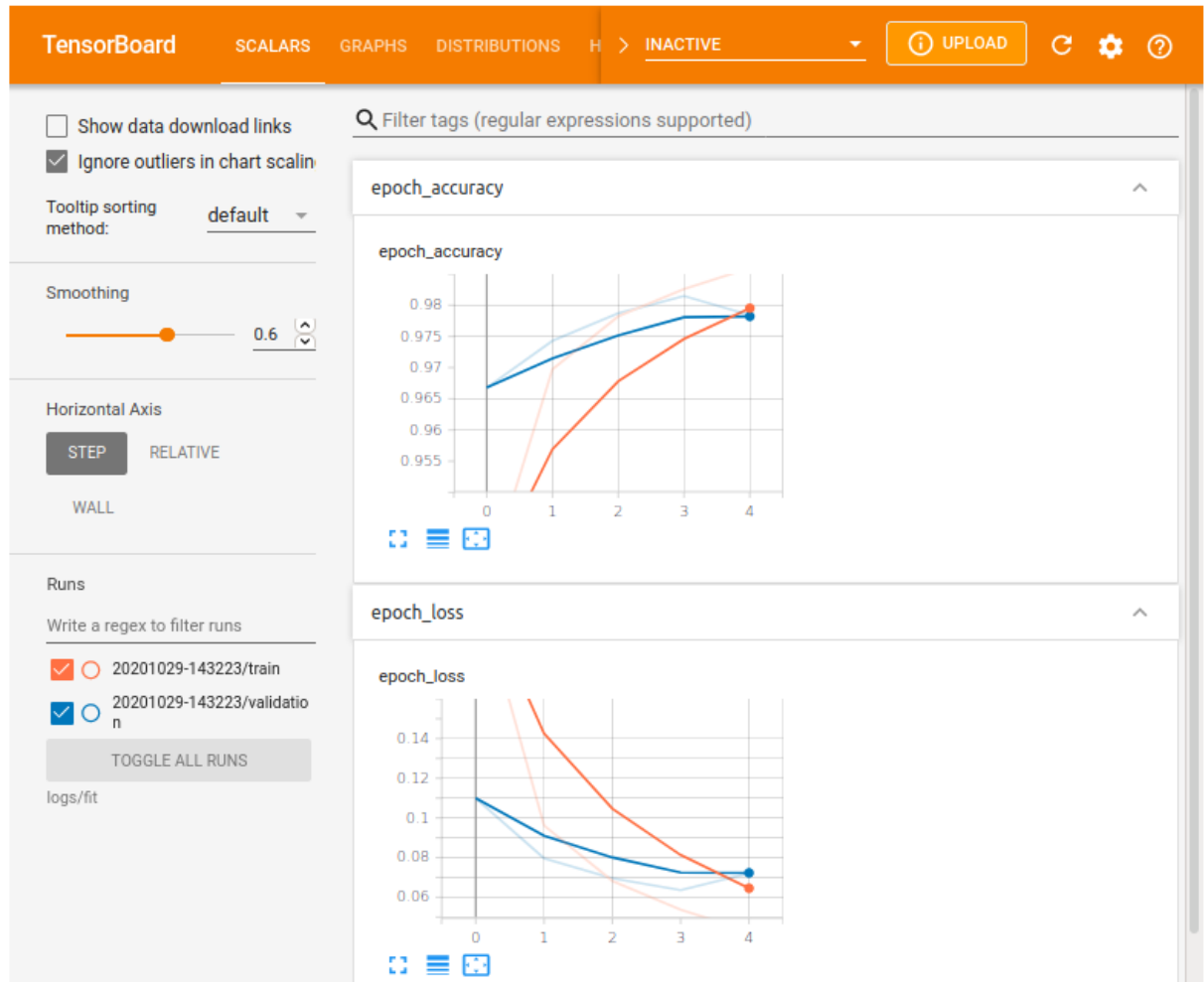
```

ssh -L 6006:localhost:6006 -o ProxyCommand="ssh -W %h:%p izar.epfl.ch" -l user i39

## with config
ssh -L 6006:localhost:6006 i39.izar

```

```
In [21]: %tensorboard --logdir logs/fit --port 6006
```



Related articles

- [Data Transfer Nodes](#)
- [Managing a lab account](#)
- [Account management](#)
- [IMOD on Izar](#)
- [Cryosparc on Izar](#)