hosted singularity-hub

# Jupyter and Singularity

**Updated: 25.11.2019, new singularity version 3.5**
**Containers are on singularity-hub now: MyCollections**

Jupyter Miniconda Python 3 and Singularity Container

This is an update from  the offical jupyter singularity container that requires root permissions to run:

- [NEW] Only need root permissions to create the container
- [NEW] Miniconda (smaller in size)
- [NEW] runscript gives informaiton
- [NEW] Using CentOS 6.10 not Ubuntu anymore
- [NEW] GLIBC 2.12 compatibility to CentOS 6.10 (Final)

If you haven't installed singularity, do that with these instructions.

1. Downlaod Receipie files:
   - Singularity.centos (Base only Centos)
   - Singularity.jupyter23 (Miniconda, Jupyter Python2 & Python 3)
   - Singularity.jupyter3 (Miniconda, Jupyter Python 3)
   - Singularity.jupyter3x (Miniconda, Jupyter Python 3, Eccodes, cfgrib from ECMWF)
   - Singualrity.jupyter3ec (Miniconda, Jupyter Python 3, Eccodes library manual build, **deprecated**)
   - Singualrity.jupyter3rttov (Miniconda, Jupyter Python 3, RTTOV from EUMETSAT (not included due to license))

   Clone the Repository and manually build containers:

```
git clone https://github.com/MBlaschek/singularity-jupyter jupyter
cd jupyter
```

2. Retrieve Containers from singularity hub:

```
singularity pull shub://MBlaschek/singularity-jupyter:[TAG]
```

Tags are the names above (centos, jupyter23, jupyter3, ...):

```
singularity pull shub://MBlaschek/singularity-jupyter:centos
```

## CREATE

First create the CentOS container that is used by all the others.

```
sudo singularity build centos610.sif Singularity.centos
```

Let's now create the notebook container:
If you build locally, then just edit the Recipie to use the local image

```
# Local centos 6.10 image
Bootstrap: localimage
From: centos610.sif
# Bootstrap: shub
# From: MBlaschek/singularity-jupyter:centos
# most recent and debian image
# BootStrap: docker
# From: continuumio/miniconda3
```

- Jupyter Python 3 Notebook Container: `Singularity.jupyter3`
- Jupyter Python 2 & 3 Notebook Container: `Singularity.jupyter23`
- Jupyter Python 2 & 3 Notebook + Eccodes Library: `Singularity.jupyter3x` (depends on the image from `jupyter3.sif`)

You can choose now if you prefer a writeable container (for development, installation of additional packages, ...) or a deployment container (read_only, default) [read more]:

```
sudo singularity build --writeable jupyter3.sif Singularity.jupyter3
```

or for deployment:

```
sudo singularity build jupyter3.sif Singularity.jupyter3
```

The Notebook server Recipies include a line at the end that is quite important for jupyter to run properly:

```
  export JUPYTER_RUNTIME_DIR=$PWD/.runtime
```

This line tells jupyter to use a specific directory for its runtime. Otherwise it would try to use the default `XDG_RUNTIME_DIR`, which is by default set to `/run/user/...` and not accessable via the container.

# RUN

Then to run the container:

```
singularity run jupyter3.sif
```

gives Information on the container and it's apps (notebook)

```
Singularity Container
    Container Centos 6.10 (docker)
    Glibc: 2.12-1.212.el6.x86_64
    Installed: wget, git, curl, bzip2 ca-certificates

  SCIF (Apps):  notebook
                                        Container.Glibc : 2.12-1.212.el6.x86_64
                                           Container.OS : CentOS 6.10
                                      Definition.Author : M. Blaschek
                                 Definition.Author.Email :
michael.blaschek@univie.ac.at
                                    Definition.File.Date : 5.11.2019
```

```
                                Definition.File.Version : 1.0
                            org.label-schema.build-date :
Thursday_28_November_2019_8:49:15_UTC
                        org.label-schema.schema-version : 1.0
                                 org.label-schema.usage :
/.singularity.d/runscript.help
  org.label-schema.usage.singularity.deffile.bootstrap : shub
       org.label-schema.usage.singularity.deffile.from : MBlaschek/singularity-
jupyter:centos
     org.label-schema.usage.singularity.runscript.help :
/.singularity.d/runscript.help
            org.label-schema.usage.singularity.version : 3.4.2
  Bye Bye
```

launch the notebook:

```
sudo singularity run jupyter3.sif notebook
```

launch the console:

```
sudo singularity run jupyter3.sif ipython
```

or as a singularity instances (background server):

```
singularity instance start jupyter3.sif Jupy3
singularity run instance://Jupy3 notebook
```

or as an instance with remote access (default is just localhost):

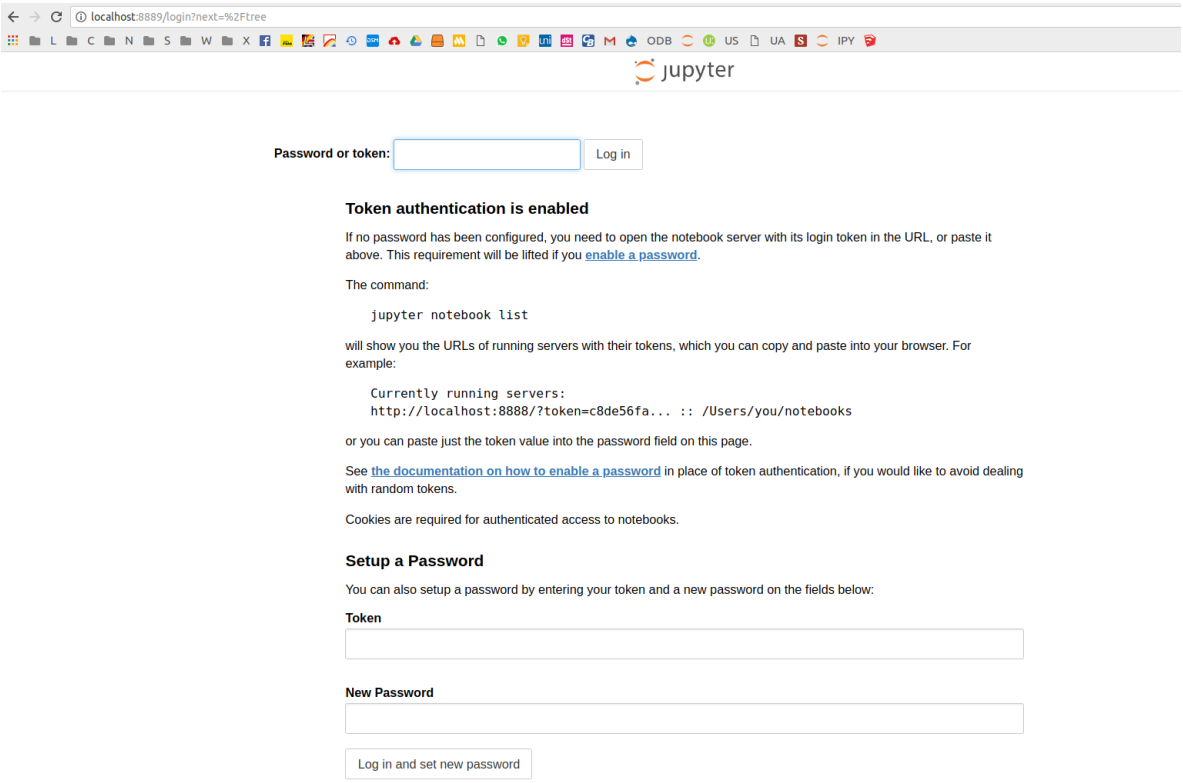
```
singularity run instance://Jupy3 notebook --ip=$(hostname)
```

Anyway you should see output like this:

```
mblaschek@nb18:~/workspace$ singularity run jupyter3.img
Starting notebook...
[W 13:06:35.366 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 13:06:35.367 NotebookApp] The port 8888 is already in use, trying another port.
[I 13:06:35.376 NotebookApp] Serving notebooks from local directory: /home/mblaschek/workspace
[I 13:06:35.376 NotebookApp] 0 active kernels
[I 13:06:35.376 NotebookApp] The Jupyter Notebook is running at:
[I 13:06:35.376 NotebookApp] http://nb18:8889/?token=...
[I 13:06:35.376 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[I 13:06:54.447 NotebookApp] 302 GET /login?next=%2Ftree (::1) 1.51ms
[I 13:08:13.180 NotebookApp] Creating new notebook in
[I 13:08:14.300 NotebookApp] Kernel started: 2709b2f0-73a1-4ec6-a1d0-c5d382e93ace
[I 13:08:15.198 NotebookApp] Adapting to protocol v5.1 for kernel 2709b2f0-73a1-4ec6-a1d0-c5d382e93ace
[I 13:10:14.264 NotebookApp] Saving file at /Untitled.ipynb
[I 13:11:51.797 NotebookApp] New terminal with automatic name: 1
[I 13:12:14.920 NotebookApp] Saving file at /Untitled.ipynb
[I 13:30:10.649 NotebookApp] EOF on FD 36; stopping reading
[I 13:30:10.750 NotebookApp] Terminal 1 closed
[I 13:30:14.874 NotebookApp] Saving file at /Untitled.ipynb
[I 13:31:34.158 NotebookApp] Saving file at /Untitled.ipynb
[I 13:31:35.944 NotebookApp] Starting buffering for 2709b2f0-73a1-4ec6-a1d0-c5d382e93ace:444ba30af65841c39b23a0003103301a
[I 13:31:36.353 NotebookApp] Kernel shutdown: 2709b2f0-73a1-4ec6-a1d0-c5d382e93ace
[I 13:31:39.714 NotebookApp] Shutting down on /api/shutdown request.
[I 13:31:39.715 NotebookApp] Shutting down 0 kernels
```

The current directory is where your server starts. In your browser you should be able to navigate to the link from the console:




There is a `.jupyter3.log` file that shows this output.

The password is **super-secret**. You can change that easily within the Singularity file.

# IPYKernel and Containers

In order to use your container with an existing notebook server you need to register your container kernel with that server.
Other people have done this:

- Tensorflow
- Kernel

To register your container, in the `${HOME}/.local/share/jupyter/kernels` create a new directory, e.g. myimage, and add a `kernel.json` file containing:

```
{
 "language": "python",
 "argv": ["/usr/bin/singularity",
   "exec",
   "/dir/to/your/image/jupyter3.sif",
   "/opt/conda/bin/python",
   "-m",
   "ipykernel",
   "-f",
   "{connection_file}"
 ],
 "display_name": "Python 3 (Singularity)"
}
```

```
```

Change the path to your image and singularity executable. Then start a jupyter notebook with

```
  jupyter notebook &
```

and there should be a usable Python 3 (Singularity) kernel option! Check your Jupyter paths, like this:

```
  jupyter --paths

  config:
      /home/user/.jupyter
      /opt/anaconda2/etc/jupyter
      /usr/local/etc/jupyter
      /etc/jupyter
  data:
      /home/user/.local/share/jupyter
      /opt/anaconda2/share/jupyter
      /usr/local/share/jupyter
      /usr/share/jupyter
  runtime:
      /run/user/1000/jupyter
```

and make sure the runtime directory is accessable from inside the container. In this example it isn't. There I need to change this to something like this, before I run the server again:

```
  export JUPYTER_RUNTIME_DIR=$HOME/.local/share/jupyter/runtime
  jupyter notebook &
```

That should solve the issue and make your contained jupyter environment accessable via your notebook server. :)

## RUNTIME DIR

I came across a few problems, which related to the `RUNTIME_DIR` and is quite import to run your server without root permissions.

```
  XDG_RUNTIME_DIR=/run/user/1000    # Default in Ubuntu/Linux (inside the
container)
```

That is not a good path. Therefore we change it to a defined path inside the container (already in the singularity file).
The following shows a way around, not necessary if you use the above recipe.

This directory `/run/user/..` is not accessable by default from inside the container.
To register your container, in the `${HOME}/.local/share/jupyter/kernels` create a new directory, e.g. myimage, and add a `kernel.json` file containing:

```
{
 "language": "python",
```

```
 "argv": ["/usr/bin/singularity",
   "exec",
   "-B",
   "/run/user:/run/user",
   "/dir/to/your/image/jupyter.img",
   "/opt/conda/bin/python",
   "-m",
   "ipykernel",
   "-f",
   "{connection_file}"
 ],
 "display_name": "Python 3 (Singularity)"
}
```

where adding the `-B /run/user:/run/user` option is important, which allows the container to have access.

# R-Studio Server

This is a lightly modified version of what nickjer has done. The Modifications allow to run the R-Studio server as an instance.

```
singularity instance start rserver.sif RStudio
```

Usually the R-Studio server runs on port 9090.