Lesson 1.1.

Getting Started

This tutorial provides detailed installation instructions for downloading and setting up GridAPPS-D, the Python notebooks, and their dependencies. The Windows installation requires considerably more steps, but provides a fully functional development environment with the ability to run Windows and Linux tools simultaneously without setting up a virtual machine.

GridAPPS-D™ is an open-source platform that accelerates development and deployment of portable applications for advanced distribution management and operations.

The GridAPPS-D[™] project is sponsored by the U.S. DOE's Office of Electricity, Advanced Grid Research. Its purpose is to reduce the time and cost to integrate advanced functionality into distribution operations, to create a more reliable and resilient grid.

GridAPPS-D enables standardization of data models, programming interfaces, and the data exchange interfaces for:

- · devices in the field
- · distributed apps in the systems
- · applications in the control room

The platform provides

- · robust testing tools for applications
- · distribution system simulation capabilities
- · standardized research capability
- · reference architecture for the industry
- · application development kit

Table of Contents

- 1. Windows 10
 - 1.1. System Requirements
 - 1.2. Verify OS Build
 - 1.3. Install Git For Windows
 - 1.4. Install Anaconda or Miniconda
 - 1.5. Install Jupyter Lab
 - 1.6. Install GridAPPSD-Python
 - 1.7. Download Python Notebooks
 - 1.8. Install Windows Subsystem for Linux
 - 1.8.1. Enable WSL
 - 1.8.2. Upgrade to WSL2
 - 1.8.3. Install Linux Ubuntu OS

- 1.8.4. Set up Ubuntu in WSL
- 1.9. Install Docker for Windows
- 1.10. Install GridAPPS-D Platform
- 2. Ubuntu Linux
 - 2.1. System Requirements
 - 2.2. Install python-pip
 - 2.3. Install Anaconda or Miniconda
 - 2.4. Install Jupyter Lab
 - 2.5. Install GridAPPSD-Python
 - 2.6. Download Python Notebooks
 - 2.7. Install GridAPPS-D Platform

1. Windows 10

A typical Windows 10 installation does not include several of the tools needed to run the GridAPPS-D Platform and Python Notebook lessons. Several software packages need to be installed prior to starting this course.

Note: The download size is quite large, so it is strongly recommended to use a fiber or ethernet interent connection, rathered than a metered hotspot to avoid excessive data usage charges.

1.1. System Requirements

- OS:
 - Windows 10 Home, Pro, Education, or Enterprise
 - For x64 systems: Version 1903 or higher, with Build 18362 or higher.
 - For ARM64 systems: Version 2004 or higher, with Build 19041 or higher.
- RAM:
 - 8GB (absolute minimum for 13 and 123 node models, may encounter memory overload during installation);
 - 16GB (preferred for small models, minimum for 9500 node model);
 - 32GB (recommended)
- Disk Space:
 - 15GB required for installation

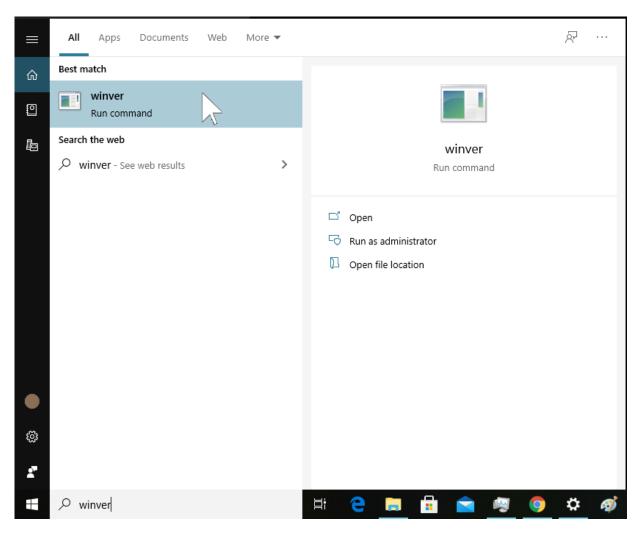
Installation Steps:

- 1.2. Verify OS Build
- 1.3. Install Git For Windows
- 1.4. Install Anaconda or Miniconda
- 1.5. Install Jupyter Lab
- 1.6. Install GridAPPSD-Python
- 1.7. Download Python Notebooks
- 1.8. Install Windows Subsystem for Linux
 - 1.8.1. Enable WSL
 - 1.8.2. Upgrade to WSL2
 - 1.8.3. Install Linux Ubuntu OS
 - 1.8.4. Set up Ubuntu in WSL

- 1.9. Install Docker for Windows
- 1.10. Install GridAPPS-D Platform

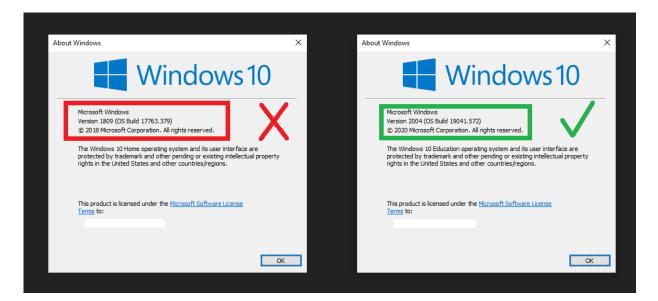
1.2. Verify OS Build

To check your OS build, type winver in the Cortana seach bar:



Check to see if your OS is

- For x64 systems: Version 1903 or higher, with Build 18362 or higher.
- For ARM64 systems: Version 2004 or higher, with Build 19041 or higher.



If not, run **Windows Update** to get the latest verion of Windows 10 available for your machine. It may take some time for the new OS to download. Multiple restarts are typical while upgrading the windows version.

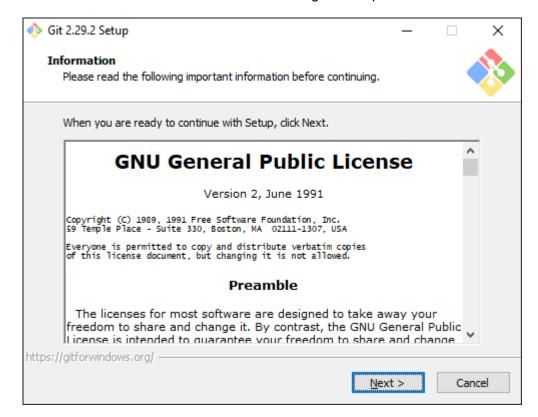
Return to Top

1.3. Install Git for Windows

Install git for windows. This package is required to download and run the python notebooks.

Open <u>gitforwindows.org (https://gitforwindows.org/)</u> and download the latest version.

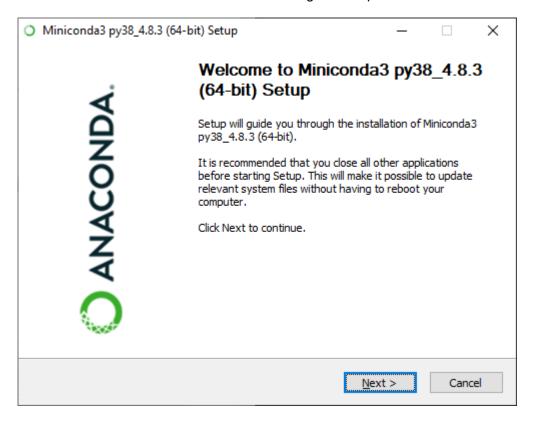
Use the installation wizard with the recommended settings to complete installation.



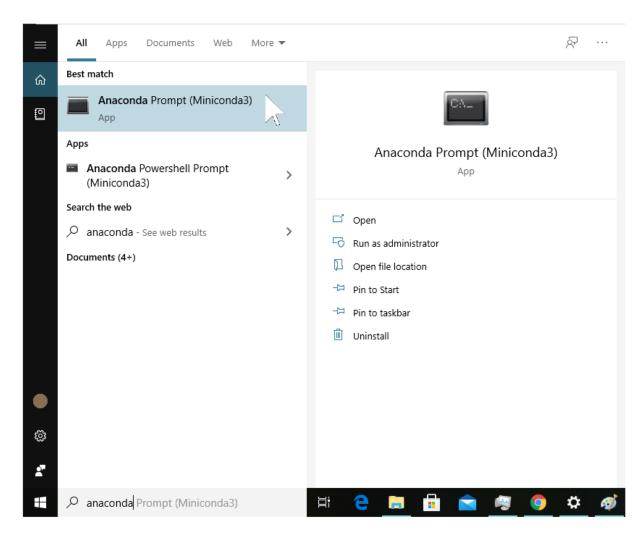
1.4. Install Anaconda or Miniconda

Download the latest version of the Miniconda for Python 3.8 from the <u>Conda.io website</u> (<u>https://docs.conda.io/en/latest/miniconda.html</u>).

Use the installation wizard with the recommended settings to complete installation.



After installation is complete, launch the **Anaconda Prompt (Miniconda3)** from the Start Menu or by typing anaconda in the Cortana toolbar



The miniconda terminal window will open



Return to Top

1.5. Install Jupyter Lab

In the miniconda terminal window, run <code>pip install jupyterlab</code> to install the Jupyter environment for executing the python notebooks. It may take a couple minutes to collect and install all the required packages.



Return to Top

1.6. Install GridAPPSD-Pvthon

In the Miniconda terminal window, download and install GridAPPSD-Python by running

pip install git+https://github.com/GRIDAPPSD/gridappsdpython.git@develop#egg=gridappsd

to download the GridAPPSD-Python library and required packages.

```
Anaconda Prompt (Miniconda3)

— X

(base) C:\Users\pip install git+https://github.com/GRIDAPPSD/gridappsd-python.git@develop#egg=gridappsd_
```

GridAPPSD-Python and all dependencies should have been automatically added to your anaconda path after completion.

Return to Top

1.7. Download Python Training Notebooks

In the miniconda terminal window, clone the python notebooks by running git clone https://github.com/GRIDAPPSD/gridappsd-hackathon to download the python training notebooks.



By default, the notebooks will be saved in the directory C:\Users\username\gridappsd-hackathon

Close the miniconda terminal

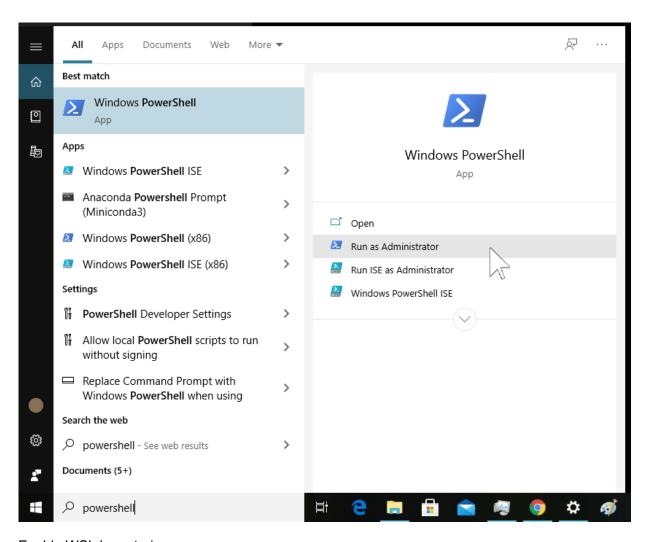
Return to Top

1.8. Install Windows Subsystem for Linux

GridAPPS-D and the associated docker containers will run using the Windows Subsystem for Linux (WSL), which is a new feature to Windows 10 that enables linux code to run natively in Windows without a separate virtual machine. The steps in this section are also available on the Microsoft website (https://docs.microsoft.com/en-us/windows/wsl/install-win10)

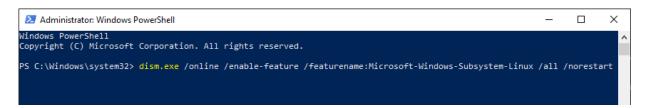
1.8.1. Enable WSL

Open Windows PowerShell as an administrator:



Enable WSL by entering

dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart



Then, without restarting, enable the virtual machine platform by entering

dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all
/norestart

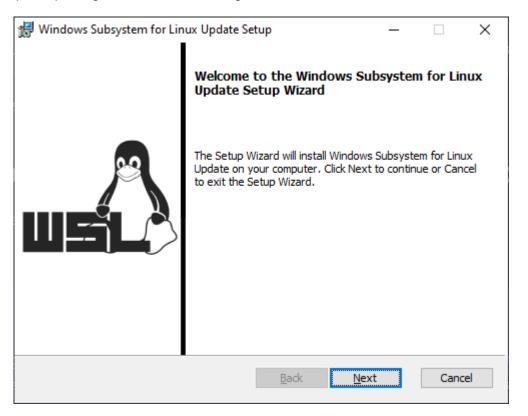
When completed, restart your machine. It may take a few minutes for the new settings to be applied while restarting.

Return to Top

1.8.2. Upgrade to WSL2

Download the latest WSL2 package .msi installer from the <u>Microsoft repository</u> (https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi)

Run the update package to install WSL2 using the wizard:



Open Windows PowerShell again and update the settings to use WSL2 by entering

wsl --set-default-version 2

```
Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

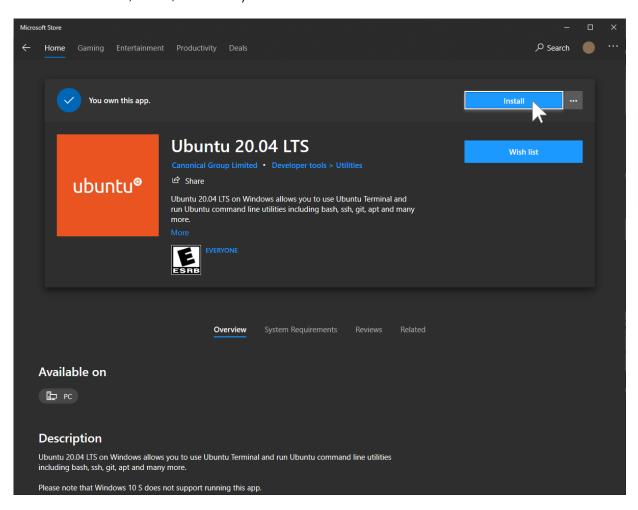
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> wsl --set-default-version 2_
```

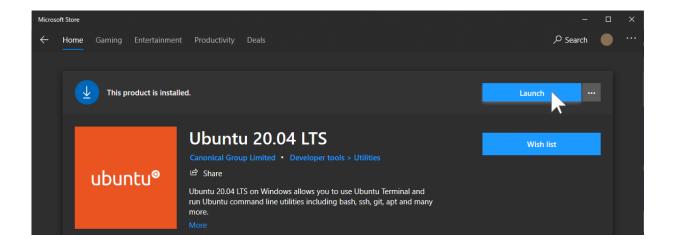
Return to Top

1.8.3. Install Linux Ubuntu OS

Open the Microsoft Store app, and search for Ubuntu and install the desired version (available versions are 16.04, 18.04, and 20.04)



When it has finished downloading, click Launch.



Return to Top

1.8.4. Set up Ubuntu in WSL

Wait for the Ubuntu OS to install.

Select a username and password. These do not need to be the same as your Windows or Microsoft Account login.

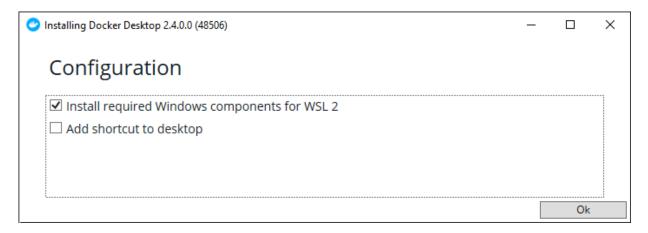
```
☑ Ubuntu 20.04 LTS
— □ X
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: demo_user
New password:
Retype new password:
_
```

Return to Top

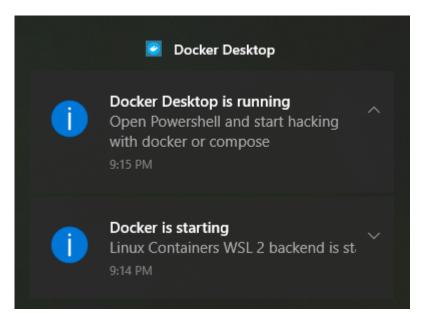
1.9. Install Docker for Windows

Download and run **Docker Desktop for Windows** from <u>Docker Hub</u> (https://desktop.docker.com/win/stable/Docker%20Desktop%20Installer.exe)

Be sure to select "Install required components for WSL2"



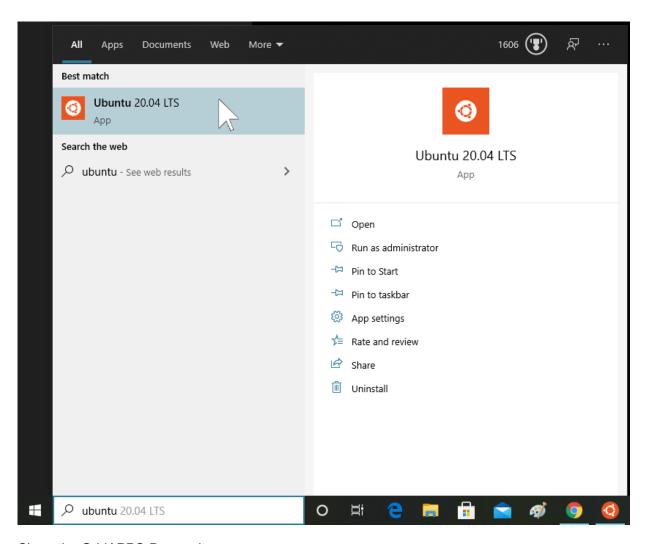
After restarting your machine, Docker should start automatically, and you will see a notification stating "Linux WSL2 containers are starting"



Return to Top

1.10. Install GridAPPS-D Platform

Open the Ubuntu terminal:



Clone the GridAPPS-D repository:

git clone https://github.com/GRIDAPPSD/gridappsd-docker

Open the **gridappsd-docker** folder and start the latest stable release of the GridAPPS-D platform.

cd gridappsd-docker

./run.sh -t releases_2020.09.0

Wait for the platform to download the required docker containers. This is a very large package and will take several minutes.

```
© demo_user@DESKTOP-06IFQ2M:~/gridappsd-docker

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo_root" for details.

demo_user@DESKTOP-06IFQ2M:~$ cd gridappsd-docker demo_user@DESKTOP-06IFQ2M:~/gridappsd-docker$ ./run.sh -t releases_2020.09.0

Create the docker env file with the tag variables

Downloading mysql data

Getting blazegraph status

Pulling updated containers
Pulling blazegraph ...
Pulling mysql ...
Pulling mysql ...
Pulling gridappsd ...
Pulling gridappsd ...
Pulling sample_app ...
Pulling sample_app ...
Pulling sample_app ...
Pulling influxdb ...
```

After the containers have finished downloading, they will automatically be created and then launched:

```
demo_user@DESKTOP-06IFQ2M:~\gridappsd-docker

demo_user@DESKTOP-06IFQ2M:~\S cd gridappsd-docker

demo_user@DESKTOP-06IFQ2M:~\gridappsd-docker\
demo_user@DESKTOP-06IFQ2M:~\gridappsd-docker\
./run.sh -t releases_2020.09.0

Create the docker env file with the tag variables

Downloading mysql data

Getting blazegraph status

Pulling updated containers
Pulling blazegraph ... done
Pulling mysql ... done
Pulling mysql ... done
Pulling mysql ... done
Pulling sample_app ... done
Pulling sample_app ... done
Pulling sample_app ... done
Pulling influxdb ... done

Starting the docker containers

Creating network "gridappsd-docker_default" with the default driver

Creating gridappsd-docker_sample_app_1 ... done
Creating gridappsd-docker_influxdb_1 ... done
Creating gridappsd-docker_viz_1 ... done
Getting blazegraph status_redis_1 ... done
Creating gridappsd-docker_blazegraph_1 ... done
Creating gridappsd-docker_blazegraph_1 ... done
Creating gridappsd-docker_blazegraph_1 ... done
Creating gridappsd-docker_blazegraph_1 ... done
```

When all the containers are running, the terminal will move inside the docker environment, which has its own internal directories and path.

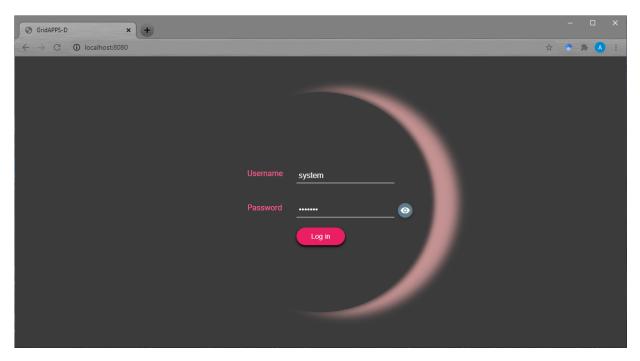
Start the GridAPPS-D platform inside the docker container by running

```
./run-gridappsd.sh
```



The GridAPPS-D platform is now installed and running.

To confirm, open <u>localhost:8080 (http://localhost:8080/)</u> to access the GridAPPS-D Visualization App:



Congratulations! You have successfully installed the GridAPPS-D Platform, and the GridAPPSD-Python development environment!

Return to Top

2. Ubuntu Linux

A typical Ubuntu installation already has most of the required packages already pre-installed, and only requires installation of the Python, Docker, and GridAPPS-D packages.

Note: The download size is quite large, so it is strongly recommended to use a fiber or ethernet interent connection, rathered than a metered hotspot to avoid excessive data usage charges.

Installation Steps

• 2.1. System Requirements

- 2.2. Install python-pip
- 2.3. Install Anaconda or Miniconda
- 2.4. Install Jupyter Lab
- 2.5. Install GridAPPSD-Python
- 2.6. Download Python Notebooks
- 2.7. Install GridAPPS-D Platform

2.1. System Requirements

- OS:
 - Ubuntu Linux, 16.04, 18.04, or 20.04
- RAM:
 - 8GB (absolute minimum for 13 and 123 node models, may encounter memory overload during installation);
 - 16GB (preferred for small models, minimum for 9500 node model);
 - 32GB (recommended)
- · Disk Space:
 - 15GB required for installation

Return to Top

2.2. Install python-pip

If not **pip** is not installed, use apt-get to install it.

• sudo apt-get install pip

Return to Top

2.3. Install Anaconda or Miniconda

Download the latest version of Anaconda or Miniconda and save it in the /Downloads folder:

- Use Python 3.8 install for 64-bit systems from the <u>Conda.io website</u> (https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86 64.sh).
- Use Python 3.7 install for 32-bit systems from the <u>Conda.io website</u> (https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86.sh)

Install Miniconda using bash

- cd /Downloads
- bash Miniconda3-latest-Linux-x86 64.sh

Follow the prompts on the installer screens. If you are unsure about any setting, accept the defaults. You can change them later. To make the changes take effect, close and then re-open your terminal window. Test your installation. In your terminal window or Anaconda Prompt, run the command conda list. A list of installed packages appears if it has been installed correctly.

Return to Top

2.4. Install Jupyter Lab

Use **pip** to install Jupyter Lab, which is need to open and execute the Python Training Notebooks

• pip install jupyterlab

Return to Top

2.5. Install GridAPPSD-Python

Use **pip** to install GridAPPSD-Python from GitHub:

pip install git+https://github.com/GRIDAPPSD/gridappsdpython.git@develop#egg=gridappsd

Return to Top

2.6. Download Python Notebooks

clone the python notebooks by running

git clone https://github.com/GRIDAPPSD/gridappsd-hackathon

to download the python training notebooks.

```
⊗ ● ① demo_user@demo-session:~

(base) demo_user@demo-session:~$ git clone https://github.com/GRIDAPPSD/gridappsd-hackathon
```

By default, the notebooks will be saved in the directory C:\Users\username\gridappsd-hackathon

Close the miniconda terminal

Return to Top

2.7. Install GridAPPS-D Platform

Clone the GridAPPS-D GitHub repository

git clone https://github.com/GRIDAPPSD/gridappsd-docker

Change directories into gridapps-docker and run the Docker installation script

- cd gridappsd-docker
- ./docker_install_ubuntu.sh

After Docker finishes installing, log out or restart the Ubuntu session.

After logging back in, change directories into gridappsd-docker and start the latest stable version of the GridAPPS-D platform, which will automatically download the required docker containers.

- cd gridappsd-docker
- ./run.sh -t releases_2020.09.0

Wait for the docker containers to finish downloading. This will take a while due to the package size.

When the containers have finished downloading and installing, start the GridAPPS-D Platform

• ./run-gridappsd.sh

```
Getting blazegraph status

Checking blazegraph data

Blazegrpah data available (1958486)

Getting viz status

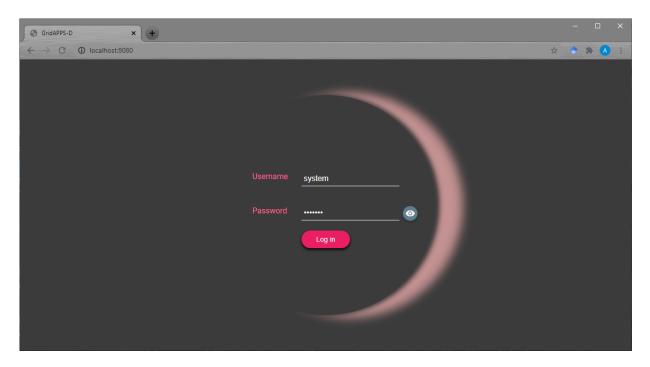
Containers are running

Connecting to the gridappsd container docker exec -it gridappsddocker_gridappsd_1 /bin/bash

gridappsd@d3bd83459470:/gridappsd$ ./run-gridappsd.sh
```

The GridAPPS-D platform is now installed and running.

To confirm, open <u>localhost:8080 (http://localhost:8080/)</u> to access the GridAPPS-D Visualization App:



Congratulations! You have successfully installed the GridAPPS-D Platform, and the GridAPPSD-Python development environment!

Return to Top

3. Mac OS

To be completed in a future release -- Install same packages for Linux using Mac releases

For GridAPPS-D install intructions on Mac, see

https://gridappsd.readthedocs.io/en/latest/installing_gridappsd/index.html (https://gridappsd.readthedocs.io/en/latest/installing_gridappsd/index.html)