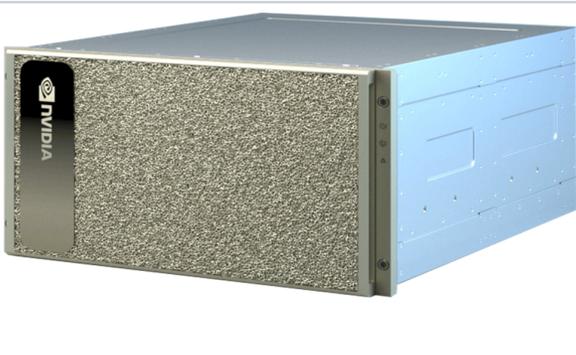


# NVIDIA DGX A100

*In publications and presentations that use results obtained on this system, please include the following acknowledgement: "This work utilizes resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign".*

[Main -> Systems](#)  
-> DGX A100

## System Description

<p>Host name: <b>hal-dgx.ncsa.illinois.edu</b></p> <p>Hardware</p> <ul style="list-style-type: none"><li><a href="#">nvidia-dgx-a100-datasheet.pdf</a></li><li><a href="#">dgxa100-user-guide.pdf</a></li></ul> <p>Software</p> <ul style="list-style-type: none"><li>RHEL 8</li><li>CUDA 12</li></ul>	
--	--

## Contact us

Request access to ISL resources:  
Application

Contact ISL staff: [Email Address](#)

Visit: [NCSA](#), room 3050E

## User Guide

To request an account: email [kindrtnk@illinois.edu](mailto:kindrtnk@illinois.edu)



Use `/home/<username>` for basic stuff only, do not put any code/data here as the `/home` partition is very small. **Quota: 50GB per User**

Use `/projects` file system for all your data/code. Create a subfolder in this partition for your username and keep your stuff there. **Quota: 2TB/10 million inodes per User**

Use `/scratch` file system for ephemeral/transient data that is created during job runs that doesn't need to be kept long term. (And do not forget to remove it after the job is done and you no longer need it!)

To access the system, submit jobs, etc.: [Access HAL-DGX and OVERDRIVE servers through HAL-LOGIN4 node](#)

CUDA tools are already in the path, just use `nvcc` to compile.

To work with Python, there are several versions installed, from 2.7 to 3.9. Create and enable virtual environment first, e.g., to use 3.8:

```
python3.8 -m venv python38-virtualenv
source python38-virtualenv/bin/activate
```

You can use `pip` to install python packages within this environment.

To start a Jupyter notebook on `hal-dgx`

- on `hal-dgx`:

```
# source your python environment, e.g., 3.8
source python38-virtualenv/bin/activate
pip install jupyter # this needs to be done only once
jupyter notebook --port=9999 # this will start the jupyter server on port
9999. Pick a different port if it fails to start
```

- on your own computer:

```
# This opens a connection to the hal-dgx.ncsa.illinois.edu Jupyter server,
and
# forwards any connection to port 8888 on the local machine to port 9999
on hal-dgx.ncsa.illinois.edu.
ssh -L 8888:localhost:9999 <userid>@hal-dgx.ncsa.illinois.edu
```

- Finally, on your own computer, open web browser and point it to the address you see after running 'jupyter notebook' on hal-dgx, something like <http://localhost:8888/?token=...>
  - Make sure to replace the port in the URL with the port on your localhost