NREL HPC Workshops: Software Environments on Eagle

Kevin Sayers

HPC User and Applications

- How can you get the software you need on Eagle?
- Three common solutions
  - Environment modules
  - Conda
  - Containers
- Each of these have associated pros/cons

What is PATH

- Linux environment variable
- Colon separated list of directories
    - /nopt/xalt/xalt/bin:/nopt/nrel/utils/bin:/nopt/slurm/current/bin:/usr/local/bin:/usr/bin
- Linux searches these directories for executable files
- Adding a directory will add it to this search
- A key component to both Modules and Conda is modifying this variable

Environment Modules

- Dynamically change environment through modulefiles
- Modulefiles specify info such as paths, environment variables, etc.
- Modules on Eagle are managed by NREL users and groups
- They may be more optimized for Eagle

```
help([[
  The GNU Compiler Collection includes front ends for C, C++,
Objective-C,
  Fortran, Ada, and Go, as well as libraries for these
languages.
]])

whatis("The GNU Compiler Collection includes front ends for C,
C++, Objective-C,
 Fortran, Ada, and Go, as well as libraries for these languages.")

local base = "/nopt/nrel/apps/base/2020-05-
12/spack/opt/spack/linux-centos7-x86_
64/gcc-4.8.5/gcc-10.1.0-iw6p5hcjkqdddphuodu6abtqifbaqzu2"


setenv("CC", pathJoin(base, "bin/gcc"))


prepend_path("PATH", pathJoin(base, "bin"))
```

Modules pros/cons

- Pros
  - Installed and ready to use for all Eagle users
  - Optimized for Eagle (GPU and CPU architectures)
  - Mix and match with Conda

- Cons
  - Not as easily portable
  - Limited number of packages
    - Ecom modules are community contributed

Conda

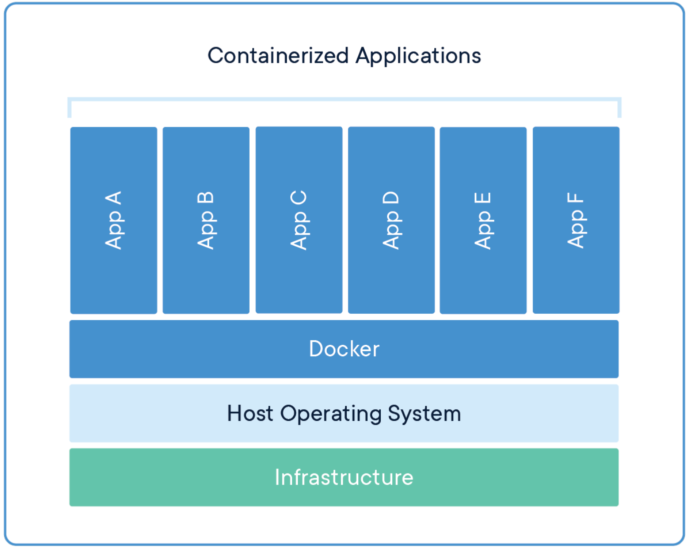
- Package manager combined with environment manager
- Available as a module on Eagle
- It is best to use the provided module and not install your own version of Conda
- Easily create and switch between environments
- Mamba: a faster alternative to Conda

Conda pros/cons

- Pros
  - Large library of packages
  - Easy to manage multiple environments
  - Environments can be setup wherever you have Conda
    - Local, HPC, Cloud
- Cons
  - May not be optimized for Eagle
  - Can easily take up space in your home directory if not careful

Containers

- Containers are an isolated environment
- Docker and Singularity
- Singularity available as a module
- Containers are built from a recipe file.
- Run anywhere a container engine is present
    - Managed cloud resources



Containers pros/cons

- Pros
  - Containers are very portable
    - Local, HPC, and Cloud
  - If container engine is present it is likely container will run
  - Isolated environment may enable installing more challenging packages
- Cons
  - Build process is involved
  - Harder to mix and match packages
  - Security concerns
  - Managing data is less straighforward

Container security

- Containers can be insecure
- Dockerhub is a mix of container images from official sources and user contributed with minimal oversight
- Docker configuration may allow privilege escalation
- It is best to pull containers from official sources or build your own

Demos

Pros/cons summary

Feature	Modules	Conda	Containers
Search	conda search gcc	ml spider gcc	Docker search gcc
NREL Managed	Yes	No	No
Portability	Not easily	Yes	Very
Mix and match	Yes	Yes	No

How I personally use these

Modules: MPI, compilers, and GPU

Conda: Python packages, analysis software

Containers: Difficult to install packages or those that I need to be portable and contained