Applied HPC Seminar Series: Introduction to Containerization for HPC

What is Containerization?

- Think of it as a “portable research environment”
- Contains everything your application needs to run:
  - Operating system files
  - Software dependencies
  - Libraries
  - Configuration files
- Like a lightweight, portable computer within your computer

Why Use Containers in Medical Imaging Research?

- **Reproducibility**: Same environment = same results
- **Portability**: Run anywhere (local machine, HPC cluster, cloud)
- **Version Control**: Track changes in your research environment
- **Collaboration**: Share exact analysis environment with colleagues
- **Dependency Management**: No more “works on my machine” problems

Real-World Example: Medical Image Processing

Before Containers:

```
❌ Install specific CUDA version
❌ Install specific Python version
❌ Install TensorFlow with GPU support
❌ Install specific NVIDIA drivers
❌ Install medical imaging libraries
❌ Hope everything works together
```

With Containers:

```
✅ Pull container image
✅ Run your analysis
```

Popular Container Technologies in HPC

Singularity/Apptainer

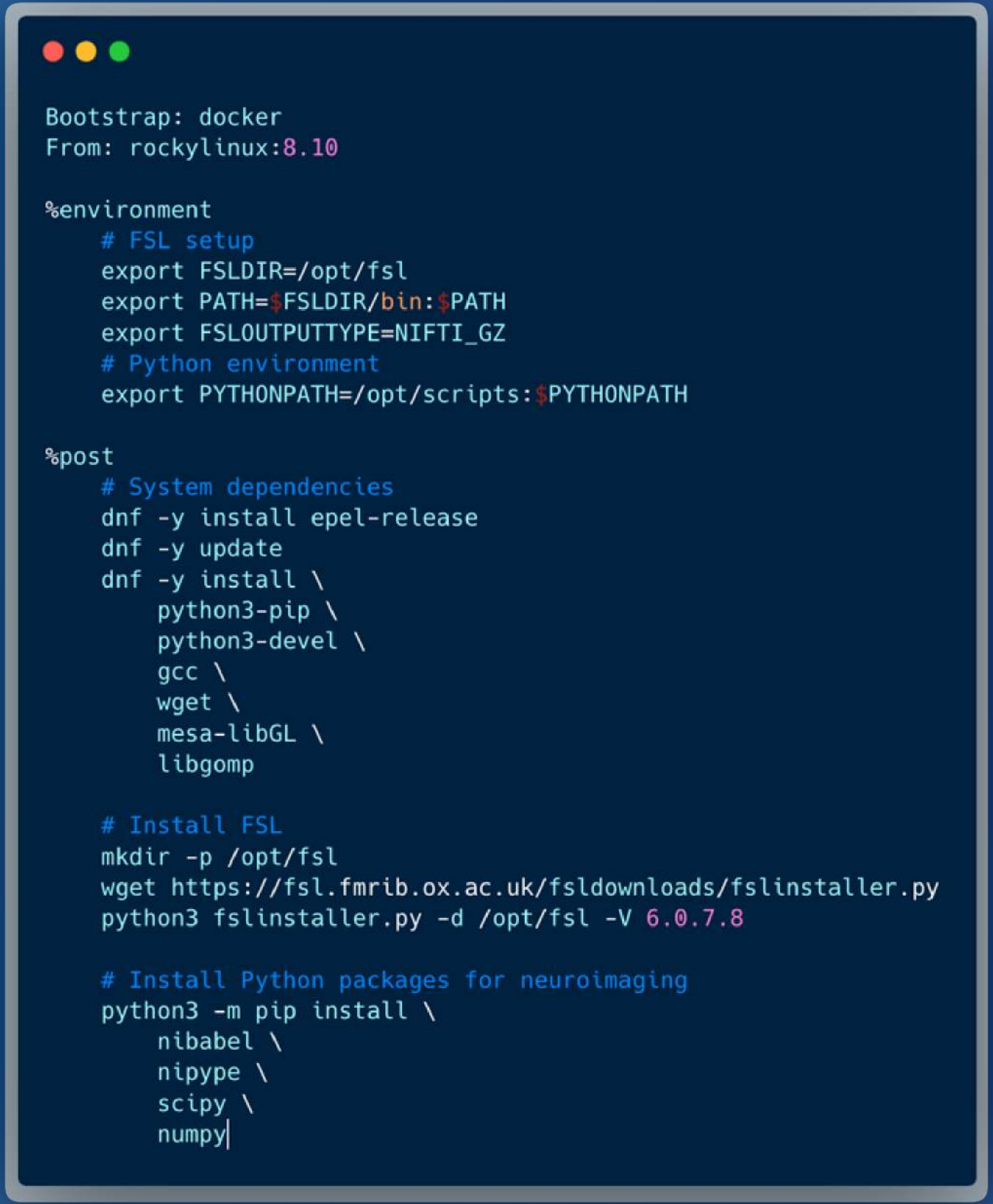
- Designed specifically for HPC
- Secure by design
- Native support for GPU acceleration
- **Perfect for medical imaging workloads**
- Supported on our HPC systems!

Basic Container Workflow

1. **Define** your environment
   1. Software versions
   2. Dependencies
   3. Configuration
2. **Build** your container
   1. Create image from definition
   2. Test locally
3. **Run** your analysis
   1. Deploy on HPC
   2. Process your data

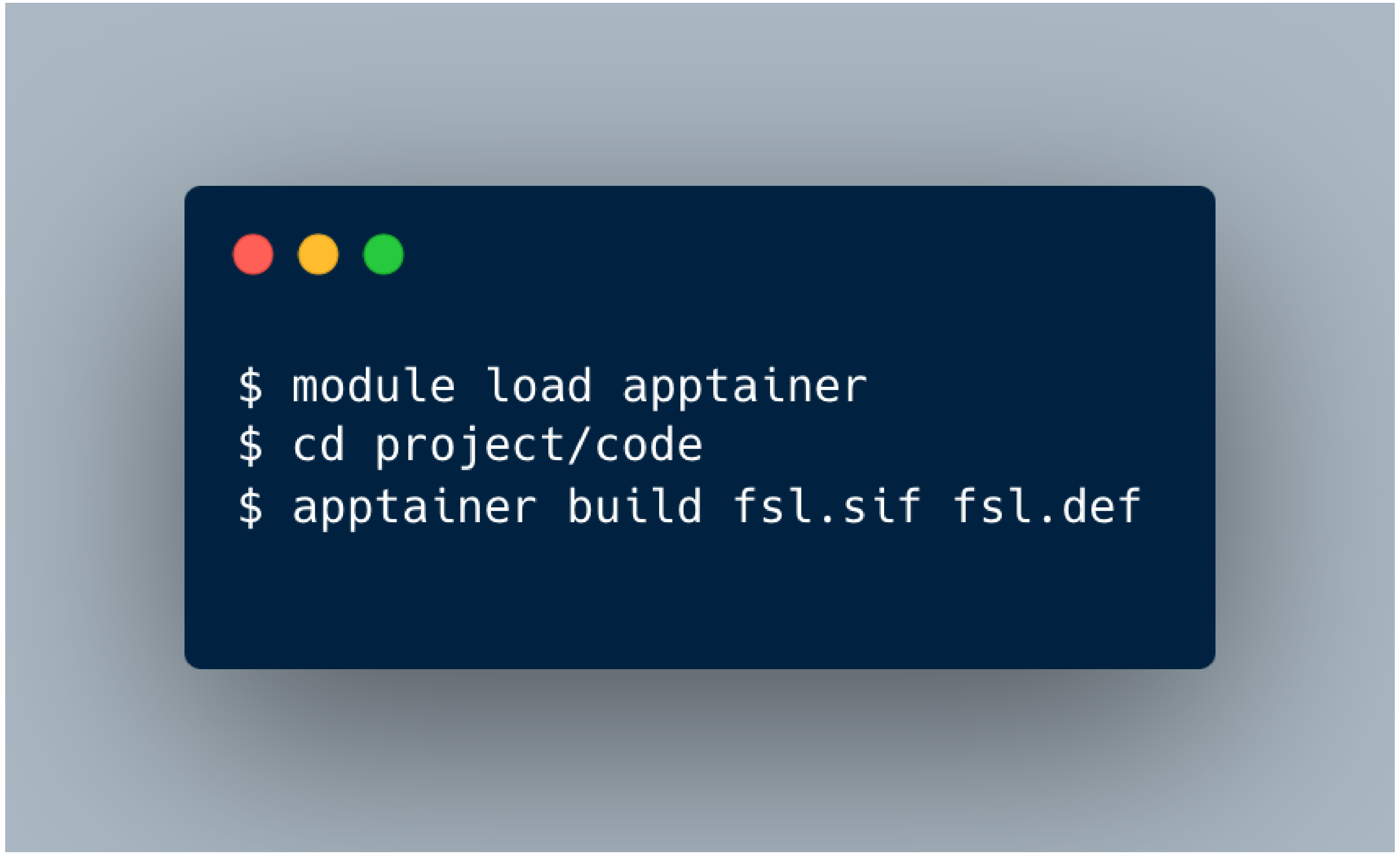
Let's think inside the Box with FSL...

Example: Neuro Image Processing with FSL



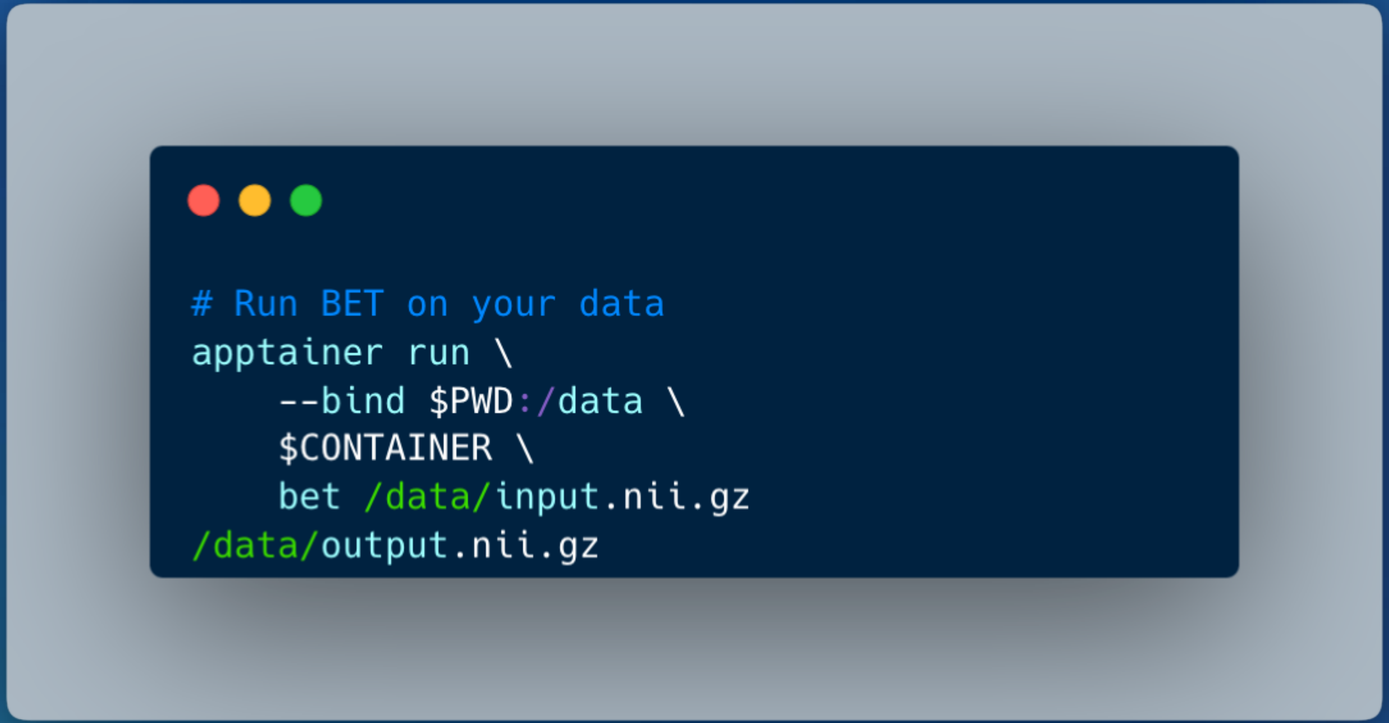
```
Bootstrap: docker
From: rockylinux:8.10

%environment
    # FSL setup
    export FSLDIR=/opt/fsl
    export PATH=$FSLDIR/bin:$PATH
    export FSLOUTPUTTYPE=NIFTI_GZ
    # Python environment
    export PYTHONPATH=/opt/scripts:$PYTHONPATH

%post
    # System dependencies
    dnf -y install epel-release
    dnf -y update
    dnf -y install \
        python3-pip \
        python3-devel \
        gcc \
        wget \
        mesa-libGL \
        libgomp

    # Install FSL
    mkdir -p /opt/fsl
    wget https://fsl.fmrib.ox.ac.uk/fsldownloads/fslinstaller.py
    python3 fslinstaller.py -d /opt/fsl -V 6.0.7.8

    # Install Python packages for neuroimaging
    python3 -m pip install \
        nibabel \
        nipype \
        scipy \
        numpy
```

Running Containers on HPC



```
$ module load apptainer
$ cd project/code
$ apptainer build fsl.sif fsl.def
```

Basic FSL Commands



```
# Run BET on your data
apptainer run \
    --bind $PWD:/data \
    $CONTAINER \
    bet /data/input.nii.gz
/data/output.nii.gz
```

Key Points to Remember

- **Important Files:**
  - fsl.def: Container definition
  - fsl.sif: Built container
- **Basic Usage**

  -
    ```
    apptainer run \
    --bind /your/data:/data \
    /path/to/fsl.sif \
    fsl-command
    ```
- **Data Access**
  - Always bind your data directory
  - Use absolute paths
  - Keep data organized

Best Practices for Medical Imaging

1. **Include GPU Support**
    1. Essential for deep learning
    2. Speeds up image processing
2. **Data Management**
    1. Mount data directories
    2. Use efficient I/O patterns
3. **Resource Optimization**
    1. Request appropriate resources
    2. Use parallel processing when possible

Common Use Cases in Medical Imaging

- MRI Analysis Pipelines
- CT Image Processing
- Deep Learning Models
- Image Registration
- Segmentation Workflows

Getting Started

1. **Start Small**
    1. Begin with a simple container
    2. Add complexity gradually
2. **Use Available Resources**
    1. Pre-built containers
    2. HPC documentation
    3. Community support
3. **Document Everything**
    1. Container definitions
    2. Run commands
    3. Workflow steps

Support and Resources

- HPC Help Desk
- Medical Imaging Communities
    - Neuroimaging Tools and Resources (NITRC)
    - BioConda

Thank You!

Remember:

Containers make your research reproducible

Start small and build up

Help is always available

Questions?

Contact: dennist@wustl.edu