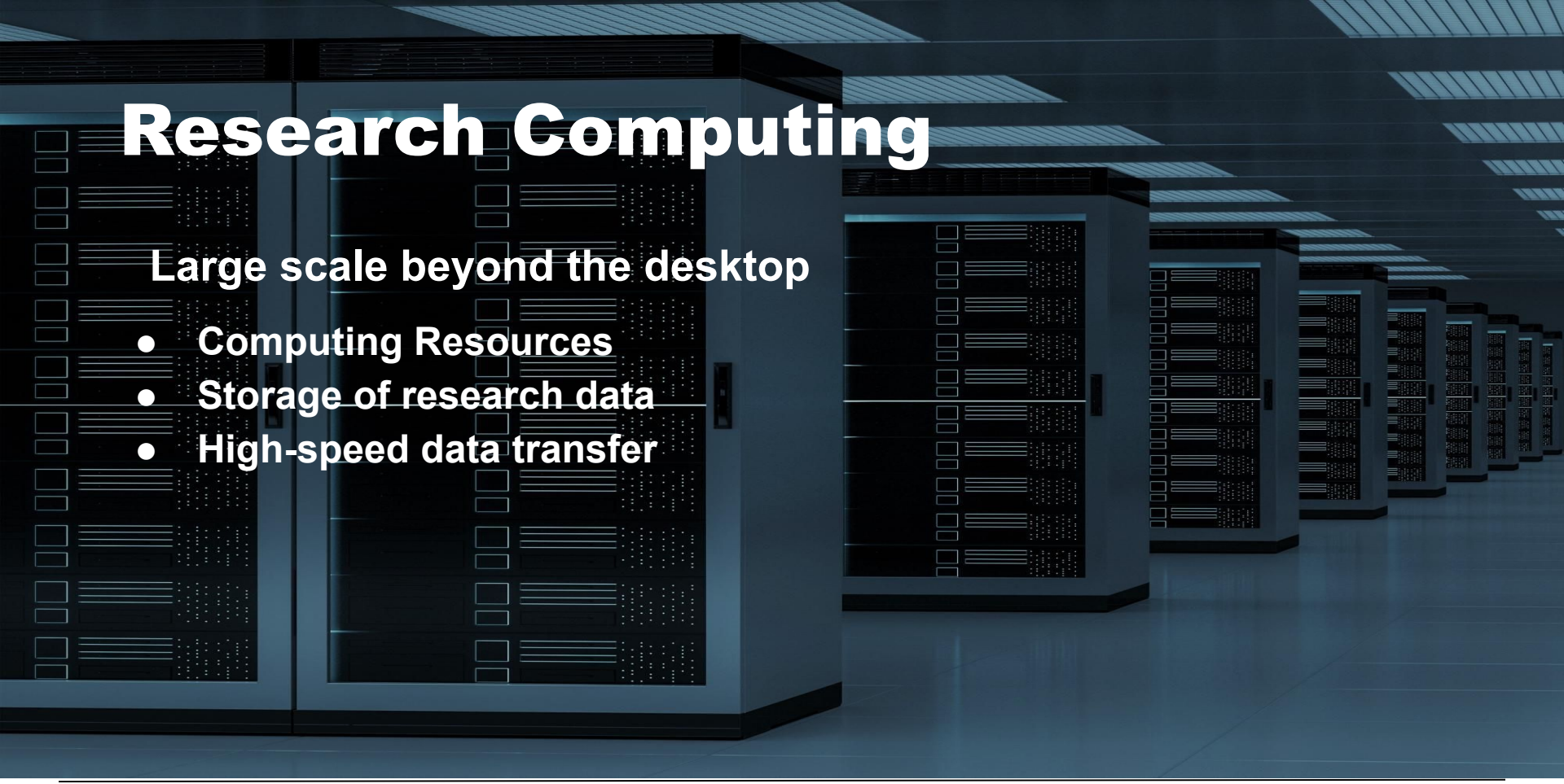

Research Computing
New User Seminar
Be Boulder.
University of Colorado Boulder

Research Computing New User Seminar

- Gerardo Hidalgo-Cuellar: gehi0941@Colorado.Edu
- Website: www.rc.colorado.edu
- Help: rc-help@colorado.edu
- Slides: https://github.com/ResearchComputing/New_User_Seminar



Research Computing

Large scale beyond the desktop

- Computing Resources
- Storage of research data
- High-speed data transfer

Goals

1. Why High Performance Computing (HPC)?
2. Understand Basic Resources
3. Get an account & log in
4. Navigate the RC system
5. Run a job
6. Help!

Things to Take Note Of

- Confusing, ambiguous, highly nuanced concepts
- Common mistakes or frustrations

Ask Questions!

Jargon

- **RC** = Research Computing
- **HPC** = High Performance Computing
- **<input>** = your input (username, password, etc.), do not include <> when you write it

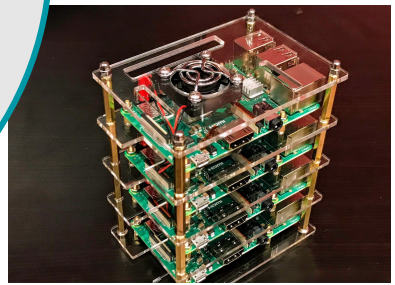
Why High Performance Computing (HPC)?

Personal Computers

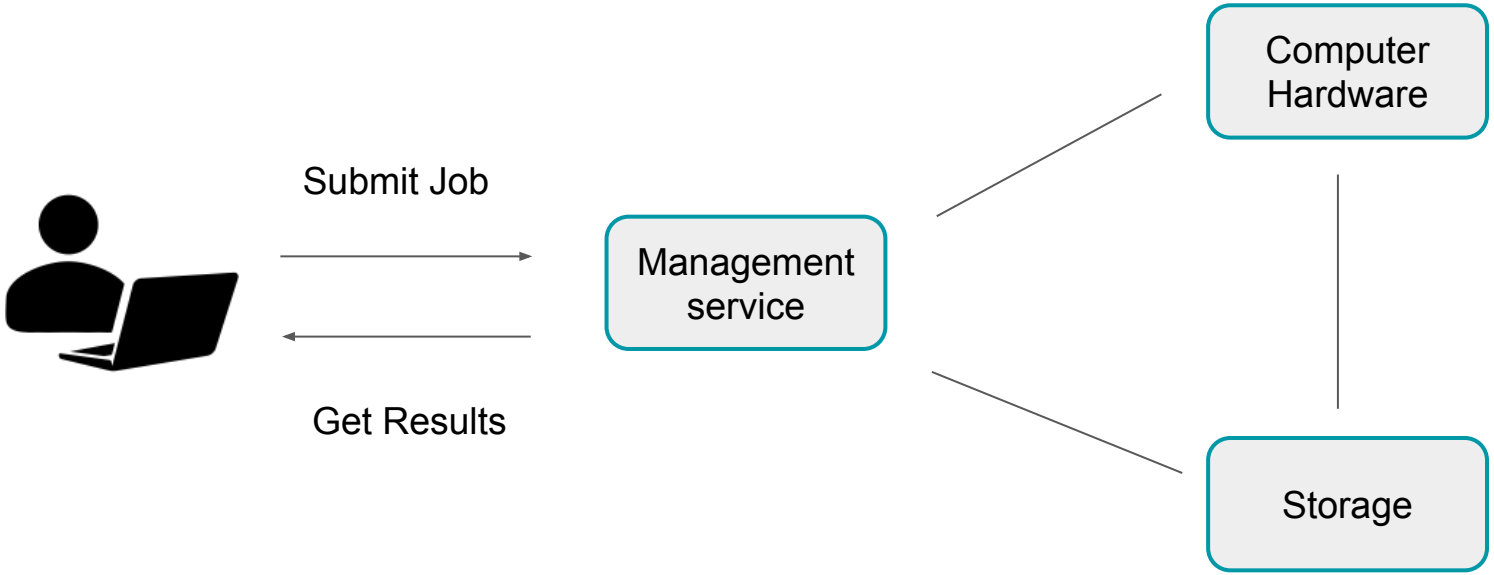
- Personal Data
- Photos
- Games
- Applications...

HPC

- Behavior analysis
- Climate modeling
- Molecular sciences
- Early universe...



High Performance Computing



Research Computing Resources

What is Research Computing?

- Provide services for researchers that include:
  - Large scale computing
  - Data storage
  - High speed data transfer
  - Data management support
  - Consulting & Training

- We are likely best known for:
  - Summit Supercomputer
  - PetaLibrary Storage

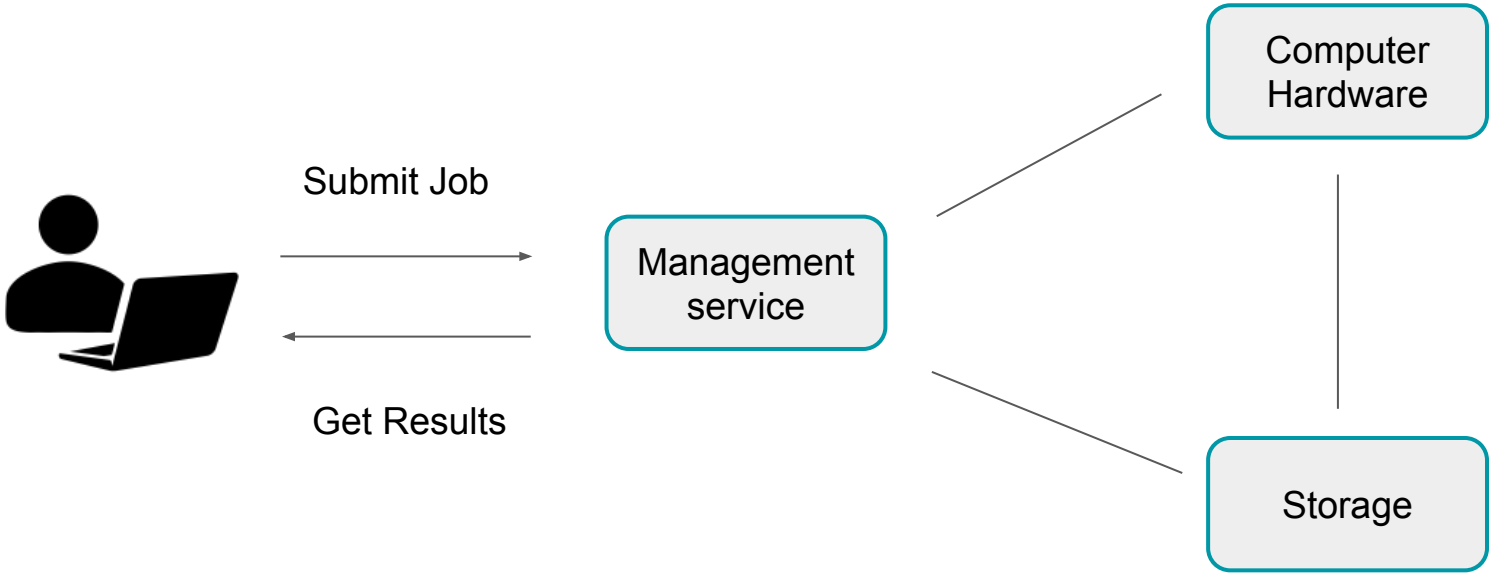
Research Computing Services

- Supercomputing Clusters
- Storage
- Gateways

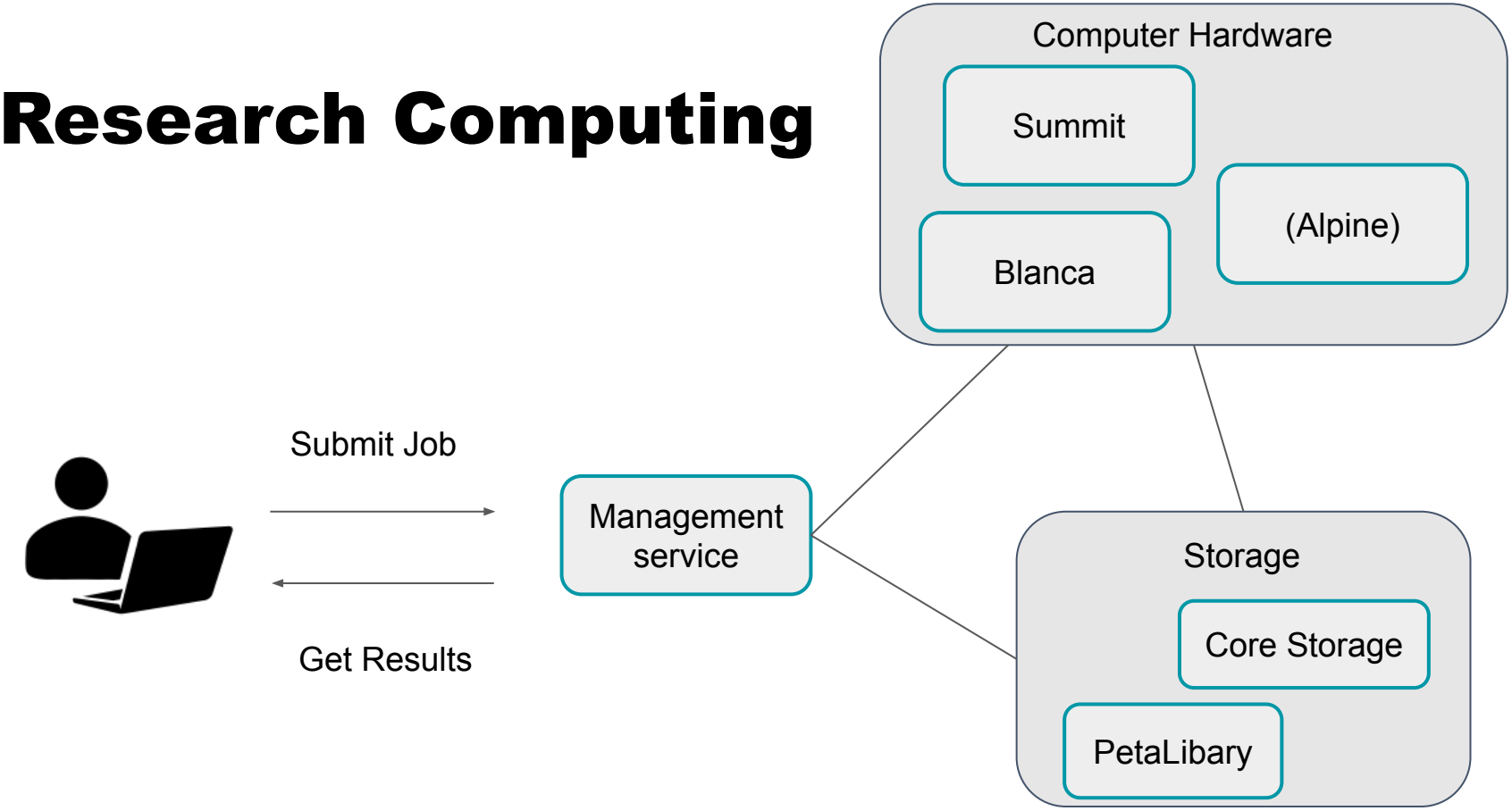
Research Computing Services

- Supercomputing Clusters
  - Summit, Blanca, (Alpine)
- Storage
  - Core, PetaLibrary
- Gateways
  - JupyterHub, EnginFrame, (CUmulus)

High Performance Computing



CU Research Computing



Summit Cluster

- Free-access, NSF-funded supercomputer

- 450+ (mostly) compute nodes
  - Also have memory, and graphics nodes

- Shared between CU, CSU, and Rocky Mountain Advanced Computing Consortium (RMACC)

Blanca Cluster

- Condo computing service
  - Buy-in Cluster

- Compute and Graphics nodes available.

PetaLibarary

- Service for:
  - Storage
  - Archive
  - Sharing of research data.

- Available at a subsidized cost to any researcher affiliated with the University of Colorado Boulder.

Accessing Research Computing

How to Access RC Resources?

1. Get an account
2. Set up two factor authentication
3. (Inform us of any specific needs)
4. Log in
5. Create greatness! (responsibly)

Getting an account

- CU Boulder users and affiliates:
  - Request an account through the RC Account request portal
  - https://rcamp.rc.colorado.edu/accounts/account-request/create/organization

- CSU Users:
  - Request an CSU eID if you don't have one
  - Fill out account application form
  - Duo authentication
  - Then get an RC user account
  - https://www.acns.colostate.edu/hpc/summit-get-started/

- RMACC Users:
  - Contact your local representative, if known. Email rc-help@colorado.edu
  - We'll guide you through the process

Demo: Getting an Account

- CU Boulder users and affiliates:
  - Request an account through the RC Account request portal
  - https://rcamp.rc.colorado.edu/accounts/account-request/create/organization

Your RC Account

Access to:

1. Summit Cluster
2. “Core” storage
3. Gateways (CU affiliates)

Two Factor Authentication

- Provides an extra level of authentication
  - We are outside the firewall!
  - Valuable resources
  - Inviting, high-profile target
  - Lost time investigating/fixing

- Duo
  - You will receive a Duo invitation when your RC account is created.

Duo Authentication

1. Most users use the Duo smartphone app
2. “Phone Call” is an alternatives
3. Physical code generator “token” available for $20

Terminal

- Mac or Linux
  - Terminal application
- Windows
  - PuTTY

Demo: Logging in

- To login to an RC login node:

```
ssh <username>@login.rc.colorado.edu
```

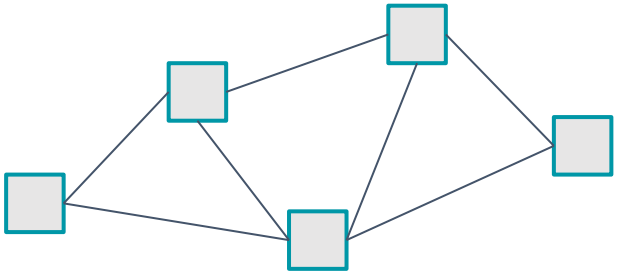
Supply your IdentiKey password and your Duo app will alert you to confirm the login

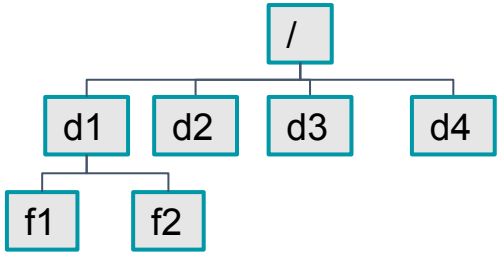
Logging In

- It’s important to note that you are NOT logging into any specific resource, Summit, etc.
- When you log in, you land on our login nodes
- From there, you can access our other resources

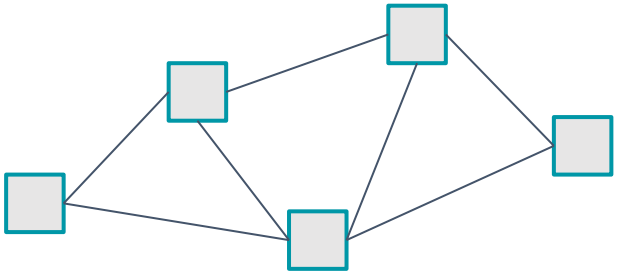
Navigating Research Computing

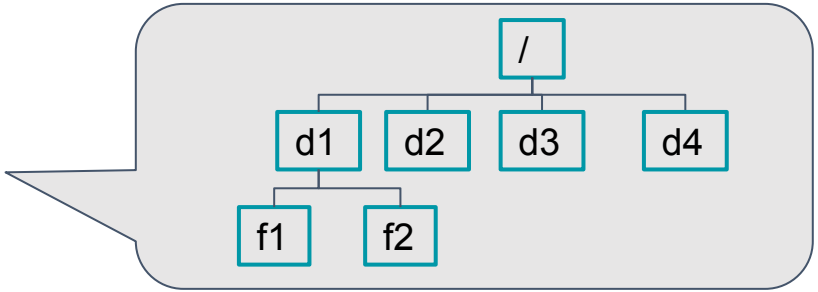
Node	File System
• One computing server • Physical hardware • Work together in parallel	• The basic tree-like layout • From any node* you have access to the entire file system



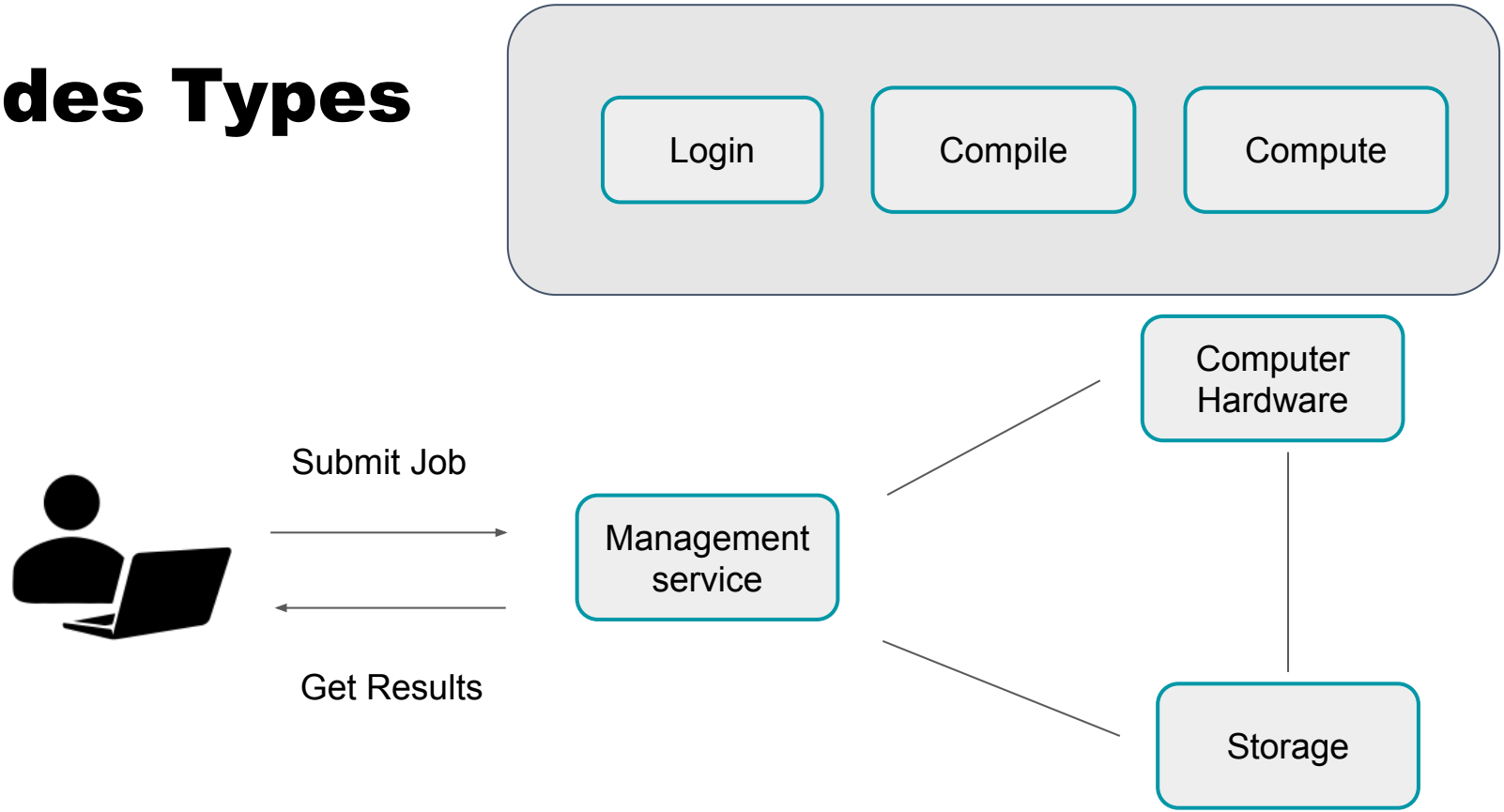


Node	File System
• One computing server • Physical hardware • Work together in parallel	• The basic tree-like layout • From any node* you have access to the entire file system





Nodes Types


Login
Compile
Compute
Computer Hardware
Submit Job
Management service
Get Results
Storage

Nodes

Login	Compile	Compute
• Where you start • For editing code, job submission • No heavy computation	• Where you compile code, install packages • Explore the Summit software environment • Edit code, submit jobs • No heavy computation	• Where scheduled jobs run • Intended for heavy computation
Ex. edit job script	Ex. Install python libs	Ex. Running Matlab

Demo: Exploring nodes

- Once logged in:

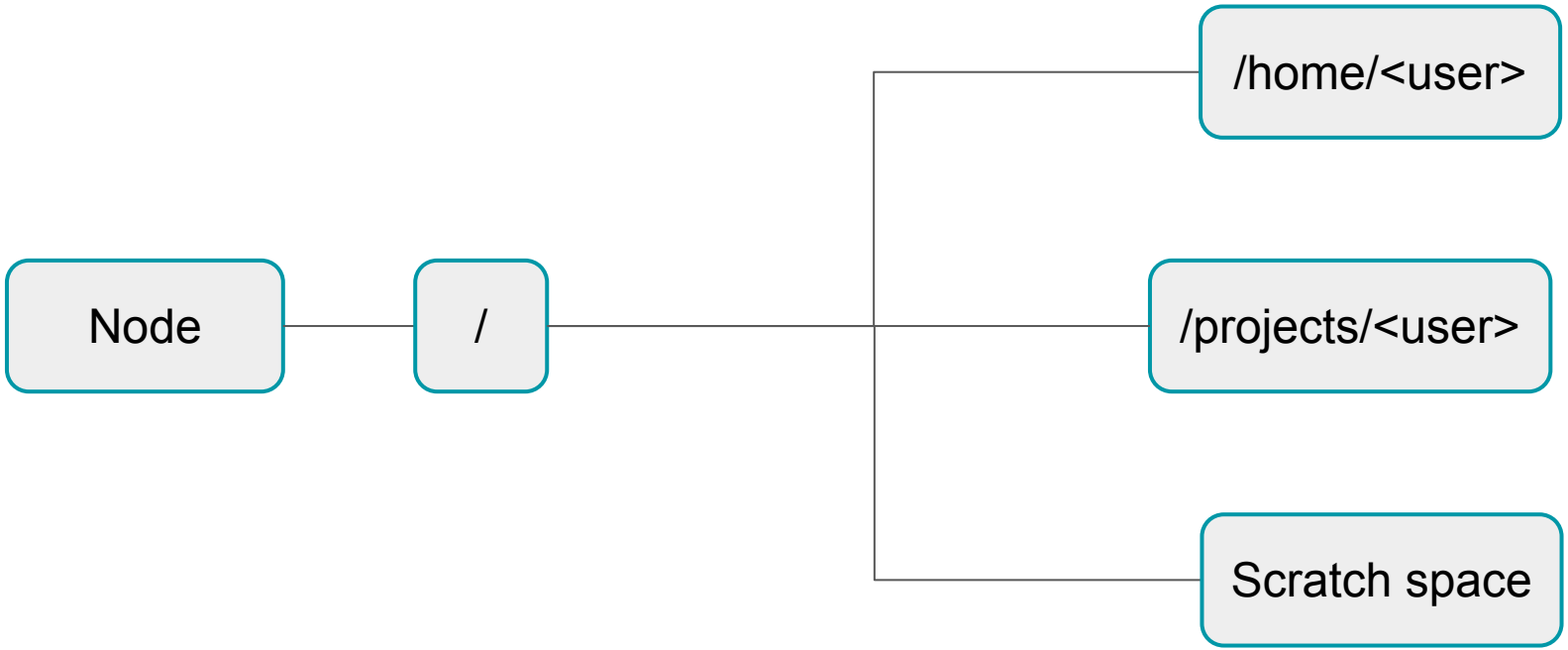
```
ssh scompile
```

To log in to a compile (or head) node.

```
module avail
```

To check currently available software

File system map



File Structures

/home (2GB)	/projects (250GB)	Scratch (10TB)
• Scripts, Code, Small, important files/directories • Not for sharing files or job output	• Code/files/libraries • Software you are installing • Sharing files • Not for job output	• Output from running jobs • Large files/datasets • Sharing files • Not for long term storage
Ex .bashrc	Ex. Shared job scripts	Ex. Data

Demo: Exploring the Filesystem

- Once logged in:

```
cd  /home/<user>
cd  /projects/<user>
cd  /scratch/summit/<user>
```

To navigate to your different directories

Using Research Computing

- We have logged on
- We have explored nodes and filesystem
- But how do we actually *use* the computing resources?

Running a Job

The fundamental “job”

What is a “job”?

- Jobs are scripted packages of work for the cluster to perform on

1. **Batch jobs**
   - Submit job script which will be executed when resources are available
     - Create or modify a script containing information about the job
     - Submit the job file to a queue
2. **Interactive jobs**
   - Work interactively at the command line of a compute node

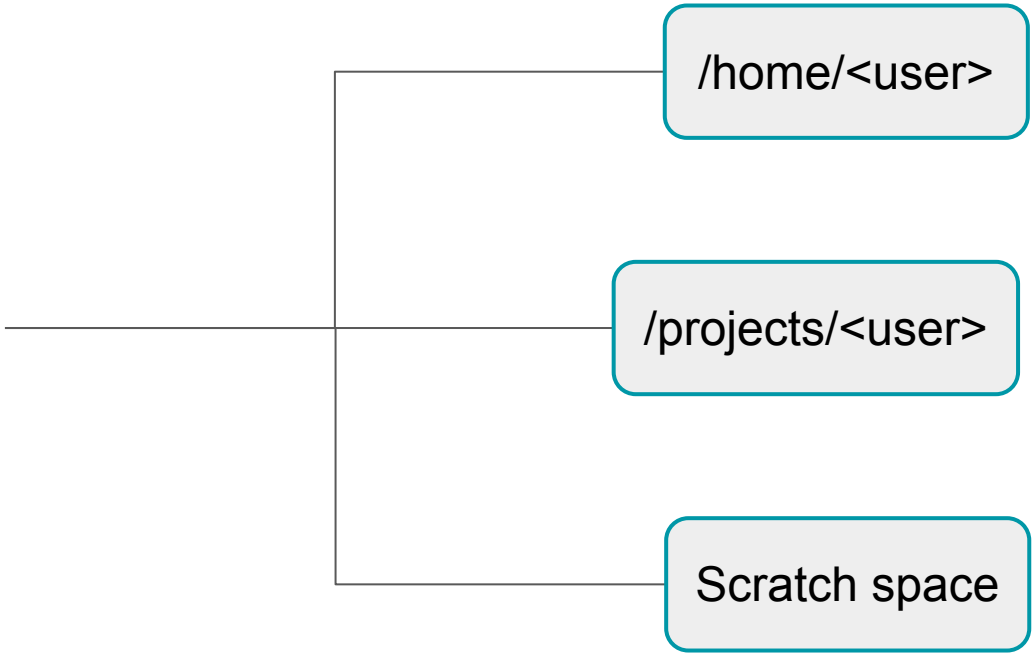
Job Scheduling

- On a supercomputer, jobs are scheduled rather than just run instantly
  - Shared system
  - Jobs are put in a queue until resources are available

- **Simple Linux Utility for Resource Management (SLURM)**
  - Keeps track of what nodes are busy/available, and what jobs are queued or running
  - Tells the resource manager when to run which job on the available resources

Your first job

- Where to put it?
- How to write it?
- How to run it?



Job Script: 3 main parts

1. Directives
   - Specify resource requirements

2. Software
   - jobs run on a different node than from where you ran
   - software that is needed must be loaded via the job script

3. User scripting
   - the actual user scripting that will execute when the job runs

Directive Options http://slurm.schedmd.com/sbatch.html

`#SBATCH <options>` `sbatch <options>`

- Allocation: `--account=<account_no>`
- Partition: `--partition=<partition_name>`
- Sending emails: `--mail-type=<type>`
- Email address: `--mail-user=<user>`
- Number of nodes: `--nodes=<nodes>`
- Number of tasks: `--ntasks=<processes>`
- Quality of service: `--qos=<qos>`
- Reservation: `--reservation=<name>`
- Wall time: `--time=<wall time>`
- Job Name: `--job-name=<jobname>`

- FYI: You do NOT actually type <> above – this designates something specific you as a user must enter about your job

Writing a Job: Hostname

- Submit a slurm job with the following instructions:

1. The job will be submitted from a bash script named hostname_summit.sh
2. The job should run the Unix “hostname” command
3. The job will run on 1 node
4. We will request 1 minute wall time
5. Run using “testing” QOS
6. Run on the shas-testing partition
7. Output should contain job #

Demo: Writing Hostname

Demo: Writing Hostname

```
#!/bin/bash
#SBATCH --nodes=1                          # Number of requested nodes
#SBATCH --time=0:01:00                     # Max wall time
#SBATCH --qos=testing                      # Specify QOS
#SBATCH --partition=shas-testing           # Specify Summit haswell
nodes
#SBATCH --output=hostname_%j.out           # Rename standard output file

# purge all existing modules
module purge

hostname
```

Submitting a Job

1. Load up the slurm module (probably not needed)
   - module load slurm/summit

2. Submit the job:
   - sbatch hostname_summit.sh

3. Check output

Help! I'm stuck, where do I go next?

We've got your back:

- Documentation: https://curc.readthedocs.io/en/latest/
- Trainings:

  https://www.colorado.edu/crdds/what-we-do/education-training
- Email: rc-help@colorado.edu

Review: Goals

1. Why High Performance Computing (HPC)?
2. Understand Basic Resources
3. Get an account & log in
4. Navigate the RC system
5. Run a job
6. Help!

Questions

Survey and feedback

http://tinyurl.com/curc-survey18