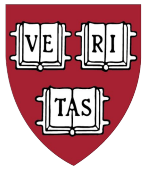
Kempner INSTITUTE



HARVARD UNIVERSITY

Introduction to Kempner AI Cluster

September 24, 2024

Objectives

By the end of this workshop, you will be able to:

- Describe cluster infrastructure, including storage locations
- Develop code on the cluster using VSCode for remote development
- Transfer data to/from the cluster using different methods based on use case
- Submit interactive and basic jobs using slurm

Agenda

1 **Orientation to Infrastructure**

Cluster access, architecture, storage tiers

2 **Development**

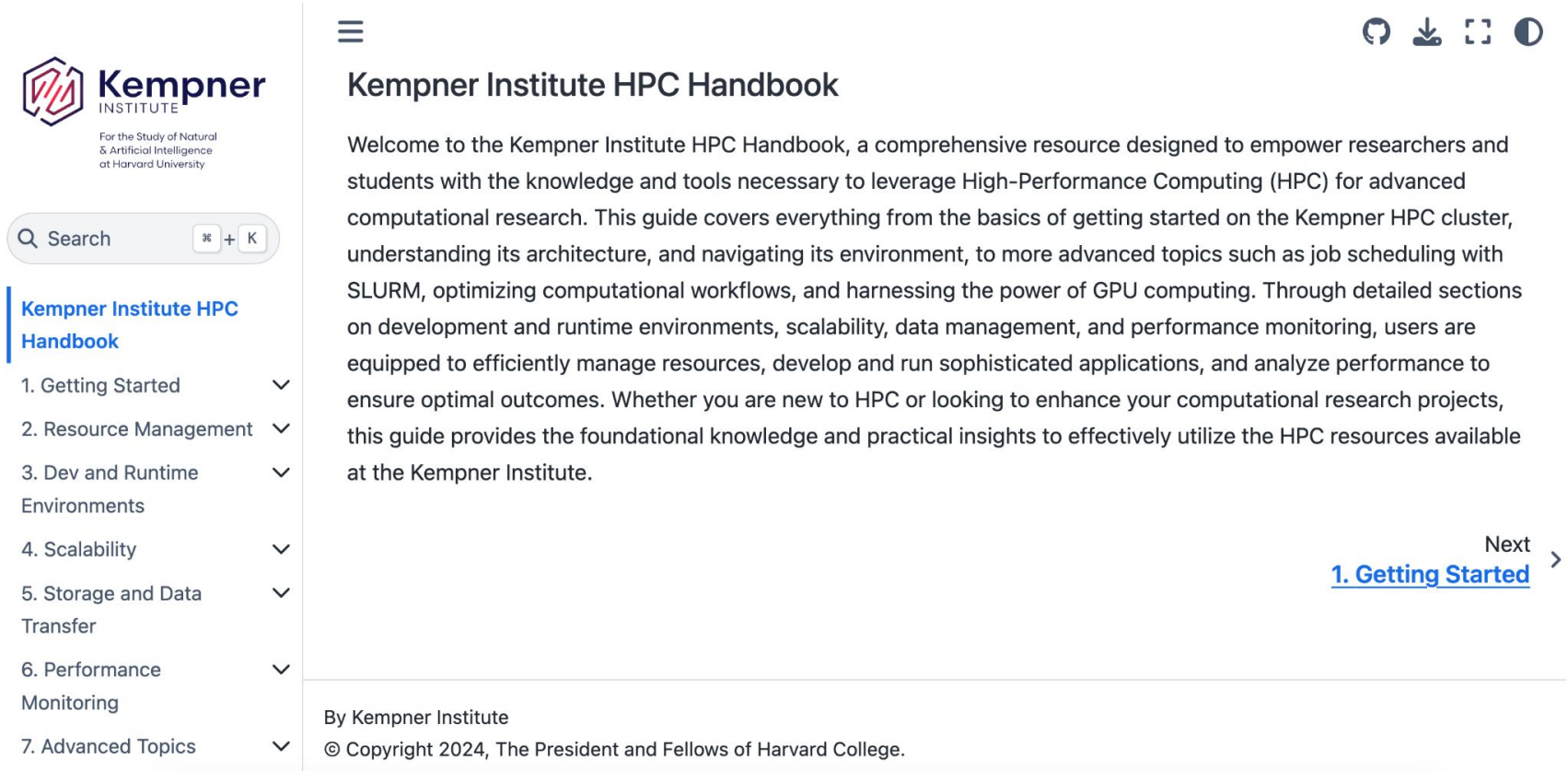
Data transfer, code synchronization, software modules & conda environments

3 **Job Management & Monitoring**

Fairshare, partitions, submitting Slurm basic & array jobs

HPC Handbook

https://kempnerinstitute.github.io/kempner-hpc-handbook



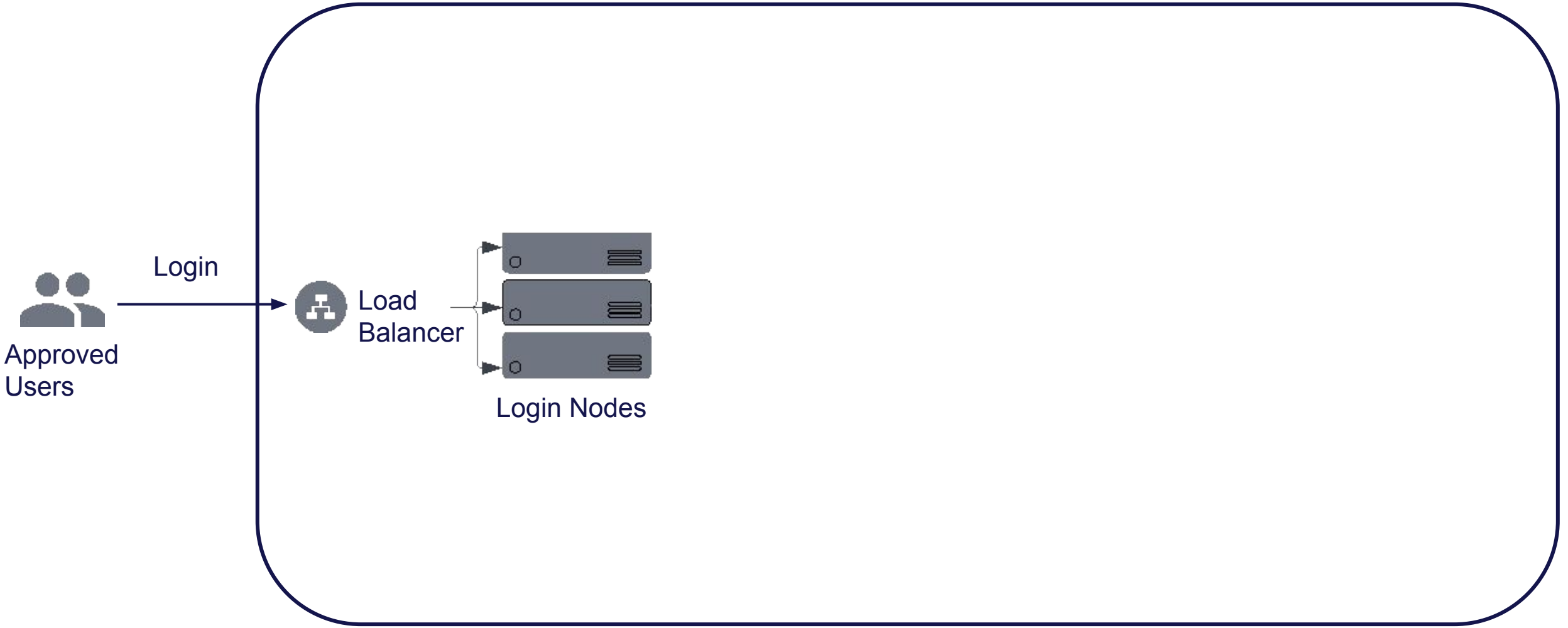
Acknowledgements

HPC Handbook developed by Research & Engineering Team

Workshop is based on Intro to HPC workshop we ran together in March

Kempner Cluster Specs

Specification	A100 40GB	H100 80GB
Total GPUs	144	384
Servers (per rack)	36	24
GPUs per Server	4	4
CPU Cores per Server	64	96
RAM per Server	1 TB	1.5 TB


Login
Approved
Users
Load
Balancer
Login Nodes

Exercise: Logging into the cluster

SSH Access

1) Open up terminal or use VSCode terminal
2) Type this in the command line:

```
ssh <username>@login.rc.fas.harvard.edu
```

3) Fill in your password & verification code (usually through DuoMobile)

   To configure 2FA: https://docs.rc.fas.harvard.edu/kb/openauth/

Bonus: Connect to the VPN and log in to the Open OnDemand Interface. Explore what it contains.

Accessing & Navigating the Cluster Page in the HPC Handbook

Persistent Storage

1. Home directory
   - 100 GB of persistent storage only accessible to you & is backuped
   - Good for long-term storage of files, checkpoints, datasets, etc
   - File path: /n/home<number>/<your user name>

2. Lab directory
   - 4 TB of persistent storage accessible all members of the lab
   - Good for long-term storage of files, checkpoints, datasets, etc
   - File path: /n/holylabs/LABS/<your lab name>
     /n/holylfs0x/LABS/<your lab name>
     /n/<your lab name>

Understanding Storage Options Page in the HPC Handbook

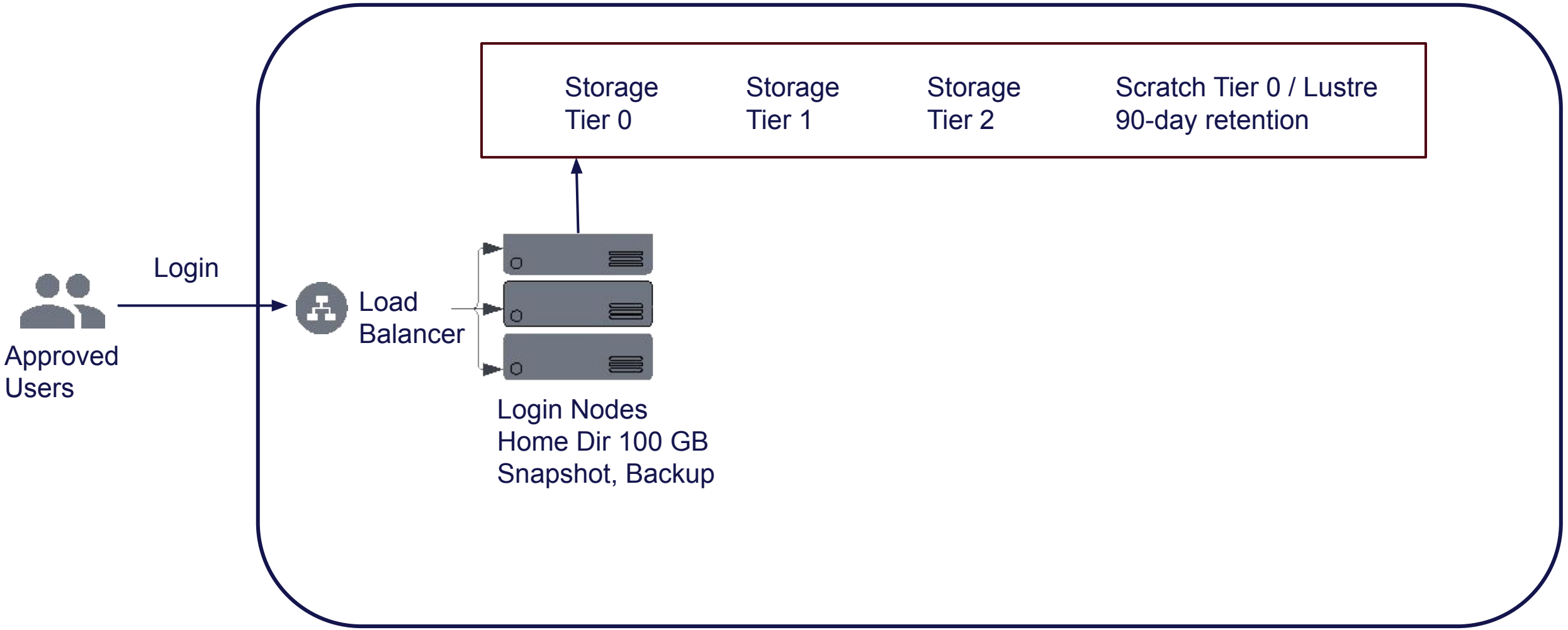
Temporary Storage

Good for storing data you are using for a job. Should be copied from persistent directories. Will be deleted after 90 days and you should treat it as if could be deleted any time

1. Scratch storage
   - 50 TB of temporary storage accessible all members of the lab
   - Need to file ticket with FASRC to get your own private directory in scratch storage space
   - `File path: /n/holyscratch01/<your lab name>`

2. VAST storage
   - 25 TB of storage for each lab
   - Offers improved performance for AI workflows, especially those requiring high I/O in computer vision ML workflows.
   - Accessible only from Kempner compute nodes for now
   - `File path: /n/vast-scratch/<your lab name>`

Understanding Storage Options Page in the HPC Handbook


Storage
Tier 0
Storage
Tier 1
Storage
Tier 2
Scratch Tier 0 / Lustre
90-day retention
Login
Approved
Users
Load
Balancer
Login Nodes
Home Dir 100 GB
Snapshot, Backup

Shared Data/Model Repository

ML Models

- CodeLlama
- EleutherAI
- OpenAI GPT 2
- Google T5 - base

ML Datasets

- C4 (for NLP)
  - “Colossal Clean Crawled Corpus” (C4) dataset, designed for training natural language processing model
- dolma
  - Dataset of 3 trillion tokens from a diverse mix of web content, academic publications, code, books, and encyclopedic materials.
- Updated ImageNet
  - Updated version of the ImageNet dataset, containing a wide variety of annotated images for visual object recognition

[Shared Data/Model Repository Page in the HPC Handbook]

Exercise: Exploring Storage

1) Check what directory you are in

```
pwd
```

2) Move to another directory (lab or scratch storage)

```
cd <file_path>
```

3) List all files in the current directory you are in

```
ls
```

Bonus: Explore the shared data repository. Read more about one of the shared models on HuggingFace (get link from Shared Data/Model Repository Page in the HPC Handbook)

Understanding Storage Options Page in the HPC Handbook

Agenda

1 **Orientation to Infrastructure**

Cluster access, architecture, storage tiers

2 **Development**

Data transfer, code synchronization, software modules & conda environments

3 **Job Management & Monitoring**

Fairshare, partitions, submitting Slurm basic & array jobs

Transferring Data

- Many possible methods for transferring data between local environments (such as your laptop) and the Kempner cluster.
- Choice will depend on the size and complexity of the data transfer

[Data Transfer Page in the HPC Handbook]

scp

Straight-forward command-line method for copying a few small files

```
scp [file path to location of file] [file path to destination of file]
```

If file is on the cluster, need to put user name and server information before that file path

Can add `-r` after scp above to copy a whole folder

Example:

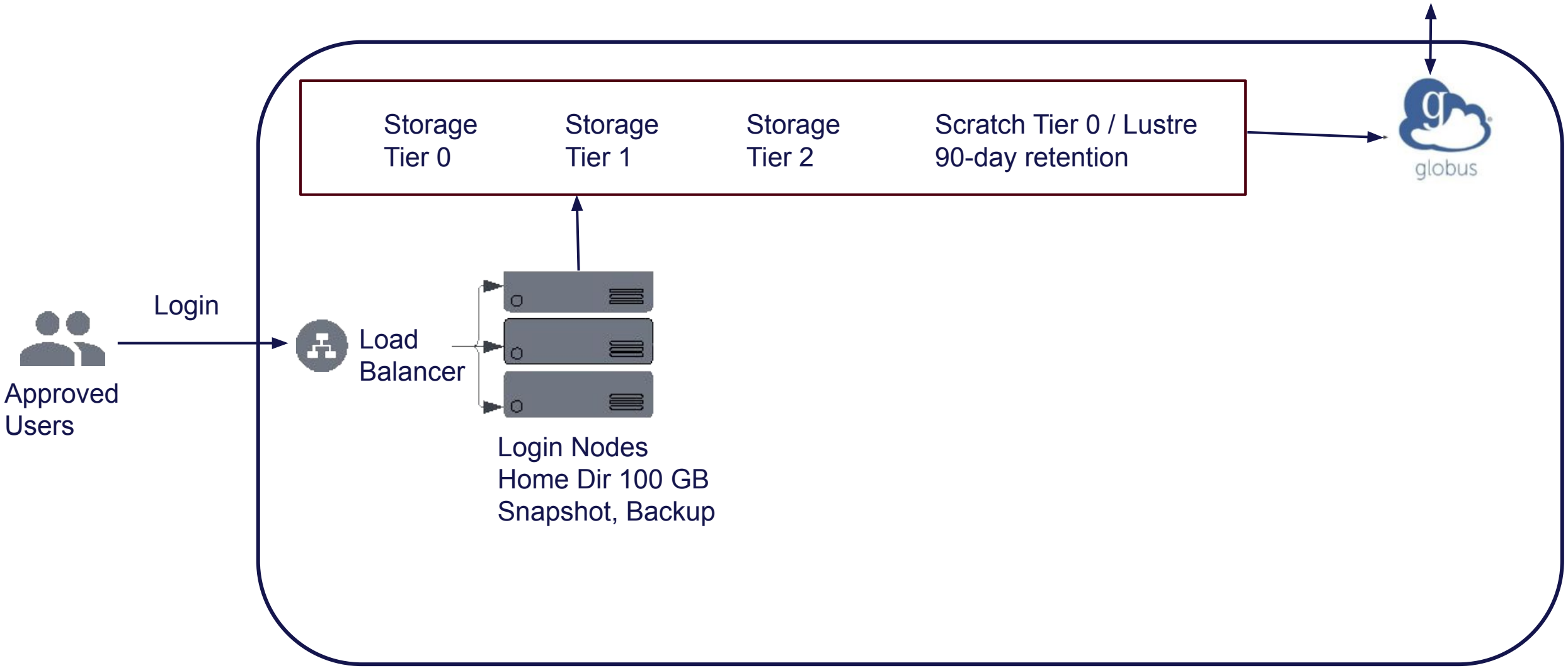
Jane Smith wants to transfer an entire directory, called results, from scratch storage on the cluster to her laptop's Documents folder. She can accomplish this with running the following command on her laptop.

```
scp -r
janesmith@login.rc.fas.harvard.edu:/n/holyscratch01/janesmith_lab/Users/janesmith/res
ults /Users/janesmith/Documents/
```

Globus

- Designed for secure and efficient transfer of large datasets
- User-friendly web interface
- Can be used for transferring between a cluster and local machine, or for sharing data with collaborators
- Set-up instructions in HPC handbook

Globus Section in the HPC Handbook


Storage
Tier 0
Storage
Tier 1
Storage
Tier 2
Scratch Tier 0 / Lustre
90-day retention
globus
Login
Approved
Users
Load
Balancer
Login Nodes
Home Dir 100 GB
Snapshot, Backup

Exercise: Transferring a File

1) Navigate to the Data_transfer_example folder here and download data.npy to your computer.
2) Use scp to transfer this data to your home directory on the cluster.

Bonus: If you'll be transferring large datasets or lots of files, start on Globus set up using instructions here.

Bonus: Read about rsync customizable arguments here.

Data Transfer Page in the HPC Handbook

Getting code on the cluster

Exercise: Synchronizing code with VSCode

1) Follow steps in Section 3.3.2 of the Using VSCode for Remote Development page of the HPC handbook. Don't move on to 3.3.3 yet
2) Create a new file in the VSCode workspace. Check your home directory on the cluster using terminal and make sure that new file is there.

Bonus: Install the Jupyter extension and create a jupyter notebook in VSCode. Open it and try running a cell of simple code. Instructions here.

Setting up your environment - Conda

A conda environment is a directory that contains a self-contained instance of Python along with a specific set of packages.

mamba is a drop-in replacement for conda that is generally much faster

Exercise: Setting up Conda Environment

1) Follow steps in Section 3.4.2 and 3.4.3 on the Using Conda Environment page to set up your first environment

   This will set it up in your home directory. For large conda environments or shared conda environments, you may want to set it up in your Lab directory

Agenda

1 **Orientation to Infrastructure**

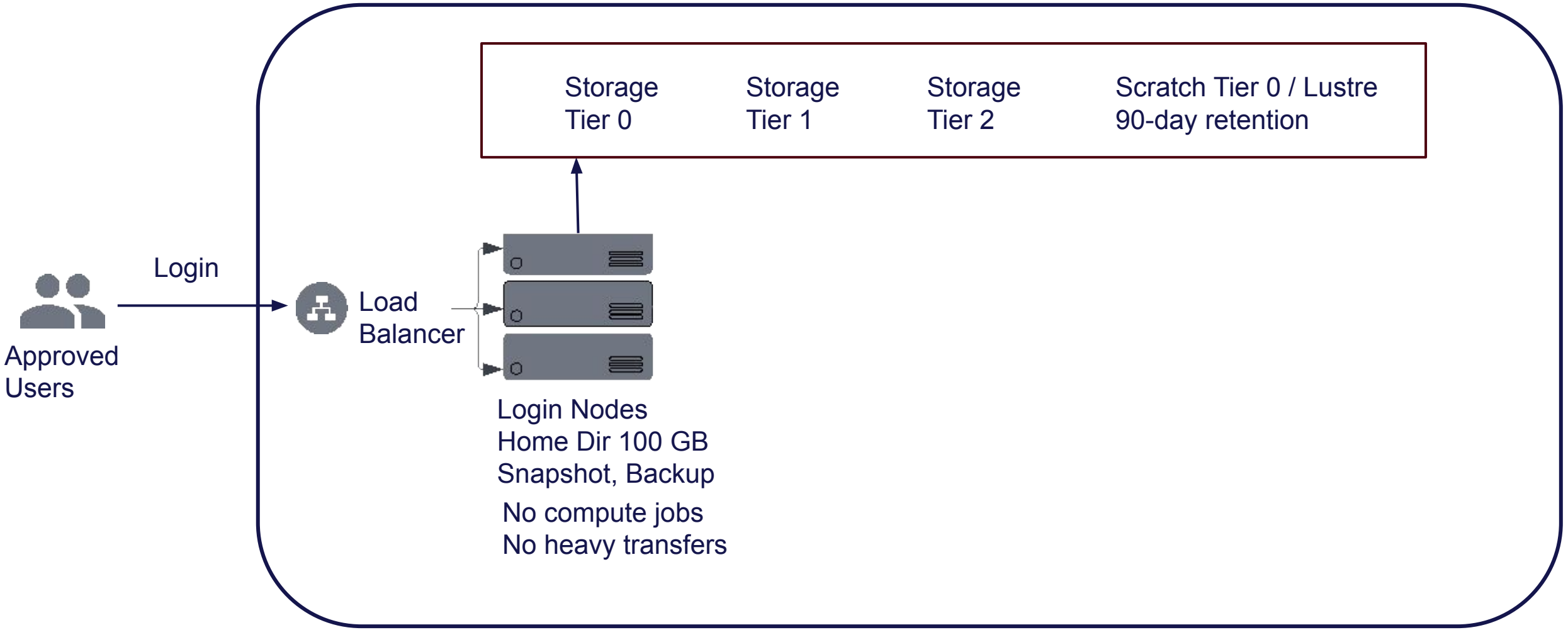
Cluster access, architecture, storage tiers

2 **Development**

Data transfer, code synchronization, software modules & conda environments

3 **Job Management & Monitoring**

Fairshare, partitions, submitting Slurm basic & array jobs


Storage
Tier 0
Storage
Tier 1
Storage
Tier 2
Scratch Tier 0 / Lustre
90-day retention
Login
Approved
Users
Load
Balancer
Login Nodes
Home Dir 100 GB
Snapshot, Backup
No compute jobs
No heavy transfers

Running Code on the Cluster

To run code on the cluster, you will need to request computational resources on compute nodes.

Each discrete request is called a job.

You can either run your jobs interactively or submit them as batch jobs. Interactive jobs are useful for testing and debugging, while batch jobs are useful for running large-scale simulations and analyses.

SLURM

SLURM is a job scheduler and resource manager, used to allocated resources to users and to schedule and manage jobs

SLURM accounts are used to track the usage of resources on the cluster. You need to specify the account to which the resources should be charged when submitting a job.

To see the accounts you have access to:

```
sshare -U -u <username>
```

[Understanding SLURM Page in the HPC Handbook]

Requesting Resources

Node:

A100: consists of 4 GPUs, 64 CPUs, with 1 TB of memory

H100: consists of 4 GPUs, 96 CPUs, with 1.5 TB of memory

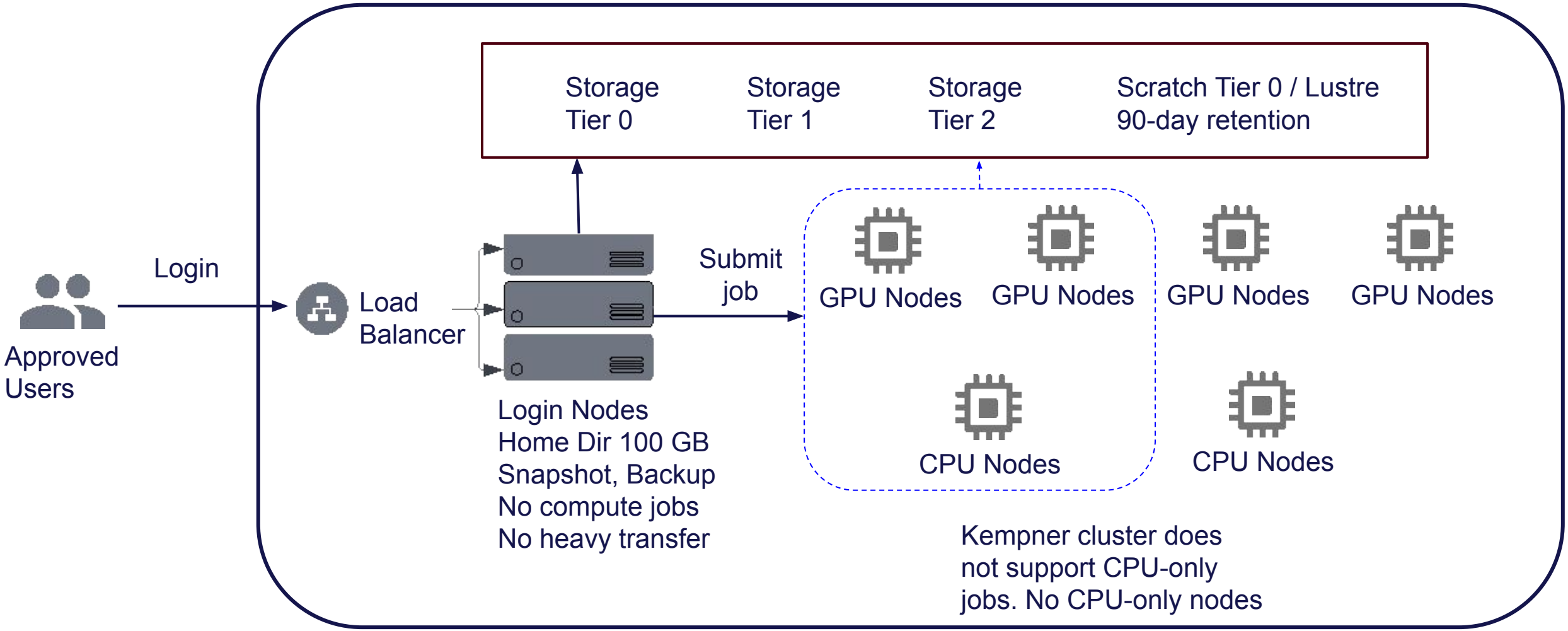
You'll need to decide # of nodes, # of GPUs, # of CPUs, and how much memory to request.

Some considerations:

- Do not request a GPU if your code does not use GPUs
  - Kempner Cluster is GPU jobs only - you can use general FASRC cluster for CPU only jobs
- Request only one GPU and one node unless your code is configured to make use of multiple (distributed training)
- Do not request more than 16/24 CPUs per GPU
- Do not request more than 250/375 GB per GPU

Kempner Cluster Specs

Specification	A100 40GB	H100 80GB
Total GPUs	144	384
Servers (per rack)	36	24
GPUs per Server	4	4
CPU Cores per Server	64	96
RAM per Server	1 TB	1.5 TB


Storage
Tier 0
Storage
Tier 1
Storage
Tier 2
Scratch Tier 0 / Lustre
90-day retention
Login
Approved
Users
Load
Balancer
Login Nodes
Home Dir 100 GB
Snapshot, Backup
No compute jobs
No heavy transfer
Submit
job
GPU Nodes
GPU Nodes
GPU Nodes
GPU Nodes
CPU Nodes
CPU Nodes
Kempner cluster does
not support CPU-only
jobs. No CPU-only nodes

SLURM Partitions

Partition	Description
`kempner`	This GPU block includes 2304 Intel Ice Lake cores and 144 Nvidia A100 40GB GPUs, with each water-cooled node featuring 64 cores, 1TB RAM, and 4 A100 GPUs, linked via HDR Infiniband, with a 7-day limit.
`kempner_h100`	This GPU block includes 2304 AMD Genoa cores, 96 Nvidia H100 80GB GPUs, water-cooled nodes with 96 cores, 1.5TB RAM, and 4 H100 GPUs, interconnected via NDR Infiniband, with a 3-day limit.
`kempner_requeue`	This partition utilizes `kempner` and `kempner_h100` partitions, designed for tasks that can be interrupted and restarted. This partition has a 7 day time limit.

[Understanding SLURM Page in the HPC Handbook]

Interactive Jobs in VSCode

Requesting interactive jobs via the command line:

```
salloc --partition=kempner_requeue --account=kempner_undergrads
--ntasks=1 --cpus-per-task=16 --mem=16G --gres=gpu:1
--time=00-03:00:00
```

You can set things up so that you are on the compute node in your interactive job in VScode, rather than the log in node

See instructions here

Open OnDemand Interactive Jobs Page in the HPC Handbook

Fairshare

- Kempner cluster uses a system called fairshare to determine prioritization and which jobs run when
- Jobs, particularly those that are resource intensive or are being run in labs with high recent usage, may not run immediately or on demand.
- Fairshare score is computed for each fairshare group (i.e. slurm account) based on prior usage of the cluster by that group and their share of the cluster
- The position of a job in the queue is based on a priority number assigned to job, which is calculated based on your fairshare score and the job age.

Fairshare Page in the HPC Handbook

SLURM Batch Job

To submit a batch job, you need a job script (i.e. my_job_script.sh) that specifies the resources required for the job and the commands to be executed. 3 sections:

- Resource and job specifications,
- Loading modules, setting environment variables, other pre-job tasks, and
- Running the job.

Submit job script with:

`sbatch my_job_script.sh`

```
#!/bin/bash
#SBATCH --job-name=my_job
#SBATCH --account=kempner_grads
#SBATCH --partition=kempner
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=16
#SBATCH --gpus-per-node=1
#SBATCH --time=0-01:00
#SBATCH --mem=256G
#SBATCH --output=my_job_output.out
#SBATCH --error=my_job_error.err
#SBATCH --mail-type=END
#SBATCH --mail-user=<username>@harvard.edu

# Load modules
module load python/3.10.9-fasrc01

# Activate conda environment (optional)

# Run the job
python my_script.py
```

Exercise: Submitting a batch job

1) Work through steps 1 - 6 SLURM Example 1 here

Bonus: Work through the rest of the steps

Batch Jobs Page in the HPC Handbook

SLURM Array Jobs

Extremely useful for scheduling multiple jobs at once, such as for parameter sweeps

Add `#SBATCH --array=1-12` to .sh file

Will execute 12 parallel tasks, each with a unique task ID that you can get with `$SLURM_ARRAY_TASK_ID`

Can make sure no more than 4 run at a time with:

```
#SBATCH --array=1-12%4
```

Exercise: Check out array job example

1) See SLURM Example 2 here

Other Examples

- **GPU Example 1**:
  - Model: ResNet-18
  - Dataset: CIFAR-10
  - Resources: 1 GPU (A100)
  - Job submission: Interactive
- **GPU Example 2**:
  - Model: ResNet-18
  - Dataset: CIFAR-10
  - Resources: 1 GPU (A100)
  - Job submission: Batch
  - Integration: Weights & Biases (wandb.ai)
- **GPU Example 3**:
  - Model: ResNet-18
  - Dataset: CIFAR-10
  - Resources: 1 GPU (A100)
  - Job submission: Batch
  - Integration: Weights & Biases (wandb.ai)
  - Job Arrays
- **GPU Example 4**:
  - Model: ResNet-50
  - Dataset: CIFAR-10
  - Resources: 2, 4 GPUs (A100)
  - Job submission: Batch
  - Integration: Weights & Biases (wandb.ai)
- **GPU Example 5**:
  - Model: ResNet-50
  - Dataset: Subset of imagenet
  - Resources: 4 GPUs, 8 GPUs (two node) (A100)
  - Job submission: Batch

Good Citizenship

- Only use the Kempner cluster for Kempner-related research

- Be considerate when submitting jobs:
  - Use the Kempner requeue for non-urgent work. This work may be requeued by higher priority work, so you should implement checkpointing in the event that something higher priority interrupts your run.
  - Never use the cluster for CPU-only jobs.
  - Do not request more than 12 A100 GPUs or 12 H100 GPUs at once.

Good Citizenship Continued

- Try to write efficient code that takes advantage of the GPUs you are requesting
  - Be mindful about tools like Jupyter notebooks, it is easy to accidentally bog down the cluster.

- Please ensure you are in the #cluster-users slack channel in the Kempner slack space. Make use of this channel if you run into any issues.

Where should you go for help?

1. HPC handbook
2. Contact FASRC via office hours or ticket
3. Post to the #cluster-users channel on the Kempner slack

Thank you