SAFARI Cluster Tutorial

June 26th 2023

ETH zürich

SAFARI
SAFARI Research Group

Executive Summary

Motivation

- We are steadily **moving** from the old **to the new cluster**
- The new cluster now has significantly **more servers**, **up-to-date software stack**, and is **more robust**

Problem

- **Using** the cluster **effectively** requires **insights on how it is set up**, its **capabilities**, and its intended use. Some of those are not widely known, and frequently asked about
- Need to re-introduce the cluster to **P&S students** and **new SAFARI members** frequently

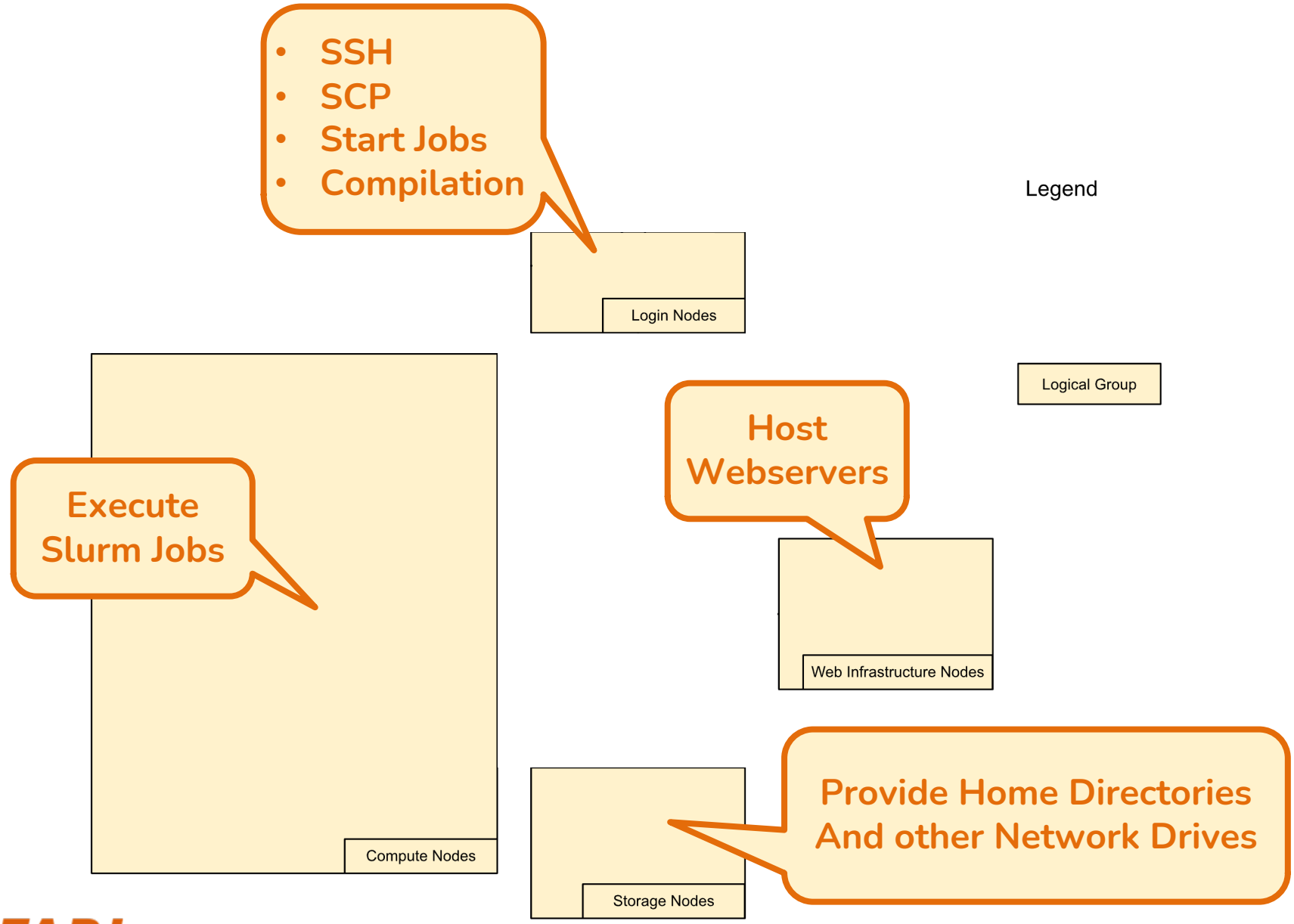
Goals

- Give you **insights** and **tips** on how to **effectively use the cluster**
- **Re-usable tutorial recording** to share with P&S students and new SAFARI members to reduce future overheads

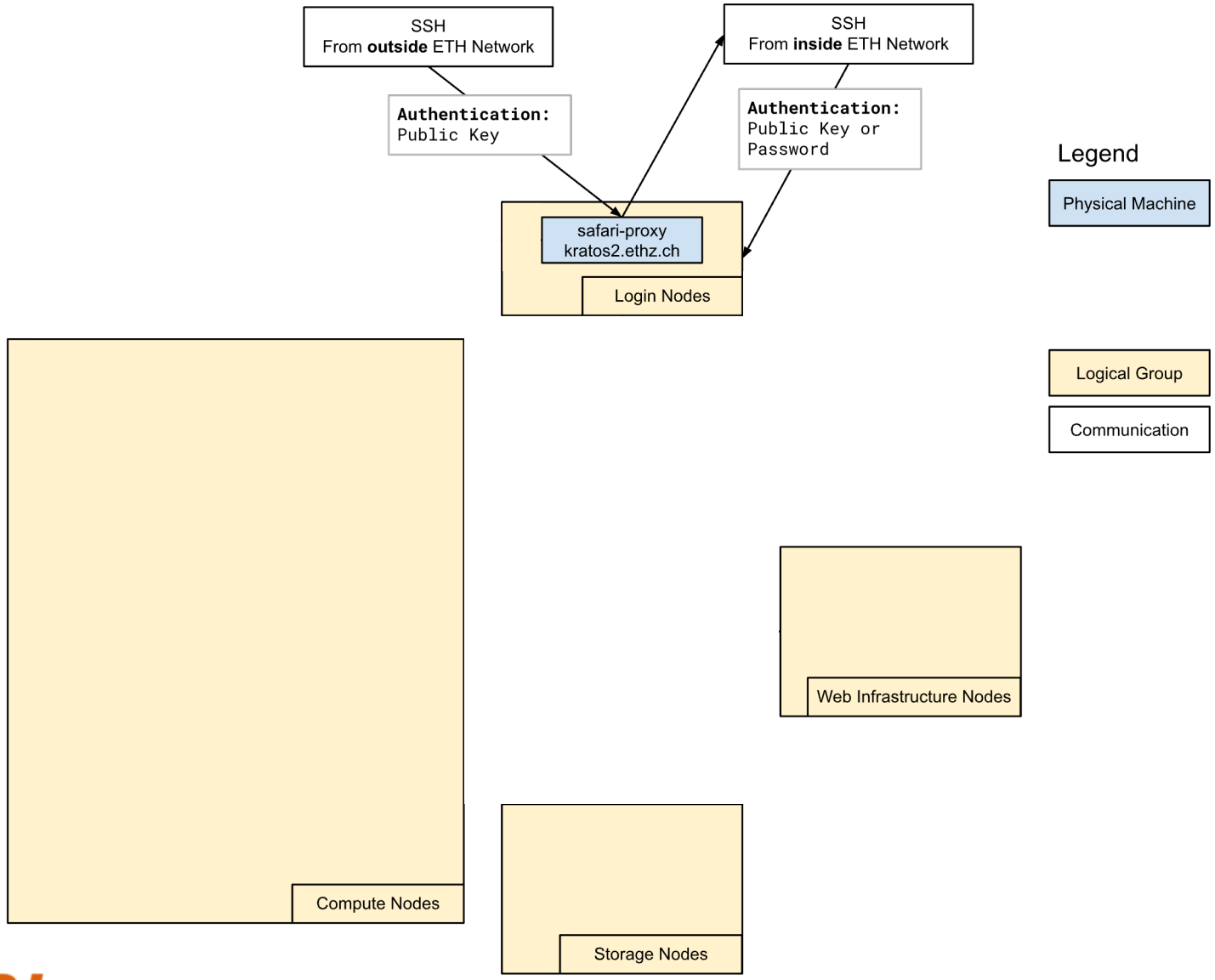
Outline

Outline

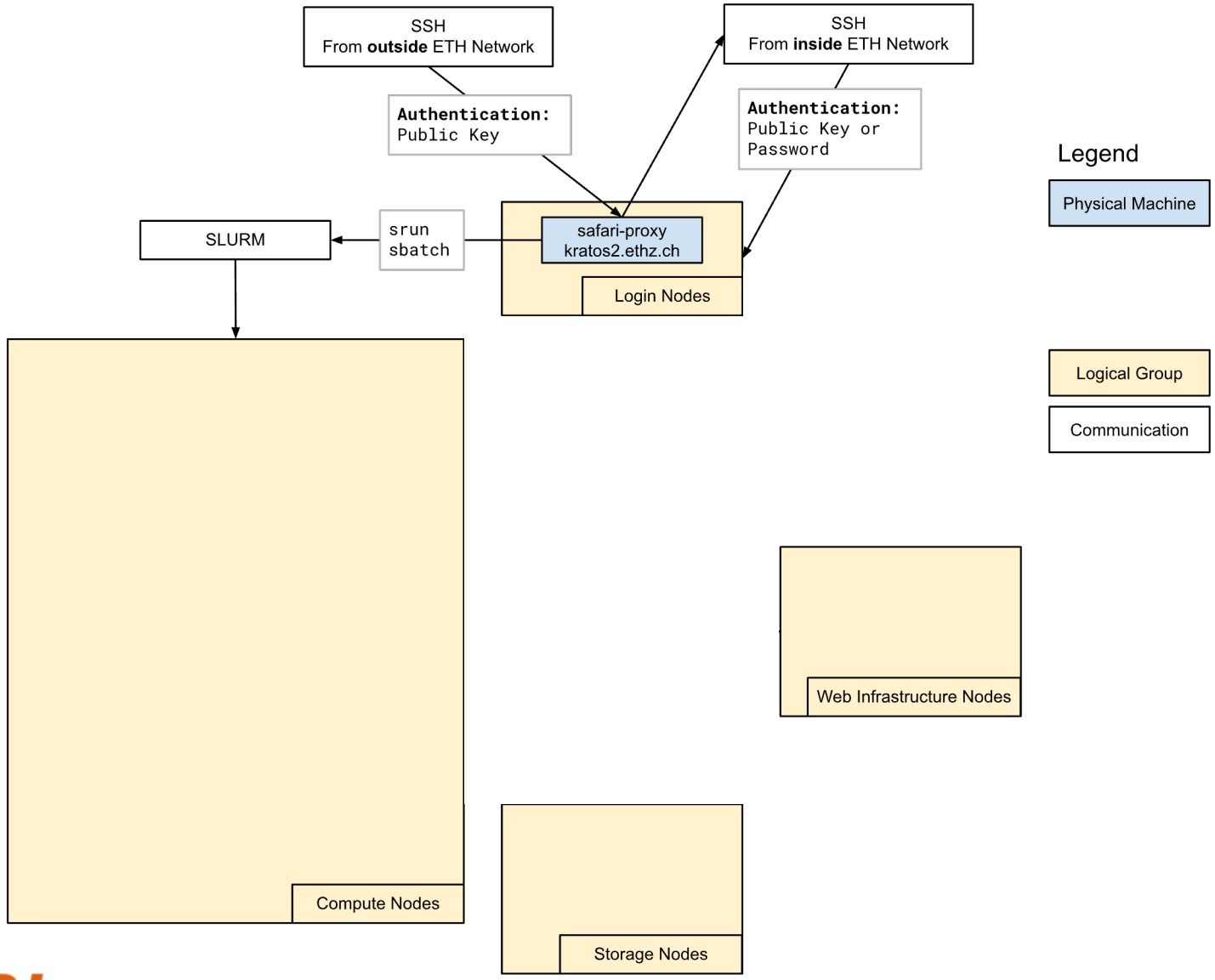
Cluster Overview



Cluster Overview

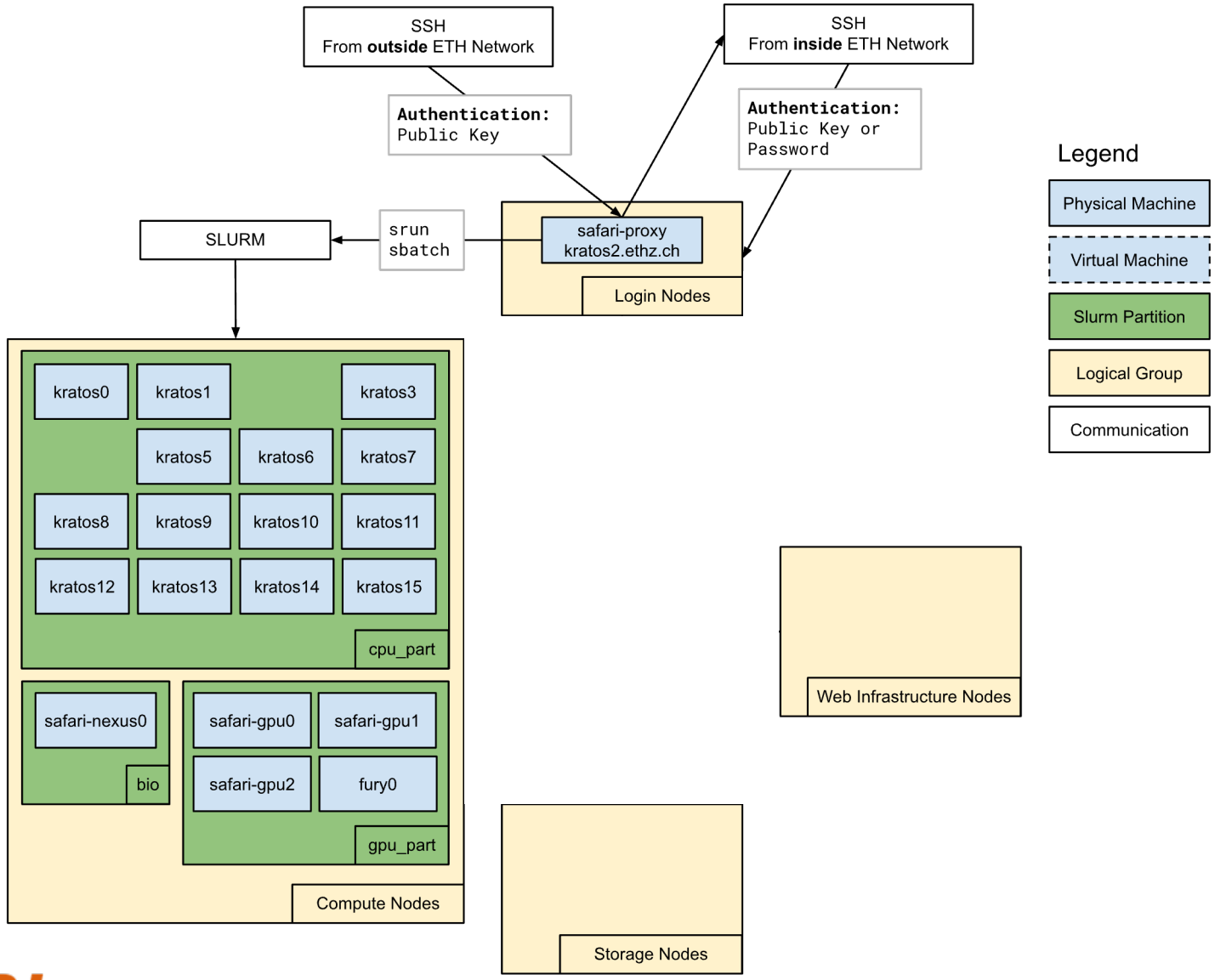


Cluster Overview

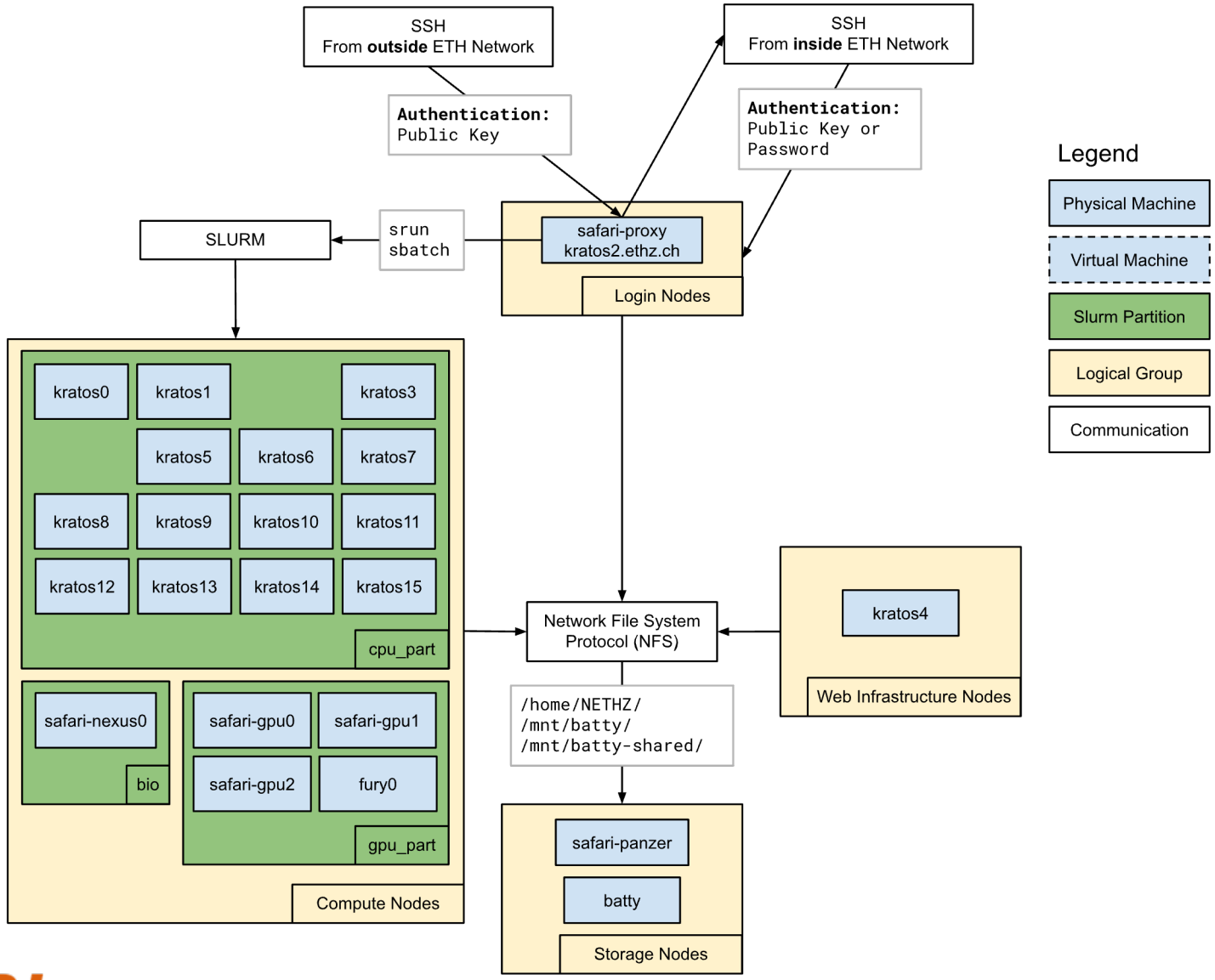


SAFARI

Cluster Overview



Cluster Overview



SAFARI

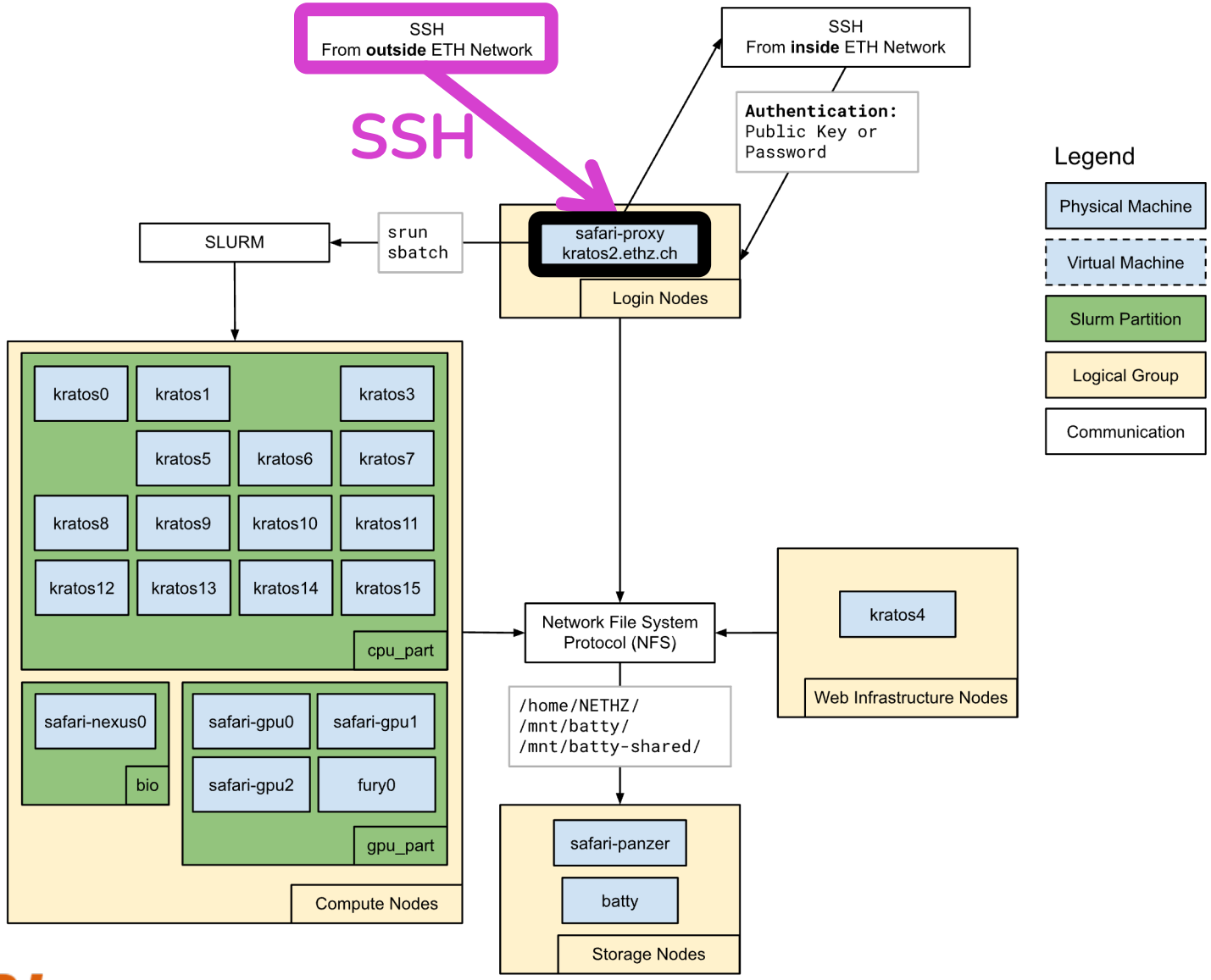
Outline

Preconditions

1. You need to be added to the cluster by one of the ITCs
   - Contact your supervisor, or directly contact one of the ITCs
     - Nisa Bostancı
     - Joël Lindegger
     - Ataberk Olgun
     - Can Firtina

2. You need to set up SSH keys
   - The cluster guide has step-by-step instructions

SSH to the Login Node (1)

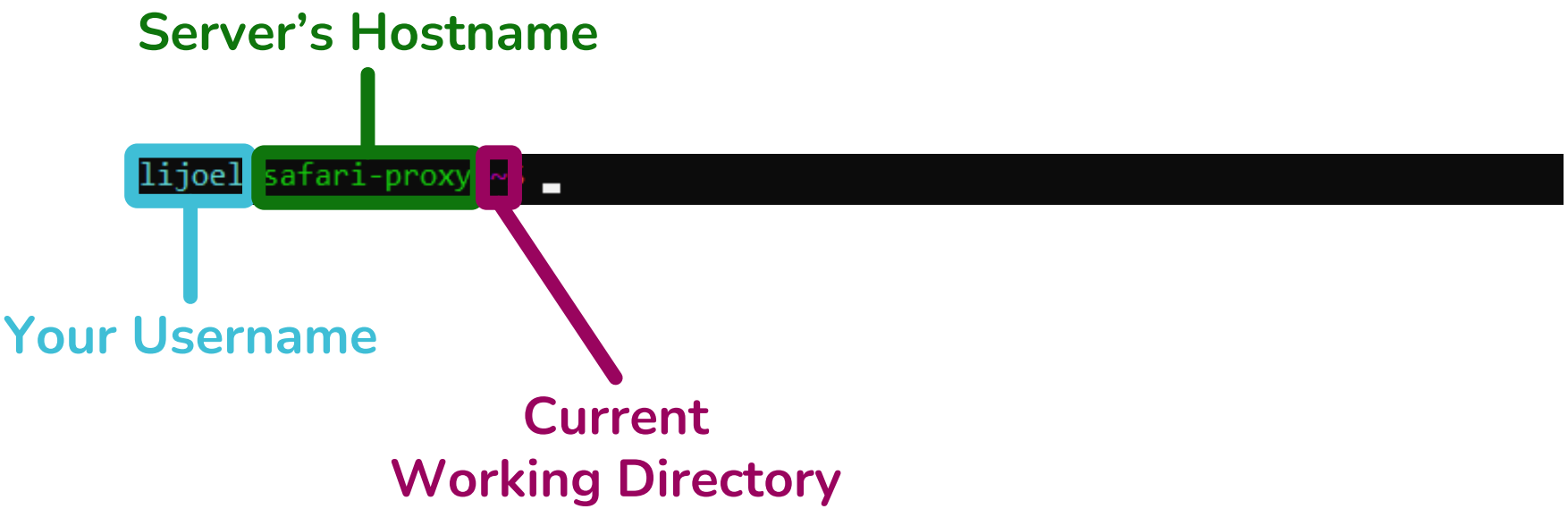


SAFARI

SSH to the Login Node (2)

```
C:\Users\joel>ssh safari
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-56-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Jun 26 04:38:19 AM CEST 2023

  System load:  0.0                Processes:               750
  Usage of /:   27.7% of 878.15GB  Users logged in:         12
  Memory usage: 6%                 IPv4 address for bond0:  129.132.63.167
  Swap usage:   1%                 IPv4 address for docker0: 172.17.0.1
  Temperature:  34.0 C

  => There are 2 zombie processes.

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

159 updates can be applied immediately.
78 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


     _ _   _                                  __          _
    | (_) (_)                 ____            / _|        (_)
  ___| |_   _ _ __ _ __ ___    / __ \   ___  __ _| |_ __ _ _ __ _
 / __| | | | | '__| '_ ` _ \  / / _` | / __|/ _` |  _/ _` | '__| |
 \__ \ | |_| | |  | | | | | | | (_| | \__ \ (_| | || (_| | |  | |
 |___/_|\__,_|_|  |_| |_| |_|  \ \__,_| |___/\__,_|_| \__,_|_|  |_|
                                \____/

Welcome to the SAFARI SLURM cluster.
Please follow the guidelines laid out here: https://docs.google.com/document/d/17p3t-5oT48FlontU2TXtIg7jTvdUT0xOJEKUQWh0jHo/edit?usp=sharing

For comments, questions, and/or requests, please contact the cluster maintainers:
        * Ataberk
        * Joel
Last login: Mon Jun 26 04:38:19 2023 from 85.2.65.130
lijoel@safari-proxy:~$
```

Bash Prompt Colors



Running Commands on the Login Node

```
lijoel@safari-proxy:~$ echo "Hello world! This is $(hostname)."
Hello world! This is safari-proxy.
```

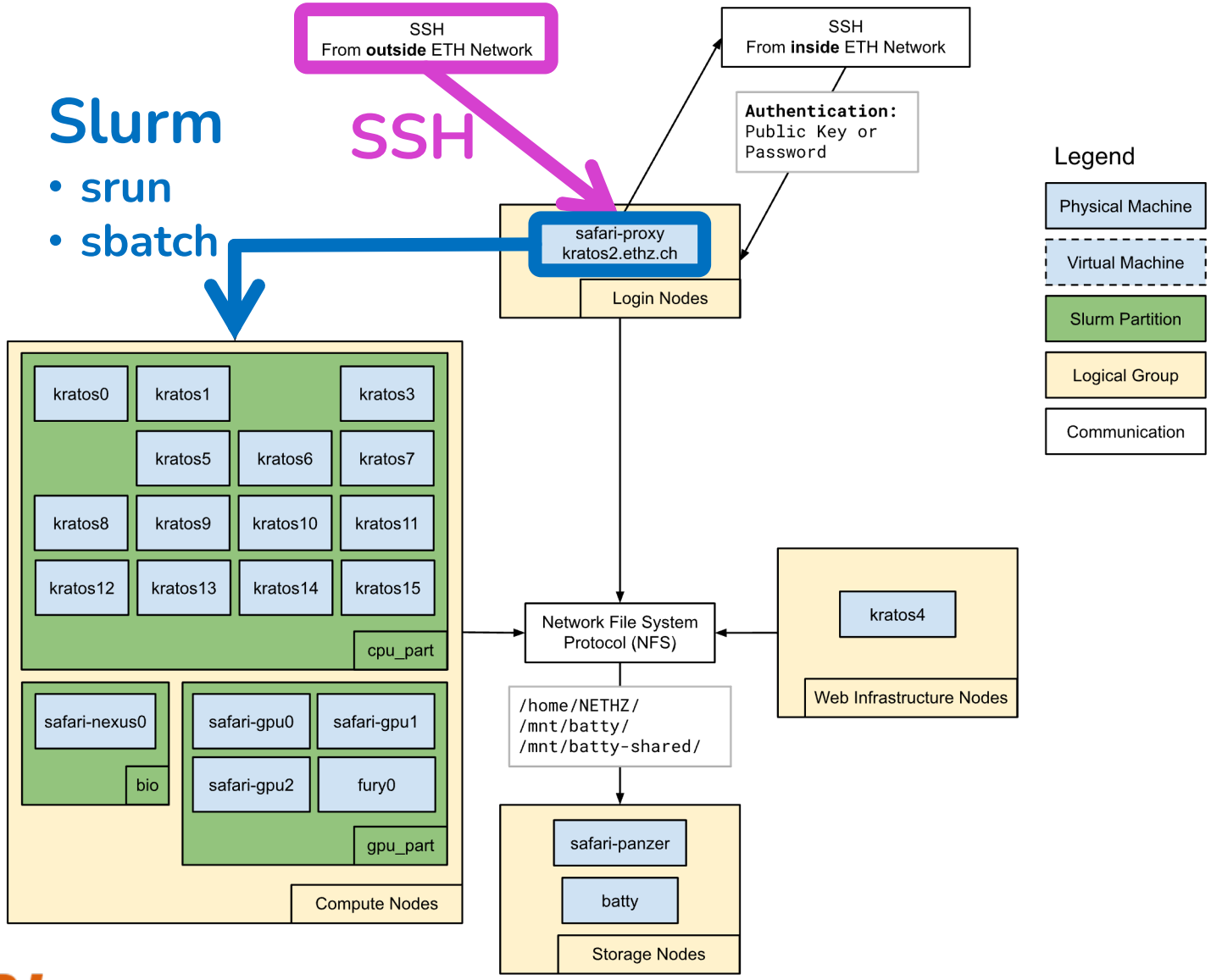
- This example runs the **echo** program directly on the **login node safari-proxy**
  - echo simply prints its command line arguments to the output
  - $(hostname) retrieves the machine's hostname (safari-proxy)

What to Run on the Login Node

- You **should** use the login node for…
  - **Moving around** or **editing files**
  - **Small compilation scripts**
    - E.g., g++ running for a few seconds
  - **Interactive applications** that aren't **compute-heavy**
    - E.g., the VSCode server
  - **Small workloads**
    - E.g., debugging your program with a small dataset that isn't compute-heavy

- You **should not** use the login node for…
  - **Heavy** and **long-running workloads** of any kind
    - E.g., applications that require 10s of GiB memory and many threads
  - **Long compilation scripts**
    - E.g., recompiling gcc
  - Run these on **computes nodes** instead!

Use Slurm to Access Compute Nodes



SAFARI

Recall: Commands on the Login Node

```
lijoel@safari-proxy:~$ echo "Hello world! This is $(hostname)."
Hello world! This is safari-proxy.
```

- This example runs the **echo** program directly on the **login node safari-proxy**
  - echo simply prints its command line arguments to the output
  - $(hostname) retrieves the machine's hostname (safari-proxy)

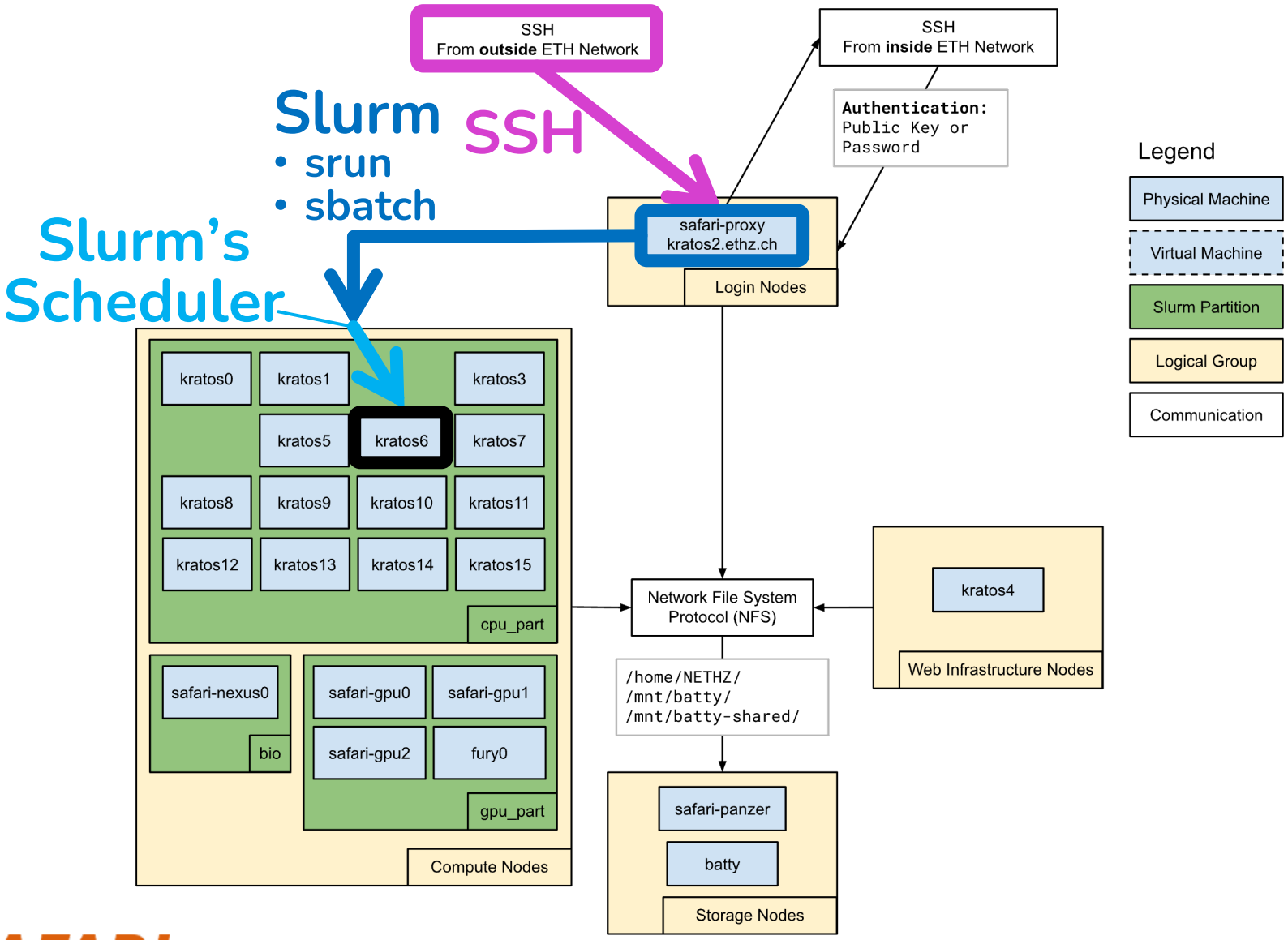
Running Commands on Compute Nodes

```
lijoel@safari-proxy:~$ srun hostname
kratos6
```

- This example runs the **hostname** program **using srun**
  - hostname prints the machine's hostname (kratos6)

- This got **executed on kratos6**, not safari-proxy!
  - Slurm automatically chose an available compute node
  - The output was returned from kratos6 to safari-proxy

Slurm Scheduling



SAFARI

“Pseudo SSH”: Interactive Bash via Slurm

- **Direct SSHing** into compute nodes is **disabled**
  - Ensures one cannot accidentally interfere with running experiments
  - Simplifies server management

- **Alternative** via Slurm: **bash over srun**

```
lijoel@safari-proxy:~$ srun --pty bash -l
lijoel@kratos6:~$
```

Slurm will immediately forward input and output of the encapsulated command

Bash in --login mode enables colored prompt and http(s)_proxy environment variables

“Pseudo SSH” to a Specific Node

- You can specifiy a node with **-w [nodename]**

```
lijoel@safari-proxy:~$ srun --pty -w kratos7 bash -l
lijoel@kratos7:~$
```

- Specifying **GPU** or **Bio** nodes requires **specifying the partition**
  - GPU nodes

```
lijoel@safari-proxy:~$ srun --pty -p gpu_part -w safari-gpu1 bash -l
lijoel@safari-gpu1:~$
```

  - Bio nodes

```
lijoel@safari-proxy:~$ srun --pty -p bio -w safari-nexus0 bash -l
lijoel@safari-nexus0:~$
```

“Pseudo SSH” is too annoying? We got you.

- We expect people to “Pseudo SSH” frequently
- To save you from re-typing the correct options every time, we’ve prepared the **slurmsh script** for you

- **Interactive bash session**

```
lijoel@safari-proxy:~$ slurmsh kratos9
lijoel@kratos9:~$
```

- **Non-interactive command**

```
lijoel@safari-proxy:~$ slurmsh kratos9 hostname
kratos9
```

Resource Reservations

- A key concept in Slurm are "**resources**"
  - CPU cores
  - Memory
  - GPUs

- By **default**, each Slurm job requests **1 CPU thread**
  - Resources can be **explicitly specified**, e.g., 16 threads:

```
lijoel@safari-proxy:~$ srun -c 16 hostname
kratos5
```

- Slurm is **aware of each machine's resources**
  - E.g., when specifying 160 threads, the job gets scheduled that has a sufficient number of threads

```
lijoel@safari-proxy:~$ srun -c 160 hostname
kratos10
```

Reservations avoid Oversubscription

- Slurm attempts to find a machine that has the requested amount of resources available
  - I.e., not currently used by another reservation
- If that's not possible, the job will be queued
- Here I force the issue by specifying kratos5 (which is busy), and requesting all of its 48 threads

```
[lijoel@safari-proxy:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             69190  cpu_part     2.sh jonschmi  R 4-16:59:38      1 kratos5
             69618  cpu_part     gups   aolgun  R 4-16:59:38      1 kratos5
             69619  cpu_part     gups   aolgun  R 4-16:59:38      1 kratos5
             69620  cpu_part     gups   aolgun  R 4-16:59:38      1 kratos5
[lijoel@safari-proxy:~$ srun -c 48 -w kratos5  hostname
 srun: job 72718 queued and waiting for resources
```

Reserving an Entire Machine

- Sometimes, it is useful to reserve an **entire machine**
  - E.g., **performance isolation** experiments

```
lijoel@safari-proxy:~$ srun --exclusive hostname
kratos6
```

- Blocks any other jobs from getting scheduled to the same machine
  - **Use sparingly**, to avoid hogging SAFARI's shared resources

Reserving GPUs

```
lijoel@safari-proxy:~$ srun --gres=gpu:1 -p gpu_part hostname
safari-gpu1
```

- Note that this requires **specifying a partition** that includes GPU machines
  - If you forget, you will receive an error as follows:

```
lijoel@safari-proxy:~$ srun --gres=gpu:1 hostname
srun: error: Unable to allocate resources: Requested node configuration is not available
```

srun vs sbatch

- **srun** is almost just a wrapper around a command

```
lijoel@safari-proxy:~$ srun hostname
kratos6
```

  - **Input is forwarded** to the command
  - **Output is printed** to the terminal
  - While the command is running, the **terminal remains blocked**
  - **Closing the terminal kills the command**

- **sbatch** is a **detached** version of srun

```
lijoel@safari-proxy:~$ sbatch --wrap "hostname"
Submitted batch job 72617
lijoel@safari-proxy:~$ cat slurm-72617.out
kratos6
```

  - **Output** is written **to file**
  - **Terminal** is **available immediately**
  - **Closing the terminal has no effect** on the command

sinfo

- Print list of available nodes
  - Status
  - Partition

```
lijoel@safari-proxy:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
cpu_part*    up   infinite      2    mix kratos[5-6]
cpu_part*    up   infinite      1  alloc kratos10
cpu_part*    up   infinite      8   idle kratos[7-9,11-15]
gpu_part     up   infinite      2   idle safari-gpu[1-2]
```

squeue

- Print list of queued and running jobs

```
lijoel@safari-proxy:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             69190  cpu_part     2.sh jonschmi  R 4-08:37:03      1 kratos5
             69618  cpu_part     gups   aolgun  R 4-08:37:03      1 kratos5
             69619  cpu_part     gups   aolgun  R 4-08:37:03      1 kratos5
             69620  cpu_part     gups   aolgun  R 4-08:37:03      1 kratos5
             72576  cpu_part     wrap omulaimi  R 1-08:57:36      1 kratos6
             72588  cpu_part     bash fmulonde  R   15:04:04      1 kratos10
```

Outline

Local Storage

- Refers to a storage device attached to the machine locally
  - I.e., not over the network
  - E.g., an SSD

- You can access the local SSD on each machine at **/mnt/local**
  - Create a directory **named** according to **your username**

```
lijoel@safari-proxy:~$ mkdir /mnt/local/lijoel
lijoel@safari-proxy:~$ ls /mnt/local
aolgun  geraldod  lijoel  swap.swap
```

- Data on local storage is not shared between nodes

```
lijoel@safari-proxy:~$ srun ls /mnt/local
swap.swap
```

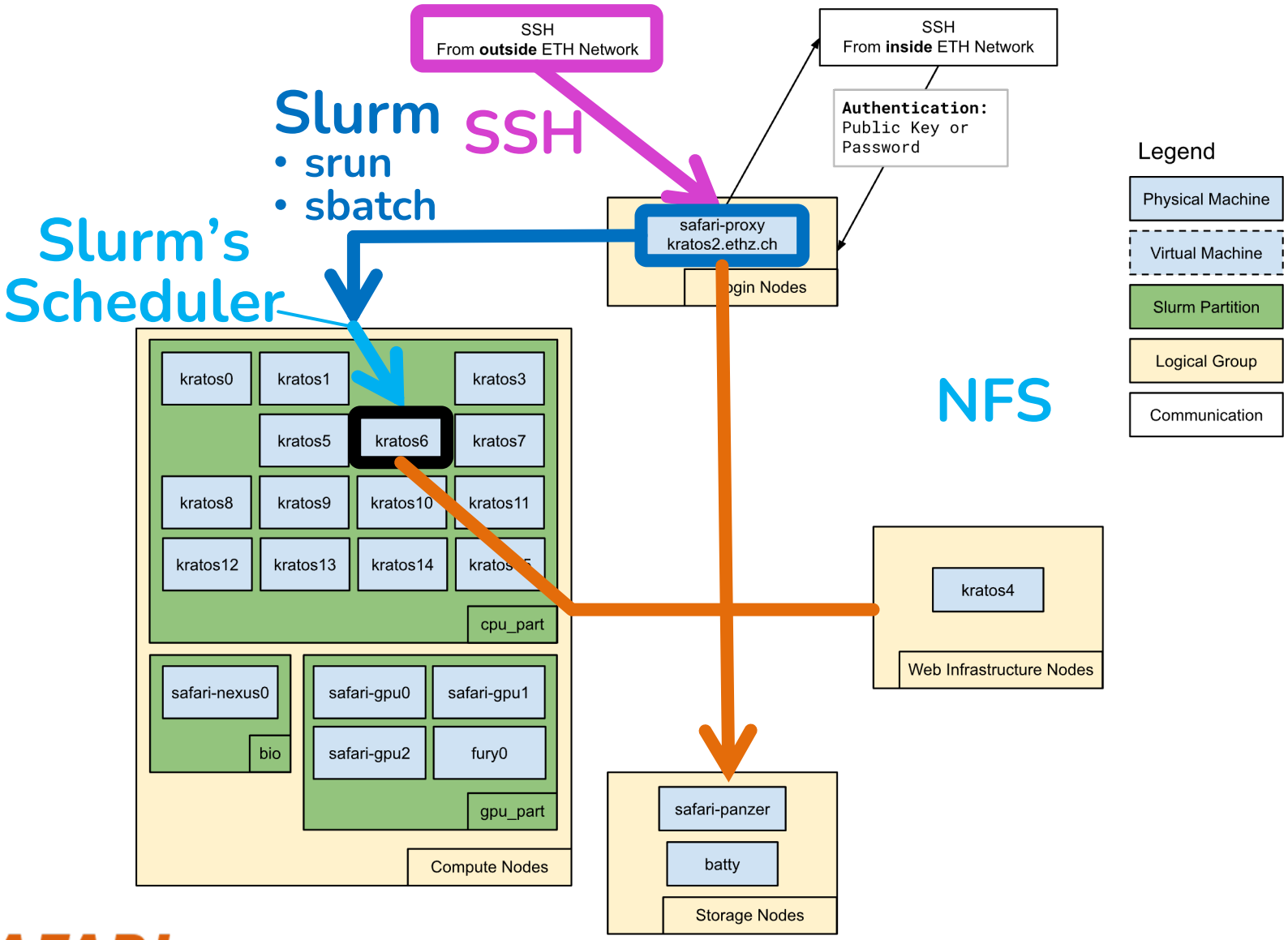
Network Drives (1)

- Refers to a storage device attached to the machine over the network
  - E.g., a NAS

- On our cluster, the following network drivers are **mounted on all machines**
  - **Home directories** at /home/[username]
    - Also at /mnt/panzer/[username]
  - **/mnt/batty/[username]**
    - You can create a directory for yourself if it doesn't exist
  - **/mnt/batty-shared**
    - Mostly bioinformatics data used by multiple SAFARI members

```
lijoel@safari-proxy:~$ echo "Hello World" > hello_world.txt
lijoel@safari-proxy:~$ srun cat hello_world.txt
Hello World
```

Network Drives (2)



When to Use Network Drivers

- You **should** use network drivers for…
  - **Infrequent file accesses**
  - **Large, batched file accesses**
  - **Mass storage**

- You **should not** use network drives for…
  - **Frequent file accesses**
    - E.g., script a script that writes to a debug log 1000 times per second
  - **High bandwidth requirement**
  - **Data that can be analyzed and reduced locally**
    - E.g., a large execution trace that can be analyzed locally, and only the summary needs to be written to a network drive
  - Run these on **local storage** instead!

An Example for Using Local Storage

```
#!/bin/bash

mkdir -p /mnt/local/lijoel/experiment123 # -p creates the entire hierarchy at once
./my_program_with_frequent_logs > /mnt/local/lijoel/experiment123/log.txt
mv /mnt/local/lijoel/experiment123/log.txt /mnt/panzer/lijoel/
rm -r /mnt/local/lijoel/experiment123 # clean up after yourself!
```

- Create directory in /mnt/local
- Write to directory (e.g., at high frequency)
- Possibly analyze data locally
- Copy output to network drive
- Clean up local storage

Outline

Methods for Installing Packages

- **User-space package managers**
  - Pip
  - Conda
- **Docker**
- **Build from Source**

Outline

Connectivity Depends on IP Type (1)

- **Public IPs**
  - Can access internet
  - Ingoing and outgoing connections possible
  - kratos2, kratos4

- **NATed Ips**
  - Can access internet
  - Outgoing connections possible
  - safari-gpu[0-2]

- **Private Ips**
  - Cannot access internet
  - All other machines

Connectivity Depends on IP Type (2)

- **Detailed list of IPs**
  - https://docs.google.com/spreadsheets/d/1L8gNyUgYbQK9Cxj74PpQQEuv6XSyMbhnndcxe5RjtXc/edit?usp=sharing

Outline

Documents

- **Cluster Guide**
  - https://docs.google.com/document/d/17p3t-5oT48FlontU2TXtlg7jTvdUT0xOJEKUQWh0jHo/edit?usp=sharing

- **Machine Properties List**
  - https://docs.google.com/spreadsheets/d/1L8gNyUgYbQK9Cxj74PpQQEuv6XSyMbhnndcxe5RjtXc/edit?usp=sharing

SAFARI Cluster Tutorial

June 26th 2023

ETH zürich

SAFARI
SAFARI Research Group