NASP集群-2024

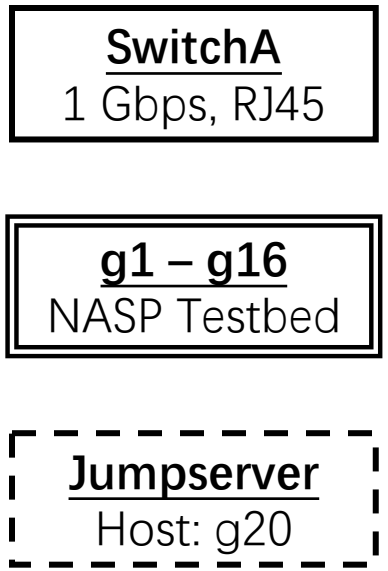
从重装到重装

熊典

2024-06-26

集群架构

图例



一台物理上存在的设备

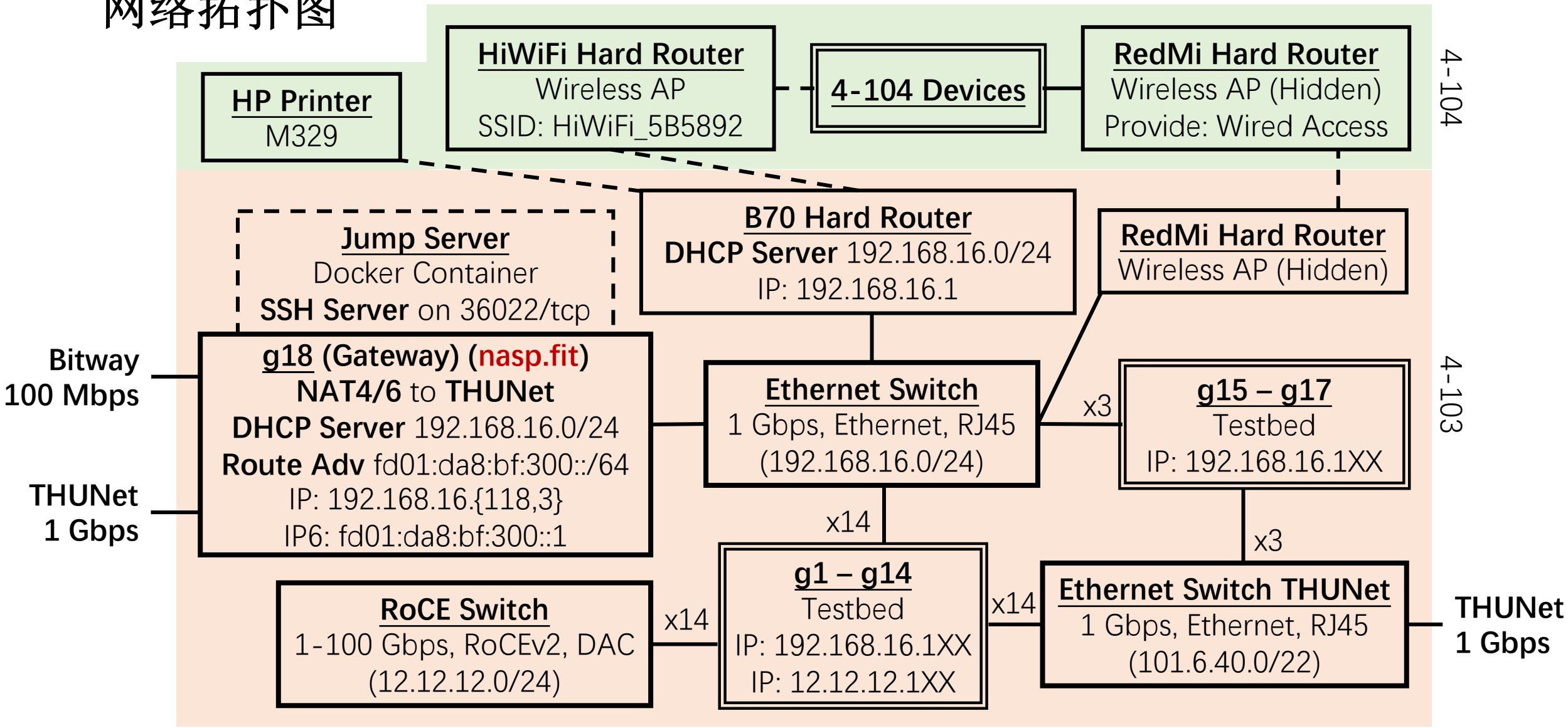
若干台物理上存在的设备的简化

一台虚拟设备（基于 KVM 或 Container）

网线（铜缆或者同轴线）

无线连接、无线桥接

NASP集群
网络拓扑图



集群总览

集群总览

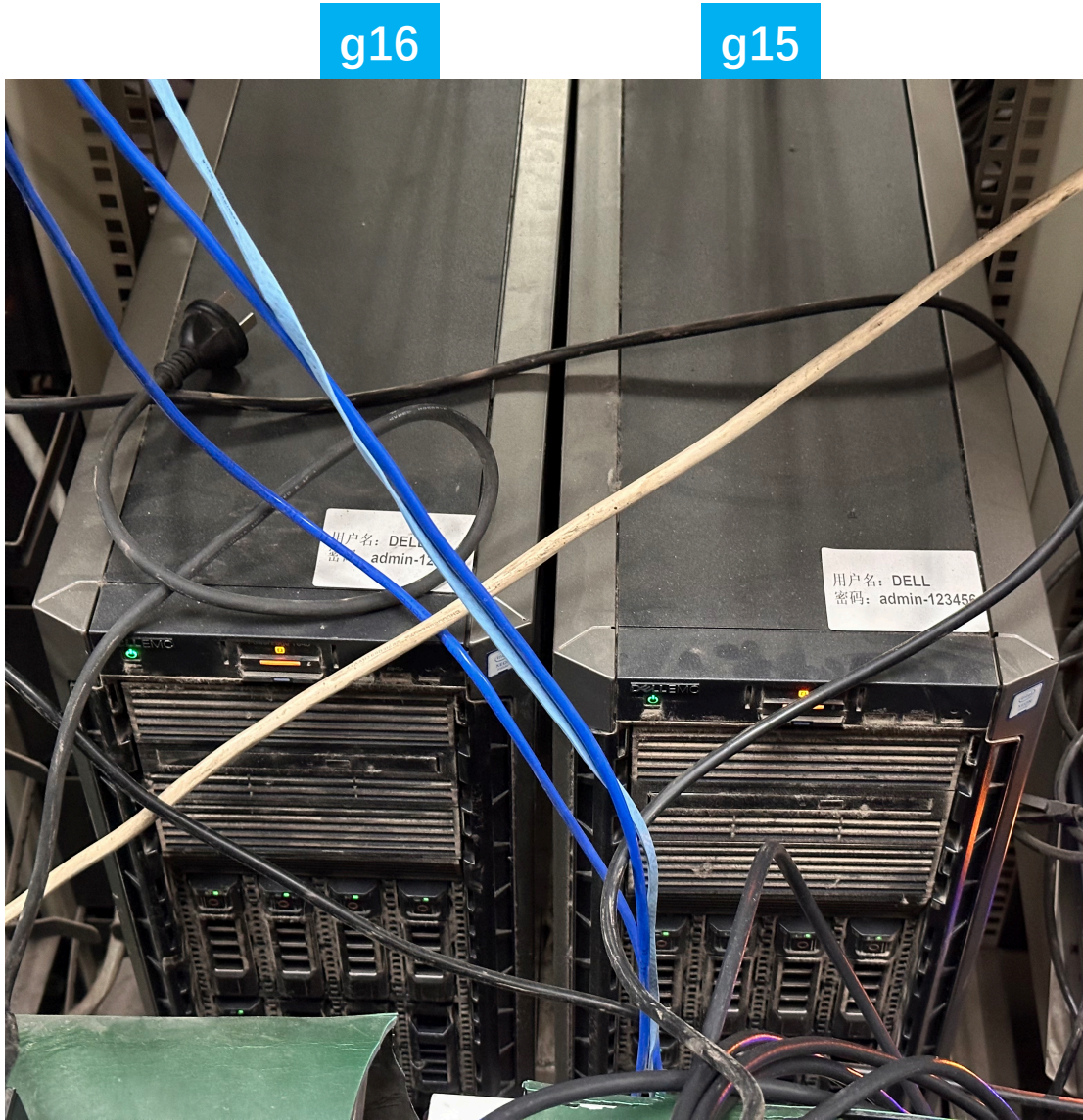

hard routers
g1
g2
g5
g6
eth switches
g7
g8
g3
g4
roce switch
kvm switch
g11
g12
g9
g10
console
g13
g14
g16
g15
g17,g18

g1 – g14（4U 塔式服务器）

- CPU
  - g1 – g9： 2 * Intel Xeon E5-2630 v3（8c16t）
  - g10 – g14： 2 * Intel Xeon E5-2680 v4（14c28t）
- GPU
  - K40c，显存 16 GB，计算能力 3.5。最高支持 cuda 11.3/11.4。
- Host Memory
  - 128 GB，DDR4，2400 MHz。
- NIC
  - Mellanox ConnectX-5，最高支持 100 Gbps。
  - 若干板载 1 GbE 网卡。

g15,g16（5U 塔式服务器）

- **Aka. lm1,lm2.**
- CPU
  - 2 * Intel Xeon Gold 5218R（20c40t）
- GPU
  - A6000，显存 48 GB，计算能力 8.6。
- Host Memory
  - 256 GB，DDR4，2666 MHz。
- NIC
  - 若干板载 1 GbE 网卡。



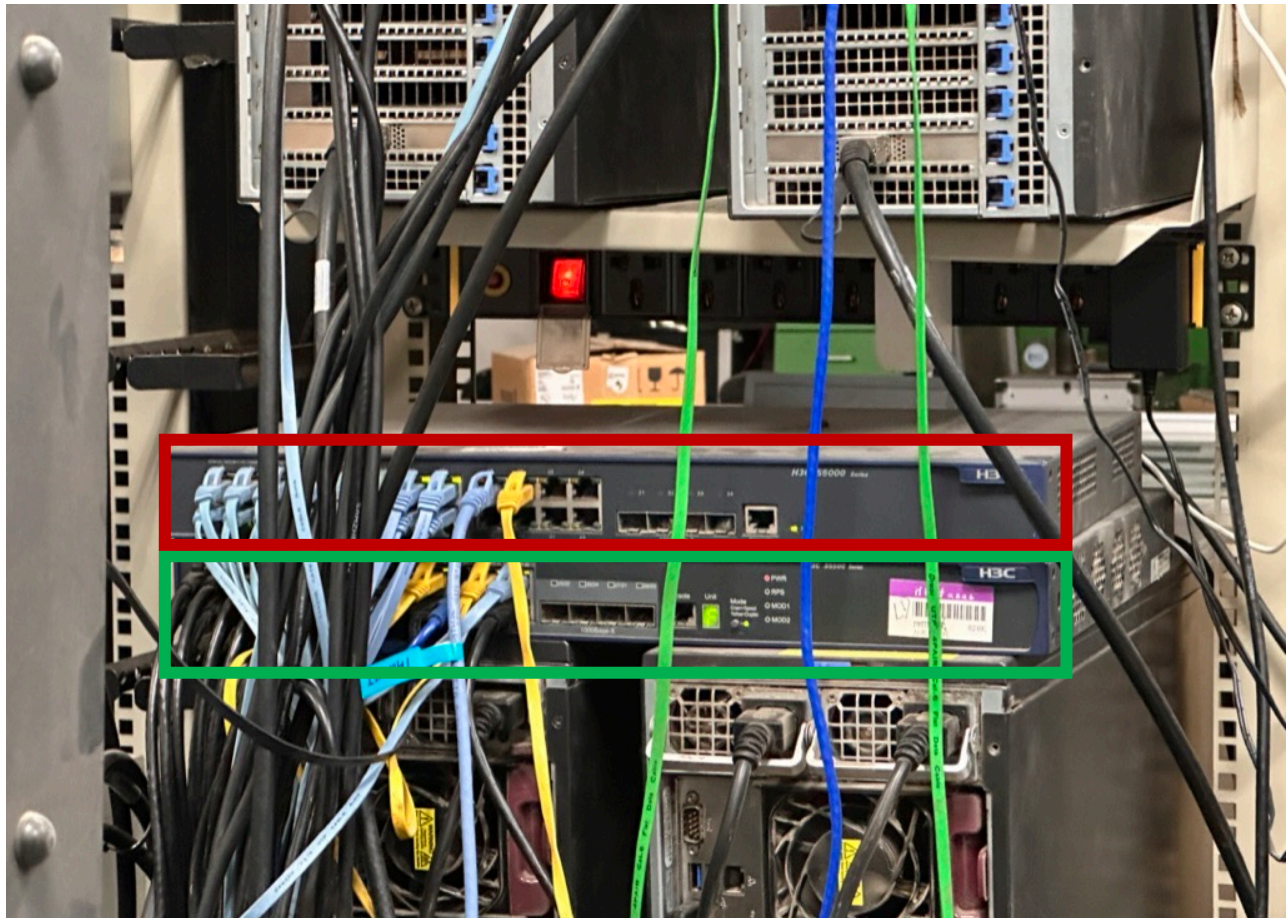
g17,g18（2U 机架式服务器）

- CPU
  - 1 * Intel Xeon Gold 6231C CPU（16c32t）。
- GPU
  - 2 * T4，显存 16 GB，计算能力 7.5。
- Host Memory
  - g17：136 GB，DDR4，2933 MHz。
  - g18：64 GB，DDR4，2933 MHz。
- NIC
  - 若干板载 1 GbE 网卡。
- *g19 已停机，将来会拆配件装到 g17 上。

前面板（上）/后面板（下）

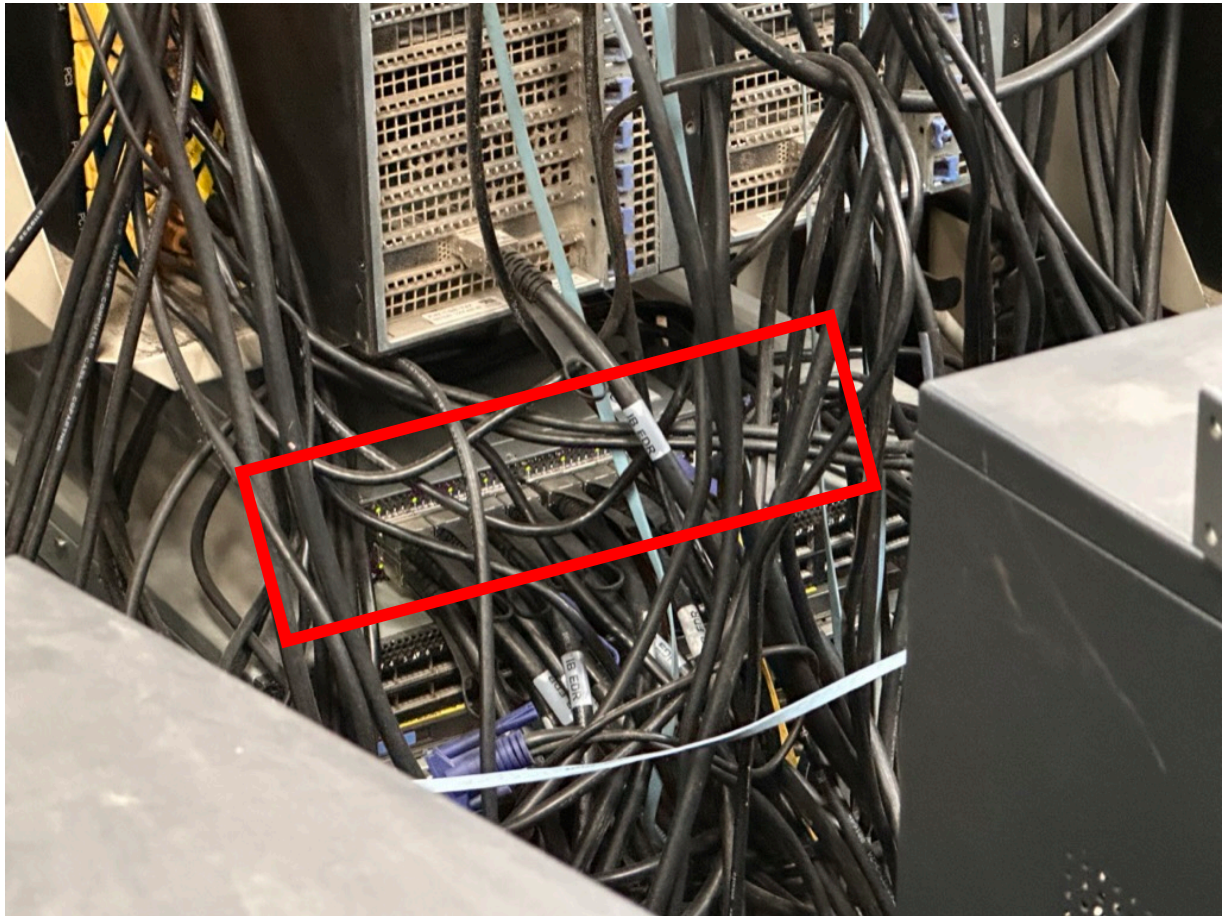
Ethernet Switch

- 24 电口 + 4 光口千兆以太网交换机
  - 上：THUNet 101.6.40.0/22
  - 下：192.168.16.0/24



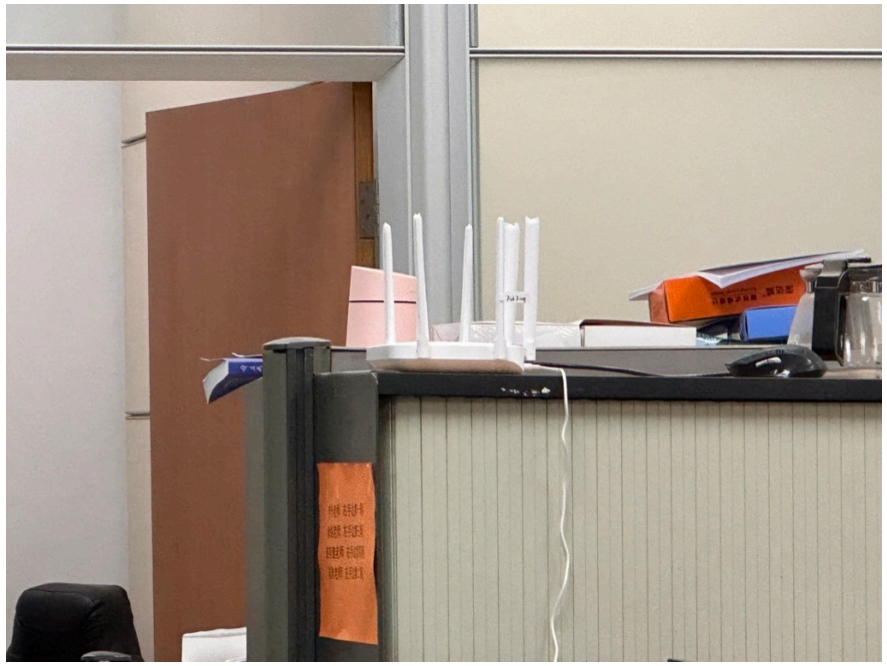
RoCE Switch

- 16 光口 100 Gbps 以太网交换机。
- 1/10/25/40/50/100 GbE 可配置。
- 当前被@飞哥物理下线。



Hard Routers

B70 和 RedMi（4-103）



RedMi（4-104）

KVM 控制台切换器

实验机网络配置

- **eno1 连接 Ethernet Switch。**
  - IP4 从 DHCP 获得：192.168.16.1XX/24。
  - IP6 从 Route Adv 生成：fd01:da8:bf:300::/64。
- **eno2 连接 Ethernet Switch THUNet。**
  - IP4 与 IP6 均从 DHCP 服务器获得。该地址全球可路由，但需要认证。认证后需要使用 `ip` 命令修改路由表。

```
sudo ip route add default via 101.6.40.1
```

- **enp2s0np0 连接 RoCE Switch。**
  - 静态 IP4：12.12.12.1XX/24。
- g18 在 Ethernet Switch 提供 NAT4/6 到 THUNet。
  - 需要一位热心群众提供校园网账号。

实验机存储配置

- g1 – g14,g17：
  - /：一块 4 TB 的 HDD 作为系统盘。
  - /home2：大小不等（ 8 TB – 12 TB ）的 HDD 阵列，持久化配置。
- g15,g16：
  - /：一块 960 GB 的 SSD 作为系统盘。
  - /home：一块 960 GB 的 SSD。
  - /data1 – /data3：三块 4 TB 的 HDD。

- ALL：
  - /gshare：所有机器共享的网络存储，共 4 TB（nfs，来自 g18）。

实验机服务配置

- SSH Server 监听 **12022** 端口。
  - 从外部登录集群必须通过跳板机。
- 每天 17:00 自动关机；每早 7:00 左右自动开机。
  - 自动关机配置 root 用户 crontab 中；自动开机由 g18 发送 wakeonlan。
- 除 g15,g16 外，所有环境配置/实验都应当发生在**容器内**。
- 除 g4,g9,g15,g16 外，通过 dnew 创建的容器会自动拉起 code-server，可在浏览器中进行开发和调试。
- 操作系统： g4,g9,g15,g16 为 Ubuntu 22.04 LTS，其余为 NixOS。

集群服务

nasp.fit (22/tcp, 36022/tcp)

- 在腾讯云注册的域名，解析托管在 Cloudflare，实验室专用。
- 顶层域名指向 g18 比威网入口。

```
nasp.fit.     1       IN      A       219.243.215.215
nasp.fit.     1       IN      AAAA    2001:da8:bf::100
```

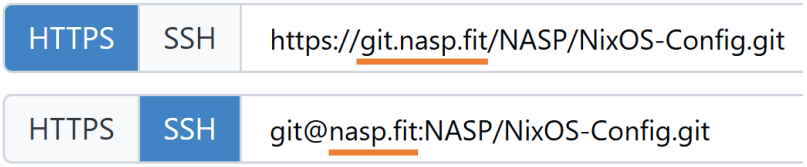
  - 跳板机是 g18 上的 docker 容器，因此登录使用该域名：

```
ssh -p 36022 ssh@nasp.fit
ssh -p 12022 -J ssh@nasp.fit:36022 <username>@g1
```

  - NASP Git 在 g18 上监听 22 端口，因此交互使用该域名：

```
git clone git@nasp.fit:NASP/registry.git
ssh git@nasp.fit
```

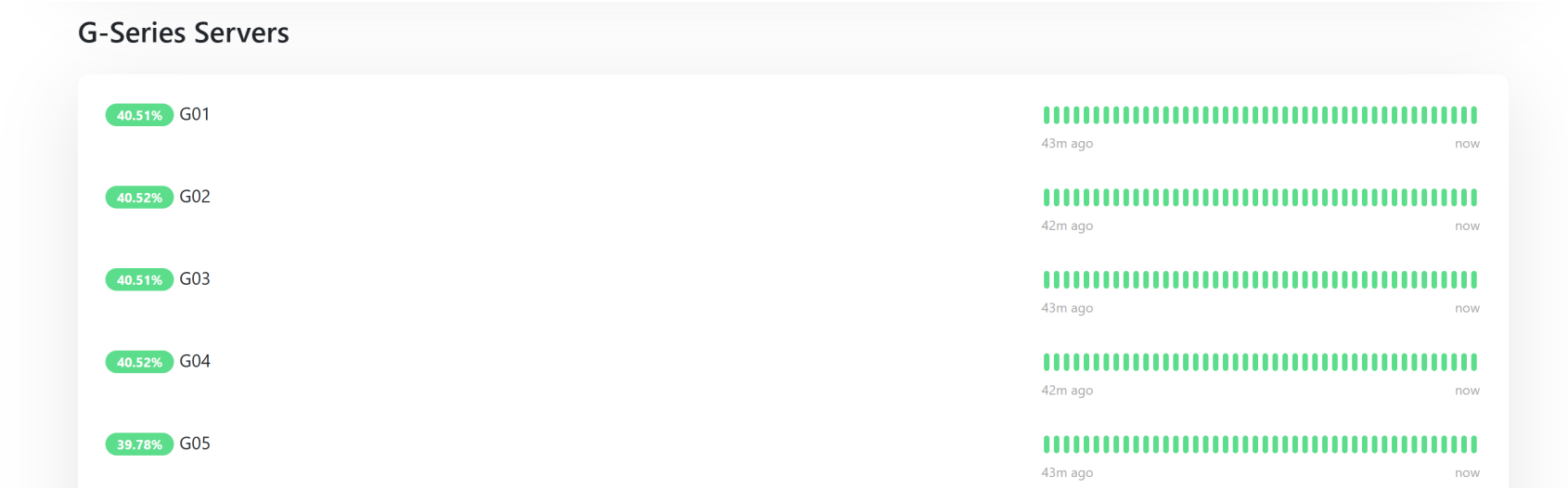
git.nasp.fit (80/tcp, 443/tcp)



- 透过 Cloudflare Tunnel 公开提供的 git 仓库托管服务。
  该地址实际动态解析到 Cloudflare 的 CDN 网络，**并不指向 g18，不能使用 ssh 等远程工具登录**，也因此无需备案。
- 集群关键组件均托管在 NASP 组织名下的仓库。**初次注册的用户需要联系管理员加入 NASP 组织**，组织外的账号会被定期清理。
  - NASP/registry：登陆公钥管理。
  - NASP/wiki：知识库。
  - NASP/NixOS-Config：操作系统配置。
  - NASP/dockerfiles：集群所涉及容器镜像的构建文件。
- 数据存储在 g18 磁盘阵列中，有冗余，能够托管重要代码。

status.nasp.fit (80/tcp, 443/tcp)

- 透过 Cloudflare Tunnel 公开提供的集群状态监控服务。
- 服务部署在 g18 上，因此：
  - 该网页访问不了：g18 网络出现问题，可能是校园网没认证。
  - 网页显示某台机器出现问题：g18 ping 不通这台机器。
  - 网页显示某个服务出现问题：该服务挂了。



proxy.nasp.fit (80/tcp, 443/tcp)

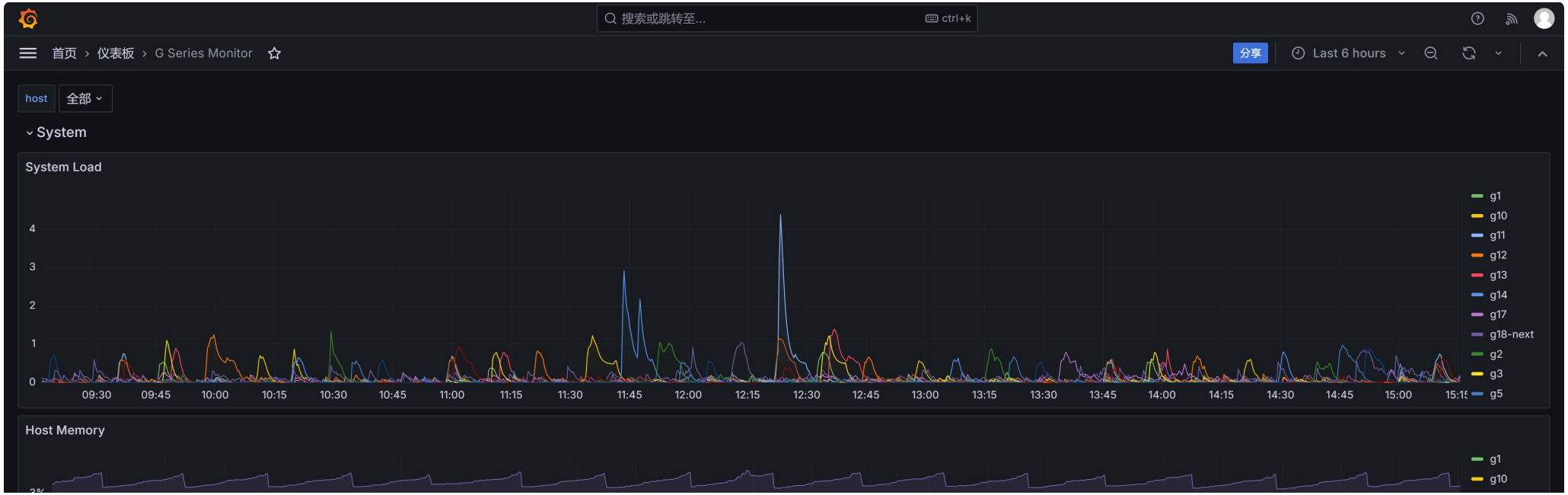
- 透过 Cloudflare Tunnel 公开提供的反向代理，即直接从外部访问内部机器。
- 目前已部署功能：
  - 反向代理到容器内 code-server：https://proxy.nasp.fit/gXX/XXX。
  - 没了。

public.nasp.fit (80/tcp, 443/tcp)

- 透过 Cloudflare Tunnel 公开提供的静态网页服务。
- 目前已部署功能：
  - /mirrors：镜像源。
  - /gshare：映射到共享网络存储文件夹 /share/public。
  - TODO：可以用来公开一些科研成果，artifacts、datasets。

grafana.nasp.fit (80/tcp, 443/tcp)

- 透过 Cloudflare Tunnel 公开提供的数据看板。
- 通过 git.nasp.fit 登录。
- **所有机器**的监控数据均已接入。



集群使用

How-to：登录实验机（1）

- 通过跳板机和密钥对访问集群。
  - 跳板机地址：nasp.fit:36022
  - 公钥由 git 仓库自动管理：https://git.nasp.fit/NASP/registry
- 添加公钥
  - 在 NASP git 上注册账户，联系管理员加入 NASP 组织。
  - 在 registry 仓库创立新的分支（或者fork一份到你的账户）。在 `authorized_keys` 目录下创建一个新文件夹，该文件夹的名字将成为你在实验机上的用户名。随后，在该文件夹下创建一个或多个含有SSH公钥的文件。
  - 提交合并请求，找任意一个其他用户批准。
  - 合并后五分钟，跳板机自动更新公钥，实验机自动创建账户并更新公钥。

How-to：登录实验机（2）

- 登录集群
  - 假设你的用户名是 naspuser。此时，持有你的私钥，你可以免密登录:

```
ssh -p 36022 ssh@nasp.fit
ssh -p 12022 -J ssh@nasp.fit:36022 naspuser@g1
```

  - SCP、SFTP等基于SSH的服务均可正常运行。
  - VS Code Remote SSH 可能有问题，建议使用 code-server（后文）。

- g15,g16 实验机登录权限/使用方法请联系@苗博。

How-to：使用实验机（1）

- 账户没有 root 权限，但可以免密 sudo 以下命令：

```
%nasp ALL = (root) NOPASSWD: /run/current-system/sw/bin/docker
%nasp ALL = (root) NOPASSWD: /run/current-system/sw/bin/whoami
%nasp ALL = (root) NOPASSWD: /run/current-system/sw/bin/nvidia-smi
%nasp ALL = (root) NOPASSWD: /run/current-system/sw/bin/shutdown
%nasp ALL = (root) NOPASSWD: /run/current-system/sw/bin/poweroff
%nasp ALL = (root) NOPASSWD: /run/current-system/sw/bin/reboot
%nasp ALL = (root) NOPASSWD: /run/current-system/sw/bin/ip
```

  - 相关配置可见 https://git.nasp.fit/NASP/NixOS-Config/src/branch/dev/hosts/modules/nasp.nix 中 `security.sudo.extraConfig` 配置项。

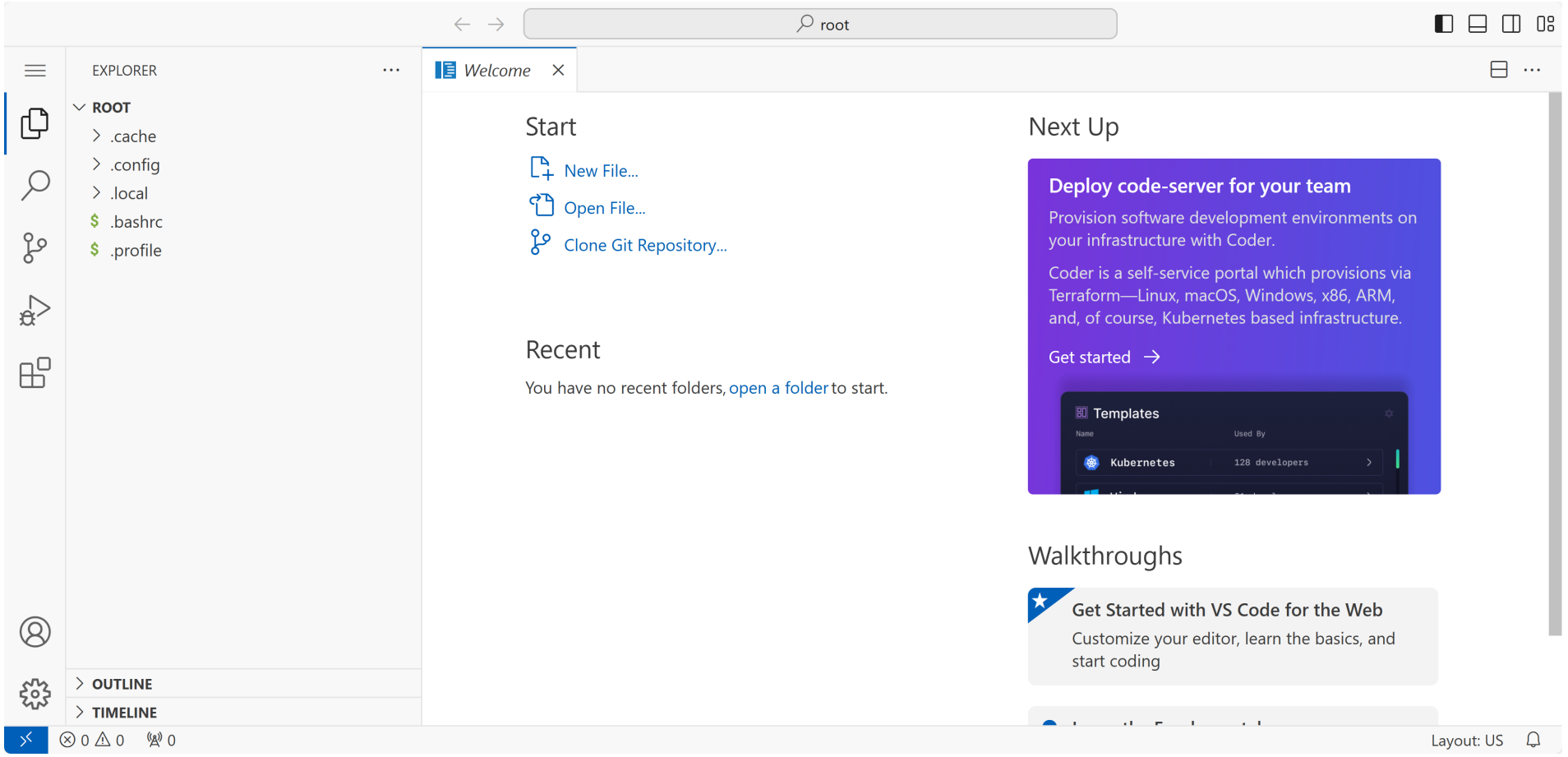
How-to：使用实验机（2）

- 进行实验，使用 `dnew` 命令创建独占容器。环境配置、开发、调试，**均在容器内进行**。
- 容器启动后，会给出容器名、 ID 与 code-server 链接。
  - 通过容器名与 ID 可以从命令行进入容器内部。
  - 通过 code-server 链接可以直接从集群外部访问网页端开发界面。
  - 如果忘记：
    - 物理机上，通过 `sudo docker ps` 命令列出正在运行的容器。
    - 容器内部，通过 `code.sh` 命令获取 code-server 配置。
- 容器基于 Ubuntu 20.04，使用上与物理机无异。
  - 容器直接使用主机网络。即，在容器内监听任意端口，等效在物理机上。
  - 容器启动脚本为 /etc/startup.sh。如需任何开机自启的命令，可置于其中。
- **重要数据请放在 /home2 或者 /gshare 目录下。**其余目录可能丢失!

How-to：使用实验机（3）

```
[dictxiong@g1:~]$ dnew
================== dnew ====================
=== contact: xd21@mails.tsinghua.edu.cn ===
Use GPU? [yN]: y
Use RDMA? [yN]: y
Will run:
=====
sudo docker run -d --net=host \
    -v /home2:/home2 -v /share:/share \
    --cap-add=SYS_NICE --cap-add=IPC_LOCK \
    --security-opt seccomp=unconfined --ulimit memlock=-1:-1 \
    --restart=unless-stopped \
    --name dictxiong_240624-154947 \
    --device nvidia.com/gpu=all \
    --device=/dev/infiniband/uverbs0 \
    -it git.nasp.fit/nasp/nasp-ubuntu /etc/startup.sh
=====
Start the container? [yN]: y
46b0f6b303d4f98a5dfb985ccdf02691fc58f85063e64dad4dc891c480edde73
Container dictxiong_240624-154947 started. You can use the following command to get in:
sudo docker exec -it dictxiong_240624-154947 bash
code-server is running
visit: https://proxy.nasp.fit/g1/eecb1ba5e11aa42ca33b4bae/
password: t8Awm7am/2nJcWUHU8IL
```

How-to：使用实验机（4）



How-to：故障排查（1）

- 如果连接不上服务器，请**使用命令行**按以下方法排查：

```
local $ ping nasp.fit                                  # 检查网络是否联通
local $ ssh -v -p 36022 ssh@nasp.fit   # 检查能否登陆跳板机
jumps $ ping g1                                        # 检查跳板机与实验机网络是否联通
local $ ssh -v -p 12022 -J ssh@nasp.fit:36022 <username>@g1   # 检查端到端
```

- 你还可以查看 https://status.nasp.fit。
- 如果查出问题，请把具体的情况及日志报告给管理员。

注：VSCode 连接性问题太多，集群服务只确保命令行能连接服务器。需要 GUI 请使用上文 code-server。

How-to：故障排查（2）

- 如果遇到这样的报错:

```
$ ssh -p 12022 -o ProxyJump=ssh@nasp.fit:36022 root@g1
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:fr7lFvhTuBLfI0cQOeTkr97KzeHFs7uAJlgg2WY0EPY.
Please contact your system administrator.
Add correct host key in /home/me/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/me/.ssh/known_hosts:72
Host key for [g1]:12022 has changed and you have requested strict checking.
Host key verification failed.
```

是目标机器的主机密钥变动导致的不匹配问题，常见于实验机重装或者跳板机重置之后。解决方法:

```
ssh-keygen -R "[g1]:12022"
```

How-to：使用 docker 命令

- 以 dnew 命令自动生成的 docker 命令为参考：

```
sudo docker run -d --net=host \
    -v /home2:/home2 -v /gshare:/gshare \
    --cap-add=SYS_NICE --cap-add=IPC_LOCK \
    --security-opt seccomp=unconfined --ulimit memlock=-1:-1 \
    --restart=unless-stopped \
    --name dictxiong_240624-154947 \
    --device nvidia.com/gpu=all \
    --device=/dev/infiniband/uverbs0 \
    -it git.nasp.fit/nasp/nasp-ubuntu /etc/startup.sh
```

- 进入 docker 内部：

```
sudo docker exec -it dictxiong_240624-154947 bash
```

How-to：使用 ssh-agent（1）

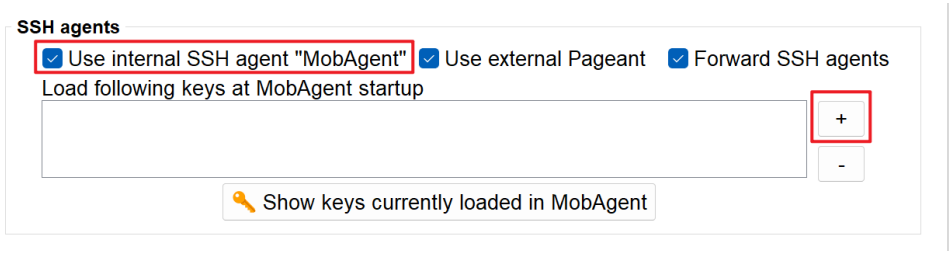
- 常规使用密钥对认证，SSH Client 会读取**当前机器上**的私钥文件。
  - 在跳板机或者实验机上，无法使用本地的私钥认证登录别的服务器，也无法使用基于 SSH 的 `git clone/pull/push`。
  - 如果粘贴私钥到服务器上，会带来**严重的安全风险**。
  - ssh-agent 能够打通机器之间的壁垒，无论跳转多少次，都可以利用本地 agent 认证。
  - ssh-agent 运行在本地，私钥没有泄露。

How-to：使用 ssh-agent（2）

- 使用方法：
  - Linux/MacOS/MinGW 等命令行环境：

```
eval `ssh-agent`
echo $SSH_AGENT_SOCK
ssh-add
ssh-add -l   # have a check anywhere & anytime
```

  - Mobaxterm/Xshell/Terminus 等 SSH 客户端：
    - 寻找软件的设置。



  - 可能需要配置 `~/.ssh/config` 或 `/etc/ssh/ssh_config`。

How-to：使用 Nix（可选）

- Nix[1] 是一种基于 Nix 表达式的包管理器，nixpkgs 仓库全世界最大的软件源。可以在 https://search.nixos.org/packages 搜索软件。
- Nix 可以在用户环境安装包而无需 root 权限。

```
nix profile list                          # 列出已安装的包
nix profile install nixpkgs#cmake         # 从 nixpkgs 安装 cmake
nix profile remove 0                      # 删除已安装的包，序号可以从 list 查到
nix shell nixpkgs#cmake                   # 临时使用而无需安装
```

  - reference: https://nix.dev/manual/nix/2.18/command-ref/new-cli/nix。
- Nix 安装的软件包存储在不可变目录中，能完美管理版本和依赖。

```
cmake -> /nix/store/q1nssraba326p2kp6627hldd2bhg254c-cmake-3.29.2/bin/cmake
```

[1] Dolstra E. The purely functional software deployment model[M]. Utrecht University, 2006.

How-to：使用 NixOS（可选）

- NixOS 是基于 Nix 构建的 Linux 发行版，以 Nix Lang 作为接口，声明式地构建操作系统。**几乎所有配置都写在一组 Nix 文件中**，可以模块化和用 git 等版本管理软件进行管理。
- 系统环境完全可复现，原子级更新和回滚。
- **不遵循 Linux 目录结构**，大部分系统文件和软件包存储在**不可变目录**下，`sudo` 也无法直接修改。一些软件可能无法正常工作。
- 集群配置托管在 https://git.nasp.fit/NASP/NixOS-Config，欢迎一同参与维护。

Q&A：如何安装软件？

- 大部分情况下请在容器内部安装软件，使用 `apt install`。
- 若真的需要在裸机上安装软件，使用 `nix profile install`。

Q&A：registry 如何实现？

- 跳板机：
  - 安装时运行：NASP/registry:/scripts/jumpserver_deploy.sh。
  - 每五分钟运行：NASP/registry:/scripts/jumpserver_cron.sh。
  - 见：NASP/dockerfiles:/nasp-jumpserver/Dockerfile。
- 实验机：
  - 每五分钟运行：NASP/registry: /scripts/testbed_cron.sh。
  - 见：NASP/NixOS-Config:/hosts/modules/nasp.nix 中 registry 相关配置。

Q&A：code-server 如何实现？

- 容器内部：
  - 见：NASP/dockerfiles:/nasp-ubuntu。
  - 构建镜像时使用官方脚本安装 code-server（Dockerfile）。
  - 启动镜像时生成随机的 socket 路径和密码并写入配置文件（startup.sh）。
  - code.sh 读取配置并输出访问提示。
- 物理机上：
  - 见：NASP/NixOS-Config:/hosts/modules/nasp.nix 中 nginx 相关配置。
  - 容器内 code-server 监听的 socket 在物理机上 /home2/run 中，通过 nginx 转发。
- g18 上：
  - 通过 nginx 正则匹配路径转发到相应的机器。

Q&A

- 希望能搬一台自己的服务器到实验室，或者希望通过有线在4-104拿到192.168.16.0/24网段的地址直接访问集群？
  - 找管理员。
- 什么时候才能晚上不断电？
  - 搬到新楼之后。在等通知。
- https://nasp.cs.tsinghua.edu.cn 是什么情况？
  - 是放在东主楼的一台主机，内容归李老师管，网络归系里管。
  - 主页已经删除，目前仅保留 /lidan.html，包含实验室近况。
- 我想参与集群管理！
  - 所有代码和配置均在NASP Git上开源。**欢迎来玩！**

谢谢大家！