

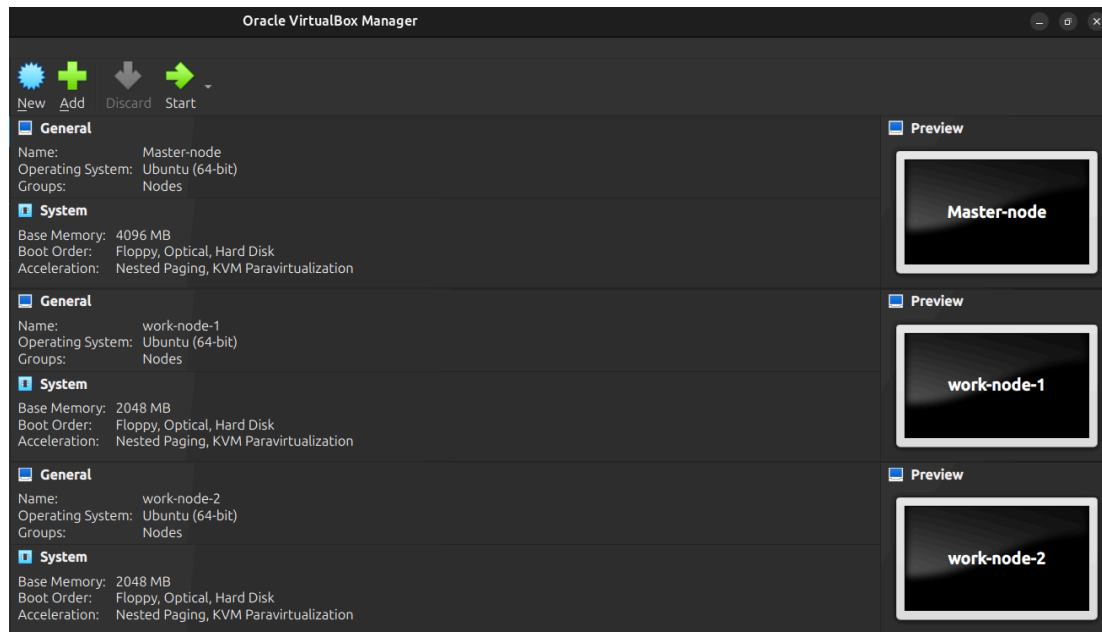
Networking Node Clusters and Virtualization

Members: Latrisha Dodson, Melissa Olivar, Timorra Rogo, Fausto Rosado, Samantha Victor

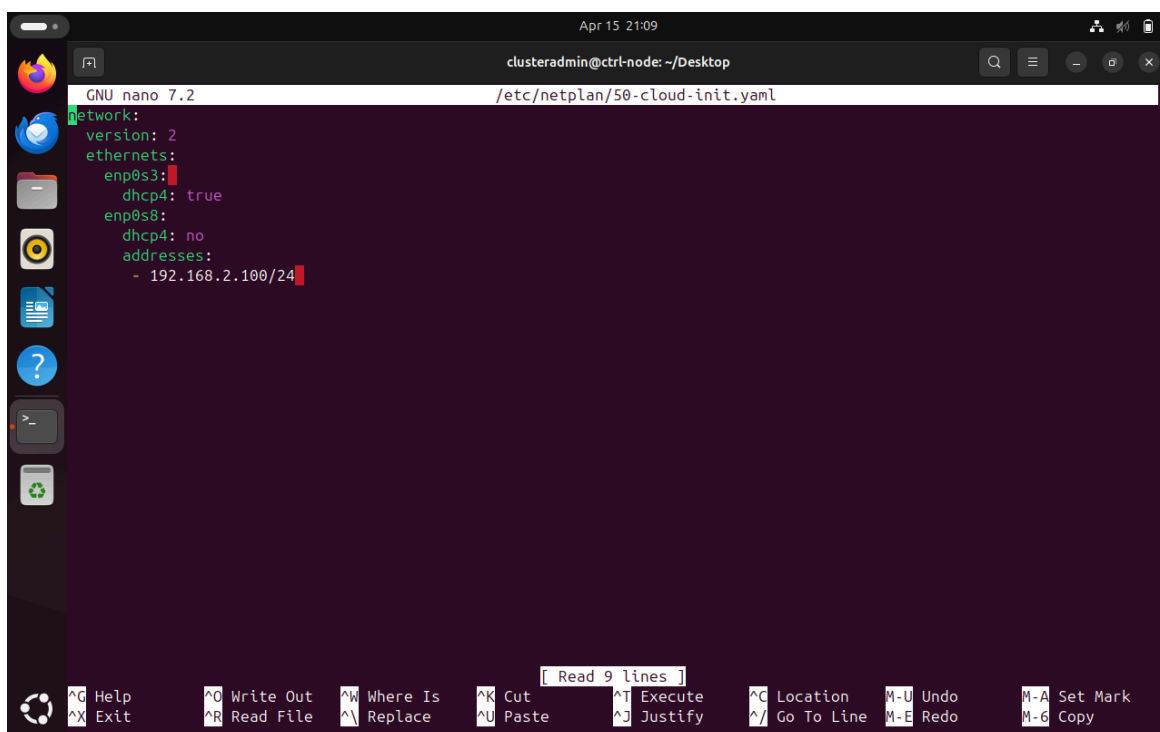
Objective: This document will guide users through the process of creating a three-node cluster from a main node using Ubuntu servers. It will compare the setup process and considerations for both the minimized and full versions of Ubuntu Server, and share lessons learned, specifically demonstrating the use of the `unminimize` command to address missing features in the minimized version.

Instructions

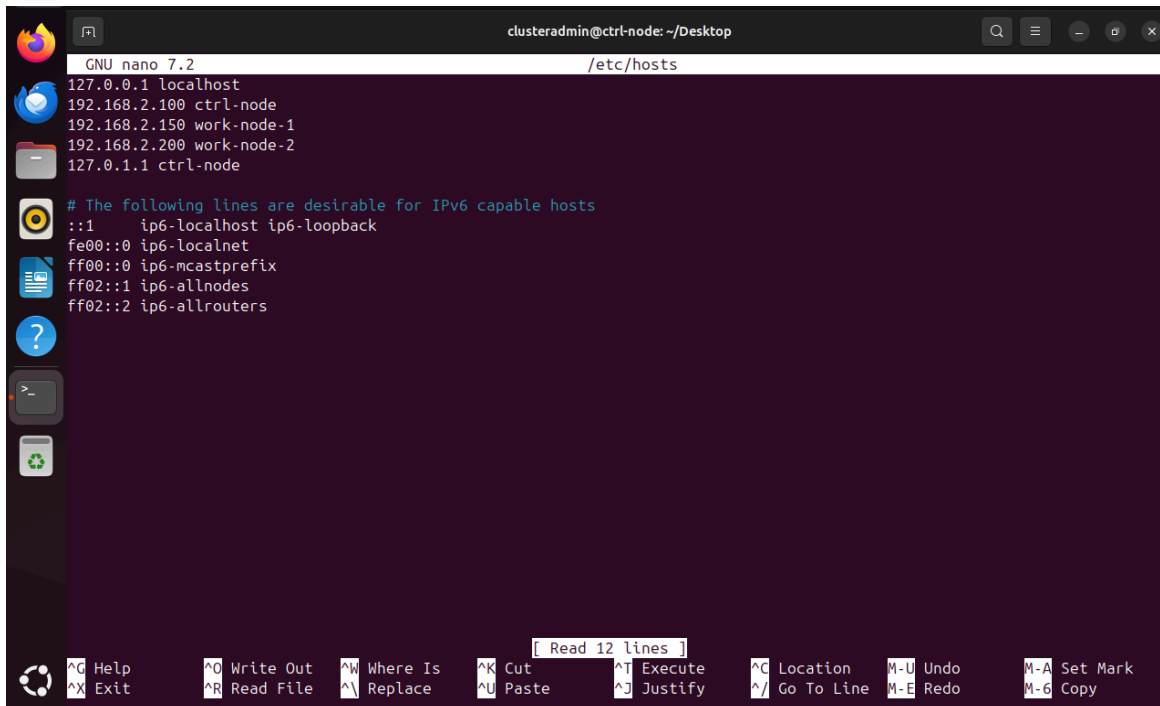
Step 1 : Spin up multiple Ubuntu VMs or physical servers. Each machine needs to have different MAC addresses in order to communicate with other.



Step 2 : Configure netplan for static internal IP for each individual node. This was done by configuring the YAML file using “sudo nano /etc/netplan/50-cloud-init.yaml”. To discover which netplan configuration file is for your devices, run “ls /etc/netplan/”.



Step 3 : Configure hostname resolution by running “sudo nano /etc/hosts”. Add the IPs and hostnames of all nodes that will be communicating.



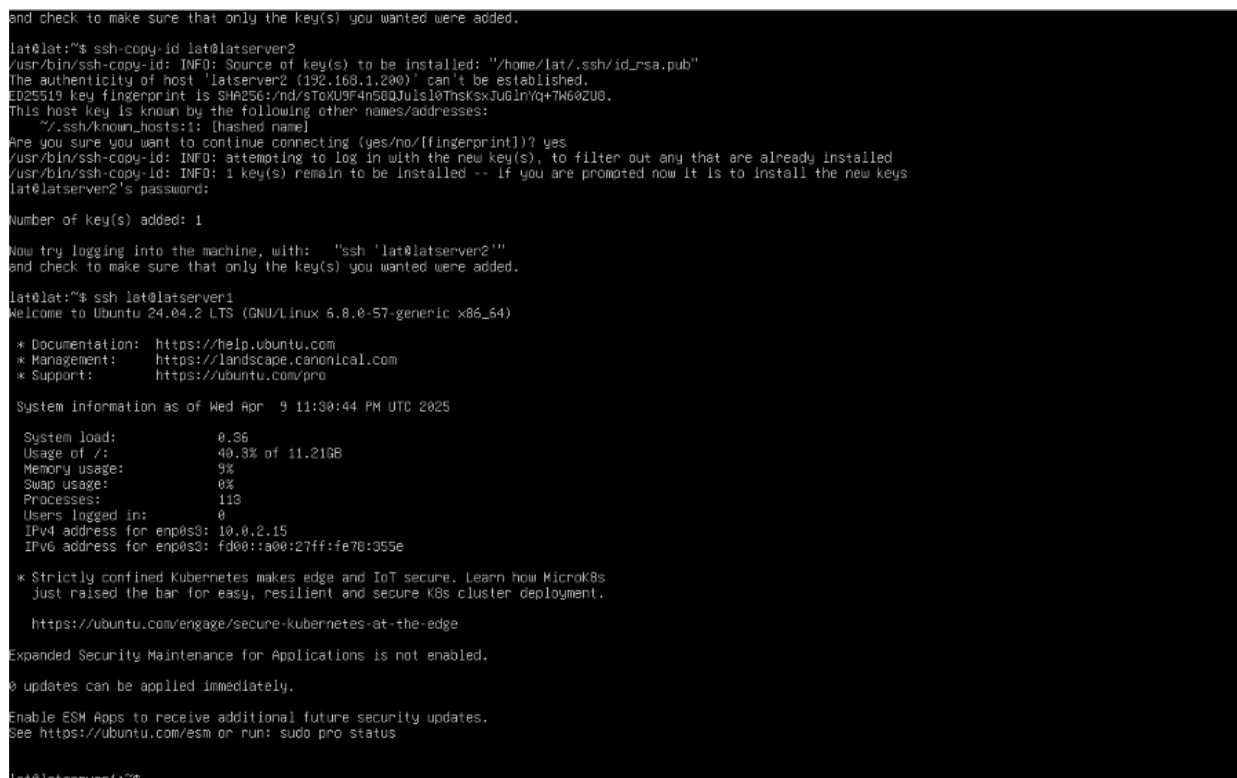
The screenshot shows a terminal window titled 'clusteradmin@ctrl-node: ~/Desktop'. The terminal is running the GNU nano 7.2 editor, editing the /etc/hosts file. The file content is as follows:

```
127.0.0.1 localhost
192.168.2.100 ctrl-node
192.168.2.150 work-node-1
192.168.2.200 work-node-2
127.0.1.1 ctrl-node

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

The terminal window includes a sidebar with application icons and a bottom status bar with various keyboard shortcuts like ^G Help, ^X Exit, ^O Write Out, etc.

Step 4 : Ensure all nodes have SSH installed. Create a SSH key from the master node and copy the key to worker nodes. The command used to create the key “ssh keygen -t rsa -b 4096”, and to copy the key to other nodes is “ssh-copy-id wn1@work-node-1”.



The screenshot shows a terminal window with the following output:

```
and check to make sure that only the key(s) you wanted were added.

lat@lat:~$ ssh-copy-id lat@latserver2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/lat/.ssh/id_rsa.pub"
The authenticity of host 'latserver2 (192.168.1.200)' can't be established.
ED25519 key fingerprint is SHA256:/nd/stoXU9F4n56QJuls10TnsksxJugIn/q+7W60ZU0.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- If you are prompted now it is to install the new keys
lat@latserver2's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'lat@latserver2'"
and check to make sure that only the key(s) you wanted were added.

lat@lat:~$ ssh lat@latserver1
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

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

System information as of Wed Apr  9 11:30:44 PM UTC 2025

System load:          0.36
Usage of /:            40.8% of 11.21GB
Memory usage:         9%
Swap usage:           0%
Processes:            113
Users logged in:      0
IPv4 address for enp0s3: 10.0.2.15
IPv6 address for enp0s3: fd00:a00:27ff:fe70:355e

 * 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

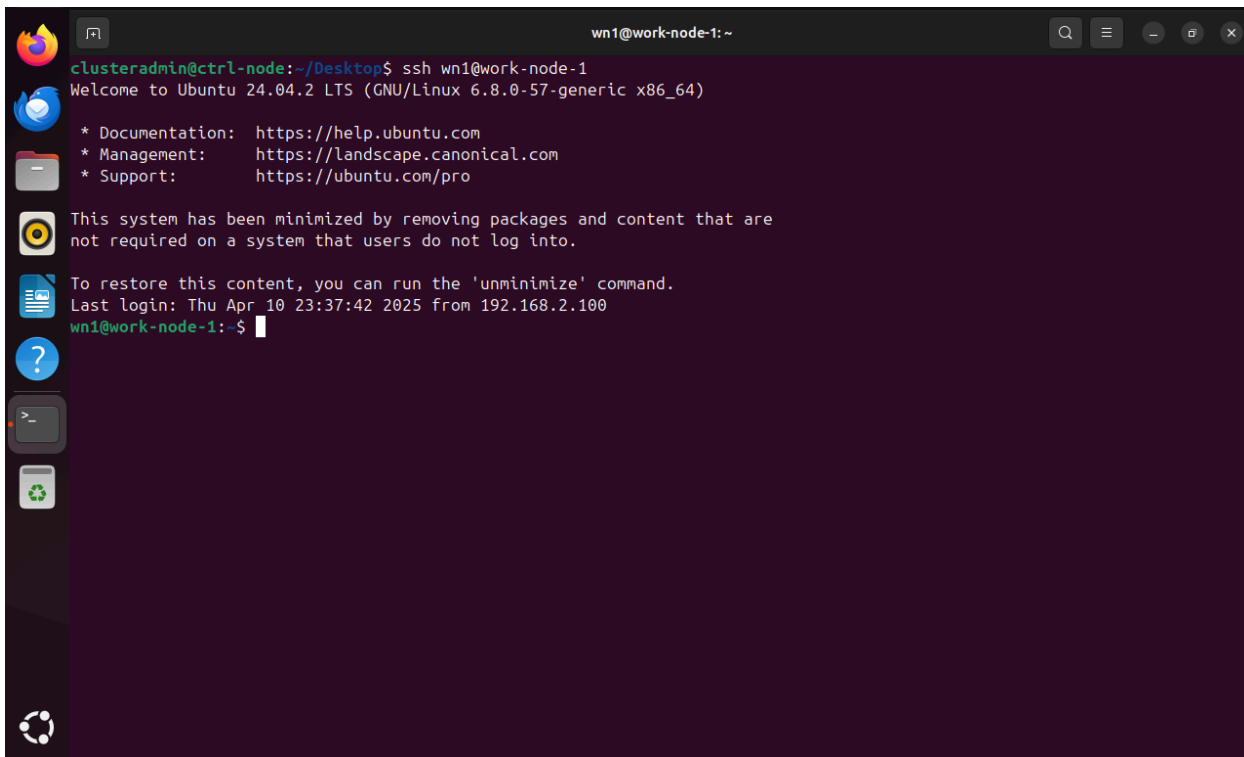
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

lat@latserver1:~$
```

Step 5: Ensure SSH access between nodes by running “ssh wn1@work-node-1”



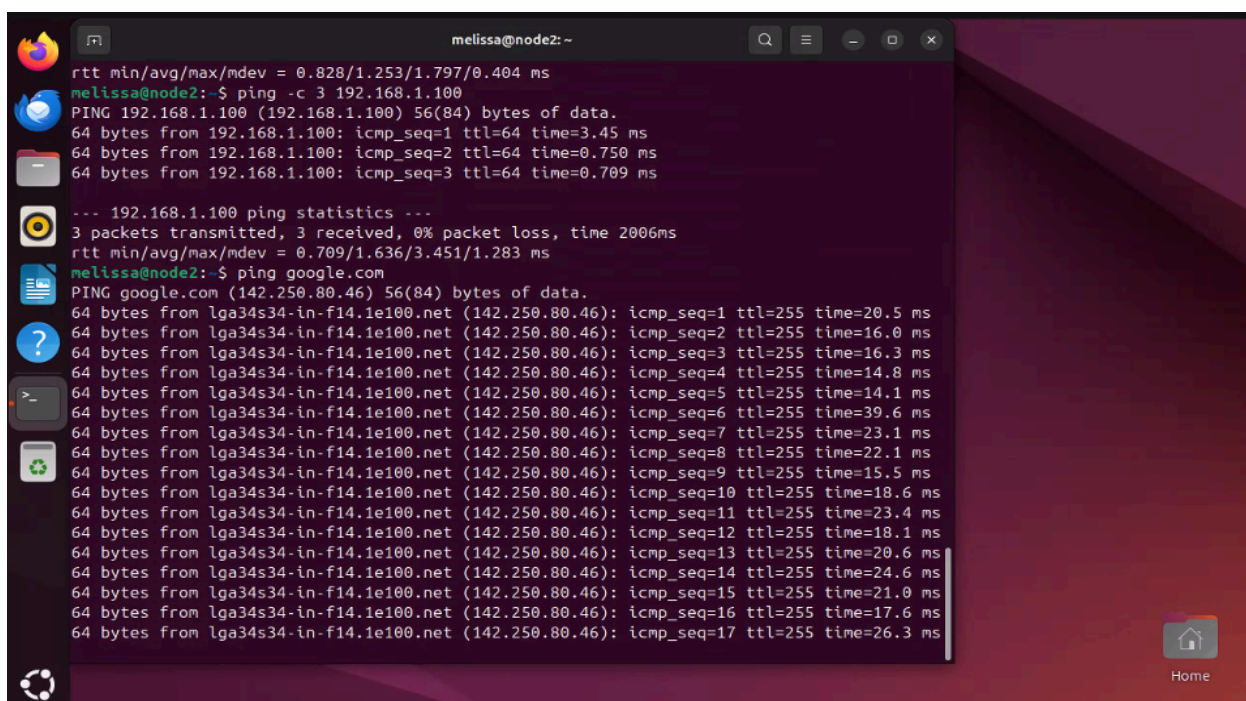
```
clusteradmin@ctrl-node:~/Desktop$ ssh wn1@work-node-1
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Apr 10 23:37:42 2025 from 192.168.2.100
wn1@work-node-1:~$
```

Step 6: Verify internal network connectivity and internet access through ping. In this example, the master node was SSH into a worker node and was then able to ping itself (verifying that they are on the same network) and the internet.



```
melissa@node2:~$ rtt min/avg/max/mdev = 0.828/1.253/1.797/0.404 ms
melissa@node2:~$ ping -c 3 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data:
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=3.45 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=0.750 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=0.709 ms

--- 192.168.1.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 0.709/1.636/3.451/1.283 ms
melissa@node2:~$ ping google.com
PING google.com (142.250.80.46) 56(84) bytes of data:
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=1 ttl=255 time=20.5 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=2 ttl=255 time=16.0 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=3 ttl=255 time=16.3 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=4 ttl=255 time=14.8 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=5 ttl=255 time=14.1 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=6 ttl=255 time=39.6 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=7 ttl=255 time=23.1 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=8 ttl=255 time=22.1 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=9 ttl=255 time=15.5 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=10 ttl=255 time=18.6 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=11 ttl=255 time=23.4 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=12 ttl=255 time=18.1 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=13 ttl=255 time=20.6 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=14 ttl=255 time=24.6 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=15 ttl=255 time=21.0 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=16 ttl=255 time=17.6 ms
64 bytes from lga34s34-in-f14.1e100.net (142.250.80.46): icmp_seq=17 ttl=255 time=26.3 ms
```

Challenges Encountered

We ran into several missing tools and had to install them one by one on the minimized image. Later on, we discovered the `unminimize` command, which converts a minimized Ubuntu system into a full-featured one.

If we had known earlier, we could have saved a lot of time by running “`sudo unminimize`”.

```
Ubuntu 24.04.2 LTS work-node-1 tty1

work-node-1 login: wn1
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
wn1@work-node-1:~$ _
```