Linux VM Provisioning, Identity Integration, and Automation Readiness

8:06 PM

This project documents the complete lifecycle of building and validating a Linux virtual machine in an enterprise-style environment, from infrastructure provisioning to automation readiness.

The virtual machine is deployed on vSphere and configured with proper networking, routing, and hostname alignment to meet organizational standards. Network interfaces, IP configuration, gateways, and DNS resolution are verified to ensure stable connectivity and interoperability with centralized services.

The system is enrolled into a FreeIPA domain to enable centralized identity and access management. Core services including SSSD, Kerberos, LDAP, and SSH are configured and validated, allowing domain users to authenticate successfully and receive appropriate group memberships and sudo privileges. Local user management is also implemented where required, following least-privilege and security best practices.

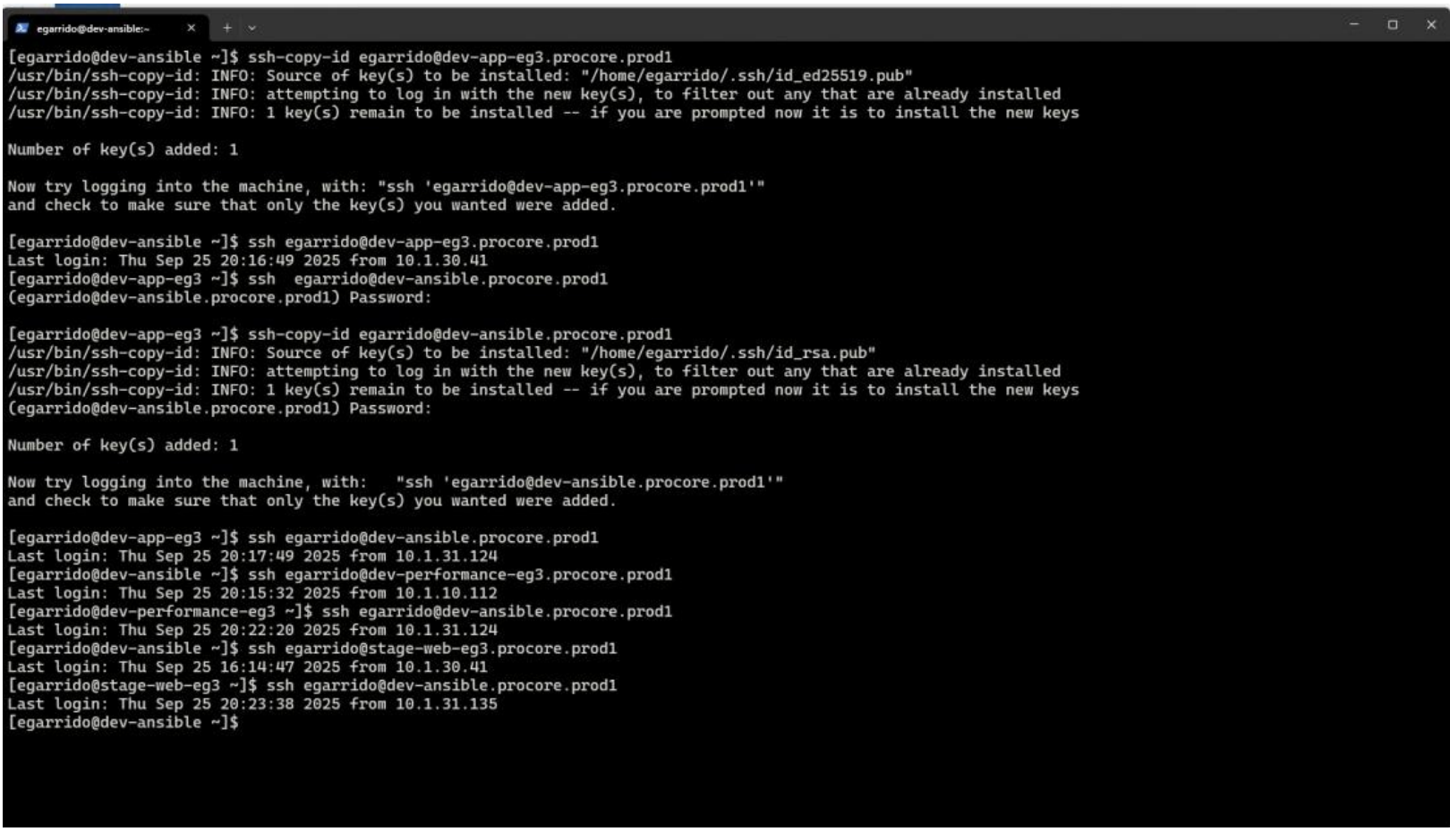
Secure access is established using SSH key-based authentication, leveraging modern cryptographic keys. SSH daemon settings are reviewed, stale host keys are cleaned up, trust relationships are rebuilt, and passwordless access between hosts is verified. Static hostname mappings are applied when necessary to support controlled environments and troubleshooting scenarios.

An Ansible inventory is created and refined to represent multiple environments and roles. Initial connectivity issues are identified and resolved, and successful communication is confirmed using the ansible -m ping module. This validates that the environment is fully prepared for repeatable, reliable automation workflows.

Finally, the system is documented in an asset management platform, capturing ownership, configuration details, and lifecycle status to ensure traceability, accountability, and operational visibility.

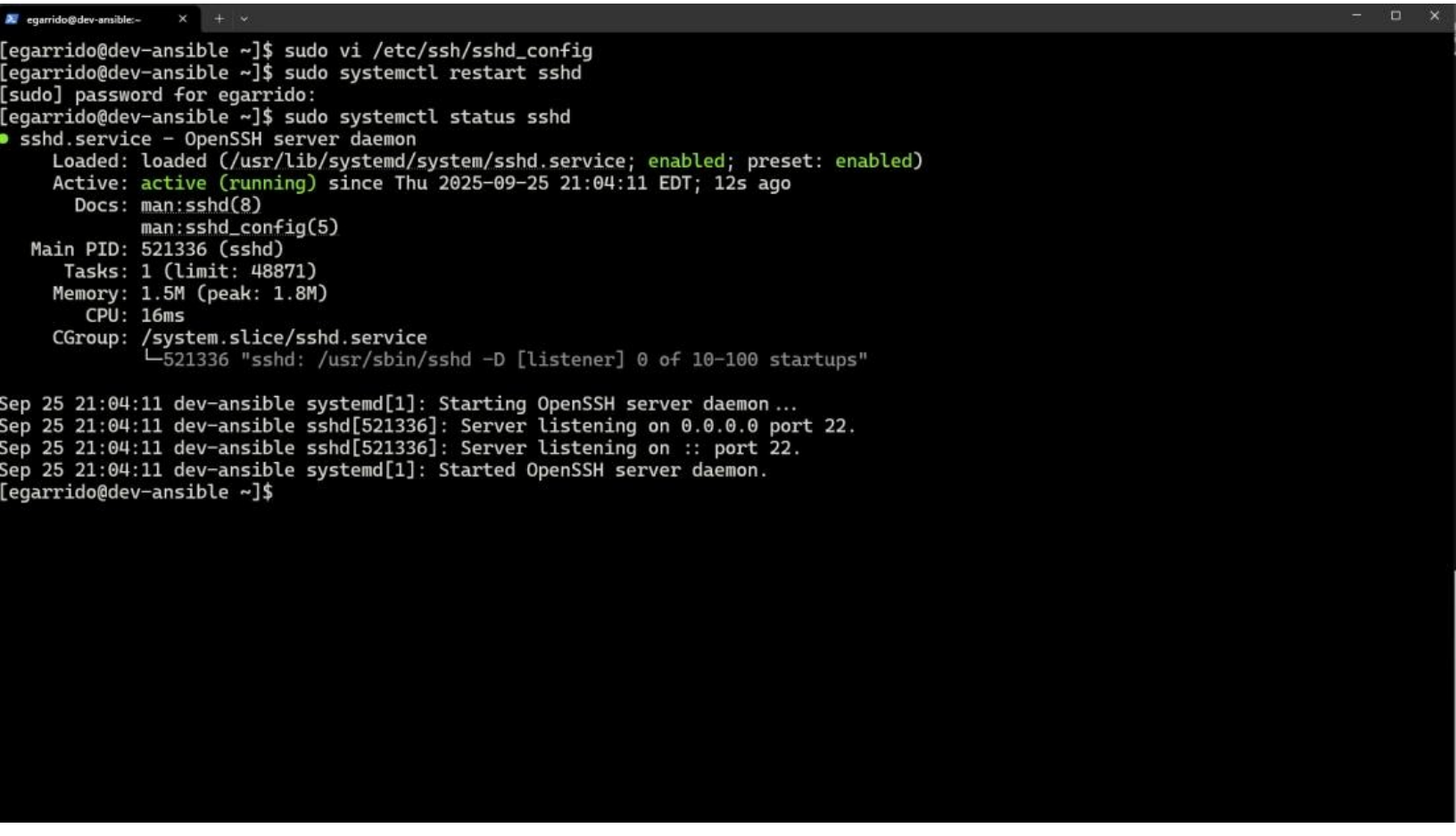
Security & Sanitization Notice

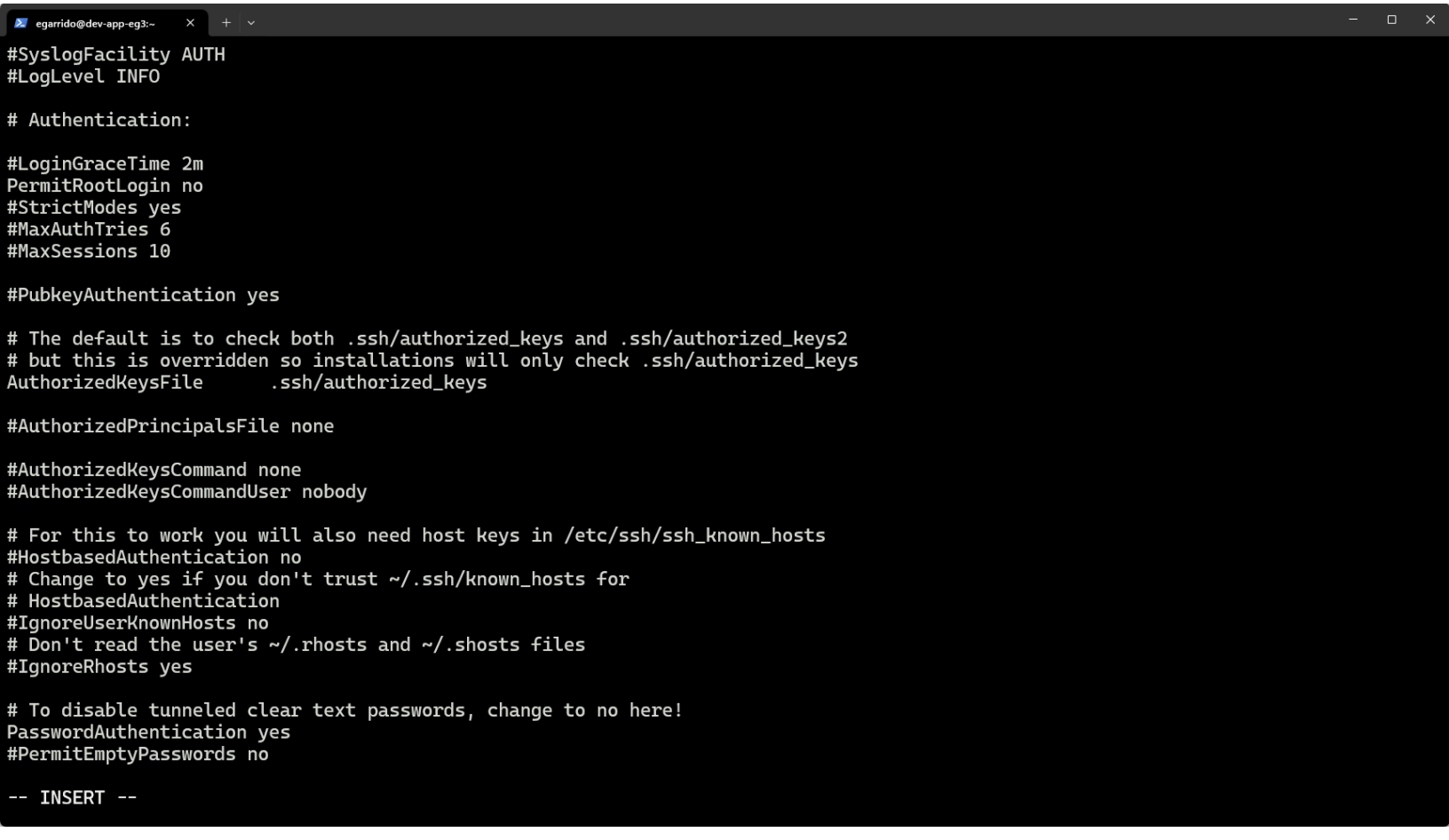
All IP addresses, hostnames, usernames, domain names, serial numbers, MAC addresses, timestamps, and environment-specific identifiers shown in screenshots, command output, and configuration files have been sanitized or obfuscated. No production credentials, sensitive infrastructure data, or real customer information are exposed in this repository.

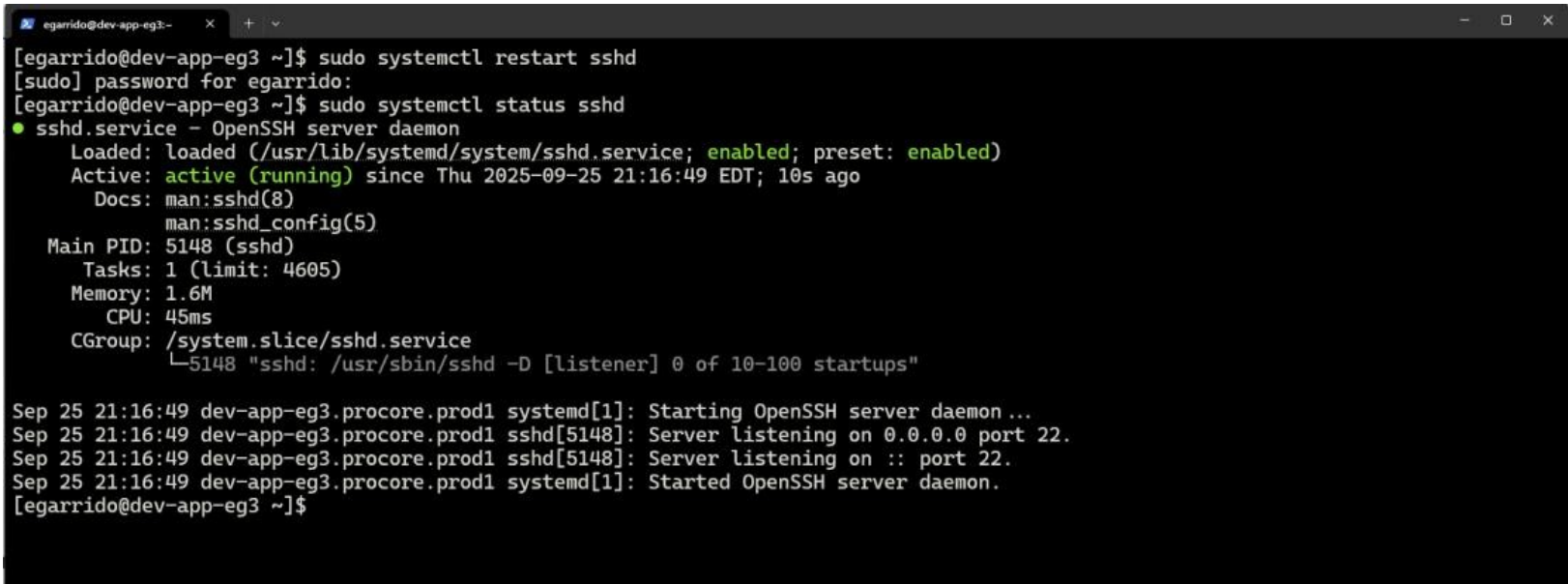

egarrido@dev-ansible:~
[egarrido@dev-ansible ~]$ ssh-copy-id egarrido@dev-app-eg3.procore.prod1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/egarrido/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/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
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'egarrido@dev-app-eg3.procore.prod1'"
and check to make sure that only the key(s) you wanted were added.
[egarrido@dev-ansible ~]$ ssh egarrido@dev-app-eg3.procore.prod1
Last login: Thu Sep 25 20:16:49 2025 from 10.1.30.41
[egarrido@dev-app-eg3 ~]$ ssh egarrido@dev-ansible.procore.prod1
(egarrido@dev-ansible.procore.prod1) Password:
[egarrido@dev-app-eg3 ~]$ ssh-copy-id egarrido@dev-ansible.procore.prod1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/egarrido/.ssh/id_rsa.pub"
/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
(egarrido@dev-ansible.procore.prod1) Password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'egarrido@dev-ansible.procore.prod1'"
and check to make sure that only the key(s) you wanted were added.
[egarrido@dev-app-eg3 ~]$ ssh egarrido@dev-ansible.procore.prod1
Last login: Thu Sep 25 20:17:49 2025 from 10.1.31.124
[egarrido@dev-ansible ~]$ ssh egarrido@dev-performance-eg3.procore.prod1
Last login: Thu Sep 25 20:15:32 2025 from 10.1.10.112
[egarrido@dev-performance-eg3 ~]$ ssh egarrido@dev-ansible.procore.prod1
Last login: Thu Sep 25 20:22:20 2025 from 10.1.31.124
[egarrido@dev-ansible ~]$ ssh egarrido@stage-web-eg3.procore.prod1
Last login: Thu Sep 25 16:14:47 2025 from 10.1.30.41
[egarrido@stage-web-eg3 ~]$ ssh egarrido@dev-ansible.procore.prod1
Last login: Thu Sep 25 20:23:38 2025 from 10.1.31.135
[egarrido@dev-ansible ~]$

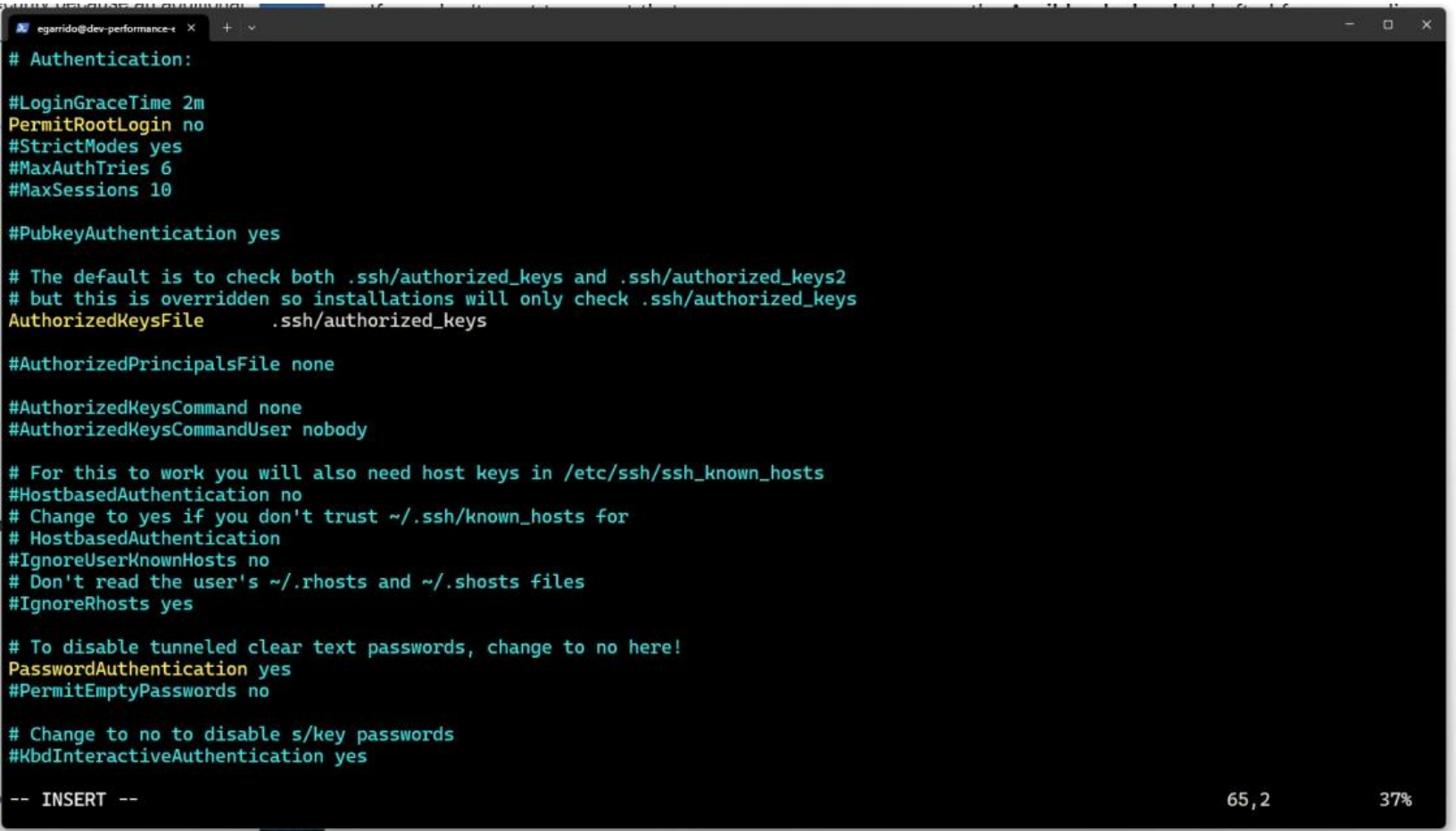
egarrido@dev-ansible:~

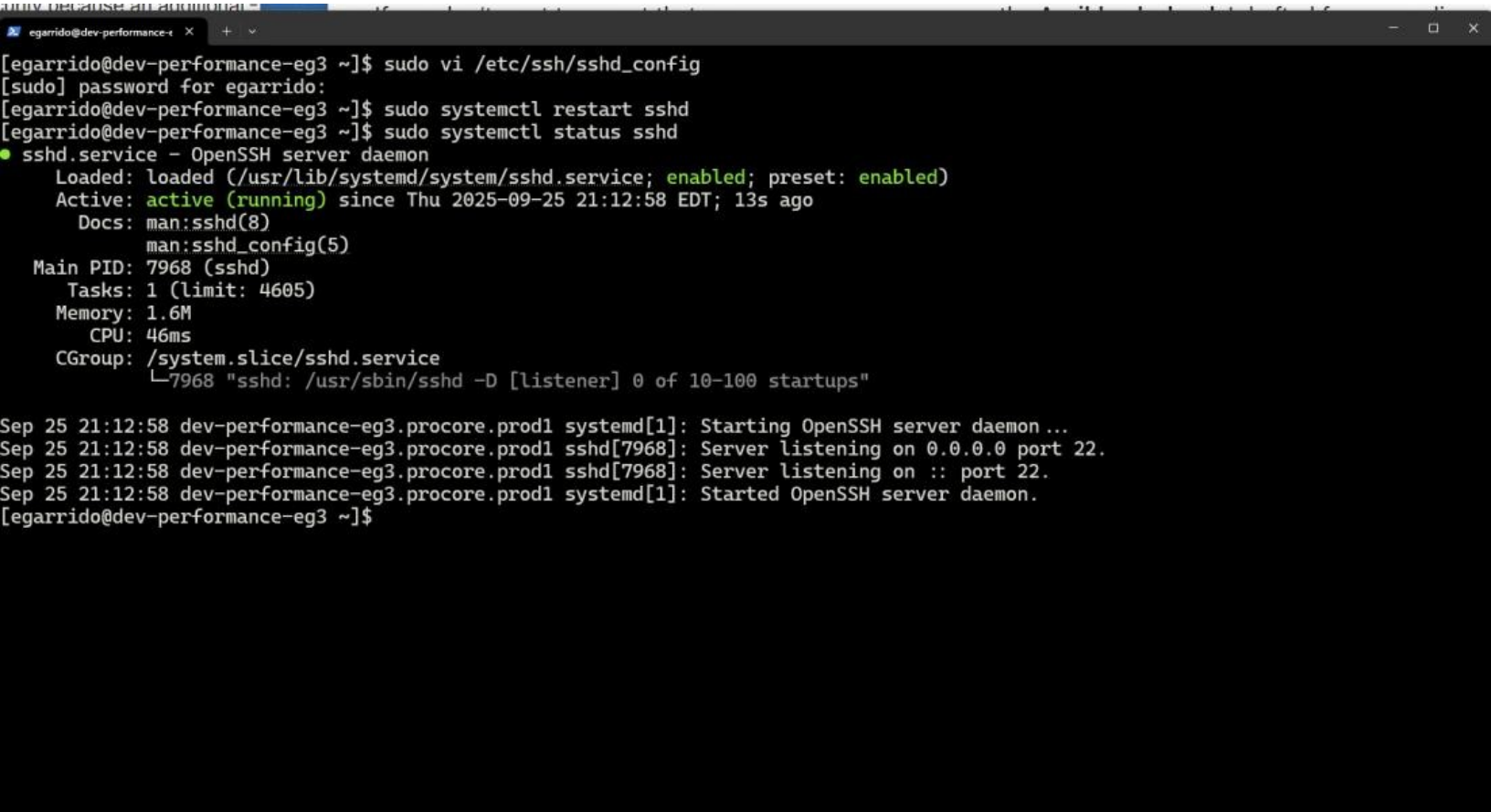
```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

-- INSERT --                                                    67,8          38%
```

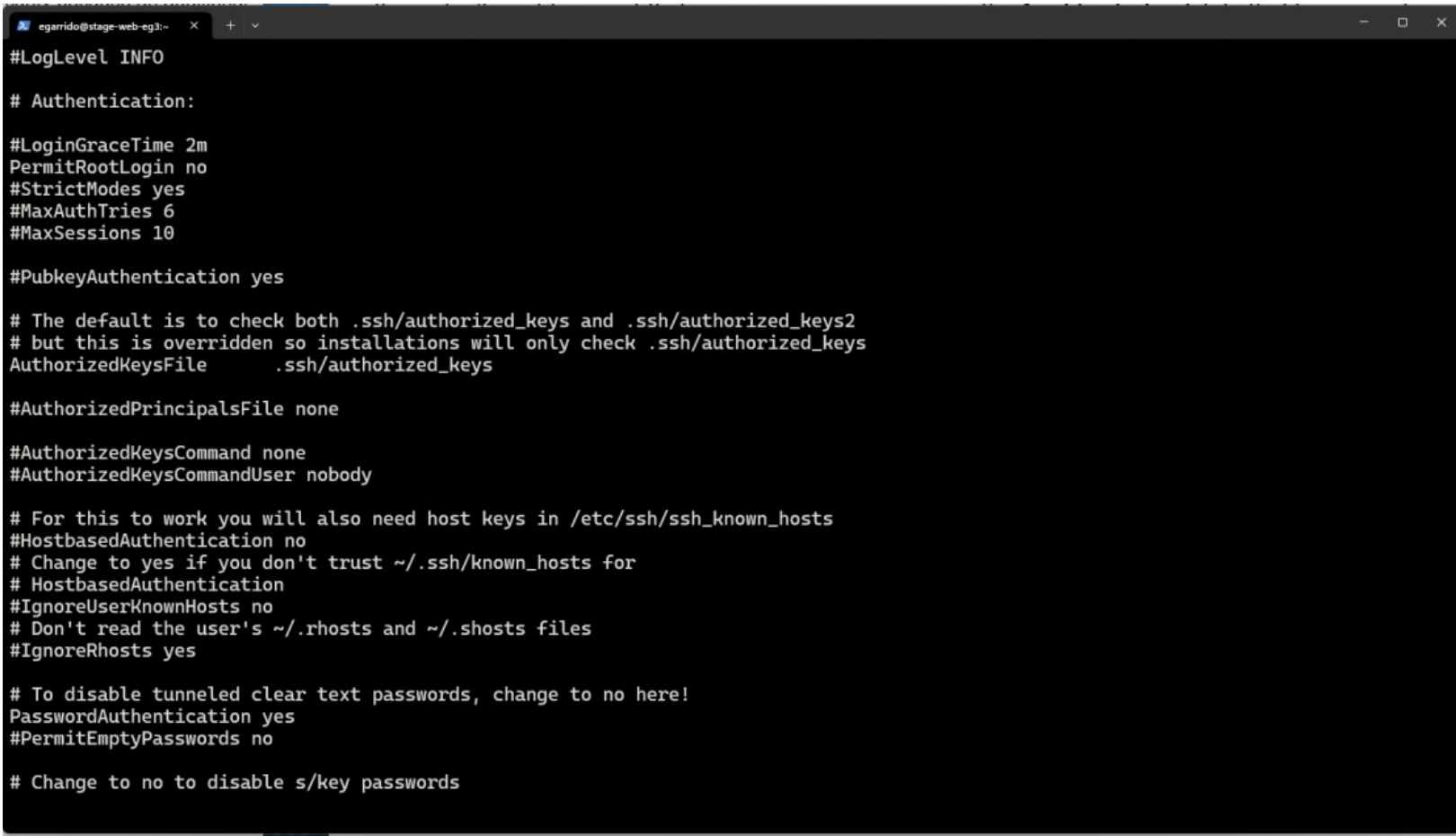

egarrido@dev-ansible:~
[egarrido@dev-ansible ~]$ sudo vi /etc/ssh/sshd_config
[egarrido@dev-ansible ~]$ sudo systemctl restart sshd
[sudo] password for egarrido:
[egarrido@dev-ansible ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
Active: active (running) since Thu 2025-09-25 21:04:11 EDT; 12s ago
Docs: man:sshd(8)
man:sshd_config(5)
Main PID: 521336 (sshd)
Tasks: 1 (limit: 48871)
Memory: 1.5M (peak: 1.8M)
CPU: 16ms
CGroup: /system.slice/sshd.service
└─521336 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
Sep 25 21:04:11 dev-ansible systemd[1]: Starting OpenSSH server daemon...
Sep 25 21:04:11 dev-ansible sshd[521336]: Server listening on 0.0.0.0 port 22.
Sep 25 21:04:11 dev-ansible sshd[521336]: Server listening on :: port 22.
Sep 25 21:04:11 dev-ansible systemd[1]: Started OpenSSH server daemon.
[egarrido@dev-ansible ~]$

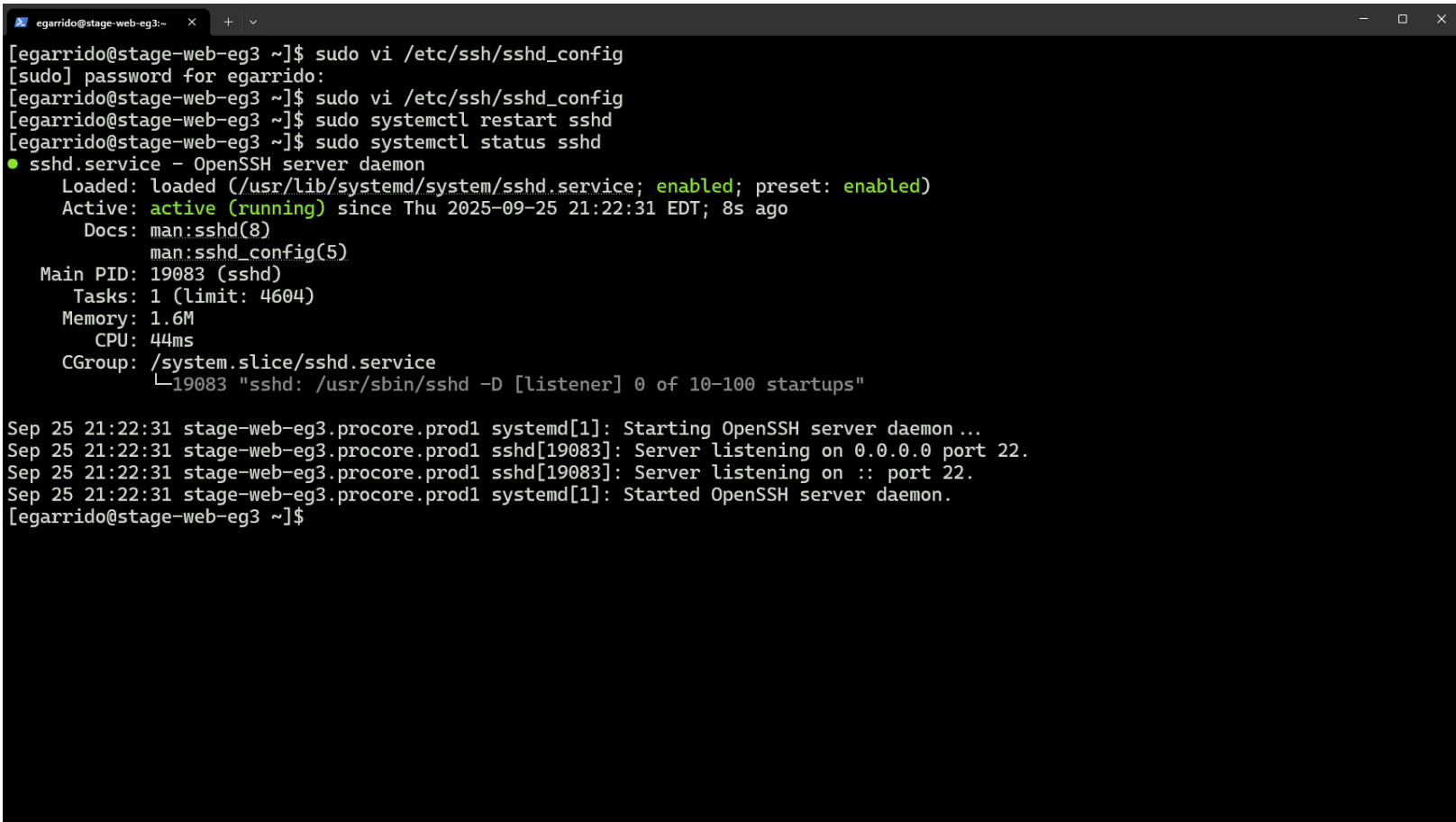

egarrido@dev-app-eg3:~
#SyslogFacility AUTH
#LogLevel INFO
Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes
The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys
#AuthorizedPrincipalsFile none
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
Change to yes if you don't trust ~/.ssh/known_hosts for
HostbasedAuthentication
#IgnoreUserKnownHosts no
Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
-- INSERT --


egarrido@dev-app-eg3:~
[egarrido@dev-app-eg3 ~]$ sudo systemctl restart sshd
[sudo] password for egarrido:
[egarrido@dev-app-eg3 ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
Active: active (running) since Thu 2025-09-25 21:16:49 EDT; 10s ago
Docs: man:sshd(8)
man:sshd_config(5)
Main PID: 5148 (sshd)
Tasks: 1 (limit: 4605)
Memory: 1.6M
CPU: 45ms
CGroup: /system.slice/sshd.service
└─5148 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
Sep 25 21:16:49 dev-app-eg3.procore.prod1 systemd[1]: Starting OpenSSH server daemon...
Sep 25 21:16:49 dev-app-eg3.procore.prod1 sshd[5148]: Server listening on 0.0.0.0 port 22.
Sep 25 21:16:49 dev-app-eg3.procore.prod1 sshd[5148]: Server listening on :: port 22.
Sep 25 21:16:49 dev-app-eg3.procore.prod1 systemd[1]: Started OpenSSH server daemon.
[egarrido@dev-app-eg3 ~]$


egarrido@dev-performance-c
Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes
The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys
#AuthorizedPrincipalsFile none
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
Change to yes if you don't trust ~/.ssh/known_hosts for
HostbasedAuthentication
#IgnoreUserKnownHosts no
Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes
-- INSERT --
65,2
37%


egarrido@dev-performance-e
[egarrido@dev-performance-eg3 ~]$ sudo vi /etc/ssh/sshd_config
[sudo] password for egarrido:
[egarrido@dev-performance-eg3 ~]$ sudo systemctl restart sshd
[egarrido@dev-performance-eg3 ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
Active: active (running) since Thu 2025-09-25 21:12:58 EDT; 13s ago
Docs: man:sshd(8)
man:sshd_config(5)
Main PID: 7968 (sshd)
Tasks: 1 (limit: 4605)
Memory: 1.6M
CPU: 46ms
CGroup: /system.slice/sshd.service
└─7968 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
Sep 25 21:12:58 dev-performance-eg3.procore.prod1 systemd[1]: Starting OpenSSH server daemon...
Sep 25 21:12:58 dev-performance-eg3.procore.prod1 sshd[7968]: Server listening on 0.0.0.0 port 22.
Sep 25 21:12:58 dev-performance-eg3.procore.prod1 sshd[7968]: Server listening on :: port 22.
Sep 25 21:12:58 dev-performance-eg3.procore.prod1 systemd[1]: Started OpenSSH server daemon.
[egarrido@dev-performance-eg3 ~]$


egarrido@stage-web-eg3:~
#LogLevel INFO
Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes
The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys
#AuthorizedPrincipalsFile none
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
Change to yes if you don't trust ~/.ssh/known_hosts for
HostbasedAuthentication
#IgnoreUserKnownHosts no
Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
Change to no to disable s/key passwords


egarrido@stage-web-eg3:~
[egarrido@stage-web-eg3 ~]$ sudo vi /etc/ssh/sshd_config
[sudo] password for egarrido:
[egarrido@stage-web-eg3 ~]$ sudo vi /etc/ssh/sshd_config
[egarrido@stage-web-eg3 ~]$ sudo systemctl restart sshd
[egarrido@stage-web-eg3 ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
Active: active (running) since Thu 2025-09-25 21:22:31 EDT; 8s ago
Docs: man:sshd(8)
man:sshd_config(5)
Main PID: 19083 (sshd)
Tasks: 1 (limit: 4604)
Memory: 1.6M
CPU: 44ms
CGroup: /system.slice/sshd.service
└─19083 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
Sep 25 21:22:31 stage-web-eg3.procore.prod1 systemd[1]: Starting OpenSSH server daemon...
Sep 25 21:22:31 stage-web-eg3.procore.prod1 sshd[19083]: Server listening on 0.0.0.0 port 22.
Sep 25 21:22:31 stage-web-eg3.procore.prod1 sshd[19083]: Server listening on :: port 22.
Sep 25 21:22:31 stage-web-eg3.procore.prod1 systemd[1]: Started OpenSSH server daemon.
[egarrido@stage-web-eg3 ~]$

Summary

This project showcases the end-to-end provisioning and validation of a Linux virtual machine in an enterprise environment. It covers network and hostname configuration, FreeIPA-based centralized identity management, secure SSH key-based access, and Ansible connectivity testing to ensure automation readiness. The system is deployed on vSphere, documented in an asset management platform, and verified for reliable operation. All IP addresses, hostnames, usernames, serial numbers, and environment-specific details shown are sanitized for security purposes.