The work includes installing and configuring system services, applying firewall rules to allow required network traffic, and enforcing security best practices through controlled user access and permissions. Service configurations are reviewed and adjusted to disable unnecessary or insecure defaults while enabling required functionality for authorized users only.

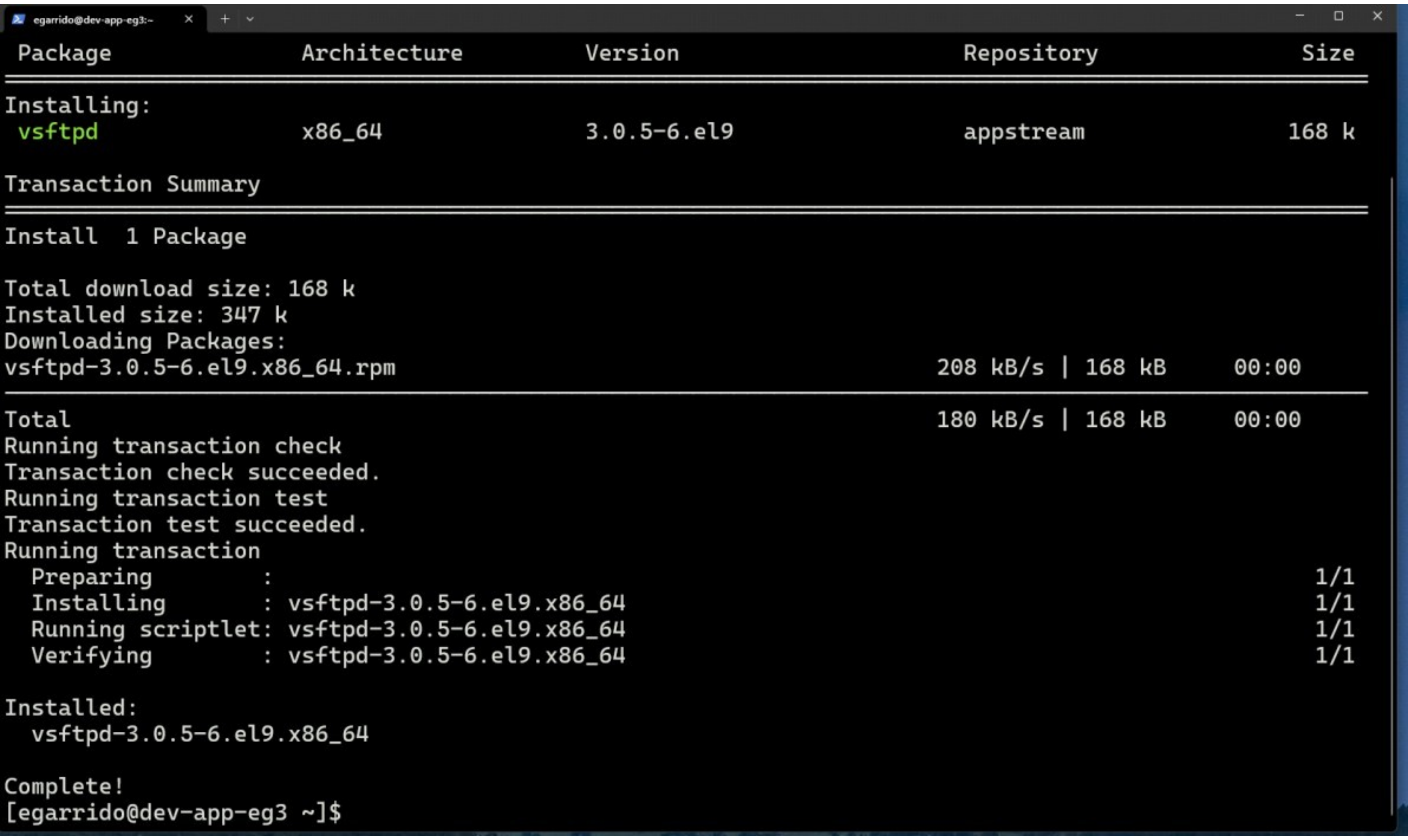
Each task is validated through service status checks, log verification, and functional testing to confirm that configurations are active and operating as intended. Where applicable, connectivity testing is performed to ensure services are reachable, properly authenticated, and able to perform real-world operations such as file transfers or remote checks.

Together, these tasks demonstrate a methodical approach to Linux system administration, highlighting the ability to deploy services, troubleshoot permission and access issues, and verify end-to-end functionality in a structured, production-style workflow.

Sanitization Notice:

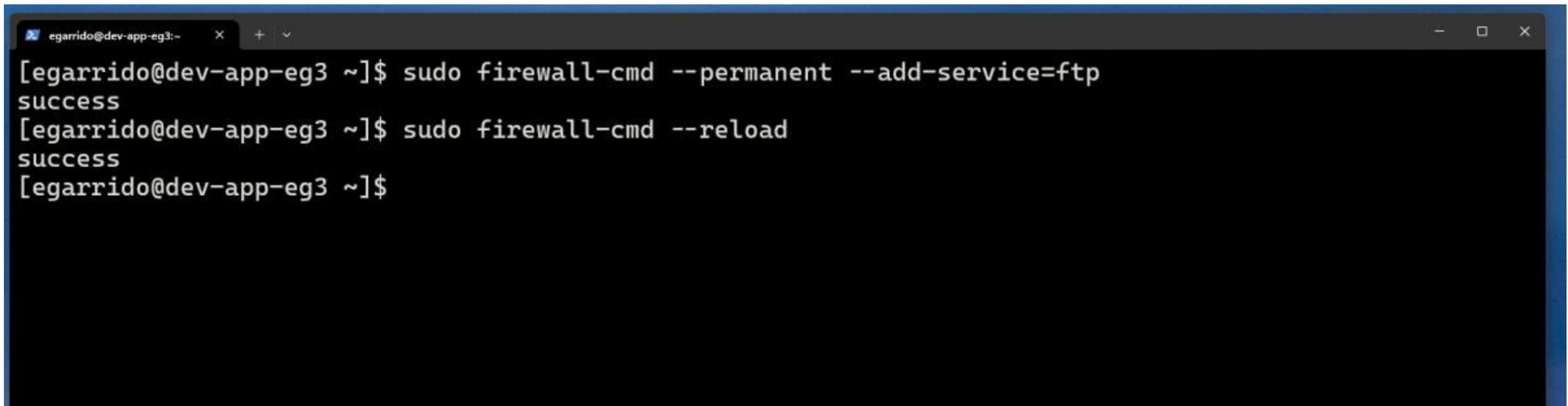
All hostnames, IP addresses, usernames, and environment identifiers shown in this documentation have been sanitized to protect sensitive infrastructure details.

The vsftpd (Very Secure FTP Daemon) package is installed successfully on the host using the system package manager. The output confirms the package version, architecture, and repository source, along with a completed transaction check and verification process. This step ensures that the FTP service binaries are present on the system and ready for subsequent configuration and service enablement



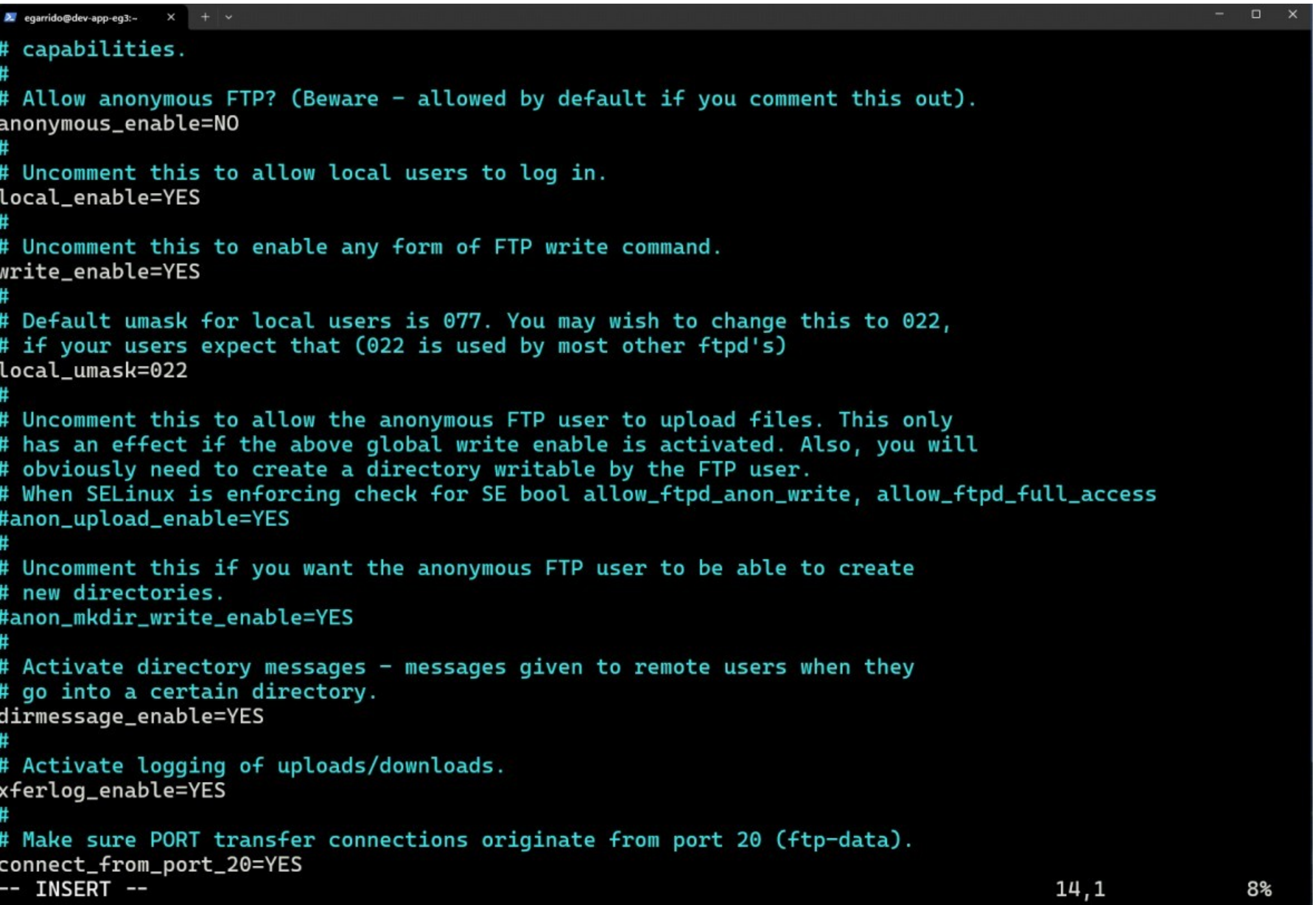
```
 Package                Architecture           Version                     Repository             Size
=========================================================================================================
Installing:
 vsftpd                 x86_64                 3.0.5-6.el9                 appstream              168 k

Transaction Summary
=========================================================================================================
Install  1 Package

Total download size: 168 k
Installed size: 347 k
Downloading Packages:
vsftpd-3.0.5-6.el9.x86_64.rpm                                       208 kB/s | 168 kB     00:00
---------------------------------------------------------------------------------------------------------
Total                                                               180 kB/s | 168 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                            1/1
  Installing       : vsftpd-3.0.5-6.el9.x86_64                                                  1/1
  Running scriptlet: vsftpd-3.0.5-6.el9.x86_64                                                  1/1
  Verifying        : vsftpd-3.0.5-6.el9.x86_64                                                  1/1

Installed:
  vsftpd-3.0.5-6.el9.x86_64

Complete!
[egarrido@dev-app-eg3 ~]$
```

The system firewall is updated to permanently allow FTP traffic by adding the FTP service to the active firewall configuration. The firewall rules are then reloaded to apply the change immediately, confirming successful rule deployment. This step ensures that the FTP service can accept incoming connections while maintaining controlled network access.



```
[egarrido@dev-app-eg3 ~]$ sudo firewall-cmd --permanent --add-service=ftp
success
[egarrido@dev-app-eg3 ~]$ sudo firewall-cmd --reload
success
[egarrido@dev-app-eg3 ~]$
```

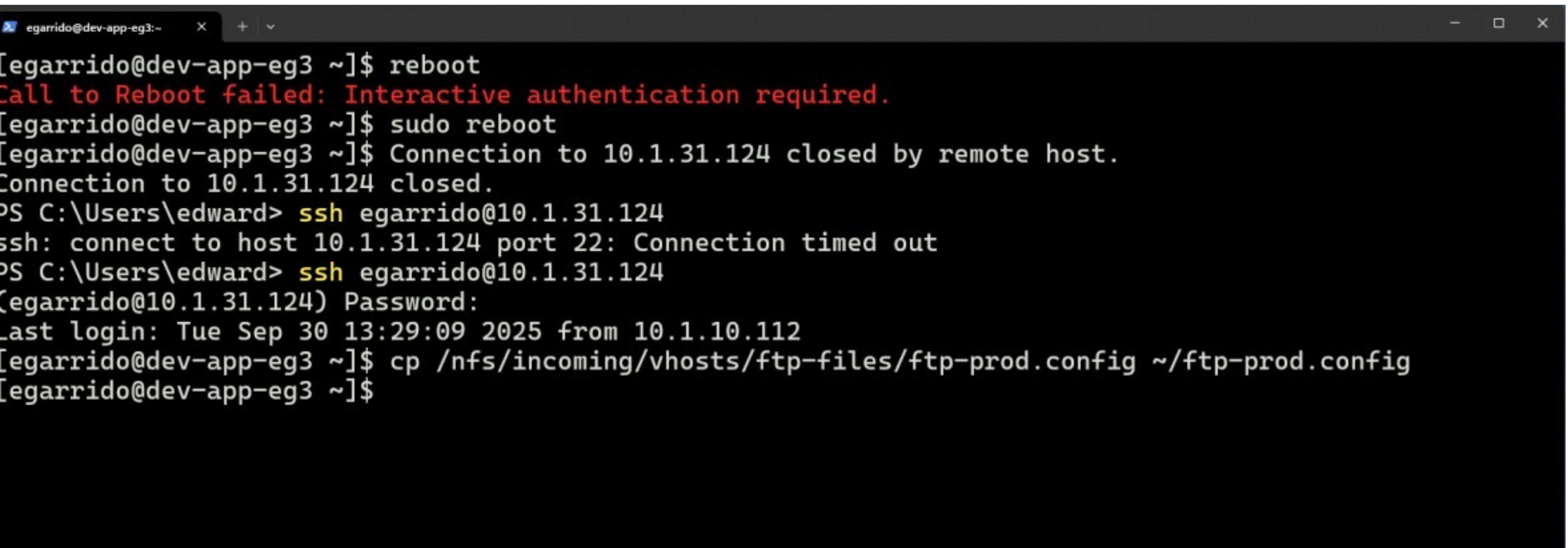
Core vsftpd configuration settings are updated to enable secure local user access while explicitly disabling anonymous FTP logins. Write permissions are enabled for authorized local users, with a defined file creation umask applied for consistent permissions. Logging is activated for FTP uploads and downloads, and standard FTP data connections are configured to originate from port 20, aligning the service with common security and operational best practices.



egarrido@dev-app-eg3:~

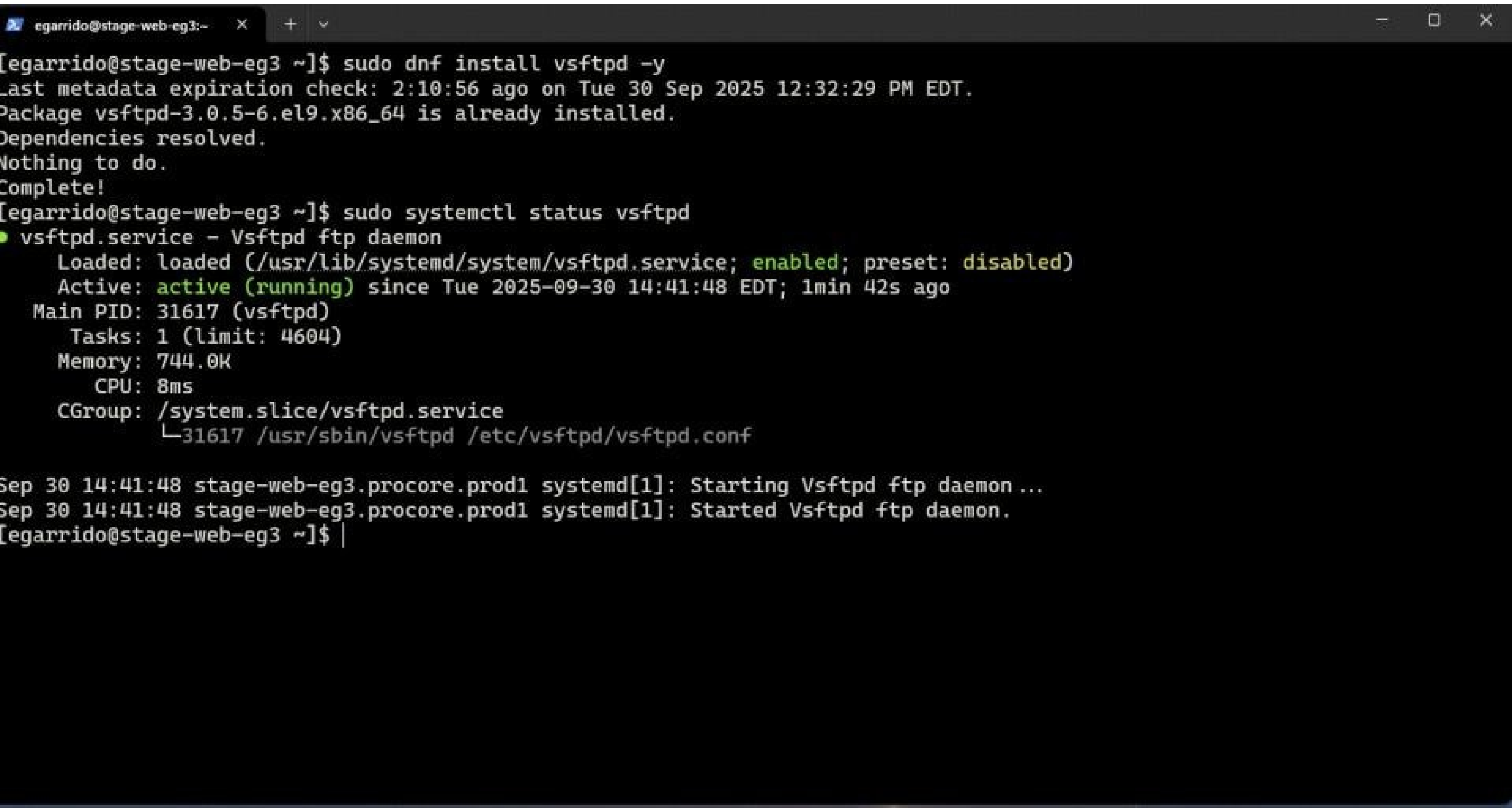
```
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write, allow_ftpd_full_access
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
-- INSERT --                                                          14,1          8%
```

A system reboot is initiated using elevated privileges after an initial attempt fails due to authentication requirements. Following the reboot, an SSH reconnectiaon is performed to restore administrative access to the host. Once access is reestablished, an FTP production configuration file is copied from a shared NFS location into the user’s home directory, preparing the system for final FTP service configuration.



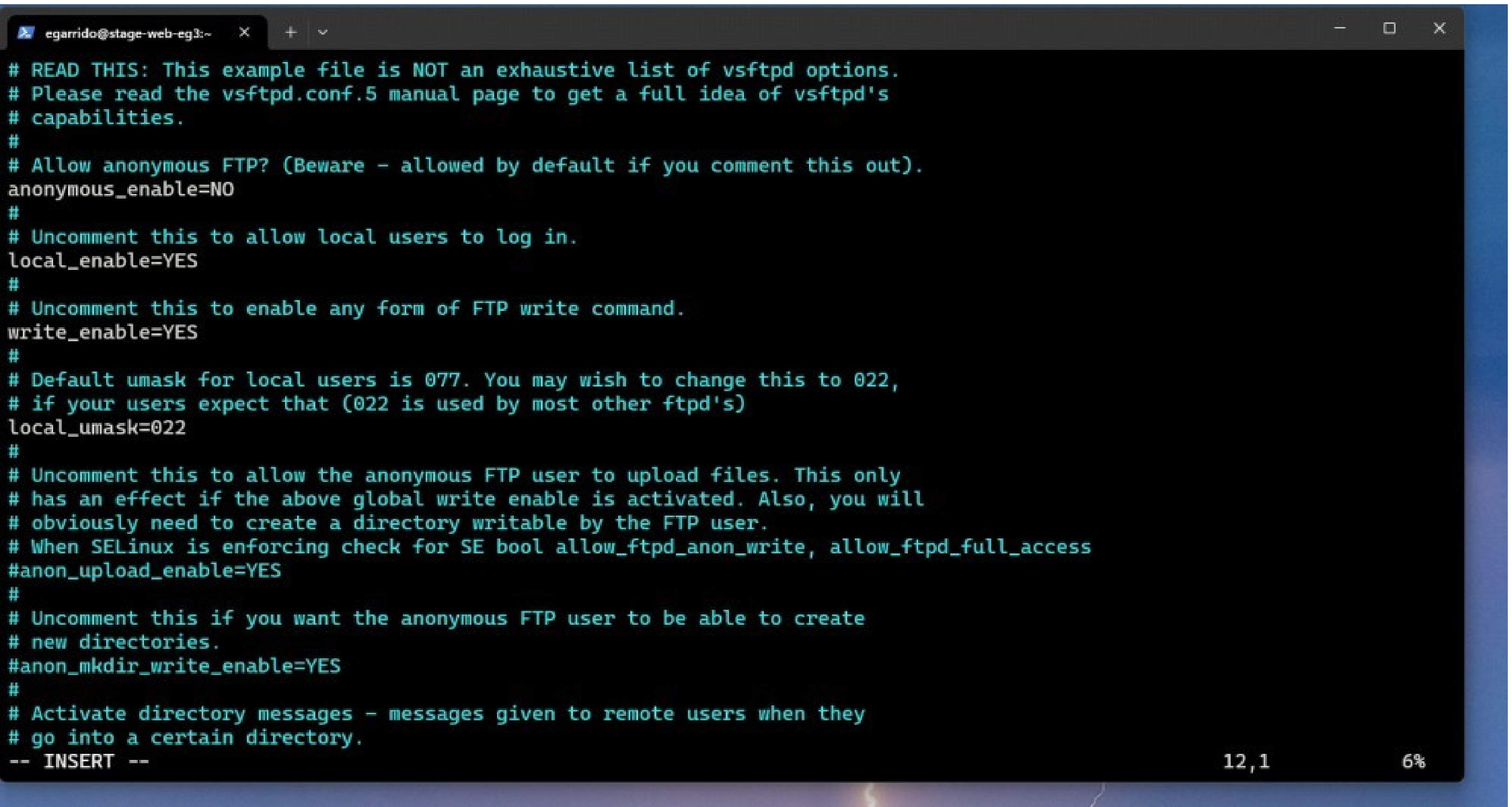
```
[egarrido@dev-app-eg3 ~]$ reboot
Call to Reboot failed: Interactive authentication required.
[egarrido@dev-app-eg3 ~]$ sudo reboot
[egarrido@dev-app-eg3 ~]$ Connection to 10.1.31.124 closed by remote host.
Connection to 10.1.31.124 closed.
PS C:\Users\edward> ssh egarrido@10.1.31.124
ssh: connect to host 10.1.31.124 port 22: Connection timed out
PS C:\Users\edward> ssh egarrido@10.1.31.124
(egarrido@10.1.31.124) Password:
Last login: Tue Sep 30 13:29:09 2025 from 10.1.10.112
[egarrido@dev-app-eg3 ~]$ cp /nfs/incoming/vhosts/ftp-files/ftp-prod.config ~/ftp-prod.config
[egarrido@dev-app-eg3 ~]$
```

The system confirms that vsftpd is already installed and proceeds to verify the FTP service status. The vsftpd.service is shown as enabled and actively running, with systemd logs indicating a successful service start. This validation step confirms that the FTP daemon is operational and using the expected configuration file, completing the service readiness check on the staging web host.



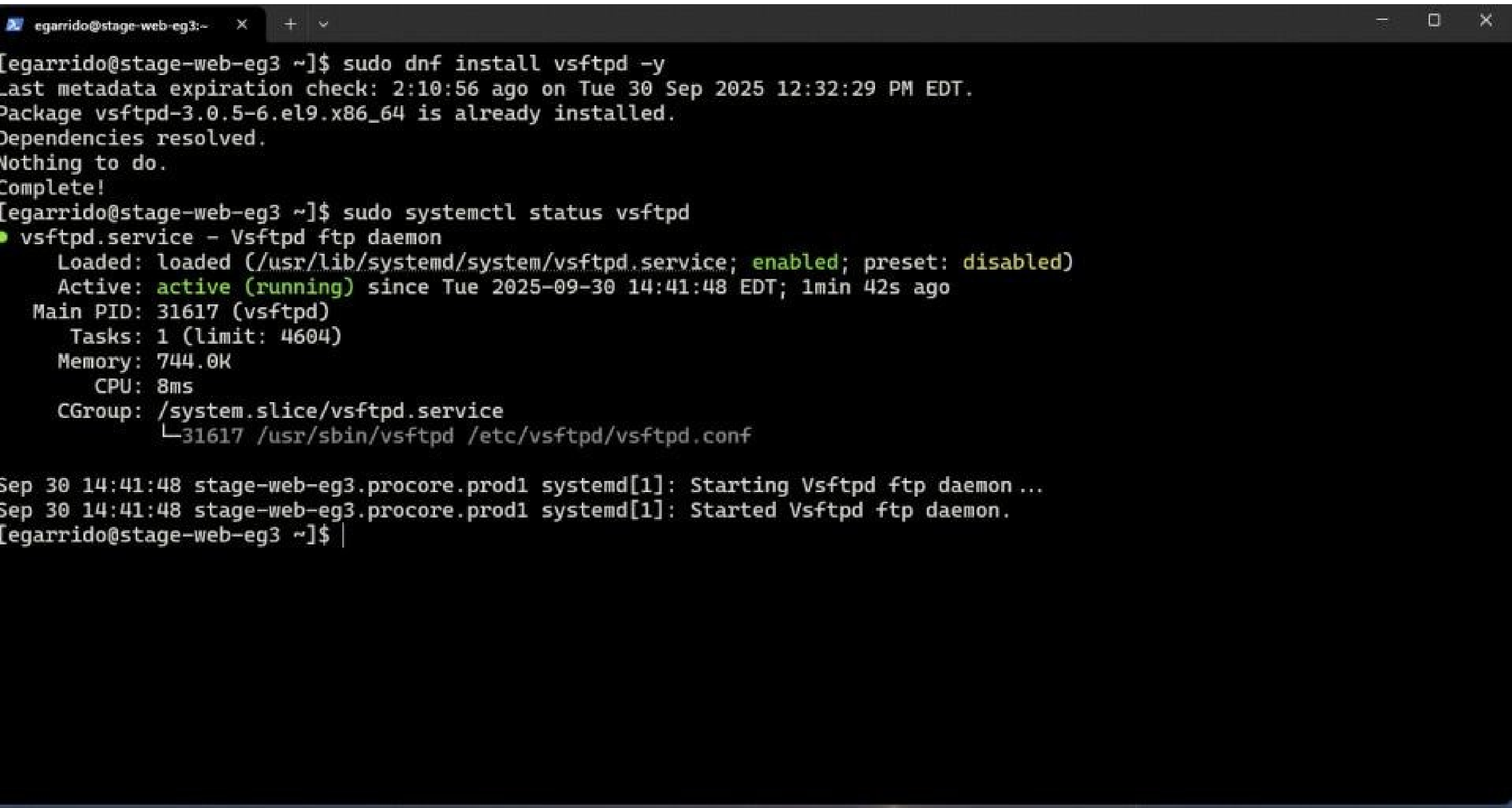
```
[egarrido@stage-web-eg3 ~]$ sudo dnf install vsftpd -y
Last metadata expiration check: 2:10:56 ago on Tue 30 Sep 2025 12:32:29 PM EDT.
Package vsftpd-3.0.5-6.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[egarrido@stage-web-eg3 ~]$ sudo systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
     Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-09-30 14:41:48 EDT; 1min 42s ago
   Main PID: 31617 (vsftpd)
      Tasks: 1 (limit: 4604)
     Memory: 744.0K
        CPU: 8ms
     CGroup: /system.slice/vsftpd.service
             └─31617 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 30 14:41:48 stage-web-eg3.procore.prod1 systemd[1]: Starting Vsftpd ftp daemon...
Sep 30 14:41:48 stage-web-eg3.procore.prod1 systemd[1]: Started Vsftpd ftp daemon.
[egarrido@stage-web-eg3 ~]$ |
```

The vsftpd configuration file is reviewed and updated to enforce secure FTP access policies. Anonymous FTP access is explicitly disabled, while local user authentication and write permissions are enabled. File creation permissions are standardized using a defined umask, and optional anonymous upload and directory creation settings remain commented to prevent unintended access. These settings ensure controlled, auditable FTP access aligned with security best practice


egarrido@stage-web-eg3:~

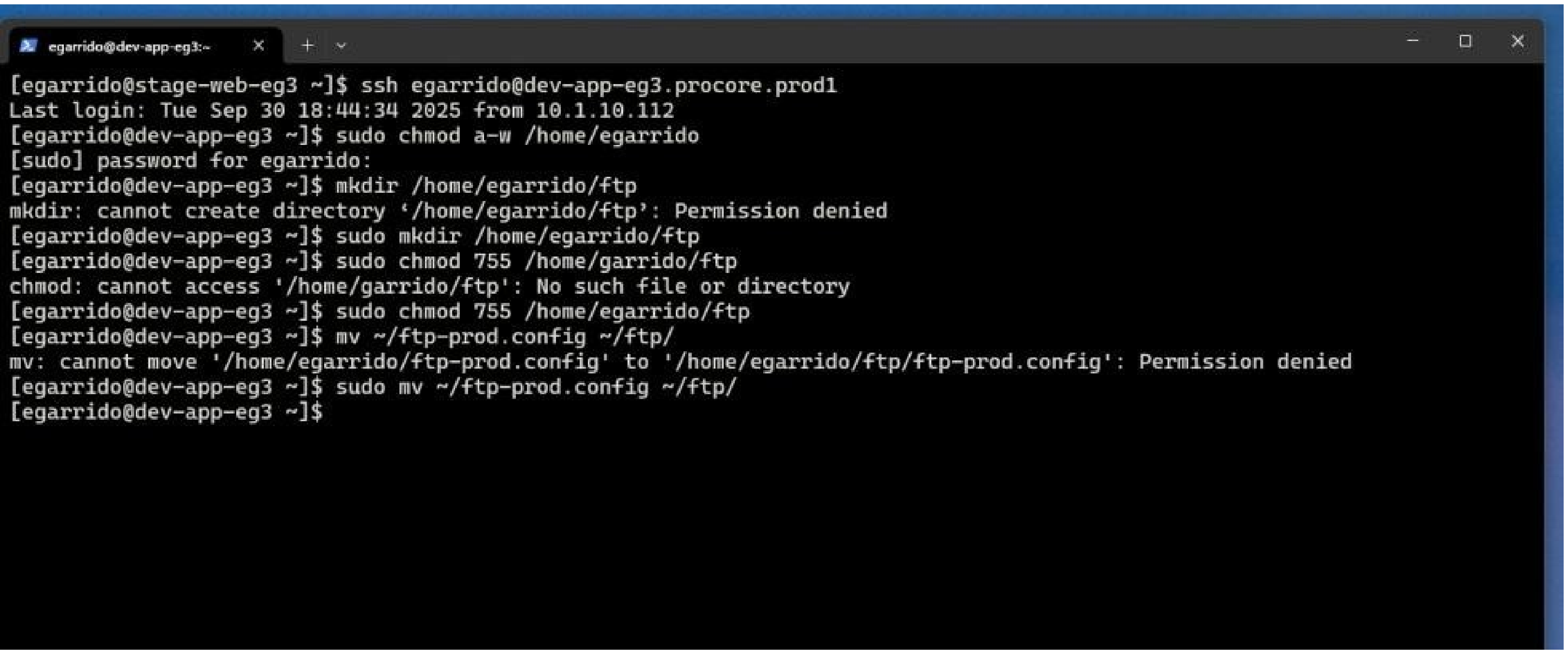
```
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write, allow_ftpd_full_access
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
-- INSERT --                                                          12,1          6%
```

The system verifies that the vsftpd package is already installed and confirms the operational status of the FTP service. The vsftpd.service is shown as enabled and actively running, with systemd logs indicating a successful daemon startup. This confirmation validates that the FTP service is correctly configured and running on the staging web host.



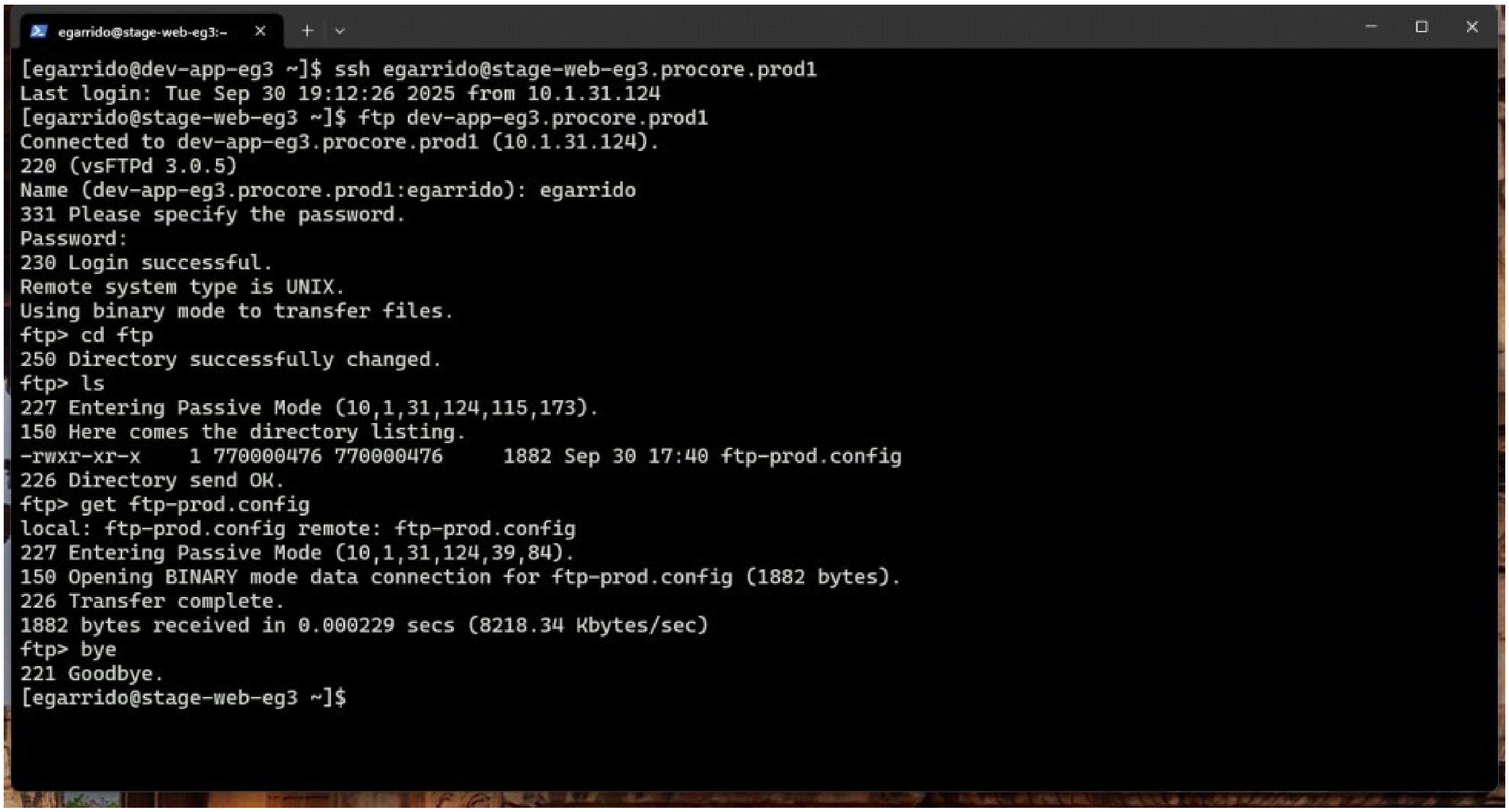
```
[egarrido@stage-web-eg3 ~]$ sudo dnf install vsftpd -y
Last metadata expiration check: 2:10:56 ago on Tue 30 Sep 2025 12:32:29 PM EDT.
Package vsftpd-3.0.5-6.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[egarrido@stage-web-eg3 ~]$ sudo systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
     Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-09-30 14:41:48 EDT; 1min 42s ago
   Main PID: 31617 (vsftpd)
      Tasks: 1 (limit: 4604)
     Memory: 744.0K
        CPU: 8ms
     CGroup: /system.slice/vsftpd.service
             └─31617 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 30 14:41:48 stage-web-eg3.procore.prod1 systemd[1]: Starting Vsftpd ftp daemon...
Sep 30 14:41:48 stage-web-eg3.procore.prod1 systemd[1]: Started Vsftpd ftp daemon.
[egarrido@stage-web-eg3 ~]$ |
```

Administrative access is established on the target host, followed by preparation of a dedicated FTP directory within the user's home path. Initial permission and ownership issues are encountered when creating and modifying the directory, requiring elevated privileges to complete the setup. After correcting permissions, the FTP production configuration file is successfully moved into the appropriate directory, completing the user-level FTP directory preparation



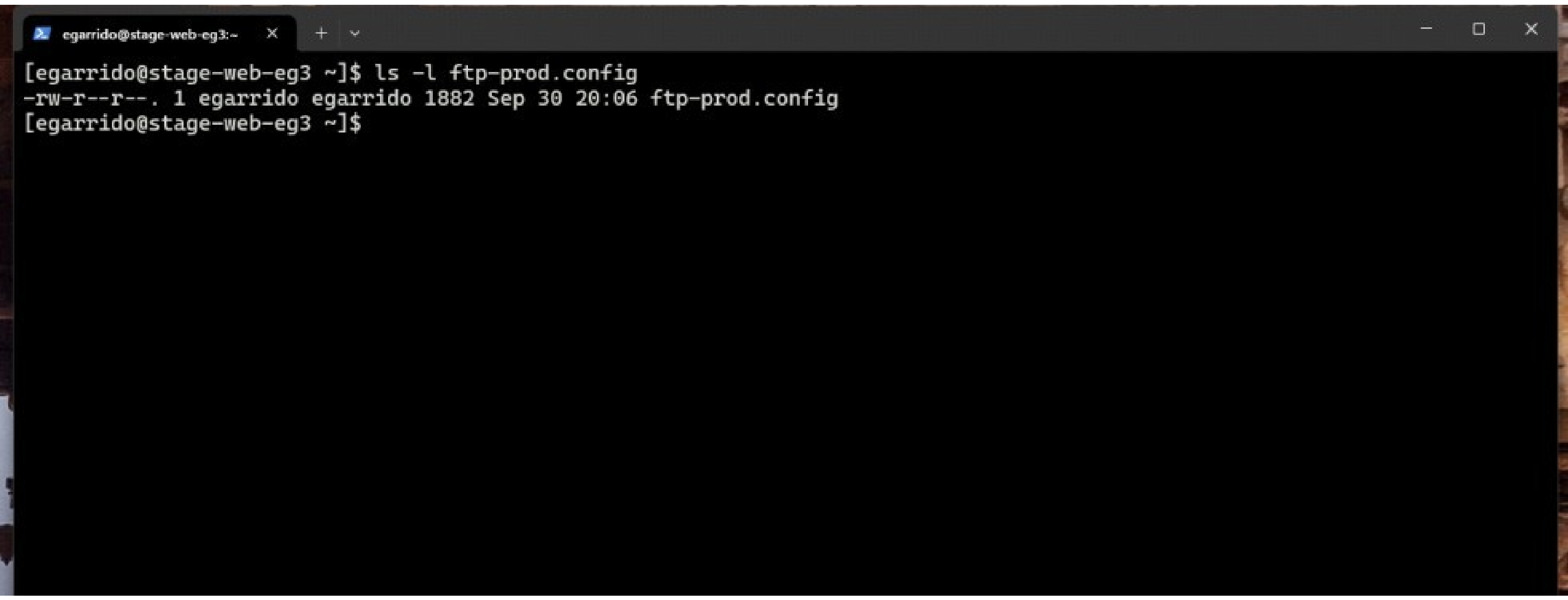
```
[egarrido@stage-web-eg3 ~]$ ssh egarrido@dev-app-eg3.procore.prod1
Last login: Tue Sep 30 18:44:34 2025 from 10.1.10.112
[egarrido@dev-app-eg3 ~]$ sudo chmod a-w /home/egarrido
[sudo] password for egarrido:
[egarrido@dev-app-eg3 ~]$ mkdir /home/egarrido/ftp
mkdir: cannot create directory ‘/home/egarrido/ftp’: Permission denied
[egarrido@dev-app-eg3 ~]$ sudo mkdir /home/egarrido/ftp
[egarrido@dev-app-eg3 ~]$ sudo chmod 755 /home/garrido/ftp
chmod: cannot access '/home/garrido/ftp': No such file or directory
[egarrido@dev-app-eg3 ~]$ sudo chmod 755 /home/egarrido/ftp
[egarrido@dev-app-eg3 ~]$ mv ~/ftp-prod.config ~/ftp/
mv: cannot move '/home/egarrido/ftp-prod.config' to '/home/egarrido/ftp/ftp-prod.config': Permission denied
[egarrido@dev-app-eg3 ~]$ sudo mv ~/ftp-prod.config ~/ftp/
[egarrido@dev-app-eg3 ~]$
```

FTP connectivity to the staging web host is validated by logging in with a local user account and establishing a successful session with the vsftpd service. After authentication, the FTP working directory is changed to the designated ftp directory, where the production configuration file is listed and retrieved using binary transfer mode. The successful file download confirms proper user authentication, directory access, and data transfer functionality for the FTP service.



```
[egarrido@dev-app-eg3 ~]$ ssh egarrido@stage-web-eg3.procore.prod1
Last login: Tue Sep 30 19:12:26 2025 from 10.1.31.124
[egarrido@stage-web-eg3 ~]$ ftp dev-app-eg3.procore.prod1
Connected to dev-app-eg3.procore.prod1 (10.1.31.124).
220 (vsFTPd 3.0.5)
Name (dev-app-eg3.procore.prod1:egarrido): egarrido
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd ftp
250 Directory successfully changed.
ftp> ls
227 Entering Passive Mode (10,1,31,124,115,173).
150 Here comes the directory listing.
-rwxr-xr-x    1 770000476 770000476     1882 Sep 30 17:40 ftp-prod.config
226 Directory send OK.
ftp> get ftp-prod.config
local: ftp-prod.config remote: ftp-prod.config
227 Entering Passive Mode (10,1,31,124,39,84).
150 Opening BINARY mode data connection for ftp-prod.config (1882 bytes).
226 Transfer complete.
1882 bytes received in 0.000229 secs (8218.34 Kbytes/sec)
ftp> bye
221 Goodbye.
[egarrido@stage-web-eg3 ~]$
```

File ownership and permissions for ftp-prod.config are confirmed using a detailed directory listing on the staging web host. The output verifies that the file exists in the expected location, is owned by the correct user and group, and has appropriate read and write permissions. This confirmation step validates the successful completion of the FTP file transfer and proper access control.



```
[egarrido@stage-web-eg3 ~]$ ls -l ftp-prod.config
-rw-r--r--. 1 egarrido egarrido 1882 Sep 30 20:06 ftp-prod.config
[egarrido@stage-web-eg3 ~]$
```

This set of work captures the configuration and validation of multiple Linux infrastructure services across development and staging environments. Tasks include service installation, secure configuration adjustments, firewall rule updates, permission and directory management, and functional testing to confirm operational readiness. Each service is verified through status checks and real-world usage to ensure reliability, security, and proper access control. Together, these activities demonstrate a disciplined, production-oriented approach to Linux system administration and service validation