

# SSH Key Authentication & Ansible Access

## Overview

This project documents the setup and verification of SSH key-based authentication to enable secure, passwordless access for Ansible-managed systems.

## Summary

Generated an RSA SSH key pair with ssh-keygen.

Verified the public key in ~/.ssh/id\_rsa.pub.

Deployed the key to a remote host using ssh-copy-id.

Confirmed passwordless SSH login.

Validated network configuration after successful access.

## Verification

```
ssh egarrido@dev-app-eg3.procore.prod1  
ip a
```

## Outcome

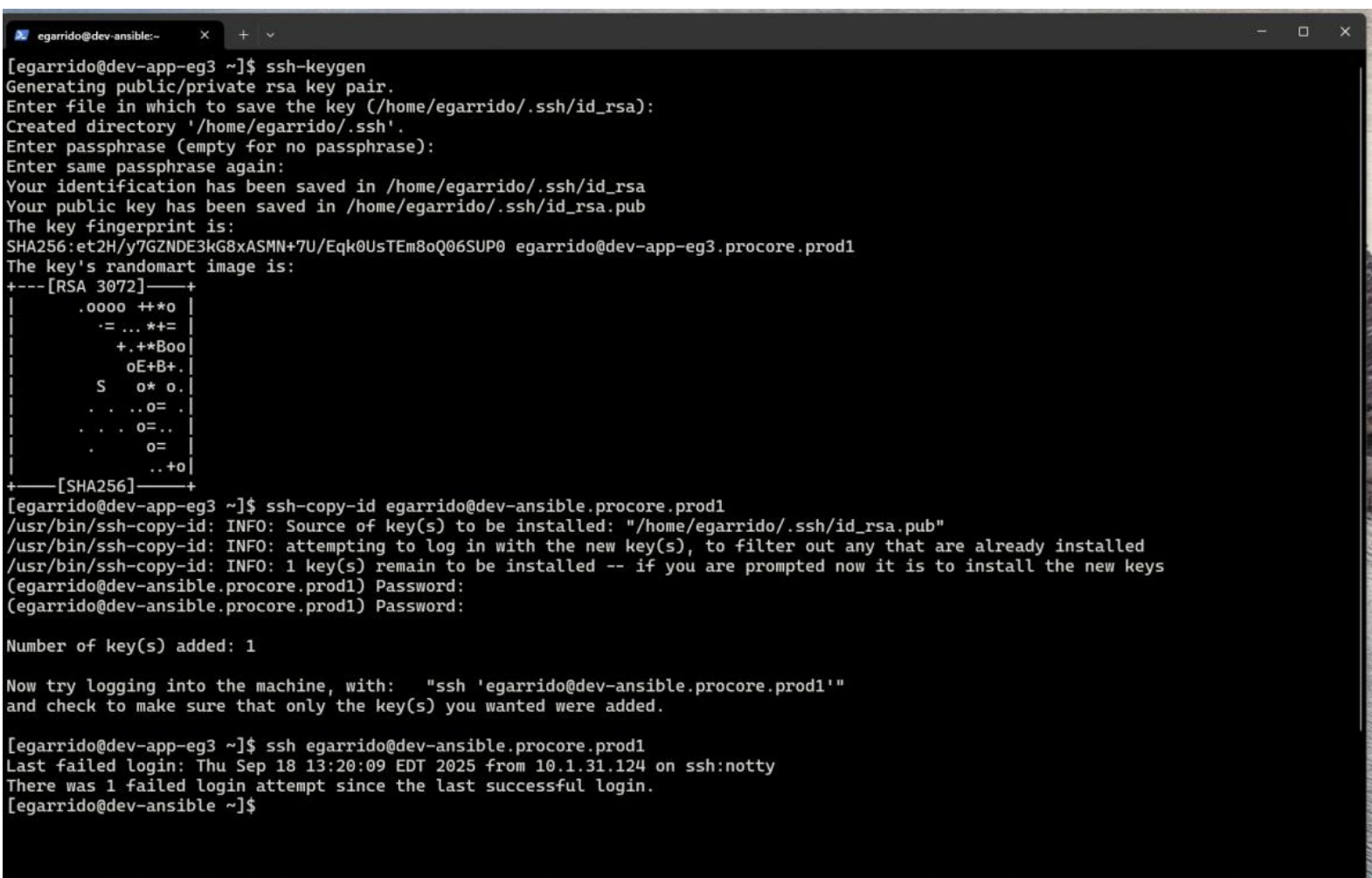
Secure SSH access is established and ready for Ansible automation.

An RSA SSH key pair is generated for the user egarrido using ssh-keygen.

The public key is copied to the remote host dev-ansible.procore.prod1 using ssh-copy-id.

Key-based authentication is successfully installed and confirmed.

Passwordless SSH access to the Ansible host is verified with a successful login.



```
[egarrido@dev-app-eg3 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/egarrido/.ssh/id_rsa):
Created directory '/home/egarrido/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/egarrido/.ssh/id_rsa
Your public key has been saved in /home/egarrido/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:et2H/y7GZNDE3kG8xASMN+7U/Eqk0UsTEm8oQ06SUP0 egarrido@dev-app-eg3.procore.prod1
The key's randomart image is:
+---[RSA 3072]-----+
|
|.0000 ++*o |
|'= ... *+=
|+.+*Boo|
|oE+B+.
|S o* o.
|. . .o= .
|. . .o=..
|. o=
|..+o
+---[SHA256]-----+
[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:
(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.

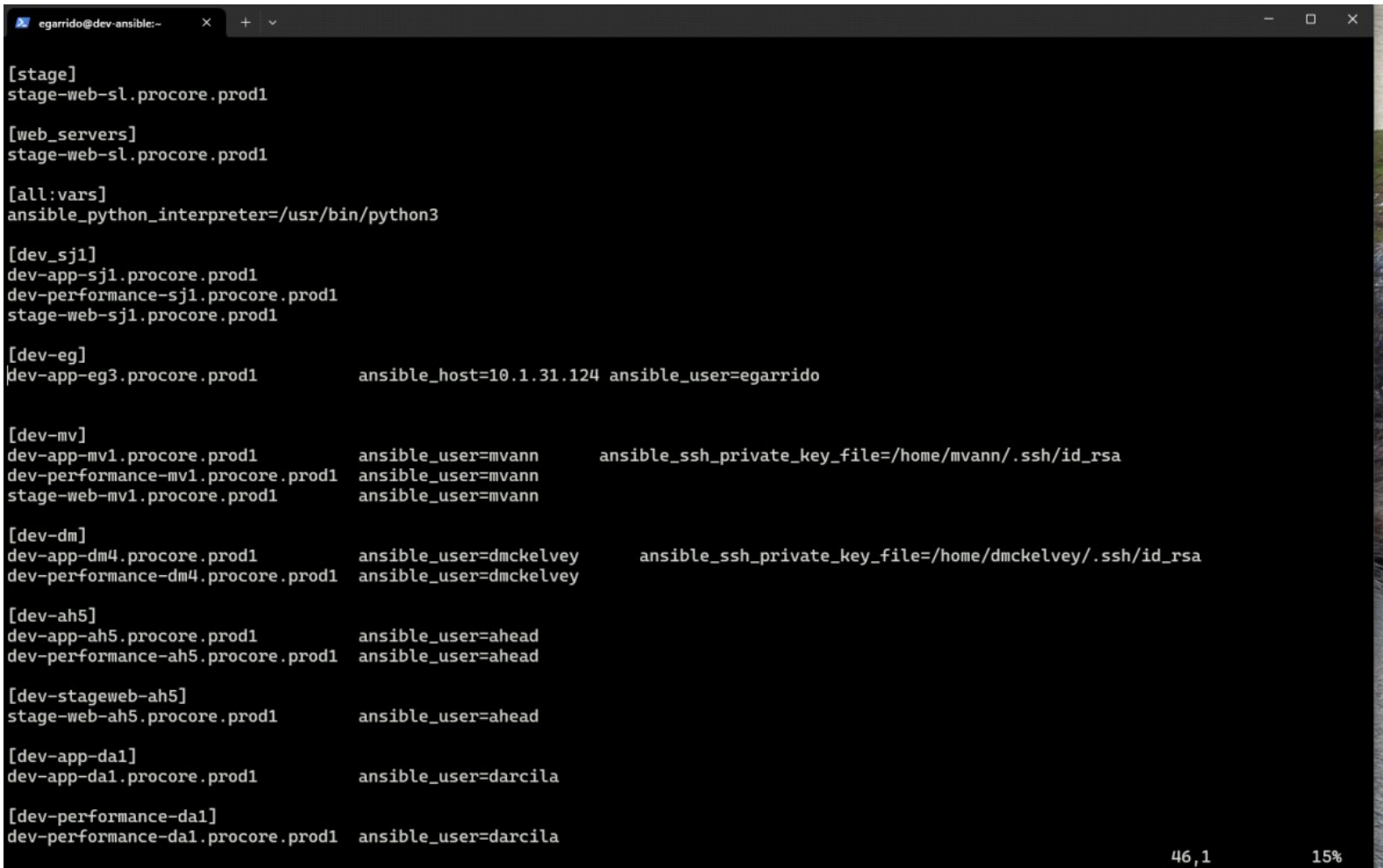
[egarrido@dev-app-eg3 ~]$ ssh egarrido@dev-ansible.procore.prod1
Last failed login: Thu Sep 18 13:20:09 EDT 2025 from 10.1.31.124 on ssh:notty
There was 1 failed login attempt since the last successful login.
[egarrido@dev-ansible ~]$
```

An Ansible inventory file is configured with multiple host groups for stage, dev, and performance environments.

Group variables define the Python interpreter for Ansible execution.

Hosts are mapped with explicit `ansible_user`, `ansible_host`, and SSH private key paths where required.

The inventory supports key-based authentication and role separation across environments, enabling consistent Ansible management at scale.

A screenshot of a terminal window with a dark background. The window title is "egarrido@dev-ansible:~". The terminal displays the content of an Ansible inventory file. It lists several host groups: [stage], [web\_servers], [all:vars], [dev\_sj1], [dev-eg], [dev-mv], [dev-dm], [dev-ah5], [dev-stageweb-ah5], [dev-app-da1], and [dev-performance-da1]. Each group lists hostnames and, for some groups, specific Ansible variables like ansible\_host, ansible\_user, and ansible\_ssh\_private\_key\_file. The bottom right corner of the terminal shows the page number "46,1" and a percentage "15%".

```
[stage]
stage-web-sl.procore.prod1

[web_servers]
stage-web-sl.procore.prod1

[all:vars]
ansible_python_interpreter=/usr/bin/python3

[dev_sj1]
dev-app-sj1.procore.prod1
dev-performance-sj1.procore.prod1
stage-web-sj1.procore.prod1

[dev-eg]
dev-app-eg3.procore.prod1      ansible_host=10.1.31.124 ansible_user=egarrido

[dev-mv]
dev-app-mv1.procore.prod1      ansible_user=mvann      ansible_ssh_private_key_file=/home/mvann/.ssh/id_rsa
dev-performance-mv1.procore.prod1 ansible_user=mvann
stage-web-mv1.procore.prod1    ansible_user=mvann

[dev-dm]
dev-app-dm4.procore.prod1      ansible_user=dmckelvey  ansible_ssh_private_key_file=/home/dmckelvey/.ssh/id_rsa
dev-performance-dm4.procore.prod1 ansible_user=dmckelvey

[dev-ah5]
dev-app-ah5.procore.prod1      ansible_user=ahead
dev-performance-ah5.procore.prod1 ansible_user=ahead

[dev-stageweb-ah5]
stage-web-ah5.procore.prod1    ansible_user=ahead

[dev-app-da1]
dev-app-da1.procore.prod1      ansible_user=darcila

[dev-performance-da1]
dev-performance-da1.procore.prod1 ansible_user=darcila
```

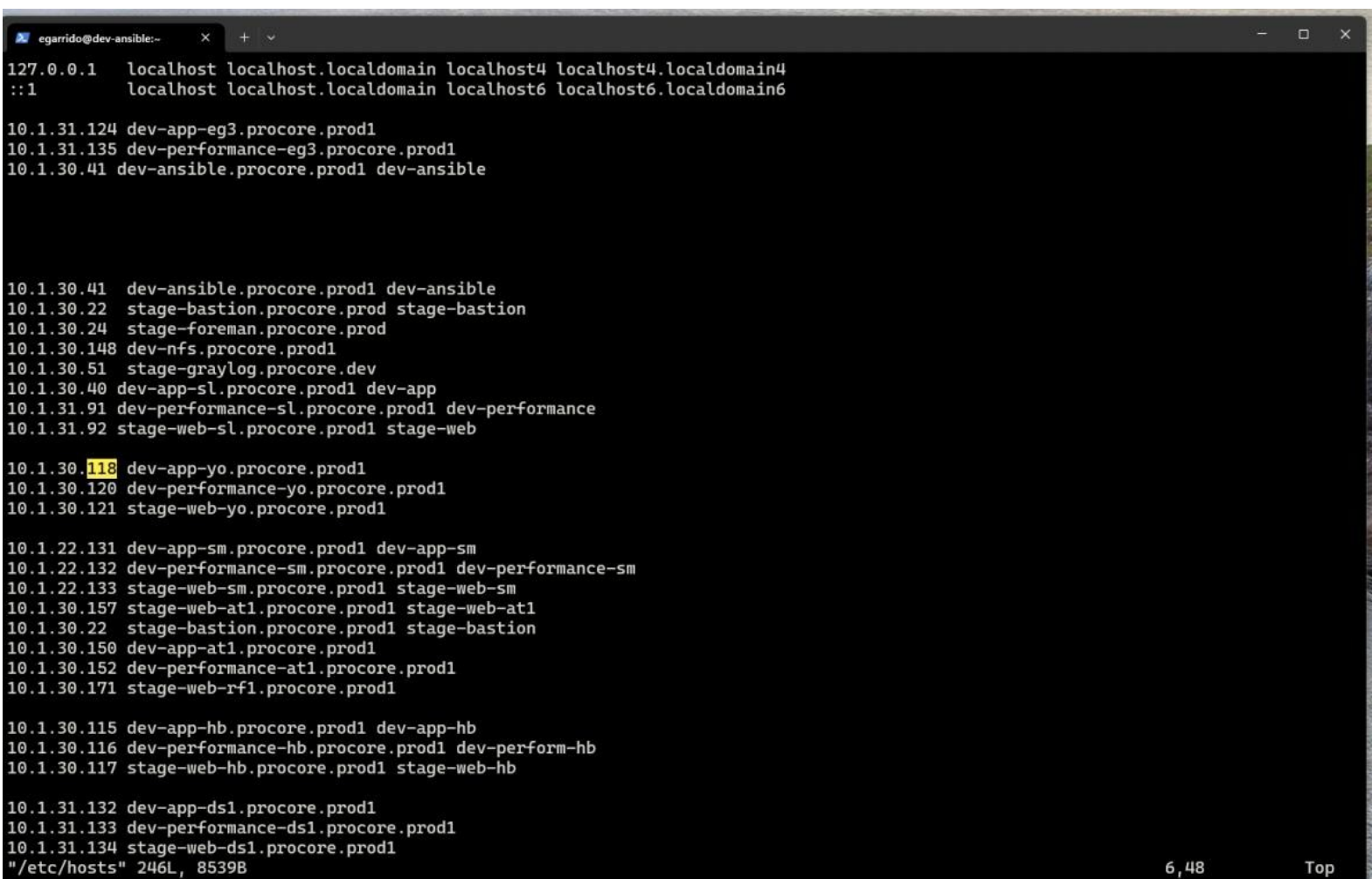
46,1 15%

The /etc/hosts file is configured with static hostname-to-IP mappings for dev, stage, and performance systems.

Entries include application servers, web servers, bastion hosts, and supporting infrastructure.

Host aliases are defined to simplify SSH access and Ansible inventory resolution.

Local name resolution ensures reliable connectivity in environments without relying solely on DNS.

A terminal window titled 'egarrido@dev-ansible:~' displays the contents of the /etc/hosts file. The file lists various IP addresses mapped to hostnames for development, stage, and performance environments. The mappings include localhost, dev-app-eg3, dev-performance-eg3, dev-ansible, stage-bastion, stage-foreman, dev-nfs, stage-graylog, dev-app-sl, dev-performance-sl, stage-web-sl, dev-app-yo, dev-performance-yo, stage-web-yo, dev-app-sm, dev-performance-sm, stage-web-sm, stage-web-at1, stage-bastion, dev-app-at1, dev-performance-at1, stage-web-rf1, dev-app-hb, dev-performance-hb, stage-web-hb, dev-app-ds1, dev-performance-ds1, and stage-web-ds1. The terminal output is as follows:

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

10.1.31.124 dev-app-eg3.procore.prod1
10.1.31.135 dev-performance-eg3.procore.prod1
10.1.30.41  dev-ansible.procore.prod1 dev-ansible

10.1.30.41  dev-ansible.procore.prod1 dev-ansible
10.1.30.22  stage-bastion.procore.prod stage-bastion
10.1.30.24  stage-foreman.procore.prod
10.1.30.148 dev-nfs.procore.prod1
10.1.30.51  stage-graylog.procore.dev
10.1.30.40  dev-app-sl.procore.prod1 dev-app
10.1.31.91  dev-performance-sl.procore.prod1 dev-performance
10.1.31.92  stage-web-sl.procore.prod1 stage-web

10.1.30.118 dev-app-yo.procore.prod1
10.1.30.120 dev-performance-yo.procore.prod1
10.1.30.121 stage-web-yo.procore.prod1

10.1.22.131 dev-app-sm.procore.prod1 dev-app-sm
10.1.22.132 dev-performance-sm.procore.prod1 dev-performance-sm
10.1.22.133 stage-web-sm.procore.prod1 stage-web-sm
10.1.30.157 stage-web-at1.procore.prod1 stage-web-at1
10.1.30.22  stage-bastion.procore.prod1 stage-bastion
10.1.30.150 dev-app-at1.procore.prod1
10.1.30.152 dev-performance-at1.procore.prod1
10.1.30.171 stage-web-rf1.procore.prod1

10.1.30.115 dev-app-hb.procore.prod1 dev-app-hb
10.1.30.116 dev-performance-hb.procore.prod1 dev-perform-hb
10.1.30.117 stage-web-hb.procore.prod1 stage-web-hb

10.1.31.132 dev-app-ds1.procore.prod1
10.1.31.133 dev-performance-ds1.procore.prod1
10.1.31.134 stage-web-ds1.procore.prod1
"/etc/hosts" 246L, 8539B
```

SSH access to dev-app-eg3.procore.prod1 is successful using key-based authentication.

Network interfaces are verified with ip a, confirming the primary interface is UP.

The system is configured with a static IPv4 address 10.1.31.124/23 on interface ens192.

Network configuration is validated, confirming proper connectivity and host accessibility

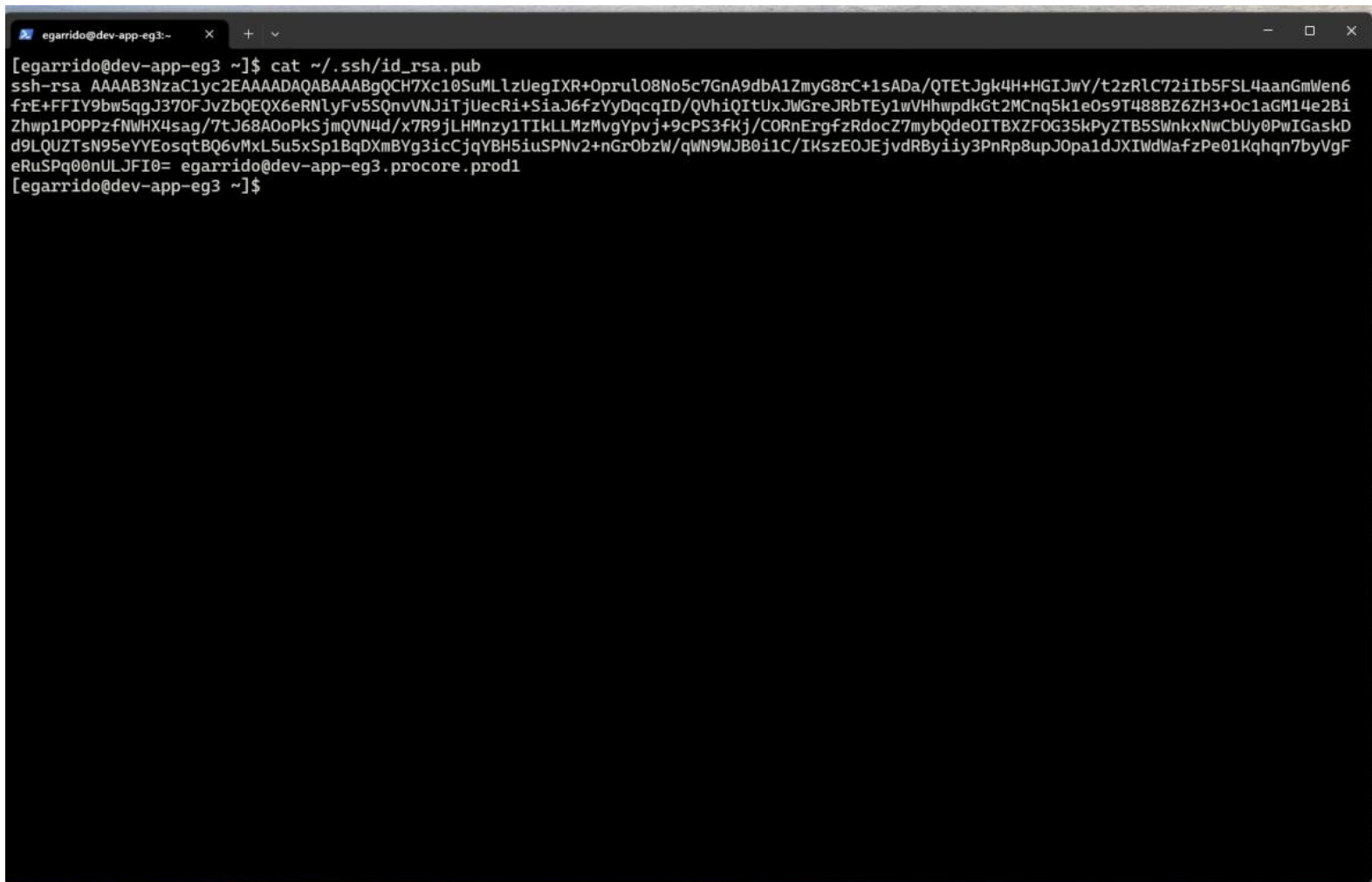
```
egarrido@dev-app-eg3:~  
[egarrido@dev-ansible ~]$ ssh egarrido@dev-app-eg3.procore.prod1  
Last login: Thu Sep 18 17:29:47 2025 from 10.1.30.41  
[egarrido@dev-app-eg3 ~]$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:50:56:8b:d4:ac brd ff:ff:ff:ff:ff:ff  
    altname enp11s0  
    inet 10.1.31.124/23 brd 10.1.31.255 scope global noprefixroute ens192  
        valid_lft forever preferred_lft forever  
    inet6 fe80::bc8f:e46c:7d3:11ac/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
[egarrido@dev-app-eg3 ~]$
```

The public SSH key is displayed from ~/.ssh/id\_rsa.pub.

The key confirms successful generation of an RSA key pair for the user egarrido.

This public key is used for key-based authentication with remote systems.

Verifies readiness for passwordless SSH access and Ansible automation.

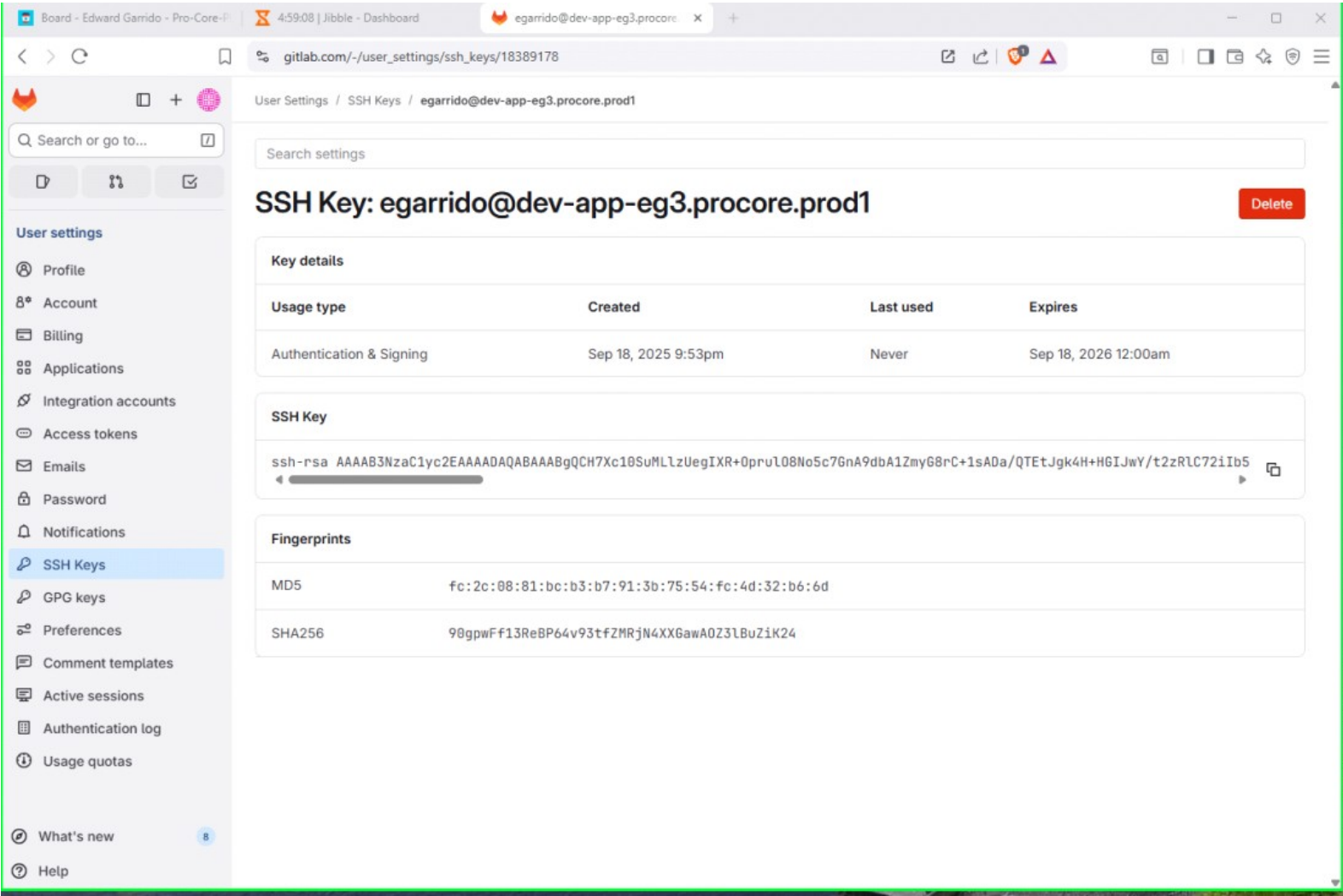
A terminal window with a dark background and light-colored text. The window title is 'egarrido@dev-app-eg3:~'. The prompt is '[egarrido@dev-app-eg3 ~]\$'. The user has entered the command 'cat ~/.ssh/id\_rsa.pub'. The output is a long string of text representing the public SSH key, starting with 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCH7Xc10SuMLlZUegIXR+Oprul08No5c7GnA9dbA1ZmyG8rC+1sADa/QTEtJgk4H+HGIJwY/t2zRlC72iIb5FSL4aanGmWen6frE+FFIY9bw5qgJ370FJvZbQEQX6eRNlyFv5SQnvVNJiTjUecRi+SiaJ6fzYyDqcqID/QVhiQITUxJWGreJRbTEy1wVHhwpdkGt2MCnq5k1e0s9T488BZ6ZH3+0c1aGM14e2BiZhwp1POPPzfNWX4sag/7tJ68A0oPkSjmQVN4d/x7R9jLHMnzy1TIkLLMzMvgYpvj+9cPS3fKj/CORnErgfzRdocZ7mybQdeOITBXZF0G35kPyZTB5SWnkxNwCbUy0PwIGaskDd9LQUZTsN95eYYEosqtBQ6vMxL5u5xSp1BqDXmBYg3icCjqYBH5iuSPNv2+nGrObzW/qWN9WJB0i1C/IKszE0JEjvdRByiiy3PnRp8upJ0pa1dJXIwdWafzPe01Kqhqn7byVgFeRuSPq00nULJFI0=' and ending with 'egarrido@dev-app-eg3.prod1'. The prompt is '[egarrido@dev-app-eg3 ~]\$'.

An SSH public key is successfully added to GitLab under user settings.

The key is configured for authentication and signing and is linked to the host dev-app-eg3.procore.prod1.

Fingerprints (MD5 and SHA256) confirm key integrity.

This enables secure, key-based access for Git operations without using passwords.



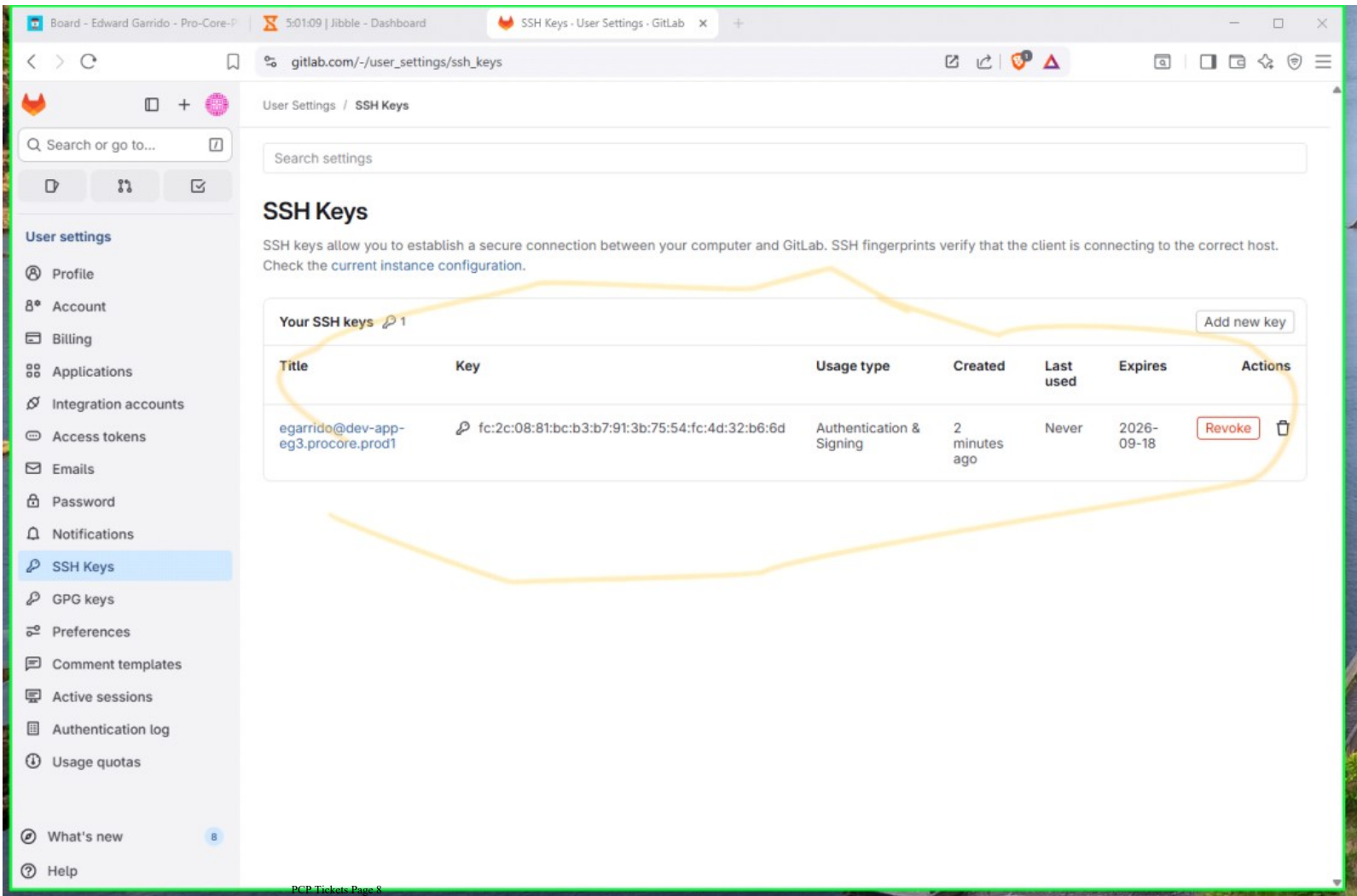


The SSH public key is successfully registered in GitLab under User Settings → SSH Keys.

The key is enabled for Authentication & Signing and associated with egarrido@dev-app-eg3.procore.prod1.

Creation time, expiration date, and fingerprint are visible, confirming proper key registration.

GitLab is now configured for secure, passwordless Git operations using SSH.





SSH connectivity to GitLab is tested using `ssh -T git@gitlab.com`.

The GitLab host fingerprint is reviewed and accepted, adding it to the local `known_hosts` file.

The SSH public key is recognized by GitLab, confirming successful authentication.

A welcome message confirms secure, passwordless SSH access for Git operations.

```
egarrido@dev-app-eg3:~$ ssh -T git@gitlab.com
The authenticity of host 'gitlab.com (<no hostip for proxy command>)' can't be established.
ED25519 key fingerprint is SHA256:eUXGGm1YGsMAS7vkcx6JOJdOGHPem5gQp4taiCfCLB8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.
[egarrido@dev-app-eg3 ~]$ sudo vi /etc/hosts
[sudo] password for egarrido:
[egarrido@dev-app-eg3 ~]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCh7Xc10SuMLlZUegIXR+Oprul08No5c7GnA9dbA1ZmyG8rC+1sADa/QTETJgk4H+HGIJwY/t2zRLC72iIb5FSL4aanGmWen6
frE+FFIY9bw5qgJ370FJvZbQEQU6eRNlyFv5S5QnvVNJiJtJuecRi+SiaJ6fzYyDqcqID/QVhiQItUxJwGreJRbTEy1wVHhwpdkGt2MCnq5k1e0s9T488BZ6ZH3+0c1aGM14e2Bi
Zhwp1POPPzfNwHX4sag/7tJ68A0oPkSjmQVN4d/x7R9jLHMnzy1TIkLLMzMvgYpvj+9cPS3fKj/CORnErgfzRdocZ7mybQde0ITBXZF0G35kPyZTB5SwnkxNmCbUy0PwIGaskD
d9LQUZTsN95eYYEosqtBQ6vMxL5u5xSp1BqDXmBYg3icCjqYBH5iuSPNv2+nGr0bzW/qWn9WJB0i1C/IKszEOJEjvdRByiiy3PnRp8upJ0pa1dJXIWdWafzPe01Kqhqn7byVgF
eRuSPq00nULJFI0= egarrido@dev-app-eg3.procure.prod1
[egarrido@dev-app-eg3 ~]$ ssh -T git@gitlab.com
The authenticity of host 'gitlab.com (<no hostip for proxy command>)' can't be established.
ED25519 key fingerprint is SHA256:eUXGGm1YGsMAS7vkcx6JOJdOGHPem5gQp4taiCfCLB8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'gitlab.com' (ED25519) to the list of known hosts.
Welcome to GitLab, @edwardgarrido3066!
[egarrido@dev-app-eg3 ~]$
```

Switched to the egarrido user and created a home directory on the performance host.

Generated a new RSA SSH key pair using ssh-keygen.

Copied the public key to the Ansible control node with ssh-copy-id.

Successfully authenticated to dev-ansible.procore.prod1 using key-based SSH.

Confirms passwordless access from the performance system to the Ansible host, enabling cross-host automation

```
egarrido@dev-ansible:~  
[root@dev-performance-eg3 ~]# su - egarrido  
Creating home directory for egarrido.  
[egarrido@dev-performance-eg3 ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/egarrido/.ssh/id_rsa):  
Created directory '/home/egarrido/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/egarrido/.ssh/id_rsa  
Your public key has been saved in /home/egarrido/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:z4jbH6909j7onpIoUmzAh/Xi2VyPmAN9BE1pkZHIwbQ egarrido@dev-performance-eg3.procore.prod1  
The key's randomart image is:  
+---[RSA 3072]---+  
|         .+++.         |  
|      . +.B          |  
|    . o oEo         |  
|  + + o o          |  
|   = * =So         |  
|    * *..+.        |  
|    o .o..*.       |  
|   . . .ooo.=.     |  
|   . . .*Bo+.      |  
+---[SHA256]---+  
[egarrido@dev-performance-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-performance-eg3 ~]$ ssh egarrido@dev-ansible.procore.prod1  
Last login: Thu Sep 18 17:34:30 2025 from 10.1.31.124
```

The Ansible inventory is updated to include both development and performance hosts.

Hosts are grouped by environment and role for organized targeting.

User-specific SSH settings (ansible\_user, ansible\_host, and private key paths) are defined per group.

A global Python interpreter is set to ensure consistent Ansible execution.

The inventory supports secure, key-based access across multiple environments for automation.

```
egarrido@dev-ansible:~  
stage-web-sl.procore.prod1  
[all:vars]  
ansible_python_interpreter=/usr/bin/python3  
[dev-sj1]  
dev-app-sj1.procore.prod1  
dev-performance-sj1.procore.prod1  
stage-web-sj1.procore.prod1  
[dev-eg]  
dev-app-eg3.procore.prod1      ansible_host=10.1.31.124  ansible_user=egarrido  
dev-performance.procore.prod1  ansible_host=10.1.31.135  ansible_user=egarrido  
[dev-mv]  
dev-app-mv1.procore.prod1      ansible_user=mvann        ansible_ssh_private_key_file=/home/mvann/.ssh/id_rsa  
dev-performance-mv1.procore.prod1  ansible_user=mvann  
stage-web-mv1.procore.prod1      ansible_user=mvann  
[dev-dm]  
dev-app-dm4.procore.prod1      ansible_user=dmckelvey    ansible_ssh_private_key_file=/home/dmckelvey/.ssh/id_rsa  
dev-performance-dm4.procore.prod1  ansible_user=dmckelvey  
[dev-ah5]  
dev-app-ah5.procore.prod1      ansible_user=ahead  
dev-performance-ah5.procore.prod1  ansible_user=ahead  
[dev-stageweb-ah5]  
stage-web-ah5.procore.prod1      ansible_user=ahead  
[dev-app-da1]  
dev-app-da1.procore.prod1      ansible_user=darcila  
[dev-performance-da1]  
dev-performance-da1.procore.prod1  ansible_user=darcila  
-- INSERT --
```

The /etc/hosts file is updated with static IP-to-hostname mappings for dev, stage, and performance environments.

Entries include application servers, web servers, bastion hosts, and supporting infrastructure.

Host aliases are defined to simplify SSH access and Ansible targeting.

Local name resolution is ensured without relying solely on DNS, supporting consistent automation and connectivity

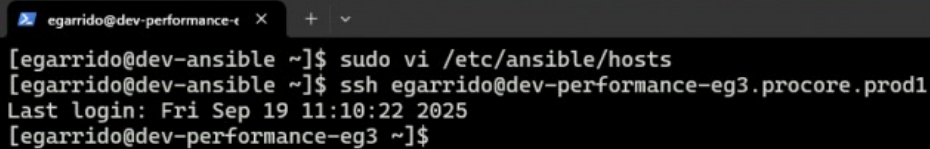
```
egarrido@dev-ansible:~  
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
10.1.31.124 dev-app-eg3.procore.prod1  
10.1.31.135 dev-performance-eg3.procore.prod1  
10.1.30.41 dev-ansible.procore.prod1 dev-ansible  
  
10.1.30.41 dev-ansible.procore.prod1 dev-ansible  
10.1.30.22 stage-bastion.procore.prod stage-bastion  
10.1.30.24 stage-foreman.procore.prod  
10.1.30.148 dev-nfs.procore.prod1  
10.1.30.51 stage-graylog.procore.dev  
10.1.30.40 dev-app-sl.procore.prod1 dev-app  
10.1.31.91 dev-performance-sl.procore.prod1 dev-performance  
10.1.31.92 stage-web-sl.procore.prod1 stage-web  
  
10.1.30.118 dev-app-yo.procore.prod1  
10.1.30.120 dev-performance-yo.procore.prod1  
10.1.30.121 stage-web-yo.procore.prod1  
  
10.1.22.131 dev-app-sm.procore.prod1 dev-app-sm  
10.1.22.132 dev-performance-sm.procore.prod1 dev-performance-sm  
10.1.22.133 stage-web-sm.procore.prod1 stage-web-sm  
10.1.30.157 stage-web-at1.procore.prod1 stage-web-at1  
10.1.30.22 stage-bastion.procore.prod1 stage-bastion  
10.1.30.150 dev-app-at1.procore.prod1  
10.1.30.152 dev-performance-at1.procore.prod1  
10.1.30.171 stage-web-rf1.procore.prod1  
  
10.1.30.115 dev-app-hb.procore.prod1 dev-app-hb  
10.1.30.116 dev-performance-hb.procore.prod1 dev-perform-hb  
10.1.30.117 stage-web-hb.procore.prod1 stage-web-hb
```

The Ansible inventory file is edited on the control node at /etc/ansible/hosts.

SSH access to dev-performance-eg3.procore.prod1 is tested and succeeds.

Successful login confirms correct inventory configuration and key-based authentication.

Validates end-to-end connectivity from the Ansible control node to a performance host.

A terminal window with a dark background and light text. The window title bar shows 'egarrido@dev-performance-eg3' and standard window controls. The terminal output shows a user at a dev-ansible host editing the Ansible inventory file, then successfully logging in to a dev-performance-eg3 host via SSH. The last login time is shown as Fri Sep 19 11:10:22 2025.

```
egarrido@dev-performance-eg3 X + v
[egarrido@dev-ansible ~]$ sudo vi /etc/ansible/hosts
[egarrido@dev-ansible ~]$ ssh egarrido@dev-performance-eg3.procore.prod1
Last login: Fri Sep 19 11:10:22 2025
[egarrido@dev-performance-eg3 ~]$
```

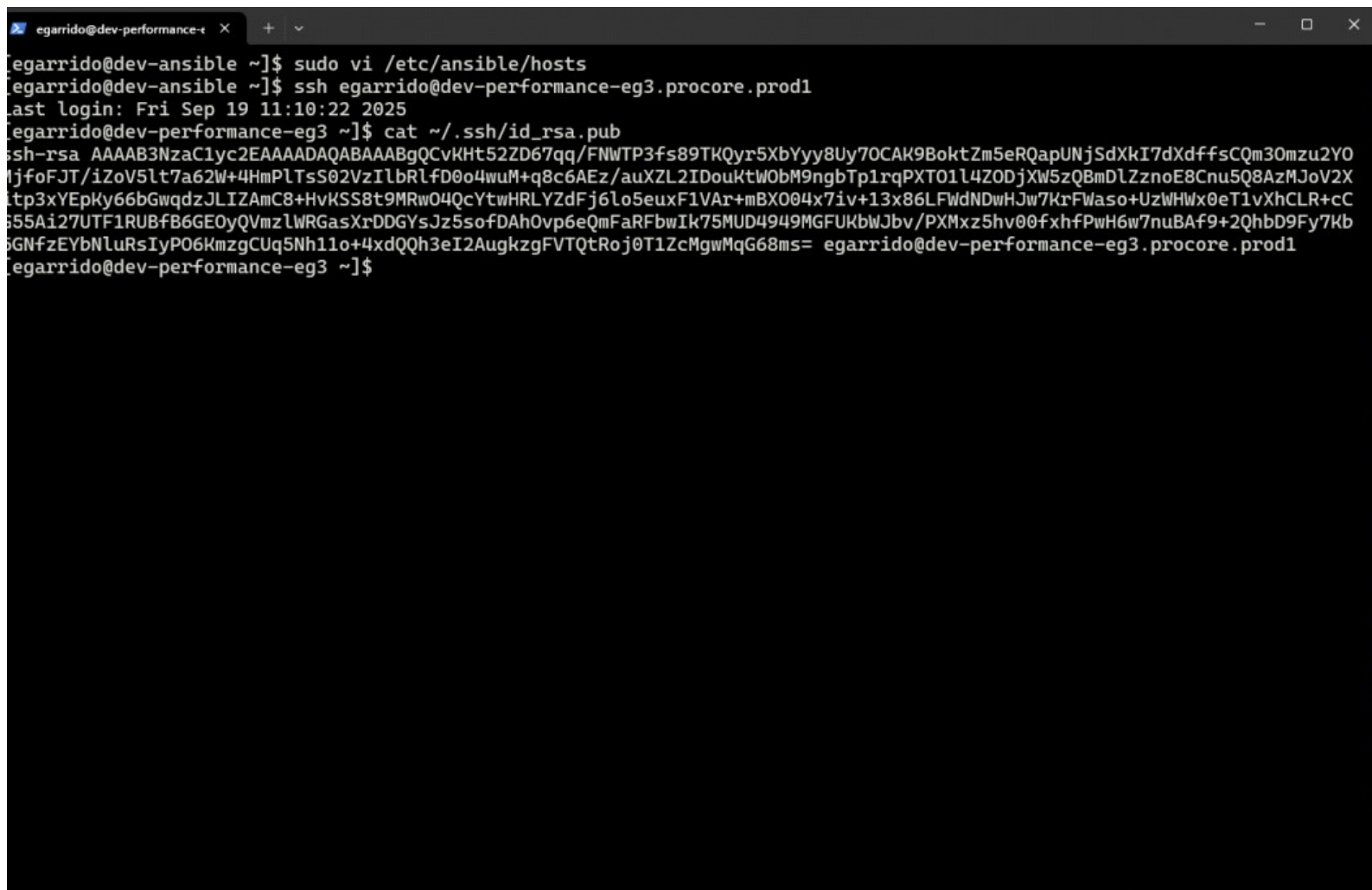


SSH access to dev-performance-eg3.procore.prod1 is confirmed from the Ansible control node.

The public SSH key is verified on the performance host at ~/.ssh/id\_rsa.pub.

Key-based authentication is successfully in place for the egarrido user.

Confirms the host is fully prepared for Ansible-managed automation without password prompts

A terminal window with a dark background and light text. The window title is 'egarrido@dev-performance-eg3'. The terminal shows a sequence of commands and their outputs. First, 'sudo vi /etc/ansible/hosts' is run. Then, 'ssh egarrido@dev-performance-eg3.procore.prod1' is run, resulting in a 'last login' message. Finally, 'cat ~/.ssh/id\_rsa.pub' is run, displaying a long SSH public key string. The prompt returns to '~]\$' after each command.

```
egarrido@dev-ansible ~]$ sudo vi /etc/ansible/hosts
egarrido@dev-ansible ~]$ ssh egarrido@dev-performance-eg3.procore.prod1
Last login: Fri Sep 19 11:10:22 2025
egarrido@dev-performance-eg3 ~]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCvKht52ZD67qq/FNWTP3fs89TKQyr5XbYyy8Uy70CAK9BoktZm5eRQapUNjSdXkI7dXdfCsCQm30mzu2Y0
HjfoFJT/iZoV5lt7a62W+4HmPLTsS02VzIlbRlfD0o4wuM+q8c6AEz/auXZL2IDouKtWObM9ngbTp1rqPXT01l4ZODjXW5zQBmDLZznoE8Cnu5Q8AzMJov2X
tp3xYEpKy66bGwqdzJLIZAmC8+HvKSS8t9MRwO4QcYtwHRLYZdFj6lo5euxF1VAr+mBX004x7iv+13x86LFwdNDwHJw7KrFWaso+UzWHWx0eT1vXhCLR+cC
655Ai27UTF1RUBfB6GEOyQVnzlWRGasXrDDGYsJz5sofDAhOvp6eQmFaRFbwIk75MUD4949MGFUKbwJbv/PXMxz5hv00fxhfPwH6w7nuBAf9+2QhbD9Fy7Kb
6GNfzEYbNluRsIyP06KmzgCUq5Nh11o+4xdQQh3eI2AugkzgFVTQtRoj0T1ZcMgwMqG68ms= egarrido@dev-performance-eg3.procore.prod1
egarrido@dev-performance-eg3 ~]$
```

A second SSH public key is successfully added to GitLab for the performance host dev-performance-eg3.procore.prod1.

The key is enabled for Authentication & Signing with a defined expiration date.

Fingerprints (MD5 and SHA256) confirm the key’s integrity.

This completes secure, passwordless Git access from the performance environment, supporting multi-host automation workflows.

Board - Edward Garrido - Pro-Core-1

egarrido@dev-performance-eg3

gitlab.com/-/user\_settings/ssh\_keys/18394723

User Settings / SSH Keys / egarrido@dev-performance-eg3.procore.prod1

Search settings

SSH Key: egarrido@dev-performance-eg3.procore.prod1

Delete

Key details

Usage type	Created	Last used	Expires
Authentication & Signing	Sep 19, 2025 3:45pm	Never	Sep 19, 2026 12:00am

SSH Key

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCvKht52ZD67qq/FNWTP3fs89TKQyr5XbYyy8Uy70CAK9BoktZm5eRQapUNjSdXkI7dXdffsCQm30mzu2Y0MjfoFJT/iZoV5lt7a62M

Fingerprints

MD5	2f:50:d2:bb:9f:8f:a4:18:2f:39:dc:76:50:00:4d:a4
SHA256	z4jBH6909j7onpIoUmzAh/Xi2VyPmAN9BE1pkZHIwbQ

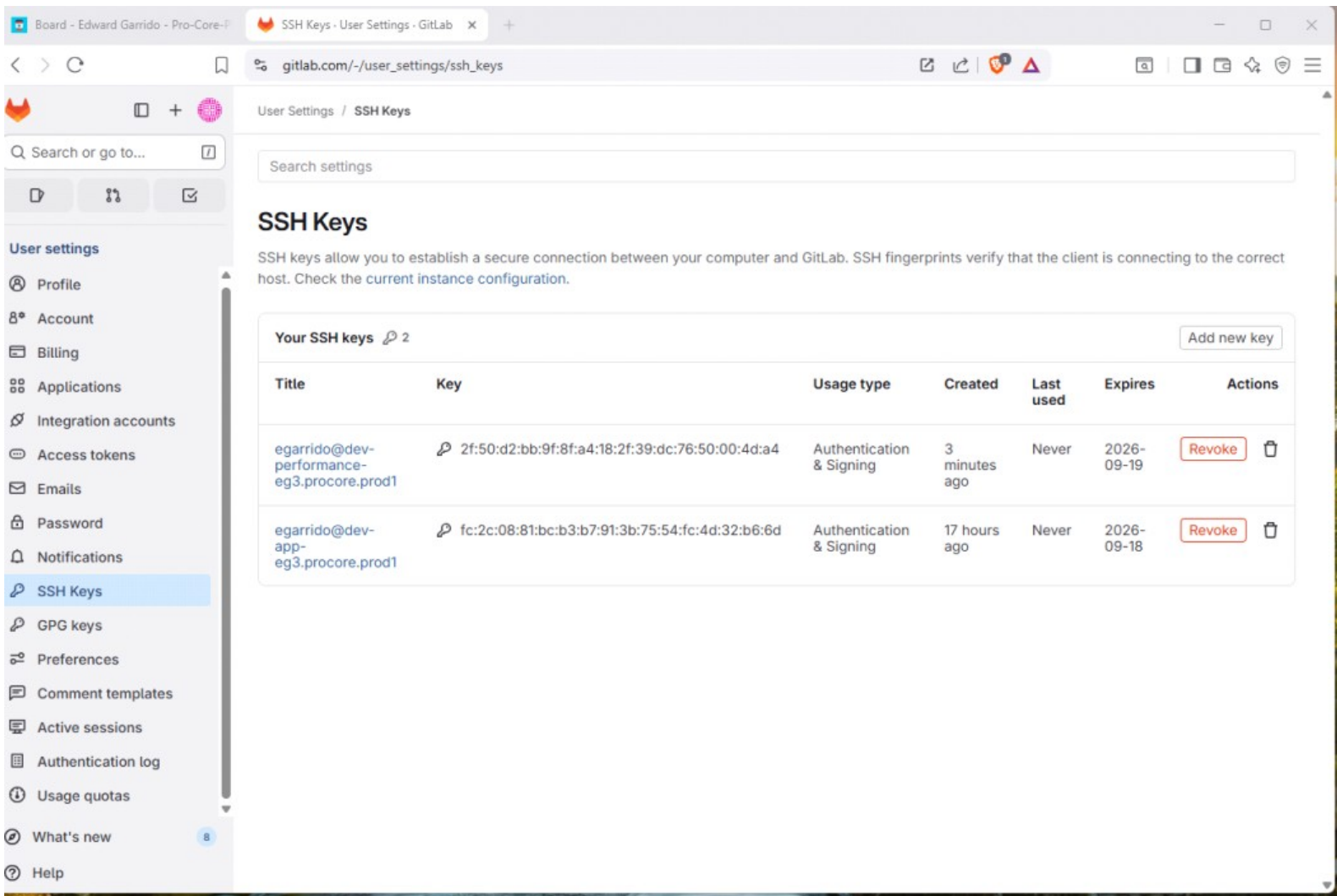


Two SSH public keys are registered in GitLab for separate hosts (dev-app-eg3 and dev-performance-eg3).

Both keys are enabled for Authentication & Signing with defined expiration dates.

Fingerprints confirm key integrity and correct association per host.

GitLab is now configured for secure, passwordless Git access from multiple environments.



Summary (last 15 screenshots):

SSH key pairs were generated on multiple systems to enable secure, passwordless authentication.

Public keys were verified locally and deployed to remote hosts using `ssh-copy-id`.

Successful SSH logins confirmed key-based access between application, performance, and Ansible control nodes.

Ansible inventory files were edited to include development, stage, and performance hosts with proper user and SSH settings.

`/etc/hosts` files were updated to provide consistent hostname resolution across environments.

Network configurations were validated using `ip a`, confirming correct interfaces and static IP assignments.

SSH keys were added to GitLab for both application and performance hosts with authentication and signing enabled.

GitLab SSH connectivity was tested and verified using `ssh -T git@gitlab.com`.

Multiple SSH keys are now managed centrally in GitLab, each associated with a specific host and expiration date.

End-to-end connectivity is confirmed across systems, ensuring readiness for Ansible automation and secure Git operations.