

# **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]---+
| .oooo +*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.

```
egarrido@dev-ansible:~ [stage] stage-web-sl.procure.prod1 [web_servers] stage-web-sl.procure.prod1 [all:vars] ansible_python_interpreter=/usr/bin/python3 [dev_sj1] dev-app-sj1.procure.prod1 dev-performance-sj1.procure.prod1 stage-web-sj1.procure.prod1 [dev-eg] dev-app-eg3.procure.prod1      ansible_host=10.1.31.124 ansible_user=egarrido [dev-mv] dev-app-mv1.procure.prod1      ansible_user=mvann      ansible_ssh_private_key_file=/home/mvann/.ssh/id_rsa dev-performance-mv1.procure.prod1  ansible_user=mvann stage-web-mv1.procure.prod1     ansible_user=mvann [dev-dm] dev-app-dm4.procure.prod1      ansible_user=dmckelvey    ansible_ssh_private_key_file=/home/dmckelvey/.ssh/id_rsa dev-performance-dm4.procure.prod1  ansible_user=dmckelvey [dev-ah5] dev-app-ah5.procure.prod1      ansible_user=ahead dev-performance-ah5.procure.prod1  ansible_user=ahead [dev-stageweb-ah5] stage-web-ah5.procure.prod1     ansible_user=ahead [dev-app-da1] dev-app-da1.procure.prod1     ansible_user=darcila [dev-performance-da1] dev-performance-da1.procure.prod1  ansible_user=darcila
```

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.

```
egarrido@dev-ansible:~      +  x
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6

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

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

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

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

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

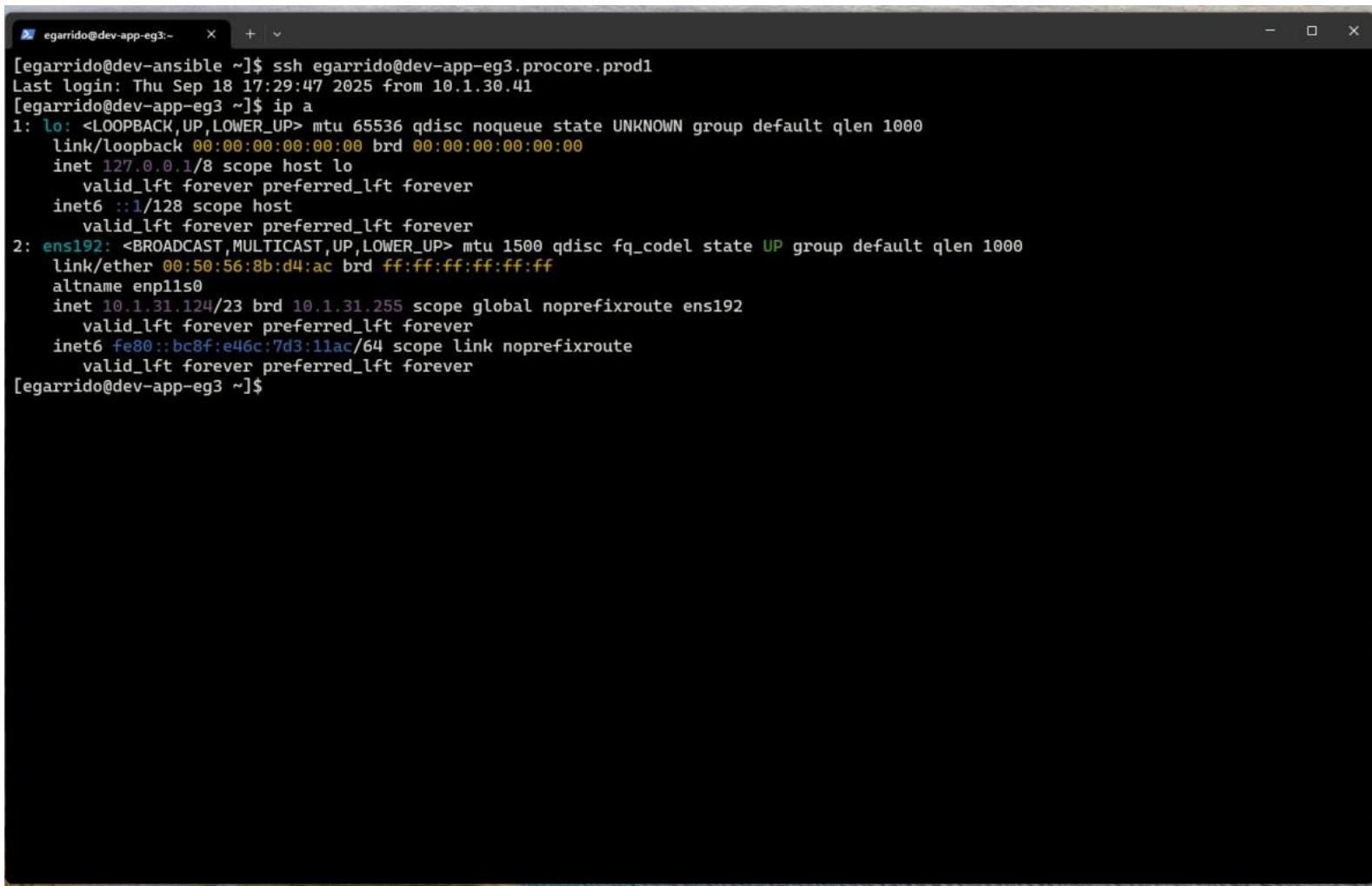
10.1.31.132 dev-app-ds1.procure.prod1
10.1.31.133 dev-performance-ds1.procure.prod1
10.1.31.134 stage-web-ds1.procure.prod1
"/etc/hosts" 246L, 8539B
```

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:~]$ 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.



```
egarrido@dev-app-eg3:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCH7Xc10SuMLlzUegIXR+Opkul08No5c7GnA9dbA1ZmyG8rC+1sADa/QTEtJgk4H+HGIJwY/t2zRlc72iIb5FSL4aanGmWen6
frE+FFIY9bw5ggJ370FJvZbQEeRNlyFv5SQnvVNj1tjUecRi+SiaJ6fzYyDqcqID/QVhiQItUxJWGreJrbTEy1wVHwpdkGt2MCnq5k1eOs9T48BBZ6ZH3+Oc1aGM14e2Bi
Zhwp1POPPzfNWjX4sag/7tJ68Ao0PkSjmQVN4d/x7R9jLHMnzy1TIkLLMzMvgYpvj+9cPS3fKj/CORnErgfzRdocZ7mybQde0ITBXZFOG35kPyZTB5SWnkxNwCbUy0PwIGaskD
d9LQUZTsN95eYYEosqtBQ6vMxL5u5xSp1BqDXmBYg3icCjqYBH5iuSPNv2+nGr0bzW/qWN9WJB0i1C/IKszEOJEjvdRByiiy3PnRp8upJ0pa1dJXIwdWafzPe01Kqhqn7byVgF
eRuSPq00nULJFI0= egarrido@dev-app-eg3.procore.prod1
[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 screenshot shows a web browser window with the URL `gitlab.com/-/user_settings/ssh_keys/18389178`. The page is titled "User Settings / SSH Keys / egarrido@dev-app-eg3.procore.prod1". The main content area displays an "SSH Key" entry for "egarrido@dev-app-eg3.procore.prod1". The key details table includes columns for Usage type (Authentication & Signing), Created (Sep 18, 2025 9:53pm), Last used (Never), and Expires (Sep 18, 2026 12:00am). The SSH Key itself is listed as:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCH7Xc10SuMLzUegIXR+OpruL08No5c7GnA9dbA1ZmyG8rC+1sADa/QTEtJgk4H+HGIJwY/t2zRLC72iIb5
```

Below the key details, there is a "Fingerprints" section showing MD5 and SHA256 values:

	MD5	SHA256
MD5	fc:2c:08:81:bc:b3:b7:91:3b:75:54:fc:4d:32:b6:6d	90gpwFf13ReBP64v93tfZMRjN4XXGawAOZ3lBuZik24

The left sidebar shows the user settings menu, with "SSH Keys" selected. Other options include Profile, Account, Billing, Applications, Integration accounts, Access tokens, Emails, Password, Notifications, GPG keys, Preferences, Comment templates, Active sessions, Authentication log, and Usage quotas. At the bottom, there are links for "What's new" and "Help".

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.

The screenshot shows the 'User Settings / SSH Keys' page on GitLab. The left sidebar has a 'User settings' section with various options like Profile, Account, Billing, Applications, Integration accounts, Access tokens, Emails, Password, Notifications, SSH Keys (which is selected and highlighted in blue), GPG keys, Preferences, Comment templates, Active sessions, Authentication log, and Usage quotas. Below that are 'What's new' and 'Help' sections. The main content area has a header 'SSH Keys' with a sub-instruction: 'SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the current instance configuration.' Below this is a table titled 'Your SSH keys'. It contains one row with the following data:

Title	Key	Usage type	Created	Last used	Expires	Actions
egarrido@dev-app-eg3.procore.prod1	fc:2c:08:81:bc:b3:b7:91:3b:75:54:fc:4d:32:b6:6d	Authentication & Signing	2 minutes ago	Never	2026-09-18	<a href="#">Revoke</a> <a href="#">Delete</a>

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:eUXGGm1YGsMAS7vkcx6J0Jd0GHPem5gQp4taiCfCLB8.
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 AAAAB3NzaC1yc2EAAAQABAAQgQCH7Xc10SuMLlzUegIXR+Oprul08No5c7GnA9dbA1ZmyG8rC+1sADa/QTEtJgk4H+HGIJwY/t2zRlc72iIb5FSL4aanGmWen6
frE+FFIY9bw5qgJ370FJvZbQEQQ6eRNlyFv55QnvVNjiTjUecRi+SiaJ6fzYyDqcqID/QVhiQItUxJWGreJrbTEy1wVHwpdkGt2MCnq5k1e0s9t488BZ6ZH3+0c1aGM14e2Bi
Zhwp1POPPzfNWlHX4sag/7tJ68AOoPkSjmQVN4d/x7R9jLHMnzy1TIkLLMzMvgYpvj+9cPS3fkj/CORnErgfzRdocZ7mybQde0ITBXZF0G35kPyZTB5SWnkxNwCbUy0PwIGaskD
d9LQUZTsN95eYYEosqtBQ6vMxL5u5xSp1BqDXmBYg3icCjqYBH5iuSPNv2+nGrObzw/qWN9WJB0i1C/IKszEOJEjvdRByiyy3PnP8upJOpal1dJXIwdWafzPe01Kqhqn7byVgF
eRuSPq00nULJFI0= egarrido@dev-app-eg3.procure.prod1
[egarrido@dev-app-eg3 ~]$ ssh -T git@gitlab.com
[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:eUXGGm1YGsMAS7vkcx6J0Jd0GHPem5gQp4taiCfCLB8.
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

```
[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.procure.prod1
10.1.31.135  dev-performance-eg3.procure.prod1
10.1.30.41   dev-ansible.procure.prod1 dev-ansible

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

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

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

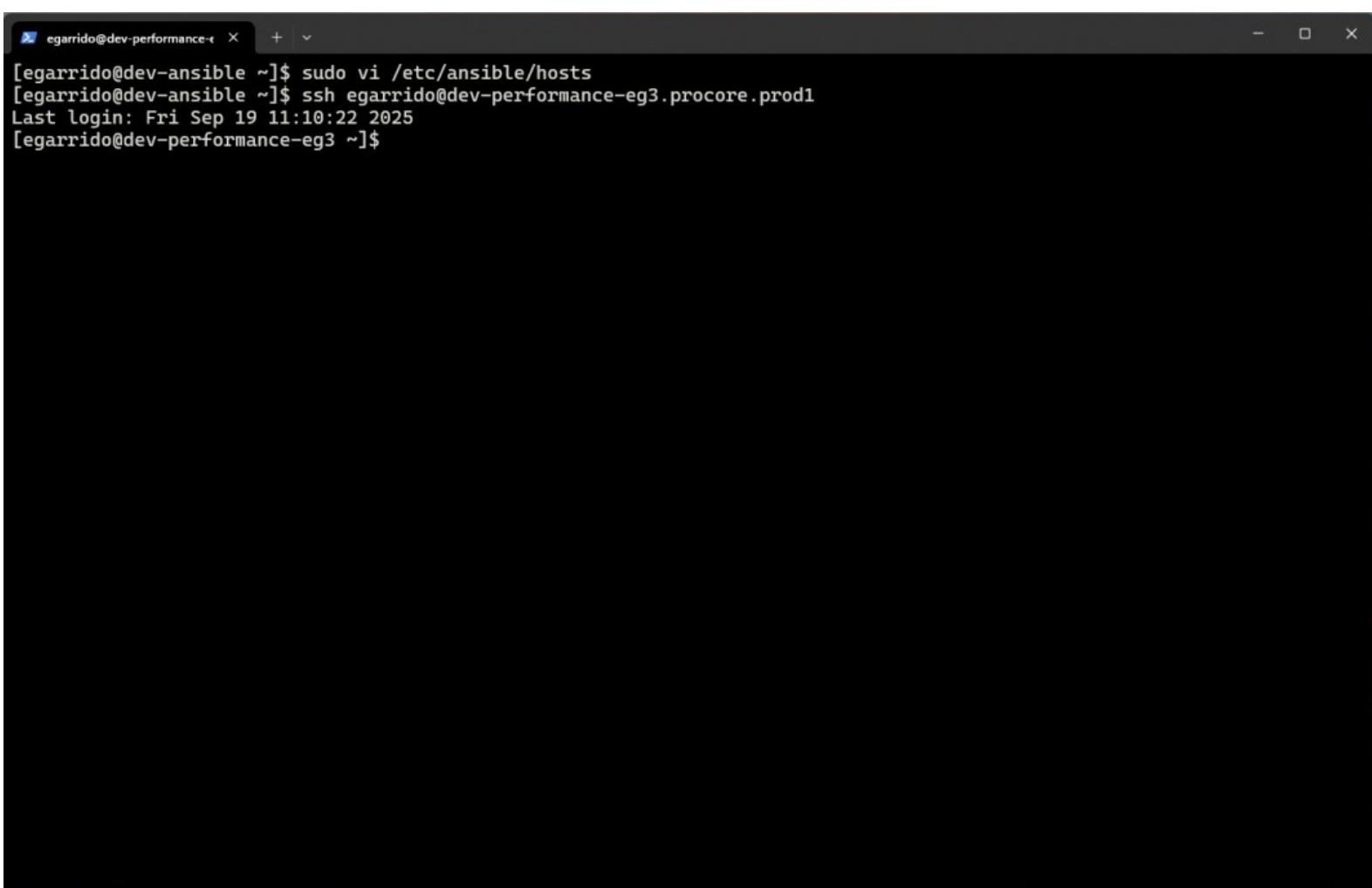
10.1.30.115  dev-app-hb.procure.prod1 dev-app-hb
10.1.30.116  dev-performance-hb.procure.prod1 dev-perform-hb
10.1.30.117  stage-web-hb.procure.prod1 stage-web-hb
```

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 screenshot of a terminal window titled "egarrido@dev-performance-eg3". The window shows a command-line session:

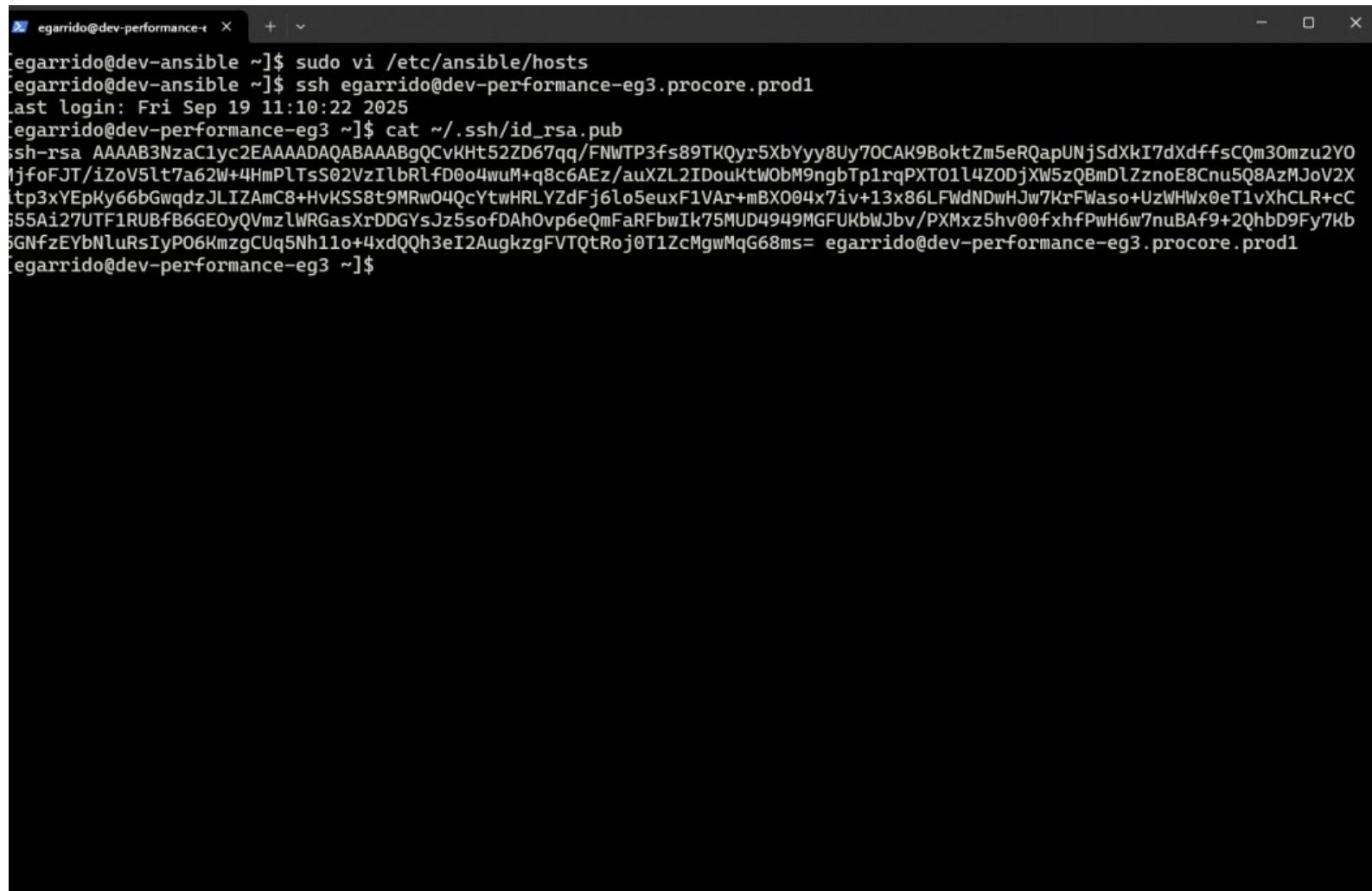
```
[egarrido@dev-ansible ~]$ sudo vi /etc/ansible/hosts
[egarrido@dev-ansible ~]$ ssh egarrido@dev-performance-eg3.procore.prod1
Last login: Fri Sep 19 11:10:22 2025
[egarrido@dev-performance-eg3 ~]$
```

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 screenshot of a terminal window titled "egarrido@dev-performance-eg3". The window shows a command-line session where the user has run several commands to verify SSH access and key exchange. The session starts with "sudo vi /etc/ansible/hosts", followed by "ssh egarrido@dev-performance-eg3.procore.prod1", which displays the last login information ("Fri Sep 19 11:10:22 2025"). The user then runs "cat ~/.ssh/id\_rsa.pub" to view the public SSH key, which is a long string of characters starting with "ssh-rsa AAAAB3NzaC1yc2EAAAQABAA...". The terminal window has a standard black background with white text and a dark gray border.

```
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 AAAAB3NzaC1yc2EAAAQABAAABgQCvKHt52ZD67qq/FNwTP3fs89TKQyr5XbYyy8Uy70CAK9BoktZm5eRQapUNjSdXkI7dXdffsCQm30mzu2Y0
ijfoFJT/iZoV5lt7a62W+4HmPLTsS02VzIlbRlfD0o4wuM+q8c6AEz/auXZL2IDouktWobM9ngbTp1rqPXT01l4Z0DjXW5zQBmDlZznoE8Cnu5Q8AzMJoV2X
itp3xYEpKy66bGwdzJLIZAmC8+HvKSS8t9MRwO4QcYtwHRLYZdFj6Lo5euxF1VAr+mBX004x7iv+13x86LFwdNDwHJw7KrFWaso+UzWHwX0eT1vXhCLR+cC
55Ai27UTF1RUBfB6GE0yQVmzlWRGasXrDDGYsJz5sofDAhOvp6eQmFaRFbwIk75MUD4949MGFUKbWJbv/PXMxz5hv00fxhfPwH6w7nuBAF9+2QhbD9Fy7kb
5GNFzEYbNluRsIyPO6KmzgCUq5Nh11o+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.

The screenshot shows the GitLab User Settings - SSH Keys page. The key 'egarrido@dev-performance-eg3.procore.prod1' is listed. Key details include:

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

The SSH Key section displays the public key:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCVKht52ZD67qq/FNWTP3fs89TKQyr5XbYyy8Uy70CAK9BoktZm5eRQapUNjSdXkI7dXdffsCQm30mzu2YOMjfoFJT/iZoV5lt7a62W
```

The Fingerprints section lists:

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.

The screenshot shows the 'User Settings / SSH Keys' page in a web browser. The left sidebar has a 'User settings' section with various options like Profile, Account, Billing, Applications, Integration accounts, Access tokens, Emails, Password, Notifications, SSH Keys (which is highlighted in blue), GPG keys, Preferences, Comment templates, Active sessions, Authentication log, Usage quotas, What's new, and Help. The main content area has a search bar and a table titled 'Your SSH keys'. The table has columns: Title, Key, Usage type, Created, Last used, Expires, and Actions. There are two entries:

Title	Key	Usage type	Created	Last used	Expires	Actions
egarrido@dev-performance-eg3.procore.prod1	2f:50:d2:bb:9f:8f:a4:18:2f:39:dc:76:50:00:4d:a4	Authentication & Signing	3 minutes ago	Never	2026-09-19	<button>Revoke</button> <button>Copy</button>
egarrido@dev-app-eg3.procore.prod1	fc:2c:08:81:bc:b3:b7:91:3b:75:54:fc:4d:32:b6:6d	Authentication & Signing	17 hours ago	Never	2026-09-18	<button>Revoke</button> <button>Copy</button>

## 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.