

Installing HashiCorp Boundary

HashiCorp Boundary is an open-source tool for secure access management. This guide provides step-by-step instructions for installing Boundary on a system.

1. Prerequisites

Before you begin, ensure you have the following:

- A supported operating system (Linux, macOS, or Windows).
- Access to a terminal or command prompt.
- Administrative privileges to install software.
- Service Principal Name (WP09_PackerPrincipal)

You can create SPN with command

```
az ad sp create-for-rbac -n "WP09_PackerPrincipal" --role="Contributor" --scopes="/subscriptions/<your-subscription>"
```

2. Provision new Azure VM

Follow the Azure portal instructions to create a new virtual machine. Ensure that the VM meets the system requirements for running Boundary.

3. Install and Configure PostgreSQL

3.1 Install PostgreSQL

Run the following command in your terminal:

```
sudo apt install postgresql -y
```

```

Setting up libtypes-serialiser-perl (1.61-1) ...
Setting up libjibson-perl (4.04000-1) ...
Setting up sysstat (12.5.2-2ubuntu0.2) ...

Creating config file /etc/default/sysstat with new version
update-alternatives: using /usr/bin/sar.sysstat to provide /usr/bin/sar (sar) in auto mode
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-collect.timer → /lib/systemd/system/sysstat-collect.timer.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-summary.timer → /lib/systemd/system/sysstat-summary.timer.
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /lib/systemd/system/sysstat.service.
Setting up libjibson-xs-perl (4.030-1build3) ...
Setting up postgresql-common (238) ...
Adding user postgres to group ssl-cert

Creating config file /etc/postgresql-common/createcluster.conf with new version
Building PostgreSQL dictionaries from installed myspell/hunspell packages...
Removing obsolete dictionary files:
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql.service → /lib/systemd/system/postgresql.service.
Setting up postgresql-14 (14.13-0ubuntu0.22.04.1) ...
Creating new PostgreSQL cluster 14/main ...
/usr/lib/postgresql/14/bin/initdb -D /var/lib/postgresql/14/main --auth-local peer --auth-host scram-sha-256 --no-instructions
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "C.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/14/main ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok
update-alternatives: using /usr/share/postgresql/14/man/man1/postmaster.1.gz to provide /usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) in auto mode
Setting up postgresql (14+238) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```

3.2 Configure PostgreSQL

Open the PostgreSQL command line:

```
sudo -u postgres psql
```

Create a new database and user:

```

CREATE DATABASE boundary;
CREATE USER boundary WITH ENCRYPTED PASSWORD 'hidden_password';
GRANT ALL PRIVILEGES ON DATABASE boundary TO boundary;

```

Quit the postgres \q

```

postgres=# CREATE DATABASE boundary;
CREATE DATABASE
postgres=# CREATE USER boundary WITH ENCRYPTED PASSWORD '*****';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE boundary TO boundary;
GRANT
postgres=# \q
root@hcl-boundary-02:~#

```

3.3 Enable Password Authentication

Edit the PostgreSQL configuration file to allow password authentication:

```
echo "host      all            all            127.0.0.1/32
password" | sudo tee -a /etc/postgresql/14/main/pg_hba.conf
```

```
root@hcl-boundary-02:~# echo "host      all            all            127.0.0.1/32
host      all            all            password" | sudo tee -a /etc/postgresql/14/main/pg_hba.conf
root@hcl-boundary-02:~# less /etc/postgresql/14/main/pg_hba.conf
root@hcl-boundary-02:~# tail -n10 /etc/postgresql/14/main/pg_hba.conf
# IPv4 local connections:
host      all            all            127.0.0.1/32            scram-sha-256
# IPv6 local connections:
host      all            all            ::1/128              scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local    replication  all            peer
host    replication  all            127.0.0.1/32            scram-sha-256
host    replication  all            ::1/128              scram-sha-256
host    all            all            127.0.0.1/32            password
root@hcl-boundary-02:~#
```

Then **Restart** the PostgreSQL service:

```
sudo systemctl restart postgresql
```

4. Install and Configure Apache2

4.1 Install Apache2

Run the following commands:

```
sudo apt install apache2 -y
sudo a2enmod ssl proxy proxy_http
sudo systemctl start apache2
```

```
root@hcl-boundary-02:~# apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil libaprutil-ltdb-sqlite3 libaprutil-ltdb liblua5.3-0 mailcap mime-support
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser ufw bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil libaprutil-ltdb-sqlite3 libaprutil-ltdb liblua5.3-0 mailcap mime-support
0 upgraded, 12 newly installed, 0 to remove and 4 not upgraded.
Need to get 2124 kB of archives.
After this operation, 8458 kB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-Subuntu4.22.04.2 [92.8 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil-ltdb-sqlite3 amd64 1.6.1-Subuntu4.22.04.2 [11.3 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil-ltdb amd64 1.6.1-Subuntu4.22.04.2 [9178 B]
Get:5 http://azure.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1buid1 [148 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.12 [1348 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.12 [165 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.12 [89.1 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu jammy/main amd64 mailcap all 3.70+nmulubuntu [23.8 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu jammy/main amd64 mime-support all 3.66 [3696 B]
Get:11 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.12 [97.9 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu jammy/main amd64 bzip2 amd64 1.0.8-5buid1 [34.8 kB]
Fetched 2124 kB in 5s (458 kB/s)
Selecting previously unselected package libapr1:amd64.
(Reading database ... 65464 files and directories currently installed.)
Preparing to unpack .../00-libapr1_1.7.0-8ubuntu0.22.04.1_amd64.deb ...
Unpacking libapr1:amd64 (1.7.0-8ubuntu0.22.04.1) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../01-libaprutil1_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil-ltdb-sqlite3:amd64.
Preparing to unpack .../02-libaprutil-ltdb-sqlite3_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil-ltdb-sqlite3:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil-ltdb:amd64.
Preparing to unpack .../03-libaprutil-ltdb_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil-ltdb:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package liblua5.3-0:amd64.
Preparing to unpack .../04-liblua5.3-0_5.3.6-1buid1_amd64.deb ...

```

```
root@hcl-boundary-02:~# sudo a2enmod ssl proxy
Considering dependency setenif for ssl:
Module setenif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
Enabling module proxy.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@hcl-boundary-02:~#
```

4.2 Create a Self-Signed Certificate

Create a directory for SSL certificates and generate a self-signed certificate:

```
sudo mkdir /etc/ssl/mycerts
```

Generate a private key with the following command:

```
sudo openssl genrsa -out /etc/ssl/mycerts/private.key 2048
```

Run this command to create a CSR. You'll be prompted to enter some information.

```
sudo openssl req -new -key /etc/ssl/mycerts/private.key -out
/etc/ssl/mycerts/self-signed.csr
```

```
root@hcl-boundary-02:/etc/ssl/mycerts# sudo openssl req -new -key /etc/ssl/mycerts/private.key -out /etc/ssl/mycerts/self-signed.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:VN
State or Province Name (full name) [Some-State]:HN
Locality Name (eg, city) []:Hanoi
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FPT
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:boundary.example.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
-----
```

Use the following command to create the self-signed certificate:

```
sudo openssl x509 -req -days 365 -in /etc/ssl/mycerts/self-signed.csr -
signkey /etc/ssl/mycerts/private.key -out /etc/ssl/mycerts/self-signed.crt
```

Output

```
root@hcl-boundary-02:/etc/ssl/mycerts# sudo openssl x509 -req -days 365 -
in /etc/ssl/mycerts/self-signed.csr -signkey /etc/ssl/mycerts/private.key
-out /etc/ssl/mycerts/self-signed.crt
Certificate request self-signature ok
subject=C = VN, ST = HN, L = Hanoi, O = FPT, CN = boundary.example.com
```

4.3 Create a Dummy SSL Site

Edit your Apache configuration file `/etc/apache2/sites-available/default-ssl.conf` to include the following lines:

```
<IfModule mod_ssl.c>
    <VirtualHost 127.0.0.1:443>
        ServerAdmin webmaster@localhost
        ServerName test.example.com
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on
        SSLProxyEngine on
        SSLProxyVerify none
        SSLProxyCheckPeerCN off
        SSLProxyCheckPeerName off
        SSLCertificateFile      /etc/apache2/ssl/cert.pem
        SSLCertificateKeyFile /etc/apache2/ssl/key.pem
```

```

<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
</Directory>

</VirtualHost>
</IfModule>

```

Enable default ssl website

```
a2ensite default-ssl
```

Reload service

```
systemctl reload apache2
```

```

root@hcl-boundary-02:/etc/ssl/mycerts# cd /etc/apache2/sites-available/
root@hcl-boundary-02:/etc/apache2/sites-available# ls -la
total 20
drwxr-xr-x 2 root root 4096 Sep 22 14:38 .
drwxr-xr-x 8 root root 4096 Sep 22 14:38 ..
-rw-r--r-- 1 root root 1332 Dec 4 2023 000-default.conf
-rw-r--r-- 1 root root 6338 Dec 4 2023 default-ssl.conf
root@hcl-boundary-02:/etc/apache2/sites-available# vi default-ssl.conf
root@hcl-boundary-02:/etc/apache2/sites-available# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@hcl-boundary-02:/etc/apache2/sites-available# systemctl reload apache2
root@hcl-boundary-02:/etc/apache2/sites-available# 
```

Test the dummy site:

```
curl -I -k https://127.0.0.1
```

```

root@cariad-dev-wp09-boundary:~# curl -I -k https://127.0.0.1
HTTP/1.1 200 OK
Date: Mon, 23 Sep 2024 07:24:05 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Mon, 23 Sep 2024 07:08:27 GMT
ETag: "29a1-622c40e350e9b"
Accept-Ranges: bytes
Content-Length: 10871
Vary: Accept-Encoding
Content-Type: text/html

```

4.4 Update /etc/hosts

Add the following lines to your /etc/hosts file:

```
echo "127.0.0.1 localhost test.example.com controller.example.com
worker.example.com boundary.example.com" | sudo tee -a /etc/hosts
```

```
root@hcl-boundary-02:/etc/apache2/sites-available# echo "127.0.0.1 localhost test.example.com controller.example.com worker.example.com boundary.example.com" | sudo tee -a /etc/hosts
127.0.0.1 localhost test.example.com controller.example.com worker.example.com boundary.example.com
root@hcl-boundary-02:/etc/apache2/sites-available# cat /etc/hosts
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::1 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
127.0.0.1 localhost test.example.com controller.example.com worker.example.com boundary.example.com
root@hcl-boundary-02:/etc/apache2/sites-available#
```

5. Install and Configure Boundary

5.1 Install Boundary

Add the HashiCorp GPG Key

```
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
```

Add the HashiCorp Repository

```
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
```

Install Boundary

```
sudo apt update
```

```
root@hcl-boundary-02:/tmp# cd
root@hcl-boundary-02:# curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
root@hcl-boundary-02:# echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
root@hcl-boundary-02:# sudo apt update
Get:1 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Err:1 https://apt.releases.hashicorp.com jammy InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY AA16FCBCA621E701
Get:6 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2061 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1125 kB]
Reading package lists... done
W: GPG error: https://apt.releases.hashicorp.com jammy InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY AA16FCBCA621E701
E: The repository 'https://apt.releases.hashicorp.com jammy InRelease' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
```

Fix GPG Key Errors (if any)

```
sudo chmod 644 /usr/share/keyrings/hashicorp-archive-keyring.gpg
```

```
root@hcl-boundary-02:~# curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
root@hcl-boundary-02:~# echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
root@hcl-boundary-02:~# sudo apt update
Get:1 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Err:1 https://apt.releases.hashicorp.com jammy InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY AA16FCBCA621E701
Get:6 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2061 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1125 kB]
Reading package lists... Done
W: GPG error: https://apt.releases.hashicorp.com jammy InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY AA16FCBCA621E701
E: The repository 'https://apt.releases.hashicorp.com jammy InRelease' is not signed.
N: Updating from one repository only, so don't warn.
N: You might want to enable 'APT::Update::Prerun' for repository creation and user configuration details.
root@hcl-boundary-02:~# curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
root@hcl-boundary-02:~# sudo apt update
Get:1 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://apt.releases.hashicorp.com jammy/main amd64 Packages [150 kB]
Fetched 163 kB in 1s (213 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@hcl-boundary-02:~#
```

Run command to install boundary

```
sudo apt install boundary -y
```

```
root@hcl-boundary-02:~# sudo apt install boundary -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  boundary
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 145 kB of archives.
After this operation, 201 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 boundary amd64 0.17.1-1 [145 kB]
Fetched 145 kB in 2s (50.1 kB/s)
Selecting previously unselected package boundary.
(Reading database ... 2225 files and directories currently installed.)
Preparing to unpack .../boundary_0.17.1-1_amd64.deb ...
Unpacking boundary (0.17.1-1) ...
Setting up boundary (0.17.1-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@hcl-boundary-02:~#
```

5.3 Create KMS Keys

Go to Auzre Portal and create new keyvault named **cariad-dev-wp09-boundary**

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance.

Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Cariad Offer
Resource group *: cariad-dev-wp09-west-us-02
Create new

Instance details
Key vault name *: cariad-dev-wp09-boundary
Region *: West US 2
Pricing tier *: Standard

Recovery options
Soft delete protection will automatically be enabled on this key vault. This feature allows you to recover or permanently delete a key vault and secrets for the duration of the retention period. This protection applies to the key vault and the secrets stored within the key vault.

To enforce a mandatory retention period and prevent the permanent deletion of key vaults or secrets prior to the retention period elapsing, you can turn on purge protection. When purge protection is enabled, secrets cannot be purged by users or by Microsoft.

Soft delete: Enabled

Previous **Next** **Review + create** **Give feedback**

Review + create

Subscription	Cariad Offer
Resource group	cariad-dev-wp09-west-us-02
Key vault name	cariad-dev-wp09-boundary
Region	West US 2
Pricing tier	Standard
Soft-delete	Enabled
Purge protection during retention period	Disabled
Days to retain deleted vaults	90 days

Access configuration

Azure Virtual Machines for deployment	Disabled
Azure Resource Manager for template deployment	Disabled
Azure Disk Encryption for volume encryption	Disabled
Permission model	Azure role-based access control

Networking

Connectivity method	Public endpoint (all networks)
---------------------	--------------------------------

Previous **Next** **Create** **Give feedback**

Grant access for the service principal (SPN) with role Key Vault Crypto Officer

Then create three keys

- Root key
- Recovery key
- Worker-auth key

The screenshot shows the Azure Key Vault interface for the 'cariad-dev-wp09-boundary' vault. The 'Keys' section is selected in the left sidebar. A success message at the top right states: 'The key 'boundary-worker-auth' has been successfully created.' The table below lists the keys:

Name	Status	Expiration date
boundary-worker-auth	✓ Enabled	
boundary-recovery	✓ Enabled	
boundary-root	✓ Enabled	

5.2 Configure Boundary Controller

Prepare TLS certificates: Run following command

Create a directory for SSL certificates and generate a self-signed certificate:

```
sudo mkdir /etc/ssl/boundary
```

Generate a private key with the following command:

```
sudo openssl genrsa -out /etc/ssl/boundary/private.key 2048
```

Run this command to create a CSR. You'll be prompted to enter some information.

```
sudo openssl req -new -key /etc/ssl/boundary/private.key -out /etc/ssl/boundary/boundary.csr
```

Use the following command to create the self-signed certificate:

```
sudo openssl x509 -req -days 365 -in /etc/ssl/boundary/boundary.csr -signkey /etc/ssl/boundary/private.key -out /etc/ssl/boundary/boundary.crt
```

```
root@hcl-boundary-02:~# sudo openssl req -new -key /etc/ssl/boundary/private.key -out /etc/ssl/boundary/boundary.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:VN
State or Province Name (full name) [Some-State]:HN
Locality Name (eg, city) []:Hanoi
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FPT
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:boundary.example.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@hcl-boundary-02:~# sudo openssl x509 -req -days 365 -in /etc/ssl/boundary/self-signed.csr -signkey /etc/ssl/boundary/private.key -out /etc/ssl/boundary/boundary.crt
Can't open "/etc/ssl/boundary/self-signed.csr" for reading, No such file or directory
80EB191C0FD7E0008:error:80000002:system library:BIO_new_file:No such file or directory:../crypto/bio/bss_file.c:67:calling fopen(/etc/ssl/boundary/self-signed.csr, r)
80EB191C0FD7E0008:error:10000080:BIO routines:BIO_new_file:no such file:..:/crypto/bio/bss_file.c:75:
Unable to load certificate request input
root@hcl-boundary-02:~# sudo openssl x509 -req -days 365 -in /etc/ssl/boundary/boundary.csr -signkey /etc/ssl/boundary/private.key -out /etc/ssl/boundary/boundary.crt
Certificate request self-signature ok
subject=C = VN, ST = HN, L = Hanoi, O = FPT, CN = boundary.example.com
root@hcl-boundary-02:~#
```

Output

```
root@hcl-boundary-02:~# sudo openssl x509 -req -days 365 -in
/etc/ssl/boundary/boundary.csr -signkey /etc/ssl/boundary/private.key -out
/etc/ssl/boundary/boundary.crt
Certificate request self-signature ok
subject=C = VN, ST = HN, L = Hanoi, O = FPT, CN = boundary.example.com
```

Edit Configuration File [/etc/boundary.d/boundary.env](#):

```
POSTGRES_CONNECTION_STRING=postgresql://boundary:*****@127.0.0.1:5432
/boundary
```

Reload OS environment

```
source /etc/boundary.d/boundary.env
```

Base Controller Configuration Create the boundary.hcl file as per requirements.

```
# disable memory from being swapped to disk
disable_mlock = true

# API listener configuration block
listener "tcp" {
    # Should be the address of the NIC that the controller server will be
    reached on
    # Use 0.0.0.0 to listen on all interfaces
    address = "127.0.0.1:9200"
    # The purpose of this listener block
    purpose = "api"

    # TLS Configuration
    tls_disable = false
    tls_cert_file = "/etc/ssl/boundary/boundary.crt"
```

```
tls_key_file = "/etc/ssl/boundary/private.key"

# Uncomment to enable CORS for the Admin UI. Be sure to set the allowed
origin(s)
# to appropriate values.
#cors_enabled = true
#cors_allowed_origins = ["https://yourcorp.yourdomain.com",
"serve://boundary"]
}

# Data-plane listener configuration block (used for worker coordination)
listener "tcp" {
    # Should be the IP of the NIC that the worker will connect on
    address = "127.0.0.1:9201"
    # The purpose of this listener
    purpose = "cluster"
}

# Ops listener for operations like health checks for load balancers
listener "tcp" {
    # Should be the address of the interface where your external systems'
    # (eg: Load-Balancer and metrics collectors) will connect on.
    address = "127.0.0.1:9203"
    # The purpose of this listener block
    purpose = "ops"
    tls_disable = false
    tls_cert_file = "/etc/ssl/boundary/boundary.crt"
    tls_key_file = "/etc/ssl/boundary/private.key"
}

# Controller configuration block
controller {
    # This name attr must be unique across all controller instances if
running in HA mode
    name = "boundary-controller-1"
    description = "Boundary controller number one"

    # This is the public hostname or IP where the workers can reach the
    # controller. This should typically be a load balancer address
    public_cluster_addr = "controller.example.com"

    # Enterprise license file, can also be the raw value or env:// value
    #license = "file:///path/to/license/file.hlic"

    # After receiving a shutdown signal, Boundary will wait 10s before
initiating the shutdown process.
    graceful_shutdown_wait_duration = "10s"

    # Database URL for postgres. This is set in boundary.env and
    #consumed via the "env://" notation.
    database {
        url = "env://POSTGRES_CONNECTION_STRING"
    }
}
```

```
}

# Events (logging) configuration. This
# configures logging for ALL events to both
# stderr and a file at /var/log/boundary/controller.log
events {
    audit_enabled      = true
    sysevents_enabled  = true
    observations_enable = true
    sink "stderr" {
        name = "all-events"
        description = "All events sent to stderr"
        event_types = ["*"]
        format = "cloudevents-json"
    }
    sink {
        name = "file-sink"
        description = "All events sent to a file"
        event_types = ["*"]
        format = "cloudevents-json"
        file {
            path = "/var/log/boundary"
            file_name = "controller.log"
        }
        audit_config {
            audit_filter_overrides {
                sensitive = "redact"
                secret     = "redact"
            }
        }
    }
}

# Root KMS Key (managed by AWS KMS in this example)
# Keep in mind that sensitive values are provided via ENV VARS
# in this example, such as access_key and secret_key
#
kms "azurekeyvault" {
    purpose      = "root"
    tenant_id    = "tenant-id"
    client_id    = "client_id"
    client_secret = "client_secret"
    vault_name   = "cariad-kms"
    key_name     = "vault1"
}
kms "azurekeyvault" {
    purpose      = "recovery"
    tenant_id    = "tenant-id"
    client_id    = "client_id"
    client_secret = "client_secret"
    vault_name   = "cariad-kms"
    key_name     = "vault2"
}
kms "azurekeyvault" {
```

```

purpose      = "worker-auth"
tenant_id    = "tenant-id"
client_id     = "client_id"
client_secret = "client_secret"
vault_name   = "cariad-kms"
key_name     = "vault3"
}

```

Please remember to change value of `tenant_id`, `client_id`, `client_secret`, `vault_name`, `key_name`.

5.3 Run the Boundary

Initialize the Database

```
boundary database init --config /etc/boundary.d/boundary.hcl
```

The output look like. You will get the initial credential from output console. Please save them to use later

```

{"id": "JmeJ8cIV7q", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault", "msg": "using plugin", "purpose": "recovery-1", "version": 1}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T03:11:42.195277857Z"}, {"id": "q1xLRP1no", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault.studio", "msg": "waiting for stdio data", "purpose": "recovery-1"}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T03:11:42.207322971Z"}, {"id": "jSGn4QIEvR", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault", "msg": "configuring client automatic mTLS", "purpose": "worker-auth-1"}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T03:11:42.875583202Z"}, {"id": "z52hR9Wt3", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault", "msg": "starting plugin", "path": "/tmp/4212868724/boundary-plugin-kms-azurekeyvault-sm6sb"}, "args": "/tmp/4212868724/boundary-plugin-kms-azurekeyvault-sm6sb"}, "msg": "starting plugin", "path": "/tmp/4212868724/boundary-plugin-kms-azurekeyvault-sm6sb"}, "purpose": "worker-auth-1"}, {"id": "BuAkn0nZy", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault", "msg": "plugin started", "path": "/tmp/4212868724/boundary-plugin-kms-azurekeyvault-sm6sb", "pid": 9361, "purpose": "worker-auth-1"}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T03:11:42.882568151Z"}, {"id": "XMNlqaswm", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault", "msg": "waiting for RPC address", "plugin": "/tmp/4212868724/boundary-plugin-kms-azurekeyvault-sm6sb", "purpose": "worker-auth-1"}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T03:11:42.883282355Z"}, {"id": "uZISV5djeG", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault.boundary-plugin-kms-azurekeyvault-sm6sb", "msg": "configuring server automatic mTLS", "purpose": "worker-auth-1", "timestamp": "2024-09-23T03:11:42.883592295Z"}}, {"id": "HHmBqNI3", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault", "msg": "using plugin", "purpose": "worker-auth-1", "version": 1}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T03:11:42.905013506Z"}, {"id": "066fitReX1", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault.boundary-plugin-kms-azurekeyvault-sm6sb", "address": "/tmp/plugin268523532", "msg": "plugin address", "network": "unix", "purpose": "worker-auth-1", "timestamp": "2024-09-23T03:11:42.904962606Z"}}, {"id": "0WOpvjt43L", "source": "https://hashicorp.com/boundary/hcl-boundary-02/boundary-database-init", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HcLogLoggerAdapter).writeEvent", "data": {"@original-log-level": "none", "@original-log-name": "azurekeyvault.studio", "msg": "waiting for stdio data", "purpose": "worker-auth-1"}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T03:11:42.917486403Z"}, Migration successfully run.
Global-scope KMS keys successfully created.

Initial login role information:
  Name: Login Grants
  Role ID: r_____
Initial authenticated user role information:
  Name: Authenticated User Grants
  Role ID: r_____
Initial auth information:
  Auth Method ID: am_____
  Auth Method Name: Generated global scope initial password auth method
  Login Name: admin
  Password: _____
  Scope ID: global
  User ID: u_____
  User Name: admin

Initial org scope information:
  Name: Generated org scope
  Scope ID: c_____
  Type: org

```

Start Boundary Service

First all first, We need to update File Permissions for `boundary`

```

sudo chown -R boundary:boundary /etc/boundary.d
sudo chown -R boundary:boundary /var/log/boundary
sudo chown -R boundary:boundary /etc/ssl/boundary

```

Then, enable boot-start and start the service

```
sudo systemctl enable boundary
sudo systemctl start boundary
```

Verify the status:

```
sudo systemctl status boundary
```

Output:

```
root@hcl-boundary-02:/etc/apache2/sites-available# systemctl status
boundary
● boundary.service - "HashiCorp Boundary – Identity-based access
management for dynamic infrastructure"
   Loaded: loaded (/lib/systemd/system/boundary.service; enabled; vendor
preset: enabled)
     Active: active (running) since Mon 2024-09-23 03:29:52 UTC; 33min ago
       Docs: https://www.boundaryproject.io/docs
      Main PID: 10134 (boundary)
        Tasks: 47 (limit: 9519)
      Memory: 292.6M
        CPU: 38.152s
       CGroup: /system.slice/boundary.service
                 ├─10134 /usr/bin/boundary server -
config=/etc/boundary.d/boundary.hcl
                 ├─10141 /tmp/3372176301/boundary-plugin-kms-azurekeyvault-
gmMTT
                 ├─10149 /tmp/3012504364/boundary-plugin-kms-azurekeyvault-
wSSDg
                 ├─10158 /tmp/3225556827/boundary-plugin-kms-azurekeyvault-
roA0R
                 ├─10168 /tmp/2276464319/boundary-plugin-aws-bMDcj
                 └─10175 /tmp/3612182203/boundary-plugin-azure-aXead

Sep 23 03:29:56 hcl-boundary-02 boundary[10134]:
{"id":"1kWltL3NT0","source":"https://hashicorp.com/boundary/hcl-boundary-
02/controller","specversion":"1.0","type":"system","data":-
{"version":"v0.1","op":"github.com/hashic>
Sep 23 03:29:56 hcl-boundary-02 boundary[10134]:
{"id":"W3acCR1pHE","source":"https://hashicorp.com/boundary/hcl-boundary-
02/controller","specversion":"1.0","type":"system","data":-
{"version":"v0.1","op":"github.com/hashic>
Sep 23 03:29:56 hcl-boundary-02 boundary[10134]:
 {"id":"mYAlwSax9z","source":"https://hashicorp.com/boundary/hcl-boundary-
02/controller","specversion":"1.0","type":"system","data":-
 {"version":"v0.1","op":"github.com/hashic>
Sep 23 03:29:56 hcl-boundary-02 boundary[10134]:
 {"id":"nAordNphSw","source":"https://hashicorp.com/boundary/hcl-boundary-
02/controller","specversion":"1.0","type":"system","data":-
```

```
{"version":"v0.1","op":"github.com/hashicorp/hcl-boundary-02 boundary [10134]:"}
Sep 23 03:29:56 hcl-boundary-02 boundary [10134]:
{"id":"v8bjYAXfKL","source":"https://hashicorp.com/boundary/hcl-boundary-02/controller","specversion":"1.0","type":"system","data":{}}
{"version":"v0.1","op":"github.com/hashicorp/hcl-boundary-02 boundary [10134]:"}
Sep 23 03:29:56 hcl-boundary-02 boundary [10134]:
{"id":"K3tih9EXZR","source":"https://hashicorp.com/boundary/hcl-boundary-02/controller","specversion":"1.0","type":"system","data":{}}
{"version":"v0.1","op":"github.com/hashicorp/hcl-boundary-02 boundary [10134]:"}
Sep 23 03:29:56 hcl-boundary-02 boundary [10134]:
{"id":"4bWoHYLDLY","source":"https://hashicorp.com/boundary/hcl-boundary-02/controller","specversion":"1.0","type":"system","data":{}}
 {"version":"v0.1","op":"server.RotateRoot>"}
Sep 23 03:56:48 hcl-boundary-02 boundary [10134]:
 {"id":"Fn2WdPWJ01","source":"https://hashicorp.com/boundary/hcl-boundary-02/controller","specversion":"1.0","type":"audit","data":{}}
 {"id":"e_b5AE2jVvVG","version":"v0.1","typ>"}
Sep 23 03:57:30 hcl-boundary-02 boundary [10134]:
 {"id":"rmRZFgKYFQ","source":"https://hashicorp.com/boundary/hcl-boundary-02/controller","specversion":"1.0","type":"audit","data":{}}
 {"id":"e_h821ZA6Mw3","version":"v0.1","typ>"}
Sep 23 03:59:07 hcl-boundary-02 boundary [10134]:
 {"id":"nGsu5tFWS0","source":"https://hashicorp.com/boundary/hcl-boundary-02/controller","specversion":"1.0","type":"audit","data":{}}
 {"id":"e_Hh506rUocN","version":"v0.1","typ>"}
lines 1-26/26 (END)
```

Verify with Boundary API

```
curl -k https://127.0.0.1/v1/auth-methods?scope_id=global
```

```
root@hcl-boundary-02:~# curl -k https://127.0.0.1/v1/auth-methods?scope_id=global | jq .
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload Upload Total Spent   Left Speed
100  566  100  566    0     0  12713      0 --:--:-- --:--:-- 12577
{
  "items": [
    {
      "id": "ampw_Lvx8PUUJsX",
      "scope_id": "global",
      "scope": {
        "id": "global",
        "type": "global",
        "name": "global",
        "description": "Global Scope"
      },
      "name": "Generated global scope initial password auth method",
      "description": "Provides initial administrative and unprivileged authentication into Boundary",
      "type": "password",
      "is_primary": true,
      "authorized_actions": [
        "authenticate"
      ]
    }
  ],
  "response_type": "complete",
  "list_token": "GnNTogSP1PnT2Wp3uHsT5bCsU7x3q1Kb9PZq535MNYYir9A2qrbtL5nGJM9zHiBZRgUSXUUSZP",
  "sort_by": "created_time",
  "sort_dir": "desc",
  "est_item_count": 1
}
```

5.4 Config reverse proxy

Create config file `/etc/apache2/site-available/boundary.conf` with content below to public Boundary Management Web UI

```
<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerName boundary.example.com
        ServerAdmin admin@example.com

        ErrorLog ${APACHE_LOG_DIR}/boundary.example.com_error.log
        CustomLog ${APACHE_LOG_DIR}/boundary.example.com_access.log
combined

        ProxyPass / https://127.0.0.1:9200/
        ProxyRequests Off

        SSLEngine on
        SSLProxyEngine on
        SSLProxyVerify none
        SSLProxyCheckPeerCN off
        SSLProxyCheckPeerName off

        SSLCertificateFile /etc/ssl/boundary/boundary.crt
        SSLCertificateKeyFile /etc/ssl/boundary/private.key

        #<LocationMatch "^/">
        #    Require all denied
        #</LocationMatch>

        #<LocationMatch "^(v1/auth-methods|v1/targets)">
        #    Require all granted
        #</LocationMatch>
    </VirtualHost>
</IfModule>
```

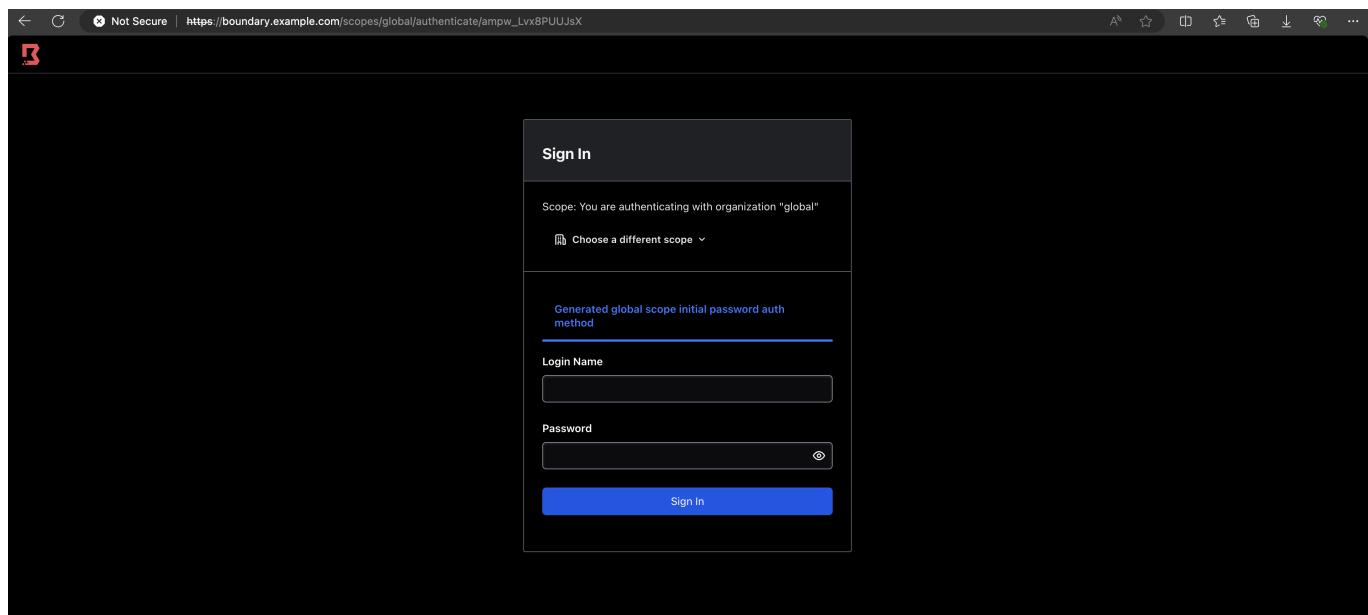
Enable boundary site

```
a2ensite boundary
```

Then reload Apache service to apply the change

```
systemctl reload apache2
```

Check access on browser:



Login with **Initial auth information** that we got from step **5.3**

5.5 Create Boundary Worker

Create worker configuration file as bellow:

```
listener "tcp" {
    address = "0.0.0.0:9202"
    purpose = "proxy"
}

# worker block for configuring the specifics of the
# worker service
worker {
    name = "worker1"
    public_addr = "worker.example.com"
    initial_upstreams = ["controller.example.com:9201"]
    tags {
        type = ["worker", "egress"]
    }
}
kms "azurekeyvault" {
    purpose          = "worker-auth"
    tenant_id       = "tenant_id"
    client_id       = "client_id"
    client_secret   = "client_secret"
    vault_name      = "cariad-kms"
    key_name        = "boundary-worker-auth"
}
```

Please remember to update **kms** information before deploying

Deploy the worker in same VM

boundary server -config=/etc/boundary.d/worker.hcl &

```
root@hcl-boundary-02:~# boundary server -config=/etc/boundary.d/worker.hcl
==> Boundary server configuration:

Azure Environment: AzurePublicCloud
  Azure Key Name: <REDACTED>
  Azure Vault Name: <REDACTED>-02
    Cgo: disabled
  Listener 1: tcp (addr: "0.0.0.0:9202", max_request_duration: "1m30s", purpose: "proxy")
    Log Level: info
      Mlock: supported: true, enabled: true
      Version: Boundary v0.17.1
  Version Sha: 6851256561b50778eaef55144cc6e5be96ce23232
  Worker Auth Current Key Id: fancied-thirsting-purchase-cursor-starless-swimmable-flaky-snowdrift
  Worker Auth Registration Request: [REDACTED]vquLBEW9U6tT9me6RMTRdz4yaMgEmFH2nca8RVRXKFTqbRp5KNd7wZndQRZNBRBKCVVZCzvQDjJC2nYgKKUo693ComrG84UnEg
InCqtk8UuwoJbyPehyUNGfw7KwIsWhsTRx59
  Worker Public Proxy Addr: worker.example.com:9202

==> Boundary server started! Log data will stream in below:
```

[{"id": "MaGQEippDw", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault", "msg": "configuring client automatic mTLS", "purpose": "worker-auth-1"}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.012347112Z"}, {"id": "aJUNwxLiSQ", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "(Worker).startAuthRotationTicking", "data": {"msg": "starting auth rotation ticking"}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.012535212Z"}, {"id": "ps3dCUGIkX", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault", "args": "/tmp/2391419214/boundary-plugin-kms-azurekeyvault-uV4E3", "msg": "starting plugin", "path": "/tmp/2391419214/boundary-plugin-kms-azurekeyvault-uV4E3", "purpose": "worker-auth-1"}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.012609512Z"}, {"id": "IaDrapCa6lB", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault", "msg": "plugin started", "path": "/tmp/2391419214/boundary-plugin-kms-azurekeyvault-uV4E3", "purpose": "worker-auth-1"}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.012689112Z"}, {"id": "fIdk", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault", "msg": "waiting for RPC address", "path": "/tmp/2391419214/boundary-plugin-kms-azurekeyvault-uV4E3", "purpose": "worker-auth-1"}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.01275612Z"}, {"id": "Jinp1k2TSG", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault.boundary-plugin-kms-azurekeyvault-uV4E3", "msg": "configuring server automatic mTLS", "purpose": "worker-auth-1", "timestamp": "2024-09-23T06:37:10.01276512Z"}, {"id": "64xcalNvF", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault", "address": "/tmp/plugin3010100645", "msg": "plugin address", "network": "unix", "purpose": "worker-auth-1", "timestamp": "2024-09-23T06:37:10.012812412Z"}, {"id": "EgBiy5bp3r", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal/event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault", "msg": "using plugin", "purpose": "worker-auth-1", "version": "1"}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.012867712Z"}, {"id": "0tzcErGZt", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "github.com/hashicorp/boundary/internal.event.(+HclogLoggerAdapter).writeEvent", "data": {"original-log-level": "none", "original-log-name": "azurekeyvault", "msg": "waiting for stdio data", "purpose": "worker-auth-1"}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.012926712Z"}, {"id": "KgdXKmpWh35", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "(Worker).upstreamDialerFunc", "data": {"msg": "worker has successfully authenticated"}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.01395790413Z"}, {"id": "PyDEFrdknx", "source": "https://hashicorp.com/boundary/hcl-boundary-02/worker", "specversion": "1.0", "type": "system", "data": {"version": "v0.1", "op": "(Worker).updateAddrs", "data": {"msg": "Upstreams after first status set to: [controller.example.com:9201]"}}, "datacontenttype": "application/cloudevents", "time": "2024-09-23T06:37:10.412564235Z"}]

The Worker will be connected to Boundary Controller automatically

From Web Admin UI, select Worker, you will see the worker is connected

Worker	Tags	Sessions	Release Version	IP Address	ID
worker1	2 tags	0	Boundary v0.17.1	worker.example.com:9202	w_1Jl2PVfrmA

6. Create user, target

6.1 Create new Org

We will create a new **org** with name **P3**



New Org (?)

An org is a type of scope used to organize projects and IAM resources. Orgs are child scopes of the global scope.

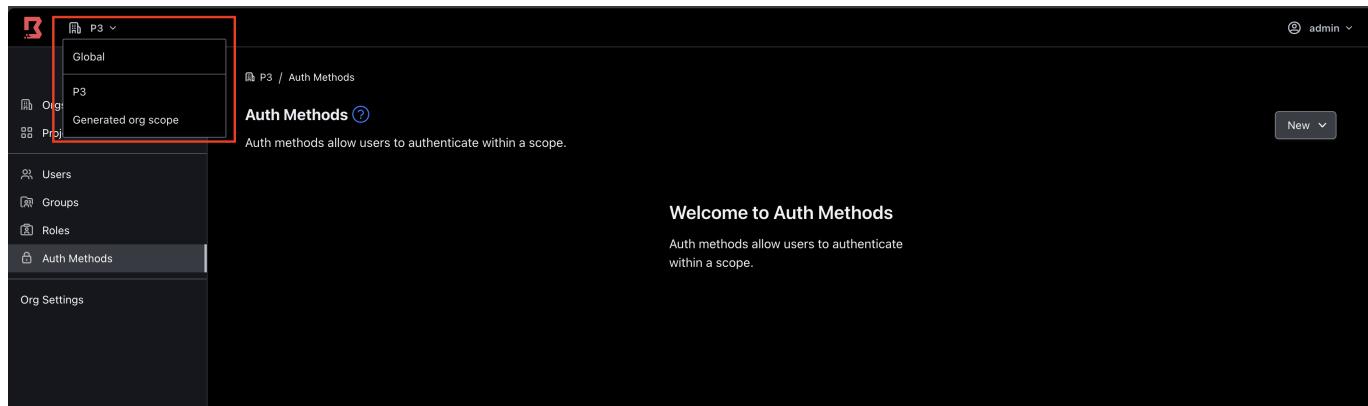
Name (Optional)
Name for identification purposes

Description (Optional)
Description for identification purposes

Save **Cancel**

6.2. Create Authentication method

Go to **P3** org



P3 (?)

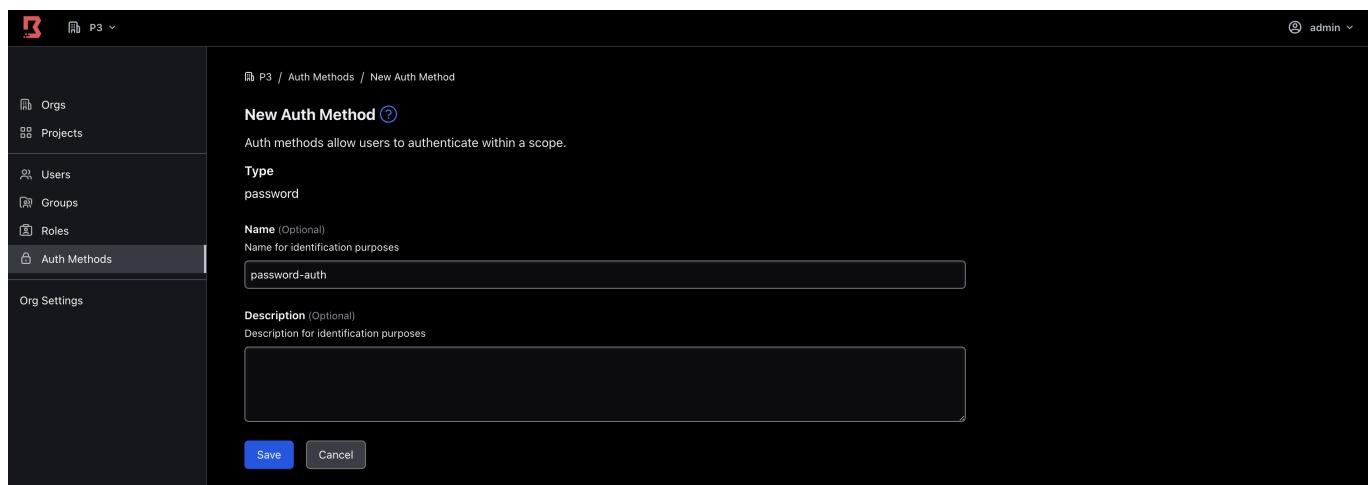
Auth Methods (?)

Auth methods allow users to authenticate within a scope.

Welcome to Auth Methods

Auth methods allow users to authenticate within a scope.

Select **Auth Methods** and create a new **Password Auth Methods**



New Auth Method (?)

Auth methods allow users to authenticate within a scope.

Type
password

Name (Optional)
Name for identification purposes

Description (Optional)
Description for identification purposes

Save **Cancel**

Then make **password-auth** as primary Auth Methods

Auth Method [?](#)
Auth methods allow users to authenticate within a scope.

ampw_VoNCs4SKBE [?](#)

Details Accounts

Type password

Name (Optional) Name for identification purposes
password-auth

Description (Optional) Description for identification purposes

Edit Form

6.3 Create new Project

We will create new project named **Project-01**

New Project [?](#)
A project is a type of scope used to organize resources such as targets and host catalogs.

Name (Optional) Name for identification purposes
Project-01

Description (Optional) Description for identification purposes

Save Cancel

6.4. Create new User

Go to **Users** and create new one

New User [?](#)
Users are entities authorized to access Boundary. Users may be assigned to roles as principals, thus receiving role grants.

Name (Optional) Name for identification purposes
user01

Description (Optional) Description for identification purposes

Save Cancel

6.5. Add new target

Next, we will create new target, that will be pointed to dummy website with domain **test.example.com**

Go to **Project-01**, then create new **Targets**

New Target [?](#)

A target is a logical collection of host sets which may be used to initiate sessions.

Name [Required](#)
Name for identification purposes

Description [\(Optional\)](#)
Description for identification purposes

Type
Generic TCP
Generic TCP supports a broad range of connection types.

Target Address [\(Optional\)](#)
Must be a valid IP address or DNS name. We recommend leaving this blank and using host catalogs and host sets instead if you want to use this target on multiple hosts.

Default Port [Required](#)
The default port on which to connect.

Default Client Port [\(Optional\)](#)
The local proxy port on which to listen by default when a session is started on a client.

6.6. Connect the target

Login in CLI

```
export BOUNDARY_ADDR=https://boundary.example.com
boundary authenticate password -auth-method-id <method-id> --tls-insecure
```

Please get auth method id that we got from step **6.2**

```
* [root@cariad-dev-wp09-boundary boundary] authenticate password -auth-method-id ampw_lsJ95jzesX --tls-insecure
Please enter the login name (it will be hidden):
Please enter the password (it will be hidden):
Authentication information:
Account ID: amcpw_6jNZeBz1h
Auth Method ID: ampw_lsJ95jzesX
Expiration Time: Mon, 30 Sep 2024 14:24:13 UTC
User ID: u_vFMeaq7Ooh
Error opening "pass" keyring: Specified keyring backend not available
The token was not successfully saved to a system keyring. The token is:
[REDACTED]
It must be manually passed in via the BOUNDARY_TOKEN env var or -token flag. Storing the token can also be disabled via -keyring-type=none.
```

Save token to env

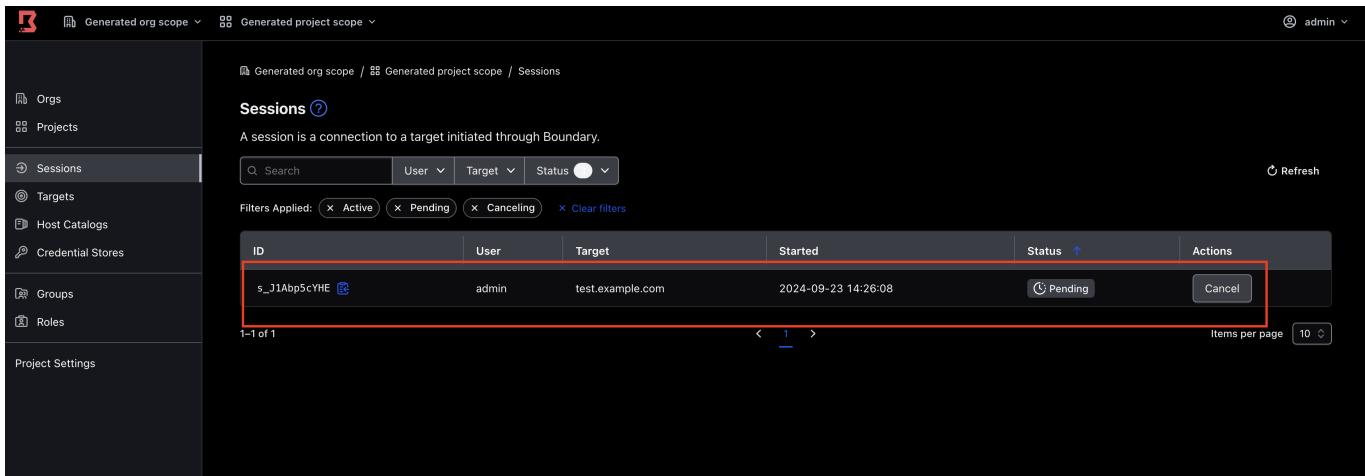
```
export BOUNDARY_TOKEN=<token>
```

Connect to target

```
boundary connect -target-id <target-id> -token env://BOUNDARY_TOKEN --tls-insecure
```

```
root@cariad-dev-wp09-boundary:~# boundary connect -target-id ttcp_Lcp210ISs1 -token env://BOUNDARY_TOKEN -tls-insecure
Proxy listening information:
Address: 127.0.0.1
Connection Limit: 0
Expiration: Mon, 23 Sep 2024 22:26:08 UTC
Port: 41205
Protocol: tcp
Session ID: s_J1Abp5cYHE
```

Check session on Web UI



The screenshot shows the Boundary web interface. On the left is a sidebar with navigation links: Orgs, Projects, Sessions (which is selected and highlighted in blue), Targets, Host Catalogs, Credential Stores, Groups, Roles, and Project Settings. The main content area has a breadcrumb navigation bar: Generated org scope / Generated project scope / Sessions. Below this is a section titled "Sessions" with a subtitle "A session is a connection to a target initiated through Boundary." There are search and filter controls: "Search" (with placeholder "Q"), "User" (dropdown), "Target" (dropdown), "Status" (dropdown set to "Pending"), "Refresh" button, and "Filters Applied": Active, Pending, Canceling, Clear filters. A table lists sessions with columns: ID, User, Target, Started, Status, and Actions. One row is highlighted with a red box: ID s_J1Abp5cYHE, User admin, Target test.example.com, Started 2024-09-23 14:26:08, Status Pending, Actions (Cancel button). At the bottom of the table are pagination controls (1-1 of 1) and "Items per page" dropdown (set to 10).

7. Custom Boundary Client

Download Boundary source code from github [Boundary Source](https://github.com/hashicorp/boundary.git)

```
git clone https://github.com/hashicorp/boundary.git
```

7.1. Hardcode Boundary URL

Open file `/api/client.go`, add content to line 33–34

```
const DEFAULT_CLIENT_URL="https://boundary.example.com"
```

```

6 import (
11   "errors"
12   "fmt"
13   "io"
14   "net"
15   "net/http"
16   "net/url"
17   "os"
18   "path"
19   "strconv"
20   "strings"
21   "sync"
22   "time"
23   "unicode"
24
25   "github.com/hashicorp/boundary/api/recovery"
26   cleanhttp "github.com/hashicorp/go-cleanhttp"
27   wrapping "github.com/hashicorp/go-kms-wrapping/v2"
28   retryablehttp "github.com/hashicorp/go-retryablehttp"
29   rootcerts "github.com/hashicorp/go-rootcerts"
30   "github.com/hashicorp/go-secure-stdlib/parseutil"
31   "golang.org/x/time/rate"
32 )
33
34 const (
35   EnvBoundaryAddr      = "BOUNDARY_ADDR"
36   EnvBoundaryCACert    = "BOUNDARY_CACERT"
37   EnvBoundaryCapath     = "BOUNDARY_CAPATH"
38
39   EnvBoundaryAddr      = "BOUNDARY_ADDR"
40   EnvBoundaryCACert    = "BOUNDARY_CACERT"
41   EnvBoundaryCapath     = "BOUNDARY_CAPATH"
42 )
43
44+ Jeff Mitchell, 4 years ago * Add basic API client (#19) ...
44+ const DEFAULT_CLIENT_URL = "https://boundary.example.com"
45
46 const (
47   EnvBoundaryAddr      = "BOUNDARY_ADDR"
48   EnvBoundaryCACert    = "BOUNDARY_CACERT"
49   EnvBoundaryCapath     = "BOUNDARY_CAPATH"
50
51   EnvBoundaryAddr      = "BOUNDARY_ADDR"
52   EnvBoundaryCACert    = "BOUNDARY_CACERT"
53   EnvBoundaryCapath     = "BOUNDARY_CAPATH"
54 )
55
56 func NewClient() (*Client, error) {
57   c := &Client{
58     Client: http.Client{
59       Transport: &Transport{
60         DialContext: (<function>),
61         RoundTrip:   (<function>),
62       },
63     },
64   }
65
66   if err := c.setAddr(DEFAULT_CLIENT_URL); err != nil {
67     return nil, err
68   }
69
70   return c, nil
71 }
```

Change the code from line 442–445 to

```
if err:=c.setAddr(DEFAULT_CLIENT_URL); err != nil {
  return nil, err
}
```

```

441     if c.Addr != "" {
442         if err := c.setAddr(c.Addr); err != nil {
443             return nil, err
444         }
445     }
446
447     return &Client{
448         config: c,
449     }, nil
450 }
451
452 // Addr returns the current (parsed) address
453 func (c *Client) Addr() string {
454     c.modifyLock.RLock()
455     defer c.modifyLock.RUnlock()
456
457     return c.config.Addr
458 }
459 }
```

```

443
444+ if err := c.setAddr(DEFAULT_CLIENT_URL); err != nil {
445+     return nil, err
446+
447     return &Client{
448         config: c,
449     }, nil
450 }
451
452 // Addr returns the current (parsed) address
453 func (c *Client) Addr() string {
454     c.modifyLock.RLock()
455     defer c.modifyLock.RUnlock()
456
457     return c.config.Addr
458 }
459 }
```

7.2. Modify the code to fetch Auth Method ID automatically

Open the file `internal/cmd/commands/authenticate/funcs.go` and paste the code in from [line 135](#)

```

/ getPasswordAuthMethodId returns the auth method ID from list of auth
methods.
func getPasswordAuthMethodId(ctx context.Context, client
*authmethods.Client, scopeId string) (string, error) {
    authMethodListResult, err := client.List(ctx, scopeId)
    if err != nil {
        return "", err
    }

    for _, m := range authMethodListResult.GetItems() {
        if m.Type == "password" && m.ScopeId == scopeId {
            return m.Id, nil
        }
    }

    return "", fmt.Errorf("Password auth method not found for scope ID:
'%s'. Please set a primary auth method on this scope or pass one
explicitly using an authenticate sub command (see 'boundary authenticate -
h') along with the --auth-method-id flag.", scopeId)
}
```

Open the file `internal/cmd/commands/authenticate/password.go` and change the code as following

```
125  
126     aClient := authmethods.NewClient(client)  
127  
128-    // if auth method ID isn't passed on the CLI, try  
-      looking up the primary auth method ID  
129    if c.FlagAuthMethodId == "" {  
130        // if flag for scope is empty try looking up global  
131        if c.FlagScopeId == "" {  
132            c.FlagScopeId = "global"  
133        }  
134  
135-        pri, err := getPrimaryAuthMethodId(c.context,  
-          aClient, c.FlagScopeId, globals.  
-          PasswordAuthMethodPrefix)  
136        if err != nil {  
137            c.PrintCliError(err)  
138            return base.CommandUserError  
139        }  
140  
141-        c.FlagAuthMethodId = pri  
142    }  
143  
144  
145-    aClient := authmethods.NewClient(client)  
146  
147+    // if auth method ID isn't passed on the CLI, try looking up  
+      the password auth method ID on global scope  
148    if c.FlagAuthMethodId == "" {  
149        // if flag for scope is empty try looking up global  
150        if c.FlagScopeId == "" {  
151            c.FlagScopeId = "global"  
152        }  
153  
154+        passwordAuthMethodId, err := getPasswordAuthMethodId(c.  
+          Context, aClient, c.FlagScopeId)  
155  
156        if err != nil {  
157            c.PrintCliError(err)  
158            return base.CommandUserError  
159        }  
160  
161+        c.FlagAuthMethodId = passwordAuthMethodId  
162    }  
163
```

7.3 Re-build the Boundary Client

Re-build boundary client with command

```
CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -o boundary  
cmd/boundary/main.go
```

Then you will get new **boundary** binary

7.4 Test new Boundary Client

Copy new binary to /tmp

Grant execution permission for binary

```
chmod +x /tmp/boundary
```

Run test command

```
./boundary authenticate password -tls-insecure
```

```
root@hcl-boundary-02:/tmp# ./boundary authenticate password -tls-insecure
Please enter the login name (it will be hidden):
Please enter the password (it will be hidden):

Authentication information:
Account ID: acctpw_8xijl0skbc
Auth Method ID: ampw_sZDgewhidG
Expiration Time: Sun, 29 Sep 2024 15:09:58 UTC
User ID: u_JNXikBSQFH
Error opening "pass" keyring: Specified keyring backend not available
The token was not successfully saved to a system keyring. The token is:
at_caiZ0Pb0eW_s13dJ328myUMaw5SfoMEEXmywNaJrTDCCo4YZ59NNGHwGPDS14NdYWcsAoRYNrctTBSX3Zw6XV32njvQcRvtjLW23Gh52pNNEjuE5xitGvFRA9dUjq92XjNtGm4VDMNxjpsEDsotnbthdsYPA
It must be manually passed in via the BOUNDARY_TOKEN env var or -token flag. Storing the token can also be disabled via -keyring-type=none.
root@hcl-boundary-02:/tmp#
```

As you can see, we can connect to the Boundary without entering the authen method id