Adding secrets to Ansible Secrets

Follow the pattern of the Ansible Secrets installation guide: create the encrypted secret file, then update the Ansible playbook to include it in the deployment.

Adding Passwords

Here is the complete, step-by-step process for adding a new password to Ansible Secrets—for example, `mssql_db_pswd`.

Step 1: Create and Encrypt the New Secret File

First, you'll create the new GPG-encrypted file inside your Ansible project.

```
# Ensure you are in the project root and your venv is
active
cd /opt/ansible_secrets
source venv/bin/activate

# Navigate to the files subdirectory
cd files

# 1. Create the new plaintext password file
printf 'AnotherS3cureP@sswOrd!' > mssql_db_pswd.txt
```

```
# 2. Encrypt it using the SAME GPG passphrase as before.
# You can retrieve it from your notes or directly from the
vault for this command:
GPG_PASSPHRASE=$(ansible-vault view
group_vars/all/vault.yml | grep app_gpg_passphrase | awk
'{print $2}' | tr -d '"')

gpg --batch --yes --symmetric --cipher-algo AES256 \
    --passphrase "$GPG_PASSPHRASE" mssql_db_pswd.txt

# 3. Verify the new .gpg file was created
ls -l mssql_db_pswd.txt.gpg

# 4. Securely delete the plaintext file
shred --remove mssql_db_pswd.txt
cd ..
```

You now have `mssql_db_password.txt.gpg` ready for deployment in your `./files` directory.

Step 2: Update the Ansible Playbook

Edit your playbook, `/opt/ansible_secrets/deploy_secrets.yml`, to add the new filename to the list of files to be deployed.

- **Find this section in `deploy_secrets.yml`:**

```
vars:
  secrets_target_dir: "/opt/credential_store"
  service_user: "service_account"
  secret_access_group: "appsecretaccess"
  encrypted_secret_files:
```

```
    - green_dm_pswd.txt.gpg
    - ldap_ro_pswd.txt.gpg
    - yellow_dm_pswd.txt.gpg
```

- **Add the new filename to the `encrypted_secret_files` list:**

```
vars:
  secrets_target_dir: "/opt/credential_store"
  service_user: "service_account"
  secret_access_group: "appsecretaccess"
  encrypted_secret_files:
    - green_dm_pswd.txt.gpg
    - ldap_ro_pswd.txt.gpg
    - yellow_db_pswd.txt.gpg
    - mssql_db_pswd.txt.gpg  # <-- Add the new file
here
```

Save the playbook file.

Step 3: Re-run the Ansible Playbook

Now, deploy the change.

```
# Ensure you are in the project root and your venv is
active
cd /opt/ansible_secrets
source venv/bin/activate

# Run the playbook
ansible-playbook deploy_secrets.yml
```

Re-running the Playbook

Because Ansible is **idempotent**, it will be very efficient. It will check the state of the existing files and find that they are already correct (they will show up as "ok" in green). It will only perform the action for the new file, copying `mssql_db_pswd.txt.gpg` to `/opt/credential_store` and setting its permissions (this will show up as "changed").

Step 4: Use the New Secret in Your Scripts

Your reusable helper scripts (`get_secret.sh` and `secret_retriever.py`) were designed for this. You can now fetch the new password without changing the helper scripts at all.

- **In a Bash script:**

```
MSSQL_PASS=$(/usr/local/bin/get_secret.sh mssql_db)
# ... use "$MSSQL_PASS"
...
unset MSSQL_PASS
```

- **In a Python script:**

```
import secret_retriever
mssql_pass = secret_retriever.get_password("mssql_db")
# ... use mssql_pass
```