# Ansible Vault

Ansible Advanced

# Agenda

- What is Ansible Vault
- Managing vault passwords

# What is Ansible Vault

Ansible Advanced

# What is Ansible Vault

- Ansible Vault is a feature of Ansible that allows users to encrypt sensitive data within their playbooks or roles
- This solution allow you to have encrypted data at rest
- As soon as your data is decrypted, starts to be in plaintext
- You need to be careful about data usage and possible prints on logs/console

# Main Features

- **Encryption**: Encrypt any structured data file used by Ansible.
- **Decryption**: Decrypt files when running playbooks, ensuring that sensitive data is only exposed when necessary.
- **Password Protection**: Files are encrypted using a password, which must be provided when editing or using the encrypted data.
- **Vault IDs**: Support for multiple passwords and labels (known as Vault IDs) to encrypt different files with different passwords.
- **Integrated with Ansible**: Seamlessly use encrypted data within playbooks, roles, and templates.

# Use Cases

- Sensitive Data: Encrypting passwords, API keys, SSL certificates, or any other sensitive data used in playbooks or roles.
- Version Control: Safely committing encrypted sensitive data to version control systems like Git, ensuring that plain-text secrets aren't exposed.
- Multi-Team Environments: Using different Vault IDs to segregate encrypted data based on teams or environments, allowing for fine-grained access control.
- Dynamic Secrets: Combining with tools like HashiCorp Vault or AWS Secrets Manager to fetch dynamic secrets and use them in playbooks.
- Auditing and Compliance: Meeting regulatory and compliance requirements by ensuring that sensitive data is encrypted at rest.

# How it works

- You create a vault and set a password for that vault
- Then anytime you want to encrypt data (single string or file) you use Ansible Vault and password you set to run AES128 algorithm
- Ansible Vault return a encrypted data that you can use on your playbooks
- When run playbook, you need to send your password on the **`ansible-playbook`** command

# Managing vault passwords

Ansible Advanced

# Managing vault passwords

- Managing your encrypted content is easier if you develop a strategy for managing your vault passwords
- A vault password can be any string you choose. There is no special command to create a vault password.
- However, you need to keep track of your vault passwords
- To develop a strategy for managing vault passwords, start with two questions:
  - Do you want to encrypt all your content with the same password, or use different passwords for different needs?
  - Where do you want to store your password or passwords?

# Single vs Multiple Password

- If you have a small team or few sensitive values, you can use a single password for everything you encrypt with Ansible Vault.
  - Store your vault password securely in a file or a secret manager
- If you have a larger team or many sensitive values, you can use multiple passwords.
  - Depending on your needs, you can use different passwords for different users, different levels of access, for each encrypted file, for each directory, or for each environment.
  - For example, you might have a playbook that includes two vars files, one for the dev environment and one for the production environment, encrypted with two different passwords.

# Using Vault IDs

- If you use multiple vault passwords, you can differentiate one password from another with vault IDs
- You use the vault ID in three ways:
  - Pass it with **`--vault-id`** to the **`ansible-vault`** command when you create encrypted content
  - Include it wherever you store the password for that vault ID
  - Pass it with **`--vault-id`** to the **`ansible-playbook`** command when you run a playbook that uses content you encrypted with that vault ID

# Demo: Use Ansible Vault

Ansible Advanced