1. **Kubernetes Secrets with Sealed Secrets**

sealed secrets

Sealed Secrets is an open-source Kubernetes controller and client-side tool that helps you encrypt sensitive data like **passwords,** API keys, and **tokens so** they can be safely stored in version control systems like Git.

Here's a breakdown of how Sealed Secrets works:

1. **Core Concept**

- **Sealed Secrets** provide a way to **encrypt confidential data** that can only be decrypted by the Sealed Secrets controller running in your Kubernetes cluster
- This allows you to commit **encrypted secrets to** your source code repository without exposing sensitive information

2. **Key Components**

- **Kubeseal**: A client-side CLI tool used to encrypt secrets
- **Sealed Secrets Controller**: A Kubernetes controller that manages the decryption of sealed secrets

3. **Installation Process**

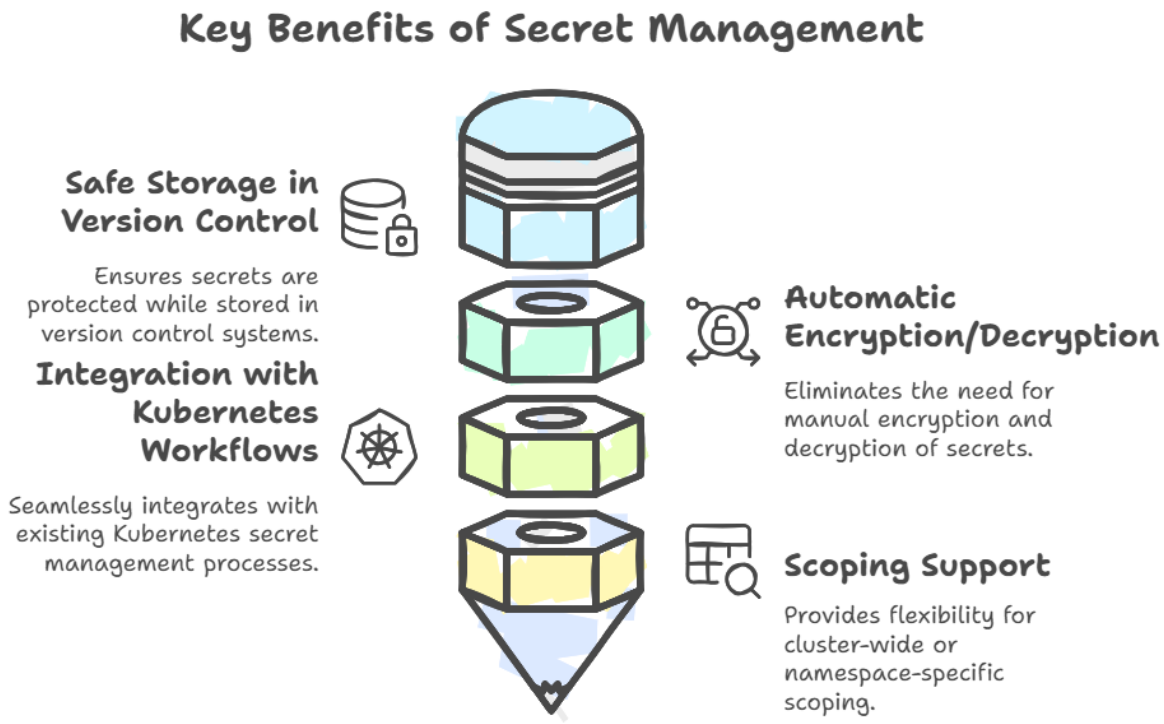
Install Sealed Secrets Controller kubectl apply -f https://github.com/bitnami-labs/sealed-secrets/releases/download/v0.24.2/controller.yaml # Install Kubeseal CLI # For Linux/macOS wget https://github.com/bitnami-labs/sealed-secrets/releases/download/v0.24.2/kubeseal-0.24.2-linux-amd64.tar.gz tar -xvzf kubeseal-0.24.2-linux-amd64.tar.gz sudo install -m 755 kubeseal /usr/local/bin/kubeseal

4. **Usage Example** Here's a step-by-step process to create and use a sealed secret:

Create a regular Kubernetes secret kubectl create secret generic my-secret \ --from-literal=username=admin \ --from-literal=password=mysecretpassword \ --dry-run=client -o yaml > secret.yaml # Seal the secret kubeseal < secret.yaml > sealed-secret.yaml # Apply the sealed secret to the cluster kubectl apply -f sealed-secret.yaml

5. **Key Benefits**

- Secrets can be safely stored in version control
- No need to manually encrypt/decrypt secrets
- Works with existing Kubernetes secret management workflows
- Support for different scoping (cluster-wide or namespace-specific)



6. **Security Considerations**

- Only the Sealed Secrets controller can decrypt the sealed secrets
- Each sealed secret is uniquely encrypted for a specific cluster
- Rotates encryption keys automatically
- Supports multiple encryption algorithms

7. **Alternatives** While Sealed Secrets is powerful, other Kubernetes secret management tools include:

- HashiCorp Vault
- AWS Secrets Manager
- Azure Key Vault
- Google Secret Manager

8. **Best Practices**

- Regularly rotate encryption keys
- Limit access to the Sealed Secrets controller
- Use namespace-scoped sealing when possible
- Avoid committing unencrypted secrets to repositories

Sealed Secrets simplifies the process of managing sensitive information in Kubernetes, allowing you to keep your secrets secure while maintaining a GitOps workflow.