Secrets

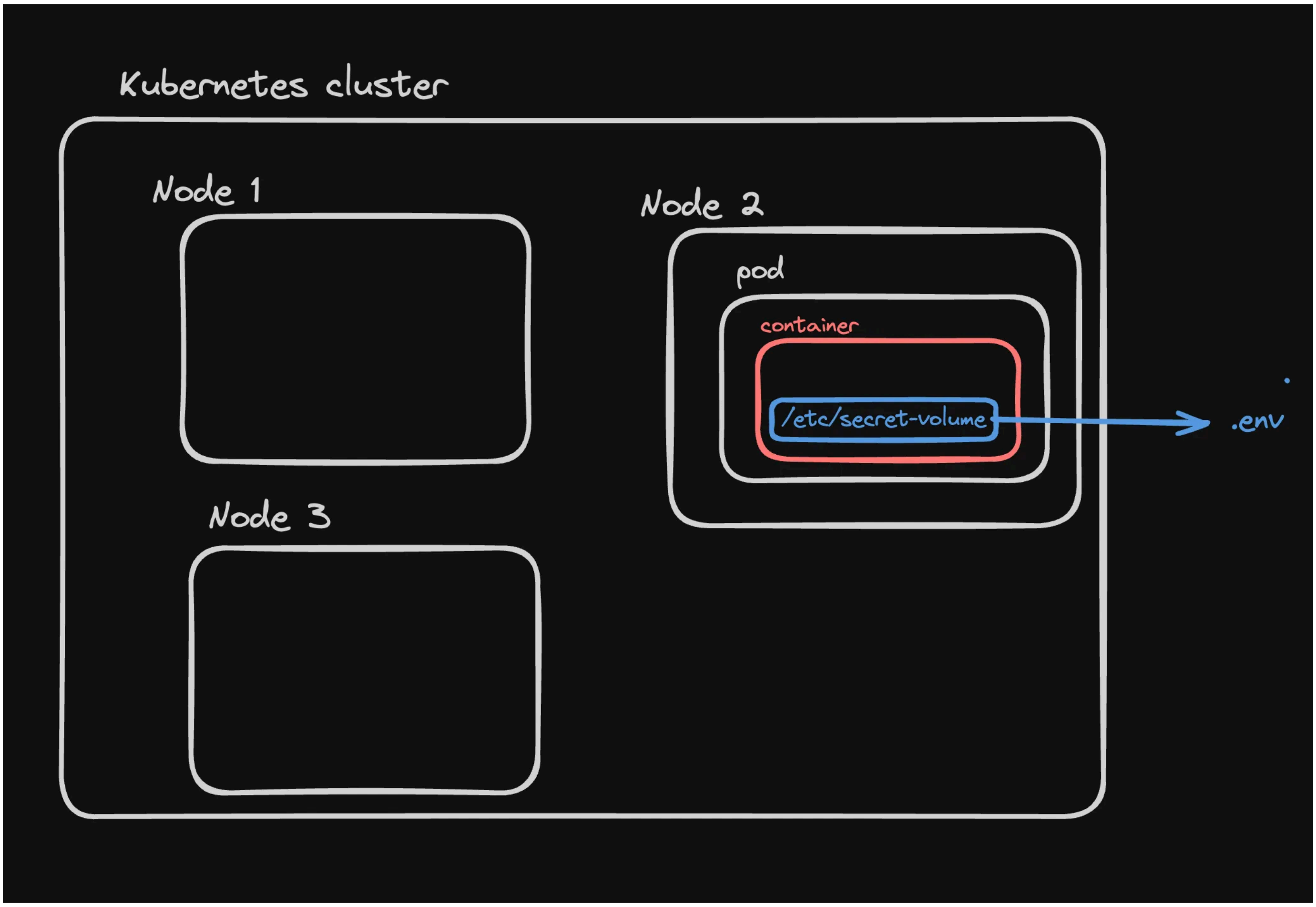
Secrets are also part of the kubernetes v1 api. They let you store `passwords` / `sensitive data` which can then be mounted on to pods as environment variables. Using a Secret means that you don't need to include confidential data in your application code.

Ref - https://kubernetes.io/docs/concepts/configuration/secret/

Using a secret

- Create the manifest with a secret and pod (secret value is base64 encoded) (https://www.base64encode.org/)

```
apiVersion: v1
kind: Secret
metadata:
  name: dotfile-secret
data:
  .env: REFUQUJBU0VfVVJMPSJwb3N0Z3JlczovL3VzZXJuYW1lOnNlY3JldEBsb2Nhb
---
apiVersion: v1
kind: Pod
metadata:
  name: secret-dotfiles-pod
spec:
  containers:
    - name: dotfile-test-container
      image: nginx
      volumeMounts:
        - name: env-file
          readOnly: true
          mountPath: "/etc/secret-volume"
  volumes:
    - name: env-file
```



- Try going to the container and exploring the .env

```
kubectl exec -it secret-dotfiles-pod /bin/bash
cd /etc/secret-volume/
ls
```

Base64 encoding

Whenever you're storing values in a secret, you need to base64 encode them. They can still be decoded , and hence this is not for security purposes. This is more to provide a standard way to store secrets, incase they are binary in nature.

For example, TLS (https) certificates that we'll be storing as secrets eventually can have non ascii characters. Converting them to base64 converts them to ascii characters.

Secrets as env variables

You can also pass in secrets as environment variables to your process (similar to how we did it for configmaps in the last slide)

- Create the secret

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
data:
  username: YWRtaW4=  # base64 encoded 'admin'
  password: cGFzc3dvcmQ=  # base64 encoded 'password'
```

- Create the pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: secret-env-pod
spec:
  containers:
  - name: my-container
    image: busybox
    command: ["/bin/sh", "-c", "echo Username: $USERNAME; echo Password: $PA
    env:
    - name: USERNAME
      valueFrom:
        secretKeyRef:
          name: my-secret
          key: username
    - name: PASSWORD
      valueFrom:
        secretKeyRef:
          name: my-secret
          key: password
```