Better Secrets Management

Justin Mitchel
Day Zero
KubeCon Chicago 2023

Justin Mitchel

- 3x Dad and Husband
- Founder of CodingforEntrepreneurs.com
- Author of Road to Kubernetes
- 21m+ views & 250k+ subs on YouTube
- 800k+ students on Udemy
- Top 25 course on freeCodeCamp

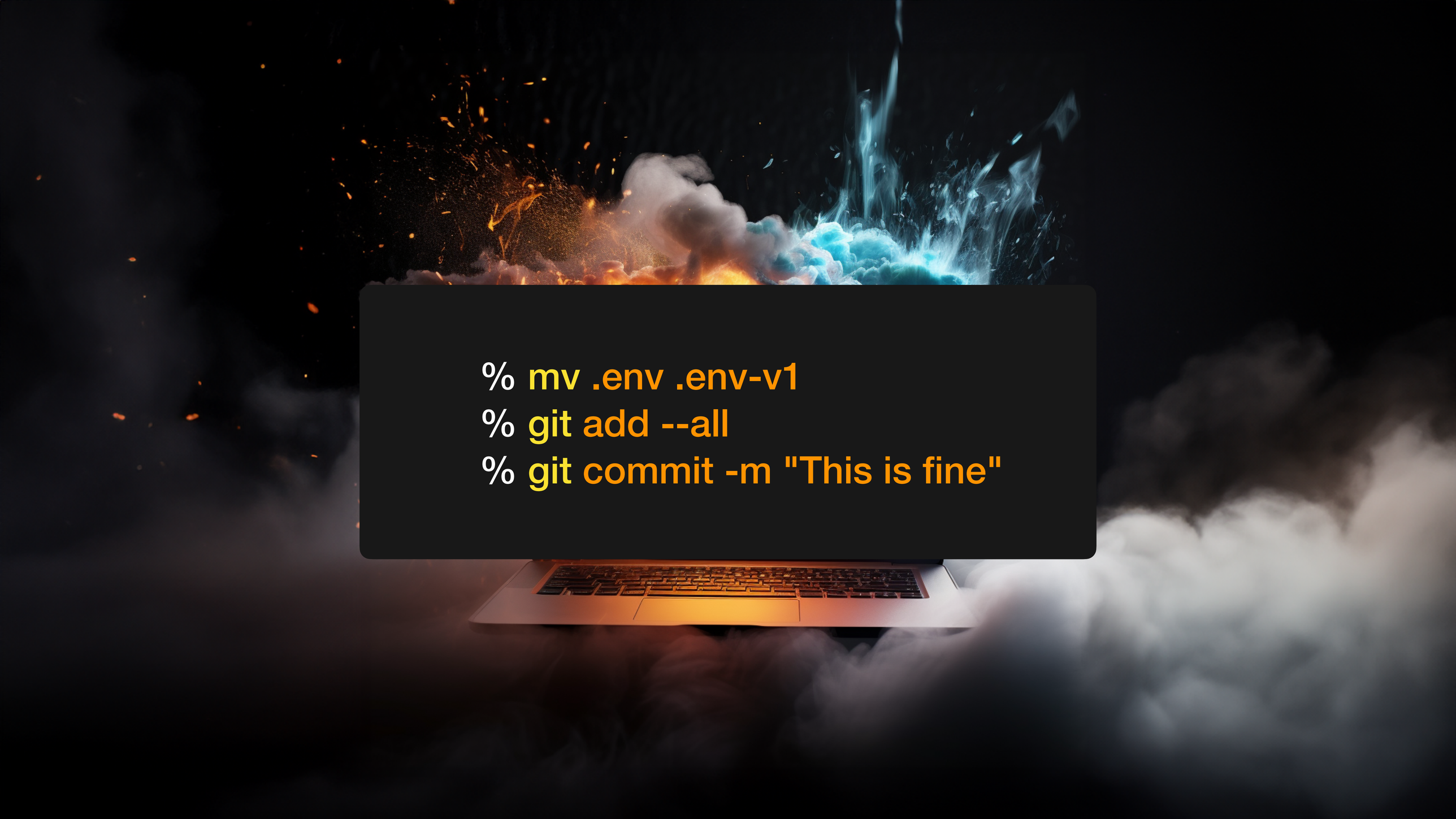
Thank you to Akamai for sponsoring this workshop.

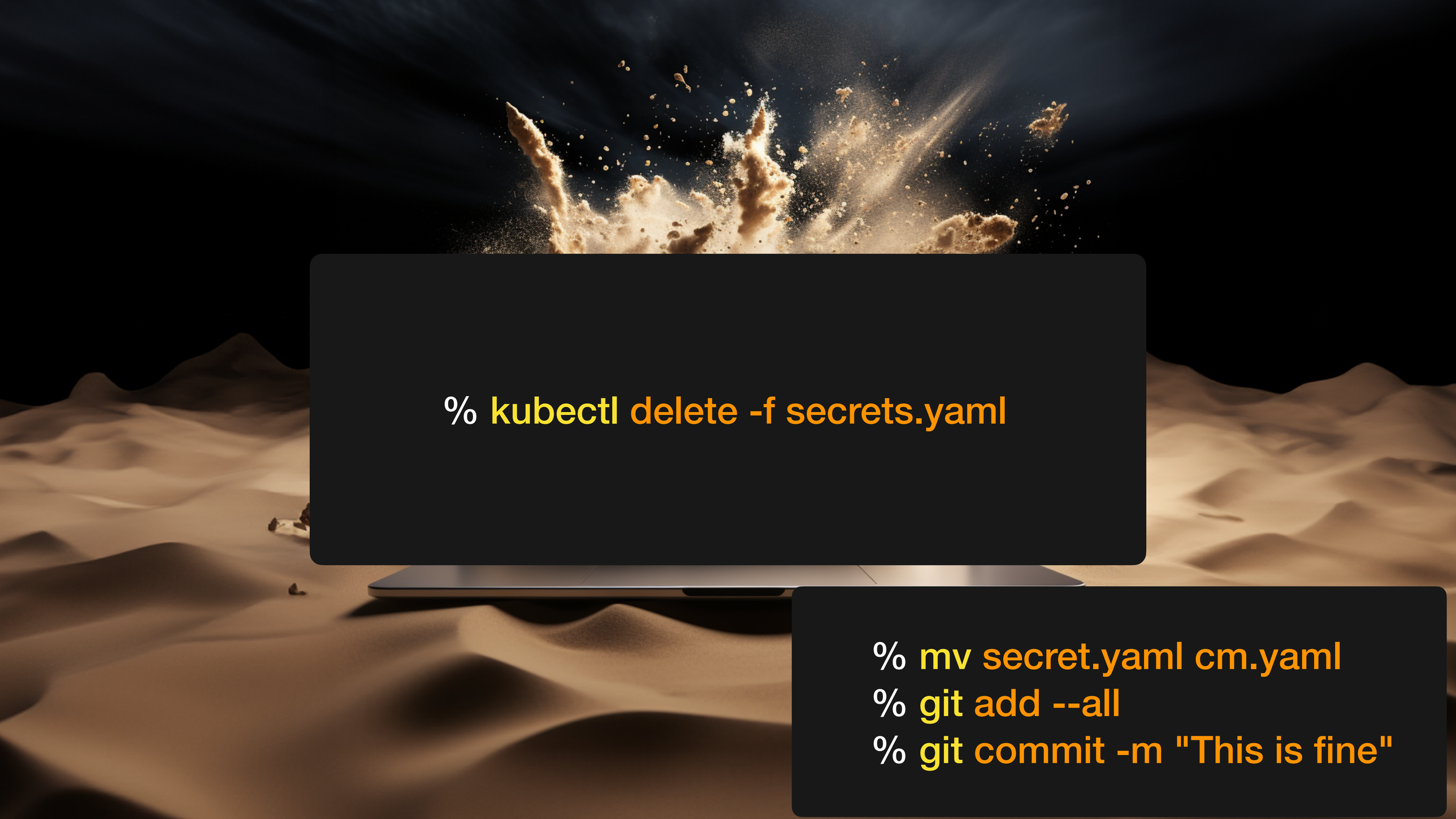
dotenv (.env)

```
API_VERSION=1
API_KEY_SECRET="some secret value"
API_KEY_PUBLIC="some non-secret value"
STAGE="prod"
```

Kubernetes Secrets Resource

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: sample-k8s-secret
  namespace: default
stringData:
  API_VERSION: 1
  API_KEY_SECRET: "some secret value"
  API_KEY_PUBLIC: "some non-secret value"
  STAGE: "prod"
---
apiVersion: v1
kind: Secret
metadata:
  name: sample-k8s-secret-2
  namespace: default
data:
  API_VERSION: MQ==
  API_KEY_SECRET: c29tZSBzZWNyZXQgdmFsdWU=
  API_KEY_PUBLIC: c29tZSBub24tc2VjcmV0IHZhbHVl
  STAGE: cHJvZA==
```


% mv .env .env-v1
% git add --all
% git commit -m "This is fine"


% kubectl delete -f secrets.yaml
% mv secret.yaml cm.yaml
% git add --all
% git commit -m "This is fine"

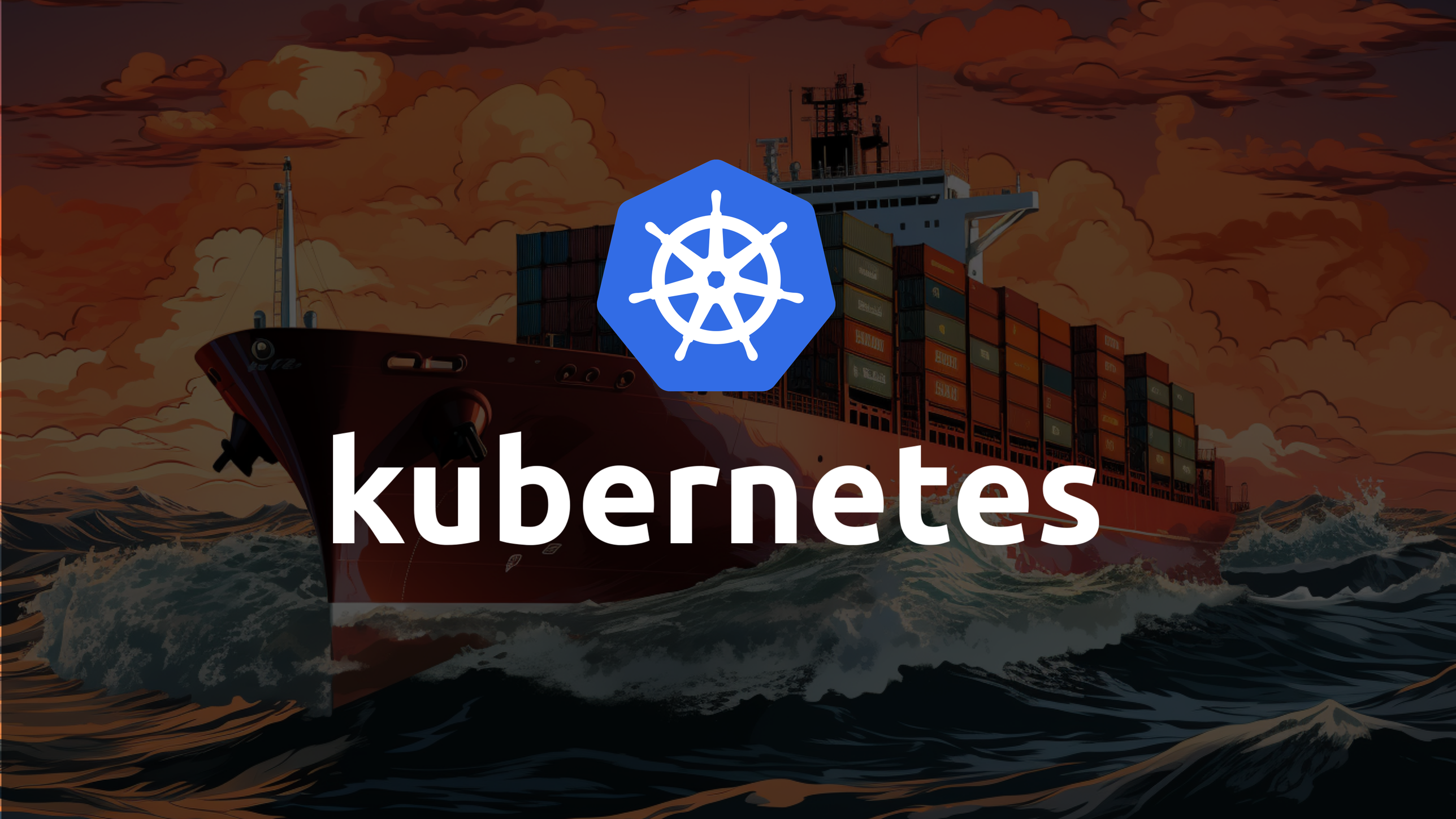
Kubernetes Secrets Resource

dotenv (.env)

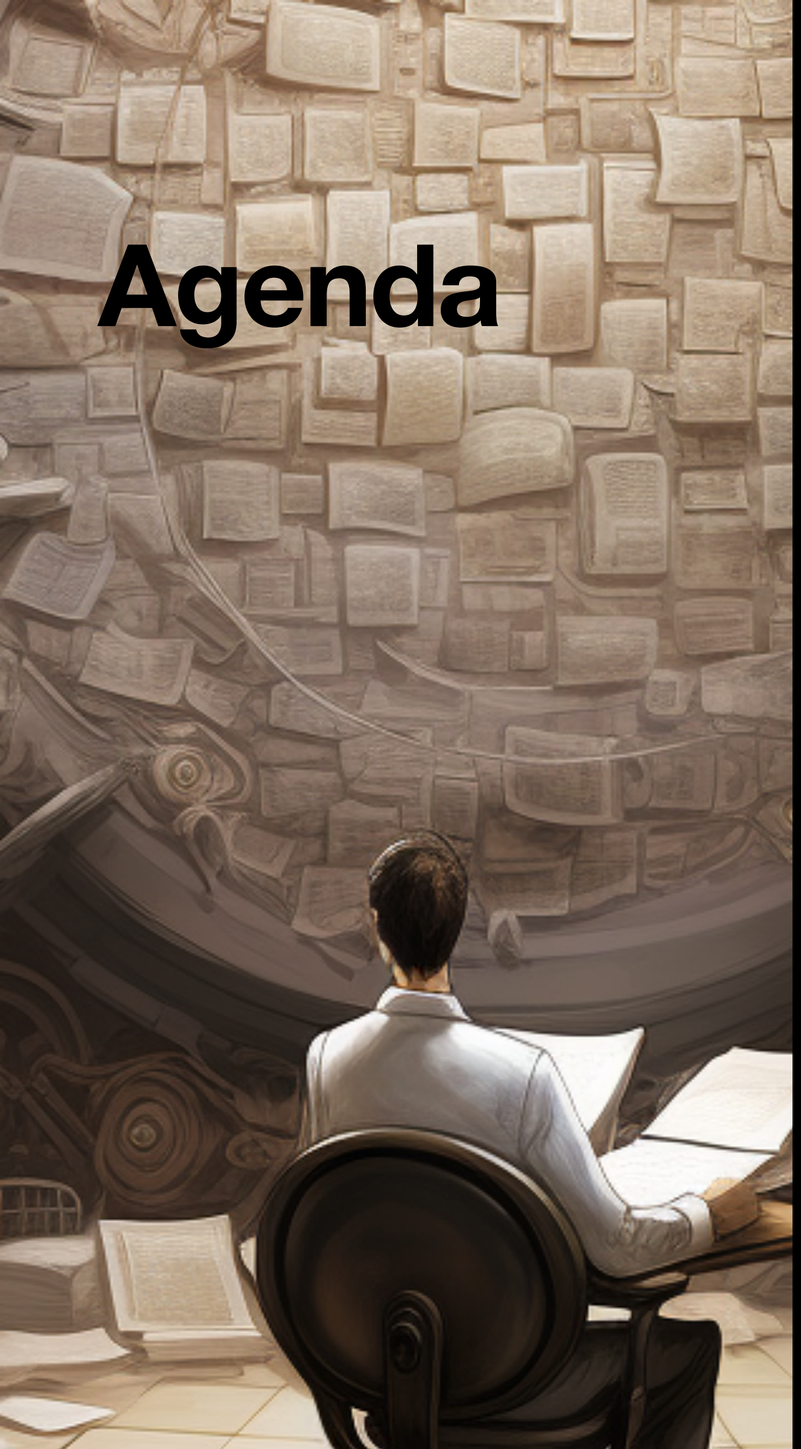
```
API_VERSION=1
API_KEY_SECRET="some secret value"
API_KEY_PUBLIC="some non-secret value"
STAGE="prod"
```

```
apiVersion: v1
kind: Secret
metadata:
  name: sample-k8s-secret
  namespace: default
stringData:
  API_VERSION: 1
  API_KEY_SECRET: "some secret value"
  API_KEY_PUBLIC: "some non-secret value"
  STAGE: "prod"
---
apiVersion: v1
kind: Secret
metadata:
  name: sample-k8s-secret-2
  namespace: default
data:
  API_VERSION: MQ==
  API_KEY_SECRET: c29tZSBzZWNyZXQgdmFsdWU=
  API_KEY_PUBLIC: c29tZSBub24tc2VjcmV0IHZhbHVl
  STAGE: cHJvZA==
```


HashiCorp
Vault


kubernetes


Vault
Secrets
Operator
Vault
Kubernetes



Agenda

- Provision VM on Akamai Connected Cloud
- Install Vault
- See a few more memes
- Vault CLI + UI
- Vault Auth + ACL Policies
- Vault Secrets Engine
- HTTP Methods for Vault API
- Provision Kubernetes Cluster
- Configure Vault Agent Operator on K8s
- Sidecar Injector with Vault

cfe

https://github.com/codingforentrepreneurs/kubecon23-chi

1. Go to cfe.sh/github
2. Navigate to kubecon23-chi
3 Clone or Fork Repo: *kubecon23-chi*

Akamai

linode.com/cfe

For VM + Managed Kubernetes

Local Installs

Vault (optional)	**https://kirr.co/xfdde3**
Kubectl:	**https://kirr.co/qqjzil**
Helm	**https://kirr.co/890eiu**
VS Code (optional):	**https://kirr.co/wai0jb**
Vault Bootstrap Script:	**https://kirr.co/6k05g8**

firewall

1. Login to linode.com
2. Navigate to Firewalls
3. Create new firewall called “vault-firewall”
4. Incoming Rule: Enable SSH
5. Incoming Rule: Custom Port 8200

vm

New Linode instance*

Image: Ubuntu 22.04 LTS

Region: Chicago, IL

Plan: Linode 2gb

Label: vault-node1

Select the ***vault-firewall***

*minimum spec

K8s

New Kubernetes cluster*

Label: vault-k8s-kubecon

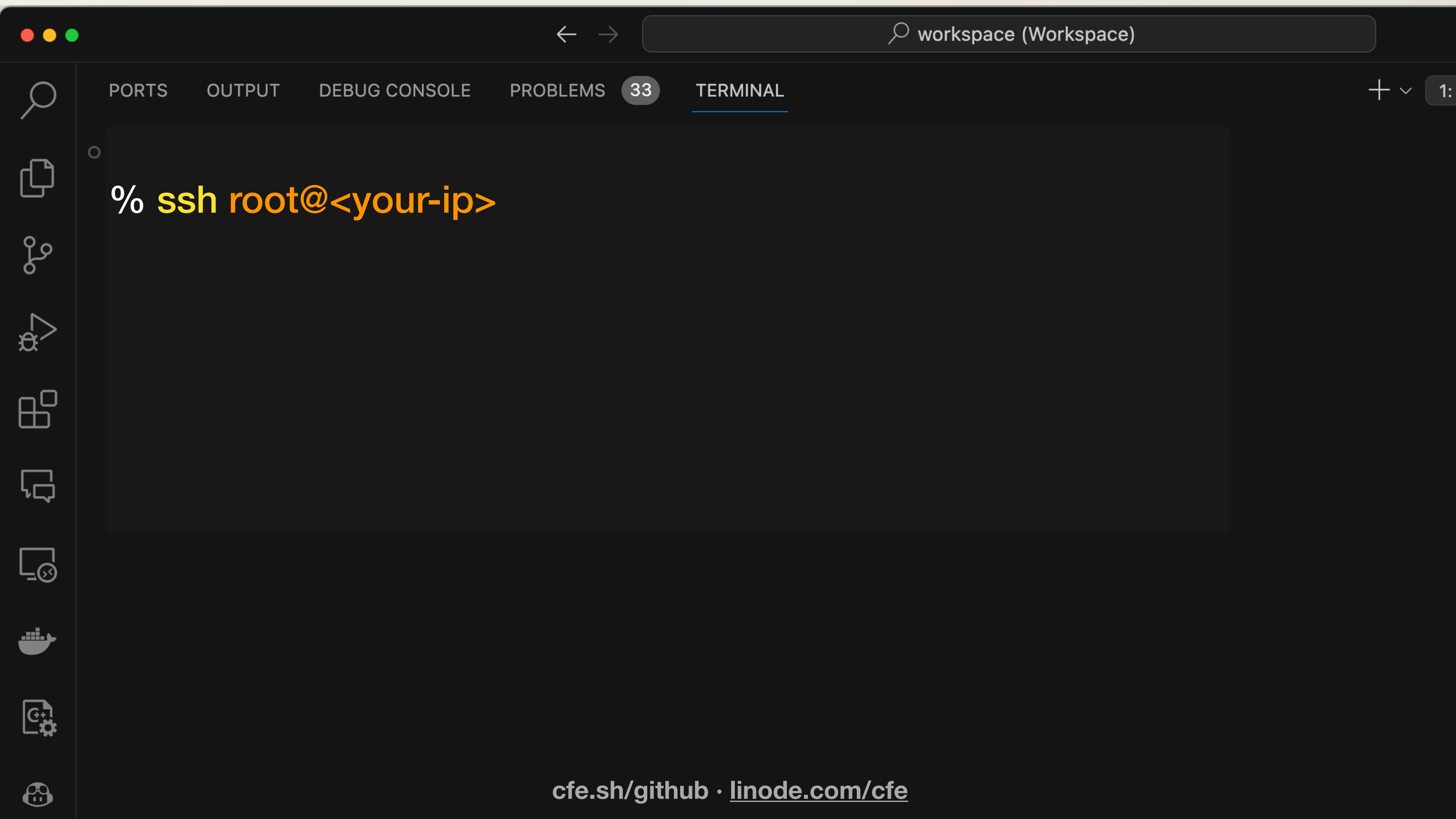
K8s Version: 1.27

HA: n/a

Node Pool Shared CPU: Linode 2gb

*minimum spec

```
VAULT_ADDR="http://<your-ip>:8200"
```


workspace (Workspace)
PORTS
OUTPUT
DEBUG CONSOLE
PROBLEMS 33
TERMINAL
% ssh root@<your-ip>
cfe.sh/github · linode.com/cfe

```
$ curl -fsSL https://kirr.co/f9t9jt -o bootstrap_vault.sh
$ sudo chmod +x bootstrap_vault.sh
$ sudo ./bootstrap_vault.sh
```

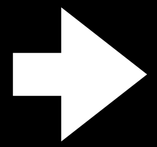
Demo

Provision VM + Install Vault

https://kirr.co/6k05g8

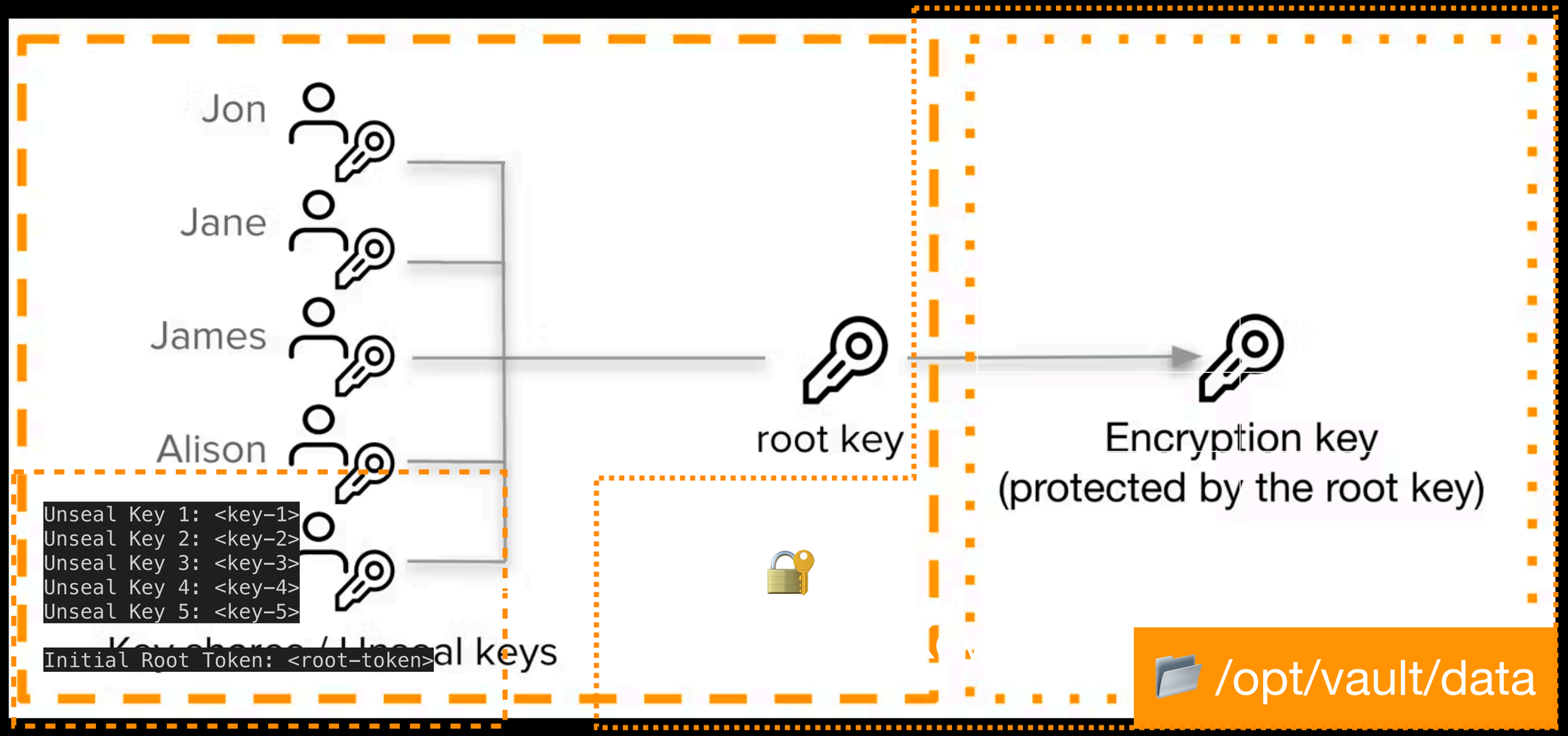
https://gist.github.com/codingforentrepreneurs/22995fd1411969127604fd9fbabf30d4

$ vault operator init



```
Unseal Key 1: <key-1>
Unseal Key 2: <key-2>
Unseal Key 3: <key-3>
Unseal Key 4: <key-4>
Unseal Key 5: <key-5>

Initial Root Token: <root-token>
```

Shamir's secret sharing




SYSTEM ACTIVATION IN PROGRESS
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key

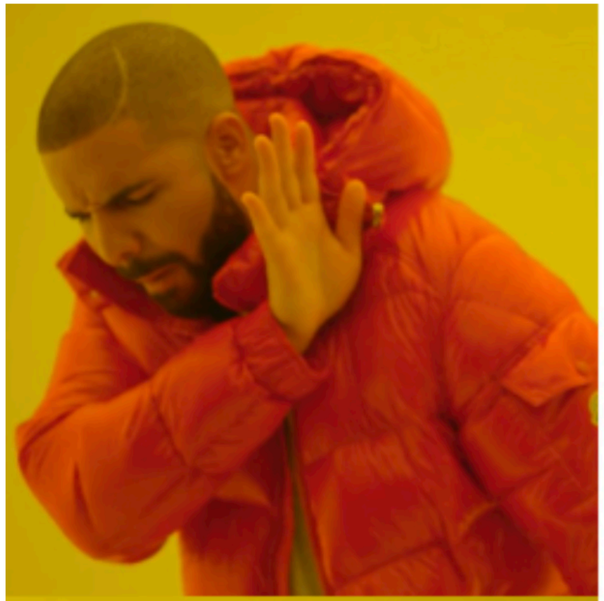
```
$ vault login
```

```
Error authenticating: error looking up token: Get "https://
127.0.0.1:8200/v1/auth/token/lookup-self": http: server gave HTTP
response to HTTPS client
```

```
$ echo 'export VAULT_ADDR="http://127.0.0.1:8200"' \
            >> ~/.bashrc
$ source ~/.bashrc
$ echo $VAULT_ADDR
```

```
$ vault login
```

```
Error authenticating: error looking up
token: Error making API request.

URL: GET http://127.0.0.1:8200/v1/auth/
token/lookup-self
Code: 503. Errors:

* Vault is sealed
```

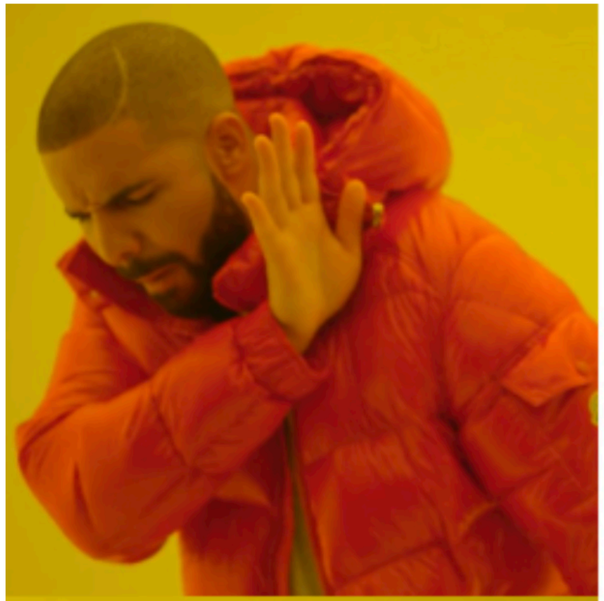


sealed
=
no access

unsealed
=
access

```
$ vault secrets enable -version=2 kv
```

```
Error enabling: Error making API request.

URL: POST http://127.0.0.1:8200/v1/sys/
mounts/kv
Code: 503. Errors:

* Vault is sealed
```



sealed
=
no access

unsealed
=
access

```
$ vault operator init
$ vault operator unseal
$ vault login
```

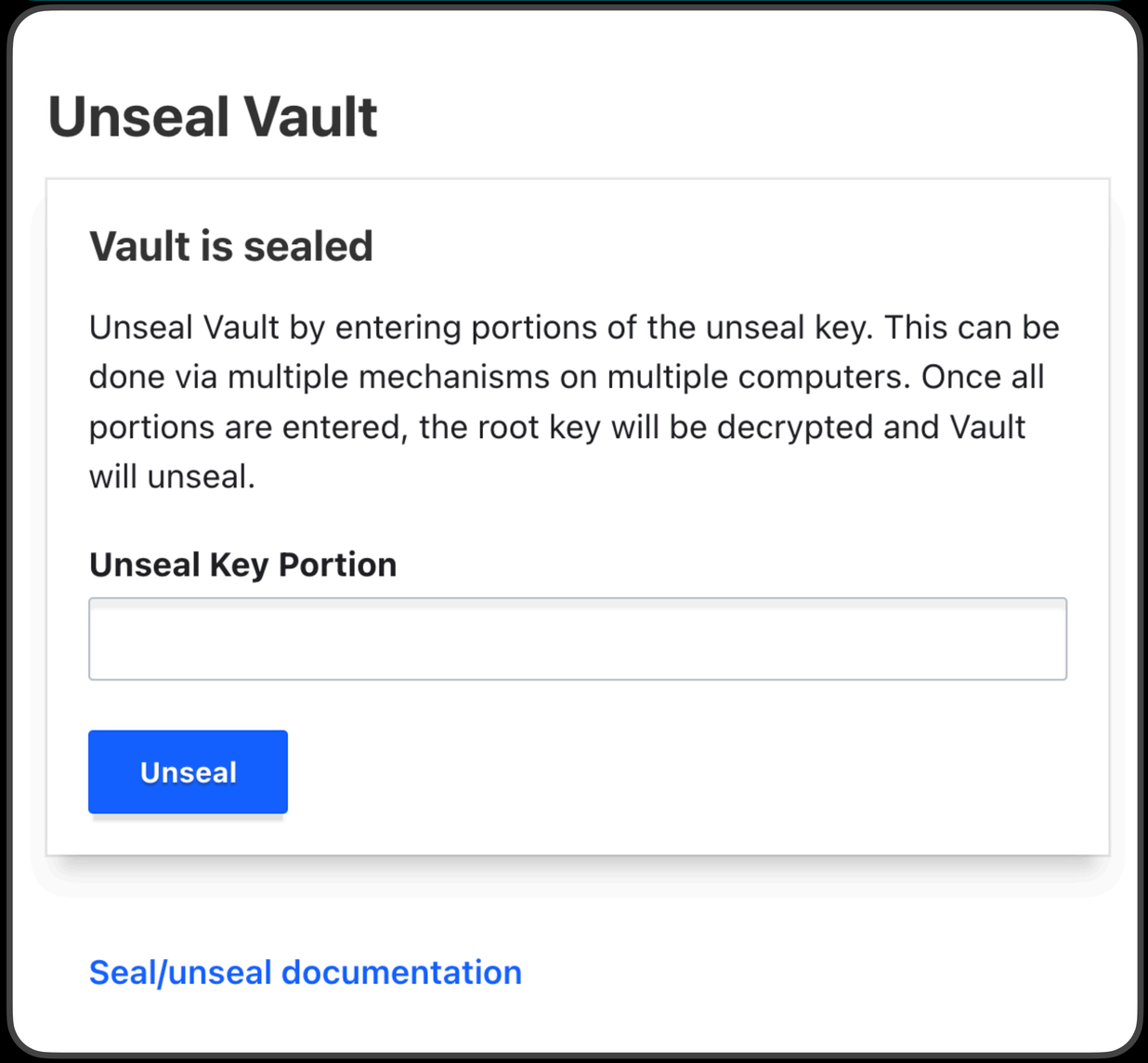

SYSTEM ACTIVATION IN PROGRESS
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key
Unseal key

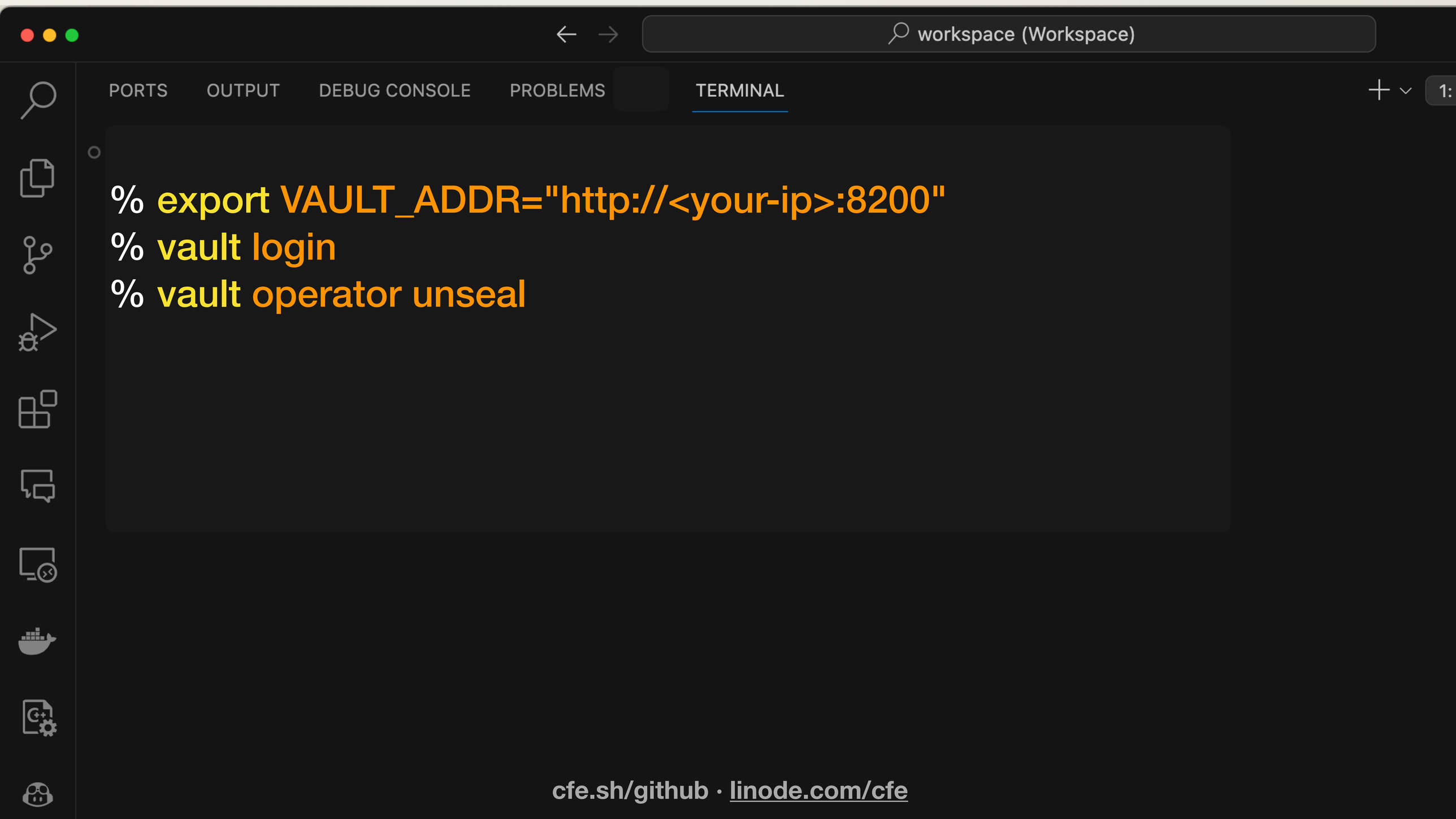
```
$ vault operator unseal
$ vault secrets enable -version=2 kv
```

```
$ vault operator seal
```

$ vault operator unseal

http://<your-ip>:8200/




workspace (Workspace)
PORTS
OUTPUT
DEBUG CONSOLE
PROBLEMS
TERMINAL
% export VAULT_ADDR="http://<your-ip>:8200"
% vault login
% vault operator unseal

Demo: Seal + Unseal

```
$ vault operator init
$ vault login
$ echo "export VAULT_ADDR=\"http://127.0.0.1:8200\"" \
                >> ~/.bashrc
$ source ~/.bashrc
$ echo $VAULT_ADDR
$ vault login
$ vault secrets enable -version=2 kv
$ vault operator unseal
$ vault secrets enable -version=2 kv
$ vault operator seal
```

Unseal vault in ui

Managing Tokens + Share Keys

```
Unseal Key 1: <key-1>
Unseal Key 2: <key-2>
Unseal Key 3: <key-3>
Unseal Key 4: <key-4>
Unseal Key 5: <key-5>
Unseal Key 6: <key-6>
Unseal Key 7: <key-7>
```


Quit
Key


I am once again asking
for your vault share key


Share Key
Don't Share

Rotate Unseal Keys*

```
$ vault operator rekey -init -key-shares=8 -key-threshold=5
$ vault operator rekey
```

*Requires Unseal Keys

Generate New Root Token….

```
$ vault token create -field=token
```

(Spoiler alert: not really)

Revoke Token

```
$ vault token revoke <vault-token>
```

Generate New Root Token*

```
$ vault operator generate-root -init
$ vault operator generate-root
$ vault operator generate-root \
      -decode=<encoded-token> \
      -otp=<otp-from-init>
```

*Requires Unseal Keys

Demo: Managing Tokens & Share Keys

```
$ vault operator rekey -init -key-shares=8 -key-threshold=5
$ vault operator rekey

$ vault token create -field=token
$ vault token revoke <vault-token>

$ vault operator generate-root -init
$ vault operator generate-root
$ vault operator generate-root \
        -decode=<encoded-token> \
        -otp=<otp-from-init>
```

Users

```
$ vault auth enable userpass
```

User Login

```
$ curl -X POST \
—data '{"password": "<pw>"}' \
$VAULT_ADDR/v1/auth/userpass/login/<un>
```

```
$ vault login -method=userpass username=username
```

Use User Token

```
$ curl -X GET --header "X-Vault-Token: <token>" \
    $VAULT_ADDR/v1/<mount>/data/path/to/secret/
```

```
$ vault kv get -mount=<mount> path/to/secret/
```

Enable Username + Passwords

```
$ vault auth enable userpass
$ vault write auth/userpass/users/<username> \
      password=<good-password>
```

*Requires Unseal Keys

Enable Username + Passwords

```
$ openssl rand -base64 32
$ vault write auth/userpass/users/cfe-user \
      password=<good-password>
```

*Requires Unseal Keys

workspace (Workspace)

PORTS OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL

% export VAULT_A
% vault login -me

HTTP

```
const password = 'foo';

vault.auths()
    .then((result) => {
        if (result.hasOwnProperty('userpass/')) return undefined;
        return vault.enableAuth({
            mount_point: mountPoint,
            type: 'userpass',
            description: 'userpass auth',
        });
    })
    .then(() => vault.write(`auth/userpass/users/${username}`, { password, policies: 'root' }))
    .then(() => vault.userpassLogin({ username, password }))
    .then(console.log)
    .catch((err) => console.error(err.message));
```

```
Python 3.11.4 (main, Jun 20 2023, 17:23:00) [Clang 14.0.3 (clang-1403.0.22.14.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import hvac
>>> client = hvac.Client()
>>> client.auth.userpass.login(
...     username='<some_username>',
...     password='<username_password>',
... )
```

Demo: Create User & Login

```
$ vault auth enable userpass
$ openssl rand -base64 32
$ vault write auth/userpass/users/cfe-user password=<password>

# on local machine
% vault login -method=userpass username=cfe-user
% vault token create
```

Policies

```
Error creating token: Error making API request.

URL: POST http://172.234.28.183:8200/v1/auth/token/create
Code: 403. Errors:

* 1 error occurred:
        * permission denied
```

Vault Policy Format

path/to/policy/doc.hcl

```
path “<mount>/path/to/resource/“ {
   capabilities =["create", "update", "patch", "read", "list",
"delete", "sudo"]
}
```

```
$ vault policy write <policy-name> path/to/policy/doc.hcl
```

Vault Policy

policies/token-policy.hcl

```
path "auth/token/*" {
   capabilities =["create", "update", "patch", "read", "delete"]
}
```

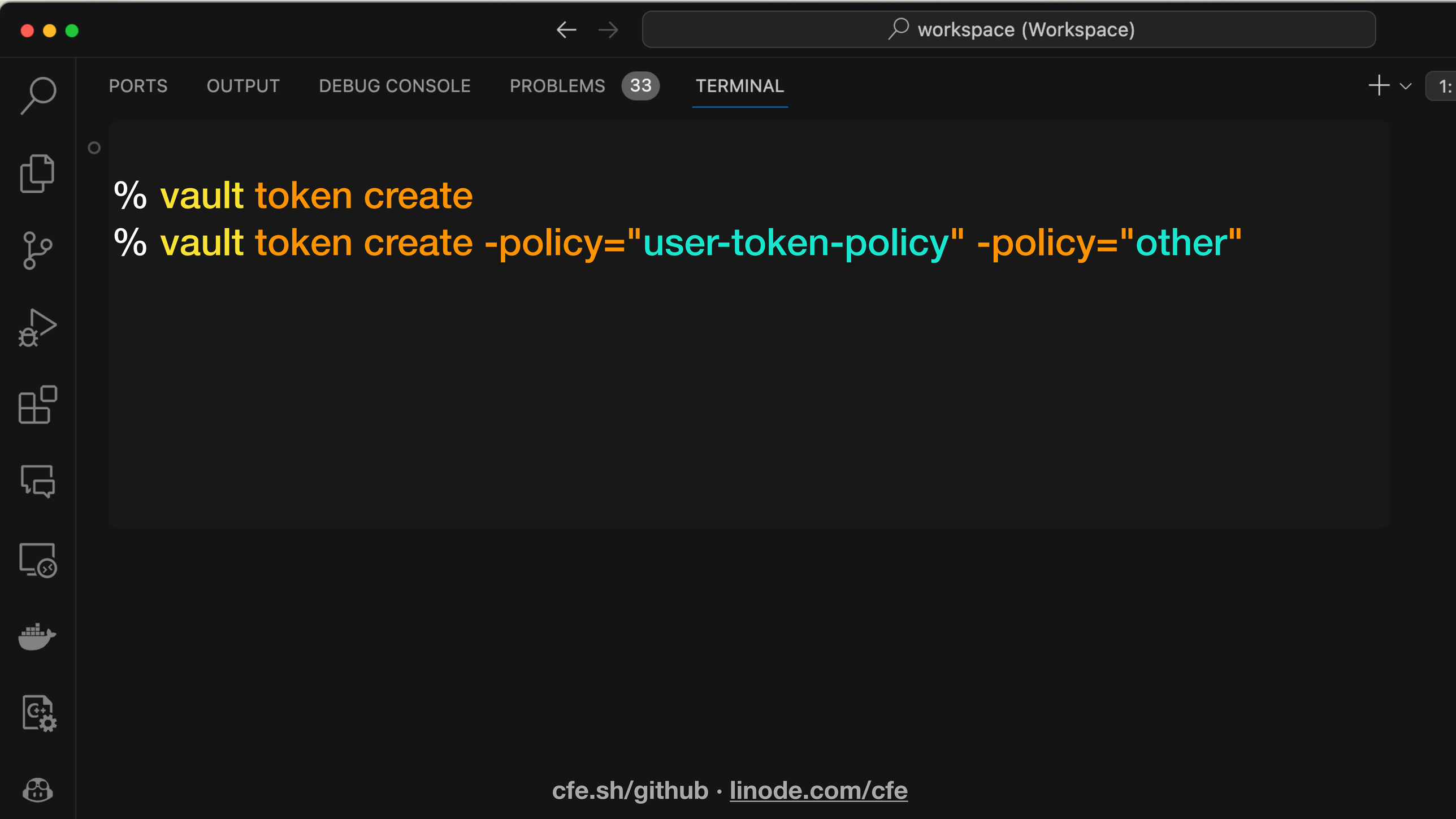
```
$ vault policy write user-token-policy ./policies/token-policy.hcl
```

Attach Policy to User

```
$ vault policy write user-token-policy ./policies/token-policy.hcl
```

```
$ vault write auth/userpass/users/cfe-user/polices
policies="user-token-policy"
```

```
$ vault read auth/userpass/users/cfe-user/
```


workspace (Workspace)
PORTS
OUTPUT
DEBUG CONSOLE
PROBLEMS 33
TERMINAL
% vault token create
% vault token create -policy="user-token-policy" -policy="other"
cfe.sh/github · linode.com/cfe

Demo: Token Policies

```
$ mkdir -p ./policies/
$ nano ./policies/token-policy.hcl

path "auth/token/*" {
   capabilities =["create", "update", "patch", "read", "delete"]
}


$ vault policy write user-token-policy ./policies/token-policy.hcl
$ vault write auth/userpass/users/cfe-user/policies policies="user-token-policy"
$ vault read auth/userpass/users/cfe-user/

# on local machine
% vault login -method=userpass username=cfe-user
% vault token create -policy=“user-token-policy"
```

Key Value Secrets Engine

Key-Value Secrets Engine

- Decide the engine mount path
- Decide the secret path
- Configure access permissions via policy

Key-Value Secrets Engine

- Engine Mount Path → cfe
- Secret Storage Path → project/dev/web
- Access Policy → <mount>/data/<secret-path>
  cfe/data/project/dev/web

- Deletion Policy →
  cfe/delete/project/dev/web
  cfe/undelete/project/dev/web
  cfe/destroy/project/dev/web

Enable Key-Value Secrets Engine

Engine mount path is cfe

```
$ vault secrets enable -path="cfe" -version=2 kv
```

Configure Secret Access Policy

Engine mount path is cfe
Secret path is project/dev/web
Secret policy is cfe-kv-secret-policy

policies/cfe-kv-secret-policy.hcl

```
path "cfe/data/project/dev/web" {
   capabilities =["create", "update", "patch", "read", "list"]
}
```

```
$ vault policy write cfe-kv-secret-policy ./policies/cfe-kv-secret-policy.hcl
```

Attach Secret Access Policy

Engine mount path is cfe
Secret path is project/dev/web
Secret policy is cfe-kv-secret-policy

```
$ vault write auth/userpass/users/cfe-user/policies policies="cfe-kv-secret-policy, user-token-policy"
```

Vault Key-Value Put: Command Line Args

Mount path is cfe
Secret path is project/dev/web

Key-value data

```
{
    "username": "justin",
    "stage": "dev",
}
```

```
$ vault kv put -mount="cfe" project/dev/web \
      username=justin \
      stage=web
```

Vault Key-Value Put: External File

Mount path is cfe
Secret path is project/dev/web

samples/key-value-data.json

```
{
    "username": "justin",
    "stage": "dev",
    "version": "1.0.1",
    "updated": "11-6-23"
}
```

```
$ vault kv put -mount="cfe" project/dev/web @samples/key-value-data.json
```

Vault Key-Value Put: API Request

Mount path is cfe
Secret path is project/dev/web

```
$ export VTOKEN=$(vault token create -field=token)
$ curl -X PUT \
      --header "X-Vault-Token: $VTOKEN" \
      --data '{"data": {"username": "jcurl", "stage": "curly"}}' \
      $VAULT_ADDR/v1/cfe/data/project/dev/web
```

Vault Key-Value GET

Mount path is cfe
Secret path is project/dev/web

```
$ vault kv get -mount=cfe project/dev/web

$ curl -X GET \
      --header "X-Vault-Token: $VTOKEN" \
      $VAULT_ADDR/v1/cfe/data/project/dev/web
```

Vault Key-Value GET Version

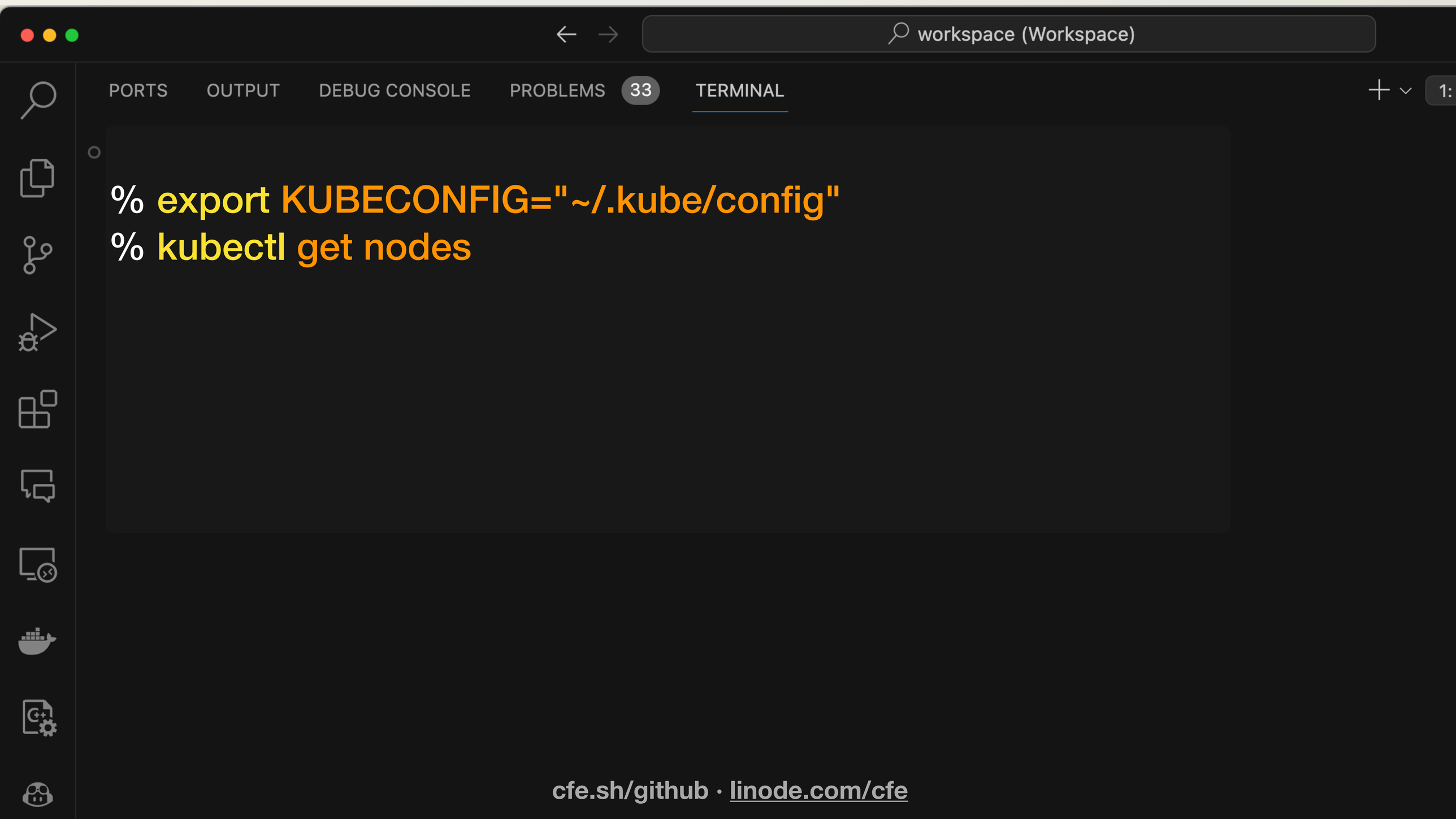
Mount path is cfe
Secret path is project/dev/web
Secret version: 2

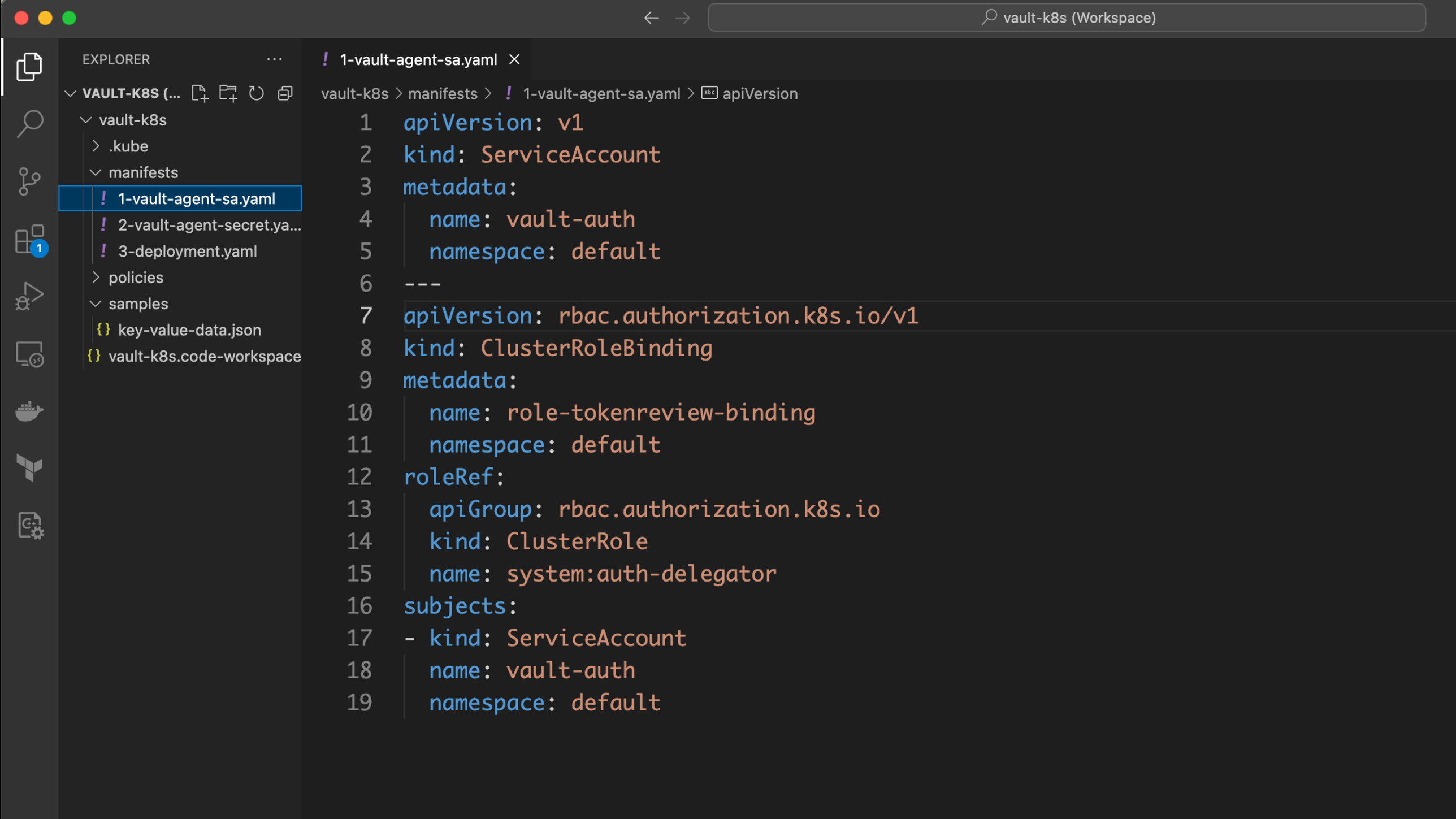
```
$ vault kv get -mount=cfe -version=2 project/dev/web

$ curl -X GET \
        --header "X-Vault-Token: $VTOKEN" \
        $VAULT_ADDR/v1/cfe/data/project/dev/web?version=2
```

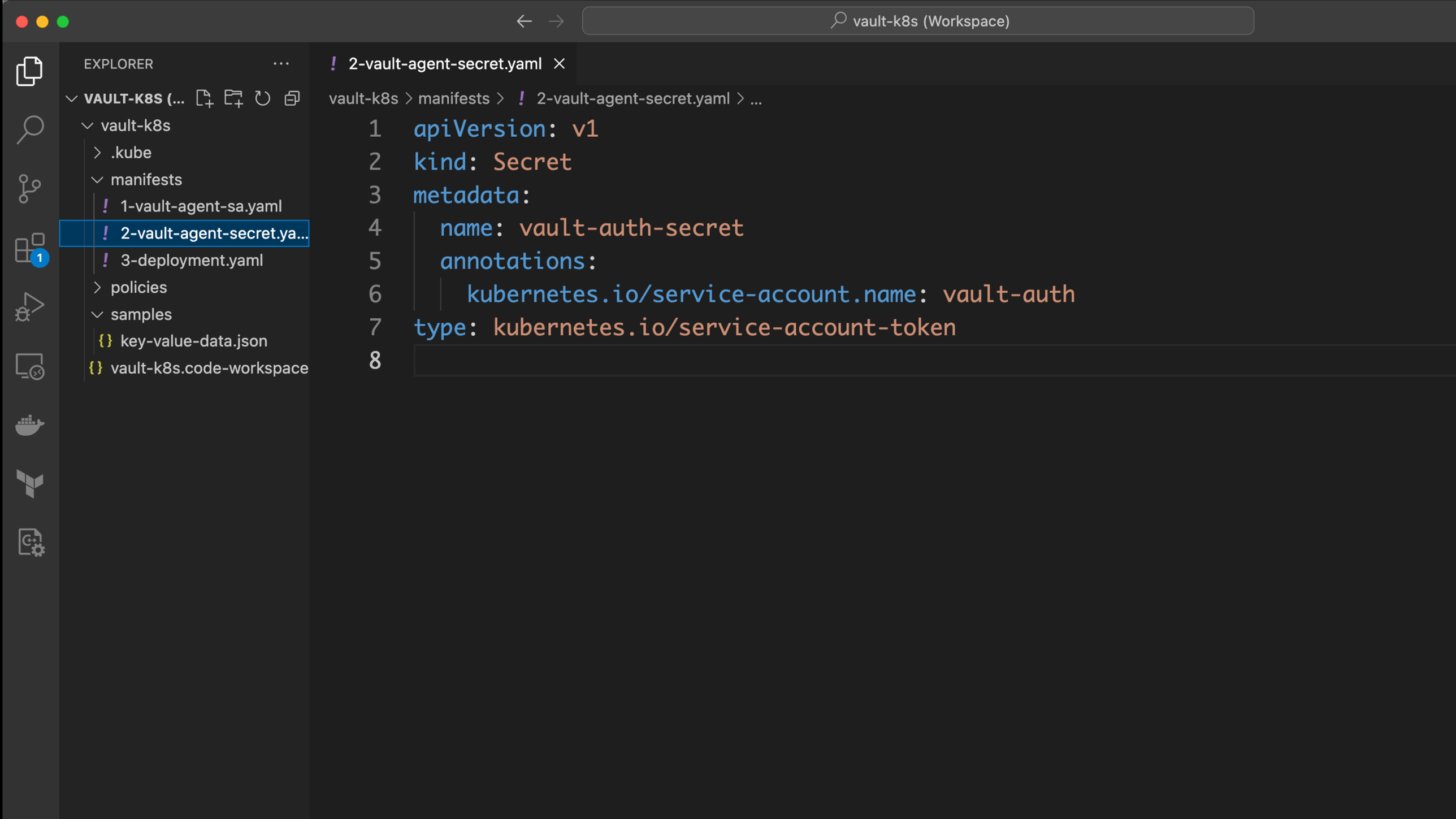
Demo: Vault Secrets Engine

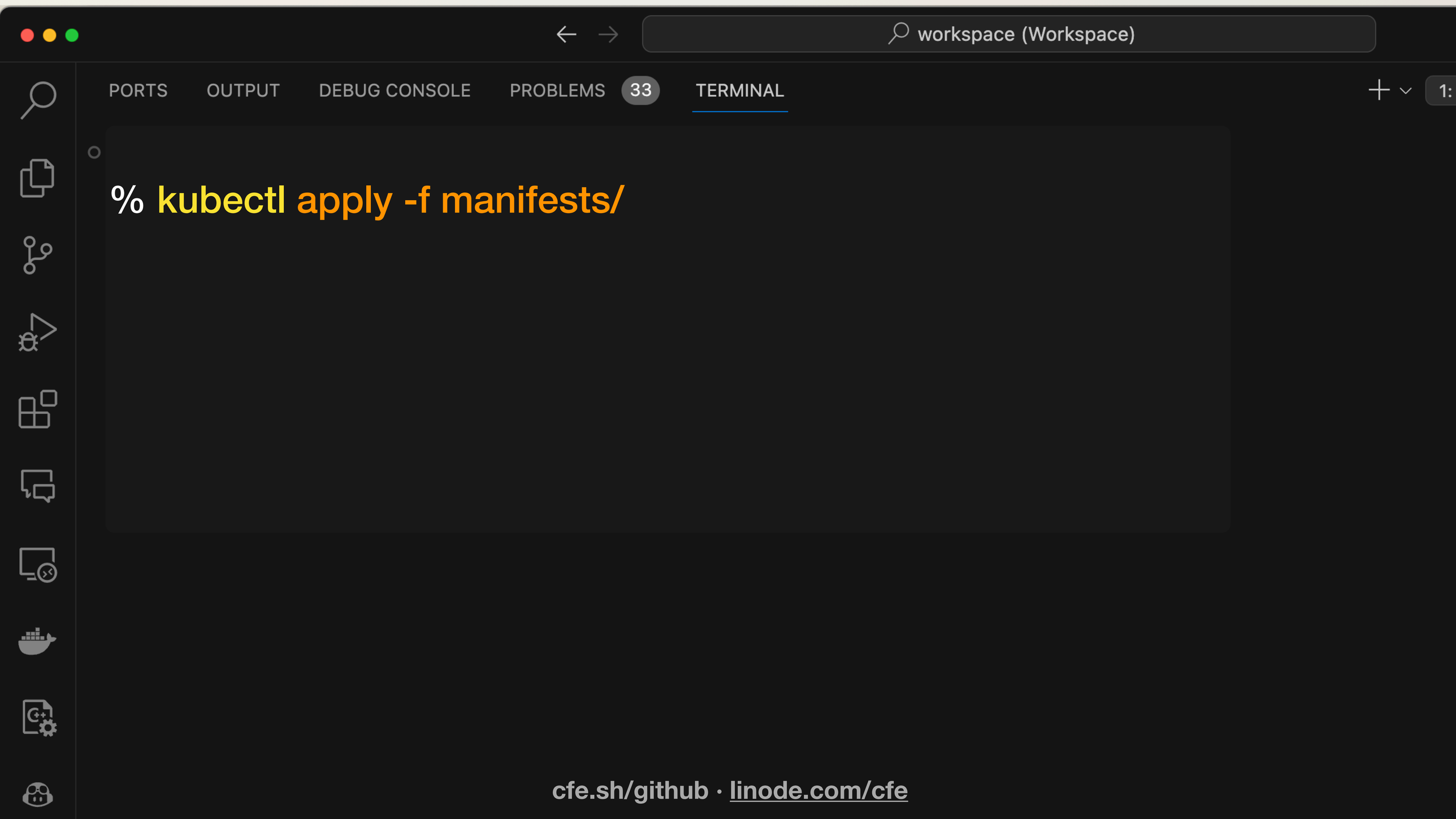
```
$ vault secrets enable -path="cfe" -version=2 kv
$ mkdir -p ./policies/
$ nano ./policies/cfe-kv-secret-policy.hcl

path "cfe/data/project/dev/web" {
   capabilities =["create", "update", "patch", "read", "list"]
}

$ vault policy write cfe-kv-secret-policy ./policies/cfe-kv-secret-policy.hcl
$ vault write auth/userpass/users/cfe-user/policies policies="cfe-kv-secret-policy, user-token-policy”
$ vault kv put -mount="cfe" project/dev/web \
        username=justin \
        stage=wew
$ vault kv get -mount="cfe" project/dev/web
$ vault kv get -mount="cfe" -version=1 project/dev/web
```

Kubernetes


workspace (Workspace)
PORTS
OUTPUT
DEBUG CONSOLE
PROBLEMS 33
TERMINAL
% export KUBECONFIG="~/.kube/config"
% kubectl get nodes
cfe.sh/github · linode.com/cfe


vault-k8s (Workspace)
EXPLORER
VAULT-K8S (...
vault-k8s
.kube
manifests
1-vault-agent-sa.yaml
2-vault-agent-secret.ya...
3-deployment.yaml
policies
samples
key-value-data.json
vault-k8s.code-workspace
1-vault-agent-sa.yaml
vault-k8s > manifests > 1-vault-agent-sa.yaml > apiVersion
1 apiVersion: v1
2 kind: ServiceAccount
3 metadata:
4 name: vault-auth
5 namespace: default
6 ---
7 apiVersion: rbac.authorization.k8s.io/v1
8 kind: ClusterRoleBinding
9 metadata:
10 name: role-tokenreview-binding
11 namespace: default
12 roleRef:
13 apiGroup: rbac.authorization.k8s.io
14 kind: ClusterRole
15 name: system:auth-delegator
16 subjects:
17 - kind: ServiceAccount
18 name: vault-auth
19 namespace: default


vault-k8s (Workspace)
EXPLORER
VAULT-K8S (...
vault-k8s
.kube
manifests
1-vault-agent-sa.yaml
2-vault-agent-secret.ya...
3-deployment.yaml
policies
samples
key-value-data.json
vault-k8s.code-workspace
2-vault-agent-secret.yaml
vault-k8s > manifests > 2-vault-agent-secret.yaml > ...
1 apiVersion: v1
2 kind: Secret
3 metadata:
4 name: vault-auth-secret
5 annotations:
6 kubernetes.io/service-account.name: vault-auth
7 type: kubernetes.io/service-account-token
8


workspace (Workspace)
PORTS
OUTPUT
DEBUG CONSOLE
PROBLEMS 33
TERMINAL
% kubectl apply -f manifests/
cfe.sh/github · linode.com/cfe

PORTS OUTPUT DEBUG CONSOLE PROBLEMS 33 TERMINAL 1: zsh

```
% export SA_JWT_TOKEN=$(kubectl get secret vault-auth-secret -o jsonpath={.data.token} | base64 -d)

% export CLUSTER_CERT =$(kubectl config view -o 'jsonpath={.clusters[].cluster.certificate-authority-data}' --raw | base64 -d)

% export K8S_HOST =$(kubectl config view -o 'jsonpath={.clusters[].cluster.server}')

# use root token
% vault login

% vault auth enable kubernetes
% vault write auth/kubernetes/config \
    token_reviewer_jwt="$SA_JWT_TOKEN" \
    kubernetes_host="$K8S_HOST" \
    kubernetes_ca_cert="$CLUSTER_CERT" \
    issuer="https://kubernetes.default.svc.cluster.local"
```

```
# copy result
% kubectl config view -o 'jsonpath={.clusters[].cluster.certificate-authority-data}' --raw

% python3
>>> import base64
>>> encoded_value = """<paste text here>"""
>>> print(base64.b64decode(encoded_value).decode())
```

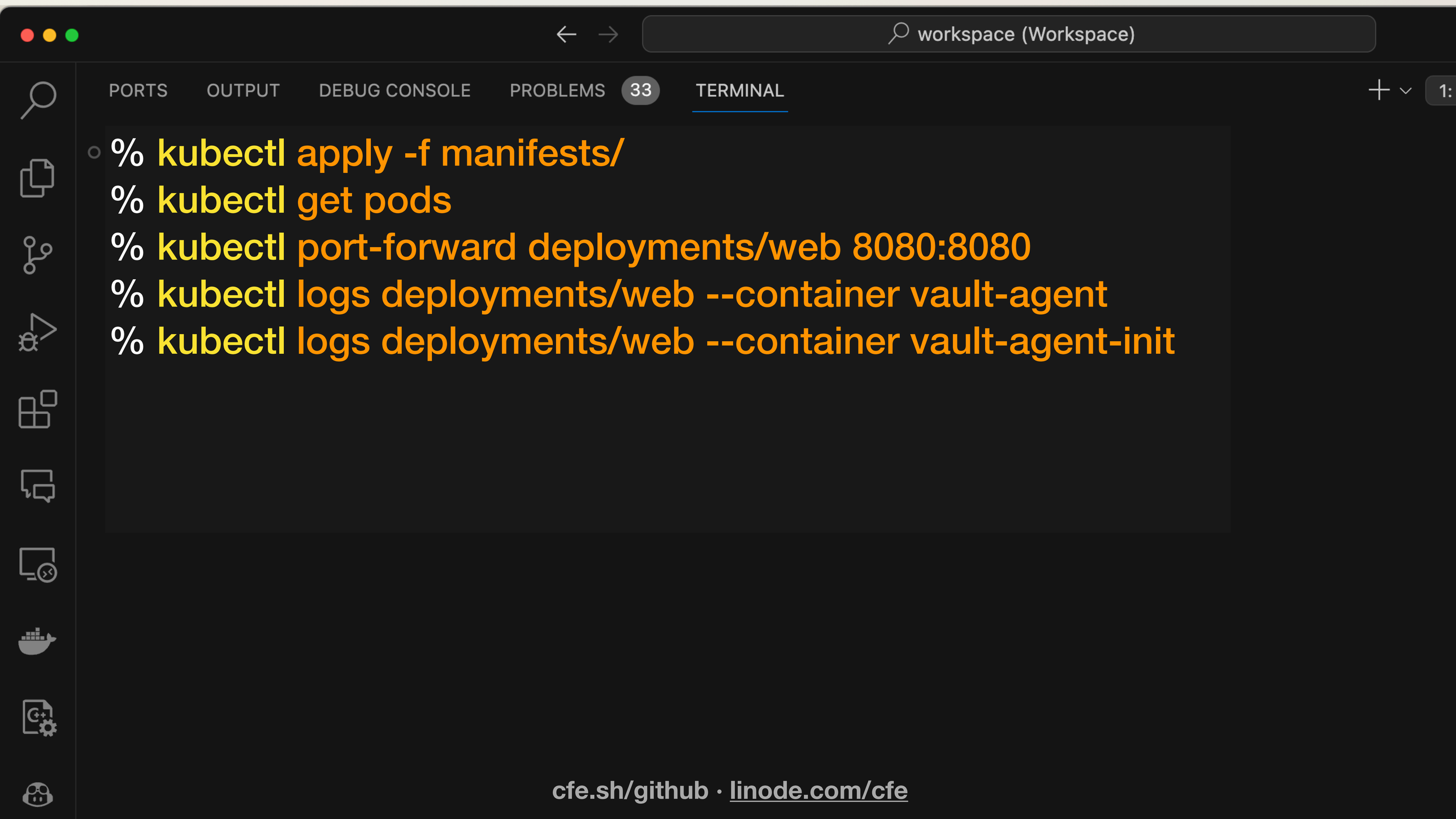
Vault Auth Kubernetes Role

```
$ vault write auth/kubernetes/role/vault-auth-sa-role \
    bound_service_account_names=vault-auth \
    bound_service_account_namespaces=default \
    token_policies=cfe-kv-secret-policy \
    ttl=24h
```

Helm Install Vault Secrets Operator

```
% helm repo add hashicorp https://helm.releases.hashicorp.com
% helm repo update
% helm search repo hashicorp/vault-secrets-operator

% helm install vault hashicorp/vault  \
      --set="injector.enabled=true" \
      --set="global.externalVaultAddr=$VAULT_ADDR" \
      --set="tlsDisable=true"
```

```yaml
apiVersion: apps/v1
kind: Deployment
spec:
  ...
  template:
    metadata:
      ..
      annotations:
        vault.hashicorp.com/agent-inject: "true"
        vault.hashicorp.com/role: "vault-auth-sa-role"
        vault.hashicorp.com/agent-inject-secret-config: "cfe/data/project/dev/web?version=1"
        vault.hashicorp.com/agent-inject-template-config: |
          {{- with secret "cfe/data/project/dev/web" -}}
          API_USER="{{ .Data.data.username }}"
          {{- end -}}
    spec:
      serviceAccountName: vault-auth
      containers:
        - name: web
          image: codingforentrepreneurs/vault-py:latest
          env:
              - name: ENV_VERSION
                value: "1"
              - name: ENV_PATH
                value: /vault/secrets/config
              - name: PORT
                value: "8080"
          ports:
            - containerPort: 8080
```


workspace (Workspace)
PORTS
OUTPUT
DEBUG CONSOLE
PROBLEMS 33
TERMINAL
% kubectl apply -f manifests/
% kubectl get pods
% kubectl port-forward deployments/web 8080:8080
% kubectl logs deployments/web --container vault-agent
% kubectl logs deployments/web --container vault-agent-init

Thank you

Justin Mitchel
cfe.sh
x.com/justinmitchel
linkedin.com/in/justinmitchel
udemy.com/u/justinmitchel