1. Go to https://github.com/bitnami-labs/sealed-secrets/releases
2. Copy the link to the controller.yaml and use wget to download it.
3. Use `kubectl` to apply the contents of the file to the cluster.
4. Go back to the downloads page and copy the link to the kubeseal-0.20.5-linux-amd64.tar.gz
5. Use wget to download the file, move it to some directory in the $PATH variable and add the execute permissions:

```
wget https://github.com/bitnami-labs/sealed-
secrets/releases/download/v0.20.5/kubeseal-0.20.5-linux-amd64.tar.gz
tar -xvf kubeseal-0.20.5-linux-amd64.tar.gz
sudo mv kubeseal /usr/local/bin
sudo chmod +x /usr/local/bin/kubeseal
```

6. Create a new directory in the `myapp` directory called `apiservice`:

```
mkdir apiservice
```

7. Inside the directory, create a `deployment.yaml` file and add the following content:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: busybox-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox
        command: ["sh", "-c", "while true; do sleep 3600; done"]
        env:
        - name: APIKEY
          valueFrom:
            secretKeyRef:
              name: appsecret
              key: apikey
      restartPolicy: Always
```

8. Encode the dummy API key into base64 format:

```
echo api_key_2a6f1d23eabc482f9032165de5a8c7 | base64
```

9. Create a new file called `secret.yaml` and add the following:

```
apiVersion: v1
kind: Secret
metadata:
  name: appsecret
type: Opaque
data:
  apikey: YXBpX2tleV8yYTZmMWQyM2VhYmM0ODJmOTAzMjE2NWRlNWE4Yzc=
```

10. Get the public key using

```
kubeseal --fetch-cert > publickey.pem
```

11. Encrypt the contents of the secret using the following command:

```
kubeseal --format=yaml --cert=publickey.pem < secret.yaml > sealedsecret.yaml
```

12. View the file contents:

```
cat sealedsecret.yaml
```

13. Copy the encrypted string and try to decode it using base64.
14. Delete the secret file:

```
rm secret.yaml
rm publickey.pem
```

15. Create a new application in the `argo-cd` directory:

```
cd ../argo-cd
vim apiservice.yaml
```

16. The file contents should be as follows:

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: apiservice
  namespace: argocd
spec:
```

```
  project: default
  source:
    repoURL: 'https://gitlab.com/[username]/samplegitopsapp.git'
    path: apiservice
    targetRevision: main
  destination:
    server: 'https://kubernetes.default.svc'
    namespace: default
  syncPolicy:
    automated:
      selfHeal: true
      prune: true
```

17. Create a branch and an MR:

```
git checkout -b "adds-apiservice"
git add -A
git commit -m "Creates the API service and the sealed secret"
git push --set-upstream origin feature/adds-apiservice
```

18. Create and approve the MR from the link.
19. Go to the Argo CD UI and refresh the argo cd applcation.