Secrets

Target: kubernetes Secrets https://kubernetes.io/docs/concepts/configuration/secret/

Using secret file

```
apiVersion: v1
kind: Secret
metadata:
  name: db-secret
data:
  # mypassword in base64 - example: # echo "mypassword" | base64
  db-password: bXlwYXNzd29yZAo=

---

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-sec
  labels:
    app: nginx-sec
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-sec
  strategy: {}
  template:
    metadata:
      labels:
        app: nginx-sec
    spec:
      containers:
      - name: nginx-sec
        image: nginx
        ports:
          - containerPort: 8088
        resources: {}
        env:
          - name: "spring.datasource.username"
            value: "ossca_plus_ipav"
          - name: "spring.datasource.password"
            valueFrom:
              secretKeyRef:
                name: db-secret
                key: db-password
```

to verify the ENV vars

```
# kubectl exec -it nginx-sec-<vour-pod-id> env | grep spring
spring.datasource.username=ossca_plus_ipav
spring.datasource.password=mypassword
```

Manage secrets in git:

- SealedSecret https://github.com/bitnami-labs/sealed-secrets https://aws.amazon.com/blogs/opensource/managing-secrets-deployment-in-kubernetes-using-sealed-secrets/

Manage secrets in jenkins

Jenkins Credentials

https://www.jenkins.io/doc/book/pipeline/jenkinsfile/#secret-text

```
// Jenkinsfile (Declarative Pipeline)
pipeline {
    agent {
        //
    }
    environment {
        AWS_ACCESS_KEY_ID     = credentials('jenkins-aws-secret-key-id')
        AWS_SECRET_ACCESS_KEY = credentials('jenkins-aws-secret-access-key')
    }
    stages {
        stage('Example stage 1') {
            steps {
                //
            }
        }
        stage('Example stage 2') {
            steps {
                //
            }
        }
    }
}
```

```
pipeline {
    agent {
        // Define agent details here
    }
    environment {
        // The MY_KUBE_SECRET environment variable will be assigned
        MY_KUBE_SECRET = credentials('my-kube-secret')
        MY_KUBE_SECRETTEXT = credentials('my-kube-secrettext') //
        MY_KUBE_SECRETFILE = credentials('my-kube-secrettext') // needs to be tested
    }
    stages {
        stage('Example stage 1') {
            steps {
                sh("kubectl --kubeconfig $MY_KUBE_SECRET get pods")
                sh('cat $MY_KUBE_SECRETTEXT > x.yaml | kubectl apply -f - ')// needs to be tested
                sh('kubectl apply -f $MY_KUBE_SECRETFILE') // needs to be tested
            }
        }
    }
}
```

Jenkins Credentials Provider Plugin

Example: kubernetes-credentials-provider-plugin

https://jenkinsci.github.io/kubernetes-credentials-provider-plugin/examples/

Kubernetes Secrets in AWS

```
aws secretsmanager get-secret-value ...
```

Example:

https://medium.com/faun/a-simple-approach-for-injecting-secrets-into-eks-using-aws-secrets-manager-and-codebuild-b0d45a8faa1