Environmental Variables:

```
controlplane:~$ cat env.yaml
apiVersion: v1
kind: Pod
metadata:
  name: env-pod
  labels:
    purpose: demonstrate-envars
spec:
  containers:
  - name: nginx-con
    image: nginx
    env:
    - name: DB_Name
      value: "sql"
    - name: DB_config
      value: "config_db"
```

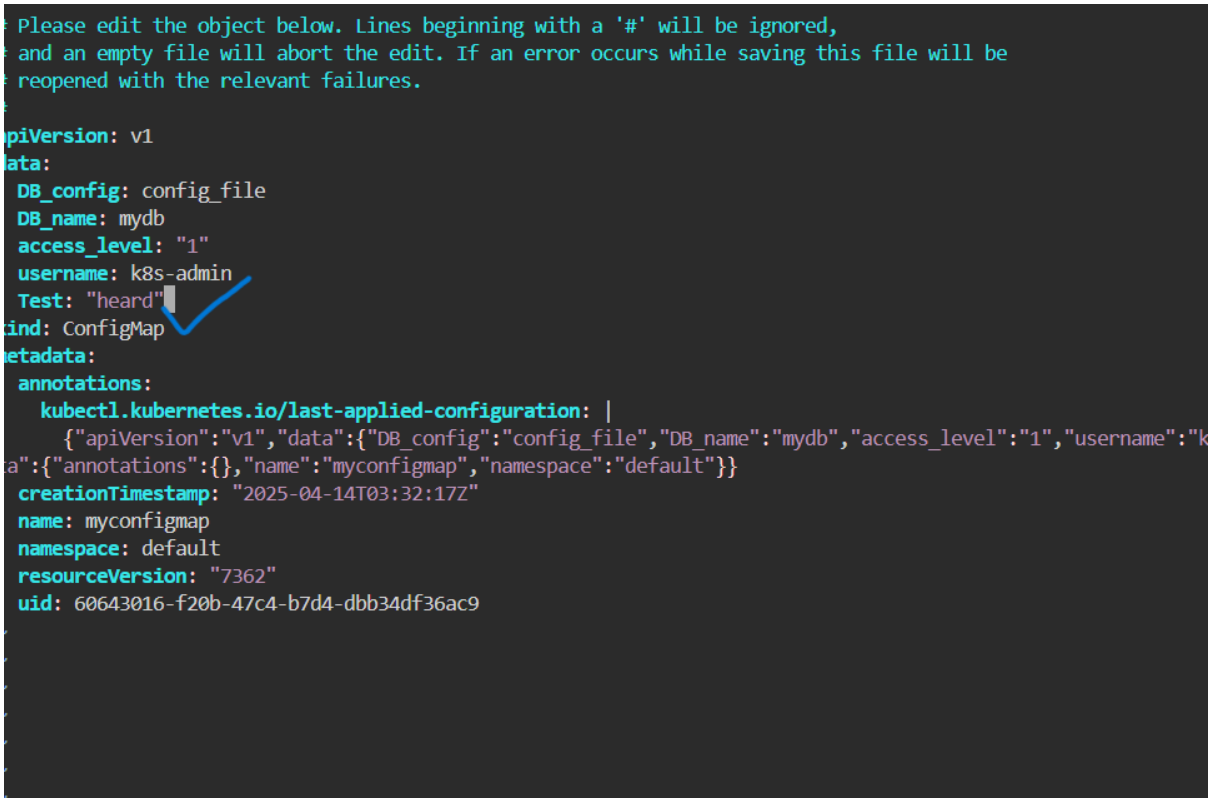
```
env.yaml  filesystem
controlplane:~$ vi env.yaml
controlplane:~$ kubectl apply -f env.yaml
pod/env-pod created
controlplane:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
env-pod   1/1     Running   0          21s
controlplane:~$ kubectl exec -it env-pod -- bin/bash
root@env-pod:/# env
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=env-pod
DB_Name=sql
DB_config=config_db
PWD=/
PKG_RELEASE=1~bookworm
HOME=/root
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
DYNPKG_RELEASE=1~bookworm
NJS_VERSION=0.8.9
TERM=xterm
SHLVL=1
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PORT=443
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
NGINX_VERSION=1.27.4
NJS_RELEASE=1~bookworm
_=/usr/bin/env
root@env-pod:/# cat env.yaml
cat: env.yaml: No such file or directory
root@env-pod:/# exit
exit
```

```
ontrolplane:~$ cat env.yaml
piVersion: v1
ind: Pod
etadata:
  name: config-pod
  labels:
    purpose: demonstrate-envars
pec:
  containers:
  - name: nginx-con
    image: nginx
    envFrom:
        - configMapRef:
            name: myconfigmap
ontrolplane:~$ kubectl get cm
AME                DATA   AGE
ube-root-ca.crt    1      22d
yconfigmap         4      4m40s
ontrolplane:~$ ls
onfig.yaml  env.yaml  filesystem
ontrolplane:~$ cat config.yaml
piVersion: v1
ind: ConfigMap
etadata:
  name: myconfigmap
ata:
  username: k8s-admin
  access_level: "1"
  DB_name: mydb
  DB_config: "config_file"
ontrolplane:~$
```

CONFIG MAP:

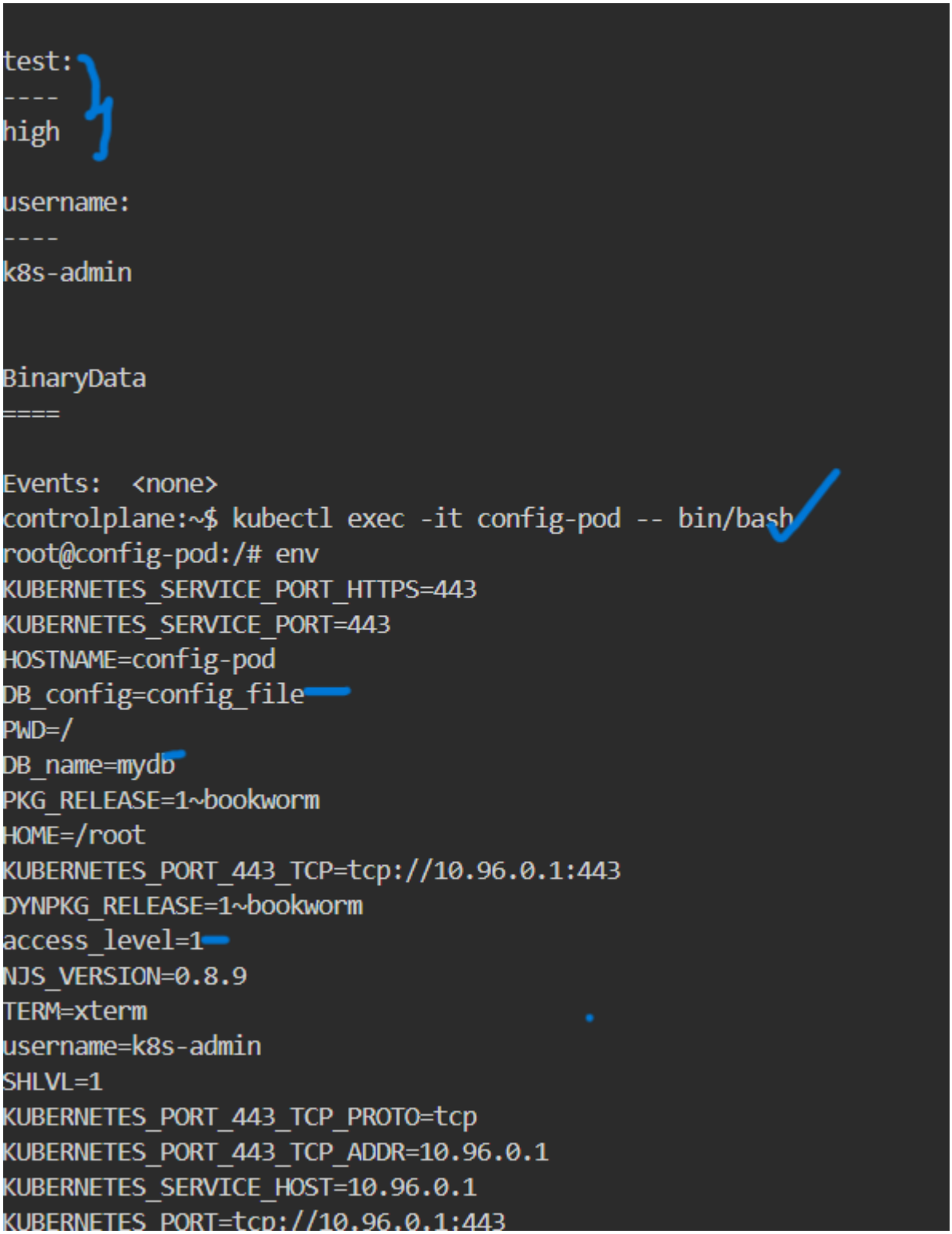
I created a ConfigMap and referenced its values as environment variables in a Pod using envFrom

```
controlplane:~$ env.yaml
env.yaml: command not found
controlplane:~$ vi env.yaml
controlplane:~$ kubectl apply -f env.yaml
pod/config-pod created
controlplane:~$ kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
config-pod   1/1     Running   0          12s
env-pod      1/1     Running   0          11m
controlplane:~$ kubectl exec -it config-pod -- /bin/bash
root@config-pod:/# env
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=config-pod
DB_config=config_file
PWD=/
DB_name=mydb
PKG_RELEASE=1~bookworm
HOME=/root
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
DYNPKG_RELEASE=1~bookworm
access_level=1
NJS_VERSION=0.8.9
TERM=xterm
username=k8s-admin
SHLVL=1
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PORT=443
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
NGINX_VERSION=1.27.4
NJS_RELEASE=1~bookworm
_=/usr/bin/env
```



I updated the ConfigMap by adding another environment variable

```
controlplane:~$ kubectl edit cm myconfigmap
configmap/myconfigmap edited
controlplane:~$ kubectl cm
error: unknown command "cm" for "kubectl"

Did you mean this?
        cp
controlplane:~$ kubectl get cm
NAME               DATA   AGE
kube-root-ca.crt   1      22d
myconfigmap        5      7m42s
controlplane:~$ kubectl describe cm myconfigmap
Name:         myconfigmap
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
DB_config:
----
config_file

DB_name:
----
mydb

access_level:
----
1

test:
----
high
```



After updating the ConfigMap, the new environment variable wasn't reflected in the Pod. I deleted the Pod and reapplied the configuration, after which the new environment variable was visible. However, this approach causes application downtime with every environment variable update. An alternative is to mount ConfigMaps as volumes; this way, changes are dynamically updated in the Pods without requiring restarts.

```
NJS_RELEASE=1~bookworm
_=/usr/bin/env
root@config-pod:/# exit
exit
controlplane:~$ kubectl delete pod config-pod
pod "config-pod" deleted
controlplane:~$ ls
config.yaml  env.yaml  filesystem
controlplane:~$ kubectl apply -f env.yaml
pod/config-pod created
controlplane:~$ kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
config-pod   1/1     Running   0          9s
env-pod      1/1     Running   0          20m
controlplane:~$ kubectl exec -it config-pod -- bin/bash
root@config-pod:/# env
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=config-pod
DB_config=config_file
PWD=/
DB_name=mydb
PKG_RELEASE=1~bookworm
HOME=/root
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
DYNPKG_RELEASE=1~bookworm
access_level=1
NJS_VERSION=0.8.9
TERM=xterm
username=k8s-admin
SHLVL=1
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_PORT=tcp://10.96.0.1:443
```

```
access_level=1
NJS_VERSION=0.8.9
TERM=xterm
username=k8s-admin
SHLVL=1
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PORT=443
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbi
NGINX_VERSION=1.27.4
NJS_RELEASE=1~bookworm
test=high
_=/usr/bin/env
root@config-pod:/# cat config.yaml
cat: config.yaml: No such file or directory
root@config-pod:/# exit
exit
command terminated with exit code 1
```

```
controlplane:~$ cat config.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: myconfigmap
data:
  username: k8s-admin
  access_level: "1"
  DB_name: mydb
  DB_config: "config_file"
```

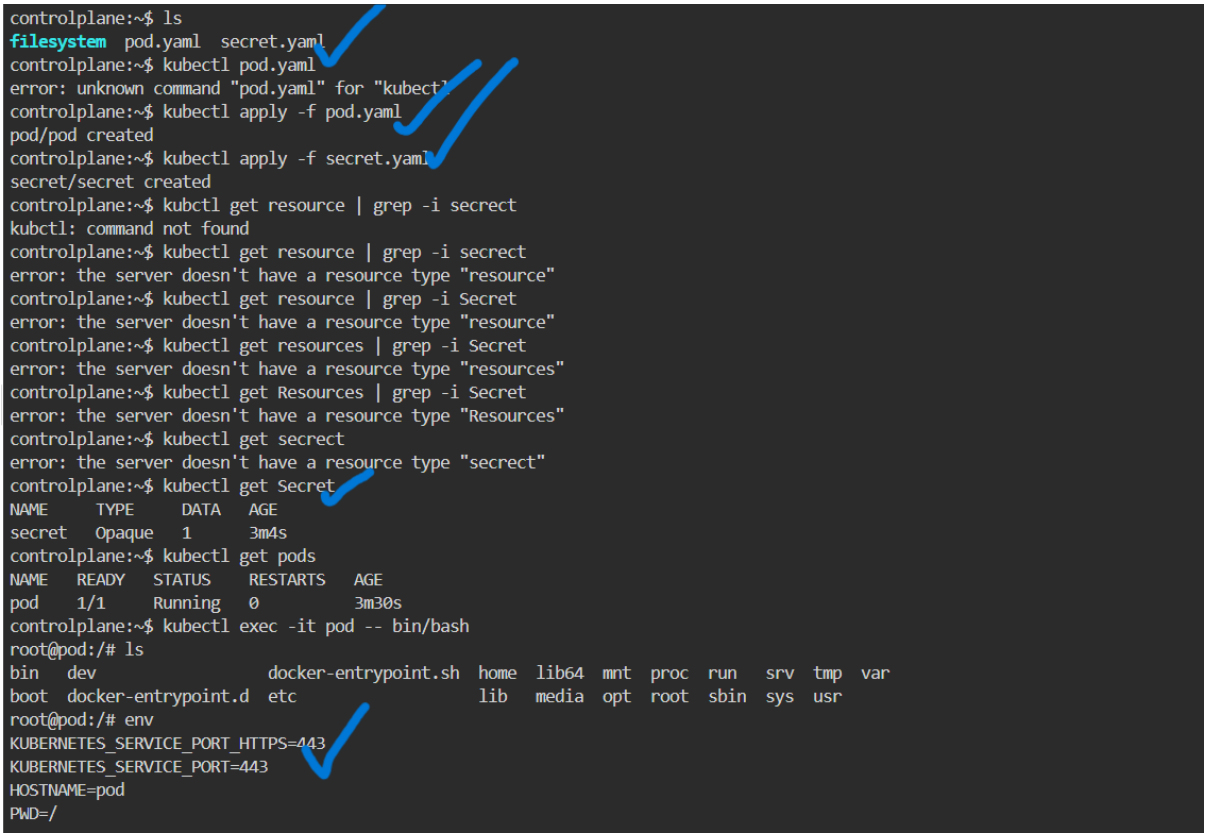
```
controlplane:~$  cat volu-config.yaml
apiVersion: v1
kind: Pod
metadata:
  name: vol-pod
  labels:
    name: vol-pod
spec:
  containers:
  - name: vol-pod
    image: nginx
    ports:
    - containerPort: 80
    volumeMounts:
    - name: volume-for-config
      mountPath: /etc/config
  volumes:
  - name: volume-for-config
    configMap:
      name: myconfigmap
controlplane:~$
```

```
controlplane:~$ vi config.yaml
controlplane:~$ ls
config.yaml  env.yaml  filesystem  volu-config.yaml
controlplane:~$ kubectl apply -f config.yaml
configmap/myconfigmap created
controlplane:~$ kubectl get pods
No resources found in default namespace.
controlplane:~$ kubectl get  cm
NAME               DATA   AGE
kube-root-ca.crt   1      22d
myconfigmap        4      13s
controlplane:~$ kubectl apply -f volu-config.yaml
pod/vol-pod created
controlplane:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
vol-pod   1/1     Running   0          8s
controlplane:~$ kubectl exec -it vol-pod -- bin/bash
root@vol-pod:/# cd /etc/config
root@vol-pod:/etc/config# ls
DB_config  DB_name  access_level  username
root@vol-pod:/etc/config# exit
exit
controlplane:~$ kubectl edit config.yaml
error: the server doesn't have a resource type "config"
controlplane:~$ kubectl edit cm config.yaml
Error from server (NotFound): configmaps "config.yaml" not found
controlplane:~$ kubectl edit cm myconfigmap
configmap/myconfigmap edited
controlplane:~$ kubectl exec -it vol-pod -- bin/bash
root@vol-pod:/# cd /etc/config
root@vol-pod:/etc/config# ls
DB_config  DB_name  Test  access_level  username
root@vol-pod:/etc/config# cat config.yaml
cat: config.yaml: No such file or directory
root@vol-pod:/etc/config# exit
```

Secret:

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret
data:
  extra: LWggcGFzc3dvcmQK
```

```
controlplane:~$ cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod
  labels:
    name: pod
spec:
  containers:
  - name: app-con
    image: nginx
    ports:
    - containerPort: 80
    envFrom:
    - secretRef:
        name: secret
```


controlplane:~$ ls
filesystem pod.yaml secret.yaml
controlplane:~$ kubectl pod.yaml
error: unknown command "pod.yaml" for "kubectl"
controlplane:~$ kubectl apply -f pod.yaml
pod/pod created
controlplane:~$ kubectl apply -f secret.yaml
secret/secret created
controlplane:~$ kubctl get resource | grep -i secrect
kubctl: command not found
controlplane:~$ kubectl get resource | grep -i secrect
error: the server doesn't have a resource type "resource"
controlplane:~$ kubectl get resource | grep -i Secret
error: the server doesn't have a resource type "resource"
controlplane:~$ kubectl get resources | grep -i Secret
error: the server doesn't have a resource type "resources"
controlplane:~$ kubectl get Resources | grep -i Secret
error: the server doesn't have a resource type "Resources"
controlplane:~$ kubectl get secrect
error: the server doesn't have a resource type "secrect"
controlplane:~$ kubectl get Secret
NAME TYPE DATA AGE
secret Opaque 1 3m4s
controlplane:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
pod 1/1 Running 0 3m30s
controlplane:~$ kubectl exec -it pod -- bin/bash
root@pod:/# ls
bin dev docker-entrypoint.sh home lib64 mnt proc run srv tmp var
boot docker-entrypoint.d etc lib media opt root sbin sys usr
root@pod:/# env
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=pod
PWD=/

```
controlplane:~$ kubectl get pods
NAME   READY   STATUS    RESTARTS   AGE
pod    1/1     Running   0          3m30s
controlplane:~$ kubectl exec -it pod -- bin/bash
root@pod:/# ls
bin   dev                  docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
root@pod:/# env
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=pod
PWD=/
DB_password=-h password

PKG_RELEASE=1~bookworm
HOME=/root
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
DYNPKG_RELEASE=1~bookworm
NJS_VERSION=0.8.9
TERM=xterm
SHLVL=1
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PORT=443
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
NGINX_VERSION=1.27.4
NJS_RELEASE=1~bookworm
_=/usr/bin/env
root@pod:/# echo DB_password
DB_password
root@pod:/# echo $DB_password
-h password
root@pod:/#
```