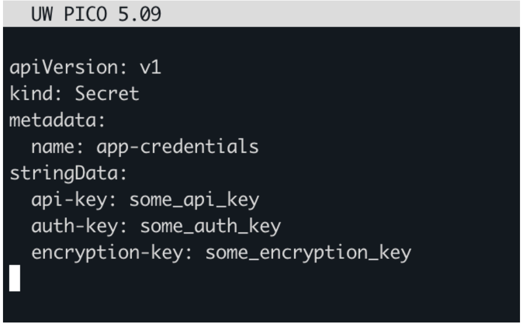
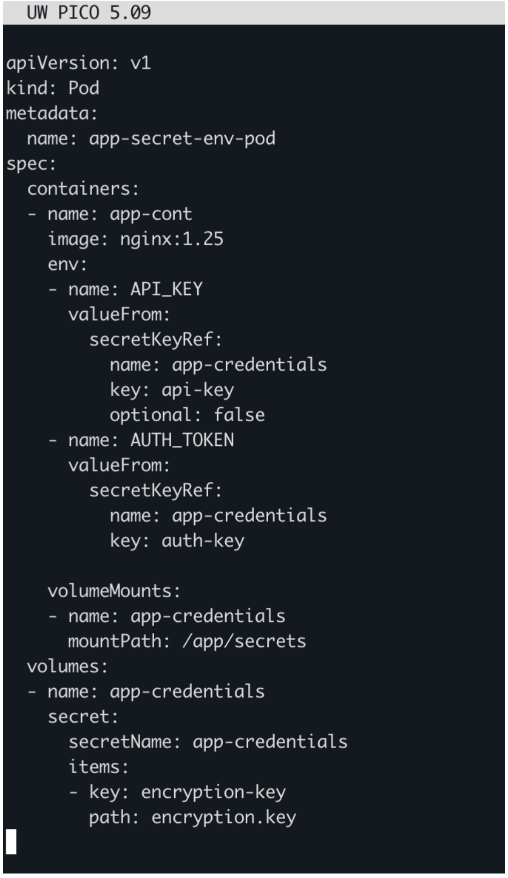
Лабораторная работа №8.

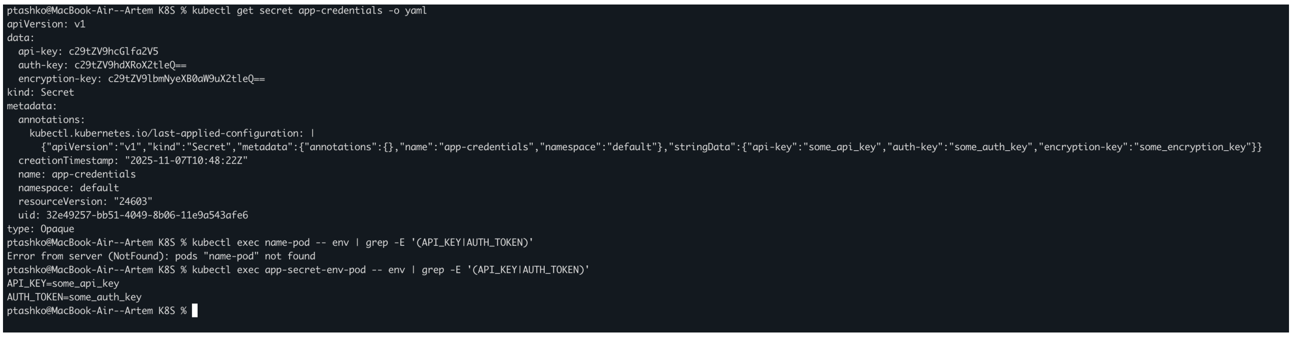
Задание 1.



```
  UW PICO 5.09

apiVersion: v1
kind: Secret
metadata:
  name: app-credentials
stringData:
  api-key: some_api_key
  auth-key: some_auth_key
  encryption-key: some_encryption_key
```

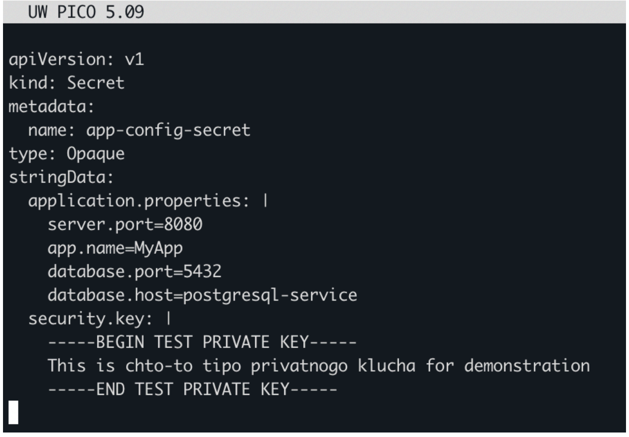


```
  UW PICO 5.09

apiVersion: v1
kind: Pod
metadata:
  name: app-secret-env-pod
spec:
  containers:
  - name: app-cont
    image: nginx:1.25
    env:
    - name: API_KEY
      valueFrom:
        secretKeyRef:
          name: app-credentials
          key: api-key
          optional: false
    - name: AUTH_TOKEN
      valueFrom:
        secretKeyRef:
          name: app-credentials
          key: auth-key

    volumeMounts:
    - name: app-credentials
      mountPath: /app/secrets
  volumes:
  - name: app-credentials
    secret:
      secretName: app-credentials
      items:
      - key: encryption-key
        path: encryption.key
```

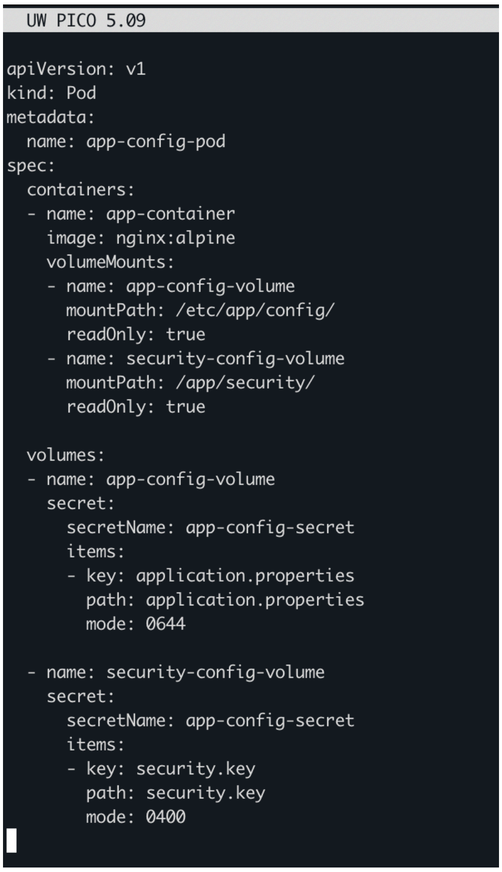


```
ptashko@MacBook-Air--Artem K8S % kubectl get secret app-credentials -o yaml
apiVersion: v1
data:
  api-key: c29tZV9hcGlfa2V5
  auth-key: c29tZV9hdXRoX2tleQ==
  encryption-key: c29tZV9lbmNyeXB0aW9uX2tleQ==
kind: Secret
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Secret","metadata":{"annotations":{},"name":"app-credentials","namespace":"default"},"stringData":{"api-key":"some_api_key","auth-key":"some_auth_key","encryption-key":"some_encryption_key"}}
  creationTimestamp: "2025-11-07T10:48:22Z"
  name: app-credentials
  namespace: default
  resourceVersion: "24603"
  uid: 32e49257-bb51-4049-8b06-11e9a543afe6
type: Opaque
ptashko@MacBook-Air--Artem K8S % kubectl exec name-pod -- env | grep -E '(API_KEY|AUTH_TOKEN)'
Error from server (NotFound): pods "name-pod" not found
ptashko@MacBook-Air--Artem K8S % kubectl exec app-secret-env-pod -- env | grep -E '(API_KEY|AUTH_TOKEN)'
API_KEY=some_api_key
AUTH_TOKEN=some_auth_key
ptashko@MacBook-Air--Artem K8S %
```

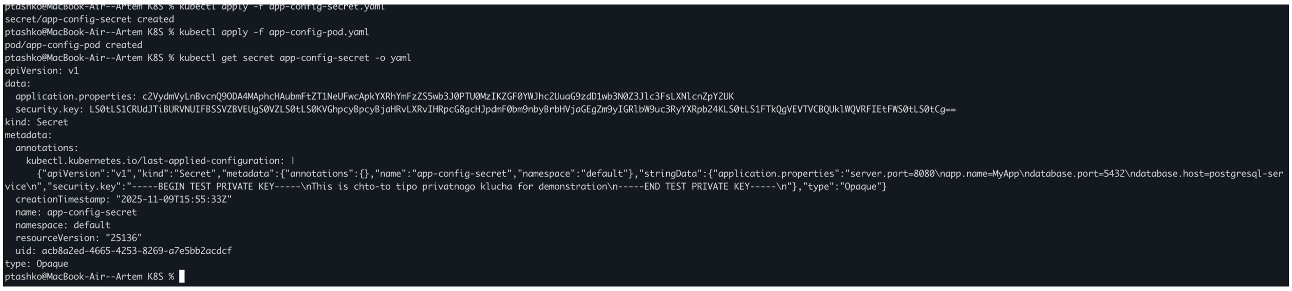
Задание 2.



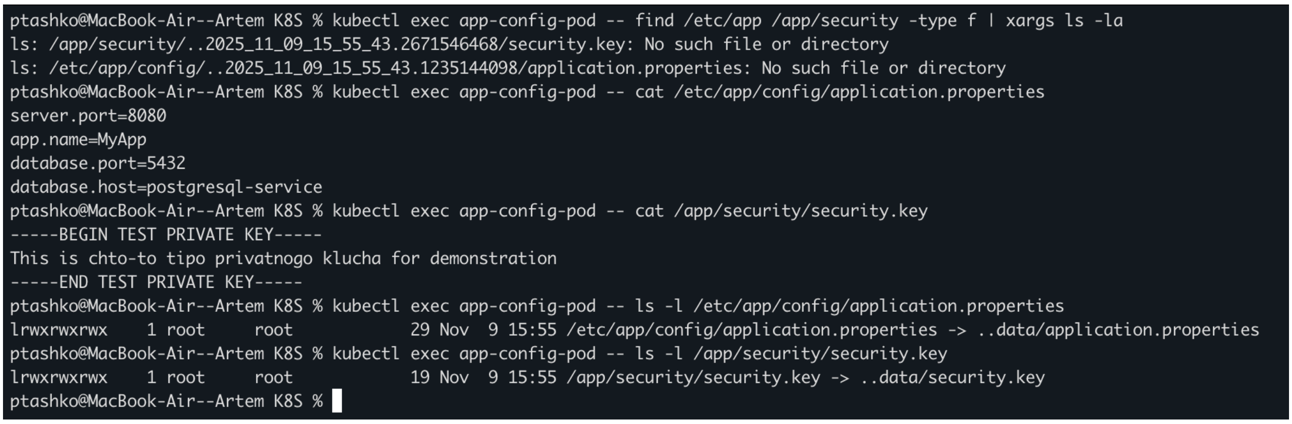
```
  UW PICO 5.09

apiVersion: v1
kind: Secret
metadata:
  name: app-config-secret
type: Opaque
stringData:
  application.properties: |
    server.port=8080
    app.name=MyApp
    database.port=5432
    database.host=postgresql-service
  security.key: |
    -----BEGIN TEST PRIVATE KEY-----
    This is chto-to tipo privatnogo klucha for demonstration
    -----END TEST PRIVATE KEY-----
```



```
  UW PICO 5.09

apiVersion: v1
kind: Pod
metadata:
  name: app-config-pod
spec:
  containers:
  - name: app-container
    image: nginx:alpine
    volumeMounts:
    - name: app-config-volume
      mountPath: /etc/app/config/
      readOnly: true
    - name: security-config-volume
      mountPath: /app/security/
      readOnly: true

  volumes:
  - name: app-config-volume
    secret:
      secretName: app-config-secret
      items:
      - key: application.properties
        path: application.properties
        mode: 0644

  - name: security-config-volume
    secret:
      secretName: app-config-secret
      items:
      - key: security.key
        path: security.key
        mode: 0400
```

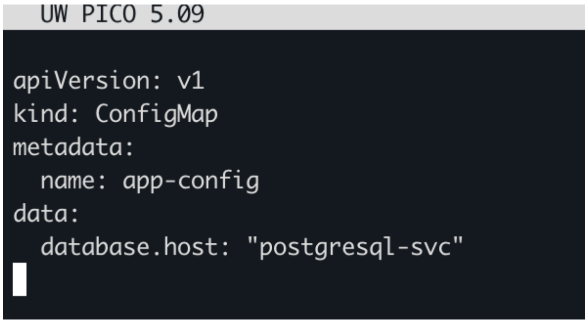


```
ptashko@MacBook-Air--Artem K8S % kubectl apply -f app-config-secret.yaml
secret/app-config-secret created
ptashko@MacBook-Air--Artem K8S % kubectl apply -f app-config-pod.yaml
pod/app-config-pod created
ptashko@MacBook-Air--Artem K8S % kubectl get secret app-config-secret -o yaml
apiVersion: v1
data:
  application.properties: c2VydmVyLnBvcnQ9ODA4MAphcHAubmFtZT1NeUFwcApkYXRhYmFzZS5wb3J0PTU0MzIKZGF0YWJhc2UuaG9zdD1wb3N0Z3Jlc3FsLXNlcnZpY2UK
  security.key: LS0tLS1CRUdJTiBURVNUIFBSSVZBVEUgS0VZLS0tLS0KVGhpcyBpcyBjaHRvLXRvIHRpcG8gcHJpdmF0bm9nbyBrbHVjaGEgZm9yIGRlbW9uc3RyYXRpb24KLS0tLS1FTkQgVEVTVCBQUklWQVRFIEtFWS0tLS0tCg==
kind: Secret
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Secret","metadata":{"annotations":{},"name":"app-config-secret","namespace":"default"},"stringData":{"application.properties":"server.port=8080\napp.name=MyApp\ndatabase.port=5432\ndatabase.host=postgresql-ser
vice\n","security.key":"-----BEGIN TEST PRIVATE KEY-----\nThis is chto-to tipo privatnogo klucha for demonstration\n-----END TEST PRIVATE KEY-----\n"},"type":"Opaque"}
  creationTimestamp: "2025-11-09T15:55:33Z"
  name: app-config-secret
  namespace: default
  resourceVersion: "25136"
  uid: acb8a2ed-4665-4253-8269-a7e5bb2acdcf
type: Opaque
ptashko@MacBook-Air--Artem K8S %
```

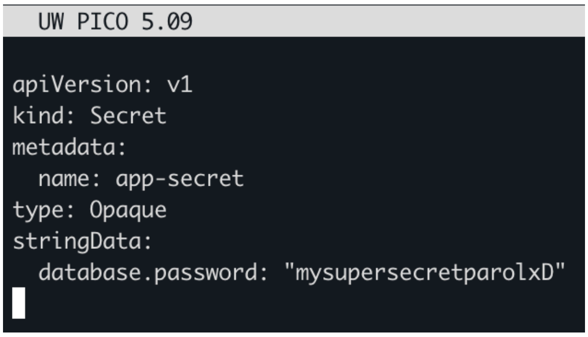


```
ptashko@MacBook-Air--Artem K8S % kubectl exec app-config-pod -- find /etc/app /app/security -type f | xargs ls -la
ls: /app/security/..2025_11_09_15_55_43.2671546468/security.key: No such file or directory
ls: /etc/app/config/..2025_11_09_15_55_43.1235144098/application.properties: No such file or directory
ptashko@MacBook-Air--Artem K8S % kubectl exec app-config-pod -- cat /etc/app/config/application.properties
server.port=8080
app.name=MyApp
database.port=5432
database.host=postgresql-service
ptashko@MacBook-Air--Artem K8S % kubectl exec app-config-pod -- cat /app/security/security.key
-----BEGIN TEST PRIVATE KEY-----
This is chto-to tipo privatnogo klucha for demonstration
-----END TEST PRIVATE KEY-----
ptashko@MacBook-Air--Artem K8S % kubectl exec app-config-pod -- ls -l /etc/app/config/application.properties
lrwxrwxrwx    1 root     root            29 Nov  9 15:55 /etc/app/config/application.properties -> ..data/application.properties
ptashko@MacBook-Air--Artem K8S % kubectl exec app-config-pod -- ls -l /app/security/security.key
lrwxrwxrwx    1 root     root            19 Nov  9 15:55 /app/security/security.key -> ..data/security.key
ptashko@MacBook-Air--Artem K8S %
```

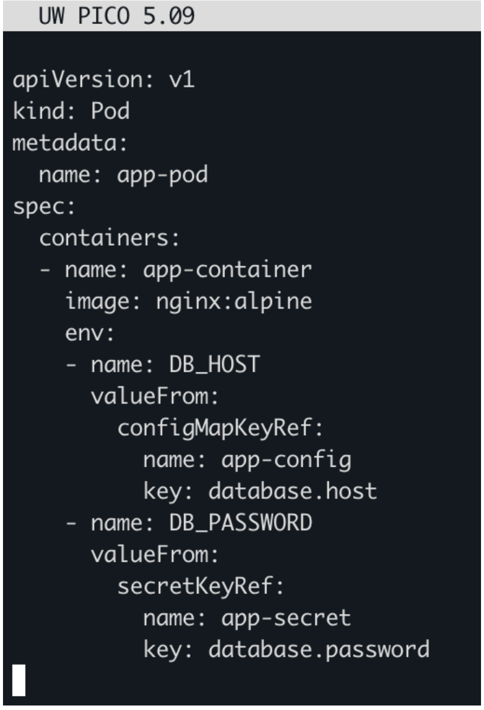
Задание 3.



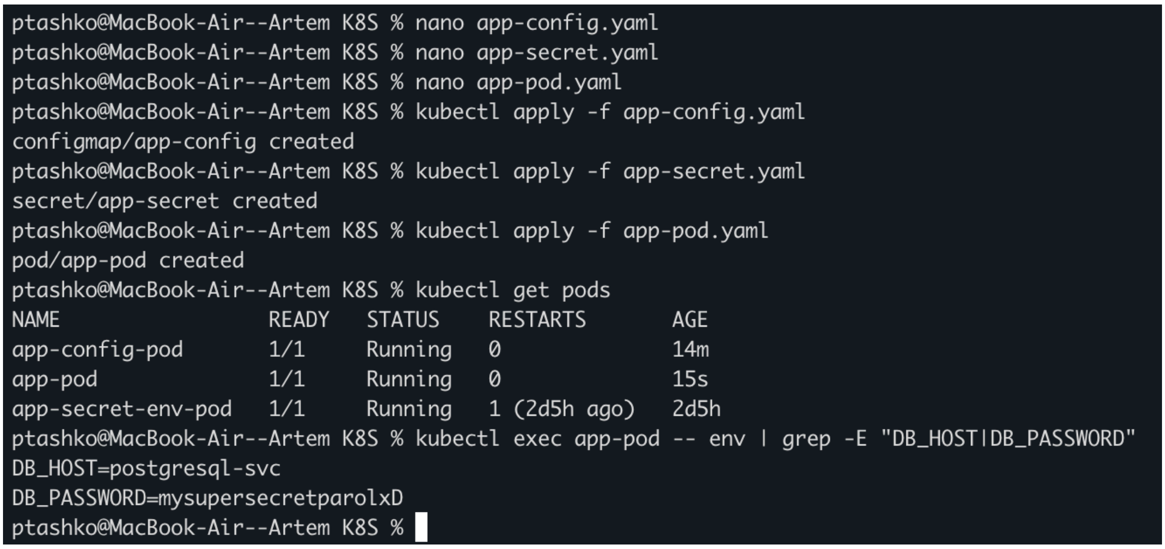
```
  UW PICO 5.09

apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  database.host: "postgresql-svc"
```



```
  UW PICO 5.09

apiVersion: v1
kind: Secret
metadata:
  name: app-secret
type: Opaque
stringData:
  database.password: "mysupersecretparolxD"
```

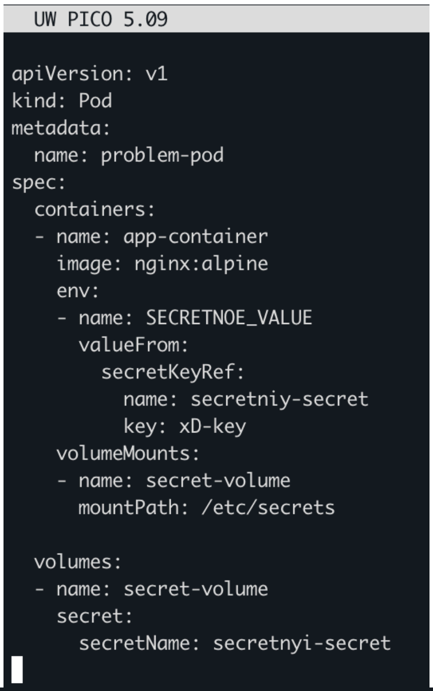


```
  UW PICO 5.09

apiVersion: v1
kind: Pod
metadata:
  name: app-pod
spec:
  containers:
  - name: app-container
    image: nginx:alpine
    env:
    - name: DB_HOST
      valueFrom:
        configMapKeyRef:
          name: app-config
          key: database.host
    - name: DB_PASSWORD
      valueFrom:
        secretKeyRef:
          name: app-secret
          key: database.password
```

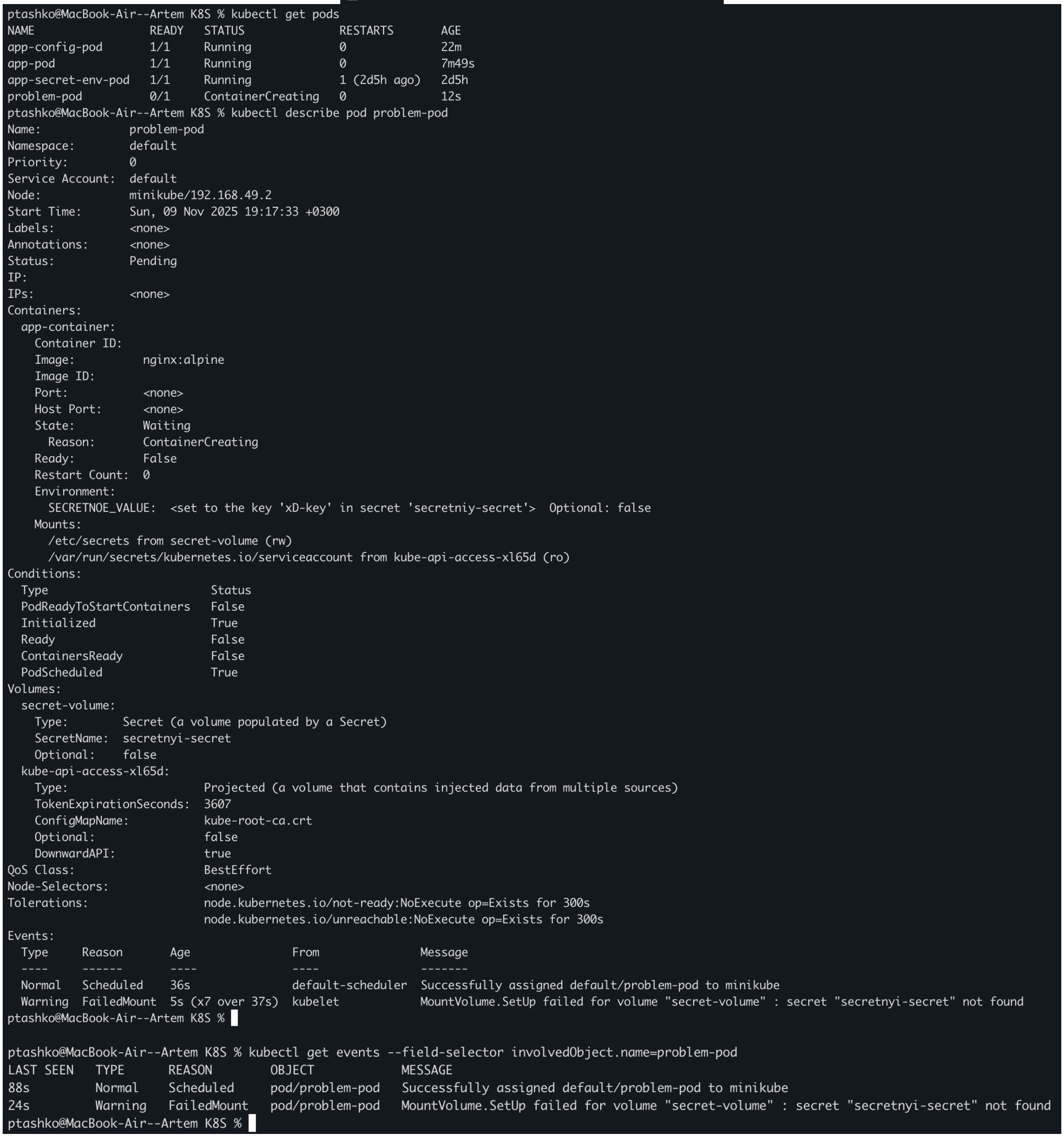


```
ptashko@MacBook-Air--Artem K8S % nano app-config.yaml
ptashko@MacBook-Air--Artem K8S % nano app-secret.yaml
ptashko@MacBook-Air--Artem K8S % nano app-pod.yaml
ptashko@MacBook-Air--Artem K8S % kubectl apply -f app-config.yaml
configmap/app-config created
ptashko@MacBook-Air--Artem K8S % kubectl apply -f app-secret.yaml
secret/app-secret created
ptashko@MacBook-Air--Artem K8S % kubectl apply -f app-pod.yaml
pod/app-pod created
ptashko@MacBook-Air--Artem K8S % kubectl get pods
NAME                 READY   STATUS    RESTARTS       AGE
app-config-pod       1/1     Running   0              14m
app-pod              1/1     Running   0              15s
app-secret-env-pod   1/1     Running   1 (2d5h ago)   2d5h
ptashko@MacBook-Air--Artem K8S % kubectl exec app-pod -- env | grep -E "DB_HOST|DB_PASSWORD"
DB_HOST=postgresql-svc
DB_PASSWORD=mysupersecretparolxD
ptashko@MacBook-Air--Artem K8S %
```

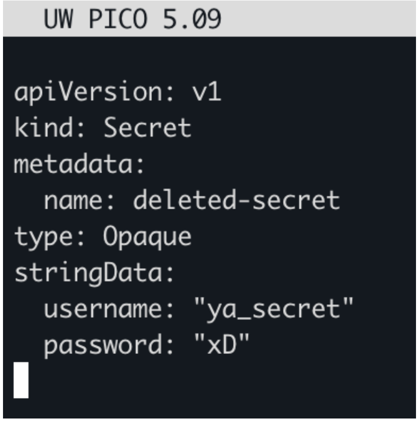
Задание 4.



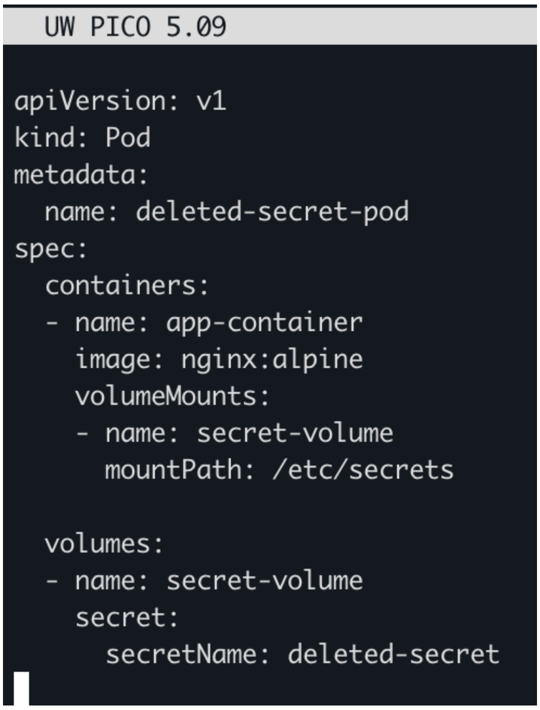
```
UW PICO 5.09

apiVersion: v1
kind: Pod
metadata:
  name: problem-pod
spec:
  containers:
  - name: app-container
    image: nginx:alpine
    env:
    - name: SECRETNOE_VALUE
      valueFrom:
        secretKeyRef:
          name: secretniy-secret
          key: xD-key
    volumeMounts:
    - name: secret-volume
      mountPath: /etc/secrets

  volumes:
  - name: secret-volume
    secret:
      secretName: secretnyi-secret
```



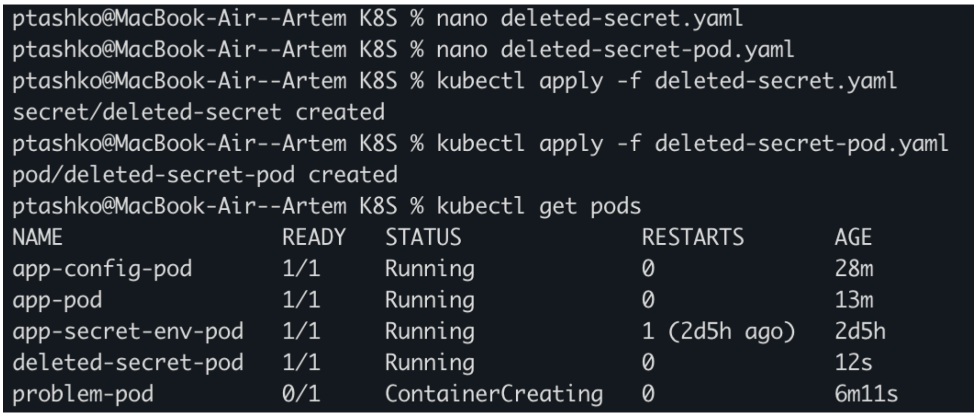
```
ptashko@MacBook-Air--Artem K8S % kubectl get pods
NAME                 READY   STATUS              RESTARTS       AGE
app-config-pod       1/1     Running             0              22m
app-pod              1/1     Running             0              7m49s
app-secret-env-pod   1/1     Running             1 (2d5h ago)   2d5h
problem-pod          0/1     ContainerCreating   0              12s
ptashko@MacBook-Air--Artem K8S % kubectl describe pod problem-pod
Name:             problem-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Sun, 09 Nov 2025 19:17:33 +0300
Labels:           <none>
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  app-container:
    Container ID:
    Image:          nginx:alpine
    Image ID:
    Port:           <none>
    Host Port:      <none>
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
    Restart Count:  0
    Environment:
      SECRETNOE_VALUE:  <set to the key 'xD-key' in secret 'secretniy-secret'>  Optional: false
    Mounts:
      /etc/secrets from secret-volume (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-xl65d (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   False
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
Volumes:
  secret-volume:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  secretnyi-secret
    Optional:    false
  kube-api-access-xl65d:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason       Age               From               Message
  ----     ------       ----              ----               -------
  Normal   Scheduled    36s               default-scheduler  Successfully assigned default/problem-pod to minikube
  Warning  FailedMount  5s (x7 over 37s)  kubelet            MountVolume.SetUp failed for volume "secret-volume" : secret "secretnyi-secret" not found
ptashko@MacBook-Air--Artem K8S % 

ptashko@MacBook-Air--Artem K8S % kubectl get events --field-selector involvedObject.name=problem-pod
LAST SEEN   TYPE      REASON        OBJECT            MESSAGE
88s         Normal    Scheduled     pod/problem-pod   Successfully assigned default/problem-pod to minikube
24s         Warning   FailedMount   pod/problem-pod   MountVolume.SetUp failed for volume "secret-volume" : secret "secretnyi-secret" not found
ptashko@MacBook-Air--Artem K8S % 
```



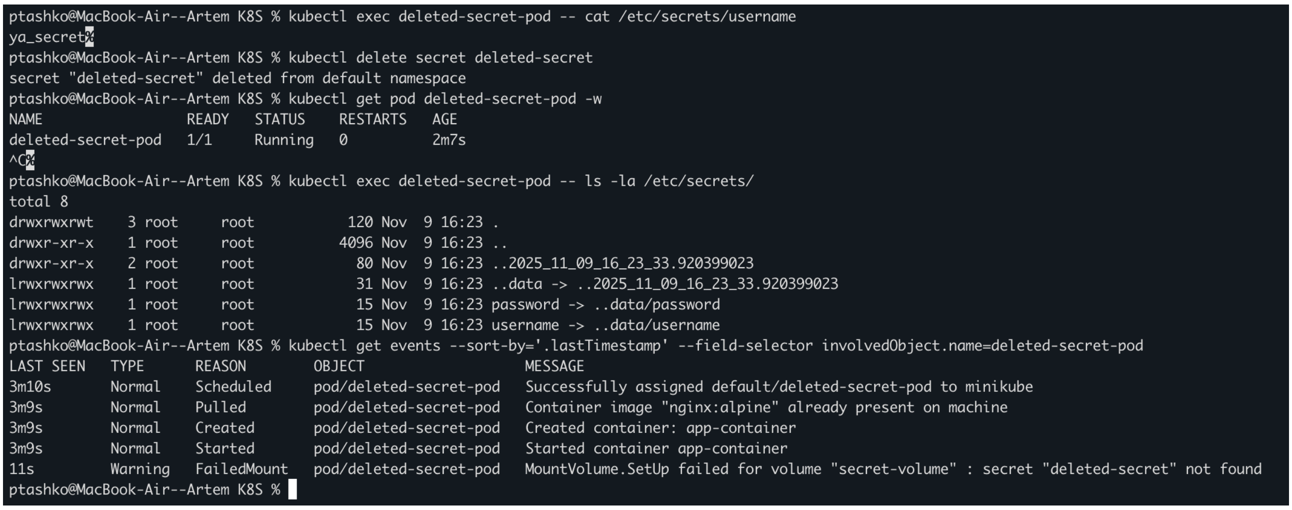
```
  UW PICO 5.09

apiVersion: v1
kind: Secret
metadata:
  name: deleted-secret
type: Opaque
stringData:
  username: "ya_secret"
  password: "xD"
```



```
  UW PICO 5.09

apiVersion: v1
kind: Pod
metadata:
  name: deleted-secret-pod
spec:
  containers:
  - name: app-container
    image: nginx:alpine
    volumeMounts:
    - name: secret-volume
      mountPath: /etc/secrets

  volumes:
  - name: secret-volume
    secret:
      secretName: deleted-secret
```



```
ptashko@MacBook-Air--Artem K8S % nano deleted-secret.yaml
ptashko@MacBook-Air--Artem K8S % nano deleted-secret-pod.yaml
ptashko@MacBook-Air--Artem K8S % kubectl apply -f deleted-secret.yaml
secret/deleted-secret created
ptashko@MacBook-Air--Artem K8S % kubectl apply -f deleted-secret-pod.yaml
pod/deleted-secret-pod created
ptashko@MacBook-Air--Artem K8S % kubectl get pods
NAME                 READY   STATUS              RESTARTS       AGE
app-config-pod       1/1     Running             0              28m
app-pod              1/1     Running             0              13m
app-secret-env-pod   1/1     Running             1 (2d5h ago)   2d5h
deleted-secret-pod   1/1     Running             0              12s
problem-pod          0/1     ContainerCreating   0              6m11s
```



```
ptashko@MacBook-Air--Artem K8S % kubectl exec deleted-secret-pod -- cat /etc/secrets/username
ya_secret%
ptashko@MacBook-Air--Artem K8S % kubectl delete secret deleted-secret
secret "deleted-secret" deleted from default namespace
ptashko@MacBook-Air--Artem K8S % kubectl get pod deleted-secret-pod -w
NAME                 READY   STATUS    RESTARTS   AGE
deleted-secret-pod   1/1     Running   0          2m7s
^C%
ptashko@MacBook-Air--Artem K8S % kubectl exec deleted-secret-pod -- ls -la /etc/secrets/
total 8
drwxrwxrwt    3 root     root           120 Nov  9 16:23 .
drwxr-xr-x    1 root     root          4096 Nov  9 16:23 ..
drwxr-xr-x    2 root     root            80 Nov  9 16:23 ..2025_11_09_16_23_33.920399023
lrwxrwxrwx    1 root     root            31 Nov  9 16:23 ..data -> ..2025_11_09_16_23_33.920399023
lrwxrwxrwx    1 root     root            15 Nov  9 16:23 password -> ..data/password
lrwxrwxrwx    1 root     root            15 Nov  9 16:23 username -> ..data/username
ptashko@MacBook-Air--Artem K8S % kubectl get events --sort-by='.lastTimestamp' --field-selector involvedObject.name=deleted-secret-pod
LAST SEEN   TYPE      REASON        OBJECT                   MESSAGE
3m10s       Normal    Scheduled     pod/deleted-secret-pod   Successfully assigned default/deleted-secret-pod to minikube
3m9s        Normal    Pulled        pod/deleted-secret-pod   Container image "nginx:alpine" already present on machine
3m9s        Normal    Created       pod/deleted-secret-pod   Created container: app-container
3m9s        Normal    Started       pod/deleted-secret-pod   Started container app-container
11s         Warning   FailedMount   pod/deleted-secret-pod   MountVolume.SetUp failed for volume "secret-volume" : secret "deleted-secret" not found
ptashko@MacBook-Air--Artem K8S %
```

Что произойдет с Pod?

Ответ: Pod продолжит работу, данные в УЖЕ смонтированных volumes останутся доступны - но НОВЫЙ Pod не запуститься и выдаст ошибку

Контрольные вопросы.

В чем основное отличие Secret от ConfigMap?
Ответ: Secret шифрует данные и служит для хранения конфиденциальной информации, а ConfigMap не шифрует данные и служит для хранения служебной или общедоступной информации.

Какие два основных способа использования Secrets в Pod вы знаете?
Ответ: как переменные окружения и как файлы.

Какой тип Secret используется по умолчанию при создании через kubectl create secret generic?
Ответ: Opaque.

В чем разница между полями data и stringData в манифесте Secret?
Ответ: data требует base64, а stringData автоматически кодируется в base64.

Что произойдет с Pod, если он ссылается на несуществующий Secret?
Ответ: при использовании в переменных окружения - не запустим Pod и не создастся контейнер, а при монтировании - Pod также не запуститься и Volume не смонтируется.

Данные в Secret хранятся в зашифрованном виде в etcd. Правда или ложь? (Поясните ответ).
Ответ: по умолчанию - НЕТ, они хранятся в закодированном виде (base64), но если настроить etcd - могут храниться в зашифрованном виде.

Как можно безопасно передать содержимое файла (например, сертификата) в Secret?
Ответ:
kubectl create secret с локальными файлами;
Чтение файла напрямую в stdin.

Какие команды вы бы использовали для диагностики проблем с Secrets?
Ответ:

```
kubectl get secrets, pods
kubectl get secret, pod <name-secret/pod> -o wide
kubectl describe secret, pod <name-secret/pod>
kubectl get events --field-selector involvedObject.name=<name-pod>
kubectl logs <name-pod>
kubectl exec <name-pod> -- mount | grep secrets
kubectl exec <name-pod> -- ls -la /etc/secrets/
kubectl exec <name-pod> -- cat /etc/secrets/username
kubectl exec <name-pod> -- env | grep SECRET
```