

# K8S 23: Kubernetes. Resources. Secret

## Описание:

Секреты почти, как configmap, также хранят key=value пары значений. Но, в отличие от configmap, они доставляются только на те ноды, которые запускают поды, использующие данные секреты. Всегда хранятся в памяти и не пишутся на диск. Также данное значение шифруется. По умолчанию base64, но есть и другие алгоритмы. Предположим, мы хотим сохранить в секрет пароль от базы данных.

```
kubectl create secret generic my-password  
--from-literal=password=mysqlpassword
```

После создания попробуйте прочитать секрет:

```
kubectl get secret my-password
```

Значение скрыто, но если добавить -o yaml или json, вы увидите зашифрованный секрет.

Конечно, base64 не очень безопасен, есть другие методы, которые куда более применимы. О них можно прочитать в этой статье [Encrypting Secret Data at Rest](#)

Секреты также можно использовать как переменные окружения и файлы.

список:

```
containers:
```

```
- image: wordpress:4.7.3-apache  
  name: wordpress
```

```
env:
```

```
- name: WORDPRESS_DB_HOST  
  value: wordpress-mysql  
- name: WORDPRESS_DB_PASSWORD
```

```
valueFrom:
```

```
  secretKeyRef:
```

```
    name: my-password  
    key: password
```

```
---
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: mypod
```

```
spec:
```

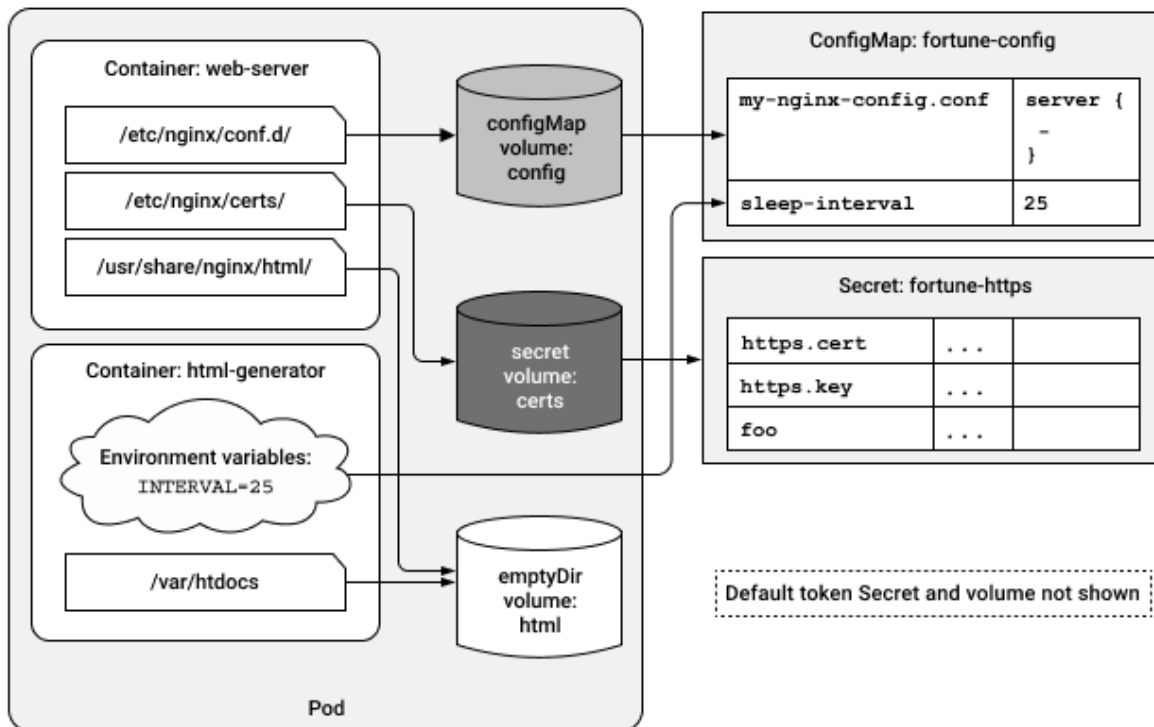
```
  containers:
```

```
- name: mypod  
  image: redis  
  volumeMounts:
```

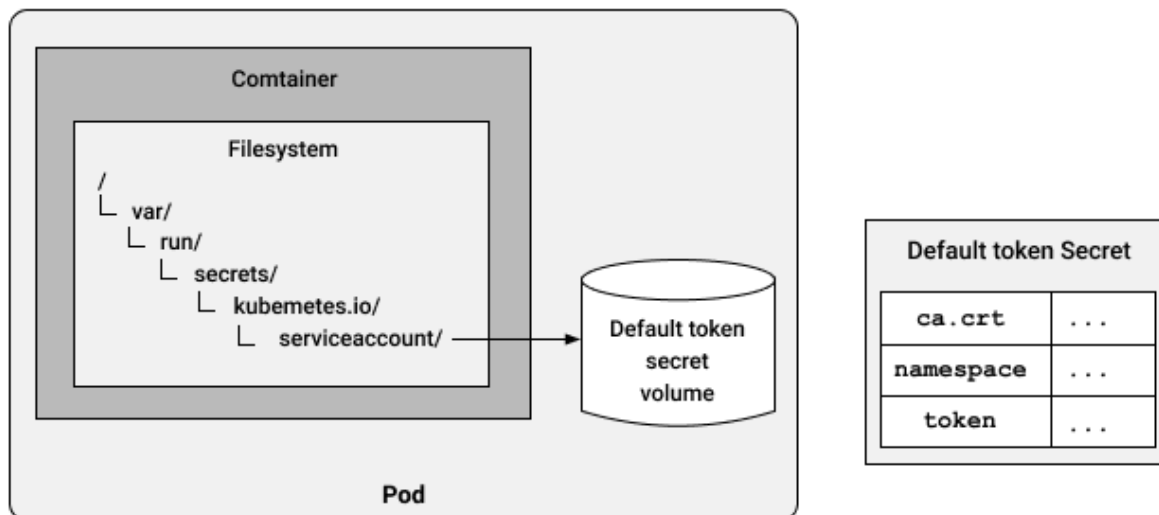
```
- name: foo
  mountPath: "/etc/foo"
  readOnly: true
volumes:
- name: foo
  secret:
    secretName: mysecret
```

Если в секрете содержится несколько значений, можно выборочно использовать требуемые:

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/etc/foo"
      readOnly: true
  volumes:
  - name: foo
    secret:
      secretName: mysecret
      items:
      - key: username
        path: my-group/my-username
```



Кстати, каждый pod по умолчанию получает секрет default-secret.



## Полезные ссылки:

- [Secrets in Kubernetes](#)
- [Знакомство с Kubernetes. Secrets](#)

## Задание:

1. Склонировать репозиторий [gocalc](#).
2. Написать Dockerfile для сборки контейнера с приложением из репозитория.
3. Собрать контейнер и загрузить его в [registry.rebrainme.com](#) (для авторизации используйте данные от [gitlab.rebrainme.com](#)).
4. Написать deployment для собранного приложения gocalc, загрузка docker-image должна производиться из [registry.rebrainme.com](#), с авторизацией. Аутентификационные данные должны быть созданы с помощью секрета.
5. Для проверки задания предоставьте deployment и secret в yaml-формате.