

Конфигурация приложений

ConfigMaps, Environment, Secrets



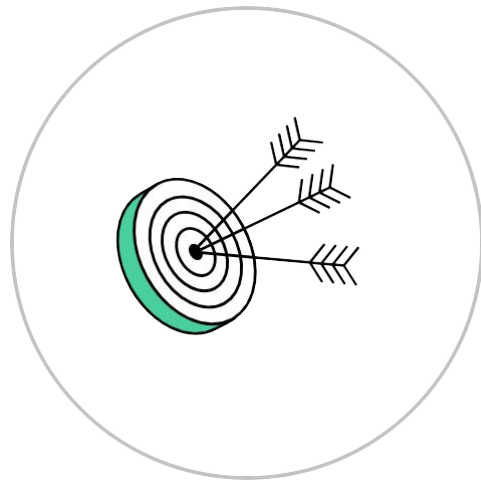
Кирилл Касаткин

DevOps-инженер, Renuе



Цели занятия

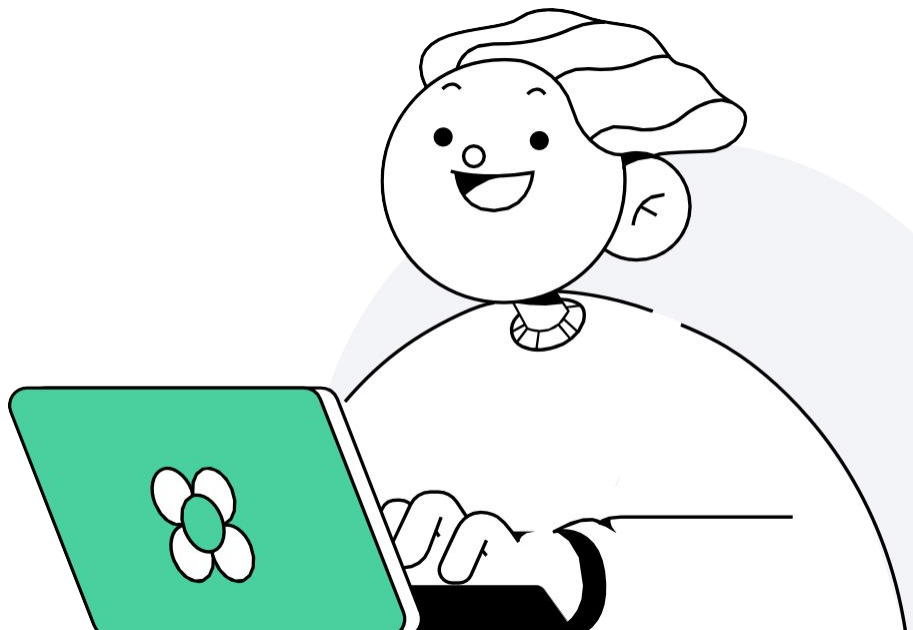
- Узнать:
 - что такое ConfigMap, Secret и чем они отличаются
 - как разделить и хранить конфигурации от Pod'ов
- Познакомиться со способами применения конфигураций
- Разобрать примеры манифестов объектов K8s



План занятия

- 1 Конфигурация приложения
- 2 ConfigMaps
- 3 Secrets
- 4 Итоги
- 5 Домашнее задание


*Нажми на нужный раздел для перехода



Конфигурация приложения



1



**Конфигурация приложения —
это возможность
динамического добавления
параметров приложению**

Конфигурация приложения

Принято разделять конфигурационные файлы и контейнеры с приложениями для того, чтобы избавиться от необходимости упаковывать конфиги в image приложения и обеспечить отказоустойчивость и безопасность.

В зависимости от содержимого есть 2 типа конфигураций :

- 1 ConfigMap
- 2 Secret

ConfigMaps



2



**ConfigMaps — объект k8s,
который позволяет хранить
нечувствительные
конфигурации в формате
ключ:значение**

Пример конфигурации

Пример конфигураций ConfigMap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
  namespace: my-ns
data:
  key1: value1
  key2: value2
  key3:
    subkey: somevalue
  key4: |
    Test
    multiple lines
    more lines
```

- metadata — имя и namespace



Пример конфигурации

Пример конфигураций ConfigMap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
  namespace: my-ns
data:
  key1: value1
  key2: value2
  key3:
    subkey: somevalue
  key4: |
    Test
    multiple lines
    more lines
```

- metadata — имя и namespace
- Данные могут быть в виде
 - ключ: значение



Пример конфигурации

Пример конфигураций ConfigMap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
  namespace: my-ns
data:
  key1: value1
  key2: value2
  key3:
    subkey: somevalue
  key4: |
    Test
    multiple lines
    more lines
```

- metadata — имя и namespace
- Данные могут быть в виде
 - ключ: значение
 - ключ: вложенные (ключ:значение)



Пример конфигурации

Пример конфигураций ConfigMap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
  namespace: my-ns
data:
  key1: value1
  key2: value2
  key3:
    subkey: somevalue
  key4: |
    Test
    multiple lines
    more lines
```

- metadata - имя и namespace
- Данные могут быть в виде
 - ключ: значение
 - ключ: вложенные (ключ:значение)
 - ключ: текст



Пример конфигурации

Подключение ConfigMap к Pod происходит несколькими способами: **ENV**

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
  namespace: my-ns
data:
  key1: value1
  key2: value2
  key3:
    subkey: somevalue
  key4: |
    Test
    multiple lines
    more lines
```

```
apiVersion: v1
kind: Pod
metadata:
  name: env-pod
  namespace: my-ns
spec:
  containers:
    - name: busybox
      image: busybox
      env:
        - name: CONFIGMAPVAR
          valueFrom:
            configMapKeyRef:
              name: my-configmap
              key: key1
```



Пример конфигурации

Подключение ConfigMap к Pod происходит несколькими способами: **Volume**

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
  namespace: my-ns
data:
  key1: value1
  key2: value2
  key3:
    subkey: somevalue
  key4: |
    Test
    multiple lines
    more lines
```

```
apiVersion: v1
kind: Pod
metadata:
  name: vol-pod
  namespace: my-ns
spec:
  containers:
    - name: busybox
      image: busybox
      volumeMounts:
        - name: configmap-volume
          mountPath: /etc/config/configmap
  volumes:
    - name: configmap-volume
      configMap:
        name: my-configmap
```



Secrets



3



**Secret — объект k8s, который
позволяет хранить
конфиденциальные
конфигурации в формате
ключ:значение**

Пример конфигурации

Пример конфигураций Secret

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
  namespace: my-ns
type: Opaque
data:
  username: <base64 String 1>
  password: <base64 String 2>
```

- metadata — имя и namespace
- значение должно храниться в кодировке base64



Пример конфигурации

Подключение Secret к Pod происходит несколькими способами: **ENV**

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
  namespace: my-ns
type: Opaque
data:
  username: <base64 String 1>
  password: <base64 String 2>
```

```
apiVersion: v1
kind: Pod
metadata:
  name: env-pod
  namespace: my-ns
spec:
  containers:
    - name: busybox
      image: busybox
      env:
        - name: SECRETVAR
          valueFrom:
            secretKeyRef:
              name: my-secret
              key: username
```



Пример конфигурации

Подключение Secret к Pod происходит несколькими способами: **Volume**

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
  namespace: my-ns
type: Opaque
data:
  username: <base64 String 1>
  password: <base64 String 2>
```

```
apiVersion: v1
kind: Pod
metadata:
  name: vol-pod
  namespace: my-ns
spec:
  containers:
    - name: busybox
      image: busybox
      volumeMounts:
        - name: secret-volume
          mountPath: /etc/config/secret
  volumes:
    - name: secret-volume
      secret:
        secretName: my-secret
```



Пример конфигурации

Типы конфигураций Secret

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
  namespace: my-ns
type: Opaque
data:
  username: <base64 String 1>
  password: <base64 String 2>
```

- **opaque (generic)** – определяемый пользователем
- **tls** – секрет из пары открытого и закрытого ключей
- **docker-registry** – секрет для доступа к хранилищу образов Docker
- и другие



Пример конфигурации

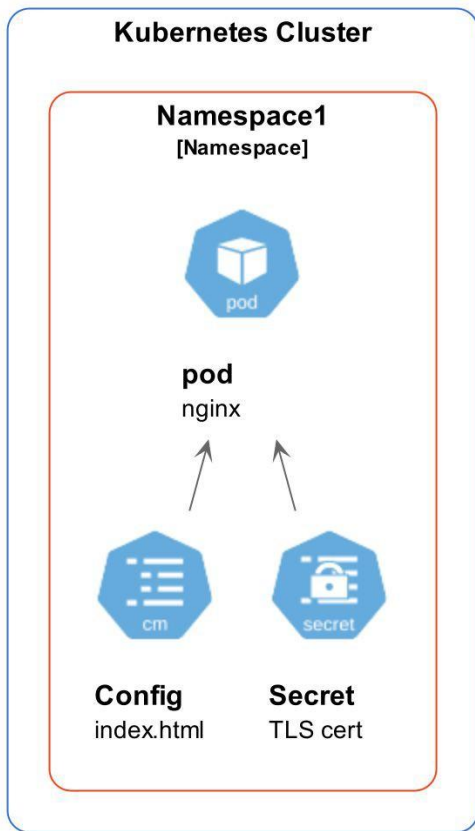
Примеры конфигураций Ingress TLS

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: my-app.com
    http:
      paths:
      - path: /
        backend:
          service:
            name: my-app-service
            port:
              number: 80
  tls:
  - hosts:
    - my-app.com
    secretName: my-app-secret-tls
```

```
apiVersion: v1
kind: Secret
metadata:
  name: my-app-secret-tls
data:
  tls.crt: base64 encode cert
  tls.key: base64 encode key
type: kubernetes.io/tls
```



Использование Secret и ConfigMap

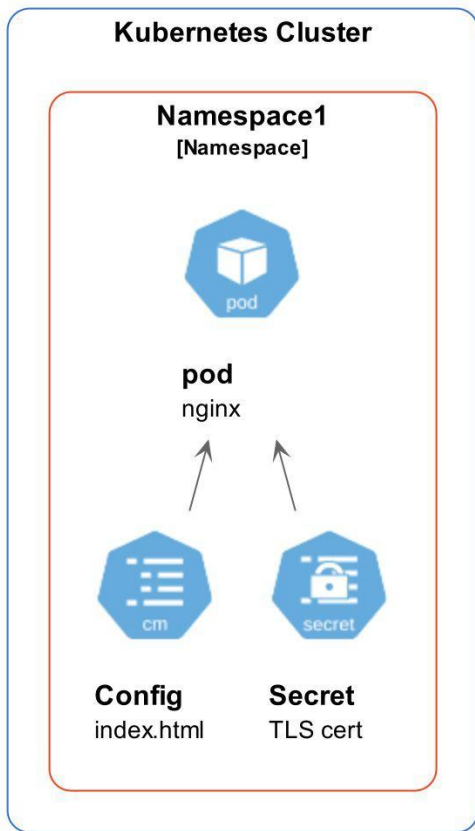


13.3

```
apiVersion: v1
kind: Pod
metadata:
  name: vol-pod
  namespace: my-ns
spec:
  containers:
  - name: busybox
    image: busybox
    env:
    - name: CONFIGMAPVAR
      valueFrom:
        configMapKeyRef:
          name: my-configmap
          key: key1
    - name: SECRETVAR
      valueFrom:
        secretKeyRef:
          name: my-secret
          key: username
```



Использование Secret и ConfigMap



```
apiVersion: v1
kind: Pod
metadata:
  name: vol-pod
  namespace: my-ns
spec:
  containers:
  - name: busybox
    image: busybox
    volumeMounts:
    - name: configmap-volume
      mountPath: /etc/config/configmap
    - name: secret-volume
      mountPath: /etc/config/secret
  volumes:
  - name: configmap-volume
    configMap:
      name: my-configmap
  - name: secret-volume
    secret:
      secretName: my-secret
```



Особенности ConfigMap и Secret

- Должны быть созданы до того, как они будут использованы в модулях.
Ссылки на несуществующие объекты предотвратят запуск Pod



Особенности ConfigMap и Secret

- Должны быть созданы до того, как они будут использованы в модулях. Ссылки на несуществующие объекты предотвратят запуск Pod
- Смонтированные конфигурации обновляются *автоматически*, но при этом не все приложения умеют перечитывать настройки, а также в случае переменных среды требуется перезапуск модуля



Особенности ConfigMap и Secret

- Должны быть созданы до того, как они будут использованы в модулях. Ссылки на несуществующие объекты предотвратят запуск Pod
- Смонтированные конфигурации обновляются *автоматически*, но при этом не все приложения умеют перечитывать настройки, а также в случае переменных среды требуется перезапуск модуля
- Динамические настройки зависят от множества переменных. Возможно применение шаблонизатора (например, Jinja)



Особенности ConfigMap и Secret

- Должны быть созданы до того, как они будут использованы в модулях. Ссылки на несуществующие объекты предотвратят запуск Pod
- Смонтированные конфигурации обновляются *автоматически*, но при этом не все приложения умеют перечитывать настройки, а также в случае переменных среды требуется перезапуск модуля
- Динамические настройки зависят от множества переменных. Возможно применение шаблонизатора (например, Jinja)
- Размер не более 1MB. Если в base64, то не более 750kB



Особенности ConfigMap и Secret

- Должны быть созданы до того, как они будут использованы в модулях. Ссылки на несуществующие объекты предотвратят запуск Pod
- Смонтированные конфигурации обновляются *автоматически*, но при этом не все приложения умеют перечитывать настройки, а также в случае переменных среды требуется перезапуск модуля
- Динамические настройки зависят от множества переменных. Возможно применение шаблонизатора (например, Jinja)
- Размер не более 1MB. Если в base64, то не более 750kB
- Создание множества мелких конфигураций может истощить память



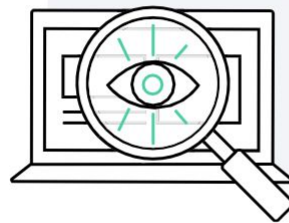
Особенности ConfigMap и Secret

- Должны быть созданы до того, как они будут использованы в модулях. Ссылки на несуществующие объекты предотвратят запуск Pod
- Смонтированные конфигурации обновляются *автоматически*, но при этом не все приложения умеют перечитывать настройки, а также в случае переменных среды требуется перезапуск модуля
- Динамические настройки зависят от множества переменных. Возможно применение шаблонизатора (например, Jinja)
- Размер не более 1MB. Если в base64, то не более 750kB
- Создание множества мелких конфигураций может истощить память
- Находятся в пространстве имён, что означает доступ только из того же пространства имён



Демонстрация работы

Работа с ConfigMap и Secret



Итоги

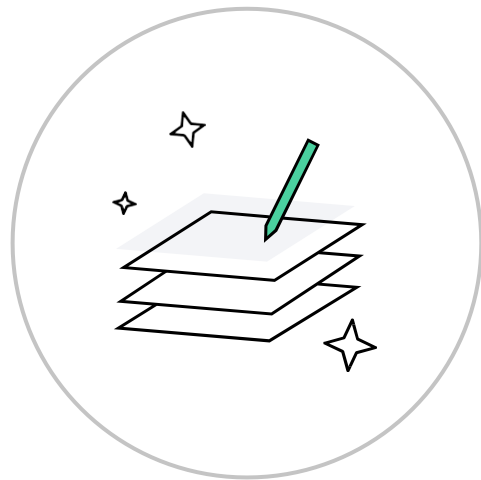
- 1 Узнали, что такое ConfigMap и Secret и чем они отличаются
- 2 Разобрались со способами подключения и применения конфигураций
- 3 Поняли, какие есть ограничения использования такого подхода
- 4 Рассмотрели примеры манифестов объектов K8s
- 5 Попробовали подключиться к кластеру и посмотреть в работе объекты, изученные на занятии



Домашнее задание

Давайте посмотрим ваше домашнее задание

- 1 Вопросы о домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



**Задавайте вопросы
и пишите отзыв о лекции**

