Переносимые сервисы

- Предоставление доступа переносимым сервисам
- Доступ к DB секретам

Предоставление доступа переносимым сервисам

По умолчанию, переносимые сервисы не имеют никакого доступа к кластеру. Чтобы решить данную проблему, необходимо выполнить следующую команду:

```
kubectl create clusterrolebinding default-pod --clusterrole cluster-admin --
serviceaccount=elma365-applets:default
```

где elma365-applets по умолчанию namespace для переносимых сервисов (может быть изменен на свой в values-elma365.yaml)

> Данный способ является **сильно** не безопасным. Крайне не рекомендуется его использовать в продуктивной среде.

Например, для библиотеки TechSU.ELMA365 иногда требуется понять, в каком namespace находится сама ELMA365. Для этого предлагается более безопасный способ:

```
cat << EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: elma365-applets
  name: namespace-reader
rules:
- apiGroups: [""]
  resources: ["namespaces"]
  verbs: ["get"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  namespace: elma365-applets
  name: namespace-reader-binding
subjects:
- kind: ServiceAccount
```

```
  name: default
  namespace: elma365-applets
roleRef:
  kind: Role
  name: namespace-reader
  apiGroup: rbac.authorization.k8s.io
EOF
```

Доступ к DB секретам

По умолчанию, переносимые сервисы расположены в отличном от основных сервисов ELMA365 namespace. Поэтому не имеют доступа к секретам.

Для предоставления доступа а так же синхронизации при изменении значений, необходимо выполнить следующие действия:

```
kubectl get secret elma365-db-connections -n default -o json > secret.json
cat secret.json | jq
'del(.metadata.namespace,.metadata.resourceVersion,.metadata.uid,.metadata.creationTimestamp,.
metadata.selfLink)' > secret-clean.json
kubectl apply -f secret-clean.json -n elma365-applets
```

Где:
-n **default** - namespace где установлена EMLA365,
-n **elma365-applets** - namespace переносимых сервисов

После этого надо выполнить действия по проброcу секретов в наш сервис в ENV, на примере как расписано тут и тут.

Если нужен прямой доступ из вашего сервиса, необходимо применить:

```
cat << EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: elma365-applets
  name: secret-reader
rules:
- apiGroups: [""]
  resources: ["secrets"]
  verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  namespace: elma365-applets
  name: secret-reader-binding
```

```
subjects:
- kind: ServiceAccount
  name: default
  namespace: elma365-applets
roleRef:
  kind: Role
  name: secret-reader
  apiGroup: rbac.authorization.k8s.io
EOF
```

После этого можно использовать библиотеки доступа в Kebernates в коде переносимого сервиса, чтобы получить данные из секретов.