

On-Premises

- Проблемы и их решение
 - Не публикуется процесс с таймером на стартовом событии
 - Отсутствуют метрики для переносимых сервисов
- Переносимые сервисы
 - Предоставление доступа переносимым сервисам
 - Доступ к DB секретам

Проблемы и их решение

Не публикуется процесс с таймером на стартовом событии

Внезапно перестали публиковаться такие процессы. Если запуск по расписанию убрать, то все нормально публикуется.

В противном случае окно публикации версии не пропадает (хоть и проверка проходит успешно), в network internal error на PUT шаблона процесса.

После глубокого изучения проблемы выяснилось, что по какие то неведанным причинам в таблице shchedule_srttings имеется две одинаковых записи вместо одной:

The screenshot shows a system interface with a list of schedule settings on the left and a data table on the right. The list on the left includes items like 'rmg_payments:incoming_invoices@buyer', 'rmg_payments:monthly_payments', and 'schedule_settings'. The table on the right has columns for 'lunch_time_from', 'lunch_time_to', 'working_time_to', 'working_time_from', and 'weekends'. It contains two identical rows of data.

	lunch_time_from integer	lunch_time_to integer	working_time_to integer	working_time_from integer	weekends integer
1	43200	46800	64800	32400	65
2	43200	46800	64800	32400	65

Удаляем одну запись (почему-то удаляется и вторая тоже автоматом, но это уже не важно).

Заходим в Администрирование - Рабочий календарь и нажимаем Сохранить: в таблице теперь как и нужно одна запись.

После этого процессы с запуском по расписанию начинают публиковаться.

Проблемы и их решение

Отсутствуют метрики для переносимых сервисов

Возможно не установлен сервер метрик для kubernetes.

Информация тут: <https://github.com/kubernetes-sigs/metrics-server>

Установка чарта: [kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml](https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml)

Переносимые сервисы

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:

```
#!/usr/bin/env bash  
  
cat << EOF | kubectl apply -f -  
apiVersion: rbac.authorization.k8s.io/v1  
kind: ClusterRole  
metadata:  
  name: elma365applets-cluster-role  
rules:  
  - apiGroups: [""]  
    resources: ["namespaces"]  
    verbs: ["get", "list", "watch"]  
  ---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: ClusterRoleBinding  
metadata:  
  name: elma365applets-cluster-role-binding
```

subjects:

- kind: ServiceAccount
name: default
namespace: elma365-applets

roleRef:

kind: ClusterRole
name: elma365applets-cluster-role
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 пробросу секретов в наш сервис в ENV, на примере как расписано [тут](#) и [тут](#).

Либо непосредственно использовать библиотеки доступа в Keubernates в коде переносимого сервиса, чтобы получить данные из секретов.