

Общее

В данном руководстве будет описан процесс обновления Deckhouse CSE с версии v1.58.2 на версию v1.64.0

Обновление

Перед обновлением должен быть один `clusteralert` о том, что deckhouse не находится на релизном канале. Это нормально, так как в редакции Deckhouse CSE нет релизных каналов. Алерт выглядит следующим образом:

```
$ kubectl get clusteralerts.deckhouse.io
NAME                                SEVERITY  AGE    LAST RECEIVED
STATUS
1d2c2f7d69f69f4b    D8DeckhouseIsNotOnReleaseChannel    9      24h    58s
firing
```

Перед обновлением необходимо выключить модуль `node-local-dns`:

```
$ kubectl apply -f - << EOF
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: node-local-dns
spec:
  enabled: false
EOF
```

Чтобы обновить deckhouse с версии 1.58.2 на версию 1.64.0 необходимо поменять тег образа в deployment deckhouse следующей командой

```
$ kubectl set image deployment -n d8-system deckhouse deckhouse=registry-
cse.deckhouse.ru/deckhouse/cse:v1.64.0
```

После обновления под `deckhouse` не сможет запуститься из-за ошибки прав доступа в директорию. В логах можно увидеть следующие строки

```
$ kubectl logs -n d8-system deckhouse-86699db77c-4pk6f -c init-external-modules
mkdir: can't create directory '/deckhouse/external-modules/modules': Permission denied
```

Чтобы это исправить, необходимо пропатчить ресурс deployment deckhouse

```
$ kubectl edit deployment deckhouse -n d8-system
```

Нужно удалить один контейнер из блока `initContainer`. Он входит в массив `initContainers` и выглядит следующим образом. Эти строки необходимо удалить и сохранить манифест:

```
- command:
- sh
- -c
- mkdir -p /deckhouse/external-modules/modules && chown -hR 64535 /deckhouse/external-
modules
```

```
/deckhouse/external-modules/modules && chmod 0700 /deckhouse/external-modules
/deckhouse/external-modules/modules
image: registry-
cse.deckhouse.ru/deckhouse/cse@sha256:e6a0f01d5fd33a2f1042ba5b1808b7eb3140c93302bd22a9291b5
910fc15eeae
imagePullPolicy: Always
name: init-external-modules
resources: {}
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /deckhouse/external-modules
  name: external-modules
```

Если после изменения поды deckhouse продолжают находиться в статусе `Terminating`, то необходимо принудительно их удалить

```
kubectl scale -n d8-system deployment deckhouse --replicas 0
```

Затем необходимо выполнить scale до исходного количества подов в кластере:

```
kubectl scale -n d8-system deployment deckhouse --replicas 3
```

После этого необходимо дождаться пока выполнится очередь. Посмотреть задачи в очереди можно следующей командой:

Для одной мастер ноды:

```
$ kubectl -n d8-system exec -it deploy/deckhouse -c deckhouse -- deckhouse-controller queue
list
```

Для мультимастер инсталляции:

```
$ kubectl -n d8-system exec svc/deckhouse-leader -c deckhouse -- deckhouse-controller queue
list
```

Если до обновления в кластере использовалась `Grafana`, то в очереди появится ошибка, связанная с созданием `ingress` для Grafana:

```
1. ModuleRun:main:prometheus:doStartup:OperatorStartup:failures 1:helm upgrade failed:
failed to create resource: admission webhook "main.validate.d8-ingress-nginx-deckhouse"
denied the request: host "grafana.158.160.58.19.sslip.io" and path "/" is already defined
in ingress d8-monitoring/grafana
```

Удаляем старый `ingress`

```
$ kubectl delete ingress -n d8-monitoring grafana
```

Далее включаем модуль `prometheus-metrics-adapter`:

```
$ kubectl apply -f - << EOF
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: prometheus-metrics-adapter
```

```
spec:  
  enabled: true  
EOF
```

Чтобы убедиться, что обновление прошло успешно необходимо проверить следующие моменты:

1. Очередь пустая

Для одной мастер ноды:

```
$ kubectl -n d8-system exec -it deploy/deckhouse -c deckhouse -- deckhouse-controller  
queue list
```

Для мультимастер инсталляции:

```
$ kubectl -n d8-system exec svc/deckhouse-leader -c deckhouse -- deckhouse-controller  
queue list
```

2. Все поды в namespace `d8-system` в статусе `Running`

```
$ kubectl get po -n d8-system  
NAME                                READY   STATUS    RESTARTS   AGE  
deckhouse-897dfcc55-gd66s          2/2    Running   0           38m  
webhook-handler-6844bbb8bc-b8llg    1/1    Running   0           37m
```

Стоит заметить, что модуль `node-local-dns` не работает с `CNI Flannel` на версии `Deckhouse CSE 1.64.0`. Поэтому перед обновлением нужно выключить модуль `node-local-dns`, а после обновления на `1.64` необходимо переключать `CNI` драйвер на `Cilium` и только потом включать модуль `node-local-dns`