Kubernetes : namespace-scoped deployments

Prepare the cluster - Admin side

Installing the Helm Chart Repositories

The first step is to start on a machine with helm v3 installed and configured with your kubernetes cluster and pull down the two charts that you will need.

First, let's add the element-updater repository to helm:

```
helm repo add element-updater https://registry.element.io/helm/element-updater --username <ems_image_store_username> --password '<ems_image_store_token>'
```

Replace `ems_image_store_username` and `ems_image_store_token` with the values provided to you by Element.

Secondly, let's add the element-operator repository to helm:

```
helm repo add element-operator https://registry.element.io/helm/element-operator --username <ems_image_store_username> --password '<ems_image_store_token>'
```

Replace `ems_image_store_username` and `ems_image_store_token` with the values provided to you by Element.

Now that we have the repositories configured, we can verify this by:

```
helm repo list
```

and should see the following in that output:

```
NAME                URL
element-operator    https://registry.element.io/helm/element-operator
```

```
element-updater         https://registry.element.io/helm/element-updater
```

Deploy the CRDs

Write the following `values.yaml` file :

```
clusterDeployment: true
deployCrds: true
deployCrdRoles: true
deployManager: false
```

To install the CRDS with the helm charts, simply run :

```
helm install element-updater element-updater/element-updater -f values.yaml
helm install element-operator element-operator/element-operator -f values.yaml
```

Now at this point, you should have the following two CRDs available:

```
[user@helm ~]$  kubectl get crds | grep element.io
elementwebs.matrix.element.io                    2023-10-11T13:23:14Z
wellknowndelegations.matrix.element.io           2023-10-11T13:23:14Z
elementcalls.matrix.element.io                   2023-10-11T13:23:14Z
hydrogens.matrix.element.io                      2023-10-11T13:23:14Z
mautrixtelegrams.matrix.element.io               2023-10-11T13:23:14Z
sydents.matrix.element.io                        2023-10-11T13:23:14Z
synapseusers.matrix.element.io                   2023-10-11T13:23:14Z
bifrosts.matrix.element.io                       2023-10-11T13:23:14Z
lowbandwidths.matrix.element.io                  2023-10-11T13:23:14Z
synapsemoduleconfigs.matrix.element.io           2023-10-11T13:23:14Z
matrixauthenticationservices.matrix.element.io   2023-10-11T13:23:14Z
ircbridges.matrix.element.io                     2023-10-11T13:23:14Z
slidingsyncs.matrix.element.io                   2023-10-11T13:23:14Z
securebordergateways.matrix.element.io           2023-10-11T13:23:14Z
hookshots.matrix.element.io                      2023-10-11T13:23:14Z
matrixcontentscanners.matrix.element.io          2023-10-11T13:23:14Z
sygnals.matrix.element.io                        2023-10-11T13:23:14Z
sipbridges.matrix.element.io                     2023-10-11T13:23:14Z
```

```
livekits.matrix.element.io                    2023-10-11T13:23:14Z
integrators.matrix.element.io                 2023-10-11T13:23:14Z
jitsis.matrix.element.io                      2023-10-11T13:23:14Z
mautrixwhatsapps.matrix.element.io            2023-11-15T09:03:48Z
synapseadminuis.matrix.element.io             2023-10-11T13:23:14Z
synapses.matrix.element.io                    2023-10-11T13:23:14Z
groupsyncs.matrix.element.io                  2023-10-11T13:23:14Z
pipes.matrix.element.io                       2023-10-11T13:23:14Z
elementdeployments.matrix.element.io          2023-10-11T13:34:25Z
chatterboxes.matrix.element.io                2023-11-21T15:55:59Z
```

Namespace-scoped role

In the namespace where the ESS deployment will happen, to give a user permissions to deploy ESS, please create the following role and roles bindings :

User role :

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: ess-additional
rules:
- apiGroups:
  - apiextensions.k8s.io
  resources:
  - customresourcedefinitions
  verbs:
  - list
  - watch
  - get
- apiGroups:
    - project.openshift.io
  resources:
    - projects
  verbs:
    - get
    - list
```

```
    - watch
```

User roles bindings :

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: ess-additional
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: ess-additional
subjects:
<role subjects which maps to the user or its groups>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: ess
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: edit
subjects:
<role subjects which maps to the user or its groups>
```

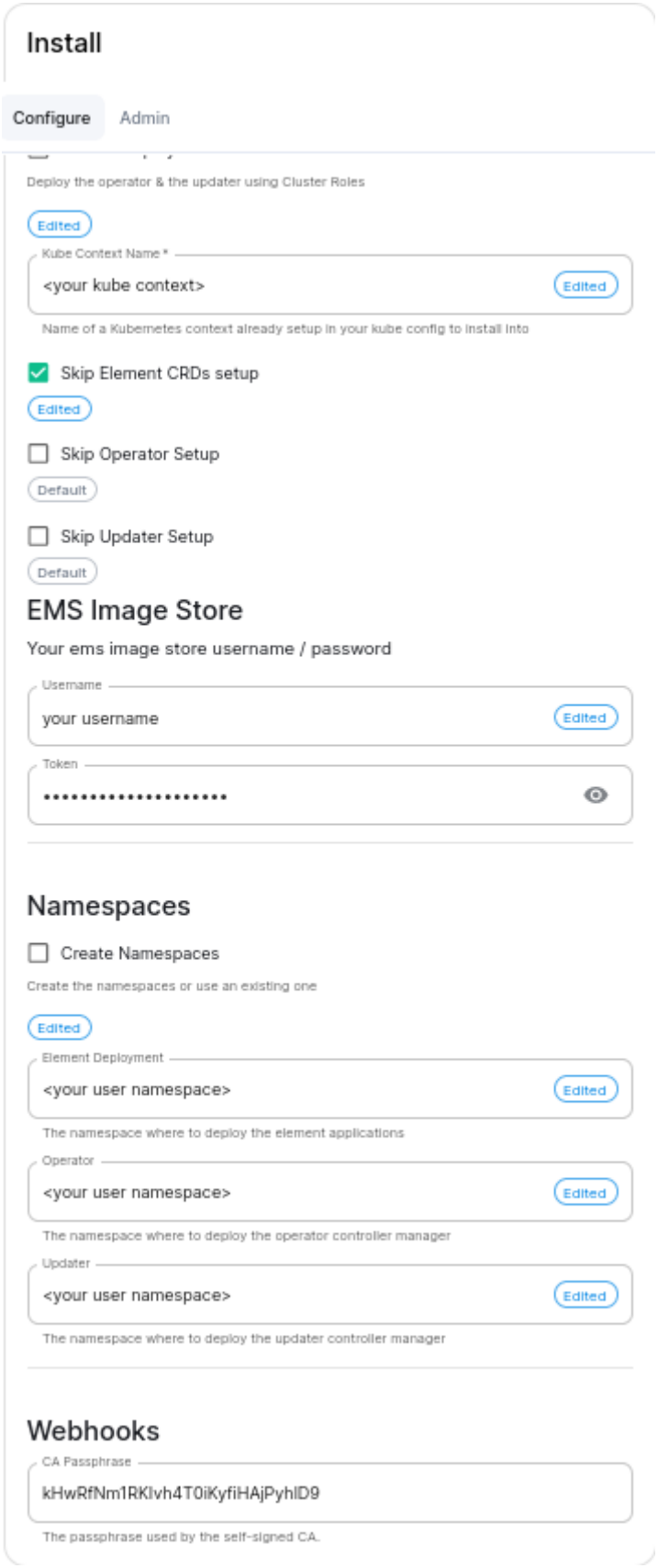
Using the installer in namespace-scoped mode

In the installer UI, on the cluster configuration screen, the user can now use the following values :

- Skip Operator Setup: unchecked
- Skip Updater Setup: unchecked
- Skip Element Crds Setup: checked
- Cluster Deployment: unchecked
- Kube Context Name:

- namespaces:
  - Create Namespaces: unchecked
  - operator:
  - updater: <same as operator, namespace to deploy ess>
  - element deployment: <same as operator, namespace to deploy ess>

See the following screenshot which describes the options. The webhooks CA passphrase value will be present in version 23.11.X and above, and will be randomly-generated on your deployment :



Revision #12
Created 13 October 2023 13:34:53 by Gaël Goinvic
Updated 6 November 2024 13:20:25 by Kieran Mitchell Lane