Using ESS with a local registry in air-gapped networks

Issue

- In some air-gapped environments, rather than downloading the air-gapped artifacts from the Element EMS service, it may be prefereble to setup your local repository to download the artifacts automatically and make them available to the air-gapped K8s cluster. This article explains the steps required to enable this.

Environment

- Element Server Suite (ESS) On-Premise
- Air-gapped Networks

Resolution

The steps are:

- Configure your repository to link to all remote repositories required by ESS.
- Create a new configmap definition with the required artifacts pointing to your local registry and inject it into the cluster.
- Configure the installer to use the new Images Digests Config Map.
- Generate secret for local registry (if it requires authentication) and add it to ESS.

ESS remote repositories

In a connected environment, the ESS Installer downloads artifacts from several remote repositories including:

- gitlab-registry.matrix.org *
- docker.io
- ghcr.io
- quay.io
- gcr.io

* requires authentication using the EMS Image Store Token and password.

In an air-gapped environment, there are two options to download the dependencies:

1. Download the air-gapped zipped repositories as described here

2. Configure a local repository to pull the artifacts from the remote repositories when required.

The `image_repository_path`, `image_repository_server` and `image_tag` for each artifact are defined in the `images_digests.yml` file supplied with the ESS Installer.

An example extract from that file (for auditbot and adminbot integrations) is below:

```
images_digests:
  adminbot:
    access_element_web:
      image_digest: sha256:1e7e1503ff73ae01af72dd27d9be387667dbbbcc53870ca02b3b02a1890696a9
      image_repository_path: ems-image-store/enterprise/access-token-elementweb
      image_repository_server: gitlab-registry.matrix.org
      image_tag: v1.11.45
    haproxy:
      image_digest: sha256:75fc37d6372cc22410bdd6ad0191a8dfeababef2a7d8d585694f21b73c2bb3d6
      image_repository_path: library/haproxy
      image_repository_server: docker.io
      image_tag: 2.8-alpine
    pipe:
      image_digest: sha256:fdef296a8cc72a6deba2f879ddc8cc94da8a49027c5da6627ababd100efa9e57
      image_repository_path: ems-image-store/enterprise/pipe
      image_repository_server: gitlab-registry.matrix.org
      image_tag: 2.7.0
  auditbot:
    access_element_web:
      image_digest: sha256:1e7e1503ff73ae01af72dd27d9be387667dbbbcc53870ca02b3b02a1890696a9
      image_repository_path: ems-image-store/enterprise/access-token-elementweb
      image_repository_server: gitlab-registry.matrix.org
      image_tag: v1.11.45
    haproxy:
      image_digest: sha256:75fc37d6372cc22410bdd6ad0191a8dfeababef2a7d8d585694f21b73c2bb3d6
      image_repository_path: library/haproxy
      image_repository_server: docker.io
      image_tag: 2.8-alpine
    pipe:
      image_digest: sha256:fdef296a8cc72a6deba2f879ddc8cc94da8a49027c5da6627ababd100efa9e57
      image_repository_path: ems-image-store/enterprise/pipe
      image_repository_server: gitlab-registry.matrix.org
      image_tag: 2.7.0
```

If you have a cluster admin role, you can include all the components in the `images_digests.yml` file.

If your cluster admins have already installed the ElementDeployment CRDs using the Helm Charts, you can omit the `Operator` and `Updater` sections from this file. If you are using an external PostgreSQL database, you can omit the `PostegreSQL` section.

Configuring local repository to download artifacts

You will need to configure your local repository to download the artifacts from the remote repositories. Once downloaded, they can be organised into folders, or added to the same base directory.

Creating the new Images Digests Config Map

In order to override the automatic download of the images from the remote repository by ESS during the install, you will need to inject a new ConfigMap which specifies the local repository to use for each artifact. To do that, you will need to inject a ConfigMap with the following example structure:

```
data:
  images_digests: |
    adminbot:
      access_element_web:
      haproxy:
      pipe:
    auditbot:
      access_element_web:
      haproxy:
      pipe:
    element_call:
      element_call:
      jwt:
      redis:
      sfu:
    element_web:
      element_web:
    groupsync:
      groupsync:
    hookshot:
      hookshot:
    integrator:
      appstore:
      integrator:
      modular_widgets:
    matrix_authentication_service:
      init:
```

```
      matrix_authentication_service:
    matrix_content_scanner:
      clam_anti_virus:
      icap:
      matrix_content_scanner:
    sliding_sync:
      api:
      poller:
    sydent:
      sydent:
    sygnal:
      sygnal:
    synapse:
      haproxy:
      redis:
      synapse:
    synapse_admin:
      synapse_admin:
    well_known_delegation:
      well_known_delegation:
```

Each container on this tree needs at least the following properties to override the source of download :

```
image_repository_path: elementdeployment/vectorim/element-web
image_repository_server: localregistry.local
```

You can also override the image tag and the image digest if you want to enforce using digests in your deployment :

```
image_digest: sha256:ee01604ac0ec8ed4b56d96589976bd84b6eaca52e7a506de0444b15a363a6967
image_tag: v0.2.2
```

For example, the required configmap manifest (e.g. `images_digest_configmap.yml`) format would be, to override the `element_web/element_web` container source path :

```
apiVersion: v1
kind: ConfigMap
metadata:
 name: <config map name>
 namespace: <namespace of your deployment>
data:
```

```
images_digests: |
  element_web:
    element_web:
      image_repository_path: elementdeployment/vectorim/element-web
      image_repository_server: localregistry.local
```

Notes:

- the `image_digest:` may need to be regenerated, or it can be removed.
- The `image_repository_path` needs to reflect the path in your local repository.
- The `image_repository_server` should be replaced with your local repository URL

The new ConfigMap can then be injected into the cluster with:

```
kubectl apply -f images_digest_configmap.yml -n <namespace of your deployment>
```

You will also need to configure the ESS Installer to use the new Images Digests Config Map by adding the <config map name> into the Cluster advanced section.

SECTIONS

🏠 Host

🌐 Domains

🔒 Certificates

✳️ Cluster

💬 Synapse

🖼️ Element Web

🛡️ Homeserver Admin

⇄ Integrator

🧩 Integrations

Cluster

Your Element Deployment runs on top of Kubernetes, a clustering software that isolates and manages your services.

Advanced ⌃

Secrets / Global ⌄

Show Values ⌄

Config

Images Digests Config Map

A configmap containing images digests metadata to override

☐ Support DNS Federation Delegation

Enable DNS Record delegation. In this mode, WellKnownDelegation is not deployed, and the domain name is served under Synapse ingress.

If your local repository requires authentication, you will need to create a new secret. So for example, if your local registry is called `localregistry` and the URL of the registry is `localregistry.local`, the command would be:

```
kubectl create secret docker-registry localregistry --docker-username=<localregistry user> --docker-password=<localregistry password> --docker-server=localregistry.local -n <your namespace>
```

The new secret can then be added into the ESS Installer GUI advanced cluster Docker Secrets:
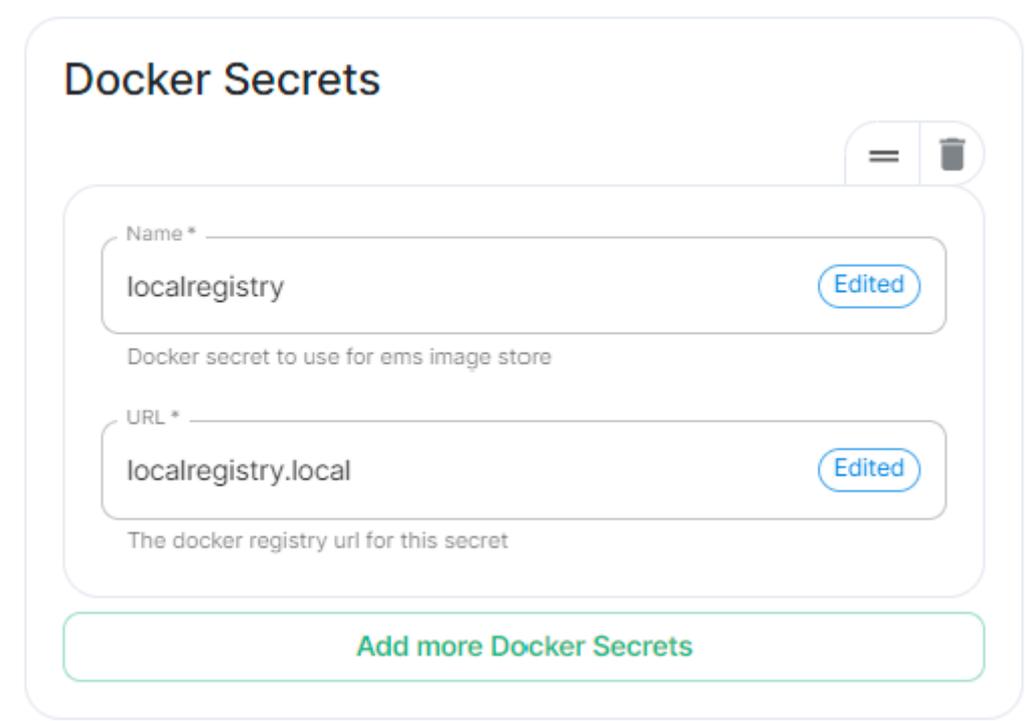
Handling new releases of ESS

If you are overriding image tags, you will need to make sure that your tags are available with the new releases of ESS. You can use a staging environment to tests the upgrades for example.

Some releases of ESS might require new containers, or might provide new components to deploy. In which case you will have to add them to your local registry to successfully deploy them.

Revision #36
Created 26 October 2023 11:33:39 by Simon Addis
Updated 6 November 2024 12:49:47 by Kieran Mitchell Lane