LDOS 1.2.0 install and config

This article outlines the installation and configuration process of the Lumada DataOps Suite 1.2.0 (GA).

Table of Contents

Before You Start

Prior to installing LDOS you need to check the following requirements:

- Kubernetes v1.18 cluster:
  - 4 nodes, each one with at least 4 CPUs / 16 GB of memory (**minimum**)
  - 300 GB of space for the default persistent volumes
- DNS hostname for the cluster
- OCI-compliant registry
  - Read more at: http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/InstallRegistry
- A NFS server is set up and configured
- Non-Debian/Ubuntu Linux machine (e.g. CentOS/Fedora/RHEL) with:
  - `docker v1.26.0+`
  - `kubectl v1.18.0+`
  - `jq 1.6+`
  - `yq 4+`
  - `7z, tar, unzip`

This guide will cover the installation of:

- Foundry 2.2.1
- Metrics Add-On 1.0.0 for Foundry
- LDOS 1.2.0

Downloads

All files required for installation are available in the release folder and can be found in the link below.

https://hcpanywhere.hitachivantara.com/a/QDDm7tWw5sBSMObK/4cc9f82c-08a8-4895-93d3-71bb0694374c?l

> You'll need your Hitachi Vantara credentials or ask Customer Success.

Kubernetes Management

To properly access the kubernetes cluster, you need to configure your `kubeconfig`.

Know where your `kubeconfig` is located - this is a YAML file that determines which cluster your `kubectl` will talk to. It is usually located under `.kube/config` at your home user folder. **You will need the path later**.

Double check that your `kubectl` is talking to the correct kubernetes cluster by running:

```
kubectl config view --minify | grep 'server\|current-context'
```

Read more at: https://kubernetes.io/docs/tasks/access-application-cluster/configure-access-multiple-clusters/

Install Foundry 2.2.1

Download the following package from the release folder:

- `Foundry-Control-Plane-2.2.1.tgz`

Follow the official installation guide for Foundry

Please refer to the official Foundry documentation for details on how to install Foundry: http://docs.foundry.wal.hds.com/docs/

Start with the prerequisites: http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites

⚠ Foundry 2.X.X no longer uses **openEBS** for its storage configuration. You have to have a default storage class before installing Foundry.

And follow through with the installation steps:

- Install the cluster services: http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices
- Add the custom resource definitions (CRDs): http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs
- Install the Foundry control plane: http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane

Access the Solution management UI

In order to access the Solution management UI, you need to get the password for the admin user `foundry`. **Keep this password for later**.

```
# get password for foundry user:
echo $(kubectl get keycloakusers -n hitachi-solutions keycloak-user -o
jsonpath='{.spec.user.credentials[0].value}')
```

Then log into the **Solution management** UI using a browser, replacing `<HOSTNAME>` by the cluster hostname:

https://`<HOSTNAME>`**/hitachi-solutions/hscp-hitachi-solutions/solution-control-plane/**

You should land on the *Installed Solutions* page, which contains a single card for the Solution Control Plane.

Upload Metrics Add-On

We are going to describe the necessary steps to upload the Metrics Add-On for the particular case of an LDOS installation. The LDOS install script will then install and configure the Metrics Add-On together with the LDOS solutions.

ℹ The Metrics Add-On is a generic extension for Foundry. It will be installed as a common solution in Foundry that will silently and transparently collect generic kubernetes metrics about any resource in the cluster.

Please refer to the official Metrics Add-On documentation for details and additional troubleshooting: http://docs.foundry.wal.hds.com/addons/metricsaddon/docs/1.0.0/

Download and unpack the Metrics Add-On

Download the following packages from the release folder:

- `metrics-addon-1.0.0.tgz`
- `Foundry-Solution-Upload-2.2.1.tgz`

Unpack the content of both archives:

```
tar -xf metrics-addon-1.0.0.tgz
mkdir -p tools
tar -C tools -xf Foundry-Solution-Upload-2.2.1.tgz
```

Upload the Metrics Add-On to Foundry

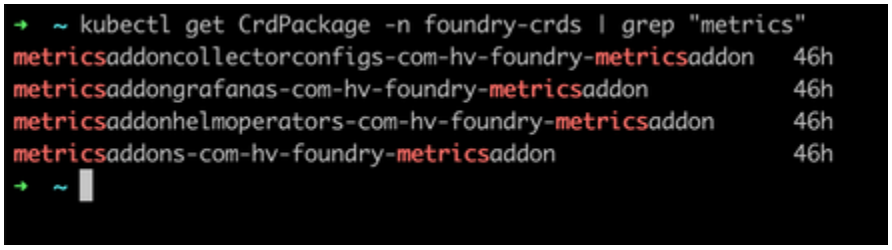
Add Metrics Add-On and Prometheus Operator CRD helm charts using `apply-crds.sh`, replacing `REGISTRY_URL` with a proper registry URL and `KUBECONFIG` by the `kubeconfig` for the cluster:

```
./tools/bin/apply-crds.sh -r <REGISTRY_URL> -C metrics-addon-1.0.0/crd-charts/ -k <KUBECONFIG> --insecure -x
```

The `--insecure` flag is used to allow an insecure registry. Please refer to the official Foundry documentation for details on the `apply-crds.sh` script: http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs/

Validate that CRD packages for Metrics Add-On were added:

```
kubectl get CrdPackage -n foundry-crds | grep "metrics"
```



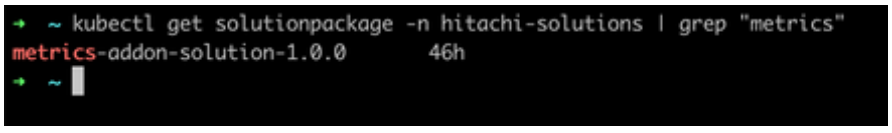
Upload the charts and images to the registry using `upload-solutions.sh`, replacing `KUBECONFIG` by the kubeconfig for the cluster:

```
./tools/bin/upload-solutions.sh -C metrics-addon-1.0.0/charts/ -I metrics-addon-1.0.0/images/ -k <KUBECONFIG> -n hitachi-solutions
```

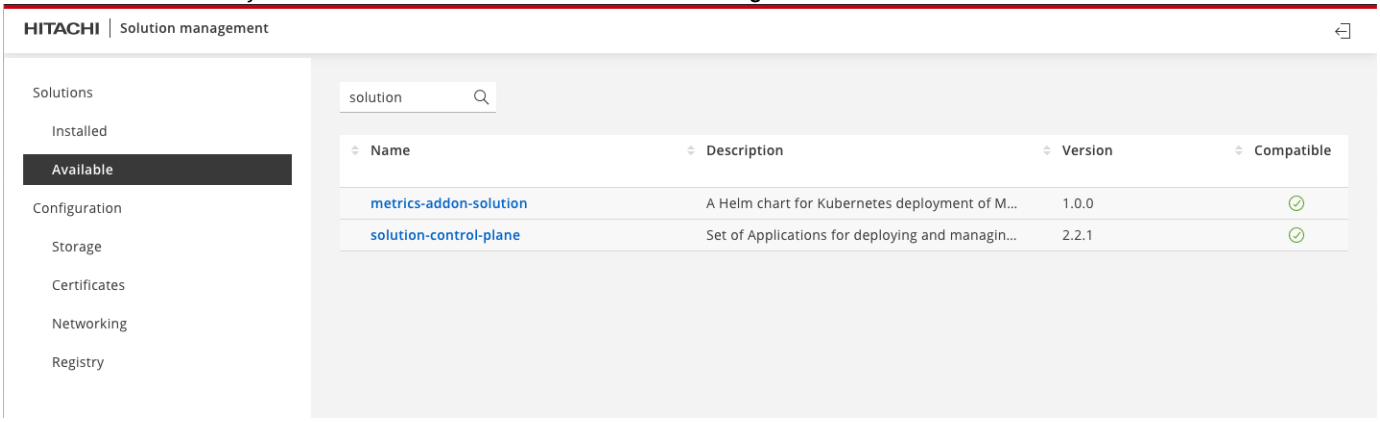
Please refer to the official Foundry documentation for details on the `upload-solutions.sh` script: http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/SolutionManagement/#using-upload-solutionssh

Validate that the solution package for Metrics Add-On was added:

```
kubectl get solutionpackage -n hitachi-solutions | grep "metrics"
```



You can also confirm that you have the solution available in the **Solution management** UI:



Troubleshooting

If you are working in a cluster with self-sign certificates, you might run into the following error when trying to apply the new CRDs:

```
docker: Error response from daemon: Get https://YOUR_CLUSTER_REGISTRY
/v2/: x509: certificate signed by unknown authority.
```

To work around this error you can perform the following actions in the machine where you are running these commands:

- Add the insecure registry to the list of insecure registries in the docker configuration
- Get the certificate from the cluster and install it in the store

```
# Get the certificate
# Replace YOUR_CLUSTER_REGISTRY and YOUR_CLUSTER_NAME accordingly
openssl s_client -showcerts -connect YOUR_CLUSTER_REGISTRY </dev/null 2>
/dev/null | openssl x509 -outform PEM > YOUR_CLUSTER_NAME-cert.pem

# Add the certificate to the store
# Use the same name of the cert file used in the previous command
certutil -user -addstore root YOUR_CLUSTER_NAME-cert.pem
```

Upload LDOS Solutions

Download the following packages from the release folder:

- `Lumada DataOps Suite Package 1.2.0/Lumada-DataOps-Suite-1.2.0.gz`
- `Lumada DataOps Suite Package 1.2.0/Lumada-DataOps-Suite-installer-1.2.0.zip`

Unpack the content of `Lumada-DataOps-Suite-1.2.0.gz`

```
7z e Lumada-DataOps-Suite-1.2.0.gz
tar -xf lumada-dataops-suite.tar
```

Unpack LDOS Installer `Lumada-DataOps-Suite-installer-1.2.0.zip`

```
unzip Lumada-DataOps-Suite-installer-1.2.0.zip
```

At the end you will have two directories: `/lumada-dataops-suite` and `/installer` folders:

- `/lumada-dataops-suite` Includes
  - `/images`
  - `/charts` with all the solution artefacts, and
  - `/control-plane` folder with scripts for uploading solutions to Foundry.
- `/installer` - Includes scripts required to prepare the upload and install LDOS solutions.

Patch the LDOS solutions charts

Some LDOS Solutions need to be patched to inject the hostname in the helm charts prior to being uploaded to Foundry.

> ⚠ Have caution with this step because it will modify helm charts and it will only work once. If for some reason, the hostname is not correct, you will have to go back, unpack the `lumada-dataops-suite.tar` to restore default helm charts and only then run `update-hostname.sh` again.

Run the following command replacing `<HOSTNAME>` with the cluster hostname:

```
./installer/update-hostname.sh -c=lumada-dataops-suite/charts -
h=<HOSTNAME>
```

Upload LDOS solutions packages to Foundry

Upload the charts and images to the registry using `upload-solutions.sh`, replacing `KUBECONFIG` by the kubeconfig for the cluster:

```
./lumada-dataops-suite/control-plane/bin/upload-solutions.sh -C lumada-
dataops-suite/charts/ -I lumada-dataops-suite/images/ -k <KUBECONFIG> -
n hitachi-solutions
```

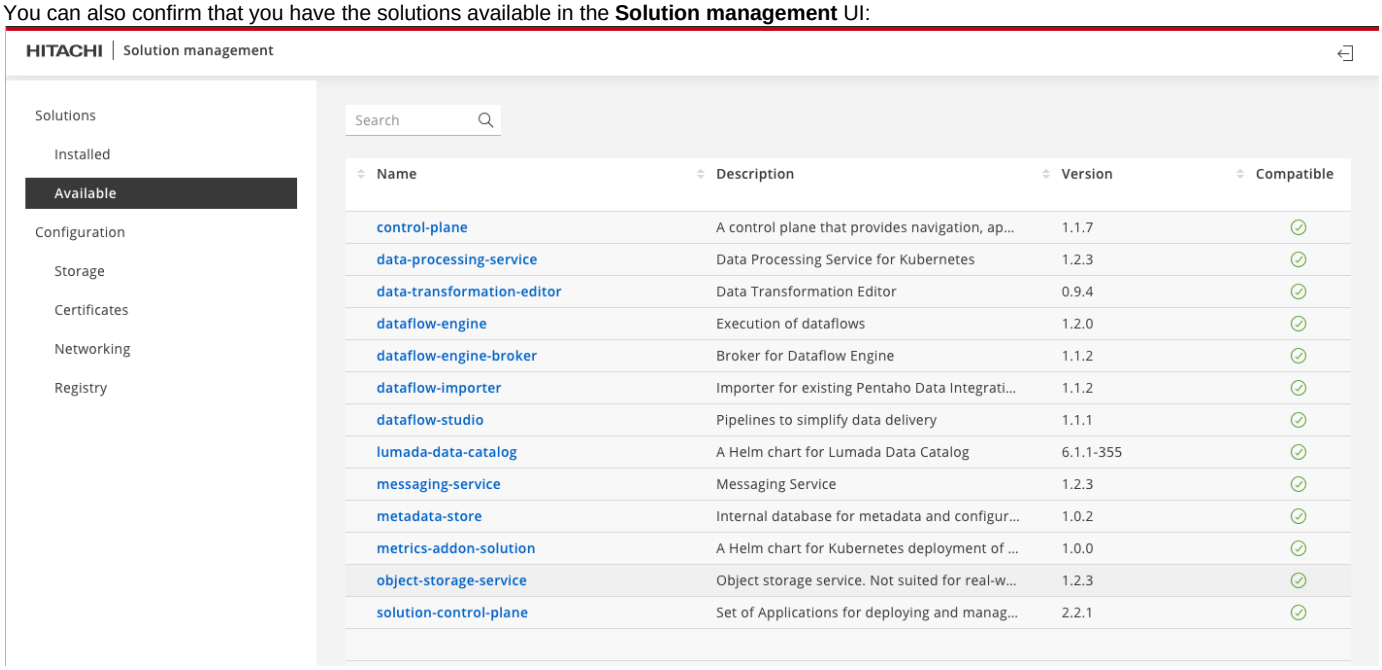
After running this command, you can validate if the solution packages were uploaded by doing:

```
kubectl get solutionpackage -n hitachi-solutions
```

> Note: The versions in the images might not match the actual versions

```
~ $ kubectl get solutionpackage -n hitachi-solutions
NAME                                AGE
control-plane-1.1.7                 41d
data-processing-service-1.2.3       41d
data-transformation-editor-0.9.4    2d19h
dataflow-engine-1.2.0               41d
dataflow-engine-broker-1.1.2        2d19h
dataflow-importer-1.1.2             2d19h
dataflow-studio-1.1.1               2d19h
lumada-data-catalog-6.1.1-355       2d19h
messaging-service-1.2.3             2d19h
metadata-store-1.0.2                2d19h
metrics-addon-solution-1.0.0        41d
object-storage-service-1.2.3        2d19h
solution-control-plane-2.2.1        2d21h
~ $
```

You can also confirm that you have the solutions available in the **Solution management** UI:



Install and Configure LDOS

Configure the properties file

Go to the `/installer` folder and modify the `env.properties` file for the cluster you are using:

```
# Cluster settings
hostname=
namespace=hitachi-solutions
realm=default
tls_mode=SIMPLE
protocol=https

# Foundry credentials used in the installation
foundry_client_name=solution-control-plane-sso-client
foundry_client_secret=
username=
password=

# NFS server settings
volume_host=
volume_path=
```

- *Cluster settings*
  - `hostname` - Hostname where the foundry instance is running, e.g. `dogfood.trylumada.com`
  - `namespace` - Namespace name, if different from the default namespace `hitachi-solutions`
  - `realm` - Keycloak realm, if different from the default Keycloak `default`
  - `tls_mode` - Ingress TLS mode for the cluster's routes. It can either be `SIMPLE`, `MUTUAL` or `NONE`.
  - `protocol` - Cluster's routes protocol. It can either be `http` (when tls mode is `NONE`) or `https` (when tls mode is `SIMPLE` or `MUTUAL`), depending on the Foundry App protocol.
- *Foundry credentials*
  - `foundry_client_name` - Foundry client id in Keycloak, if different from the default `solution-control-plane-sso-client`
  - `foundry_client_secret` - Foundry client secret in Keycloak
  - `username` - Username with admin permissions in Foundry, e.g. `foundry`
  - `password` - Password for the user with admin permissions
- *How to get the* `foundry_client_secret`

```
# get client secret for solution-control-plane-sso-client
echo $(kubectl get secrets/keycloak-client-secret-solution-control-
plane-sso-client -n hitachi-solutions --template={{.data.
CLIENT_SECRET}} | base64 --decode)
```

- *How to get the* `password` *for the user* `foundry`

```
# get password for foundry user:
echo $(kubectl get keycloakusers -n hitachi-solutions keycloak-user -o
jsonpath='{.spec.user.credentials[0].value}')
```

- *NFS volume settings*

LDOS needs to point to a NFS server to store files for the Data Transformation Editor, Dataflow Importer, and Dataflow Engine.

`volume_host` - NFS server host, e.g. `my-nfs-server.example.com`
`volume_path` - Path for the volume root folder in the NFS server, e.g. `/ldos-volume`

All these properties are case-sensitive.
The `env.properties` file also includes other properties that control the installation. You can, for example, do a partial installation of LDOS by changing the `install_mode` between `LDOS`, `LDI` and `LDC`. For more advanced settings see the included `README.md` file.

Run the install script

Go to the `/installer` folder and run the following command:

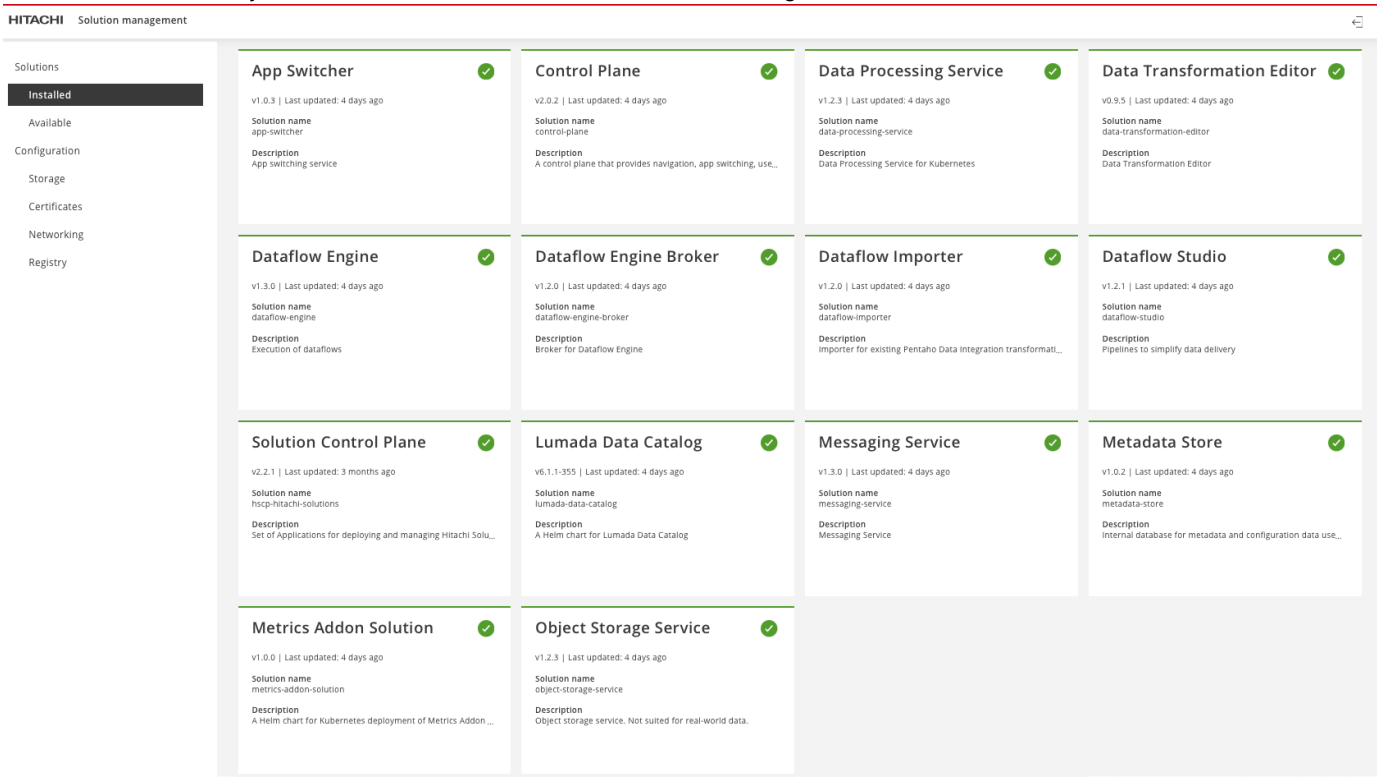
```
./install.sh
```

The script performs all the necessary steps to install and configure LDOS including default roles and users:

Role	User	Password	Brief Description
Administrator	cmoore	cmoore	Full access to LDOS, including to the Solution management and Keycloak
Data Engineer	bwayne	bwayne	Access to Dataflow operations, including to the Data Transformation Editor Access to the Catalog (as Analyst)

Data Steward	mpayton	mpayton	Limited access to Dataflow operations Access to the Catalog (as Steward)
Analyst	cparker	cparker	Limited access to Dataflow operations Access to the Catalog (as Analyst)
Guest	jdoe	jdoe	View-only access

You can then confirm that you have all the solutions installed in the **Solution management** UI:



Configure DNS entries

Create the following DNS alias to the cluster hostname, replacing `<HOSTNAME>` with the cluster hostname:

- `catalog-<HOSTNAME>`
- `dte-<HOSTNAME>`

Add licenses

> The LDOS package doesn't contain licenses. Please contact **Customer Success** or **Product Management** on how to get a license.

Licenses for Lumada Data Integration

The Data Transformation Editor needs a Pentaho EE license to run. The Dataflow Engine also needs a license for executing transformations that use Pentaho EE steps.

Upload the Pentaho EE licenses file to the NFS volume under `/<volume_path>/licenses/.installedLicenses.xml`

File must be literally named `.installedLicenses.xml`

License for Lumada Data Catalog

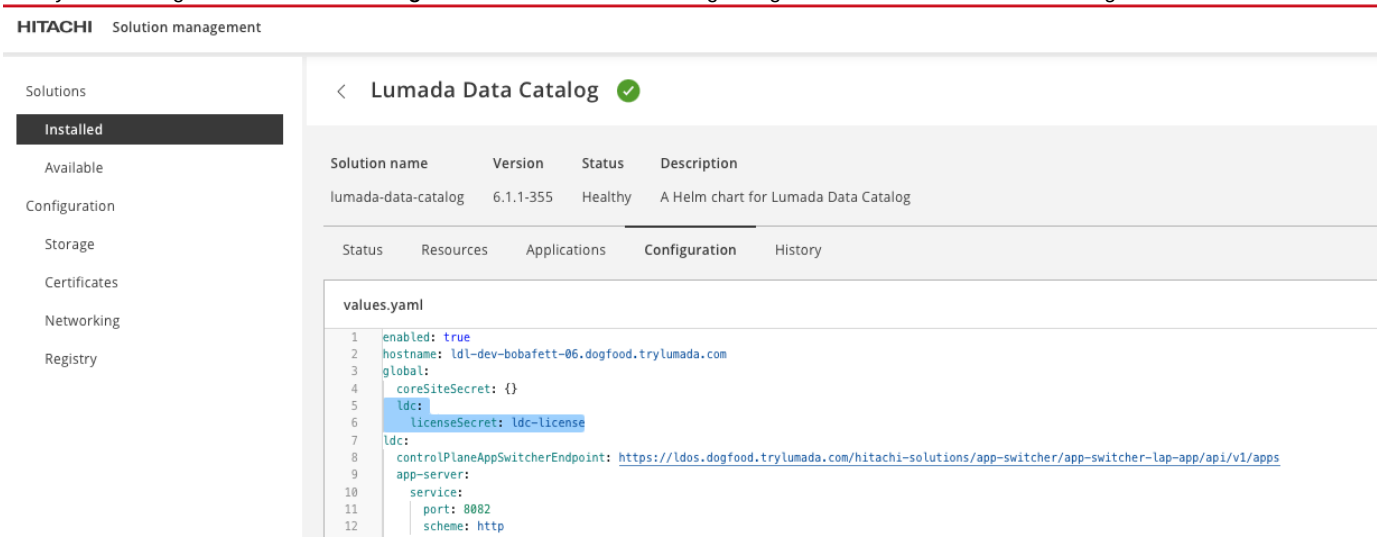
The Lumada Data Catalog is by default a light version (some functions are disabled).

In order to upgrade the license to have a full version of the Catalog, run the following command:

```
kubectl create secret generic ldc-license --from-file=license-features.
yaml --from-file=ldc-license-public-keystore.p12 -n hitachi-solutions
```

Files must be literally named `license-features.yaml` and `ldc-license-public-keystore.p12`

Then you need to go to the **Solution management** UI and add the following configuration in the *Lumada Data Catalog*:

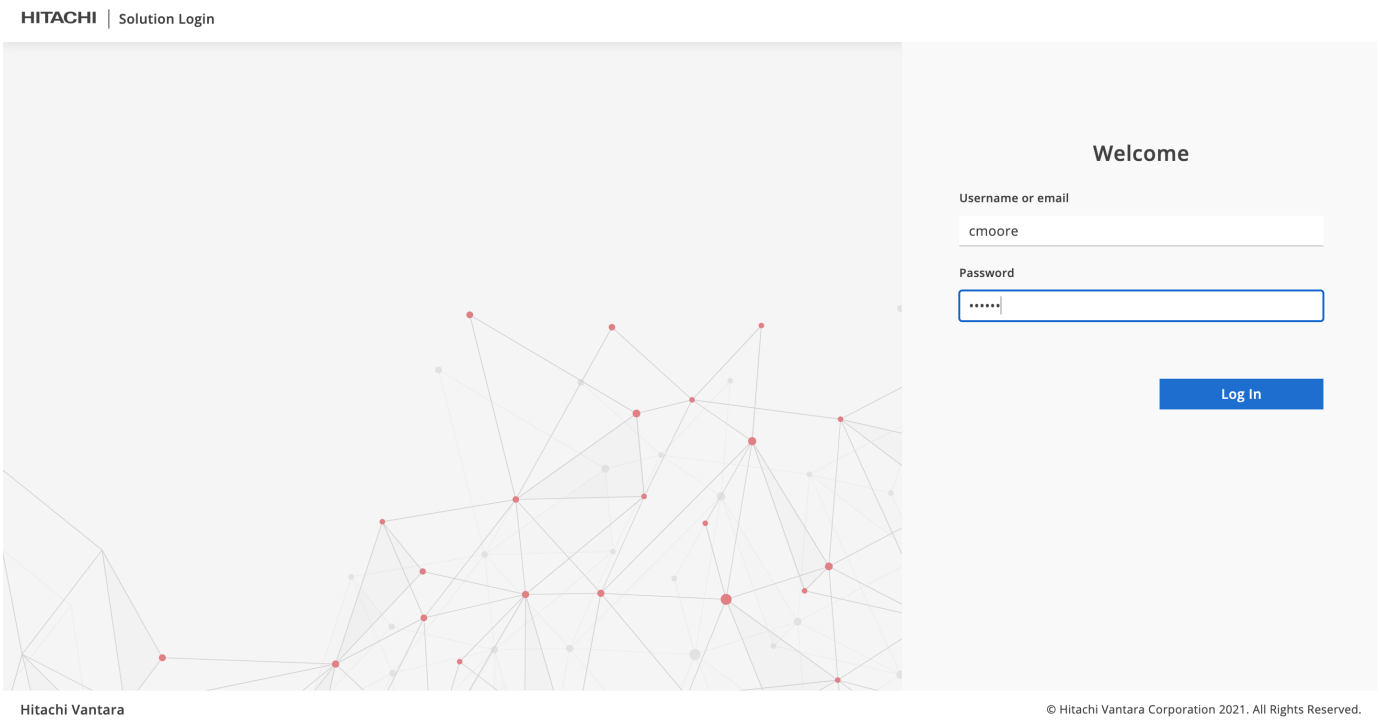


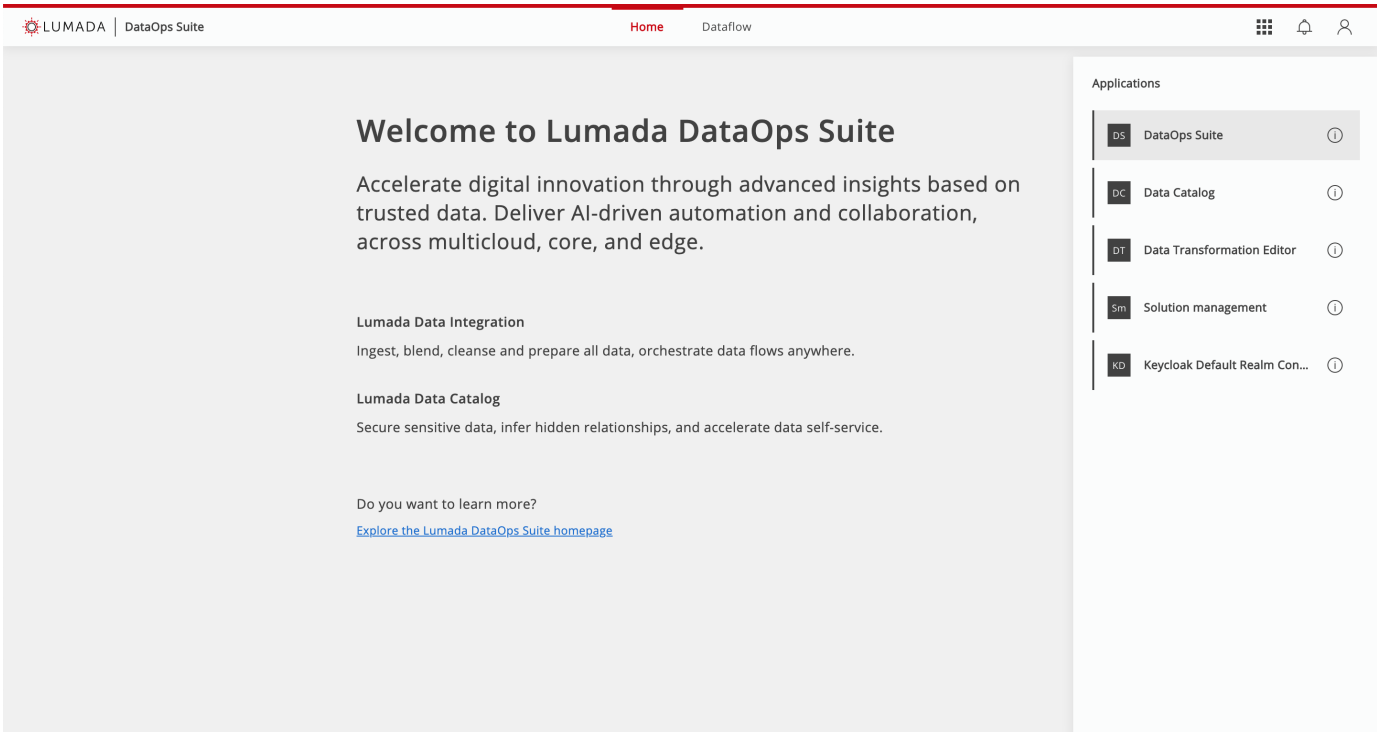
Click on *Save* and the Lumada Data Catalog will upgrade to activate the license.

Login

After the installation is completed, you can log into the **Lumada DataOps Suite** using a browser, replacing <HOSTNAME> by the cluster hostname:

https://<HOSTNAME>**/hitachi-solutions/control-plane/control-plane-lcp-app/**




LUMADA | DataOps Suite
Home
Dataflow
Welcome to Lumada DataOps Suite
Accelerate digital innovation through advanced insights based on trusted data. Deliver AI-driven automation and collaboration, across multicloud, core, and edge.
Lumada Data Integration
Ingest, blend, cleanse and prepare all data, orchestrate data flows anywhere.
Lumada Data Catalog
Secure sensitive data, infer hidden relationships, and accelerate data self-service.
Do you want to learn more?
Explore the Lumada DataOps Suite homepage
Applications
DS DataOps Suite
DC Data Catalog
DT Data Transformation Editor
Sm Solution management
KD Keycloak Default Realm Con...