

KNIME Business Hub Installation Guide - Existing Cluster

KNIME AG, Zurich, Switzerland
Version 1.10 (last updated on 2024-06-26)



Table of Contents

Introduction	1
Installation planning	2
Kubernetes	2
Software prerequisites	3
Hardware prerequisites	4
Single node installation	5
Multi-node installation	6
Networking prerequisites	9
Distribution specific prerequisites	11
Supported Kubernetes distributions	11
OpenShift	11
KNIME Business Hub installation	13
Installing KOTS	15
Next steps in the installation	16
Access the KOTS Admin Console	17
Provide a Replicated .yaml license file	19
Configure the installation	21
Preflight checks	33
Wait for the installation to complete	34
KNIME Business Hub airgapped installation	36
Airgapped install	36
Getting started	36
Getting Kubernetes ready	38
Installing KNIME Business Hub	38
Advanced settings	41
Browser Security	41
Kubernetes Cluster Management	42
Nodes and Extensions	43
PostgreSQL Database	44
DNS Entries and TLS Certificates	45
Istio	46
Create namespaces manually	47
Use existing Ingress control	47
Post-installation steps	49

Version updates.	49
User registration	49
Keycloak setup	52
Update your KNIME Business Hub license.....	55
Uninstalling KNIME Business Hub	57

Introduction

KNIME Business Hub is a customer-managed KNIME Hub instance.

Once you have a license for it and proceed with installation you will have access to Hub resources and will be able to customize specific features, as well as give access to these resources to your employees, organize them into Teams and give them the ability to manage specific resources.

Once you have access to a KNIME Business Hub instance available at your company, you can use KNIME Business Hub to perform a number of tasks such as:

- collaborate with your colleagues,
- test execution of workflows,
- create and share data apps, schedules, and API services
- keep track of changes with versioning.

The following is a guide for installing KNIME Business Hub into an existing cluster.

To administrate a KNIME Business Hub instance please refer instead to the following guide:

- [KNIME Business Hub Administration Guide](#)

Installation planning

KNIME Business Hub supports the following types of installations:

- **Installation by downloading container images from public image registries:** This type of installation requires an environment with externally accessible networks. This is a standard installation process and is documented in the [KNIME Business Hub installation](#) section. Just follow the steps [here](#), after making sure you fulfill the prerequisites in the next section.
- **Airgapped installation:** This type of installation is required for installing into air gap environments, or into an environment that, even though has a network with external access, is not allowed to download container images from public image registries. If your environment does not have access to external networks you can follow the steps in the [airgapped installation](#) section.

Kubernetes

Since KNIME Business Hub is a product based on microservices that are orchestrated and managed via Kubernetes, a Kubernetes cluster is needed when installing KNIME Business Hub.

KNIME Business Hub 1.10 supports the following versions of Kubernetes:

- 1.25
- 1.26
- 1.27
- 1.28
- 1.29

Software prerequisites

- **kubectl**: only required if installing into an existing cluster, or when remotely managing a cluster. When installing the embedded cluster with kURL `kubectl` is automatically installed on the host machine.
- **Helm**: only required if uninstalling KNIME Business Hub.

Hardware prerequisites

This guide covers the installation of KNIME Business Hub.

The following **Operating Systems** are supported:

- Ubuntu Server 20.04 LTS
- Ubuntu Server 22.04 LTS
- RHEL 8.6, 8.8, 8.9, 9.0, 9.2, 9.3
- Amazon Linux 2

Note that only x86, x64 and x86_64 processors are supported.



For installations via AWS please refer to the [KNIME Business Hub on AWS Marketplace](#) guide, for installations via Azure please refer to the [KNIME Business Hub on Azure Marketplace](#) guide.

The following sections cover the prerequisites for single node and multi node installations.

Here are some recurrent terms used in the next sections:

- **Hub core:** Refers to Hub core services that are responsible for all functionalities of Business Hub besides execution. Including authentication services, UI, database services, etc.
- **Hub execution:** Refers to executor resources, e.g. executor pod resources.
- **Disk:** The attached storage size requirement for the persistent components of Business Hub. Used to provide storage to databases, store data files for KNIME workflows, etc.



The CPU and Memory requirement amounts shown in the following chapters of this document refer to the “Total Capacity” of the nodes. Hence not all of these resources are allocatable for Kubernetes workloads, some needs to be available or reserved for system-related-services and should not be allocated, or reserved for Kubernetes related workloads. By default usually only some Memory is reserved from the total capacity, therefore if there are no reservation for CPU vCores, it is advised to either reserve or leave some vCores un-reserved by Kubernetes workloads. The reserved Memory for system-related-services is usually 100 MB by default for Kurl installations.



The "vCore" term used in this guide stands for virtual cores. They are a measure of the processing power allocated to a virtual machine (VM) or a computing instance in cloud computing environments. KNIME makes no representation about how the vCores licensed for execution are correlated or mapped onto the physical system where the software is running. The same principle also extends to the vCores necessary for operating core services.

Single node installation

Here we provide recommended requirements for single node installations.

KNIME's advice is to strive to meet the **recommended system requirements** for better performance and more scalability.

Find the advanced install option for installation on host machines with smaller sized root volumes in the [Installation on hosts with undersized root volumes](#) section.

Enterprise Plan

Using an **Enterprise** license provides the possibility to set the desired amount of vCores.

Recommended

- CPU: 26+ vCores total
 - Hub Core: 10 vCores
 - Hub Execution: 16 vCores
- Memory: 64 GB+
 - Hub Core: 20 GB
 - Hub Execution: 44 GB+ available memory to be allocated for executors
- Disk: 500GB+

Examples for setting up executors based on available resources with Enterprise license:

Limitations:

- *at least 1 vCore per executor*
- *at least 2 GB Memory per executor*



Allocatable RAM for executors in case of a 64 GB Ram installation is **44 GB**.

Examples with 16 execution vCores for an installation with recommended resource requirements:

- Scenario 1:
 - 8 executors. Each executor running with 2 vCore and 2 GB or more Memory. (total of 44 GB RAM can be shared amongst executors)
- Scenario 2:
 - 4 executors. Each executor running with 4 vCores and 2 GB or more Memory. (total of 44 GB RAM can be shared amongst executors)
- Scenario 3:
 - 2 executor. Each executor running with 8 vCores and 2 GB or more Memory. (total of 44 GB RAM can be shared amongst executors)

Multi-node installation

Here we provide installation requirements for multi node installations.

See the [advanced install options](#) section for configuring highly available clusters and installing on instances with smaller sized root volumes.

Also we distinguish two cases based on whether or not `nodeAffinity` and `nodeSelection` is used.

Read more about `nodeAffinity` and `nodeSelection` in the [KNIME Business Hub Admin Guide](#).

Highly-available Multinode installation without `nodeAffinity` and `nodeSelection`

- Node count: 3+ (Shared between Hub Core services and Execution services)
- The cluster needs to have at least **three** nodes with at least:
 - 16 vCores and
 - 32 GB RAM per instance
- Disk: 100GB+ per instance for the root volume
- Additional Attached Disks: 1 or more additional, attached, unformatted disks are required for multinode installations to handle data replication between nodes

- See the advanced install options for configuring highly available clusters and installing on instances with smaller sized root volumes

Example

3 nodes. 16 vCores and 32 GB RAM each. (Shared between Hub Core services and Execution services)

Running the installation on the above resources will allow the user to allocate 8 vCores and 16 GB RAM for execution.

Highly-available Multinode installation with nodeAffinity and nodeSelection

- Node count: 3+ (Allocated for Hub Core services only)
- For Hub Core workloads the cluster needs to have at least **three** nodes with at least:
 - 12 vCores and
 - 32 GB RAM per instance.
- For execution workloads: additional nodes need to be added to the cluster depending on the performance needs for execution.
- Disk: 100GB+ per instance for the root volume
- Additional Attached Disks: 1 or more additional, attached, unformatted disks are required for multinode installations to handle data replication between nodes
- See the advanced install options for configuring highly available clusters and installing on instances with smaller sized root volumes

Example

3 nodes for Hub Core services.

1 node for Execution services.

3 nodes with nodeAffinity or nodeSelection set to "core", 12 vCores each, with 32 GB RAM.

At least 1 node with nodeAffinity or nodeSelection set to "execution",

- 16+ vCores and 64+ GB RAM (Enterprise)



Nodes annotated with `nodeAffinity` or `nodeSelection` set to “execution” can use approximately 75% of the total allocatable resources for execution resources, due to some resource being allocated to system-related-services, some to default Kubernetes workloads running on the node and some to certain kurl add-ons, achieving high availability.

Networking prerequisites

Network Ports:

- 80 (HTTP)
- 443 (HTTPS)
- 22 (SSH) ADMIN USE ONLY
- 6443 (kubectl) ADMIN USE ONLY
- 8800 (KOTS Admin Console) ADMIN USE ONLY

Security Warnings:

- Ports 22, 6443, and 8800 are vulnerable access points for a KNIME Hub installation. If a malicious actor gained access to any of those ports, they would be able to perform destructive actions on the cluster and/or gain access to sensitive configuration. Access to these ports must be restricted to only the IP address(es) of the machine(s) which will administer the installation.
- Security-Enhanced Linux (SELinux) is not currently supported. If enabled, the installer script will notify the user via a prompt and disable SELinux before proceeding.

The following domains need to be accessible from servers performing online installations:

Trusted Host	Domain
KNIME	*.knime.com
Replicated	See the Firewall Openings for Online Installations guide.

Also, in order to be able to pull Docker images the following Docker registries need to be accessible from servers performing online installations:

```
docker.elastic.co
docker.io
ghcr.io
quay.io
registry.k8s.io (for this domain image layers are pulled from the domain below)
*.pkg.dev
registry.opensource.zalan.do
proxy.replicated.com
```



Some features in KNIME Business Hub, e.g. the **Job Viewer**, use websockets. If an external proxy or load balancer is used it needs to be websocket compatible.

Distribution specific prerequisites



The installation on **existing** Kubernetes clusters is only available for Enterprise edition licenses.

Supported Kubernetes distributions

- KURL
- AKS
- EKS
- GKE
- **OpenShift**

OpenShift

Security Context Constraints

Before installing Hub on OpenShift please ensure the following **Security Context Constraints** are configured. These are the most restrictive policies currently supported.

```
oc adm policy add-scc-to-group nonroot-v2 system:serviceaccounts:knime
oc adm policy add-scc-to-group nonroot-v2 system:serviceaccounts:hub
oc adm policy add-scc-to-group nonroot-v2 system:serviceaccounts:hub-execution
```

Istio & UIDs



For OpenShift you will need to install the Istio service mesh manually. To do so follow the instructions in the *** Istio*** section of this guide.

Business Hub will need to be configured to **not** deploy Istio. This can be done when configuring Business Hub by selecting the *View Advanced Settings* option then selecting the *Show Advanced Istio Configuration* option under *Networking*, then de-selecting the *Enable Istio* option in the **Networking: Istio** section that follows in the configuration dialog.

When installing Istio in OpenShift, Business Hub needs to be made aware of the UID used by the Istio Proxy Sidecar as this can vary depending on the method of installation.

If Istio is installed using via OpenShift Service Mesh 2.x then the Istio Proxy UID to enter will be **1002**.

If Istio is installed manually or the upcoming OpenShift Service Mesh 3.x is used then the UID of the Istio Proxy will need to be identified before installation. OpenShift will currently set this value to the **maximum** UID available in the hub namespace. Check the `openshift.io/sa.scc.uid-range` namespace annotation to find this value. e.g. for a value of `openshift.io/sa.scc.uid-range=1000990000/10000` the UID used will be 1000999999.

Networking: Istio

Advanced configuration for the [Istio service mesh](#).

Enable Istio Recommended

The [Istio service mesh](#) manages network traffic for KNIME Business Hub deployments. Its functionality depends on the Ingress-NGINX Controller. **Please note: it is highly recommended to leave this option enabled.**

Istio Proxy UID

The known UID of the Istio sidecar proxy in the hub namespace. Set this only in cases where the cluster forces a non default (1337) value for pods, e.g. OpenShift. OpenShift will currently set this value to the maximum UID available in the 'hub' namespace. Check the [openshift.io/sa.scc.uid-range](#) namespace annotation to find this value.

Default value: **1337**

DNS Entries and TLS Certificates



For OpenShift you need to go to the [DNS Entries and TLS Certificates](#) section of this guide to set up DNS Entries/TLS Certificates for your environment.

KNIME Business Hub installation

For the commands demonstrated below, replace anything shown in <brackets> with real values.



If your environment does not have access to external networks you can follow the steps in the [airgapped installation](#) section.

Before you start:

The following supplemental install instructions assume that:

- The install is performed with access to the internet on an existing Kubernetes cluster (refer to [this section](#) to know which versions are supported)
- The Kubernetes cluster needs to have **volume provisioner** (CSI) installed

Istio can optionally be installed manually. Refer to the [Istio](#) section of this guide for installation instructions.

Additionally, the following steps also assume that namespaces, [DNS entries and related TLS certificates](#) are provisioned before proceeding with the install.

If you need to create namespaces manually please refer to the [Create namespaces manually](#) section.

If you want to use an existing Ingress control follow the instructions in the [Use existing Ingress control](#) section.

Installing KOTS

The KOTS admin UI can be installed by running the following from a host machine where `kubectl` has already been configured to access the Kubernetes cluster

```
curl https://kots.io/install/<version> | bash kubectl kots install knime-hub
```



The currently supported `<version>` of KOTS is 1.108.11

When prompted, specify the desired namespace (i.e. `kots`, `default`, etc.) where you prefer the KOTS stack to run.

Once installation completes, a port-forward tunnel will be automatically opened to allow the browser to connect to the KOTS UI on <https://localhost:8800>.

When installing KOTS into an existing Kubernetes cluster, a tunnel will need to be opened anytime the KOTS UI needs to be accessed for security reasons.

The KOTS UI tunnel can be re-established by running:

```
kubectl kots admin-console -n kots
```

(`-n` defines the namespace where KOTS is running and needs to be updated accordingly)

Alternatively, you can directly establish the port-forward tunnel to kots via:

```
kubectl -n kots port-forward service/kotsadm 8800:3000
```



In case you need a Minimal RBAC Install you can find more information [here](#).

Next steps in the installation

Once the following supplemental steps above have been completed, you should be able to proceed with the remainder of the install.

The high-level remaining steps include:

- **Logging into the KOTS Admin UI**
- **Uploading the Replicated license** which will fetch the latest release of Business Hub
- **Enter appropriate configuration parameters** in the KOTS Admin Console
 - Specifically, note that there is an option under the *Networking* section to use a "provided" Ingress controller. This will block Business Hub from attempting to deploy its own ingress-nginx.
 - Ensure the option is checked for "Enable TLS" if TLS is configured either at the ingress-nginx layer or somewhere else in front of the cluster.
 - If Istio was manually installed prior to the installation, then Business Hub will need to be configured to not deploy Istio. This can be done by clicking the "Show Advanced Istio Configuration" option under *Networking*, then de-selecting the "Enable Istio" option in the *Networking: Istio* section that follows.
- Save the changes, allow the **pre-flight** checks to run, then click **Deploy** and **Wait for the installation to complete**.

Access the KOTS Admin Console

Navigate to the KOTS Admin Console URL and take note of the password.

```
Installation
Complete ✓

Kotsadm: http://[redacted]:8800
Login with password (will not be shown again): [redacted]
```

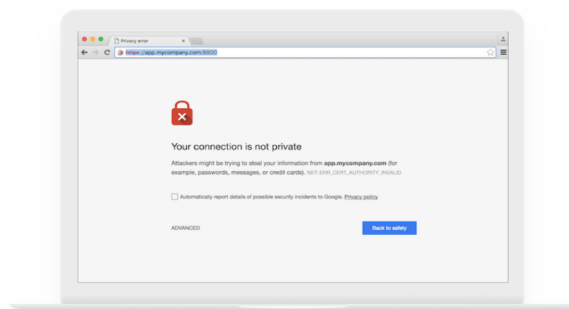
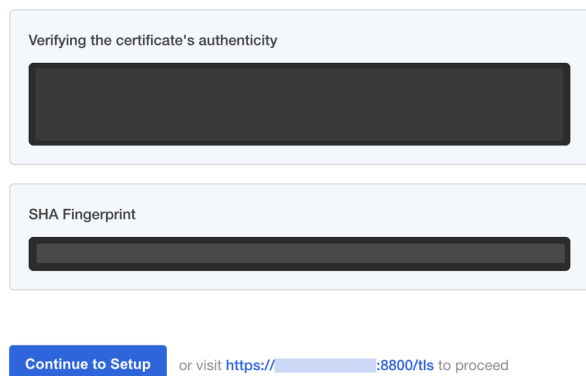
The first page that will display is a warning regarding Transport Layer Security (TLS) configuration. Follow the on-screen instructions to proceed.

Bypass browser TLS warning

We use a self-signed SSL/TLS Certificate to secure the communication between your local machine and the Admin Console during setup. You'll see a warning about this in your browser, but you can be confident that this is secure.

Chrome

On the next screen, click "Advanced", then click "Proceed" to continue to the Admin Console.



You will then be prompted to provide your own TLS cert to secure traffic to the admin console.



For everything to work correctly and securely, setting up TLS is strongly recommended.



HTTPS for KNIME Business Hub the admin console

Certificate type

Self-signed Upload your own

A self-signed TLS certificate is currently used to secure communication between your browser and the admin console. You will see a warning in your browser every time you access the admin console unless you upload your own TLS certificate.

Hostname (optional)

Ensure this domain is routable on your network.

Continue

You should then see a prompt for a password. Enter the admin console password to proceed (this password can be changed later).




Log in to KNIME Hub

Enter the password to access the KNIME Hub admin console.


Log in

Provide a Replicated .yaml license file

After logging in, you should be prompted for a license file. This is the Replicated license file that your KNIME customer care representative has provided to you and has a .yaml extension. Please contact your customer care representative if you need assistance with your license.



Upload your license file



Drag your license here or [choose a file](#)

This will be a .yaml file. Please contact your account rep if you are unable to locate your license file.

You may be prompted to install KNIME Business Hub into an “airgapped environment” if airgap installations are enabled on your license. This feature is particularly useful for installing KNIME Hub onto a machine that has no outbound internet access. Follow the instructions in the [KNIME Business Hub airgapped installation](#) section.

In most cases, it is fine to click the **Download KNIME Hub from the Internet** button if this prompt appears.



Install in airgapped environment

To install on an airgapped network, the images in KNIME Hub will be uploaded from the bundle you provide to the cluster.

Drag your airgap bundle here or [choose a bundle to upload](#)

This will be a .airgap file KNIME Hub provided. Please contact your account rep if you are unable to locate your .airgap file.

Optionally you can [download KNIME Hub from the Internet](#)

Configure the installation

If all prior steps were successful, you should now be prompted to configure your KNIME Business Hub installation. A number of settings will display for you to customize. Please note that all configuration settings in this view can be changed post-installation, except for the settings under “Initialization of KNIME Business Hub”.

Global overview

Global

High-level settings for the KNIME Business Hub deployment.

KNIME Business Hub Deployment Name Required

The KNIME Business Hub deployment name is displayed in several places in the web UI. The value may contain spaces.

KNIME Business Hub

Default value: **KNIME Business Hub**

KNIME Business Hub Mountpoint ID Required

Name of the Mountpoint ID when adding this KNIME Business Hub deployment in the KNIME AP. The value may not contain spaces (it is recommended to use dashes - instead).

knime-business-hub

Default value: **knime-business-hub**

KNIME Business Hub License Required

Upload your KNIME Business Hub License .xml file. You have received this file from your account manager, in addition to the Replicated License .yaml file that you used during a previous step. Contact your KNIME account manager if you have not yet received a .xml file containing the KNIME Business Hub License.

Upload a file

📁 Browse files for KNIME Business Hub License

View Advanced Settings

- **KNIME Business Hub Deployment Name** - This will be the name of your deployment for KNIME Business Hub
- **KNIME Business Hub Mountpoint ID** - This is where your KNIME Business Hub Mountpoint will be.
- **KNIME Business Hub License** - You should have received this from your account manager. This is a different file than the Replicated `.yaml` license file. The KNIME

Business Hub license file is a `.xml` file that contains information needed for your installation. Upload it here by clicking *Browse files for KNIME Business Hub License* and navigating to the file on your local machine.

- (OPTIONAL) select *View Advanced Settings*: - This will add additional fields underneath the *Notifications* section. This are explained in the *Advanced* options section.

Initialization of KNIME Business Hub

During the very first installation of KNIME Business Hub a number of one-time initializations are made, like creating an admin user, team, space, and execution context. Changing fields in this section after installation won't have any effect on the deployed application. The admin user can change these after the installation in the browser.

Initialization of KNIME Business Hub

This section contains configuration options that are applied during the initial installation of KNIME Business Hub. These options will be hidden after the installation has completed, and any future changes will require an administrator to manually edit the affected resource(s).

KNIME Business Hub Admin Username Required

The username for the admin user of KNIME Business Hub.

Default value: **knimeadmin**

KNIME Business Hub Admin Password Required

The initial password for the admin user. This password can be changed in Keycloak post-installation.

Initial Team Name

The name for the initial KNIME Business Hub Team.

Initial Team Description

The description for the initial KNIME Business Hub Team.

Initial Space Name

The name of the initial KNIME Business Hub Space (public) which will be associated with the initial KNIME Business Hub Team.

Initial Execution Context Name

The name for the initial KNIME Business Hub Execution Context. This Execution Context is provisioned with minimal CPU/memory resources, and the resource allocation can be increased post-installation via the KNIME Business Hub REST API.

Initial Execution Context Docker Image Repository

The image repository for the initial Execution Context.

Default value: `registry.hub.knime.com/knime/knime-full`

Initial Execution Context Docker Image Tag

The image tag for the initial Execution Context.

Default value: `r-5.2.3-477`

- **KNIME Business Hub Admin Username** - Choose a username for the KNIME Business Hub
- **KNIME Business Hub Admin Password** - Choose a password for the KNIME Business Hub
- **Initial Team name** - Choose a name for the initial team that KNIME Business Hub will create
- **Initial Team description** - Give a brief description of the Initial Team name
- **Initial Space Name** - This will be a public facing space name associated with the Initial Team Name
- **Initial Execution Context Name** - The name for the initial KNIME Execution Context Name
- **Initial Execution Context Docker Image Repository** - This is where the docker image link will go
- **Initial Execution Context Docker Image Tag** - This is where the initial Execution Context will go

Initialization of KNIME Business Hub

This section contains configuration options that are applied during the initial installation of KNIME Business Hub. These options will be hidden after the installation has completed, and any future changes will require an administrator to manually edit the affected resource(s).

KNIME Business Hub Admin Username Required

The username for the admin user of KNIME Business Hub.

Default value: **knimeadmin**

KNIME Business Hub Admin Password Required

The initial password for the admin user. This password can be changed in Keycloak post-installation.

Initial Team Name

The name for the initial KNIME Business Hub Team.

Initial Team Description

The description for the initial KNIME Business Hub Team.

Initial Space Name

The name of the initial KNIME Business Hub Space (public) which will be associated with the initial KNIME Business Hub Team.

Initial Execution Context Name

The name for the initial KNIME Business Hub Execution Context. This Execution Context is provisioned with minimal CPU/memory resources, and the resource allocation can be increased post-installation via the KNIME Business Hub REST API.

Initial Execution Context Docker Image Repository

The image repository for the initial Execution Context.

Default value: **registry.hub.knime.com/knime/knime-full**

Initial Execution Context Docker Image Tag

The image tag for the initial Execution Context.

Default value: **4.7.0**

The execution context has minimal resources (1CPU, 2GB memory) and a default executor provided by KNIME, to enable basic execution. For any production use of execution you should configure the execution context and assign more resources or use a different executor docker image.

Configure KNIME Business Hub URLs

URLs for KNIME Business Hub need to have the structure of:

- **Base URL**

- `<base-url>` (e.g. `hub.example.com`).
- The URL scheme (`http://` or `https://`) should not be included in the Base URL.
- The `<base-url>` must include the top-level domain (e.g. `.com`), and cannot be an IP address.
- This is the URL you use to view the KNIME Business Hub in your browser.
- Valid examples:
 - `hub.example.com`
 - `example.com`
- Invalid examples:
 - `https://hub.example.com/`
 - `example.com/hub`
 - `myhub`
 - `12.34.56.78`

- **Subdomains**

- `apps.<base-url>`
- `api.<base-url>`
- `ws.<base-url>`
- `auth.<base-url>`
- `storage.<base-url>`
- `registry.<base-url>`

The Base URL is the only URL that can be customized. The rest of the URLs are generated automatically.

- Webapp URL - The URL to the KNIME Business Hub webapp
- DataApps URL - The URL for the KNIME Business Hub DataApps
- API URL - The URL to the KNIME Business Hub API
- Websocket URL - The URL to the KNIME Business Hub Websocket
- Authentication URL - The URL to the KNIME Business Hub authentication endpoint
- Storage URL - The URL to the KNIME Business Hub persistent storage endpoint
- Registry URL - The URL to the KNIME Business Hub Embedded registry endpoint

If you are testing KNIME Business Hub without DNS configured, it is recommended to create `/etc/hosts` entries on your local machine pointing to the public IPv4 address of the instance running the cluster. This will redirect traffic from your local machine to the appropriate IPv4 address when you enter URLs such as `http://hub.example.com/` into your browser.

Notice that the values in `/etc/hosts` below are for `hub.example.com`. The values must match the config in the URLs section of the **Config** tab in the KOTS Admin Console, as demonstrated above. You can always use `hub.example.com` as the Base URL for local test installations.

```
<public ip> hub.example.com
<public ip> api.hub.example.com
<public ip> auth.hub.example.com
<public ip> storage.hub.example.com
<public ip> apps.hub.example.com
<public ip> registry.hub.example.com
```

On Windows machines you can find the `/etc/hosts` file in `<windows dir>\system32\drivers\etc\hosts`.

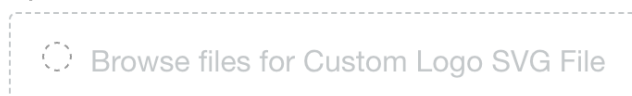
Branding

Enable Branding Options

Custom Logo SVG File

The image will be scaled to fit 183px wide and 48px high logo area. Images with significantly more height than width may appear small when displayed.

Upload a file

 Browse files for Custom Logo SVG File

If you would like to have branding options enabled check the *Enable Branding Options* box.

You will now be able to upload your own logo. Click *Browse files for Custom Logo SVG File* and navigate to select the file from you local machine.

If customizing the logo, the file being uploaded must be an .svg file in XML format such as the one below.

example.svg

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" viewBox="0 0 183 48" version="1.1"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
xml:space="preserve"
xmlns:serif="http://www.serif.com/" style="fill-rule:evenodd;clip-rule:evenodd;stroke-
linejoin:round;stroke-miterlimit:2;">
  <g transform="matrix(0.673983,0,0,0.673983,-2.4399,8.02946)">
    <text x="6.739px" y="43.245px" style="font-family:'Arial-BoldMT', 'Arial', sans-
serif;font-weight:700;font-size:54.619px;">EXAMPLE</text>
  </g>
</svg>
```

Execution Image Builder

Execution Image Builder

Configuration for the Execution Image Builder for building custom execution docker images and pushing them to the embedded registry or an external registry.

Enable Execution Image Builder

Enable the use of Execution Image Builder. Note that the container for building an image has elevated privileges.

Here you can configure the Execution Image Builder to build custom execution Docker images and push them to the embedded registry or to an external registry.

If you check the *Enable Execution Image Builder* box, two additional fields will appear.

Execution Image Builder

Configuration for the Execution Image Builder for building custom execution docker images and pushing them to the embedded registry or an external registry.

- Enable Execution Image Builder**
Enable the use of Execution Image Builder. Note that the container for building an image has elevated privileges.
- Enable Execution Image Builder Cache**
Enable the use of the Execution Image Builder cache. Note that enabling the cache leads to more occupied disk space in the destination repository.
- Skip TLS for Execution Image Builder**
Skips the TLS verification for pulling and pushing images.

By default *Enable Execution Image Builder Cache* will be selected. The option *Skip TLS for Execution Image Builder* skips the TLS verification for pulling and pushing images to the registry.

Execution Contexts

Execution Contexts

KNIME Business Hub uses Execution Contexts for hosting and executing KNIME workflows.

- Enable Custom Execution Image Pull Secret**
Enable the use of custom execution images that require Docker authentication to pull.

Executor Startup Timeout

The timeout in seconds after which the startup of an executor fails, is aborted and is tried again. Only needs to be changed if an executor image is used that takes a long time to start and never finishes startup within the configured time. Minimum value is 30.

Default value: 60

By default *Enable Custom Execution Image Pull Secret* is unchecked. By checking it you will get another option that will appear:

Execution Image Pull Secret Name

The name of the Kubernetes Secret resource which contains the Docker image pull secret for custom execution image(s). The secret should exist in the namespace defined in the **Execution Context Namespace** config option.

Default value: **regcred**

- Execution Image Pull Secret Name - The name of the Kubernetes Secret resource which contains the Docker image pull secret for custom execution image(s).
- Executor Startup Timeout - The timeout in seconds after which the startup of an executor fails, is aborted and is tried again

Job Instrumentation Data

Job Instrumentation Data

KNIME Business Hub can collect instrumentation data on workflow executions.

Do you want to collect job instrumentation data?

Job Info Expiration

How long instrumentation data about a job is stored, in days.

Default value: **30**

- Select the box *Do you want to collect job instrumentation data?* to allow the KNIME Business Hub instance to collect data about workflows execution.
- Job Info Expiration - This number reflects how long the instrumentation data about a job is stored in days

Configure networking

In the section *Networking* of the Admin Console you can deploy an external load balancer and enable and configure the Transport Layer Security (TLS).

It is anyways possible to configure this section in a second step.

To have more information on how to configure the networking section read the [KNIME](#)

Business Hub Admin Guide.

Networking

Configuration for networking options such as Transport Layer Security (TLS).

Enable TLS Recommended

Enable Transport Layer Security (TLS). This option is highly recommended for any KNIME Business Hub deployed in a production environment. A certificate must be created for all URLs configured above, including a wildcard certificate for `hub.example.com` and `*.hub.example.com`.

Notifications

In this section you can configure the notification service, which notifies for various events.

Enable Notifications Recommended

Connect KNIME Business Hub to your mail server.

Enable HTML in Email Notifications Recommended

Enable to allow HTML in email notifications, or disable for plain text.

Mail Server Configuration

Enter configuration properties for the Jakarta Mail library used by the Notification Service below. Consult the KNIME Business Hub Installation Guide, or see the Jakarta Mail API documentation for all possible parameters:

<https://jakarta.ee/specifications/mail/1.6/apidocs/index.html>

```
mail.from=hub.email@example.com (KNIME Business Hub)
mail.smtp.host=<host>
mail.smtp.port=25
```

By enabling the *Enable Notifications* box you get a new option that is checked by default.

- Enable HTML in Email Notifications - allows HTML in email notifications, or disable this option for plain text
- Mail Server Configuration - Enter configuration properties for the Jakarta Mail library



You can consult the Jakarta Mail API documentation for all possible parameters in the [Jakarta Mail API documentation](#).

The default code value is as follows:

Name	Value
Description	mail.from
hub.email@example.com	Address from which all mails are sent, required
mail.smtp.host	<host>
SMTP server host address	mail.smtp.port
25	SMTP port, default 25
mail.smtp.auth	true
Set to true if the mail server requires authentication	mail.smtp.user
<user>	Username for SMTP authentication; optional
mail.password	<password>
Password for authentication; optional	mail.smtp.starttls.enable
false	If true, enables the use of the STARTTLS command (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands.
mail.smtp.ssl.enable	false

Preflight checks

The final step before installing is the preflight checks, which is a set of automated tests to help identify if KNIME Business Hub is ready for installation. It will check the Kubernetes distribution, Kubernetes version, resources available to the cluster, and other mission-critical settings.

It is highly recommended to never skip the pre-flight checks during installation or upgrades.

Preflight checks

Preflight checks validate that your cluster meets the minimum requirements. Required checks must pass in order to deploy the application. Optional checks are recommended to ensure that the application will work as intended.

Results Re-run

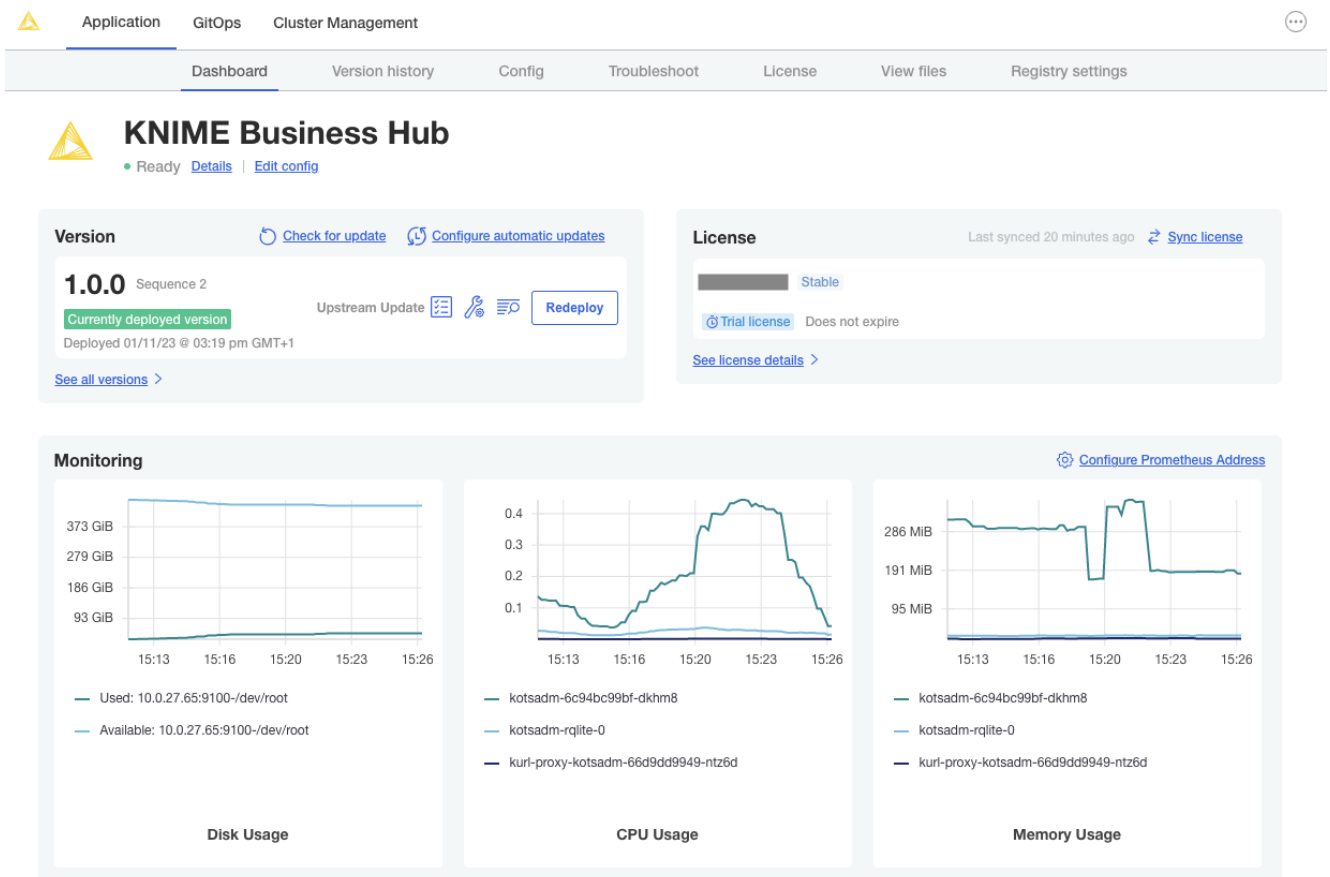
- ✓ **Required Kubernetes Version**
Your cluster meets the recommended and required versions of Kubernetes.
- ✓ **KOTS Admin Console Version**
Your cluster meets the required version of KOTS.
- ✓ **Container Runtime**
Containerd runtime was found.
- ✓ **Check Kubernetes environment.**
KURL is a supported distribution.
- ✓ **Total CPU Cores in the cluster is 16 or greater.**
There are at least 16 cores in the cluster.
- ✓ **PostgreSQL configuration**
PostgreSQL configuration has no syntax errors.
- ✓ **KNIME Business Hub URL validation**
URL passes validation for alphanumeric and special characters.
- ✓ **KNIME Business Hub URL validation: IP address**
URL is not an IP address.

Deploy

Wait for the installation to complete

If the preflight checks all passed and you opted to continue, the only thing left to do is wait for a few minutes until KNIME Hub finishes installing! You should see the installation turn the **Ready** status (top left) to green after 15-20 minutes.

If you cannot access the KNIME Business Hub Webapp URL after the **Ready** status has turned green, the first troubleshooting step would be to check the **Config** tab in the KOTS Admin Console and ensure the URLs are configured properly.



Navigating to the Webapp URL should display the KNIME Business Hub landing page.

Welcome to the
KNIME Hub

Solutions for data science: find workflows, nodes and components, and collaborate in spaces.

0	4 191	0	220
Workflows	Nodes	Components	Extensions

KNIME Business Hub airgapped installation

Airgapped install

The term airgapped is used to indicate a computer network environment that has no external network access. This type of networking environment is useful for securing sensitive data, applications, and computing resources.

An airgapped environment may be completely isolated or have a firewall between it and other networks with wider access. That wider access may be to the external world or within the wider organization.

Installing modern software in an airgapped environment can be challenging. Containerization and public registries for container images work on the assumption that all software installs have access to the public internet. To get around this challenge, software installs have to package all the required artifacts and images into a bundle that can be copied to portable media for transport into the airgapped environment.

The KNIME Business Hub supports installation into environments with externally accessible networks and into airgapped environments. Of course airgapped environments require an airgapped install approach. But even networks with external access may require airgapped installations. For instance, many organizations do not allow downloading container images from public image registries. In this instance, an airgapped install will be required.

Getting started

To get started with an airgapped install you will need the link to the Download Portal of Replicated and a password that you will receive from your KNIME customer care representative.

Enter the Download Portal from a machine that has access to the internet and has also access to the installation machine.

Here, you need to select the type of installation that you need to perform.

Select "Bring my own cluster" option for existing cluster installation of the KNIME Business Hub.

KNIME Business Hub

The installation process is different depending on if you have an existing cluster or an embedded cluster on a VM.

- Bring my own cluster**
Existing cluster installation
- Embedded cluster**
Embedded cluster on a VM

1. License

gbanyai-dev Stable

<> Dev license Does not expire

Airgap enabled Snapshots enabled GitOps enabled

[Download license](#)

Select application version

Selecting the application version ensures that compatible versions of the KOTS CLI and the KOTS airgap bundle are also selected.

One or more of the application versions available are required. If you are attempting to upgrade past a required version, ensure that you upload the airgap bundle for the required version before uploading a later version.

1.10.1 Sequence 14456

2. KOTS CLI

To be installed on your workstation as a `kubectl` plugin. Provides commands for installation, updates, and more.

v1.109.10

Jun 6, 2024 @ 3:49pm



[Download CLI plugin](#)

3. KOTS Admin Console Bundle

This includes the admin console images, manifests, and dependencies. This can be downloaded to your local workstation and then automatically pushed to your private registry with the KOTS tooling.

v1.109.10

Jun 6, 2024 @ 3:49pm



[Download admin bundle](#)

[Show other bundles](#)

4. KNIME Business Hub App Bundle

This includes the application images, manifests, and dependencies. This can be downloaded to your local workstation and then automatically pushed to your private registry with the KOTS tooling.

1.10.1 Sequence 14456

Jun 4, 2024 @ 7:46pm [Show Checksum](#)



[Download airgap bundle](#)

Optional

Latest Preflight CLI

To be installed on your workstation as a `kubectl` plugin. Provides commands for validating your system requirements.

v0.93.1

May 31, 2024 @ 9:11am



[Download CLI plugin](#)

Latest Support Bundle CLI

To be installed on your workstation as a `kubectl` plugin. Provides debugging via collectors, analyzers, and redactors.

v0.93.1

May 31, 2024 @ 9:11am



[Download CLI plugin](#)

From the download page you need to download:

- The Replicated license file (`.yaml`) with airgapped install enabled.
- The airgap bundle for the KNIME Business Hub (*knime-hub Airgap Bundle*).

Getting Kubernetes ready

Next step is to install the new Kubernetes cluster.

You will need to install the KNIME Business Hub instance into an existing Kubernetes cluster.

Installing into an existing cluster with air gap

Before installing KNIME Business Hub into an existing cluster, the KOTS extension for Kubernetes must be installed into your Kubernetes cluster. See the above section on how to download artifacts from the KNIME Business Hub download site.

For airgapped installs, download the “KOTS Airgap Bundle” which can be found in the “Bring my own cluster” section of the download site.

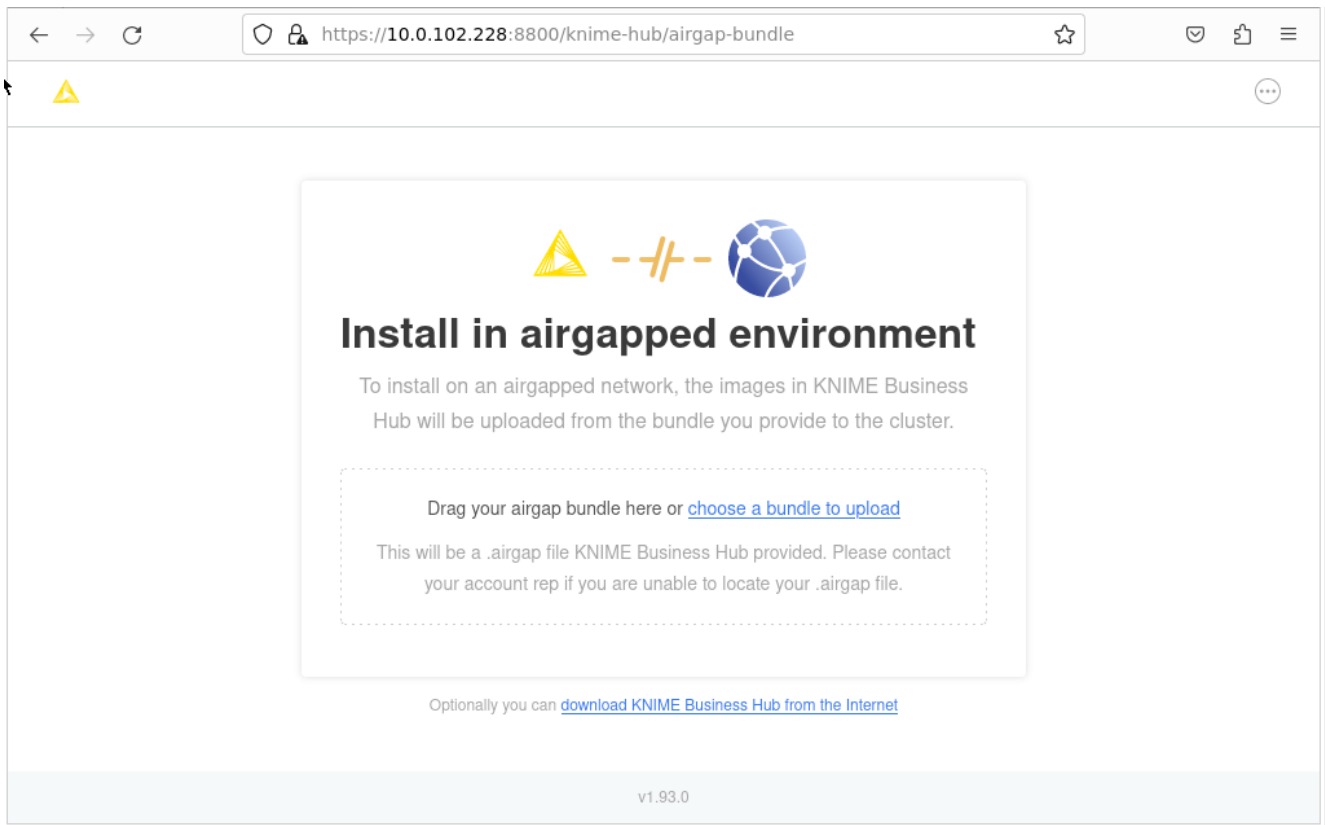
Installing KNIME Business Hub

Once your Kubernetes cluster is configured with the KOTS extension, you are ready to install the KNIME Business Hub.

Now you need to proceed with the airgap install on a machine that:

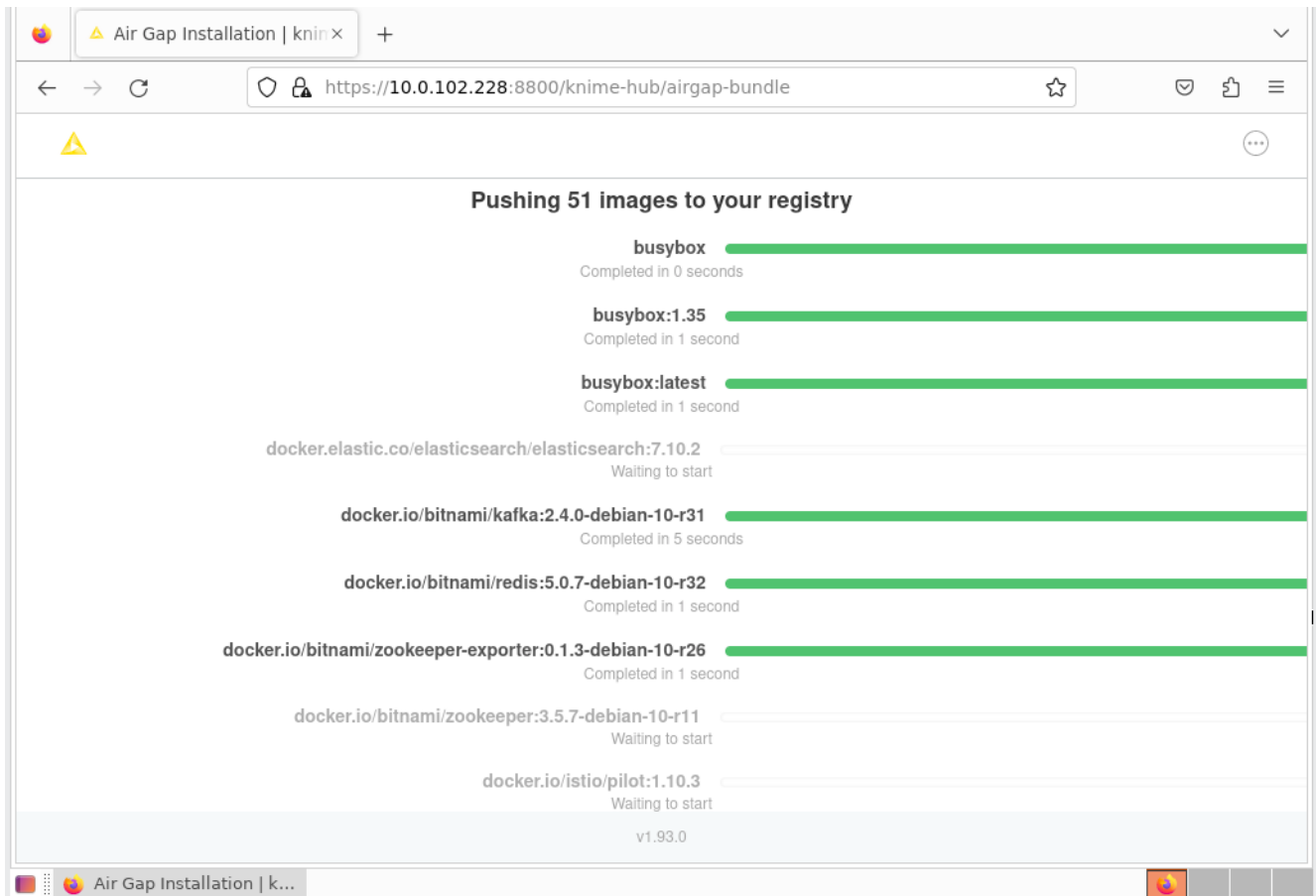
- Hosts the web browser used for the install and has access to the internet
- Has access to the machine into which you are installing your KNIME Business Hub instance

On the machine that can access the installation machine, you can open the KOTS Admin Console URL and enter the kotsadm password. Follow the instructions for installing the KNIME Business via a browser. You will first need to upload the license and then the airgap bundle you downloaded from the Download Portal.



Once you select the KNIME Business Hub airgap bundle to use, the bundle will be uploaded to the install computer. The bundle is large and will take some time to upload.

Once the bundle has completed loading, the installer will extract all the required images from the bundle and store them in a local image registry. Again this will take some time. In your browser, you will see the progress of the extraction and load process. It will look like the screen shot below. Once the load into the registry is complete, the configuration page will be shown. Proceed as normal with the install at this point.



If all prior steps were successful, you should now be prompted to configure your KNIME Business Hub installation.

To do so follow the steps in the [Configure the installation](#) section.

Advanced settings

The following section explains you how to set up advanced settings for your KNIME Business Hub instance. These settings are only visible if you checked the *View Advanced Settings* box under the *Global* tab.

You can skip to the [Initialization of KNIME Business Hub](#) section if you do not need any of the following settings.

Browser Security

Advanced: Browser Security

This section contains settings for browser security.

Enable Content Security Policy for Data Apps

Enabling this option allows you to set a custom Content Security Policy for Data Apps below. If disabled, no Content Security Policy header is set.

Content Security Policy for Data Apps

Specifies a custom Content Security Policy for Data App execution. It may be necessary to override the default if you are using BIRT report generators or custom JavaScript views that load external resources. The default works for all standard KNIME views. For more information about how to write the CSP statement, please refer to this [resource](#).

```
default-src 'self'; script-src 'unsafe-inline' 'unsafe-eval' 'self'; style-src 'unsafe-inline' 'self'; img-src 'self' data;; font-src 'self'
```

Default value: `default-src 'self'; script-src 'unsafe-inline' 'unsafe-eval' 'self'; style-src 'unsafe-inline' 'self'; img-src 'self' data;; font-src 'self' data;;`

X-Frame-Options Header

Sets the [X-Frame-Options](#) header to the selected option, or doesn't set the header if `none` is selected. This header is used to avoid click-jacking attacks, by ensuring that the sites content is not embedded into other sites. See [here](#) for more information.

SAMEORIGIN DENY none

Whitelist URLs for Execution REST API

Sets additional URLs as allowed origins for the CORS policy and the CSRF filter. This option is used to allow browser based apps to call execution REST endpoints, e.g. for Service Deployments. Multiple URLs must be either newline or space separated, e.g. `http://localhost:9000 https://example.webapp.com`.

- Enable Context Security Policy for Data Apps - this box is checked by default and gives you access to the following options:

Content Security Policy for Data Apps

Specifies a custom Content Security Policy for Data App execution. It may be necessary to override the default if you are using BIRT report generators or custom JavaScript views that load external resources. The default works for all standard KNIME views. For more information about how to write the CSP statement, please refer to this [resource](#).

```
default-src 'self'; script-src 'unsafe-inline' 'unsafe-eval' 'self'; style-src 'unsafe-inline' 'self'; img-src 'self' data:; font-src 'self'
```

Default value: `default-src 'self'; script-src 'unsafe-inline' 'unsafe-eval' 'self'; style-src 'unsafe-inline' 'self'; img-src 'self' data:; font-src 'self' data:;`

- Content Security Policy for Data Apps - this specifies a custom Content Security Policy for Data App execution.



For more information on how to write your own CSP statement please refer to this link: [X-Frame-Options](#)

X-Frame-Options Header

Sets the [X-Frame-Options](#) header to the selected option, or doesn't set the header if [none](#) is selected. This header is used to avoid click-jacking attacks, by ensuring that the sites content is not embedded into other sites. See [here](#) for more information.

SAMEORIGIN DENY none

- X-Frame-Options Header - Sets the X-Frame-Options header to the selected option, or doesn't set the header if none is selected

Whitelist URLs for Execution REST API

Sets additional URLs as allowed origins for the CORS policy and the CSRF filter. This option is used to allow browser based apps to call execution REST endpoints, e.g. for Service Deployments. Multiple URLs must be either newline or space separated, e.g. [http://localhost:9000 https://example.webapp.com](#).

- Whitelist URLs for Executions REST API - Sets additional URLs as allowed origins for the CORS policy and the CSRF filter

Kubernetes Cluster Management

Advanced: Kubernetes Cluster Management

This section contains settings for how resources (e.g. Deployments) are scheduled in the Kubernetes cluster.

Mandatory Node Affinity

KNIME Hub has support for scheduling core and execution resources to specific nodes using the `hub.knime.com/role=core` and/or `hub.knime.com/role=execution` node labels. By default, pods will be scheduled to nodes with the appropriate label using a best attempt strategy. If this setting is enabled, pods will fail to be scheduled unless there is a node with the appropriate label.

Execution Context Update Strategy

The `Recreate` strategy involves terminating and recreating execution contexts when they are updated. This may cause downtime but reduces the possibility of resource allocation conflicts during provisioning. On the other hand, the `Rolling Update` strategy brings a new execution context with updated configuration online before terminating the previous version, minimizing downtime. This strategy is often used to prevent misconfigured execution contexts from causing downtime for end users. Although `Rolling Update` is effective for minimizing downtime, `Recreate` is a better strategy for minimizing resource utilization. For more information on deployment strategies, please refer to the [Kubernetes documentation](#).

Rolling Update Recreate

Enable Automatic Deployment and Updates for Custom Resource Definitions (CRDs)

Recommended

Enable the `crd-updater` job (and corresponding cluster role) which applies the appropriate version of each required CRD to the cluster during initial installation and subsequent upgrades/deployments. If this job is disabled, the cluster role will not be created but any new and/or updated CRDs will have to be applied manually by the customer. See [Kubernetes docs](#) for more information on custom resource definitions.

- **Mandatory Node Affinity** - by default this box is unchecked. KNIME Hub has support for scheduling core and execution resources to specific nodes using `hub.knime.com/role=core` and/or `hub.knime.com/role=execution` node labels.
- **Execution Context Update Strategy** - The *Recreate* strategy involves terminating and recreating execution contexts when they are updated. The *Rolling Update* strategy brings a new execution context with updated configuration online before terminating the previous version, minimizing downtime

Nodes and Extensions

Advanced: Nodes and Extensions

Configuration for the initial import of nodes and extensions to KNIME Business Hub.

- Enable Importing Nodes and Extensions** Recommended
Enable an initial import of nodes and extensions to the KNIME Business Hub instance upon first installation.
- Import from Trusted Update Sites Only**
Filter the initial import of nodes and extensions to trusted Update Sites only.

- **Enable Importing Nodes and Extensions** - this option is checked by default. It will enable an initial import of nodes and extensions to the KNIME Business Hub instance upon first installation
- **Import from Trusted Update Sites Only** - this option is unchecked by default. It allows you to filter the initial import of nodes and extensions to trusted Update Sites only

PostgreSQL Database

Advanced: PostgreSQL Database

KNIME Business Hub uses [PostgreSQL](#) to store metadata about users, workflows, jobs, and more. You can fine tune PostgreSQL performance below. Changing settings here may trigger a restart of the PostgreSQL instance.

Max Connections

Number of [maximum concurrent connections](#) that PostgreSQL allows.

Default value: 200

CPU Resources

In millicores, 1000m = 1 CPU core. See [kubernetes documentation](#).

Default value: 1000

Memory Resources

In megabytes. See [kubernetes documentation](#).

Default value: 1024

Enable CDDS database

Enable the database that can be used by the KNIME Continuous Deployment for Data Science (CDDS) extension.

The activation of this option is necessary only if you intend to use the CDDS extension in conjunction with the Hub's internal database. See [installation guide of CDDS](#).

- Max Connections - this number represents the number on maximum concurrent connections that PostgreSQL will allow
- CPU Resources - this number shows the CPU Resources in milicores (1000m = 1 CPU core).
- Memory Resources - this number shows the Memory Resources in megabytes
- Enable CDDS database - this value is unchecked by default. This option allows you to enable the database that can be used by the KNIME Continuous Development for Data Science (CDDS) extension

DNS Entries and TLS Certificates

KNIME Business Hub will be accessed via a base URL, with a number of subdomains also configured for more targeted purposes. The following will use `hub.example.com` as an example base address to highlight how the base address and subdomains are to be

configured.

DNS entries will need to be created for the following endpoints:

- `hub.example.com` - root URL also used for the Business Hub UI
- `apps.hub.example.com` - used for exposing workflow-defined Data Apps
- `api.hub.example.com` - exposed the Business Hub API
- `ws.hub.example.com` - used for websocket communication between Browser/AP and Business Hub
 - Currently used for exposing KNIME AI service available in Business Hub Standard or Enterprise 1.8.0 and higher
 - Requires the Loadbalancer in front of the Ingress controller supports Layer-4/TCP/TLS (i.e. AWS Application Load Balancer or Network Load Balancer, etc.)
- `auth.hub.example.com` - used for exposing the embedded Keycloak user store
- `storage.hub.example.com` - used for exposing the embedded Minio object store where workflows and data files are located

Note that the above domain and subdomains can be either directly set to the relevant Loadbalancer for the Ingress controller, or can alternatively be set by adding the necessary annotations to the Ingress rules below if using an automated DNS provisioner like External-DNS.

Additionally, a TLS cert will need to be provisioned either directly or via adding relevant Certificate resources and settings to the the Ingress rules below if using an automated Certificate provisioner like Cert-Manager.

If creating the cert explicitly, it is typically recommended to create a cert with the common name `hub.example.com`, and listing all domains above as subject alternative names, or using a wildcard for the subject alternative name (i.e. `*.hub.example.com`)

Istio

When required Istio Service Mesh can be manually installed and configured in the target cluster if cluster security configurations or cluster types (e.g. OpenShift) disallow Business Hub from creating the mesh itself.

Detailed instruction for installing Istio as well as sample configuration can be found [here](#).

If installing Istio manually, then the option to include Istio in the KOTs Configuration Dialog (when configuring the Hub release) will need to be disabled. If the Configuration Dialog

specified that Istio is enabled (the **default** value), then the KNIME Business Hub release process will attempt to install Istio and its related Custom Resource Definitions (CRDs).

Please notice that the following instructions are only necessary if you have a highly restricted environment. The Hub installer will provide all of the needed pieces by default.

Create namespaces manually

During the installation of the Business Hub namespaces are created automatically. However, in case you have restrictions and you need to create namespaces manually you can follow these instructions.

Namespaces:

The following namespaces need to be initially created:

- `kots` - namespace where the KOTS Admin UI is deployed to (This namespace is specified when running the KOTS install shell script to deploy KOTS as it will act as the deployment administration tool for the Business Hub release. This can alternatively be `default` or any other namespace the admin prefers to use.)
- `istio-system` - the namespace where the Istio service mesh controller will run. This namespace may already exist if Istio was deployed previously during the above steps.
- `knime` - namespace where Hub persistence layers run
- `hub` - namespace where Hub microservices run
 - The hub namespace additionally needs the `istio-injection=enabled` label set for inclusion in the service mesh
- `hub-execution` - namespace where KNIME executors will run

Use existing Ingress control

If you want to use an existing Ingress control follow these instructions.

Ingress rules:

Ingress rules will need to be manually created prior to install. A working template of the required Ingresses can be found in [this](#) `.yaml` file.

Ingress rules may need to be further updated depending on use of other tools to correctly set DNS, TLS, etc.

See the *Deploy Ingress Resources* section [here](#) for more information.

Post-installation steps

Version updates

If you save any changes in the **Config** tab of the KNIME Business Hub Admin Console, or check for updates and see a new version that you can upgrade to, then the new version will be visible in the **Version history** tab. New versions of KNIME Business Hub will not be deployed automatically unless automatic updates have been configured. Preflight checks will execute prior to deployment and the deployment itself can be triggered by clicking the **Deploy** button.

The screenshot shows the 'Version history' tab in the KNIME Business Hub Admin Console. The navigation bar includes 'Dashboard', 'Version history' (selected), 'Config', 'Troubleshoot', 'License', 'View files', and 'Registry settings'. The main content area is divided into three sections:

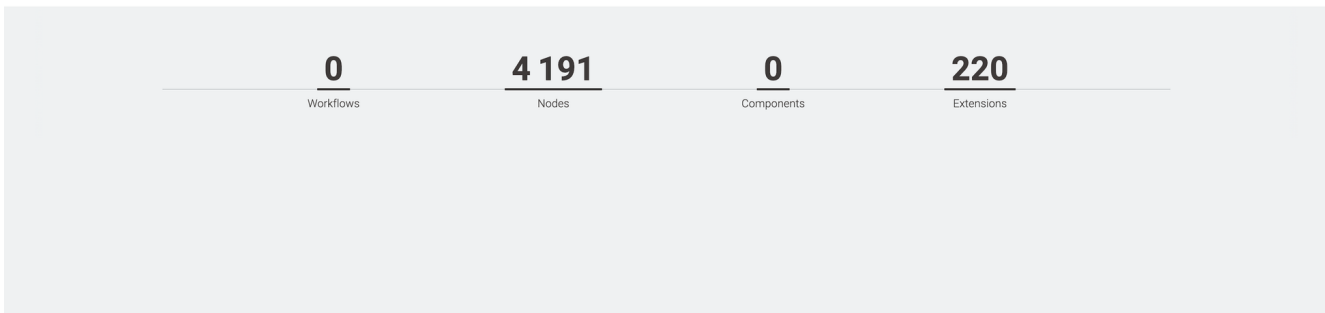
- Currently deployed version:** Shows version 1.0.0, Sequence 2, deployed on 01/11/23 at 03:19 pm GMT+1. It includes icons for configuration and deployment.
- New version available:** Shows version 1.0.0, Sequence 3, with a 'Config Change' and '7 files changed'. It includes a 'View diff' link, icons for configuration and deployment, and a 'Deploy' button. Action links include 'Check for update', 'Configure automatic updates', and 'Diff versions'.
- All versions:** A list of versions with a 'Results per page: 20' dropdown. It shows two entries:
 - Version 1.0.0, Sequence 3, 'Config Change', '7 files changed', 'View diff', 'Deploy' button.
 - Version 1.0.0, Sequence 2, 'Upstream Update', 'Released 12/21/22 @ 07:18 pm GMT+1', '22 files changed', 'View diff', 'Currently deployed version' badge, 'Redeploy' button.

User registration

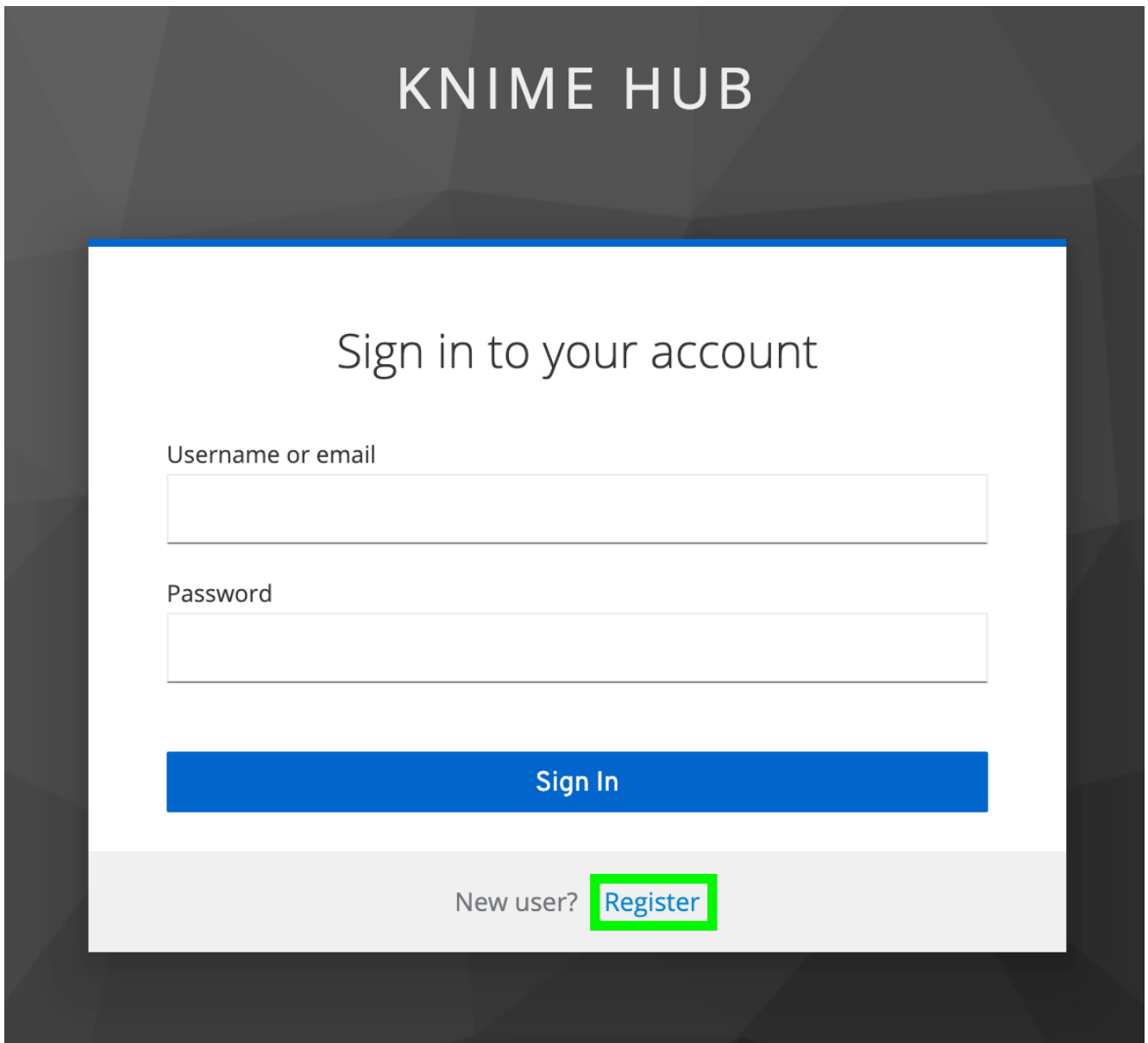
After initial installation, start the process of creating the first user by clicking the **Sign In** button.

Welcome to the
KNIME Hub

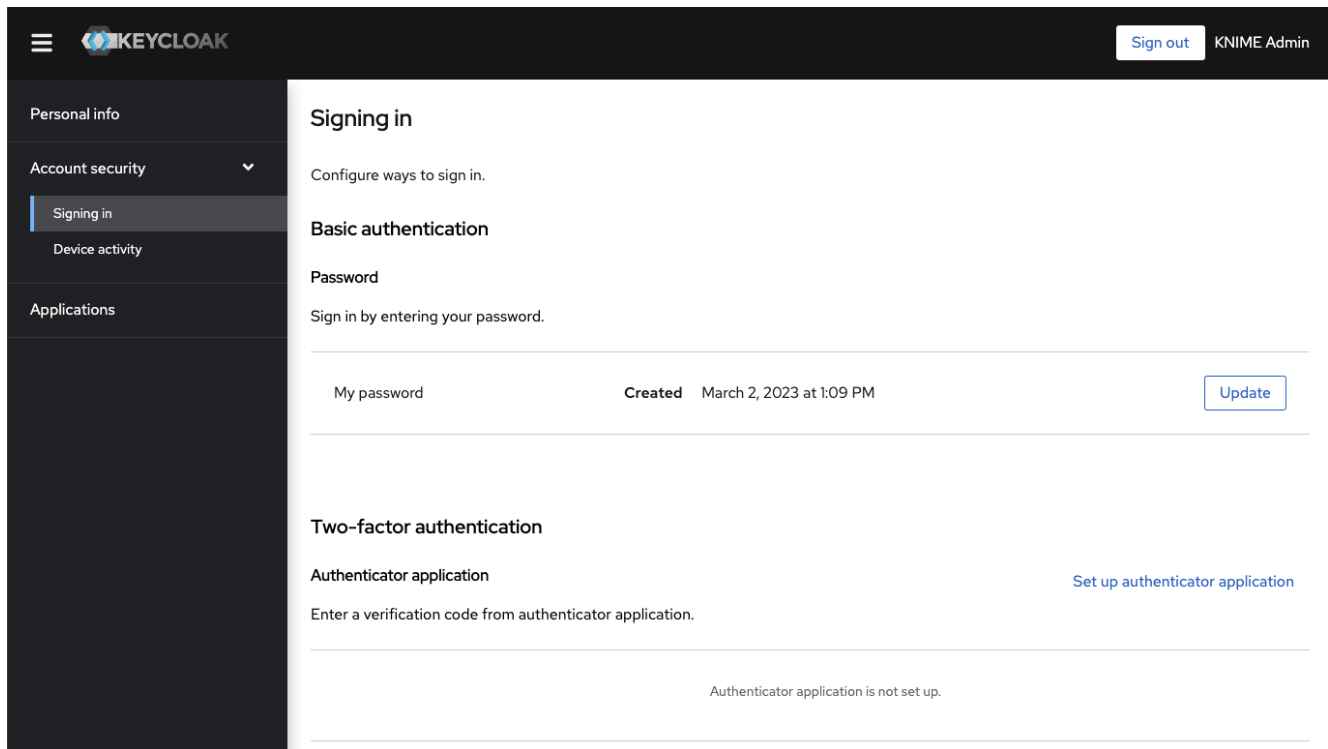
Solutions for data science: find workflows, nodes and components, and collaborate in spaces.



Next, click the **Register** button to proceed with creating a new account. You will be prompted for user information and will be logged in automatically.



Users can change their password by going to `auth.<base-url>/auth/realms/knime/account` (e.g. `http://auth.hub.example.com/auth/realms/knime/account`) and navigating to **Account Security** → **Signing In**.



Keycloak setup

Keycloak is an open source software product that KNIME Business Hub leverages for authenticating users with Single-Sign On. Keycloak offers Identity Brokering and User Federation to external Identity Providers.

You can manage your Keycloak setup by going to `auth.<base-url>/auth/` (e.g. `http://auth.hub.example.com/auth/`), clicking Administration Console and logging in with the Keycloak admin credentials. These credentials are stored in a kubernetes secret called `credential-knime-keycloak` in the `knime` namespace.



When you first log in to the Keycloak Administration Console you are shown the `master` realm. However, all KNIME Business Hub related configurations are in the `knime` realm. You can select the realm via the dropdown menu in the top left corner.

For configuring your Keycloak setup, e.g. for adding User Federation or an external Identity Providers, consult the [Keycloak Server Administration Guide](#).

Integrating an external Identity Provider (IdP)

While Keycloak itself can be used as the user base, KNIME Business Hub can also federate to an external IdP. To do this the external IdP needs to support a protocol known to Keycloak,

such as SAML v2.0, OpenID Connect v1.0, or OAuth v2.0. For configuring Keycloak to use an external IdP please see the [Integrating identity providers section](#) in Keycloaks own documentation on this topic.



If you are using **Microsoft Entra ID** as external Identity Provider you can restrict who can register to your KNIME Business Hub instance by deactivating the option "[Assignment required?](#)" in Microsoft Entra ID to Yes and then allowing only a subset of users or groups to access the KNIME Business Hub instance as explained in the [Microsoft Entra ID documentation](#).

Requirements

Keycloak and KNIME Business Hub have the following requirements for a user or external Identity Provider:

- Usernames can only include letters and numbers as first character. Letters, numbers, underscore, and @ as last character. Letters, numbers, space, dot, underscore, hyphen, and @ for remaining characters. The minimum length is 3 characters, the maximum is 64. Special characters like umlauts (äöü) are not allowed.
 - KNIME Business Hub uses this regex to validate usernames: `^[a-zA-Z0-9][a-zA-Z0-9-_.@]{1,62}[a-zA-Z0-9_]$`.
- When a new user is added to keycloak, it needs to have values for the *Username*, *Email*, *First name* and *Last name* fields. If any of those are not filled in automatically you need to create a mapper for it in the Identity Provider settings. Otherwise the user will be prompted to fill in details themselves when they log in for the first time.
- Each user needs to have an email address, even if notifications are not configured in KNIME Business Hub.

Identity Provider Mapper

After adding an external IdP to your Keycloak instance you can create Identity Provider Mappers. These mappers can be used to fill in attributes if they are not getting automatically added, like *Email*. See the [Mapping claims and assertions section](#) in Keycloaks documentation.

In Keycloak, in the `knime` realm, go to the Identity Providers section, click your provider and go to the Mappers tab. Click the *Add Mapper* button and configure the mapper. Some useful Mapper types:

- *Username Template Importer*: you can use this mapper to generate a username from

multiple claims if a username claim is not available. Example template: `${CLAIM.given_name}.${CLAIM.family_name}`. `given_name` and `family_name` are the given claims here. Nesting is possible with e.g. `${CLAIM.user_info.given_name}`.

- There are three transformers you can use here: lowercase, uppercase, and localpart.
- *Attribute Importer*: map an Identity Provider claim to a Keycloak attribute. Useful target attributes: `email` (*Email*), `given_name` (*First name*), and `family_name` (*Last name*). The Claim field is not a template like above, but accepts the claim name directly, e.g. `family_name`.
- Use the *Hardcoded Attribute* mapper if an attribute can't be constructed from a claim.

For finding available claims see the section below.

Tips for debugging Keycloak when integrating an external IdP

To create appropriate mappers in the IdP configuration one needs to know what the external IdP is actually returning to Keycloak in the token. This can be done by increasing the loglevel of the `org.keycloak.social.user_profile_dump` keycloak logger to debug. To do so you need to modify an instance of the `keycloaks.k8s.keycloak.org` CustomResource. You can get the current yaml definition of it by executing:

```
kubectl get keycloaks.k8s.keycloak.org -n knime knime-keycloak -o yaml
```

You can either save this output to a file, modify the file, and apply the changes to the cluster with:

```
kubectl apply -f <file>
```

Or you can edit it directly in the cluster using `kubectl edit`. This will open the default editor in your current shell. Make the modification, save and exit the editor, and it will automatically be applied to the cluster:

```
kubectl edit keycloaks.k8s.keycloak.org -n knime knime-keycloak
```

The modification that needs to be done is adding an additional configuration option under `spec.additionalOptions`:

```
spec:
  additionalOptions:
    - name: log-level
      value: INFO,org.keycloak.social.user_profile_dump:debug
```

Do not remove or change any of the other lines.

After this modification has been applied the keycloak-operator will automatically reconcile the `knime-keycloak-0` pod in the `knime` namespace. After the pod is ready again attempt a login with the configured external IdP in the KNIME Business Hub UI (e.g. <http://hub.example.com>) and look at the log output of the `knime-keycloak-0` pod. The interesting lines contain `org.keycloak.social.user_profile_dump` with the full JSON that was received from the external IdP:

```
2024-05-23 08:45:21,103 DEBUG [org.keycloak.social.user_profile_dump] (executor-thread-2) User Profile JSON Data for provider oidc: {"sub":"02aebec3-aaba-4e96-86a7-1d12531e8fb2","email":"admin@example.com","email_verified":false,"name":"Admin Admin","preferred_username":"admin","zoneinfo":"Europe/Berlin"}
```

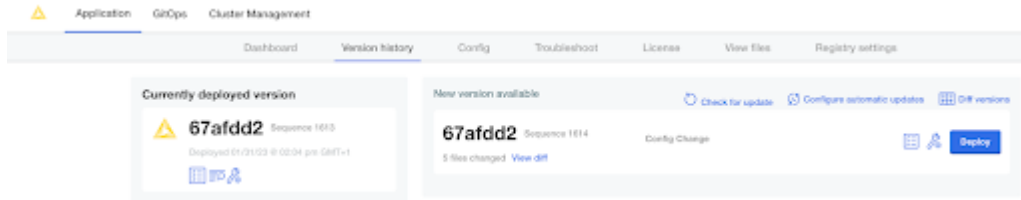
To turn the debug logging off again in Keycloak, remove the `additionalOptions` item that you added in the earlier step.

Update your KNIME Business Hub license

In order to deploy a new Business Hub license, please go to the Replicated console. There, navigate to the Config tab and find your current license file.

The screenshot shows the configuration interface for the KNIME Business Hub. The 'Config' tab is selected, and the 'Global' section is expanded. The 'KNIME Business Hub License' field is highlighted, indicating that a license file has been uploaded. The uploaded file is named 'enterprise_license4.xml'.

Click “select a different file”, and choose the `.xml` file provided by your KNIME customer care representative. Afterwards, scroll all the way to the bottom to confirm the configuration change. Click “go to updated version” next. This brings you to the “version history”, where you need to click on “deploy” to switch to the new license.



Uninstalling KNIME Business Hub

Uninstalling KNIME Business Hub is a **highly destructive action** that can have permanent implications. Please ensure you are connected to the right cluster and 100% sure you want to uninstall all resources related to KNIME Business Hub before proceeding. Please ensure you have retrieved all data or backups from the cluster that you want to preserve.

To completely remove a KNIME Business Hub instance, the following commands can be run from a terminal which has `kubectl` access to the cluster running KNIME Business Hub.

Both `kubectl` and `Helm` must be installed to successfully run the commands. Please confirm that the proper Kubernetes context is set before executing.

```
# Remove the KNIME Business Hub app from the KOTS Admin Console.
kubectl kots remove knime-hub -n default --force

# List all helm releases that will be deleted in a later step. This does not delete
anything yet.
# Ensure there are no helm releases that you deployed yourself.
helm ls -a -n hub && helm ls -a -n knime

# Delete all helm releases in the hub namespace.
helm ls -a -n hub | awk 'NR > 1 { print "-n "$2, "$1}" | xargs -L1 helm delete;

# Delete the knime-hub-keycloak helm release, and wait 10seconds to give the knime-
keycloak-operator time to reconcile.
helm delete -n knime knime-hub-keycloak; kubectl delete
keycloakrealms.legacy.k8s.keycloak.org knime-realm -n knime; sleep 10;

# Delete all helm releases in the knime namespace.
helm ls -a -n knime | awk 'NR > 1 { print "-n "$2, "$1}" | xargs -L1 helm delete;

# Remove finalizer in Keycloak realm resource
kubectl patch keycloakrealms.keycloak.org -n knime knime-realm -p
'{"metadata":{"finalizers":null}}' --type=merge

# Finally, delete all namespaces that were created by KNIME Business Hub.
kubectl delete namespace istio-system hub hub-execution knime
```

If the commands above run successfully, all KNIME Business Hub resources will be removed from the cluster. You could then re-install KNIME Business Hub into the same cluster by navigating to the KOTS Admin Console in a browser and following this installation guide again.

KNIME AG
Talacker 50
8001 Zurich, Switzerland
www.knime.com
info@knime.com