

KNIME Edge User Guide

KNIME AG, Zurich, Switzerland
Version 1.2 (last updated on 2024-07-10)



Table of Contents

Introduction	1
Interacting with KNIME Edge from KNIME Server	2
KNIME Edge Licensing	2
License States	3
Inspecting the Cluster and License Status	3
Registering KNIME runtime images	3
Managing Inference Deployments	4
Creating an Inference Deployment	5
Verifying an Inference Deployment	6
Updating an Inference Deployment	6
Deleting an Inference Deployment	7
Logging	7
Retrieving execution logs from an Inference Deployment	7

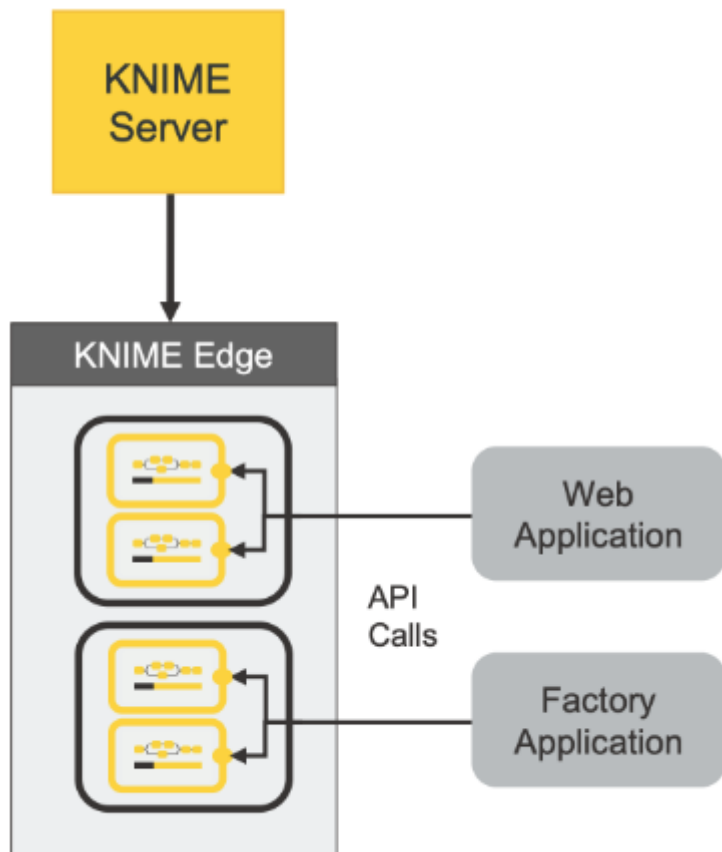
Introduction

This guide outlines the requirements, considerations, and steps for interacting with a KNIME Edge cluster and creating Inference Deployments.

- i** The examples below will utilize the [REST API for Sentiment Analysis](#) workflow.
- i** The examples below will assume that the Edge workflows have been deployed to an Edge/ workflow group on KNIME Server.
- i** The URL to the KNIME Server instance used to configure the Edge Deployment will be referred to as `knimeServerUrl`.

KNIME Edge is a distributed, container-based platform that moves consumption of models directly to where data is generated. Built on top of Kubernetes, KNIME Edge offers the ability to deploy inference-oriented workflows as highly available and scalable endpoints. This allows for high throughput and low latency while also decentralizing execution by deploying into datacenters, manufacturing facilities, multiple cloud providers, and more.

One or more KNIME Edge clusters can be remotely managed by leveraging KNIME Server. Using KNIME Server's Webportal, a user can select and deploy workflows to any connected KNIME Edge clusters. Once a workflow is deployed, KNIME Edge creates locally consumable endpoints while managing execution, scaling, uptime, resiliency and more to ensure model application can scale seamlessly with demand.



Interacting with KNIME Edge from KNIME Server

The KNIME Edge Control Plane workflows must be deployed to KNIME Server in order to interact with KNIME Edge cluster(s). The control plane workflows (and data apps) provide capabilities for:

- Initializing KNIME Server as a control plane for KNIME Edge
- Initializing and updating the PostgreSQL database for KNIME Edge
- Creating, reading, updating, and deleting Inference Deployments

For instructions on deploying the KNIME Edge Control Plane workflows, see [Configuring KNIME Server for KNIME Edge](#).

KNIME Edge Licensing

License States

A KNIME Edge cluster can be in one of three different states, indicated by colors:

- **Green:** The license is valid and the Edge cluster works without restrictions.
- **Yellow:** The license has recently expired or become underspecified and is working within a grace period. The Edge cluster still works without restrictions but its state is about to turn red.
- **Red:** Either no license is provided or the provided license is invalid, expired or underspecified (and the grace period is over). Already running deployments keep working but new deployments are rejected.

A license is considered to be underspecified if the number of provisioned CPU cores by the cluster exceeds the license specification. Note that if a node selector is configured, only the CPU cores of the selected nodes will be counted.

Inspecting the Cluster and License Status

The license state of a KNIME Edge cluster can be inspected by running the `View Edge Clusters` workflow. Additionally, when creating new or updating existing deployments, the state of the cluster will be indicated with a color (as depicted below). A cluster with a red state will reject a new deployment.

Select the edge locations for deployment:

Excludes



Includes



Registering KNIME runtime images

Before an Inference Deployment can be created and assigned to a KNIME Edge cluster, one or more runtime execution Docker images need to be added as references to a KNIME Server.

Note, KNIME Server does not directly pull or use these images, but they serve as a catalog of available runtime images that will be used for workflows deployed to KNIME Edge clusters.

Check the **knime-execution** project on the [KNIME Artifact Registry](#) for a list of KNIME published runtime images.



Note that a user will need access enabled to view and pull images from the **knime-execution** project.

To add a reference for a new runtime image, run the **Add Execution Image** workflow found in the **/apps** section of the KNIME Edge workflows.

Add Execution Image

Add available KNIME runtime docker images so they can be assigned to deployments. If adding KNIME hosted images from the KNIME Artifact Registry (<https://registry.hub.knime.com>), ensure connected Edge clusters can pull the needed images from the "knime-execution" project.

Notify me when workflow has run

 Run

The **Image Name** is the full registry URL for an image.

The **Image Description** field is optional.



Home > Edge-CC > apps >  Add Execution Image >  Add Execution Image 2021-09-26 17.39.13

Image Name:

Image Description:

Managing Inference Deployments

Creating an Inference Deployment

1. Navigate to the KNIME Edge Control Plane workflows on KNIME Server.
2. In KNIME WebPortal, navigate to Edge -> apps -> Create Inference Deployment and execute the workflow.
3. A view will display with a handful of critical parameters (see below).
4. The remaining parameters can be filled out using default values or adjusted as needed.

Parameter	Description
Deployment Name	The value provided here becomes the API endpoint for <code>inferenceDeployments</code> ; this will be referred to as <code>edge_deployment_name</code> in this guide.
Deployment Description	Friendly text to describe the purpose & intent of the deployment.
Select a workflow to deploy	A visual file navigator to find and select the target workflow to deploy.
Select a Base Image Instance	The location of the Docker image which serves as the KNIME Edge Inference Agent. To follow this example, set the value to <code>registry.hubdev.knime.com/knime-edge/inference-agent:v4-4-2021-0630</code> .
Select the Edge location(s) for deployment	This determines the KNIME Edge cluster(s) that will propagate the Inference Deployment. This corresponds to the value for the <code>knimeEdgeName: <edge location></code> key used to create the edge cluster in a previous step.



A new snapshot of the selected workflow will be created and deployed instead of the workflow itself. The deployment description will be set as snapshot comment to help distinguish different snapshots.

Verifying an Inference Deployment

The following command should list the `edge_deployment_name` created in the previous step:

```
% curl -sL http://localhost:8081 | jq
{
  "inferenceDeployments": [
    {
      "endpoint": "http://localhost:8081/<edge_deployment_name>"
    }
  ]
}
```

The following command demonstrates how to interact with the deployed workflow. The request JSON is specific to the aforementioned [REST API for Sentiment Analysis](#) workflow and would need to be adapted for other workflows:

```
% curl -s -X POST -H "Content-Type: application/json" --data '{"content":["happy happy joy joy","sad bad mad mad"]}' http://localhost:8081/<edge_deployment_name> |jq
[
  {
    "Prediction (Document class) (Confidence)": 0.6459559392130747,
    "Prediction (Sentiment)": "Positive"
  },
  {
    "Prediction (Document class) (Confidence)": 0.5321978494130959,
    "Prediction (Sentiment)": "Positive"
  }
]
```

Updating an Inference Deployment

1. Navigate to the KNIME Edge Control Plane workflows on KNIME Server.
2. In KNIME WebPortal, navigate to Edge -> apps -> Update Inference Deployment and execute the workflow.
3. A view will display that allows you to change the parameters of an existing Inference Deployment with the exception of its name. The selected workflow cannot be changed but its version by selecting a different snapshot. You can also select to create and deploy a new snapshot of the workflow.

Deleting an Inference Deployment

1. Navigate to the KNIME Edge Control Plane workflows on KNIME Server.
2. In KNIME WebPortal, navigate to Edge -> apps -> Delete Inference Deployment and execute the workflow.
3. A view will display that allows you to select the Inference Deployment you want to delete.

Logging

Inference Deployments create logs during execution that can be helpful to troubleshoot problems. Each time a workflow is executed, events are logged similar as in KNIME Analytics Platform. When creating or updating an Inference Deployment (see above), the logging level can be specified to be either DEBUG, INFO, WARN or ERROR. From ERROR to DEBUG the logs become more detailed and verbose. By default, the logging level is set to WARN. If you encounter problems with an Inference Deployment and need more detailed information, you can update the Inference Deployment (see above) and set a more verbose logging level.

Retrieving execution logs from an Inference Deployment

There are two ways to inspect the logs for Inference Deployments.

a) Using the workflow on KNIME Server

1. Navigate to the KNIME Edge Control Plane workflows on KNIME Server.
2. In KNIME WebPortal, navigate to Edge -> apps -> Get Logs of Deployment and execute the workflow.
3. A view will display that allows you to select the Edge cluster you want to get logs from.
4. A view will display that allows you to select the Inference Deployment and the time range you want to get logs for.
5. The logs will be collected and provided as a downloadable zip file.



If you want to get the most recent logs, you might need to wait a few minutes before executing the workflow as it takes some time for the Edge cluster to collect and provide the logs.

b) Using the terminal

The following command demonstrates how to get logs from the Inference Deployment; in this example the log corresponds to the previously run scoring job run against the aforementioned [REST API for Sentiment Analysis](#) workflow:

```
> kubectl [-n <namespace>] logs <edge_deployment_name>-<podID>
Sep 01, 2021 3:31:02 PM org.apache.cxf.bus.osgi.CXFExtensionBundleListener addExtensions
INFO: Adding the extensions from bundle org.apache.cxf.cxf-rt-frontend-jaxrs (388)
[org.apache.cxf.jaxrs.JAXRSBindingFactory]
Sep 01, 2021 3:31:02 PM org.apache.cxf.bus.osgi.CXFExtensionBundleListener addExtensions
INFO: Adding the extensions from bundle org.apache.cxf.cxf-rt-transport-http (391)
[org.apache.cxf.transport.http.HTTPTransportFactory,
org.apache.cxf.transport.http.HTTPWSDLExtensionLoader,
org.apache.cxf.transport.http.policy.HTTPClientAssertionBuilder,
org.apache.cxf.transport.http.policy.HTTPServerAssertionBuilder,
org.apache.cxf.transport.http.policy.NoOpPolicyInterceptorProvider]
Sep 01, 2021 3:31:02 PM org.apache.cxf.bus.osgi.CXFExtensionBundleListener addExtensions
INFO: Adding the extensions from bundle org.apache.cxf.cxf-rt-transport-http-hc (392)
[org.apache.cxf.transport.http.HTTPConduitFactory,
org.apache.cxf.transport.ConduitInitiator]
Initializing Scoring Agent...
WARN      KNIME-Worker-1-Document Vector Applier 5:303 Node      The structures of both
active input data tables are not compatible.
WARN      KNIME-Worker-2-Category To Class 5:275 Node      The structures of both active
input data tables are not compatible.
WARN      KNIME-Worker-0-Gradient Boosted Trees Predictor (deprecated) 5:371 Node
The structures of both active input data tables are not compatible.
WARN      KNIME-Worker-1-Rule Engine 5:352 Node      The structures of both active input
data tables are not compatible.
```

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