

Technical Document

Containerized Niagara Guide

January 4, 2025

niagara⁴

Legal Notice

Tridium, Incorporated

3951 Western Parkway, Suite 350
Richmond, Virginia 23233
U.S.A.

Confidentiality

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation (Tridium). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, and Sedona Framework are registered trademarks, and Workbench are trademarks of Tridium Inc. All other product names and services mentioned in this publication that are known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and patent notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Copyright © 2025 Tridium, Inc. All rights reserved.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

For an important patent notice, please visit: <http://www.honpat.com>.

Contents

About this guide	5
Document change log	5
Related documentation	5
Chapter 1. Overview	7
Features	7
Using a container instead of a virtual machine	8
Where to use containers	8
Host system requirements	9
Limitations	9
Chapter 2. Backing up and restoring a registered container	11
Backing up the Docker container	11
Restoring backup DIST file to Docker container	13
Re-registering Docker image with licensing server	16
Chapter 3. Upgrading a Niagara container	19
Upgrading a container using Docker	19
Chapter 4. Licensing	23
Creating subscription license order	23
Setting up a container license	28
Copying a station from a container version	31
Verifying license check-in parameters	33
Deregistering a license from host ID	34
Replacing a host ID bound to license	37
Chapter 5. Glossary	43

About this guide

This topic contains important information about the purpose, content, context, and intended audience for this document.

Product Documentation

This document is part of the Niagara technical documentation library. Released versions of Niagara software include a complete collection of technical information that is provided in both online help and PDF format. The information in this document is written primarily for Systems Integrators. To make the most of the information in this book, readers should have some training or previous experience with Niagara software, as well as experience working with JACE network controllers.

Document Content

This document describes how to set up and use Containerized Niagara. Sections in this guide include chapters about container license, system requirements, and limitations.

Document change log

Changes to this document are listed in this topic.

January 4, 2025

- Added "Deregistering a license from host ID" and "Replacing a host ID bound to license" topics to the "Licensing" chapter (as of Niagara 4.15).

October 21, 2024

- Added "Backing up and restoring registered container" chapter.
- Added "Upgrading a Niagara container" chapter.

November 30, 2023

- In "*Creating subscription license order*" chapter, removed "Email Licenses (Optional)" checkbox.

Initial release publication: September 5, 2023

Related documentation

Additional information is available in the following documents.

- *Niagara Platform Guide*

Chapter 1. Overview

Containerized Niagara is a packaging mechanism for deploying and updating Niagara core. It is a Niagara framework stack that is packaged into a Docker[®] container with all required dependencies for easy deployment.

A container bundles the code defining a specific software application with its related configuration files, system libraries, system tools and dependencies such that the package is fully functional regardless of the cloud or non-cloud computing environment.

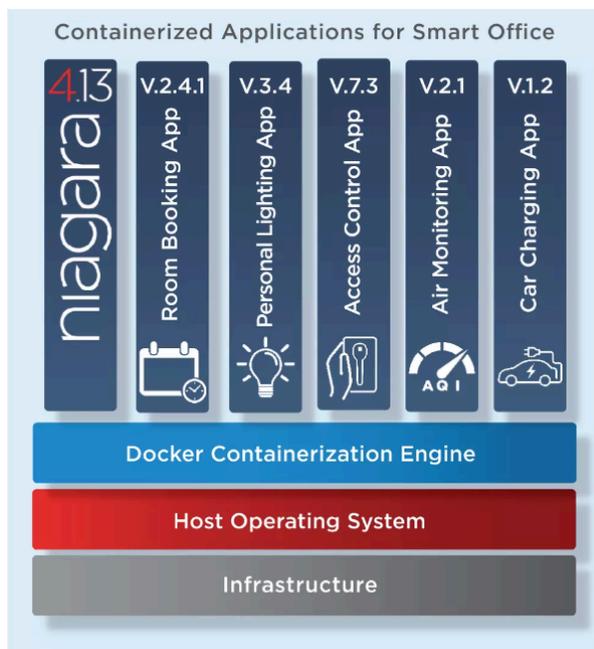
Advantages of using containers:

- Faster delivery of software
- More agile software development processes
- Easier portability of code from one computing environment to another

There are two primary use cases for deploying Niagara via container:

- Embedded deployment as an alternative to the Niagara Portability Software Development Kit. By bundling Niagara core (JRE, Niagara Framework, and modules) into a single deployment unit, you simplify the start and upgrade of Niagara systems compared to using a traditional Niagara Portability Software Development Kit (NPSDK) or Windows/Linux server installation.
- Niagara Development partner deploys Containerized Niagara to a third-party device or server.

Figure 1. Example deployment of Containerized Niagara on a partner-built mobile hub



Features

The following features are available in Containerized Niagara.

- The Niagara Framework is supplied in a Docker[®] container.
- You can use the hosting platform of your choice (for example, Azure, AWS, Google)

- If you use Enterprise Scale, you can choose the orchestration tool.
- AMD_x86 and Arm64 architecture are supported.
- Custom support is available for different architectures.
- Authorization models: file domain authorization or native domain authorization
- Subscription-based licensing using OPex instead of CAPex for purchasing flexibility
- Models that are released with Niagara 4.13:

Type	Part #
Supervisor	NCC-SUP-0
	NCC-SUP-1
	NCC-SUP-10
	NCC-SUP-100
	NCC-SUP-500
Supervisor Upgrade	NCC-SUP-UP-1
	NCC-SUP-UP-10
	NCC-SUP-UP-100
Supervisor Device Pack	NCC-SUP-DEV-10
	NCC-SUP-DEV-50
	NCC-SUP-DEV-100
	NCC-SUP-DEV-500
	NCC-SUP-DEV-1000

Type	Part #
Hardware Embedded	NCC-CPN-0001
	NCC-CPN-0002
	NCC-CPN-0005
	NCC-CPN-0010
	NCC-CPN-0025
	NCC-CPN-0100
	NCC-CPN-0200
Hardware Embedded Device Pack	NCC-CPN-DEV-1
	NCC-CPN-DEV-2
	NCC-CPN-DEV-10
	NCC-CPN-DEV-25
	NCC-CPN-DEV-50

Using a container instead of a virtual machine

Advantages using a container instead of a virtual machine are as follows:

- You do not need to maintain the host operating system.
- Containers require less storage and RAM overhead.
 - A container reuses the Linux Kernel of the host operating system.
 - Containers require less storage and RAM overhead.
 - A container reuses the Linux Kernel of the host OS.
 - Typically, a container is one hundredth of MB versus multiple GB for the virtual machine.
- Containers are more portable.

Where to use containers

There are two primary use cases for deploying Niagara as a container: embedded deployment as an alternative for NPSDK (Niagara Portability Software Development Kit) and Supervisor deployment in a cloud service or on a server device locally.

Next generation NPSDK deployment

Note that containers are not appropriate for devices targeted by Smart Edge SDK. If a device is capable of running a container, it can run full Niagara.

- OEM-specific engagement to support custom hardware
- Works out of box
- Differences from NPSDK
 - Faster onboarding time by OEM engineering team
 - Limited opportunities for customization. See also "Limitations".

Standard cloud service deployment

- N4Supervisor for AWS or Azure.

Host system requirements

Before you start to use Containerized Niagara, check the host system requirements in this section.

- Linux x86_64 (AMD64) or Linux ARM64 system with Docker Engine installed.
 - Docker runtime on Windows or Mac is not supported.
 - (Optional) install Docker Compose to use `docker-compose.yml` for configuring containers.
- Docker volume support
 - Niagara stores station, history, and alarm data as standard files on volumes.
- RAM/disk requirements
 - RAM requirements vary by application. You can adjust the Java heap size allocated to Niagara station using container environment variables.
 - Standard Niagara container size on disk is about 350MB, which includes JRE and requires OS libraries and all standard modules.
- Container management
 - You can use Kubernetes or other orchestration tools to start, stop, upgrade, and configure the Niagara container.

Limitations

The following limitations exist when using Niagara containers.

Scalability

- Niagara containers do not provide horizontal scalability. It is a single instance of Niagara. You can increase the capacity by allocating additional memory or using a faster CPU.
- Containerized Niagara uses IP-based drivers only.
- Serial drivers and custom drivers such as BACnet MS/TP, CCN, BPort are not supported. The container engine does not permit applications running in container direct access to serial ports.

No data recovery service

If the container is terminated or the hosting platform has a power loss, there is no support for data recovery. All data will be lost since the last station backup.

Station only

Containerized Niagara is not suitable for deploying Workbench. You need another laptop or PC to interact with the container and run Workbench.

Outbound internet connectivity required

A container deployment must be able to connect to the Entitlement Server to validate its license.

Chapter 2. Backing up and restoring a registered container

The following sections describe how to back up and restore an already registered Niagara container using Niagara Workbench.

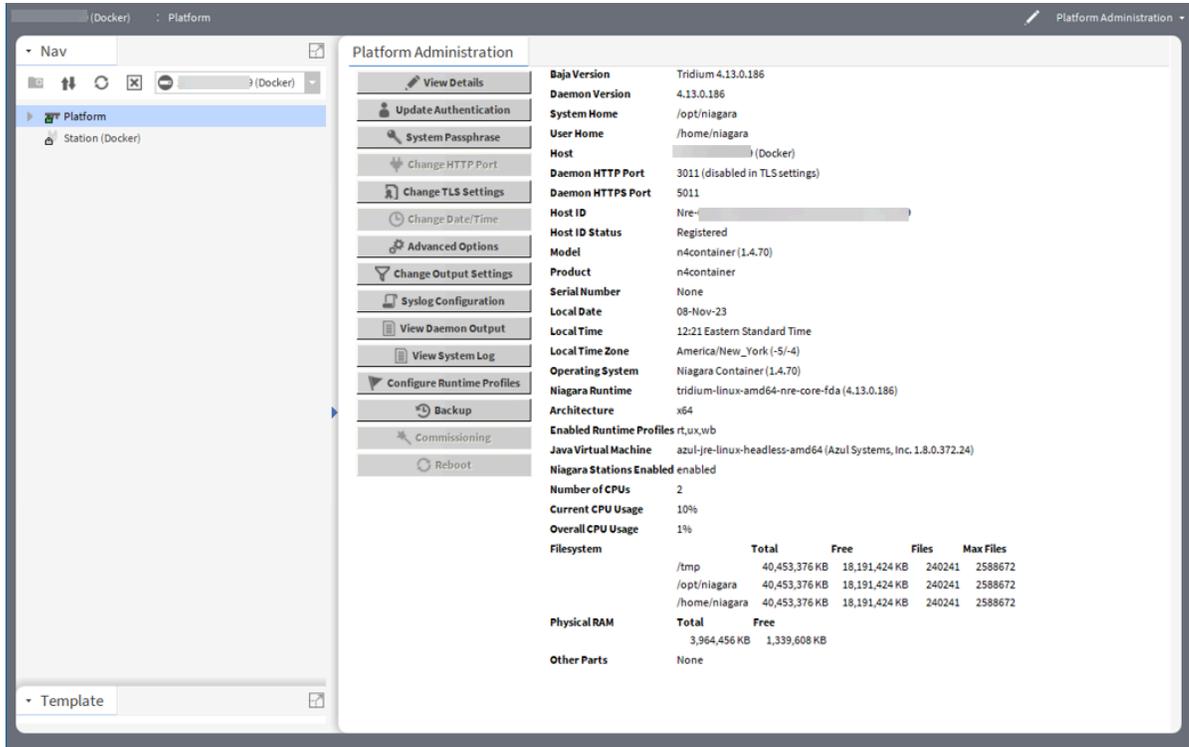
Backing up the Docker container

The following section describes how to back up a Docker container in Niagara Workbench.

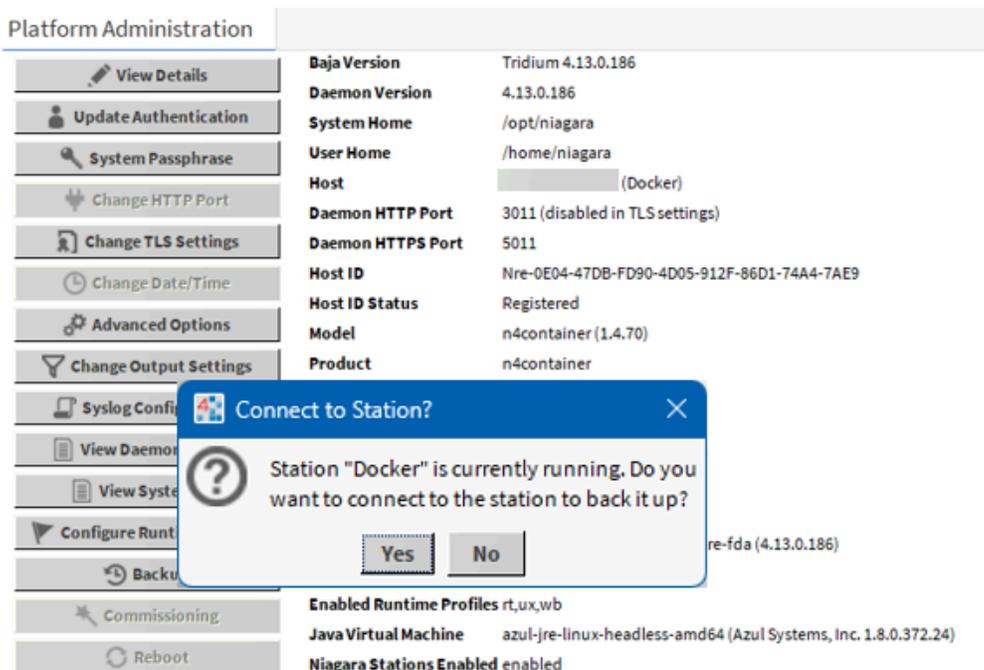
Prerequisites:

- You have a running Docker container.

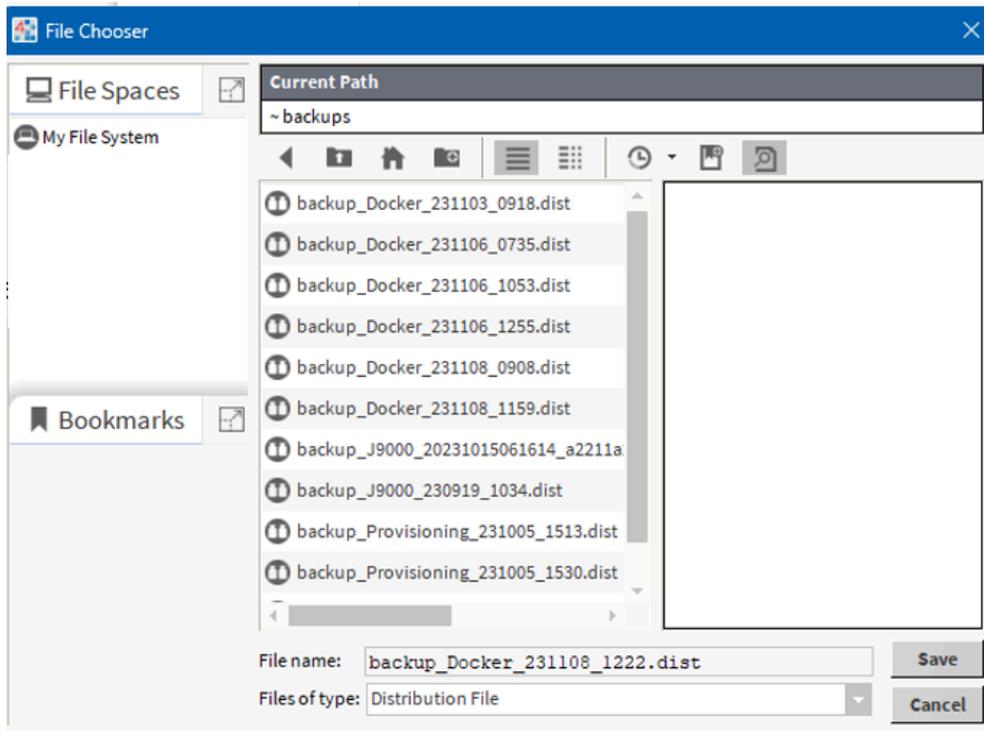
Step 1. In Workbench, after connecting to the container's platform, open the **Platform Administration** and click **Backup**.



In the **Connect to Station?** window, you are prompted to allow Workbench to authenticate and connect the station in order to back it up.



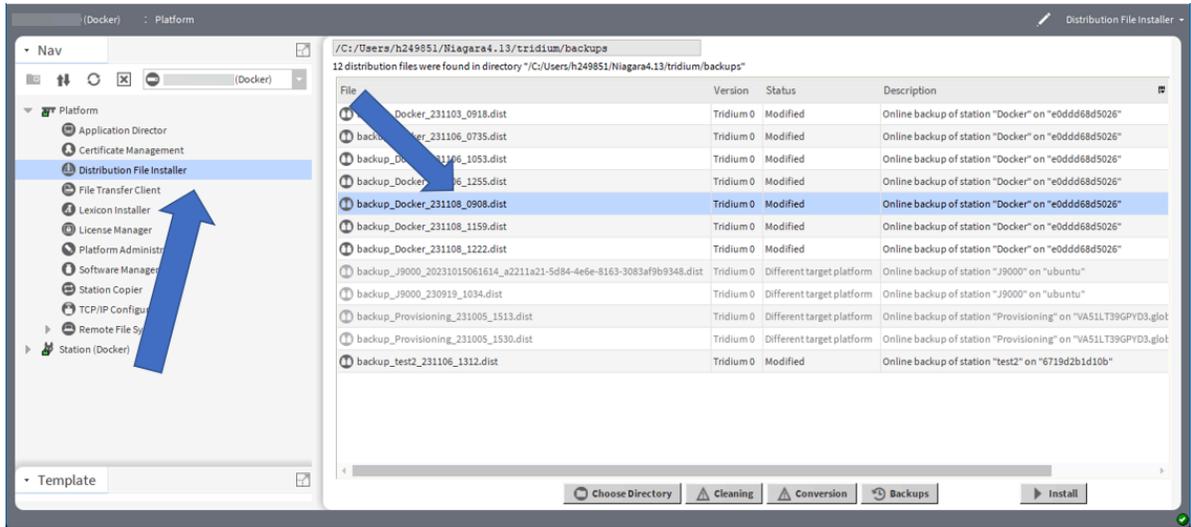
Step 2. In the **File Chooser** window, which defaults to the `backups` folder, click **Save** to save the backup file.



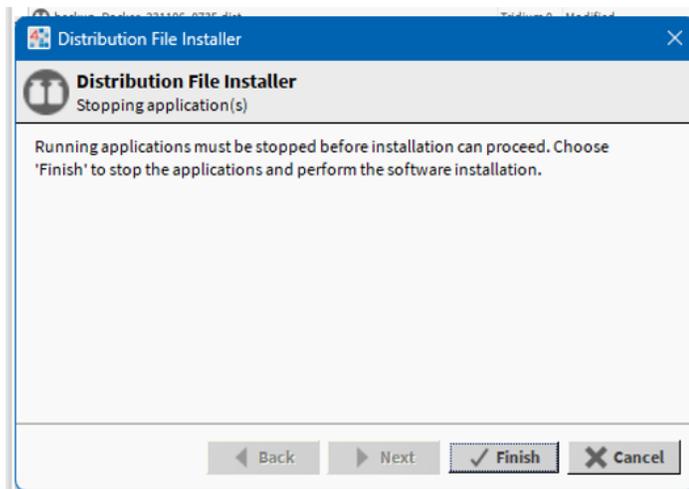
Restoring backup DIST file to Docker container

The following section describes how to restore the backup .dist file to the Docker Container in Niagara Workbench.

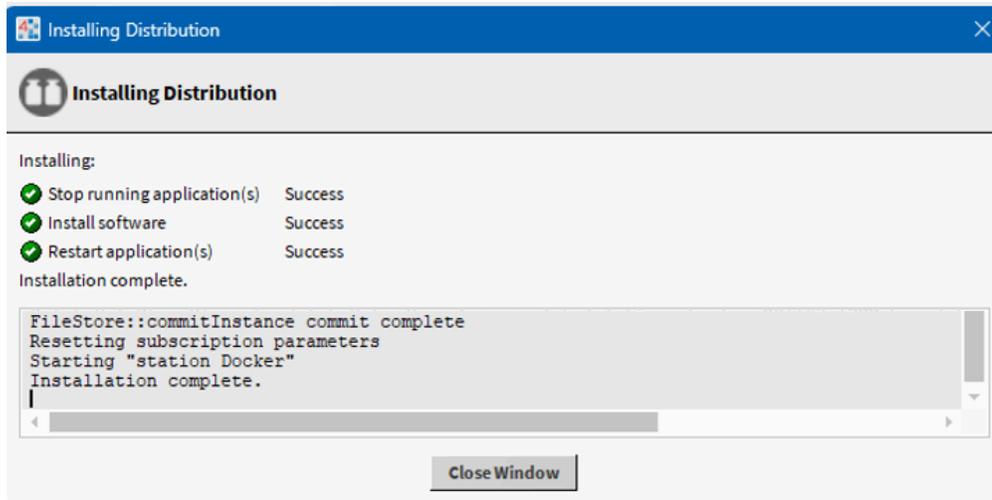
- Step 1. In the Workbench platform, to select and install the backup, select the **Distribution File Installer**.



The Distribution File Installer window opens.



- Step 2. To stop all running applications within Niagara and allow Workbench to install the backup to the container, click **Finish**.
- Step 3. Observe the installing status and after completion, click **Close Window**.



Result

- In the Docker console, you can view the installation of the backup.

```

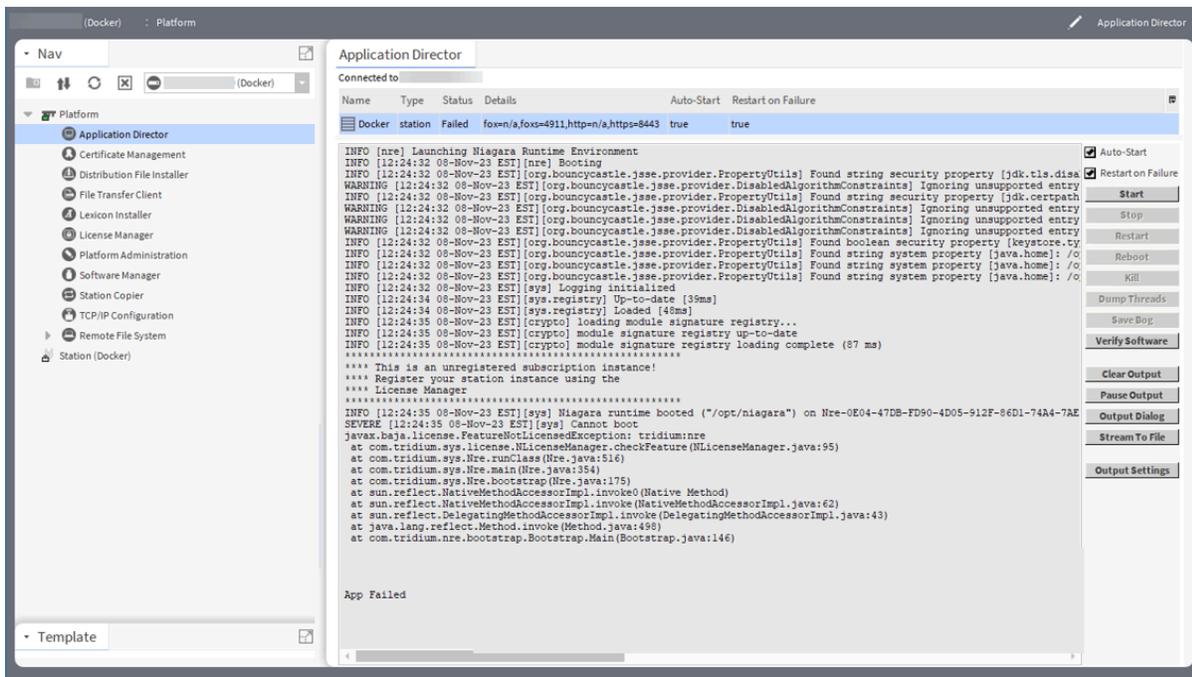
niagara-ntagara-1 INFO [12:24:25 08-Nov-23 EST][stationRegistry] station Docker stopping
niagara-ntagara-1 INFO [12:24:28 08-Nov-23 EST][stationRegistry] station Docker exited with status 0
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/console.txt deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/console_backup_231108_1218.txt deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/console_backup_231108_1202.txt deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/etc/platform.bog deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/etc/platform_backup_231108_1222.bog deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/Alarm/Alarm.adb deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/Alarm/ deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg0/SecurityHistory.hdb deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg0/ deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg4/LogHistory.hdb deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg4/DeviceNetworkJobHistoryRecord.hdb deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg4/ deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteDirectory /home/niagara/stations/Docker/history/station/seg4' deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg7/AuditHistory.hdb deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteDirectory /home/niagara/stations/Docker/history/station/seg7' deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg1/AuditHistoryRecord.hdb deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteFile /home/niagara/stations/Docker/history/station/seg1/NetworkStepHistoryRecord.hdb deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteDirectory /home/niagara/stations/Docker/history/station/seg1' deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteDirectory /home/niagara/stations/Docker/history/station' deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] DeleteFileStoreElement::deleteDirectory /home/niagara/stations/Docker/history' deleted
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] FileCachedFileStoreElement::commit file written /home/niagara/stations/Docker/config.bog
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] FileCachedFileStoreElement::commit file written /home/niagara/etc/platform.bog
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] KeyRingImportFileStoreElement::commit imported data to keyring in /home/niagara/security/.kr
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] FileCachedFileStoreElement::commit file written /home/niagara/daemon/daemon.properties
niagara-ntagara-1 INFO [12:24:30 08-Nov-23 EST][file] FileCachedFileStoreElement::commit file written /home/niagara/etc/license.properties
    
```

- The Docker console displays the following information after the restore has completed and the station has attempted to start.

```

niagara-ntagara-1 INFO [12:24:31 08-Nov-23 EST][stationRegistry] station Docker watchdog thread started [tid = 51]
niagara-ntagara-1 INFO [12:24:31 08-Nov-23 EST][stationRegistry] station Docker starting
niagara-ntagara-1 INFO [12:24:35 08-Nov-23 EST][stationRegistry] station Docker exited with status -3
niagara-ntagara-1 SEVERE [12:24:35 08-Nov-23 EST][stationRegistry] station Docker failed, rc = -3
    
```

- At this point, in Workbench re-register the subscription into the backup. In Workbench, the Application Director displays that the station startup may have failed. The console shows `FeatureNotLicensedException`.

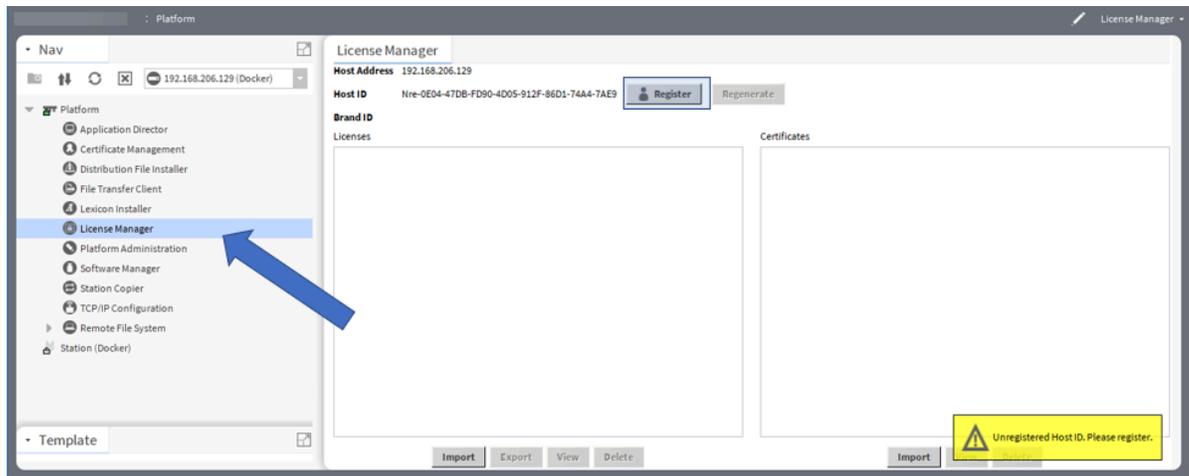


Re-registering Docker image with licensing server

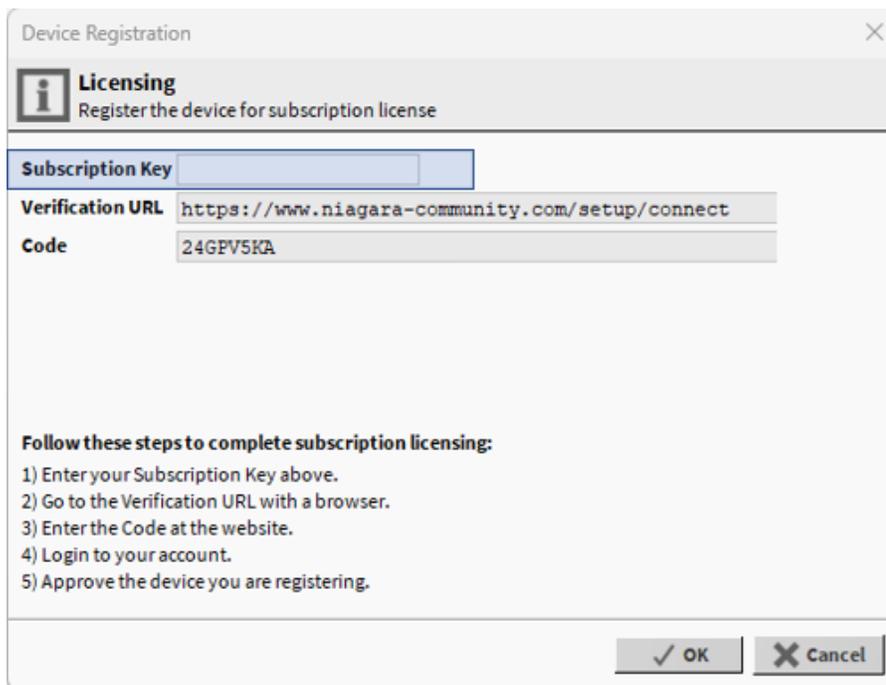
The following section describes how to re-register the Docker image with the licensing server in Workbench.

The platform **License Manager** allows you to re-register the backup and link it back to the subscription.

- Step 1. In Workbench, open the platform, and select **License Manager**. In the lower right corner of the screen, a yellow popup window informs you that you must register the Host ID.



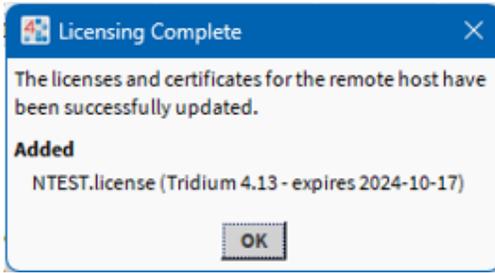
- Step 2. In the **License Manager**, click **Register**.



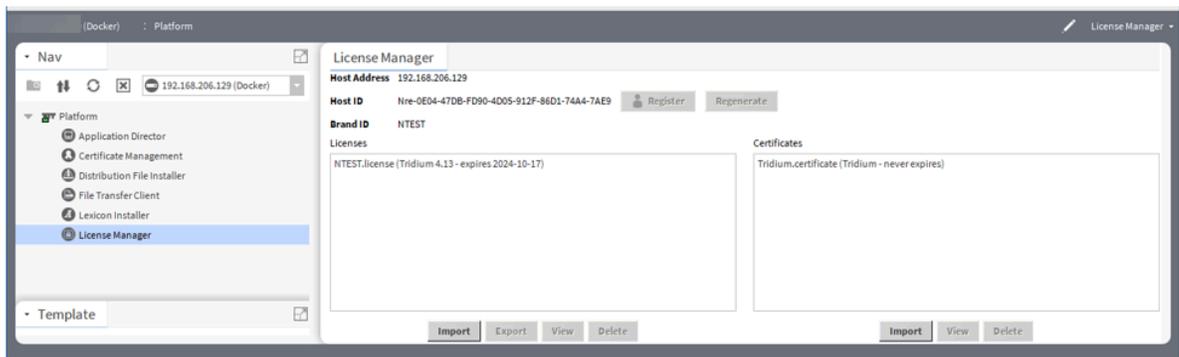
The **Device Registration** window opens.

- Step 3. For **Subscription Key**, enter the original subscription key used for this container's NRE Host ID. The subscription key is delivered via email, or you can contact your System Integrator (SI) if you have questions.
- Step 4. For **Verification URL**, use the same procedure you used to activate the container for the first time.

Step 5. For Code, click the URL, enter the temporary code, and click OK.



The Licensing Complete popup window notifies you that the licenses and certificates for the remote host have been successfully updated. Your backup is re-registered with the Licensing server, the container is restored from the backup, and the subscription has been re-attached and will start up.



Step 6. Restart the station using the Application Director.

Result

The Docker console displays that the station and the subscriptions have successfully completed.

```

niagara-niagara-1 INFO [12:26:11 08-Nov-23 EST][licensing.subscription] Polling for registration status (will poll for 5 minutes)
niagara-niagara-1 INFO [12:26:11 08-Nov-23 EST][updatedaemon] access token poll started, poll interval 5 seconds
niagara-niagara-1 INFO [12:27:00 08-Nov-23 EST][licensing.subscription] Access token retrieval successful.
niagara-niagara-1 INFO [12:27:17 08-Nov-23 EST][licensing.subscription] This device is registered with the subscription licensing system.
niagara-niagara-1 INFO [12:27:18 08-Nov-23 EST][licensing.subscription] Certificate request completed successfully.
niagara-niagara-1 INFO [12:27:18 08-Nov-23 EST][updatedaemon] license request successful
niagara-niagara-1 INFO [12:27:18 08-Nov-23 EST][file] FileCachedFileStoreElement::commit file written /home/niagara/security/subscription/licenses/NTEST.license
niagara-niagara-1 INFO [12:27:46 08-Nov-23 EST][stationRegistry] station Docker watchdog thread started [tid = 63]
niagara-niagara-1 INFO [12:27:46 08-Nov-23 EST][stationRegistry] station Docker starting
niagara-niagara-1 INFO [12:27:58 08-Nov-23 EST][stationRegistry] station Docker startup complete
    
```

Chapter 3. Upgrading a Niagara container

The following procedure describes how to upgrade a Niagara container using Docker.

Upgrading a container using Docker

Prerequisites:

Docker is installed and running on your device.

- Step 1. To view what Docker images are loaded, open a command prompt and run the command: `docker image ls`.

```
tridium@tridium-virtual-machine:~/niagara$ docker image ls
REPOSITORY          TAG          IMAGE ID        CREATED         SIZE
vykon-fda-amd64    4.13.2.18   456e91cb8f27   10 months ago  410MB
vykon-fda-amd64    4.13.1.14   9736cce5525a   11 months ago  391MB
vykon-fda-amd64    4.13.0.186  2c2b7bfa6a15   12 months ago  405MB
hello-world        latest      9c7a54a9a43c   17 months ago  13.3kB
```

A list of all Docker images stored locally will be displayed including details such as repository name, tag, image ID and creation time.

- Step 2. To view what Docker volumes are used, run the `docker volume ls` command.

```
tridium@tridium-virtual-machine:~/niagara$ docker volume ls
DRIVER      VOLUME NAME
local      niagara_niagara-user-home
tridium@tridium-virtual-machine:~/niagara$
```

A list of all Docker volumes that are currently available on your device are displayed.

NOTE: It uses `niagara_niagara-user-home` which is mapped in the `docker-compose.yml`.

- Step 3. To load a new Docker image from Tridium, run the `docker load <` command.

```
tridium@tridium-virtual-machine:~/NiagaraContainerImages$ docker load<vykon-fda-amd64-4.14.0.162.tar
4a1518ebc26e: Loading layer [=====>] 75.19MB/75.19MB
0da75023adea: Loading layer [=====>] 67.86MB/67.86MB
f12d5ea90c4f: Loading layer [=====>] 1.659MB/1.659MB
e27f25c4915f: Loading layer [=====>] 4.096kB/4.096kB
06e94a0cabe4: Loading layer [=====>] 3.072kB/3.072kB
0f7d796aee40: Loading layer [=====>] 3.072kB/3.072kB
ef54f5a60d74: Loading layer [=====>] 136.5MB/136.5MB
5f604a116c7f: Loading layer [=====>] 123.8MB/123.8MB
86f184d8f1ef: Loading layer [=====>] 2.56kB/2.56kB
39513d1309b7: Loading layer [=====>] 46.59kB/46.59kB
77e3d6e8463e: Loading layer [=====>] 15.87kB/15.87kB
03fc19d57226: Loading layer [=====>] 21.5kB/21.5kB
a6de71766449: Loading layer [=====>] 39.94kB/39.94kB
15a61fb92d87: Loading layer [=====>] 34.3kB/34.3kB
21e61d50c794: Loading layer [=====>] 34.3kB/34.3kB
d04d6f0fd5b3: Loading layer [=====>] 1.226MB/1.226MB
Loaded image: vykon-fda-amd64:4.14.0.162
```

The system integrator loads the `.tar` file, which is, in this example, the `vykon-fda-amd64-4.14.0.162.tar` file.

- Step 4. To confirm the Docker loaded image for use, run the `docker image ls` command.

```
tridium@tridium-virtual-machine:~/niagara$ docker image ls
REPOSITORY          TAG                IMAGE ID           CREATED            SIZE
vykon-fda-amd64     4.14.0.162        bb51a5298092      3 months ago      402MB
vykon-fda-amd64     4.13.2.18         456e91cb8f27      10 months ago     410MB
vykon-fda-amd64     4.13.1.14         9736cce5525a      11 months ago     391MB
vykon-fda-amd64     4.13.0.186        2c2b7bfa6a15      12 months ago     405MB
hello-world         latest            9c7a54a9a43c      17 months ago     13.3kB
tridium@tridium-virtual-machine:~/niagara$
```

In this example, you can use five Docker images.

Step 5. Edit the `docker-compose.yml` file.

```
26 # image: vykon-fda-amd64:4.13.0.186 # Specify the provided image name and version, for example: niagara/fda/amd64/base:4.13.0.182_1.4.67.
27 # image: vykon-fda-amd64:4.13.0.186 # use docker load~/NiagaraContainerImages/Niagara_NDA_Amd64_Base_4.13.0.186.tar
28 # image: vykon-fda-amd64:4.13.1.14 # use docker load~/NiagaraContainerImages/Niagara_NDA_Amd64_Base_4.13.1.14.tar
29 # image: vykon-fda-amd64:4.13.2.18 # use docker load~/NiagaraContainerImages/Niagara_NDA_Amd64_Base_4.13.2.18.tar
30 image: vykon-fda-amd64:4.14.0.162 # use docker load~/NiagaraContainerImages/Niagara_NDA_Amd64_Base_4.14.0.162.tar
```

In this example, the active image line is copied and commented out with the `#`, then it is pasted back. The image file name and tag are changed as follows: `image: Vykon-fda-amd64:4.14.0.162`.

Step 6. To start the container, run the `docker compose up` command.

```
tridium@tridium-virtual-machine:~$ cd niagara
tridium@tridium-virtual-machine:~/niagara$ docker compose up
WARN[0000] /home/tridium/niagara/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 1/0
✔ Container niagara-niagara-1 Created                                0.0s
Attaching to niagara-1
niagara-1 | INFO | init.sh | Initializing Niagara container
niagara-1 | INFO | init.sh | Container is starting with root-level commands
niagara-1 | INFO | init.sh | Setting timezone to America/New_York
niagara-1 | INFO | copy_modules.sh | Checking for removed modules
niagara-1 | INFO | copy_modules.sh | Copying core Niagara modules to /home/niagara/modules
niagara-1 | INFO | init.sh | niagarad.sh running as PID 1561
niagara-1 | INFO | niagarad.sh | Container version is 1.5.71
niagara-1 | INFO | niagarad.sh | Hard open file limit is 1048576
niagara-1 | INFO | niagarad.sh | Starting niagarad.
niagara-1 | INFO | niagarad.sh | niagarad running as PID 1564
niagara-1 | INFO [12:04:01 10-Oct-24 EDT][crypto.registry] module signature registry up-to-date
niagara-1 | INFO [12:04:01 10-Oct-24 EDT][crypto.registry] module signature registry load complete (85ms)
niagara-1 | INFO [12:04:01 10-Oct-24 EDT][niagarad] starting, niagara_user_home=/home/niagara
niagara-1 | WARNING [12:04:01 10-Oct-24 EDT][webserver] using default TLS server certificate 'default' is not recommended
niagara-1 | INFO [12:04:02 10-Oct-24 EDT][webserver] web server threads started [threadCount = 5]
niagara-1 | INFO [12:04:02 10-Oct-24 EDT][stationRegistry] station registry starting
niagara-1 | INFO [12:04:02 10-Oct-24 EDT][stationRegistry] station Docker watchdog thread started [tid = 32]
niagara-1 | INFO [12:04:02 10-Oct-24 EDT][niagarad] startup complete (3314ms)
niagara-1 | INFO [12:04:02 10-Oct-24 EDT][stationRegistry] station Docker starting
niagara-1 | INFO [12:04:19 10-Oct-24 EDT][stationRegistry] station Docker startup complete
```

Step 7. Confirm the upgrade via a platform connection in Workbench.

Platform Administration

View Details	Baja Version Tridium 4.14.0.162
Update Authentication	Daemon Version 4.14.0.162
System Passphrase	System Home /opt/niagara
Change HTTP Port	User Home /home/niagara
Change TLS Settings	Host 192. (Docker)
Change Date/Time	Daemon HTTP Port 3011 (disabled in TLS settings)
Advanced Options	Daemon HTTPS Port 5011
Change Output Settings	Host ID Nre-
Syslog Configuration	Host ID Status Registered
View Daemon Output	Model n4container (1.5.71)
View System Log	Product n4container
Configure Runtime Profiles	Serial Number None
Backup	Local Date 10-Oct-24
Commissioning	Local Time 12:11 Eastern Daylight Time
Reboot	Local Time Zone America/New_York (-5/-4)
	Operating System Niagara Container (1.5.71)
	Niagara Runtime tridium-linux-amd64-nre-core-fda (4.14.0.162)
	Architecture x64
	Enabled Runtime Profiles rt,ux,wb
	Java Virtual Machine azul-jre-linux-headless-amd64 (Azul Systems, Inc. 1.8.0.412.20)
	Niagara Stations Enabled enabled

In this example, an Ubuntu virtual machine is used to run the Docker environment, which does not contain Workbench. A localhost is used as the IP address on the PC in Workbench. You will connect to the platform to verify the update and use the IP address of your environment.

Chapter 4. Licensing

Niagara container deployment is tightly coupled to the Niagara Entitlement Server (licensing service). The container has to reach out to the Entitlement Server at least once a day to re-authenticate and to receive the authentication token to continue running.

Traditional Niagara licensing uses a unique and permanent Host ID as a key to license. Containers are inherently portable and intended to operate in a variety of virtualized environments. Containers include a single hard-coded Host ID and node-locked license with an expiration date. A separate Host ID and thus container image is required for each license variant.

The Entitlement Server provides an online service to monitor and distribute licenses. While each container will be assigned a uniquely generated identity, the central Entitlement Server can detect abuses such as trying to reuse the same identity. Since a license must be periodically renewed, the opportunity for abuse is limited. In addition, the risk of abuse is mitigated by the following:

- Contractual auditing of units is deployed.
- Providing a unique container variant to customer.
- Each license has an expiration date.

Creating subscription license order

The following section describes how to create an order for a subscription license.

Prerequisites:

You have created a Niagara Central account (<https://www.niagara-community.com>) and you have access to Niagara Licensing.

- Step 1. To create a subscription license order, select **Products > Product Catalogue > Buy Model**, and click **Next**.

The screenshot shows the Niagara Licensing web interface. At the top, there is a navigation bar with the logo and links for Asset Manager, Marketplace, Community, Software, University, and Tridium. Below this is a breadcrumb trail: Organization > Production > Products > Licenses > Orders > Software. A search bar is present with a 'Search' button. The main content area features a progress bar with six steps: 1. Destination, 2. Req'd Option, 3. Add'l Option (current), 4. Dep't Option, 5. Order summary, and 6. Check out. The 'Add'l Option' step contains two sections: 'Product Information' with fields for Product (SUP-100-SUBSCRIPTION), Description (Test Subscription Model), and Quantity (10); and 'Who gets it?' with fields for Select Org and Project Name (currently showing '_01_SUBSCRIPTION'). At the bottom of the form are 'CANCEL' and 'NEXT' buttons.

Step 2. On the **Add'l Option** tab, add upgrades and add-ons, and click **Next**.

The screenshot displays the Niagara Licensing web application interface. At the top, the logo 'niagara licensing' is visible, along with navigation links for 'Asset Manager', 'Marketplace', 'Community', 'Software', 'University', and 'Tridium'. Below the logo, there are dropdown menus for 'Organization', 'Production', 'Products', 'Licenses', 'Orders', and 'Software', with 'Admin, S' on the right. A search bar with a 'Search' button is present. The breadcrumb trail reads 'Home > Product Catalogue > Buy Model'. A progress bar at the top of the main content area shows six steps: '1. Destination', '2. Req'd Option', '3. Add'l Option' (highlighted), '4. Dep't Option', '5. Order summary', and '6. Check out'. The 'Product Information' section includes the following details:

Product	: SUP-100-SUBSCRIPTION
Description	: Test Subscription Model
Quantity	: 10
Organization Name	:
Project Name	: _01_SUBSCRIPTION

The 'Configure Options' section shows three expandable categories: 'SWOptions (0) >>', 'Maintenance (0) >>', and 'Upgrades (0) >>'. A large box below these categories contains the text 'No record available.'. At the bottom of the form, there are 'CANCEL' and 'NEXT' buttons, with a mouse cursor hovering over the 'NEXT' button.

- Step 3. On the **Order summary** tab, select **Additional Options** and/or **Dependent Options** if applicable and click **CHECK OUT NOW**.
- Step 4. On the **Check out** tab, enter and confirm the PO number, and click **Process Cart** to finalize the license order.

Additional Emails:

Submitted To: Tridium, Inc.

Submitted By:

PO Number **: DEMO_SUB_ORDERING

Confirm PO Number **: DEMO_SUB_ORDERING

Order Notes:
300 characters limit

Verticals: BAS Energy
HVAC Retail
Industrial Lighting
DCIM Security

Item List

Line Number	Quantity	Product	Destination
10	10	SUP-100-SUBSCRIPTION	_01_SUBSCRIPTION

The license order is complete.

niagara licensing

Asset Manager Marketplace Community Software University Tridium

Organization Production Products Licenses **Orders** Software Admin, S

Category... Search... Search

Home > Placed Order

Purchase Order Information

Tridium, Inc.
SO : (Unbilled)
PO : DEMO_SUB_ORDERING [Change](#)

Created On : Feb 17,2023
Last Updated : Feb 17,2023
Billing Report : -
Total Qty : 10

Notes : -
Email : -
POUpdatedBy :

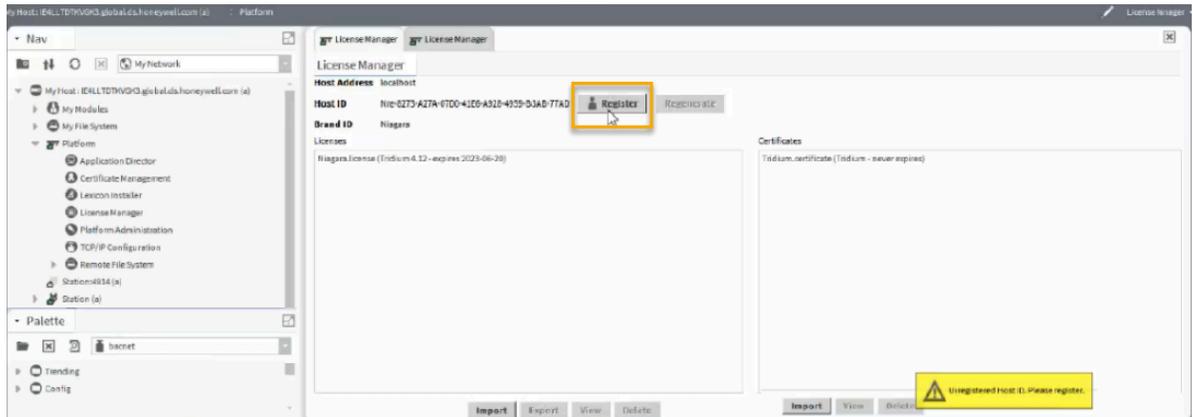
Order Lines Audit

Line #	Qty	Product	Order Type	Created By	Placed Order
10	10	SUP-100-SUBSCRIPTION	New License	Admin, SystemTest	-

You can view the generated license keys, which are unbound, by clicking the appropriate license subscription in the **Product** column.

NRE ID	Subscription License Key	Status	Expiration Date
-	B15E-	disabled	Feb 17,2024
-	6536-	disabled	Feb 17,2024
-	8399-	disabled	Feb 17,2024
-	8F92-	disabled	Feb 17,2024
-	50E2-	disabled	Feb 17,2024
-	BBF9-	disabled	Feb 17,2024

Step 5. To register the subscription instance, connect to the Workbench platform, navigate to the **License Manager** view, and click **Register**.



Step 6. From the newly created list of subscription license keys, paste the license key into the **Subscription Key** field, and follow the steps provided in the **Device Registration** window to complete the subscription licensing.

Device Registration

Licensing
Register the device for subscription license

Subscription Key

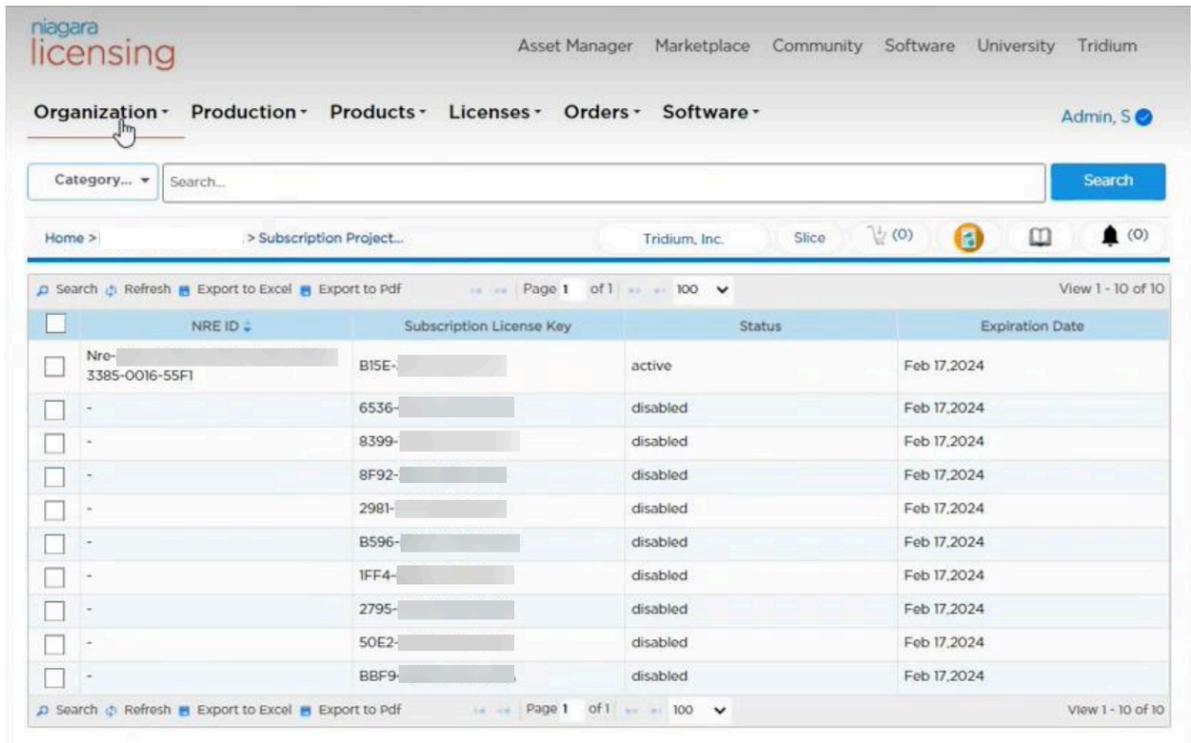
Verification URL

Code

Follow these steps to complete subscription licensing:

- 1) Enter your Subscription Key above.
- 2) Go to the Verification URL with a browser.
- 3) Enter the Code at the website.
- 4) Login to your account.
- 5) Approve the device you are registering.

Step 7. To allow license binding, authorize the device.



Your subscription is activated and bound to Host ID (NRE-ID).

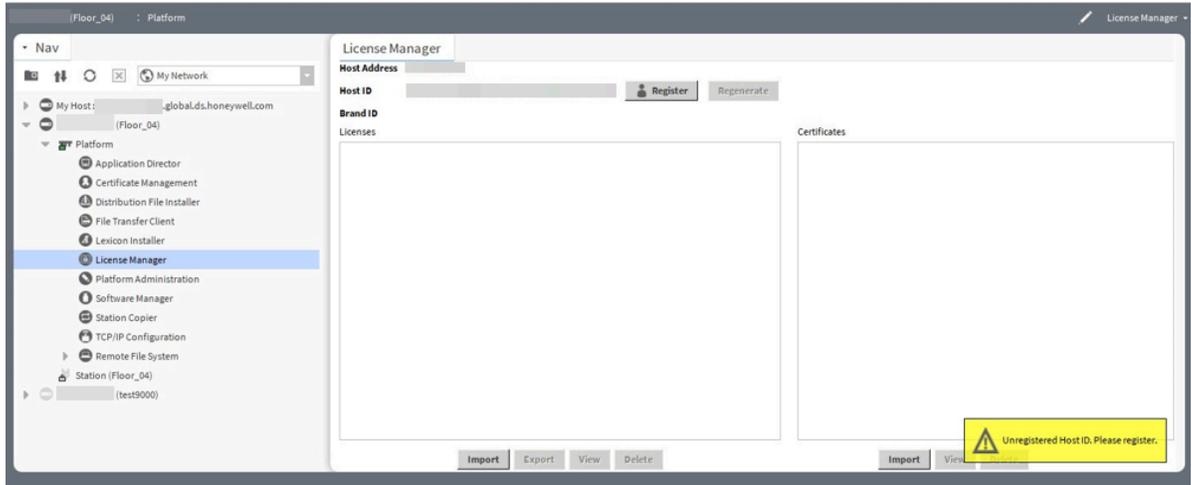
Setting up a container license

In the following section, you learn how to set up a container license.

Prerequisites:

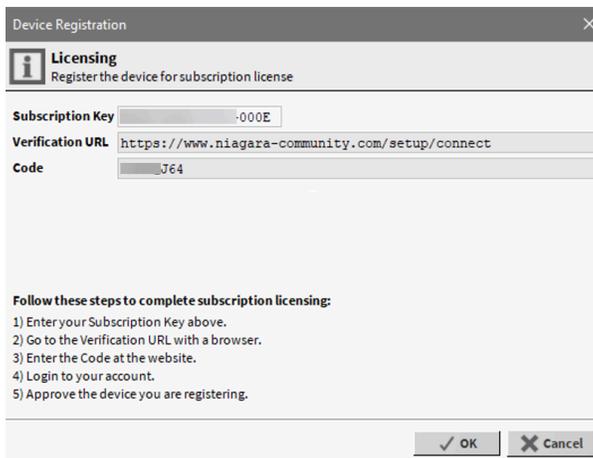
You have created a Niagara Community account. Signing up is free and the account is free of charge. If you do not have an account, you can sign up here: https://www.niagara-community.com/Sign_Up.

- Step 1. Start the Docker container.
- Step 2. In Workbench, to open a platform connection, from the menu bar, select **File > Open > Open Platform**.
- Step 3. Change default system passphrase and default platform credentials.
- Step 4. Open the **License Manager** view.



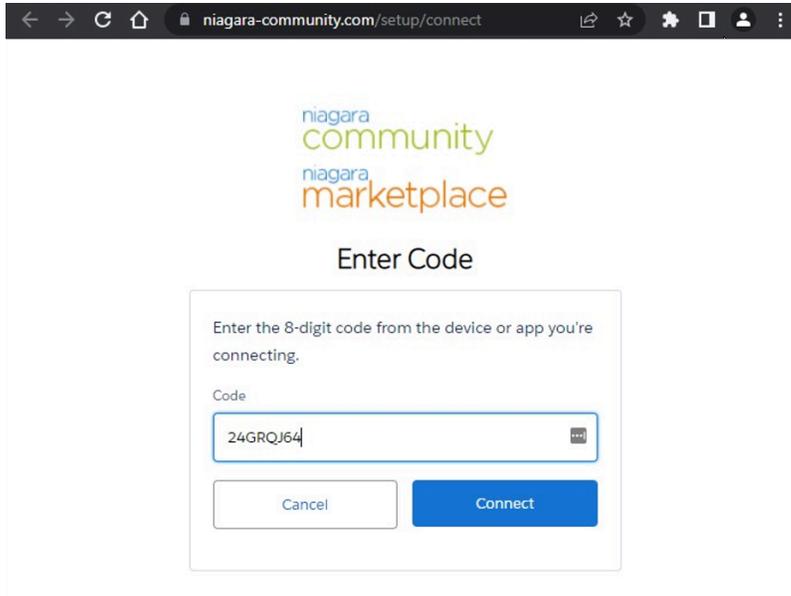
You should see the Host ID in the top left corner and a yellow warning dialog in the bottom right corner.

Step 5. Click **Register** and paste the subscription license key into the empty field.



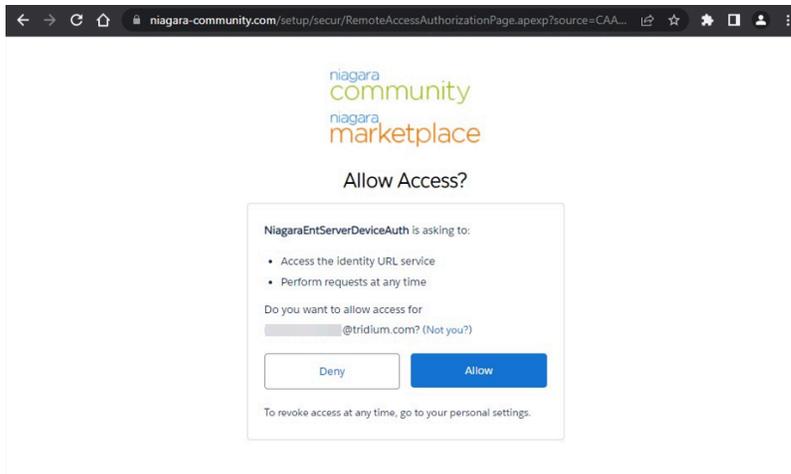
Step 6. To open the browser link, click in the **Verification URL** field.

Step 7. Paste the 8-digit code from the **Device Registration** dialog box into the **Code** field in the browser and click **Connect**.

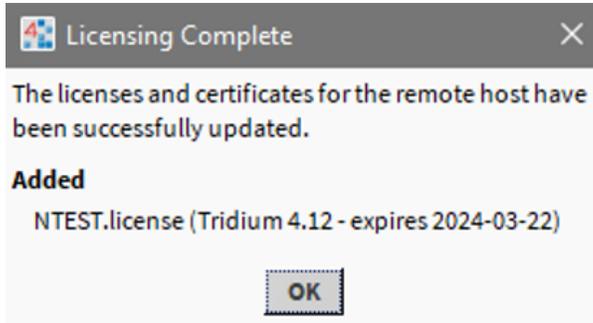


NOTE: If you were not previously logged into Niagara Central, provide your login credentials here.

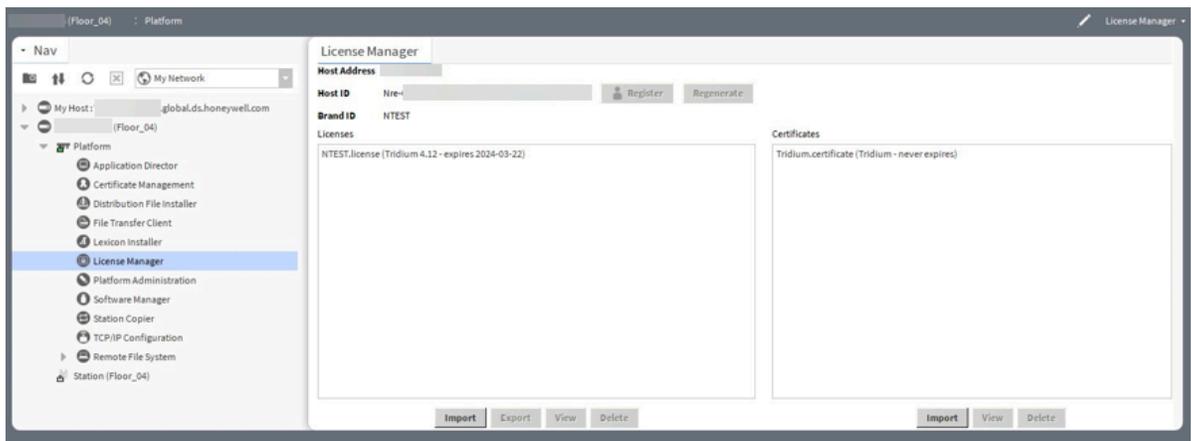
- Step 8. On the **Remote Access Authorization** page, click **Allow** to complete license binding in Niagara Central, and click **Continue**.



- Step 9. Return to Workbench and click **OK** in the **Device Registration** dialog box. A dialog box indicates whether the licensing of the device was successful or failed.



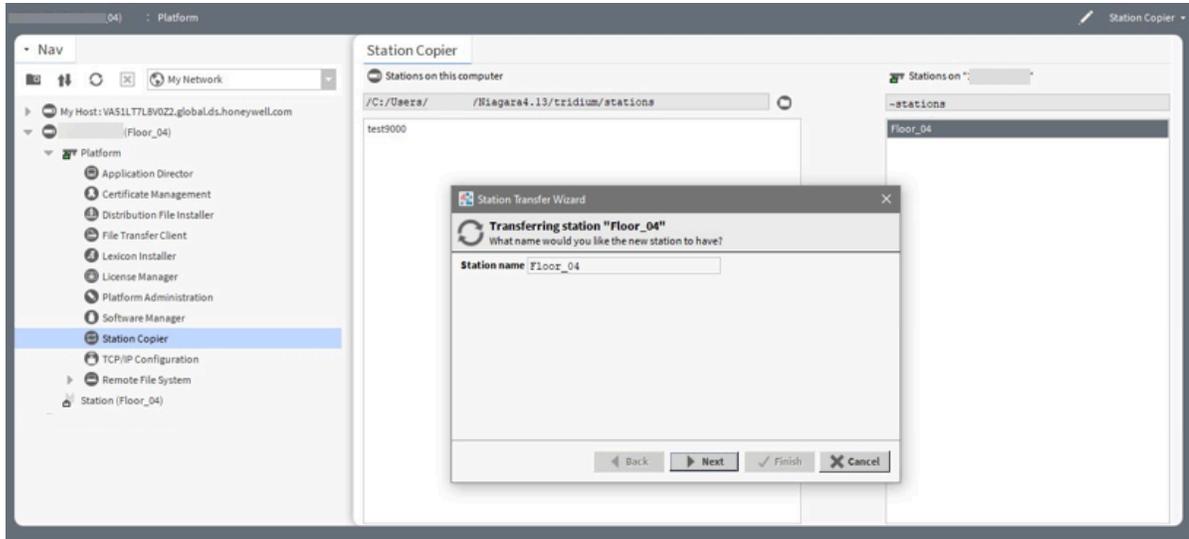
Both the license and the certificate are displayed in the License Manager view.



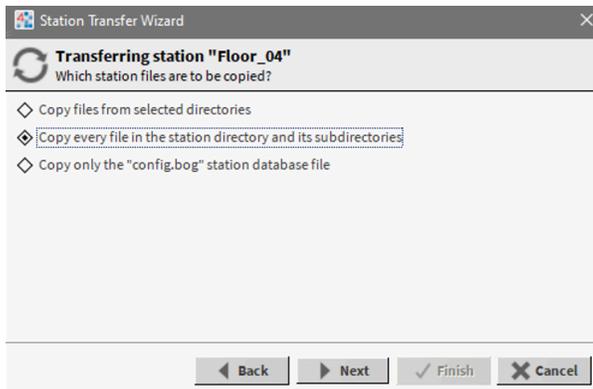
Copying a station from a container version

This section describes how to copy a station from a previous container version.

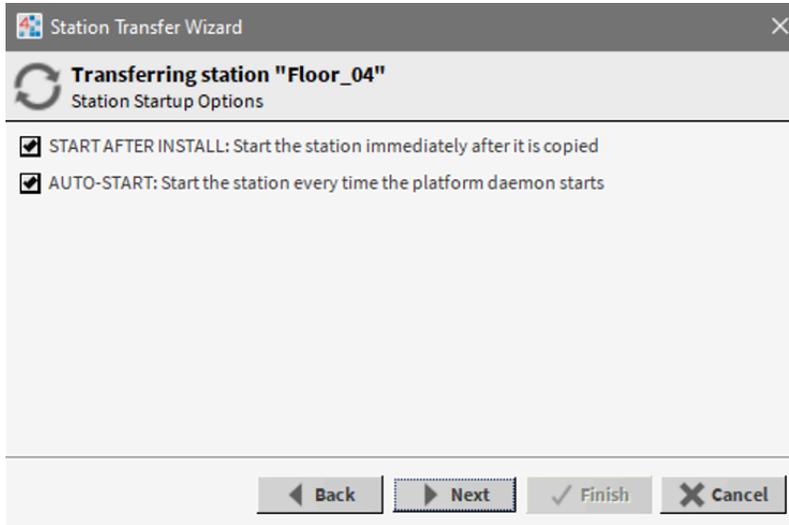
- Step 1. Before upgrading the container, open the station copier from the platform and copy the station to your local files to store it on your Workbench PC.



- Step 2. Select the Copy every file in the station directory and its subdirectories option and click Finish.



- Step 3. Start a new container image and make a platform connection. Repeat the licensing steps above if necessary.
- Step 4. Expand the platform container in the Nav tree, double-click the System Copier, select the station that you want to copy to the container and click Copy.
- Step 5. Click Next, select the Copy every file in the station directory and its subdirectories option, and click Next again.
- Step 6. Select Start After Install: Start the station immediately after it is copied and Auto-Start: Start the station every time the platform daemon starts options, click Next, and click Finish to complete.



You can check the status of the transfer in the **Transferring station** window.

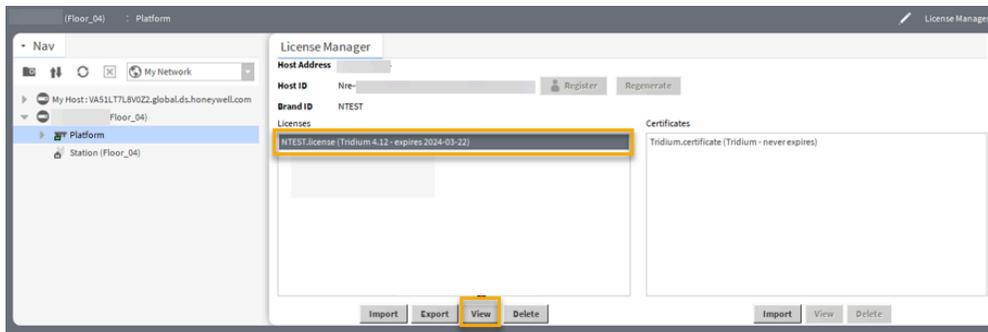
Step 7. Check the station status in the platform’s **Application Director**.

Verifying license check-in parameters

This section describes how to verify the check-in parameters of a license.

Step 1. Connect to the platform and open the **License Manager**.

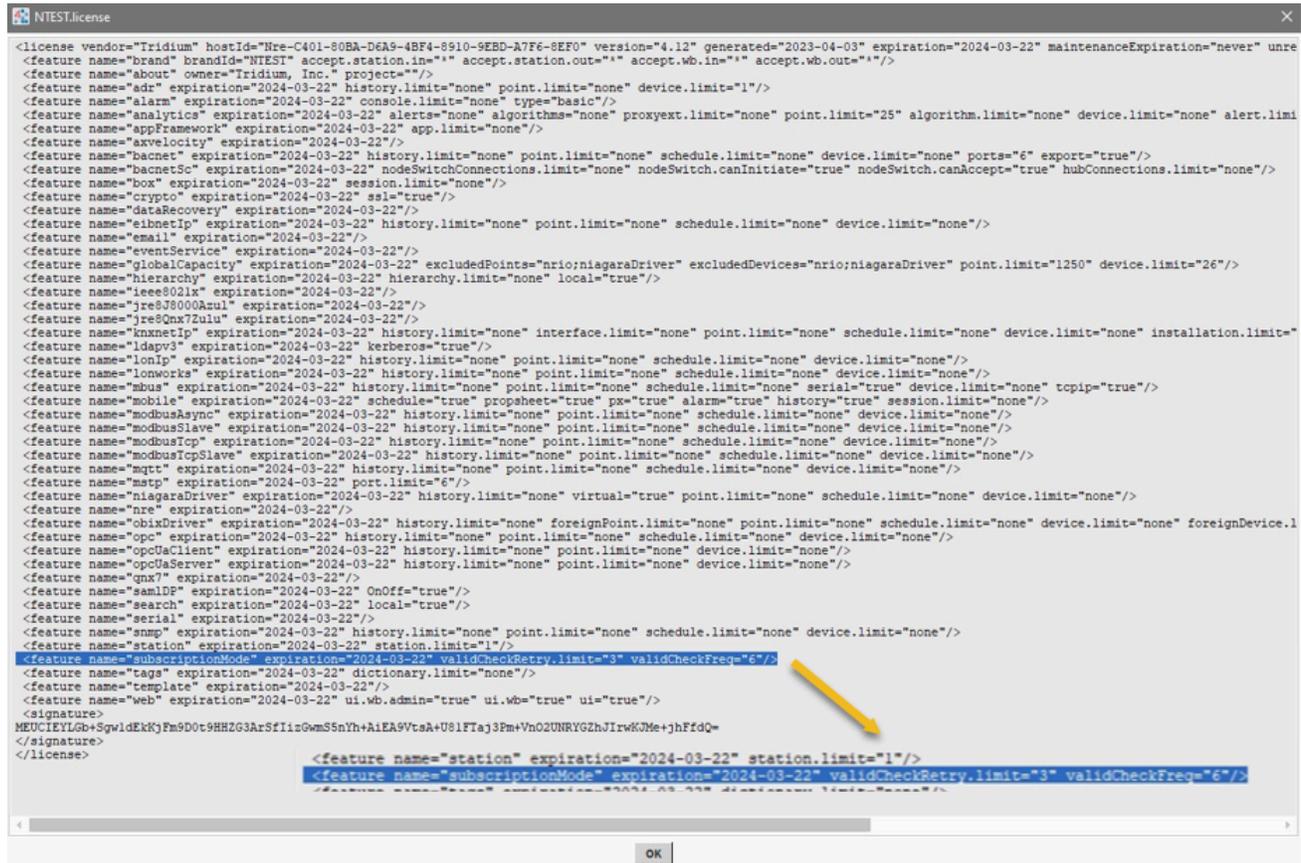
Step 2. In the **Licenses** window, select the appropriate license and click **View**.



Towards the bottom of the file, look for the `subscriptionMode` license feature. It indicates your check-in interval and number of allowed failed check-ins prior to the station shutdown.

Example:

```
<feature name="subscriptionMode" expiration="2024-03-22" validCheckRetry.limit="3"
validCheckFreq="6"/>
```



Deregistering a license from host ID

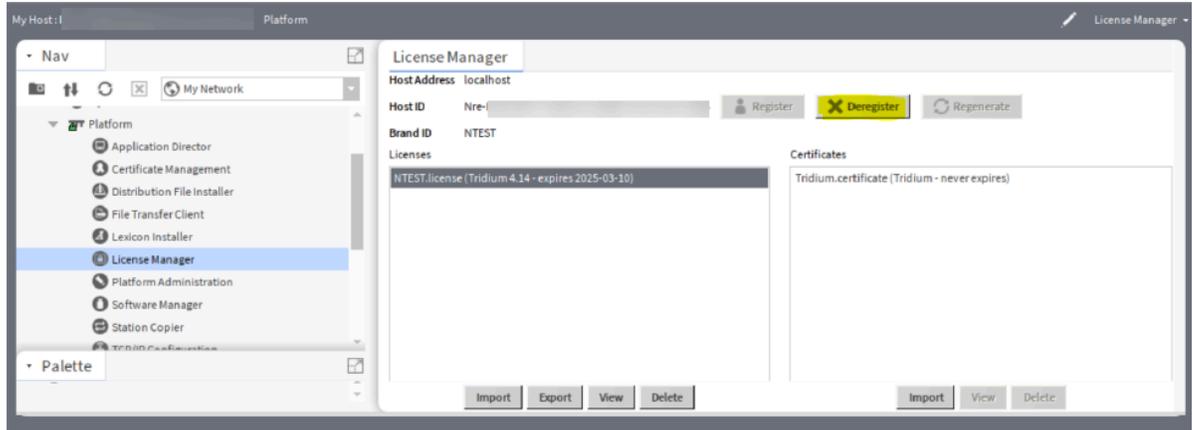
As of Niagara 4.15, you can deregister a Niagara license from a host ID by connecting Workbench to the Niagara platform or by using the NRE command. You can perform the deregistration of a license directly from your station. Use the following steps to deregister a host ID.

Prerequisites:

- You have created a Niagara Central account (<https://www.niagara-community.com>) and you have access to Niagara Licensing.
- You are a Niagara Licensing portal user with Write/WriteX permissions on Subscription License.

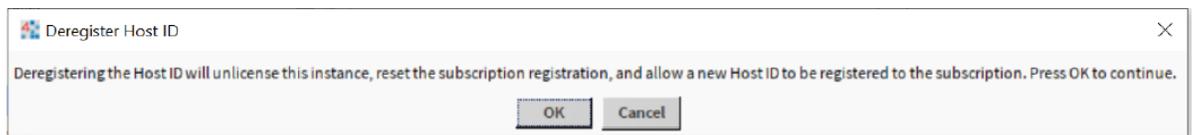
Using Workbench

- Step 1. Open Workbench and make a platform connection to the Niagara instance.
- Step 2. In the Nav tree, double-click on the License Manager.



Step 3. Verify that the Niagara instance is registered to a host ID and the **Register** button is disabled.

Step 4. To deregister the license, click **Deregister**.
The **Deregister Host ID** window opens to ask you if you want to proceed with the deregistration.



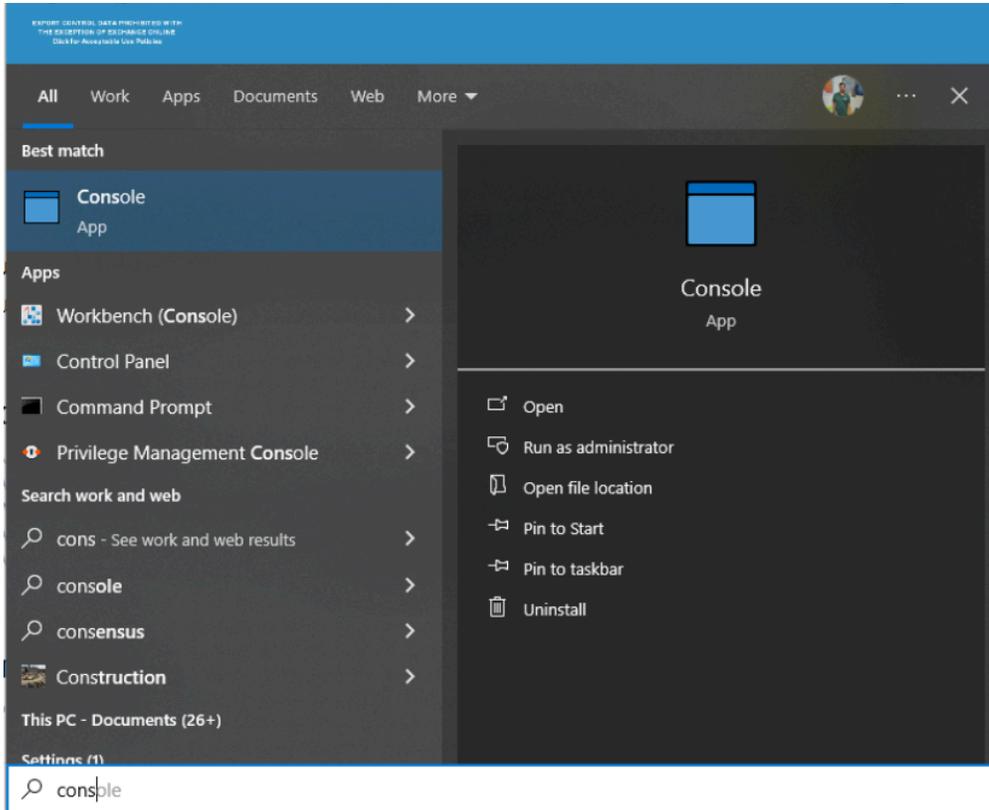
Step 5. Click **OK** to continue.

Step 6. Upon successful deregistration, a dialog window opens confirming that the host ID is deregistered, along with the unbound license key. You can use the unbound licensing key to register a new host ID.

NOTE: Retain a copy of the licensing key. To register a new device to the subscription, you can use the unbound license key. The license key has been exported to the remote host's `security/subscription/deregistration` folder.

Using NRE command

Step 7. Open the Niagara console command line for the Niagara instance to be deregistered. On a Linux Niagara instance, open the shell application.



Step 8. Check if the host ID is currently bound by running the NRE command: `nre -checksub` and registered with the entitlement system.

```
Administrator: Niagara Command Line
c:\niagara\niagara-4.15.0.115.1557>
c:\niagara\niagara-4.15.0.115.1557>nre -checksub
*****
**** DEVELOPER BUILD FOR INTERNAL TRIDIUM USE ONLY ****
*****
INFO [nre] Launching Niagara Runtime Environment
INFO [10:59:13 01-Feb-25 IST][nre] Booting
INFO [10:59:14 01-Feb-25 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string security property [jdk.tls.disabledAlgorithms]: SSLv3, RC4, DES, MD5withRSA, DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC, anon, NULL, include jdk.disabled.namedCurves
WARNING [10:59:14 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.tls.disabledAlgorithms': include jdk.disabled.namedCurves
INFO [10:59:14 01-Feb-25 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string security property [jdk.certpath.disabledAlgorithms]: MD2, MD5, SHA1 jdkCA & usage TLSserver, RSA keySize < 1024, DSA keySize < 1024, EC keySize < 224, SHA1 usage SignedJAR & denyAfter 2019-01-01, include jdk.disabled.namedCurves
WARNING [10:59:14 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgorithms': SHA1 jdkCA & usage TLSserver
WARNING [10:59:14 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgorithms': SHA1 usage SignedJAR & denyAfter 2019-01-01
WARNING [10:59:14 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgorithms': include jdk.disabled.namedCurves
INFO [10:59:16 01-Feb-25 IST][sys.registry] Up-to-date [48ms]
INFO [10:59:16 01-Feb-25 IST][sys.license] This Host Id nre-8944-EFAC-548E-4CCB-8B25-F04A-FE2C-1AE5 has been registered with the entitlement system.

c:\niagara\niagara-4.15.0.115.1557>
```

Step 9. To deregister the license from the host ID, run the command `nre -deregister`.

Upon successful deregistration, the console will display a message confirming that the device is deregistered. The original subscription key will also be displayed and can be bound to another host ID.

Replacing a host ID bound to license

As of Niagara 4.15, you can replace an already registered Niagara license to a host ID with new host ID. You can handle the replacement of the license directly from your station using the NRE command. Use the following steps to replace a host ID.

- You have created a Niagara Central account (<https://www.niagara-community.com>) and you have access to Niagara Licensing.
- You are a Niagara Licensing portal user with Write/WriteX permissions on **Subscription License**.
- The subscription key is available and the host ID is already bound.

Using Workbench

- Step 1. Open Workbench and make a platform connection to the Niagara instance.
- Step 2. In the Nav tree, double-click the **License Manager** and click **Register**. The **Device Registration** window opens.

Device Registration

Licensing
Register the device for subscription license

Subscription Key: 5C24-

Replaces existing Host ID bound to this Subscription Key

Host ID: D-B

Verification URL: <https://www.niagara-community.com/setup/connect>

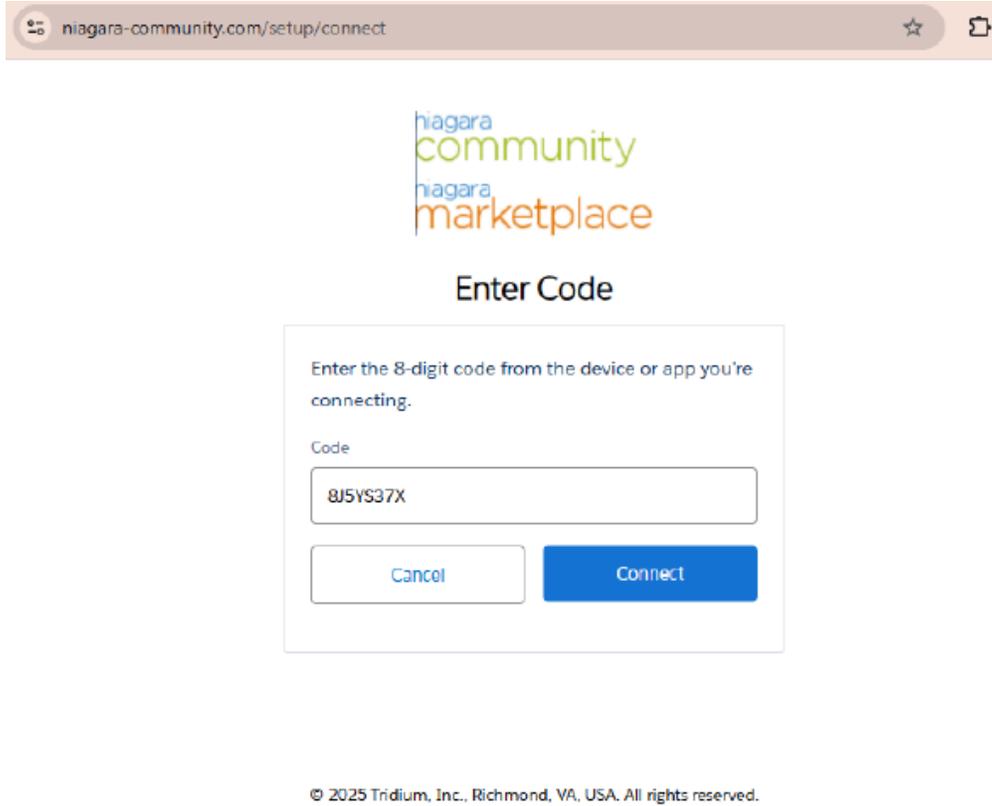
Code: 8J5P4Y4V

Follow these steps to complete subscription licensing:

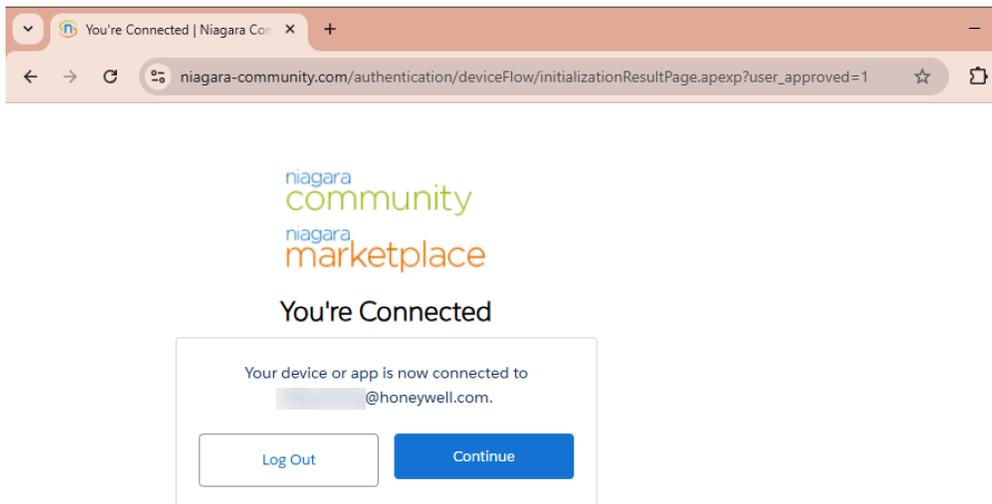
- 1) Enter your Subscription Key above.
- 2) Go to the Verification URL with a browser.
- 3) Enter the Code at the website.
- 4) Login to your account.
- 5) Approve the device you are registering.

OK Cancel

- Step 3. Enter the subscription key, select the **Replaces existing Host ID bound to this Subscription Key** check box, and enter the host ID that is already registered with the entered subscription key.
- Step 4. Click on the **Verification URL** to navigate to the web browser, paste the copied code from Workbench in the **Code** field, and click on **Connect**.

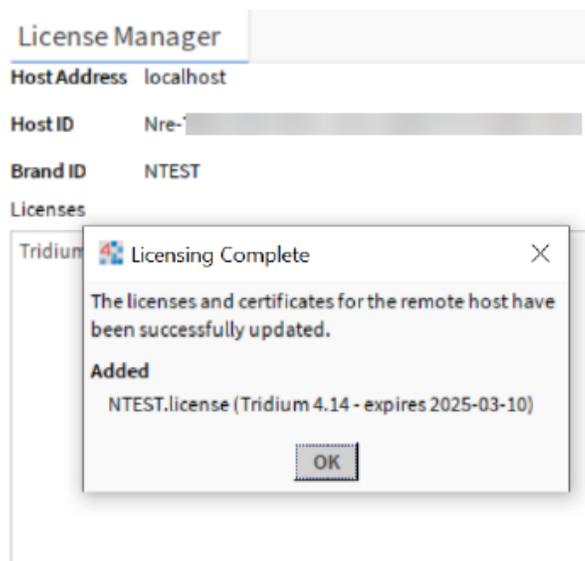


- Step 5. On the **Allow Access** screen, click **Allow** to permit the Niagara instance to connect to the Niagara license server and to continue. The **You're Connected** screen opens to inform you that your device or app is now connected.



- Step 6. Click **Continue** to replace the existing host ID bound to this subscription key.
- Step 7. In Workbench > **License Manager**, the **Licensing Complete** dialog window informs you that the

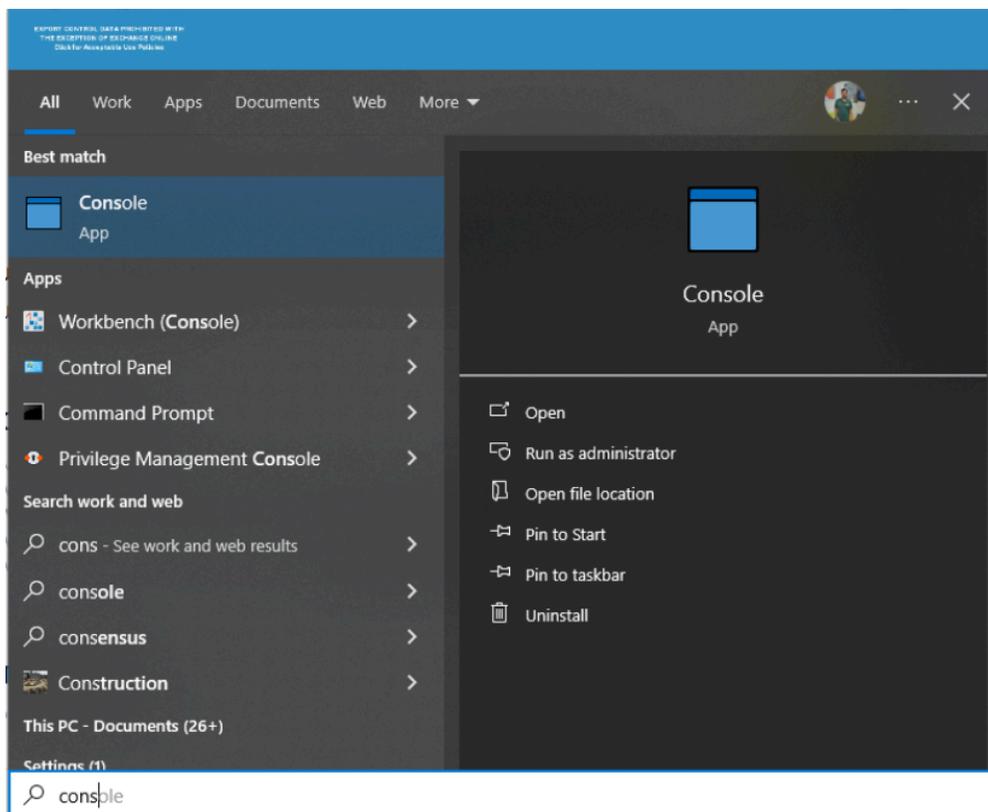
Niagara instance has been successfully registered and the host ID has been replaced.



Step 8. Click **Ok**.

Using NRE command

Step 9. Open the console command line for the Niagara instance whose host Id needs to be replaced. On a Linux Niagara instance, open the shell application.



Step10. Run the command `nre -replace <Existing Host ID> <Subscription Key>` to initiate the replacement of the existing host ID.

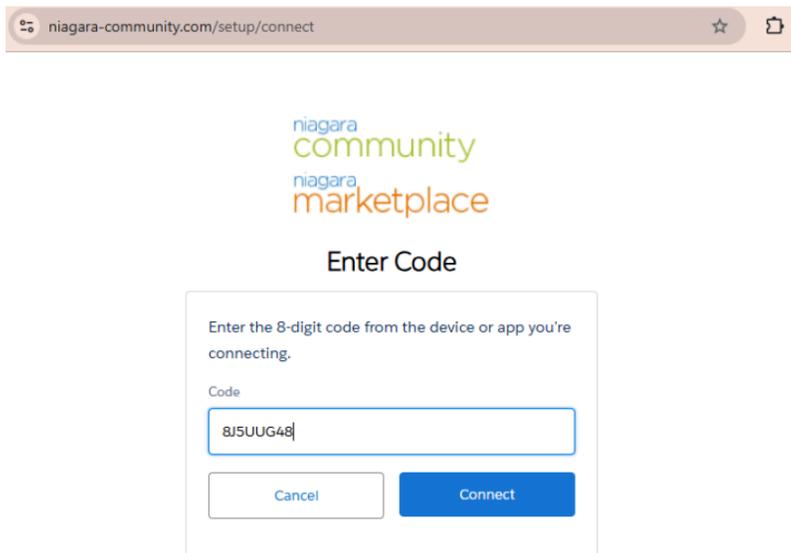
```
Administrator: Niagara Command Line
ithms': SHA1_jdkCA & usage_TLSServer
WARNING [14:13:17 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgor
ithms': SHA1_usage_SignedJAR & denyAfter_2019-01-01
WARNING [14:13:17 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgor
ithms': include_jdk.disabled.namedCurves
INFO [14:13:17 01-Feb-25 IST][sys] Please provide a valid Host ID followed by a valid Subscription Key as parameters.

c:\niagara\niagara-4.15.0.123.1560>nre -replace Nre-
***** DEVELOPER BUILD FOR INTERNAL TRIDIUM USE ONLY *****
INFO [nre] Launching Niagara Runtime Environment
INFO [14:13:51 01-Feb-25 IST][nre] Booting
INFO [14:13:51 01-Feb-25 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string security property [jdk.tls.disabledAlgorithms]: SSLv3, RC4, DE
S, MD5withRSA, DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC, anon, NULL, include_jdk.disabled.namedCurves
WARNING [14:13:51 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.tls.disabledAlgorithms
': include_jdk.disabled.namedCurves
INFO [14:13:51 01-Feb-25 IST][org.bouncycastle.jsse.provider.PropertyUtils] Found string security property [jdk.certpath.disabledAlgorithms]: MD2, MD5,
SHA1_jdkCA & usage_TLSServer, RSA keySize < 1024, DSA keySize < 1024, EC keySize < 224, SHA1_usage_SignedJAR & denyAfter_2019-01-01, include_jdk.disab
led.namedCurves
WARNING [14:13:51 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgor
ithms': SHA1_jdkCA & usage_TLSServer
WARNING [14:13:51 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgor
ithms': SHA1_usage_SignedJAR & denyAfter_2019-01-01
WARNING [14:13:51 01-Feb-25 IST][org.bouncycastle.jsse.provider.DisabledAlgorithmConstraints] Ignoring unsupported entry in 'jdk.certpath.disabledAlgor
ithms': include_jdk.disabled.namedCurves
INFO [14:13:53 01-Feb-25 IST][sys.registry] Out-of-date: Module changed "aaphp-rt"
INFO [14:13:53 01-Feb-25 IST][sys.registry] Rebuilding registry...
INFO [14:13:57 01-Feb-25 IST][sys.registry] Loaded [24ms]
INFO [14:13:57 01-Feb-25 IST][sys.registry] Rebuilt: 8655 types [3728ms]
*****
**** Please go to the Verification URL and enter the User Code to approve
**** the subscription for this Niagara Instance
**** Verification URL: https://www.niagara-community.com/setup/connect
**** User Code: 8J5UUG48
*****
INFO [14:13:58 01-Feb-25 IST][licensing.subscription] Polling for registration status (will poll every 5 seconds for 10 minutes)

c:\niagara\niagara-4.15.0.123.1560>
```

Step11. To navigate to the **Enter Code** screen in your web browser, locate and copy the **Verification URL** in the **Niagara Command Line** window and paste it into your web browser.

Step12. Similarly, from the **Niagara Command Line** window, copy the **User Code**, paste it in the **Code** entry field, and click **Connect**.



Step13. On the **Allow Access** screen, click **Allow** to permit the Niagara instance to connect to the Niagara license server.

NOTE: If you have successfully granted access to a specific application 5 times, which is the maximum allowed, the following applies: If you approve the 6. request, the oldest approval will automatically be revoked. To prevent this, you can either deny this request or manually revoke an approval in your personal settings.



Allow Access?

NiagaraEntServerDeviceAuth is asking to:

- Access the identity URL service
- Perform requests at any time

Do you want to allow access for
[redacted]@honeywell.com? (Not you?)

To revoke access at any time, go to your personal settings.
Caution: You have granted access to this application 5 times, which is the limit. Approving this request automatically revokes your oldest approval. To avoid revoking your oldest approval, deny this request or manually revoke an approval in your personal settings.

The **You're Connected** screen opens to inform you that your device or app is now connected.

- Step14. Click **Continue** to replace the existing host ID bound to this subscription key. Upon registration, the Niagara Command Line window confirms that the device is registered with the subscription licensing system and that the existing host ID was replaced successfully.

Chapter 5. Glossary

The following glossary entries relate specifically to the topics that are included as part of this document. To find more glossary terms and definitions refer to glossaries in other individual documents.

Alphabetical listing

Carrier Communication/Comfort Network (CCN)

The Carrier Communication/Comfort Network (CCN driver) integrates CCN devices and data into the Niagara Framework environment.

Java SE Runtime Environment (JRE)

Niagara Portability Software Development Kit (NPSDK)