Rocket software

# Verastream Host Integrator

## Installation Guide

7.9 SP1

# Table of contents

# 1. Introducing Verastream Host Integrator

Verastream Host Integrator provides an innovative approach that encapsulates host data as a web service and makes it available to other parts of your enterprise. It requires no changes to existing host systems and causes no interruption to end-user operating environments.

- Use the Design Tool to navigate through the application and select fields and screens to build a model. You can organize model data into tables that can be queried by SQL statements.
- When your model is complete, Host Integrator generates a WS-I compliant Web service that is deployed and executed using the Host Integrator Session Server. Along with the auto-generated Web service you can also use Web Builder to generate a Web-based application and use connectors to create new applications that access host data.
- The client/server or Web application is deployed in an enterprise production environment using Host Integrator’s multi-tier domain architecture with load balancing and failover support.
- The Administrative Console is the central hub for all Host Integrator security, server, directory service, and network domain configuration. You can also monitor live sessions and query session logs using the Administrative Console.
- The management server provides greater control over sessions and pools, logging of session information including activity, use and status as well as SNMP and JMX support for third-party consoles. It is also possible to configure a multi-server management cluster for scalable performance and redundancy.

# 2. Installation

## 2.1 Host Integrator Components

The Verastream Host Integrator includes two core products:

- The Development Kit provides developers with the tools to model and build applications that integrate host application information. It contains all the Host Integrator components.
- The Server Kit is used to deploy those applications.

  Contains all the components, except the Design Tool and Web Builder. With a Server Kit installation you must manually start the Web server service as well as the Host Emulator service.

### 2.1.1 About Host Integrator Components

Design Tool

Web Builder

Management Server

Administrative Console

Host Integrator Server

Host Emulator

#### Design Tool

You use the Design Tool to model existing host applications for encapsulation and integration into client/server and web applications. When you deploy a model to the session server a WS-I compliant web service is automatically generated and saved to the session server along with the project files. The Design Tool runs on Windows platforms and requires the ability to connect to the host via a Telnet, Telnet Extended connection, or NS/VT for HP connections. You cannot run the Design Tool when logged on to Windows as a "guest".

#### Web Builder

With Web Builder you can quickly and easily generate and deploy a web application or component interface. Web Builder generates an HTML 5 web application and .NET and Java components based on the procedures of a host application model. Web Builder can auto-generate web pages that are not dependent on the appearance or flow of the host application.

### Management Server

The Management Server handles authentication and authorization, as well as session pool scheduling and automatic failover support. Each Host Integrator server must be able to connect to a Management Server.

### Administrative Console

**Note**

The Administrative Console is only installed on a Windows platform, but it is used to manage both Linux and Windows servers.

The Administrative Console takes advantage of the Eclipse workbench features to help administrators manage Host Integrator servers, security, directory services, and domains, as well as monitor sessions and pools in real time. The Administrative Console also provides access to complete logging capabilities and allows a user to manage both Linux and Windows servers from a Windows-based installation.

### Host Integrator Server

The Host Integrator Server is a session server that supports multi-tier client/server and web application architectures and provides access to host information systems to thousands of web application users.

Using Host Integrator Server, a single web or client/server application can concurrently access data on a variety of host systems, including IBM mainframes and compatibles using the 3270 terminal protocol; IBM AS/400 systems using the 5250 terminal protocol; VAX/OpenVMS, HP-UX, and other ASCII hosts using the VT-420 terminal protocol (including VT-52 and VT-100); HP 3000 hosts using the 700/92 terminal protocol via Telnet or NS/VT. Telnet Secure Socket Layer (SSL) and Transport Layer Security (TLS) security protocols are available for 3270 and 5250 session types, and Telnet Extended SSL/TLS support is available for 3270 session types. SSH connections are available when you need secure, encrypted communications between a trusted VT host and your network.

The Host Integrator Server supports SHA-256 security certificates with 2048 bits keys. All certificates coming from the server, for example the HTTPS Verastream web services use this algorithm. SHA-256 security certificates are designed by the National Security Agency and meet the United States information processing standards.

**⚠ Caution**

Due to a vulnerability in the SSL 3.0 protocol, beginning in VHI version 7.7, SSL 3.0 is disabled by default. See Knowledge Base article 7021975 for information on the vulnerability.

### Host Emulator

Use Host Emulator to run 3270, 5250, and VT models created with the Design Tool without having a live connection to a host. You can choose to install the Host Emulator as part of a custom installation when you select the Pacific Department Store component. The Host Emulator is managed in the Administrative Console.

## 2.1.2 Host Integrator Connectors

You use connectors to manage host connections and sessions, and read and write data to fields.

Verastream Host Integrator supports these connectors:

- Java connector – this connector is used to create Java-based (compatible CORBA framework) applications.
- JDBC connector – this connector is used to provide an industry standard Structured Query Language (SQL) interface to Host Integrator servers. Although the server is not a relational database system, the Design Tool's Table and Procedure feature provides access to the host application in a way that simulates traditional relational database tables.
- .NET connector – this connector uses Microsoft’s .NET Framework. The Verastream .NET Class Library is a library of classes and interfaces that comply with the Microsoft .NET Framework SDK.
- COM connector – this connector is used to create Visual Basic and ASP applications. When you choose this option from the main installation program Component Selection panel, both a 32-bit and 64-bit connector is installed.See Installation Use Cases for instructions.
- C connector – this connector is used to create C or non-COM C++ applications for Windows- or Linux-based development platforms that integrate host data into Web applications or client/ server applications.
- Web services provide reusable APIs for creating portals, web applications, and other business solutions. They are technology independent, and can run on any platform.
- HLLAPI Adapters – These adapters provide a migration path from legacy APIs to a centralized server-based platform. These adapters look and feel like the legacy API, but provides an implementation that interfaces with the Host Integrator session server.

For information about working with Host Integrator connectors and adapters, see the Development Kit online help.

# 2.2 System Requirements

The requirements for Host Integrator vary depending on your choice of component and platform.

- Hardware and software requirements apply to all installation options.
- Connector requirements apply to interfaces used by developers to create client/server, or Web applications.
- See Ports used by Host Integrator for information on ports used by Host Integrator. This information is helpful when configuring firewall access.

## 2.2.1 About IPv4 and IPv6

Verastream Host Integrator uses IPv4 by default, but if you have a fully working IPv6 network, you can configure VHI to use IPv6.

### To configure VHI to use IPv6

In the following files, change the value (True) to false. For example, `-Djava.net.preferIPv4Stack=false`.

| Component | File name and location |
|---|---|
| Management Server | `/ManagementServer/conf/container.conf` |
| Management Server's JConsole | `/ManagementServer/bin/jconsole` (bat and/or sh) |
| Administrative Console | `/AdministrativeConsole/Administrative Console.ini` |
| Host Emulator | `/HostIntegrator/hostemulator/conf/container.conf` |
| Host Emulator JConsole | `/HostIntegrator/hostemulator/bin/jconsole` (bat and /or sh) |
| Web Server | `/HostIntegrator/servletengine/conf/container.conf` |
| Design Tool | `/HostIntegrator/etc/destool.conf` |
| Log Manager | `/HostIntegrator/etc/logmgr.conf` |
| Session Server | `/HostIntegrator/etc/sesssrvr.conf` |

| Component | File name and location |
|---|---|
| Web Builder | `/HostIntegrator/etc/webbuilder.conf` |

And also these files in the `/HostIntegrator/bin` directory:

`activatemodel` (bat and/or sh)

`deactivatemodel` (bat and/or sh)

`resetsessionserver` (bat and/or sh)

## 2.2.2 Hardware and Software Requirements

The following requirements are for all Verastream Host Integrator components. Some development components can only be installed in a Windows environment.

### Design Tool, Web Builder, Session Server and Administrative Console Requirements (Dev Kit)

#### Supported platforms (64-bit)

Windows Server 2022

Windows Server 2019

Windows Server 2016

Windows Server 2012

Windows 11

Windows 10

#### Disk Space Requirements

All the Host Integrator components require less than 1GB of disk space. However, it is a good idea to allow additional space for data (models, projects, and logging) and to allow for Windows platform install programs or .NET Framework

#### Additional requirements

- RAM - 512 MB for first 100 users and an additional 64 MB for each additional 100 users.
- To use the Design Tool, a minimum video resolution of 1024 x 768 is required.

**Note**

Applications are generated for Microsoft Visual Studio 2008, and for older .NET versions; if they are opened with newer versions, Visual Studio will convert them. Projects can be opened and rebuilt using Visual Studio 2005 without errors or warnings.

## Web Builder

Additional software requirements for Web Builder generated services, components, and Web applications derived from host application model procedures include:

- ASP .NET must be installed to run legacy .NET Web applications with Web Builder. In older versions of Windows, ASP .NET is installed by default when you install Internet Information Services (IIS). This is not the case when running Windows 2008. See .NET Web Application Properties in the Web Builder online help for instructions on how to install ASP .NET on those platforms.
- Client interfaces – Java Bean (JDK 1.8.0 or higher) and C# .NET 4.5 Class Library

Verastream Host Integrator follows HTML and JavaScript specifications and takes advantage of the powerful features they offer. We support all modern browsers that support these specifications such as Internet Explorer, Firefox, Chrome, Safari and Opera.

Client Web browsers that meet these minimal requirements:

Internet Explorer version 11.0 +

Firefox 27+

Chrome 33+

Microsoft Edge

**Note**

To generate .NET Web applications on computers running Microsoft IIS 7.0, select the IIS 6 Management Compatibility option when installing IIS. For Vista, install IIS 7, under Programs and Features. For Windows Server 2008, use Server Manager to Add the Web Server Role (IIS).

## Session Server and Management Server requirements (Server Kit)

### Supported Windows platforms (64-bit)

Windows Server 2022

Windows Server 2019

Windows Server 2016

Windows Server 2012

### Supported Linux platforms (64-bit)

- Red Hat Enterprise Linux (RHEL)

  RHEL 9

  RHEL 9 for IBM System z

  RHEL 8

  RHEL 8 for IBM System z

  RHEL 7.x (x86_64)

  RHEL 7/x for IBM System z

- SUSE Enterprise Linux

  SUSE Enterprise Server 15 for IBM System z

  SUSE Enterprise Server 15 (x86 64)

  SUSE Enterprise Server 12 (x86 64)

  SUSE Enterprise Server 12 for IBM System z

### Other requirements

RAM - 512 MB for first 100 users and an additional 64 MB for each additional 100 users.

### Additional information

- Because the current version of Verastream Host Integrator includes Java 8 with the installation, the product works as expected on all platforms where Java is installed. However, if you are using zLinux and an IBM JDK, you must supply additional information to ensure the product works with Java 8.

  To make VHI compatible with Java 8, newly introduced serialization filters must be loosely configured to adhere to security guidelines and must be configured for the IBM JDK installed on your machine. See the Configure Java Serialization Filters section of Technical Note 7021305 for instructions.

- On some Linux platforms you may encounter delays caused by insufficient entropy. An indication of this issue is a stalled process that does not consume CPU time. You can inspect the amount on entropy on a Linux system with the command `cat /proc/sys/kernel/random/entropy_avail`.

To test and temporarily work around entropy issues, you can change the `java.security` file, modifying the property `securerandom.strongAlgorithms` to `NativePRNGNonBlocking`.

- If, when installing on Red Hat Linux 7 server, you encounter errors when asked to specify the administrative password, which stops the installation, you should install the following files, (available for download from the Web): `glibc-2.12-1.80.el6.i686.rpm` and `nss-softokn-freebl-3.12.9-11.el6.i686.rpm`.

  These files must be installed prior to successfully installing VHI. You can install them as root using this command: `yum install glibc-2.12-1.80.el6.i686.rpm nss-softokn-freebl-3.12.9-11.el6.i686.rpm`.

## Connector Requirements

| Connector | Requirements |
|---|---|
| COM | To create a client application using C++, you need a Windows-based operating system and Microsoft Visual Studio version 6.0 or higher. There is a separately installed 32-bit COM connector available. See Installing a 32-bit COM Connector. |
| C | The C connector is included in the installation of the AppConn COM connector. You can use the C connector to create C or non-COM C++ applications for Windows or Linux-based development platforms that integrate host data into Web, client, or server applications. Visual Studio 2010 C libraries, which are installed with the product, have replaced the previously used Visual Studio 2005 C libraries. The C API requires inclusion of appconnapi.h and appconndef.h in the source file. See the C connector topic under Verastream Host Integrator > Connectors and API’s in the Help for more information. |
| Java | The Host Integrator AppConn Java Connector’s .JAR files require JDK 1.8. When you install the Development Kit, a copy of the Azul Zulu OpenJDK version 1.8 is installed in the `Verastream\java` folder. For more information on Azul Zulu OpenJDK version 1.8, see the Azul Web site. |

| Connector | Requirements |
|---|---|
| .NET | To use the .NET connector to develop an application that accesses the host application model file on your Host Integrator Server, you need the .NET development tool—Microsoft Visual Studio. You also need .NET Framework SDK, version 3.5 to link to information on .NET data types from the Host Integrator MSDN-style compiled help system. |

# 2.3 Port and Network Configuration

## 2.3.1 Port and Network Configuration

These are the ports used by Verastream Host Integrator (VHI) version 7.0 and higher. This information may be helpful if you are configuring your firewall to allow VHI access.

### Management Server ports

| Listening Port | Purpose/From |
|---|---|
| 9641 (WCP) | Client connector, connecting via load distribution domain / Session server (for authentication and authorization, on behalf of clients) / Deployment tools (activatemodel, deactivatemodel, Design Tool) |
| 33000 (SSL JMX) - 33001 (SSL JMX export callback) | Administrative Console (JConsole API for configuration and monitoring) |
| 8095 (HTTP) | Web Services Explorer (Jetty/JSP) Web application for testing Web services |
| 7846 | Peer management servers in cluster (for data replication) |
| 1636 (LDAPS) | Administrative Console (for authorization profile configuration; aggregates user data from all configured directory servers) |
| 48620 and higher (WCP) | From client connector (connecting “via domain") to domain server function in Management Server (one port used per load distribution domain) |
| 32000 | Wrapper service for JVM |

| Listening Port | Purpose/From |
|---|---|
| Varies (UDP) | Session server (notification of activity; to display in Administrative Console) |

## Session Server ports

| Listening Port | Purpose/From |
|---|---|
| 9623 (WCP) | Client connector, deployment tools, and Management Server |
| 9680 (HTTP) | Web services clients, including Web Services Explorer. See Changing Web Service Ports |
| 9681 (HTTPS) | Web services clients, with SSL encryption |
| 35000 (SSL JMX) - 35001 (SSL JMX export callback) | Management server (Web services management) |
| 9653 | Script engine for model Java event handlers (on local system) |
| 9662 | Script engine for model .NET event handlers (on local system) |
| 5005, or next available | External debugger for Java event handler scripting (if enabled) |

## Log Manager and Host Emulator ports

### Log Manager

| Listening Port | Purpose/From |
|---|---|
| 9640 (WCP) | Session Server (adding messages)/Management Server (querying messages)/ LogExport utility (querying messages) |

### Host Emulator

| Listening Port | Purpose/From |
|---|---|
| 9670-9671, 1096-1099 | Default "master" models, for demo connections from Session Server |

| Listening Port | Purpose/From |
|---|---|
| 36000 (secure JMX/RMI) | Management Server (JConsole API for configuration) |
| 36001 (JMX) | JConsole API |

## Design Tool ports (Windows Development Kit only)

| Listening Port | Purpose/From |
|---|---|
| 5006, or next available | External debugger for Java event handler scripting |
| 9654 through 9661 | Script manager for Java event handlers (one port per Design Tool instance) |
| 9663 through 9669 | Script manager for .NET event handlers (one port per Design Tool instance) |

## Web Server and other ports

| Component | Listening Port | Purpose/From |
|---|---|---|
| Web server | 8081 (HTTP) | HTML5/Java web application projects generated in Web Builder |
| Web server | 8443 (HTTPS) | HTML5 web application projects generated in Web Builder, with encrypted communications |
| Your hosts | 23 (Telnet, default) / 992 (Telnet SSL, default) /1570 (HP NS/VT, default) | Session server |
| Your network monitoring system (optional) | 161 UDP (SNMP) | Log Manager (notifications) |
| Your e-mail server (optional) | 25 (SMTP, default) | Log Manager (notifications) |
| | | |

| Component | Listening Port | Purpose/From |
|---|---|---|
| Your directory server (optional) | 636 (LDAPS, default) /389 (LDAP, default) | Management Server (authentication requests for Administrative Console, deployment tools, and client connector) |

**Tip**

Some components may use additional ephemeral ports. When configuring Windows Firewall exceptions, in addition to specifying port numbers, it is recommended you add program `C:\Program Files\RocketSoftware\Verastream\java\bin\java.exe`.

## Configuring Host Integrator Networking

If you need to reassign the port numbers used by Host Integrator components, do the following:

Stop all Host Integrator services.

Create a file called `vhibaseport.properties` and place it in the `vhi/etc` directory.

Add `vhibaseport=n` to the file, where "n" is a valid port number.

Restart all Host Integrator services.

After you reassign the VHI base port, the Host Integrator components are assigned these ports:

| Component | Port Assignment |
|---|---|
| vhibaseport | n |
| session server | n > n+16 |
| logging utility | n+17 |
| management server | n+18 |
| script manager | n+30 |

### Changing Web service ports

To change the ports used for Web services (SOAP, and REST):

1. Edit the configuration file, `service-cfg.xml` file in `%VHI_ROOT%\sesssrvr\services\ws\META-INF`.

> **Note**
>
> This file is created after web services are configured in Administrative Console.

2. Add or modify the appropriate value:

   - SOAP HTTP - hostPort port 9680 (default)
   - SOAP HTTPS - secureHostPort port 9681 (default)

   For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>com.attachmate.ws.impl.ServiceMainConfigImpl</comment>
    <entry key="hostPort">9666</entry>
    <entry key="secureHostPort">9644</entry>
</properties>
```

3. Restart the session server.

## 2.3.2 Customize LDAP TLS certificates and ciphers

The VHI LDAP server is used to store credentials and settings for the Management Server. To facilitate automatic setup, the LDAP server is provided with a self-signed TLS certificate and uses preset ciphers.

A production system will most likely require a custom certificate and may also require custom ciphers. Follow the instructions below to install a custom certificate and configure custom ciphers for LDAP over TLS.

### Install a custom certificate

1. Construct a keystore in PKCS12 format that contains the new certificate and private key. The key type must be RSA.
2. Edit `path-to\RocketSoftware\Verastream\ManagementServer\conf\container.properties` set `ldap.port.enabled=true`, and restart the Management Server service.
3. From a command prompt, run `java -cp jar-file com.attachmate.vhi.vms.UpdateLdapCert keystorefile keystorepassword adminpassword`. This command will update the LDAP server certificate.

   Replace `keystorefile`, `keystorepassword` and `adminpassword` with their respective actual values.

4. Revert the changes to `path-to\RocketSoftware\Verastream\ManagementServer\conf\container.properties` or set `ldap.port.enabled=false`, and restart the Management Server service.

> **Note**
>
> Replace `-cp jar-file` in the Java command with `-cp "path-to\RocketSoftware\Verastream\ManagementServer\services\vhi\lib\*"` where `path-to` is the actual path to where VHI is installed. In a default installation, use `-cp "C:\Program Files\RocketSoftware\Verastream\ManagementServer\services\vhi\lib\*"`. The quotation marks are needed. On a Linux system, use `-cp /opt/rocketsoftware/verastream/managementserver/services/vhi/lib/*`

> **Note**
>
> On Linux, the default VHI install path is `/opt/rocketsoftware/verastream/...`, all lowercase.

## Configure custom ciphers

1. Edit the file: `path-to\RocketSoftware\Verastream\ManagementServer\conf\container.conf`

   set `wrapper.java.additional.x=-Djdk.tls.server.cipherSuites=comma-delimited-cipher-names` where `x` is the next additional setting in the file sequence, and `comma-delimited-cipher-names` is the list of ciphers to be used.

2. Restart the Management Server service.

> **Note**
>
> Changing the cipher list will affect all endpoints running on this JVM: the Management Server endpoints, including LDAP.

# 2.4 Planning Your Installation

## 2.4.1 Use Cases

Host Integrator components are typically installed on multiple machines. This makes it necessary for you to carefully plan your installation before you install any components or modify an existing installation.

In a typical environment, you have separate installations for development, testing, and production. Within an installation environment, you can add servers for failover and load balancing capabilities.

The following use cases represent different installation scenarios:

- Install all the components on a single machine. If you are installing a Developer Kit, use the Typical option during setup. This option installs all the necessary components, including a local Host Integrator Server system.
- Install a subset of the components on a single machine. (ex: A developer who only needs the connectors to work with existing models that are hosted on existing servers)
- The user’s intention is to have a single session server with failover management (1 session server with any number of management servers)
- The user’s intention is to have multiple session servers grouped in a single domain (n session servers with 1 management server)
- A combination of failover management and session servers grouped in a domain, a combination of the last two cases.

**Note**

If you are installing session servers and management servers on multiple computers, it is important that you make sure that each server can communicate (either using IP addresses or machine names) with every other server. This is more important if you are not using Domain Name System (DNS). The machine names and IP addresses of both the local server and remote servers must be entered in your server machines' hosts files.

### Installing a 32-bit COM Connector

Verastream Host Integrator is a 64-bit product, and from the main installation program both a 64-bit and 32-bit COM connector is installed. However if it is necessary to install on a 32-bit system, Host Integrator provides a separate installer that installs the COM Connector, the .NET Connector, and the HLLAPI Adapters.

1. If you installed from the download site, follow the procedures described in the download instructions to start the Host Integrator setup program.
2. After you run the .exe file, unpacked files are stored in a temporary directory prior to you running the setup.exe. From this temporary directory, open the x86 directory.

## 2.4.2 Using Docker

The Docker open platform has excellent documentation which you should read and understand.

### Why Docker?

Docker is a container-based platform that enables you to develop, deploy, and run applications within a container. Your application, plus any dependencies your application requires, such as binaries and libraries, and configuration information are held within the container. You can deploy multiple containers, all of which run in Docker and on top of the operating system.

Using Docker you can scale your applications vertically, meaning multiple instances of the session server can exist on one server and each instance will perform exactly as it did when you created and tested it.

### What are the benefits?

Containerization delivers multiple benefits:

- Performance

  Virtual machines are an alternative to containers, however containers do not contain an operating system (unlike VMs). This means containers are faster to create, quicker to start, and have a much smaller footprint.

- Agility

  Because containers are more portable and have better performance, you can take advantage of more agile and responsive development practices.

- Isolation

  Docker containers are independent of one another. This is important because a Docker container containing one application, including the required versions of any supporting software, will not interfere with another container of the same application which requires different supporting software. You can have total confidence that at each stage of development and deployment the image you create will perform exactly as expected.

## Terminology

There are basic terms you need to be familiar with when working with Docker. For more information see the Docker Documentation site.

### Container

A run-time instance of an image. A container is usually completely isolated from the host environment, only able to access host files and ports if it has been configured to do so. To run an image in a container you use the docker run command.

### Docker Hub

A cloud-based community resource for working with Docker. Docker Hub is typically used for hosting images, but can be used for user authentication and automating the building of images. Anyone can publish images to Docker Hub.

### Docker Compose

Compose is a tool that uses YAML files to configure your application services and then define and run multi-container Docker applications. To learn more about Compose visit the Docker Compose documentation.

### Dockerfile

A text document containing the commands to build a Docker image. You can specify complex commands (such as specifying an existing image to use as a base) or simple ones (such as copying files from one directory to another). To build an image from a Dockerfile you use the docker build command.

### Image

A standalone, executable package that runs in a container. A Docker image is a binary that includes everything needed to run a single Docker container, including its metadata. You can build your own images (using a Dockerfile) or use images that have been built by others and then made available in a registry (such as Docker Hub). To build an image from a Dockerfile you use the docker build command. To run an image in a container you use the docker run command.

## Getting started with Docker and Verastream Host Integrator

When you install Host Integrator, if you choose to use Docker, the install package contains an initial Dockerfile and accompanying application jar file to get you started using the session server in a container. These files are available prior to your installation.

There are examples located in the `examples/docker` folder.

> **Note**
>
> Make sure you are running the latest version of Docker and Docker Compose.

There are six steps involved in creating the base image:

1. Install Docker. Follow the instructions on the Docker web site.
2. From the download site, download and extract the file, `tar xvf vhisrv-x.x.xx-prod-linux-64`. A subdirectory, `linux64` which contains a Dockerfile example is created.
3. Open the directory `linux64/examples/docker`. This directory includes the following files: `Dockerfile`, `extract_install_files.sh` and `stage_container.sh`.
4. Execute `./stage_container.sh` and accept the license, or execute `./stage_container.sh --licenseagreed` to accept the license from the command line.
5. Build the Docker image.
6. Run the Docker image.

### Build the Docker image

> **Note**
>
> One session server container per Docker host is supported.

Assuming you have followed steps one and two; installed Docker and extracted and located the Dockerfile, the next step is to build the base Docker image of the session server.

1. Run this command from the folder containing the Dockerfile:

   ```
   docker build -t vhi/sessionserver:<version> .
   ```

   Replace `<version>` with the version of the session server. If a version is not available, the default tag `(-t)` is `latest`.
2. Verify that the image was successfully created. Run:

   ```
   docker images
   ```

   The output should contain information about the image you just built.

### Run the image

Before you can run the session server image in a Docker container, you must complete the following steps:

- **Expose the needed ports**

  To specify the ports to use, run: `-p 9623:9623 -p 9680:9680 -p 9681:9681 -p 35000:35000 -p 35001:35001`

- **Map your configuration directory to the one in the container**

  A volume mount mounts a file or directory on the host machine into a container. The file or directory is referenced by its full or relative path on the host machine.

  This volume mounts the volumes named *etc* and *deploy* on the host to the Docker container. If a volume named *etc* or *deploy* does not exist it will be created. The first time the container is run the configuration files and deployed models will be copied to the volumes for retaining the session server settings.

- **Add Host Machine Name Resolution**

  The VHI Session Server needs to know the name of its host, so we add an environment variable, `--env VHI_SSHOSTNAME=<host-name>`. We also `--add-host` so the Docker network can resolve the host by name. Specify the fully qualified `<host-name>`, e.g., "name.example.com". You can use `nslookup <host-name>` to find the `<host-address>`.

```
   docker run -d \
  --add-host <host-name>:<host-address> --env VHI_SSHOSTNAME=<host-name> \
  -p 9623:9623 -p 9680:9680 -p 9681:9681 -p 9640:9640 -p 35000:35000 -p
35001:35001 \
  --mount source=etc,target=/opt/rocketsoftware/verastream/hostintegrator/
etc \
  --mount source=deploy,target=/opt/rocketsoftware/verastream/
hostintegrator/deploy \
   sessionserver:<version>
```

In Windows PowerShell:

```
   docker run -d `
   --add-host <host-name>:<host-address> --env VHI_SSHOSTNAME=<host-name>
`
   -p 9623:9623 -p 9680:9680 -p 9681:9681 -p 9640:9640 -p 35000:35000 -p
35001:35001 `
   --mount source=etc,target=/opt/rocketsoftware/verastream/
hostintegrator/etc `
   --mount source=deploy,target=/opt/rocketsoftware/verastream/
hostintegrator/deploy `
  sessionserver:<version>
```

**Tip**

This `docker run` command might work better without the line breaks on some terminals.

### Docker Desktop for Windows

To run VHI in Docker Desktop for Windows, the instructions given above are valid for both Linux and Windows, however be aware of these additional points:

- We have tested the extraction, staging, and Docker build and run instructions on Windows 10 with good results. Recent updates of Windows 10 have the `tar` command, and will also run shell scripts. The `./stage_container.sh` script does not seem to run interactively on Windows, so it behaves as if it were run as `./stage_container.sh --licenseagreed`. All commands can be typed verbatim using PowerShell; but if using the Windows Command shell, omit the `./` when running `stage_container.sh`.
- Using the Docker-recommended version two of Windows Subsystem for Linux (WSL2), users may encounter unusably slow network response times for VHI client requests, due to low entropy for secure network connections. This includes attempts to configure the Session Server from the Administrative Console. Low entropy is a problem best addressed on the Docker host machine. Our research has not turned up any entropy configuration options for WSL2. WSL1 seems to work better, so reverting back to WSL1 may be an option. In a Docker farm in the cloud, host machine access might not be possible, and the best solution may be something like harbur/haveged on Docker Hub. We have tested this, and it is convenient to use, and it works very well.
- Docker Desktop for Windows modifies the Windows hosts file ( `%WINDIR%/System32/drivers/etc/hosts` ) with the following:

```
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
```

This may cause problems for some applications. If you're not using kube context, try commenting it out by adding a hash tag to the beginning of the line: `127.0.0.1 kubernetes.docker.internal`.

## 2.4.3 Using Kubernetes

Verastream Host Integrator (VHI) now has the ability to run on Kubernetes, an open-source platform for building and managing highly scalable workloads. Setup the environment by using the built-in Linux installer, or, the Kubernetes self setup.

**Early Adopter Program**

This topic, part of our Early Adopter Program, describes the setup of VHI in a Kubernetes environment. We want to hear from early adopters about their experiences, issues and future needs.

Consult the Kubernetes documentation site for information on getting started.

**Info**

The setup procedures below assume that the user is experienced working in a Linux operating environment.

### Minimum CPU and Memory Requirements

The following minimum resources are required for each node. These requirements assume that no other production software is installed on the node. If additional software is running on the node, more resources need to be added to accommodate the other software accordingly.

- **8 CPU Cores**
- **16 GB** RAM - with swap space disabled
- **100 GB** disk space (SSD) with **80 GB** delegated to `/var/opt` and **20 GB** for `/opt`

## Extract files

VHI Kubernetes setup is provided in the file `vhik8s-99.9.8298-prod-eap.zip`. This file contains all of the images and dependencies required for the VHI Kubernetes example setup.

The zip file contains the following files:

- `install-vhi-99.9.8298.sh` -- Built-in Linux Installer
- `vhi-99.9.8298.tgz` -- Helm chart for installing VHI
- `vhi-airgap-images-99.9.8298.tgz` -- VHI runtime container images
- `vhi_thirdpartynotices.txt` -- Third party license information
- `VHI-Early-Adopter-Program.pdf` -- EAP features and limitations
- `Rocket.package-sign.pub` -- PGP key to verify that the file you are downloading has not been manipulated by a third party

Extract the zip file to a convenient location. You do not need to extract the `tgz` files.

## Using the Built-in Linux Installer

After the system requirements have been met, perform the following steps:

1. Extract the Linux installer script (`install-vhi-99.9.8298.sh`) from the zip file.
2. Enable execute permissions for the installer:

```
chmod 744 install-vhi-99.9.8298.sh
```

3. Ensure that an operating system firewall is not blocking any required ports and that masquerading is enabled.
4. With elevated privileges (for example, `sudo`), run the Linux install script (.sh) to install the product.
5. Verastream Host Integrator (VHI) uses a PGP key to verify the file you are downloading has not been manipulated by a third party. If the displayed signing information represents a known and trusted entity, such as Rocket Software, then enter `y` to install the public key and continue.

   Alternatively, refer to this Knowledge Base article to download the key separately and verify the file.
6. When the install completes, a verification tool is automatically executed.
7. If verification succeeds, then the services automatically start and you can move on to the next steps. If verification fails, see Troubleshooting.

## Troubleshooting

Once the issues are addressed, run `sudo cspctl start` to start the system. Then run `sudo cspctl enable` to start the system automatically after server restarts.

## Using the Linux Installer Tools

To use commands please make sure user is in the sudoers. Most of the tools require root.

```
sudo -s
```

To check the status of the installation:

```
/opt/opentext/csp/bin/status-vhi.sh
```

Add commands to path:

```
. /opt/opentext/csp/bin/env.sh
```

Check the status of VHI pods:

```
kubectl get pods -n vhi
```

Manage your clusters with k9s

```
k9s -A
```

You must state the VHI namespace for any Kubernetes command `-n vhi`. To use VHI after the installation jump to Using VHI in Kubernetes.

## Kubernetes Self Setup

This guide walks you through how to configure a VHI Kubernetes sample setup consisting of one VHI management server and two VHI session servers, all running in a light-weight Kubernetes environment K3s.

K3s provides most of the elements necessary to work with VHI in Kubernetes. VHI has been developed, run, and tested in K3s.

This VHI Kubernetes sample setup uses the Kubernetes `default` namespace. Production systems usually use a dedicated namespace to isolate concerns, and make the system more reliable and secure.

There are a few steps involved in setting up the VHI Kubernetes sample setup. It is best to follow the steps in the order that they are presented. Once you have K3s installed on Linux, and the VHI container images imported into your container registry, you can use our Helm setup chart to install VHI.

## Setting Up K3s

Install K3s on a Linux environment that supports the K3s requirements. This guide was prepared using Red Hat Enterprise Linux 8, however, other supported distributions of Linux will work.

After you install K3s:

Set an environment variable `KUBECONFIG`. Run the command:

```
export KUBECONFIG=~/.kube/config
```

Create the config file:

```
mkdir ~/.kube 2> /dev/null
sudo k3s kubectl config view --raw > "$KUBECONFIG"
chmod 600 "$KUBECONFIG"
```

Add `KUBECONFIG` to your `~/.profile` or `~/.bashrc` to make it persist on reboot.

## Install Helm Package Manager

The Helm package manager is what you use to install, update or remove VHI in Kubernetes. Refer to the Helm install instructions.

## Install Docker command-line interface

Because K3s does not provide container management, you need to provide this. The Docker command line is the standard interface. For example, you can easily obtain podman on Red Hat Linux using the command:

`sudo dnf install docker`

**Tip**

For convenience, set up an alias `docker=podman`.

## Configure a container registry

You need a container registry to store the VHI images. You can use an existing registry, or set one up locally.

To run a test registry:

```
docker run --rm -d -p 6000:5000 --name registry registry:2
```

This test registry does not require authentication. In a production environment your registry should have access control configured. For this, VHI Helm setup can be used with an `imagePullSecret` name if necessary.

Follow these instructions to configure an `imagePullSecret`, and add the name to your `custom-values.yaml` file. See Configure Custom Settings, below.

## Import VHI images to your container registry

Issue the following commands to import the images into your local container registry.

- To import VHI images:

  `docker load -i vhi-airgap-images-99.9.8298.tgz`

- View the loaded images:

  `docker image ls`

- To tag the images:

```
docker tag <image-id> localhost:6000/vhi-mngt:99.9.8298
docker tag <image-id> localhost:6000/vhi-ssvr:99.9.8298
docker tag <image-id> localhost:6000/busybox:1.36
```

> **Tip**
>
> Replace `<image-id>` with the actual `IMAGE ID`s from `docker image ls`

- Verify your tagged images:

  `docker image ls`

- Upload images to the registry:

```
docker push localhost:6000/vhi-mngt:99.9.8298
docker push localhost:6000/vhi-ssvr:99.9.8298
docker push localhost:6000/busybox:1.36
```

> **Note**
>
> You may need to use `docker push --tls-verify=false ...` if you encounter TLS errors with this command.

### Using TLS certificates

You should obtain a server certificate for the machine that will be hosting K3s, and install it as a Kubernetes TLS secret. You specify this ingress secret name in the next section.

> **Note**
>
> You will need to establish trust on client machines (including Java cacerts) when using self-signed or personal-CA certificates.

### Set the cluster certificate manually for the built-in Linux installer

Use the following commands to manually set the cluster certificate:

```
kubectl delete secret traefik-secret -n csp-system
kubectl create secret tls traefik-secret --cert=/path/to/cert/example.crt --
key=/path/to/key/example.key -n csp-system
```

### Configure Custom Settings

For the next steps, you need to obtain your fully qualified host name, using the command: `hostname --fqdn`

In the same directory where `vhi-99.9.8298.tgz` is located, create a file named `custom-values.yaml` ,containing the following entries:

```
    # Custom values for VHI Kubernetes Example Setup
    # K3s uses persistent volume storage class "local-path"
modelPVCStorageClass: local-path
    # FQDN hosting your K3s for VHI administration
adminHostName: <fqdn-hostname>
    # FQDN ingress host and TLS certificate secret name
ingress:
  enabled: true
  host: <fqdn-hostname>
  tls:
    secretName: <tls-secret-name>
    # VHI Management Server image location/name:tag
mngtImage: localhost:6000/vhi-mngt:99.9.8298
    # VHI Session Server image location/name:tag
ssrvrImage: localhost:6000/vhi-ssvr:99.9.8298
    # Init Container Image
initContainerImage: localhost:6000/busybox:1.36
    # VHI Node Port base (change only if port conflicts occur)
basePort: 30500
```

**Note**

More advanced values can be found here.

## Check your setup templates

Test your configuration using the `helm install --dry-run` option.

This command highlights syntax and configuration errors in the .yaml files and provides a preview of what will be submitted to Kubernetes:

```
helm install --dry-run --values custom-values.yaml vhi vhi-99.9.8298.tgz
```

## Install VHI using Helm

Next, install VHI using the `helm install` command:

```
helm install --values custom-values.yaml vhi vhi-99.9.8298.tgz
```

When helm install completes, you should see something like the following:

```
Release "vhi" has been installed. Happy Helming!
NAME: vhi
LAST DEPLOYED: Wed Apr 19 20:08:36 2023
NAMESPACE: default
STATUS: deployed
REVISION: 3
TEST SUITE: None
NOTES:
Installed vhi-mngt and vhi-ss!
```

Run `kubectl get pods` to verify if VHI started properly. This should return a list of the running pods, similar to this:

```
NAME                                   READY   STATUS    RESTARTS   AGE
vhi-mngt-deployment-69f54cb475-xc4xr   1/1     Running    0         10s
vhi-ss-deployment-85454985d8-958wn     1/1     Running    0          7s
vhi-ss-deployment-85454985d8-lntff     1/1     Running    0          6s
```

The first install may take a minute or two for the pod status to get to `Running`. You may see `Initializing` or `Pending` before you see `Running`. Repeat the command to see the current status. You may see some transient errors and restarts, as things are getting set up. Persistent non-running status indicates a problem. See the section, In Case of Difficulty, below.

## Using VHI in Kubernetes

If you are accustomed to using VHI in a traditional environment, there are some things you need to be aware of when running in a Kubernetes environment.

- VHI TCP ports are exposed on NodePort, in the range 30000-32767. The default base port for VHI is 30500.

  When connecting the VHI Administrative Console to the Management Server, provide a port number: `<fqdn-hostname>:30500` (base + 0)

  When using VHI connectors, provide the port number: `<fqdn-hostname>:30523` (base + 23)

- VHI web services are published on standard ports. This includes `/vhi`, `/vhi-ws`, `/vhi-rs` and `/vhi-xe`. You may need to update existing web service clients.
- The default Administrative Console login credentials are username: `admin` and password: `secretpassword`. You can change the password using the VHI Administrative Console: Management Server Explorer, and right click on the Management Cluster. Select Change Admin Password... This new password will persist until you re-install VHI.
- Deploying Models is different using Kubernetes. When you deploy a model, the session servers will need to be restarted before they publish the model. To restart all session server pods, use the command `kubectl rollout restart deployment vhi-ss-deployment`.
- Web Service Explorer is not installed in Kubernetes.
- Design Tool cannot deploy deployment descriptors to a Kubernetes environment. There is a workaround, first define your deployment in the Design Tool. Then you copy the descriptors (typically "deploy_desc.xml") from `<model>/deploy/design_tool` into the folder `<model>/deploy`. After that you can use the deploy function in the Design Tool or the package/activate scripts.
- Testing deployed models is disabled via the Design Tool.
- If you re-install VHI (Helm uninstall vhi, followed by `helm install ... vhi ...`), you will need to re-deploy your models.

  This is because uninstalling reclaims persistent volume storage. If you only need to make changes to your VHI install, use `helm upgrade` instead. Your models will be preserved.

- Configuration changes that you make to individual session server pods using the VHI Administrative Console only persist for the lifetime of the pod. We are evaluating early adopter feedback to determine future needs.
- Host Emulator is not installed in Kubernetes. If you use our demo models, your Host Emulator will need to be installed at an accessible network location outside Kubernetes, and the model should be modified prior to deployment, to access the Host Emulator using that host name or IP address.
- Stateful services will only work with a single session server replica. If you plan to use stateful web services (using `wsResourceCreate`, `rsCreateSessionId` or suspend/resume session with VHI connectors), add the following value to `custom-values.yaml`:

```
    # Number of Session Server replicas
    ssrvrReplicas: 1
```

Then run: `helm upgrade --values custom-values.yaml vhi vhi-99.9.8298.tgz`

## Checking your VHI installation

You can verify that VHI has been properly set up, by running some quick tests.

- Start VHI Administrative Console and connect to the management server. In the Connect dialog, enter `<fqdn-hostname>:30500`. Enter username: `admin` and password: `secretpassword` when prompted for credentials. In the Host Integrator Session Server Explorer panel, you should see two servers, identified by the service pod IP address.
- Connect to VHI Web Services. In a web browser, enter `https://<fqdn-hostname>/vhi-ws`. You should see the SOAP Services catalog page. If there are no models deployed, it is normal for the list of services to be empty. If the server certificate is installed correctly, the browser should display a green or gray lock icon to the left of the browser address bar.

Congratulations! You should now be able to deploy models and start using VHI.

## Trust store for connections between Session Server and host

If your model uses a TLS connection to the host, then certificate validation may fail, causing error "SSL Error - Could not complete the SSL connection". One possible cause is that the certificate returned by the host is self-signed or is issued by a proprietary CA. If the host cannot be changed to use a well-known CA, then you can overlay the trust store with your own.

- Determine which certificate(s) are needed for the connection.
- Place the certificates, in PEM format, in a file with the name `ca-bundle.pem`.
- Create a secret: `kubectl --namespace default create secret generic vhi-ss-truststore --from-file=ca-bundle.pem`
- In your `custom-values.yaml` add `changeTrustStore=true`.
- Upgrade your helm deployment.

## In case of difficulty

- Setup failures: The best approach is to start with a clean setup, and stay as close to the defaults given in this guide as possible. Once you have a basic system up and running, add your customizations one at a time. If something breaks, resolve that problem, or set it aside, and move on to the next thing.

- Image pull backoff: This means that Kubernetes was not able to pull the container image as specified in the `custom-values.yaml` file. Double check for typing errors, and retrace your import process. Verify the container registry image names using `docker image ls`. If your container registry requires authentication, make sure you have correctly configured your image pull secret.
- Crash retry backoff: A problem with provisioning the K3s system, or another environmental problem is not allowing the VHI services to initialize, start or run correctly. Run the command `helm uninstall vhi` to remove VHI. Double check your setup and configuration. Retrace your steps from the beginning and reinstall as necessary.
- Most of the time, `helm upgrade` is sufficient to change your VHI configuration. In many cases, an upgrade will not even cause interruptions in service.
- Run kubectl commands: The Kubectl cheat sheet has many helpful commands for diagnosing problems and general system monitoring.
- Run K3s Dashboard: Kubernetes Dashboard is a web-based visual tool for viewing Kubernetes resources in real time.
- Install K9s: K9s is a text UI dashboard that provides real-time status of multiple pods, live logs, easy access to resource descriptions and other useful diagnostic tools. Follow these instructions to install K9s in your development environment.

## 2.4.4 Verastream Host Integrator Helm Values

When deploying to Kubernetes, Helm uses the values you specify to populate the Helm chart's templates and tailor the deployment to your specification.

| Value Name | Description | Default Value |
|---|---|---|
| ssrvrReplicas | Number of session server replicas. | 2 |
| adminHostName | Change this to the actual FQDN name of your K8s server. | localhost |
| basePort | Base Node Port. The Administrative Console connects to adminHostName:basePort+adminRegistryPortOffset. | 30500 |
| adminRegistryPortOffset | Admin JMX/RMI registry port offset. This value is added to the basePort value. | 0 |

| Value Name | Description | Default Value |
|---|---|---|
| adminExportPortOffset | Admin JMX/RMI export port offset. This value is added to the basePort value. | 1 |
| mngtWcpPortOffset | Management Server WCP node port offset. This value is added to the basePort value. | 41 |
| ssrvrWcpPortOffset | Session Server WCP node port offset. This value is added to the basePort value. | 23 |
| logmgrWcpPortOffset | Log Manager WCP node port offset. This value is added to the basePort value. | 40 |
| wcpNodePortEnabled | When disabled, mngtWcpPortOffset, ssrvrWcpPortOffset, logmgrWcpPortOffset will not be exposed on the NodePort. | true |
| tcpIngressEnabled | When enabled, a Traefik TCP ingress may be used in place of mngtWcpPortOffset, ssrvrWcpPortOffset, logmgrWcpPortOffset. Standard VHI ports will be used. | false |
| modelPVCStorageClass | Storage class used for persistent volume claim. | "" |
| mngtImage | Management server container image. | "" |
| ssrvrImage | Session server container image. | "" |
| snmpPort | Internal SNMP trap target port. | 162 |
| initContainerImage | Container Image used to initialize the Session Server and Management Server pods. | busybox:1.36 |
| ingress.host | The name of your virtual host pointing to your Kubernetes platform. A subdomain may be specified here. | "" |
| ingress.secret | The name of the Kubernetes secret for your server certificate and key. | "" |
| pod.dnsPolicy | DNS name resolution policy. See Kubernetes Documentation. | "ClusterFirst" |

| Value Name | Description | Default Value |
|---|---|---|
| imagePullSecrets.name | The name of the Kubernetes secret used to pull images from a container registry. | "" |
| changeTrustStore | When enabled, overlay the session server trust store with a Kubernetes secret. | false |

# 2.5 Installing on Windows

There are two installation options available when installing for the first time; Typical or Custom.

## 2.5.1 Install steps

1. Download the product from the download site, following the instructions to start the Host Integrator setup program.
2. Run the install setup, read and accept the license agreement, click Continue and then select an installation type; Typical or Custom.

   - Typical - All components are installed and configured to the local machine using the default configuration values.. All services will be automatically started when the install is complete and the session server will be registered with the management server.

     The default installation directory is: `C:\Program Files\RocketSoftware\Verastream`

     Specify a password for the administrative account of the management server. For example, `admin`. This password is important. You need this password when you first log in to the Administrative Console.

   - Custom - If you select Custom, you can select an installation directory, choose the components you want to install, join an existing installation, or configure advanced installation options available from the tabbed menu pages.

     To install a Host Integrator Server without installing a management server, you must join an existing installation, which supports failover and load distribution.

     To do this, you need to supply the necessary credentials, the existing server address, and port number. Login credentials include either the user name and password created when the first server was installed or the credentials of a user added to the administrator profile in the Administrative Console.

You can also register a management server with an existing installation forming a management server cluster. When you combine this option with the registration of the session server, you attain automatic replication to the management server cluster.

3. Click Install Now to continue the installation. By default, all services start automatically.

> **Note**
>
> If the configuration fails for either the management cluster or management server, you cannot re-enter the information. If your installation fails, an error message displays asking you to either retry the install, stop the uninstall, or to ignore. The Ignore option leaves the system installed, but not configured. If any of the installation steps fail, check the `addmsrvr.log`, `addssrvr.log`, or `changepwd.log` in the `%TMP%` directory for the reason.

### About Windows services

The Management Server, Log Manager, Session Server, Web Server, and Host Emulator are installed as Windows services. For example, Management Server is listed as Verastream Host Integrator Management Server in the Services panel available from the Windows Control Panel. You can modify the Stop and Start settings for these services there.

Host Integrator servers and Host Emulators can be started from the Administrative Console.

## 2.5.2 Creating an unattended Windows install file

To install VHI silently, you can specify all necessary installation and configuration options in a separate file.

From the root of the CD, locate and open the setup.ini file.

Depending on whether you want to join an existing management server cluster, add the appropriate code to the end of the file.

To do an independent install:

```
[VHI]

JoinExisting=0

MgmtAddress=localhost

MgmtUsername=admin

Password=secretpassword
```

5. To join an existing installation:

```
[VHI]
```

```
JoinExisting=1
MgmtAddress=<address of management server>
MgmtUsername=<user name existing installation>
Password=<password of management server>
```

6. After you save the file, from the command line run `setup/install`.

   You can add these options:

| Option | Command |
|---|---|
| Run in unattended mode displaying only the progress bar without a Cancel button. | `/passive` |
| Run in quiet mode which requires no user interaction. | `/quiet` |
| Optional. Specify the components you want to install. These are described on the component install panels. | `ADDLOCAL=componentID`, `componentID`, `componentID` For example, `setup /install ADDLOCAL=HostIntegrator, COMConnector, Server` |

If necessary, you can change the installation location during a silent install using the property, INSTALLDIR. The value must be quoted if it contains spaces.

For example: `setup /install INSTALLDIR="C:\Program Files\RocketSoftware\Verastream"`

## 2.5.3 Removing Host Integrator from Windows

Use the Add/Remove Programs option from the Windows Control Panel.

If you want to remove a session server with security enabled, you will be prompted to supply a User ID and password. The credentials must be those that you specified in the Administrator profile in the Administrative Console when the server was registered.

The uninstall program removes all installed Host Integrator components, but does not remove log files, user-supplied model files, or Web Builder projects.

### Adding or removing Host Integrator Components in Windows

1. If you installed from the download site, follow the procedures described in the download instructions to start the Host Integrator setup program.

2. To start the installation program, click Start > Run, type the path to the installation program (for example, D:\Setup.exe), and then click OK.
3. When the maintenance screen displays, select Modify from the available options. Click Continue.
4. From the Select Components screen specify the components you want to add or remove. Components that are currently installed are selected. Clear the check box to remove a component. Continue to use the check boxes to specify the components you want to add or remove, and then click Continue.
5. If Setup is adding or removing a Host Integrator Session Server with security enabled from an existing installation, it will prompt you for the user ID and password for an Administrator found in the Authorization panel of the Administrative Console with which the management server is registered. Enter the user ID and password that belongs to the administrator specified in the Authorization panel of the Administrative Console, click Continue.
6. As Setup adds and removes components, it may encounter a locked file. To ensure that you have no unnecessary files remaining on your machine after the installation completes you may be prompted to re-start the computer.

## 2.6 Installing on Linux

On the download site, choose the installer that identifies the operating system and version of Host Integrator that you want to install. Install programs are designated with either –prod for the production version or –eval for the evaluation version. Each installer performs the following steps, which cannot be executed individually:

Installs the chosen products.

Continues with the setup of the installed products.

Configures the servers.

### 2.6.1 Interactive installation

**Note**

These instructions are for a system without a current version of VHI installed. If you have a previous version of VHI installed, see the Upgrading on Linux section.

In this example the Linux installer is used to install the production version (-prod) of Host Integrator.

To install Host Integrator components on Linux systems:

1. Log in as root. You can create a new non-root user and group specifically for running Host Integrator, such as vhiuser and vhigroup. To specify the user and group, use the --owner option. See Additional Installation and Configuration Options.
2. Type the following command on the Linux prompt: `sudo sh ./vhisrv-7.x.x-prod-<linux> [--owner vhiuser:vhigroup]`
3. Use either the `y` or `n` command to respond to the license agreement. If you select `n`, the installation stops.
4. Choose to initiate a Typical (t) or Custom (c) install process.

   - Typical - A standalone installation which includes all default components and configurations. The session server and the management server are installed and the session server is automatically registered with the local management server.
   - Supply and verify an administrative password. This password is required for later configurations.

   The installation continues with the default installation of all VHI components.

   - Installation proceeds with no option to “go back”.
   - Custom - In a custom installation you manually select components, join existing servers, and configure other required options from the configuration options menu. Continue customizing your installation, choosing the components you want to install.

5. To continue you must specify the following menu category options:

   - Components - To select or clear a component enter the corresponding number and press: ENTER. By default all components are selected.
   - Options - Installation location and owner. The default location is `/opt/rocketsoftware/verastream`. To specify the owner:
   - Choose the owner (and optional group) for all files and directories under the base location where the product is being installed.

   To specify a non-root user and group ownership use one of the following methods: menu option "o" in interactive custom installation, --owner option on the installer command line, or owner= line in the install-input file for an unattended automated installation.

   - Specify user or user:group

     After installation by root is complete, you can directly log in as the specified user to manage the product.

   - Actions - Specify whether to continue, quit or display help.

> ⚠ **Caution**
>
> If there is still a partial VHI installation on the machine, the default value will hold the value from that installation. This is generally true for all configurable options and settings.

Depending on what components you selected, there may be further configuration needed:

| If selected | Do this |
|---|---|
| Host Integration Server | To successfully install a Host Integration server you must either install a management server or join an existing installation. |
| Management Server or Host Integration Server and Management Server | If you choose either of these scenarios, you can associate the management server with an existing management server cluster. You can combine this option with the registration of the Host Integrator Server, allowing for automatic replication to the management server cluster. |
| Any other combination of components | The option c) continue will be changed to i) install |

If the Host Integration Server is selected without a management server, you need to specify the following options to join an existing installation:

```
Verastream Host Integrator Version 7.x
Installation
Configurations:
1) Management Server address:
2) Management Server username:
3) Management Server password:
Actions:
i)   Install
q)   Quit
h)   Help
Make a selection:
```

- Server address - Enter 1 - The address is checked for format (Ex: 192.168.1.1:3000), address and port number) and existence of the server.
- Server username - Enter 2 - Either the built-in “admin” user, or another user that has been given administrator rights in the management server.
- Server password - Enter 3 - (The password is neither echoed nor shown)

**Note**

If a failure occurs while registering the management server, session server, or changing the password, check the log file located in the installation path for information.

When prompted to join an existing cluster, if you select yes, specify the configuration options in the screen below. See the previous table for a description of the commands.

```
Verastream Host Integrator Version 7.x
Installation
Configurations:
1) Join an existing cluster: Yes
2) Management Cluster address:
3) Management Cluster username:
4) Management Cluster password:
Actions:
i)   Install
q)   Quit
h)   Help
Make a selection:
Run the i) install command to complete the installation.
```

## 2.6.2 Installing additional components

If you did not install all the components initially, you can re-start the installer and add components to an existing installation

## 2.6.3 Additional installation and configuration options

In addition to the traditional installation method you may elect to install from the Linux command prompt where you have full control over the installation using the following commands:

For a list of commands, type: `sh vhisrv-7.x.x-prod-linux --help`

Use these commands to perform the following:

| Command | Description |
|---|---|
| basepath | Choose another base installation location. |

| Command | Description |
|---|---|
| owner | Choose the owner (and optional group) for all files and directories under the base location where the product is installed. This same owner is used to start all server processes. Specify user or user:group. After the installation by root it will be possible to directly log in as the specified user and manage the product. |
| licenseagreed | The EULA license is not shown and the install program assumes you have read and accepted it. You can initiate a complete unattended install when --input is used. |
| input | Specify a file that contains all necessary information to install the product without interaction. By default the install only outputs errors and warnings. Additionally if you specify `--verbose` the different executed steps are written to standard output. |
| createinput filepath | When this option is used with an interactive install, all choices taken are written to the specified file. This file can next be used as input for a silent install. |
| verbose | Provides detailed output when `--input` is specified. |
| no_entropy_check | Do not check available entropy before starting the session server |

In addition to the above commands the following commands are configurable. You can use these options to specify or override the default values as explained in the previous Interactive Installation section.

| Configuration Command | Description |
|---|---|
| join_cluster | True or False |
| cluster_address | Set the management cluster address in the format (address:port) |
| cluster_user | Set the management cluster user name |
| cluster_password | Set the management cluster password. Use this option if you do not want the management cluster password recorded in the input file. You must enter the password once at the start of the installation, and then the install is silent. |
| | |

| Configuration Command | Description |
|---|---|
| password | Set the password |

### Examples using commands

```
sh vhisrv-7.x.x-prod-linux --owner vhi:vhigroup --basepath /opt/vhi
```

The above command specifies the owner and group used to start all server processes and manually specifies the installation path.

```
sh vhisrv-7.x.x-prod-linux --join_cluster true --cluster_address cluster_server.com:33000 --cluster_user admin --cluster_password secretpassword
```

This is used to join an existing cluster by specifying the required credentials and cluster configurations.

## 2.6.4 Automatically start services

To configure the session server to run as a system daemon and have the services start automatically as system services, you need to add a script to your system init.d or rc.tcpip configuration.

These instructions will start all services, including the management server if it is installed.

Create a file called vhi containing the following and enter in your installation directory.

To start a particular service, replace the parameters with one of the following:

server —VHI session server

mgmtserver —VHI management server

logmgmr —VHI log manager

hostemul —VHI host emulator

```
### BEGIN INIT INFO
# Provides: VHI
# Required-Start: $network
# Should-Start: $network
# Required-Stop: $network
# Should-Stop: $network
# Default-Start: 3 5
# Default-Stop: 0 1 2 4 6
# Description: Rocket Verastream Host Integrator Services
### END INIT INFO
INSTALL_DIR=<enter installation directory>
BIN_DIR=$INSTALL_DIR/hostintegrator/bin
case "$1" in
start)
echo "Starting Verastream"
$BIN_DIR/atstart -start all
RETVAL=0
;;
stop)
echo "Stopping Verastream"
$BIN_DIR/atstart -stop all
RETVAL=0
;;
status) echo "Current Verastream status"
$BIN_DIR/atstart -status
RETVAL=0
;;
restart) echo "Restart Verastream"
echo "-- stopping all components --"
$BIN_DIR/atstart -stop all
echo "-- starting all components --"
$BIN_DIR/atstart -start all
RETVAL=0
;;
*)
echo "Usage: $0 {start|stop|status|restart}"
RETVAL=1
;;
esac
exit $RETVAL
```

Select your platform and complete the following steps.

Linux and zLinux

Copy the file to the `/etc/init.d` directory

Set the file permission. Run `chmod` using the value `755`. For example, `chmod 755 vhi`

Run `chkconfig` to add the initialization script. For example, `chkconfig --add vhi`

To run the services as a non-root user, create a new script that will run the provided sample script.

### Testing changes

You should test manually stopping and starting services while logged in as the non-root user and verify that services are automatically started after you restart the system.

- For more information on manually starting and stopping Host Integrator services, see the online help topic, Starting and Stopping Services.
- To verify services are successfully running, see Technical Note 7021540.
- If services do not start successfully, check the operating system log as described in Technical Note 7021303, Operating System Logs section.

### Local authentication requires root privileges

One of the Host Integrator processes may need to run as root, depending on your Host Integrator version and security configuration:

- If you enable local OS groups in Administrative Console (Management > Directories > Properties), then the management server service, along with the other services, must run as root. However, this configuration is typically not necessary as VHI provides the following alternative for authentication security:

  Secured administrative access using built-in “admin” user name and administrative password (set during installation or in Administrative Console).

  Improved support for LDAP directory services, such as Microsoft Active Directory. You can add users and groups from your directory server to the Administrator, Developer, and User authorization profiles.
- See Technical Note 7021354 for more information on this topic.

## 2.6.5 Creating an unattended install file

You may install VHI silently, by specifying all necessary installation and configuration options in a separate file. Assuming this file is called install-input, it is enough to type this command on the command line:

```
sh vhisrv-7.x.x-prod-linux --input install-input --licenseagreed
```

**Note**

An ‘unattended install’ presumes that you have already accepted the software license agreement applicable to this product (most likely, in the course of a software download process, or an initial product installation). A copy of the software license agreement is provided in the root directory of the product, in case you wish to review it before proceeding with this setup. Optional switch --verbose provides additional output. By default only errors and warnings are shown.

### Example

File `install-input` (below) contains all previously described options and is a sample delivered with the installation. In this file all components are selected and all configuration options are commented out.

```
#Component selection
sessionserver=true
managementserver=true
hostemulator=true
connectors=true
#Specify the action to perform when there is already a VHI installed
#upgrade=false
#add=false
# Installation location
# basepath=/opt/rocketsoftware/verastream
# Owner of installation and startup id servers
# owner=vhiuser:vhigroup
# Management Server values to register Session Server with another
Management Server
# vms_address=mgmt_server.rocket.com:33000
# Does the Management Server join a Management Cluster?
# join_cluster=true
# Management Cluster configuration
# cluster_address=cluster_server.rocket.com:33000
# Perform an entropy check for the session server?
# entropy_check=true
# General username and password; for new or existing Management # # Server
or Management Cluster
# server_username=admin
# server_password=secretpassword
```

### Notes

If sessionserver, managementserver,hostemulator or connectors are set to true, the component is installed. If you set the value to false, add # in front of the option or leave the component out; the component will not be installed.

If a line begins with ‘#’, the option is ignored. In the example above all configuration options are commented out in this way.

If you use the example file above (if the comments are removed), the session server, web server, management server, Host Emulator, and connectors are installed. The management server is registered with cluster server cluster_server.rocket.com and the session server is registered in the cluster through replication.

You only need to use the vms_address option when no management server is installed (managementserver=false). When that is the case the vms_address option specifies where to register the session server. Otherwise, this option is ignored. When the Host Emulator is installed along with the session server, it is registered to the same management server.

No interaction is necessary and no output is given unless an error occurs.

### Entropy

```
WARNING! Available entropy is critically low.

Low entropy will decrease performance and cause the installation to fail.

To override, use the --no_entropy_check flag
```

When you install the session server, the installer checks the level of entropy available.

Low entropy causes performance issues and can lead to the install program failing to register the session server. If there is less than 1000 bytes of entropy available, the install program will exit.

To override this behavior, you can use the --no_entropy_check flag or set the entropy_check variable to true in the unattended install file.

See Technical Note 7025092 for more information on viewing and increasing entropy.

## 2.6.6 Uninstalling Host Integrator

To run the uninstall process follow the steps below:

> **Tip**
>
> The file `vhi_uninstall.sh` is located in the Verastream Host Integrator installation directory.

From the base installation directory, type: `sh vhi_uninstall.sh --sessionserver --managementserver`

2. Confirm the removal:

   Choose yes to stop all servers and remove all files and directories including the base installation directory. When yes is chosen, all servers are stopped and all known files and directories in the base installation directory are removed. When the base directory is empty, the directory itself is also removed.

   You can remove either the session server or the management server. See step 1 above and enter just `--sessionserver` or `--managementserver`. This option only removes files and directories not being used by the remaining component.

> **Note**
>
> The installer does not remove the registration of the session server or management server. You must make registration changes manually with the help of the Administrative Console.

3. Type `sh vhi_uninstall.sh --help` to display the help screen if needed.

```
Rocket Verastream Host Integrator and/or Management Server uninstall
Remove all files from a prior installation,
including all configuration data that may be present.
Usage: ./vhi_uninstall.sh [options]
Options:
  --basepath path         : Base installation path
  --managementserver      : Remove Management Server
  --sessionserver         : Remove Host Integration Server
  --hostemulator          : Remove Host Emulator
  --connectors            : Remove Connectors
  --all                   : Remove all installed VHI components
  --verbose               : Detailed output
  --yes                   : Answer yes to all questions
  --help                  : Display this message.
```

## 2.7 Upgrading from Previous Versions of Host Integrator

If you are upgrading to Verastream Host Integrator 7.8.x from a previous version read this section to determine how best to proceed with the installation.

If you are upgrading from versions earlier than 7.7.x, see the Upgrade section of the support site.

You cannot use this upgrading process to apply VHI hot fixes. See the support download page on the Technical Support for instructions on applying hot fixes.

**Note**

All VHI versions 7.5 and higher support SHA-256 security certificates with 2048 bits keys. When you upgrade from a previous version, existing certificates (installed with the previous version) are maintained.

## 2.7.1 Upgrading on Windows

To begin the migration process, perform the following steps:

As a good practice, back up your existing project files.

Run setup.exe from the current version’s install image.

A message confirms that a previous installation has been detected. You are prompted to continue. Click Yes.

The installation completes with a message indicating its success.

## 2.7.2 Upgrading on Linux

This process includes the following steps:

Make a note of your existing administrative password set during original installation or changed in Management>Servers>Management Cluster>Change Admin Password.

Back up existing files

Install VHI.

A message confirms that a previous installation has been detected. You are prompted to continue. Click Yes.

The installation completes with a message indicating its success.

You can specify `--owner user:group` to install VHI as non-root. Whatever user is used, either non-root or root, that user must also be used to upgrade, uninstall, and reinstall VHI. If root needs to assume ownership of a non-root installation, VHI must be uninstalled and the directory, in this example, `/opt/rocketsoftware/verastream` is deleted. The new installation will be owned by root or whatever user you specify.

## 2.7.3 Upgrading Model Files and Web Applications

Your deployed models from the previous version will continue to function on a new session server. No manual upgrade is required. When you open a model from the previous version in the Design Tool, it is saved as a model for the current version. There are additional compatibility switches to allow older models to run without being affected by some newer enhancements.

However, your Java Web applications are not automatically deployed after an upgrade and you must rebuild the Web Builder project and redeploy it.

# 3. Legal Notice