

Cloud Native Unica V12.1.8 Getting Started Guide



Contents

- Chapter 1. An overview to the Cloud Native Unica Getting Started Guide..... 1**
- Chapter 2. Cloud Native Unica Overview..... 3**
 - Features and Benefits.....3
- Chapter 3. Docker, Kubernetes, and Helm..... 5**
 - Docker overview..... 5
 - Kubernetes overview.....5
 - Helm overview.....5
 - Helm charts overview..... 6
- Chapter 4. ELK Overview..... 7**
- Chapter 5. Minimum hardware requirements.....8**
- Chapter 6. Software requirements..... 9**
- Chapter 7. Supported cloud platforms..... 10**
- Chapter 8. Minimum required resources for containers..... 11**
- Chapter 9. Downloading Cloud Native Unica image..... 12**
- Chapter 10. Cloud Native Unica Helm release architecture.....13**
- Chapter 11. Cloud Native Unica Reference Materials..... 14**

Chapter 1. An overview to the Cloud Native Unica Getting Started Guide

Cloud Native Unica Getting Started Guide gives you a quick glimpse about the basic details of the Cloud Native Unica product

How to benefit from this guide?

The Getting Started Guide provides you the following information:

- An overview of Cloud Native Unica.
- The features and benefits of Cloud Native Unica.
- Introductory details related to Docker, Kubernetes, Helm, Helm charts, and ELK.
- The minimum or recommended hardware and software requirements.
- The supported cloud platforms.
- The minimum or recommended required resources for containers.
- How to download the Cloud Native Unica image.
- Helm architecture of Cloud Native Unica.

We recommend new users of Unica and existing users of Unica who are new to Cloud Native Unica to consume all the information available in the guide to get a basic understanding about Cloud Native Unica. The existing users of Cloud Native Unica can refer only the topics related to system requirements to see if there are any updates related to the requirements.

Who can benefit from this guide?

The following users can benefit from the guide:

- New users of Unica
- Existing users of Unica who are new to Cloud Native Unica
- Existing users of Cloud Native Unica

Users	Topics to be consumed
<ul style="list-style-type: none">• New Users• Existing Users of Unica who are new to Cloud Native Unica	<p>Since you are new to Cloud Native Unica, we recommend you to refer all the following topics to get the introductory information:</p> <ul style="list-style-type: none">• Cloud Native Unica Overview on page 3• Features and Benefits on page 3• Docker, Kubernetes, and Helm on page 5• ELK Overview on page 7• Minimum hardware requirements on page 8• Software requirements on page 9• Supported cloud platforms on page 10

Users	Topics to be consumed
	<ul style="list-style-type: none">• Minimum required resources for containers on page 11• Downloading Cloud Native Unica image on page 12• Cloud Native Unica Helm release architecture on page 13
Existing users of Cloud Native Unica	<p>Since you are the existing users of Cloud Native Unica, you can only refer the following topics to see if there are any updates in the new version:</p> <ul style="list-style-type: none">• Minimum hardware requirements on page 8• Software requirements on page 9• Supported cloud platforms on page 10• Minimum required resources for containers on page 11

Chapter 2. Cloud Native Unica Overview



Features and Benefits

The features and benefits of Cloud Native Unica are as follows:

- **Why Unica Cloud-Native?**

We are amidst an information technology revolution driven by the cloud. The emergence of containers, with their fast startup, standardized application packaging, and isolation model, is further contributing to efficiency and agility.

Along with the benefits of Docker like reducing hardware resource costs, ease-of-use, portability, scalability, and modularity of deployments, Kubernetes provides container-orchestration capabilities for automating application deployment (rollout and rollbacks), workload scaling, and high availability. Helm charts leverage Kubernetes packages to streamline the installation and management of applications deployed on Kubernetes.

- **Faster release pace**

Unica Docker Images will be rolled out for each release and/or fix pack. Time to market has become the key differentiator between the most innovative organizations and their lagging competition. With this deployment approach, Unica can build and ship more value to its customers. With this deployment approach, it is very easy for customers to consume new releases.

- **Better CX**

HCL Unica shipd Docker Images with new features faster and keep iterating continuously. An extensive set of APIs in the Unica Products would enable integrations with enterprise data stores. Overall, Cloud-native applications allow you to improve your customer experiences.

- **Ease of Application management**

Cloud-native also has many options to make infrastructure management effortless. With helm charts, application management, monitoring, deployments will be easy, automated, and configuration driven.

- **Reduced cost through containerization**

Containers make it easy to manage and secure applications independently of the infrastructure that supports them. The industry is now consolidating around Kubernetes for the management of these containers at scale. Alongside Kubernetes, there are a host of powerful cloud-native tools and standardization of infrastructure and tooling. This, along with open-source technologies, drives down costs. Pricing flexibility models are all enabled with a cloud-native deployment approach.

- **Building Reliable systems**

With modern cloud-native approaches like Kubernetes in the cloud, you can more easily build applications to be fault-tolerant with resiliency and self-healing built-in. Because of this design, even when failures happen, you can easily isolate the impact of the incident, so it doesn't take down the entire application. Instead of servers and monolithic applications, cloud-native microservices helps you achieve higher uptime and thus further improve the user experience.

- **Deploy Anywhere**

Unica solutions can be deployed on any cloud. Unica can be deployed on Managed Kubernetes Clusters like EKS, GKE, AKS, etc.

Chapter 3. Docker, Kubernetes, and Helm

In this release, we have implemented Cloud Native Unica using Docker, Kubernetes, and Helm.

The following topics provide an overview of the mentioned technologies:

- For an overview on Docker, see [Docker overview on page 5](#).
- For an overview on Kubernetes, see [Kubernetes overview on page 5](#).
- For an overview on Helm, see [Helm overview on page 5](#).
- For an overview on Helm Chart, see [Helm charts overview on page 6](#).

Docker overview

Docker is an open source software that makes it easy to create, deploy, and manage virtualized applications using containers.

You can use containers to package applications with the necessary components to run the applications, like libraries and other dependencies. Because a container has all the components requires for its execution, it is not dependent on other containers and can run in an isolated manner.

The software that hosts the containers is called a Docker Engine. The Docker Engine creates containers on top of an operating system and automates application deployment on the container.

Docker-based architecture also offers standardization. With standardization of service infrastructure, every team member can work on a production parity environment.

For more information on Docker, its prerequisites, and the minimum system requirements, see <https://docs.docker.com/>.

Kubernetes overview

Kubernetes is an open source container orchestration system that provides a platform to automate deployment, scaling, and management of application containers across clusters of hosts.

If you have a cluster containing groups of hosts running Linux containers, you can use Kubernetes to manage the clusters efficiently. These clusters can span hosts across public, private, or hybrid clouds. This makes Kubernetes an ideal platform for hosting cloud-native applications that require rapid scaling.

For more information on Kubernetes, its prerequisites, and the minimum system requirements, see <https://kubernetes.io/docs/home/>.

Helm overview

Helm is a package manager for Kubernetes that can be used by developers and operators to easily package, configure, and deploy applications and services onto Kubernetes clusters.

Helm can:

- install and upgrade software
- automatically install software dependencies

- configure software deployments
- fetch software packages from repositories

Helm packages are called charts and they contain a few `YAML` files and templates that are rendered into the Kubernetes manifest files.

For more information on Helm, its prerequisites, and the minimum system requirements, see <https://helm.sh/>.

Helm charts overview

Helm Charts are packaging formats. A chart is a collection of files that describe a related set of Kubernetes resources.

You can use a single chart to deploy something simple, like a `memcached pod`, or something complex, like a full web application stack with `HTTP` servers, databases, caches, etc.

Charts are created as files laid out in a specific directory tree, and you can package charts into versioned archives for deployment.

Chapter 4. ELK Overview

The ELK Stack is a collection of three open-source products: Elasticsearch, Logstash, and Kibana.

The features of the ELK stack are as follows:

- Centralized logging and monitoring to identify problems with containers, or applications, hosted inside the pod
- Visualization tools to represent data using a graph or a chart
- Host ELK as a standalone system or a Docker container or a Kubernetes pod

Filebeat

Filebeat collects and ships log files, and is also the most commonly used beat. You can install Filebeat on almost any operating system, or also as a Docker container. It includes internal modules for platforms like Apache, MySQL, Docker, MariaDB, Kafka, and many more.

Filebeat is very efficient and it displays this in how it handles backpressure. If Logstash is busy, Filebeat slows down its read rate and picks up the beat once the slowdown is completed.

For Unica, Filebeat is deployed as a sidecar container inside Platform pod, and Platform and Filebeat, as a sidecar, runs as a container inside a pod. The Filebeat has a read-only access to persistent volume. It reads the logs from the logs folder, specified in configmap for all Unica applications, to keep sending logs to ELK.

metricbeat

Metricbeat collects and reports various system-level metrics for various systems and platforms. Metricbeat also supports internal modules for collecting statistics from specific platforms.

For Unica, metricbeat is deployed as a daemonset in `kube-system` namespace to collect metrics from the metric-server in OpenShift. In Kubernetes, it connects to both the Kubernetes API-server and the metric server.

Fluentd

It is a smart metrics and log shipper. For Unica, Fluentd is deployed as a daemonset in custom namespace to collect metrics from Kubernetes.

Advantages of using ELK

Logging, keeping historical logs, or monitoring the logs is a real challenge in containerized applications. If you destroy a container, everything is lost, including the logs. The advantages of using ELK are as follows:

- Maintains and keeps all the data, and makes this activity easy, even if the cluster, pod, or node is destroyed.
- Allows searching of all the logs in a single place.
- Helps find issues in multiple servers, or pods, by connecting logs during a specific time frame.

Chapter 5. Minimum hardware requirements

The minimum hardware requirements for Cloud Native Unica are as follows:

Memory	64 GB RAM
Storage	150 GB Hard disk drive
CPU	8 CPU machines

Chapter 6. Software requirements

The recommended software requirements for Cloud Native Unica are as follows:

Table 1. Recommended Software Requirements

Software type	Recommended requirement
Operating System	Cloud Native Unica supports: <ul style="list-style-type: none"> • CentOS version 8 • RHEL version 8 • SUSE Linux Enterprise Server 12 Service Pack 5 (hosted operating system)
Base Operating System	The Base Operating System used for all images is Red Hat Universal Base Image 9.
Database	Cloud Native Unica supports: <ul style="list-style-type: none"> • DB2 • MariaDB • Microsoft SQL Server • OneDB • Oracle • PostgreSQL
Application Server	Cloud Native Unica supports: <ul style="list-style-type: none"> • Red Hat® JBoss® Enterprise Application Platform (EAP) 7.4.0 • Apache Tomcat® 9.0.86 • Oracle® WebLogic Server 12.2.1.4.0
Other Software	<ul style="list-style-type: none"> • Apache ActiveMQ 5.15.8 • Docker Enterprise version 20.10.17 • Kubernetes versions 1.21, 1.22, and above • Helm version 3 • EKS, AKS, GKE, and OpenShift • Ingress: New ingress controller repo: https://kubernetes.github.io/ingress-nginx NGINX Ingress Controller 2.0 • Kafka Standalone is not packaged with Cloud Native Unica for Unica Journey. You must either have Kafka Standalone installed on your system or you must purchase Kafka Standalone and install it.

Chapter 7. Supported cloud platforms

Cloud Native Unica supports the following Cloud platforms:

- Amazon Elastic Kubernetes Service (Amazon EKS)
- Google Cloud Platform (GCP)
- Microsoft® Azure
- RedHat OpenShift®

Chapter 8. Minimum required resources for containers

The following table lists the minimum resources required by each container:

Table 2. Resources required by each container

Resource Name	Resource Value
CPU	2000m (2000 millicpu per container).
Memory	2048 MB.
Storage	NFS is supported with Unica as persistent volume Openshift - NFS Azure - Azure Files Google - File storage If the containers fail and restart, the installation and the database data persist. However, setups and JDBC drivers are not packaged with the solution and they need to be placed at the mount point.



Note:

- The configuration will impact the minimum hardware requirements. It is recommended that you use a configuration that is slightly higher than the minimum requirements. If a container does not get the required resources, it might hang.
- **Docker Image Security Vulnerabilities:** For Operating System package vulnerabilities, like NSS, SQLite, etc., run the yum updates within the containers.

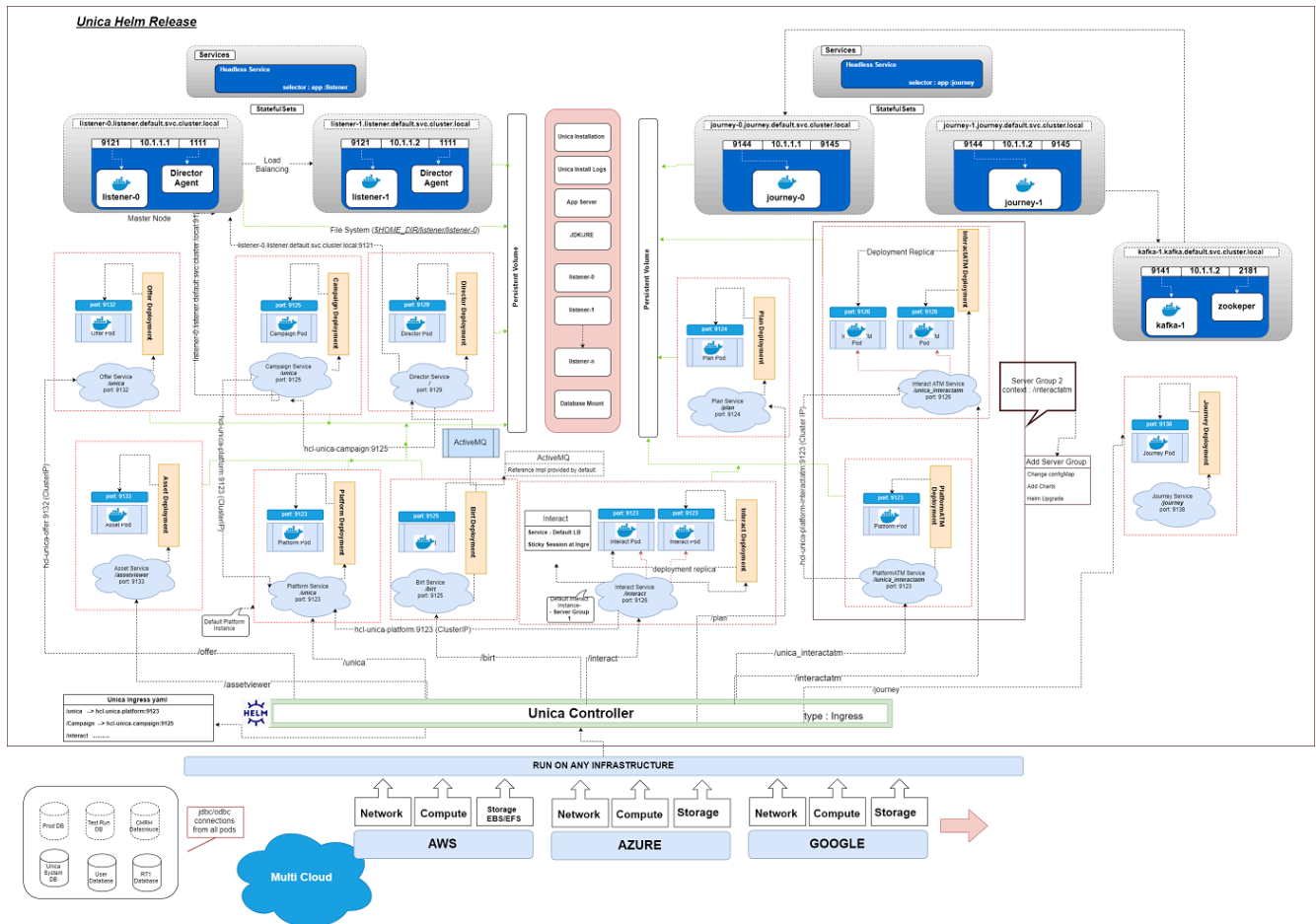
Chapter 9. Downloading Cloud Native Unica image

Download the required image from Flex Net Operations (FNO). By default, the chart uses Cloud Native Unica image.

- To roll out the Helm charts, you must specify the offering-related details and requirements. Contact Unica support team to get a Helm chart.
- Although Cloud Native Unica images are CentOS-based or RHEL-based, it may work on any Unix-based operating systems. HCL Software will provide support only for CentOS and RHEL operating system. HCL will not provide support for any other operating systems.
- Although Oracle drivers are bundled with Cloud Native Unica images, HCL Software will not provide support for any issues related to third-party software or third-party drivers bundled with Cloud Native Unica images.

Chapter 10. Cloud Native Unica Helm release architecture

The following is a diagrammatic representation of the Cloud Native Unica Helm release architecture:



[Click here](#), to access HTML help, for a better resolution of the image.

Chapter 11. Cloud Native Unica Reference Materials

After downloading the Cloud Native Unica image from FNO, access the Cloud Native Unica product documentation to implement it on your setup.

The product documentation for Unica is available for consumption in the PDF format and HTML format. You can access the Unica documentation from the following URLs:

- [PDF format](#)
- [HTML format](#)

For details related to latest release of Cloud Native Unica, see *Cloud Native Unica Release Notes*. This document is ideal for new users and existing users to know the following details:

- New features in the release
- Defects fixed from earlier releases
- Known issues or limitations
- Contacting HCL Technical Support

For details related to installation and configuration of Cloud Native Unica, see *Cloud Native Unica Installation and Configuration Guide*.

For exhaustive information related to implementing Cloud Native Unica using helm charts, see the following application server-specific implementation guides:

- *Cloud Native Unica Implementation Guide for Apache Tomcat.*
- *Cloud Native Unica Implementation Guide for Red Hat JBoss Enterprise Application Platform.*
- *Cloud Native Unica Implementation Guide for Oracle WebLogic Server.*