

Enterprise Edition

On this page

- [What is YugaWare?](#)
- [How does the installation work for YugaByte DB Enterprise?](#)
- [What are the OS requirements and permissions to run YugaWare?](#)
- [What are the OS requirements and permissions to run the YugaByte DB data nodes?](#)
- [How are the build artifacts packaged and stored for YugaByte DB Enterprise?](#)
- [How does the Admin Console interact with the YugaByte DB data nodes?](#)
- [Would we have access to the database machines that get spawned in public clouds?](#)
- [How many machines would I need to try out YugaByte DB against my load?](#)
- [Can we control the properties \(such as VPC, IOPS, tenancy etc.\) of the machines YugaWare is spinning up?](#)

What is YugaWare?

YugaWare, shipped as a part of YugaByte DB Enterprise, is the Admin Console for YugaByte DB. It has a built-in orchestration and monitoring engine for deploying YugaByte DB in any public or private cloud.

How does the installation work for YugaByte DB Enterprise?

YugaWare first needs to be installed on any machine. The next step is to configure YugaWare to work with public and/or private clouds. In the case of public clouds, YugaWare spawns the machines to orchestrate bringing up the data platform. In the case of private clouds, you add the nodes you want to be a part of the data platform into YugaWare. YugaWare would need ssh access into these nodes in order to manage them.

What are the OS requirements and permissions to run YugaWare?

• OS requirements

Only Linux-based systems are supported by Replicated at this point. This Linux OS should be 3.10+ kernel, 64bit and ready to run docker-engine 1.7.1 - 17.06.2-ce (with 17.06.2-ce being the recommended version). Some of the supported OS versions are:

- Ubuntu 16.04+
- Red Hat Enterprise Linux 6.5+
- CentOS 7+

- Amazon AMI 2014.03 / 2014.09 / 2015.03 / 2015.09 / 2016.03 / 2016.09

The complete list of operating systems supported by Replicated are listed [here](#)

- **Permissions necessary for Internet-connected host**

- Connectivity to the Internet, either directly or via a http proxy
- Ability to install and configure [docker-engine](#)
- Ability to install and configure [Replicated](#), which is a containerized application itself and needs to pull containers from its own Replicated.com container registry
- Ability to pull YugaByte container images from [Quay.io](#) container registry, this will be done by Replicated automatically

- **Permissions necessary for airgapped host**

An “airgapped” host has no path to inbound or outbound Internet traffic at all. For such hosts, the installation is performed as a sudo user.

- **Additional requirements**

For airgapped hosts a supported version of docker-engine (currently 1.7.1 to 17.03.1-ce). If you do not have docker-engine installed, follow the instructions [here](#) to first install docker-engine.

- Following ports should be open on the YugaWare host: 8800 (replicated ui), 80 (http for yugaware ui), 22 (ssh)
- Attached disk storage (such as persistent EBS volumes on AWS): 100 GB minimum
- A YugaByte license file (attached to your welcome email from YugaByte Support)
- Ability to connect from the YugaWare host to all the YugaByte DB data nodes. If this is not setup, [setup passwordless ssh](#).

What are the OS requirements and permissions to run the YugaByte DB data nodes?

Prerequisites for the YugaByte DB data nodes are listed [here](#).

How are the build artifacts packaged and stored for YugaByte DB Enterprise?

The Admin Console software is packaged as a set of docker container images hosted on [Quay.io](#) container registry and managed by [Replicated](#) management tool. Installation of the admin console starts with installing Replicated on a Linux host. Replicated installs the [docker-engine](#), the Docker container runtime, and then pulls its own container images the Replicated.com container registry. YugaWare then becomes a managed application of Replicated, which starts by pulling the YugaWare container images from Quay.io for the very first time. Replicated ensures that YugaWare remains highly available as well as allows for instant upgrades by simply pulling the incremental container images associated with a newer YugaWare release. Note that if the host running the admin console does not have Internet connectivity, then a fully airgapped installation option is also available.

The data node software is packaged into the YugaWare application. YugaWare distributes and installs the data node software on the hosts identified to run the data nodes. Since it's already packaged into existing artifacts, the data node does not require any Internet connectivity.

How does the Admin Console interact with the YugaByte DB data nodes?

The YugaWare Admin Console does a password-less ssh to interact with the data nodes. It needs to have the access key file (like a PEM file) uploaded into it via the UI. The setup on each of the data nodes to configure password-less ssh is documented [here](#).

A REST API is also exposed by the admin console to the end users in addition to the UI as another means of interacting with the data platform.

Would we have access to the database machines that get spawned in public clouds?

Yes, you would have access to all machines spawned. The machines are spawned by YugaWare. YugaWare runs on your machine in your AWS region/datacenter. If you have configured YugaWare to work with any public cloud like AWS or GCP, it will spawn YugaByte nodes using your credentials on your behalf. These machines run in your account, but are created and managed by YugaWare on your behalf. You can log on to these machines anytime, and YugaWare will additionally show you some stats graphed into a built in dashboard either per node or per universe.

How many machines would I need to try out YugaByte DB against my load?

You would need:

- One machine to install YugaWare on
- Minimum as many data nodes as the replication factor. So just one machine for replication factor 1, and 3 machines in case of rf=3
- A machine to run the load tests on

Typically you can saturate a database machine (or three in case of replication factor 3) with just one large enough test machine running a synthetic load tester that has a light usage pattern. YugaByte ships some synthetic load-testers with the product which can simulate a few different workloads. For example, one load tester simulates a timeseries/IoT style workload and another does stock-ticker like workload. But if you have a load tester that emulates your planned usage pattern, nothing like it!

Can we control the properties (such as VPC, IOPS, tenancy etc.) of the machines YugaWare is spinning up?

Yes, you can control what YugaWare is spinning up. For example:

- You can choose if YugaWare should spawn a new VPC with peering to the VPC on which application servers are running (to isolate the database machines into a separate VPC) AWS, or ask it to re-use an existing VPC.
- You to choose dedicated IOPs EBS drives on AWS and specify the number of dedicated IOPS you need.

In general, we are be able to fill the gaps quickly if we are missing some features. But as a catch all, YugaWare allows creating these machines out of band and import these as "on-premise" install.

[FAQs](#)



[Architecture](#)

[FAQs](#)



Apache Cassandra Compatibility in YCQL

Give Feedback

Was this page helpful?