

Distributed System Deployment

This guide contains instructions on how to install VoipNow on a distributed environment (more than one computer) or to upgrade the software deployed on multiple computers.

- [Understand our cloud](#)
 - [Infrastructure](#)
 - [The software](#)
- [Requirements](#)
 - [Infrastructure](#)
 - [Operating system](#)
- [What's next?](#)

Fast Track

Unfortunately, there is no Fast Track. Deploying and managing a distributed infrastructure requires highly advanced knowledge. In this section, we only cover the particularities of the VoipNow distributed infrastructure. We do not cover the fundamentals of service distribution.

If you are not convinced about the distributed deployment, check the [single node installation](#). It is still flexible enough and able to support thousands of users.

Understand our cloud

We have already detailed why you would want to use a [VoipNow system installed on multiple computers](#).

To keep things short, it looks like you need massive scalability that goes beyond the possibilities of a single hardware machine and you still want to have a single system to manage, not multiple autonomous systems.

Before going further with a distributed setup, it is necessary to understand the key elements of our cloud technology.

Infrastructure

The Infrastructure represents the physical layer that supports the Cloud and VoipNow. The Infrastructure is made of bare-metal servers, switches, routers, storage systems, etc. It can also be provided by an IaaS service such as the Amazon EC2.

Without infrastructure, it is impossible to run anything, let alone VoipNow.

Low level

The current document is not meant to describe how a reliable infrastructure should be built. This is not its purpose. We will only be looking at the container, as the computing entity that runs the software. Since this is a virtual machine, it is highly abstracted and standardized, so we do not care what lies behind it. Basically, it is the same philosophy you adopt when buying many IaaS services.

Container

The Container represents a virtual computing instance, with its own computing power, memory, storage, network addresses and operating system. The Containers are a logical unit, provided by the virtualization software.

Containers are:

- **Flexible:** they can be started and stopped and resources can be changed (even without reboot), etc.
- **Part of the infrastructure:** In many cases, they are the only part, which is exposed.
- **Given at least a network address** (only private, only public, or both types)

Do not confuse our container terminology with specific virtualization techniques that call their virtual machines "containers". Our containers are abstract entities that can be deployed using any virtualization technology.

The software

Node

VoipNow always runs on what we call Service Nodes, or simply Nodes. The Node is part of the software and its scope is to deliver some of the functions of the software. In short, **a VoipNow installation on a single computer has a single node that does everything**.

Do not confuse the Node with the Container!

The Node is a Software entity, while the Container is an Infrastructure entity. The Node requires a Container in order to run, but it does not require a specific container. It might change during the software lifetime operation and the software will not give error messages.

Role

VoipNow has a lot of functions. It is a complex software that covers many flows. Nodes are specialized. Some handle database functionality, some handle the web management interface, while others cover signaling for telephony functions, etc. We call such a specialization **Role**.

So, when you have to manage nodes, you will see that each node is assigned a specific role.

Sounds too complicated?

Think of the Infrastructure and Software like a courier company. Each company has many employees delivering stuff. Some are specialized in delivering to customers, others are delivering between hubs, contractors, etc.

The employee is responsible with delivering the packages. Usually one drives a car, but the car might be shared between shifts, the car might break down, etc. It is just a method to fulfill the goal, it can be replaced, taken out of service, etc.

Now that you know more, we want to introduce the new terminology - **it is no longer the distributed system that runs on multiple computers, but the distributed system that runs on multiple nodes.**

Requirements

This is what you need in order to run VoipNow in a distributed environment.

Infrastructure

It does not matter if you run your own infrastructure or use a public cloud service. The following conditions must be met:

- You must be able to provision virtual machines easily.
- The hardware specifications of the virtual machines must be flexible enough because, based on the role of the node, they might require more CPU power, memory or I/O.
- Both public (routable) and private (non-routable) network are required. It is in your best interest to keep some traffic private and to expose only the services your customers will use.

As virtualization is necessary, it does not really matter what hardware is used as long as it is supported by the virtualization technology vendor. Naturally, if you use a cloud service, hardware should be the last of your worries.

The public network can be accessible using a carrier-grade NAT layer.

Operating system

When running distributed, VoipNow is compatible with the Linux distributions:

- Red Hat Enterprise Linux 7.x (x86_64)
- Red Hat Enterprise Linux 8 (x86_64)
- CentOS 7.x (x86_64)
- CentOS 8 (x86_64)

What's next?

There are many things you must become familiar with before going further with the deployment. Take it step by step, as follows:

- [Software Roles](#): Learn more general things about Roles in VoipNow.
- [Roles Requirements](#): Review the requirements of each role and what you must do in order to implement each of them.
- [Storage](#): Learn how you should implement storage.
- [Start the Installation](#): Learn how to perform the installation of a distributed environment.