

# VoipNow Roles Requirements

This document describes the VoipNow roles and what you should know about them in order to provision them properly.

- [Networking requirements](#)
  - [Public and private](#)
  - [Addressing requirements](#)
- [High Availability](#)
  - [Recommendations](#)
- [Hardware requirements](#)
  - [Role based resource consumption](#)
  - [How many nodes?](#)
  - [When to add new nodes?](#)

## Fast Track

In this section, we will detail each role in order to determine provisioning constraints such as:

- [Networking Requirements](#)
- [High Availability](#)
- [Hardware Requirements](#)

You are free to go solely through the tables if you are confident about your infrastructure. At the same time, we understand that some things might get complicated, that's why we provide infrastructure design consulting services.

## Networking requirements

### Public and private

While it is required to design your infrastructure around private and public networks, it is important to remember that roles have different requirements. When you assign a role on a node, such requirements should be considered for the container that runs the node:

Role	Private Network *	Public Network	Notes
SQL	Required	Not required	We strongly discourage exposing SQL nodes in a public network.
Distributed Database	Required	Not required	We strongly discourage exposing Distributed Database nodes in a public network. If there are more than 3 extensions on the server, the Distributed Database role <b>cannot be assigned</b> .
Elasticsearch	Required	Not required	We strongly discourage exposing Elasticsearch nodes in a public network.
Web Management Interface	Required	Required	Technically could run on a private IP as well, but a HTTP load balancer (that obviously runs on a public IP) is required.
SIP	Required	Required	Private IP required for management purposes.
PBX	Required	Required	Private IP required for management purposes.
Infrastructure Controller	Required	Optional	The public IP is necessary only if it's impossible to access the infrastructure controller management interface through private network.
Worker	Required	Not required	We strongly discourage exposing Worker nodes exposed in a public network.

\* The private network is also required for administrative tasks by all roles.

### Addressing requirements

We have the following recommendations:

- Use an addressing scheme even in the private network. By using DNS, when an IP renumbering occurs in the infrastructure, you do not have to reconfigure roles.
- Carefully plan the private network to allow expansion. If you do it properly on the network level, the private space is pretty much unlimited.
- Always use an addressing scheme in the public network. Ask your customers to connect to a hostname (yourservice.com), not directly to the IP address.

With a cloud IaaS service, you don't have to worry about these, as all of them support public and private networks, automatic addressing, and virtually unlimited resources.

## High Availability

Achieving high-availability is usually expensive. [As already explained](#), distribution does not mean high-availability, yet it will improve the overall system resiliency on certain class of events.

For the sake of the argument, let us assume that the software system is a car. If the AC breaks down, the car will still be able to run, maybe causing some inconvenience to its passengers (depending on the weather). The same car might be able to run at a very low speed even with a punctured tire, although it is obvious that such a damage is more important. There are also critical damages. For example, when the engine breaks down, the car will not be able to run anymore.

The same happens in our environment. It might be able to work with some damages, but clearly functionality is impacted.

In this chapter we determine the high-availability requirements based on the role type.

The table below shows what happens if nodes with a certain role go down due to any hardware fault.

Role	Outcome	Recommended HA strategy
SQL	System is not affected if one of the slave nodes is lost, but system functionality is totally lost when master MySQL node goes down.	<b>OS LEVEL</b> Deploy a MySQL cluster. Protect MySQL master node with virtualization layer HA technologies.
Distributed Database	System is not affected as long as quorum exists..	<b>APP LEVEL</b> Deploy a HubRing cluster.
Elasticsearch	System is not affected as long as quorum exists..	<b>APP LEVEL</b> Deploy an Elasticsearch cluster.
Web Management Interface	System is not affected as long as a health based balancer is configured to distribute requests to web management interface nodes.	<b>APP LEVEL</b> Use a redundant load balancer in front of web management interface nodes.
SIP	A group of customers is affected, phone functionality is lost, current calls are not be dropped.	<b>OS LEVEL</b> Protect SIP nodes using virtualization layer HA technologies.
PBX	A group of customers is affected, phone calls are dropped, no loss of functionality after.	<b>OS LEVEL</b> Protect PBX nodes using virtualization layer HA technologies.
Infrastructure Controller	Infrastructure provisioning affected.	<b>OS LEVEL</b> Protect Infrastructure Controller node using virtualization layer HA technologies.
Worker	System is not affected as long as a sufficient number of worker nodes survive the event in order to be able to process requests.	<b>APP LEVEL</b> Deploy a sufficient number of Worker nodes.

## Recommendations

Based on the above conclusions, it becomes obvious that distribution does not come with High Availability guarantees.

### Application level HA **APP LEVEL**

Some VoipNow roles are designed to work in fault tolerant clusters. This is an ideal situation because no other high availability technology is required. This provides application level high availability.

### Virtualization layer HA **OS LEVEL**

Unfortunately not all software is designed with high availability in mind. For instance, MySQL is not resilient to failures. As you can see in the table above, when the master MySQL nodes dies, the system goes **totally down**. There are promising projects that deliver multi master MySQL replication and VoipNow plays friendly with them, but for the moment we recommend to protect MySQL master node from failures using high availability technologies on the virtualization layer.

The Virtualization layer HA comes with a price - more overhead, more hardware resources, sometimes even extra licensing costs. But the nice thing is that from the hardware perspective, with virtualization layer HA *you do not need 2N resources*.

It is not always necessary to deploy virtualization layer HA. For example if a SIP or PBX node goes down, only a group of customers is affected. If you can live with this, then it's not necessary to deploy it, but you can use alternate strategies (for example to automatically deploy a replacement node).

## Cloud services

Cloud services with built-in HA are also typically more expensive. Always check for the service provider availability guarantees before deploying VoipNow in the cloud.

## Hardware requirements

This chapter covers the system sizing based on usage. Even when dealing with a distributed system, it is necessary to understand:

- the recommended hardware specifications based on the role of the node that runs on the container
- how many nodes should be deployed on each role

We are not making any specific hardware recommendations because hardware selection and sizing must be based on monitoring and usage information. Instead, we are explaining what to expect from each role.

## Role based resource consumption

This shows what type of resources nodes on various roles consume.

Role	CPU	Memory	I/O	Example: Amazon EC2 Instance Type *
<b>SQL</b>	High	Very High	Very High	High I/O Quadruple Extra Large Instance
<b>Distributed Database</b>	Moderate	High	Moderate	Large Instance
<b>Elasticsearch</b>	Moderate	High	Moderate	Large Instance
<b>Web Management Interface</b>	High	Low	Low	High-CPU Extra Large Instance
<b>SIP</b>	High	Moderate	Low	High-CPU Extra Large Instance
<b>PBX</b>	High	Moderate	Moderate	High-CPU Extra Large Instance
<b>Infrastructure Controller</b>	Low	Low	Low	Small Instance
<b>Worker</b>	High	Moderate	Low	High-CPU Extra Large Instance

As you can see, there are different requirements on each role. The nice part about it is that you are flexible no matter if:

- You deploy your own infrastructure - by using virtualization, you can distribute resources for the virtual machines running the nodes
- You use a cloud service - service providers have different instance profiles based on what you want to achieve

\* The Amazon EC2 example provided above is purely for comparison purposes and it is appropriate for a high performance infrastructure.

## How many nodes?

The system is flexible - the more users you get, the more nodes you can add.

Role	Node Type	Min /Max Nodes	Comments
<b>SQL</b>	Master	1/1	A single MySQL node can sustain a large infrastructure. We support sharding to address the cases when a single master MySQL node becomes a limitation.
<b>SQL</b>	Slave	0/4	Slaves offload some queries from the master. It's not necessary to deploy slaves, but if you do, keep their number to a maximum of four.
<b>Distributed Database</b>	-	1/128	The number of distributed database nodes must not be changed for the system lifetime, but this is not a problem because, for the start, you can use virtual machines or instances with very limited resources.
<b>Elasticsearch</b>	-	1/Cluster	It is recommended to deploy an Elasticsearch cluster.
<b>Web Management Interface</b>	-	1/No limit	You can add nodes dynamically, based on the web interface utilization.
<b>SIP</b>	-	1/No limit	The SIP role uses dynamic sharding. This means that customers are assigned automatically to one of the existing SIP roles. That's why you cannot remove SIP nodes after these are provisioned. You can start with one and add more as capacity increases demand it.
<b>PBX</b>	-	1/No limit	The PBX nodes are dynamically chosen by VoipNow. This means that you can remove PBX nodes that are not used. You can start with one and add more capacity when telephony utilization information shows this.
<b>Infrastructure Controller</b>	-	1/3	A single node running the infrastructure controller is necessary, but it can be protected with Fault Tolerance technologies.
<b>Worker</b>	-	1/No limit	You can add nodes dynamically, based on the worker layer utilization.

## When to add new nodes?

The answer to this question can be provided by the same data which also allows you to upgrade the hardware of the existing nodes.

All nodes must be monitored in order to have timely and insightful information available that allows to take infrastructure scaling decisions.