**Introduction**

**HyperCX Overview**

REDrackX HyperCX Bento is a hyper-converged Cloud platform designed to provide Infrastructure as a Service (IaaS). REDrackX started as a Private Cloud provider, but nowadays, it also provides Public Cloud based on the same technology leveraged on HyperCX Private Clusters. Users can request REDrackX for private clouds, or register and start using resources from REDrackX own managed public cloud. Besides this, owners of a Private HyperCX Cluster can convert them into a Public Cluster without intervention from REDrackX.

**Note**

REDrackX HyperCX Bento is the newest HyperCX architecture, REDrackX no longer maintains the original HyperCX stack. In this documentation, the term HyperCX alone will always refer to HyperCX Bento.

Being a hyper-converged platform, a REDrackX HyperCX cluster integrates the following components:

· Network Subsystem: Provides connectivity to all cluster components and some workloads. Constitutes physical components as well as virtual components. These components can vary depending on the specific deployment. When REDrackX deploys in one of its data centers, REDrackX own managed switches will be used, but this is optional when deploying on-prem for a user. In the case of the firewall, something similar happens. By default, REDrackX recommends using its own virtual routers. These devices will provide connectivity to all the critical cluster components and the default virtual network that is created using private IPs. These virtual routers can be fully replaced by physical firewalls if desired, or use together to physical firewalls.

· Management Subsystem: HyperCX leverages an extended monitoring system tightly integrated to all its components. This system will send monitoring data to REDrackX central management servers, allowing REDrackX to monitor the health of every cluster constantly. Besides this, a local version of this monitoring system will be installed locally only on Bento clusters, accessible to the cluster's administrator to check advanced metrics not available through the orchestrator. This system offers:

* + Proactive and reactive monitoring.
  + Failure, accounting and performance monitoring.
  + Active and passive probes.
  + Optional email notifications.
* Storage Subsystem: A Software Defined Storage (SDS) solution deployed as a Storage Area Network (SAN) is used with the following characteristics:
  + Bare metal installation.
  + Replicated data.
  + Optimized performance for VMs and containers.
* Orchestration Subsystems: This subsystem provisions and manages resources on the cluster. It also handles more advanced features like High Availability.
* Compute Subsystems: HyperCX being a hyper-converged platform, shares the same cluster between workloads and the rest of the subsystems (storage, management, compute and orchestration). The compute subsystem offers KVM Virtual Machines and LXD containers. KVM Virtual Machines are mandatory and will be available on every HyperCX cluster. LXD containers are optional.

So far, we have mentioned HyperCX Bento and miniBento. HyperCX Bento is REDrackX flagship architecture designed to handle big and critical deployments. It is fully redundant, and the smallest configuration starts on three nodes. There are use cases where the smallest deployments are needed, like Edge Computing and development applications. HyperCX miniBento is more suited for these cases since it can be built on top of a single node (or more). To make it as lightweight as possible, High Availability is not supported on miniBento deployments. The local monitoring system is only available on HyperCX Bento configuration, although both Bento and miniBento will be remotely monitored. The last major difference is that HyperCX Bento is designed to be scaled out up to hundreds of nodes, but this is not the case for miniBento. Notice then, in this case, the limiting factor is not the technology, but the lack of redundancy. Since HyperCX stores all the workloads on a SAN, and there is no redundancy on miniBento if any server or any drive fails, most of the information can be lost or corrupted, so it is not feasible to scale-out miniBento instances since the more hardware running the more chances of failure. The following table summarizes the differences between both:

| **Feature** | **Bento** | **miniBento** |
| --- | --- | --- |
| HA | Yes | No |
| Minimum amount of nodes | 3 | 1 |
| Integrated Monitoring System | Yes | No |
| Remotely monitored | Yes | Yes |
| Scalable design | Yes | Partially |
|  |  |  |

For any Cloud Deployment model offered by REDrackX, the hardware can be installed on-prem (inside the user's facilities) and off-prem (hosted in one of REDrackX data centers). Each approach has advantages and disadvantages, some of those are summarized next:

| **On-prem** | **Off-prem** |
| --- | --- |
| Hardened security | Support |
| Low latency | Reliability |
| Less Internet BW required | Faster deployments |
|  |  |

Regarding the installation mode, two options are available: Full Stack and Software Only. The only difference is related to the hardware provider. A HyperCX cluster delivered by REDrackX, including the hardware, is a full-stack configuration. On the other hand, the user can provide the hardware, and REDrackX will build the cluster. It is called a software-only configuration. Full-stack configurations can be deployed inside REDrackX data centers (off-prem) or be shipped to different locations (on-prem). HyperCX software only clusters, on the other hand, can only be built off-prem, inside the user's facilities.

The following documentation is designed for HyperCX users. Most of this documentation, except for the **Public Cloud** section, is meant for administrators of a Private Cluster. This configuration exposes all of REDrackX HyperCX features, which are considerably more, hence why there is a single section for REDrackX Public Clouds.

If the goal is to quickly start spinning up virtual instances, you should go to the [Quick Start Guide](https://docs.virtalus.com/QuickStart/QuickStart)

# Quick Start guide

## Overview

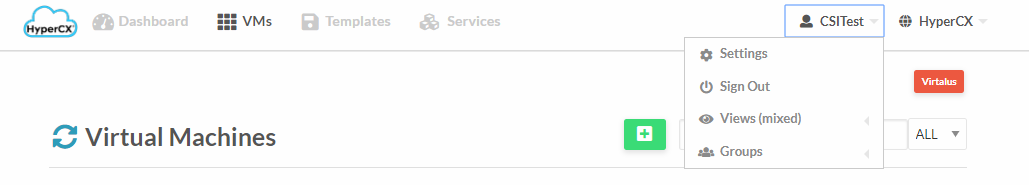
HyperCX can be set up with just a click of a button, and it’s that easy. Everything else has already been taken care of, although for full control over a Virtual Machine (VM), it’s images, and it’s context, we recommend understanding VM templates and images. But if you need a VM up and running with few or no modifications, you can follow this guide without worrying about that part.

**Note**

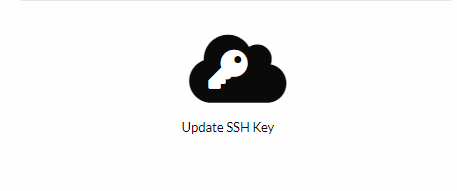
Some HyperCX clusters are only accessible from inside a VPN. If this is your case, make sure you are connected using the provided VPN configuration file. For detailed steps on how to connect to VPN [learn more](https://docs.virtalus.com/Misc/Others/#connect-to-openvpn-windows)

## Getting Started

1. (Optional) With HyperCX users can utilize password less SSH (Only for Linux), for windows a password must be provided when launching the VM. If you intend to launch a Windows VM you can skip this part. All a user needs to do is navigate to settings



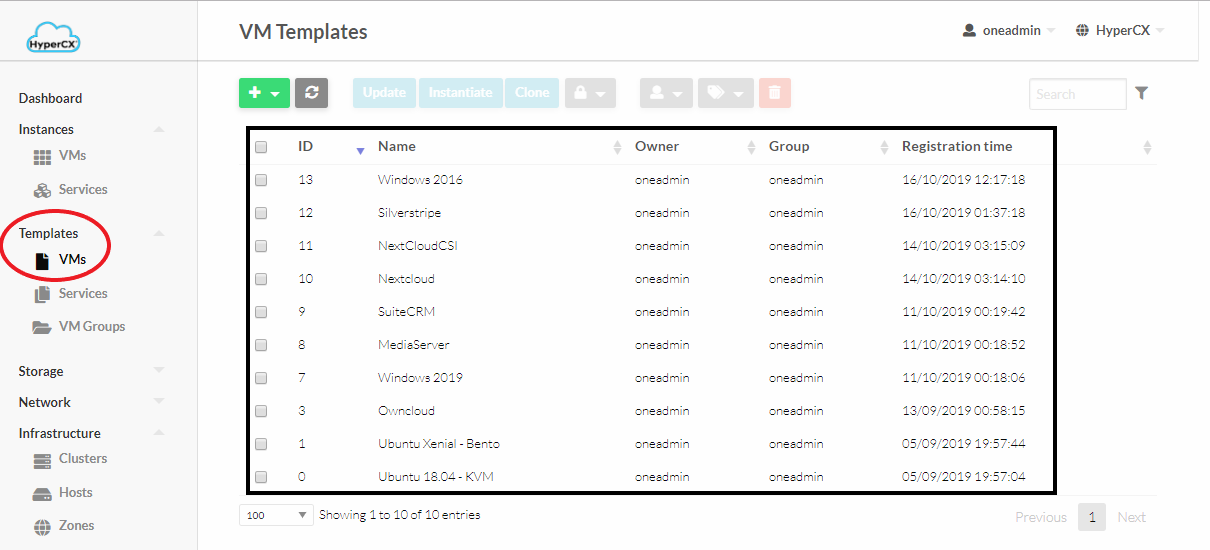
1. Then navigate to update SSH key



Add their public SSH key then click on update SSH key. From now on, this public key will be configured inside the **root** user for each new Linux VM or container deployed.

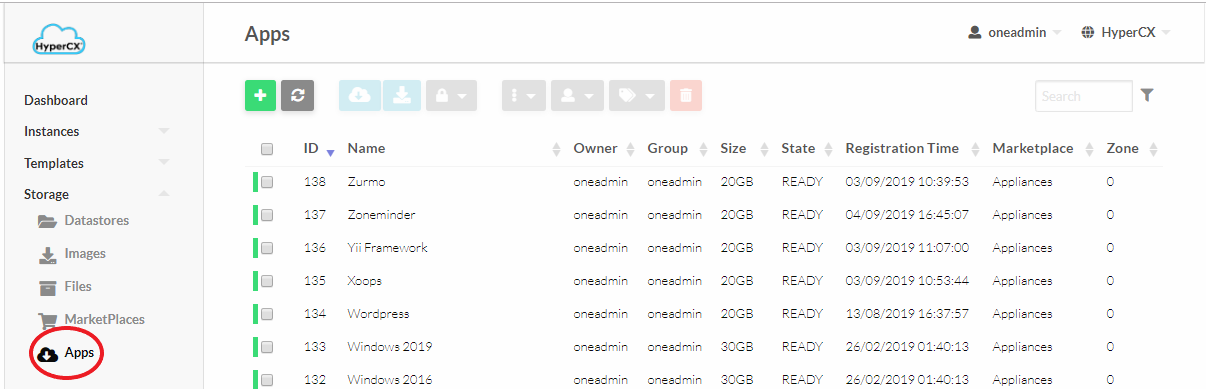
### Downloading a template to HyperCX

VMs are created from templates. If this is a new cluster, chances are the needed template will not be available (although some clusters come with some pre-downloaded templates, like in the case of Proof of Concept (PoC)). You can check if you’re template is already available by heading over to templates->VMs

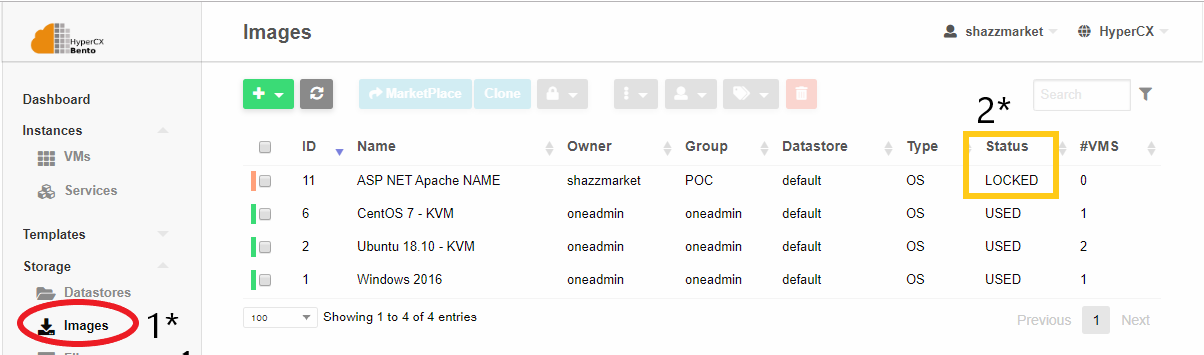


If your desired template is available you can head over to the next subchapter. If not, you will need to download the required template or appliance from one of REDrackX marketplaces.

1. Appliances can be accessed under Storage 🡪 Apps.



1. Once in the Apps UI select an existing appliance and click on the appliance to view its description.
2. Now click on https://docs.virtalus.com/images/QS6.png icon to start the download template creation process.. This process will download the OS image and create a new template.
3. Enter the template name, its Image name and click onhttps://docs.virtalus.com/images/QS7.png.
4. Wait for the Image to be downloaded. It’s status can be checked by accessing Storage -> Images



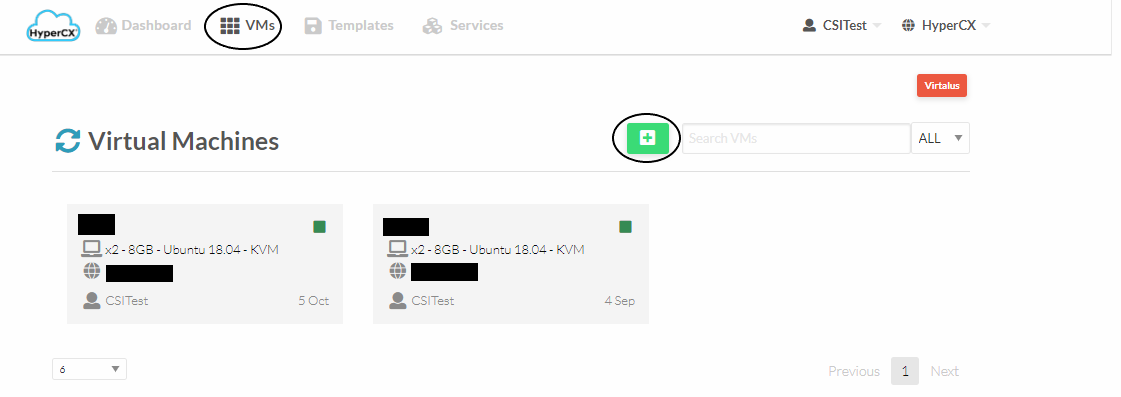
The status of the image will be LOCKED while the download is in progress. Once the image is downloaded its status is changed from LOCKED to READY, the new template will be ready to be instantiated.

[Click here](https://docs.virtalus.com/UserGuides/TemplateManagement/) to learn more about templates and images.

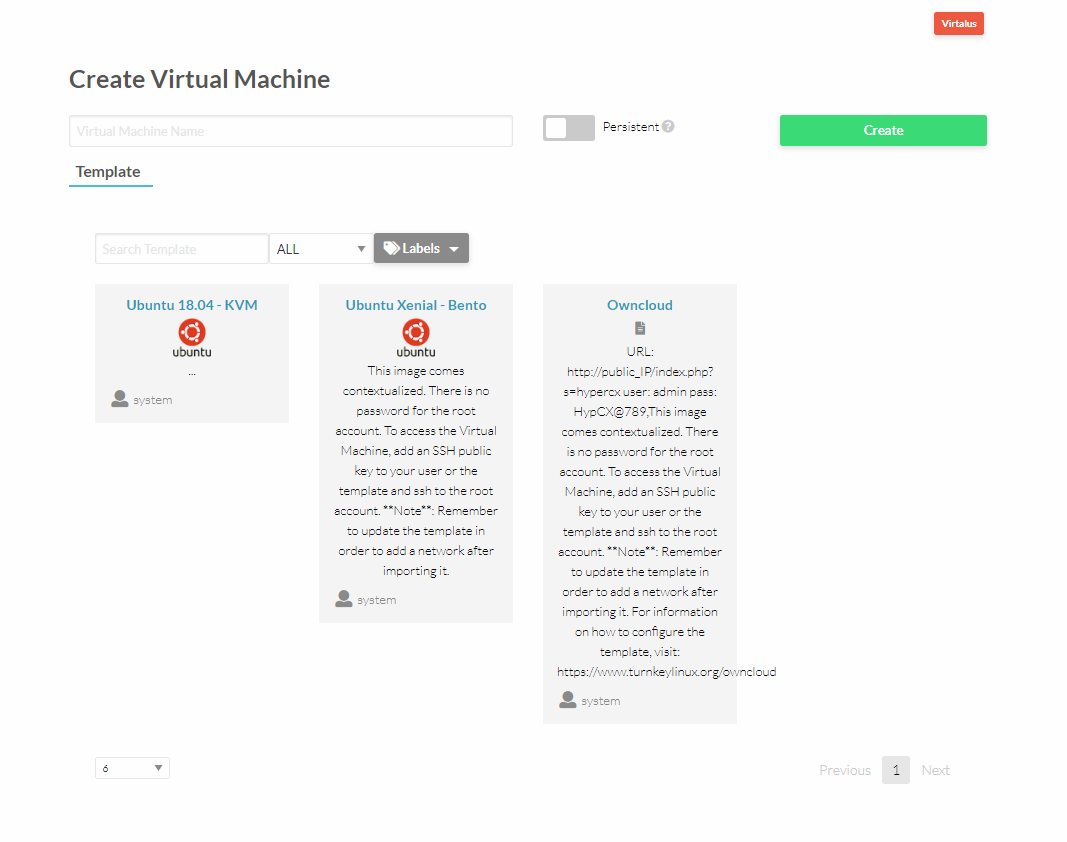
### Instantiating a VM

Instantiating a VM can be easily done by,

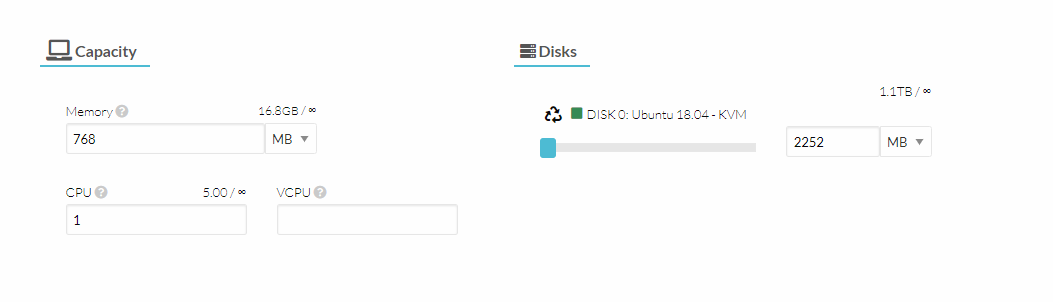
* Navigate to VMs and then clicking on the add button.



* Now select a predefined template that you want to use, it can be an OS or prebuilt image with everything preinstalled. Now enter a name for the virtual machine(This is optional and can be left blank)

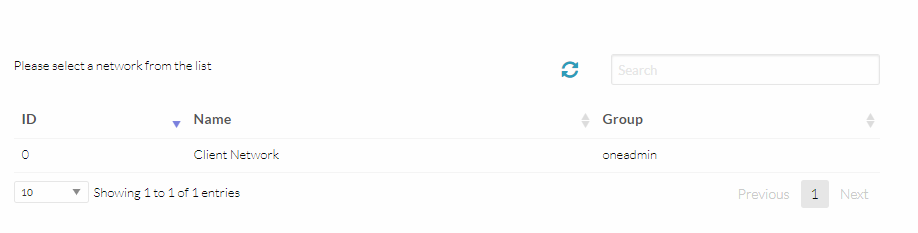


* Once the VM template is selected just specify capacity of the VM.



* Now carry on to the network part and attach the required network interface.





* Now scroll back up and click on the create button that’s all required to instantiate a VM.

### Accessing the VM

Determine the IP address of the VM by navigating to VMs and then under your newly created VM IP address will be mentioned.

#### Linux VM

Linux VMs can be accessed using password-less ssh (if the public key was previously defined). If a password were specified, this would be the root password, but ssh access using the root user and a password is not allowed for security reasons. If this is your case, you must use the VNC screen and either remove this restriction from the VM's SSH server or create a new user (recommended).

#### Windows VM

Remotely access the VM by using Remote Desktop Protocol (RDP) using the IP address determined earlier.

#### VNC

Users can also get a direct display of the VM using VNC by navigating to VMs and then clicking on the respective VM, now click on https://docs.virtalus.com/images/QS14.png

For more detailed information about how to manage VMs and it’s templates follow our documentation.

## FAQ

* Q: I don’t know the password of my windows or Linux machine`

A: Your virtual instance (VM o container) will only contain a password if this was specified when instantiating the instance. If not, there is no password associated with the instance. Windows VMs require a password; if not, they will not be able to access them later on via VNC or RDP. For Linux VMs, a password is optional, although public keys are recommended. If a password was set, it would be found on the VM configuration parameters.