Xen Orchestra Installer / Updater

VM image build date 2024-06-29 VM image build date 2024-06-29 VM image build date 2024-06-29 VM image build date 2024-06-29

VM image build date 2024-06-29

lint passing lint passing

Script to install/update Xen Orchestra and all of it's dependencies on multiple different Linux distributions. Separate script to be used on XenServer/XCP-ng host that installs a readymade VM image that has Xen Orchestra installed utilizing the same installer script.

How about docker? No worries, check Docker hub

What is Xen Orchestra?

Xen Orchestra is a web interface used to manage XenServer/XCP-ng hosts and pools. It runs separately and one can manage multiple different virtualization environments from one single management interface.

Xen Orchestra is developed by company called Vates. They offer Xen Orchestra as a turnkey appliance with different pricing models for different needs and even a free version with limited capabilities. This is the preferred and only supported method of using Xen Orchestra product as the appliance goes through QA and each of the plans come with support. I highly recommend using the official appliance if you plan on using Xen Orchestra in production environment, to support a great product and it's development now, and in the future.

Why to use this script?

If you're a home user/enthusiast with simple environment you want to manage but can't justify the cost of Xen Orchestra appliance and don't need the support for it.

Since Xen Orchestra is open source and majority of the paid features included in the official appliance are part of the sources, one can build it themself. This procedure is even documented. Note that even though this method is documented, it's not supported way of using Xen Orchestra and is intended to be used only for testing purposes and not in production.

This script offers an easy way to install all dependencies, fetch source code, compile it and do all the little details for you which you'd have to do manually otherwise. Other than that, it follows the steps described in the official documentation. All source code for Xen Orchestra is by default pulled from the official repository.

This script is not supported or endorsed by Xen Orchestra. Any issue you may have, please report it first to this repository.

The very first version of this script i did purely for myself. Now i'm mainly trying to keep it up to date for others who might already rely on it frequently. My intentions are to offer an easy way for people to get into Xen Orchestra without restricted features which could potentially help this piece of software to evolve and grow.

Preparations

First thing you need is a VM (or even a physical machine if you wish) where to install the software. This should have at least 4GB of RAM and ~1GB of free disk space. Having more CPU does speed a the build procedure a bit but isn't really a requirement. 2vCPU's on most systems are more than fine.

Supported Linux distributions and versions:

- CentOS 9 Stream
- CentOS 8 Stream
- AlmaLinux 9
- AlmaLinux 8
- Rocky Linux 9
- Rocky Linux 8
- Debian 12
- Debian 11
- Debian 10
- Ubuntu 22.04
- Ubuntu 20.04

Only x86_64 architecture is supported. For all those raspberry pi users out there, check container instead.

All OS/Architecture checks can be disabled in `xo-install.cfg` for experimental purposes. Not recommended obviously.

I suggest using a fresh OS installation, let script install all necessary dependencies and dedicate the VM for running Xen Orchestra.

If you plan on using the prebuilt VM image for XenServer/XCP-ng, see the image section below.

Installation

Start by cloning this repository to the machine you wish to install to.

See Wiki for common configuration options

There is a file called `sample.xo-install.cfg` which you should copy as `xo-install.cfg`. This file holds some editable configuration settings you might want to change depending on your needs.

When done editing configuration, just run the script with root privileges:

```
sudo ./xo-install.sh
```

There are few options you can choose from:

- `Install`

install all dependencies, necessary configuration and xen orchestra itself * `Update`

update existing installation to the newest version available * `Rollback`

should be self explanatory. if you wish to rollback to another installation after doing update or whatever

- `Install proxy`

install all dependencies, necessary configuration and xen orchestra backup proxy

- `Update proxy`

update existing proxy installation to newest version available

Each of these options can be run non interactively like so:

```
sudo ./xo-install.sh --install [--proxy]
sudo ./xo-install.sh --update [--proxy] [--force]
sudo ./xo-install.sh --rollback
```

As mentioned before, Xen Orchestra has some external dependencies from different operating system packages. All listed below will be installed if missing:

```
rpm:
- curl
- epel-release
- nodejs (v14)
- npm (v3)
- yarn
- gcc
- gcc+
- make
- openssl-devel
- redis
- libpng-devel
- python3
- git
- nfs-utils
- libvhdi-tools
- cifs-utils
- lvm2
- ntfs-3g
- dmidecode
- sudo (if set in xo-install.cfg)

deb:
- apt-transport-https
- ca-certificates
- libcap2-bin
- curl
- yarn
- nodejs (v14)
- npm (v3)
- build-essential
- redis-server
- libpng-dev
- git
- python3-minimal
- libvhdi-utils
- lvm2
- nfs-common
- cifs-utils
- gnupg (debian 10/11)
- software-properties-common (ubuntu)
- ntfs-3g
- dmidecode
- sudo (if set in xo-install.cfg)
```

Following repositories will be installed if needed and repository install is enabled in xo-install.cfg

```
rpm:
- forensics repository
- epel repository
- nodesource repository
- yarn repository

deb:
- universe repository (ubuntu)
- nodesource repository
- yarn repository
```

Backup proxy **Proxy installation method is experimental, use at your own risk. Proxy installation from sources is not documented by Xen Orchestra team. Method used here is the outcome of trial and error.**

Proxy source code will be edited slightly to disable license check which only works with official XOA and there is no documented or working procedure to bypass it properly (there used to be but not anymore)

Backup proxy can be used to offload backup tasks from the main Xen Orchestra instance to a proxy which has a direct connection to remote where backups are stored.

Requirements for proxy VM are otherwise the same as mentioned above, in addition the VM needs to live inside XCP-ng/XenServer pool managed by Xen Orchestra instance and have xen tools installed. VM needs to have access to pool master host and Xen Orchestra needs to be able to access this VM via TCP/443.

Majority of xo-install.cfg variables have no effect to proxy installation. Proxy process will always run as root user and in port 443.

Since there is no way in Xen Orchestra from sources to register a proxy via UI, the installation will output a piece of json after the proxy is installed. You need to copy this json string and save to a file. Then use the config import option in Xen Orchestra settings to import this piece of json to add proxy. This works as a partial config import and won't overwrite any existing config. Although it's good to take a config export backup just in case.

Xen Orchestra figures out the correct connection address from the VM UUID which is part of this json. This is why the VM needs to be part of pool managed by Xen Orchestra. Connection details cannot be changed manually.

Note that for obvious reasons some of the proxy features seen in Xen Orchestra UI aren't working, like upgrade button, upgrade check, redeploy, update appliance settings.

Plugins Plugins are installed according to what is specified in `PLUGINS` variable inside `xo-install.cfg` configuration file. By default all available plugins that are part of xen orchestra repository are installed. This list can be narrowed down if needed and 3rd party plugins included.

Image

If you don't want to first install a VM and then use `xo-install.sh` script on it, you have the possibility to import VM image which has everything already setup. Use `xo-vm-import.sh` to do this, it'll download a prebuilt Debian 11 image which has Xen Orchestra and XenOrchestraInstallerUpdater installed.

Details of image build process here

Run on your Xenserver/XCP-ng host with root privileges:

```
sudo bash -c "$(curl -s https://raw.githubusercontent.com/ronivay/XenOrchestraInstallerUpdater/master/xo-vm-import.sh)"
```

Default username for UI is `admin@admin.net` with password `admin`

SSH is accessible with username `xo` with password `xopass`

Remember to change both passwords before putting the VM to actual use.

Xen Orchestra is installed to /opt/xo, it uses self-signed certificates from /opt/ssl which you can replace if you wish. Installation script is at /opt/XenOrchestraInstallerUpdater which you can use to update existing installation in the future.

xo-server runs as a systemd service.

xo user has full sudo access. Xen Orchestra updates etc should be ran with sudo.

This image is updated weekly. Latest build date and MD5/SHA256 checksum can be checked from here

Built and tested on XCP-ng 8.x

Tests and VM image

I run my own little implementation of automation consisting of ansible and virtual machines to test the installation on a regular basis with different operating systems. Test results are visible in badges on top of this readme.

VM image is also built totally by me and distributed from webservers i maintain.

Contributing

Pull requests and issues (either real issues or just suggestions) are more than welcome. Note that i do not wish to make any modifications to Xen Orchestra source code as part of this script.

Support

If you find this project useful and want to support the development by covering some of the hosting costs that come from maintaining an XCP-ng server in a data center, use paypal donation link below.

Donate PayPal