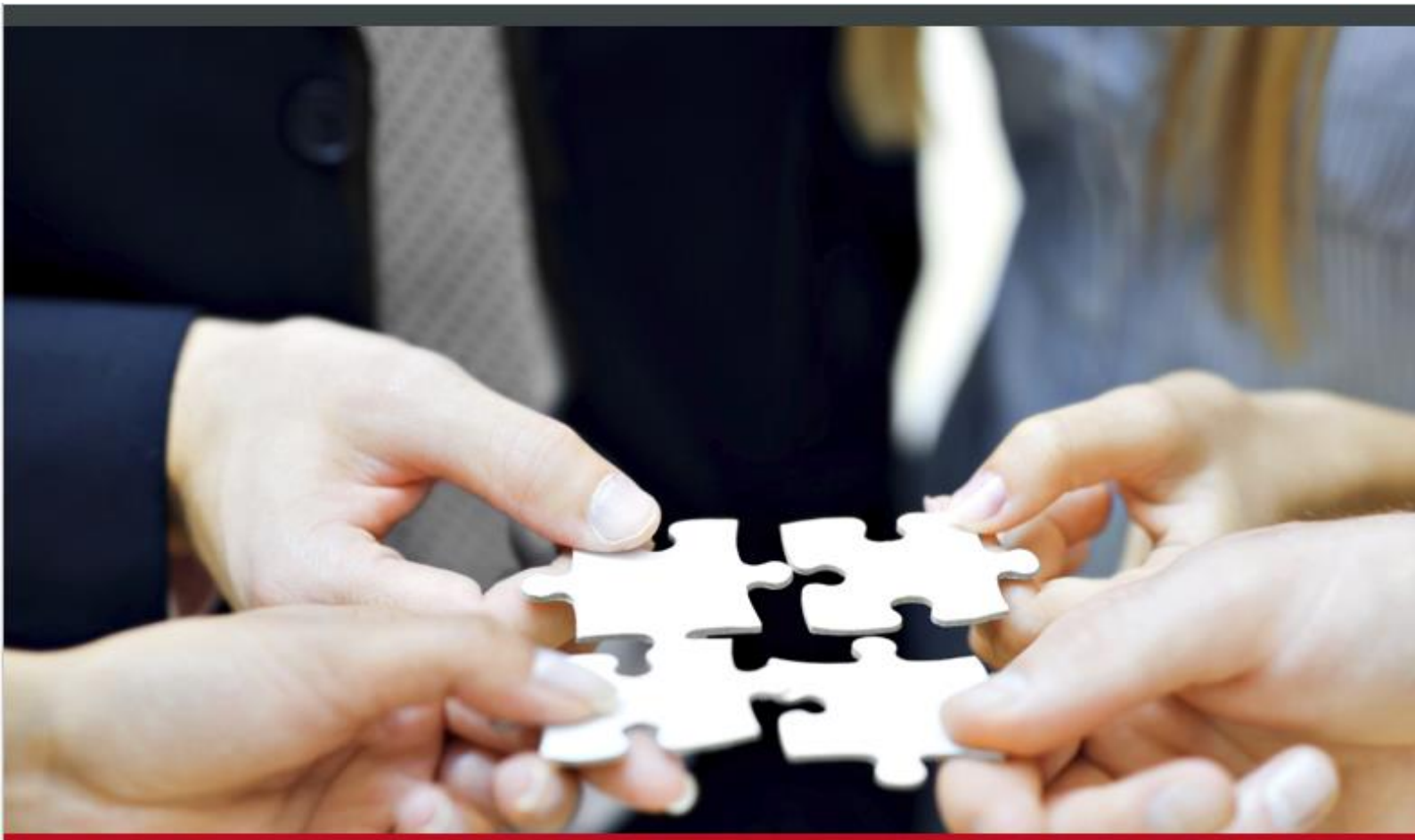
FRITZ & MACZIOL
software, systems and services



Coopto™ – VMware plug-in

User Guide

Author Robert Szymczak

Date 12. January 2015
Version 0.0.2-beta

Hörvelsinger Weg 17 – 89081 Ulm – Phone +49 731 1551-0 – Fax +49 731 1551-555 – www.fum.de – info@fum.de
FRITZ & MACZIOL Software und Computervertrieb GmbH

Table of contents

Table of figures

1 Licensing & Legal

Coopto – from now on "this project", "this program" or "this software" – is an open source project proudly presented by Fritz & Macziol Software und Computervertrieb GmbH.

This software may include *"Open Source Software"*, which means various open source software components licensed under the terms of applicable open source license agreements included in the materials relating to such software. Open Source Software is composed of individual software components, each of which has its own copyright and its own applicable license conditions. Information about the used Open Source Software and their licenses can be found in the *open_source_licenses.md* file. The Product may also include other components, which may contain additional open source software packages. One or more such *open_source_licenses.md* files may therefore accompany this Product.

2 Abstract

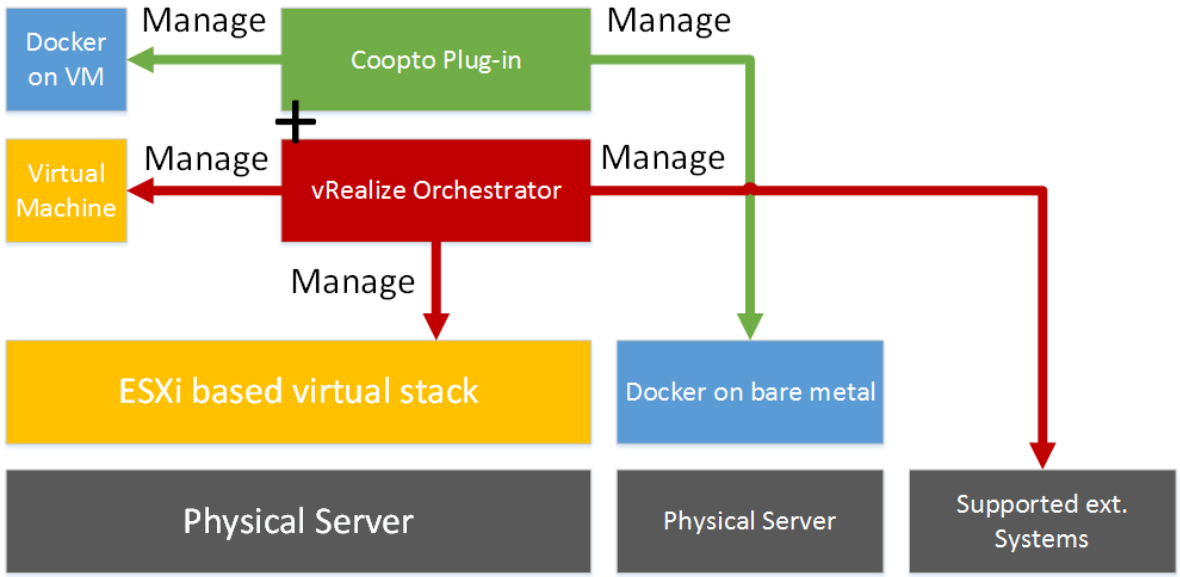
2.1 About Coopto

Coopto is a plug-in for VMware's orchestration engine vRealize Orchestrator. It aims to provide full Docker functionality within the central automation component of the VMware stack in order to utilize and combine the power of container technology with virtualization technology.

We think that containers and VM may not only coexist but greatly benefit from each other. Parts where virtualization alone so far has greatly failed due to the high variation of competing formats in a very dynamic market can be moved to the shiny parts or container technology, e.g. the independent and therefore shiftable format. Other aspects of cluster computation where virtualization has matured and proven enterprise ready within the last years can furthermore be implemented on the trusted and well known virtualization stack currently in use in most modern datacenter.

Digging in deeper the combination of both can result in an even more powerful computation stack then possible with just one of the technologies. A typical real world use-case would be providing a persistent storage to a container by leveraging existing vSphere storage APIs to create and attach a virtual HDD exclusively for that container, implicitly simplifying container management and backup for business critical data.

In order to bring both worlds together we use a simple to use and yet very common orchestration engine: vRealize Orchestrator. In fact: if you're running on a vSphere stack, you probably already own vRealize Orchestrator.



3 Downloading Coopto

There are two methods how you can get Coopto. The first and recommended is by using VMware's Solution Exchange platform in order to download a ready to run binary (*.vmoapp format).
Due to its open source nature you also can download only the source code and compile it yourself.
Whatever you choose to do: the following chapters provide you with instructions for both methods.

3.1 Downloading binaries from VSX

Coopto is available as ready-to-use binary via VMware's Solution Exchange (VSX). In order to download Coopto, please follow the provided steps.

▼ ***Note***

> The look and feel of the provided screenshots may change in future. The basic process however should be identical independent of VSX portal updates.

1. Visit VMware Solution Exchange on solutionexchange.vmware.com.
2. Login using existing VSX credentials or create a new account by using the *register* → *customer* link as shown in Figure 1.

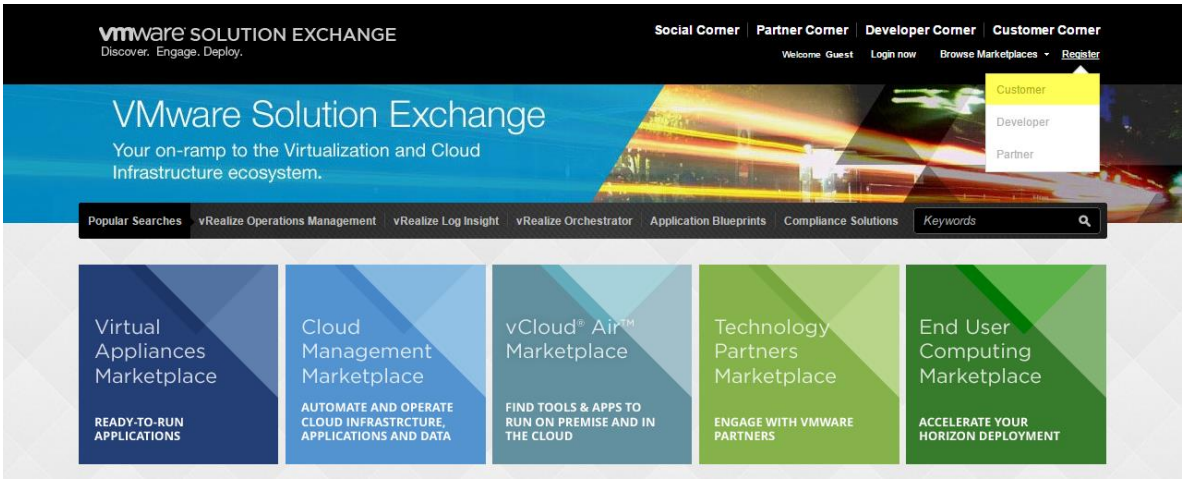


Figure 1: VMware Solution Exchange portal

3. Visit the "Cloud Management Marketplace" for vRealize Orchestrator by clicking the *vRealize* Orchestrator link.
4. Find the Coopto plug-in by using the search function within VSX or by simply browsing through the marketplace.
5. Open the Coopto plug-in page by clicking the Coopto solution in the marketplace.
6. Download Coopto by clicking the try button on the right hand side of the Coopto plug-in page.
7. You should be provided with a *.vmoapp binary of Coopto.

3.2 Downloading source from GitHub

The source code is distributed using GitHub as shown in Figure 2. Visit the project page if you wish to get more information. The readme file as well as the project WiKi will provide you with the necessary information in order to compile the most current source into a binary *.vmoapp.

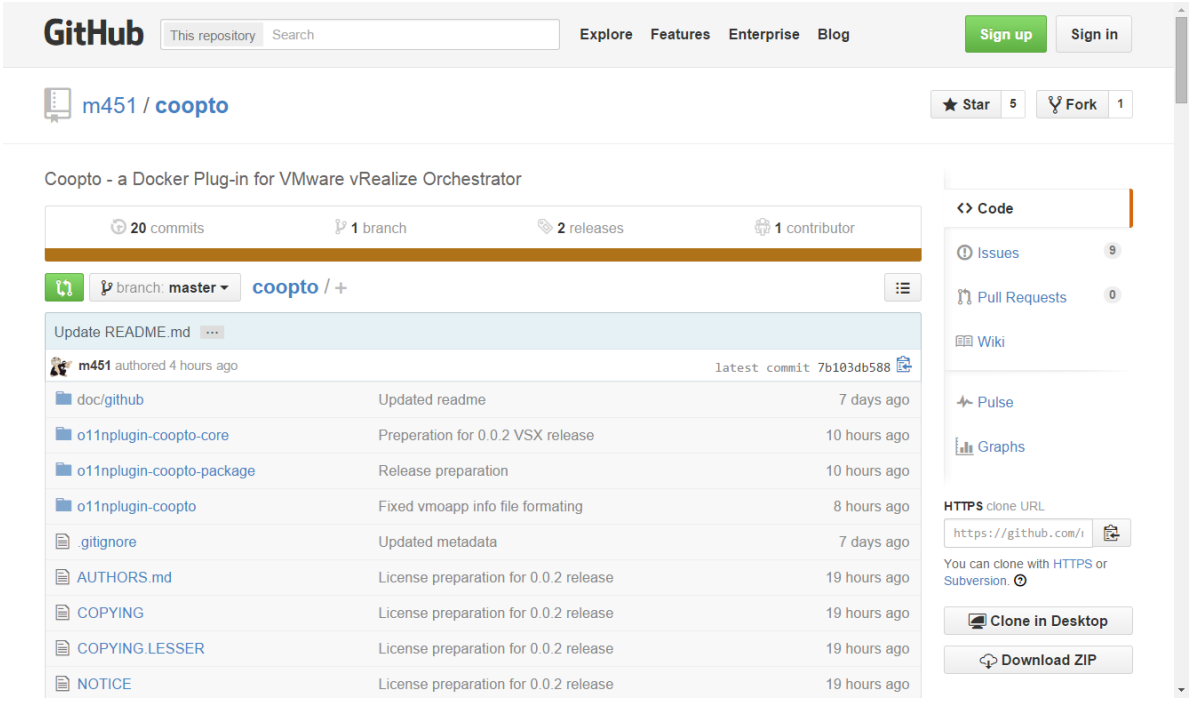


Figure 2: Coopto GitHub project page

4 Requirements

The following chapter provides information about the environment requirements for Coopto. The requirements are as follows:

- Docker with a remote API version 1.15 - 1.16 [1]
- Docker remote API service running and listening on a TCP (TLS disabled) [2]
- The Docker remote API port is reachable from vRealize Orchestrator

▼ *Note [1]*

Note that the current development process is based on Dockers remote API 1.16. If useing different versions, keep in mind:

Versions <1.15 may work and we even provide a selection down to 1.13 but please note that those versions are not tested and may result in unexpected side effects or poor user experience.

Versions >1.16 should work with a limited feature set. If you want to use a Docker host with remote API >1.16, select the version closest to the Docker host version you're using when running the "add a docker node" workflow. This will explicitly set the remote API to use for every API call.

As all this may change in future, before you upgrade your Coopto installation you have to verify that your Docker hosts are still supported.

▼ *Note [2]*

Coopto doesn't support Dockers TLS feature yet. If you want to secure the setup so your Docker remote API is not exposed to your entire network, giving everyone on that network control over your Docker service, consider protecting the remote API port using iptables.

An alternative could be adding a second network interface only used by the remote API that runs on an isolated VLAN that only exists for the Docker hosts and your vRO server.

5 Using Coopto

This chapter will show you how to install, configure and use Coopto.

5.1 Installing Coopto

If the requirements mentioned in chapter 4 are met, you may proceed with the installation, which is documentet in details in this chapter.

1. Login to your vRealize Orchestrator configuration page, available via https://your-vro-address:8283 and navigate to the plug-ins section as shown in Figure 3.
2. Within the plug-ins section, click on the magnifier, browse and select the *coopto.vmoapp* you downloaded and start the installation by clicking the *upload and install* button.

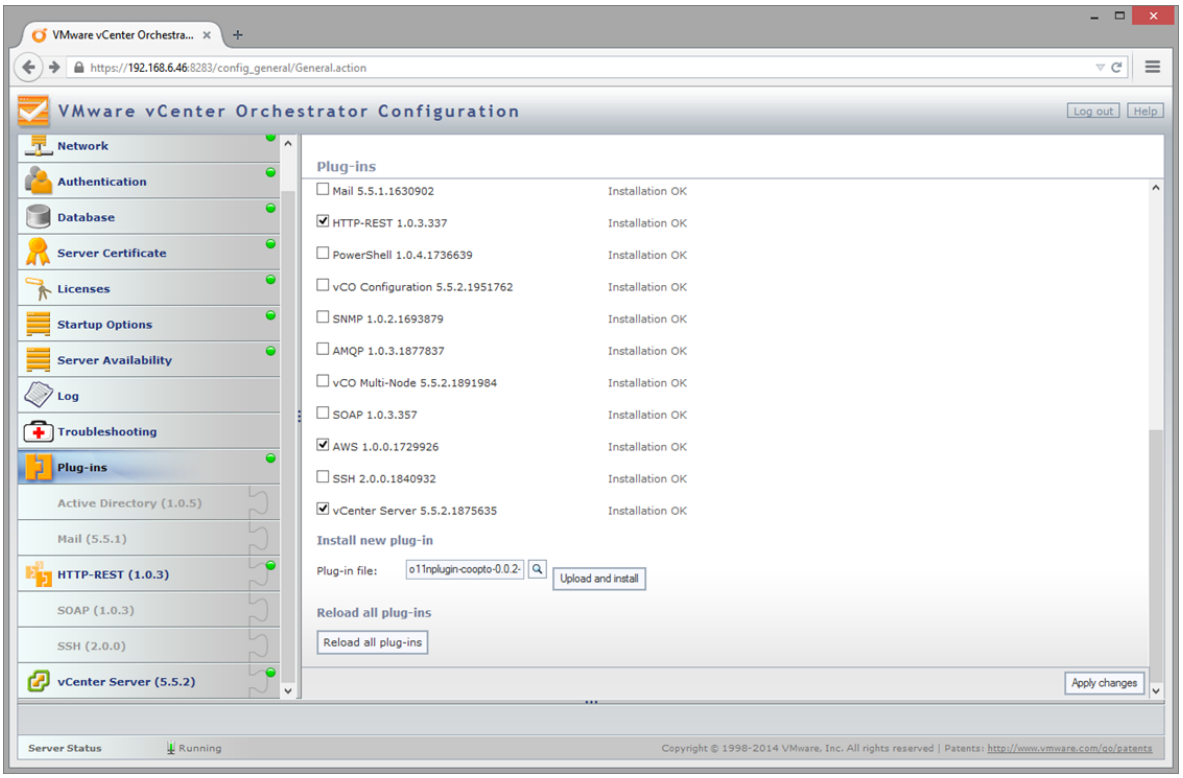


Figure 3: vRO configuration page – plug-ins section

3. On the next page you'll have to read and accept the license agreement, see Figure 4.

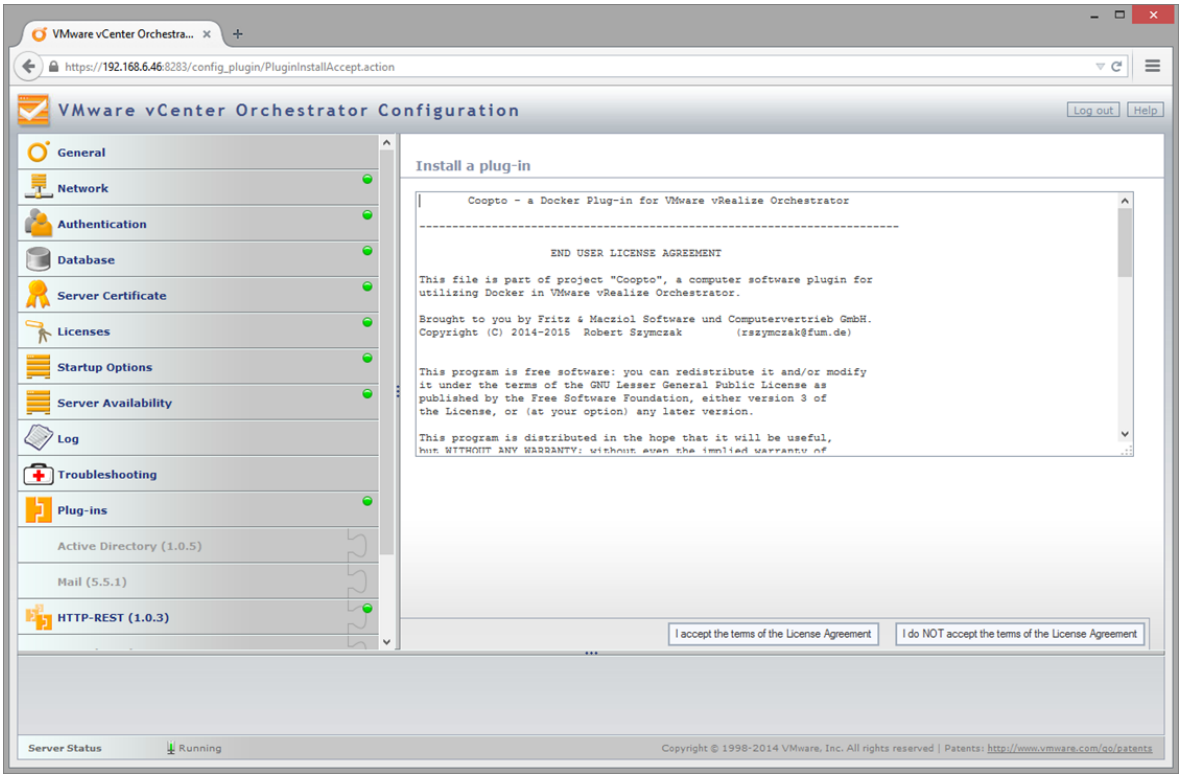


Figure 4: vRO configuration page - Coopto installation process

4. The plug-in should install and you should be returned to the plug-ins section, where you'll see the installation status listed on the top. It should read "*plug-in installed*" if everything went good, e.g. Figure 5.

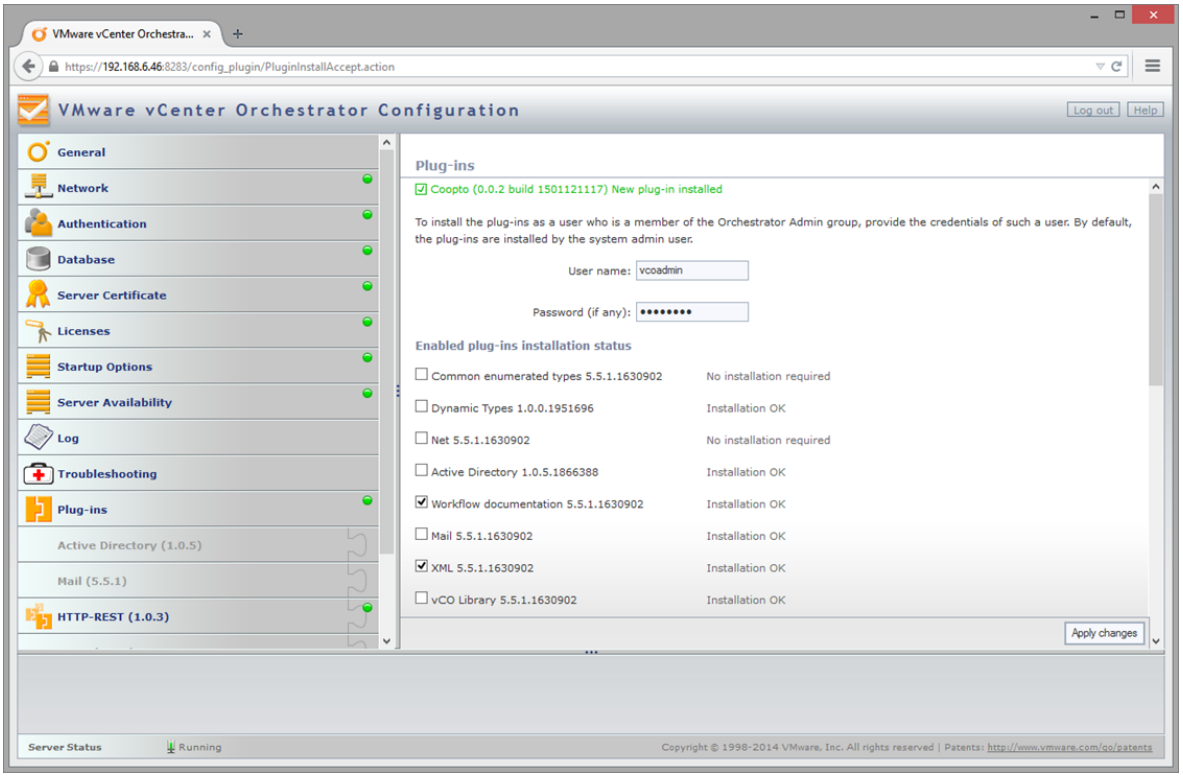


Figure 5: vRO configuration page - successfull plug-in installation

5. In order to use Coopto, you'll need to restart the orchestration service of your vRO. This will stop all running workflows, so you probably want to do this within a service window. To restart the orchestration service, browse to the *startup options* section and click *"restart services"* as shown in Figure 6.

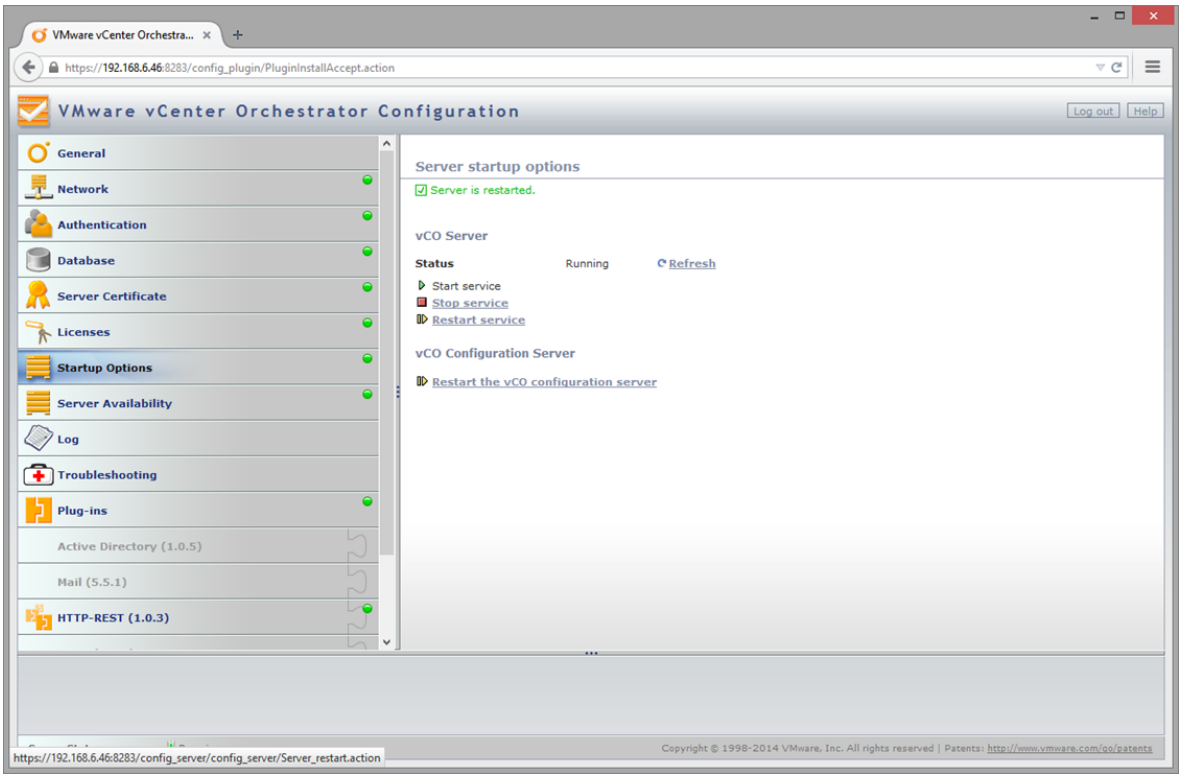


Figure 6: vRO configuration page - startup options section

6. Once the orchestration service is restarted, you should be able to use Coopto within your vRO client as documented in chapter 5.2.

5.2 Adding a Docker node

This chapter will provide you with basic information you need to verify that Coopto is working as expected and lead you through the process of linking your Docker host with your vRO installation.

1. Open your vRO client and login with sufficient credentials, see Figure 7.

Figure 7: vRO client loging page

2. Once you logged in, change the view to *design* and browse to the package tab. If the Coopto installation was successful, you should see a package called *de.fum.coopto*. If you don't see that package it might be because something went wrong while installing the plug-in. In that case, repeat the steps documented in chapter 5.1.

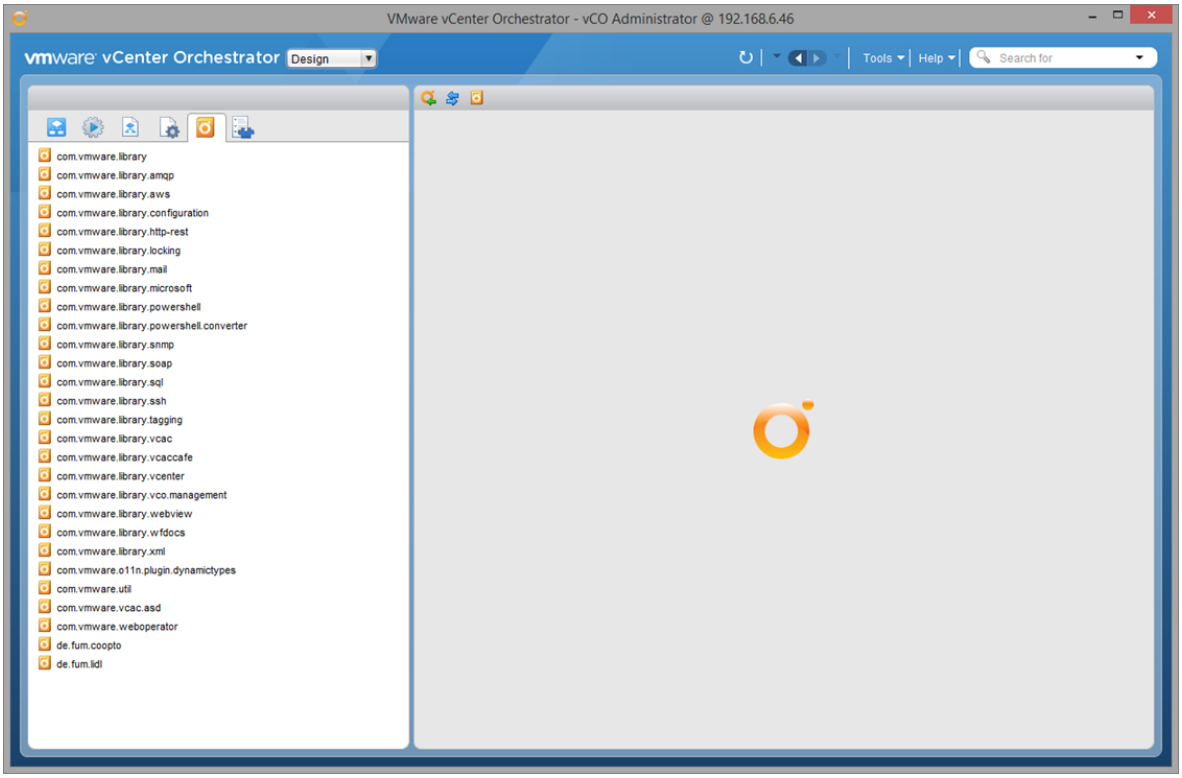


Figure 8: vRO client - Coopto package listed after successfull instalation

3. When opening the workflow tab, you should see multiple workflows listed at *Library/Coopto/* as shown in Figure 9.

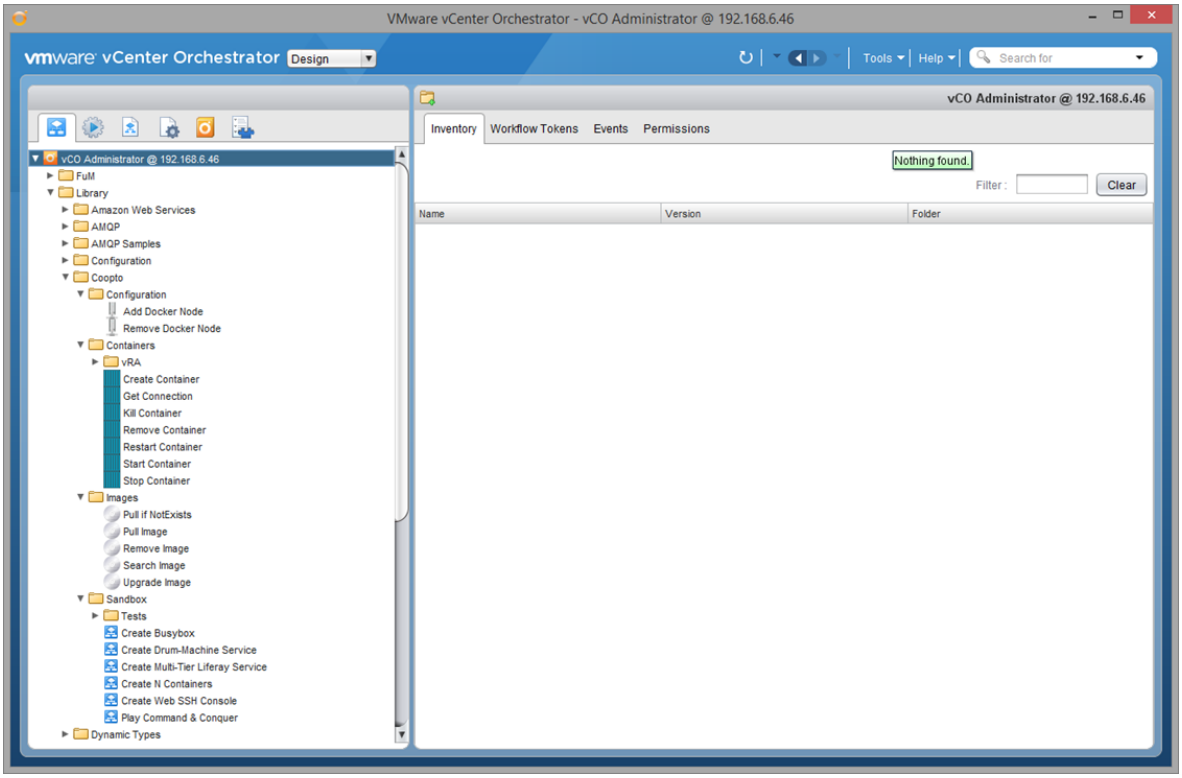


Figure 9: vRO client - Coopto workflows after successfull installation

4. Before you can start using Coopto for container management you first have to add one or more Docker hosts. Right click the workflow *"add Docker node"* from *Library/Coopto/Configuration* and select *"start workflow"* from the context menu that opens, see Figure 10.

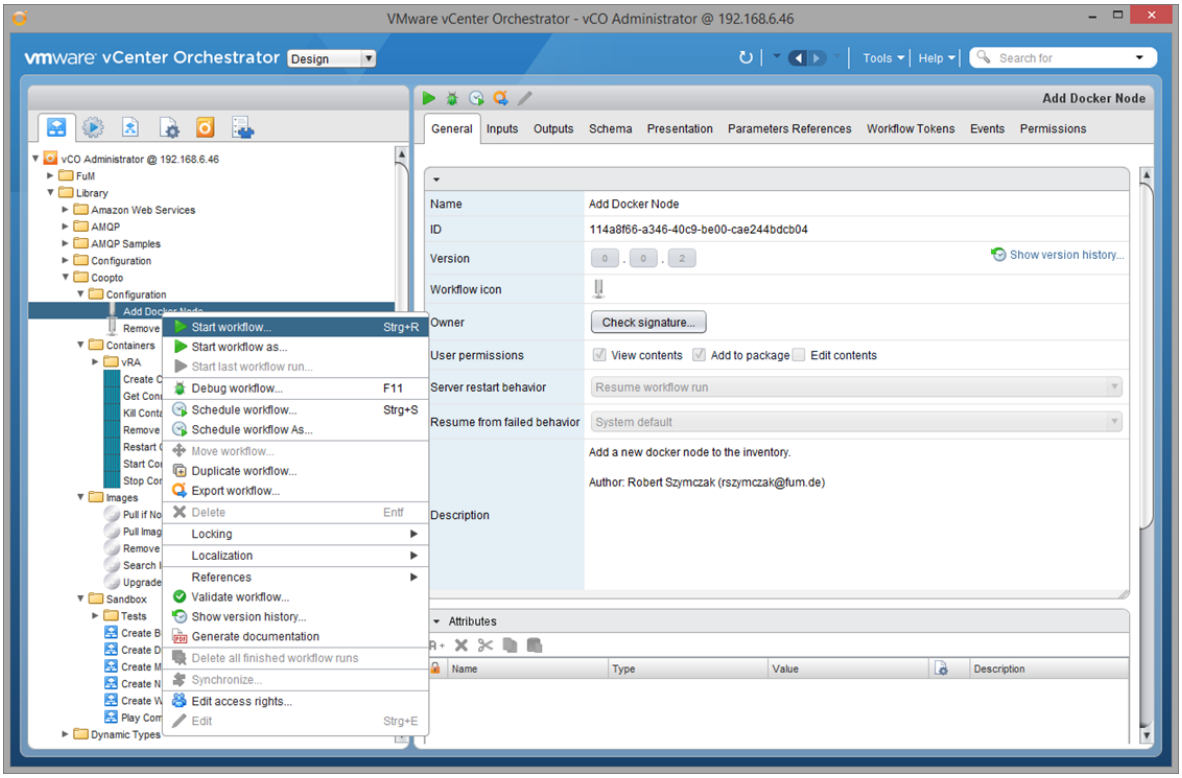


Figure 10: vRO client - running Coopto "add Docker node" workflow

5. The *"add Docker node"* workflow should open and present you with some input fields. Fill them with the appropriate parameters, Figure 11 shows an example configuration. After you've filled out the form, click *submit* to start the workflow.

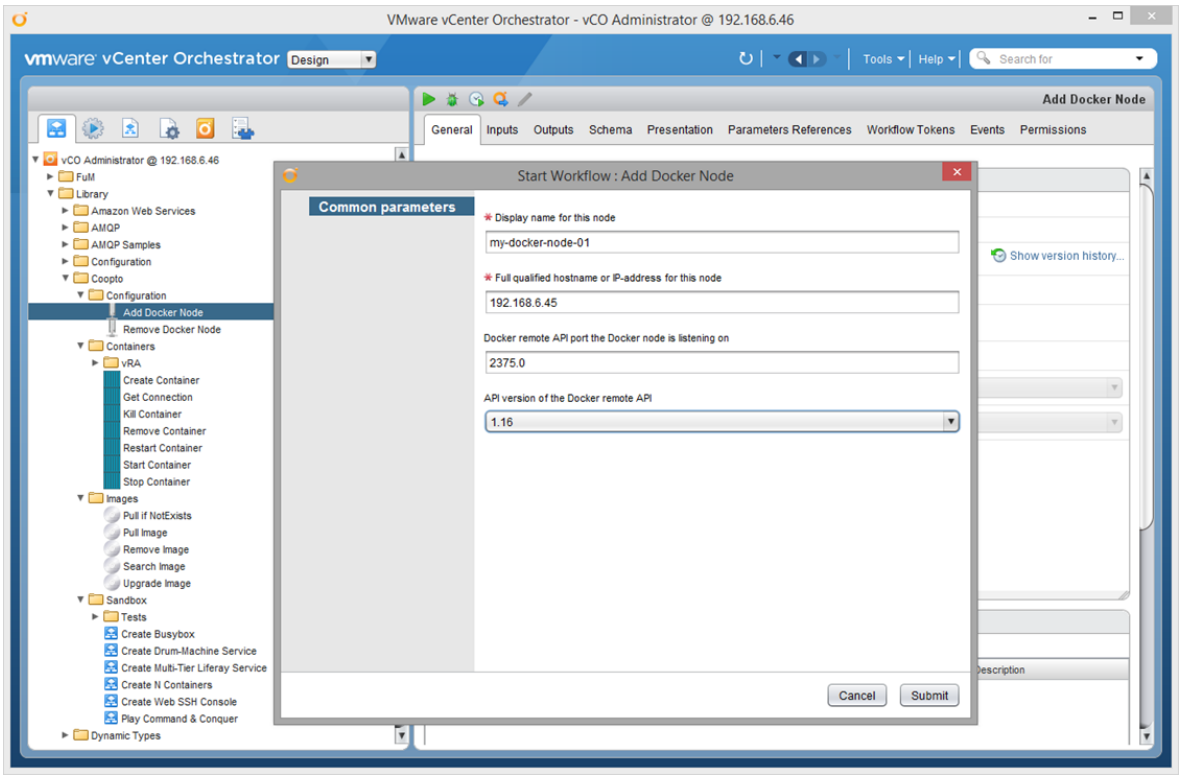


Figure 11: vRO client - "add Docker node" workflow input fields

6. If everything worked as expected, you should see a successful workflow run, e.g. Figure 12.

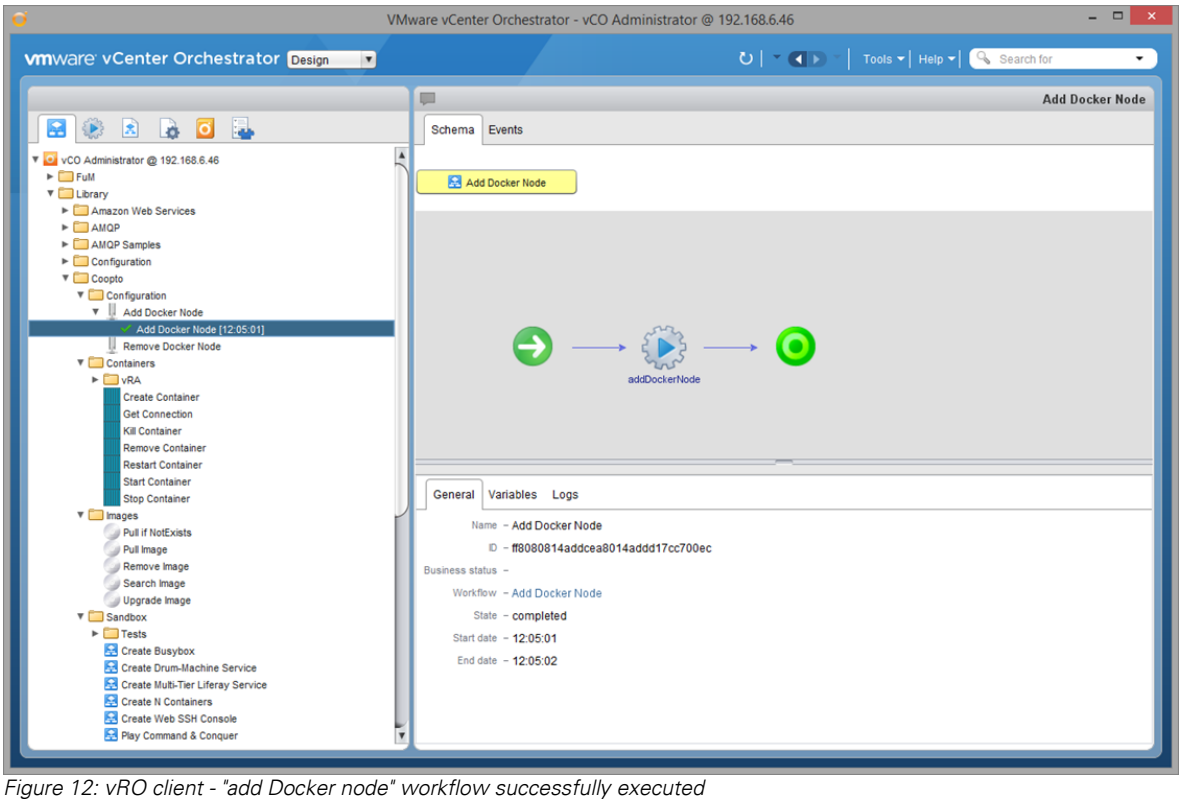


Figure 12: vRO client - "add Docker node" workflow successfully executed

7. Open the inventory tab in your vRO client. You should see that the Coopto inventory now lists the Docker node you just added. If vRO is able to communicate with the Docker host, then the *status* attribute should read *"online"*, see Figure 13.

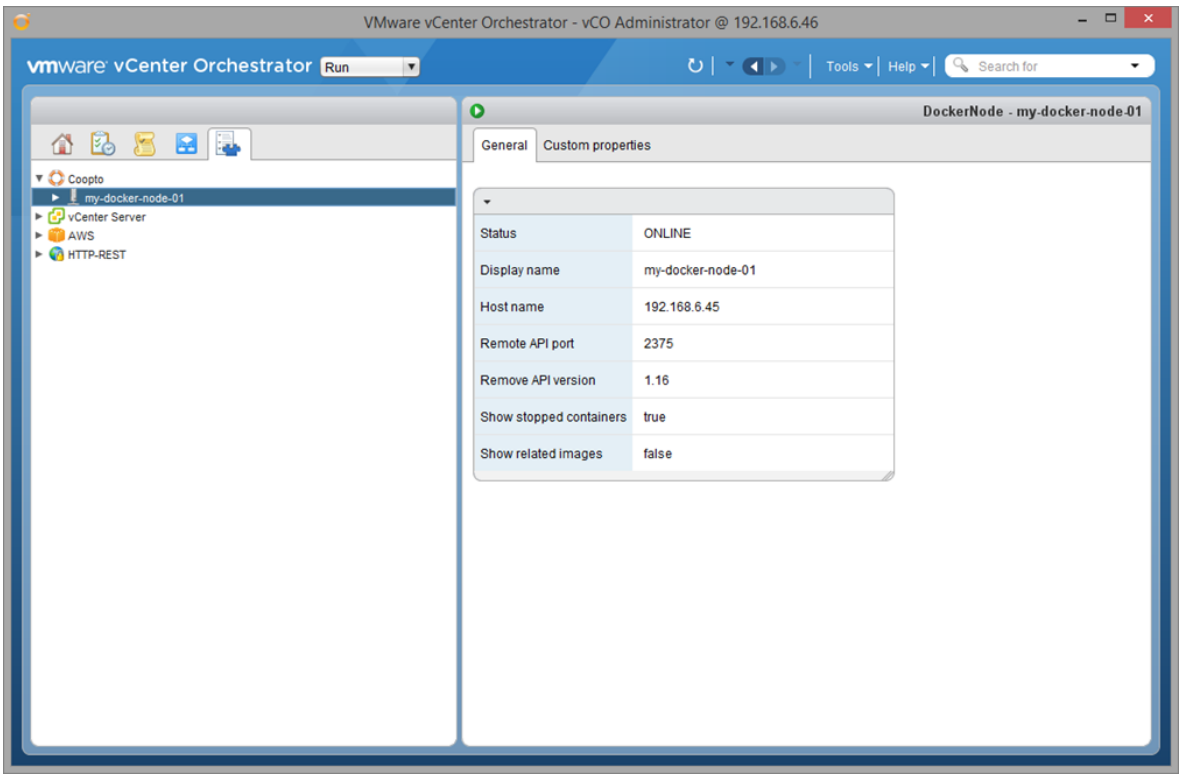


Figure 13: vRO client - Docker node visible in inventory view after workflow run

8. If vRO is not able to connect to your Docker host, and therefore the *status* attribute displays *"offline"*, you probably ran into some network issues. Make sure the requirements mentioned in chapter 4 are met before you proceed. You can use the vRO refresh button on the top right in your vRO client's inventory tab to update the *status* attribute.

5.3 First steps in Coopto

This chapter will illustrate a basic example operations with Coopto so you can get familier with the plug-in.

1. Within your vRO client, browse to the inventory tab and right-click the Docker node you added as described in chapter 5.2. You should get a context menu like shown in Figure 14. Select the *"pull image"* workflow.

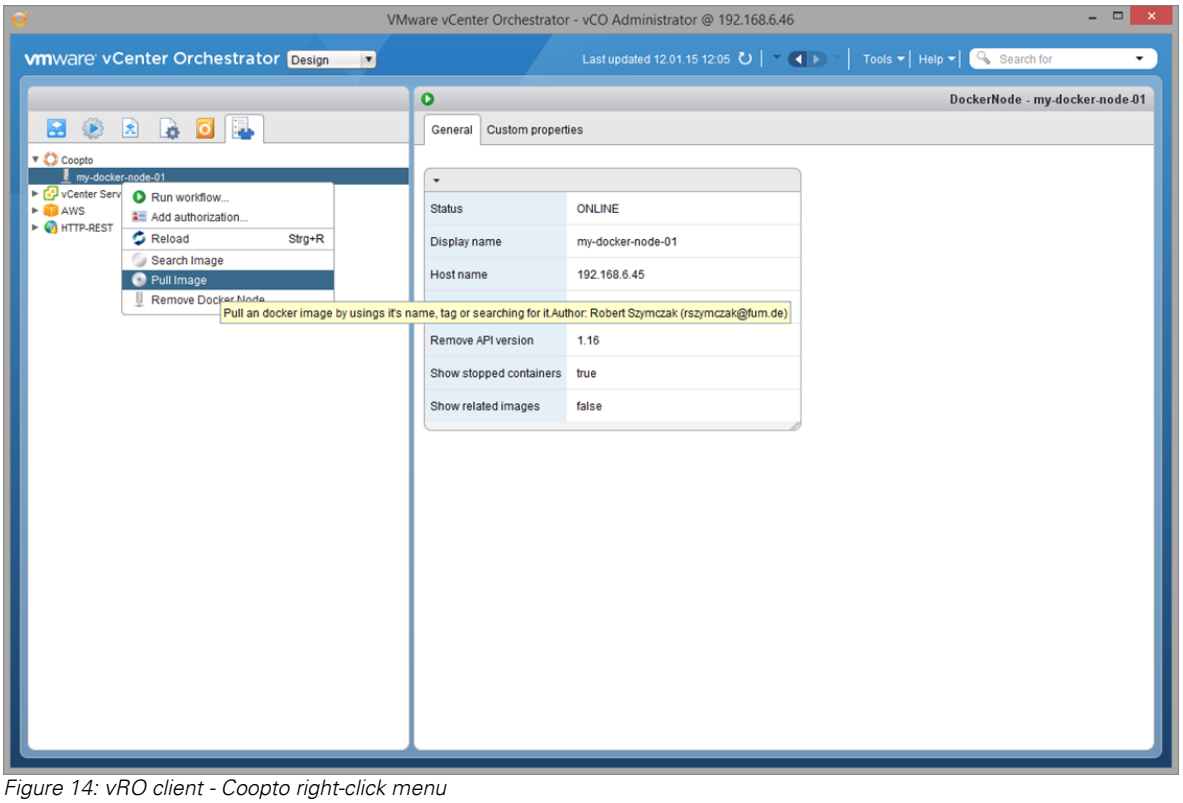


Figure 14: vRO client - Coopto right-click menu

2. The pull workflow allows you to download a new Docker image from the central Docker hub repository. In order for this to work your Docker host needs to have a connection to the Internet, so it can reach the Docker hub. You may reconfigure your Docker host to use a private image repository. Coopto will use whatever repository you configured as default within your Docker host. Within the pull workflow you may search for images to pull or – if you know the exact tag of an image – you can specify the image by entering its tag. In this example we choose to input the image name and thus select "no" as parameter for the search mode, as shown in Figure 15.

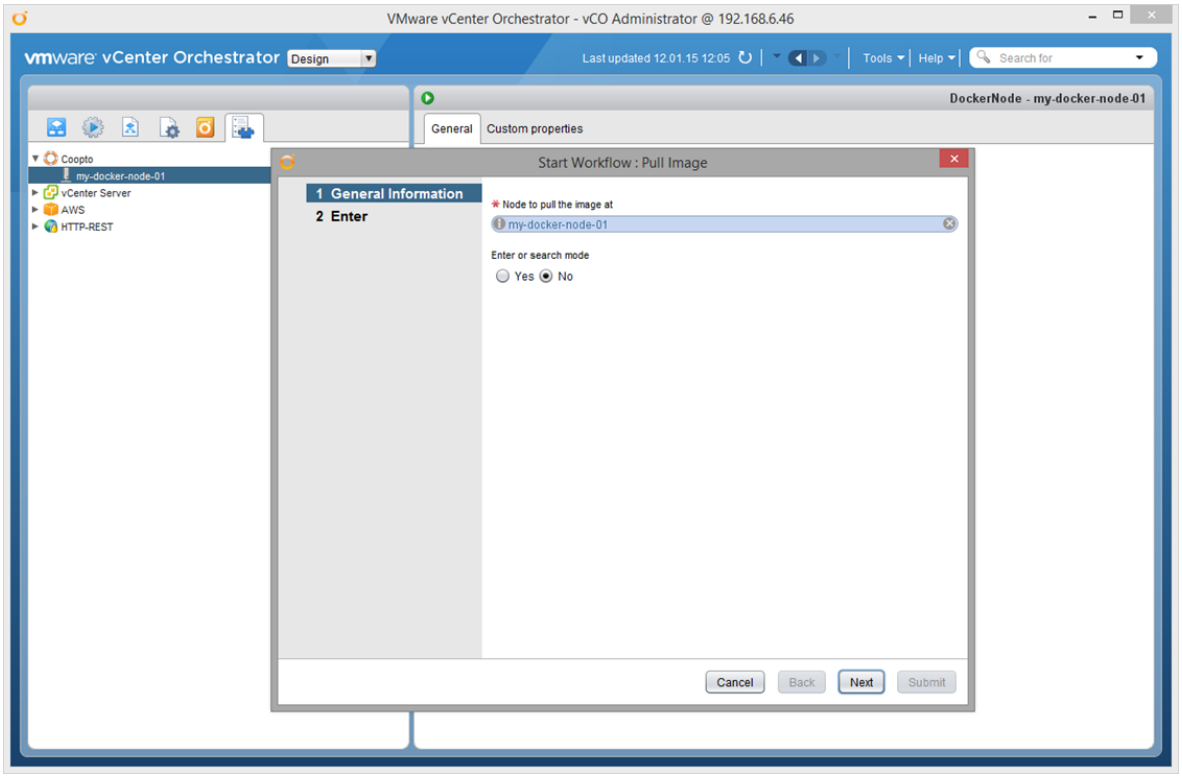


Figure 15: vRO client - Coopto "pull image" workflow

3. Enter any image tag you want that is available in your configured repository – which is Docker hub by default. In this example we enter *"busybox:latest"*, which will download the latest release of busybox.

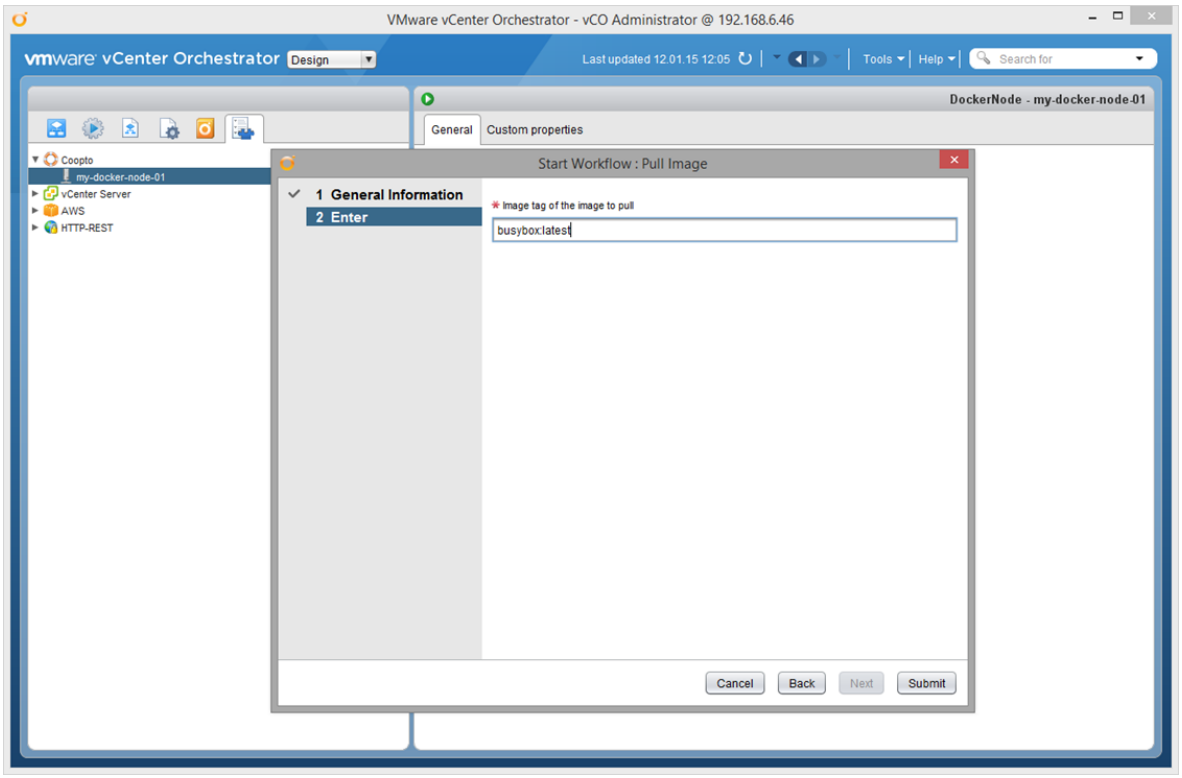


Figure 16: vRO client - Coopto "pull image" workflow ready to execute

4. Depending on the uplink connection speed to your Docker repository, the pull operation may take a few minutes. Once the pull finishes, the workflow execution should be successful and your Coopto inventory should automaticly update and list the new image you just pulled, see Figure 17. Also shown in Figure 17 is the right-click context menu provided for Docker image objects.

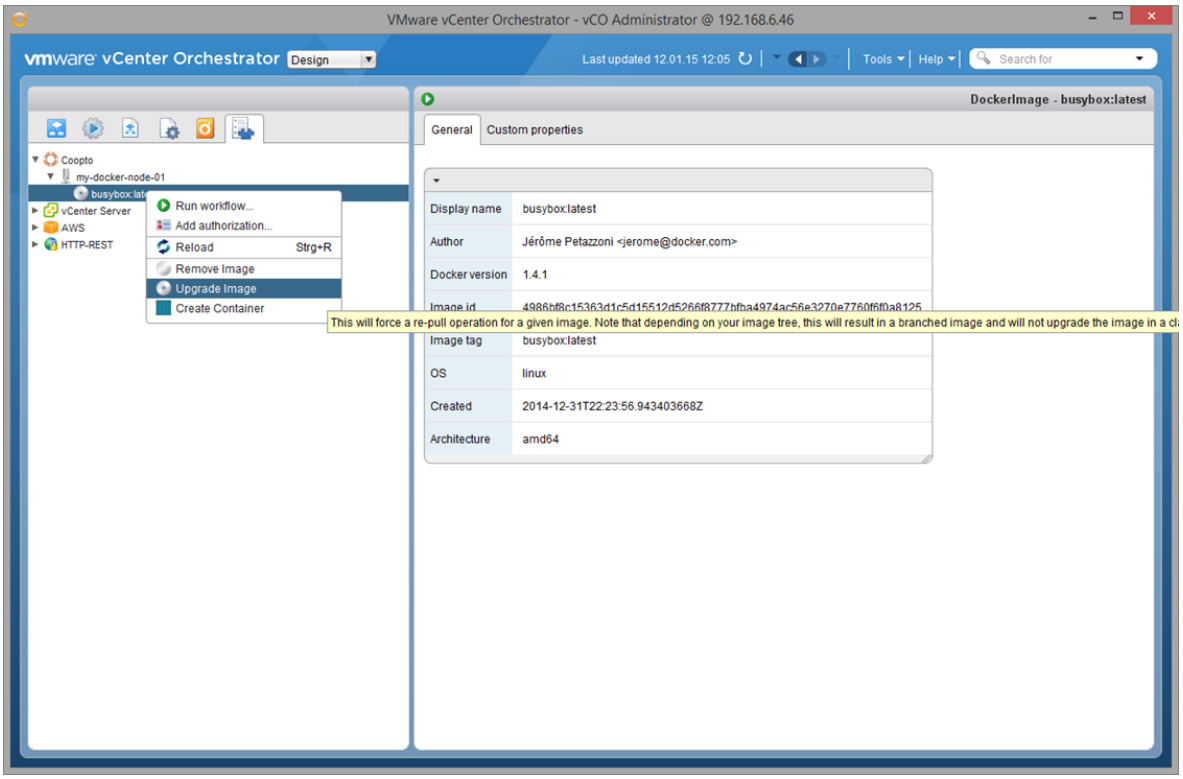


Figure 17: vRO client - Coopto right-click menu for images

5. All right-click operations are available thoughout workflows within the */Library/Coopto* workflow folder. The sandbox subfolder includes a set of workflows that will give you examples how to build custom, container based services using Coopto, see Figure 18.

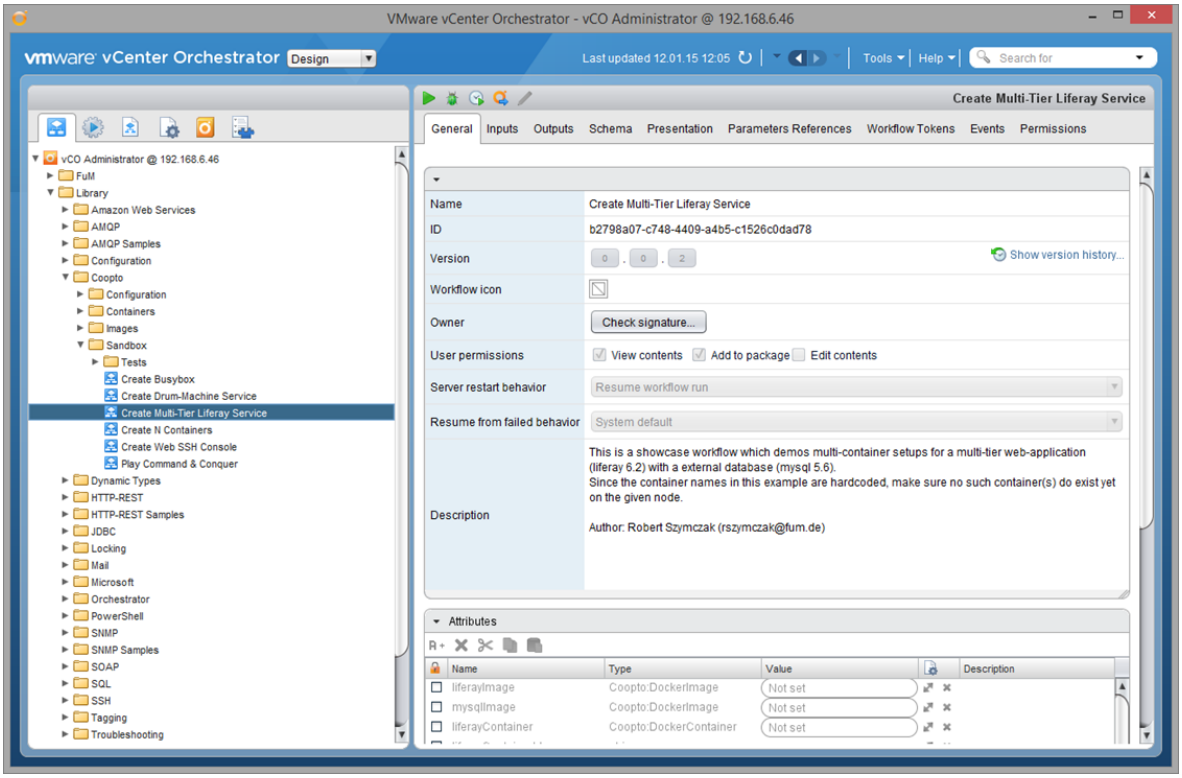


Figure 18: vRO client - Coopto sandbox workflows for quick-start purpose

6. You now should have a basic idea of how Coopto is handled. For further information you can always visit the Coopto project homepage on GitHub or the vRO community forums.

▼ *Note*

In order to get you started we build some workflows which leverage nearly every function that Coopto currently provides, so called "sandbox" workflows. Be aware that those are ***not*** production ready examples.

For more information you can read the WiKi about useage examples and sandbox workflows on the GitHub project page.

6 Support

To provide you with the support you need to get things going we've put together a cosy wiki page for you. There's also a section about debugging. This is the first place you should visit if you got any issues or questions.
If the issue you're facing isn't covered within the wiki you might consider visiting the Coopto thread within the community forums of vRO. Feel like you run into a software bug related to Coopto itself? Be free to submit a new issue on the issues page.

Please understand that the open source nature of this project also means that we – as in we the Coopto open source community – cannot guarantee to provide you with support and even less with support within a certain response time.

If you require enterprise support with costs you might consider contacting vmware@fum.de but please be aware that this is out of scope of the open source process this project is based on and thus should not be submitted into the projects issues section.

7 Security Disclosure

If you have any issue regarding security, please disclose the information responsibly by sending an email to vmware@fum.de.