QualiSystems CloudShellTM

NFV Installation Automation

User Guide

Version 1.0.0

Release date: July 2016

Document version ()

[1 Overview 4](#_Toc466640197)

[2 Package Content 5](#_Toc466640198)

[2.1 NFV Sandbox 5](#_Toc466640199)

[2.2 Quick Start 5](#_Toc466640200)

[2.3 Deploying the CloudShell appliance 6](#_Toc466640201)

[2.3.1 Configuring CloudShell VM 7](#_Toc466640202)

[2.3.2 Installing CloudShell license token 8](#_Toc466640209)

[2.3.3 CloudShell Services overview 9](#_Toc466640210)

[2.4 CloudShell Portal 10](#_Toc466640211)

[2.5 Importing the NFV Package 12](#_Toc466640212)

[2.5.1 Installing the NFV Solution package on the CloudShell VM 12](#_Toc466640213)

[2.6 Inputs 13](#_Toc466640214)

[2.7 Logs 13](#_Toc466640218)

[2.8 Executing in steps 13](#_Toc466640219)

[3 Compatibility 17](#_Toc466640220)

[3.1 CloudShell compatibility 17](#_Toc466640221)

[4 Technical Guide 18](#_Toc466640222)

[4.1 OnRack Discovery Workflow 18](#_Toc466640223)

[4.2 Main Workflow 18](#_Toc466640224)

[4.3 Shells & Functions 18](#_Toc466640225)

[4.4 vCenter Configuration 21](#_Toc466640226)

[4.4.1 vCenter Configuration breakdown 21](#_Toc466640227)

[4.5 Versa configuration & workflow 21](#_Toc466640228)

[4.5.1 Versa Configuration breakdown 23](#_Toc466640229)

[4.6 Nagios 24](#_Toc466640230)

[4.6.1 Nagios relationship with CloudShell 24](#_Toc466640231)

[4.6.2 Quick overview on Nagios component & how-to 25](#_Toc466640232)

[4.7 vROPS 25](#_Toc466640233)

[4.8 vLog Insight 26](#_Toc466640234)

[4.9 NSX 26](#_Toc466640235)

[5 Relevant HW/SW compatibility 28](#_Toc466640236)

[6 Managing scripts and drivers 29](#_Toc466640237)

[6.1 Downloading a script/driver file from CloudShell 30](#_Toc466640238)

[6.2 Updating a script file 30](#_Toc466640239)

[6.3 Updating a driver 30](#_Toc466640240)

[7 Post-Deployment How-to and extra steps 32](#_Toc466640241)

[7.1 Using the ScaleIO web plugin 32](#_Toc466640242)

[7.2 Nagios 32](#_Toc466640243)

[7.3 vLog Insight, vRA, vROPS & NSX 33](#_Toc466640244)

[8 Known Gaps 35](#_Toc466640245)

[8.1 Versa Gaps 35](#_Toc466640246)

[8.2 vCenter Gaps 35](#_Toc466640247)

[8.3 Nagios Gaps 36](#_Toc466640248)

[8.4 ScaleIO Gaps 36](#_Toc466640249)

[9 Known Issues 37](#_Toc466640250)

[10 Changes from the previous code release 38](#_Toc466640251)

[Legal Notice 39](#_Toc466640252)

# Overview

QualiSystems CloudShell is the orchestration platform that enables end-to-end provisioning and configuration of the EMC NFV solution.

The NFV installation automation performs an unattended. All manual steps are replaced by automation using a variety of techniques, including SSH, REST APIs, PowerCLI, and browser automation. No user intervention is required during the course of deployment.

The deployment requires inputs such as IPs, hostnames, existing and new credentials, DNS and NTP servers. It allows optional customization of many vSphere entity names, addresses, configurations, and other settings.

# Package Content

CloudShell NFV Installation Automation package contains the following elements:

CloudShell appliance containing drivers, scripts, and NFV installation files

## NFV Sandbox

The NFV sandbox is a visual representation of the NFV components.

Multiple sandbox environments have been defined, corresponding to different deployment types:

* SIO vRA NSX

Note: at the moment only the vRA environment is supported.

Objects within a sandbox corresponds to a VM or group of VMs. The sequence of operations is defined in the workflow code rather than the visual arrangement.

## Quick Start

The overall workflow

1. Deploy the CloudShell NFV installation appliance.
2. Open the CloudShell portal in a web browser, e.g. <http://cloudshell1> and provide an activation key
3. Extract the ‘NFVInstaller.zip’ file on the CloudShell appliance, and run the ‘Install.cmd’ file to deploy all of the required packages.
4. Open the CloudShell portal in a web browser, e.g. <http://cloudshell1> and log in as admin/admin
5. Under Inventory in the portal, create a Site Manager resource and fill the desired IP ranges and other network settings (DNS, NTP, domain suffix) – There should be one Site Manager per Cloud deployment.
6. Create an OnRack resource and fill the IP and credentials
7. Select a sandbox type (e.g. SIO vRA NSX) and reserve it. Choose a duration greater than 8 hours.
8. Execute **Set Inputs** to generate IPs
9. Customize settings on the different components (vCenter, NSX, etc.)
10. Execute **Run Orchestration Steps**
11. When you need to refresh the OnRack inventory: Reserve the sandbox OnRack Discovery. It will automatically add all OnRack resources defined in step #5, run autodiscovery on them, and sync CloudShell with the OnRack server inventory. End the reservation after this process completes. You can repeat this discovery step whenever the OnRack server inventory may have changed, or schedule it to run daily.

## Deploying the CloudShell appliance

The CloudShell appliance is a Windows 2012 OVA provided by QualiSystems.

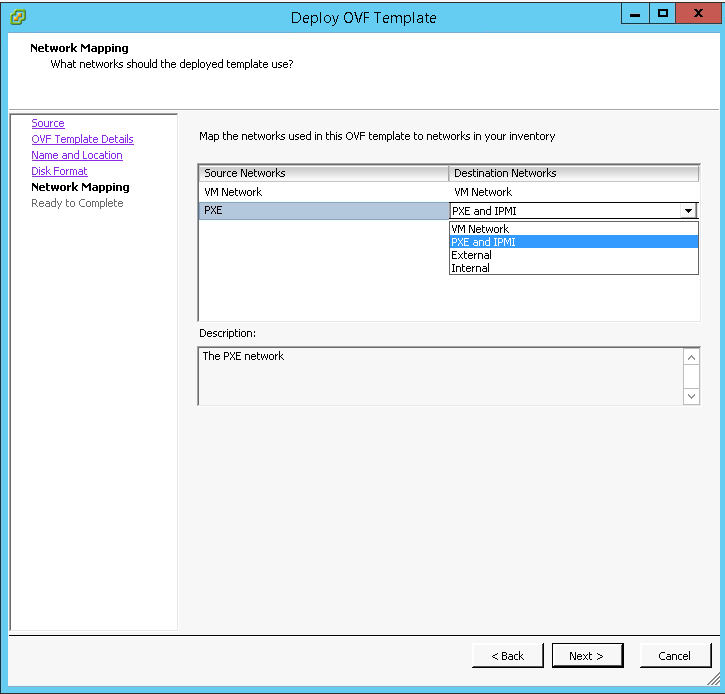
The Minimum VM hardware requirement are;

* 2 vCPU
* 16GB RAM
* 150GB of storage

Recommended VM hardware;

* 4 vCPU
* 16GB RAM
* 150G of storage

The CloudShell appliance has two vNICs, one for the management network and one for the OnRack PXE network. Assign these to the appropriate portgroups when deploying:

* 

Windows runs in a trial mode until a license is provided.

QualiSystems will provide a CloudShell license token that will require internet access in order to load it.

Most software and files are preinstalled on the CloudShell appliance. For licensing reasons, it may be necessary to download certain software directly from the vendor and install it on the appliance. If the following are not present, they must be obtained from the vendor.

* VMware PowerCLI 6.0R3
* VMware ovftool 4.1.0 installed in the default location
* JRE 7 (not 6 or 8) 64-bit installed in the default location
* NSX Manager OVA in c:\deploy
* vROPS OVA in c:\deploy
* vLog Insight OVA in c:\deploy
* Versa OVAs in c:\deploy
* ScaleIO OVAs and installation script in c:\deploy\sio
* vRA OVA in c:\deploy

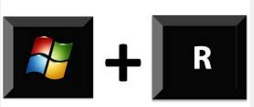
### Configuring CloudShell VM

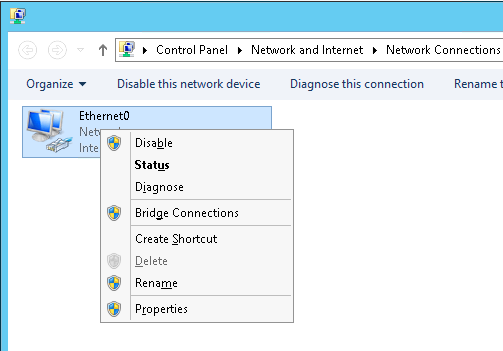
After deploying the CloudShell OVA, it is needed to configure the network settings,

The CloudShell OVA comes with static IP settings, which needs to be modified to reflect the customer network.

The Default Username and Password are (local user) “Administrator” with password “Password1”

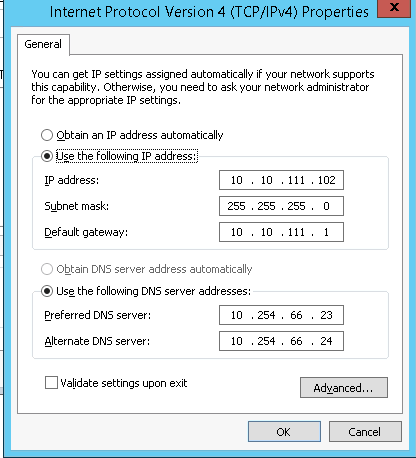
Once the VM has started, open the Console view of it from the vCenter webgui or vSphere client.

Press the Win key + “R” () on the keyboard to open the ‘run’ window, and type “ncpa.cpl” alternatively, you can go to “Control Panel” > “Network and Internet” > “Network and Sharing Center” > “Change adapter settings”.

Right click on the “Ethernet” and select “properties” 

Click on the “Internet Protocol Version 4 (TCP/IPv4)” and click on “Properties”

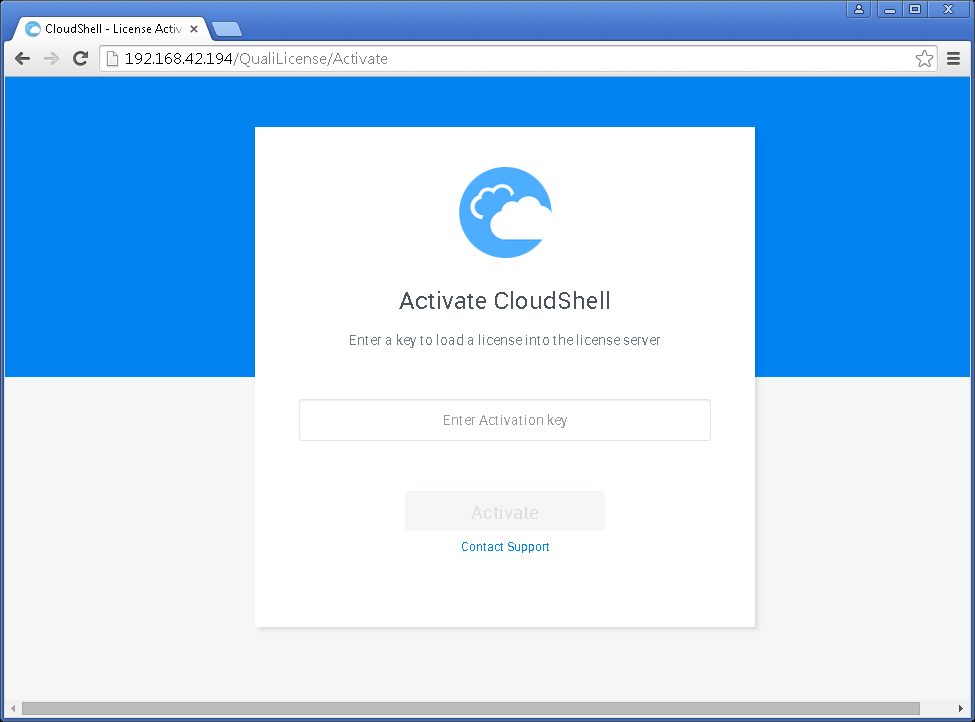
Once there, change the “IP address”, “Subnet mask”, “Default gateway” and DNS servers fields



Repeat this process for Ethernet 2, assigning an address on the 172.31.128.0 network with a netmask of 255.255.252.0 to communicate on the OnRack PXE network.

### Installing CloudShell license token

On first time lunching the CloudShell Portal, you’ll be redirected to the “License Activation” page.



Enter the license key you got from QualiSystems on the “Enter Activation Key” field and click on “Activate”

Note: using this activation method requires Internet connection.

### CloudShell Services overview

The CloudShell OVA comes preinstalled with all necessary CloudShell components, including:

Quali Server

CloudShell Portal

CloudShell Execution Server

SQL 2008 Express with Database preconfigured (Named “QUALISYSTEMS2008”)

#### Quali Server

This service in charge of running the CloudShell software, you can start\stop it from the “Services” windows screen. It is set to automatically start when the VM power on.

#### CloudShell Portal

The CloudShell Portal is running under the IIS installed on the CloudShell VM. To manage it, use the IIS Manager.

#### CloudShell Execution Server

The CloudShell Execution Server is in charge of running all of the tasks, job and commands that are being initiated from the Portal, the service run as a ‘console’. The Service under “Services” is disabled. In its place, there’s a tray icon that runs the Execution Server.



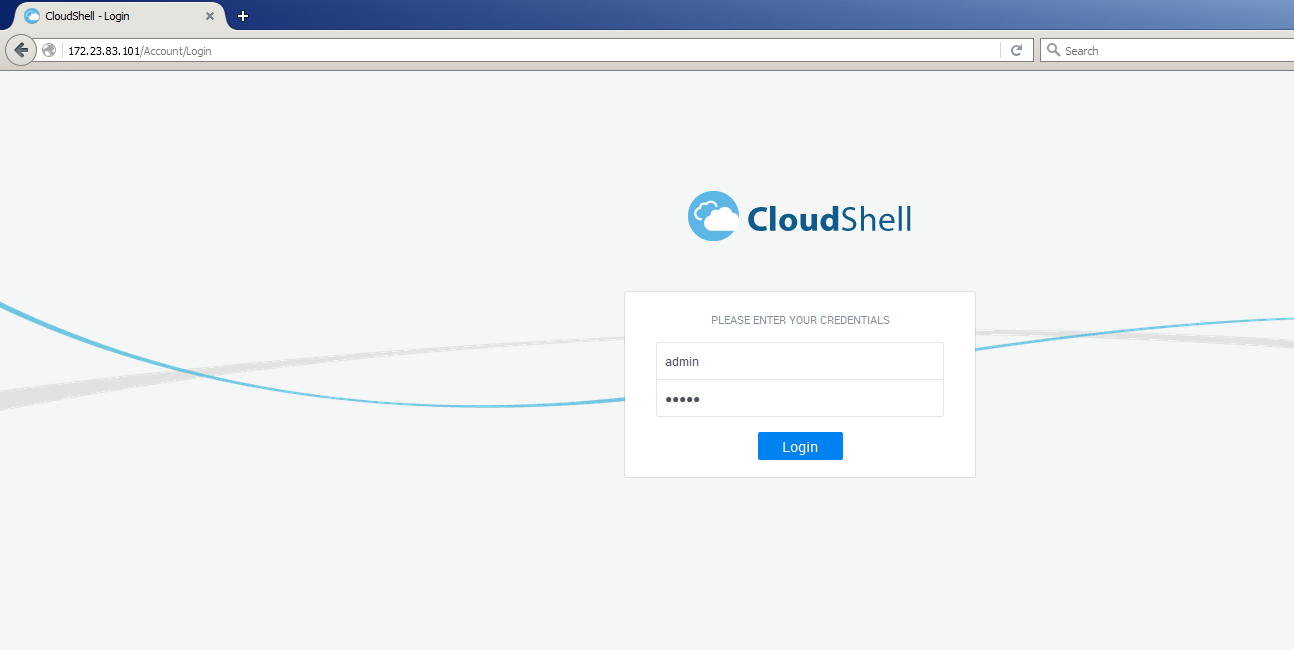
To stop the Execution Server, you can right click on it and choose “Close”

To start it, open CMD, navigate to “C:\Program Files (x86)\Qualisystems\TestShell\ExecutionServer” and run “QsExecutionServer.exe tray”

Alternatively, you can use the “Start Execution Server” shortcut on the desktop or execute the batch script under “C:\Start Execution Server.cmd"

## CloudShell Portal

Log in:



The default administrator credentials to use to log in to the portal are:  
User: admin

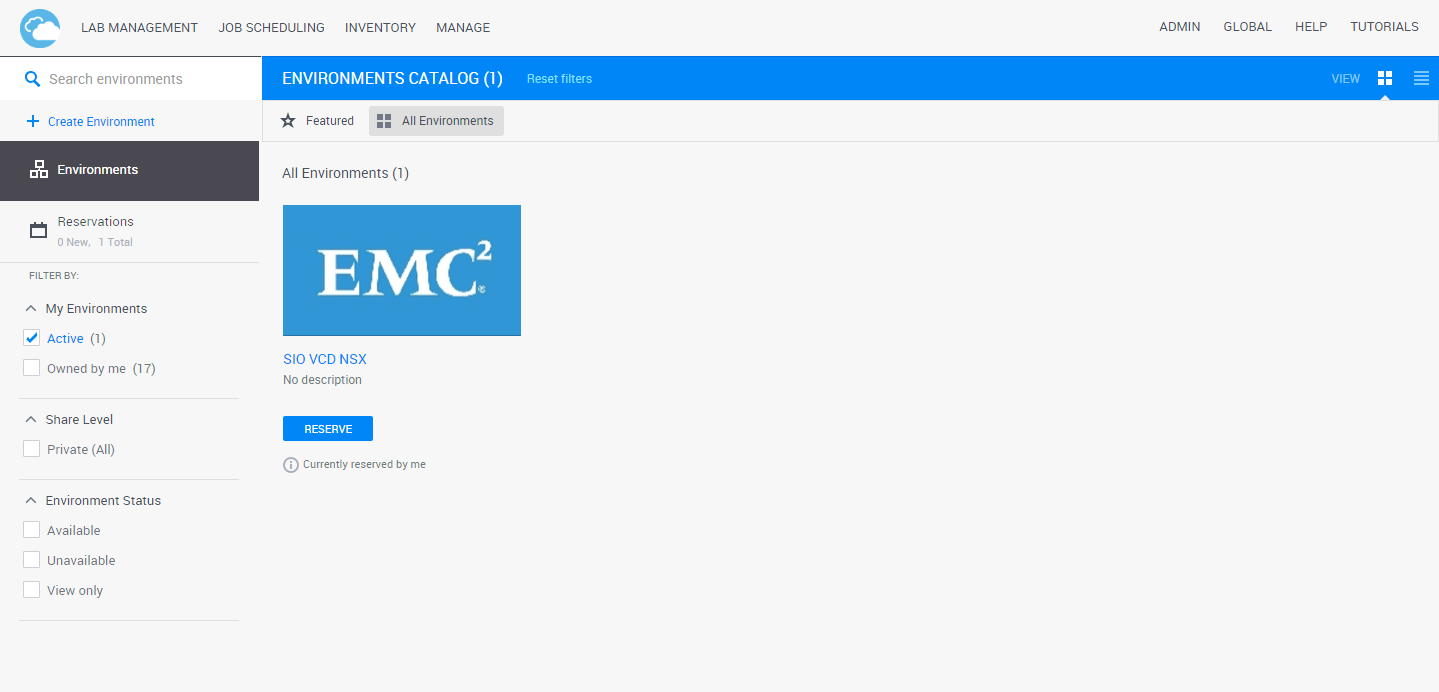
Password: admin

At the first log in from each machine, you’ll have to provide the local time zone as well.

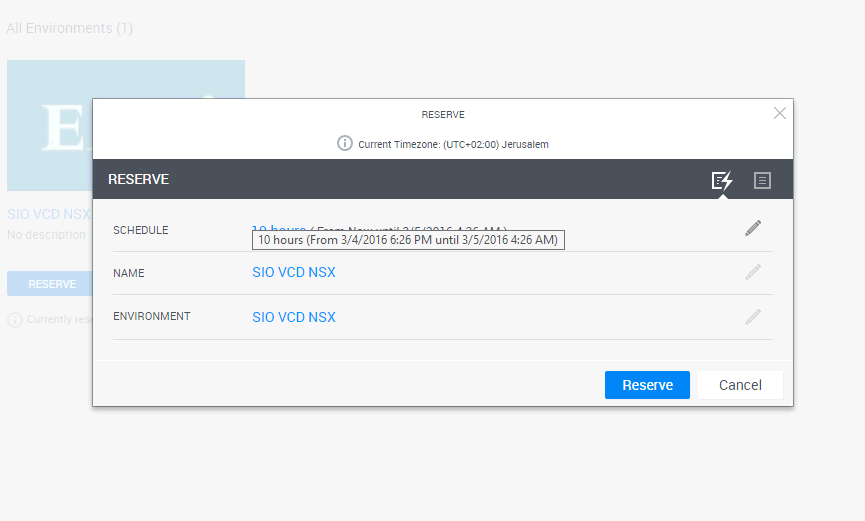
Once logged in, you’ll need to import the automation package (refer to “Importing the NFV Package and configuring Python”).

Once imported, you’ll be able to continue with the following on any other use of CloudShell.

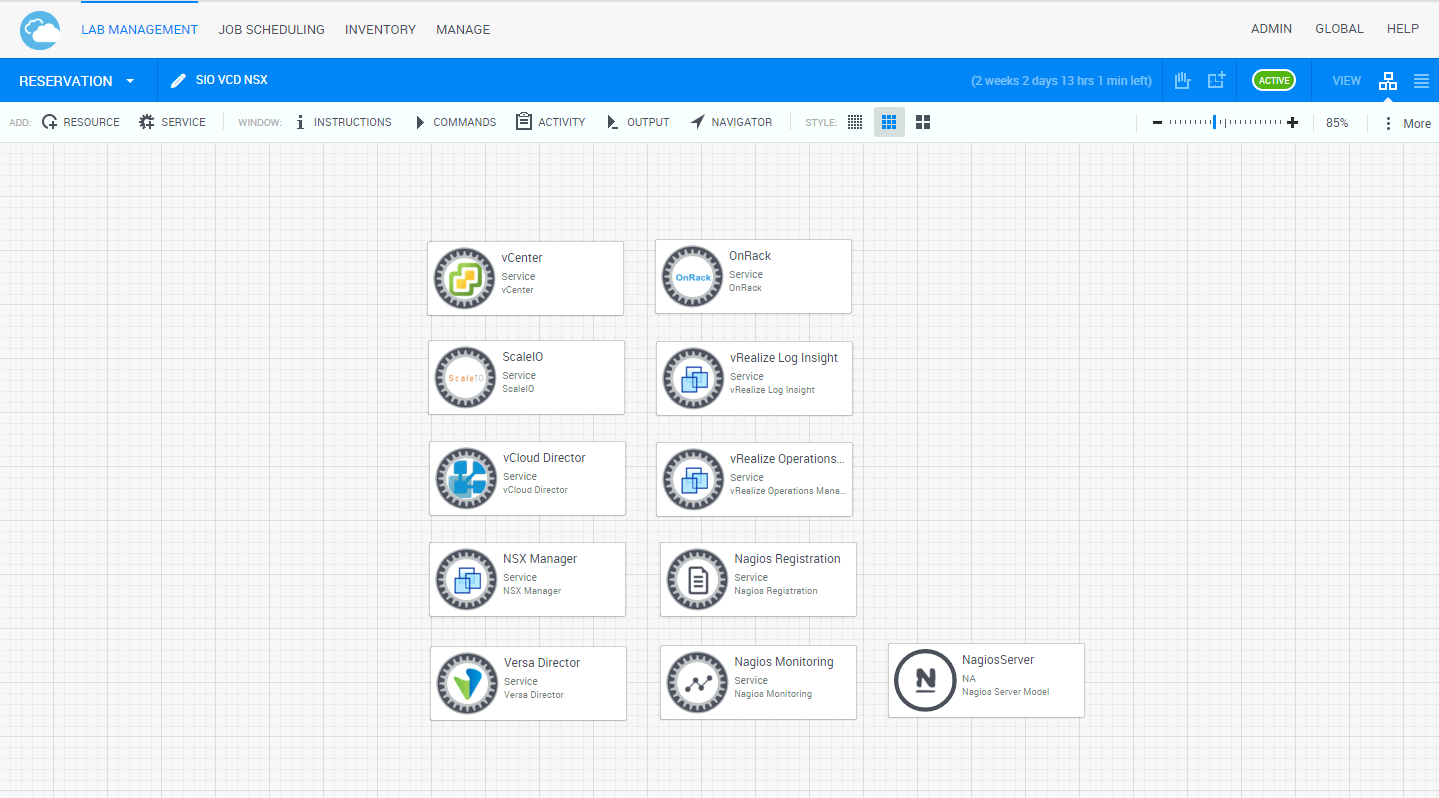
Go to Environments section and select a sandbox:



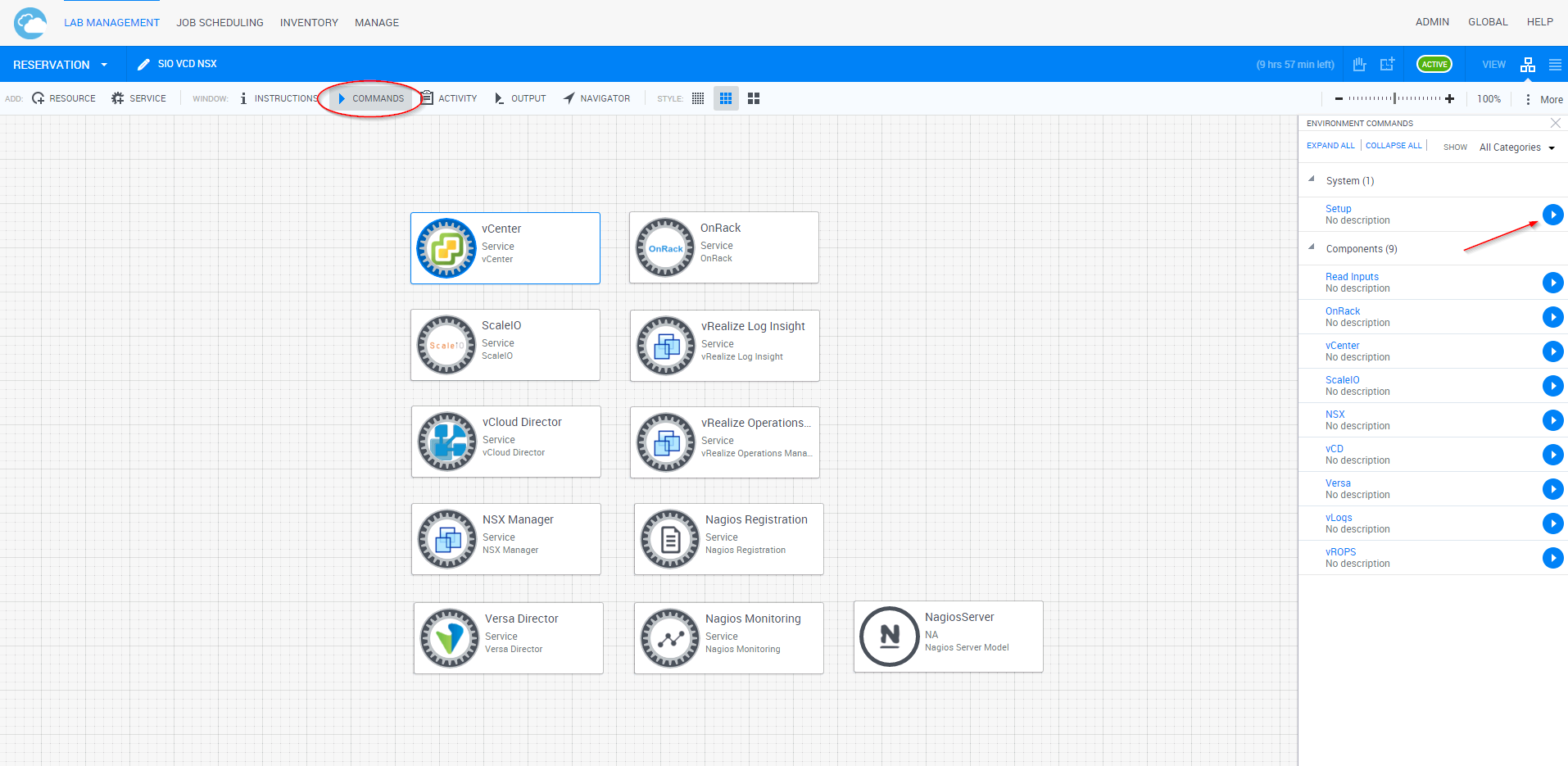
Reserve a sandbox:



Reserved sandbox:



Executing Setup function to deploy NFV:



## Importing the NFV Package

The NFV Environment and Services need to be imported to the CloudShell VM.

### Installing the NFV Solution package on the CloudShell VM

After setting the CloudShell license, log in to the CloudShell desktop. Obtain NFVInstall.zip and extract it locally on the CloudShell desktop. It contains the following files:

* compute-shell.zip
* onrack-shell.zip
* site\_manager\_shell.zip
* NFV Build Environment.zip
* Python site packages.zip
* worker.py
* INSTALL.cmd

Double-click INSTALL.cmd. This will install dependencies to the CloudShell Python directory and automatically import all required shell and environment packages into CloudShell. At this point, the sandboxes will be available in the portal and you can enter your settings and run an autodiscovery and deployment.

When you receive an updated NFV software release, repeat this procedure with the new NFVInstall.zip. The installer will perform an in-place upgrade of all dependencies and code.

## Inputs

Inputs are divided according to component. Before starting to deploy, you must run **Set Inputs** to set the service attributes of all the components in the reservation. Most values are already set and are hidden by default. Some values are set to auto-generated IP addresses based on the pools defined in the Site Manager.

## Logs

All automation activity is logged to **C:\ProgramData\QualiSystems\Logs\<reservation id>\Orchestration.log** (summary of high-level steps) and **C:\ProgramData\QualiSystems\Logs\<reservation id>\Shells.log** (all details).

Shells.log includes:

* Names of functions called.
* Function inputs.
* Full contents of generated config files uploaded to remote machines.
* The full conversations in CLI or API calls.
* Any relevant error messages.

In the event of failures, especially those from invalid inputs, the log is the most important diagnostic tool. Command lines and API calls are reproduced in full so they can be executed manually when troubleshooting.

The ‘Attach Logs’ function zips all logs related to the current reservation and attaches logs.zip to the reservation. After running, refresh the page and download logs.zip using the paperclip icon near the upper right corner. This can help you get the relevant logs without logging into the CloudShell vm.

The ‘Clean Logs’ function moves orchestration.log and shells.log to backup files and clears them. This can be useful to use before starting a new run.

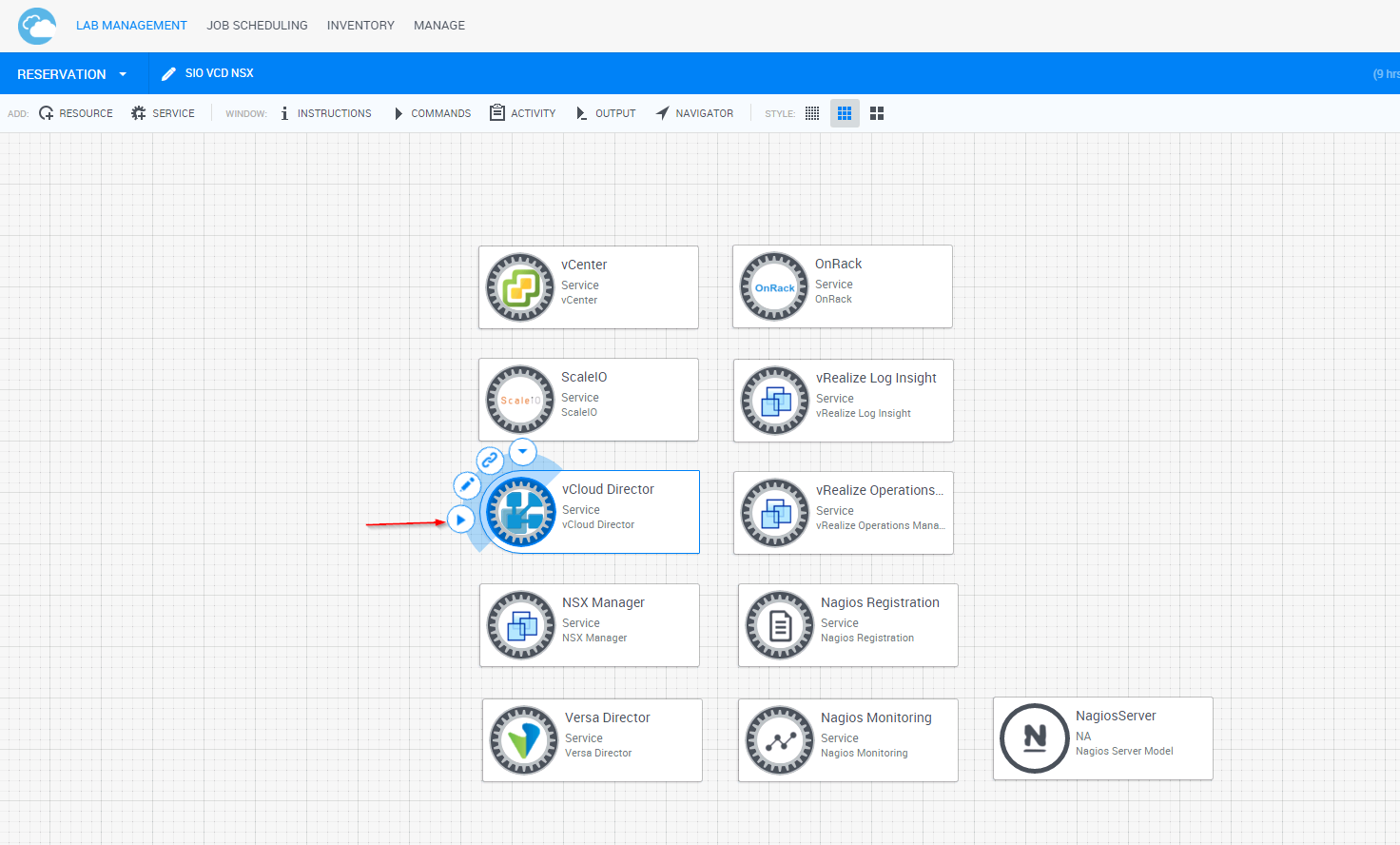
## Executing in steps

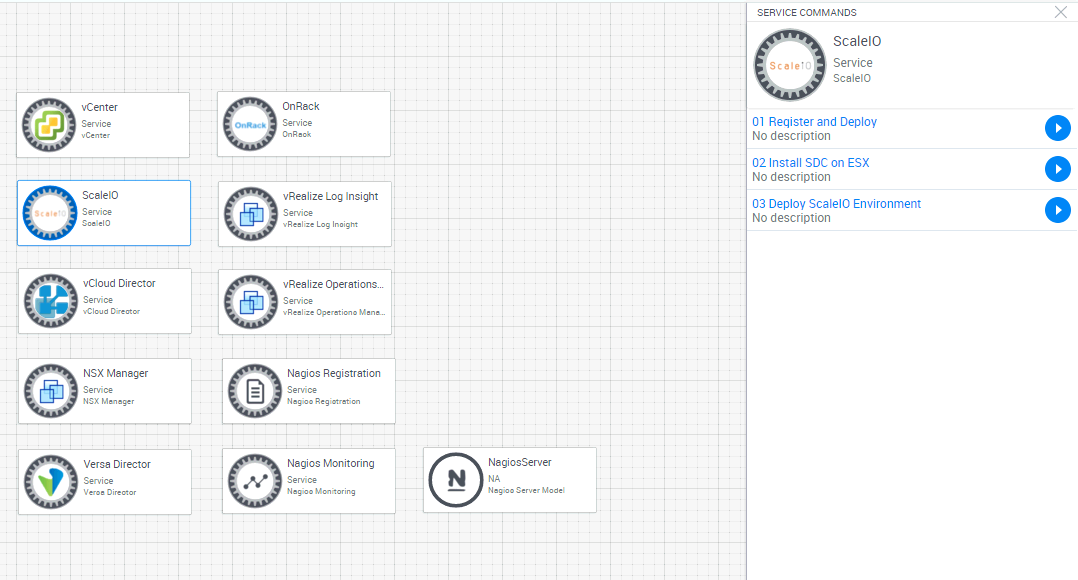
The NFV deployment is mainly intended to be run by a single monolithic orchestration function Setup.

You can run a custom subset of steps using ‘Run Orchestration Steps’, specifying a list of step numbers to include and/or exclude. This can be helpful when you need to skip steps, resume an interrupted deployment, or repeat some orchestration steps after changing inputs. If you know you need to make custom manual adjustments in the middle of the deployment, you can program the orchestration to perform only the steps leading up to that point, then later run the rest, without having to manually start and stop or start from the beginning.

To get the step numbers, run ‘Generate Orchestration Steps File’, refresh the page, and download ‘steps.csv’ using the paperclip icon near the upper right. The same steps information is also written to the Output window.

You can also select one of the components in the diagram and execute one/several steps of it from there:

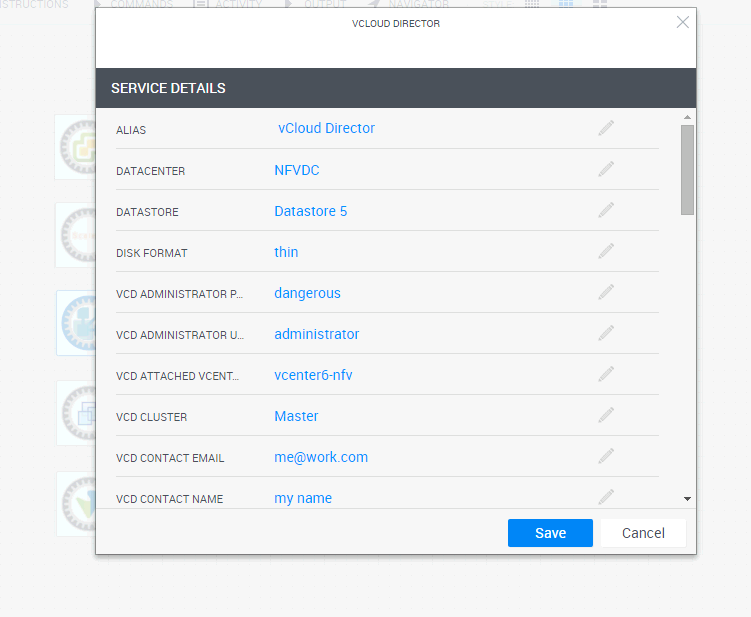




Functions take their inputs from the component’s attributes, which can be edited directly in the reservation. For example, if an input was found to be incorrect, rather than rerunning the entire deployment from the beginning, it is easier to change the input, run the function from the portal, and check the result.

Editing an attribute:





# Compatibility

## CloudShell compatibility

Supported versions:

CloudShell 7.1 Patch 1

vCenter 6.0.0-2800571

ScaleIO 2.0.0.2

vCD 8.0.0-3017494

NSX 6.2.4-4292526

vLog Insight 3.0.0-3021606

vRealize Automation 7.1.0.710-4270058

vROPS 6.1.0.3038036\_OVF10

Versa

Analytics 2dfd2b0-15.2R2

Director 11f7d34-15.2R2

Flex-vnf 15.2R2

Nagios-Core 4.1.1

# Technical Guide

The NFV deployment is subdivided into workflows and shell functions. A top-level Setup workflow function executes smaller workflows, which execute shell functions.

## OnRack Discovery Workflow

Create an OnRackShell resource and set the IP and credentials. Then reserve the environment ‘OnRack Discovery’. This environment will automatically load all OnRackShell resources and run an auto-load operation. The latest server inventory from OnRack will be synced to CloudShell resources.

## Main Workflow

The NFV deployment is sequential. The top-level Setup function calls are:

* Copy Prerequisites
* Copy Inputs
* OnRack deploy ESX
* vCenter deployment and configuration (datacenter, clusters, vds)
* ScaleIO deployment and configuration
* NSX deployment and configuration
* vCD deployment and configuration (if present)
* vROPS deployment (configuration is optional)
* vLog Insight (configuration is optional)
* Versa deployment and configuration (if present)
* Nagios deployment and configuration (if present)
* vRealize Automation deployment (if present)

## Shells & Functions

|  |  |  |
| --- | --- | --- |
| **Shell** | **Function** | **Description** |
| OnRack | Populate Resources | Runs a discovery to get all of the available compute resources and adds them to the CloudShell inventory + active build environment |
| OnRack | Deploy ESXs | Runs an ESX OS deploy for each of the detected hosts |
| vCenter | 01 Deploy vCenter | Deploys a vCenter appliance on a predefined host |
| vCenter | 02 Create Infrastructure | Create datacenter, clusters, assign hosts |
| vCenter | 03 Create VDS | Create all the required vds and assigns the vmnics as well as vmkernel IPs |
| ScaleIO | 01 Install SDC | Install the SDC component on all ESXs that will be part of SIO deployment and reboots them. |
| ScaleIO | 02 Deploy ScaleIOVMs | Deploy SVM (ScaleIO VM) on each ESX that will contribute storage for the SIO system. |
| ScaleIO | 03 Configure ScaleIOVMs | Configure each SVM to its specific role, configure the SIO system and create a ESX Storage. |
| NSX Manager | 01 Deploy NSX | Deploy VMware NSX appliance from OVA |
| NSX Manager | 02 Set NSX SSO | Connect to vCenter SSO server |
| NSX Manager | 03 Set vCenter Details in NSX | Connect to vCenter |
| NSX Manager | 04 Add NSX Admin User | Add enterprise admin role |
| NSX Manager | 05 Create IP Pools | Define a pool of IP addresses for use by appliances deployed by NSX |
| NSX Manager | 07 Create VXLAN Segment | Create a range of VXLAN segment |
| NSX Manager | 08 Install VIBs on ESXi Servers | Install VXLAN VIB drivers on ESXi hosts |
| NSX Manager | 09 Configure Host VTEP | Configure VXLAN tunnel endpoints on ESXi hosts |
| NSX Manager | 10 Create Transport Zone | Create transport zone, aka scope |
| NSX Manager | 11 Create Logical Switch | Create logical switch, aka virtual wire |
| NSX Manager | 06 Deploy NSX Controller | Deploy NSX controller |
| vCloud Director | 01 Deploy vCD | Deploy CentOS for vCD |
| vCloud Director | 02 Prepare vCD Database | Creates a db in CloudShell’s sql server |
| vCloud Director | 03 Create SSL Certificates For vCD | Creates ssl certificates for vCD |
| vCloud Director | 04 Install and Configure vCD | Installs vCD and configures the db |
| vCloud Director | 05 vCD Setup | Runs the vCD wizard |
| vRealize Log Insight | 01 Deploy vLog | Deploy vLog Insight OVA |
| vRealize Log Insight | 02 Run Start-up Configuration Wizard | Configure DNS, SMTP, license, NTP |
| vRealize Log Insight | 03 Install vLog Patch PAK file | Apply patch PAK file |
| vRealize Log Insight | 04 vCenter Registration | Register vCenter server |
| vRealize Log Insight | 05 vROPS Registration | Register vRealize Operations Manager server |
| vRealize Operations Manager | 01 Deploy vROPS | Deploy vRealize Operations Manager OVA |
| vRealize Operations Manager | 02 Run Start-up Configuration Wizard | New installation, set admin password, set license mode, set NTP |
| vRealize Operations Manager | 03 Start vROPS Service | Start service |
| vRealize Operations Manager | 04 Enter vROPS License | Accept EULA and enter license |
| vRealize Operations Manager | 05 Register vROPS With vCenter | Register vCenter server, enter vCenter monitoring settings |
| Nagios Monitoring | 01 Deploy Nagios | Deploy Nagios OVA |
| Nagios Monitoring | 02 Configure Nagios | Configures Nagios to support ESX monitoring |
| Nagios Monitoring | Enable | Starts the monitoring |
| Nagios Monitoring | Disable | Stops the monitoring |
| Nagios Registration | RegisterResources | Adds a resource to the Nagios monitoring |
| Nagios Registration | RemoveResources | Removes a resource from the Nagios monitoring |
| Versa | 02 Deploy Analytics | Deploy Versa Analytics OVA |
| Versa | 04 Deploy Branch 1 | Deploy Versa Branch OVA |
| Versa | 05 Deploy Branch 2 | Deploy Versa Branch OVA |
| Versa | 03 Deploy Controller | Deploy Versa Controller OVA |
| Versa | 01 Deploy Director | Deploy Versa Director OVA |
| Versa | 06 Post-deployment Configurations | Configures network for the different versa components and basic cli commands |
| Versa | 07 Organization Administration | Create Provider and Customer Organizations |
| Versa | 08 Create Controller Profile | Create the Controller profile |
| Versa | 09 Controller Administration | Assigning the Controller to the Customer |
| Versa | 10 Organization Configurations | Create Organization specific profiles |
| Versa | 11 Create Analytics Profile | Adding the Analytics to the Director context |
| Versa | 12 Create IPSec Profiles | Create IPSec profiles |
| Versa | 13 Branch Administration | Adding Branch VMs to the inventory |
| Versa | 14 Creating Branch Template | Adding Branch Staging templates |
| Versa | 15 Updating Branch Templates | Adding Branch PostStaging templates |
| Versa | 16 Branches Startup Configurations | Config the Branch VMs with basic IPSec setting to begin Versa Director flow |
| vRealize Automation | 01 Deploy | Set the desire interfaces of the Brocade switch to the desired VLAN ID with access mode. |

## vCenter Configuration

This section describes how the vCenter is being deployed & configured in the NFV rack.

There are 4 steps in the vCenter orchestration process,

* Deployment of the vCenter Appliance
* Creating vCenter hierarchy of Datacenters & Clusters
* Adding ESXi servers to their clusters
* Creating VDS switches

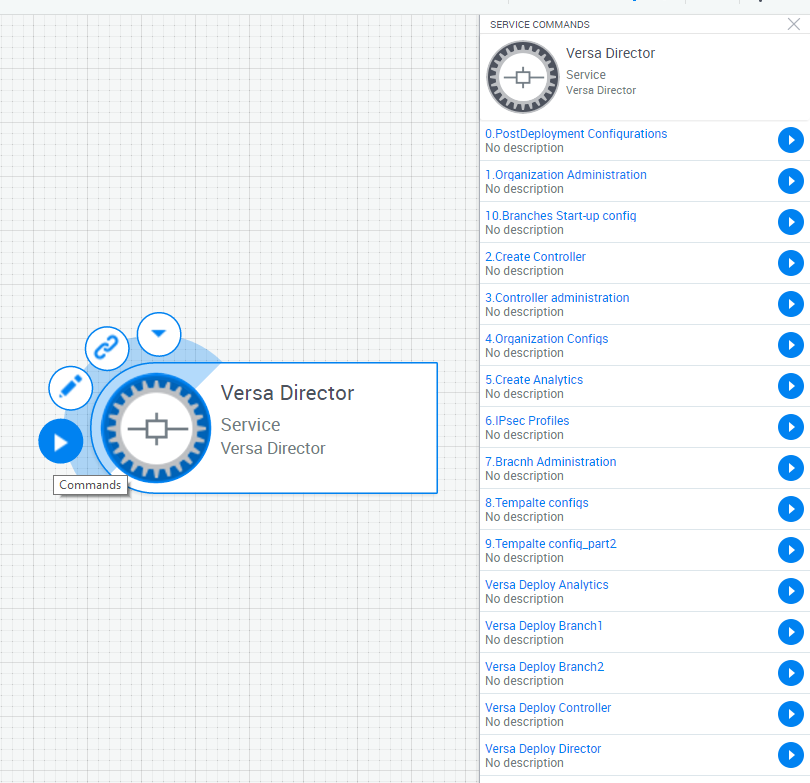
### vCenter Configuration breakdown

1. Deployment of the vCenter appliance using ovftools, the vCenter will be deployed on the master ESXi, using default network and storage settings. (“VM Network”/”datastore1”)
2. Using Python pyvmomi to create DC and up to 5 Clusters, depends on the inputs from the excel sheet.
3. Adding the ESXi servers to their relevant cluster using pyvmomi
4. Creating & assigning ESXi to each VDS, up to a maximum of 4 VDS. Each VDS will be created with an additional portgroup corresponding to its role. I.e. Data VDS will have DataPortgroup.

## Versa configuration & workflow

This section describes the deployment & configuration of the Versa components

Overview of the Versa Service as it is on CloudShell canvas



The main service is the “Versa Director” which have 5 deploy commands and 11 configuration steps.

### Versa Configuration breakdown

1. Deployment

* Scripts: Versa Deploy Director, Versa Deploy Controller, Versa Deploy Analytics, Versa Deploy Branch1, Versa Deploy Branch2
  + Versa Director, Analytics, Controller, Branch 1 & Branch 2
  + Deployment also include basic Ubuntu OS network configurations, VM configuration to add additional vNICs and powering on the VMs.

\*\*Network configuration is being done using PowerCLI “Invoke-Script” cmdlet

1. Post deployment configurations

* Script: 0.PostDeployment Configurations
  + Configuring Versa Controller, Branch1 & Branch2 using Versa CLI to use the static management IP. (Default behavior is DHCP), rewriting the Ubuntu OS network config file & rebooting the VMs
  + Restarting the Analytics Database for new deployment and doing basic log-gathering configs.
  + Disabling 2 way authentication on the Versa Director

1. Organization Configuration

* Scripts: 1.Organization Administrator, 4.Organization Configs
  + Setting up 2 organizations, 1 for the ‘Customer’ and one for a ‘Provider’
  + Configuring basic network related settings for each organization

1. Controller Configuration

* Scripts: 2.Create Controller, 3.Controller Administrator
  + Adding the deployed Controller to the Director context
  + Associating the Controller to an organization
  + Configuring the network interfaces & IP addresses for the Controller NICs

1. Analytics

* Script: 5.Create Analytics
  + Adding the Analytics VM to the Director context
  + Associating the Analytics to an organization
  + Adding log templates

1. IPSec Profiles

* Script: 6.IPsec Profiles
  + Adding 3 IPSec profiles to the Controller, pre-staging, staging & poststaging

1. Branch Configuration

* Script: 7.Branch Administration
  + Adding the Branch VMs to the Director inventory
  + Creating the basic templates for staging & poststaging
  + Creating device groups of the branches for the templates

1. Template Configuration

* Scripts: 8.Template configs, 9.Template config\_part2
  + Adding network settings, interfaces configs, IPSec profiles, name and role to the staging and poststaging templates
* Steps 3-9 are being done using the Versa Director REST API

1. Branches initializing

* Script: 10.Branches Start-up config
  + Sending basic IPSec information to connect to the Controller
  + Suppling each Branch VM with its own unique name & serial to be recognized by the Director

\*\* The commands are being sent using SSH.

## Nagios

The Nagios Shell is in charge of everything Nagios related;

* Deployment
* Network configuration
* Commands to collect logs
* Group based log collection
* Displaying the information on the CloudShell Portal

### Nagios relationship with CloudShell

Nagios is retrieving information from the ESXi servers using scripts, this information is displayed in the Nagios web portal.

CloudShell pulls information from the Nagios and display it in user-friendly way in the CloudShell Portal.

### Quick overview on Nagios component & how-to

The Nagios Shell is composed from 3 main components,

* Nagios Server
  + A Resource representing the actual Nagios machine
  + The resource also contains group, template & other general Nagios information
* Nagios Monitor Service
  + The Monitor Service is in charge of periodically checking and updating the information from the Nagios Server to the resources that we want to monitor
  + There’s a main excel sheet that the monitor service use for knowing which Nagios information to which CloudShell resource & attribute to update
* Nagios Registration Service
  + The Nagios Registration Service is in charge of ‘registering’ new VMs or physical machine to the Nagios server to monitor
  + Machines can be added to a predefined Nagios group or template

For more information about Nagios, please refer to the Nagios documentation

https://support.qualisystems.com/entries/50010383?Solution%20Packs/Nagios/6.3%20CloudShell%20Integration%20with%20Nagios%20Solution%20Pack%20v1.4.pdf

## vROPS

The vROPS Shell will deploy & configure the vROPS VM, as well as connect to it to the vCenter, to achieve that, we have several steps;

1. Deployment

* Script: vrops\_deploy
  + Deployment of the vROPS VM using ovftools, as well as configuring network settings, time zone and deployment location

1. Initial setup wizard & License

* Scripts: vrops\_configure1, vrops\_configure2, vrops\_configure3
  + Completing the first time installation wizard, setting admin password, NTP servers & cluster information
  + Starting the vROPS service
  + Configuring the vROPS license

1. Configuring vROPS with vCenter

* Script: vrops\_configure4
  + Adding the vCenter information to the vROPS system

## vLog Insight

The vLog Shell will deploy, configure & connect the vLog to both vROPS and vCenter for log collection.

1. Deployment

* Script: vra\_log\_insight\_deploy
  + Deployment of the vLog VM using ovftools, as well as configuring network settings and deployment location

1. Initial setup wizard & License

* Scripts: vra\_log\_insight\_configure1
  + Completing the first time installation wizard, setting admin password, NTP servers & SMTP information
  + Configuring the vLog license

1. Configuring vLog with vCenter

* Script: vra\_log\_insight\_configure3
  + Adding the vCenter information to the vLog system

1. Configuring vROPS info

* Script: vra\_log\_insight\_configure4
  + Adding vROPS information to the vLog system

## NSX

The NSX Shell will deploy the NSX Manager, and with REST API will configure the Manager to deploy a Controller and configure the NSX appliance.

1. Deployment

* Script: nsx\_deploy
  + Deployment of the NSX Manager appliance, as well as configuring the network setting

1. Configuring vCenter connection

* Scripts: nsx\_configure1, nsx\_configure2
  + Configuring the vCenter SSO information and vCenter info on the NSX Manager

1. Configuring NSX network related settings

* Scripts: nsx\_configure4, nsx\_configure6, nsx\_configure7, nsx\_configure8, nsx\_configure9
  + Create IP pools for each MGMT, Edge & Compute VDS
  + Create VXVLAN segments for the IP pools
  + Create transport zone between the segments

1. Adding Admin group to manage the NSX

* Script: nsx\_configure3
  + Add the desire group name to the NSX Administrators

1. Deploying NSX Controller & configuring ESXi hosts

* Scripts: nsx\_configure5, nsx\_configure7
  + Deploy NSX Controller using NSX Manager REST API
  + Make the necessary vibs installation on each ESXi in the NSX scope

# Relevant HW/SW compatibility

All of the above shells and workflows were tested and validated with:

* 5 Dell servers
* vCenter 6
* CloudShell 7.1 Patch 1

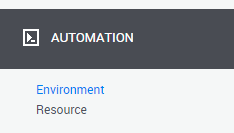
# Managing scripts and drivers

All the scripts and drivers are managed by system administrators.

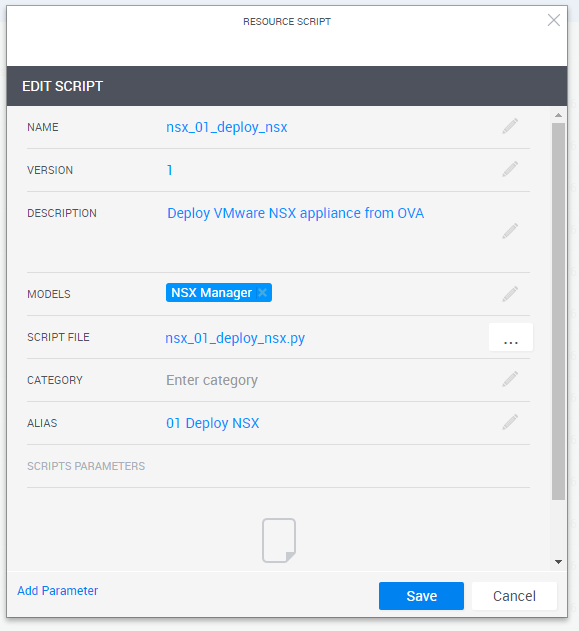
Once logged in as admin, you’ll be able to go to the Manage page:



There, under the Automation sub-menu on the left, you’ll be able to manage both the environment drivers/scripts and the resource drivers/scripts.



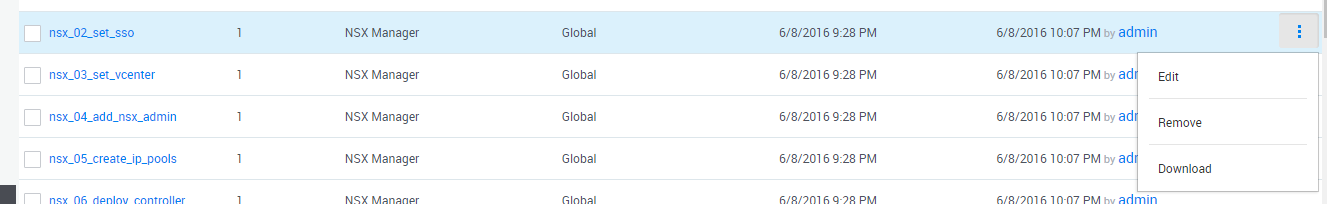
For each script/driver, you can open the properties page (either by clicking on the script name, or by the Edit option in the context menu on the right side of each script). This allows you to edit their name/alias/description/version:



**Note:** the environment driver is using the names of the scripts to execute the commands in a correct sequence. You should not change the name of the scripts without updating the environment driver as well.  
Changing the alias, description, category and version does not affect the automation.

## Downloading a script/driver file from CloudShell

Each row in the scripts/drivers window has a context menu on the right side, that allows you to download the script/driver:



## Updating a script file

If you got a new version of a script file, or would like to update one of the script files, open the properties windows of that script as shown above, then press on the browse button in the “Script file” field and select a .py or .zip file.

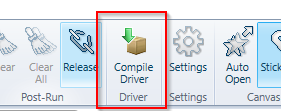
When done, click on the Save button to upload the new file.

## Updating a driver

To create a new version of a driver, you’ll need access to the project files to be able to open, modify and compile the project in CloudShell Authoring.

The latest projects and scripts are available in a private github repository, and you’ll need to get access to this repository first.

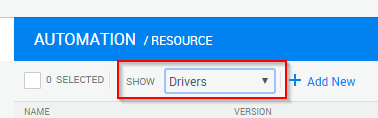
Once you got access to such a driver project that is not python based, you’ll be able to open it in CloudShell Authoring, and when done modifying the driver, you’ll be able to compile it:



This will create a new folder with the compiled project (a .dll or .exe file, depends on the project type).

Once the new version of the driver was compiled, you’ll be able to upload it to the portal.

Make sure that you first show the resource/environment drivers list by selecting “Drivers” in the “Show” drop down:



# Post-Deployment How-to and extra steps

## Using the ScaleIO web plugin

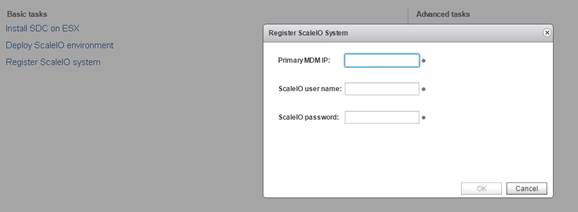
In order to use the ScaleIO web plugin, you first need to register manually the deployed ScaleIO environment with the vCenter.

To register the deployed ScaleIO environment with the installed vCenter ScaleIO Plugin, a user needs go to the vCenter web client, and do the following steps:

1. Login to the vCenter Web Client
2. Open the ScaleIO Plugin



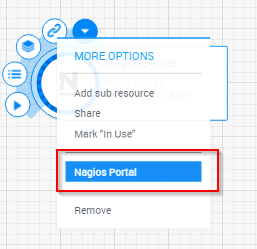
1. Select “Register ScaleIO System”
2. Fill in the relevant info (Primary MDM IP = Info from Excel sheet \ vCenter view of “SIOVM-Primary”, ScaleIO user name = ‘admin’, ScaleIO password = Info from Excel)



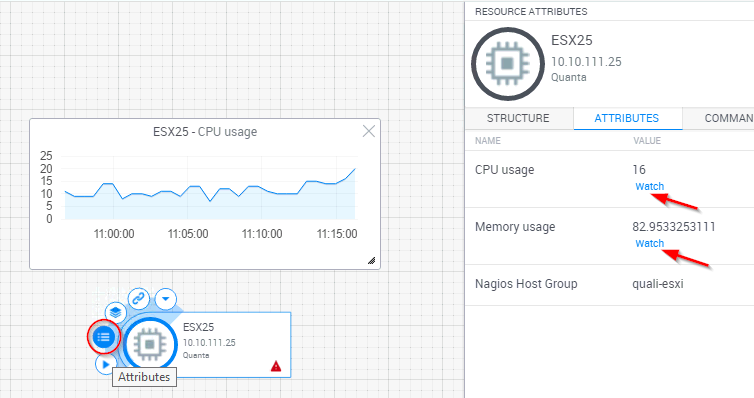
## Nagios

* To open the Nagios Server website, you can browse to http://<Nagios\_IP>/nagios

The <Nagios\_IP> can be taken from the excel or from the vCenter view of the Nagios VM.   
The credentials are: Username: ‘nagiosadmin’, Password:’Password1’.

You can also use the shortcut available on the ‘NagiosServer’ resource:  


* To view the live status of the attributes we monitor using Nagios, go to the CloudShell Portal and select the ESX you wish to see, opening the “Attributes” tab. From there clicking on the ‘watch’ link will open the monitoring window.



Note: the times on the graph are displayed in UTC (no matter which timezone was selected by the user), but displays the last 60 minutes always

## vLog Insight, vRA, vROPS & NSX

* As part of the current automation workflow, only the deployment of the above is being done (NSX configuration continues until and including step 4).
* To fully utilize the above, a user must login to each corresponding IP address web and configure the VM.

# Known Gaps

|  |  |
| --- | --- |
| Gap | Description |
| Deployment & VM configuration only supported on vCenter version 6. | Deployment, Network, Storage and other VM related tasks are supported only on vCenter environment. |
| Order of the orchestration cannot be changed | The order of each Shell, i.e. deploying the NSX before SIO is not supported, due to the fact that each step depends on the step before, for example, vROPS require SIO storage to be present for the deployment. |
|  |  |

## Versa Gaps

|  |  |
| --- | --- |
| Gap | Description |
| Deployment & configuration only support 2 Branch VMs and a single Controller, Analytics and Director VMs | The Versa Shell support a single fixed use case of 2 Branches as VMs and single deployment of other Versa components.  Different uses cases are not support at the moment. |
|  |  |

## vCenter Gaps

|  |  |
| --- | --- |
| Gap | Description |
| Configuring the vCenter with more than 4 VDS is not supported | At the moment, only up to 4 VDS are supported in the automation workflow. |
| Configuring the vCenter with more than 5 Clusters is not supported | At the moment, only up to 5 Clusters are supported in the automation workflow. |
|  |  |

## Nagios Gaps

|  |  |
| --- | --- |
| Gap | Description |
| Only Memory usage is operational | You’re only able to see a graph on the Memory usage of the different ESXi servers |
|  |  |

## ScaleIO Gaps

|  |  |
| --- | --- |
| Gap | Description |
| Configuration only support using RDM as the method to attach storage to the SVMs | Unable to use ESX with only 1 local disk or an ESX with an already configured Datastore, due to inability to configure VMDK. |
| Required Faultsets number must match the amount of SDS in the SIO system | For Faultset use, there must be an exact match in the number of SDS VMs and the number of desired Faultsets. |
| ScaleIO plug-in is not deployed | Due to a known issue, the plug-in is not being deployed. |

# Known Issues

|  |  |
| --- | --- |
| Issue | Description |
| When Running Versa Automation step, on rare occasion it might fail on step 2 with an “Controller Out of Sync” error. | Due to unknown issue, the controller sometimes shows as “Out of Sync” and is unable to process any additional commands. The Workaround for it, until Versa will release a fix, is to delete the Controller from the Versa Director WebGUI (Administratorion > Connectors - > Organization)    And from the Resource Pool under the same subtree    Once done, re-run step 2-10. Ignore “Error 408 ‘data already exists’” |
| vRA Automation appliance is left powered off | There may be a compatibility issue on some hosts that blocks the VM from powering on.  It may be possible to respond to this prompt programmatically in the future.  For now, manually power on the VM and answer Yes to the prompt when it appears:cid:image001.png@01D23C5E.68548E50 |

# Changes from the previous code release

|  |  |
| --- | --- |
| Feature | Description |
|  |  |

# Legal Notice

Information in this document is subject to change without notice. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of QualiSystems Ltd.

QualiSystems may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except if expressly provided in any written license agreement from QualiSystems, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

QualiSystems, CloudShell, CloudShell Authoring, CloudShell Resource Manager, CloudShell Remote Runner, CloudShell Runtime, CloudShell Monitor, CloudShell Spy, CloudShell Portal, the QualiSystems logo, the CloudShell logo, and the CloudShell application logos, and all other QualiSystems product names and logos are trademarks or registered trademarks of QualiSystems Ltd. The absence of a trademark from this list does not constitute a waiver of QualiSystems intellectual property rights concerning that trademark.

All other trademarks, brand and product names are property of their respective holders.

© 2016 QualiSystems Ltd. All rights reserved.