Univa Corporation

Navops Suite

Navops Launch Documentation

Author:
Univa Engineering

Version:
0.6

April 19, 2016

Copyright ©2012–2016 Univa Corporation. All rights reserved.

Contents

1 Navops Launch Preview Documentation

1.1 Welcome to Navops Launch!

Thanks for trying out our Navops Launch Preview. Navops Launch is probably the easiest way to install and configure a Kubernetes cluster that contains, Kubernetes, and Docker all running on Fedora Atomic host machines.

Navops Launch is designed to install a **physical**, **virtual machine** or **cloud** based Kubernetes cluster. If you have a group of spare servers that can network boot or you have a machine where you can install Docker Toolbox™ then it is very simple to create a complete Kubernetes cluster using Navops Launch. Also if you want to try out Navops Launch by utilizing resources on Google Cloud Platform or Amazon Web Services then it is straight forward to do that as well.

1.2 Prerequisites

1.2.1 Machine with Docker Installed

Before you download Navops Launch from our website (http://navops.io) you will need either a Linux machine with CentOS 7 with Docker installed on the machine or a Mac OS X machine with Docker Toolbox installed.

- Instructions for installing Docker on CentOS 7
- Instructions for installing Docker Toolbox on Mac OS X

If you already have a machine with Docker installed please make sure it is at least Docker 1.7. There are issues running Navops Launch on older versions of Docker.

We recommend that the machine have at least 1GB of memory and 5GB of available storage for the Navops Launch containers. Additional resources will be required if you wish to run local Kubernetes worker VMs.

1.2.2 Networking

You can skip this section if you are planning to install Navops Launch on cloud resources.

Creating a Kubernetes cluster using Navops Launch inside your existing network infrastructure can be a complex operation. The Navops Launch software is designed to automatically detect nodes that PXE boot and request an IP using DHCP. These operations may conflict with existing DHCP services in your corporate network. There are two ways to configure the node discovery in Navops Launch they are:

- **DHCP Relay**: Navops Launch does not run a DHCP server. When a node is added to the Navops Launch cluster the MAC address to Name mapping is stored in your corporate DHCP server or Firewall/DNS/DHCP server. In this configuration the IP and name of the machine is registered at the corporate server.
- **DHCP Server**: Navops Launch system runs a DHCP server and when a node is added to the cluster the DHCP Server in Navops provides the IP and registers a MAC to name mapping. If a DHCP server exists in your corporate network the Navops Launch DHCP server will conflict unless Navops is behind a NAT/Router, on a VLAN or in a different subnet.

It is very important to understand that there is typically only one DHCP server in a network segment and routers do not normally propagate DHCP requests unless the corporate network Admins have configured the routers to relay. Please see the networking diagrams in the Navops Launch Network Configuration section to understand the difference between DHCP Server mode and DHCP Relay Mode.

We highly recommend that you build your Kubernetes cluster in a subnet where you can run a DHCP Server, or a VLAN where you can run a DHCP server or by building your Kubernetes cluster behind a firewall/router (even a home router will work). This avoids complications with DHCP and DNS that may occur when using the DHCP Relay installation.

1.2.3 Downloading

To download Navops Launch visit navops.io and click the Download button. After you have filled out the web form and answered some questions you will receive an email from navops.io (be sure to use a real email address). Click on or copy and paste the single line command into a terminal window to begin the download and installation process of the Navops Launch Docker container. The terminal session to which you paste needs to be enabled to run Docker commands.

Please proceed to the following sections for detailed installation instructions:

- Installing Navops Launch on CentOS 7 with Docker.
- Installing Navops Launch on Google Cloud Platform.
- Installing Navops Launch on Amazon Web Services.
- Installing Navops Launch on Mac OS X with Docker Toolbox

1.3 Installing Navops Launch on CentOS 7 with Docker

Here is a schematic picture of the deployment model of Navops for a bare metal infrastructure. You start with a CentOS 7 node with Docker already installed on it. On that you will install the Navops installer via a command which has been sent to you via e-mail. Once the installer is setup you can use the Navops web-UI or CLI to add a Kubernetes master node and then any number of worker nodes. These nodes are bare metal nodes which you will have to PXE-boot to start the provisioning and operating system installation.

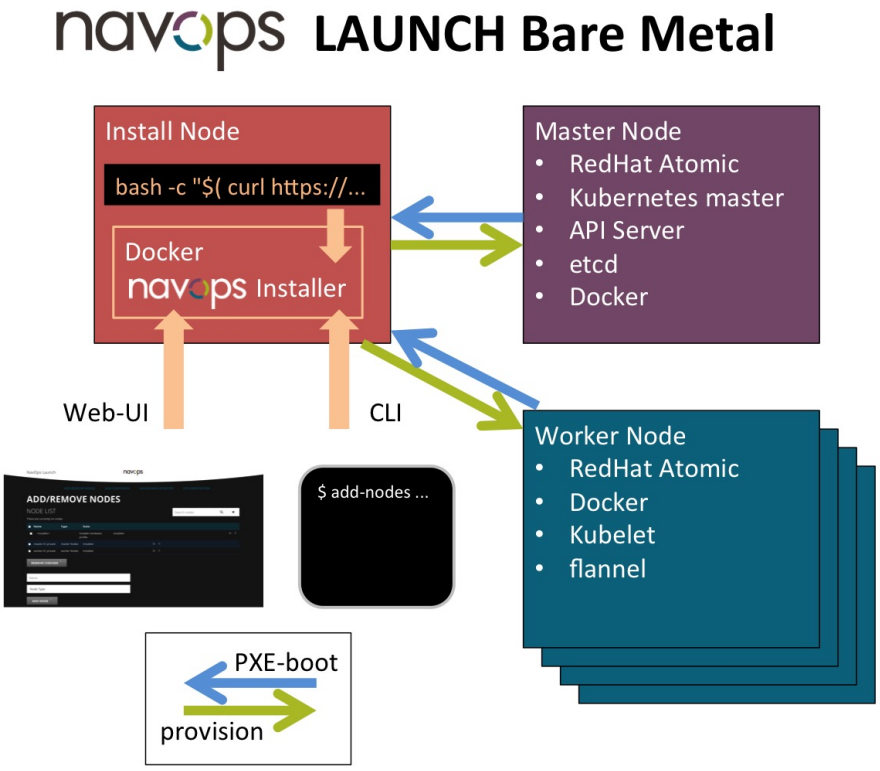

Figure 1:

If you do not have a CentOS 7 node with Docker already installed then please first follow the steps in section Instructions for Installing CentOS 7 with Docker.

With Docker on a CentOS 7 node available, login as root on that node, open a terminal session and run the command provided in the e-mail that was sent to you after registering for the Navops download. It should look something like this (all one line - line breaks only inserted for better readability):

```
[root@nucky ~]# bash -c "$( curl https://dev.navops.io?token=\
c3RhZ2luZ191c2VyOmU3MTI2MDg0NTI4MGExOTYwOGMyYjFjYzAzYjdhYTg2\&tag=\
v0.0.2-65-g7a19bfae02 | bash )"
```

It takes about 5-10 minutes to download the container and execute the installation from a 30Mbit Internet connection. Faster or slower Internet connections will result in different download and installation times. Very slow connections can take 30 minutes or more.

The command above will download the latest container packages, automatically install them and configure the system to provision a Kubernetes cluster. When the script finishes you should have several containers running on the host. You can check this with the following command:

```
[root@nucky ~]# docker ps
CONTAINER ID     IMAGE                                                    COMMAND
      CREATED          STATUS             PORTS                     NAMES
4d7dbd9b2409     registry.navops.io/unicloud:v0.0.1-1-gc929dfcb38    "/usr/sbin/init"
      58 minutes ago  Up 58 minutes                                 unicloud
44f309f35af4     registry.navops.io/repo-server                       "/run-apache.sh"
      2 days ago       Up About an hour   0.0.0.0:9000-\>80/tcp   fedora-atomic-repo
```

Once the containers have started you have a working Navops Launch system. You can proceed with section Creating a Kubernetes Cluster from the command line to create a Kubernetes cluster from the command line or with section Creating a Kubernetes cluster from the Navops web interface for doing the same from the Navops web interface.

1.4 Installing Navops Launch on Google Cloud Platform

If you do not have bare-metal resources available on-site then you can use Google Cloud platform to test the functionality and ease-of-use of Navops Launch. Below is a schematic view of how Navops Launch would install Kubernetes in such an environment.

Installing on Google Cloud Platform requires to create a project in Google Cloud Platform (or use an existing one). You will need to configure that project for use through Navops Launch so that you then can download and run the Navops Launch installation process.

1.4.1 Create a Project on Google Cloud Platform

You can skip this step if you already have a project on Google Cloud Platform which you would like to use for your Navops Launch testing.

For creating a project on Google Cloud Platform you obviously need an account there. A trial account can be create easily by going to `https://cloud.google.com/` and then follow the free trial instructions from there.

If you already have an account on Google Cloud Platform then you can use that as well, of course.

Once you have an account you need to create a project for use through Navops Launch, e.g. calling it `navops-test`. Please follow the workflow recommended for Google Cloud Platform for that. Once the project has been created please note the ID of the project as it will be used later on when installing Navops Launch.

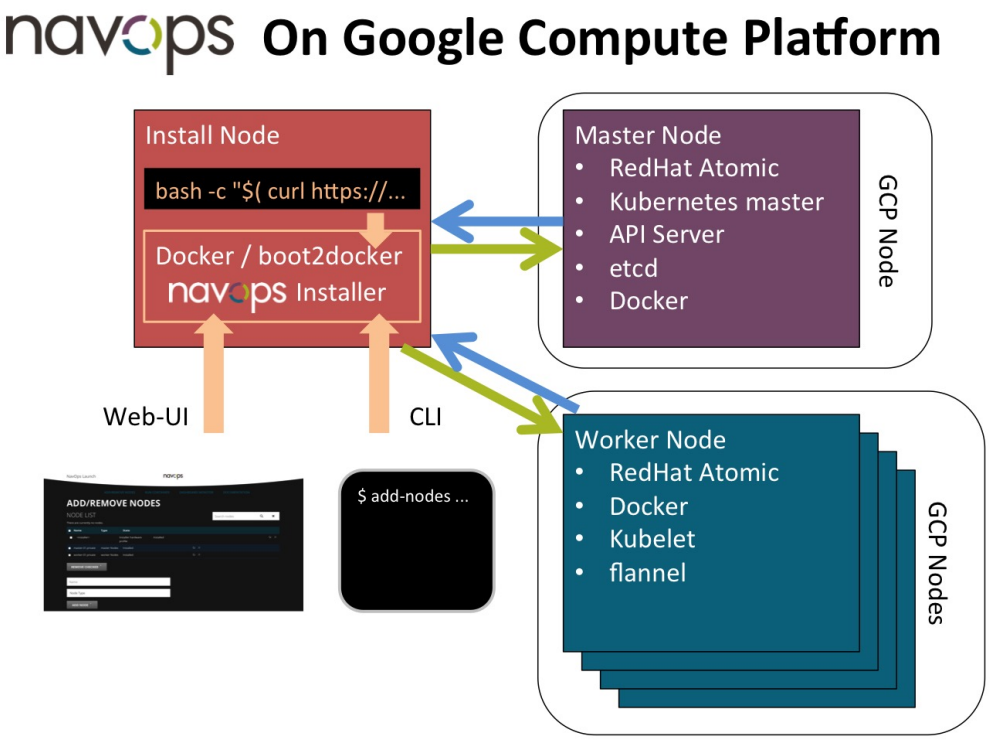
navops On Google Compute Platform
Install Node
bash -c "$(curl https://...
Docker / boot2docker
navops Installer
Web-UI
CLI
Master Node
RedHat Atomic
Kubernetes master
API Server
etcd
Docker
GCP Node
Worker Node
RedHat Atomic
Docker
Kubelet
flannel
GCP Nodes
ADD/REMOVE NODES
$ add-nodes ...

Figure 2:

1.4.2 Configure a Google Cloud Platform Project for Navops Launch

In the project you have designated for your Navops Launch test, please go to API management and enable the following APIs:

- Google Compute Engine API
- Google Identity and Access Management API
- Google Cloud Resource Manager API

Your project is now ready to be used during the next step which will install Navops Launch.

1.4.3 Downloading and Installing Navops Launch on Google Cloud Platform

In the following we will assume you are using CentOS 7 as an installer node with Docker already installed. Should that not be the case then you can follow the steps in section Instructions for Installing CentOS 7 with Docker.

Alternatively other docker-enabled environments like Docker Toolbox on Mac OS X should work as well. Note that on Mac OS X with Docker Toolbox you will not need to works with root privileges as described for a CentOS installer host below. For simplicity will want to you the Docker Quickstart Terminal for executing the below commands, however.

With Docker on a CentOS 7 node available, login as root on that node, open a terminal session and paste (do not run it yet!) the command provided in the e-mail that was sent to you after registering for the Navops download. It should look something like this (all one line - line breaks only inserted for better readability):

```
# bash -c "$( curl https://dev.navops.io?token=\
c3RhZ2luZ191c2VyOmU3MTI2MDg0NTI4MGExOTYwOGMyYjFjYzAzYjdhYTg2\&tag=\
v0.0.2-65-g7a19bfae02\&mode=2\&mode_data=<GCE_PROJECT_ID> | bash )"
```

Before running that command you have to change the placeholder `<GCE_PROJECT_ID>` to the ID of the project you have created and configured as described in the previous two sections. Assuming the project name was `navops-test` the commend you would run becomes:

```
# bash -c "$( curl https://dev.navops.io?token=\
c3RhZ2luZ191c2VyOmU3MTI2MDg0NTI4MGExOTYwOGMyYjFjYzAzYjdhYTg2\&tag=\
v0.0.2-65-g7a19bfae02\&mode=2\&mode_data=navops-test | bash )"
```

Run this command and it will kick off the Navops Launch installation process downloading and configuring multiple containers. Depending on the speed of your Internet connection this can take as little as 5 minutes or, for very slow connections, 30 minutes and more.

Towards the end of installation process a URL will be displayed and you will be asked to paste it into a web browser. After doing so you will need to login with your Google Cloud Platform account (the same with which the project used for testing was created) and a credential key will be displayed. Paste this key then into the installation terminal and hit return to conclude the installation.

You then can proceed with section Creating a Kubernetes Cluster from the command line to create a Kubernetes cluster from the command line or with section Creating a Kubernetes cluster from the Navops web interface for doing the same from the Navops web interface.

1.5 Installing Navops Launch on Amazon Web Services

1.5.1 Create an Amazon Web Services (AWS) Account

If you do not already have an AWS account you wish to use with Navops Launch, one will need to be opened and associated with valid billing information before beginning the installation. Visit https://aws.amazon.com and select `Create AWS Account` to begin the process.

1.5.2 Create an AWS User for Navops Launch

In order to manage AWS resources, the Navops Launch installation requires access to your AWS account. While any user with the appropriate access rights will suffice, we recommend that one be created specifically for Navops Launch.

This can be accomplished by logging into the AWS console using credentials that have IAM access for the given AWS account, and navigating to `Identity & Access Management`. Select `Users` from the navigation, followed by `Create New Users`. Input the desired username, and click `Create`.

Next, you will be presented with an option to download the user's security credentials. The credentials are required by the Navops Launch installer, so download them and store them in a secure location.

Granting AWS Access From the IAM users list, select the newly created user and navigate to the `Permissions` tab. Attach the policies listed below to ensure that the Navops Launch user has the necessary access rights.

- AmazonEC2FullAccess
- AmazonVPCFullAccess

1.5.3 Installing Navops Launch

Navops Launch requires a CentOS 7 host with Docker, or a Mac OS X host with Docker Machine. This host can be located in AWS, or externally with the appropriate internet access to reach the AWS API.

Installation is accomplished by issuing a single command on the Navops Launch host. Replace `<AWS_ACCESS_KEY_ID>` with the Navops Launch AWS user's access key ID (acquired when creating the user).

During the installation process, you will be prompted to input the secret access key associated with the provided access key ID, so be sure to have it available before starting the installation.

Be aware that a Navops Launch AWS VPC and security group will be created during installation.

```
# bash -c "$( curl https://dev.navops.io?token=\
    c3RhZ2luZ191c2VyOmU3MTI2MDg0NTI4MGExOTYwOGMyYjFjYzAzYjdhYTg2\&tag=\
    v0.0.2-65-g7a19bfae02\&mode=3\&mode_data=<AWS_ACCESS_KEY_ID> | bash )"
```

Once the installation has finished successfully, you can move on to the section Creating a Kubernetes Cluster from the command line (#creating-a-kubernetes-cluster-from-the-command-line) to create a Kubernetes cluster from the command line or with section Creating a Kubernetes cluster from the Navops web interface for doing the same from the Navops web interface.

1.6 Installing Navops Launch on Mac OS X with Docker Toolbox

You can furthermore test Navops on Mac OS X to get a basic understanding for how the Navops deployment automation works before you deploy onto bare metal servers. Running all components on your Mac OS X host and, as you will learn, in virtual machines is clearly not a production grade set-up. Several concurrent virtual machines can easily overload your system, especially when run on a Laptop with many other, resource consumptive applications executing at the same time. For the best experience we recommend rebooting your Mac OS X system and have as few as possible other applications running.

Docker does not run natively on Mac OS X because it is a Linux only container technology. However you can run Docker on Mac OS X using Docker Toolbox which downloads and installs a VirtualBox virtual machine to your Mac and runs Docker in a virtual machine. If you do not have Docker installed on your Mac OS X system then please follow the instructions in section Installing Docker Toolbox for Mac OS X.

Docker Toolbox contains `boot2docker` and `docker-machine`. Docker-machine provides functionality for launching and controlling Docker virtual machines on Mac OS X. You can perform several actions using `docker-machine` for more information please read the 'docker-machine' documentation

The following schematic picture shows the Navops deployment model on Mac OS X. As you can see, the Navops installer will get run in a boot2docker virtual machine and also the Kubernetes master node and worker nodes will run in virtual machines. You will need to PXE-boot those virtual machines to provision them (see the instructions in section PXE-Booting a VirtualBox VM).

1.6.1 Downloading Navops Launch on Mac OS X

Mac OS X does not natively support docker so a 'docker virtual machine' will get launched in order to use the Navops Launch image on Mac OS X. We suggest you use the following steps:

1. Open a new Mac Terminal Session Window. For simplicity we recommend you use the Docker Quickstart Terminal.
2. Cut and paste the Navops URL script line you received by email into the Terminal (i.e. a command of the form `bash -c "$( curl https://...)"`)
3. The script will download the containers and install it into Docker. If you have the Virtualbox GUI running you will see that a Navops virtual machine is created. If no Navops virtual machine is created then there is a configuration issue with your VirtualBox.
4. Go to the web Interface at the URL provided by the install script and skip to section Creating a Kubernetes cluster from the Navops web interface or continue with the next instruction for using the command-line to continue installing
5. Set docker-machine to the Navops Virtual Machine with the following command in a Terminal session:

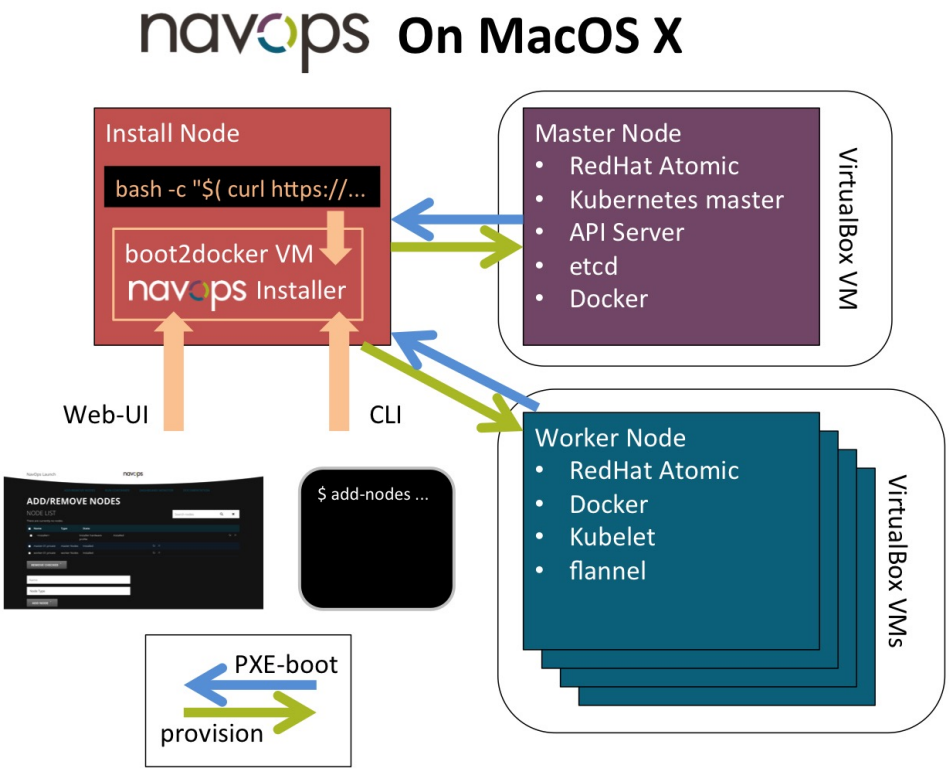
navops On MacOS X
Install Node
bash -c "$(curl https://...
boot2docker VM
navops Installer
Web-UI
CLI
Master Node
RedHat Atomic
Kubernetes master
API Server
etcd
Docker
VirtualBox VM
Worker Node
RedHat Atomic
Docker
Kubelet
flannel
VirtualBox VMs
ADD/REMOVE NODES
$ add-nodes ...
PXE-boot
provision

Figure 3:

- `# eval "$(docker-machine env Navops)"`

6. Check to see if the containers that were downloaded are running with the following command
   - `# docker ps`
7. For installing your Kubernetes cluster from the command-line follow the instructions in section Creating a Kubernetes Cluster from the command line.

1.7 Creating a Kubernetes cluster from the Navops web interface

After having installed the Navops installer (via the `bash -c "$( curl https://..."` command) a URL gets printed for the Navops web interface. Please point your browser to that URL to display the following introductory screen:

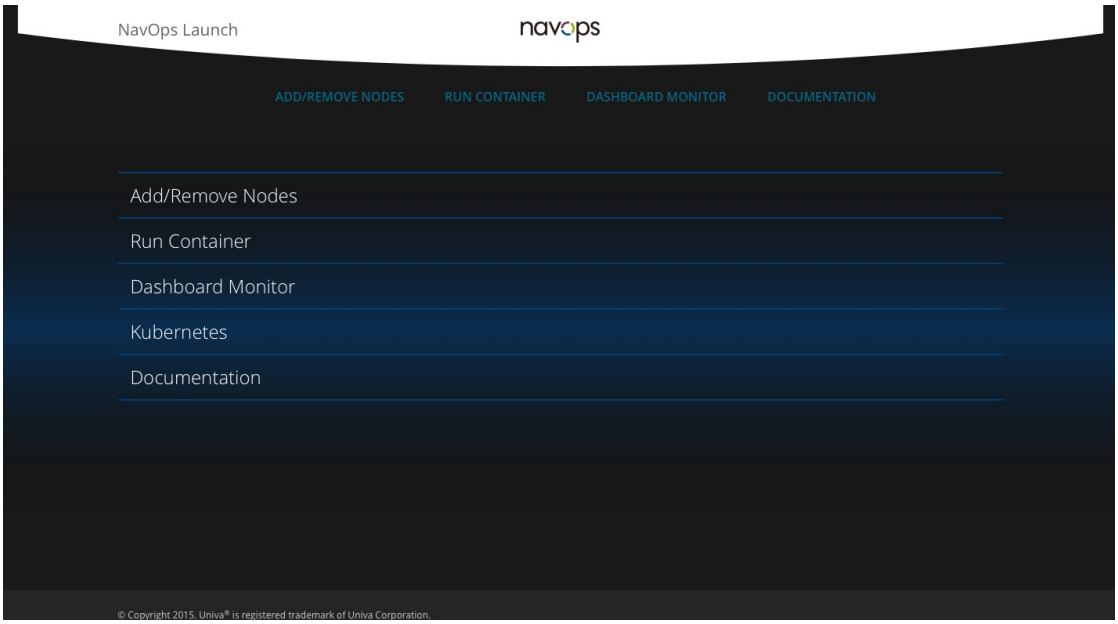

Figure 4:

The first thing you will want to do is adding nodes to build up your cluster. For that please click on 'Add/Remove Nodes'. We will first describe how adding nodes is done for the different installation cases in general before we describe the sequence of node types (master and worker) you have to add.

1.7.1 Adding Nodes for a Google Cloud Platform or AWS Installation

If you are not installing on a public cloud platform then please skip to section Adding Nodes for all Other Cases.

Adding nodes in the case where you install them on a public cloud provider is very simple. In the 'Add/Remove Nodes' dialog you just select the node type and then you click 'Add Node'. A corresponding pop-up message will get displayed and when the node is ready it will show up in the node table.

1.7.2 Adding Nodes for all Other Cases

The following steps for node installation should be followed when you install on either bare-metal nodes or on VirtualBox VMs on Mac OS X. If you are installing on a public cloud platform, however, then please use the steps described in Adding Nodes for a Google Cloud Platform or AWS Installation

If you plan to install bare-metal nodes then you will have to switch off the firewall on your installer host. Please follow the instructions in section Remvoing the Firewall for On-Premise Installation from CentOS Installer Node for this before you proceed.

In the 'Add/Remove Nodes' dialog you enter a node name, such as `master` or `worker-01`, and you select the corresponding node type then you click 'Add Node'. For cases where you want to or need to ensure a node only gets installed on a specific server you may want to select the MAC address box and enter the MAC address of that host before you add the node.

After having added the node you PXE-boot the node. You need to do this for bare-metal nodes as well as for VirtualBox VMs when installing on Mac OS X. If you are installing on Mac OS X then please refer to the description in section PXE-Booting a VirtualBox VM. You may also want to refer to section Atomic Installation Steps during PXE-Booting for hints on the bootstrap process of the RedHat Atomic OS which gets deployed.

While PXE-booting is going on, the 'Add/Remove Nodes' dialog will first display the master node as 'Discovered' and will switch to 'Installed' when installation is complete.

1.7.3 Adding a Kubernetes Master

You have to add the master node first. Depending on where you will install your nodes (Google Cloud Platform or other) you should either follow the steps in Adding Nodes for a Google Cloud Platform or AWS Installation or in Adding Nodes for all Other Cases.

This screen shot shows the corresponding situation for the Google Cloud Platform case:

And this screen shot represents the case of installing on bare-metal or on Mac OS X using VirtualBox VMs:

1.7.4 Adding a Kubernetes Worker

The next step is to add your first Kubernetes worker. For that use the 'Add/Remove Nodes' screen again. Depending on where you will install your nodes you should either follow the steps in Adding Nodes for a Google Cloud Platform or AWS Installation or in Adding Nodes for all Other Cases.

and fill in the 'Name' field with 'worker-01' and select 'Worker' for 'Node Type' as shown in the following screenshot:

This screen shot shows the corresponding situation for the Google Cloud Platform case:

And this screen shot represents the case of installing on bare-metal or on Mac OS X using VirtualBox VMs:

You can now repeat this step to add further nodes or you can skip to section Installation Summary.

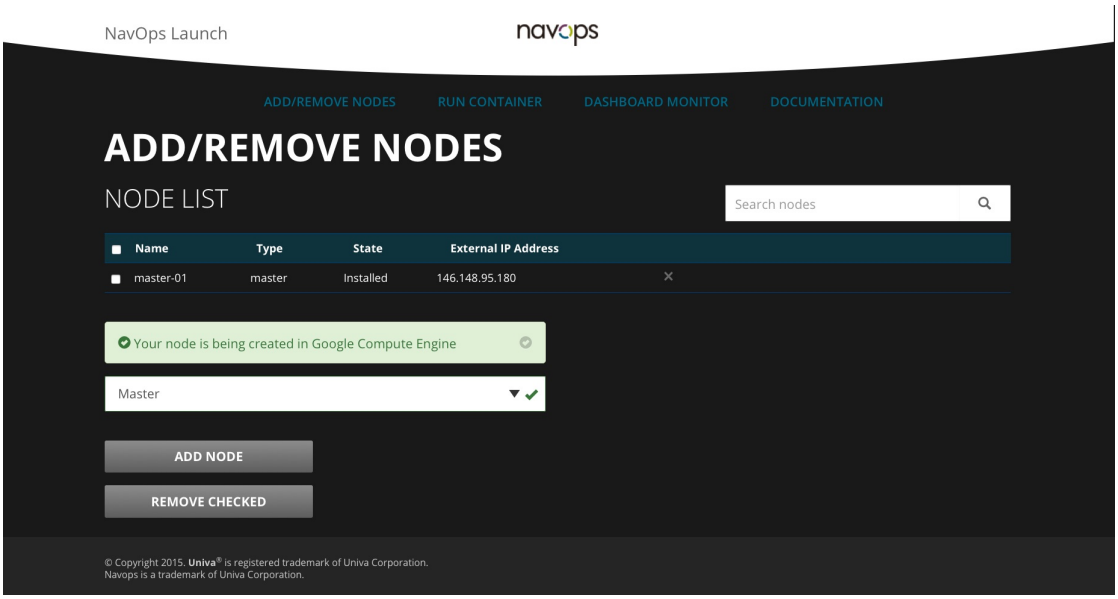

Figure 5:

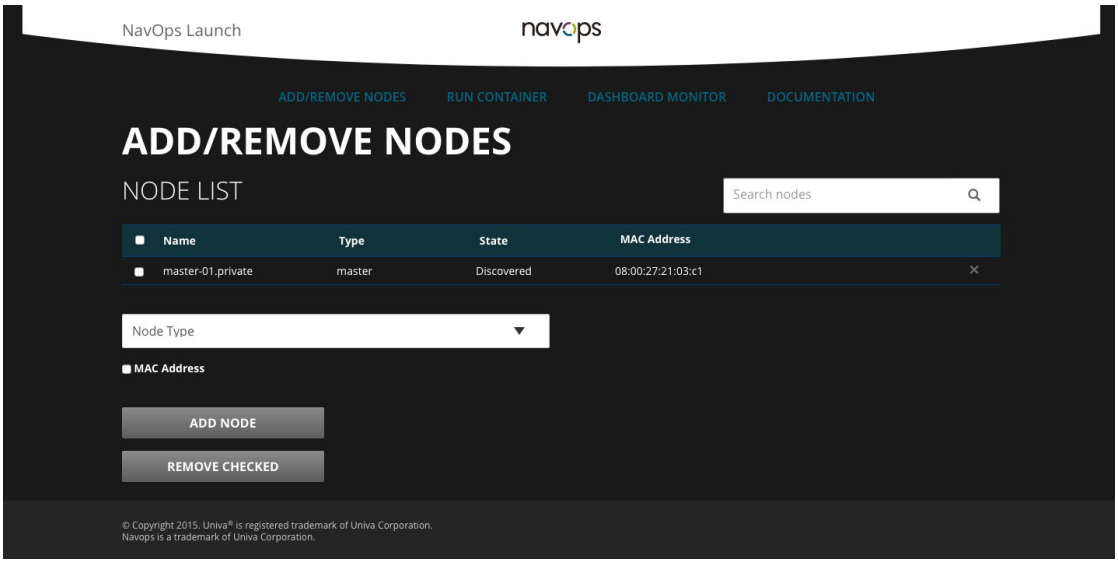

Figure 6:

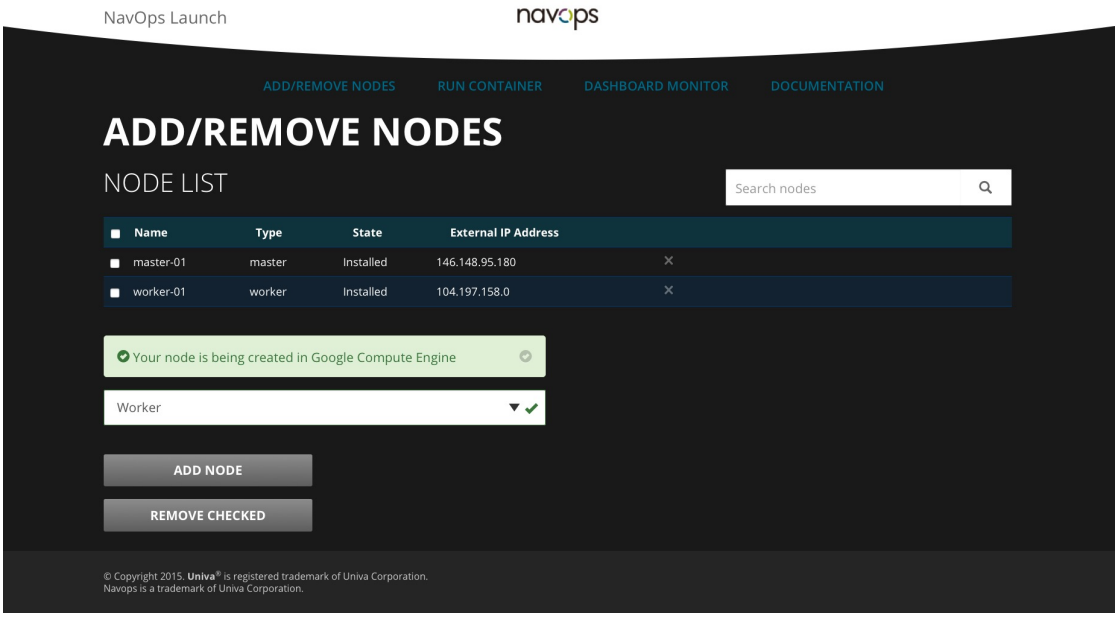

Figure 7:

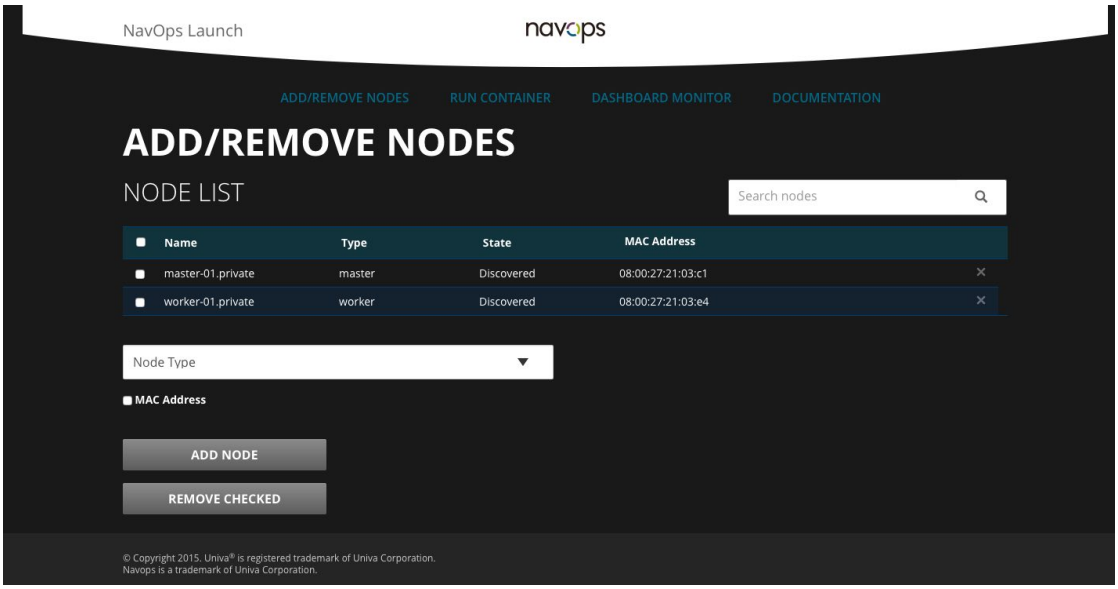

Figure 8:

1.8 Creating a Kubernetes cluster from the command line

Navops Launch provides a GUI for adding/removing nodes and basic monitoring of the nodes running the Kubernetes master and Kubernetes workers. If you prefer not to use the GUI the cluster can be assembled using the command line.

1.8.1 Adding a Kubernetes Master

If you are installing a bare-metal cluster then please first disable the firewall on the installer host following the step described in section Remvoing the Firewall for On-Premise Installation from CentOS Installer Node.

Now make sure you have a shell in the Navops container with the command below. To execute this docker command you need to logged in as root when installing from a CentOS host. From Mac OS X this is not necessary but need to connect your terminal session to the Navops Docker Machine via `eval $(docker-machine env navops)`.

```
# docker exec -it navops bash
```

You will notice the command prompt is `/`. When connected to the Docker Machine on Mac OS X the host name – here `nucky` – becomes `Navops`. Then run the following command to see if the navops service is operational

```
[root@nucky /]# get-node-status

Installer (OS: centos-7.1-x86_64) ------------------------------------------

<installer> (enp0s25: 172.20.1.49)
    Hardware Profile: Installer
    Boot: Disk
    Status: Installed/Active, Locked: HardLocked
```

Now you can add a Kubernetes master with the following command:

```
[root@nucky /]# add-nodes --software-profile master --hardware-profile
master --count 1 --mac-addr 00:18:71:e7:0c:f2
Waiting for new node #1 of 1...
Added node [master-01.example.com] IP [172.20.1.50] MAC
[00:18:71:e7:0c:f2]
The following nodes were added to the system:
master-01.example.com
```

The command above will add the node to the cluster and configure it properly. In a case where you add your node on a public cloud platform this is all you need to do. For other cases you will need to PXE-boot the node. Please refer to section Atomic Installation Steps during PXE-Booting for some hints on how to get through the provisioning of the RedHat Atomic OS which will occur during PXE-booting.

In case of bare-metal nodes you can also modify the command like below

```
[root@nucky /]# add-nodes --software-profile master --hardware-profile
master --count 1 --mac-addr 00:18:71:e7:0c:f2
```

in order to execute the node installation on the server with the MAC address `00:18:71:e7:0c:f2`.

If the installation process doesn't start and the node fails to boot please look at the Troubleshooting section.

1.8.2 Adding Kubernetes Worker Nodes

In analogy to the master node you can now add worker nodes via the following command:

```
[root@nucky /]\# add-nodes --software-profile worker --hardware-profile
worker --count 1 --mac-addr b8:ae:ed:78:29:85
Adding 1 node...
Added node [worker-01.example.com] IP [172.20.1.51] MAC [b8:ae:ed:78:29:85]
Adding nodes: 100% (1 of 1 complete)
The following nodes were added to the system:
worker-01.example.com
```

As for the master node no further action is required to add a worker node to a public cloud installation while the other installation cases require that you PXE-boot the node.

Again you can specify a MAC address for bare-metal installation cases where you want to ensure only a specific server gets selected. Use a command like this in such a case:

```
[root@nucky /]\# add-nodes --software-profile worker --hardware-profile
worker --count 1 --mac-addr b8:ae:ed:78:29:85
```

If the installation process doesn't start and the node 'fails to boot' please look at the Troubleshooting section.

1.8.3 Connecting to the Kubernetes Master

When you have a shell session on the Navops container you can simply ssh to the Kubernetes master with the following command:

```
[root@nucky /]# ssh fedora@master-01
Last login: Fri Oct 30 19:42:33 2015 from 172.20.1.49
[fedora@master-01 \~]$
```

1.8.4 Connecting to the Kubernetes Workers

Normally you do not connect to the Kubernetes workers, however if you need to troubleshoot Kubernetes on the nodes then you can connect to the Kubernetes workers directly from the Navops container with the following command:

```
[root@nucky /]# ssh fedora@worker-01
Last login: Fri Oct 30 19:42:38 2015 from 172.20.0.176
[fedora@worker-01 \~]$
```

Or you can connect to the Kubernetes workers by ssh forwarding credentials from the Navops container to Kubernetes master and then to workers, like this:

```
[root@nucky /]# eval $(ssh-agent)
Agent pid 12903
[root@nucky /]# ssh-add
Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
[root@nucky /]# ssh -A fedora@master-01
Last login: Fri Oct 30 19:40:54 2015 from 172.20.1.49
[fedora@master-01 \~]$ ssh -A fedora@worker-01
Last login: Fri Oct 30 19:42:18 2015 from 172.20.1.49
```

1.9 Testing your Navops Installation

Kubernetes is a very comprehensive container orchestration system and for a complete understanding of Kubernetes you should read the Kubernetes documentation at the Kubernetes Website.

1.9.1 First Simple Test

For this first simple test your cluster will need connectivity to the Internet in order to be able to download an image (`nginx` in this case) from Docker Hub. In case your cluster is not connected to the Internet please skip this section.

For simple tests the Navops Launch installer package provides the `kubectl.sh` script which allows to execute Kubernetes `kubectl` commands from the installer node. On a CentOS 7 installer node you will need to be logged in as root to be able to run the commands below. On Mac OS X you only will need to connect to the Navops Docker Machine via `eval $(docker-machine env navops)`.

First run:

```
# ~/.Navops/bin/kubectl.sh run nginx --image=nginx
CONTROLLER   CONTAINER(S)   IMAGE(S)   SELECTOR       REPLICAS
nginx        nginx          nginx      run=nginx      1
```

This will pull an image called ‘nginx’ from Docker Hub and run it as a so call ed ‘pod’ via a ‘replication controller’ under the same name on your Kubernetes cluster (see the Kubernetes documentation for details on the terms ‘pod’ or ‘replication controller’).

To see whether the command has worked run

```
# ~/.Navops/bin/kubectl.sh get pods
NAME             READY     STATUS    RESTARTS   AGE
nginx-l8n3i      1/1       Running   0          1m
```

And to delete the pod again you will have to delete its replication controller via the command

```
# ~/.Navops/bin/kubectl.sh delete rc nginx
replicationcontrollers/nginx
```

A follow-on check for running pods (the second command above) will show no more nginx pods are running.

1.9.2 Launching a 'Hello World' container in Kubernetes

The following steps will configure and run a simple "Hello World" container in your new Kubernetes cluster. This is a quick way to show that everything is working as expected and it does not require an Internet connection of your cluster. You will, however, need to be logged into the master node of your Kubernetes cluster. See section Connecting to the Kubernetes Master for how to do that.

Kubernetes launches containers in 'pods'. In this example we will create a simple "Hello World" container and run it in a Kubernetes 'pod'.

First you need to create a YAML file that contains the following. Please do so while being logged into your Kubernetes master node:

```
apiVersion: v1
kind: Pod
metadata:
 name: hello-world
spec:
 restartPolicy: Never
 containers:
   - name: hello-world
     image: "centos:7"
     command: ["/bin/echo", "hello", "world"]
```

Save this file as `hello-world.yaml` and run it in Kubernetes with the following command:

```
# kubectl create -f ./hello-world.yaml
```

You can check to see if your pod is running the hello-world container with the following command.

```
# kubectl get pods
```

Your pod should be running or will be pending and will run soon.

This particular container did not actually run much, just echoed 'Hello World'. You can view the output of the container with the following command.

```
# kubectl logs hello-world
hello world
```

Finally you can remove a pod with the following command:

```
# kubectl delete pods/hello-world
pods/hello-world
```

1.10 Installation Summary

At this point the following activities have been completed:

1. CentOS 7 installed and configured on a machine if you are using Linux as installer node
2. Docker Installed on the CentOS 7 machine or Docker Toolbox installed on Mac OS X.
3. Navops container downloaded and installed on the Mac machine or on the CentOS 7 machine
4. A shell session is created to the Navops container and the Navops provisioning tool is working
5. A Kubernetes master is added to the Navops cluster
6. One or more Kubernetes workers are added to the Navops cluster
7. Kubernetes cluster status is confirmed operational
8. A simple 'Hello World' container is launched and the output reviewed.

1.11 Next Steps

Now that you have a working Kubernetes cluster you should check out the Kubernetes.io web pages for more information on Kubernetes and how to launch containers and services at http://kubernetes.io

In addition to Kubernetes information you can also checkout Docker for more details on Docker, how it works and the tools available at: http://docker.com

1.12 Re-Installing

If you need to re-install Navops at any time you can simply do so via the command `\~/.Navops/bin/reinstall.sh`. When installing on CentOS 7 just login to your installer node as root, open a terminal and run this command. When installing on Mac OS X it is easiest to open a Docker Quickstart Terminal (part of the Docker Toolbox) and run the command there.

1.13 De-Installing

De-installing Navops Launch can simply be performed via two steps:

1. Removing the Kubernetes all worker nodes and the master node
2. De-installing the Navops containers from your installer host

1.13.1 Removing Kubernetes Nodes

It is easiest to remove Kubernetes hosts from the Navops web-UI using the 'Add/Remove Nodes' dialog. Simply select the nodes you want to be removed and click on the 'Delete' button. Note that you can select all nodes by clicking into the box next to 'Name' in the node list table heading.

If you prefer to use the command-line then log into the Navops container. On CentOS be sure you are logged in as root and on Mac OS X you may want to use a Docker Quickstart Terminal and execute `eval $(docker-machine env navops)` in it. Then run a command sequence similar to the below:

```
# docker exec -it navops bash
[root@nucky /]# delete-node --node=master-01,worker-01,worker-02
```

The second line would delete three nodes from your Navops Kubernetes cluster, the master nodes as well as two worker nodes.

1.13.2 De-Installing Navops Containers

The command

```
# ~/.Navops/bin/uninstall.sh --keep-images
```

will de-install the Navops containers from your installer host. It will, however, keep the container images that were downloaded from the Navops repository during installation. This will speed up the process of installing again as it will then only download containers which have been updated in the repository in the meantime.

If you want Navops to be removed completely then just drop the `--keep-images` from the above command.

On Mac OS X you can also remove the Navops Docker Machine alternatively via the command

```
# docker-machine rm Navops
```

In all described cases it is then safe to do a complete re-installation by following the instruction above, i.e. start with the `bash -c "$( curl https://..."` command.

1.14 Updating a Navops Cluster

You can also update a Navops cluster if your current installation is Navops Launch version 0.5 or later. The process involves the following steps:

1. Backup the cluster configuration as described in Backing-Up a Navops Cluster
2. De-install Navops as described in De-Installing
3. Install Navops anew as described in sections
   - Installing Navops Launch on CentOS 7 with Docker.
   - Installing Navops Launch on Google Cloud Platform.
   - Installing Navops Launch on Amazon Web Services.
   - Installing Navops Launch on Mac OS X with Docker Toolbox
4. Restore the backed-up configuration as per the instructions in section Restoring a Navops Cluster

This update process should be used when you want to upgrade your master node to a new Kubernetes and OS version while preserving the configuration you currently have in Kubernetes. When performing the upgrade you will have a slight outage between the time the new master is operational and the backed up configuration is loaded. Backup and restore also only works when upgrading or re-installing the master. Attempts to downgrade the master may yield poor results (this would be due to Kubernetes not understanding the configuration data if it was downgraded).

1.14.1 Backing-Up a Navops Cluster

To backup a Navops cluster you can use the command

```
# ~/.Navops/bin/backup.sh
```

This command needs to be executed from the installer host. On CentOS you should be logged in as root. On Mac OS X you will want to use a Docker Quickstart Terminal and point your terminal session to the Navops container by executing `eval $(docker-machine env navops)`.

The `backup.sh` script will essentially do the following:

1. Stop the Kubernetes API daemon
2. Stop the etcd daemon
3. Make a tar ball of the etcd data direcetory
4. Copy the tar ball off of the master host

1.14.2 Restoring a Navops Cluster

For restoring a Navops cluster the command

```
# ~/.Navops/bin/restore.sh
```

is available. It needs to be executed in the same environment as the `backup.sh` script described above and should be executed after a successful, fresh installation of Navops. The `restore.sh` script will do the following in essence:

1. Stop the Kubernetes API daemon
2. Stop the etcd daemon
3. Remove or backup the existing etcd data directory
4. Copy the the backup tar ball to the master host
5. Extract the backup tar ball
6. Start the etcd daemon
7. Start the Kubernetes API daemon

1.15 Best Practices and HowTos

The following sub-sections contain best practices and hints for accomplishing certain administrative tasks which may be required during the Navops installation process. These sections are referred to from the main installation guide above.

1.15.1 PXE-Booting a VirtualBox VM

When adding Kubernetes master and worker nodes to your virtual Navops cluster on Mac OS X you will need to PXE-boot VirtualBox VMs. Use the following instructions:

1. Create a new VM selecting Linux and Fedora 64bit for it and accept all defaults. You may want to choose a name like `kmaster`, `kworker-01`, `kworker-02`, ... for it
2. When the VM is created but still powered off then right-click it and open the 'Change ...' menu. In there modify the following settings:
3. Select 'System' and under boot options turn on booting from the network and move it to the top of the boot options
4. Select 'Network' and connect the VM to 'NAT-Network' and chose 'NavopsNat' as name
5. Save your changes by clicking 'OK'
6. You can just start the VM now when the Navops installation procedure instructs you to do so

1.15.2 Atomic Installation Steps during PXE-Booting

When PXE-booting Kubernetes master or worker nodes during the Navops Launch installation process RedHat Atomic will get installed on those nodes. You will have to select a disk upon which the Atomic OS will get deployed or confirm the one which has been auto-selected. When the selected disk already has data on it then you may get an error message that not enough space is available on that disk. That is always the case if you boot a VM using a virtual disk (as you will when installing on Mac OS X). In that case you will need to allow the Atomic installer to re-format that disk.

For any other questions that you will be asked during bootstrapping Atomic you can accept defaults (usually by typing 'c').

1.15.3 Remvoing the Firewall for On-Premise Installation from CentOS Installer Node

This section only applies when you install on bare-metal servers and from a CentOS Installer node. It is not relevant for installation on Google Cloud Platform or on VirtualBox on Mac OS X.

If you are installing an on-premise cluster and you have deployed the installer containers on a local, docker-enabled node such as CentOS 7 then you will need to disable the firewall there. On your CentOS installer node you should do the following:

1. `systemctl stop firewalld`
2. `systemctl disable firewalld`

You obviously will need system administrator permissions for these commands so you may have to prepend them with `sudo` or be logged in as root.

1.15.4 Instructions for Installing CentOS 7 with Docker

Before you can install Navops Launch you will need a CentOS 7 machine with Docker installed. Navops Launch may work with any Linux Operating system that can run Docker 1.7, however we have not tested all Linux distributions.

Installing CentOS 7

1. Boot the CentOS 7 installation media.
2. Enable the network card on the machine, set the name to something different other than `localhost` . It should be `<machine_name>.<yourdomain>`, for example `navopsdocker.example.com`
3. Choose Server type, typically we choose "Server with GUI"
4. Change the partitioning, choose "I will configure partitioning"
5. The installer will tell you "You haven't created any mount points for your CentOS 7 installation"
6. Make sure you have enough 'Available Space' (Bottom Left Corner) for the installation and the Docker Images.
7. If you are installing CentOS 7 on a machine that has an old OS installed or other partitions that have data then you may not have enough space for the installation.
8. Delete any old partitions that may be left around from a previous install
9. Select 'click here to create them automatically" (Top Left Corner)
10. That will create a partition layout for the OS and home directory.
11. Delete 'home' and use the space for Docker images later.
12. You should end up with a large amount of available space free (depending on the size of your disk).
13. Set the Root password
14. Create any other users you need, for example we create a 'univa' account.
15. Installation should proceed and finish normally.
16. Don't forget to remove media when you are finished installing.

Once the newly installed CentOS 7 operating system reboots it should present a GUI login prompt. Login as `root` to continue with Installing Docker on CentOS 7.

Installing Docker on CentOS 7

1. Login as `root`
2. Open a command line terminal
3. Set selinux to permissive mode with the following commands:
    1. `# vi /etc/sysconfig/selinux`
    2. edit the following line: `SELINUX=enforced`
    3. to: `SELINUX=permissive`
    4. save and exit the file
4. reboot the machine for the changes to take effect
5. Login as `root`
6. Open a command line terminal
7. If you are logged in as an account other than `root`, you will likely have to add your non admin account to sudoers file if you haven't already.
8. Run the following command `yum install docker`
9. The Docker packages will be downloaded by `yum` and installed on the machine
10. Once complete you have to start the docker daemon on the machine with the following commands:
    - `# systemctl enable docker`

- `# systemctl start docker`

11. Now you can check to see if docker is running using the following command
    - `# docker ps`
    - `# systemctl status docker`
12. Finally before you add any Kubernetes Master or Worker nodes you will have to disable the firewall on the CentOS 7 machine hosting the containers with the following commands:
    - `# systemctl stop firewalld`
    - `# systemctl disable firewalld`

Partitioning Extra Disk Space on CentOS 7 for Docker If you followed the instructions for installing CentOS 7 then part of the disk is unused because the `/home` partition was deleted during partitioning. This additional space can be used by Docker to store the docker images. Storing docker images in the LVM partition is far superior to the default storage on CentOS. Follow the instructions below to partition the unused space as LVM, extend the LVM volume group and then run the Docker storage tool to configure Docker to use the new space.

1. run `sudo fdisk /dev/sda` (it could be a different disk on your machine, you could even have multiple disks)
2. Create a new partition using the empty space on the disk. You should create a primary partition and use the rest of the empty space on the disk.
3. Change the partition id to `Linux LVM` which was type `8e` in CentOS 7
4. Make sure you write down the device name, in our example it was `/dev/sda3`
5. Exit by 'writing the table and exit'
6. Reboot the machine.
7. Determine the volume group name on your machine with the following command: `sudo vgdisplay`
8. Extend the volume group to add the new space. `sudo vgextend <volume_group_name> /dev/sda3`
9. At this point the extra disk space has been added to the volume group.
10. Now run docker-storage-setup so that docker can use the new extended LVM. with the command `sudo docker-storage-setup`
11. Now you can start the installation process of NavOps Launch.

1.15.5 Changing the node naming for Kubernetes nodes

Navops Launch provisions a Kubernetes cluster, sets the Kubernetes master name to `master-01` and `worker-01` to `worker-NN`. This naming convention works well when you have one Kubernetes cluster in your enterprise, however if you have more than one Navops Kubernetes cluster the names will conflict because both clusters will have the same names and will conflict in DNS when using DHCP relay. To avoid this situation you can change the default naming scheme for nodes in your Kubernetes cluster avoiding any potential name collisions.

To change the default master naming scheme use the following commands.

From the CentOS 7 Docker host:

```
# docker exec -it navops bash -lc "update-hardware-profile\
--name master --name-format master-foo-#NN"
```

From the Navops Launch container:

```
# update-hardware-profile --name master --name-format master-foo-#NN
```

To change the default worker naming scheme use the following command.

From the CentOS 7 Docker host:

```
# docker exec -it navops bash -lc "update-hardware-profile\
--name worker --name-format worker-foo-#NN"
```

From the Navops Launch container:

```
# update-hardware-profile --name worker --name-format worker-foo-#NN
```

In both of the previous examples `foo-#NN` can be replaced with any unique name that you choose to avoid name conflicts between other clusters.

1.15.6 Installing the Docker Toolbox for Mac OS X

If you do not have Docker installed on your Mac OS X system then you should install the Docker Toolbox. The Docker team has packaged several of the most commonly used tools and boot2docker into the 'Docker Toolbox' package. You can download the Docker toolbox from the following location: https://www.docker.com/docker-toolbox

1. Go to https://www.docker.com/docker-toolbox
2. Select 'Download Mac'
3. Run the Installer
4. Once the installer completes 'boot2docker' and other Docker tools will be installed

Please take note of the Docker Quickstart Terminal which gets installed along with the Docker Toolbox. Do not launch the Docker Quickstart terminal before installing the Navops Containers.

1.15.7 Navops Launch Networking Configuration

With a DHCP Server behind a Router/Firewall

No DHCP Server (DHCP Relay) on a switch configuration

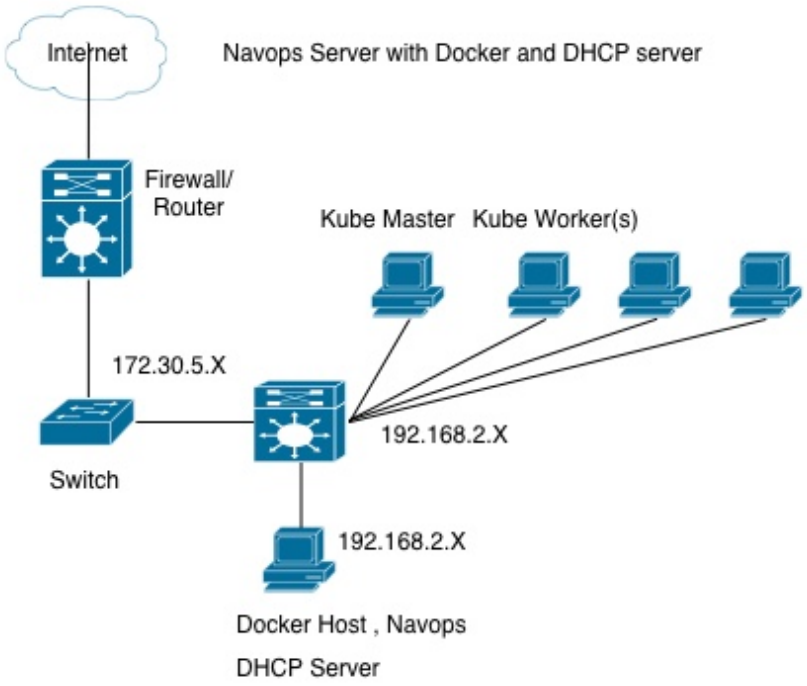

Figure 9:

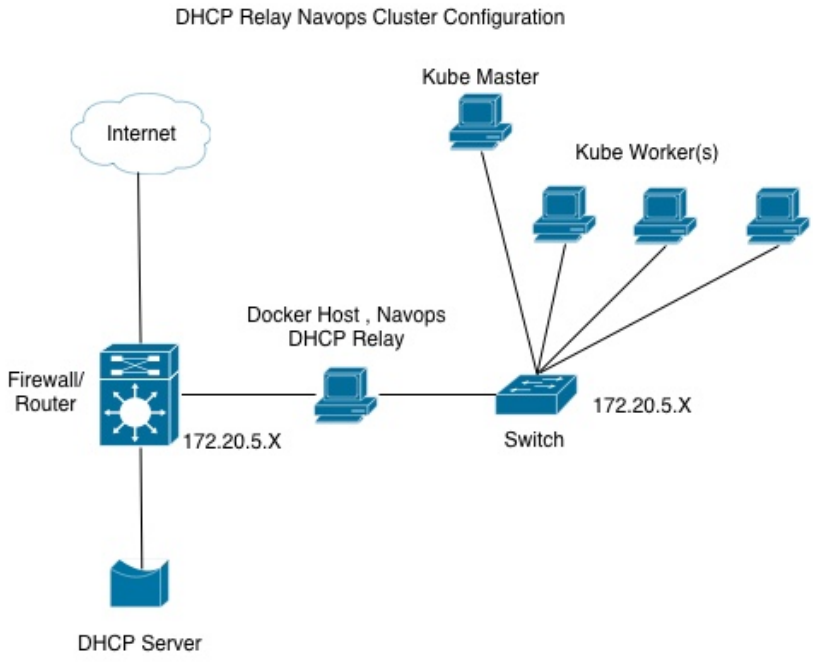

Figure 10:

1.16 Troubleshooting

1.16.1 Issue: TFTP Failure

Symptom: Node PXE-boots, receives an IP address and attempts to start the operating system installation by downloading a file using TFTP but it fails.

Reason: This can happen when there are other DHCP servers on your network. When you PXE-boot a machine it broadcasts across the network for any DHCP server to respond. If a different DHCP server responds and provides an IP address to the node it will not complete the OS installation because it is looking in the wrong location.

Solution 1: You can use the `--mac-addr` flag to the `add-nodes` command and provide the MAC address of the machine you are PXE-booting. This will create a complete configuration in the Navops installer for the node and will update the DHCP server with the MAC address of the machine. The next time you PXE-boot the node you want to add to the cluster the DHCP server will respond and install the node as expected.

1.16.2 Issue: Timeout - Cannot Get Address from DHCP Server

Symptom: You PXE-boot and the boot process spins when trying to acquire an IP address from the DHCP server. Then it times out.

Possible Reason: When registering for your Navops download you have given the wrong answer when asked whether your environment already has a DHCP server. You may have stated, for instance, that your environment has one while installing on Mac OS X where the DHCP server is embedded in the Navops component. So the answer to the question should have been 'No'.

Solution: Edit the file `~/.Navops/settings.sh` and modify the parameter `DHCP_ENABLED`. If in the case of a Mac OS X installation this parameter has the value `0` then set it to `1` and save the file. Now you will have to re-install the Navops installer. Please follow the instructions in section Re-Installing for that.

1.16.3 Issue: Cannot get tftp file to continue booting

Symptom: The node PXE boots, however just after receiving an IP address from the DHCP server the boot process attempts to download using TFTP the next file in the boot process but it fails to download the file and the installation stops.

Solution: It is likely that the firewall is still running on the CentOS 7 Docker Host machine. You should disable the firewall and then PXE boot the node again. Disable the firewall using the following commands:

```
# systemctl stop firewalld
# systemctl disable firewalld
```

1.16.4 Issue: Screen Flashes Or Goes Blank

Symptom: Node PXE-boots, installation works, node reboots and then while booting the OS the screen flashes and then goes blank.

Possible Reason: It is very likely that one of the kernel boot options is causing an issue with your hardware. If you see the screen flash then it is likely the video driver is trying to switch to a video mode that is not supported by the hardware. This can cause the boot process to hang and the node will not start properly.

Solution: If it is a video issue then adding `nomodeset` to the kernel boot parameters may allow the node to boot. You can add this when the node is booting by interrupting the boot process at the GRUB screen and 'edit' by pressing 'e' the kernel boot parameters.

1.16.5 Issue: Not Enough Disk

Symptom: Node PXE-boots, installation works but stops complaining about not enough disk space on the designated installation disk.

Reason: The disk that got auto-selecting already has data on it. That is always the case when booting a VM (like for the Mac OS X installation case) and using a virtual disk.

Solution: Follow the instructions provided here: Atomic Installation Steps during PXE-Booting

1.16.6 Issue: Issues After Reboot on Mac OS X

Symptom: After a reboot of Mac OS X without previously halting the Navops VMs (e.g. in case of reboot after a crash) the Navops system may not come up cleanly even if you restart the VMs. E.g. not all Navos containers may get restarted or commands like `docker ps` may hang.

Possible Reason: These symptoms are likely caused by issues in VirtualBox or Docker's boot2docker integration on top of it.

Solution: You may have to re-install Navops. For that you can follow the steps in section De-Installing. Should the de-installation process hang, however, then you may need to interrupt it (Ctrl-C), then remove the Navops VM from the VirtualBox console and then re-run the de-installation command.

1.16.7 Issue: When installing a Kubernetes master or worker the node PXE boots, installs properly, however the node name is incorrect on the machine.

For example: When adding a Kubernetes master the node name by default is `master-01` and when the machine installs and boots it is another name such as `worker-01`

Reason: This issue can happen when you install NavOps using the DHCP Relay mode. When DHCP Relay is used during node installation the NavOps management server does not run a DHCP server but instead the DHCP request from booting nodes is typically handled by the corporate DHCP server. The corporate DHCP server responds to the node providing it an IP address, however the DHCP server may optionally have configured a name for the node. . . .i.e. it may have a IP to node name association for the IP address. When this happens the node will be provided the incorrect node name. You can check to see if this happened by asking the DHCP server Administrator(s) to check for the IP to node name name resolution.

Solution: You will have to shutdown the node that booted with the wrong name and ask the DHCP and DNS server Administrator to remove the DHCP lease for the node, also make sure

that any IP to node name associations are removed for the node. Then add the node back into the Navops Kubernetes cluster using the instructions in this guide.

1.16.8 Issue: When adding nodes to the cluster using the 'add-nodes' command the wrong node joins the cluster. It is another computer that PXE booted.

Reason: When the `add-nodes` command is used to add new nodes to the Kubernetes cluster the DHCP server is directed to accept any node that sends a DHCP request - i.e. any node that PXE boots in the network. In a network that already has many machines that DHCP boot it is quite common that the wrong node will be added to the cluster.

Solution: The `add-nodes` command has an additional command line parameter called `--mac-addr`. If the node is added with the `--mac-addr` then the proper node will always be added to the cluster because the DHCP server will have configuration for the specified MAC address.

© Copyright Univa Corporation. Navops is a trademark of Univa Corporation.