Kubernetes Installation and Operations

Contents

[1 Overview 4](#_Toc51654842)

[1.1 Introduction 4](#_Toc51654843)

[2 Application to UCI ESSL 5](#_Toc51654844)

[3 Preparations 6](#_Toc51654845)

[3.1 Verify Presence of Tools 6](#_Toc51654846)

[3.2 DHCP Re-Installation 6](#_Toc51654847)

[3.3 SW Installation 6](#_Toc51654848)

[3.3.1 Docker 6](#_Toc51654849)

[3.3.2 Kubernetes 6](#_Toc51654850)

[3.3.3 NGINX 6](#_Toc51654851)

[3.3.4 Helm 7](#_Toc51654852)

[3.3.5 Prometheus 7](#_Toc51654853)

[3.3.6 Elastic Search 7](#_Toc51654854)

[3.3.7 Grafana 7](#_Toc51654855)

[4 Kubernetes Cluster 8](#_Toc51654856)

[4.1 Set-Up of Cluster Nodes 8](#_Toc51654857)

[4.2 Cluster Networking 8](#_Toc51654858)

[4.3 Cluster Storage Heirarchy 8](#_Toc51654859)

[4.4 Cluster Monitoring 8](#_Toc51654860)

[5 Kubernetes Cluster Operations 9](#_Toc51654861)

[5.1 Using KUBECTL for Cluster Management 9](#_Toc51654862)

[5.2 Operations Monitoring 9](#_Toc51654863)

[5.3 Data Base and File Back-Up 9](#_Toc51654864)

[5.4 Purging of Log Files 9](#_Toc51654865)

[5.5 Cluster Power-Down 9](#_Toc51654866)

[5.6 Cluster Power-Up 9](#_Toc51654867)

[5.7 Trouble Shooting 9](#_Toc51654868)

[6 Application Pod Set-Up 10](#_Toc51654869)

[6.1 Helm Charts 10](#_Toc51654870)

[6.1.1 Introduction to Helm Charts 10](#_Toc51654871)

[6.1.2 Helm Registry 10](#_Toc51654872)

[6.1.3 Creation of Helm Charts 10](#_Toc51654873)

[6.2 Docker Images 10](#_Toc51654874)

[6.2.1 Introduction to Docker 10](#_Toc51654875)

[6.2.2 Docker Registry 10](#_Toc51654876)

[6.2.3 Creation of Docker Inages 10](#_Toc51654877)

[6.3 oTree Work-Space Creation 10](#_Toc51654878)

[6.4 zTree Work-Space Creation 10](#_Toc51654879)

[7 Parked Text Copy 11](#_Toc51654880)

[7.1 Reminders / Check-Lists 11](#_Toc51654881)

[7.1.1 List One 11](#_Toc51654882)

[7.1.2 List Two 11](#_Toc51654883)

[7.2 Texts for Inclusion 11](#_Toc51654884)

[7.3 URL Pointers 11](#_Toc51654885)

# Overview

## Introduction

This is not a text book, rather this file collects succinct descriptions and pointers for operating a Kubernetes cluster.

There are a lot of sections in the document (see above) which will be fleshed out as we will go along with the cluster definition and set-up on the one hand or environment preparations on the other.

## Next Steps

* detail the DHCP data base back-up for relocating to a new machine, setting up the service on the new machine, verifying it working, and subsequently purging DHCP from the original node
* preparing three machines for Kubernetes installation

# Application to UCI ESSL

this chapter will address the specific of the UCI lab where a Kubernetes cluster will be set up

this chapter is not for general use

# Preparations

These instructions are applicable to Debian and Ubuntu.

There may be slight modifications of the command required if CentOS or Linux are used. It is left to the reader to figure out which changes need to be made.

## Verify Presence of Tools

apt, git, curl

## DHCP Re-Installation

DHCP (Dynamic Host Configuration Protocol) is used to automatically configure hosts with crucial IP parameters across the network when they are booted up. This way machines on a network will have addresses drawn from a pre-defined range, will know which default router to use, and where to find the DNS servers.

In some cases DHCP is also used to configure hosts for Windows-based IP communications.

This section has been created using the [Debian DHCP Server Configuration](https://www.debianhelp.co.uk/dhcp.htm) web-page as a guide. Some details have been taken from the [Debian DHCP Server](https://wiki.debian.org/DHCP_Server) wiki-page.

For in-depth administration guidelines check out the Oracle’s [Network Address Configuration](https://docs.oracle.com/cd/E37670_01/E41138/html/ol_netaddr.html) manual.

Note:

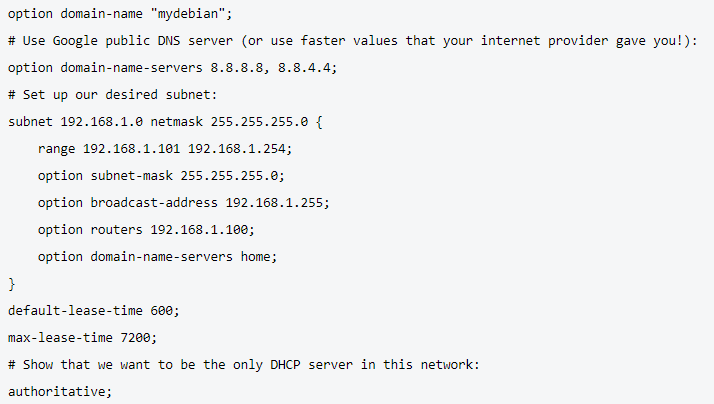
save existing configuration depending on current implementation

### Saving Existing DHCP Data-Base

On Ubuntu the DHCP configuration file is at **/etc/dhcp/dhcpd.conf**.

For Windows-Server look in: **%SystemRoot%System32\DHCP\backup**

Display the file contents to make sure the contents are what you are looking for in the context of your network. You should expect data similar to this:



Transfer the DHCP content to a storage area where it will be accessible for loading into the targeted new DHCP server.

### Verify Availability of DHCP on Debian Target Host

This section has been created using the [Debian DHCP Wiki-Page](https://wiki.debian.org/DHCP_Server) web-page as a guide.

To check whether DHCP is already available run:  
**sudo service isc-dhcp-server status**

To check whether DHCP is already available run:  
**sudo service isc-dhcp-server status**

### Configure DHCP Data-Base

text

### Verify DHCP Server Running

On the server running the DHCP service confirm that the service is indeed up and running:  
**sudo service isc-dhcp-server status**

Make sure the output from the command is indicating that the service is **running** or **active**.

Next go to a workstation, which is supposed to receive its IP configuration from the DHCP server.

### Configure Debian DHCP Clients

Boot up the Debian-Client.

Open the file **/etc/network/interfaces**; and verify that its contents look like this:

**auto lo eth0**  
**auto lo inet loopback**  
**iface eth0 inet dhcp**

If not make the necessary changes and re-boot the machine.

Once the machine is up and running, log-in and verify its IP configuration by running: **ifconfig**

Make sure the output matches what has been configured on the DHCP server. In particular check the IP address shown for the DHCP and DNS servers as well as the default router.

### Configure Windows DHCP Clients

On a Windows client run: **ipconfig /all**

On Debian client run: **ifconfig**

Once the machine is up and running, log-in and verify its IP configuration.

On a Windows client run: **ipconfig /all**

On Debian client run: **ifconfig**

Make sure the output matches what has been configured on the DHCP server. In particular check the IP address shown for the DHCP and DNS servers as well as the default router.

## SW Installation

text

|  |  |  |
| --- | --- | --- |
| Packag*e* | Native | K8S |
| Docker | X |  |
| Kubernetes | X |  |
| NGINX |  | X |
| Helm | X | X |
| Grafana |  | X |
| Elastic Search |  | X |
| Prometheus |  | X |

### Docker

### Kubernetes

install Kubernetes control plane (kubeadm, kubelet, kubectl)

### NGINX

### Helm

### Prometheus

### Elastic Search

### Grafana

# Kubernetes Cluster

text

## Set-Up of Cluster Nodes

text.

1. have two or three Linux machines on hand (one master, at least one worker)
2. make sure you have superUser access to the machines (uid/pwd)
3. configure network access so that both machines are accessible across Internet
4. verify git, curl, and Docker are installed, if not add
5. set up a Docker registry or make sure Docker hub is reachable from machines in K8s cluster
6. set up a basic cluster following tutorial instructions at  
   either: <https://www.digitalocean.com/community/tutorials/how-to-create-a-kubernetes-cluster-using-kubeadm-on-debian-9>  
   or https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/  
   or: https://www.howtoforge.com/tutorial/how-to-install-kubernetes-on-ubuntu/  
   (may mention previous steps 1. — 4.; ignore and/or selectively complete)
7. implement and verify cluster monitoring to be set-up and working  
    (Grafana - ElasticSearch - Prometheus)
8. create a basic application Helm chart to make sure cluster is working properly;  
   execute the Helm chart (this step may have been part of the tutorials!)
9. decide on load-balancing scheme for appropriately distributing traffic to worker nodes
10. configure Kubernetes cluster network;  
    make sure to use ‘service endpoints’ and not IP addresses to access applications
11. configure a storage hierarchy consisting of ‘in-memory’ storage for direct pod access  
    and ‘persisted’, physical (NFS) file and data-base (PostgreSQL, MySQL) storage  
    (user PostgreSQL for oTree/Django runtime support!!)
12. extend Kubernetes cluster as other HW worker nodes become available;
13. make sure these nodes are visible in the monitoring and participate in the load-sharing

text

## Cluster Networking

configure Kubernetes IP traffic ingress/egress

## Cluster Storage Heirarchy

text.

## Cluster Monitoring

text.

# Kubernetes Cluster Operations

text

## Using KUBECTL for Cluster Management

text.

## Operations Monitoring

text.

## Data Base and File Back-Up

text.

## Purging of Log Files

text.

## Cluster Power-Down

text.

## Cluster Power-Up

text.

## Trouble Shooting

text.

# Application Pod Set-Up

use of Helm charts and Docker images

there a are essentially two strategies: prepare the POD contents from scratch using Helm charts or pre-configure Docker images that get loaded and activates using appropriate Helm charts;

for running in Kubernetes pod create a Helm chart for loading and starting the image

## Helm Charts

### Introduction to Helm Charts

basic structure and functionality

### Helm Registry

Helm registry as source of Helm charts

local registry

### Creation of Helm Charts

basic functionality

## Docker Images

### Introduction to Docker

basic functionality

### Docker Registry

Docker registry as source of Docker images

local registry

### Creation of Docker Inages

basic functionality

## oTree Work-Space Creation

for the creation of oTree images create a Docker file [template] and image that has installed:

Debian, git, curl, Python 3, PIP, virtualenv, Django, PyCharm

either extend the template with further Docker directives or

load the image and then complete the oTree installation / set-up (may suffice if ready to load oTree app)

text.

## zTree Work-Space Creation

text.

# Parked Text Copy

text

## Reminders / Check-Lists

### List One

understand the Kubernetes theory-of-operations  
know the role of Kubernetes master and workers  
understand Kubernetes networking

know how to prepare the available infra-structure  
make a plan that assigns roles/functions to all available HW nodes  
know how to install Kubernetes control plane (kubeadm, kubelet, kubectl)  
install nginx and configure Kubernetes IP traffic ingress/egress

experience with Docker (installation, registry set-up, image creation, image uploading)

understand Helm, know how to install and interwork it with Kubernetes

know how to create Helm charts that reflect a specific HW infra-structure and that instantiate PODs conducive for supporting oTree lab-environment

alternatively, proof of several successfully installed Kubernetes cluster environments and documented by associated Helm charts

### List Two

* lokales Docker-Registry für die Speicherung von Images mit benutzerbezogenen O-Tree images
* Betriebsüberwachung (Prometheus, Elastic Search, Grafana)
* Datenbank Datensicherung (Export der PostgreSQL DB-Daten in Sicherungsdateien)
* Datei-Sicherung (Sicherung der Dateien vom Datei-Server)
* Revolvierende Journale (Log-Dateien werden über mehrere Tage gespeichert und dann gelöscht)
* Regelmäßiges Löschen temporärer Dateinen und unwichtiger Journale

## To Be Checked / Verified

Verify the purpose of **/etc/resolv.conf**)

## Texts for Inclusion

text

## URL Pointers

url

[Kubernetes Application Deployment with Helm -- Beginners Guide](https://deref-1und1.de/mail/client/1IQ44OytFZM/dereferrer/?redirectUrl=https%3A%2F%2Fwww.slideshare.net%2FmKrishnaKumar1%2Fkubernetes-application-deployment-with-helm-a-beginner-guide)

<https://itnext.io/unleash-the-power-of-microk8s-your-own-production-like-kubernetes-cf356ccb75f7>  
[https://github.com/ubuntu/microk8s/issues/695](https://deref-1und1.de/mail/client/ENv1PpkCmfc/dereferrer/?redirectUrl=https%3A%2F%2Fgithub.com%2Fubuntu%2Fmicrok8s%2Fissues%2F695)  
[https://serverfault.com/questions/958912/helm-under-microk8s-giving-error-could-not-find-a-ready-tiller-pod](https://deref-1und1.de/mail/client/aH_aCayH35U/dereferrer/?redirectUrl=https%3A%2F%2Fserverfault.com%2Fquestions%2F958912%2Fhelm-under-microk8s-giving-error-could-not-find-a-ready-tiller-pod)

url