National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Anant Sakharam Pednekar
Student ID: 21188947

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Anant Sakharam Pednekar
Student ID:	21188947
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	15/12/2023
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Anant
Date:	29th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Anant Sakharam Pednekar
21188947

1 Introduction

This article is an in-depth guide for setting up a Kubernetes cluster on a Ubuntu server. It lays out the essential steps for building the cluster and deploying a robust Pod network in great detail and provides the essential technologies.

The manual's last section explores benchmarking tool configuration methods, shedding light on the complexities of k-bench and sysbench deployment. Insights gained from this investigation will help readers assess the efficiency and effectiveness of their Kubernetes environment, which can lead to better decisions on the allocation of resources.

2 Tools and Technologies Required

The following table outlines the essential tools and technologies utilized in this experiment, encompassing the clustering platform, application-container operating system, container software, container orchestrator, platform tools for monitoring, design specification, and other key elements necessary for the successful execution of the experiment.

Category	**Details**
Server	AWS EC2
Host Operating System (OS)	Ubuntu Server
Container Orchestrator	Kubernetes , Openshift
Tools for Benchmarking	Kbench, Benchmark-operator
Slave & Master Config	t4g.2xlarge (8 vCPUs and 32GB)

Table 1: Tools & Technologies

3 Creation of Kubernetes Cluster

This portion of the configuration manual utilizes AWS Cloud services, specifically EC2 instances, to construct a Kubernetes cluster. Refer 1.

The selected operating system for this research is Ubuntu Server. This approach is chosen for its compatibility and support in effortlessly creating Kubernetes clusters.

To set up the Kubernetes cluster, it is necessary to have both a master node and one or more slave nodes. For this benchmarking, the master and slave nodes are built using

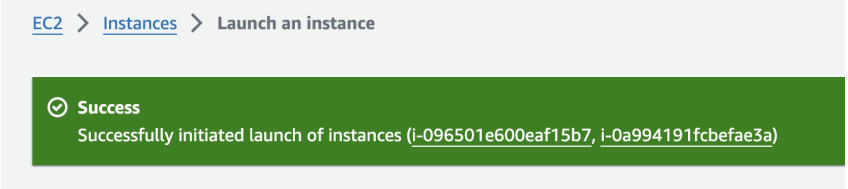

Figure 1: AWS Instances Launch

the t4g.2xlarge instance type, which has 8 virtual CPUs and 32GB of memory.

Before initiating the cluster, installing specific prerequisites on all nodes is necessary Refer 2 . Each node should perform the following commands [1]:

```
sudo apt-get install golang make tmux -y
export GOROOT=/usr/lib/go
export GOPATH=/home/ubuntu/go
```

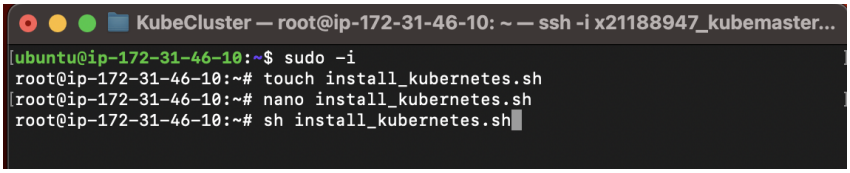

Figure 2: Install Kubernetes

The command in figure 3 is to be executed only on the master node :

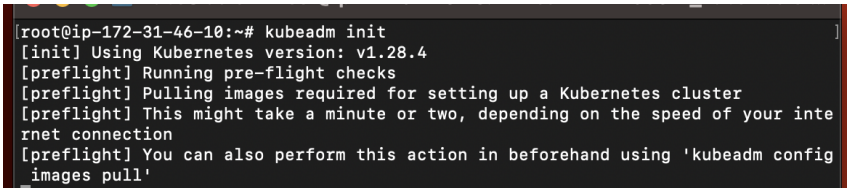

Figure 3: Kubernetes init

The master node will then create the cluster using kubeadm and generate a token. This token is essential for the slave/worker nodes to join the cluster. Refer 4

On each worker node, the following command should be executed using the token obtained from the master node :

[1]Link to install_kubernetes.sh

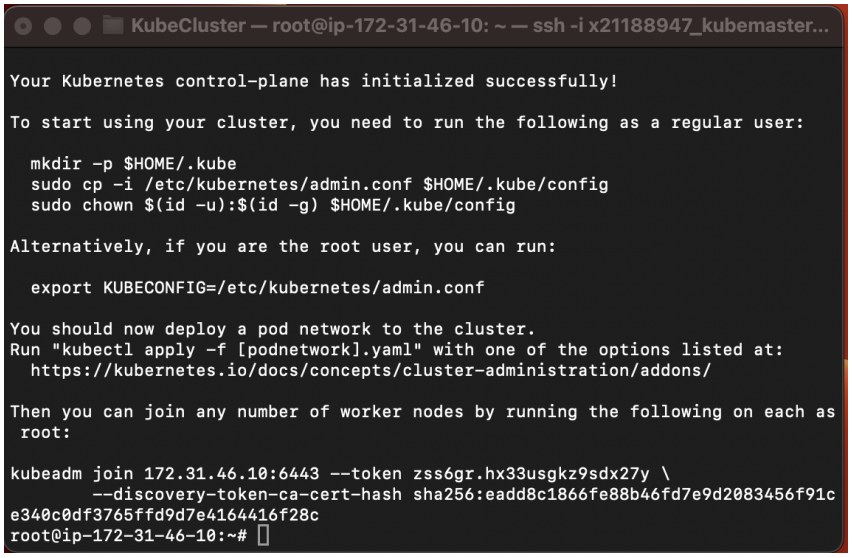

Figure 4: Master Ready

```
kubeadm join 172.31.22.210:6443 \
--token mndegl.v4ez3pj9ru3i7qkp \
--discovery-token-ca-cert-hash  \
sha256:a1fbbcb61c03305f43d352a8f988aebf4b1a1909c574d6ac313e9ce3509578ed
```

Once joined, the output should resemble the one shown in figure 5

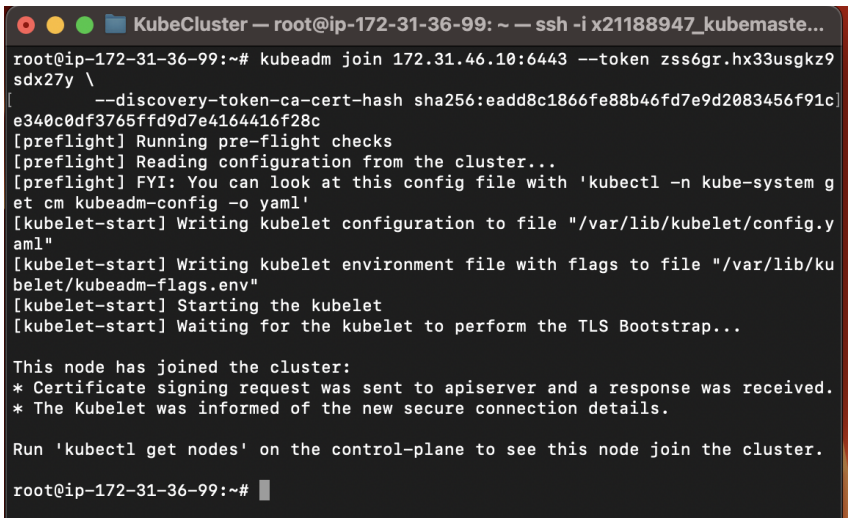

Figure 5: Worker init

To verify the functionality of cluster and install the Pod Network, the following commands are executed:

```
# Verify kubectl functionality
kubectl get nodes
```

```
# Install Pod Network (Weave)
kubectl apply -f \
https://github.com/weaveworks/weave
    /releases/download/v2.8.1/weave-daemonset-k8s-1.9.yaml
```

Once done the output should resemble the one shown in figure 6

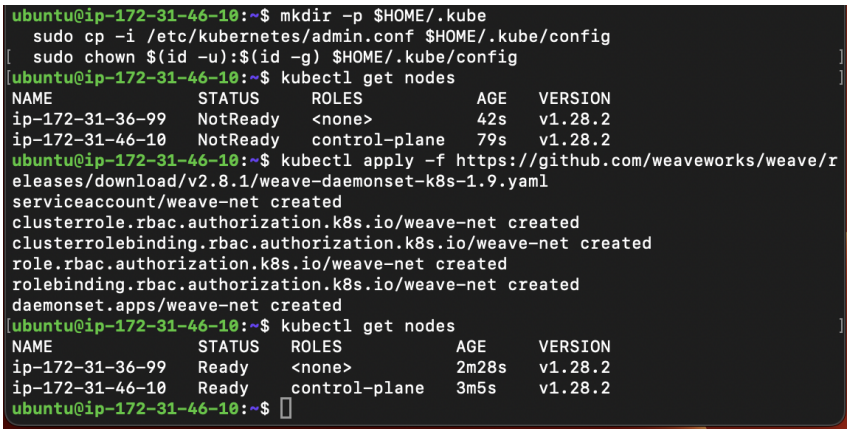

Figure 6: Cluster Ready

4 Install Benchmarking operator

In this section, we focus on the installation process of the benchmarking operator, a crucial component for executing performance tests using sysbench within the Kubernetes environment.

The installation process is depicted in Figure 7.

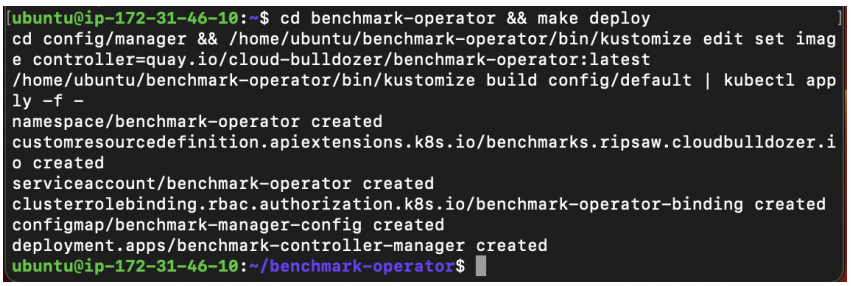

Figure 7: Install Benchmarking operator

4.1 Running sysbench tests

To initiate the sysbench tests, execute the following command. The YAML configuration file utilized is sourced from the standard templates provided by the benchmark operator.

```
kubectl apply -f config/samples/sysbench/cr.yaml
```

This command triggers the deployment of the sysbench workload with the specified configurations, allowing for comprehensive benchmarking within the Kubernetes cluster.

5 Install K-bench operator

This section details the installation process for the K-bench operator, an essential tool for conducting benchmark tests within a Kubernetes environment.

Before installing the K-bench operator, ensure that Kustomize is installed. Execute the following command to install Kustomize:

```
curl -s "https://raw.githubusercontent.com/kubernetes-sigs/\
    kustomize/master/hack/install_kustomize.sh"  | bash
```

Clone the K-bench repository from GitHub and navigate to the repository directory. Execute the installation script and move the K-bench executable to a directory included in the system's PATH.

```
git clone https://github.com/vmware-tanzu/k-bench.git
cd k-bench
go get -u github.com/kubernetes/kompose
go get golang.org/x/oauth2/google@v0.0.0-20200107190931-bf48bf16ab8d
./install.sh

sudo mv /home/ubuntu/go/bin/kbench
```

5.1 Running Tests with K-bench

To perform benchmark tests using K-bench, use the provided run scripts. Here are examples of running different tests:

```
./run.sh -r "kbench-run-dp_redis-1"  -t "dp_redis" -o "./"   \
| tee test-dp_redis.txt
./run.sh -r "kbench-run-dp_redis_service-1"  -t "dp_redis_service" -o "./" \
| tee test-dp_redis_service.txt
./run.sh -r "kbench-run-cp_heavy_12client-1"  -t "cp_heavy_12client" -o "./" \
| tee test-cp_heavy_12client.txt
```