Automatic Your Wordpress Application with Helm

On Kubernetes Cluster

Nugi Abdiansyah

openSUSE End User

✉ nugi@abdiansyah.com

nugiabdiansyah

@nugiabdiansyah

openSUSE Asia Summit 2019

Get this slide at https://bit.ly/35dKLum

My Profile

My Name is Nugi Abdiansyah, work at PT Excellent Infotama Kreasindo as Product Manager.

I am very enthusiast with Linux server, i did use all linux variant like Ubuntu, CentOS, Redhat, SUSE but almost of all i very like openSUSE.

I learning a lot of a thing related about Information Technology. I expert in manage email service with Zimbra Mail Server and manage website using WHM/cPanel

I also write a portfolio on my Blog at https://nugi.biz in Bahasa.

REFERENCES

- Kubernetes Documentation: https://kubernetes.io/docs/home/
- Kubernetes Cookbook - Sébastien Goasguen, Michael Hausenblas, 2018
- Kubernetes in Actions - Marko Lukša, 2017
- Helm.sh Documentation: https://helm.sh/docs/
- Wordpress Codex: https://codex.wordpress.org/

WORDPRESS (1)

An amazing open source sofware also known as Content Management System (CMS) with trusted by the best 34% of the web uses WordPress, from hobby blogs to the biggest news sites online.

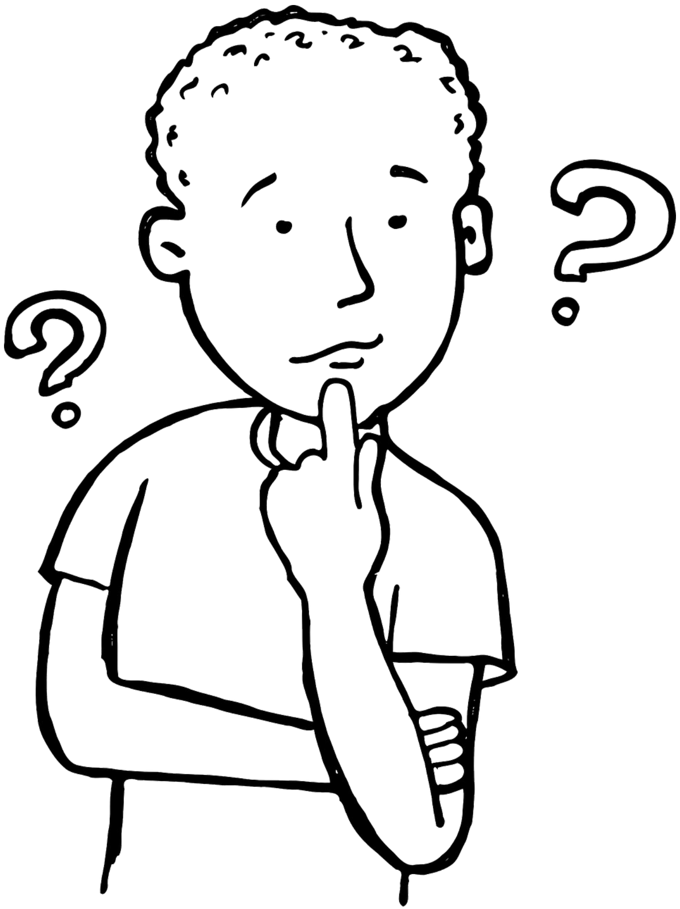
Image from: wordpress.org

WORDPRESS (2)

Wordpress also featured with Beautiful designs, powerful features, and the freedom to build anything you want. WordPress is both free and priceless at the same time.

Image from: wordpress.org

What is difficulty?



1) Wordpress is Easy to Install
2) We need server for each wordpress
3) We use Kubernetes as a server

So if we are using Kubernetes, we need Helm to solved our problem.

Let’s rocks!

Image from: pixabay.com

What Is Kubernetes?

- "Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications."
- Kubernetes comes from the Greek word κυβερνήτης:, which means helmsman or ship pilot.
- People pronounce Kubernetes in a few different ways. Many pronounce it as Koo-ber-nay-tace, while others pronounce it more like Koo-ber-netties.
- Kubernetes is also referred to as k8s, as there are 8 characters between k and s.
- Kubernetes was started by Google and, with its v1.0 release in July 2015, Google donated it to the Cloud Native Computing Foundation (CNCF).

Kubernetes Features

- Automatic binpacking
- Self-healing
- Horizontal scaling
- Service discovery and Load balancing
- Automated rollouts and rollbacks
- Secrets and configuration management
- Storage orchestration
- Batch execution

HELM

- Helm is a tool that streamlines installing and managing Kubernetes applications.
- Think of it like zypper/apt/yum/homebrew for Kubernetes.
- Helm has two parts: a client (helm) and a server (tiller) Tiller runs inside of your Kubernetes cluster, and manages releases (installations) of your charts.
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application.

Prerequisite

- 2 VM openSUSE Leap 15.1
- Minimum RAM 2 GB each server
- Minimum vCPU 2 GB each server
- Enable Virtualization Technology (Intel VT/AMD V)
- 1 NIC each server
- 2 IP Private

Get your own openSUSE ISO

- International:

  http://download.opensuse.org/distribution/leap/15.1/iso/openSUSE-Leap-15.1-DVD-x86_64.iso

- Local ID:

  https://repo.opensuse.id/distribution/openSUSE-stable/iso/openSUSE-Leap-15.1-DVD-x86_64.iso

Setup Server

- Get latest service:

```
zypper ref && zypper dup -y
```

- Setup /etc/hosts:

```
127.0.0.1 localhost
192.168.99.101 kube-master.nugi.biz kube-master
192.168.99.102 kube-worker0.nugi.biz kube-worker0
```

Install Docker

- Install Docker CE 18.09.6, run command:

```
zypper in docker-18.09.6_ce
```

- Add some line on /etc/docker/daemon.json after { like this:

```
…
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "storage-driver": "overlay2",
...
```

Restart & Enable Service Docker

- Restart Docker Service, run command:

  systemctl restart docker

- Add Docker Service to automatic running when boot:

  systemctl enable docker

modprobe & sysctl

- Run modprobe command:

```
modprobe overlay
modprobe br_netfilter
```

- Add Docker Service to automatic running when boot:

```
net.ipv4.ip_forward = 1
net.ipv4.conf.all.forwarding = 1
net.bridge.bridge-nf-call-iptables = 1
```

- Run this command to apply:

```
sysctl -p
```

Add k8s Repository

- Add newk8s repository, Run command:

```
zypper addrepo --type yum --gpgcheck-strict --refresh https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64 google-k8s
```

- Add gpg key for repository, run command:

```
rpm --import https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

```
rpm --import https://packages.cloud.google.com/yum/doc/yum-key.gpg
```

- Refresh repository, run command:

```
zypper refresh google-k8s
```

Install kubeadm,kubectl & kubelet

- Install a bundling package to completed your kubernetes cluster:

```
zypper in kubelet-1.15.4-0 kubernetes-cni kubeadm-1.15.4-0 cri-tools kubectl-1.15.4-0 socat
```

- Ignoring conntrack breakout, just pick:

```
 Solution 2: break kubelet-1.15.4-0.x86_64 by ignoring some of its dependencies
Choose from above solutions by number or skip, retry or cancel [1/2/s/r/c] (c): 2
...
 Solution 3: break kubelet-1.13.3-0.x86_64 by ignoring some of its dependencies
Choose from above solutions by number or skip, retry or cancel [1/2/3/s/r/c] (c): 3
```

Disable Swap & Enable kubelet

- Kubernetes don’t support swap, we must disable swap manually to install kubernetes.
- Check swap with command:

  swapon -s
- Disable swap with command:

  swapoff -a
- Enable kubelet service on boot:

  systemctl enable kubelet

Kubeadm init (do only in master)

- Run kubeadm init like this:

```
kubeadm init --pod-network-cidr=10.244.14.0/16
```

- Run this command to working with kubectl:

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

- Watch pods creating process with command:

```
kubectl get pods --all-namespaces --watch
```

- Hint: Copy kubeadm join to active a worker

Install flannel (do only in master)

- Install flannel to connect the cluster with command:

  ```
  wget https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
  ```

  ```
  kubectl apply -f kube-flannel.yml
  ```

- Watch pods creating process with command:

  ```
  kubectl get pods --all-namespaces --watch
  ```

Kubeadm join (do only in worker)

- Run kubeadm join with the last copied after kubeadm init from admin. Like this command:

  ```
  kubeadm join 192.168.99.101:6443 --token x8wb20.f8czwt7sdxbvprdh --discovery-token-ca-cert-hash sha256:5226d23fa710d7ca86443ca52665c5b7d0526aced2985da4b88b3cfdcd0deb97
  ```

- After join get to the master and run this command to check the worker is ready:

  ```
  kubectl get nodes
  ```

Install Helm (do only in master)

- Install helm with this command:

```
wget -c https://get.helm.sh/helm-v2.14.3-linux-amd64.tar.gz
```

```
tar xzvf helm-v2.14.3-linux-amd64.tar.gz
```

```
mv linux-amd64/helm /usr/local/bin/helm
```

- Initializing helm for the first time:

```
helm init
```

Install Tiller (do only in master)

- Install tiller with this command:

```
kubectl create serviceaccount --namespace kube-system tiller
kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller
kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'
```

- Check helm and tiller version are the same:

```
helm version
```

Get Your Own NFS Server

- Create a nfs server with helm before install wordpress with this command:

```
wget -c https://raw.githubusercontent.com/nugiabdiansyah/k8s-opensuse/master/nfs-values.yaml
```

```
helm install --name nfs-wordpress -f nfs-values.yaml stable/nfs-server-provisioner
```

- Install nfs-client on master & worker with command:

```
zypper in nfs-client
```

Install Wordpress with Helm

- Easy install wordpress server with helm with this command:

```
wget -c https://raw.githubusercontent.com/nugiabdiansyah/k8s-opensuse/master/wordpress.yaml
```

```
helm install --name wordpress-nugi -f wordpress.yaml --set persistence.storageClass=nfs --set persistence.size=2Gi --set service.type=NodePort stable/wordpress --set mariadb.master.persistence.storageClass=nfs
```

- Check pods until running with command:

```
kubectl get pods wordpress-nugi
```

Ingress Nginx (1)

- Install ingress-nginx to expose our wordpress applications, with this command:

  kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mandatory.yaml

  kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/provider/baremetal/service-nodeport.yaml

- And create ingress-rewrite with command:

  kubectl create -f https://github.com/nugiabdiansyah/k8s-opensuse/raw/master/ingress.yaml

Ingress Nginx (2)

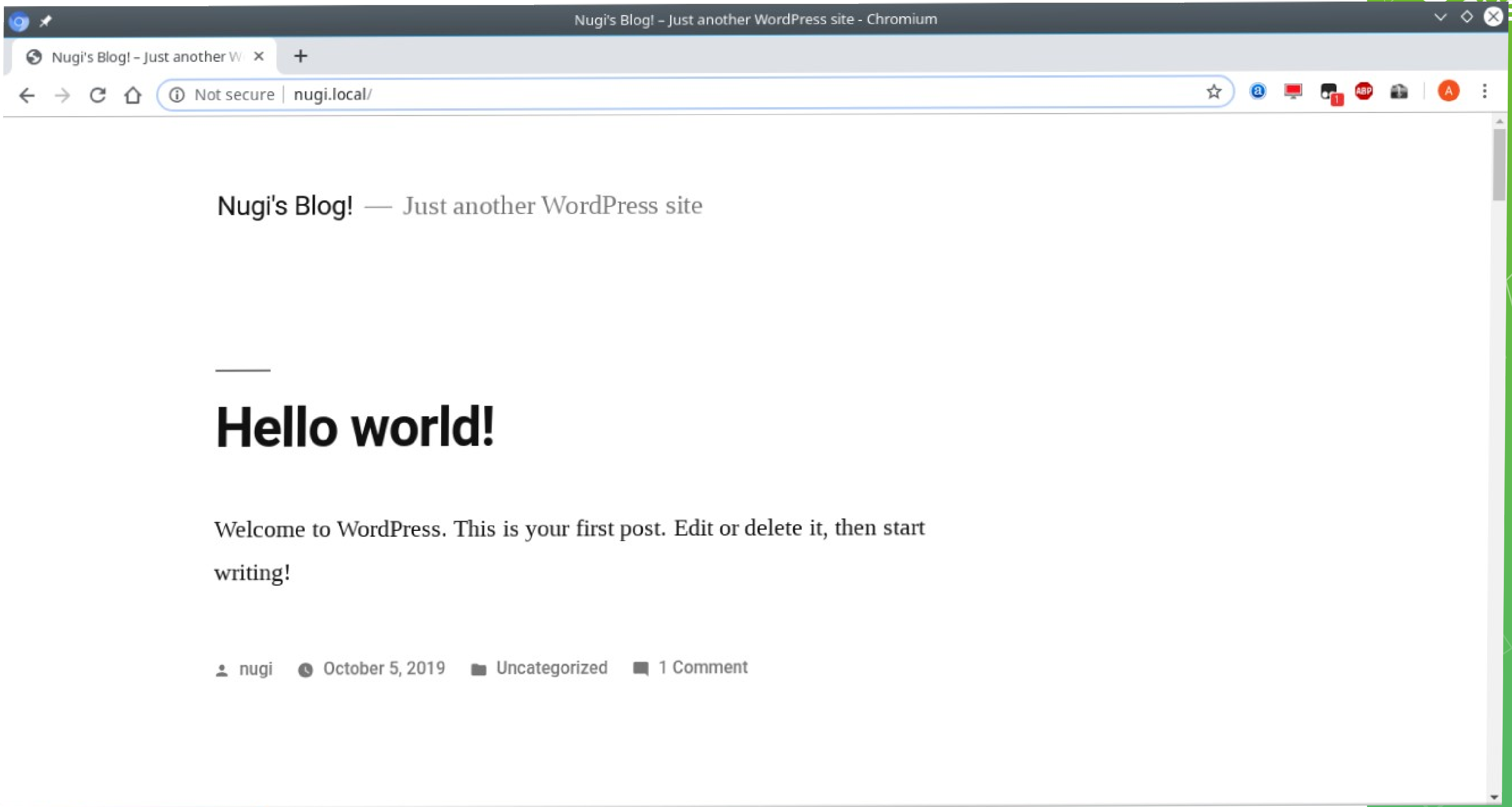
- Edit your ingress-service with command:

  kubectl edit svc -n ingress-nginx ingress-nginx

- Add externalIPs after this line:

```
...
 selector:
  app: ingress-nginx
 externalIPs:
 - IP-MASTER-LOCAL
...
```

Access Your Wordpress



Nugi's Blog! – Just another WordPress site - Chromium
Nugi's Blog! – Just another W
Not secure
nugi.local/

Nugi's Blog! — Just another WordPress site

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

nugi October 5, 2019 Uncategorized 1 Comment

Thank you for your attention

openSUSE®