**Các bước cài đặt Matomo trên Kubernetes**

**Bước 1: Tạo Helm Chart**

**-Cài đặt Helm trên ubuntu:**

Apt update -y

Apt install helm -y

**-Tạo Helm chart:**

+Các thành phần để cài đặt thành công Matomo: Matomo app,database(mysql), matomo volume(NFS),Database volume(NFS)

Helm create matomo

Helm create mysql

Helm create matomo-vol

Helm create db-vol

**Bước 2: Viết file template và values cho helm chart**

**Template của các service gồm các file yaml:**

- Matomo: service , deployment

- Mysql: service, deployment, secret

- Matomo-vol: pv, pvcs

- Db-vol: pv,pvc

- Các thông số metadata của service sử dụng releasename: {{ .Release.Name }}

- Các thông số phần template sử dụng giá trị trong file values.yaml: {{ .Values.x }}

**Bước 3: Cài đặt nginx ingress controller deamonset để đảm bảo nó chạy trên tất cả các node:**

Git clone <https://github.com/nginxinc/kubernetes-ingress/>

Cd kubernetes-ingress/deployment/helmchart

kubectl apply -f common/ns-and-sa.yaml

kubectl apply -f common/default-server-secret.yaml

kubectl apply -f common/nginx-config.yaml

kubectl apply -f rbac/rbac.yaml

kubectl apply -f daemon-set/nginx-ingress.yaml

**Bước 4: Tạo ingress resource để sử dụng ingress controller bằng file yaml,host là tên miền muốn trỏ đến,backend là service matomo**

Để truy cập được vào ingress trong môi trường baremetal cần có 1 proxy đứng trước và trỏ vào nodeport của service nginx-controller

**Các lệnh helm:**

**- Cài đặt các helm chart đã tạo:**

Helm install release\_name path\_of\_chart

VD: helm install matomo-app matomo/

**- Cài đặt helm chart từ 1 repo có sẵn:**

Helm repo add chart\_name link\_repo

VD: helm repo add nginx-stable <https://helm.nginx.com/stable>

Helm repo update

**- Upgrade 1 helm chart khi có chỉnh sửa trong file:**

Helm upgrade release\_name path\_of\_chart

**- Xem các chart đã cài đặt:**

Helm list

**- Gỡ cài đặt 1 chart:**

Helm uninstall release\_name

Các lệnh kubernetes:

**- Khởi tạo 1 Cluster:**

Trên node master: Kubeadm init --pod-network-cidr=192.168.0.0/16

Trên các node worker: copy lệnh chưa token sinh ra trên node master khi init

**- Cài đặt plugin mạng Calico**

kubectl create -f <https://projectcalico.docs.tigera.io/manifests/tigera-> operator.yaml

kubectl create -f <https://projectcalico.docs.tigera.io/manifests/custom-> resources.yaml

**- Reset cluster:**

Trên tất cả các node: kubeadm reset

**- Cài đặt 1 resouces dựa trên manifest(file yaml):**

Kubectl apply -f file\_name.yaml

VD: kubectl apply -f deploymatomo.yaml

**- Xem các tài nguyên đã cài đặt:**

Kubectl get all

Kubectl get all -o wide để hiện thị chi tiết

Kubectl get nodes -o wide : xem tất cả các node trong cluster

Kubectl get deamonset: xem các daemonset trong cluster

Kubectl get all -o wide -n namespace\_name : xem các tài nguyên trong namespace được chỉ ra.

Kubectl get secret

Kubectl get storageclass

Kubectl get ingress

Kubectl get ingressclass

**- Xóa các tài nguyên:**

Kubectl delete resource\_type resource\_name

Hoặc: kubectl delete resource\_type/resource\_name

Kubectl delete resource\_name -n namespace\_name

VD: kubectl delete pod/pod\_12345

Kubectl delete deployment/nginx

Hoặc: kubectl delete deployment nginx

Kubectl delete namespace nginx

**- Xem thông số của 1 resource:**

Kubectl describe resource\_type resource\_name

Hoặc: kubectl describe resource\_type/resource\_name

VD: Kubectl describe pod/pod\_12345

Kubectl describe service matomo-app

**- Xem log của 1 pod:**

Kubectl logs pod pod\_name

VD: kubectl logs pod pod\_12345 hoặc kubectl logs pod/pod\_12345

**- Xem các event xảy ra trong cluster:**

Kubectl get ev

**- Vào shell của 1 pod chỉ có 1 container:**

kubectl exec --stdin --tty pod\_name -- /bin/bash

**- Vào shell của 1 pod có nhiều container:**

Kubectl exec -I -t pod\_name --container container\_name -- /bin/bash

**Cài đặt k9s để quản lí kubernets cluster chạy trên docker:**

docker run --rm -it -v $KUBECONFIG:/root/.kube/config quay.io/derailed/k9s

**Cài đặt metrics-server để thu thập thông tin CPU-RAM:**

Wget <https://github.com/kubernetes-sigs/metrics-server.git>

Sau khi tải xong sẽ có file component.yaml

Sửa file này,ở phần args,thêm phần

- --kubelet-insecure-tls

Sau khi sửa kubectl apply -f component.yaml