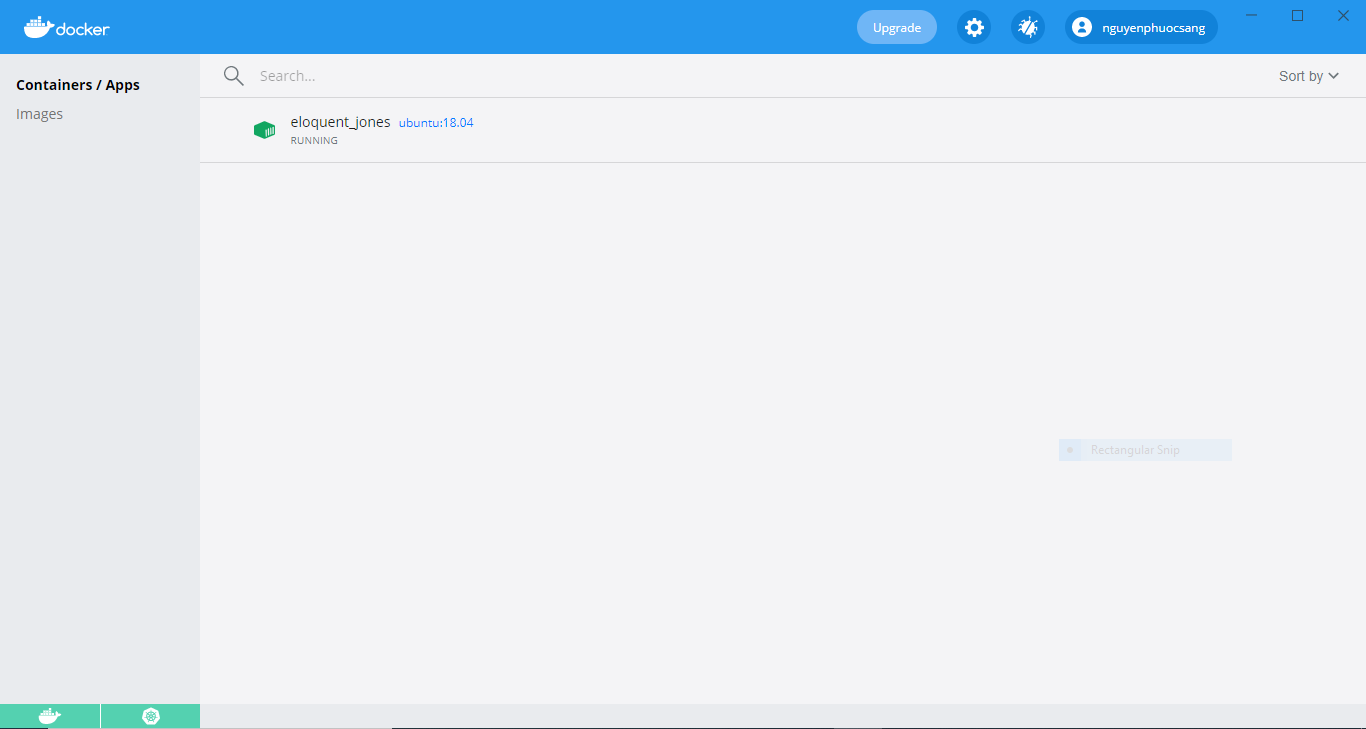
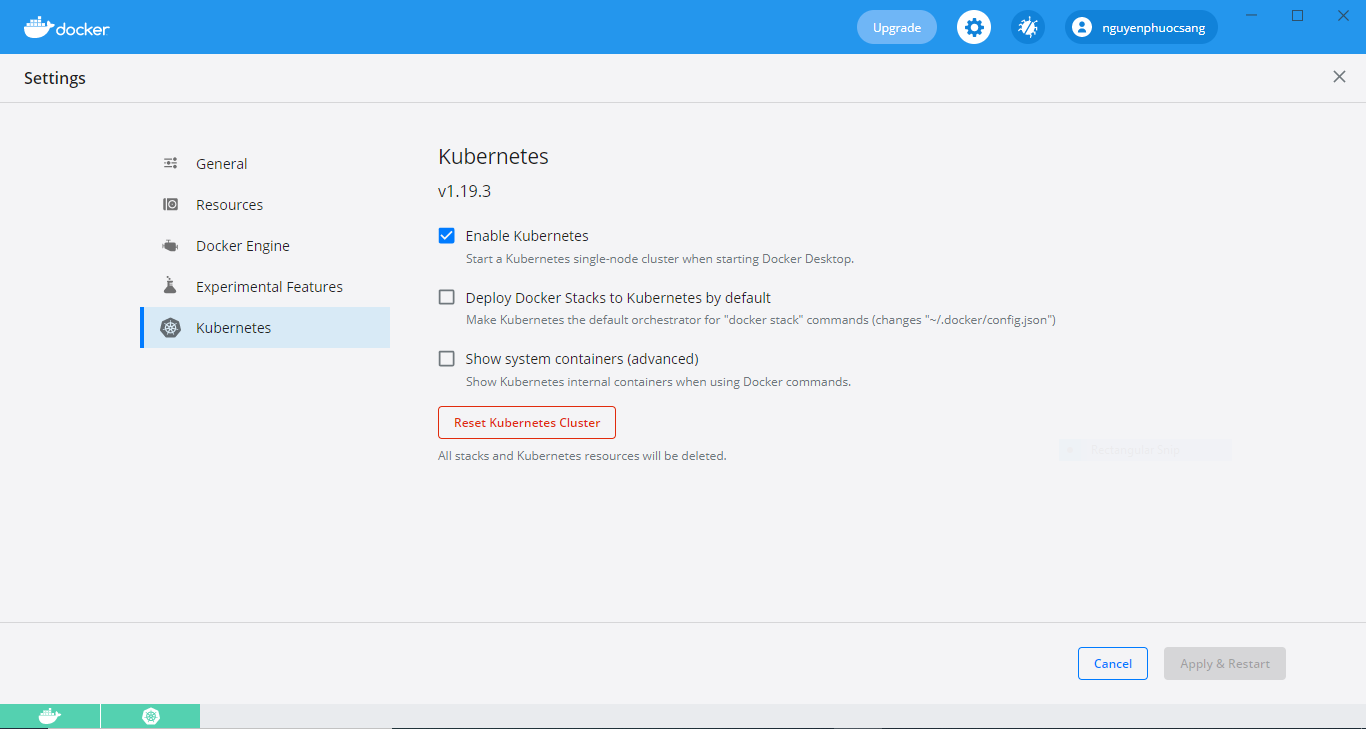
# Nguyễn Phước Sang – 17133055

# Nguyễn An Bình – 15110013

# Triển khai ứng dụng PHP Guestbook với Redis

##### Đầu tiên cần cài đặt docker desktop, ubuntu18.04 và khởi động Kubernetes





Khởi động Redis Master

Ứng dụng guestbook sử dụng Redis để lưu trữ dữ liệu của nó. Nó ghi dữ liệu của nó vào một cá thể chính của Redis và đọc dữ liệu từ nhiều cá thể Redis slave.

Tạo triển khai Redis Master

Tệp kê khai, bao gồm bên dưới, chỉ định bộ điều khiển Triển khai chạy một Pod chính Redis bản sao duy nhất.

[application/guestbook/redis-master-deployment.yaml](https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/guestbook/redis-master-deployment.yaml)

**apiVersion**: apps/v1

**kind**: Deployment

**metadata**:

**name**: redis-master

**labels**:

**app**: redis

**spec**:

**selector**:

**matchLabels**:

**app**: redis

**role**: master

**tier**: backend

**replicas**: 1

**template**:

**metadata**:

**labels**:

**app**: redis

**role**: master

**tier**: backend

**spec**:

**containers**:

- **name**: master

**image: k8s.gcr.io/redis:e2e # or just image**: redis

**resources**:

**requests**:

**cpu**: 100m

**memory**: 100Mi

**ports**:

- **containerPort**: 6379

1. Khởi chạy cửa sổ dòng lệnh trong thư mục bạn đã tải xuống tệp kê khai.
2. Áp dụng Triển khai Redis Master từ tệp redis-master-deployment.yaml:
3. kubectl apply -f https://k8s.io/examples/application/guestbook/redis-master-deployment.yaml
4. Truy vấn danh sách các Pod để xác minh rằng Redis Master Pod đang chạy:
5. kubectl get pods

Câu trả lời phải tương tự như sau:

NAME READY STATUS RESTARTS AGE

redis-master-1068406935-3lswp 1/1 Running 0 28s

1. Chạy lệnh sau để xem nhật ký từ Redis Master Pod:
2. kubectl logs -f POD-NAME

**Lưu ý:** Thay POD-NAME bằng tên Pod của bạn.

Tạo dịch vụ Redis Master

Ứng dụng guestbook cần giao tiếp với Redis master để ghi dữ liệu của nó. Bạn cần áp dụng một [Dịch vụ](https://kubernetes.io/docs/concepts/services-networking/service/) để ủy quyền lưu lượng truy cập đến Redis master Pod. Dịch vụ xác định chính sách để truy cập các Pod.

[application/guestbook/redis-master-service.yaml](https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/guestbook/redis-master-service.yaml)

**apiVersion**: v1

**kind**: Service

**metadata**:

**name**: redis-master

**labels**:

**app**: redis

**role**: master

**tier**: backend

**spec**:

**ports**:

- **name**: redis

**port**: 6379

**targetPort**: 6379

**selector**:

**app**: redis

**role**: master

**tier**: backend

1. Áp dụng Redis Master Service từ tệp redis-master-service.yaml sau :
2. kubectl apply -f https://k8s.io/examples/application/guestbook/redis-master-service.yaml
3. Truy vấn danh sách Dịch vụ để xác minh rằng Dịch vụ Redis Master đang chạy:
4. kubectl get service

Câu trả lời phải tương tự như sau:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 1m

redis-master ClusterIP 10.0.0.151 <none> 6379/TCP 8s

**Lưu ý:** Tệp kê khai này tạo một Dịch vụ có tên redis-mastervới một tập hợp các nhãn phù hợp với các nhãn đã xác định trước đó, vì vậy Dịch vụ định tuyến lưu lượng mạng đến Redis master Pod.

Khởi động Redis Slaves

Mặc dù Redis master là một Pod duy nhất, nhưng bạn có thể làm cho nó luôn sẵn sàng để đáp ứng nhu cầu lưu lượng bằng cách thêm các Redis slave bản sao.

Tạo triển khai Redis Slave

Quy mô triển khai dựa trên cấu hình được đặt trong tệp kê khai. Trong trường hợp này, đối tượng Triển khai chỉ định hai bản sao.

Nếu không có bất kỳ bản sao nào đang chạy, Triển khai này sẽ bắt đầu hai bản sao trên cụm vùng chứa của bạn. Ngược lại, nếu có nhiều hơn hai bản sao đang chạy, nó sẽ giảm quy mô cho đến khi hai bản sao đang chạy.

[application/guestbook/redis-slave-deployment.yaml](https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/guestbook/redis-slave-deployment.yaml)

**apiVersion**: apps/v1

**kind**: Deployment

**metadata**:

**name**: redis-slave

**labels**:

**app**: redis

**spec**:

**selector**:

**matchLabels**:

**app**: redis

**role**: slave

**tier**: backend

**replicas**: 2

**template**:

**metadata**:

**labels**:

**app**: redis

**role**: slave

**tier**: backend

**spec**:

**containers**:

- **name**: slave

**image**: gcr.io/google\_samples/gb-redisslave:v3

**resources**:

**requests**:

**cpu**: 100m

**memory**: 100Mi

**env**:

- **name**: GET\_HOSTS\_FROM

**value**: dns

*# Using `GET\_HOSTS\_FROM=dns` requires your cluster to*

*# provide a dns service. As of Kubernetes 1.3, DNS is a built-in*

*# service launched automatically. However, if the cluster you are using*

*# does not have a built-in DNS service, you can instead*

*# access an environment variable to find the master*

*# service's host. To do so, comment out the 'value: dns' line above, and*

*# uncomment the line below:*

*# value: env*

**ports**:

- **containerPort**: 6379

1. Áp dụng Triển khai Redis Slave từ redis-slave-deployment.yamltệp:
2. kubectl apply -f https://k8s.io/examples/application/guestbook/redis-slave-deployment.yaml
3. Truy vấn danh sách các Pod để xác minh rằng Redis Slave Pods đang chạy:
4. kubectl get pods

Câu trả lời phải tương tự như sau:

NAME READY STATUS RESTARTS AGE

redis-master-1068406935-3lswp 1/1 Running 0 1m

redis-slave-2005841000-fpvqc 0/1 ContainerCreating 0 6s

redis-slave-2005841000-phfv9 0/1 ContainerCreating 0 6s

Tạo dịch vụ Redis slave

Ứng dụng guestbook cần giao tiếp với Redis slave để đọc dữ liệu. Để làm cho Redis slave có thể phát hiện được, bạn cần thiết lập Dịch vụ. Một Dịch vụ cung cấp khả năng cân bằng tải trong suốt cho một bộ Pod.

[application/guestbook/redis-slave-service.yaml](https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/guestbook/redis-slave-service.yaml)

**apiVersion**: v1

**kind**: Service

**metadata**:

**name**: redis-slave

**labels**:

**app**: redis

**role**: slave

**tier**: backend

**spec**:

**ports**:

- **port**: 6379

**selector**:

**app**: redis

**role**: slave

**tier**: backend

1. Áp dụng Redis Slave Service từ redis-slave-service.yamltệp sau :
2. kubectl apply -f https://k8s.io/examples/application/guestbook/redis-slave-service.yaml
3. Truy vấn danh sách Dịch vụ để xác minh rằng dịch vụ Redis slave đang chạy:
4. kubectl get services

Câu trả lời phải tương tự như sau:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 2m

redis-master ClusterIP 10.0.0.151 <none> 6379/TCP 1m

redis-slave ClusterIP 10.0.0.223 <none> 6379/TCP 6s

Thiết lập và hiển thị giao diện guestbook

Ứng dụng guestbook có giao diện người dùng web phục vụ các yêu cầu HTTP được viết bằng PHP. Nó được cấu hình để kết nối với redis-masterDịch vụ cho các yêu cầu ghi và redis-slavedịch vụ cho các yêu cầu Đọc.

Tạo triển khai giao diện người dùng guestbook

[application/guestbook/frontend-deployment.yaml](https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/guestbook/frontend-deployment.yaml)

**apiVersion**: apps/v1

**kind**: Deployment

**metadata**:

**name**: frontend

**labels**:

**app**: guestbook

**spec**:

**selector**:

**matchLabels**:

**app**: guestbook

**tier**: frontend

**replicas**: 3

**template**:

**metadata**:

**labels**:

**app**: guestbook

**tier**: frontend

**spec**:

**containers**:

- **name**: php-redis

**image**: gcr.io/google-samples/gb-frontend:v4

**resources**:

**requests**:

**cpu**: 100m

**memory**: 100Mi

**env**:

- **name**: GET\_HOSTS\_FROM

**value**: dns

*# Using `GET\_HOSTS\_FROM=dns` requires your cluster to*

*# provide a dns service. As of Kubernetes 1.3, DNS is a built-in*

*# service launched automatically. However, if the cluster you are using*

*# does not have a built-in DNS service, you can instead*

*# access an environment variable to find the master*

*# service's host. To do so, comment out the 'value: dns' line above, and*

*# uncomment the line below:*

*# value: env*

**ports**:

- **containerPort**: 80

1. Áp dụng Triển khai giao diện người dùng từ tệp frontend-deployment.yaml:
2. kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-deployment.yaml
3. Truy vấn danh sách các Pod để xác minh rằng ba bản sao giao diện người dùng đang chạy:
4. kubectl get pods -l app=guestbook -l tier=frontend

Câu trả lời phải tương tự như sau:

NAME READY STATUS RESTARTS AGE

frontend-3823415956-dsvc5 1/1 Running 0 54s

frontend-3823415956-k22zn 1/1 Running 0 54s

frontend-3823415956-w9gbt 1/1 Running 0 54s

Tạo dịch vụ giao diện người dùng

Các redis-slavevà redis-masterDịch vụ bạn đã áp dụng chỉ có thể truy cập được trong cụm vùng chứa vì loại mặc định cho Dịch vụ là [ClusterIP](https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services---service-types) . ClusterIPcung cấp một địa chỉ IP duy nhất cho tập hợp các Pod mà Dịch vụ đang trỏ đến. Địa chỉ IP này chỉ có thể truy cập được trong cụm.

Nếu bạn muốn khách có thể truy cập guestbook của mình, bạn phải định cấu hình Dịch vụ giao diện người dùng để hiển thị bên ngoài, vì vậy khách hàng có thể yêu cầu Dịch vụ từ bên ngoài cụm vùng chứa. Minikube chỉ có thể hiển thị Dịch vụ thông qua NodePort.

**Lưu ý:** Một số nhà cung cấp đám mây, như Google Compute Engine hoặc Google Kubernetes Engine, hỗ trợ các bộ cân bằng tải bên ngoài. Nếu nhà cung cấp dịch vụ đám mây của bạn hỗ trợ bộ cân bằng tải và bạn muốn sử dụng nó, chỉ cần xóa hoặc nhận xét type: NodePort, và bỏ ghi chú type: LoadBalancer.

[application/guestbook/frontend-service.yaml](https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/guestbook/frontend-service.yaml)

**apiVersion**: v1

**kind**: Service

**metadata**:

**name**: frontend

**labels**:

**app**: guestbook

**tier**: frontend

**spec**:

*# comment or delete the following line if you want to use a LoadBalancer*

**type**: NodePort

*# if your cluster supports it, uncomment the following to automatically create*

*# an external load-balanced IP for the frontend service.*

*# type: LoadBalancer*

**ports**:

- **port**: 80

**selector**:

**app**: guestbook

**tier**: frontend

1. Áp dụng Dịch vụ giao diện người dùng từ frontend-service.yamltệp:
2. kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml
3. Truy vấn danh sách Dịch vụ để xác minh rằng Dịch vụ giao diện người dùng đang chạy:
4. kubectl get services

Câu trả lời phải tương tự như sau:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

frontend NodePort 10.0.0.112 <none> 80:31323/TCP 6s

kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 4m

redis-master ClusterIP 10.0.0.151 <none> 6379/TCP 2m

redis-slave ClusterIP 10.0.0.223 <none> 6379/TCP 1m

Xem Dịch vụ giao diện người dùng qua NodePort

Nếu bạn đã triển khai ứng dụng này cho Minikube hoặc một cụm cục bộ, bạn cần tìm địa chỉ IP để xem Guestbook của mình.

1. Chạy lệnh sau để lấy địa chỉ IP cho Dịch vụ giao diện người dùng.
2. minikube service frontend --url

Câu trả lời phải tương tự như sau:

http://192.168.99.100:31323

1. Sao chép địa chỉ IP và tải trang trong trình duyệt của bạn để xem guestbook của bạn.

Xem Dịch vụ giao diện người dùng qua LoadBalancer

Nếu bạn đã triển khai frontend-service.yamltệp kê khai với loại: LoadBalancerbạn cần tìm địa chỉ IP để xem Guestbook của mình.

1. Chạy lệnh sau để lấy địa chỉ IP cho Dịch vụ giao diện người dùng.
2. kubectl get service frontend

Câu trả lời phải tương tự như sau:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

frontend ClusterIP 10.51.242.136 109.197.92.229 80:32372/TCP 1m

1. Sao chép địa chỉ IP bên ngoài và tải trang trong trình duyệt của bạn để xem guestbook của bạn.

Quy mô giao diện người dùng web

Mở rộng quy mô lên hoặc giảm xuống dễ dàng vì máy chủ của bạn được định nghĩa là một Dịch vụ sử dụng bộ điều khiển Triển khai.

1. Chạy lệnh sau để tăng số lượng các Pod giao diện người dùng:
2. kubectl scale deployment frontend --replicas=5
3. Truy vấn danh sách các Pod để xác minh số lượng các Pod giao diện người dùng đang chạy:
4. kubectl get pods

Câu trả lời sẽ giống như sau:

NAME READY STATUS RESTARTS AGE

frontend-3823415956-70qj5 1/1 Running 0 5s

frontend-3823415956-dsvc5 1/1 Running 0 54m

frontend-3823415956-k22zn 1/1 Running 0 54m

frontend-3823415956-w9gbt 1/1 Running 0 54m

frontend-3823415956-x2pld 1/1 Running 0 5s

redis-master-1068406935-3lswp 1/1 Running 0 56m

redis-slave-2005841000-fpvqc 1/1 Running 0 55m

redis-slave-2005841000-phfv9 1/1 Running 0 55m

1. Chạy lệnh sau để giảm số lượng Pod giao diện người dùng:
2. kubectl scale deployment frontend --replicas=2
3. Truy vấn danh sách các Pod để xác minh số lượng các Pod giao diện người dùng đang chạy:
4. kubectl get pods

Câu trả lời sẽ giống như sau:

NAME READY STATUS RESTARTS AGE

frontend-3823415956-k22zn 1/1 Running 0 1h

frontend-3823415956-w9gbt 1/1 Running 0 1h

redis-master-1068406935-3lswp 1/1 Running 0 1h

redis-slave-2005841000-fpvqc 1/1 Running 0 1h

redis-slave-2005841000-phfv9 1/1 Running 0 1h

Dọn dẹp

Xóa các Triển khai và Dịch vụ cũng xóa mọi Pod đang chạy. Sử dụng nhãn để xóa nhiều tài nguyên bằng một lệnh.

1. Chạy các lệnh sau để xóa tất cả Pod, Triển khai và Dịch vụ.
2. kubectl delete deployment -l app=redis
3. kubectl delete service -l app=redis
4. kubectl delete deployment -l app=guestbook
5. kubectl delete service -l app=guestbook

Các câu trả lời phải là:

deployment.apps "redis-master" deleted

deployment.apps "redis-slave" deleted

service "redis-master" deleted

service "redis-slave" deleted

deployment.apps "frontend" deleted

service "frontend" deleted

1. Truy vấn danh sách các Pod để xác minh rằng không có Pod nào đang chạy:
2. kubectl get pods

Câu trả lời phải là:

No resources found.