KUBERNETES ON MINIKUBE

Alessandro Bocci

name.surname@phd.unipi.it

Advanced Software Engineering (Lab)

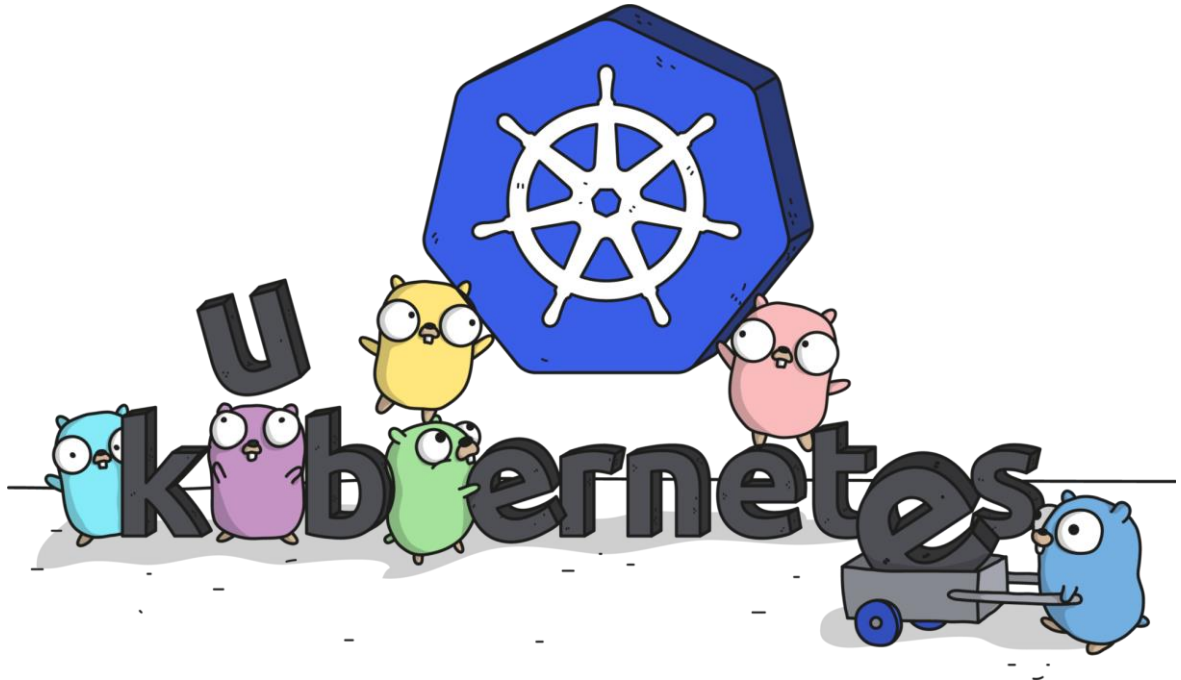
27/10/2023

What is minikube?

- Minikube is a tool that lets you run Kubernetes locally.
- It runs an all-in-one or a multi-node local Kubernetes cluster on your personal so that you can try out Kubernetes, or for daily development work.
- Minikube has an internal docker environment, images and containers of your system are different from the ones within it.

What will you do?

- Write microase2324 manifests for K8s Deployment and Services.
- Deploy and run them!
- Change some conditions and observe how K8s reacts.



Software Prerequisites

- Minikube and its base docker image `kicbase/stable:v0.0.40`
- Previous lab Docker images:
  - `alebocci/gateway:latest`
  - `alebocci/math-service:latest`
  - `alebocci/string-service:latest`

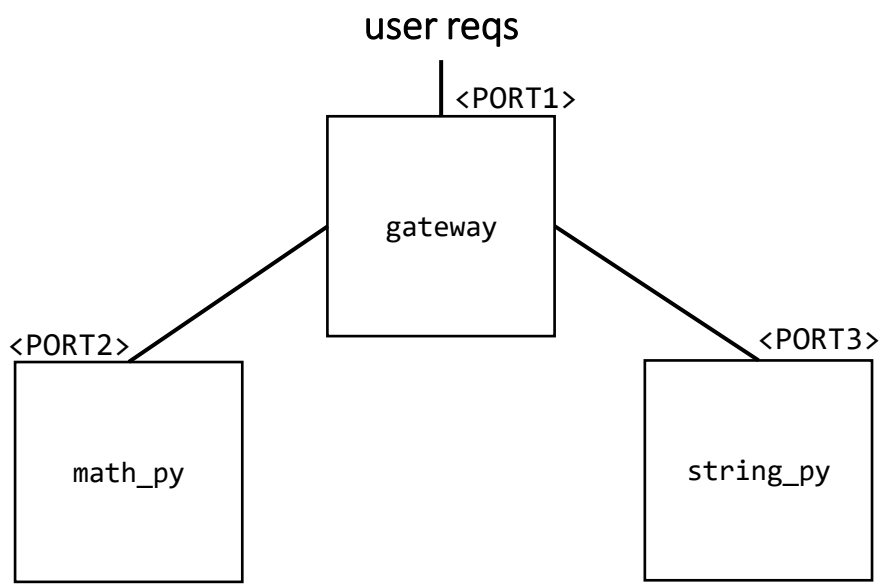
Today's Lab

PART ONE

1. Start minikube with 3 nodes.
2. Load in minikube `microase2324` docker images.
3. Complete the manifest for the deployment.
4. Complete the manifest for the services.
5. 'Apply' them with K8s.

PART TWO

1. Individuate the `gateway` endpoint.
2. Perform some operation (previous lab API calls).
3. Increase the number of replicas of the `math-service`.
4. Change the number of minikube nodes.



Minkube Useful commands

- `minikube start --nodes <number_of_nodes>`

  it starts a minikube cluster with `n_o_n` nodes
- `minikube stop`

  it stops the minikube cluster
- `minikube delete`

  it delete the minkube cluster
- `minikube image load <image> [--daemon]`

  it load a docker image from your local docker to minikube's docker
- `minikube service list`

  it shows the list of services running on the minikube cluster
- `minikube kubectl -- <command>`

  it allows to run a Kubernets (`kubetcl`) command

K8s Useful commands

- `kubectl get <resource>`

  it shows the status of <resource> (e.g. `pods`, `nodes`, `services`).

- `kubectl apply -f <manifest.yaml>`

  it creates or update a resource from a .yaml file.

- `kubectl drain <node_name> [--ignore-daemonsets]`

  it remove everything from the node <node_name>.

- `kubectl uncordon <node_name>`

  it marks <node_name> as schedulable.

- `kubectl delete <resource>`

  it delete the <resource>.

Before starting

You need the docker image of the `microase2324` application.
There are 3 alternatives:

- You have them from the last lab lecture;
- You pulled the requested images;
- You build them from the code of the .zip file.

In every case you have to load them in Minikube

(PART ONE, point 2. of the following slide)

PART ONE in detail

Download from the Moodle **`microase2324.zip`** and in the microase2324 folder:

1. Start the minikube cluster with 3 nodes (2 if your pc doesn't have enough resources).
2. Load in minikube the docker images of **`gateway`**, **`math-service`** and **`string-service`**.
3. Complete the **`deployment.yml`** file and for **`math-service`** and **`string-service`** give 2 replicas each. (You will have 3 pods and their replicas, one for each service).
4. Create and check the deployment (you should have 5 running containers after the setup time).
5. Complete the **`service.yml`** file giving to the **`gateway`** the correct type.
   (https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types)
6. Create and check the services (you should see your services running plus kubernetes).

PART TWO in detail

1. Obtain from minikube the URL of the `gateway`.
2. Try the deployed services e.g. (previous lecture examples):

`http://IP:PORT/math/add?a=2&b=1`	should return a JSON with a field `s = 3`
`http://IP:PORT/math/div?a=2&b=1`	should return a JSON with a field `s = 2`
`http://IP:PORT/math/div?a=2&b=0`	should return an error
`http://IP:PORT/str/concat?a=2&b=1`	should return a JSON with a field `s = "21"`
`http://IP:PORT/str/upper?a=ase`	should return a JSON with a field `s = "ASE"`
`http://IP:PORT/str/lower?a=aSE`	should return a JSON with a field `s = "ase"`

3. Check the pods and the nodes (`kubectl get pods -o wide --show-labels`)
4. Drain one node and check again.
5. Uncord one node and check again.
6. Add one replica to the math-service in the deployment file and apply it.
7. Check the pods and the nodes again.

BONUS STAGE!

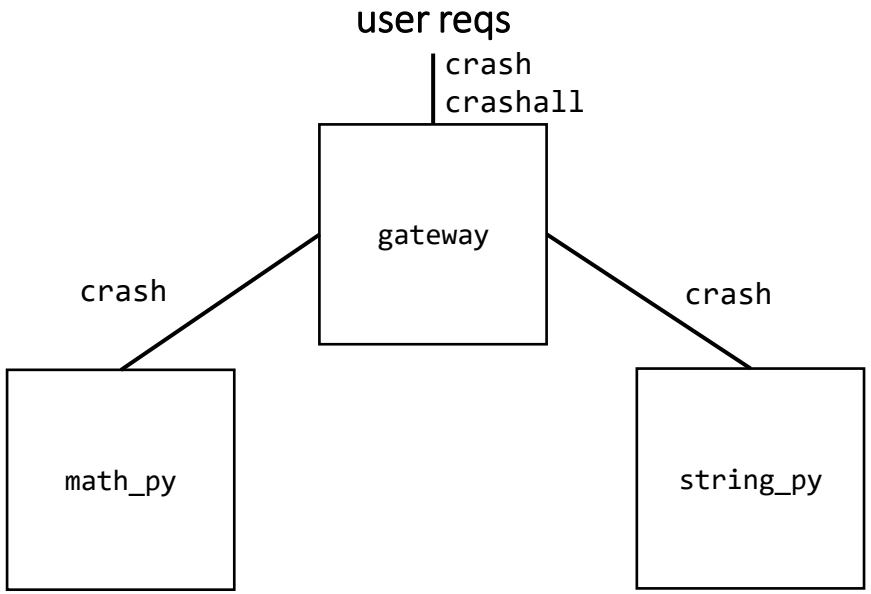

TEST YOUR MIGHT
LOW ATTACKS INCREASE POWER
BLOCK BUTTON STRIKES
RUBY
9
DIAMOND

Bonus stage

Modify the free services adding API calls that simulates ‘crashes’.

1. Add an endpoint `/crash` that stops the service execution.
2. Only for the **`gateway`**, add and endpoint `/crashall` that calls the `/crash` of the other services and stops the execution.
3. Build the docker images and load them in minikube.
4. Stop the previous deployment and services and substitute them with the new ones.
5. Call the new endpoints and check how K8s reacts.

What happen when you crash a service?



Lab take away

- ❑ Set up a Minkube multinode cluster
- ❑ Set up a K8s cluster with pods and services.
- ❑ Understand how K8s reacts to events.
  (e.g. node drain, adding replicas).

PART ONE: deployment manifest example

```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gateway
spec:
  replicas: 1
  selector:
    matchLabels:
      app: gateway
  template:
    metadata:
      labels:
        app: gateway
    spec:
      containers:
      - name: gateway
        imagePullPolicy: Never
        image: gateway
        ports:
        - containerPort: 5000
```

local loaded image

```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: math-service
spec:
  replicas: 2
  selector:
    matchLabels:
      app: math-service
  template:
    metadata:
      labels:
        app: math-service
    spec:
      containers:
      - name: math-service
        image: alebocci/math_py:latest
        ports:
          - containerPort: 5000
```

pulled image from repo

```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: string-service
spec:
  replicas: 2
  selector:
    matchLabels:
      app: string-service
  template:
    metadata:
      labels:
        app: string-service
    spec:
      containers:
      - name: string-service
        image: alebocci/string_py:latest
        ports:
          - containerPort: 5000
```

PART ONE: service manifest example

```
---
apiVersion: v1
kind: Service
metadata:
  name: gateway
spec:
  type: LoadBalancer|NodePort
  selector:
    app: gateway
  ports:
    - protocol: TCP
      port: 5000
      targetPort: 5000
```

```
---
apiVersion: v1
kind: Service
metadata:
  name: math-service
spec:
  selector:
    app: math-service
  ports:
    - protocol: TCP
      port: 5000
      targetPort: 5000
```

```
---
apiVersion: v1
kind: Service
metadata:
  name: string-service
spec:
  selector:
    app: string-service
  ports:
    - protocol: TCP
      port: 5000
      targetPort: 5000
```

PART ONE in detail solution

Download from the Moodle **`microase2324.zip`** and in the microase2324 folder:

1. `minikube start --nodes 3`
2. `minikube image load <image> [--daemon]`
   `<image>` = `gateway`, `math-service`, `string-service`
3. Previous slides + `minikube kubectl -- apply -f deployment.yaml`
4. `minikube kubectl -- get pods`
5. Previous slides + `minikube kubectl -- apply -f service.yaml`
6. `minikube kubectl -- get services`

PART TWO in detail solution

1. `minikube service list`
2. HTTP GET with browser, Postman, curl etc.
3. `minikube kubectl -- get pods -o wide --show-labels`
4. `minikube kubectl -- drain <node_name> [--ignore-daemonsets]`
   `(node_name =\= minikube)`
   `minikube kubectl -- get pods -o wide --show-labels`
5. `minikube kubectl -- uncord <node_name> [--ignore-daemonsets]`
   `(node_name == drained node)`
   `minikube kubectl -- get pods -o wide --show-labels`
6. Change deployment file + `minikube kubectl -- apply -f deployment.yaml`
7. `minikube kubectl -- get pods -o wide --show-labels`