Partie 2 : Déploiement d'applications

Dans un premier temps créer le dossier memos-data pour les volumes de l'application memos :

root@ecole-it-master:/home/farouck#mkdir memos-data

Créer les fichiers de configurations, memos-pv-pvc.yaml, memos-deployment.yaml, memos-services.yaml, php-deployment-service.yaml, et memos-ingress.yaml

Ecrire dans chaque fichier yaml avec la commande nano :

Ex : root@ecole-it-master:/home/farouck# nano memos-deployment.yaml

- **Configuration du stockage persistant et de la base de données**

Déployer mariadb via helm

copier les valeurs de la registry dans le fichier values.yaml

helm show values oci://registry-1.docker.io/bitnamicharts/mariadb > values.yaml

```
root@ecole-it-master:/home/farouck# helm show values oci://registry-1.docker.io/bitnamicharts/mariadb > values.yaml
Pulled: registry-1.docker.io/bitnamicharts/mariadb:18.0.1
Digest: sha256:2a0b02bc5c4b08611449cc1f5bd293bbe3a8d6ef04109a345f0d7699c4f8d7fd
```

Modifier le fichier values.yaml

root@ecole-it-master:/home/farouck#nano values.yaml

```
##
auth:
  ## @param auth.rootPassword Password for the `root` user. Ignored if existing secret is provided.
  ## ref: https://github.com/bitnami/containers/tree/main/bitnami/mariadb#setting-the-root-password-on-first-run
  ##
  rootPassword: "memos"
  ## @param auth.database Name for a custom database to create
  ## ref: https://github.com/bitnami/containers/blob/main/bitnami/mariadb/README.md#creating-a-database-on-first-run
  ##
  database: memos-db
  ## @param auth.username Name for a custom user to create
  ## ref: https://github.com/bitnami/containers/blob/main/bitnami/mariadb/README.md#creating-a-database-user-on-first-run
  ##
  username: "memos"
  ## @param auth.password Password for the new user. Ignored if existing secret is provided
  ##
  password: "memos"
  ## @param auth.replicationUser MariaDB replication user
```

On peut voir à quoi va ressembler le template avec la commande :

helm template mariadb -f values.yaml oci://registry-1.docker.io/bitnamicharts/mariadb

```
service:
  ## @param primary.service.type MariaDB Primary Kubernetes service type
  ##
  type: NodePort
  ports:
    ## @param primary.service.ports.mysql MariaDB Primary Kubernetes service port for MariaDB
    ##
    mysql: 3306
    ## @param primary.service.ports.metrics MariaDB Primary Kubernetes service port for metrics
    ##
    metrics: 9104
  ## @param primary.service.nodePorts.mysql MariaDB Primary Kubernetes service node port
  ## ref: https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport
  ##
  nodePorts:
    mysql: "31640"
  ## @param primary.service.clusterIP MariaDB Primary Kubernetes service clusterIP IP
  ##
```

Déploiement via helm :

helm install mariadb -f values.yaml oci://registry-1.docker.io/bitnamicharts/mariadb

```
root@ecole-it-master:/home/farouck# helm install mariadb -f  values.yaml oci://registry-1.docker.io/bitnamicharts/mariadb
Pulled: registry-1.docker.io/bitnamicharts/mariadb:18.0.1
Digest: sha256:2a0b02bc5c4b08611449cc1f5bd293bbe3a8d6ef04109a345f0d7699c4f8d7fd
NAME: mariadb
LAST DEPLOYED: Sun Apr 14 18:57:43 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: mariadb
CHART VERSION: 18.0.1
APP VERSION: 11.3.2

** Please be patient while the chart is being deployed **

Tip:

  Watch the deployment status using the command: kubectl get pods -w --namespace default -l app.kubernetes.io/instance=mariadb

Services:

  echo Primary: mariadb.default.svc.cluster.local:3306

Administrator credentials:

  Username: root
  Password : $(kubectl get secret --namespace default mariadb -o jsonpath="{.data.mariadb-root-password}" | base64 -d)

To connect to your database:

  1. Run a pod that you can use as a client:

      kubectl run mariadb-client --rm --tty -i --restart='Never' --image  docker.io/bitnami/mariadb:11.3.2-debian-12-r1 --names

  2. To connect to primary service (read/write):
```

Stockage persistent pour l'application memos

root@ecole-it-master:/home/farouck#nano memos-pv-pvc.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: memos-pv
spec:
  capacity:
    storage: 2Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/home/farouck/memos-data"
  storageClassName: hostpath

---

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: memos-pvc
spec:
  resources:
    requests:
      storage: 1Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  storageClassName: hostpath
  volumeName: memos-pv
```

Enregistrer et taper la commande :

root@ecole-it-master:/home/farouck#kubectl apply -f memos-pv-pvc.yaml

- **déployer l'application de site Web prédéfinie sur le cluster Kubernetes**

root@ecole-it-master:home/farouck#nano memos-deployment.yaml

```
kind: Deployment
metadata:
  name: memos-deployment
  labels:
    app: memos
spec:
  replicas: 4
  selector:
    matchLabels:
      app: memos
  template:
    metadata:
      labels:
        app: memos
    spec:
      containers:
      - name: memos
        image: neosmemo/memos:stable
        env:
          - name: MEMOS_DRIVER
            value: mysql
          - name: MEMOS_DSN
            value: memos:memos@tcp(mariadb:3306)/memos-db
        ports:
        - containerPort: 5230
        volumeMounts:
        - name: memos-data
          mountPath: "/var/opt/memos"
      volumes:
      - name: memos-data
        persistentVolumeClaim:
          claimName: memos-pvc
```

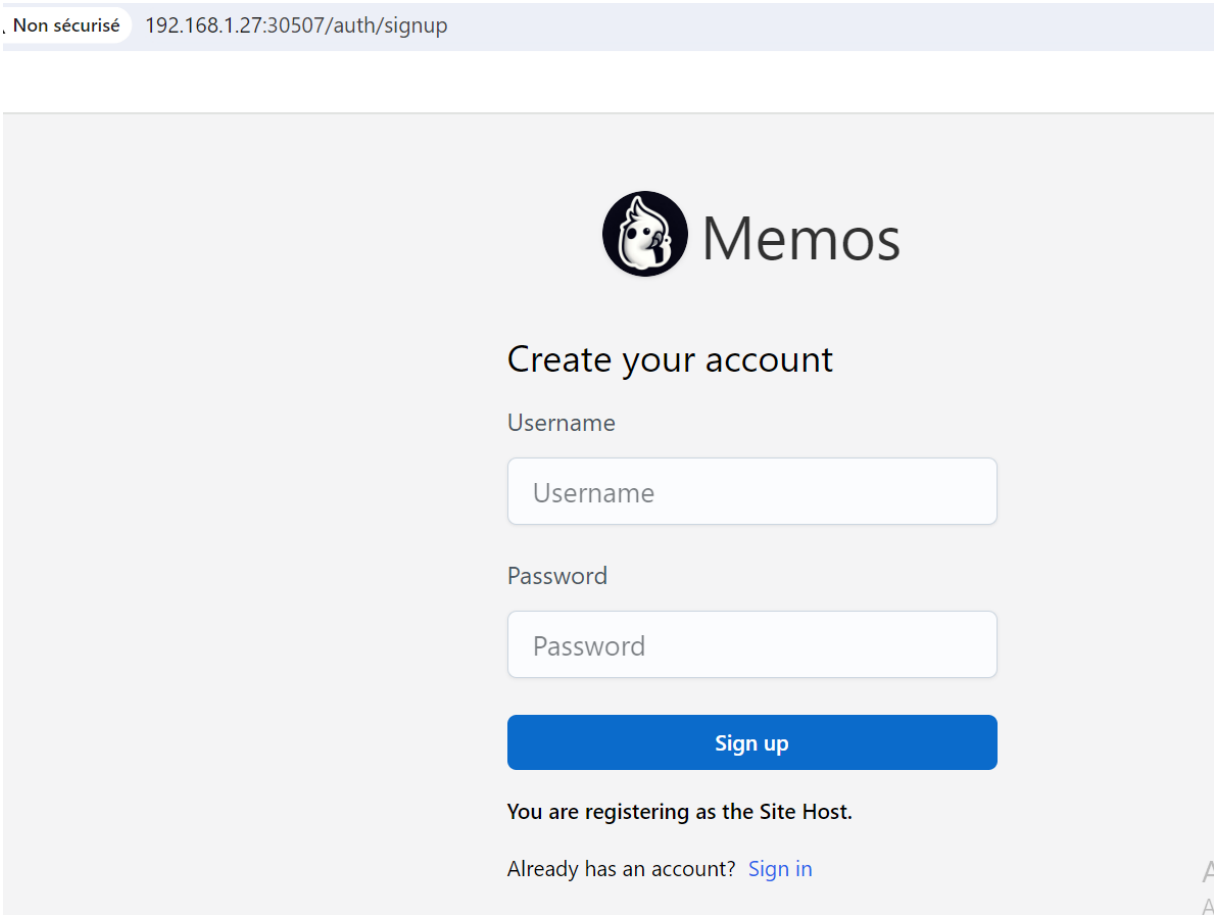
root@ecole-it-master:/home/farouck#kubectl apply –f memos-deployment.yaml

root@ecole-it-master:home/farouck#nano memos-services.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: memos-services
spec:
  type: NodePort
  selector:
    app: memos
  ports:
  - port: 5230
    targetPort: 5230
```

root@ecole-it-master:/home/farouck#kubectl apply –f memos-services.yaml

Accès de l’application via l’adresse IP du worker1 :



Déploiement de mariadb sur phpmyadmin

root@ecole-it-master:/home/farouck#nano php-deployment-service.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: phpadmin
spec:
  strategy:
    type: Recreate
  selector:
    matchLabels:
      component: phpadmin
  template:
    metadata:
      labels:
        component: phpadmin
    spec:
      containers:
      - name: phpadmin
        image: phpmyadmin
        env:
          - name: MYSQL_ROOT_PASSWORD
            value: "memos"
          - name: MYSQL_DATABASE
            value: "memos-db"
          - name: MYSQL_USER
            value: "memos"
          - name: MYSQL_PASSWORD
            value: "memos"
          - name: PMA_HOST
            value: "mariadb"
        ports:
        - containerPort: 80
---
```

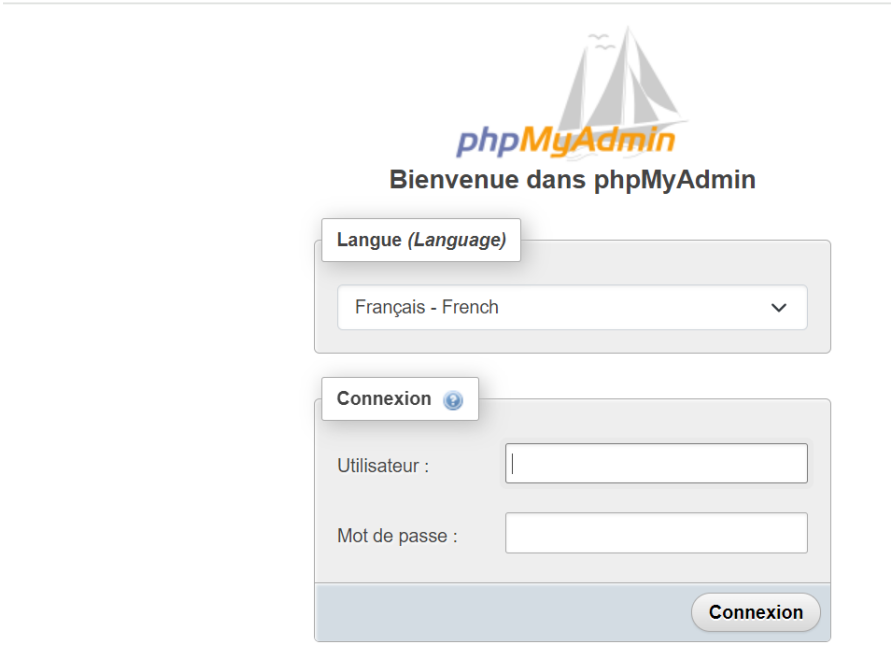
```
apiVersion: v1
kind: Service
metadata:
  name: phpadmin
spec:
  type: NodePort
  selector:
    component: phpadmin
  ports:
  - port: 8000
    targetPort: 80
root@ecole-it-master:/home/farouck# kubectl apply -f php-deployment-services.yaml
error: the path "php-deployment-services.yaml" does not exist
root@ecole-it-master:/home/farouck# kubectl apply -f php-deployment-service.yaml
deployment.apps/phpadmin created
service/phpadmin created
```

root@ecole-it-master:/home/farouck#kubectl apply -f php-deployment-service.yaml

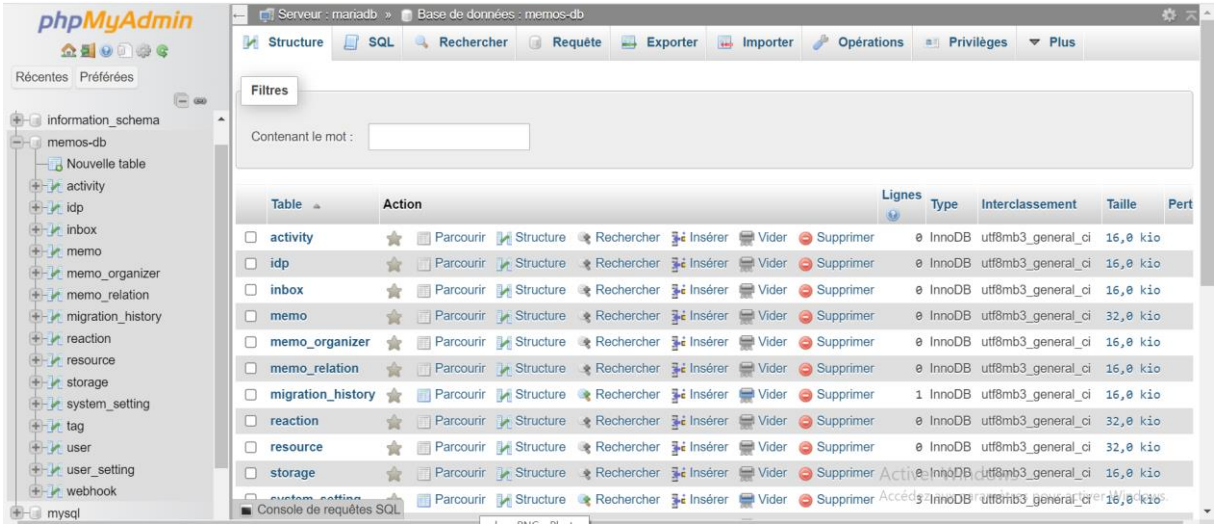
```
phpadmin                                  NodePort       10.43.129.71    <none>        8000:31275/TCP
s
```

Accéder à phpmyadmin via le master : http://192.168.1.25:31275

192.168.1.25:31275



Se connecter avec Utilisateur : root, Mot de passe : memos



- **<u>Implémentation du controller Traefik :</u>**

  lien de la documentation : https://doc.traefik.io/traefik/getting-started/quick-start-with-kubernetes/

Création des fichiers : 00-role.yaml, 00-account.yaml, 01-role-binding.yaml, 02-traefik.yaml, 02-traefik-services.yaml

Ecriture dans chaque fichier

root@ecole-it-master:/home/farouck# nano 00-role.yaml

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: traefik-role

rules:
  - apiGroups:
      - ""
    resources:
      - services
      - endpoints
      - secrets
    verbs:
      - get
      - list
      - watch
  - apiGroups:
      - extensions
      - networking.k8s.io
    resources:
      - ingresses
      - ingressclasses
    verbs:
      - get
      - list
      - watch
  - apiGroups:
      - extensions
      - networking.k8s.io
    resources:
      - ingresses/status
    verbs:
      - update
```

root@ecole-it-master:/home/farouck# nano 00-account.yaml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: traefik-account
```

root@ecole-it-master:/home/farouck# nano 01-role-binding.yaml

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: traefik-role-binding

roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: traefik-role
subjects:
  - kind: ServiceAccount
    name: traefik-account
    namespace: default
```

root@ecole-it-master:/home/farouck# nano 02-traefik.yaml

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: traefik-deployment
  labels:
    app: traefik

spec:
  replicas: 1
  selector:
    matchLabels:
      app: traefik
  template:
    metadata:
      labels:
        app: traefik
    spec:
      serviceAccountName: traefik-account
      containers:
        - name: traefik
          image: traefik:v2.11
          args:
            - --api.insecure
            - --providers.kubernetesingress
          ports:
            - name: web
              containerPort: 80
            - name: dashboard
              containerPort: 8080
```

root@ecole-it-master:/home/farouck# nano 02-traefik-services.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: traefik-dashboard-service

spec:
  type: NodePort
  ports:
    - port: 8080
      targetPort: dashboard
  selector:
    app: traefik
---
apiVersion: v1
kind: Service
metadata:
  name: traefik-web-service

spec:
  type: NodePort
  ports:
    - targetPort: web
      port: 80
  selector:
    app: traefik
```

faire un apply sur chaque fichier

```
root@ecole-it-master:/home/farouck# kubectl apply -f 00-role.yaml
clusterrole.rbac.authorization.k8s.io/traefik-role created
root@ecole-it-master:/home/farouck# kubectl apply -f 00-account.yaml
serviceaccount/traefik-account created
root@ecole-it-master:/home/farouck# kubectl apply -f 01-role-binding.yaml
clusterrolebinding.rbac.authorization.k8s.io/traefik-role-binding created
root@ecole-it-master:/home/farouck# kubectl apply -f 02-traefik.yaml
deployment.apps/traefik-deployment created
root@ecole-it-master:/home/farouck# kubectl apply -f 02-traefik-services.yaml
service/traefik-dashboard-service created
service/traefik-web-service created
```

Ecrire dans le fichier memos-ingress.yaml pour créer un ingress et faire un apply:

root@ecole-it-master:/home/farouck# nano memos-ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: memos-ingress
spec:
  rules:
  - host: memos.localhost.com
    http:
      paths:
      - pathType: Prefix
        path: "/"
        backend:
          service:
            name: memos-services
            port:
              number: 5230
```

root@ecole-it-master:/home/farouck# kubectl -f memos-ingress.yaml

liste de tous les services et ingress

```
root@ecole-it-master:/home/farouck# kubectl get services
NAME                     TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes               ClusterIP   10.43.0.1       <none>        443/TCP          30h
mariadb                  NodePort    10.43.61.116    <none>        3306:31640/TCP   24h
memos-services           NodePort    10.43.90.135    <none>        5230:30507/TCP   24h
nginx-service            NodePort    10.43.169.50    <none>        80:31148/TCP     29h
phpadmin                 NodePort    10.43.129.71    <none>        8000:31275/TCP   23h
traefik-dashboard-service   NodePort    10.43.63.184    <none>        8080:32719/TCP   64m
traefik-web-service      NodePort    10.43.218.77    <none>        80:31907/TCP     64m
root@ecole-it-master:/home/farouck# kubectl get ingress
NAME           CLASS     HOSTS                ADDRESS                       PORTS   AGE
memos-ingress   traefik   memos.localhost.com   192.168.1.25,192.168.1.27   80      58m
```

Modifier le fichier /etc/hosts dans le master dans la machine virtuelle debian et enregistrer

root@ecole-it-master:/home/farouck# nano /etc/hosts

```
127.0.0.1       localhost
127.0.1.1       ecole-it-master
192.168.1.25    memos.localhost.com
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Modifier Le fichier C:\Windows\System32\drivers\etc\hosts sur windows en tant qu'administrateur

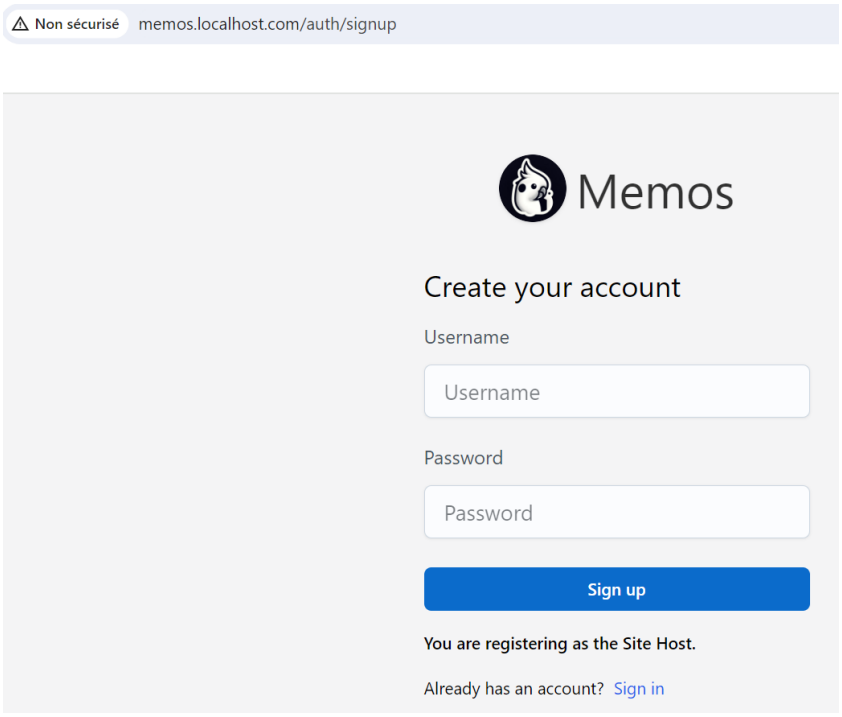
```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       127.0.0.1       farouck
#       ::1             localhost
192.168.1.25 memos.localhost.com
```

Preuve que l'ingress a marché :

Sur le master de la machine virtuelle :

```
root@ecole-it-master:/home/farouck# curl memos.localhost.com
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png" />
    <link rel="icon" type="image/webp" href="/logo.webp" />
    <link rel="manifest" href="/site.webmanifest" />
    <meta name="theme-color" media="(prefers-color-scheme: light)" content="#f4f4f5" />
    <meta name="theme-color" media="(prefers-color-scheme: dark)" content="#18181b" />
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
    <!-- memos.metadata.head -->
    <title>Memos</title>
    <script>
      // Prevent flash of light mode.
      const appearance = localStorage.getItem("appearance");
      if (appearance === `"dark"`) {
        document.documentElement.classList.add("dark");
      }
    </script>
    <script type="module" crossorigin src="/assets/index-BlexmZ_L.js"></script>
    <link rel="stylesheet" crossorigin href="/assets/index-CcXvRfz2.css">
  </head>
  <body>
    <div id="root"></div>
    <!-- memos.metadata.body -->
  </body>
</html>
```

Preuve sur mon pc windows :



Preuve de réussite de toutes les configurations :

```
root@ecole-it-master:/home/farouck# kubectl get pods
NAME                                READY   STATUS    RESTARTS       AGE
mariadb-0                           1/1     Running   2 (115m ago)   25h
memos-deployment-756f9658cc-4n4dc   1/1     Running   6 (114m ago)   25h
memos-deployment-756f9658cc-s9lcj   1/1     Running   8 (114m ago)   25h
memos-deployment-756f9658cc-tpd6f   1/1     Running   1 (114m ago)   25h
memos-deployment-756f9658cc-wrcbt   1/1     Running   1 (114m ago)   25h
nginx-server-84764cd945-rnxmb       1/1     Running   1 (114m ago)   29h
phpadmin-575d8f45b-5nmzb            1/1     Running   1 (114m ago)   24h
traefik-deployment-8b68dc9fd-vpf47  1/1     Running   0              79m
root@ecole-it-master:/home/farouck# kubectl get services
NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes                ClusterIP   10.43.0.1      <none>        443/TCP          30h
mariadb                   NodePort    10.43.61.116   <none>        3306:31640/TCP   25h
memos-services            NodePort    10.43.90.135   <none>        5230:30507/TCP   25h
nginx-service             NodePort    10.43.169.50   <none>        80:31148/TCP     29h
phpadmin                  NodePort    10.43.129.71   <none>        8000:31275/TCP   24h
traefik-dashboard-service NodePort    10.43.63.184   <none>        8080:32719/TCP   83m
traefik-web-service       NodePort    10.43.218.77   <none>        80:31907/TCP     83m
root@ecole-it-master:/home/farouck# kubectl get ingress
NAME            CLASS     HOSTS                ADDRESS                       PORTS   AGE
memos-ingress   traefik   memos.localhost.com   192.168.1.25,192.168.1.27   80      76m
root@ecole-it-master:/home/farouck# kubectl get pvc
NAME            STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
data-mariadb-0  Bound    pvc-f42a33d8-e556-4ab1-a581-11d89f13e13e   8Gi        RWO            local-path     <unset>                 25h
memos-pvc       Bound    memos-pv                                   2Gi        RWO            hostpath       <unset>                 25h
```