Fixing minikube external IP issue

Tuesday, August 5, 2025 10:19 AM

To access a web application pod from external IP address, we need to implement Ingress to expose the pod to external IP. Here are the steps to implement Ingress:

1. `minikube start`

2. `minikube addons enable ingress`

3. `minikube addons list`

   Verify that the NGINX Ingress controller is running

4. `kubectl get pods -n ingress-nginx`

   Deploy your pods, services and deployments

5. `kubectl apply -f mongo-config.yaml`
6. `kubectl apply -f mongo-secret.yaml`
7. `kubectl apply -f mongo.yaml`
8. `kubectl apply -f webapp.yaml`
9. `kubectl apply -f webapp-ingress.yaml`

   Verify your pods, services and deployments are created and running

10. `kubectl get all`

    Get service IP address

11. `minikube service web --url`

    You will get similar output ;ole this:
    http://127.0.0.1:51859
    You can check the output: `curl http://127.0.0.1:51859`

    Add an entry in `hosts` file
    The `hosts` file is located at `C:\Windows\System32\drivers\etc`

12. `127.0.0.1  rama.example`

    Run this command in another command window

13. `minikube tunnel`

    Verify that URL is accessible

14. `curl rama.example -i`

    You should see similar to this: `HTTP/1.1 200 OK` and html output

15. Browse in these URLs in the browser and you should see the UI output
    http://rama.example
    http://rama.example/venky

mongo-config.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: mongo-config
data:
  mongo-url: mongo-service
```

mongo-secret.yaml

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mongo-secret
type: Opaque
data:
  mongo-user: bW9uZ291c2Vy
```

```
  mongo-password: bW9uZ29wYXNzd29yZA==
```

Install openssl on windows
Use this command to encrypt secrets:

- echo -n mongouser |openssl base64

mongo.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo-deployment
  labels:
    app: mongo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongo
  template:
    metadata:
      labels:
        app: mongo
    spec:
      containers:
      - name: mongodb
        image: mongo:5.0
        ports:
        - containerPort: 27017
        env:
        - name: MONGO_INITDB_ROOT_USERNAME
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-user
        - name: MONGO_INITDB_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-password
---
apiVersion: v1
kind: Service
metadata:
  name: mongo-service
spec:
  selector:
    app: mongo
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017
```

webapp.yaml

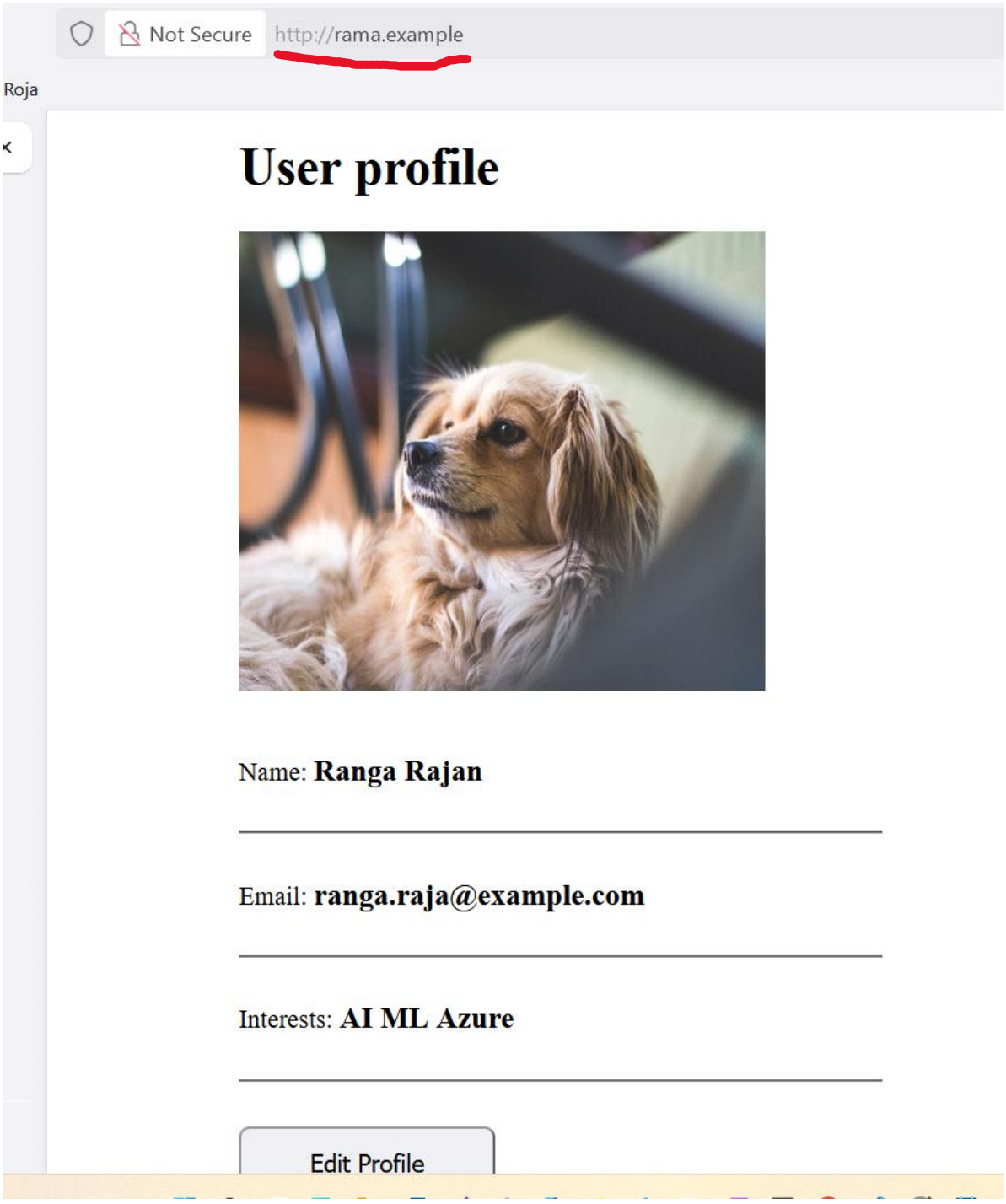
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-deployment
  labels:
    app: webapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp
  template:
    metadata:
      labels:
        app: webapp
    spec:
      containers:
      - name: webapp
        image: nanajanashia/k8s-demo-app:v1.0
        ports:
        - containerPort: 3000
        env:
        - name: USER_NAME
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-user
        - name: USER_PWD
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-password
        - name: DB_URL
          valueFrom:
            configMapKeyRef:
              name: mongo-config
              key: mongo-url
---
apiVersion: v1
kind: Service
metadata:
  name: webapp-service
spec:
  type: NodePort
  selector:
    app: webapp
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
```

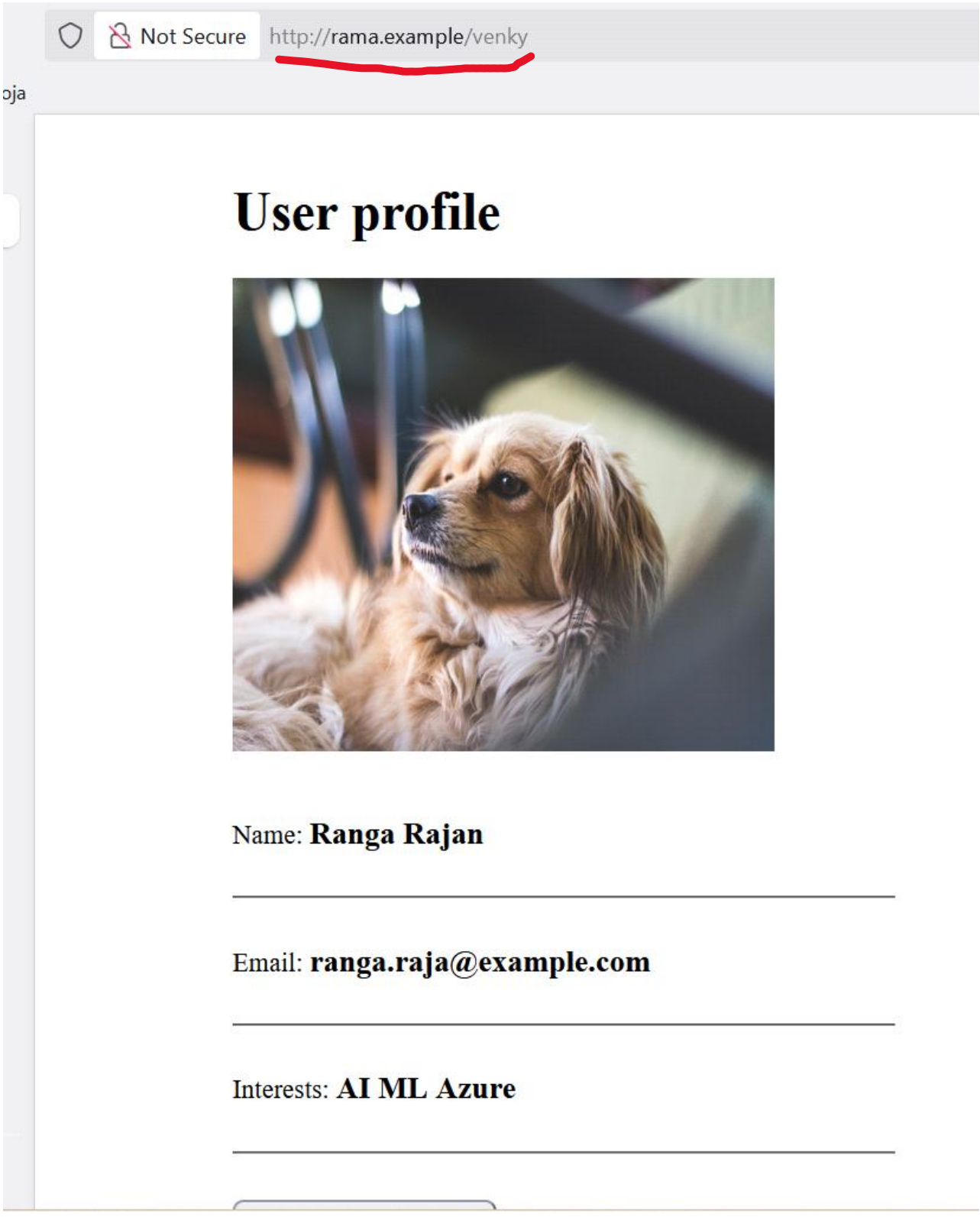
```
  - protocol: TCP
    port: 3000
    targetPort: 3000
    nodePort: 30100
```

Webapp-ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: webapp-ingress
spec:
  rules:
  - host: rama.example
    http:
      paths:
      - path: /venky
        pathType: Prefix
        backend:
          service:
            name: webapp-service
            port:
              number: 3000
      - path: /
        pathType: Prefix
        backend:
          service:
            name: webapp-service
            port:
              number: 3000
```

Host name I gave to my laptop (minikube cluster) using this yaml is `rama.example`. The intent is to access the web app (pod) using the below URLs:
http://rama.example
http://rama.example/venky


Not Secure
http://rama.example
Roja
User profile
Name: Ranga Rajan
Email: ranga.raja@example.com
Interests: AI ML Azure
Edit Profile



Deploy hello world pods from Google

```
16. kubectl create deployment web --image=gcr.io/google-samples/hello-app:1.0
17. kubectl create deployment web2 --image=gcr.io/google-samples/hello-app:2.0
```

Update the ingress service, add the below lines to

```
18. webapp-ingress.yaml
```

Append these lines to existing webapp-ingress.yaml

```
    - path: /v1
      pathType: Prefix
      backend:
        service:
          name: web
          port:
            number: 8080
    - path: /v2
      pathType: Prefix
      backend:
        service:
          name: web2
          port:
            number: 8080
```

19. `kubectl apply -f webapp-ingress.yaml`

Verify your pods, services and deployments are created and running

20. `kubectl get all`

You should see `web` and `web2` pods also in the output

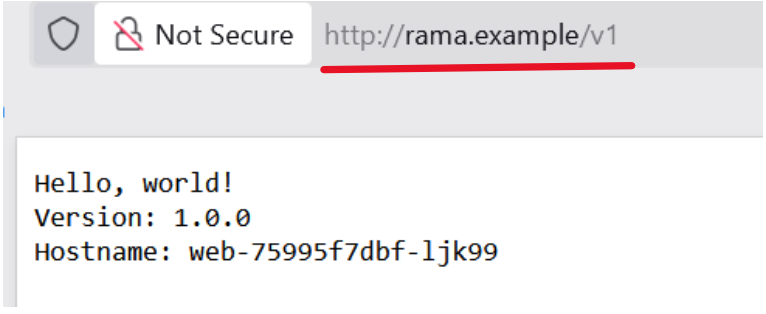
Verify that URL is accessible

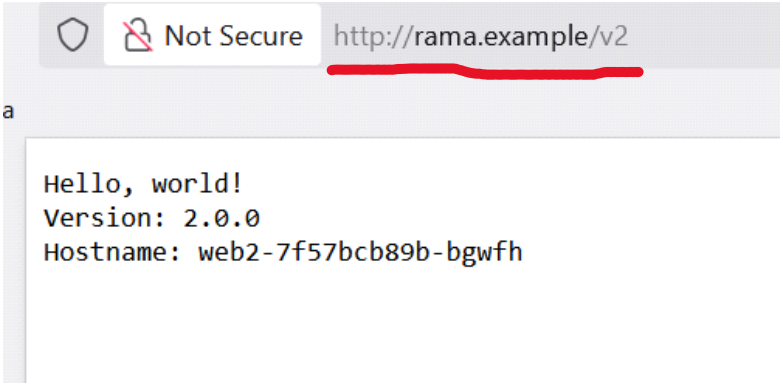
21. `curl rama.example -i`

You should see similar to this: `HTTP/1.1 200 OK` and html output

22. Browse in these URLs in the browser and you should see the UI output

http://rama.example/v1
http://rama.example/v2





References:

- Kubernetes Crash Course for Absolute Beginners [NEW]

- https://gitlab.com/nanuchi/k8s-in-1-hour