Ch-134.DevOps/Cisco

Demo 4

IT Academy

softserve

Team 2

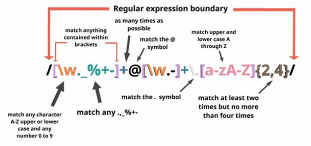
Ivan Kuvila
Pavlo Tarnovetskyi
Ivan Danyliuk
Vladyslav Boreiko

Ch-134.DevOps/Cisco

softserve

Ivan Danyliuk

https://github.com/idanylyuk/DevOps

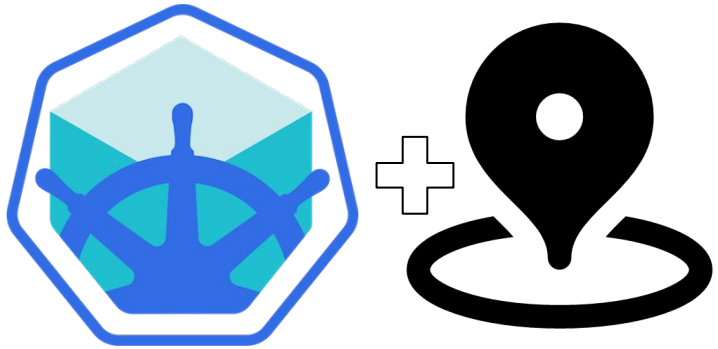


https://github.com/idanylyuk/DevOps

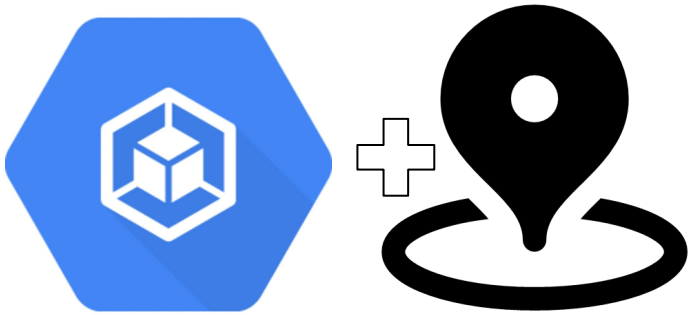
Report plan

Kubernetes

Minikube GeoCitizen Deployment



GKE Geocitizen Deployment



softserve

Ivan Danyliuk

Kubernetes

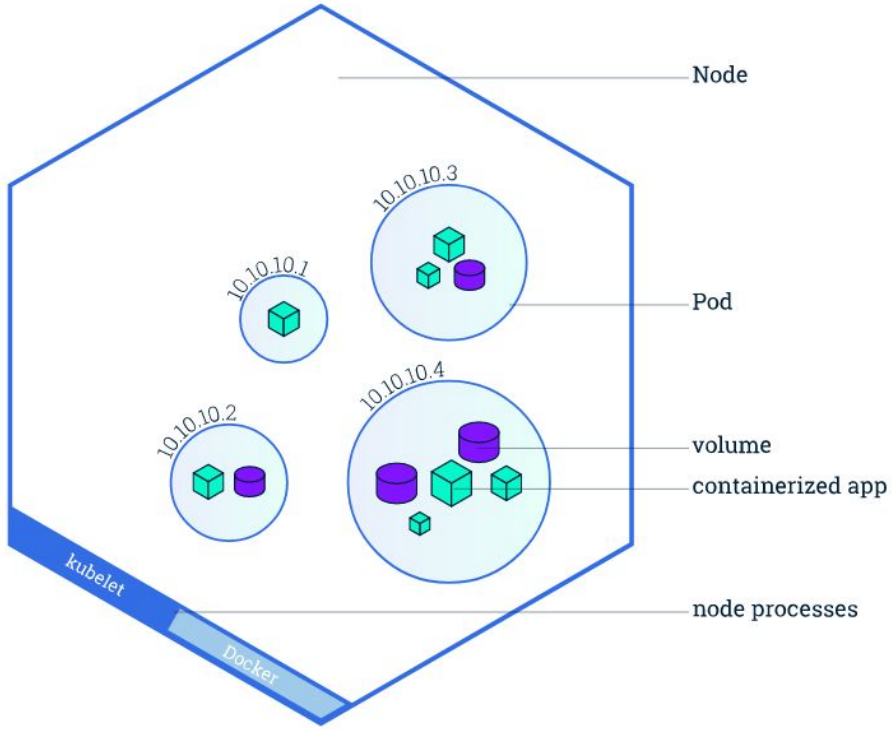
https://kubernetes.io

Kubernetes originates from Greek, meaning helmsman or pilot

K8(eight letters between K and s in “Kubernetes” word)s

Google open-sourced the Kubernetes project in 2014

A Kubernetes (K8s) cluster
is a grouping of nodes
that run containerized apps in an
- efficient,
- automated,
- distributed,
- scalable
manner.



softserve

Ivan Danyliuk

Kubernetes Cluster Architecture

Nodes

Control Plane - Node Communication

Controllers

Cloud Controller Manager

Container Runtime Interface (CRI)

Garbage Collection

softserve
Ivan Danyliuk

Kubernetes cluster creating tools

Learning Environment

- kind
- minikube
- kubeadm

Production environment

- kubeadm
- kops
- Kubespray

Cloud Solutions

https://landscape.cncf.io/card-mode?category=certified-kubernetes-hosted&grouping=category

softserve

Ivan Danyliuk

minikube

Local Kubernetes

focusing on making it easy to learn and develop Kubernetes

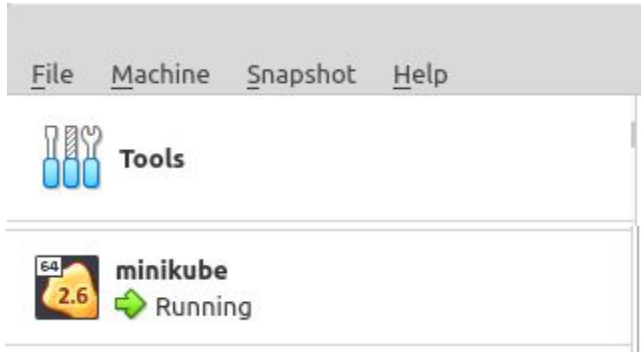
Requirements:

- **2 CPUs or more**
- **2GB of free memory**
- **20GB of free disk space**
- **Internet connection**
- **Container or virtual machine manager, such as: Docker, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox, or VMware Fusion/Workstation**

Simple install / Simple Start

softserve
Ivan Danyliuk

minikube



All minikube files are stored in directory ~/.minikube

minikube start

```
ivan@Dell-NB:~$ minikube start
😄  minikube v1.25.2 on Linuxmint 20
✨  Using the virtualbox driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🔄  Restarting existing virtualbox VM for "minikube" ...
🐳  Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
    ▪ kubelet.housekeeping-interval=5m
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Start with parameters

```
ubuntu@docker2:~$ minikube start --cpus=2 --memory=2.5gb --disk-size=8gb
😄  minikube v1.25.2 on Ubuntu 18.04 (vbox/amd64)
✨  Automatically selected the docker driver
❗  Your cgroup does not allow setting memory.
    ▪ More information: https://docs.docker.com/engine/install/linux-postinstall/
rt-cgroup-swap-limit-capabilities
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔥  Creating docker container (CPUs=2, Memory=2560MB) ...
🐳  Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
    ▪ kubelet.housekeeping-interval=5m
```

minikube stop

```
ubuntu@docker2:~$ minikube stop
✋  Stopping node "minikube"  ...
🛑  Powering off "minikube" via SSH ...
🛑  1 node stopped.
ubuntu@docker2:~$
```

minikube delete

```
ubuntu@docker2:~$ minikube delete
🔥  Deleting "minikube" in docker ...
🔥  Deleting container "minikube" ...
🔥  Removing /home/ubuntu/.minikube/machines/minikube ...
💀  Removed all traces of the "minikube" cluster.
```

softserve
Ivan Danyliuk

kubectl

interaction with cluster

```
ivan@Dell-NB:~$ kubectl version --client --output=yaml
clientVersion:
  buildDate: "2022-04-14T08:49:13Z"
  compiler: gc
  gitCommit: ad3338546da947756e8a88aa6822e9c11e7eac22
  gitTreeState: clean
  gitVersion: v1.23.6
  goVersion: go1.17.9
  major: "1"
  minor: "23"
  platform: linux/amd64

ivan@Dell-NB:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.59.103:8443
CoreDNS is running at https://192.168.59.103:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

`kubectl get nodes`

```
ivan@Dell-NB:~$ kubectl get nodes
NAME       STATUS   ROLES                  AGE   VERSION
minikube   Ready    control-plane,master   30h   v1.23.3
```

Add new node to cluster

```
ivan@Dell-NB:~$ minikube node add
😄  Adding node m02 to cluster minikube
❗  Cluster was created without any CNI, adding a node to it might cause broken networking.
👍  Starting worker node minikube-m02 in cluster minikube
🔥  Creating virtualbox VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
🐳  Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
```

```
ivan@Dell-NB:~$ kubectl get nodes
NAME           STATUS   ROLES                  AGE   VERSION
minikube       Ready    control-plane,master   30h   v1.23.3
minikube-m02   Ready    <none>                 40s   v1.23.3
```

```
ivan@Dell-NB:~$ minikube node list
minikube        192.168.59.103
minikube-m02    192.168.59.104
```

```
ivan@Dell-NB:~$ minikube node delete minikube-m02
🔥  Deleting node minikube-m02 from cluster minikube
🔥  Deleting "minikube-m02" in virtualbox ...
💀  Node minikube-m02 was successfully deleted.
```

softserve

Ivan Danyliuk

Working with pods

Starts new pod with name app-geo, docker image tomcat:9 and port 8080:

```
$ kubectl run app-geo --image=tomcat:9 --port=8080
```

Get pods info

```
$ kubectl get pods
$ kubectl describe pods app-geo
```

Delete pod

```
$ kubectl delete pods app-geo
```

Login to created pod (app-geo)

```
$ kubectl exec -it geo-deployment-autoscaling-84d4998d94-6np4m -- bash
```

```
ivan@Dell-NB:~$ kubectl exec -it geo-deployment-autoscaling-84d4998d94-6np4m -- bash
root@geo-deployment-autoscaling-84d4998d94-6np4m:/usr/local/tomcat#
```

View log files of pod

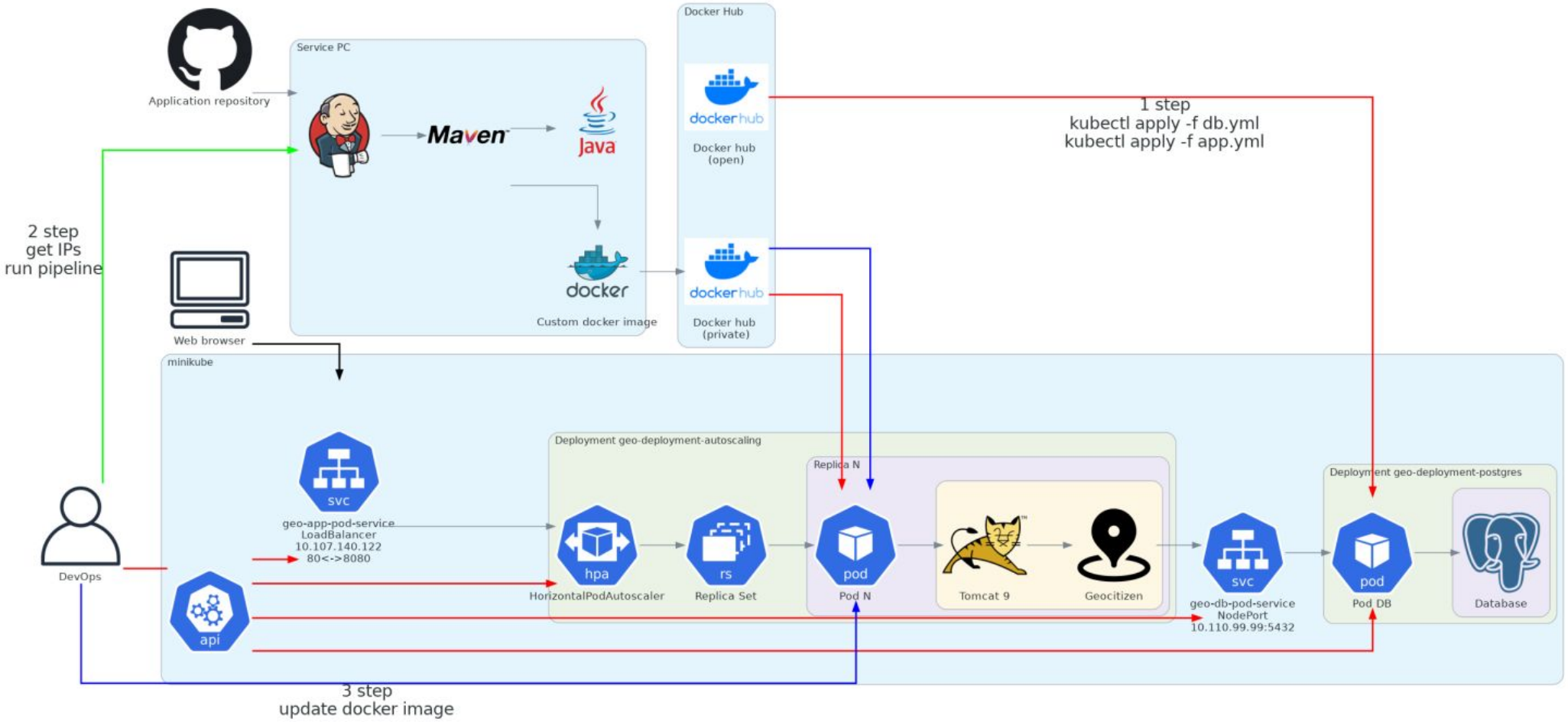
```
$ kubectl logs app-geo
```

Port forwarding

```
$ kubectl port-forward app-geo 8081:8080
```

softserve
Ivan Danyliuk

minikube GeoCitizen deployment



softserve

Ivan Danyliuk

minikube GeoCitizen deployment

1.Build application war-file with fake addresses and push it to docker hub.

```
$ docker login -u <user>
$ docker push xbuyer/data:geo_minikube
```

2.Generate in docker hub access token

3.Create secret with kubectl

```
$ kubectl create secret docker-registry geosecret
--docker-server='https://index.docker.io/v1/'
--docker-username='-----'
--docker-password='--------------'
--docker-email='--------'
```

4.Create Infrastructure

```
$ kubectl apply -f db.yml
$ kubectl apply -f app.yml
```

5.Get Ip-addresses

```
ivan@Dell-NB:~$ kubectl get services
NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)          AGE
geo-app-pod-service  LoadBalancer   10.107.140.122   10.107.140.122   80:31777/TCP     31h
geo-db-pod-service   NodePort       10.110.99.99     <none>           5432:30926/TCP   31h
kubernetes           ClusterIP      10.96.0.1        <none>           443/TCP          31h
```

softserve

Ivan Danyliuk

minikube GeoCitizen deployment

6.Run minikube tunnel and leave it working

```
$ minikube tunnel
```

```
ivan@Dell-NB:~/kubernetes$ minikube tunnel
Status:
        machine: minikube
        pid: 382183
        route: 10.96.0.0/12 -> 192.168.59.103
        minikube: Running
        services: [geo-app-pod-service]
    errors:
                minikube: no errors
                router: no errors
                loadbalancer emulator: no errors
```

7.Rebuild application war-file with real addresses and push it to docker hub with new tag.

```
$ docker login -u <user>
$ docker push xbuyer/data:geo_minikube_v2
```

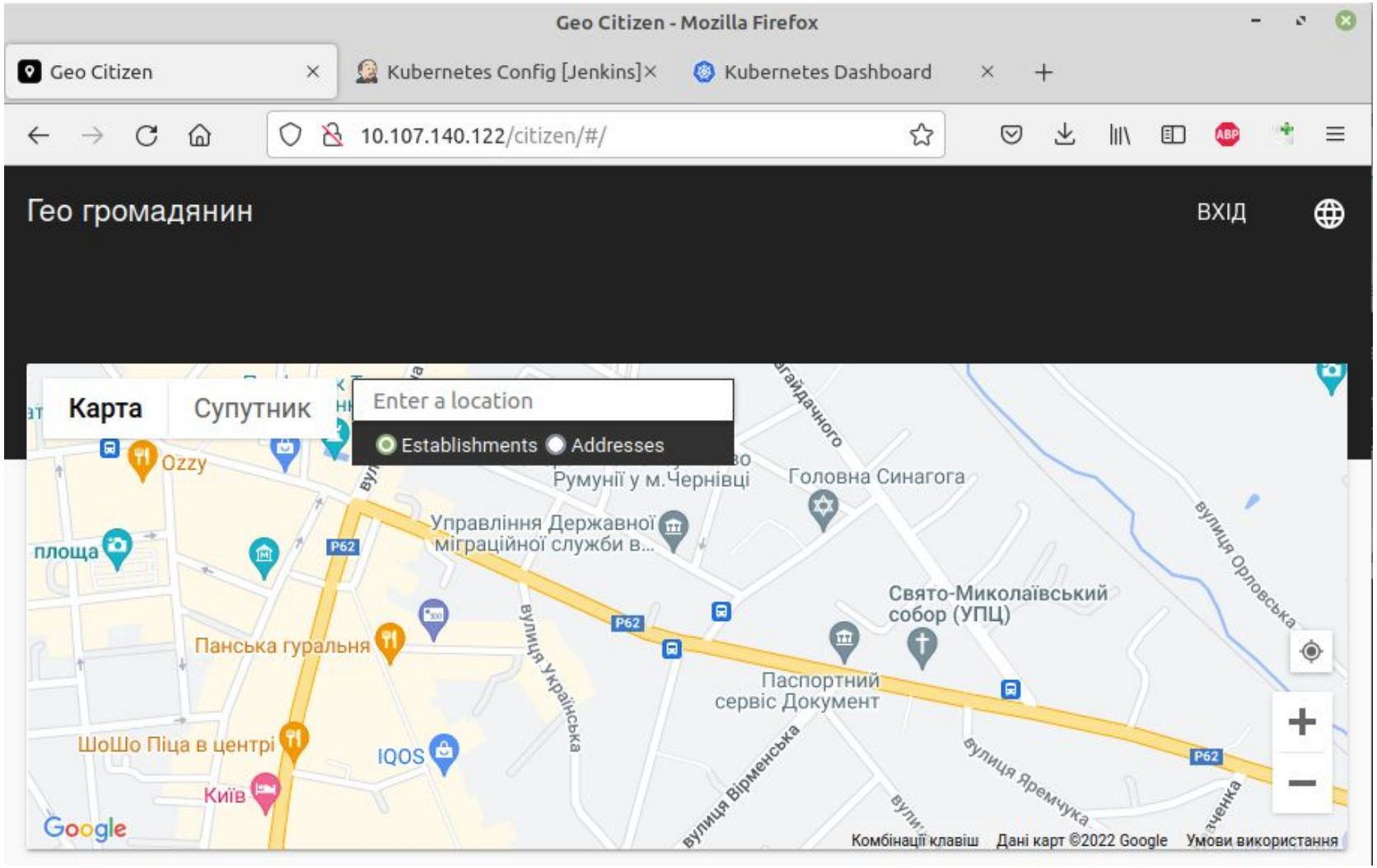
8.Update image for App Load Balancer Pods

```
$ kubectl set image deployment/geo-deployment-autoscaling \
  app-web=docker.io/xbuyer/data:geo_minikube_v2
```

softserve
Ivan Danyliuk

minikube GeoCitizen deployment

9.Use application

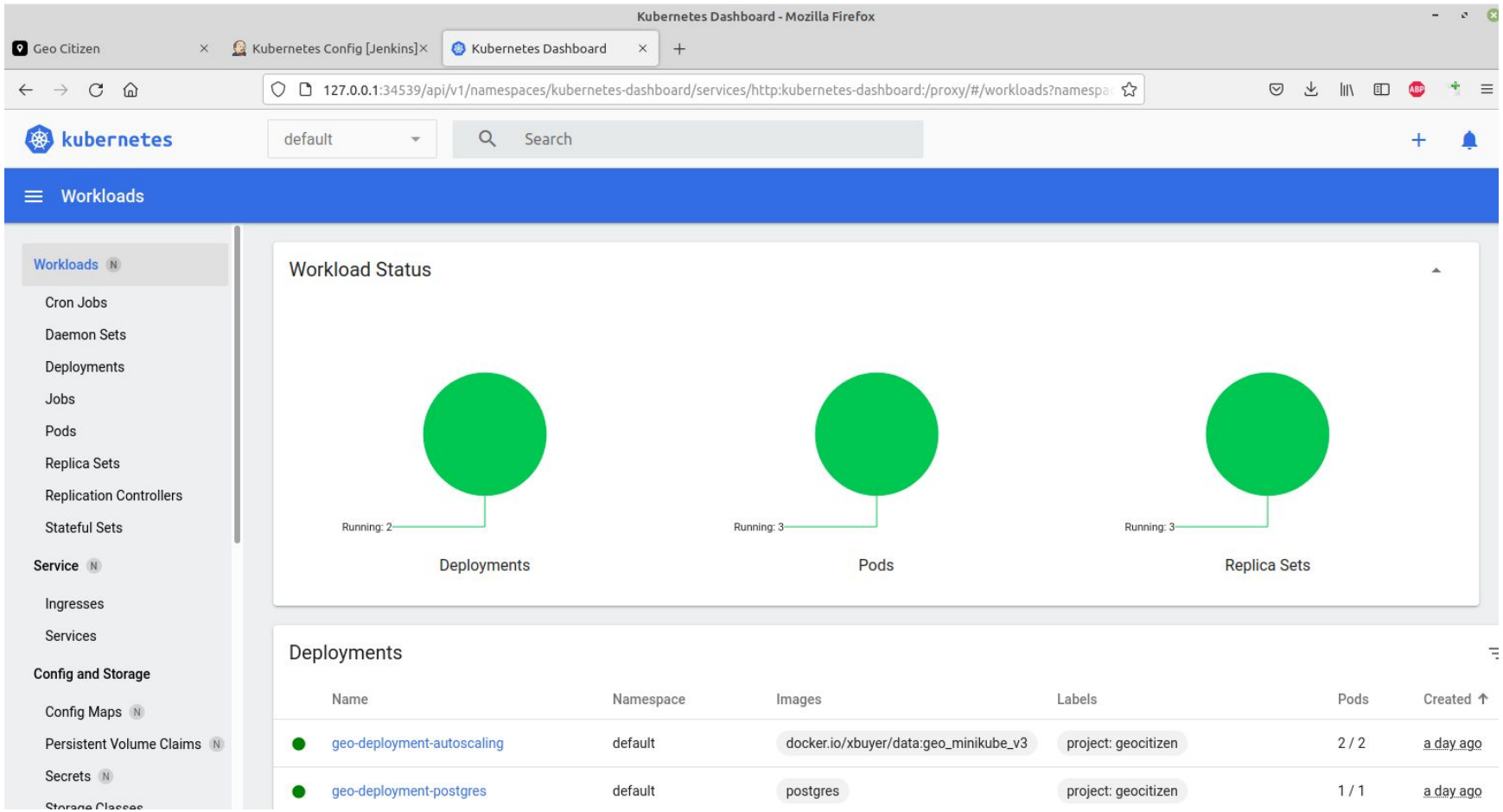


softserve
Ivan Danyliuk

minikube GeoCitizen deployment

10.Dashboard with created infrastructure

```
$ minikube dashboard
```



softserve

Ivan Danyliuk

db.yml

```yaml
apiVersion : apps/v1
kind: Deployment
metadata:
  name: geo-deployment-postgres
  labels:
    project : geocitizen
spec:
  selector:
    matchLabels:
      project: geocitizen-db
  template:
    metadata:
      labels:
        project: geocitizen-db  # Service will look
    spec:
      containers:
        - name : app-db
          env:
          - name: POSTGRES_DB
            value: Geo
          - name: POSTGRES_USER
            value: Geo
          - name: POSTGRES_PASSWORD
            value: GeoCitizenDocker
          image: postgres
          ports:
            - containerPort: 5432
```

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: geo-db-pod-service
  labels:
    env  : test
    owner: uixcoder
spec:
  selector:
    project:  geocitizen-db     # Selecting PODs
  ports:
    - name       : db-listener
      protocol   : TCP
      port       : 5432  # Port on Load Balancer
      targetPort: 5432  # Port on Pod
  type: NodePort
```

softserve

Ivan Danyliuk

app.yml

```yaml
apiVersion : apps/v1
kind: Deployment
metadata:
  name: geo-deployment-autoscaling
  labels:
    project : geocitizen
spec:
  selector:
    matchLabels:
      project: geocitizen-app
  template:
    metadata:
      labels:
        project: geocitizen-app  # Service will look
    spec:
      containers:
      - name : app-web
        image: docker.io/xbuyer/data:geo_minikube_v2
        imagePullPolicy: Always
        ports:
        - containerPort: 8080
      imagePullSecrets:
      - name: geosecret
```

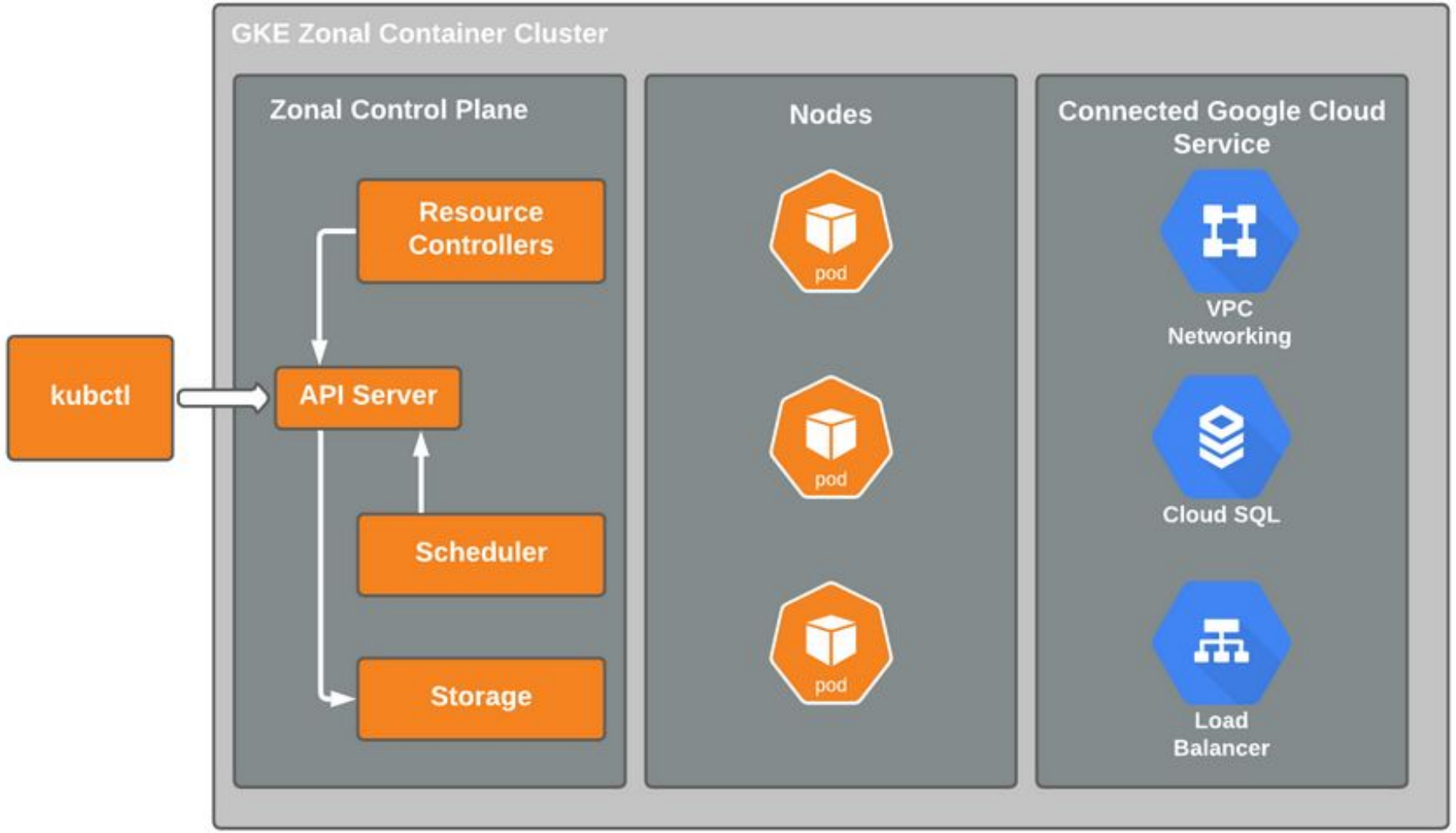
```yaml
---
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: geo-autoscaler
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: geo-deployment-autoscaling
  minReplicas: 2
  maxReplicas: 6
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 50
    - type: Resource
      resource:
        name: memory
        target:
          type: Utilization
          averageUtilization: 50
```

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: geo-app-pod-service
  labels:
    env  : test
    owner: uixcoder
spec:
  selector:
    project:  geocitizen-app      # Selecting PODs
  ports:
    - name      : app-listener
      protocol  : TCP
      port      : 80  # Port on Load Balancer
      targetPort: 8080  # Port on Pod
  type: LoadBalancer
```

softserve

Ivan Danyliuk

Kubernetes on GCP (GKE)



Google Kubernetes Engine (GKE)

softserve
Ivan Danyliuk

GKE start

Kubernetes clusters CREATE DEPLOY REFRESH OPERATIONS

OVERVIEW COST OPTIMIZATION

Filter Enter property name or value

Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notificat
✓	geocluster	europe-west1	3	6	6 GB	

VM instances

✓	gke-geocluster-default-pool-b44af8a5-8jhb	europe-west1-b
✓	gke-geocluster-default-pool-b44af8a5-c02b	europe-west1-b
✓	gke-geocluster-default-pool-b44af8a5-l406	europe-west1-b

Clusters EDIT DELETE OPERATIONS HELP ASSISTANT

geocluster

DETAILS NODES STORAGE LOGS

Cluster basics

Name	geocluster	🔒
Location type	Regional	🔒
Region	europe-west1	🔒
Default node zones	europe-west1-b	✎
Release channel	Stable channel	✎ UPGRADE AVAILABLE
Version	1.21.10-gke.2000	
Total size	3	ⓘ
Endpoint	34.76.47.135 Show cluster certificate	🔒

Node Pools

Filter Filter node pools

Name	Status	Version	Number of nodes	Machine type	Image type
default-pool	✓ Ok	1.21.10-gke.2000	3	e2-small	Ubuntu with Docker (ubuntu)

softserve

Ivan Danyliuk

GKE start

Install the gcloud CLI

Install kubectl and configure cluster access

```
$ gcloud init --console-only
```

Follow the instructions to authorize the gcloud CLI

!!! Do not set zone. Only region later by command

```
$ gcloud config set compute/region ....
```

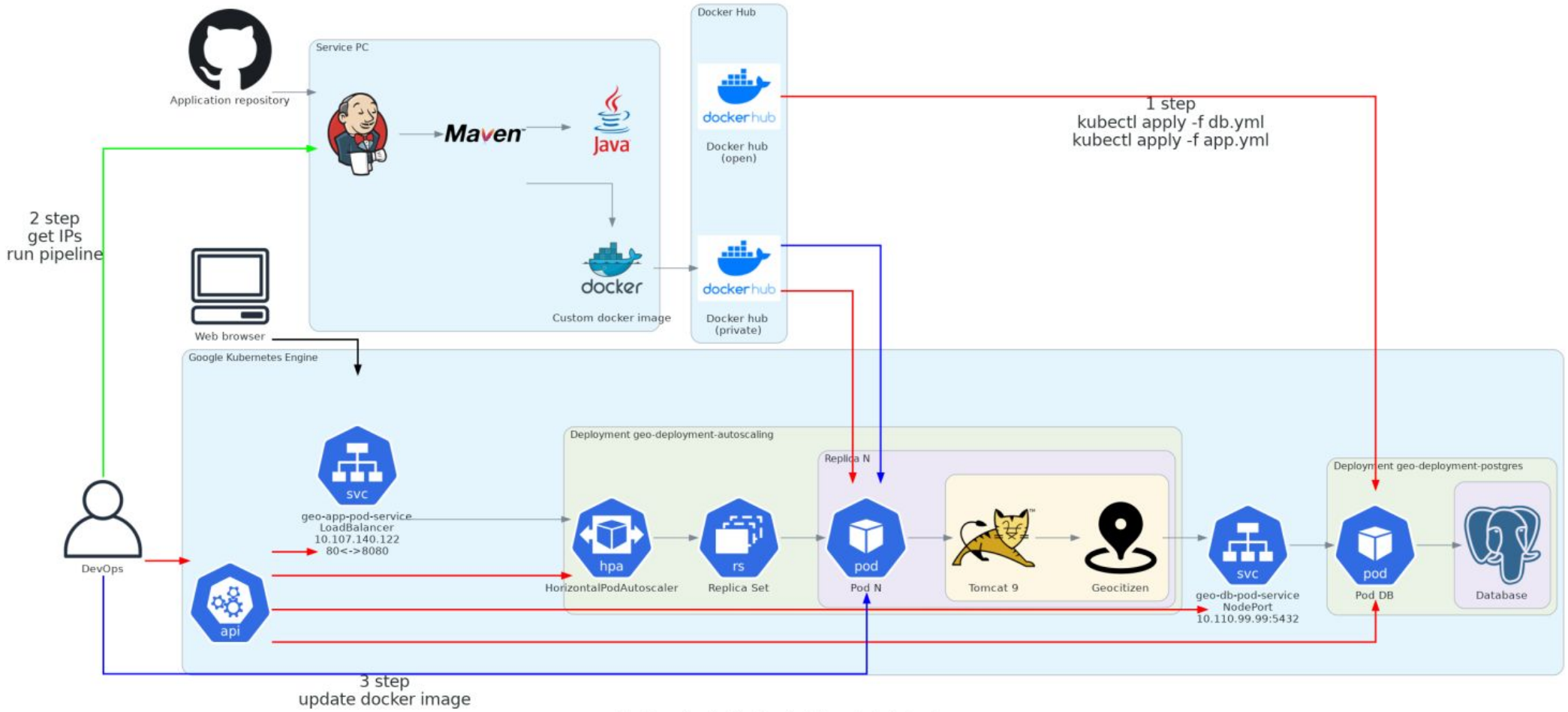
Install required plugins and connect to previously created cluster

```
$ sudo apt-get install google-cloud-sdk-gke-gcloud-auth-plugin
$ gcloud container clusters get-credentials CLUSTER_NAME
```

```
ubuntu@gcloud:~$ gcloud config set compute/region europe-west1
Updated property [compute/region].
ubuntu@gcloud:~$ gcloud container clusters get-credentials geocluster
Fetching cluster endpoint and auth data.
kubeconfig entry generated for geocluster.
```

softserve
Ivan Danyliuk

GKE GeoCitizen deployment



softserve

Ivan Danyliuk

GKE GeoCitizen deployment

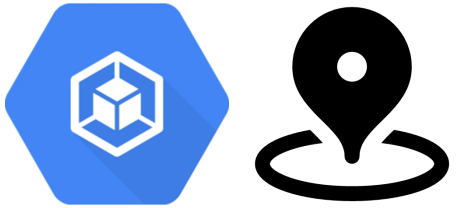
Create Infrastructure as for minikube and get IP addresses

```
ubuntu@gcloud:~$ kubectl get pods
W0511 05:56:13.463999    1366 gcp.go:120] WARNING: the gcp auth plugin is deprecated in v1.22+, unavailable in v1.25+; use gcloud in
stead.
To learn more, consult https://cloud.google.com/blog/products/containers-kubernetes/kubectl-auth-changes-in-gke
NAME                                        READY   STATUS    RESTARTS   AGE
geo-deployment-autoscaling-694b4647b6-2njlg   1/1     Running   0          7h
geo-deployment-autoscaling-694b4647b6-b8bls   1/1     Running   0          7h
geo-deployment-postgres-6fdd65557-z77bd       1/1     Running   0          7h30m
ubuntu@gcloud:~$ kubectl get services
W0511 05:56:25.815691    1370 gcp.go:120] WARNING: the gcp auth plugin is deprecated in v1.22+, unavailable in v1.25+; use gcloud in
stead.
To learn more, consult https://cloud.google.com/blog/products/containers-kubernetes/kubectl-auth-changes-in-gke
NAME                 TYPE           CLUSTER-IP    EXTERNAL-IP     PORT(S)          AGE
geo-app-pod-service  LoadBalancer   10.32.8.202   104.199.12.71   80:31298/TCP     7h30m
geo-db-pod-service   NodePort       10.32.5.140   <none>          5432:31594/TCP   7h31m
kubernetes           ClusterIP      10.32.0.1     <none>          443/TCP          8h
ubuntu@gcloud:~$ kubectl get deployments
W0511 05:56:38.333751    1374 gcp.go:120] WARNING: the gcp auth plugin is deprecated in v1.22+, unavailable in v1.25+; use gcloud in
stead.
To learn more, consult https://cloud.google.com/blog/products/containers-kubernetes/kubectl-auth-changes-in-gke
NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
geo-deployment-autoscaling   2/2     2            2           7h31m
geo-deployment-postgres      1/1     1            1           7h31m
```

Rebuild application war-file with real addresses and push it to docker hub with new tag, update image for App Load Balancer Pods

softserve
Ivan Danyliuk

app.yml for GKE



```yaml
apiVersion : apps/v1
kind: Deployment
metadata:
  name: geo-deployment-autoscaling
  labels:
    project : geocitizen
spec:
  selector:
    matchLabels:
      project: geocitizen-app
  template:
    metadata:
      labels:
        project: geocitizen-app  # Service will look
    spec:
      containers:
      - name : app-web
        image: docker.io/xbuyer/data:geo_minikube_v2
        imagePullPolicy: Always
        ports:
        - containerPort: 8080
      imagePullSecrets:
      - name: geosecret
```

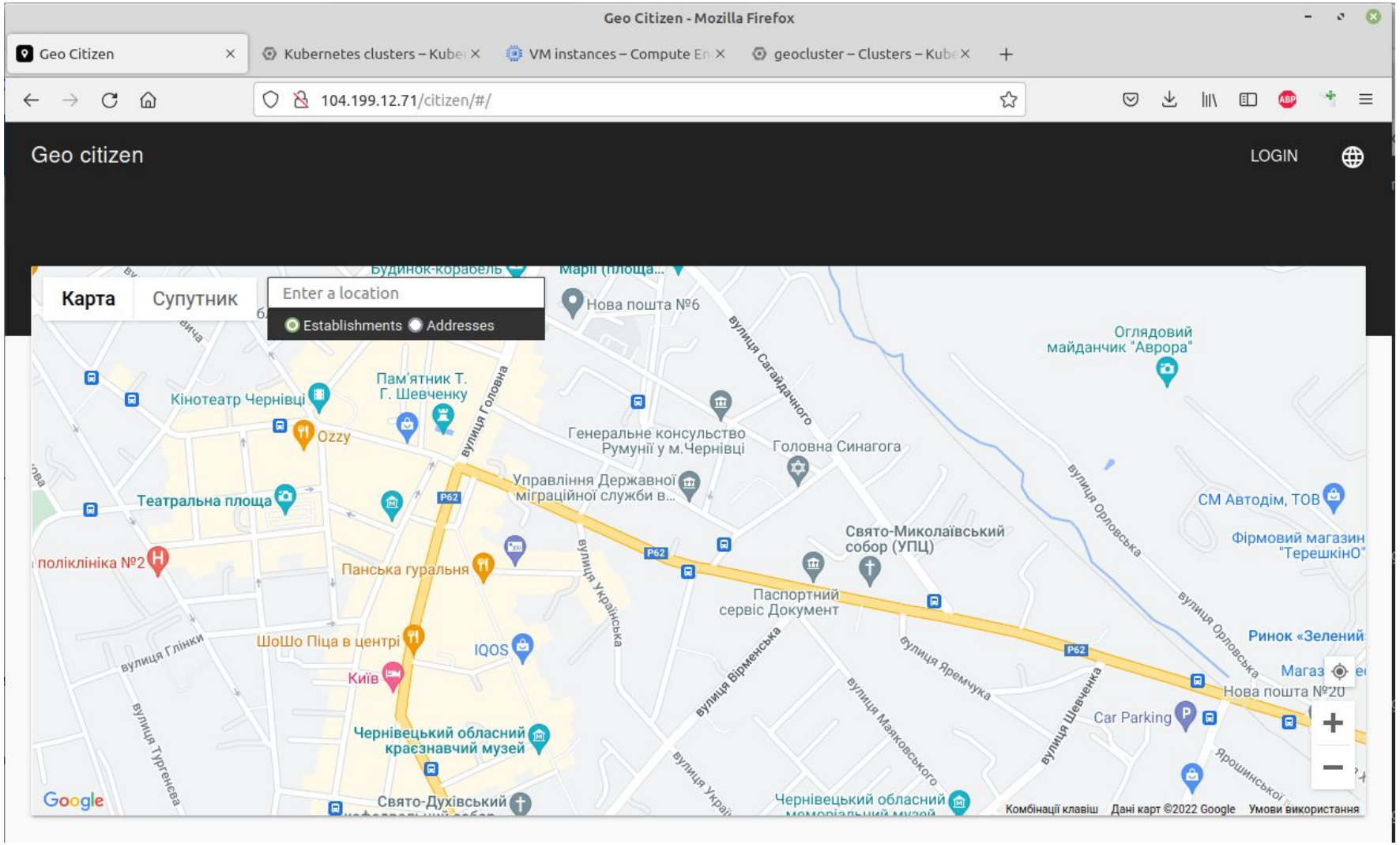
```yaml
---
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: geo-autoscaler
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: geo-deployment-autoscaling
  minReplicas: 2
  maxReplicas: 6
  targetCPUUtilizationPercentage: 50
```

Modified for GKE

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: geo-app-pod-service
  labels:
    env    : test
    owner: uixcoder
spec:
  selector:
    project:  geocitizen-app      # Selecting PODs
  ports:
    - name      : app-listener
      protocol  : TCP
      port      : 80  # Port on Load Balancer
      targetPort: 8080  # Port on Pod
  type: LoadBalancer
```

softserve
Ivan Danyliuk

GKE GeoCitizen deployment



softserve
Ivan Danyliuk

softserve