ASSIGNMENT 3

STEPS:

Step 1: Clone the Repository

Step 2: Build the Docker Image

```
docker build -t e-commerce-app .
```

Step 3: Start Minikube

```
minikube start --force
minikube status
```

Step 4: Load the Docker Image into Minikube

```
minikube image load e-commerce-app
```

Verify the image is loaded:

```
minikube image list # Ensure "e-commerce-app" is listed
```

Step 5: Deploy the Application

```
kubectl apply -f deployment.yml
kubectl get deployments
kubectl get pods
```

If you need a NodePort service, apply it:

```
kubectl apply -f Nodeport.yaml
```

Step 6: Fix Image Pull Issues (if necessary)

If Kubernetes tries to pull the image from a registry instead of using the local image, patch the deployment kubectl patch deployment react-ecommerce-deployment --type='json' p='[{"op": "replace", "path": "/spec/template/spec/containers/0/imagePullPolicy", "value": "Never"}]'

Step 7: Expose the Service & Access the App

```
minikube ip
minikube service react-ecommerce-service
```

Step 8:

Push to Git:

git init # If not already initialized git

add Dockerfile deployment.yml

Nodeport.yaml

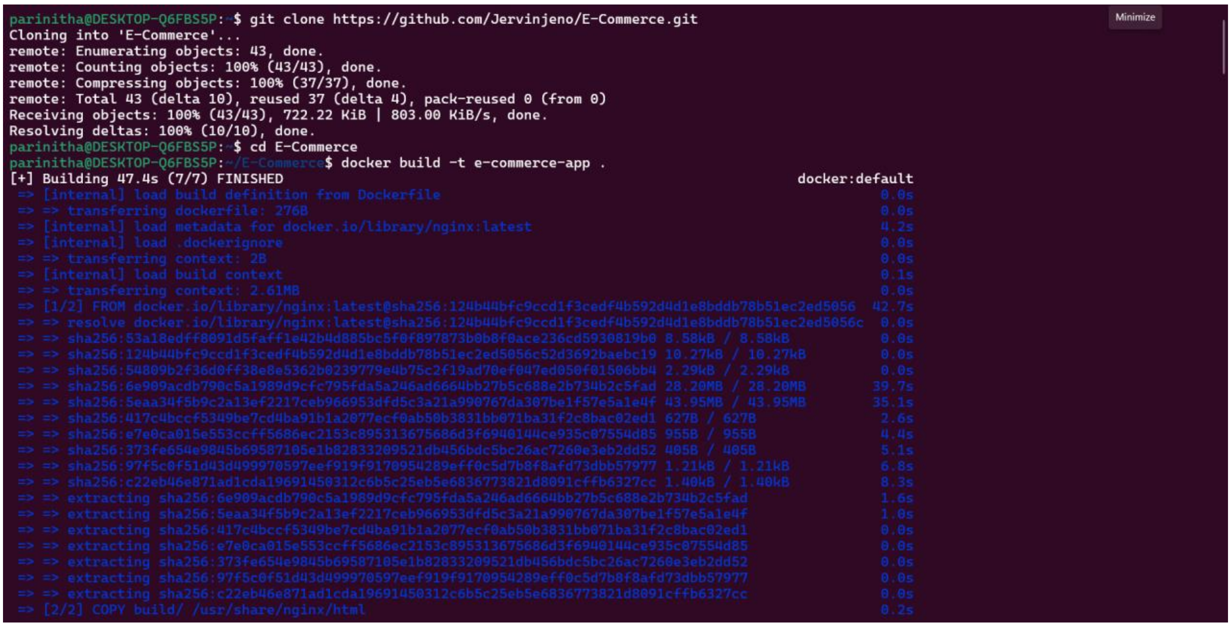
git commit -m "Kubernetes deployment for React e-commerce app"

git remote add origin

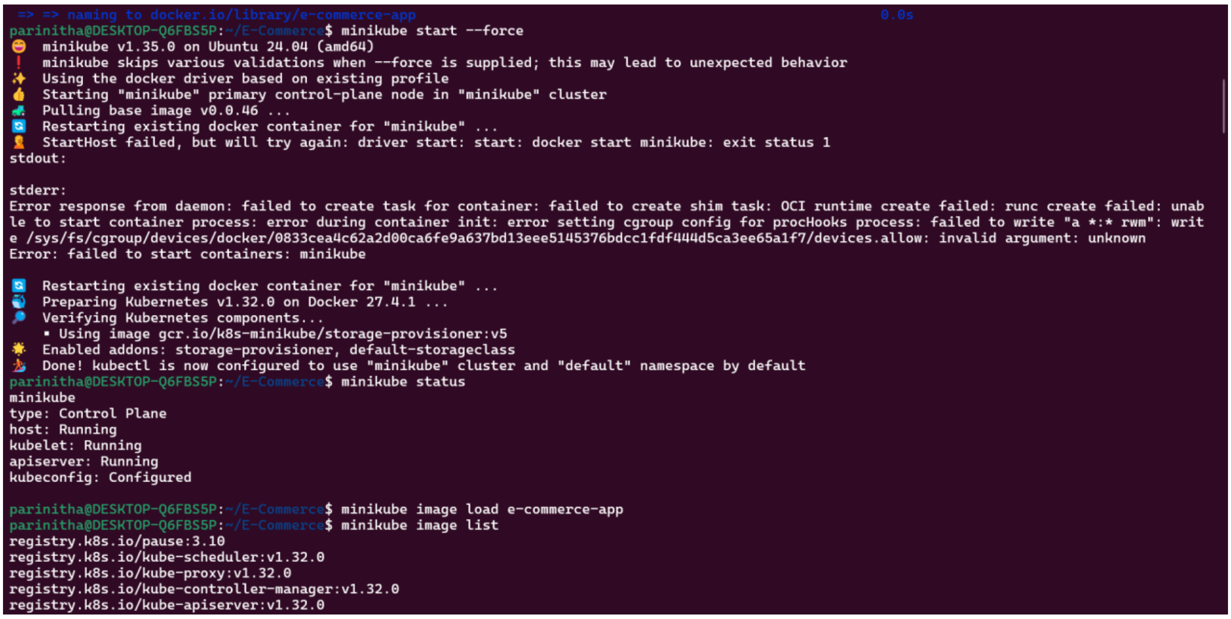
git branch -M main

git push -u origin main

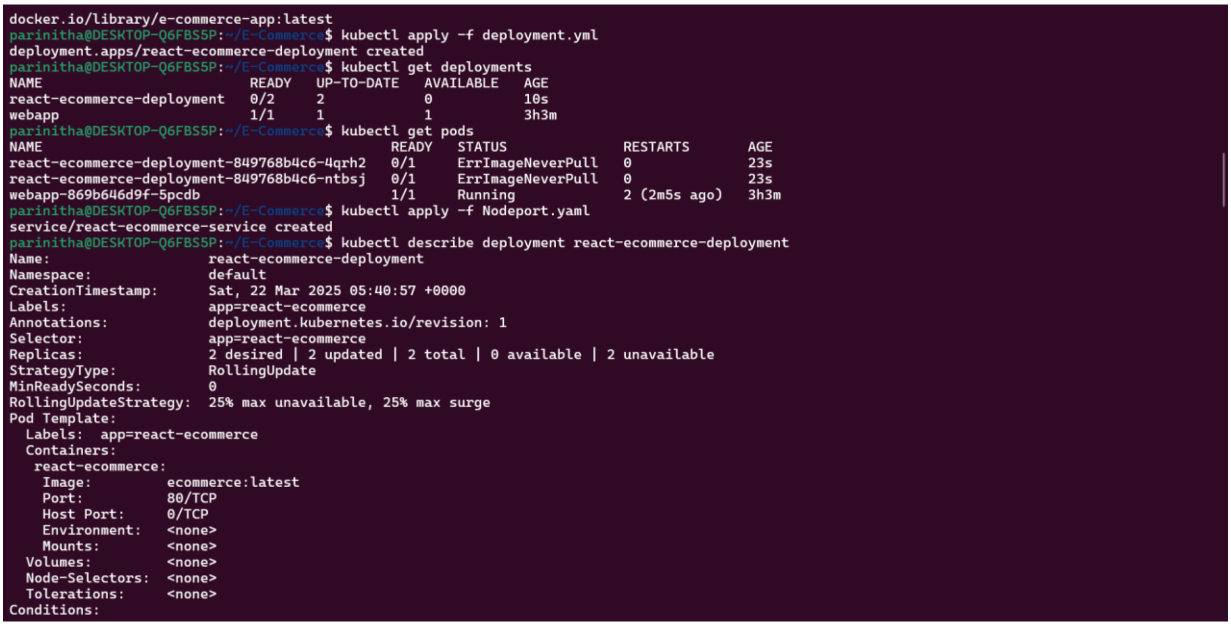
OUTPUT:



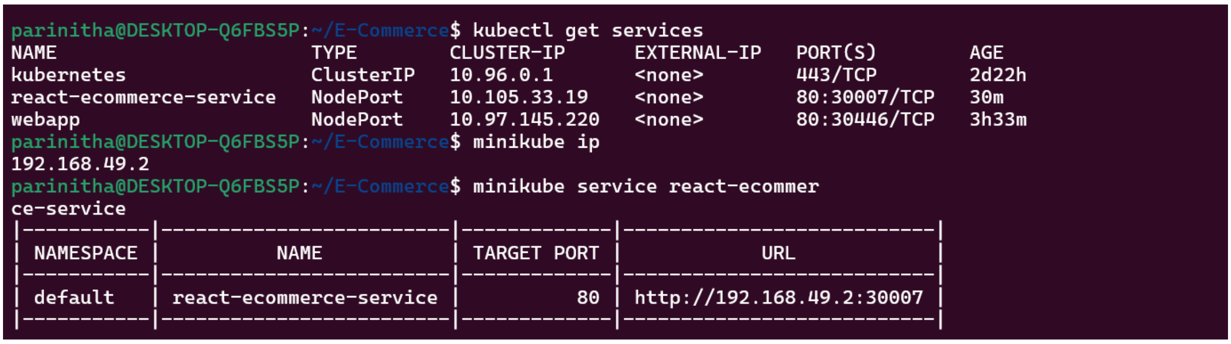
```
parinitha@DESKTOP-Q6FBS5P:~$ git clone https://github.com/Jervinjeno/E-Commerce.git
Cloning into 'E-Commerce'...
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 43 (delta 10), reused 37 (delta 4), pack-reused 0 (from 0)
Receiving objects: 100% (43/43), 722.22 KiB | 803.00 KiB/s, done.
Resolving deltas: 100% (10/10), done.
parinitha@DESKTOP-Q6FBS5P:~$ cd E-Commerce
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ docker build -t e-commerce-app .
[+] Building 47.4s (7/7) FINISHED                                                 docker:default
 => [internal] load build definition from Dockerfile                                        0.0s
 => => transferring dockerfile: 276B                                                        0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                             4.2s
 => [internal] load .dockerignore                                                           0.0s
 => => transferring context: 2B                                                             0.0s
 => [internal] load build context                                                           0.1s
 => => transferring context: 2.61MB                                                         0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056  42.7s
 => => resolve docker.io/library/nginx:latest@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c  0.0s
 => => sha256:53a18edff8091d5faff1e42b4d885bc5f0f897873b0b8f0ace236cd5930819b0 8.58kB / 8.58kB          0.0s
 => => sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19 10.27kB / 10.27kB        0.0s
 => => sha256:54809b2f36d0ff38e8e5362b0239779e4b75c2f19ad70ef047ed050f01506bb4 2.29kB / 2.29kB          0.0s
 => => sha256:6e909acdb790c5a1989d9cfc795fda5a246ad6664bb27b5c688e2b734b2c5fad 28.20MB / 28.20MB       39.7s
 => => sha256:5eaa34f5b9c2a13ef2217ceb966953dfd5c3a21a990767da307be1f57e5a1e4f 43.95MB / 43.95MB       35.1s
 => => sha256:417c4bccf5349be7cd4ba91b1a2077ecf0ab50b3831bb071ba31f2c8bac02ed1 627B / 627B               2.6s
 => => sha256:e7e0ca015e553ccff5686ec2153c895313675686d3f6940144ce935c07554d85 955B / 955B               4.4s
 => => sha256:373fe654e9845b69587105e1b82833209521db456bdc5bc26ac7260e3eb2dd52 405B / 405B               5.1s
 => => sha256:97f5c0f51d43d499970597eef919f9170954289eff0c5d7b8f8afd73dbb57977 1.21kB / 1.21kB          6.8s
 => => sha256:c22eb46e871ad1cda19691450312c6b5c25eb5e6836773821d8091cffb6327cc 1.40kB / 1.40kB          8.3s
 => => extracting sha256:6e909acdb790c5a1989d9cfc795fda5a246ad6664bb27b5c688e2b734b2c5fad               1.6s
 => => extracting sha256:5eaa34f5b9c2a13ef2217ceb966953dfd5c3a21a990767da307be1f57e5a1e4f               1.0s
 => => extracting sha256:417c4bccf5349be7cd4ba91b1a2077ecf0ab50b3831bb071ba31f2c8bac02ed1               0.0s
 => => extracting sha256:e7e0ca015e553ccff5686ec2153c895313675686d3f6940144ce935c07554d85               0.0s
 => => extracting sha256:373fe654e9845b69587105e1b82833209521db456bdc5bc26ac7260e3eb2dd52               0.0s
 => => extracting sha256:97f5c0f51d43d499970597eef919f9170954289eff0c5d7b8f8afd73dbb57977               0.0s
 => => extracting sha256:c22eb46e871ad1cda19691450312c6b5c25eb5e6836773821d8091cffb6327cc               0.0s
 => [2/2] COPY build/ /usr/share/nginx/html                                                 0.2s
```



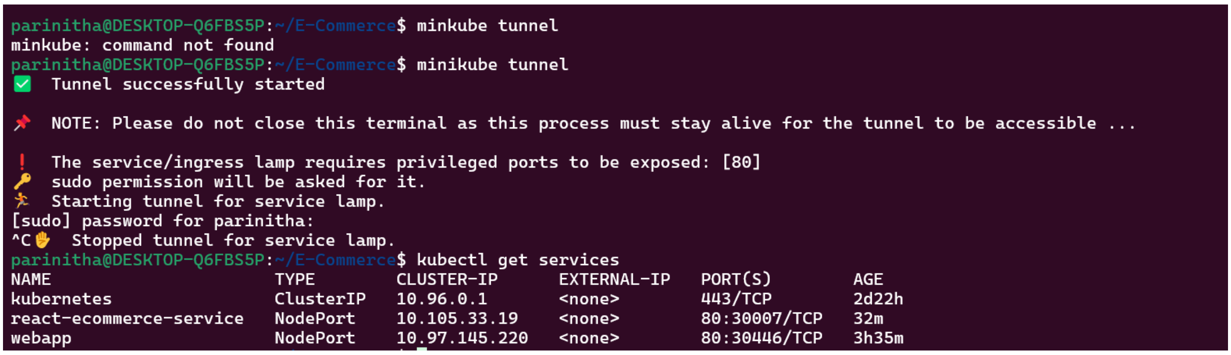
```
 => => naming to docker.io/library/e-commerce-app                                                   0.0s
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minikube start --force
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
❗  minikube skips various validations when --force is supplied; this may lead to unexpected behavior
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🔄  Restarting existing docker container for "minikube" ...
🤦  StartHost failed, but will try again: driver start: start: docker start minikube: exit status 1
stdout:

stderr:
Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unab
le to start container process: error during container init: error setting cgroup config for procHooks process: failed to write "a *:* rwm": writ
e /sys/fs/cgroup/devices/docker/0833cea4c62a2d00ca6fe9a637bd13eee5145376bdcc1fdf444d5ca3ee65a1f7/devices.allow: invalid argument: unknown
Error: failed to start containers: minikube

🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minikube image load e-commerce-app
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minikube image list
registry.k8s.io/pause:3.10
registry.k8s.io/kube-scheduler:v1.32.0
registry.k8s.io/kube-proxy:v1.32.0
registry.k8s.io/kube-controller-manager:v1.32.0
registry.k8s.io/kube-apiserver:v1.32.0
```



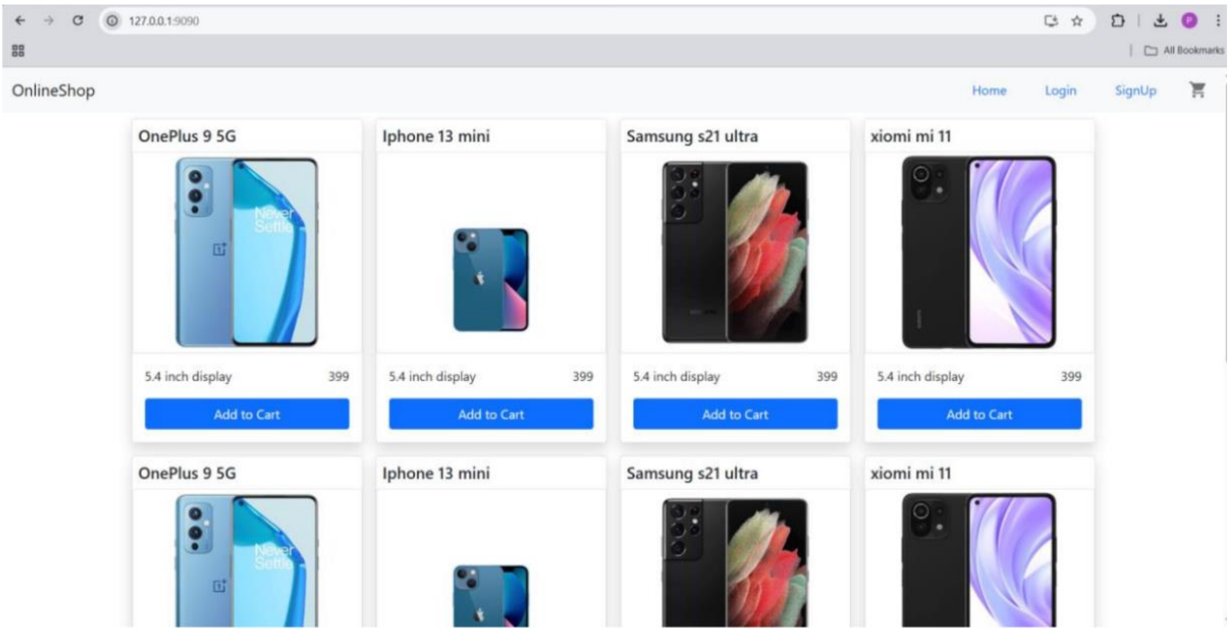
```
docker.io/library/e-commerce-app:latest
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ kubectl apply -f deployment.yml
deployment.apps/react-ecommerce-deployment created
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ kubectl get deployments
NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
react-ecommerce-deployment   0/2     2            0           10s
webapp                       1/1     1            1           3h3m
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ kubectl get pods
NAME                                          READY   STATUS              RESTARTS       AGE
react-ecommerce-deployment-849768b4c6-4qrh2   0/1     ErrImageNeverPull   0              23s
react-ecommerce-deployment-849768b4c6-ntbsj   0/1     ErrImageNeverPull   0              23s
webapp-869b646d9f-5pcdb                       1/1     Running             2 (2m5s ago)   3h3m
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ kubectl apply -f Nodeport.yaml
service/react-ecommerce-service created
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ kubectl describe deployment react-ecommerce-deployment
Name:                   react-ecommerce-deployment
Namespace:              default
CreationTimestamp:      Sat, 22 Mar 2025 05:40:57 +0000
Labels:                 app=react-ecommerce
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=react-ecommerce
Replicas:               2 desired | 2 updated | 2 total | 0 available | 2 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=react-ecommerce
  Containers:
   react-ecommerce:
    Image:        ecommerce:latest
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
  Node-Selectors: <none>
  Tolerations:    <none>
Conditions:
```



```
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ kubectl get services
NAME                      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                ClusterIP   10.96.0.1       <none>        443/TCP        2d22h
react-ecommerce-service   NodePort    10.105.33.19    <none>        80:30007/TCP   30m
webapp                    NodePort    10.97.145.220   <none>        80:30446/TCP   3h33m
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minikube ip
192.168.49.2
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minikube service react-ecommer
ce-service
|-----------|-------------------------|-------------|---------------------------|
| NAMESPACE |          NAME           | TARGET PORT |            URL            |
|-----------|-------------------------|-------------|---------------------------|
| default   | react-ecommerce-service |          80 | http://192.168.49.2:30007 |
|-----------|-------------------------|-------------|---------------------------|
```



```
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minkube tunnel
minkube: command not found
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ minikube tunnel
✅  Tunnel successfully started

📌  NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

❗  The service/ingress lamp requires privileged ports to be exposed: [80]
🔑  sudo permission will be asked for it.
🏃  Starting tunnel for service lamp.
[sudo] password for parinitha:
^C✋  Stopped tunnel for service lamp.
parinitha@DESKTOP-Q6FBS5P:~/E-Commerce$ kubectl get services
NAME                      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                ClusterIP   10.96.0.1       <none>        443/TCP        2d22h
react-ecommerce-service   NodePort    10.105.33.19    <none>        80:30007/TCP   32m
webapp                    NodePort    10.97.145.220   <none>        80:30446/TCP   3h35m
```



127.0.0.1:9090
All Bookmarks
OnlineShop
Home
Login
SignUp
OnePlus 9 5G
5.4 inch display
399
Add to Cart
Iphone 13 mini
5.4 inch display
399
Add to Cart
Samsung s21 ultra
5.4 inch display
399
Add to Cart
xiomi mi 11
5.4 inch display
399
Add to Cart
OnePlus 9 5G
Iphone 13 mini
Samsung s21 ultra
xiomi mi 11