



MSA Application Deployment

- Deploy each MicroService Manually
 - 상품, 주문, 배송 서비스 배포
 - 인증, 게이트웨이 배포
 - Front-End 배포

MSA Application - 12 Street

12 STREET

LOGOUT

USER INFO

마일리지 : 0 M


Home

Products

My Page(CQRS)

My Page(UI mashup)

10%
적립




[상품번호] TV
남은수량:999997

10000 원

@eal 한 일반 상품


10%
적립



[상품번호] TV
남은수량:999997

10000 원


10%
적립



[상품번호] MASK
남은수량:1999998

20000 원


10%
적립



[상품번호] NOTEBOOK
남은수량:2999999

30000 원

10%
적립

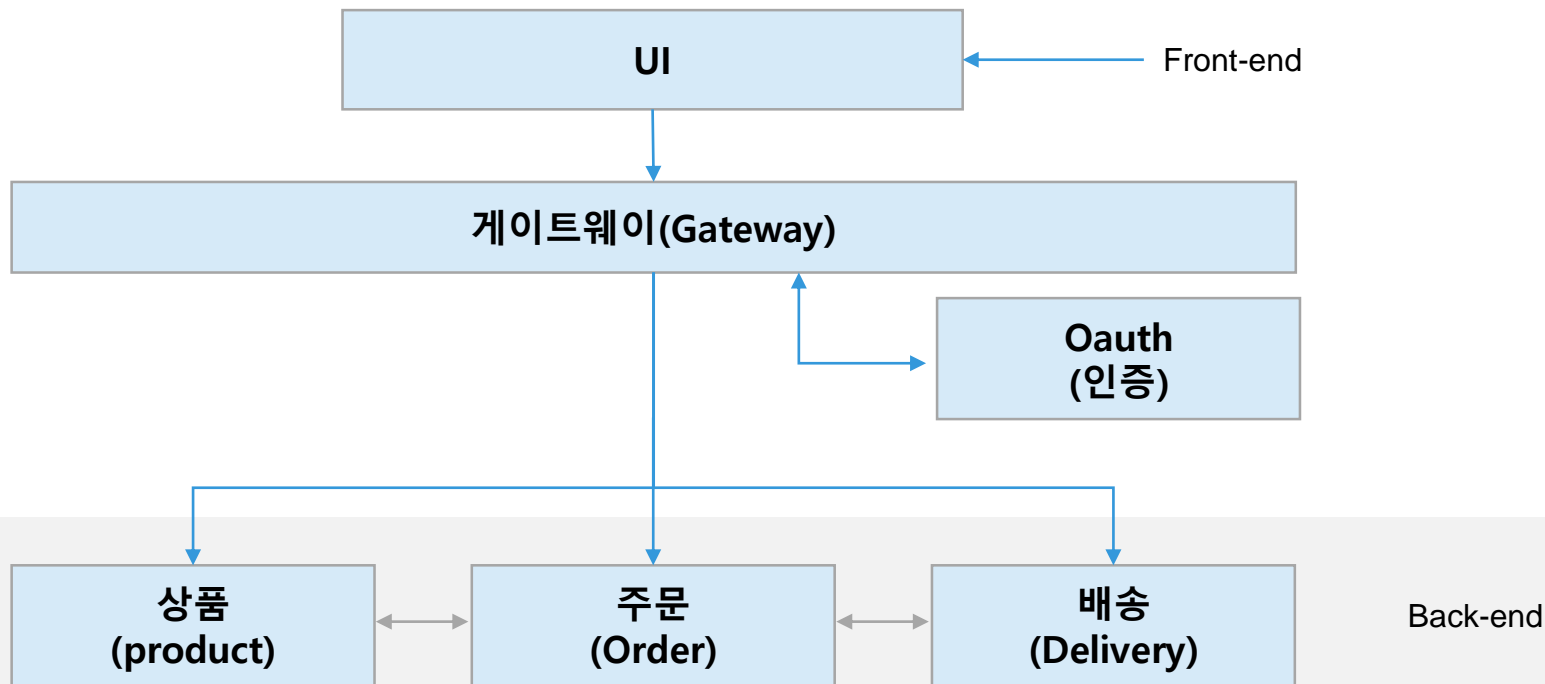


[상품번호] TABLE
남은수량:4000000

40000 원

2

MSA Service Structure



상품(Product) 서비스

1. Ubuntu 실행

- 실습스크립트 : <https://workflowy.com/s/msa/27a0ioMCzlpV04lb#/98ec94e6e26d>

2. Shell에서 아래 스크립트 실행

```
Mkdir MSA-Sample  
cd MSA-Sample
```

```
git clone https://github.com/event-storming/reqres\_products.git  
cd reqres_products  
mvn package -Dmaven.test.skip=true
```

```
az acr build --registry [acr-registry-name] --image [acr-registry-name].azurecr.io/products:latest .
```

```
kubectl run products --image= [acr-registry-name].azurecr.io/products:latest  
kubectl expose deploy products --type="ClusterIP" --port=8080
```

```
cd ..
```

주문(Order) 서비스

1. 공통된 작업을 편하게 하기 위해 Container Registry 를 환경변수로 설정
2. export 환경변수명=변수값
3. 확인

```
echo ${환경변수명}
```

```
export CRNAME=[acr-registry-name]  
export ACR=${CRNAME}.azurecr.io
```

```
git clone https://github.com/event-storming/regres\_orders.git  
cd regres_orders  
export IMAGENAME=orders
```

```
mvn package -Dmaven.test.skip=true  
az acr build --registry ${CRNAME} --image ${ACR}/${IMAGENAME}:latest .  
kubectl run ${IMAGENAME} --image=${ACR}/${IMAGENAME}:latest  
kubectl expose deploy ${IMAGENAME} --type="ClusterIP" --port=8080
```

```
cd ..
```

배송, 인증, 게이트웨이 서비스

- git clone https://github.com/event-storming/reqres_delivery.git
- export IMAGENAME=delivery
- git clone <https://github.com/event-storming/oauth.git>
- IMAGENAME=oauth
- git clone <https://github.com/event-storming/gateway.git>
- IMAGENAME=gateway
- 게이트웨이는 --type="LoadBalancer" 로 배포해야 함

UI 서비스

1. VueJs 로 되어있어서 nodejs 및 npm 으로 build 및 package
2. UI 에서 게이트웨이 주소를 바로 연결하기 위하여 env 를 설정

```
git clone https://github.com/event-storming/ui.git
```

```
cd ui
```

```
export IMAGENAME=ui
```

```
npm install
```

```
npm run build
```

```
az acr build --registry ${CRNAME} --image ${ACR}/${IMAGENAME}:latest .
```

```
_GATEWAY_IP=$(kubectl get -o jsonpath="{.status.loadBalancer.ingress[0].ip}" svc gateway --ignore-not-found)echo  
${_GATEWAY_IP}
```

UI 서비스 배포

```
cat <<EOF | kubectl apply -f -
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ${IMAGENAME}
  labels:
    app: ${IMAGENAME}
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ${IMAGENAME}
  template:
    metadata:
      labels:
        app: ${IMAGENAME}
    spec:
      containers:
        - name: ${IMAGENAME}
          image: ${ACR}/${IMAGENAME}:latest
          ports:
            - containerPort: 8080
          env:
            - name: VUE_APP_API_HOST
              value: http://${_GATEWAY_IP}:8080
EOF
```

```
kubectl expose deploy ${IMAGENAME} --type="LoadBalancer" --port=8080
cd ..
```