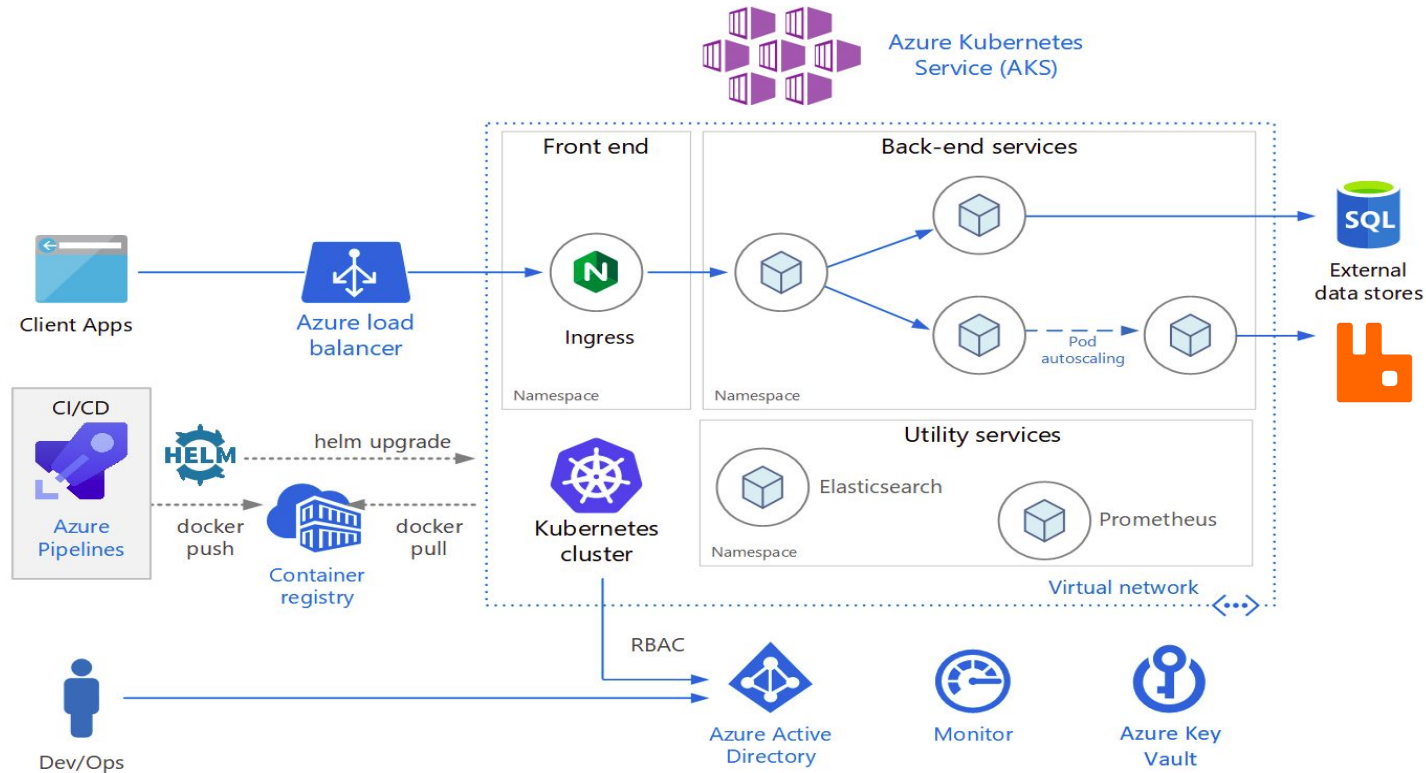


Microservice DevOps Architecture & Infrastructure

Prepared by Ashadullah Shawon

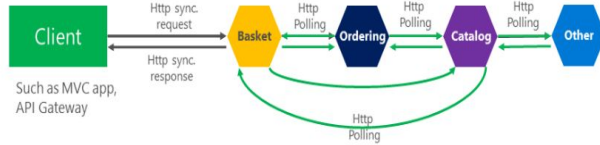
Microservice Cloud Infrastructure on Azure for Ecommerce



Microservice Architecture & Communication

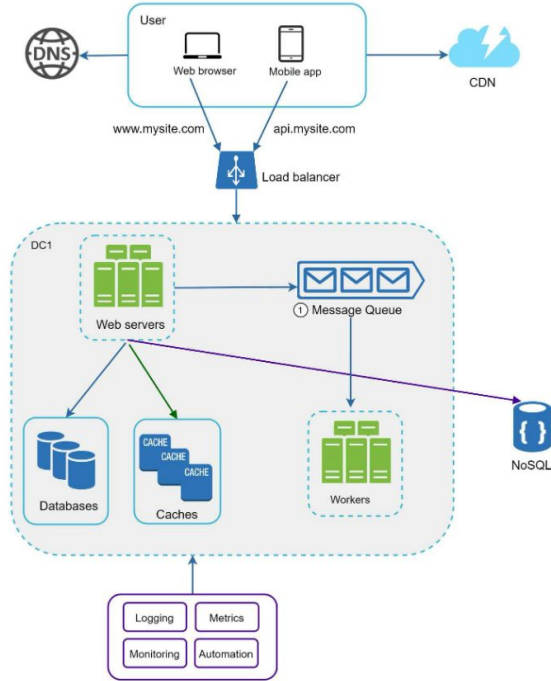
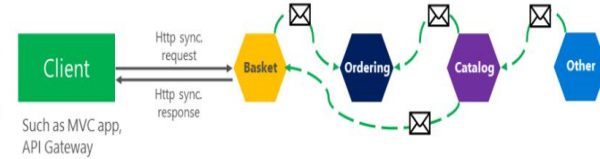
"Asynchronous"

Comm. across internal microservices
(Polling: **Http**)



Asynchronous

Comm. across internal microservices
(EventBus: like **AMQP**)

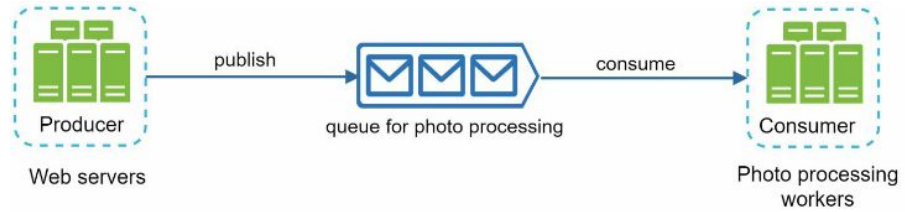
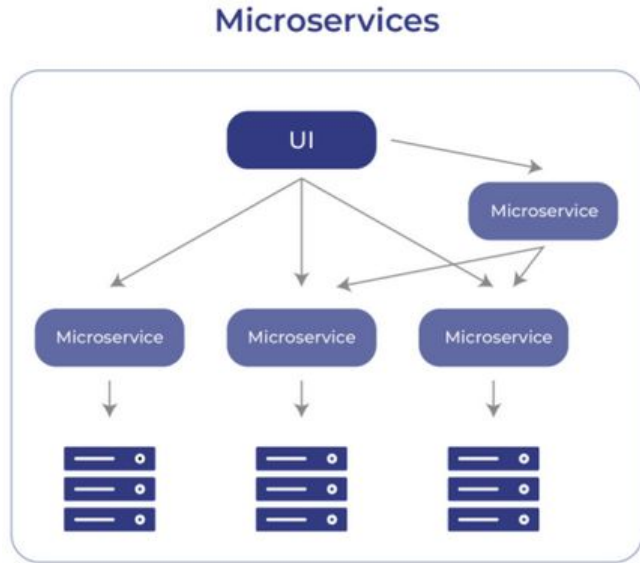


Event Driven Microservice Architecture

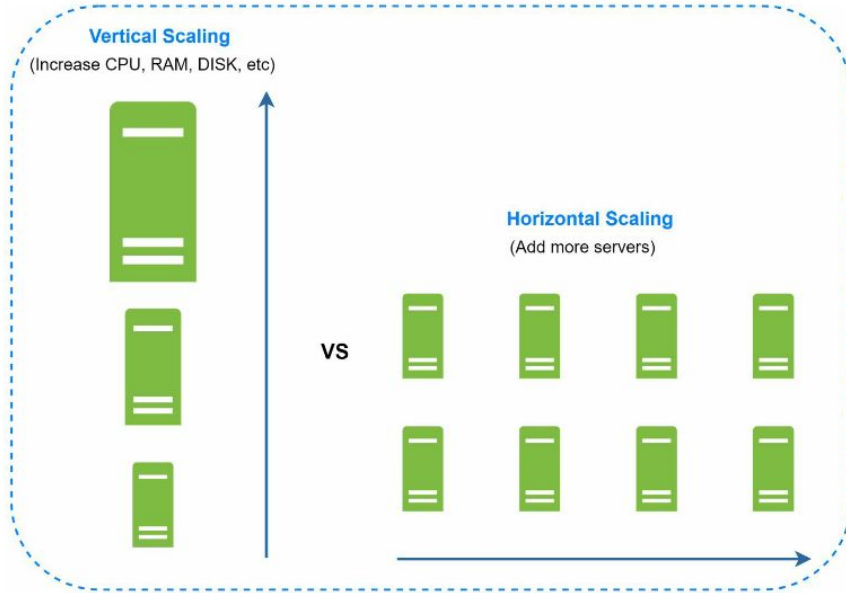
Microservices and event-driven computing have recently gained popularity. Modern microservices designs are

- Decentralized
- Loosely Coupled/ Connected
- Reactive and Event driven
- Asynchronous
- Lightweight
- Highly Scalable
- Highly Available
- Fault Tolerant

Event Driven Microservice Architecture



Horizontal & Vertical Auto Scaling for High Availability



Horizontal & Vertical Scaling in Kubernetes

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: web-servers
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: web-servers
  minReplicas: 1
  maxReplicas: 3
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 80
  - type: Resource
    resource:
      name: memory
      target:
        type: AverageValue
        averageValue: 30Mi
```

```
apiVersion: autoscaling.k8s.io/v1beta2
kind: VerticalPodAutoscaler
metadata:
  name: nginx-vpa
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind: Deployment
    name: nginx
```

Fault Tolerant

- Readiness Probe
- Liveness Probe

Pod Health checks

	Liveness	Readiness
On failure	Kill container	Stop sending traffic to pod
Check types	Http , exec , tcpSocket	Http , exec , tcpSocket
Declaration example (Pod.yaml)	<code>livenessProbe: failureThreshold: 3 httpGet: path: /healthz port: 8080</code>	<code>readinessProbe: httpGet: path: /status port: 8080</code>

Microservice Routing

Path Based Routing

Identity: `microservices.clodageskill.com/api/identity/v1/token`

Storage: `microservices.cloudageskill.com/api/storage/v1/storageservice`

Notification: `microservices.cloudageskill.com/api/notification/notifierservice`

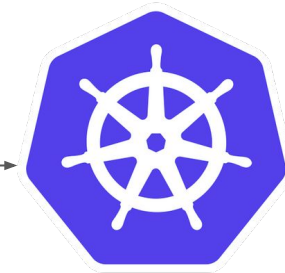
Mailservice: `microservices.cloudageskill.com/api/mailservice/mailservice`

GraphQL : `microservices.cloudageskill.com/api/graphql/gqlservice`

Microservice Automated Deployment with CI/CD



Docker Image			
✓	Build and Push (DEV)	1m 27s	
⊖	Initialize job	<1s	
✓	Checkout selise01/eca...	4s	
✓	Build Docker Image	1m 19s	
✓	Push Docker Image	3s	
✓	Post-job: Checkout s...	<1s	
⊖	Finalize Job	<1s	



Deploy To Kubernetes		
✓	Helm Generate and De...	13s
⊖	Initialize job	<1s
✓	Checkout selise01/eca...	2s
✓	Deploy with Helm Te...	9s
✓	Post-job: Checkout s...	<1s

Security

- Encrypt all communications (using https or transport layer security).
- Authenticate all access requests.
- Do not hard code certificates, passwords or any form of secrets within the code.
- Use DevSecOps tools designed for microservice architecture environments to scan code as it is developed.
- Define the APIs and strictly make sure all communications comply.
- Use Managed Load Balancer to protect microservices from cyber attacks

Challenges

- Operational complexity is likely to increase
- New skill sets will be required by your development teams and your operational teams (including security)
- You will need to have a microservice-friendly infrastructure in place to support your new continuous delivery workflows.
- Latency will necessarily be introduced by the need to traverse the network in order to execute a complete workflow.
- Hard to follow the best practices