

```

apiVersion: v1
kind: Secret
metadata:
  name: gift-secret
  labels:
    app: gift-service
    project: gift-service
    tier: backend
# From source_5 and source_11
data:
  JWT_SECRET: WW4ya2ppYmRkRkFXdG5QSjJBRmxMOFdYbW9oSk1Ddm1nUWdnYUV5cGE1RT0=
  KAKAO_CLIENT_ID: YWMyMGQ4NGY5N2I2NzlmMGJiZDlhODA2NWJhODIzZjM=
  KAKAO_CLIENT_SECRET: RnczSm1MNVBrcGV2WFZaOU9pVk04YmZiZ0hVUEQ1RUE=
# From source_5
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: gift-config
  labels:
    app: gift-service
    project: gift-service
    tier: backend
# From source_5 and source_11
data:
  API_GATEWAY_URI: http://api-gateway-svc:8080
  SERVICE_USER_URI: http://user-service-svc:8084
  SERVICE_PRODUCT_URI: http://product-service-svc:8081
  SERVICE_WISH_URI: http://wish-service-svc:8082
  SERVICE_ORDER_URI: http://order-service-svc:8083
  KAKAO_REDIRECT_URI: http://localhost:8080/members/login/oauth2/code/kakao
  FRONT_DOMAIN: http://localhost:8080
# From source_5
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-gateway
  labels:
    app: api-gateway
    project: gift-service
    tier: backend
# From source_1, source_6, and source_11
spec:
  replicas: 1
  selector:
    matchLabels:
      app: api-gateway
  template:
    metadata:
      labels:
        app: api-gateway

```

```

    project: gift-service
    tier: backend
spec:
  containers:
  - name: api-gateway
    image: sk124590/api-gateway:2.1
    ports:
    - containerPort: 8080
    resources:
      requests:
        cpu: "100m"
        memory: "256Mi"
      limits:
        cpu: "500m"
        memory: "512Mi"
  env:
  - name: SERVICE_USER_URI
    valueFrom:
      configMapKeyRef:
        name: gift-config
        key: SERVICE_USER_URI
  - name: SERVICE_PRODUCT_URI
    valueFrom:
      configMapKeyRef:
        name: gift-config
        key: SERVICE_PRODUCT_URI
  - name: SERVICE_WISH_URI
    valueFrom:
      configMapKeyRef:
        name: gift-config
        key: SERVICE_WISH_URI
  - name: SERVICE_ORDER_URI
    valueFrom:
      configMapKeyRef:
        name: gift-config
        key: SERVICE_ORDER_URI
  - name: JWT_SECRET
    valueFrom:
      secretKeyRef:
        name: gift-secret
        key: JWT_SECRET
# From source_6
---
apiVersion: v1
kind: Service
metadata:
  name: api-gateway-svc
  labels:
    app: api-gateway
    project: gift-service
    tier: backend
# From source_1 and source_11
spec:
  type: ClusterIP

```

```
ports:
- port: 8080
  targetPort: 8080
selector:
  app: api-gateway
# From source_12
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: user-service
  labels:
    app: user-service
    project: gift-service
    tier: backend
# From source_1, source_8, and source_11
spec:
  replicas: 1
  selector:
    matchLabels:
      app: user-service
  template:
    metadata:
      labels:
        app: user-service
        project: gift-service
        tier: backend
    spec:
      containers:
      - name: user-service
        image: sk124590/user-service:2.1
        ports:
        - containerPort: 8084
        resources:
          requests:
            cpu: "150m"
            memory: "512Mi"
          limits:
            cpu: "1000m"
            memory: "1Gi"
        envFrom:
        - configMapRef:
            name: gift-config
        - secretRef:
            name: gift-secret
# From source_8
---
apiVersion: v1
kind: Service
metadata:
  name: user-service-svc
  labels:
    app: user-service
    project: gift-service
```

```
    tier: backend
# From source_1 and source_11
spec:
  type: ClusterIP
  ports:
    - port: 8084
      targetPort: 8084
  selector:
    app: user-service
# From source_12
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: product-service
  labels:
    app: product-service
    project: gift-service
    tier: backend
# From source_1, source_10, and source_11
spec:
  replicas: 1
  selector:
    matchLabels:
      app: product-service
  template:
    metadata:
      labels:
        app: product-service
        project: gift-service
        tier: backend
    spec:
      containers:
        - name: product-service
          image: sk124590/product-service:2.1
          ports:
            - containerPort: 8081
          resources:
            requests:
              cpu: "150m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1Gi"
# From source_10
---
apiVersion: v1
kind: Service
metadata:
  name: product-service-svc
  labels:
    app: product-service
    project: gift-service
    tier: backend
```

```
# From source_1 and source_11
spec:
  type: ClusterIP
  ports:
    - port: 8081
      targetPort: 8081
  selector:
    app: product-service
# From source_12
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wish-service
  labels:
    app: wish-service
    project: gift-service
    tier: backend
# From source_1, source_4, and source_11
spec:
  replicas: 1
  selector:
    matchLabels:
      app: wish-service
  template:
    metadata:
      labels:
        app: wish-service
        project: gift-service
        tier: backend
    spec:
      containers:
        - name: wish-service
          image: sk124590/wish-service:2.1
          ports:
            - containerPort: 8082
          resources:
            requests:
              cpu: "100m"
              memory: "512Mi"
            limits:
              cpu: "700m"
              memory: "1Gi"
          envFrom:
            - configMapRef:
                name: gift-config
# From source_4
---
apiVersion: v1
kind: Service
metadata:
  name: wish-service-svc
  labels:
    app: wish-service
```

```
    project: gift-service
    tier: backend
# From source_1 and source_11
spec:
  type: ClusterIP
  ports:
    - port: 8082
      targetPort: 8082
  selector:
    app: wish-service
# From source_12
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: order-service
  labels:
    app: order-service
    project: gift-service
    tier: backend
# From source_1, source_7, and source_11
spec:
  replicas: 1
  selector:
    matchLabels:
      app: order-service
  template:
    metadata:
      labels:
        app: order-service
        project: gift-service
        tier: backend
    spec:
      containers:
        - name: order-service
          image: sk124590/order-service:2.1
          ports:
            - containerPort: 8083
          resources:
            requests:
              cpu: "100m"
              memory: "512Mi"
            limits:
              cpu: "700m"
              memory: "1Gi"
          envFrom:
            - configMapRef:
                name: gift-config
            - secretRef:
                name: gift-secret
# From source_7
---
apiVersion: v1
kind: Service
```

```
metadata:
  name: order-service-svc
  labels:
    app: order-service
    project: gift-service
    tier: backend
# From source_1 and source_11
spec:
  type: ClusterIP
  ports:
    - port: 8083
      targetPort: 8083
  selector:
    app: order-service
# From source_12
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gift-app
  labels:
    app: gift-app
    project: gift-service
    tier: frontend
# From source_1, source_8, and source_11
spec:
  replicas: 1
  selector:
    matchLabels:
      app: gift-app
  template:
    metadata:
      labels:
        app: gift-app
        project: gift-service
        tier: frontend
    spec:
      containers:
        - name: gift-app
          image: sk124590/gift-app:2.1
          ports:
            - containerPort: 8085
          resources:
            requests:
              cpu: "100m"
              memory: "512Mi"
            limits:
              cpu: "500m"
              memory: "1Gi"
      envFrom:
        - configMapRef:
            name: gift-config
        - secretRef:
            name: gift-secret
```

```
# From source_10
---
apiVersion: v1
kind: Service
metadata:
  name: gift-app-svc
  labels:
    app: gift-app
    project: gift-service
    tier: frontend
# From source_1 and source_11
spec:
  type: LoadBalancer
  ports:
    - port: 8080
      targetPort: 8085
  selector:
    app: gift-app
# From source_12
```

YAML 위치와 내용	참조된 Source	비고
gift-secret.Secret.metadata.labels = {app: gift-service, project: gift-service, tier: backend}	source_5, source_11	시크릿 공통 라벨 구성 값
gift-secret.Secret.data.JWT_SECRET = WW4y...T0=	source_5	JWT 시크릿(Base64)
gift-secret.Secret.data.KAKAO_CLIENT_ID = YWMY...ZjM=	source_5	카카오 클라이언트 ID(Base64)
gift-secret.Secret.data.KAKAO_CLIENT_SECRET = Rncz...RUE=	source_5	카카오 클라이언트 시크릿 (Base64)
gift-config.ConfigMap.metadata.labels = {app: gift-service, project: gift-service, tier: backend}	source_5, source_11	컨피그맵 공통 라벨 구성 값
gift-config.ConfigMap.data.API_GATEWAY_URI = http://api-gateway-svc:8080	source_5	마이크로서비스 내부 통신 주소
gift-config.ConfigMap.data.SERVICE_USER_URI = http://user-service-svc:8084	source_5	사용자 서비스 주소
gift-config.ConfigMap.data.SERVICE_PRODUCT_URI = http://product-service-svc:8081	source_5	상품 서비스 주소
gift-config.ConfigMap.data.SERVICE_WISH_URI = http://wish-service-svc:8082	source_5	위시 서비스 주소
gift-config.ConfigMap.data.SERVICE_ORDER_URI = http://order-service-svc:8083	source_5	주문 서비스 주소

YAML 위치와 내용	참조된 Source	비고
gift-config.ConfigMap.data.KAKAO_REDIRECT_URI = http://localhost:8080/members/login/oauth2/code/kakao	source_5	카카오 OAuth 리다이렉트 URI
gift-config.ConfigMap.data.FRONT_DOMAIN = http://localhost:8080	source_5	프론트 도메인
api-gateway.Deployment.spec (전체) = replicas:1, 이미지/포트/리소스/환경 등	source_1, source_6, source_11	API Gateway 배포 스펙 전반(이미지/리소스/포트/환경변수 기준)
api-gateway.Deployment.spec.template.spec.containers[0].env.* (SERVICE_*, JWT_SECRET 참조 설정) = configMapKeyRef/secretKeyRef	source_6	환경 변수 주입 방식 및 키 매핑 기준
api-gateway-svc.Service.spec (전체) = type: ClusterIP, ports: 8080→8080, selector.app: api-gateway	source_1, source_11	API Gateway 서비스 노출 스펙
user-service.Deployment.spec (전체) = replicas:1, 이미지/포트/리소스/환경 등	source_1, source_8, source_11	사용자 서비스 배포 스펙 전반
user-service.Deployment.spec.template.spec.containers[0].envFrom = [{configMapRef: gift-config}, {secretRef: gift-secret}]	source_8	환경 변수 일괄 주입 기준
user-service-svc.Service.spec (전체) = type: ClusterIP, ports: 8084→8084, selector.app: user-service	source_1, source_11	사용자 서비스 노출 스펙
product-service.Deployment.spec (전체) = replicas:1, 이미지/포트/리소스 등	source_1, source_10, source_11	상품 서비스 배포 스펙 전반
product-service.Deployment.spec.template.spec.containers[0].resources = requests/limits 설정	source_10	리소스 요청/제한 기준
product-service-svc.Service.spec (전체) = type: ClusterIP, ports: 8081→8081, selector.app: product-service	source_1, source_11	상품 서비스 노출 스펙
wish-service.Deployment.spec (전체) = replicas:1, 이미지/포트/리소스/환경 등	source_1, source_4, source_11	위시 서비스 배포 스펙 전반
wish-service.Deployment.spec.template.spec.containers[0].envFrom = [{configMapRef: gift-config}]	source_4	환경 변수 일괄 주입 기준
wish-service-svc.Service.spec (전체) = type: ClusterIP, ports: 8082→8082, selector.app: wish-service	source_1, source_11	위시 서비스 노출 스펙

YAML 위치와 내용	참조된 Source	비고
order-service.Deployment.spec (전체) = replicas:1, 이미지/포트/리소스/환경 등	source_1, source_7, source_11	주문 서비스 배포 스펙 전반
order-service.Deployment.spec.template.spec.containers[0].envFrom = [{configMapRef: gift-config}, {secretRef: gift-secret}]	source_7	환경 변수 일괄 주입 기준
order-service-svc.Service.spec (전체) = type: ClusterIP, ports: 8083→8083, selector.app: order-service	source_1, source_11	주문 서비스 노출 스펙
gift-app.Deployment.spec (전체) = replicas:1, 이미지/포트/리소스/환경 등	source_1, source_8, source_11	프론트엔드(BFF) 배포 스펙 전반
gift-app.Deployment.spec.template.spec.containers[0].envFrom = [{configMapRef: gift-config}, {secretRef: gift-secret}]	source_10	환경 변수 일괄 주입 기준
gift-app-svc.Service.spec (전체) = type: LoadBalancer, ports: 8080→8085, selector.app: gift-app	source_1, source_11	프론트엔드 서비스 노출 스펙