

```
apiVersion: v1
kind: Secret
metadata:
  name: gift-secret
  labels:
    app: gift-service
    project: gift-service
    tier: backend
  # Secret resource for storing sensitive information
  # From source_5 and source_13

data:
  JWT_SECRET: WW4ya2ppYmRkRkFXdG5QSjJBmRmxMOFdYbW9oSk1Ddm1nUWdnYUV5cGE1RT0=
  KAKAO_CLIENT_ID: YWMyMGQ4NGY5N2I2NzlmMGJiZDlhODA2NWJhODIzZjM=
  KAKAO_CLIENT_SECRET: RnczSm1MNVBrcGV2WFZaOU9pVk04YmZiZ0hVUEQ1RUE=
  ---
apiVersion: v1
kind: ConfigMap
metadata:
  name: gift-config
  labels:
    app: gift-service
    project: gift-service
    tier: backend
  # ConfigMap resource for storing non-sensitive configuration
  # From source_5

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
  ---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-gateway
  labels:
    app: api-gateway
    project: gift-service
    tier: backend
  # Deployment for api-gateway
  # 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"
        envFrom:
          - configMapRef:
              name: gift-config
          - secretRef:
              name: gift-secret
    ---
  apiVersion: v1
  kind: Service
  metadata:
    name: api-gateway-svc
    labels:
      app: api-gateway
      project: gift-service
      tier: backend
    # Service for api-gateway
    # From source_12
  spec:
    type: ClusterIP
    ports:
      - port: 8080
        targetPort: 8080
    selector:
      app: api-gateway
    ---
  apiVersion: apps/v1
  kind: Deployment
  metadata:
    name: user-service
    labels:
      app: user-service
      project: gift-service
      tier: backend
    # Deployment for user-service
    # 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
---
apiVersion: v1
kind: Service
metadata:
  name: user-service-svc
  labels:
    app: user-service
    project: gift-service
    tier: backend
  # Service for user-service
  # From source_12
spec:
  type: ClusterIP
  ports:
    - port: 8084
      targetPort: 8084
  selector:
    app: user-service
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: product-service
  labels:
    app: product-service
    project: gift-service
    tier: backend
  # Deployment for product-service
  # 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"
---
apiVersion: v1
kind: Service
metadata:
  name: product-service-svc
  labels:
    app: product-service
    project: gift-service
    tier: backend
  # Service for product-service
  # From source_12
spec:
  type: ClusterIP
  ports:
    - port: 8081
      targetPort: 8081
  selector:
    app: product-service
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wish-service
  labels:
    app: wish-service
    project: gift-service
    tier: backend
  # Deployment for wish-service
  # 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
---
apiVersion: v1
kind: Service
metadata:
  name: wish-service-svc
  labels:
    app: wish-service
    project: gift-service
    tier: backend
  # Service for wish-service
  # From source_12
spec:
  type: ClusterIP
  ports:
    - port: 8082
      targetPort: 8082
  selector:
    app: wish-service
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: order-service
  labels:
    app: order-service
    project: gift-service
    tier: backend
  # Deployment for order-service
  # 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
    ---
  apiVersion: v1
  kind: Service
  metadata:
    name: order-service-svc
    labels:
      app: order-service
      project: gift-service
      tier: backend
    # Service for order-service
    # From source_12
  spec:
    type: ClusterIP
    ports:
      - port: 8083
        targetPort: 8083
    selector:
      app: order-service
    ---
  apiVersion: apps/v1
  kind: Deployment
  metadata:
    name: gift-app
    labels:
      app: gift-app
      project: gift-service
      tier: frontend
    # Deployment for gift-app
```

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

YAML 위치와 내용	참조된 Source	비고
gift-secret.Secret.data.JWT_SECRET = WW4ya2ppYmRkRkFXdG5QSjJBRmxMOFdYbW9oSk1DdmInUWdnYUV5cGE1RT0=	source_5, source_13	내부 문서의 시크릿 값과 “민감 정보는 Secret에 저장” 정책을 반영
gift-secret.Secret.data.KAKAO_CLIENT_ID = YWMYMGQ4NGY5N2I2NzlmMGJiZDIhODA2NWJhODIzZjM=	source_5, source_13	카카오 클라이언트 ID 및 Secret을 Secret 리소스에 보관하도록 한 정책 준수
gift-secret.Secret.data.KAKAO_CLIENT_SECRET = RnczSmlMNVBrcGV2WFZaOU9pVk04YmZiZ0hVUEQ1RUE=	source_5, source_13	같은 맥락의 민감정보 관리 기준 적용
gift-config.ConfigMap.data.API_GATEWAY_URI = http://api-gateway-svc:8080	source_5	공통 구성 값(서비스 URI) 정의를 반영
gift-config.ConfigMap.data.SERVICE_USER_URI = http://user-service-svc:8084	source_5	공통 구성 값(서비스 URI) 정의를 반영
gift-config.ConfigMap.data.SERVICE_PRODUCT_URI = http://product-service-svc:8081	source_5	공통 구성 값(서비스 URI) 정의를 반영
gift-config.ConfigMap.data.SERVICE_WISH_URI = http://wish-service-svc:8082	source_5	공통 구성 값(서비스 URI) 정의를 반영
gift-config.ConfigMap.data.SERVICE_ORDER_URI = http://order-service-svc:8083	source_5	공통 구성 값(서비스 URI) 정의를 반영

YAML 위치와 내용	참조된 Source	비고
gift-config.ConfigMap.data.KAKAO_REDIRECT_URI = http://localhost:8080/members/login/oauth2/code/kakao	source_5	공통 리디렉트 URI 구성 반영
gift-config.ConfigMap.data.FRONT_DOMAIN = http://localhost:8080	source_5	프론트엔드 도메인 구성 반영
api-gateway.Deployment.spec.template.spec.containers[0].image = sk124590/api-gateway:2.1	source_1, source_6, source_11	서비스별 이미지/버전 및 배포 규칙 근거
api-gateway.Deployment.spec.template.spec.containers[0].resources.requests = {cpu: "100m", memory: "256Mi"}	source_1, source_6, source_11	자원 요청/제한 정책 근거
api-gateway.Deployment.spec.template.spec.containers[0].resources.limits = {cpu: "500m", memory: "512Mi"}	source_1, source_6, source_11	자원 상한 정책 근거
api-gateway.Deployment.spec.template.spec.containers[0].envFrom.configMapRef.name = gift-config	source_1, source_6, source_11	공통 ConfigMap 주입 기준
api-gateway.Deployment.spec.template.spec.containers[0].envFrom.secretRef.name = gift-secret	source_1, source_6, source_11	공통 Secret 주입 기준
api-gateway-svc.Service.spec.type = ClusterIP	source_12	서비스 노출 방식 기준
api-gateway-svc.Service.spec.ports[0] = {port: 8080, targetPort: 8080}	source_12	포트 매핑 규칙 근거
user-service.Deployment.spec.template.spec.containers[0].image = sk124590/user-service:2.1	source_1, source_8, source_11	서비스 이미지/버전 및 배포 규칙 근거
user-service.Deployment.spec.template.spec.containers[0].resources.requests = {cpu: "150m", memory: "512Mi"}	source_1, source_8, source_11	자원 요청 정책 근거
user-service.Deployment.spec.template.spec.containers[0].resources.limits = {cpu: "1000m", memory: "1Gi"}	source_1, source_8, source_11	자원 상한 정책 근거

YAML 위치와 내용	참조된 Source	비고
user-service.Deployment.spec.template.spec.containers[0].envFrom.configMapRef.name = gift-config	source_1, source_8, source_11	공통 ConfigMap 주입 기준
user-service.Deployment.spec.template.spec.containers[0].envFrom.secretRef.name = gift-secret	source_1, source_8, source_11	공통 Secret 주입 기준
user-service-svc.Service.spec.type = ClusterIP	source_12	서비스 노출 방식 기준
user-service-svc.Service.spec.ports[0] = {port: 8084, targetPort: 8084}	source_12	포트 매핑 규칙 근거
product-service.Deployment.spec.template.spec.containers[0].image = sk124590/product-service:2.1	source_1, source_10, source_11	서비스 이미지/버전 및 배포 규칙 근거
product-service.Deployment.spec.template.spec.containers[0].resources.requests = {cpu: "150m", memory: "512Mi"}	source_1, source_10, source_11	자원 요청 정책 근거
product-service.Deployment.spec.template.spec.containers[0].resources.limits = {cpu: "1000m", memory: "1Gi"}	source_1, source_10, source_11	자원 상한 정책 근거
product-service-svc.Service.spec.type = ClusterIP	source_12	서비스 노출 방식 기준
product-service-svc.Service.spec.ports[0] = {port: 8081, targetPort: 8081}	source_12	포트 매핑 규칙 근거
wish-service.Deployment.spec.template.spec.containers[0].image = sk124590/wish-service:2.1	source_1, source_4, source_11	서비스 이미지/버전 및 배포 규칙 근거
wish-service.Deployment.spec.template.spec.containers[0].resources.requests = {cpu: "100m", memory: "512Mi"}	source_1, source_4, source_11	자원 요청 정책 근거
wish-service.Deployment.spec.template.spec.containers[0].resources.limits = {cpu: "700m", memory: "1Gi"}	source_1, source_4, source_11	자원 상한 정책 근거

YAML 위치와 내용	참조된 Source	비고
wish-service.Deployment.spec.template.spec.containers[0].envFrom.configMapRef.name = gift-config	source_1, source_4, source_11	공통 ConfigMap 주입 기준
wish-service-svc.Service.spec.type = ClusterIP	source_12	서비스 노출 방식 기준
wish-service-svc.Service.spec.ports[0] = {port: 8082, targetPort: 8082}	source_12	포트 매핑 규칙 근거
order-service.Deployment.spec.template.spec.containers[0].image = sk124590/order-service:2.1	source_1, source_7, source_11	서비스 이미지/버전 및 배포 규칙 근거
order-service.Deployment.spec.template.spec.containers[0].resources.requests = {cpu: "100m", memory: "512Mi"}	source_1, source_7, source_11	자원 요청 정책 근거
order-service.Deployment.spec.template.spec.containers[0].resources.limits = {cpu: "700m", memory: "1Gi"}	source_1, source_7, source_11	자원 상한 정책 근거
order-service.Deployment.spec.template.spec.containers[0].envFrom.configMapRef.name = gift-config	source_1, source_7, source_11	공통 ConfigMap 주입 기준
order-service.Deployment.spec.template.spec.containers[0].envFrom.secretRef.name = gift-secret	source_1, source_7, source_11	공통 Secret 주 입 기준
order-service-svc.Service.spec.type = ClusterIP	source_12	서비스 노출 방식 기준
order-service-svc.Service.spec.ports[0] = {port: 8083, targetPort: 8083}	source_12	포트 매핑 규칙 근거
gift-app.Deployment.spec.template.spec.containers[0].image = sk124590/gift-app:2.1	source_1, source_8, source_11	프론트엔드 앱 이미지/버전 및 배포 규칙 근거
gift-app.Deployment.spec.template.spec.containers[0].resources.requests = {cpu: "100m", memory: "512Mi"}	source_1, source_8, source_11	자원 요청 정책 근거

YAML 위치와 내용	참조된 Source	비고
gift-app.Deployment.spec.template.spec.containers[0].resources.limits = {cpu: "500m", memory: "1Gi"}	source_1, source_8, source_11	자원 상한 정책 근거
gift-app.Deployment.spec.template.spec.containers[0].envFrom.configMapRef.name = gift-config	source_1, source_8, source_11	공통 ConfigMap 주입 기준
gift-app.Deployment.spec.template.spec.containers[0].envFrom.secretRef.name = gift-secret	source_1, source_8, source_11	공통 Secret 주입 기준
gift-app-svc.Service.spec.type = LoadBalancer	source_12	외부 노출 정책 근거
gift-app-svc.Service.spec.ports[0] = {port: 8080, targetPort: 8085}	source_12	프론트 서비스 포트 매핑 규칙 근거

특징: 배포 성공 및 모든 서비스 동작