

```

apiVersion: v1
kind: Secret
metadata:
  name: gift-secret # From source_5
  labels:
    app: gift-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
data:
  JWT_SECRET: WW4ya2ppYmRkRkFXdG5QSjJBRmxMOFdYbW9oSk1Ddm1nUWdnYUV5cGE1RT0= # From
source_5
  KAKAO_CLIENT_ID: YWMyMGQ4NGY5N2I2NzlmMGJiZDlhODA2NWJhODIzZjM= # From source_5
  KAKAO_CLIENT_SECRET: RnczSmIMNVBrcGV2WFZaOU9pVk04YmZiZ0hVUEQ1RUE= # From
source_2
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: gift-config # From source_5
  labels:
    app: gift-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
data:
  API_GATEWAY_URI: http://api-gateway-svc:8080 # From source_5
  SERVICE_USER_URI: http://user-service-svc:8084 # From source_5
  SERVICE_PRODUCT_URI: http://product-service-svc:8081 # From source_5
  SERVICE_WISH_URI: http://wish-service-svc:8082 # From source_5
  SERVICE_ORDER_URI: http://order-service-svc:8083 # From source_5
  KAKAO_REDIRECT_URI: http://localhost:8080/members/login/oauth2/code/kakao # From
source_5
  FRONT_DOMAIN: http://localhost:8080 # From source_5
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-gateway # From source_1
  labels:
    app: api-gateway # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  replicas: 1 # From source_6
  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 # From source_6
    ports:
    - containerPort: 8080 # From source_6
    envFrom:
    - configMapRef:
        name: gift-config
    - secretRef:
        name: gift-secret
  resources:
    requests:
      cpu: "100m" # From source_6
      memory: "256Mi" # From source_6
    limits:
      cpu: "500m" # From source_6
      memory: "512Mi" # From source_6
---
apiVersion: v1
kind: Service
metadata:
  name: api-gateway-svc # From source_1
  labels:
    app: api-gateway # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  type: ClusterIP # From source_12
  selector:
    app: api-gateway
  ports:
  - port: 8080 # From source_3
    targetPort: 8080 # From source_6
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: user-service # From source_1
  labels:
    app: user-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  replicas: 1 # From source_8
  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 # From source_8
      ports:
        - containerPort: 8084 # From source_8
      envFrom:
        - configMapRef:
            name: gift-config
        - secretRef:
            name: gift-secret
      resources:
        requests:
          cpu: "150m" # From source_8
          memory: "512Mi" # From source_8
        limits:
          cpu: "1000m" # From source_8
          memory: "1Gi" # From source_8
  ---
apiVersion: v1
kind: Service
metadata:
  name: user-service-svc # From source_1
  labels:
    app: user-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  type: ClusterIP # From source_12
  selector:
    app: user-service
  ports:
    - port: 8084 # From source_12
      targetPort: 8084 # From source_8
  ---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: product-service # From source_1
  labels:
    app: product-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  replicas: 1 # From source_10
  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 # From source_10
    ports:
    - containerPort: 8081 # From source_10
    resources:
      requests:
        cpu: "150m" # From source_10
        memory: "512Mi" # From source_10
      limits:
        cpu: "1000m" # From source_10
        memory: "1Gi" # From source_10
---
apiVersion: v1
kind: Service
metadata:
  name: product-service-svc # From source_1
  labels:
    app: product-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  type: ClusterIP # From source_12
  selector:
    app: product-service
  ports:
  - port: 8081 # From source_12
    targetPort: 8081 # From source_10
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wish-service # From source_1
  labels:
    app: wish-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  replicas: 1 # From source_4
  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 # From source_4
        ports:

```

```

    - containerPort: 8082 # From source_4
  envFrom:
  - configMapRef:
      name: gift-config
  resources:
    requests:
      cpu: "100m" # From source_4
      memory: "512Mi" # From source_4
    limits:
      cpu: "700m" # From source_4
      memory: "1Gi" # From source_4
---
apiVersion: v1
kind: Service
metadata:
  name: wish-service-svc # From source_1
  labels:
    app: wish-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  type: ClusterIP # From source_12
  selector:
    app: wish-service
  ports:
    - port: 8082 # From source_12
      targetPort: 8082 # From source_4
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: order-service # From source_1
  labels:
    app: order-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  replicas: 1 # From source_7
  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 # From source_7
          ports:
            - containerPort: 8083 # From source_7
          envFrom:

```

```

    - configMapRef:
      name: gift-config
    - secretRef:
      name: gift-secret
  resources:
    requests:
      cpu: "100m" # From source_7
      memory: "512Mi" # From source_7
    limits:
      cpu: "700m" # From source_7
      memory: "1Gi" # From source_7
---
apiVersion: v1
kind: Service
metadata:
  name: order-service-svc # From source_1
  labels:
    app: order-service # From source_11
    project: gift-service # From source_11
    tier: backend # From source_11
spec:
  type: ClusterIP # From source_12
  selector:
    app: order-service
  ports:
    - port: 8083 # From source_12
      targetPort: 8083 # From source_7
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gift-app # From source_1
  labels:
    app: gift-app # From source_11
    project: gift-service # From source_11
    tier: frontend # From source_11
spec:
  replicas: 1 # From source_10
  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 # From source_10
          ports:
            - containerPort: 8085 # From source_10
          envFrom:

```

```
- configMapRef:
  name: gift-config
- secretRef:
  name: gift-secret
resources:
  requests:
    cpu: "100m" # From source_10
    memory: "512Mi" # From source_10
  limits:
    cpu: "500m" # From source_10
    memory: "1Gi" # From source_10
---
apiVersion: v1
kind: Service
metadata:
  name: gift-app-svc # From source_1
  labels:
    app: gift-app # From source_11
    project: gift-service # From source_11
    tier: frontend # From source_11
spec:
  type: LoadBalancer # From source_12
  selector:
    app: gift-app
  ports:
    - port: 8080 # From source_12
      targetPort: 8085 # From source_10
```

YAML 위치와 내용	참조된 Source	비고
gift-secret.Secret.metadata.name = gift-secret	source_5	시크릿 리소스 명
gift-secret.Secret.metadata.labels.app = gift-service	source_11	공통 레이블 (app)
gift-secret.Secret.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
gift-secret.Secret.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
gift-secret.Secret.data.JWT_SECRET = WW4ya2pp...T0=	source_5	JWT 서명 키 (Base64)
gift-secret.Secret.data.KAKAO_CLIENT_ID = YWMY...MzM=	source_5	카카오 REST API 키(Base64)
gift-secret.Secret.data.KAKAO_CLIENT_SECRET = Rncz...RUE=	source_2	카카오 시크릿 키(Base64)

YAML 위치와 내용	참조된 Source	비고
gift-config.ConfigMap.metadata.name = gift-config	source_5	컨피그맵 리소스명
gift-config.ConfigMap.metadata.labels.app = gift-service	source_11	공통 레이블 (app)
gift-config.ConfigMap.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
gift-config.ConfigMap.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
gift-config.ConfigMap.data.API_GATEWAY_URI = <a href="http://api-gateway-svc:8080">http://api-gateway-svc:8080</a>	source_5	게이트웨이 내부 DNS
gift-config.ConfigMap.data.SERVICE_USER_URI = <a href="http://user-service-svc:8084">http://user-service-svc:8084</a>	source_5	사용자 서비스 URI
gift-config.ConfigMap.data.SERVICE_PRODUCT_URI = <a href="http://product-service-svc:8081">http://product-service-svc:8081</a>	source_5	상품 서비스 URI
gift-config.ConfigMap.data.SERVICE_WISH_URI = <a href="http://wish-service-svc:8082">http://wish-service-svc:8082</a>	source_5	위시 서비스 URI
gift-config.ConfigMap.data.SERVICE_ORDER_URI = <a href="http://order-service-svc:8083">http://order-service-svc:8083</a>	source_5	주문 서비스 URI
gift-config.ConfigMap.data.KAKAO_REDIRECT_URI = <a href="http://localhost:8080/members/login/oauth2/code/kakao">http://localhost:8080/members/login/oauth2/code/kakao</a>	source_5	카카오 리다이렉트
gift-config.ConfigMap.data.FRONT_DOMAIN = <a href="http://localhost:8080">http://localhost:8080</a>	source_5	프론트 도메인
api-gateway.Deployment.metadata.name = api-gateway	source_1	배포 대상 이름
api-gateway.Deployment.metadata.labels.app = api-gateway	source_11	공통 레이블 (app)
api-gateway.Deployment.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
api-gateway.Deployment.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
api-gateway.Deployment.spec.replicas = 1	source_6	레플리카 수
api-gateway.Deployment.spec.template.spec.containers[api-gateway].image = sk124590/api-gateway:2.1	source_6	컨테이너 이미지
api-gateway.Deployment.spec.template.spec.containers[api-gateway].ports[0].containerPort = 8080	source_6	컨테이너 포트



YAML 위치와 내용	참조된 Source	비고
api-gateway.Deployment.spec.template.spec.containers[api-gateway].resources.requests.cpu = 100m	source_6	요청 CPU
api-gateway.Deployment.spec.template.spec.containers[api-gateway].resources.requests.memory = 256Mi	source_6	요청 메모리
api-gateway.Deployment.spec.template.spec.containers[api-gateway].resources.limits.cpu = 500m	source_6	한도 CPU
api-gateway.Deployment.spec.template.spec.containers[api-gateway].resources.limits.memory = 512Mi	source_6	한도 메모리
api-gateway-svc.Service.metadata.name = api-gateway-svc	source_1	서비스 이름
api-gateway-svc.Service.metadata.labels.app = api-gateway	source_11	공통 레이블 (app)
api-gateway-svc.Service.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
api-gateway-svc.Service.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
api-gateway-svc.Service.spec.type = ClusterIP	source_12	서비스 타입
api-gateway-svc.Service.spec.ports[0].port = 8080	source_3	서비스 포트
api-gateway-svc.Service.spec.ports[0].targetPort = 8080	source_6	타겟 포트
user-service.Deployment.metadata.name = user-service	source_1	배포 대상 이름
user-service.Deployment.metadata.labels.app = user-service	source_11	공통 레이블 (app)
user-service.Deployment.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
user-service.Deployment.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
user-service.Deployment.spec.replicas = 1	source_8	레플리카 수
user-service.Deployment.spec.template.spec.containers[user-service].image = sk124590/user-service:2.1	source_8	컨테이너 이미지
user-service.Deployment.spec.template.spec.containers[user-service].ports[0].containerPort = 8084	source_8	컨테이너 포트
user-service.Deployment.spec.template.spec.containers[user-service].resources.requests.cpu = 150m	source_8	요청 CPU
user-service.Deployment.spec.template.spec.containers[user-service].resources.requests.memory = 512Mi	source_8	요청 메모리

YAML 위치와 내용	참조된 Source	비고
user-service.Deployment.spec.template.spec.containers[user-service].resources.limits.cpu = 1000m	source_8	한도 CPU
user-service.Deployment.spec.template.spec.containers[user-service].resources.limits.memory = 1Gi	source_8	한도 메모리
user-service-svc.Service.metadata.name = user-service-svc	source_1	서비스 이름
user-service-svc.Service.metadata.labels.app = user-service	source_11	공통 레이블 (app)
user-service-svc.Service.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
user-service-svc.Service.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
user-service-svc.Service.spec.type = ClusterIP	source_12	서비스 타입
user-service-svc.Service.spec.ports[0].port = 8084	source_12	서비스 포트
user-service-svc.Service.spec.ports[0].targetPort = 8084	source_8	타겟 포트
product-service.Deployment.metadata.name = product-service	source_1	배포 대상 이름
product-service.Deployment.metadata.labels.app = product-service	source_11	공통 레이블 (app)
product-service.Deployment.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
product-service.Deployment.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
product-service.Deployment.spec.replicas = 1	source_10	레플리카 수
product-service.Deployment.spec.template.spec.containers[product-service].image = sk124590/product-service:2.1	source_10	컨테이너 이미지
product-service.Deployment.spec.template.spec.containers[product-service].ports[0].containerPort = 8081	source_10	컨테이너 포트
product-service.Deployment.spec.template.spec.containers[product-service].resources.requests.cpu = 150m	source_10	요청 CPU
product-service.Deployment.spec.template.spec.containers[product-service].resources.requests.memory = 512Mi	source_10	요청 메모리
product-service.Deployment.spec.template.spec.containers[product-service].resources.limits.cpu = 1000m	source_10	한도 CPU
product-service.Deployment.spec.template.spec.containers[product-service].resources.limits.memory = 1Gi	source_10	한도 메모리

YAML 위치와 내용	참조된 Source	비고
product-service-svc.Service.metadata.name = product-service-svc	source_1	서비스 이름
product-service-svc.Service.metadata.labels.app = product-service	source_11	공통 레이블 (app)
product-service-svc.Service.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
product-service-svc.Service.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
product-service-svc.Service.spec.type = ClusterIP	source_12	서비스 타입
product-service-svc.Service.spec.ports[0].port = 8081	source_12	서비스 포트
product-service-svc.Service.spec.ports[0].targetPort = 8081	source_10	타겟 포트
wish-service.Deployment.metadata.name = wish-service	source_1	배포 대상 이름
wish-service.Deployment.metadata.labels.app = wish-service	source_11	공통 레이블 (app)
wish-service.Deployment.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
wish-service.Deployment.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
wish-service.Deployment.spec.replicas = 1	source_4	레플리카 수
wish-service.Deployment.spec.template.spec.containers[wish-service].image = sk124590/wish-service:2.1	source_4	컨테이너 이미지
wish-service.Deployment.spec.template.spec.containers[wish-service].ports[0].containerPort = 8082	source_4	컨테이너 포트
wish-service.Deployment.spec.template.spec.containers[wish-service].resources.requests.cpu = 100m	source_4	요청 CPU
wish-service.Deployment.spec.template.spec.containers[wish-service].resources.requests.memory = 512Mi	source_4	요청 메모리
wish-service.Deployment.spec.template.spec.containers[wish-service].resources.limits.cpu = 700m	source_4	한도 CPU
wish-service.Deployment.spec.template.spec.containers[wish-service].resources.limits.memory = 1Gi	source_4	한도 메모리
wish-service-svc.Service.metadata.name = wish-service-svc	source_1	서비스 이름
wish-service-svc.Service.metadata.labels.app = wish-service	source_11	공통 레이블 (app)

YAML 위치와 내용	참조된 Source	비고
wish-service-svc.Service.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
wish-service-svc.Service.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
wish-service-svc.Service.spec.type = ClusterIP	source_12	서비스 타입
wish-service-svc.Service.spec.ports[0].port = 8082	source_12	서비스 포트
wish-service-svc.Service.spec.ports[0].targetPort = 8082	source_4	타겟 포트
order-service.Deployment.metadata.name = order-service	source_1	배포 대상 이름
order-service.Deployment.metadata.labels.app = order-service	source_11	공통 레이블 (app)
order-service.Deployment.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
order-service.Deployment.metadata.labels.tier = backend	source_11	공통 레이블 (tier)
order-service.Deployment.spec.replicas = 1	source_7	레플리카 수
order-service.Deployment.spec.template.spec.containers[order-service].image = sk124590/order-service:2.1	source_7	컨테이너 이미지
order-service.Deployment.spec.template.spec.containers[order-service].ports[0].containerPort = 8083	source_7	컨테이너 포트
order-service.Deployment.spec.template.spec.containers[order-service].resources.requests.cpu = 100m	source_7	요청 CPU
order-service.Deployment.spec.template.spec.containers[order-service].resources.requests.memory = 512Mi	source_7	요청 메모리
order-service.Deployment.spec.template.spec.containers[order-service].resources.limits.cpu = 700m	source_7	한도 CPU
order-service.Deployment.spec.template.spec.containers[order-service].resources.limits.memory = 1Gi	source_7	한도 메모리
order-service-svc.Service.metadata.name = order-service-svc	source_1	서비스 이름
order-service-svc.Service.metadata.labels.app = order-service	source_11	공통 레이블 (app)
order-service-svc.Service.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
order-service-svc.Service.metadata.labels.tier = backend	source_11	공통 레이블 (tier)

YAML 위치와 내용	참조된 Source	비고
order-service-svc.Service.spec.type = ClusterIP	source_12	서비스 타입
order-service-svc.Service.spec.ports[0].port = 8083	source_12	서비스 포트
order-service-svc.Service.spec.ports[0].targetPort = 8083	source_7	타겟 포트
gift-app.Deployment.metadata.name = gift-app	source_1	배포 대상 이름
gift-app.Deployment.metadata.labels.app = gift-app	source_11	공통 레이블 (app)
gift-app.Deployment.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
gift-app.Deployment.metadata.labels.tier = frontend	source_11	공통 레이블 (tier)
gift-app.Deployment.spec.replicas = 1	source_10	레플리카 수
gift-app.Deployment.spec.template.spec.containers[gift-app].image = sk124590/gift-app:2.1	source_10	컨테이너 이미지
gift-app.Deployment.spec.template.spec.containers[gift-app].ports[0].containerPort = 8085	source_10	컨테이너 포트
gift-app.Deployment.spec.template.spec.containers[gift-app].resources.requests.cpu = 100m	source_10	요청 CPU
gift-app.Deployment.spec.template.spec.containers[gift-app].resources.requests.memory = 512Mi	source_10	요청 메모리
gift-app.Deployment.spec.template.spec.containers[gift-app].resources.limits.cpu = 500m	source_10	한도 CPU
gift-app.Deployment.spec.template.spec.containers[gift-app].resources.limits.memory = 1Gi	source_10	한도 메모리
gift-app-svc.Service.metadata.name = gift-app-svc	source_1	서비스 이름
gift-app-svc.Service.metadata.labels.app = gift-app	source_11	공통 레이블 (app)
gift-app-svc.Service.metadata.labels.project = gift-service	source_11	공통 레이블 (project)
gift-app-svc.Service.metadata.labels.tier = frontend	source_11	공통 레이블 (tier)
gift-app-svc.Service.spec.type = LoadBalancer	source_12	외부 노출 타입
gift-app-svc.Service.spec.ports[0].port = 8080	source_12	외부 포트
gift-app-svc.Service.spec.ports[0].targetPort = 8085	source_10	타겟 포트