

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: gift-app
  labels:
    app: gift-app # From source_12
    project: gift-service # From source_11
    tier: frontend # From source_12
spec:
  replicas: 1 # From source_2
  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_2
          ports:
            - containerPort: 8085 # From source_2
          env:
            - name: API_GATEWAY_URI
              valueFrom:
                configMapKeyRef:
                  name: gift-config
                  key: API_GATEWAY_URI # From source_2
            - name: KAKAO_REDIRECT_URI
              valueFrom:
                configMapKeyRef:
                  name: gift-config
                  key: KAKAO_REDIRECT_URI # From source_2
            - name: JWT_SECRET
              valueFrom:
                secretKeyRef:
                  name: gift-secret
                  key: JWT_SECRET # From source_2
      resources:
        requests:
          cpu: "150m" # From source_2
          memory: "512Mi" # From source_2
        limits:
          cpu: "1000m" # From source_2
          memory: "1Gi" # From source_2
---
apiVersion: v1
kind: Service
metadata:

```

```
name: gift-app-svc
labels:
  app: gift-app # From source_12
  project: gift-service # From source_11
  tier: frontend # From source_12
spec:
  type: LoadBalancer # From source_13
  ports:
    - port: 8080 # From source_13
      targetPort: 8085 # From source_13
  selector:
    app: gift-app
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: gift-config
  labels:
    app: gift-app # From source_12
    project: gift-service # From source_11
spec:
  data:
    API_GATEWAY_URI: "http://api-gateway-svc:8080"
    KAKAO_REDIRECT_URI: "http://localhost:8080/members/login/oauth2/code/kakao"
---
apiVersion: v1
kind: Secret
metadata:
  name: gift-secret
  labels:
    app: gift-app # From source_12
    project: gift-service # From source_11
    tier: frontend # From source_12
spec:
  data:
    JWT_SECRET: WW4ya2ppYmRkRkFXdG5QSjJBRmxMOFdYbW9oSk1Ddm1nUWdnYUV5cGE1RT0= #
From source_3
    KAKAO_CLIENT_ID: YWMyMGQ4NGY5N2I2NzlmMGJiZDlhODA2NWJhODIzZjM= # From source_8
    KAKAO_CLIENT_SECRET: RnczSm1MNVBrcGV2WFZaOU9pVk04YmZiZ0hVUEQ1RUE= # From
source_8
```

YAML 위치와 내용	참조된 Source	비고
gift-app.Deployment.metadata.labels.app = gift-app	source_12	레이블 규칙에 따라 app 라벨 정의
gift-app.Deployment.metadata.labels.project = gift-service	source_11	모든 리소스에 project: gift-service 포함 규칙

YAML 위치와 내용	참조된 Source	비고
gift-app.Deployment.metadata.labels.tier = frontend	source_12	레이블 규칙에 따라 tier(frontend) 지정
gift-app.Deployment.spec.replicas = 1	source_2	gift-app 마이크로서비스 정의에서 복제 수 1
gift-app.Deployment.spec.template.spec.containers[0].image = sk124590/gift-app:2.1	source_2	gift-app BFF 서비스의 컨테이너 이미지
gift-app.Deployment.spec.template.spec.containers[0].ports[0].containerPort = 8085	source_2	gift-app 컨테이너 포트
gift-app.Deployment.spec.template.spec.containers[0].env.API_GATEWAY_URI.key = API_GATEWAY_URI	source_2	gift-app이 api-gateway에 의존
gift-app.Deployment.spec.template.spec.containers[0].env.KAKAO_REDIRECT_URI.key = KAKAO_REDIRECT_URI	source_2	카카오 인증 리다이렉트 URI 환경 변수
gift-app.Deployment.spec.template.spec.containers[0].env.JWT_SECRET.key = JWT_SECRET	source_2	JWT_SECRET 값을 secret에서 가져옴
gift-app.Deployment.spec.template.spec.containers[0].resources.requests.cpu = 150m	source_2	gift-app 리소스 요청 CPU 값
gift-app.Deployment.spec.template.spec.containers[0].resources.requests.memory = 512Mi	source_2	gift-app 리소스 요청 Memory 값
gift-app.Deployment.spec.template.spec.containers[0].resources.limits.cpu = 1000m	source_2	gift-app 리소스 한계 CPU 값
gift-app.Deployment.spec.template.spec.containers[0].resources.limits.memory = 1Gi	source_2	gift-app 리소스 한계 Memory 값

YAML 위치와 내용	참조된 Source	비고
gift-app-svc.Service.metadata.labels.app = gift-app	source_12	서비스 리소스에도 동일 app 라벨 적용
gift-app-svc.Service.metadata.labels.project = gift-service	source_11	서비스 리소스에도 동일 project 라벨 적용
gift-app-svc.Service.metadata.labels.tier = frontend	source_12	서비스 tier: frontend
gift-app-svc.Service.spec.type = LoadBalancer	source_13	외부 트래픽 노출은 LoadBalancer 타입 사용
gift-app-svc.Service.spec.ports[0].port = 8080	source_13	외부 서비스 포트 8080
gift-app-svc.Service.spec.ports[0].targetPort = 8085	source_13	gift-app 컨테이너 내부 포트 8085
gift-config.ConfigMap.metadata.labels.app = gift-app	source_12	ConfigMap에도 동일 app 라벨 적용
gift-config.ConfigMap.metadata.labels.project = gift-service	source_11	ConfigMap에도 동일 project 라벨 적용
gift-secret.Secret.metadata.labels.app = gift-app	source_12	Secret 리소스 app 라벨
gift-secret.Secret.metadata.labels.project = gift-service	source_11	Secret 리소스 project 라벨
gift-secret.Secret.metadata.labels.tier = frontend	source_12	Secret 리소스 tier 라벨
gift-secret.Secret.data.JWT_SECRET = WW4y... (Base64)	source_3	gift-secret에 JWT_SECRET Base64 값 정의

YAML 위치와 내용	참조된 Source	비고
gift-secret.Secret.data.KAKAO_CLIENT_ID = YWMY... (Base64)	source_8	Secret에 카 카오 클라이 언트 ID 저장
gift-secret.Secret.data.KAKAO_CLIENT_SECRET = Rncz... (Base64)	source_8	Secret에 카 카오 클라이 언트 Secret 저장

특징: 질문에 카카오톡 선물하기 MSA라고 하니, gift-app에 해당하는 배포 명세 파일만 작성해  
줌. 모든 서비스에 대한 배포 파일을 작성하게 하려면, 해당 MSA에 대한 모든 서비스를 명시해  
주는 것이 좋아 보임. 배포에는 실패함. (CreateContainerConfigError) - ConfigMap/Secret 리소  
스는 최상위에 spec이 존재하지 않지만 최상위 필드에 spec을 넣어서 생성해버림. -> 환경변수  
참조 실패 -> 리소스 형식 혼동: Deployment/Service처럼 대부분이 spec을 쓰니, 템플릿 복사/  
수정하다가 그대로 붙여넣음. 생성 자동화/LLM 산출물: 자동 생성 프롬프트가 "모든 리소스에  
spec을 둔다"는 잘못된 일반화를 학습/적용.