

```
cat <<EOF > Dockerfile
FROM centos:7
ADD jdk-8u212-linux-x64.tar.gz /opt/
ENV JAVA_HOME=/opt/jdk1.8.0_212 \
    PATH=$PATH:/opt/jdk1.8.0_212/bin:/usr/tomcat/bin
ENTRYPOINT bash
EOF
```

```
docker build -t centos7-jdk8:v1 .
```

```
cat <<EOF > Dockerfile
FROM centos7-jdk8:v1
COPY sp05-eureka-0.0.1-SNAPSHOT.jar /opt/
ENTRYPOINT ["java", "-jar", "/opt/sp05-eureka-0.0.1-SNAPSHOT.jar"]
CMD ["--spring.profiles.active=eureka1"]
EOF
```

```
docker build -t sp-eureka:v1 .
```

```
cat <<EOF > Dockerfile
FROM centos7-jdk8:v1
COPY sp12-config-0.0.1-SNAPSHOT.jar /opt/
ENTRYPOINT ["java", "-jar", "/opt/sp12-config-0.0.1-SNAPSHOT.jar"]
EOF
```

```
docker build -t sp-config:v1 .
```

```
cat <<EOF > Dockerfile
FROM centos7-jdk8:v1
COPY sp02-itemservice-0.0.1-SNAPSHOT.jar /opt/
ENTRYPOINT ["java", "-jar", "/opt/sp02-itemservice-0.0.1-SNAPSHOT.jar"]
EOF
```

```
docker build -t sp-item:v1 .
```

```
cat <<EOF > eureka1-svc.yml
apiVersion: v1
kind: Service
metadata:
  name: eureka1
spec:
  type: NodePort          # 在每个节点上开放访问端口
  ports:
    - port: 2001          # 集群内部访问该服务的端口
```

```

    targetPort: 2001      # 容器的端口
    nodePort: 30123      # 外部访问端口
  selector:
    app: eureka1
EOF

cat <<EOF > eureka2-svc.yml
apiVersion: v1
kind: Service
metadata:
  name: eureka2
spec:
  type: NodePort          # 在每个节点上开放访问端口
  ports:
    - port: 2002          # 集群内部访问该服务的端口
      targetPort: 2002    # 容器的端口
      nodePort: 30124     # 外部访问端口
  selector:
    app: eureka2
EOF

cat <<EOF > eureka1.yml
apiVersion: apps/v1      # RS 是 apps/v1中提供的资源类型
kind: ReplicaSet         # 资源类型
metadata:
  name: eureka1          # RS 命名为 kubia
spec:
  replicas: 1             # pod 副本数量
  selector:
    matchLabels:          # 使用 label 选择器
      app: eureka1        # 选取标签是 "app=kubia" 的pod
  template:
    metadata:
      labels:
        app: eureka1      # 为创建的pod添加标签 "app=kubia"
    spec:
      containers:
        - name: eureka1    # 容器名
          image: eureka:v1 # 镜像
          ports:
            - containerPort: 2001 # 容器暴露的端口
              protocol: TCP
EOF

cat <<EOF > eureka2.yml
apiVersion: apps/v1      # RS 是 apps/v1中提供的资源类型
kind: ReplicaSet         # 资源类型
metadata:
  name: eureka2          # RS 命名为 kubia
spec:

```

```

replicas: 1                # pod 副本数量
selector:
  matchLabels:             # 使用 Label 选择器
    app: eureka2           # 选取标签是 "app=kubia" 的pod
template:
  metadata:
    labels:
      app: eureka2         # 为创建的pod添加标签 "app=kubia"
  spec:
    containers:
      - name: eureka2      # 容器名
        image: eureka:v1   # 镜像
        args: ["--spring.profiles.active=eureka2"]
EOF

```

```

cat <<EOF > config.yml
apiVersion: apps/v1        # RS 是 apps/v1中提供的资源类型
kind: ReplicaSet           # 资源类型
metadata:
  name: config             # RS 命名为 kubia
spec:
  replicas: 1              # pod 副本数量
  selector:
    matchLabels:           # 使用 Label 选择器
      app: config          # 选取标签是 "app=kubia" 的pod
  template:
    metadata:
      labels:
        app: config        # 为创建的pod添加标签 "app=kubia"
    spec:
      containers:
        - name: config     # 容器名
          image: sp-config:v1 # 镜像
EOF

```

```

cat <<EOF > item1.yml
apiVersion: apps/v1        # RS 是 apps/v1中提供的资源类型
kind: ReplicaSet           # 资源类型
metadata:
  name: item1              # RS 命名为 kubia
spec:
  replicas: 1              # pod 副本数量
  selector:
    matchLabels:           # 使用 Label 选择器
      app: item1           # 选取标签是 "app=kubia" 的pod
  template:
    metadata:
      labels:
        app: item1         # 为创建的pod添加标签 "app=kubia"
    spec:
      containers:
        - name: item1      # 容器名
          image: sp-item:v1 # 镜像
          args: ["--spring.profiles.active=item1"]

```

EOF

```
cat <<EOF > item2.yml
apiVersion: apps/v1          # RS 是 apps/v1中提供的资源类型
kind: ReplicaSet             # 资源类型
metadata:
  name: item2                 # RS 命名为 kubia
spec:
  replicas: 1                 # pod 副本数量
  selector:
    matchLabels:              # 使用 label 选择器
      app: item2              # 选取标签是 "app=kubia" 的pod
  template:
    metadata:
      labels:
        app: item2            # 为创建的pod添加标签 "app=kubia"
    spec:
      containers:
        - name: item2          # 容器名
          image: sp-item:v1     # 镜像
          args: ["--spring.profiles.active=item2"]
EOF
```