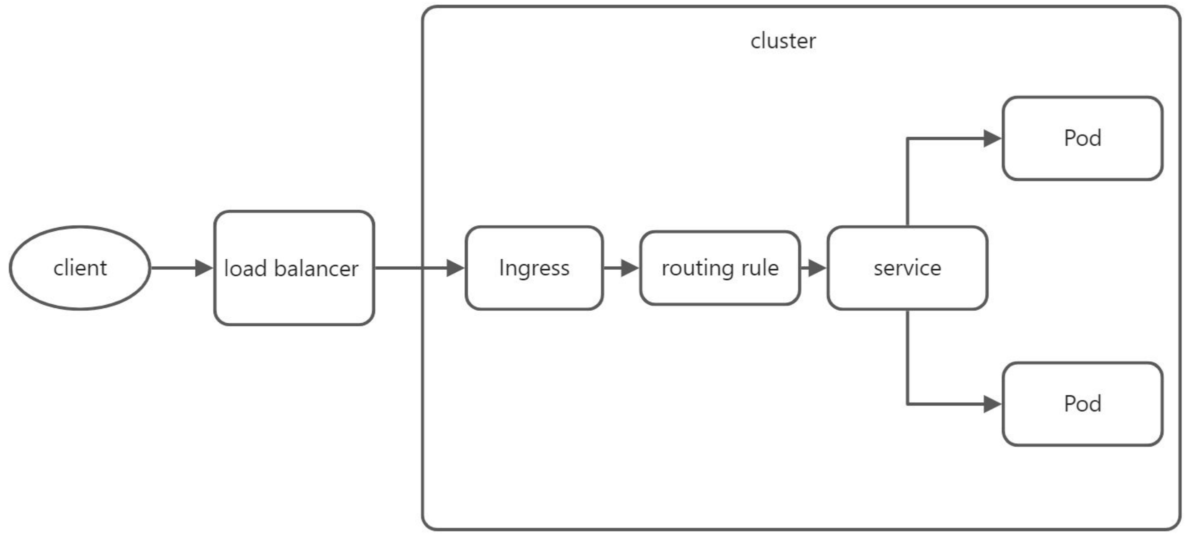
是什么

Ingress 公开从集群外部到集群内服务的 HTTP 和 HTTPS 路由。 流量路由由 Ingress 资源上定义的规则控制。简而言之，Ingress为一组路由规则。下图为简单的ingress 示意图：



Ingress为路由规则，那么在集群中使用Ingress 则必须在集群中安装Ingress controller。Ingress controller 负责通过负载均衡来实现Ingress 规则。

相关概念

1. ingress：k8s中的一种资源对象，用于定义软件层面的路由规则
2. ingress controller：一组pod负责解析ingress路由规则，根据规则将请求转发到对应的service
3. ingressClass：每一个ingress都需要指定一个class，用于表示采取哪个ingress controller解析规则
4. Kubernetes 作为一个项目，目前支持和维护 AWS、 GCE 和 Nginx Ingress 控制器

nginx ingress controller

文档

部署文档：https://kubernetes.github.io/ingress-nginx/deploy/
源码：https://github.com/kubernetes/ingress-nginx/tree/controller-v1.5.1

安装

下载部署配置文件

https://github.com/kubernetes/ingress-nginx/blob/controller-v1.5.1/deploy/static/provider/cloud/deploy.yaml

修改配置文件镜像地址

```
# 修改 ingress-nginx-controller镜像地址

image: registry.k8s.io/ingress-
nginx/controller:v1.5.1@sha256:4ba73c697770664c1e00e9f968de14e08f606ff961c76e5d7
033a4a9c593c629
# 修改为阿里云的镜像地址
image: registry.aliyuncs.com/google_containers/nginx-ingress-controller:v1.5.1

image: registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20220916-
gd32f8c343@sha256:39c5b2e3310dc4264d638ad28d9d1d96c4cbb2b2dcfb52368fe4e3c63f61e1
0f
# 修改为阿里云的镜像地址
image: registry.aliyuncs.com/google_containers/kube-webhook-certgen:v20220916-
gd32f8c343
```

修改过后的配置文件

```
apiVersion: v1
kind: Namespace
metadata:
  labels:
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
  name: ingress-nginx
---
apiVersion: v1
automountServiceAccountToken: true
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx
  namespace: ingress-nginx
---
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-admission
  namespace: ingress-nginx
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  labels:
```

```yaml
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx
  namespace: ingress-nginx
rules:
- apiGroups:
  - ""
  resources:
  - namespaces
  verbs:
  - get
- apiGroups:
  - ""
  resources:
  - configmaps
  - pods
  - secrets
  - endpoints
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - services
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses/status
  verbs:
  - update
- apiGroups:
  - networking.k8s.io
  resources:
  - ingressclasses
  verbs:
  - get
  - list
  - watch
```

```yaml
- apiGroups:
  - ""
  resourceNames:
  - ingress-nginx-leader
  resources:
  - configmaps
  verbs:
  - get
  - update
- apiGroups:
  - ""
  resources:
  - configmaps
  verbs:
  - create
- apiGroups:
  - coordination.k8s.io
  resourceNames:
  - ingress-nginx-leader
  resources:
  - leases
  verbs:
  - get
  - update
- apiGroups:
  - coordination.k8s.io
  resources:
  - leases
  verbs:
  - create
- apiGroups:
  - ""
  resources:
  - events
  verbs:
  - create
  - patch
- apiGroups:
  - discovery.k8s.io
  resources:
  - endpointslices
  verbs:
  - list
  - watch
  - get
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
```

```yaml
  name: ingress-nginx-admission
  namespace: ingress-nginx
rules:
- apiGroups:
  - ""
  resources:
  - secrets
  verbs:
  - get
  - create
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx
rules:
- apiGroups:
  - ""
  resources:
  - configmaps
  - endpoints
  - nodes
  - pods
  - secrets
  - namespaces
  verbs:
  - list
  - watch
- apiGroups:
  - coordination.k8s.io
  resources:
  - leases
  verbs:
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - nodes
  verbs:
  - get
- apiGroups:
  - ""
  resources:
  - services
  verbs:
  - get
  - list
  - watch
- apiGroups:
```

```
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - events
  verbs:
  - create
  - patch
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses/status
  verbs:
  - update
- apiGroups:
  - networking.k8s.io
  resources:
  - ingressclasses
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - discovery.k8s.io
  resources:
  - endpointslices
  verbs:
  - list
  - watch
  - get
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-admission
rules:
- apiGroups:
  - admissionregistration.k8s.io
  resources:
  - validatingwebhookconfigurations
  verbs:
  - get
  - update
```

```yaml
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx
  namespace: ingress-nginx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: ingress-nginx
subjects:
- kind: ServiceAccount
  name: ingress-nginx
  namespace: ingress-nginx
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-admission
  namespace: ingress-nginx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: ingress-nginx-admission
subjects:
- kind: ServiceAccount
  name: ingress-nginx-admission
  namespace: ingress-nginx
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  labels:
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ingress-nginx
subjects:
```

```yaml
- kind: ServiceAccount
  name: ingress-nginx
  namespace: ingress-nginx
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-admission
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ingress-nginx-admission
subjects:
- kind: ServiceAccount
  name: ingress-nginx-admission
  namespace: ingress-nginx
---
apiVersion: v1
data:
  allow-snippet-annotations: "true"
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-controller
  namespace: ingress-nginx
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-controller
  namespace: ingress-nginx
spec:
  externalTrafficPolicy: Local
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - appProtocol: http
```

```yaml
    name: http
    port: 80
    protocol: TCP
    targetPort: http
  - appProtocol: https
    name: https
    port: 443
    protocol: TCP
    targetPort: https
  selector:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
  type: LoadBalancer
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-controller-admission
  namespace: ingress-nginx
spec:
  ports:
  - appProtocol: https
    name: https-webhook
    port: 443
    targetPort: webhook
  selector:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
  type: ClusterIP
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-controller
  namespace: ingress-nginx
spec:
  minReadySeconds: 0
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app.kubernetes.io/component: controller
```

```
      app.kubernetes.io/instance: ingress-nginx
      app.kubernetes.io/name: ingress-nginx
  template:
    metadata:
      labels:
        app.kubernetes.io/component: controller
        app.kubernetes.io/instance: ingress-nginx
        app.kubernetes.io/name: ingress-nginx
    spec:
      containers:
      - args:
        - /nginx-ingress-controller
        - --publish-service=$(POD_NAMESPACE)/ingress-nginx-controller
        - --election-id=ingress-nginx-leader
        - --controller-class=k8s.io/ingress-nginx
        - --ingress-class=nginx
        - --configmap=$(POD_NAMESPACE)/ingress-nginx-controller
        - --validating-webhook=:8443
        - --validating-webhook-certificate=/usr/local/certificates/cert
        - --validating-webhook-key=/usr/local/certificates/key
        env:
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: LD_PRELOAD
          value: /usr/local/lib/libmimalloc.so
        # image: registry.k8s.io/ingress-
nginx/controller:v1.5.1@sha256:4ba73c697770664c1e00e9f968de14e08f606ff961c76e5d7
033a4a9c593c629
        image: registry.aliyuncs.com/google_containers/nginx-ingress-
controller:v1.5.1
        imagePullPolicy: IfNotPresent
        lifecycle:
          preStop:
            exec:
              command:
              - /wait-shutdown
        livenessProbe:
          failureThreshold: 5
          httpGet:
            path: /healthz
            port: 10254
            scheme: HTTP
          initialDelaySeconds: 10
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 1
        name: controller
        ports:
        - containerPort: 80
```

```yaml
          name: http
          protocol: TCP
        - containerPort: 443
          name: https
          protocol: TCP
        - containerPort: 8443
          name: webhook
          protocol: TCP
        readinessProbe:
          failureThreshold: 3
          httpGet:
            path: /healthz
            port: 10254
            scheme: HTTP
          initialDelaySeconds: 10
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 1
        resources:
          requests:
            cpu: 100m
            memory: 90Mi
        securityContext:
          allowPrivilegeEscalation: true
          capabilities:
            add:
            - NET_BIND_SERVICE
            drop:
            - ALL
          runAsUser: 101
        volumeMounts:
        - mountPath: /usr/local/certificates/
          name: webhook-cert
          readOnly: true
      dnsPolicy: ClusterFirst
      nodeSelector:
        kubernetes.io/os: linux
      serviceAccountName: ingress-nginx
      terminationGracePeriodSeconds: 300
      volumes:
      - name: webhook-cert
        secret:
          secretName: ingress-nginx-admission
---
apiVersion: batch/v1
kind: Job
metadata:
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-admission-create
  namespace: ingress-nginx
```

```
spec:
  template:
    metadata:
      labels:
        app.kubernetes.io/component: admission-webhook
        app.kubernetes.io/instance: ingress-nginx
        app.kubernetes.io/name: ingress-nginx
        app.kubernetes.io/part-of: ingress-nginx
        app.kubernetes.io/version: 1.5.1
      name: ingress-nginx-admission-create
    spec:
      containers:
      - args:
        - create
        - --host=ingress-nginx-controller-admission,ingress-nginx-controller-
admission.$(POD_NAMESPACE).svc
        - --namespace=$(POD_NAMESPACE)
        - --secret-name=ingress-nginx-admission
        env:
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        #image: registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20220916-
gd32f8c343@sha256:39c5b2e3310dc4264d638ad28d9d1d96c4cbb2b2dcfb52368fe4e3c63f61e1
0f
        image: registry.aliyuncs.com/google_containers/kube-webhook-
certgen:v20220916-gd32f8c343
        imagePullPolicy: IfNotPresent
        name: create
        securityContext:
          allowPrivilegeEscalation: false
      nodeSelector:
        kubernetes.io/os: linux
      restartPolicy: OnFailure
      securityContext:
        fsGroup: 2000
        runAsNonRoot: true
        runAsUser: 2000
      serviceAccountName: ingress-nginx-admission
---
apiVersion: batch/v1
kind: Job
metadata:
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-admission-patch
  namespace: ingress-nginx
spec:
  template:
    metadata:
```

```yaml
      labels:
        app.kubernetes.io/component: admission-webhook
        app.kubernetes.io/instance: ingress-nginx
        app.kubernetes.io/name: ingress-nginx
        app.kubernetes.io/part-of: ingress-nginx
        app.kubernetes.io/version: 1.5.1
      name: ingress-nginx-admission-patch
    spec:
      containers:
      - args:
        - patch
        - --webhook-name=ingress-nginx-admission
        - --namespace=$(POD_NAMESPACE)
        - --patch-mutating=false
        - --secret-name=ingress-nginx-admission
        - --patch-failure-policy=Fail
        env:
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        #image: registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20220916-
gd32f8c343@sha256:39c5b2e3310dc4264d638ad28d9d1d96c4cbb2b2dcfb52368fe4e3c63f61e1
0f
        image: registry.aliyuncs.com/google_containers/kube-webhook-
certgen:v20220916-gd32f8c343
        imagePullPolicy: IfNotPresent
        name: patch
        securityContext:
          allowPrivilegeEscalation: false
      nodeSelector:
        kubernetes.io/os: linux
      restartPolicy: OnFailure
      securityContext:
        fsGroup: 2000
        runAsNonRoot: true
        runAsUser: 2000
      serviceAccountName: ingress-nginx-admission
---
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: nginx
spec:
  controller: k8s.io/ingress-nginx
---
apiVersion: admissionregistration.k8s.io/v1
kind: ValidatingWebhookConfiguration
metadata:
```

```
  labels:
    app.kubernetes.io/component: admission-webhook
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
    app.kubernetes.io/version: 1.5.1
  name: ingress-nginx-admission
webhooks:
- admissionReviewVersions:
  - v1
  clientConfig:
    service:
      name: ingress-nginx-controller-admission
      namespace: ingress-nginx
      path: /networking/v1/ingresses
  failurePolicy: Fail
  matchPolicy: Equivalent
  name: validate.nginx.ingress.kubernetes.io
  rules:
  - apiGroups:
    - networking.k8s.io
    apiVersions:
    - v1
    operations:
    - CREATE
    - UPDATE
    resources:
    - ingresses
  sideEffects: None
```

1. 应用配置

```
kubectl apply -f deploy.yaml
```

设置默认ingress controller

```
# 编辑nginx ingress controller 的 ingressclass
kubectl edit -n ingress-nginx ingressclasses.networking.k8s.io nginx

# 在annotations字段下，添加注解 ingressclass.kubernetes.io/is-default-class: "true"
annotations:
    ingressclass.kubernetes.io/is-default-class: "true"
```

查看ingress-nginx下的资源

```
kubectl get  -f deploy.yaml
```

```
NAME                                                    CREATED AT
role.rbac.authorization.k8s.io/ingress-nginx            2022-12-14T12:59:43Z
role.rbac.authorization.k8s.io/ingress-nginx-admission  2022-12-14T12:59:43Z

NAME                                                           CREATED AT
clusterrole.rbac.authorization.k8s.io/ingress-nginx            2022-12-14T12:59:43Z
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission  2022-12-14T12:59:43Z

NAME                                                           ROLE                          AGE
rolebinding.rbac.authorization.k8s.io/ingress-nginx            Role/ingress-nginx            104s
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission  Role/ingress-nginx-admission  104s

NAME                                                                  ROLE                                 AGE
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx            ClusterRole/ingress-nginx            104s
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission  ClusterRole/ingress-nginx-admission  104s

NAME                                     DATA   AGE
configmap/ingress-nginx-controller       1      104s

NAME                                          TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                        AGE
service/ingress-nginx-controller              LoadBalancer   10.98.249.2     <pending>     80:32137/TCP,443:31259/TCP     104s
service/ingress-nginx-controller-admission    ClusterIP      10.103.103.64   <none>        443/TCP                        104s

NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ingress-nginx-controller      1/1     1            1           104s

NAME                                          COMPLETIONS   DURATION   AGE
job.batch/ingress-nginx-admission-create      1/1           17s        104s
job.batch/ingress-nginx-admission-patch       1/1           18s        103s

NAME                                      CONTROLLER             PARAMETERS   AGE
ingressclass.networking.k8s.io/nginx      k8s.io/ingress-nginx   <none>       104s

NAME                                                                                    WEBHOOKS   AGE
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission    1          104s
```

Ingress的使用

准备三组service

myservice1

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysvc1-deployment
  labels:
    name: mysvc1-app
spec:
  replicas: 2
  selector:
    matchLabels:
      name: mysvc1-app
  template:
    metadata:
      labels:
        name: mysvc1-app
    spec:
      containers:
      - name: myhello
        image: xlhmzch/hello:1.0.2
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 80
        env:     # 注入到容器的环境变量
        - name: env1
          value: "myservice1"
```

```
apiVersion: v1
kind: Service
metadata:
  name: mysvc1
```

```
  labels:
    name: mysvc1-app
spec:
  ports:
  - port: 8081
    protocol: TCP
    targetPort: 80
    name: http
  selector:
    name: mysvc1-app
```

```
kubectl create -f mysvc1-deployment.yaml
kubectl create -f mysvc1.yaml
kubectl get -f mysvc1.yaml
# 访问service
curl http://<serviceIP>:8081/print/env
```

myservice2

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysvc2-deployment
  labels:
    name: mysvc2-app
spec:
  replicas: 2
  selector:
    matchLabels:
      name: mysvc2-app
  template:
    metadata:
      labels:
        name: mysvc2-app
    spec:
      containers:
      - name: myhello
        image: xlhmzch/hello:1.0.2
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 80
        env:    # 注入到容器的环境变量
        - name: env1
          value: "myservice2"
```

```
apiVersion: v1
kind: Service
metadata:
  name: mysvc2
  labels:
    name: mysvc2-app
spec:
  ports:
```

```
  - port: 8082
    protocol: TCP
    targetPort: 80
    name: http
  selector:
    name: mysvc2-app
```

```
kubectl create -f mysvc2-deployment.yaml
kubectl create -f mysvc2.yaml
kubectl get -f mysvc2.yaml
# 访问service
curl http://<serviceIP>:8082/print/env
```

myservice3

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysvc3-deployment
  labels:
    name: mysvc3-app
spec:
  replicas: 2
  selector:
    matchLabels:
      name: mysvc3-app
  template:
    metadata:
      labels:
        name: mysvc3-app
    spec:
      containers:
      - name: myhello
        image: xlhmzch/hello:1.0.2
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 80
        env:    # 注入到容器的环境变量
        - name: env1
          value: "myservice3"
```

```
apiVersion: v1
kind: Service
metadata:
  name: mysvc3
  labels:
    name: mysvc3-app
spec:
  ports:
  - port: 8083
    protocol: TCP
    targetPort: 80
    name: http
```

```
  selector:
    name: mysvc3-app
```

```
kubectl create -f mysvc3-deployment.yaml
kubectl create -f mysvc3.yaml
kubectl get -f mysvc3.yaml
# 访问service
curl http://<serviceIP>:8083/print/env
```

暴露单个service

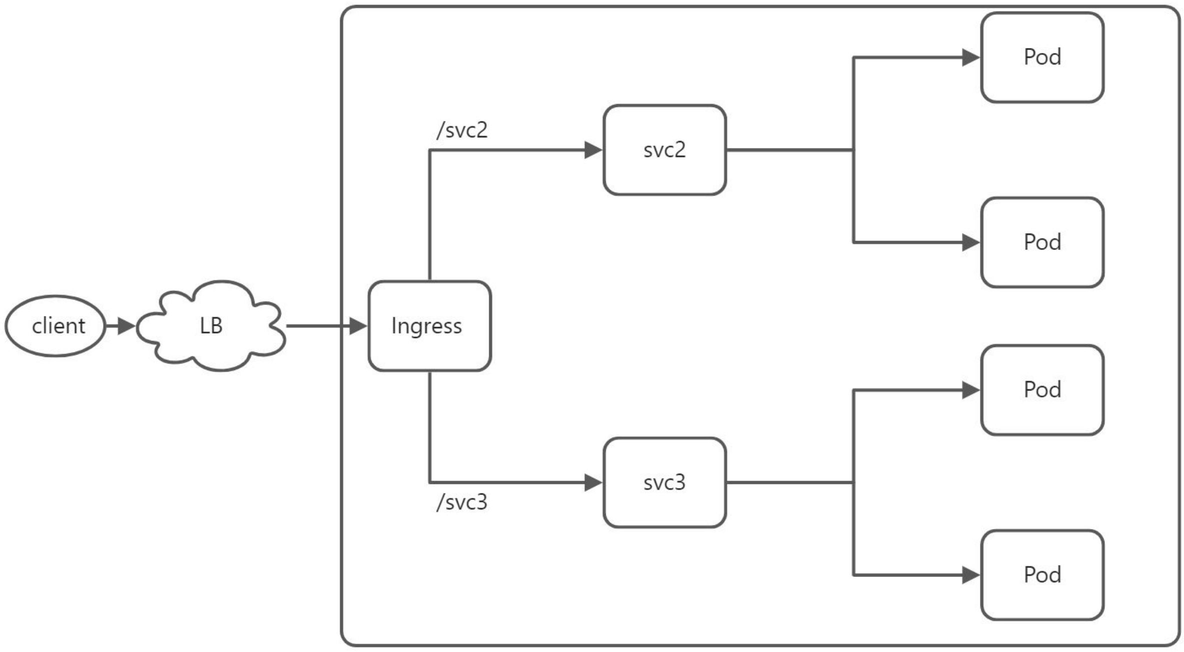
```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: default-backend-ingress
spec:
  defaultBackend:
    service:
      name: mysvc1
      port:
        number: 8081
```

```
kubectl create -f defaultBackend-ingress.yaml
kubectl get -f defaultBackend-ingress.yaml

# 访问
curl http://<ADDRESS>/print/env
```

Ingress 扇出

一个扇出（fanout）配置根据请求的 HTTP URI 将来自同一 IP 地址的流量路由到多个 Service。 Ingress 允许将负载均衡器的数量降至最低。如下图：




```
apiVersion: networking.k8s.io/v1
kind: Ingress
```

```
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: "/$1"
  name: fanout-ingress
spec:
  rules:
  - host: www.nick.com
    http:
      paths:
      - path: /svc2/(.*)$
        pathType: Prefix
        backend:
          service:
            name: mysvc2
            port:
              number: 8082
      - path: /svc3/(.*)$
        pathType: Prefix
        backend:
          service:
            name: mysvc3
            port:
              number: 8083
```

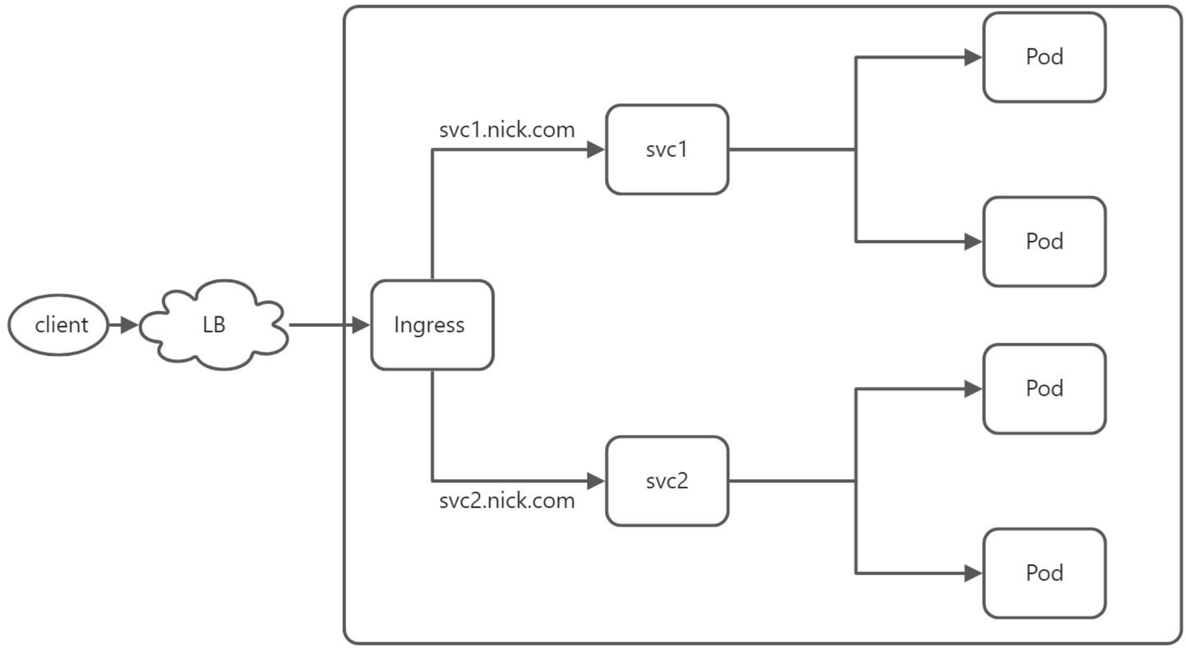
```
kubectl create -f fanout-ingress.yaml
kubectl get -f fanout-ingress.yaml
kubectl describe  -f fanout-ingress.yaml
```

```
# 修改/etc/hosts,增加 <ADDRESS> www.nick.com 映射
# ADDRESS 为分配的负载均衡IP地址或ingress controller service集群IP
192.168.1.240 www.nick.com

# 访问
curl http://www.nick.com/print/env
```

Ingress基于名称虚拟托管

基于名称的虚拟主机支持将针对多个主机名的 HTTP 流量路由到同一 IP 地址上。如下图：




```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: hostname-ingress
spec:
  rules:
  - host: svc1.nick.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: mysvc1
            port:
              number: 8081
  - host: svc2.nick.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: mysvc2
            port:
              number: 8082
```

```
kubectl create -f hostname-ingress.yaml
kubectl get -f hostname-ingress.yaml
kubectl describe  -f hostname-ingress.yaml
```

```
# 修改/etc/hosts,增加 <ADDRESS> svc1.nick.com 映射,增加 <ADDRESS> svc2.nick.com 映射
# ADDRESS 为分配的负载均衡IP地址 或ingress controller service集群IP
192.168.1.240 svc1.nick.com
192.168.1.240 svc2.nick.com

# 访问
curl http://svc1.nick.com/print/env
curl http://svc2.nick.com/print/env
```

Ingress TLS

注意事项:

1. 自签证书中的CN字段必须为全限定域名，例如：https-www.nick.com
2. 自签证书中的CN字段必须与ingress定义中rules.host匹配
3. ingress定义中tls.hosts 字段的值必须与rules.host完全匹配
4. TLS Secret 的数据中必须包含用于 TLS 的以键名 tls.crt 保存的证书和以键名 tls.key 保存的私钥，即 tls.crt为键值保存证书，tls.key为键值保存私钥

自签证书

```
# 创建公钥和相对应的私钥
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /tmp/fanouttls.key -out /tmp/fanouttls.crt -subj "/CN=https-www.nick.com/O=https-www.nick.com"
```

新建Secret

```
# 创建secret
kubectl create secret tls fanout-ingress-tls --cert=/tmp/fanouttls.crt --key=/tmp/fanouttls.key
# 查看secret
kubectl get secrets fanout-ingress-tls -o yaml
```

新建ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: "/$1"
  name: fanout-tls-ingress
spec:
  tls:
  - hosts:
      - https-www.nick.com
    secretName: fanout-ingress-tls
  rules:
  - host: https-www.nick.com
    http:
      paths:
      - path: /svc2/(.*)$
```

```yaml
        pathType: Prefix
        backend:
          service:
            name: mysvc2
            port:
              number: 8082
      - path: /svc3/(.*)$
        pathType: Prefix
        backend:
          service:
            name: mysvc3
            port:
              number: 8083
```

```
kubectl create -f  fanout-tls-ingress.yaml
kubectl get -f  fanout-tls-ingress.yaml
kubectl describe -f  fanout-tls-ingress.yaml
```

```
# 修改/etc/hosts,增加 <ADDRESS> https-www.nick.com 映射
# ADDRESS 为分配的负载均衡IP地址 或ingress controller service集群IP
192.168.1.240 https-www.nick.com

# 访问
curl  https://https-www.nick.com/svc2/print/env --cacert /tmp/fanouttls.crt
curl  https://https-www.nick.com/svc3/print/env --cacert /tmp/fanouttls.crt
```