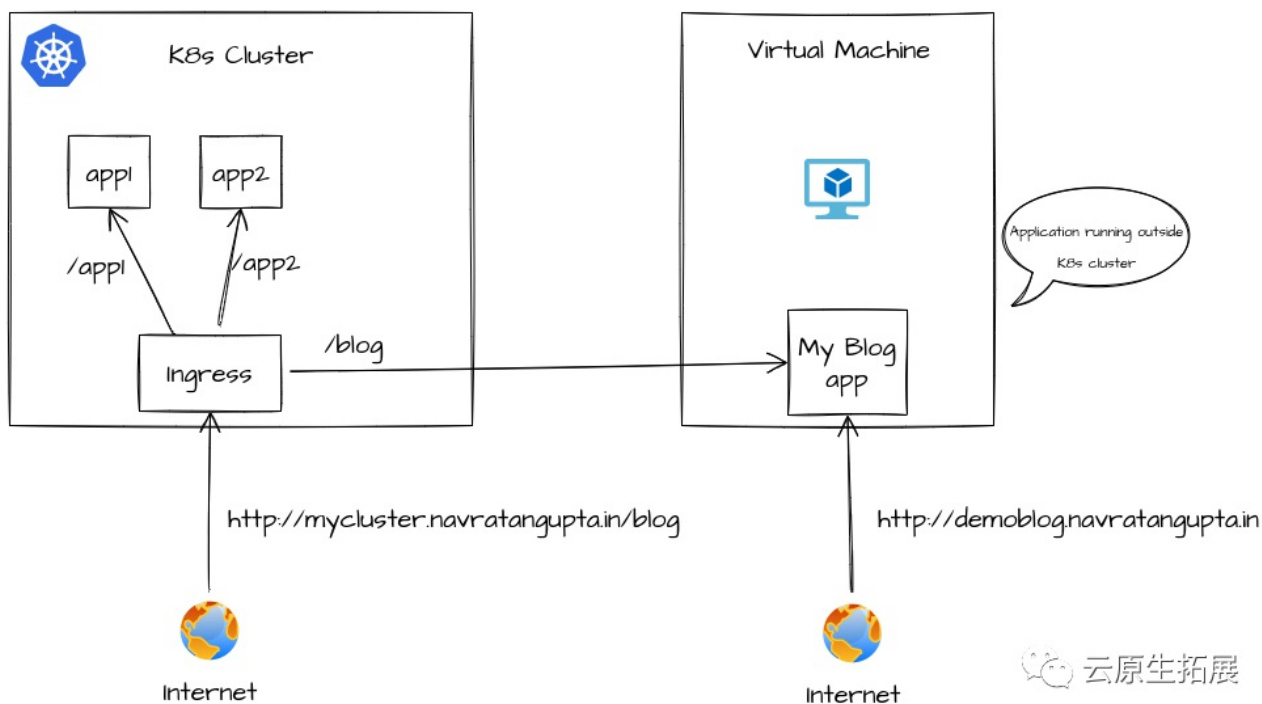


## Kubernetes 系列（十六）Ingress 代理集群外服务

让我们考虑一下，我们有一个运行在独立虚拟机上的应用程序。我们还有一个运行了一些应用程序的 Kubernetes 集群，他们可以通过 ingress 访问。我们希望使用访问 Kubernetes 集群上运行的所有应用程序相同的域名来访问运行在独立 VM (K8s 集群之外) 上的应用程序，而不需要将应用程序移动到 K8s 集群。

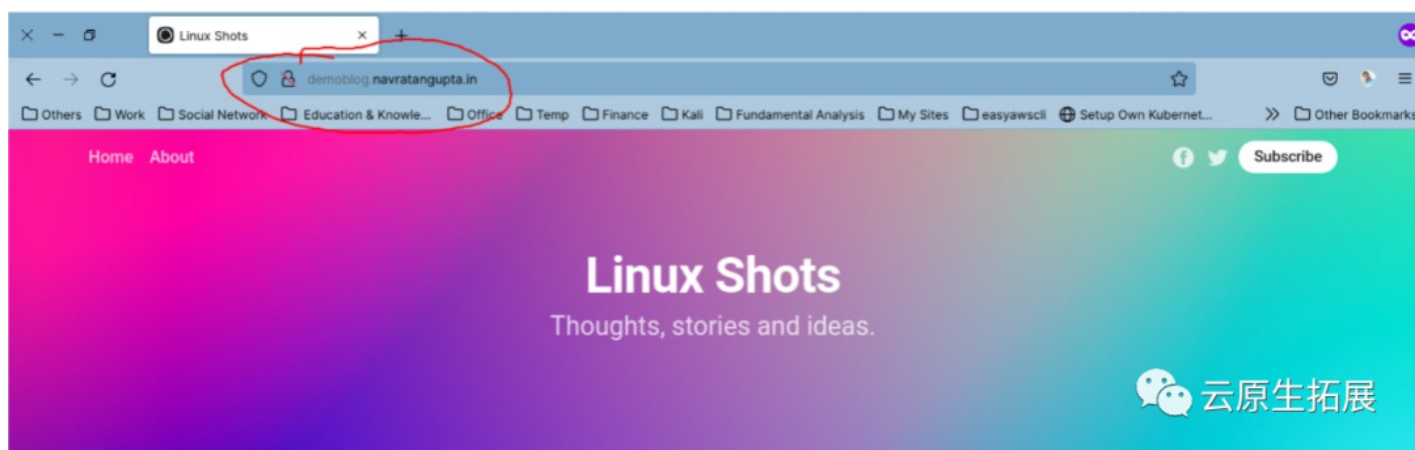
本文将演示如何将 K8s 集群外运行的应用程序置于在 K8s 集群上运行的 ingress 之后。因此，运行在 K8s 集群之外的应用程序可以使用与我们访问运行在 K8s 集群上的应用程序相同的域名进行访问。

基本概念如下图：



在我们开始之前，假设我们已经熟悉 Kubernetes 的概念和资源。总的来说，Kubernetes 中的 Ingress 是一个资源，它可以帮助您将应用程序放在反向代理之后，并使我们能够从互联网访问应用程序。K8s 中的 Service 是一种帮助我们访问运行在 pod 中的应用程序的资源。

在这里，我有一个在虚拟机上运行的博客，可以通过 <http://demoblog.navratangupta.in> 访问。我有一个 K8s 集群，其中运行一些应用程序，可以从 [http://mycluster.navratangupta.in/\\*](http://mycluster.navratangupta.in/*) 访问。



我们将把我的博客放在 K8s 集群域后面的 path /blog 下。所以这个博客可以通过 <http://mycluster.navratangupta.in/blog> 访问。

为此，我们需要在 K8s 集群中创建 ExternalName 类型的 K8s service 和一个 Ingress。ExternalName 类型的 service 通过返回一条 CNAME 记录。

## 1. 创建 service.yml 文件:

```
apiVersion: v1
kind: Service
metadata:
  name: blog
  namespace: demo
spec:
  type: ExternalName
  externalName: demoblog.navratangupta.in
```

这里的 `.spec.externalName` 的值是那个外部虚拟机上的博客访问域名(没有额外的路径)。

将 service 部署到集群中:

```
kubectl apply -f service.yaml
```

## 2. 创建 ingress.yaml 文件:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: blog
  namespace: demo
spec:
  rules:
  - host: mycluster.navratangupta.in
    http:
      paths:
      - backend:
          service:
            name: blog
            port:
              number: 80
        path: /blog
        pathType: ImplementationSpecific
```

这里的 `.spec.rules[0].host` 具有在 K8s 上运行的应用程序可以访问的域名。

`.spec.rules[0].http.paths[0].backend.service.name` 为步骤1中创建的ExternalName类型服务名称。

`.spec.rules[0].http.paths[0].backend.service.port.number` 表示可访问虚拟机中的应用程序的端口号。在我们的例子中, 它的端口是80。

`.spec.rules[0].http.paths[0].path` 包含了可以访问我们博客的路径(或端点)。在本例中, 我们希望在/blog路径下访问博客。

如果应用程序监听端口443, 并且可以通过 HTTPS 协议上访问, 那么我们必须要在 `.metadata.annotations`下添加下面的注释

```
nginx.ingress.kubernetes.io/ backend-protocol: HTTPS
```

假设 Nginx Ingress 控制器已经部署并运行在K8s集群上。我们在集群中部署 ingress:

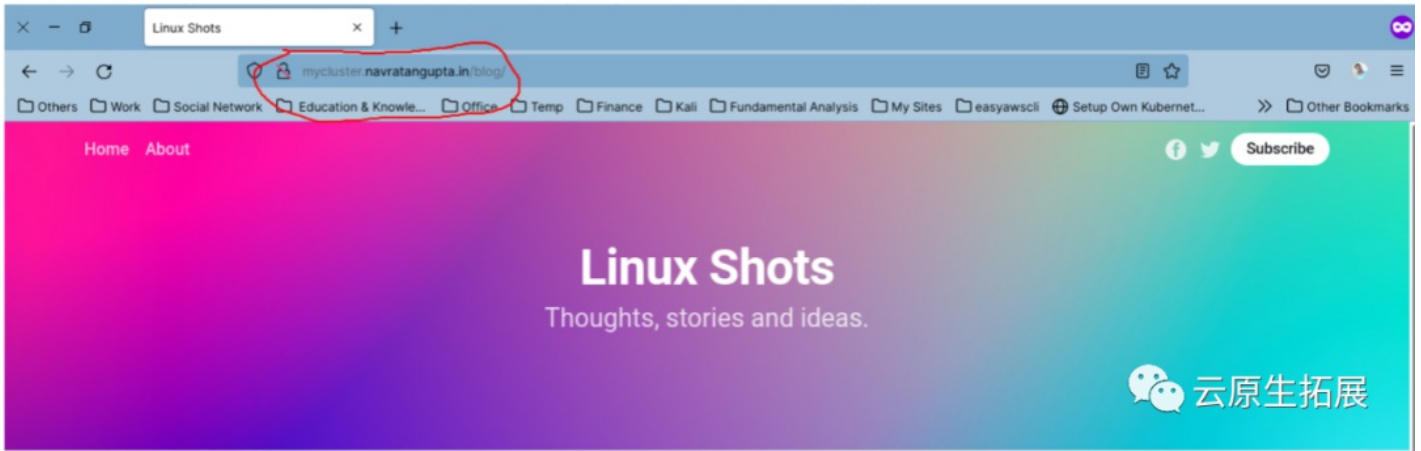
```
kubectl apply -f ingress.yaml
```

```
ubuntu ~$ kubectl -n demo get svc
NAME    TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
blog    ExternalName <none>         demoblog.navratangupta.in <none>       101m
ubuntu ~$ kubectl -n demo get ing
NAME    CLASS    HOSTS                                ADDRESS    PORTS    AGE
blog    <none>   mycluster.navratangupta.in          80        64s
ubuntu ~$
```

## 重新配置虚拟机中的应用

一旦在K8s集群上创建了 Service 和 Ingress，应用程序端就需要更改配置。这些更改可能是更改应用程序配置文件中的域名和路径。这可能是必要的，也可能不是，取决于应用程序。查看应用程序文档，了解所需的任何域和代理更改。

一旦所有配置调整到位，通过 K8s集群的 ingress 访问域名及路径进行访问。对我们来说，它将访问<http://mycluster.navratangupta.in/blog>



就是这样。我们应该能够在访问K8s应用程序的同一个域中访问在集群外部运行的应用程序。

欢迎关注我的公众号“云原生拓展”，原创技术文章第一时间推送。