

Maninderjit (Mani) Bindra

243 Followers

About

Follow



Sign in to Medium with Google

**Moshe Sharon**

moshes@gmail.com

Continue as Moshe

To create your account, Google will share your name, email address and profile picture with Medium. See Medium's [privacy policy](#) and [Terms of Service](#).

Using nginx-ingress controller to restrict access by IP (ip whitelisting) for a service deployed to a Kubernetes (AKS) cluster

**Maninderjit (Mani) Bindra** May 19, 2018 · 3 min read

While working on a project earlier this week we were given the following requirements :

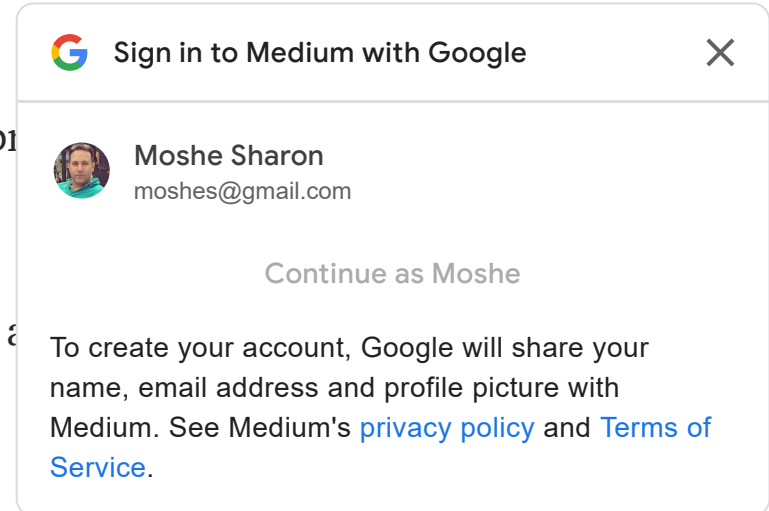
1. Create a Managed **Kubernetes** cluster (AKS) on Azure, in a existing Azure **Subnet** using an ARM template / az commands

2. For a service deployed to the AKS cluster, restrict access to certain client source IPs (**Ip Whitelisting**)

3. For a service deployed to the AKS cluster auto provision certificates. You can follow the [post](#) for this

This post details point 2 above. I will be creating a service for point 1 and 3 above, in the near future.

The rest of this post assumes that the AKS Kubernetes cluster is available, you have helm installed, and we have already executed the helm init command.



Create or update the nginx-ingress controller

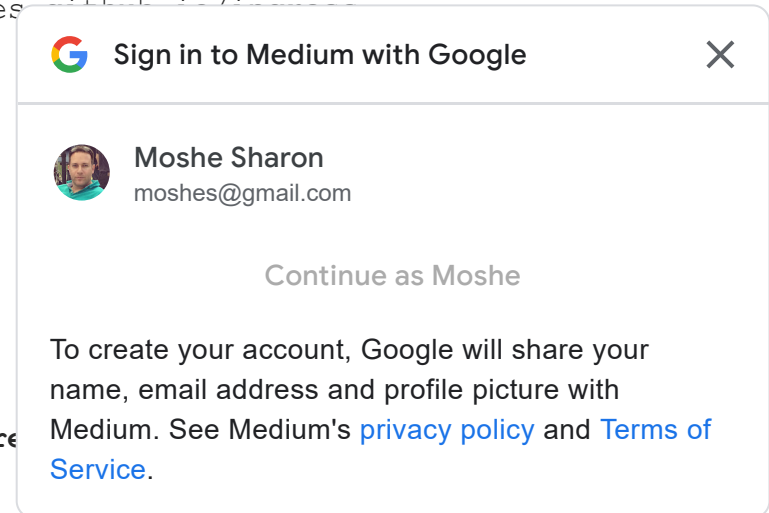
The first thing we do now is install the ingress-ingress controller using helm. The github page for the **nginx-ingress** controller **helm** chart is at [nginx-ingress](#). The install command to be used is :

Helm Repo Initialization

```
$ helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
$ helm repo update
```

Ingress controller install command

```
$ helm install ingress-nginx ingress-nginx/ingress-nginx --set controller.service.externalTrafficPolicy=Local
```



The default value of **controller.service.externalTrafficPolicy** in the nginx ingress helm chart is 'Cluster', we need to change this value to 'Local'. With the default value of 'Cluster' the ingress controller does not see the actual source ip from the client request but an internal IP. After setting this value to '**Local**' the ingress controller gets the unmodified source ip of the client request.

Before we apply the ingress rule with source ip whitelisting for a service, let us create a sample web app deployment and service:

Create the the hello world web server deployment and service to test the whitelisting

Create Deployment

```
$ kubectl run web --image=tutum/hello-world --port=80
```

Expose the deployment as a service

```
$ kubectl expose pod web --port=80
```

Check if service and pod is up

```
$ kubectl get svc web
```

NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
web	10.3.30.224	<none>	80/TCP	5m

```
$ kubectl get pod -l=run=web
```

 Sign in to Medium with Google ✕

 **Moshe Sharon**
moshes@gmail.com

Continue as Moshe

To create your account, Google will share your name, email address and profile picture with Medium. See Medium's [privacy policy](#) and [Terms of Service](#).

NAME	READY	STATUS	RESTARTS	AGE
web-5bff8ffd8c-twxwp	1/1	Running	0	

Apply the ingress IP Whitelisting rule to the service

The annotation (`nginx.ingress.kubernetes.io/whitelist-source-range`)we need to apply to the kubernetes ingress resource using nginx-ingress is detailed at [nginx-ingress](#).

We now create the ingress rule

Ingress YAML file

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: web-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
```

 Sign in to Medium with Google



Moshe Sharon
 moshes@gmail.com

Continue as Moshe

To create your account, Google will share your name, email address and profile picture with Medium. See Medium's [privacy policy](#) and [Terms of Service](#).

```

    nginx.ingress.kubernetes.io/use-regex: "true"
    nginx.ingress.kubernetes.io/rewrite-target: /$1
    nginx.ingress.kubernetes.io/whitelist-source-range: 10.0.0.0/24
# # depending on the ingress controller version
# the above may need to be modified to remove the prefix
ingress.kubernetes.io/whitelist-source-range: 49
spec:
  rules:
  - host: web.manitestdomain.com
    http:
      paths:
      - path: /(.*)
        pathType: Prefix
        backend:
          service:
            name: web
            port:
              number: 80

```



Sign in to Medium with Google

**Moshe Sharon**

moshes@gmail.com

Continue as Moshe

To create your account, Google will share your name, email address and profile picture with Medium. See Medium's [privacy policy](#) and [Terms of Service](#).

Apply the ingress rule

```
$ kubectl apply -f web-ingress-whitelist.yaml
```

Thats it whitelisting is in place.

Testing and Debugging the whitelisting rules

Before steps below were executed dns configuration the domain manitestdomain.com was pointing to the ingress controller service.

Request from whitelisted ip

```
$ curl ipinfo.io/ip  
49.36.X.X
```

```
$ curl -I web.manitestdomain.com  
HTTP/1.1 200 OK  
Server: nginx/1.13.5  
Date: Sat, 19 May 2018 06:07:01 GMT  
Content-Type: text/html; charset=UTF-8  
Connection: keep-alive  
X-Powered-By: PHP/5.6.14
```

Request from Non Whitelisted ip

```
$ curl ipinfo.io/ip  
  
223.Y.Y.Y
```

 Sign in to Medium with Google ✕



Moshe Sharon
moshes@gmail.com

Continue as Moshe

To create your account, Google will share your name, email address and profile picture with Medium. See Medium's [privacy policy](#) and [Terms of Service](#).

```
$ curl -I web.manitestdomain.com
HTTP/1.1 403 Forbidden
Server: nginx/1.13.5
Date: Sat, 19 May 2018 06:15:47 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
```

Checking nginx-ingress controller logs

```
$ kubectl get pods
NAME                                     READY
STATUS   RESTARTS   AGE
nginx-ingress-nginx-ingress-controller-586c47b885-rxm72    1/1
Running   0          1d
nginx-ingress-nginx-ingress-default-backend-65f4cd97fb-sbh7c 1/1
Running   0          1d
web-5bff8ffd8c-twxwp
```

```
$ kubectl logs nginx-ingress-nginx-ingress-controller-586c47b885-rxm72 -f
```

```
..
..
..
49.36.x.x - [49.36.x.x] - - [19/May/2018:06:11:21 +0000] "HEAD /
HTTP/1.1" 200 0 "-" "curl/7.52.1" 87 0.002 [default-web-80]
10.2.1.7:80 0 0.002 200
```



Sign in to Medium with Google



Moshe Sharon

moshes@gmail.com

Continue as Moshe

To create your account, Google will share your name, email address and profile picture with Medium. See Medium's [privacy policy](#) and [Terms of Service](#).

Thanks for reading this post. I hope you liked it. Please leave your comments and views about the same over here.

It has been pointed to me by @brunzefb in his [tweet](#) about an issue when using `externalTrafficPolicy=Local` in `nginx` along with AWS ELB. As per his request I am sharing a relevant stack overflow post :

<https://stackoverflow.com/questions/66648243/deploying-ingress-nginx-controller-elb-in-eks-cluster-with-multiple-nodes> .

Kubernetes Aks Nginx Ingress Whitelisting Ingress

 Sign in to Medium with Google ✕



Moshe Sharon
moshes@gmail.com

Continue as Moshe

To create your account, Google will share your name, email address and profile picture with Medium. See Medium's [privacy policy](#) and [Terms of Service](#).

About Write Help Legal