

Deploying application with SSL in Google Nginx

This example will use SSL using ACM. First import your SSL certificate in ACM and get the ARN.

We will try to deploy tea application here.

Below is the deployment file of tea.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: tea-dep
  namespace: ingress-nginx
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: tea
    spec:
      containers:
        - name: tea
          image: nginxdemos/hello
          ports:
            - containerPort: 80
```

Post that service file of the application.

```
apiVersion: v1
kind: Service
metadata:
  name: tea-svc
  namespace: ingress-nginx
  labels:
    app: tea
spec:
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
      name: http
  selector:
    app: tea
```

Now time for ingress.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  namespace: ingress-nginx
  name: google-nginx-aggregator
spec:
  rules:
    - host: googlenginx.awsstar.com
      http:
        paths:
          - backend:
              serviceName: tea-svc
              servicePort: 80
            path: /tea
```

Now to attach SSL with ELB we have to modify the service file of google ingress controller. And then apply it.

```
kind: Service
apiVersion: v1
metadata:
  name: ingress-nginx
  namespace: ingress-nginx
  labels:
    app: ingress-nginx
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-ssl-cert:
"arn:aws:acm:us-east-1:23xxxxxxxxx6:certificate/d001145a-b5fd-455d-b76c-
dafcc79be381"
    service.beta.kubernetes.io/aws-load-balancer-backend-protocol:
"http"
    service.beta.kubernetes.io/aws-load-balancer-ssl-ports: "https"

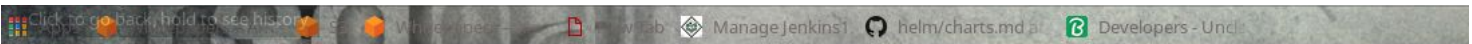
service.beta.kubernetes.io/aws-load-balancer-connection-idle-timeout:
'3600'
spec:
  type: LoadBalancer
  selector:
    app: ingress-nginx
  ports:
    - name: http
      port: 80
      targetPort: http
    - name: https
      port: 443
      targetPort: http
```

Now modify the configmap and disable proxy control.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
  namespace: ingress-nginx
  labels:
    app: ingress-nginx
data:
  use-proxy-protocol: "false"
```

Apply the files.

Now map the elb with your domain and hit it with https



NGINX

Server name:	tea-dep-3874333905-sncc5
Server address:	10.42.178.12:80
URI:	/tea
Date:	16/Dec/2017:06:43:28 +0000