ALB

- **Note: We are provisioning ALB (Application Load Balancer) with ALB controller ingress deployed in Cluster.**

- Here's the link to deploy/set up the ALB controller in the cluster.
- Installing the AWS Load Balancer Controller add-on - Amazon EKS
- Currently, we have created only 2 Load balancers for this deployment using Kubernetes ingress.
  Its currently used for Production deployment and staging deployment

- **Also when we move towards the final production deployment we'll create the 2 load balancers as per the deployment environments (Production, Staging).**
- **We have attached the WAF with ALB by using Ingress annotations.**

Setting Up External DNS:

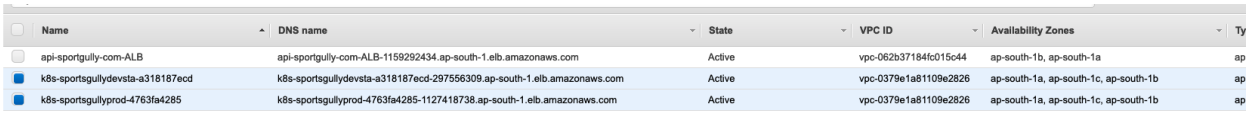
- In Order to add the entries to the Route53 hosted zone we have set up the External DNS component to our Kubernetes Cluster.
- Setup External DNS - AWS LoadBalancer Controller

Annotations Used in ALB:

```
- alb.ingress.kubernetes.io/load-balancer-name: <name-of-load-balancer>
- alb.ingress.kubernetes.io/backend-protocol-version: HTTP2
- alb.ingress.kubernetes.io/healthcheck-protocol: HTTP
- alb.ingress.kubernetes.io/healthcheck-interval-seconds: '10' (as per requirement)
- alb.ingress.kubernetes.io/healthcheck-timeout-seconds: '8' (as per requirement)
- alb.ingress.kubernetes.io/healthy-threshold-count: '2' (as per requirement)
- alb.ingress.kubernetes.io/unhealthy-threshold-count: '2' (as per requirement)
- alb.ingress.kubernetes.io/load-balancer-attributes: routing.http2.enabled=true
  #enable HTTP2
- alb.ingress.kubernetes.io/load-balancer-attributes: idle_timeout.timeout_seconds=60
  (as per requirement)
```

```
-   alb.ingress.kubernetes.io/target-group-attributes: slow_start.duration_seconds=30
    (min value is 30, rest you can set based on requirement)

-   alb.ingress.kubernetes.io/target-group-attributes:
    deregistration_delay.timeout_seconds=60-80

-   alb.ingress.kubernetes.io/wafv2-acl-arn: ARN of WAF

-   kubernetes.io/ingress.class: alb

-   alb.ingress.kubernetes.io/group.name: for grouping the ingress for different
    namespace to same alb

-   alb.ingress.kubernetes.io/scheme: internet-facing

-   alb.ingress.kubernetes.io/load-balancer-attributes: idle_timeout.timeout_seconds=600

-   alb.ingress.kubernetes.io/backend-protocol: HTTP or HTTPS ( based on our backend )

-   alb.ingress.kubernetes.io/certificate-arn:  ARN of ACM certificate we have issued

-   alb.ingress.kubernetes.io/subnets:  Subnet ID of public subnets

-   alb.ingress.kubernetes.io/listen-ports: '[{"HTTPS":443}, {"HTTP":80}]'

-   alb.ingress.kubernetes.io/actions.ssl-redirect: '{"Type": "redirect",
    "RedirectConfig": { "Protocol": "HTTPS", "Port": "443", "StatusCode": "HTTP_301"}}'

-   external-dns.alpha.kubernetes.io/hostname: for External DNS to enter the Records to
    route53 Hosted Zone

-   alb.ingress.kubernetes.io/tags:  give Tags to ALB.

-   alb.ingress.kubernetes.io/actions.response-200: >
    {"type":"fixed-response","fixedResponseConfig":{"Port":"443","Protocol":"HTTPS","Cont
    entType":"text/plain","StatusCode":"200","MessageBody":"200 error text"}}     #we can
    use this for any specific route like (wp-admin, env etc)

-   alb.ingress.kubernetes.io/actions.ssl-redirect: |-
         {
           "Type": "redirect",
           "RedirectConfig": {
             "Protocol": "HTTPS",
             "Port": "443",
             "StatusCode": "HTTP_301"
           }
         } (then we have to set http rule backend in to redirect port 80 to 443 like
    below)

-
```

```
spec:
 rules:
   - http:
       paths:
         - backend:
             serviceName: ssl-redirect
             servicePort: use-annotation
(this rule will change default rule of HTTP and redirect HTTP to HTTPS)
```



Name	DNS name	State	VPC ID	Availability Zones	Ty
api-sportgully-com-ALB	api-sportgully-com-ALB-1159292434.ap-south-1.elb.amazonaws.com	Active	vpc-062b37184fc015c44	ap-south-1b, ap-south-1a	ap
k8s-sportsgullydevsta-a318187ecd	k8s-sportsgullydevsta-a318187ecd-297556309.ap-south-1.elb.amazonaws.com	Active	vpc-0379e1a81109e2826	ap-south-1a, ap-south-1c, ap-south-1b	ap
k8s-sportsgullyprod-4763fa4285	k8s-sportsgullyprod-4763fa4285-1127418738.ap-south-1.elb.amazonaws.com	Active	vpc-0379e1a81109e2826	ap-south-1a, ap-south-1c, ap-south-1b	ap