**#Summary:**

Created hello-world java REST application and deployed to Elastic Kubernetes Service (Amazon EKS).

I used Amazon Elastic Container Service (ECS) to store docker image.

I implemented login API as well.

Users need to provide username password to access the endpoint.

I also implemented logout REST API using POST method.

Application logs messages for each REST call.

Application can be accessed at endpoint: [**http://35.174.60.226:30443/hello**](http://35.174.60.226:30443/hello)

Username: user

Password : password

**#Implementation**

details:

-----------------------

**##**

**Application:**

1) I developed

**## Create Infrastructure:**

1) I created EKS cluster in my AWS account.

2) I created worker nodes using cloud formation template and added them to the EKS cluster by configuration mapping of AWS authentication

3) I downloaded docker to my laptop and built docker image for the application jar file.

4) I used Amazon Elastic Container Registry(ECR) to host docker images.

I created a repository and authenticated docker to it. (https://docs.aws.amazon.com/AmazonECR/latest/userguide/Registries.html).

\*Below commands are copied from AWS ECR console.

```

aws ecr get-login --region us-east-1 --no-include-email

```

**## Build docker image**

1)

I built image locally and pushed to the ECR repository using below commands.

\*Below commands are copied from AWS ECR console.

```

docker build -t project .

docker tag project:latest 927286686937.dkr.ecr.us-east-1.amazonaws.com/project:latest

docker push 927286686937.dkr.ecr.us-east-1.amazonaws.com/project:latest

```

**##Deploy docker image:**

1) Installed kubectl on my laptop.

2) I Downloaded kubernetes config using command:

```

aws eks update-kubeconfig --name final

``’

* We can create resources using cloud formation templates, I created a stack using official aws eks documents or we can also create it manually following below steps

3) Created pod resource file (project.yml).

4) I created kubernetes resources.

```

kubectl create -f project.yml

```

**## Enable access for the endpoint to public.**

1) By going to EC2 console, enabled access for endpoint to public by updating inbound rules in security groups.

**#Testing:**

1) Try accessing without username and password

```

curl http://35.174.60.226:30443/hello

```

Output: {"timestamp":"\*\*\*","status":401,"error":"Unauthorized","message":"Unauthorized","path":"/hello"}

2) Try accessing with username and password

```

curl http://35.174.60.226:30443/hello -u user:password

```

Output: {"message": "hello world"}

3) Logout from the endpoint.

```

curl -X POST http://35.174.60.226:30443/logmeout

```

Output: {"message": "you have been logged out successfully"}

**#Security:**

1) Access to endpoint can be configures using EC2 security groups.

2) Endpoint is secure with username & password.

#

**Logging:**

1) Applicaation logs messages for each REST call. Below is snippet of log:

```

2020-04-15 02:41:23.971 WARN 1 --- [nio-9011-exec-5] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' not supported]

2020-04-15 02:41:25.324 WARN 1 --- [nio-9011-exec-7] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' not supported]

2020-04-15 02:41:35.978 INFO 1 --- [nio-9011-exec-1] c.l.s.r.s.hello.HelloWorldController : accessing hello API

2020-04-15 02:41:37.955 INFO 1 --- [nio-9011-exec-3] c.l.s.r.s.hello.HelloWorldController : accessing hello API

2020-04-15 02:41:50.874 INFO 1 --- [nio-9011-exec-6] c.l.s.r.s.hello.HelloWorldController : accessing hello API

2020-04-15 02:45:54.751 INFO 1 --- [nio-9011-exec-2] c.l.s.r.s.hello.HelloWorldController : accessing logmeout API

2020-04-15 02:45:55.865 INFO 1 --- [nio-9011-exec-1] c.l.s.r.s.hello.HelloWorldController : accessing logmeout API

2020-04-15 02:46:18.334 INFO 1 --- [nio-9011-exec-3] c.l.s.r.s.hello.HelloWorldController : accessing hello API

2020-04-15 02:47:03.238 INFO 1 --- [nio-9011-exec-4] c.l.s.r.s.hello.HelloWorldController : accessing logmeout API

2020-04-15 03:23:51.417 INFO 1 --- [nio-9011-exec-2] c.l.s.r.s.hello.HelloWorldController : accessing hello API

2020-04-15 03:24:37.138 INFO 1 --- [nio-9011-exec-3] c.l.s.r.s.hello.HelloWorldController : accessing logmeout API

2020-04-15 03:25:24.899 INFO 1 --- [nio-9011-exec-8] c.l.s.r.s.hello.HelloWorldController : accessing hello API

2020-04-15 03:25:41.712 INFO 1 --- [nio-9011-exec-3] c.l.s.r.s.hello.HelloWorldController : accessing hello API

```



