**Paytm-Insider Assignment**

**Write a simple Kubernetes manifest to expose a NodeJS server with the following config:**

* Use a custom docker image hosted on ECR called nodejs-test:latest (any region).
  + Created a docker image locally and pushed into ECR

A screenshot of a computer screen

Description automatically generated

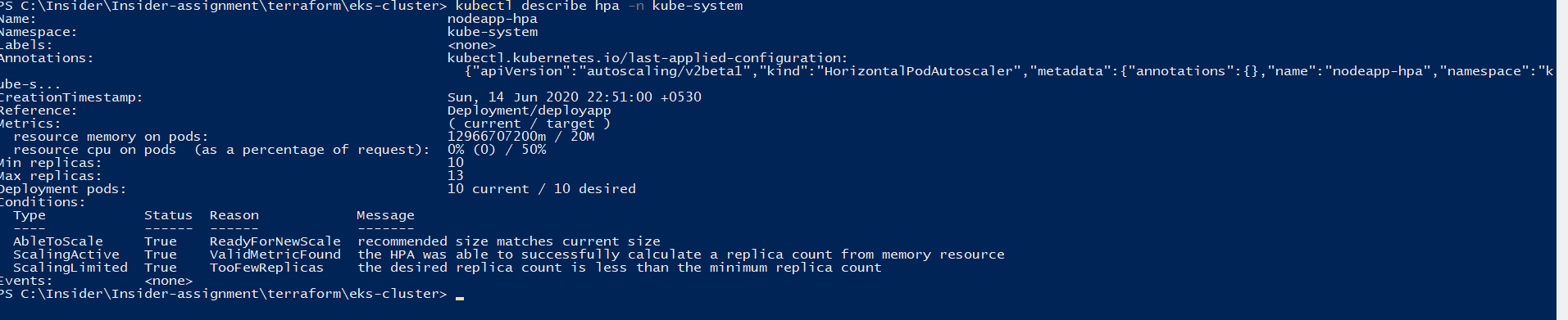
* Bonus points if you include how to login and pull an image from ECR
  + Using ECR Push commands:
    - aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 167947914732.dkr.ecr.us-west-2.amazonaws.com
    - docker build -t nodejs-test .
    - docker tag nodejs-test:latest 167947914732.dkr.ecr.us-west-2.amazonaws.com/nodejs-test:latest
    - docker push 167947914732.dkr.ecr.us-west-2.amazonaws.com/nodejs-test:latest
* Expose the app on port 3000 via an EC2 classic load balancer:A screenshot of a computer

  Description automatically generated
* Bonus points if you do the task as code i.e. using terraform or any other configuration language of your choice.
  + Used terraform to create reources in aws:
    - EKS Cluster
    - Worker Nodes
    - Security groups
    - IAM role and Policy
* There should be 10 replicas running:

A picture containing circuit, computer

Description automatically generated

* Should have higher priority than daemonset pods
  + Priority Class is used to provide high priority
* The deployment should auto scale at average 50% cpu and 60% memory:
  + HPA and Cluster-autoscaler deployed in cluster for scale in and scale out



* Any change to the deployment should always ensure at least 7 replicas are always running
  + Used Replica strategy Rolling Update to make ensure 7 replicas should available all the time.
* Bonus points if you also load test the application and include the test results in your submission.

A screenshot of a computer

Description automatically generated

**Tools:**

* **Awscli**
* **Kubectl**
* **Aws-iam-authenticator**
* **Docker**
* **EKS components:**
  + **Cluster-autoscaler**
  + **Metrics-API**
  + **Priority Class**
  + **HPA (Horizontal pod autoscaling)**