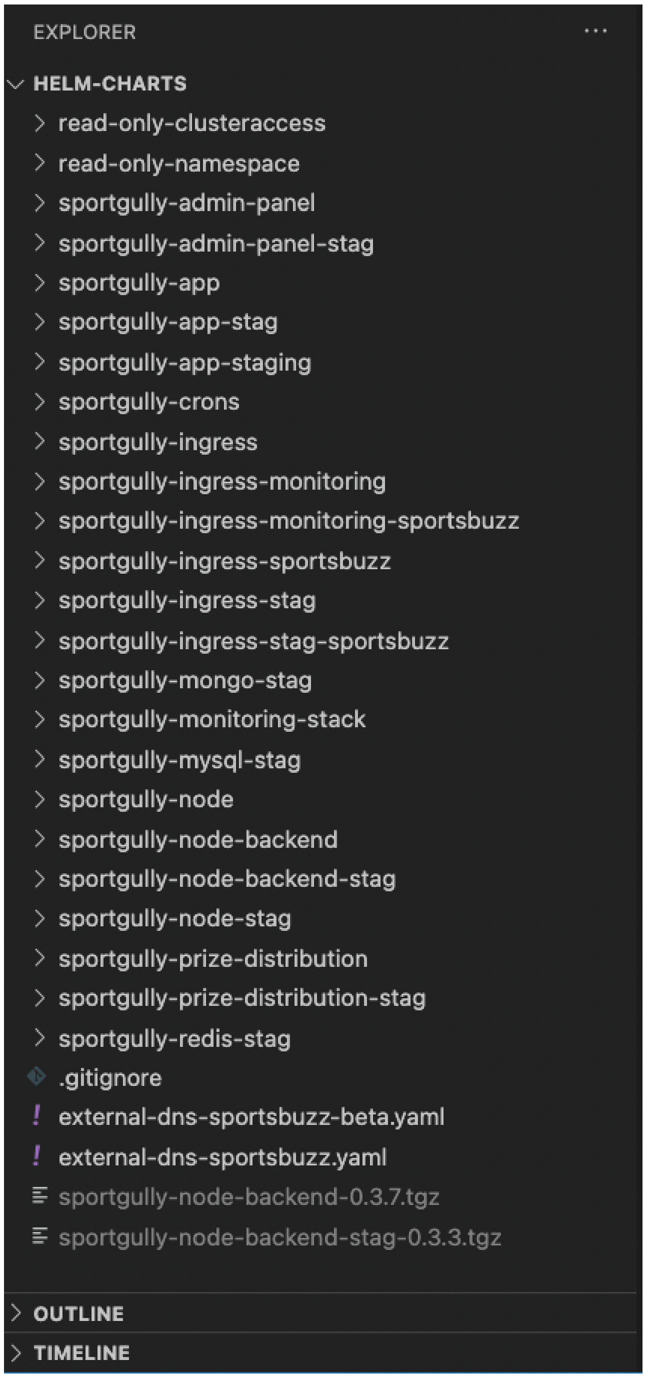
Configuring Helm Chart in the local system

First, install helm in your system. It can be downloaded from here, for different OS versions, various methods are defined.

After installing helm, clone the helm repo in your system at your desired location, here you can find the repo link.

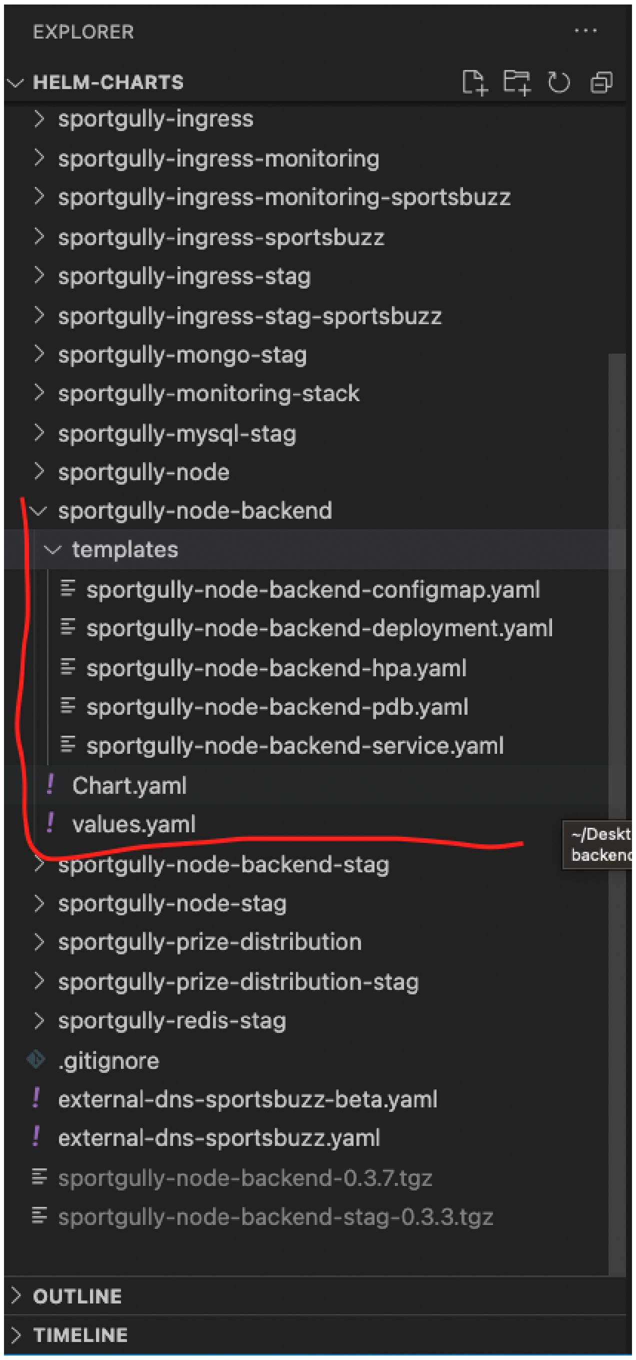
Open it in your favorite editor, it will look something like this



```
read-only-clusteraccess
read-only-namespace
sportgully-admin-panel
sportgully-admin-panel-stag
sportgully-app
sportgully-app-stag
sportgully-app-staging
sportgully-crons
sportgully-ingress
sportgully-ingress-monitoring
sportgully-ingress-monitoring-sportsbuzz
sportgully-ingress-sportsbuzz
sportgully-ingress-stag
sportgully-ingress-stag-sportsbuzz
sportgully-mongo-stag
sportgully-monitoring-stack
sportgully-mysql-stag
sportgully-node
sportgully-node-backend
sportgully-node-backend-stag
sportgully-node-stag
sportgully-prize-distribution
sportgully-prize-distribution-stag
sportgully-redis-stag
.gitignore
external-dns-sportsbuzz-beta.yaml
external-dns-sportsbuzz.yaml
sportgully-node-backend-0.3.7.tgz
sportgully-node-backend-stag-0.3.3.tgz
```

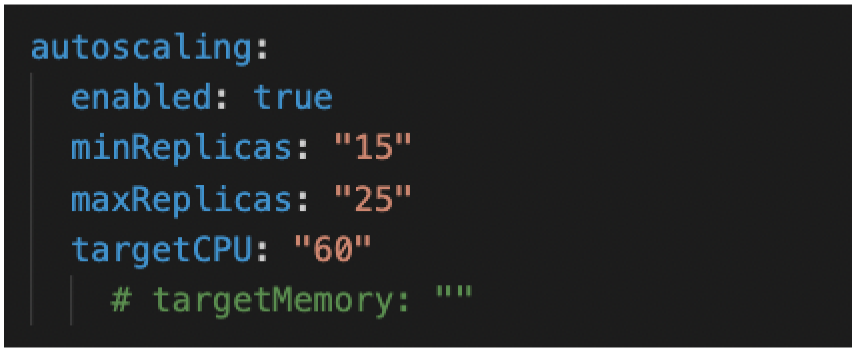
For every microservice, separate helm charts are created, so if you want to edit any value in the chart for a particular service, open that service's helm chart and edit it. For e.g. let's say there's a need to change the HPA value in the node backend,

1. Open the node backend chart



```
EXPLORER
HELM-CHARTS
  sportgully-ingress
  sportgully-ingress-monitoring
  sportgully-ingress-monitoring-sportsbuzz
  sportgully-ingress-sportsbuzz
  sportgully-ingress-stag
  sportgully-ingress-stag-sportsbuzz
  sportgully-mongo-stag
  sportgully-monitoring-stack
  sportgully-mysql-stag
  sportgully-node
  sportgully-node-backend
    templates
      sportgully-node-backend-configmap.yaml
      sportgully-node-backend-deployment.yaml
      sportgully-node-backend-hpa.yaml
      sportgully-node-backend-pdb.yaml
      sportgully-node-backend-service.yaml
    Chart.yaml
    values.yaml
  sportgully-node-backend-stag
  sportgully-node-stag
  sportgully-prize-distribution
  sportgully-prize-distribution-stag
  sportgully-redis-stag
  .gitignore
  external-dns-sportsbuzz-beta.yaml
  external-dns-sportsbuzz.yaml
  sportgully-node-backend-0.3.7.tgz
  sportgully-node-backend-stag-0.3.3.tgz
OUTLINE
TIMELINE
```

2. Now head over to **values.yaml** and edit the autoscaling section, whichever value needs to be changed.



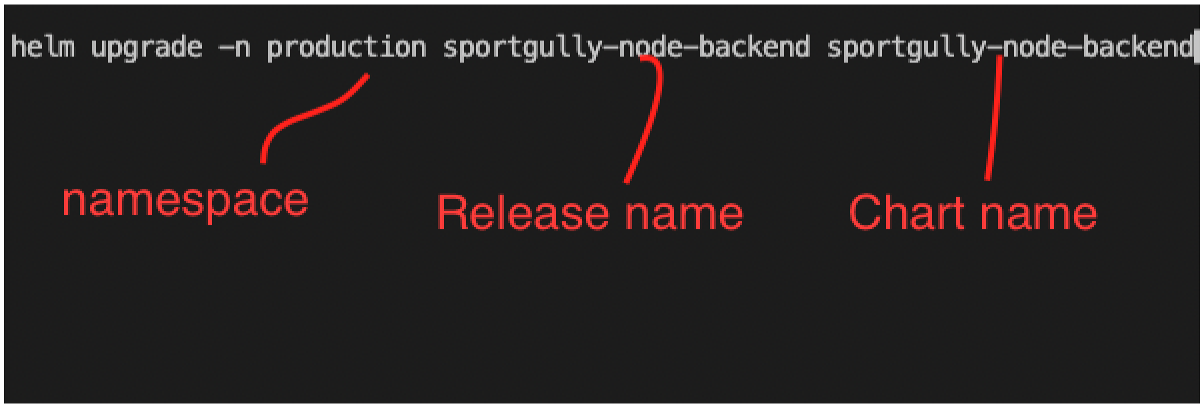
```yaml
autoscaling:
  enabled: true
  minReplicas: "15"
  maxReplicas: "25"
  targetCPU: "60"
    # targetMemory: ""
```

3. First, check the chart name using the command
   helm ls -n production (instead of production change as per the environment)

4. This repo is managed using git so you can use all git commands, and do a git pull every time before making any change in the files.

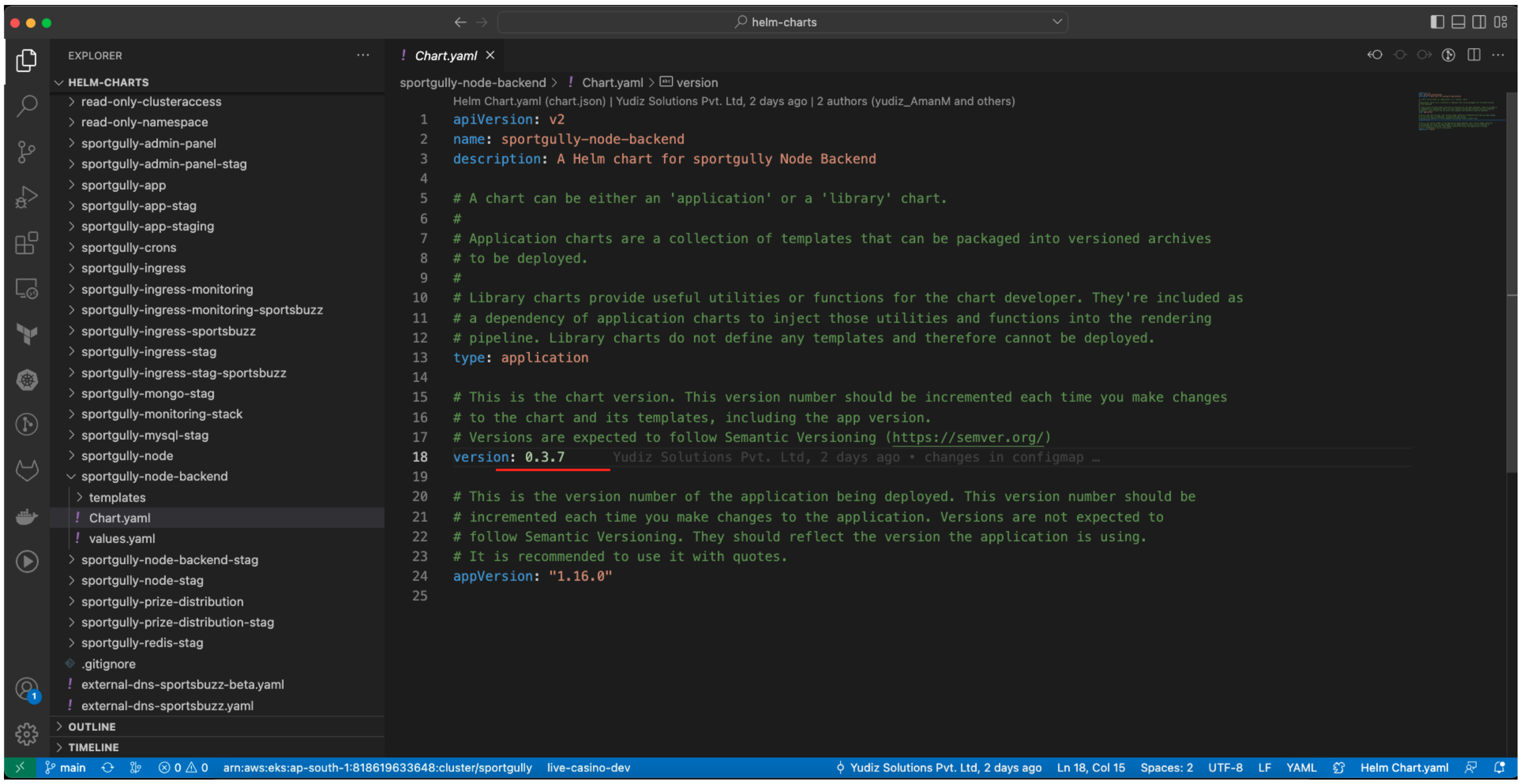
5. **Before upgrading, it is very much important to change the cluster in which you want to update.**

6. Now upgrade it in the cluster using a command, which is as follows



```
helm upgrade -n production sportgully-node-backend sportgully-node-backend
```

7. Now update the chart number first by editing the chart version in the file


helm-charts
EXPLORER
HELM-CHARTS
read-only-clusteraccess
read-only-namespace
sportgully-admin-panel
sportgully-admin-panel-stag
sportgully-app
sportgully-app-stag
sportgully-app-staging
sportgully-crons
sportgully-ingress
sportgully-ingress-monitoring
sportgully-ingress-monitoring-sportsbuzz
sportgully-ingress-sportsbuzz
sportgully-ingress-stag
sportgully-ingress-stag-sportsbuzz
sportgully-mongo-stag
sportgully-monitoring-stack
sportgully-mysql-stag
sportgully-node
sportgully-node-backend
templates
Chart.yaml
values.yaml
sportgully-node-backend-stag
sportgully-node-stag
sportgully-prize-distribution
sportgully-prize-distribution-stag
sportgully-redis-stag
.gitignore
external-dns-sportsbuzz-beta.yaml
external-dns-sportsbuzz.yaml
OUTLINE
TIMELINE
Chart.yaml
sportgully-node-backend > Chart.yaml > version
Helm Chart.yaml (chart.json) | Yudiz Solutions Pvt. Ltd, 2 days ago | 2 authors (yudiz_AmanM and others)
main
arn:aws:eks:ap-south-1:818619633648:cluster/sportgully
live-casino-dev
Yudiz Solutions Pvt. Ltd, 2 days ago
Ln 18, Col 15
Spaces: 2
UTF-8
LF
YAML
Helm Chart.yaml

```yaml
apiVersion: v2
name: sportgully-node-backend
description: A Helm chart for sportgully Node Backend

# A chart can be either an 'application' or a 'library' chart.
#
# Application charts are a collection of templates that can be packaged into versioned archives
# to be deployed.
#
# Library charts provide useful utilities or functions for the chart developer. They're included as
# a dependency of application charts to inject those utilities and functions into the rendering
# pipeline. Library charts do not define any templates and therefore cannot be deployed.
type: application

# This is the chart version. This version number should be incremented each time you make changes
# to the chart and its templates, including the app version.
# Versions are expected to follow Semantic Versioning (https://semver.org/)
version: 0.3.7      Yudiz Solutions Pvt. Ltd, 2 days ago • changes in configmap …

# This is the version number of the application being deployed. This version number should be
# incremented each time you make changes to the application. Versions are not expected to
# follow Semantic Versioning. They should reflect the version the application is using.
# It is recommended to use it with quotes.
appVersion: "1.16.0"
```

8. Now make a package of the chart which was updated, in order to reflect the changes whenever CICD is triggered. To package a chart using this command **(P.S. these all are compulsory steps)**

    helm package <chart-name>(here sportgully-node-backend)

9. **Install the cm-push plugin from here**

10. Login into the helm-repo using this command

    ```
    helm repo add \
    --username nisarg.satani \
    --password glpat-Prpng987UuWqeW9xsQAH \
    Sportsbuzz11 \
    https://gitlab.com/api/v4/projects/40317289/packages/helm/stable
    ```

11. After that push the chart to the package registry using this command

    ```
    helm cm-push sportgully-node-backend(change the name as required) Sportsbuzz11
    ```