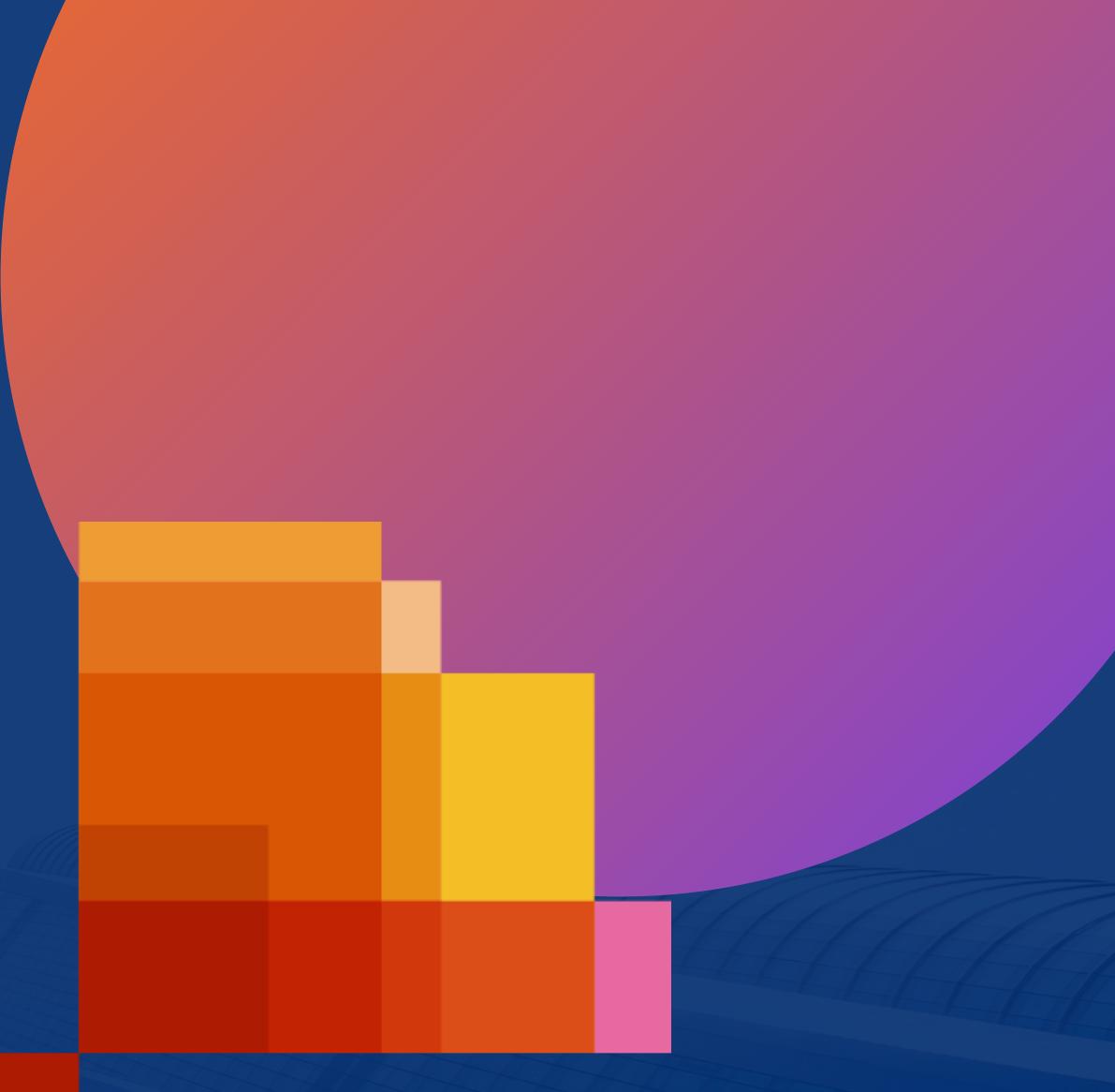


# Capstone Project

-Team 3

**pwc**



# Meet Our Team



Sandeep  
bandaru



Mellacheruvu  
Harini



Likhitha  
Chowdary



Soumyanil Ain



Mohan Manoj  
Sakala

Team leader - Helmchart

Ansible

Blockdiagram and scheme

Introduction and Application

Docker and kubernetes

A photograph of a person's hands working on a laptop keyboard. The person is wearing a striped shirt and a white watch. The laptop screen is visible in the background, showing some documents. The background of the slide is white, with a large, semi-transparent orange-to-purple gradient shape covering the lower half. Two smaller, solid orange and purple circles are positioned on the left and right edges of this shape.

# Problem Statement

Deploying a sample online voting system using K8s through containerization, container management and automating K8s through helm chart and terraform.

# Problem Description



Online voting systems can speed up election results and lower the cost of conducting an election by significantly reducing the number of people required to operate a polling place and tabulate results. A primary concern with e-voting, however, is how to store votes so they can be recounted if required

The Online Voting System becomes more automated with the help of Cloud Eks and kubernetes and there is no need for continuous management of the portal. There is no need for third party tools in order to manage the servers when they get down with the help of kubernetes having an auto scaling feature.

# Application Features

01

**Vote preview**

02

**Multiple votes**

03

**View real time result  
in dashboard**

04

**Changeable  
order of  
positions to  
show in ballot**

05

**CRUD voters,  
candidates,  
positions**

06

**Print voting  
result in PDF**



# Technology Stack

**Application language:**

PHP

**Databases used:**

MySQL

**Design used:**

Bootstrap

**Cloud Infrastructure:**

AWS

**Infrastructure Automation:** Terraform

Docker

**Orchestrating:** Kubernetes

Helm Chart, Ansible

**Package Management:** VPC, EKS, EC2, Security group

, Firewall

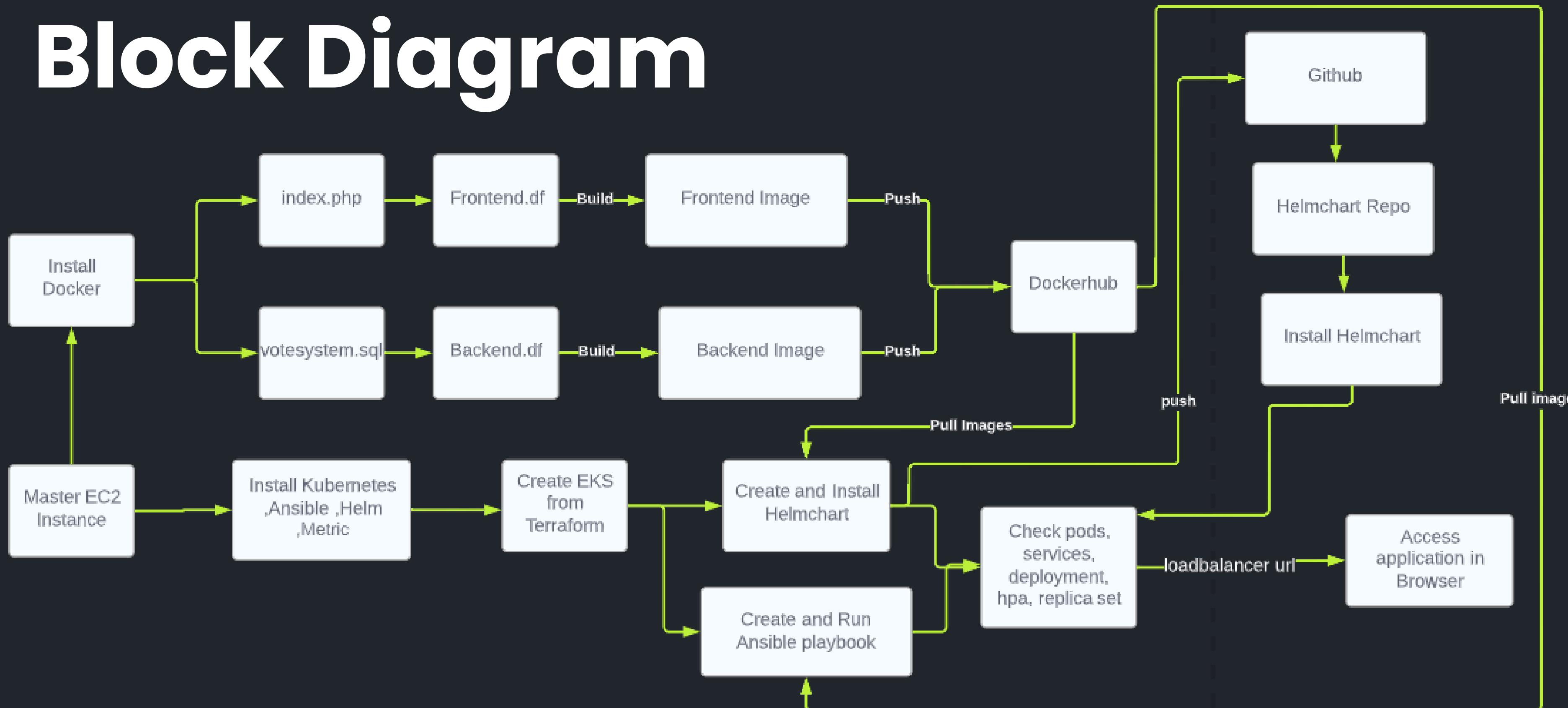
**Cloud services:**

Dockerhub, Github

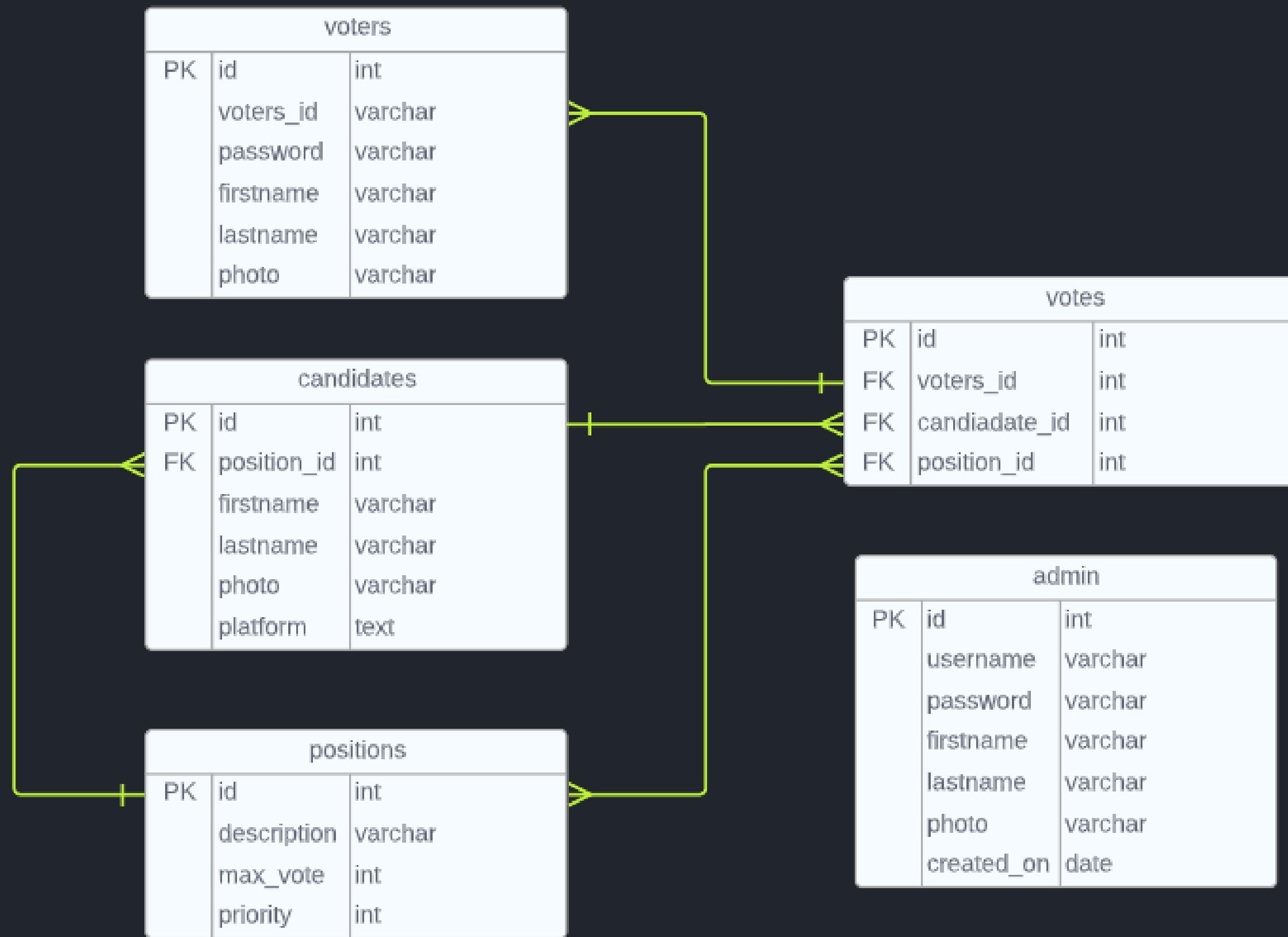
**Repositories:**



# Block Diagram



# Database schema



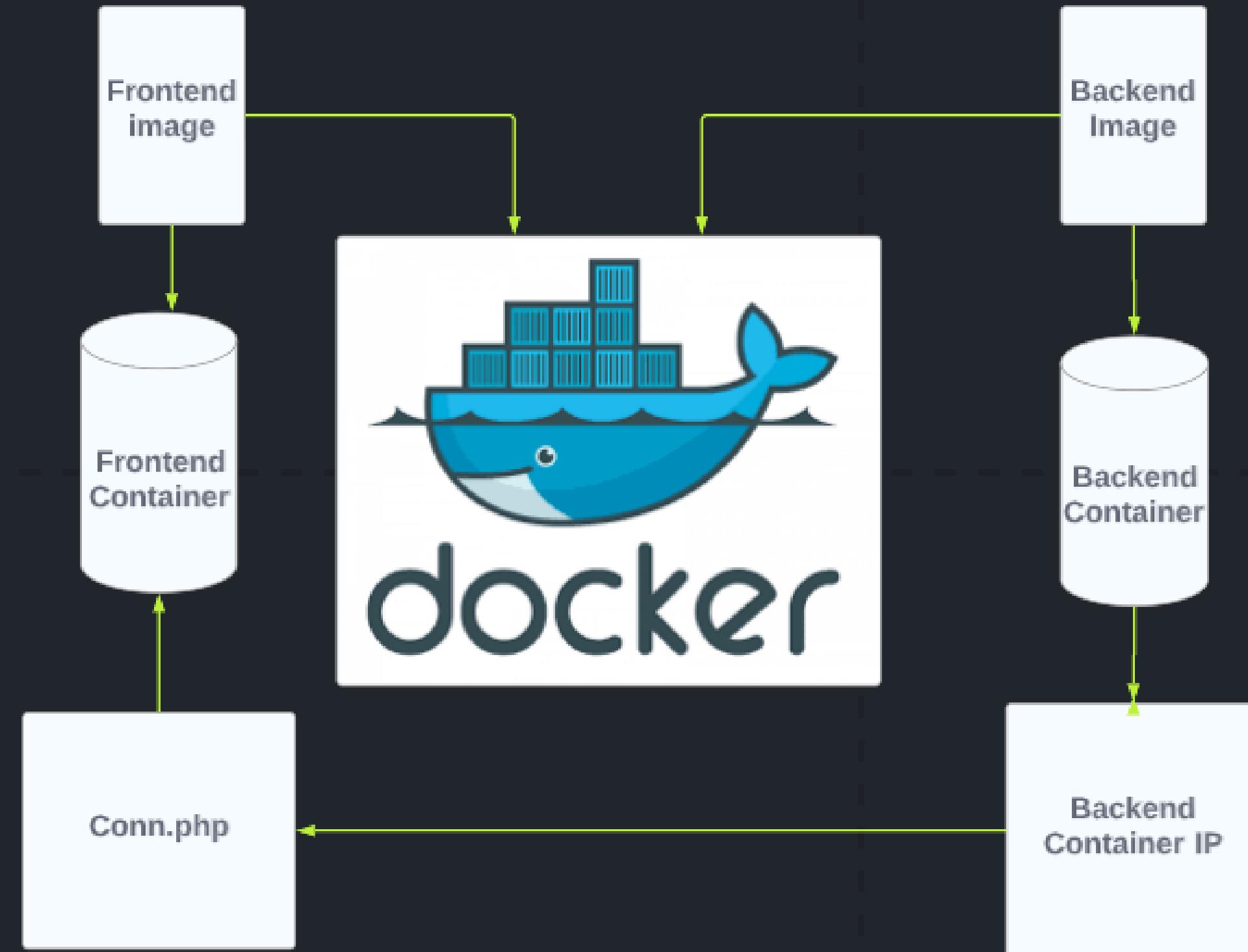
# Php.df

```
FROM centos:7
MAINTAINER "Team 3"
LABEL "App"="Development"
RUN yum install httpd -y
RUN yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm -y
RUN yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm -y
RUN yum install yum-utils -y
RUN yum-config-manager --enable remi-php72 -y
RUN yum install php php-mcrypt php-cli php-gd php-curl php-mysql php-ldap php-zip php-fileinfo -y
EXPOSE 80
ADD votesystem /var/www/html
RUN systemctl enable httpd
CMD ["httpd", "-D", "FOREGROUND"]
```



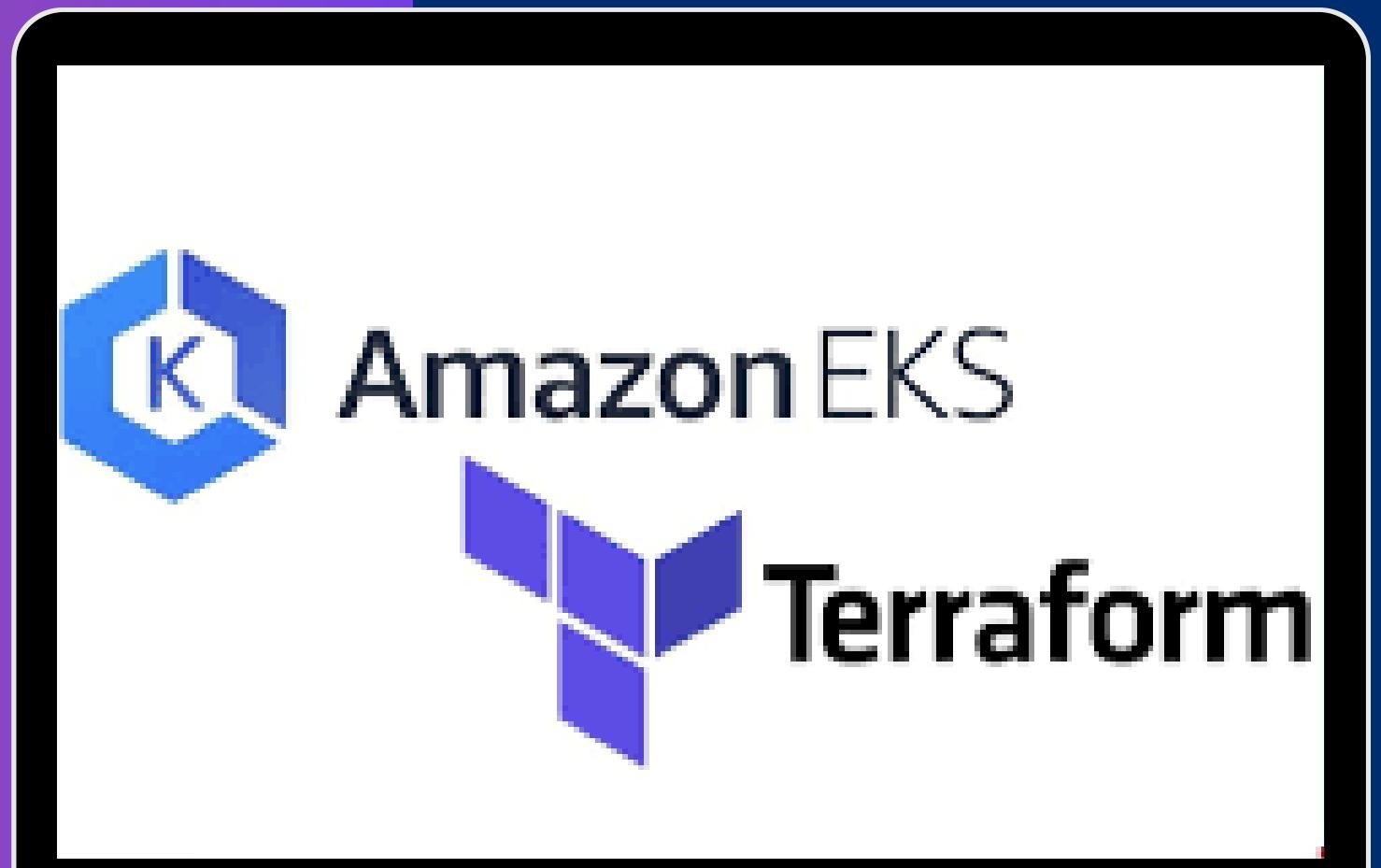
# Backend.df

```
FROM mysql
ENV MYSQL_DATABASE votesystem
COPY ./votesystem.sql /docker-entrypoint-initdb.d/
COPY ./alter.sql /docker-entrypoint-initdb.d/
```



# EKS-Cluster Using Terraform

1. Clone From Hashi Crop Git Hub Repo
2. Make Changes as per our requirement
3. Terraform init,plan,apply
4. Installing Kubernetes
5. Install Metric Server

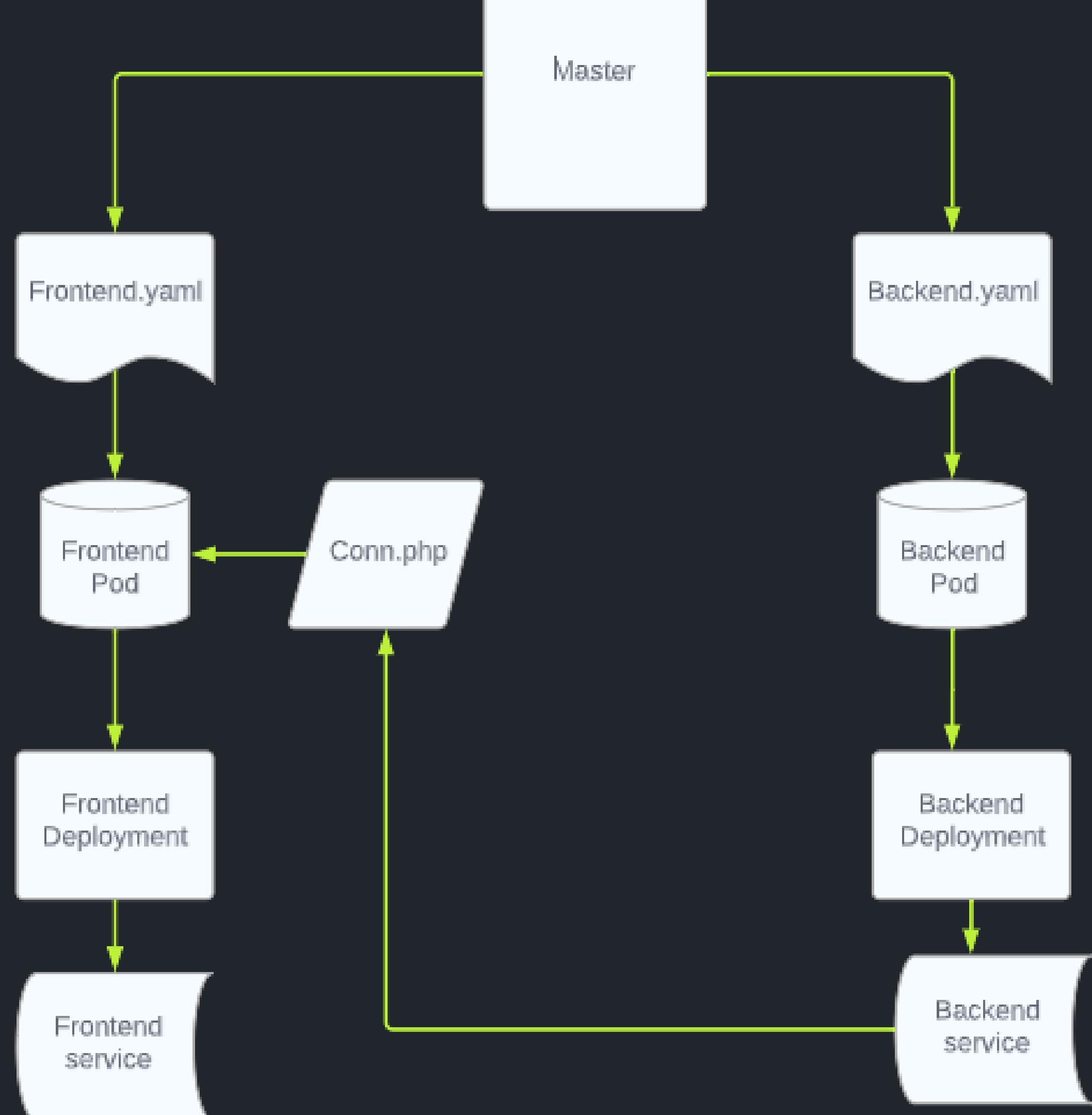


# Frontend.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: frontend
  name: frontend
  namespace: sandeep
spec:
  replicas: 1
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
        - image: sandeep9583/vote:frontend
          name: sandeep
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: frontend
  name: frontend
  namespace: sandeep
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: frontend
  type: LoadBalancer
```

# Backend.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: backend
  name: backend
  namespace: sandeep
spec:
  replicas: 1
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
        - image: sandeep9583/vote:backend
          name: sandeep
      env:
        - name: MYSQL_ROOT_PASSWORD
          value: '123'
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: backend
  name: backend
  namespace: sandeep
spec:
  ports:
    - port: 3306
      protocol: TCP
      targetPort: 3306
  selector:
    app: backend
  type: ClusterIP
```



# Playbook Tasks

## Task1

```
- name: create namespace
  k8s:
    name: harini
    api_version: v1
    kind: Namespace
    state: present
```

## Task3

```
- name: create a backend deployment
  k8s:
    api_version: v1
    namespace: harini
    definition:
      kind: Deployment
      spec:
        containers:
          - name: backend
            image: sandeep9583/vote:backend
            env:
              - name: MYSQL_ROOT_PASSWORD
                value: '123'
```

## Task2

```
- name: create a frontend deployment
  k8s:
    api_version: v1
    namespace: harini
    definition:
      kind: Deployment
      spec:
        containers:
          - name: frontend
            image: sandeep9583/vote:frontend
            resources:
              limits:
                cpu: 500m
              requests:
                cpu: 200m
```



# Playbook Tasks

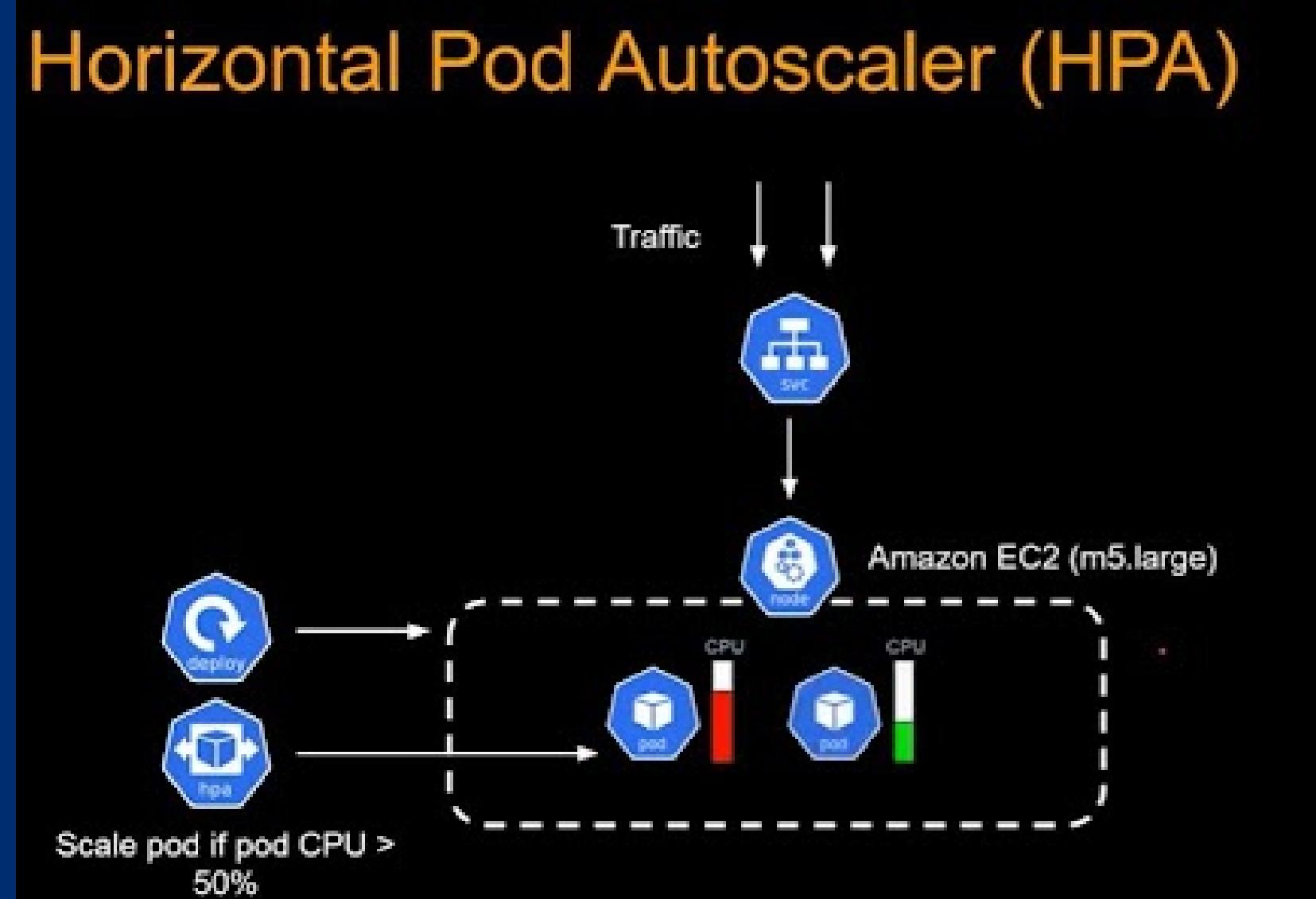
## Task4

```
- name: create a frontend service
  k8s:
    api_version: v1
    namespace: harini
    definition:
      kind: Service
      spec:
        ports:
          - port: 80
            protocol: TCP
            targetPort: 80
        selector:
          app: frontend
        type: LoadBalancer
```

## Task5

```
- name: create a backend service
  k8s:
    api_version: v1
    namespace: harini
    definition:
      kind: Service
      metadata:
        labels:
          app: backend
          name: backend
          namespace: harini
      spec:
        ports:
          - port: 3306
            protocol: TCP
            targetPort: 3306
        selector:
          app: backend
        type: ClusterIP
```

# HPA



# Playbook Tasks

```
- name: create a frontend deployment
  k8s:
    api_version: v1
    namespace: harini
    definition:
      kind: Deployment
    spec:
      containers:
        - name: frontend
          image: sandeep9583/vote:frontend
      resources:
        limits:
          cpu: 500m
        requests:
          cpu: 200m
```

## Task6

```
-- name: Deploy hpa
  k8s:
    api_version: v1
    namespace: harini
    definition:
      kind: HorizontalPodAutoscaler
    metadata:
      name: frontend
      namespace: harini
    spec:
      scaleTargetRef:
        apiVersion: apps/v1
        kind: Deployment
        name: frontend
      minReplicas: 2
      maxReplicas: 10
      targetCPUUtilizationPercentage: 30
```



# Helm Chart

## Value.yaml

```
frontend:
  name: frontend
  namespace: sandeep
  enabled: true
  replicaCount: 1
  image:
    repository: sandeep9583/vote:frontend

backend:
  name: backend
  namespace: sandeep
  enabled: true
  replicaCount: 1
  image:
    repository: sandeep9583/vote:backend

hpa:
  name: frontend
  namespace: sandeep
```

```
team
  charts
  Chart.yaml
  templates
    deployment.yaml
    hpa.yaml
    service.yaml
  values.yaml
```

## Frontend

### Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: {{ .Values.frontend.name}}
    name: {{ .Values.frontend.name}}
    namespace: {{ .Values.frontend.namespace}}
spec:
  replicas: {{ .Values.frontend.replicaCount }}
  selector:
    matchLabels:
      app: {{ .Values.frontend.name}}
  template:
    metadata:
      labels:
        app: {{ .Values.frontend.name}}
    spec:
      containers:
        - image: {{ .Values.frontend.image.repository}}
          name: {{ .Values.frontend.name}}
          resources:
            limits:
              cpu: 500m
            requests:
              cpu: 200m
```

### Service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: {{ .Values.frontend.name}}
    name: {{ .Values.frontend.name}}
    namespace: {{ .Values.frontend.namespace}}
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: {{ .Values.frontend.name}}
  type: LoadBalancer
```

### hpa.yaml

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: {{ .Values.hpa.name}}
  namespace: {{ .Values.hpa.namespace}}
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: {{ .Values.hpa.name}}
  minReplicas: 2
  maxReplicas: 10
  targetCPUUtilizationPercentage: 30
```



# Helm Chart

## Value.yaml

```
frontend:
  name: frontend
  namespace: sandeep
  enabled: true
  replicaCount: 1
  image:
    repository: sandeep9583/vote:frontend

backend:
  name: backend
  namespace: sandeep
  enabled: true
  replicaCount: 1
  image:
    repository: sandeep9583/vote:backend

hpa:
  name: frontend
  namespace: sandeep
```

## Backend

### Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: {{ .Values.backend.name}}
    name: {{ .Values.backend.name}}
    namespace: {{ .Values.backend.namespace}}
spec:
  replicas: {{ .Values.backend.replicaCount }}
  selector:
    matchLabels:
      app: {{ .Values.backend.name}}
  template:
    metadata:
      labels:
        app: {{ .Values.backend.name}}
    spec:
      containers:
        - image: {{ .Values.backend.image.repository}}
          name: {{ .Values.backend.name}}
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: "123"
```

### Service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: {{ .Values.backend.name}}
    name: {{ .Values.backend.name}}
    namespace: {{ .Values.backend.namespace}}
spec:
  ports:
    - port: 3306
      protocol: TCP
      targetPort: 3306
  selector:
    app: {{ .Values.backend.name}}
  type: ClusterIP
```

# Steps for Installing Helm Charts from Repositories

1. `helm search repo votesystem_helm/team`
2. `helm install <Any name> votesystem_helm/team`
3. `watch kubectl get all`

# Application Admin Page

**VotingSystem**

≡

 Admin Team Online

REPORTS

 [Dashboard](#)

 [Votes](#)

MANAGE

 [Voters](#)

 [Positions](#)

 [Candidates](#)

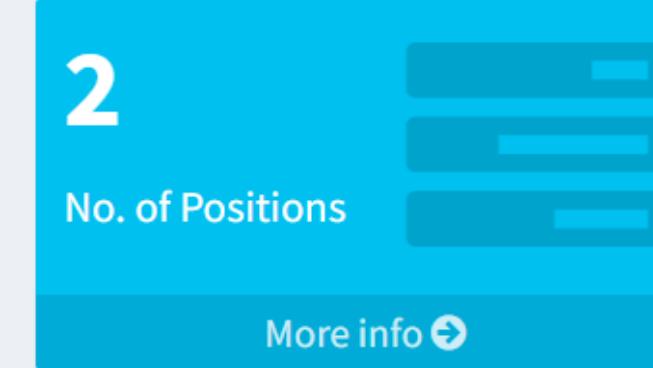
SETTINGS

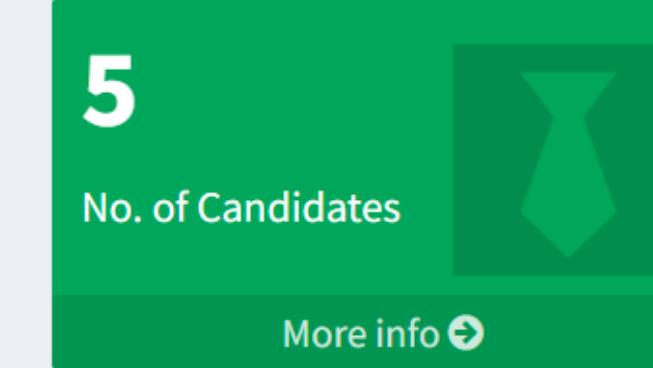
 [Ballot Position](#)

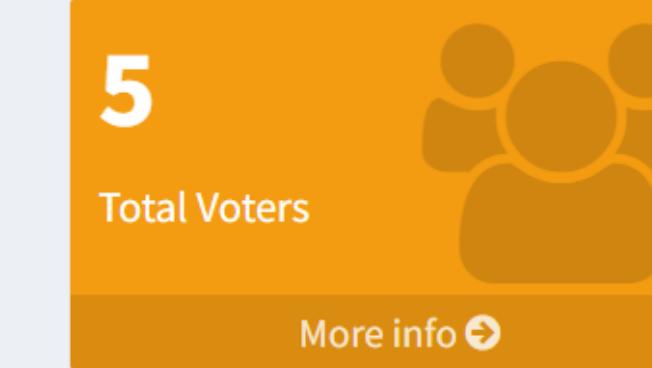
 [Election Title](#)

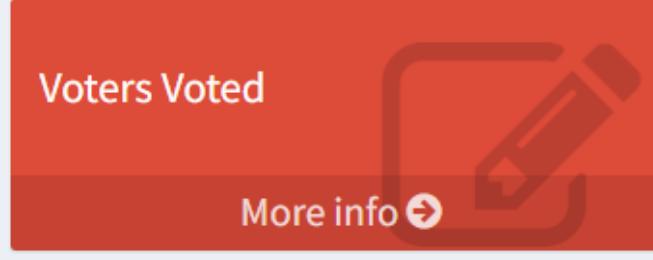
≡

**Dashboard**

 **2** No. of Positions [More info](#)

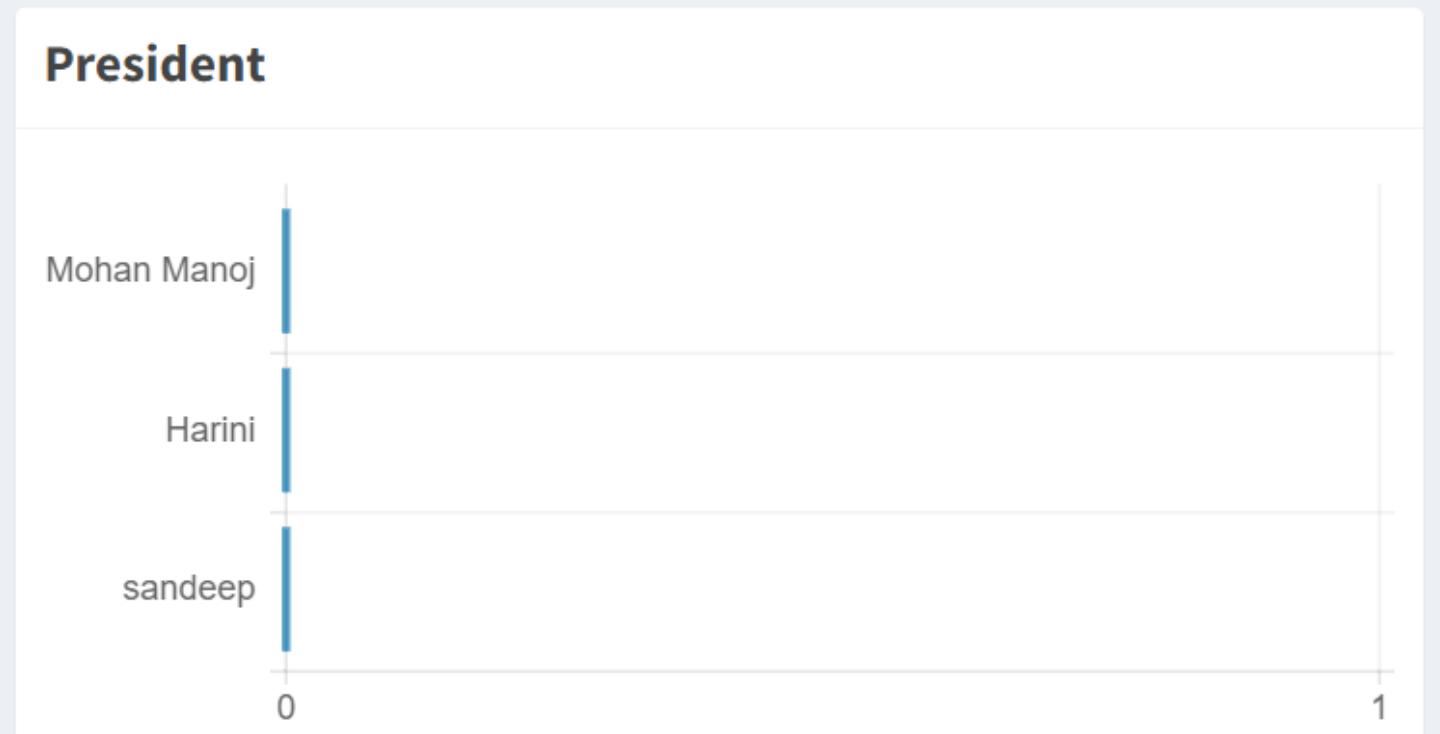
 **5** No. of Candidates [More info](#)

 **5** Total Voters [More info](#)

 **Voters Voted** [More info](#)

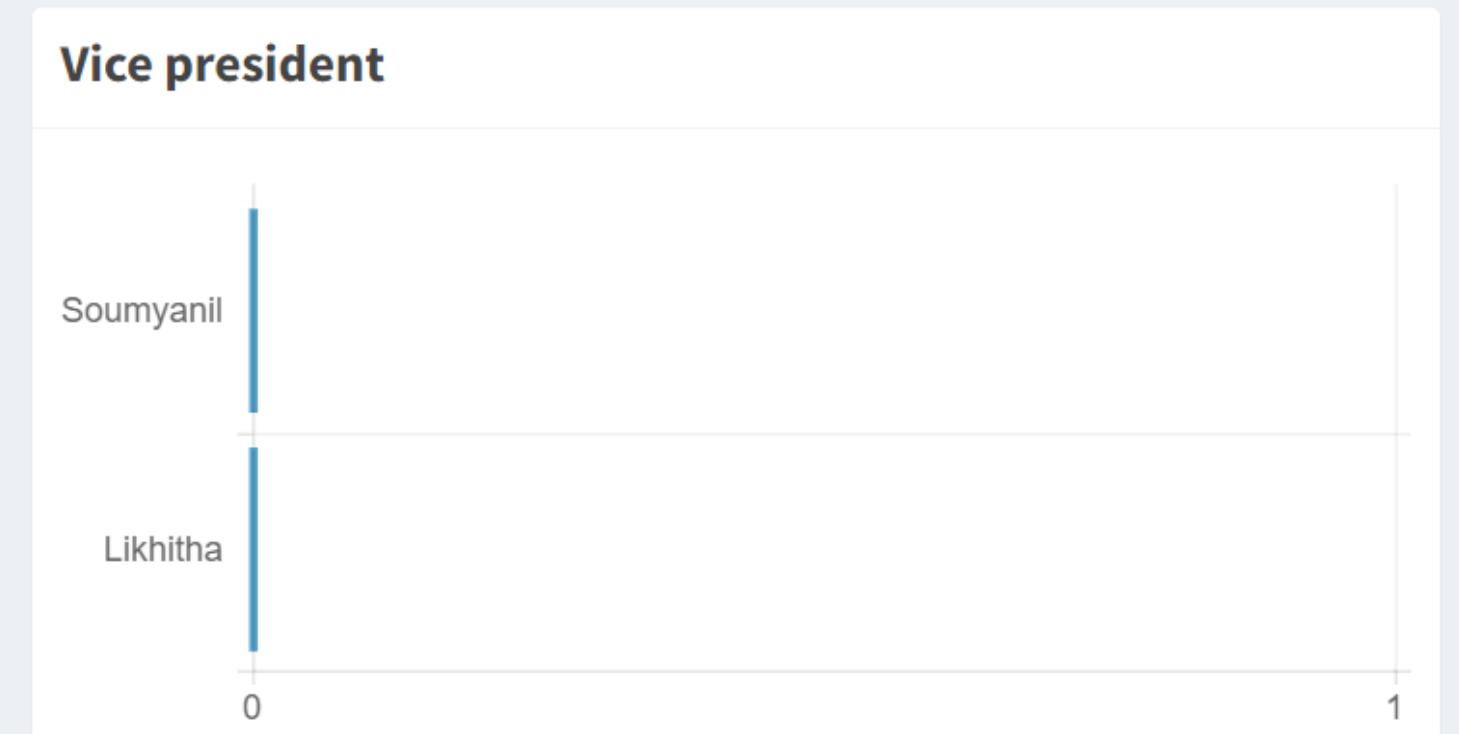
**Votes Tally**

**President**



Candidate	Votes
Mohan Manoj	1
Harini	1
sandeep	1

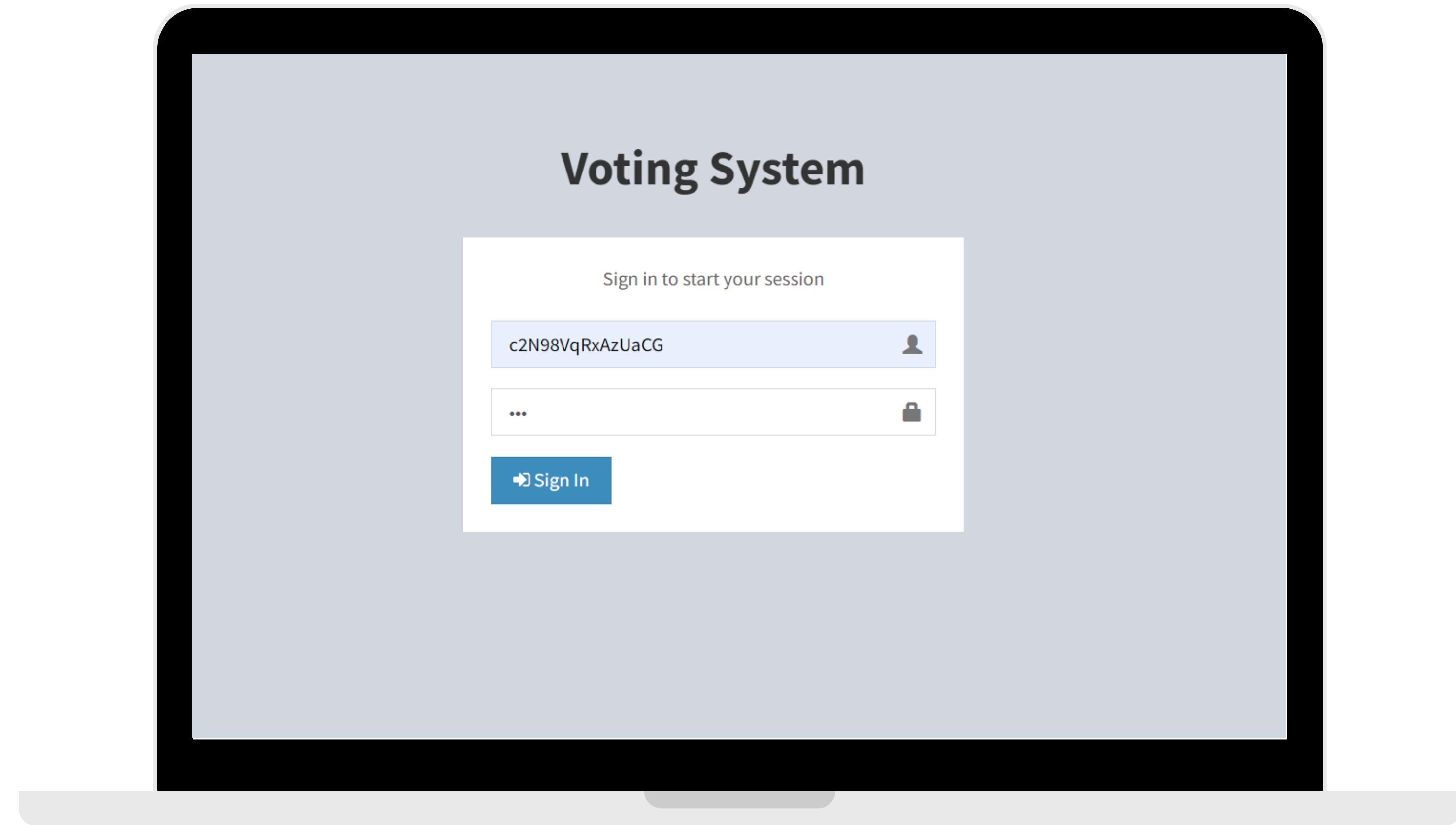
**Vice president**



Candidate	Votes
Soumyanil	1
Likhitha	1

 [Print](#)

# Application Login Page



# Application Voter Page

VotingSystem

Voter3 Voter3 [LOGOUT](#)

## 2022 ELECTIONS

**President**

Select only one candidate

[Platform](#)  Bandaru sandeep

[Platform](#)  Mellacheruvu Harini

[Platform](#)  Sakala Mohan Manoj

**Vice president**

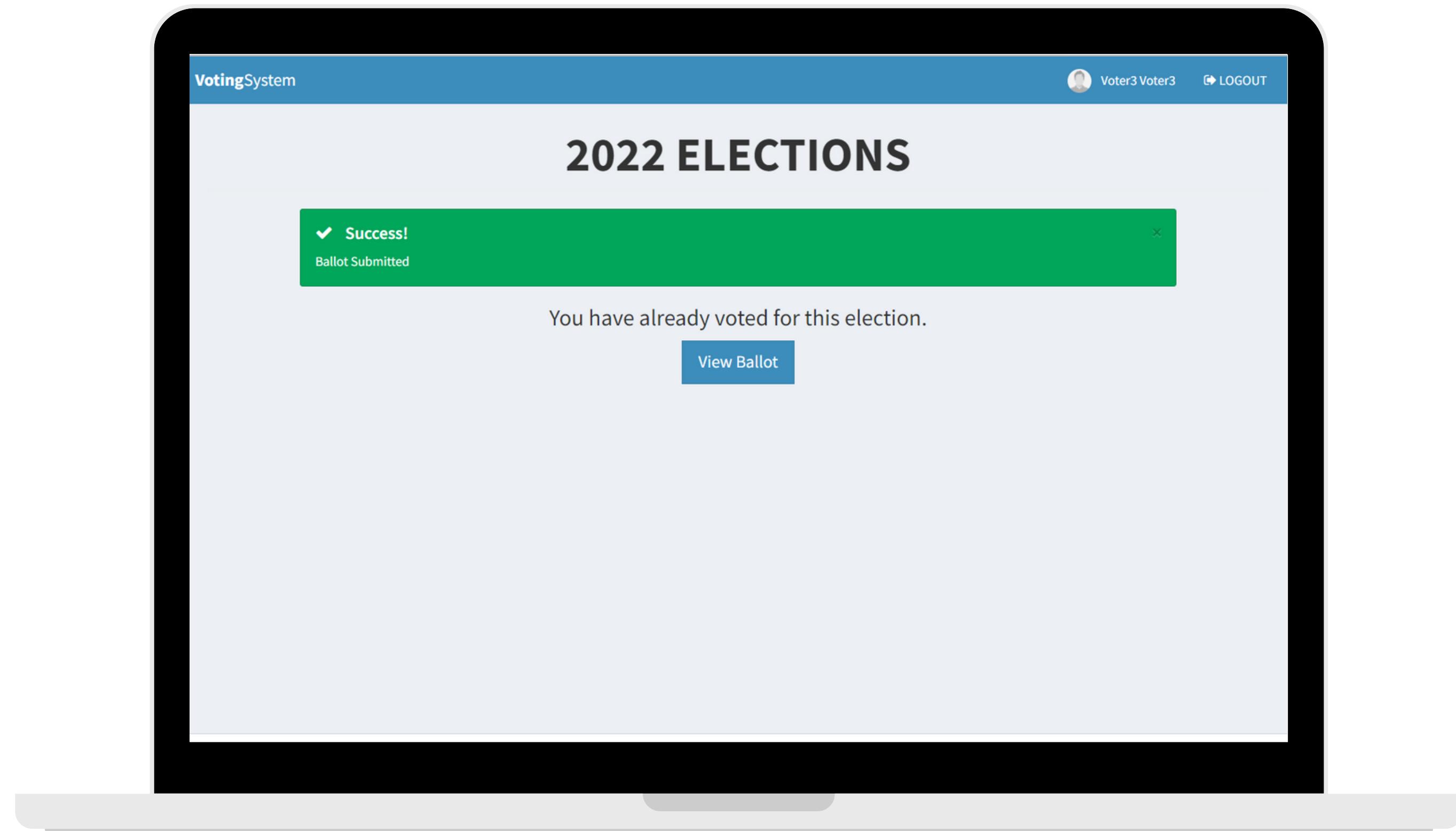
Select only one candidate

[Platform](#)  Mandava Likhitha

[Platform](#)  Ain Soumyanil

[Preview](#)  [Submit](#)

# Success Page After Voting



# Result Webpage after Casting the Vote

**VotingSystem**

≡

 Admin Team Online

REPORTS

 Dashboard

 Votes

MANAGE

 Voters

 Positions

 Candidates

SETTINGS

 Ballot Position

 Election Title

Admin Team

Home > Votes

 Reset

Show 10 entries Search:

Position	Candidate	Voter
President	Bandaru sandeep	Voter3 Voter3
President	Mellacheruvu Harini	Voter4 Voter4
Vice president	Ain Soumyanil	Voter3 Voter3
Vice president	Mandava Likhitha	Voter4 Voter4

Showing 1 to 4 of 4 entries

Previous 1 Next

# Admin Dashboard After Polling

**VotingSystem**

≡

 Admin Team Online

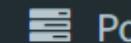
REPORTS

 [Dashboard](#)

 [Votes](#)

MANAGE

 [Voters](#)

 [Positions](#)

 [Candidates](#)

SETTINGS

 [Ballot Position](#)

 [Election Title](#)

 Admin Team

Home > Dashboard

## Dashboard

**2**  No. of Positions [More info](#)

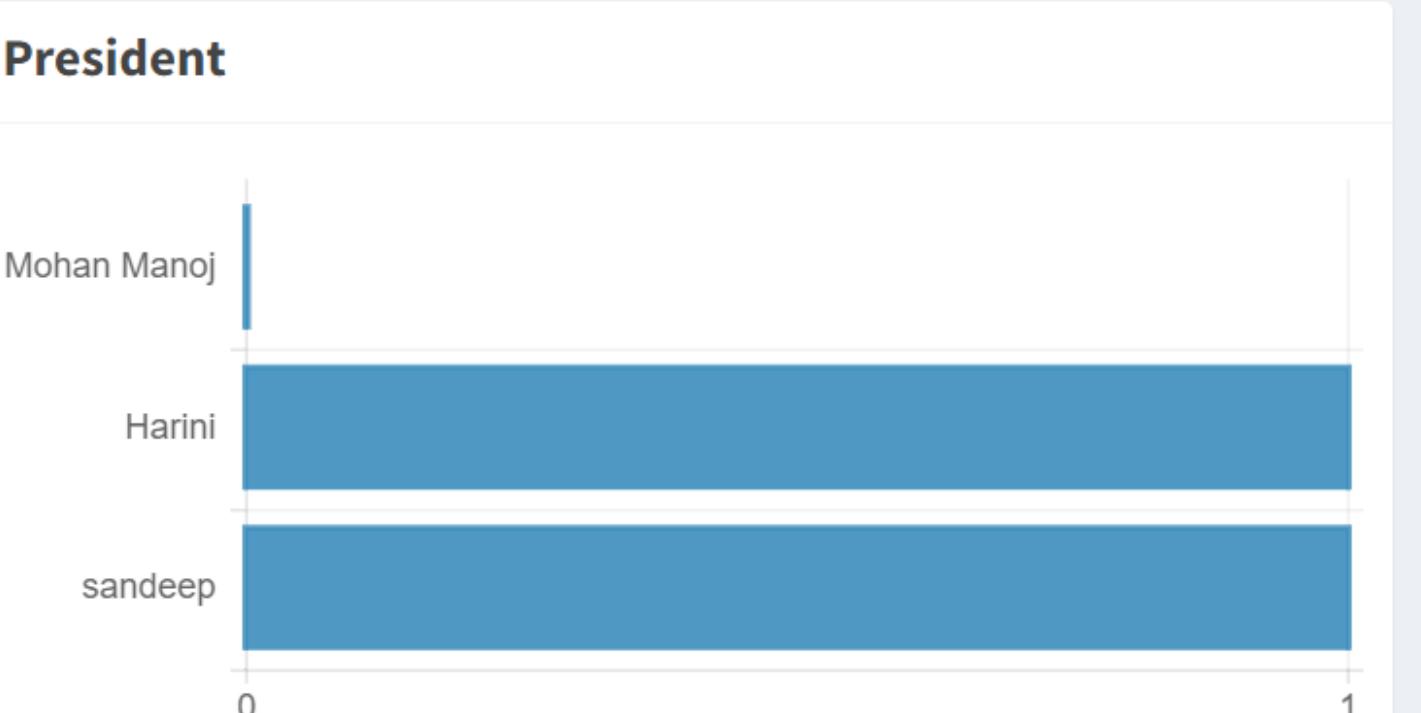
**5**  No. of Candidates [More info](#)

**5**  Total Voters [More info](#)

**Voters Voted**  [More info](#)

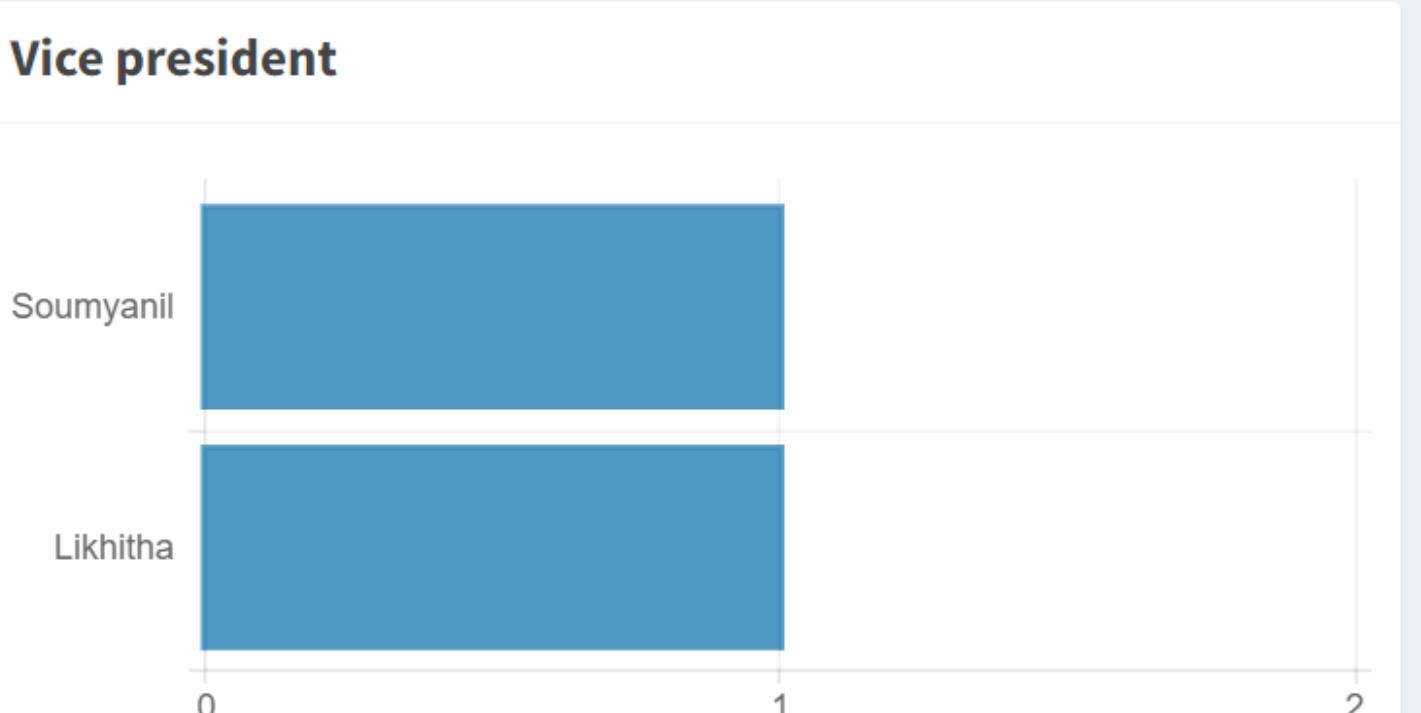
## Votes Tally

**President**



Candidate	Votes
Mohan Manoj	0
Harini	1
sandeep	1

**Vice president**



Candidate	Votes
Soumyanil	1
Likhitha	1

 [Print](#)

# Result Pdf

## 2022 Elections

### Tally Result

<b>President</b>	
<b>Candidates</b>	<b>Votes</b>
Harini, Mellacheruvu	1
Mohan Manoj, Sakala	0
sandeep, Bandaru	1
<b>Vice president</b>	
<b>Candidates</b>	<b>Votes</b>
Likhitha, Mandava	1
Soumyanil, Ain	1



## Summary

*"By summarizing the project, our team has successfully deployed a sample online voting system using K8s through containerization, container management and automating K8s through helm chart/terraform and with the help of above technology stack like AWS,terraform,docker,Ansible,helm charts."*



# Thank You

**pwc**