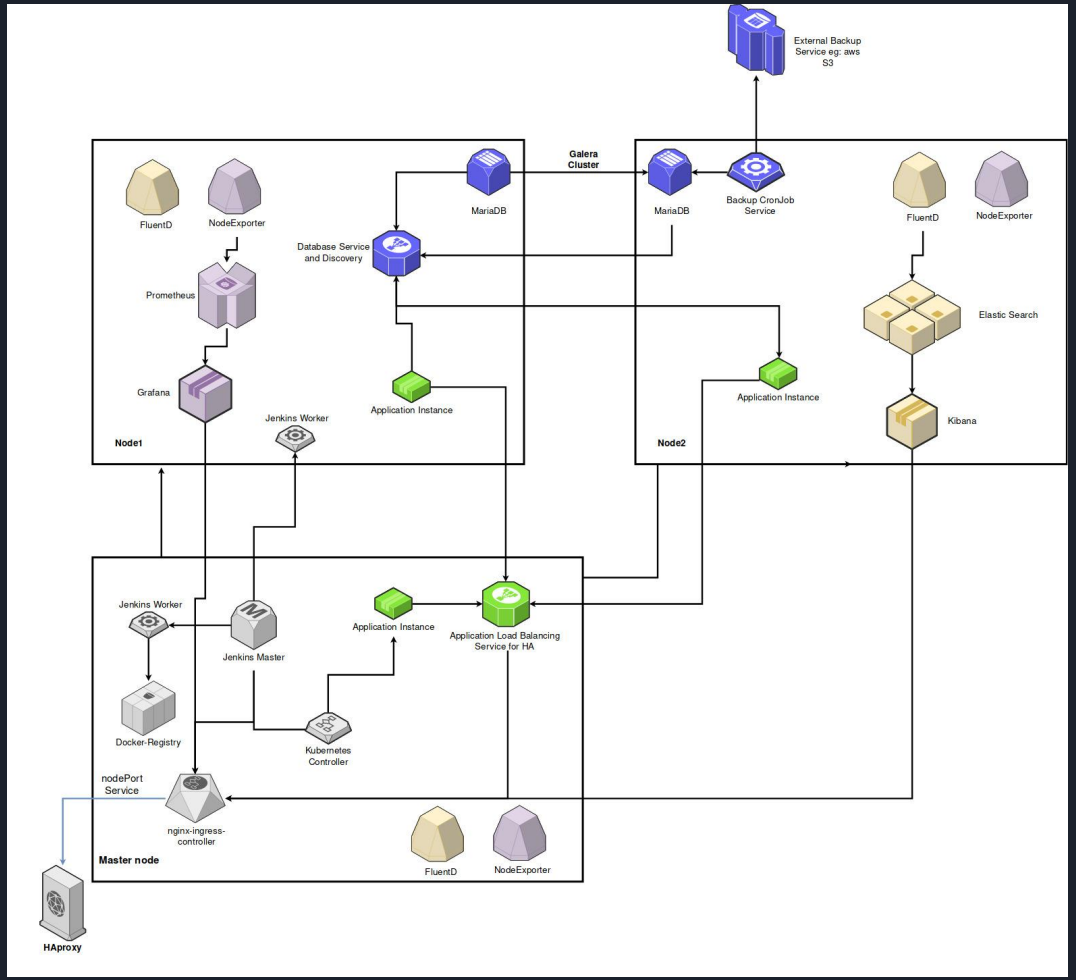
Kubernetes Cluster Breakdown

Break Down Chart



Cluster Layout

Notes : Due to demo capacity the cluster is not the ideal for high availability

- Master Node : Containing the nginx controller that acts as a gateway for the cluster and the kubernetes api server
- Node1 : general purpose node containing at least 1 mysql data node
- Node2 : general purpose node containing at least 1 mysql data node

Ideal setup would be :

- 3 Master nodes and 3 data nodes

Services listing

1. Jenkins CI/CD
2. Prometheus , Node Exporters and Grafana
3. Kibana, FluentD and Elastic Search
4. MariaDB Galera Cluster
5. Rails Application
6. Kubernetes Services
7. Nginx Ingress

Jenkins

- Jenkins is Deployed as a stateful set inside of the kubernetes cluster
- Workers are deployed on demand by the jenkins master to build and test the app
- In case of build success Jenkins starts a rolling deployment of the application to maintain high availability

Prometheus

- Prometheus collects data from exporters in nginx and the nodes and the application even metrics from itself
- It provides grafana with the data in form of time series database
- Grafana can be used to draw useful charts for the availability of the app
- An alert manager can be deployed to alert the user in case of any metrics going dangerously low (emails - slack notifications - any APIS)

EFK Stack

- FluentD log exporters collects logs from every part of the kubernetes cluster and send it to elastic search data storage
- Elastic search indexes and identifies the logs making it easy to access
- Kibana is used to search the logs provided to it through elastic search
- Elastic search can be a cluster but one node is deloyed for demo purposes

MariaDB Cluster

- 2 Database servers are deployed on different nodes to make sure at least 1 node survives
- The database servers are in master master galera cluster and more servers can be added to increase the availability
- 1 node can go offline and then rejoin the cluster and syncs back it’s missing part of the data, in case of network outage a split brain can happen causing problems so a network fail safe is made to make sure network access is provided before creating new cluster
- In ideal situations 3 database servers can be deployed to insure no split brain happens

Rails application

- Rails application is changed to use mysql other than sqlite3
- A gem is added for production server eg : unicorn
- A gem is added to provide prometheus exporter functions to collect metrics

Kubernetes services

- Kubernetes services acts as simple load balancers and discovery services for the cluster
- More Advanced methods like consul can be used for service discovery
- Services do round-robin load balancing
- Services provide high availability for underlying applications
- A service is made for the application and database elasticsearch cluster and prometheus backend

Nginx Ingress

- Nginx handles the routing of outside addresses to applications inside our cluster
- Nginx is exposed as nodeport on the master node
- HAproxy is used to route traffic to nginx controller inside the cluster