Luis Fernando Valverde Cárdenas
jueves, 30 de mayo de 2024

IES Cura Valera

PRESENTACIÓN IDEA DE PROYECTO INTEGRADO DE ASIR

Curso: 2023/2024 (2º ASIR)
Grado Superior en Administración de Sistemas Informáticos en Red

ÍNDICE

Introducción

Descripción de la idea

La idea de este proyecto consiste en **centralizar** las aplicaciones de uso cotidiano destinadas a la realización de tareas administrativas dentro del ámbito de una empresa ofreciendo algunas de las siguientes **ventajas**:

1. **Acceso desde cualquier lugar:** Desde cualquier dispositivo con conexión a Internet y un navegador web compatible reduciendo así los requisitos **hardware** del lado del cliente proporcionando una mayor flexibilidad y movilidad.
2. **Seguridad:** Medidas de seguridad más robustas, como firewalls, cifrado de datos y autenticación de usuarios.
3. **Gestión centralizada de datos:** Implementación de políticas de acceso,control de datos, copias de seguridad y recuperación de datos garantizando la integridad y la confidencialidad de los datos de manera centralizada.
4. **Escalabilidad:** Más fácil escalar y ajustar la capacidad según las necesidades de la empresa permitiendo un crecimiento flexible y escalable.

El objetivo como se ha explicado anteriormente es el de **centralizar** las aplicaciones de los clientes reduciendo cargas de trabajo,medidas de seguridad adicionales,datos sensibles… en el lado del cliente llevándoselo al lado del servidor para una **fácil y mejor** administración, mantenimiento, seguridad, disponibilidad, escalabilidad entre otras características.

Desarrollo de la idea

El **contexto** de esta idea es una empresa que le interesa las ventajas de la **centralización** donde solicita lo siguiente:

1. Un **servicio de almacenamiento compartido** con los equipos del dominio destinado a las **operaciones** del departamento TI como mantenimiento o nuevas implementaciones.
2. Un **servicio de directorio** para almacenar y organizar los distintos recursos de red, como usuarios, grupos, equipos y otros objetos del dominio.
3. Una aplicación para la **gestión,inventario y operaciones del dominio** destinada a los administradores del mismo.

Componentes

Servidores

Orquestador

Como hemos mencionado anteriormente toda la **infraestructura** va a estar centralizada en un servidor **linux** con **ubuntu server 22.04.4 LTS** como sistema operativo.

Este servidor es el que se encarga de lanzar,administrar y orquestar los **micro servicios** que componen las aplicaciones web y los recursos o complementos de dichas aplicaciones.

Ubuntu Server Docs

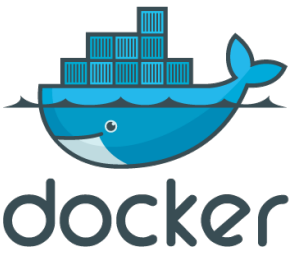
TrueNAS

Este **servidor** contendrá todos los datos de los **micro servicios** y de la empresa centralizado en un servidor especializado en el almacenamiento de los datos con un repertorio de tecnologías para ello como sistemas de archivos como **ZFS**...

TrueNAS Docs

Componentes del orquestador

Docker



Utilizaremos **docker** para construir los diferentes **micro servicios** que sustentarán las aplicaciones web como también los recursos o complementos que utilizarán dichas aplicaciones web todo esto en un **marco inicial de desarrollo** y pruebas, más adelante se desplegará en un **marco de producción** junto con **kubernetes**.

Docker Docs

Ansible

Ansible será nuestra herramienta de **automatización** para llevar a cabo en los equipos del dominio las operaciones de mantenimiento o nuevas implementaciones.

Ansible Docs

Kubernetes

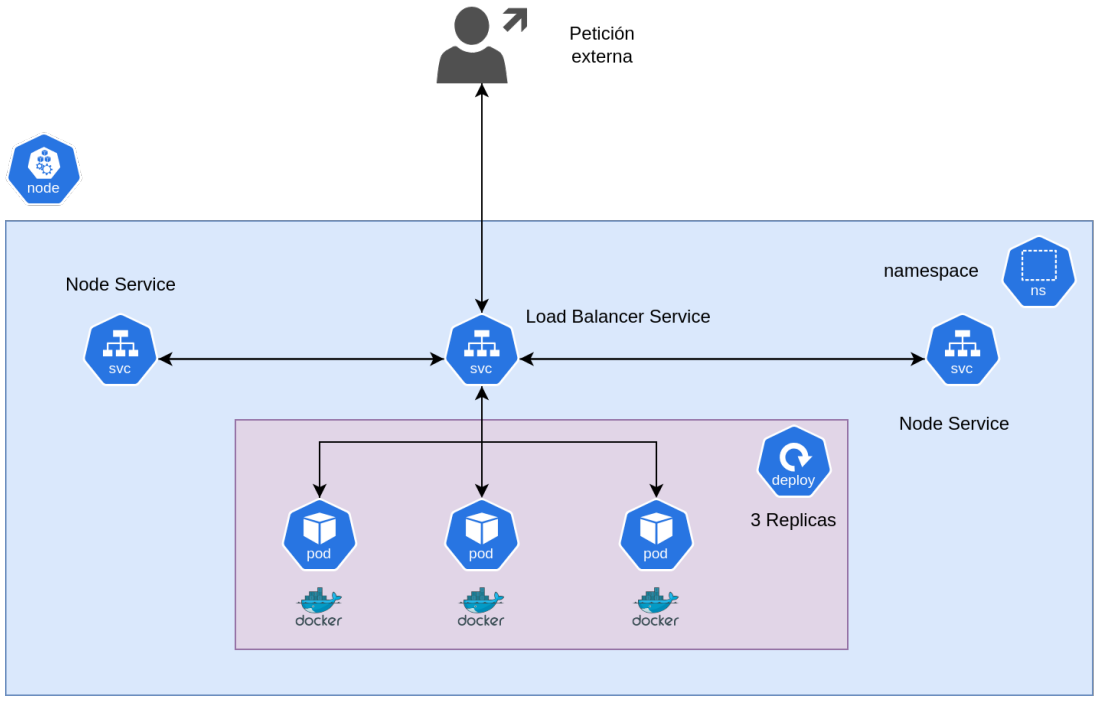
Kubernetes será nuestra tecnología de orquestación de los diferentes **micro servicios** construidos con **docker**, nos ayudará a desplegar de una manera **organizada** dichos micro servicios ofreciendo **alta disponibilidad**.

Esta será la principal tecnología de nuestro servidor **orquestador** de ahí dicho nombre.

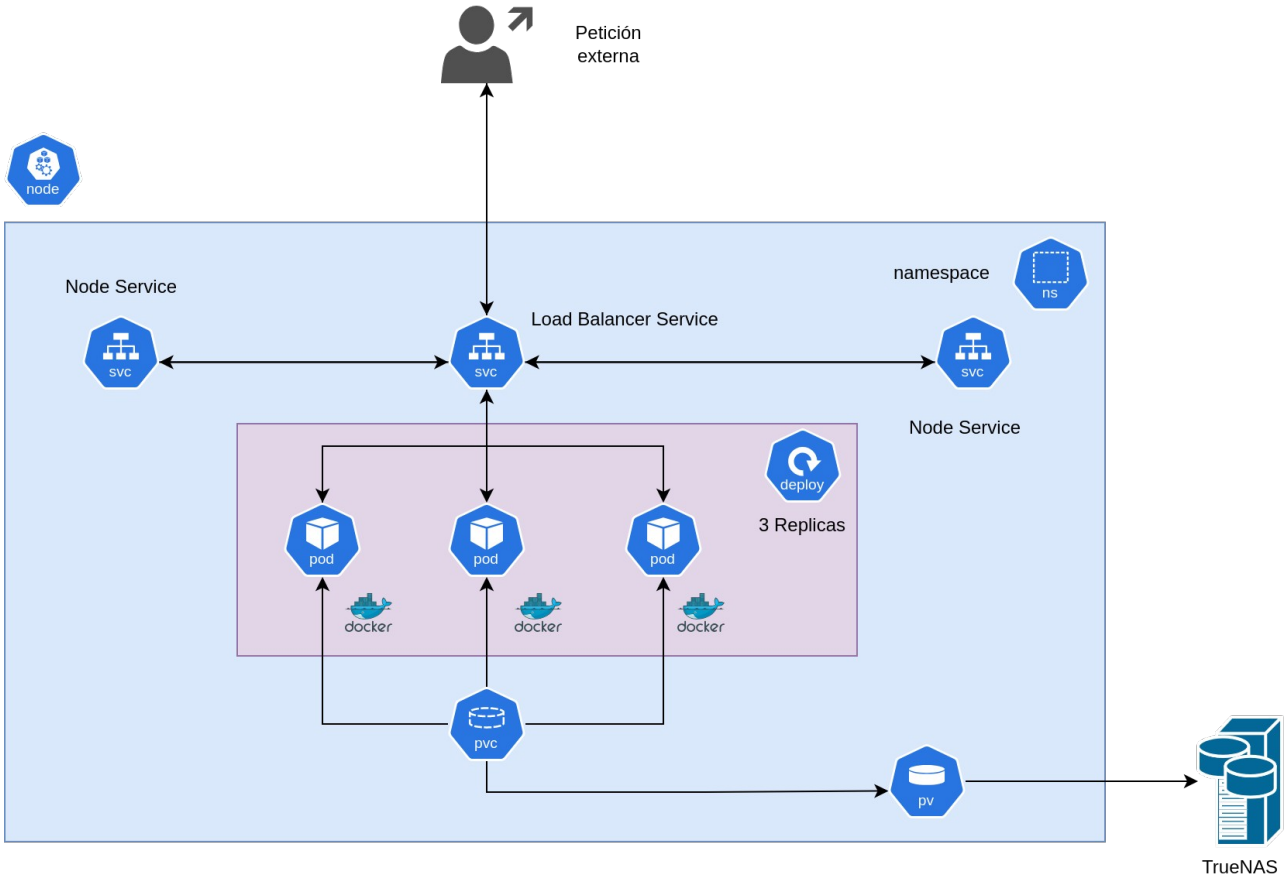
Kubernetes Docs

Un ejemplo de un servicio externo que comunica un despliegue hacia el exterior para recibir las peticiones externas que redirigirá a otros servicios internos y como resultado responderá hacia el exterior.

Estos servicios están situados en una **namespace** dentro de un **nodo worker**.



Un ejemplo pero con un volumen de datos persistentes para los datos ubicado en el servidor **TrueNAS**.



Componentes comunes (Kubernetes)

Nginx ingress controller

Nginx ingress controller será uno de los dos servicios expuestos de cara al exterior dentro de lo que es el **Clúster** de **kubernetes**.

Su función es la de actuar como **load balancer** frente a peticiones **http** o **https** redireccionándolas a servicios internos dentro del clúster de kubernetes que resuelvan este tipo de peticiones.

Nginx Ingress Controller

OpenLdap

OpenLdap será nuestro **servicio de directorio** siendo uno de los dos servicios expuestos de cara al exterior dentro del **clúster** de **kubernetes**.

Su función es **almacenar** y controlar los usuarios,grupos,equipos entre otros objetos de red del dominio.

Se encargará de la **autentificación** de ciertas aplicaciones.

Este servicio dispondrá de un **volumen persistente** donde se almacenarán los datos de usuarios…

OpenLdap Docs

MySQL operator

MySQL operator es una herramienta que administra un **MySQL InnoDB Clúster** dentro de **kubernetes.**

Un **operador** es un software que se ejecuta dentro del clúster de kubernetes y el operador interactúa con la **API** de kubernetes para observar los recursos y servicios para ayudar a kubernetes con la gestión del ciclo de vida.

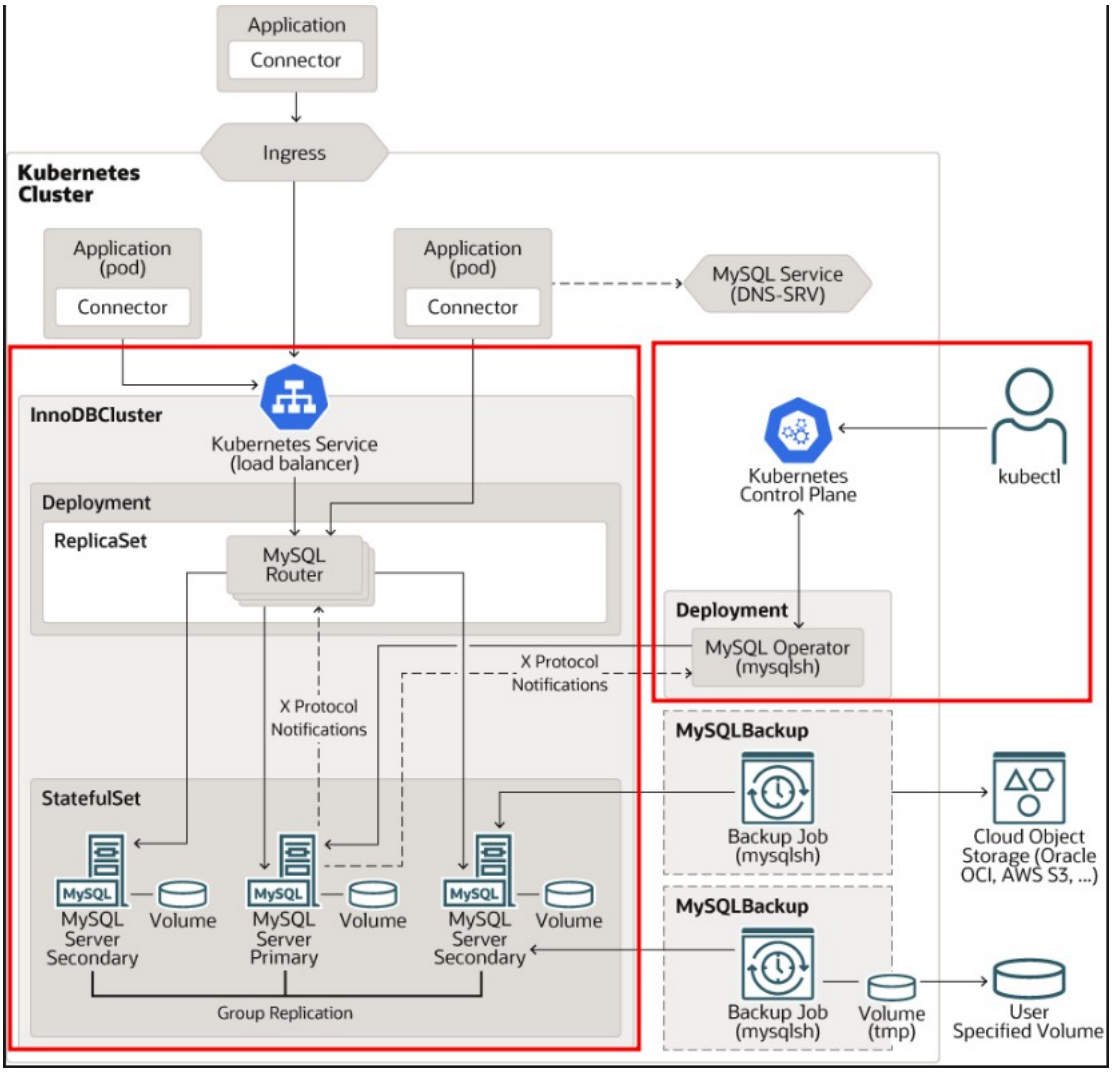
La idea de alta disponibilidad continúa aquí con un clúster situado en un **namespace** aparte en kubernetes donde este está formado por:

- **MySQL Router** → Los **enrutadores MySQL** son servicios sin estado que enrutan la aplicación al primario actual o a una réplica, según la elección de la aplicación. El operador puede aumentar o reducir la cantidad de enrutadores según lo requiera la carga de trabajo del clúster.

- **MySQL Operator (mysqlsh)** → Es la **shell** con la que nos conectamos a **MySQL InnoDB Clúster** para realizar las tareas de administración y mantenimiento dentro de **kubernetes**.
- **StatefulSet** → Este gestiona los **pods** y asigna el volumen persistente de almacenamiento correspondiente. Cada pod administrado por este **StatefulSet** ejecuta múltiples contenedores. Varios proporcionan una secuencia de pasos de inicialización para preparar la configuración del servidor **MySQL** y el directorio de datos, y luego dos contenedores permanecen activos para el modo operativo. Uno de esos contenedores (llamado 'mysql') ejecuta el servidor MySQL, y el otro (llamado 'sidecar') es un **sidecar** de **kubernetes** responsable de la gestión local del nodo en coordinación con el propio operador.

Estos son los apartados donde nos centraremos:

MySQL Operator Docs



Admin domain tool

Distribución de la app

La aplicación es una herramienta destinada a los administradores del dominio por lo tanto nada más entrar...

Login

En el **login** tendremos que **autentificarnos** mediante un **usuario** y **contraseña** donde estas credenciales las tendrá que **verificar** el servidor del dominio **(OpenLdap)**, el usuario tendrá permisos para acceder perteneciendo este al grupo de **Administradores**, cualquier otra situación donde:

- El usuario no existe en **OpenLdap**.

- La contraseña no corresponde con el usuario.

- El usuario no pertenece al grupo de **Administradores**.

Será rechazada.

Dashboard

Una vez autentificados nos encontraremos en la **dashboard** donde con **grafana** obtendremos y representaremos las métricas de rendimiento de cada uno de los servidores del dominio además conectaremos las tecnología de grafana con la api de kubernetes para también obtener y representar las métricas de los despliegues.

En la parte izquierda habrá una barra vertical donde tendremos los siguientes apartados donde si clicamos en…

Grafana Docs

Usuarios

Obtendremos **listado** de todos los **usuarios** del dominio en formato **tabla**.

Grupos

Obtendremos **listado** de todos los **grupos** del dominio en formato **tabla**.

Equipos

Obtendremos **listado** de todos los **equipos** del dominio en formato **tabla**.

Unidades Organizativas

Obtendremos **listado** de todas las **unidades organizativas** del dominio en formato de **cartas**.

Inventario

Tendremos que **identificar** el equipo o acceder desde el apartado “equipos” seleccionando un equipo para que nos aparezca el historial,logs,características acerca del equipo identificado.

En la parte de historial podremos **modificar** los datos (borrar,actualizar y insertar historial) acerca del equipo.

En referencia a los **apartados** de **usuarios**,**grupos**,**equipos** y **unidades organizativas** podremos realizar las siguientes acciones:

- Realizar una inserción.
- Realizar una inserción masiva mediante ficheros como CSV o JSON.
- Realizar un borrado de los registros seleccionados.
- Realizar un actualizado de los registros seleccionados.

En referencia a las **tablas** tendrán las siguientes características:

- Paginación.
- Orden ascendente y descendente por columnas.
- Posibilidad de marcar los registros para posteriormente trabajar sobre ellos y botón para seleccionar todos los registros.
- Ocultación de columnas.
- Filtrado de registros a través de coincidencias por columnas.

En referencia a el **apartado equipos** habrá un desplegable con una serie de tareas a ejecutar como instalación de software,actualización,configuraciones… que se ejecutarán en los equipos del dominio seleccionados mediante **ansible.**

NextJS

NextJS es un **framework** de **React** utilizado para desarrollar el **frontend** de la aplicación descrita anteriormente.

Para las funcionalidades de las tablas se utilizará bibliotecas de componentes como **NextUI**.

Para los estilos de la aplicación se utilizará **frameworks** como **TailWindCSS** o **Bootstrap**.

NextJS Docs

NestJS

NestJS es un **framework** de **NodeJS** utilizado para desarrollar el **backend** de la aplicación.

Su función principal es la de comunicarse con **OpenLdap** para la administración de los usuarios,grupos,equipos… y mediante **TypeORM** se comunicará con **MySQL** para el inventario de equipos.

NestJS Docs