Capacitación Kubernetes

Servicio de capacitación y entrenamiento ARIU

Clase 1
Introducción

Agenda

- **Presentación**
- Temario curso
- Contenedores
- Introducción kubernetes
- Configuración de producción

Gerardo Barud
Título: Lic. en Ciencias de la Computación

Experiencia Laboral:

- Trabajo en UNSJ como SysAdmin - 8 años
- Fui Docente de la Licenciatura por 10 años
- Trabajando con Kubernetes desde hace 3 años
- Trabajo para el SIU desde hace 3 años

Correo electrónico: **gbarud@siu.edu.ar**

Leandro Adrián Gómez
Título: Ing. Sistemas - Master en Administración de empresas

Experiencia Laboral:

- Trabajo en UNICEN hace 17 años, soy Jefe de Sistemas
- Docente de Sistemas hace 17 años
- Trabajando con Kubernetes desde hace 5 años
- Trabajo para el SIU desde hace 4 años

Correo electrónico: **lgomez@rec.unicen.edu.ar**

Agenda

- Presentación
- **Temario curso**
- Contenedores
- Introducción kubernetes
- Configuración de producción

Duración: 8 clases

Actividades entre clases: Autoevaluaciones, videos y prácticas para validar conocimientos.

Objetivos:
Conocimiento básico de Kubernetes.
Creación y gestión de un cluster funcional.

Temario del Curso

Clase 1: Resumen de Docker, Introducción a Kubernetes.

Clase 2: Revisar requisitos, Iniciar cluster, instalar herramientas.

Clase 3: Descripción de recursos básicos.

Clase 4: Descripción de recursos adicionales.

Clase 5: Almacenamiento.

Clase 6: Despliegue de aplicaciones.

Clase 7: Monitoreo.

Clase 8: Actualización, Backup y Alta disponibilidad.

Conocimiento de contenedores:

- Familiaridad con Docker y sus ventajas.
- Diferencias entre contenedores y máquinas virtuales.

Experiencia con Kubernetes:

- Nivel de experiencia previa con Kubernetes.

Expectativas:

- Crear un cluster en producción.
- Primer contacto con la tecnología.
- Aumentar los conocimientos previos.

Agenda

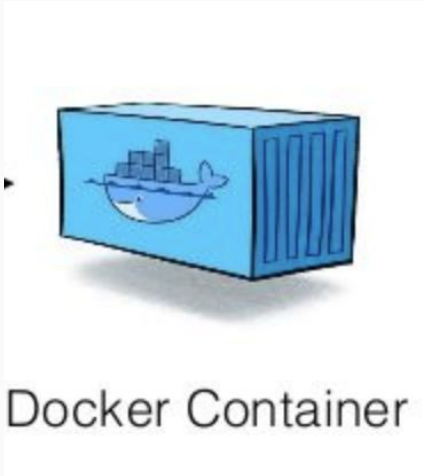
- Presentación
- Temario curso
- **Contenedores**
- Introducción kubernetes
- Configuración de producción

Asociación Redes de Interconexión Universitaria

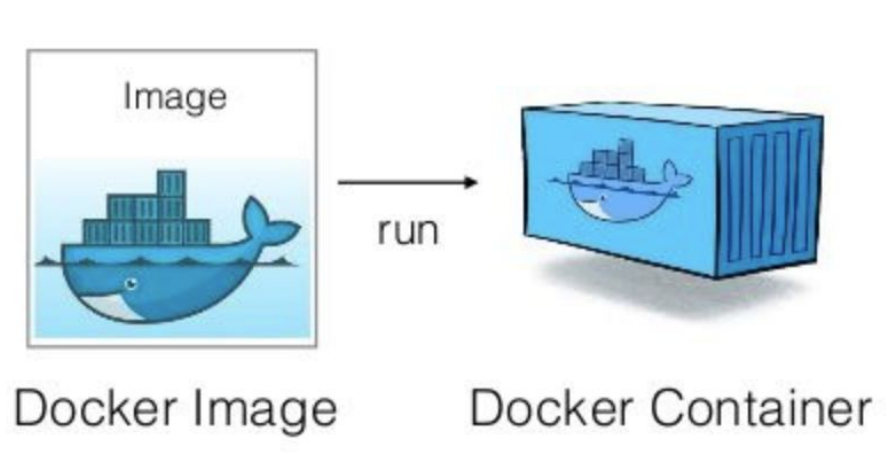
Definición de Contenedores (Docker)

Los contenedores son una tecnología de virtualización ligera que permite empaquetar aplicaciones y sus dependencias en un entorno aislado. Docker, una plataforma popular, facilita la distribución y ejecución eficiente de aplicaciones en diversos entornos al encapsularlas en unidades autónomas llamadas "imágenes".

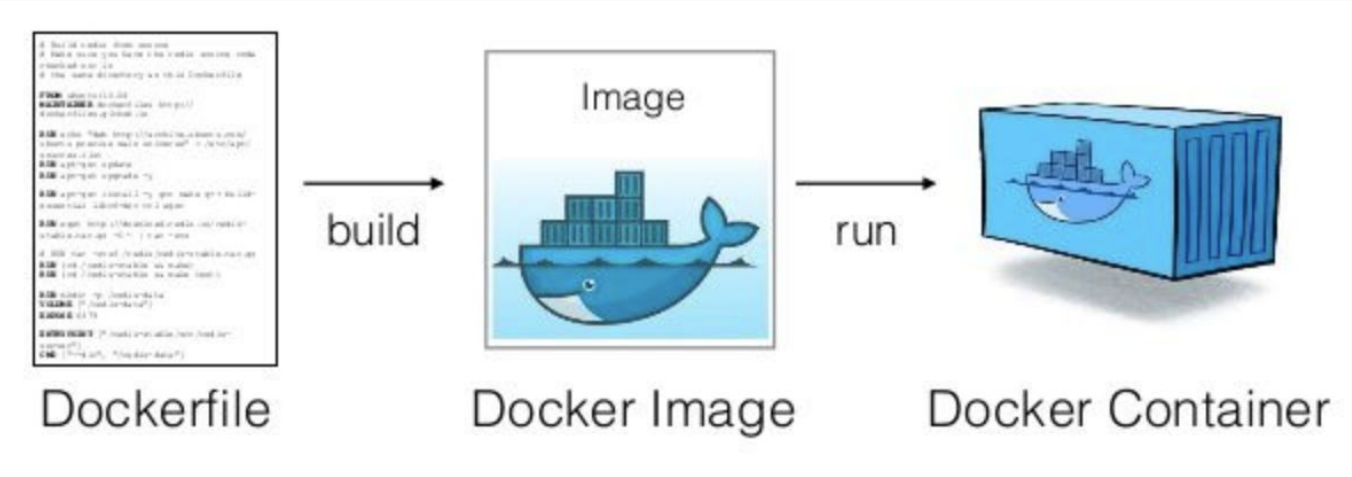
- Se crea para una o varias aplicaciones.
- Posee todo el software (librerías y binarios) para que la aplicación funcione.
- Pueden conectarse a él redes y volúmenes
- Provee “movilidad” de la aplicación.
- Es creado a partir de una imagen.



- Plantilla para la creación de un contenedor.
- Contiene toda la información necesaria para la creación.
- Cada imagen comienza desde una imagen base, y una plantilla.
- Es creada a partir de un dockerfile y archivos necesarios



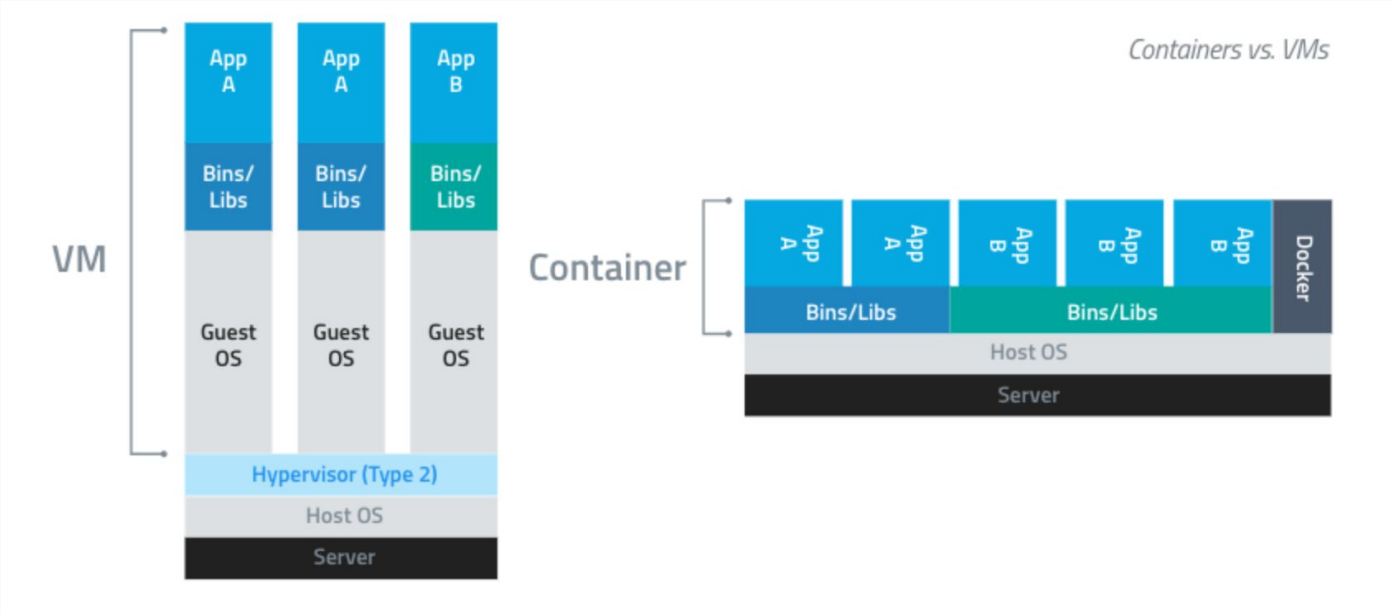
- Documento de texto que tiene todos los comandos que un usuario podría ejecutar para ensamblar una imagen (receta).



- Docker no necesita instalar un SO como las MV
  - No necesita espacio de disco para el SO
  - No utiliza memoria para ejecutar el SO
  - Reduce el tiempo de arranque.
- El servidor de Docker asigna memoria y cpu según se necesite, sin necesidad de declararlo previamente.
- Docker nativo no permite ejecutar contenedores que no sean para el sistema operativo host.
  - Para ejecutar contenedor linux sobre windows, docker utiliza una máquina virtual linux.

Si vemos los procesos ejecutando en el anfitrión (ps -ax) podremos observar cada uno de los procesos ejecutados dentro del contenedor

- Los contenedores proporcionan aislamiento de procesos a nivel del sistema operativo, mientras que las máquinas virtuales ofrecen aislamiento en la capa de abstracción de hardware.



Beneficios de usar contenedores

- **Ágil creación y despliegue de aplicaciones**
- **Desarrollo, integración y despliegue continuo**: Con sólo crear la imagen del contenedor nuevo se puede poner en producción el nuevo software desarrollado
- **Separación de tareas entre Dev y Ops**: Se puede crear el contenedor y en otro momento realizar la puesta en producción
- **Consistencia entre los entornos de desarrollo, pruebas y producción**
- **Portabilidad entre nubes y distribuciones**
- **Microservicios distribuidos, elásticos, liberados y débilmente acoplados**
- **Aislamiento de recursos**: Hace el rendimiento de la aplicación más predecible

Agenda

- Presentación
- Temario curso
- Contenedores
- **Introducción kubernetes**
- Configuración de producción

A R I U

Asociación Redes de Interconexión Universitaria

¿Qué es Kubernetes?

Kubernetes (K8s) es una plataforma de contenedores que orquesta la infraestructura de cómputo, redes y almacenamiento buscando que el estado del cluster se mantenga según fue declarado

Qué hace de Kubernetes una plataforma

- Permite definir componentes y herramientas que hacen más fácil el desplegar, escalar y administrar aplicaciones
- API accesible para desarrolladores y usuarios finales

Que NO es Kubernetes

- No hace deployment ni compilación de código fuente.
- No provee servicios en capa de aplicación como middleware, bases de datos o sistemas de almacenamiento.
- No adopta un sistema o lenguaje de configuración específico.

Componentes de Kubernetes

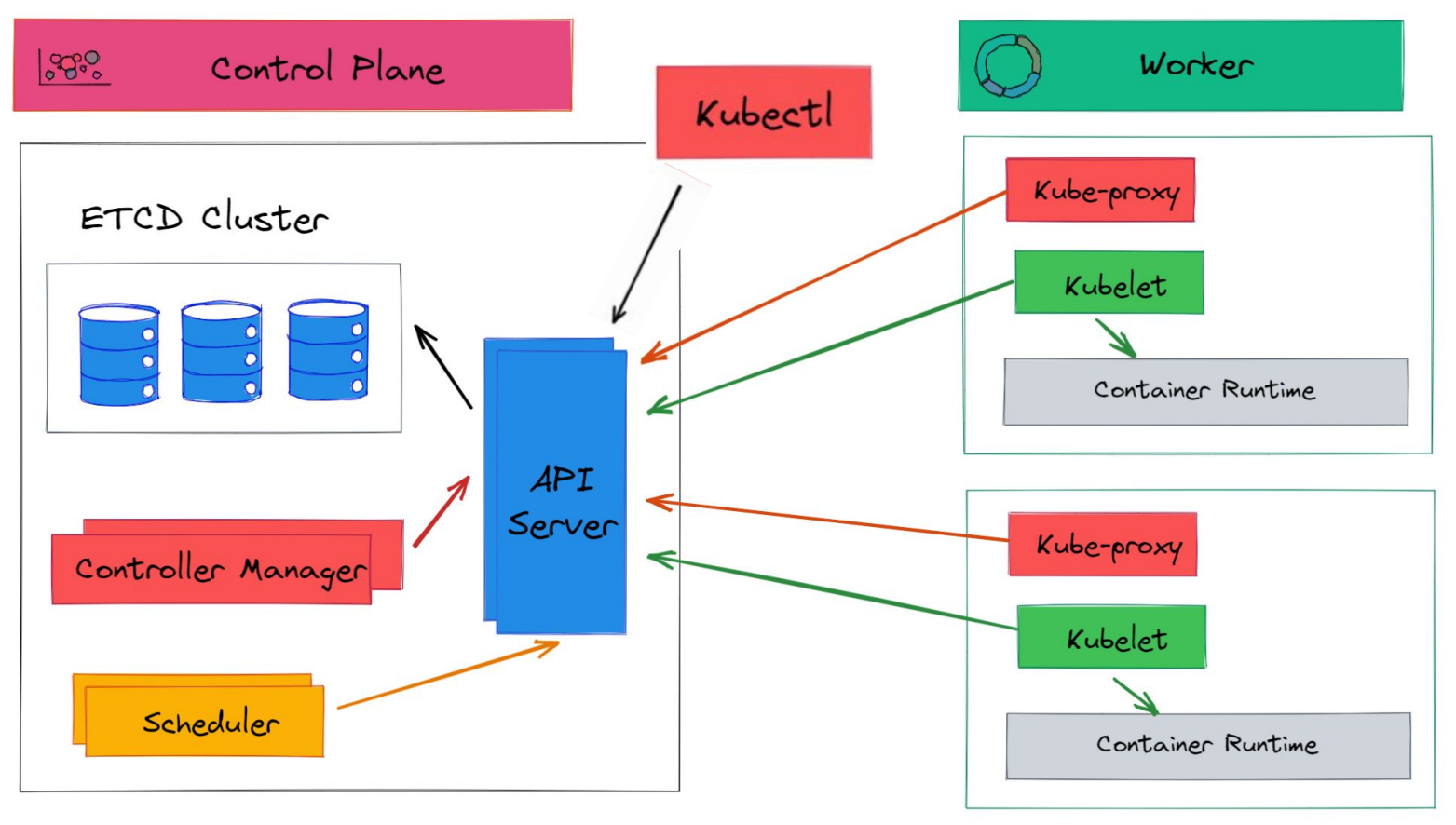
Control Plane

- Administra y coordina las operaciones en el clúster de Kubernetes
- Permite configurar múltiples instancias para garantizar tolerancia a fallos y redundancia

Worker

- Ejecuta las cargas de trabajo y los contenedores en el clúster
- Permite agregar o eliminar nodos según sea necesario para adaptarse a la carga de trabajo y aumentar la capacidad del clúster

Componentes del plano de control



Componentes del plano de control

- **Kube-apiserver:** Expone la API de Kubernetes
- **Etcd:** Almacena de forma segura y distribuida la configuración y el estado del clúster.
- **Kube-scheduler:** Asigna las cargas de trabajo a los nodos disponibles en base a políticas de asignación.
- **Kube-controller-manager:** Gestiona el ciclo de vida de los objetos de Kubernetes y realiza verificaciones de estado (Mantiene las réplicas, levanta nodos caídos, etc)

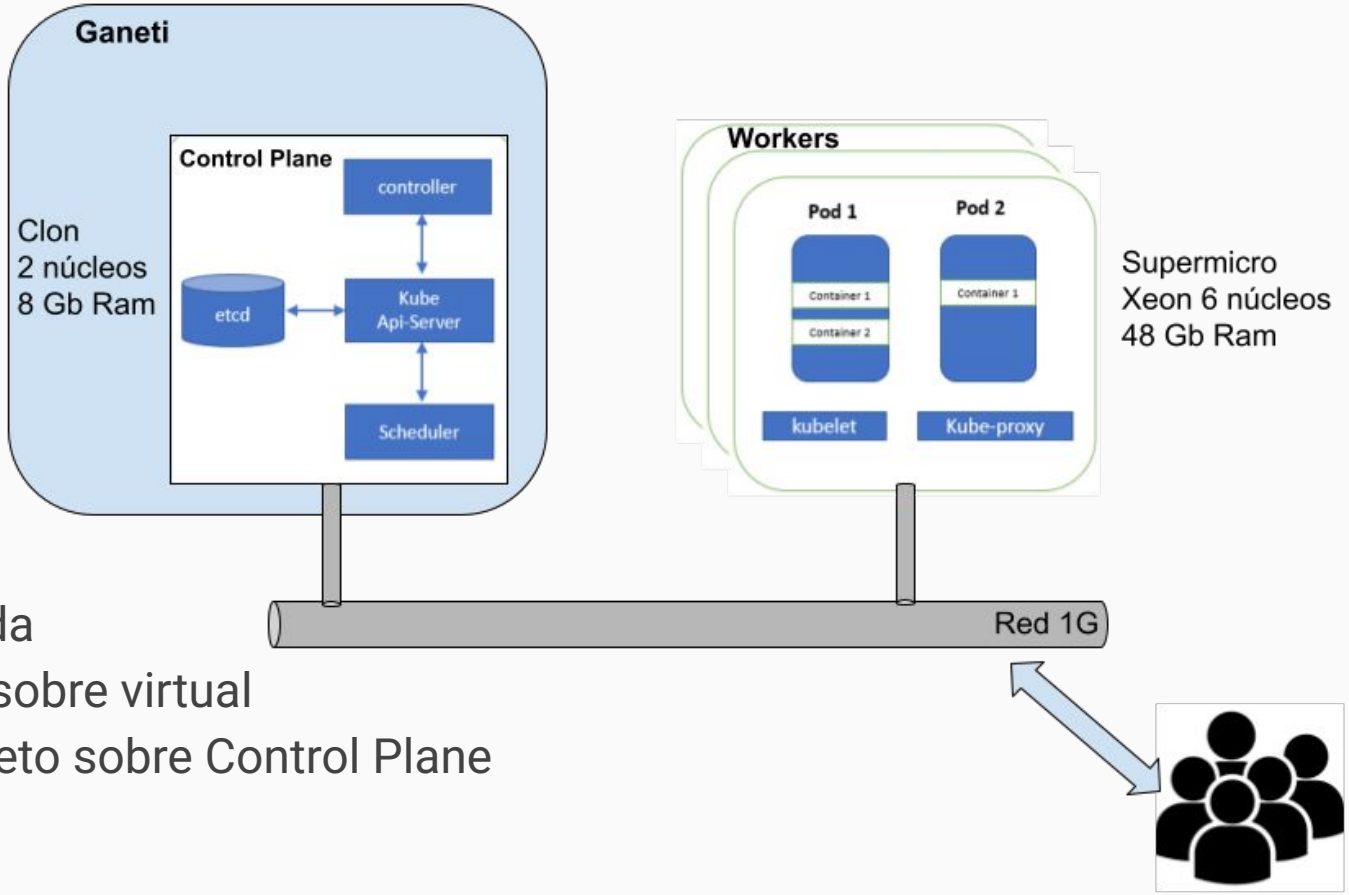
Componentes del Worker

- **Kubelet:** Agente que se ejecuta en cada nodo y administra los contenedores y los recursos del nodo.
- **Container Runtime:** Proporciona el entorno de ejecución para los contenedores (ej. Docker, Containerd).
- **Kube-proxy:** Facilita la comunicación de red entre los servicios dentro del clúster.

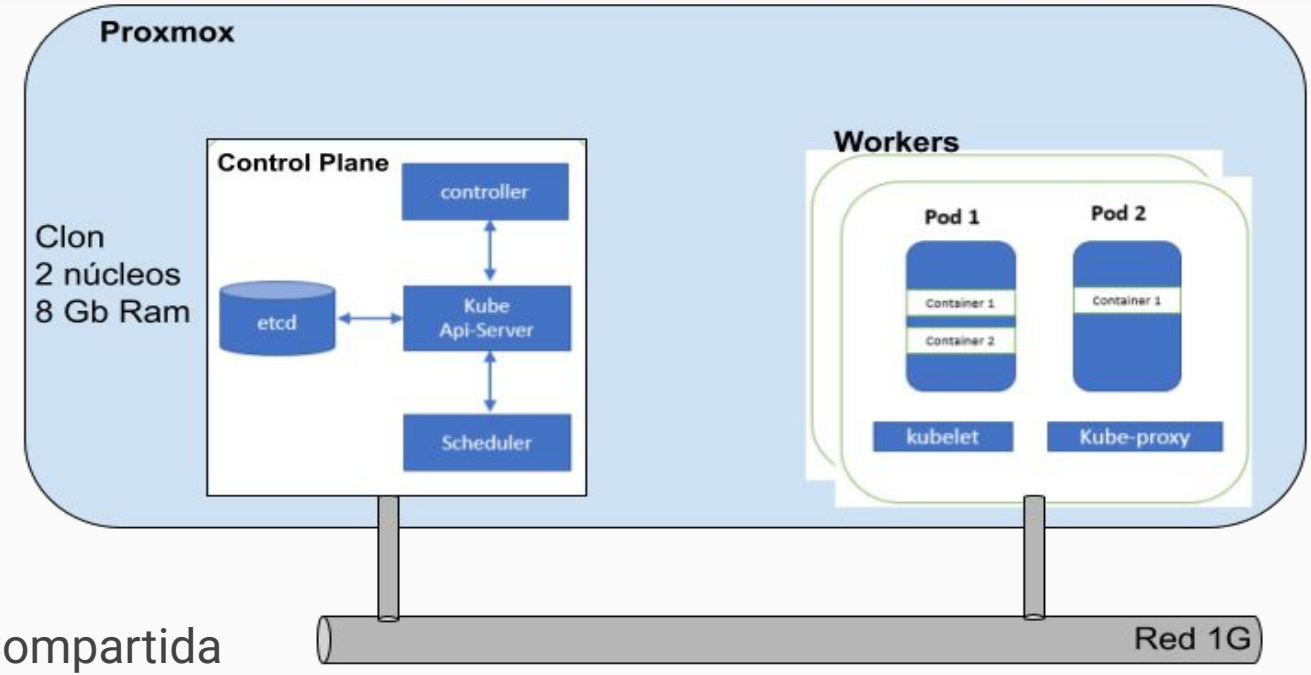
Agenda

- Presentación
- Temario curso
- Contenedores
- Introducción kubernetes
- **Configuración de producción**

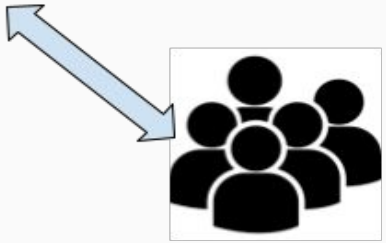
Experiencia - Inicios Unicen



- Red Compartida
- Control Plane sobre virtual
- Backup completo sobre Control Plane



- Red Compartida
- Control Plane sobre virtual
- Workers sobre virtuales



Autoevaluación Doker

Realizar Autoevaluación Docker moodle

Objetivos:

- Validación de Conocimientos
  - Comprobar comprensión y retención
  - Identificar fortalezas y debilidades
- Reflexión y Pensamiento Crítico
  - Reflexionar sobre los conceptos aprendidos
  - Evaluar progreso y desempeño personal

Instalación y Preparación del Servidor/s

Pasos a Seguir y Requisitos

Ejecutar el Script de Instalación y Preparación

- Script para Debian o ubuntu: instalar-k8s-debian12-ubuntu2404.sh
- Descargar en el servidor los archivos para continuar la instalación la siguiente clase.

Elección del Número de Servidores

- Opción 1: Un único servidor:
  - Servirá tanto como Control Plane y Worker
- Opción 2: Múltiples servidores
  - Control Plane inicial: Mínimo necesario
  - Control Plane adicional: Sumar redundancia
  - Workers: Los servidores restantes se utilizarán exclusivamente como Workers.

Requisitos para inicializar el cluster

- Repasar Moodle
- Revisar Requisitos de instalación

ARIU

Asociación Redes de Interconexión Universitaria