RELACIÓN DEL MUNDO REAL CON NUESTRO PROYECTO

aws | amazon | NETFLIX | Google Cloud

TECNOLOGÍAS CLOUD-NATIVE UTILIZADAS EN PRODUCCIÓN

Escalabilidad • Alta Disponibilidad • Automatización • Resiliencia • Observabilidad

KubeBackup360

DevOps Lab

KUBERNETES

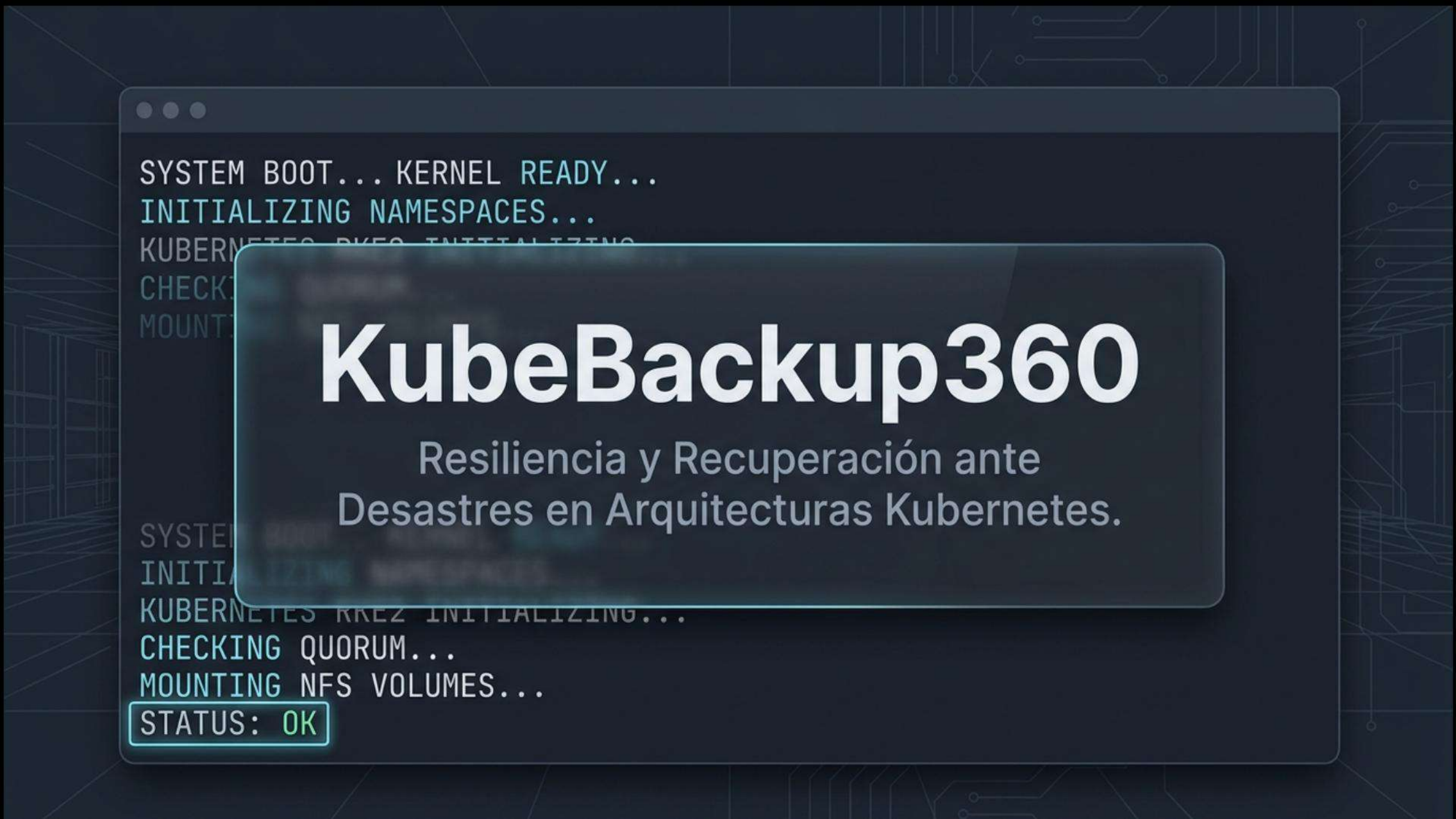
ALMACENAMIENTO PERSISTENTE

BACKUPS Y RECUPERACIÓN

MONITORIZACIÓN

SEGURIDAD Y RESILIENCIA

“Traemos las mejores prácticas del mundo real a nuestro laboratorio”


SYSTEM BOOT... KERNEL READY...
INITIALIZING NAMESPACES...
CHECKING QUORUM...
MOUNTING NFS VOLUMES...
STATUS: OK
KubeBackup360
Resiliencia y Recuperación ante
Desastres en Arquitecturas Kubernetes.

El Desafío: La Naturaleza Efímera de Kubernetes

La Promesa (Poder)

Escalabilidad automática, orquestación distribuida, autorecuperación de servicios (Self-Healing).

La Realidad (Vulnerabilidad)

Contenedores efímeros. Si un nodo cae o un Namespace se elimina por error, la pérdida de datos es catastrófica sin un almacenamiento externo y un Plan de Recuperación ante Desastres (DRP).

KubeBackup360: De Efímero a Resiliente

El Enfoque KubeBackup360

Requisito: Orquestación	Tecnología: RKE2 / Rancher	Función: Plano de control distribuido.
Requisito: Persistencia	Tecnología: NFS	Función: Almacenamiento externo compartido.
Requisito: DRP y Backups	Tecnología: Velero + MinIO	Función: Repositorio estanco compatible con S3.
Requisito: Observabilidad	Tecnología: Prometheus + Grafana	Función: Telemetría y auditoría proactiva.

Topografía de la Arquitectura

Capa de Red

Segmentada, Router/NAT interno, DNS interno (CoreDNS).

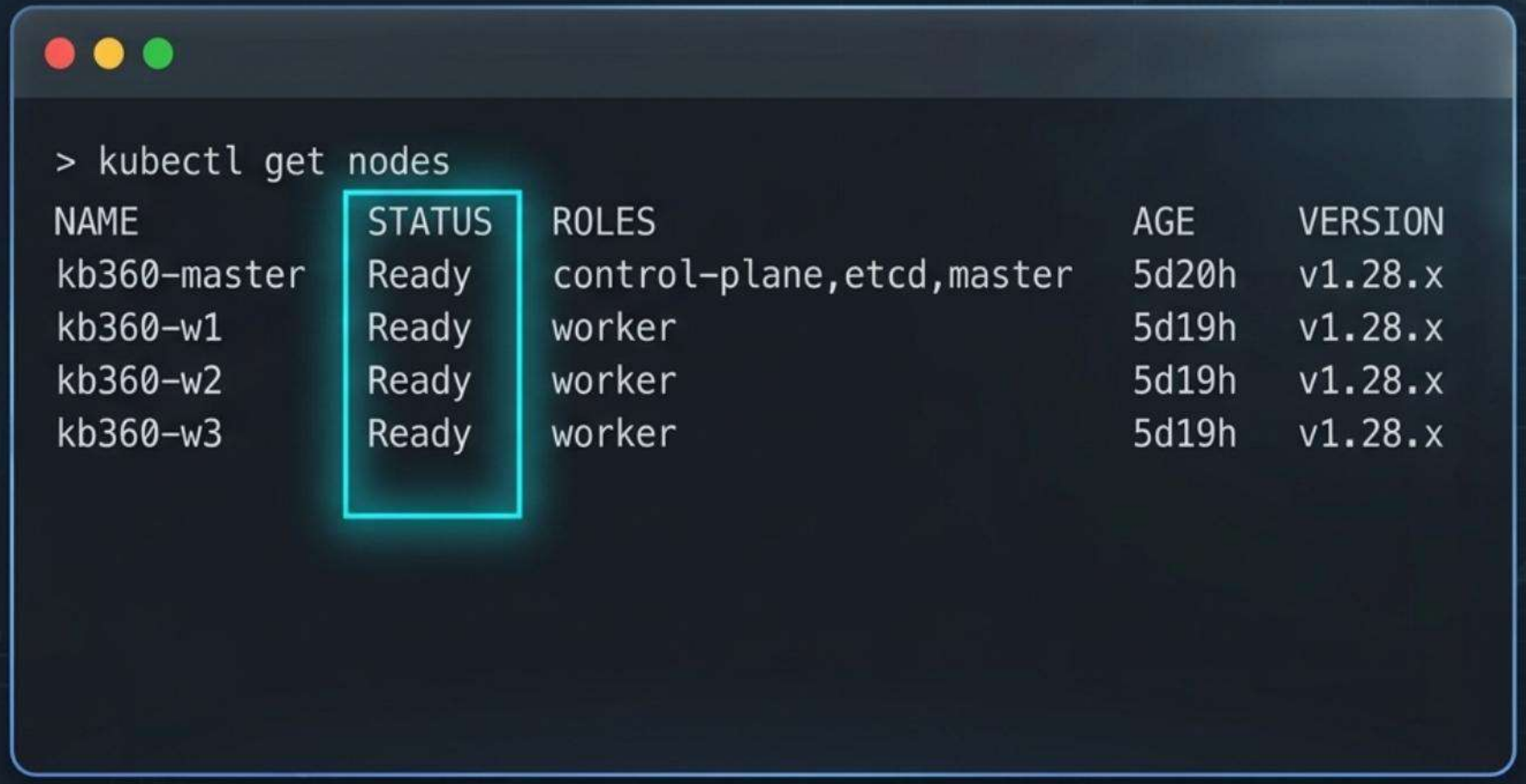
Capa de Computación

1 Nodo Master, 3 Nodos Worker sobre Ubuntu Server virtualizado.

Capa de Continuidad

Backup orquestado y almacenamiento S3 aislado.

Pilar 1: Infraestructura (El Clúster en Producción)



Distribución automática y quórum validado. Entorno Kubernetes estable.

Pilar 1: Contenerización y Despliegue (Docker -> K8s)

Panel 1: Construcción (Docker)

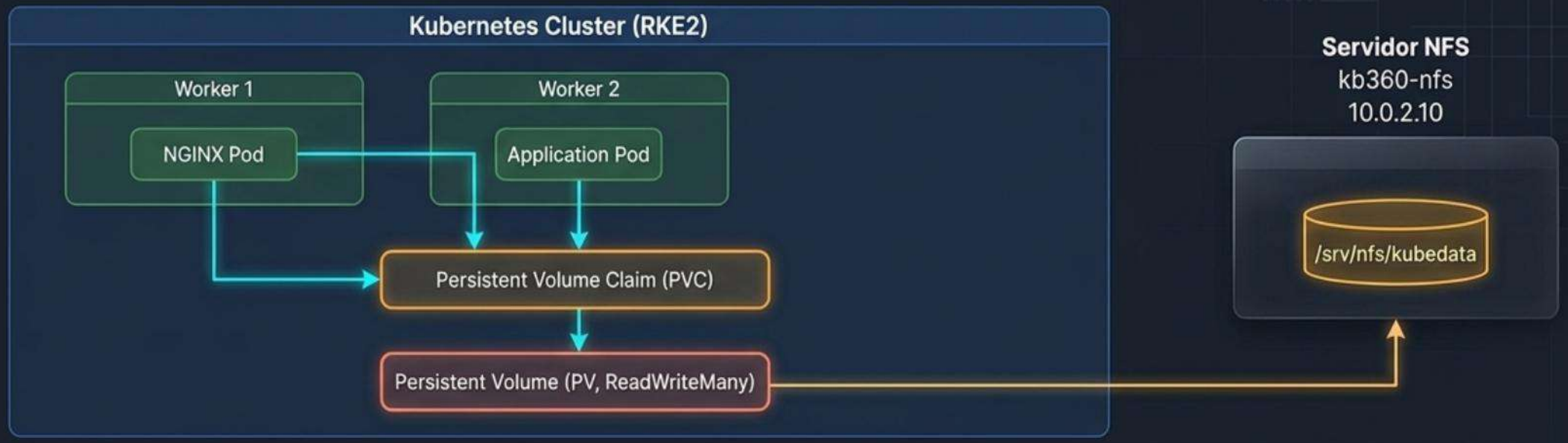
```
> docker ps -a
CONTAINER ID   IMAGE                          STATUS
a1b2c3d4e5f6   usuario/kubebackup360-web:v1   Up 2 hours
```

Panel 2: Ejecución (Kubernetes)

```
> kubectl run kubebackup360-web --image=usuario/kubebackup360-web:v1

> kubectl get pods -o wide
NAME                        READY   STATUS    IP           NODE
kubebackup360-web-7b9c...   1/1     Running   10.42.1.15   kb360-w2
```

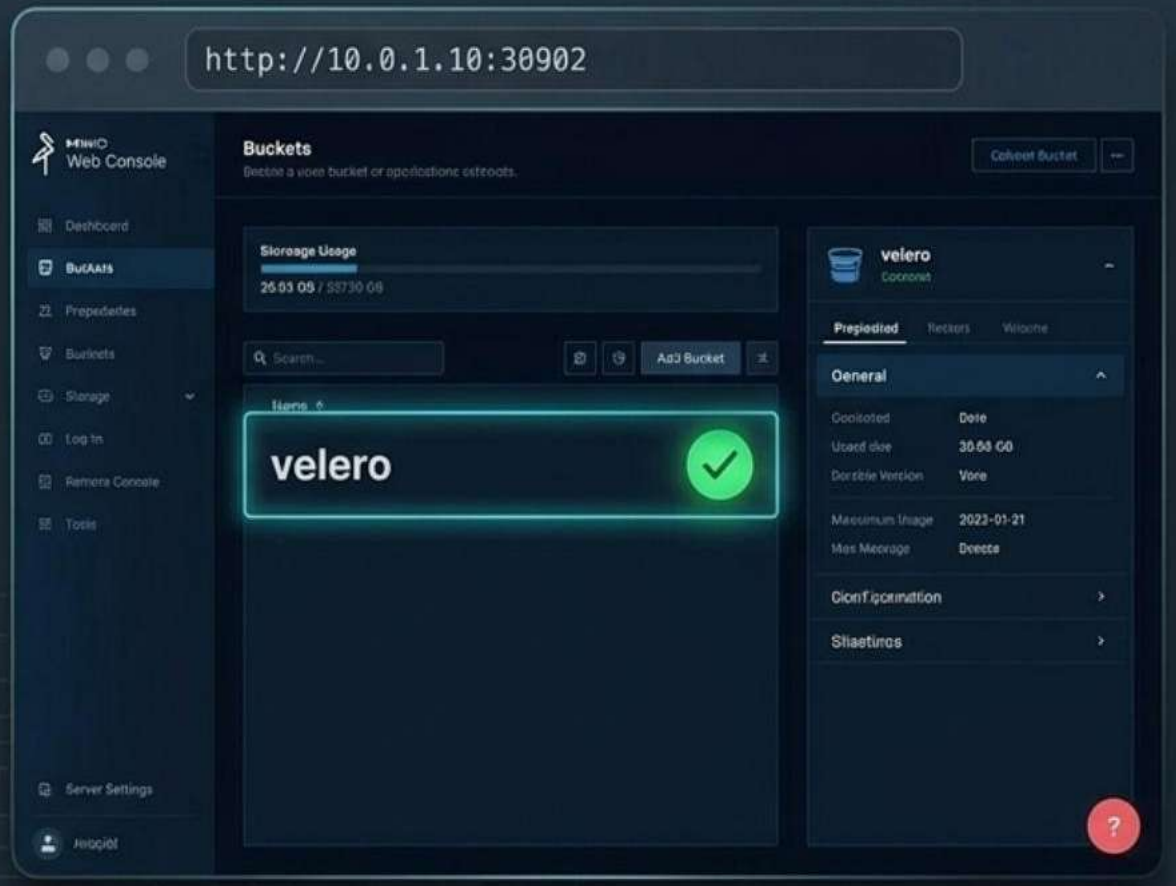
Pilar 2: Desacoplando el Almacenamiento (Persistencia NFS)




```
> kubectl exec -it pod-nfs-test -- cat /data/test.txt
[INFO]: Conexión exitosa al volumen persistente NFS.
```

Insight Clave: Si el contenedor se destruye, los datos permanecen intactos en el servidor central NFS.

Pilar 2: Repositorio S3 Estanco (MinIO)



¿Por qué MinIO?

Emula la API de Amazon S3 localmente sin coste.

Función:

Caja fuerte digital y destino estanco para los backups generados por la infraestructura.

Pilar 2: El Motor DRP (Velero)


```
> velero install --provider aws --plugins velero/velero-plugin-for-aws:v1.10.0 --bucket
velero --secret-file ./credentials-velero --use-node-agent ...
> velero schedule get
NAME           STATUS    SCHEDULE       LAST BACKUP
daily-backup   Enabled   0 */6 * * *    23m ago
```

Gobernanza DRP Automatizada.

La Prueba de Fuego: Simulación de Desastre

```
> kubectl delete namespace produccion
namespace 'produccion' deleted
> kubectl get pods -n produccion
No resources found in produccion namespace.
```

Pérdida Crítica Simulada. Eliminación controlada de Deployments, Pods, Services y rutas lógicas. Estado del sistema: **Caído.**

La Prueba de Fuego: Restauración y Sanación

```
> velero restore create --from-backup daily-backup-20231024
Restore request 'restore-disaster-01' submitted successfully.
> velero restore get
NAME                   STATUS      ERRORS   WARNINGS   CREATED
restore-disaster-01    Completed   0        0          1m ago
```

100% de la infraestructura y metadatos recuperados desde MinIO.
Continuidad de negocio garantizada.

Pilar 3: Telemetría y Acceso Seguro (Túneles SSH)


```
> ssh -L 30080:10.0.1.10:30080 -L 30090:10.0.1.10:30090 usuario@192.168.0.75
```

En lugar de exponer la monitorización abiertamente, utilizamos un puente seguro (Port-Forwarding SSH) para enrutar el tráfico de Prometheus y Grafana hacia el cliente, replicando prácticas estrictas de seguridad empresarial.

Pilar 3: El Pulso del Clúster en Tiempo Real

Insight Clave: La observabilidad transforma la administración reactiva en proactiva. Permite validar visualmente la redistribución de cargas de trabajo durante las pruebas de failover.

Síntesis: El Ecosistema KubeBackup360

Operación (Pilar 1)
Nodos RKE2 ejecutando contenedores Docker.

Auditoría (Pilar 3)
Prometheus extrayendo telemetría –> Grafana visualizando el rendimiento global.

Protección (Pilar 2)
Datos escritos en NFS -> Capturados por **Velero** -> Aislados en MinIO S3.

Conclusión 1: Infraestructura y Contenedores

Lo que queríamos hacer	Diseñar e implementar un entorno de laboratorio virtualizado, segmentado y seguro, capaz de orquestar contenedores emulando una infraestructura empresarial real.
Lo que hemos hecho	Desplegamos un clúster Kubernetes mediante RKE2 (1 Master, 3 Workers). Construimos imágenes personalizadas en Docker, las distribuimos vía Docker Hub y las instanciamos usando `Deployments` y `Services`.
Resultado obtenido	Un clúster 100% operativo con distribución automática de cargas. Se verificó la tolerancia a fallos, la auto-reparación de Pods y la escalabilidad horizontal con éxito total.

Conclusión 2: Persistencia y Backups DRP

Lo que queríamos hacer	Garantizar que los datos sobrevivieran a la eliminación de contenedores y crear un Plan de Recuperación ante Desastres (DRP) robusto y automatizado.
Lo que hemos hecho	Implementamos almacenamiento externo `NFS` vinculado mediante Persistent Volumes. Desplegamos `MinIO` como repositorio S3 y configuramos `Velero` para automatizar copias de seguridad cada 6 horas. Ejecutamos un simulacro de desastre controlado.
Resultado obtenido	El sistema recuperó el 100% de los metadatos y volúmenes persistentes tras el borrado crítico (`ERRORS: 0` en `Velero`). Se validó la resiliencia absoluta del entorno.

Conclusión 3: Monitorización

Lo que queríamos hacer	Transformar la administración reactiva en una gestión proactiva, obteniendo visibilidad total sobre el estado de salud, rendimiento y consumo del hardware.
Lo que hemos hecho	Integramos `Prometheus` como motor de recolección de métricas y `Grafana` como plataforma visual. Desarrollamos dashboards personalizados con consultas `PromQL` y aseguramos el acceso mediante túneles SSH.
Resultado obtenido	Una plataforma de telemetría en tiempo real que permitió validar visualmente el comportamiento de la red, los nodos y los Pods antes, durante y después del proceso de Disaster Recovery. Proyecto auditado y validado.