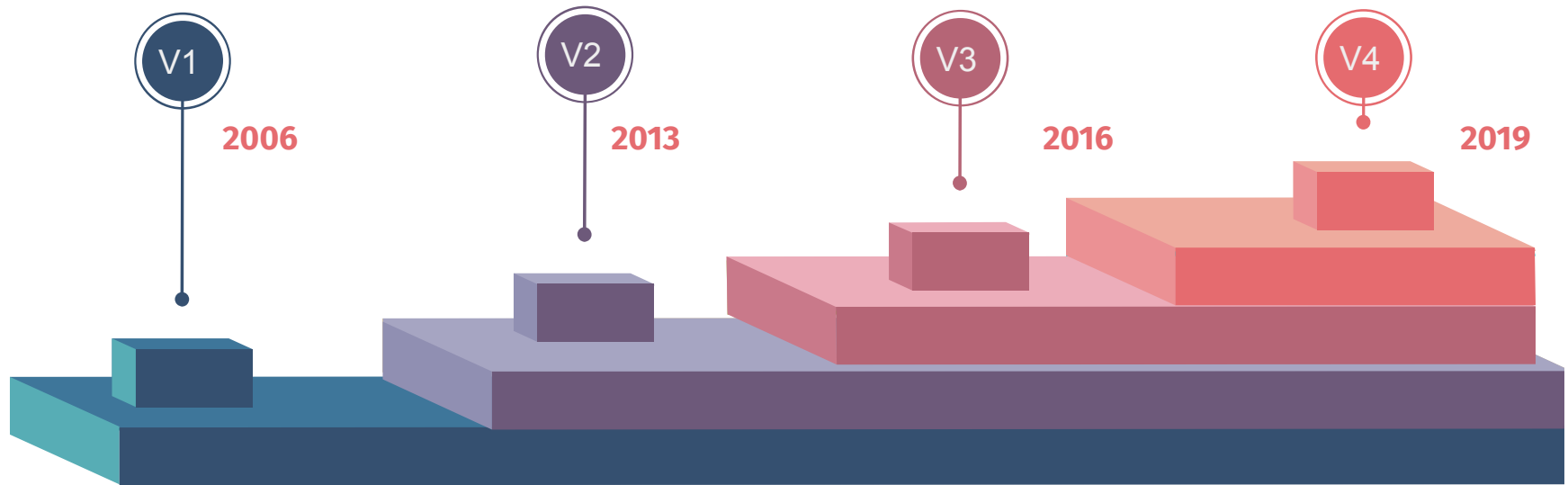


Elabs v3. Provisión de servicios efímeros de computación
personalizados sobre un almacenamiento persistente.
Jupyter Hub y Kubernetes.



Jose Luis Castro López
Luis Guillermo Torres Sanjuán
Servicio de Informática
Estación biológica de Doñana

Evolución de la arquitectura tecnológica en EBD-ICTS-RBD



EBD-ICTSv1

no elabs

EBD-ICTSv2

elabs v1: herramientas de propósito general; colaboración; publicación

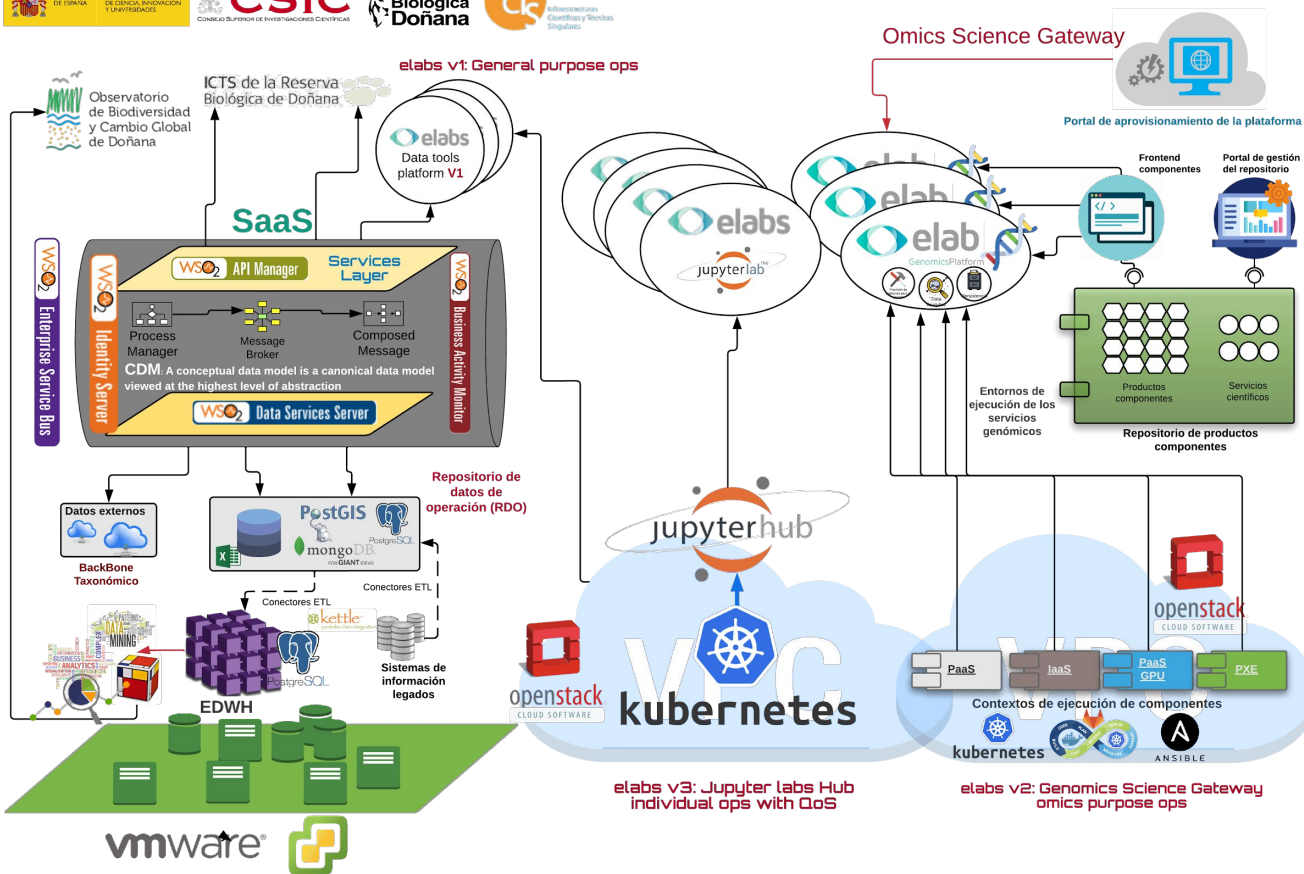
EBD-ICTSv3

elabs v2: herramientas específicas; colaboración

EBD-ICTSv4

elabs v3: herramientas de propósito general; entornos personales

Arquitectura tecnológica de la EBD y la ICTS-RBD



Por qué Jupyter

WebApp

Además, su arquitectura hace posible despliegues personales

Estándar

Es un estándar para el desarrollo del scripting sobre notebooks

Seguro

Los accesos pueden gestionarse desde un único endpoint contra LDAP

JupyterHub

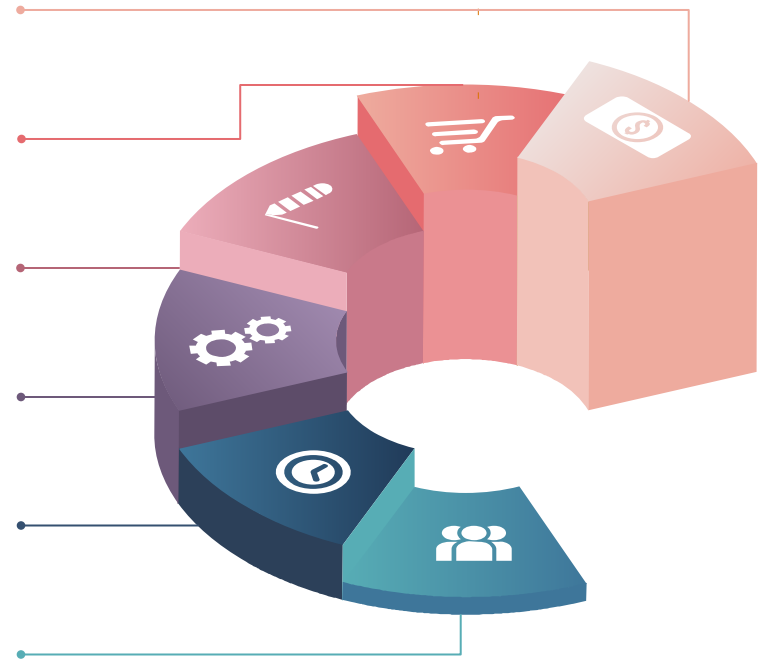
Servicio que permite ofrecer Jupyter-labs a grandes grupos de usuarios

Avanzado

Oferta una arquitectura de despliegue sobre kubernetes

Escalable

Es susceptible de incorporar nuevas herramientas (!)



Por qué Jupyter en Kubernetes

Arquitectura de contenedores

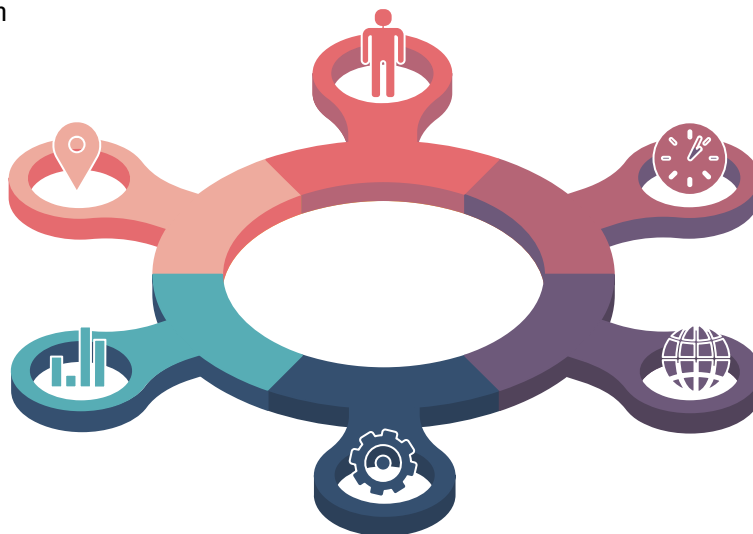
Mejor vincular la instancia a un contenedor/pod que utilizar una provisión local

Autoescalado

JupyterHub Kubernetes Spawner enables JupyterHub to spawn single-user notebook servers on a Kubernetes cluster.

Externalización

Los entornos de ejecución pueden estar en varias nubes. Es posible diseñar estrategias de workloads



Estándar de facto

(Lo que aprendas con k8s te va a ayudar cuando los Reyes Magos te regalen licencias de Openshift)

Prevenir morir de éxito

Si tu plataforma docker o tu cluster HPC comienza a ser utilizado a “jierro”, vas a tener problemas...

Gestión de recursos

En la configuración del Pod se pueden controlar fácilmente en las secciones Requests y Limits del fichero de despliegue YAML

Elementos clave de JupyterHub sobre k8s



Hub: administra las cuentas de usuarios, autenticación y coordina la creación de los **Single User Notebooks Servers** para cada usuario utilizando un **Spawner**.



Single-User Notebook Server: se despliega un servidor Jupyter Notebook para cada usuario autorizado en el sistema al inicio de cada sesión. El objeto que gestiona servidores notebook de un solo usuario se denomina **Spawner**.



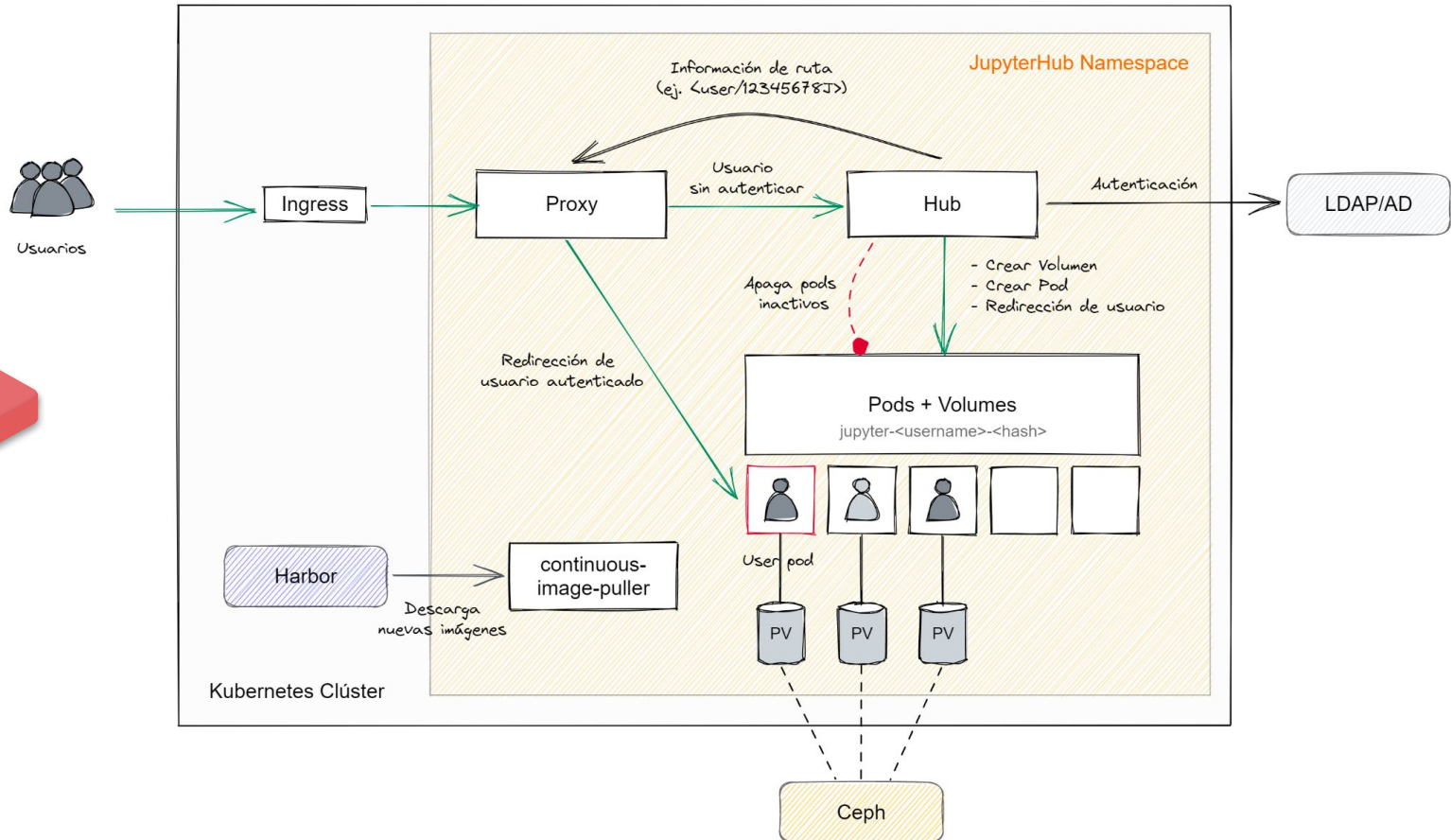
Proxy: es la parte pública de JupyterHub que utiliza un proxy dinámico para enrutar las solicitudes HTTP a los servidores Hub y Notebook de cada usuario.



Continuos-image-puller: se ejecuta como daemonset en todos los workers del clúster y se encarga de asegurar que la última imagen de los notebooks de usuario esté siempre descargada en el nodo para agilizar la creación de instancias de usuario.

Por qué JupyterHub (en k8s)

El esquema de comunicación entre servicios de nuestro sistema JupyterHub-k8s es tal que así:



El resultado

The screenshot shows a JupyterLab interface. On the left is a file browser with a search bar and a table of files. The main area displays a 'Specify' dialog with three sections: Notebook, Console, and Other. The right sidebar contains panels for VARIABLES, CALLSTACK, BREAKPOINTS, SOURCE, and KERNEL SOURCES.

Name	Last Modified
arbolAves.csv	11 days ago
ArbolAvesConSu...	10 days ago
ArbolAvesConSu...	10 days ago
ArbolAvesConSu...	10 days ago
ArbolMamiferos.csv	7 days ago
citations.dmp	11 days ago
delnodes.dmp	11 days ago
division.dmp	11 days ago
excludedfrontype...	11 days ago
fillingSubFamili...	10 days ago
fillingTree.ipynb	7 days ago
fullnamelineage.d...	11 days ago
genocode.dmp	11 days ago
host.dmp	11 days ago
images.dmp	11 days ago
merged.dmp	11 days ago
names.dmp	11 days ago
new_taxdump.tar...	11 days ago
nodes.dmp	11 days ago
rankedlineage.dmp	11 days ago
taxidlineage.dmp	11 days ago
typematerial.dmp	11 days ago
typeotype.dmp	11 days ago

Specify

Notebook

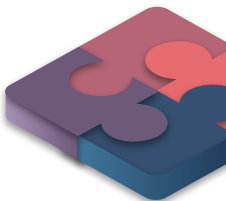
- Python 3 (pyKernel)
- Julia 1.8.1
- OpenRefine [*]
- R
- RStudio [*]

Console

- Python 3 (pyKernel)
- Julia 1.8.1
- R

Other

- Terminal
- Text File
- Markdown File
- Julia File
- Python File
- R File
- Show Contextual Help



Características de un elab v3: los (demás) conceptos



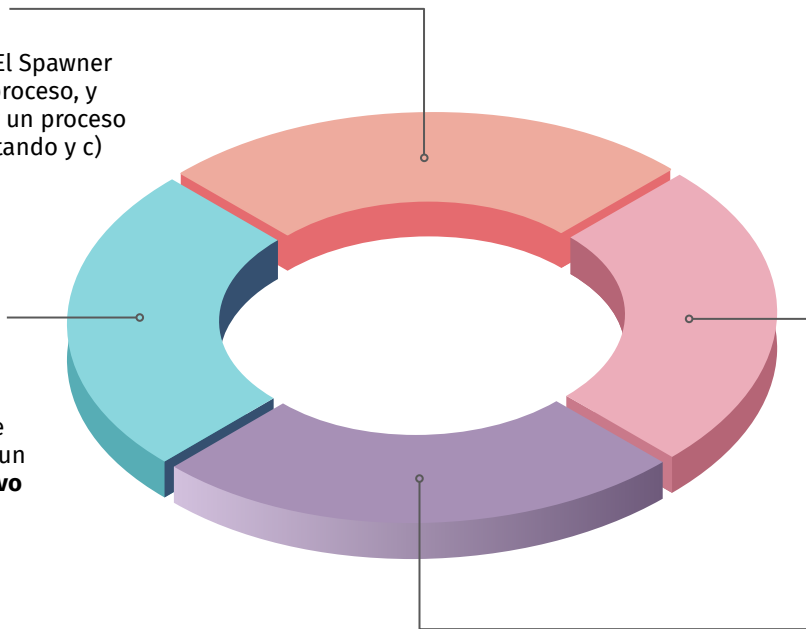
Gestión: spawning y culling en JupyterHub (y k8s)

spawner

Inicia cada jupyter lab de un solo usuario. El Spawner representa una interfaz abstracta para un proceso, y debe poder realizar tres acciones: a) **iniciar** un proceso b) **sondear** si un proceso aún se está ejecutando y c) **detener** un proceso

culling

Un servidor se puede apagar mediante la eliminación selectiva (**culling**). De forma predeterminada, el servicio de selección de JupyterHub está configurado para replegar un servidor de usuario que haya estado **inactivo durante una hora**.



kubespawner

En Kubernetes, un **PersistentVolume** (PV) representa el **disco duro**. **KubeSpawner** creará un **PersistentVolumeClaim** que solicita un PV desde la nube. De forma predeterminada, eliminar el PVC hará que la nube elimine el PV.

persistencia y economía

Aunque el estado en memoria de una sesión anterior se pierda, los cuadernos abiertos se guardan periódicamente en una carpeta `.ipynb_checkpoints`, lo cual consigue evitar la pérdida del trabajo almacenado en ficheros.

Evolución: aplicaciones externas en JupyterHub (y k8s)

Integración

La definición de la imagen utilizada para lanzar los **notebooks** está basada en la imagen [jupyter/datascience-notebook:ubuntu-20.04](https://hub.docker.com/r/jupyter/datascience-notebook) a la que se le han añadido OpenRefine, RStudio y la herramienta [jupyter-server-proxy](#)

Adecuación

Es posible añadir nuevos aplicativos a la imagen del **notebook** utilizada por los usuarios. El acceso a ellos queda así unificado a través de una misma interfaz web y se beneficia del mecanismo de autenticación de JupyterHub, como sucede con los servicios de OpenRefine y RStudio que se encuentran integrados actualmente.



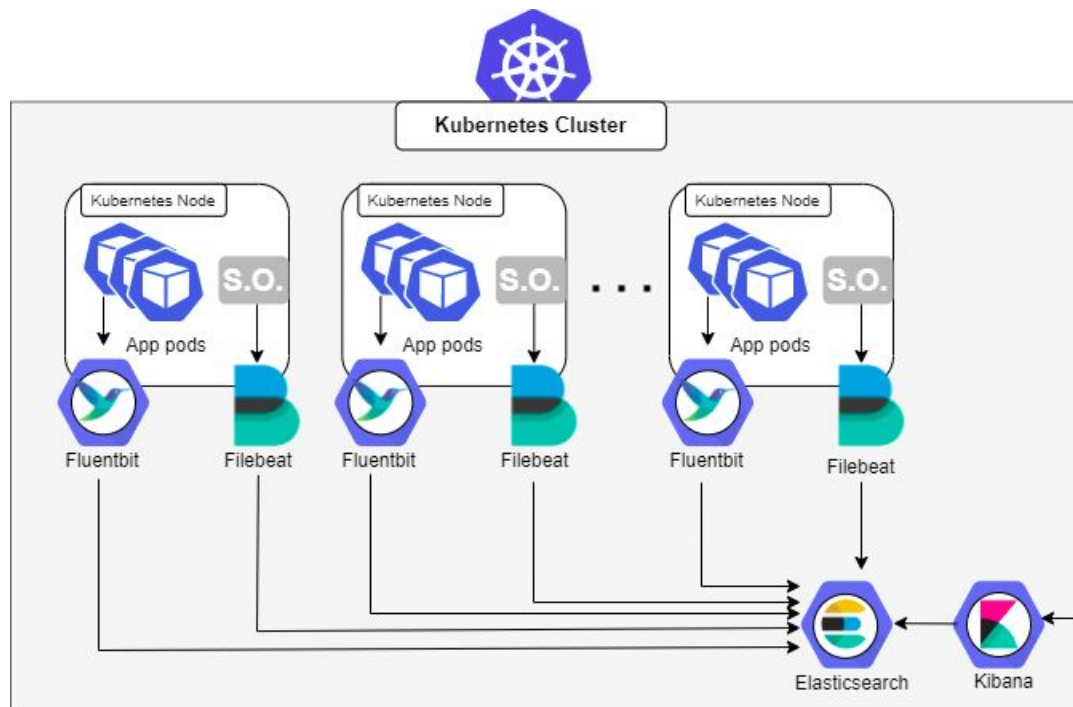
Personalización

Una instancia de OpenRefine y RStudio única para cada usuario dentro del propio **notebook**, proporcionando así un entorno único protegido por la autenticación de JupyterHub y que se puede iniciar simplemente pulsando un botón en el launcher del **notebook**.

Monitorización de la infraestructura: logs

Stack EFK ([Elastic](#), [Fluentbit](#), [Filebeat](#) y [Kibana](#)) para la recolección, transformación, normalización y visualización de este tipo de datos.

- **Filebeat:** Permite recolectar logs de los nodos del cluster, sistema operativo y reenviarlos hacia Elasticsearch donde serán almacenados y consultados.
- **Fluentbit:** Conecta distintas fuentes de entrada de información, normalizarlas y las reenvía hacia un punto de recepción (Elasticsearch en nuestro caso)
- **Elasticsearch:** Punto de recepción de los datos estructurados y, además, motor de búsqueda para interactuar con la información de forma eficiente.
- **Kibana:** Componente de visualización interactúa con Elasticsearch para obtener información y la presenta de forma más amigable, permitiendo crear y personalizar dashboards, consultas, diagramas y tablas.



Monitorización de la infraestructura: métricas



Prometheus server: Recoge las métricas de los targets y las almacena en un modelo de datos multidimensional que podemos consultar utilizando el lenguaje de consulta PromQL.



Prometheus Pushgateway: permite que los pods y jobs efimeros expongan sus métricas a Prometheus.



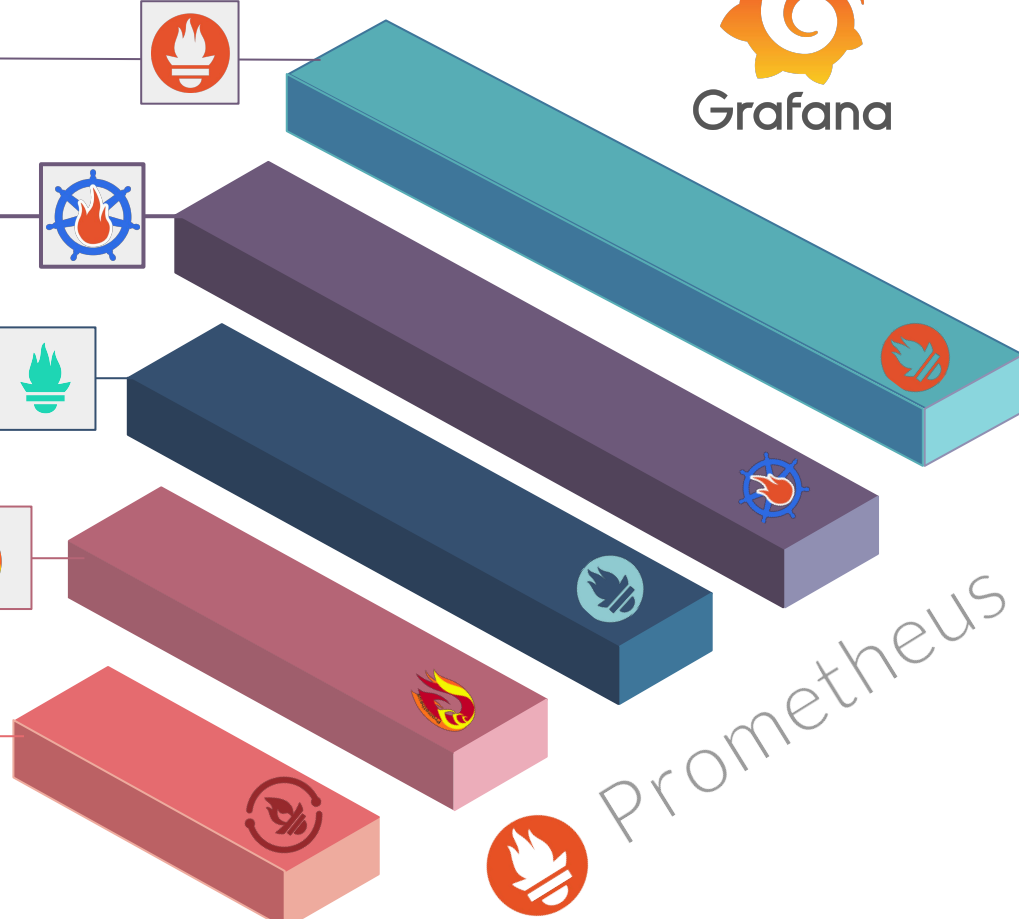
Kube-state-metrics: Genera métricas sobre el estado del clúster usando la información recolectada a través del apiserver de Kubernetes.



Node-exporter: Recoge las métricas del host (uso de CPU, uso de memoria, carga de trabajo, uso de los filesystems, etc.).



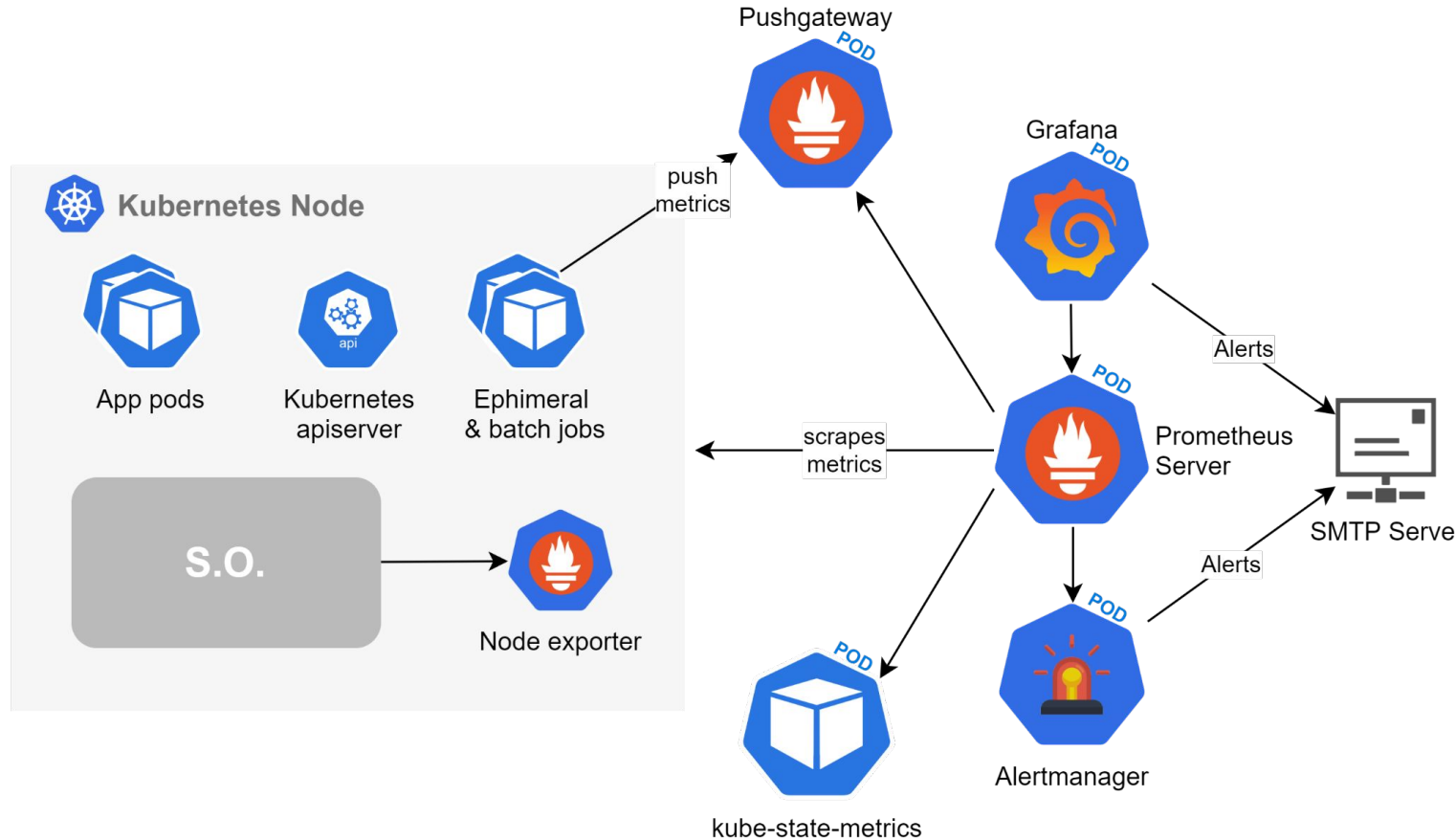
Alertmanager: Sistema de avisos del propio Prometheus; permite definir alertas más complejas que Grafana.



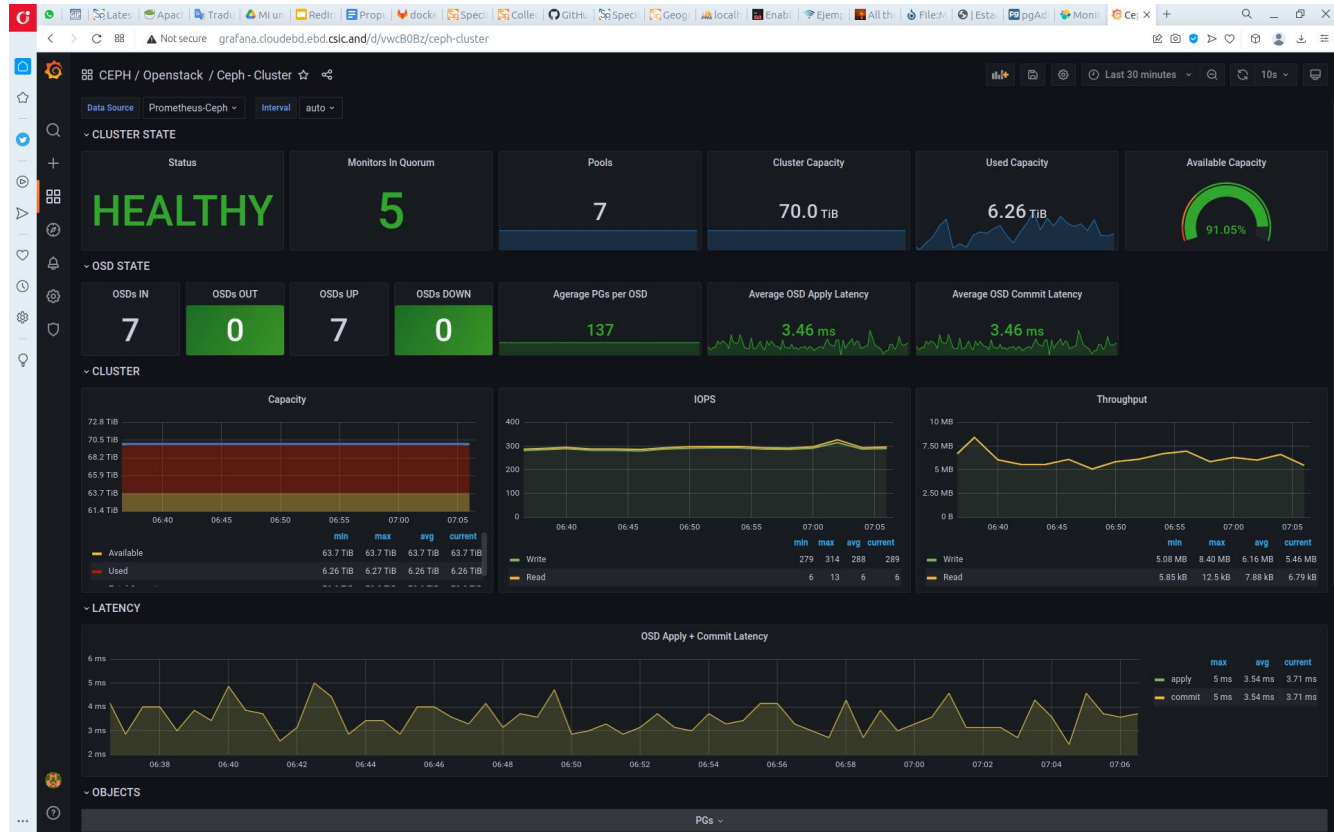
Monitorización de la infraestructura: despliegue

Para la recolección y almacenamiento de métricas del clúster se despliega un stack de Prometheus junto con la herramienta Grafana para explorar y visualizar las métricas.

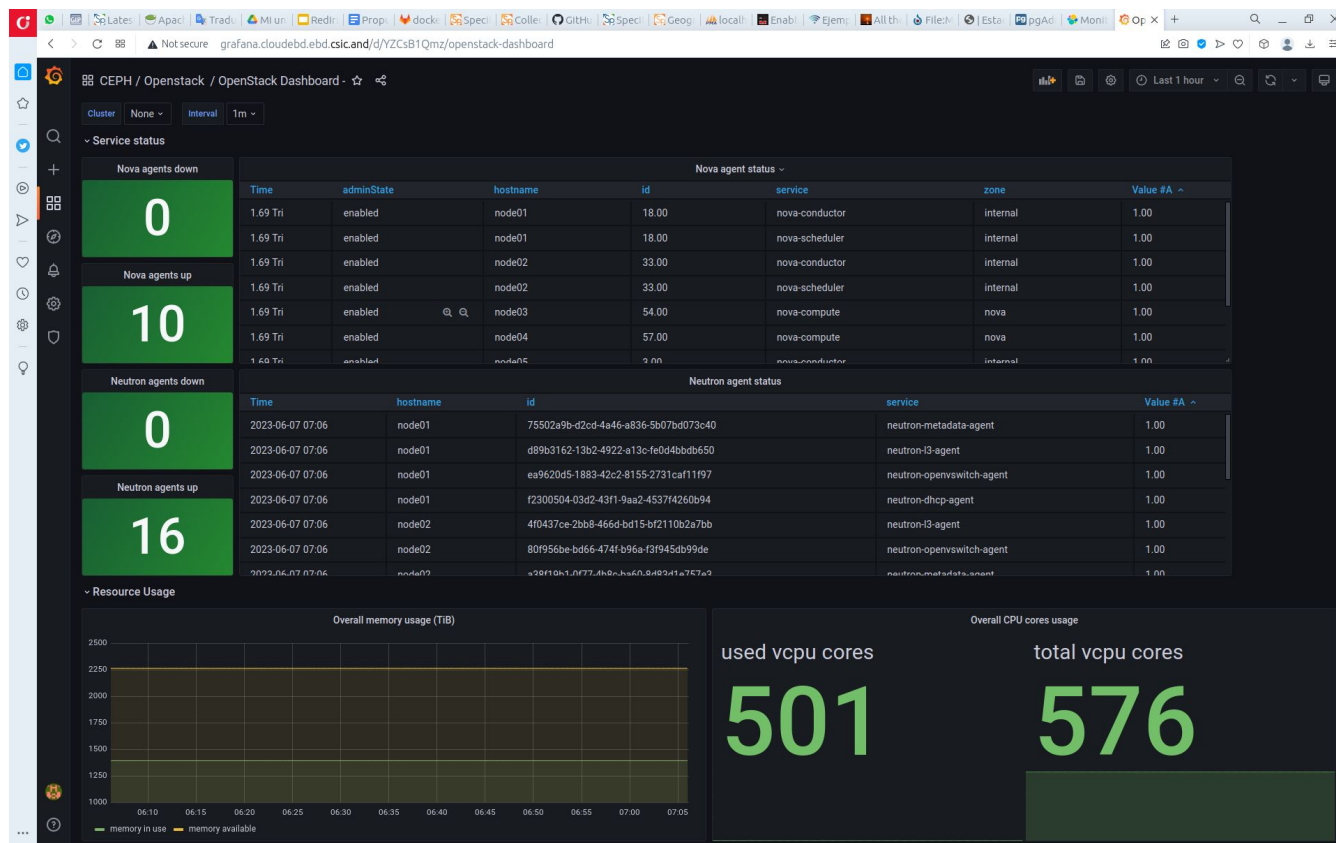
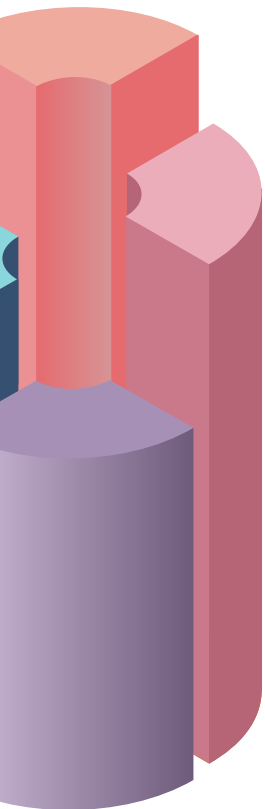
Las métricas se almacenan de forma persistente con un periodo de retención de 30 días.



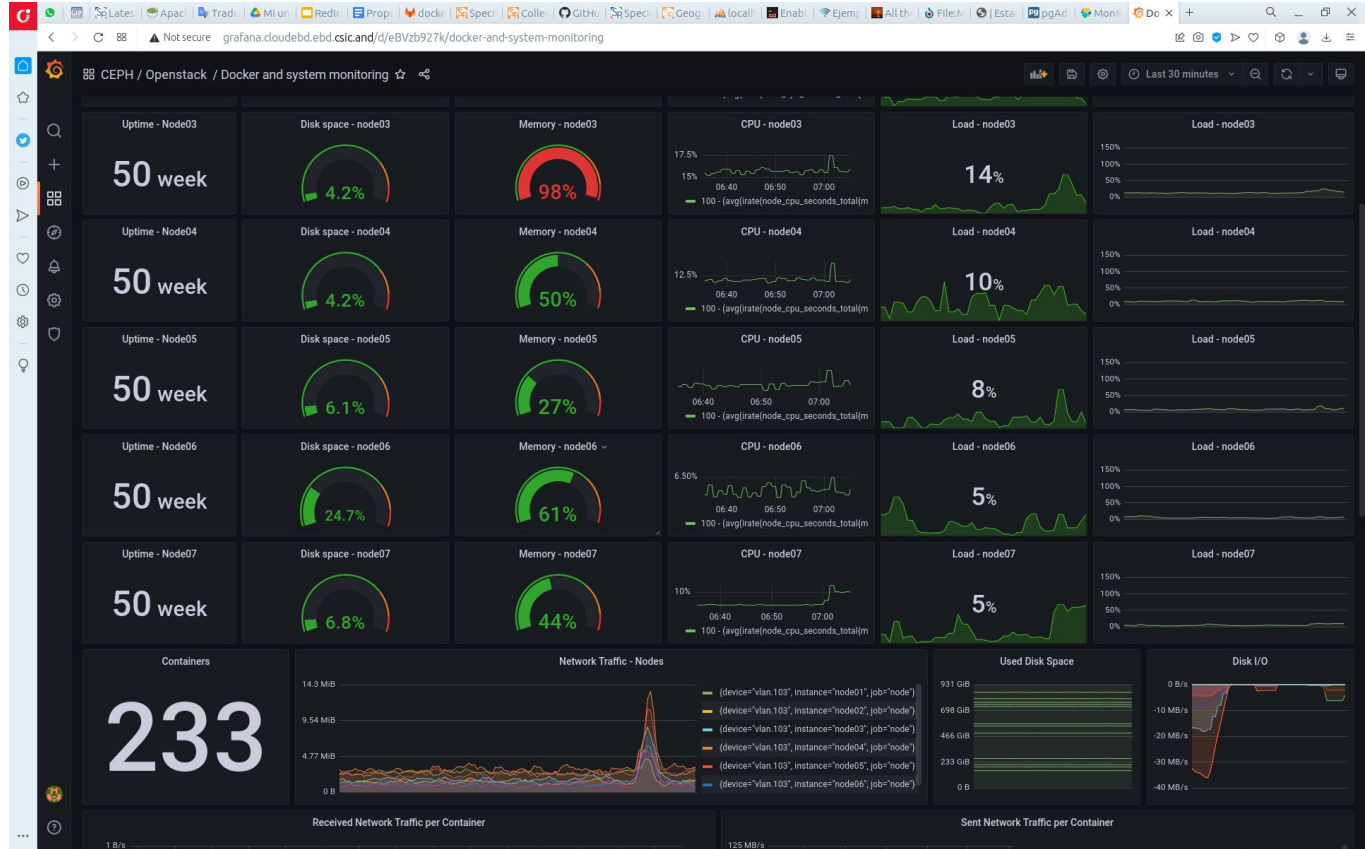
Monitorización de la infraestructura: SDS Ceph



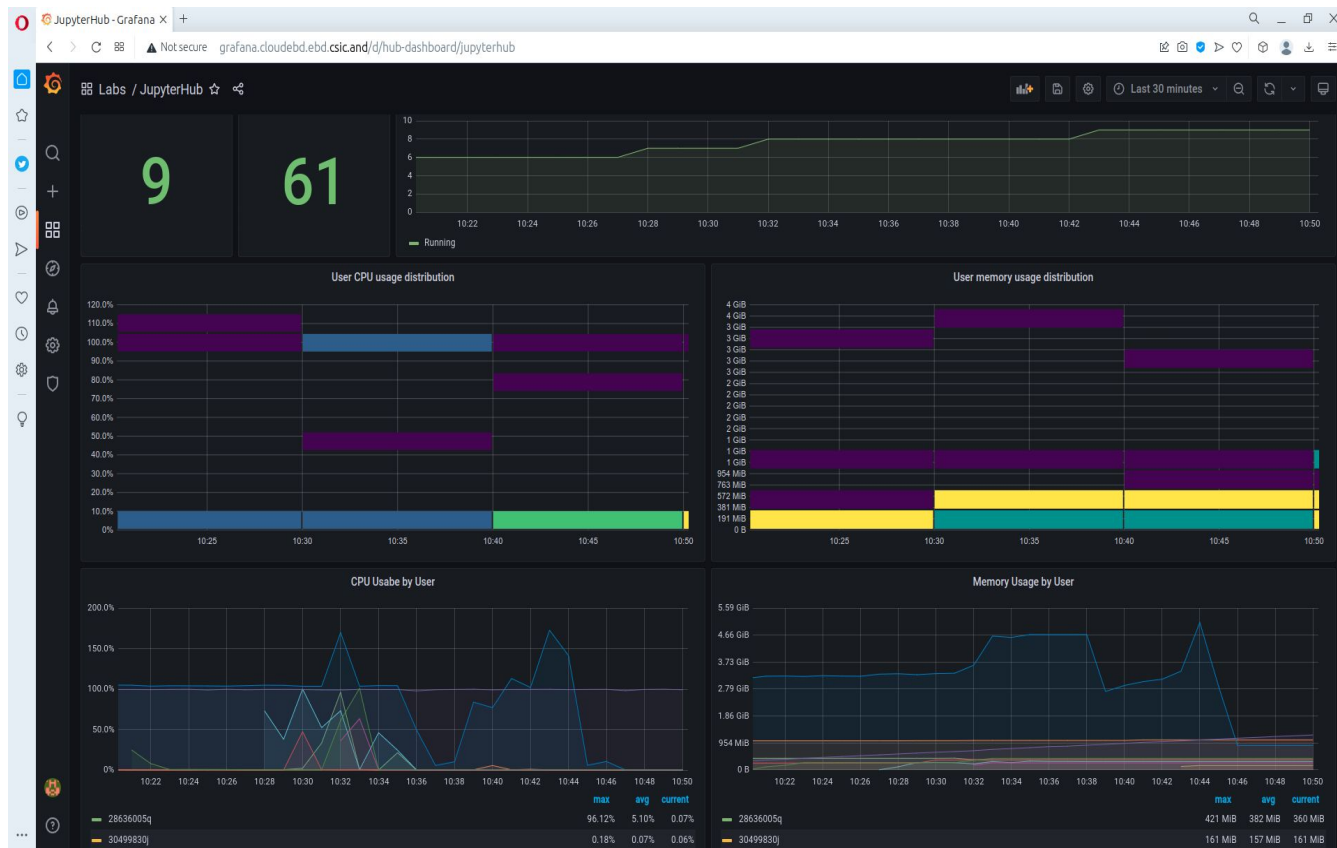
Monitorización de la infraestructura: nube privada



Monitorización de la infraestructura: containers



Monitorización de la infraestructura: pods



Recapitulando. Jupyterhub y kubernetes:

Notebooks **efimeros** pero **persistentes** gracias a k8s

Con **Jupyter-Kubespawner** se vincula el escalado horizontal del cluster a despliegues de nuevos servidores Jupyter

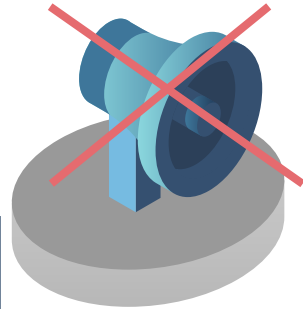
Incorporando **aplicaciones externas** es posible la adecuación a nuevas necesidades

La gestión de recursos disponibles se optimiza con los sistemas de **culling**

Privacidad, **aislamiento** de la información y facilidad de conexión con sistemas de **acreditación** corporativos



Claves de un elab v3



Simplicidad

Sin entornos de colaboración y publicación

Sistemas efímeros: los recursos del elab se devuelven al pool si están inactivos 1 hora

Ahorro

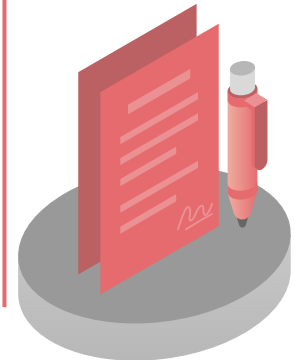


Privacidad

Contextos de almacenamiento persistentes y privados

Configuraciones personalizables, persistentes y evolucionables

Autonomía



Gracias.

Preguntas ?

Jose Luis Castro López
Luis Guillermo Torres Sanjuán
Servicio de Informática
Estación biológica de Doñana