Universidad Nacional del Noroeste de la Provincia de Buenos Aires

Título

Optimización del desempeño, escalabilidad y estabilidad de aplicaciones mediante el uso de contenedores

Carrera: Ingeniería en Informática

Práctica Profesional Supervisada

Estudiante: Marcelo Emanuel Guiguet

Tutor Docente: Javier Charne

Tutor de Empresa/Institución/Organización: Carlos Di Cicco

Fecha de presentación: 19 de octubre de 2022

UNNOBA
Universidad Nacional
Noroeste • Buenos Aires

Índice

UNNOBA
Universidad Nacional
Noroeste • Buenos Aires

Tablas de contenidos

1. Introducción

El Sistema Integral de Gestión y Evaluación (SIGEVA) es un conjunto de aplicaciones informáticas desarrollado por la Dirección de Informática de la Gerencia de Organización y Sistemas del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

El sistema nace en 2005 debido a la necesidad del CONICET de contar con una herramienta adecuada para la gestión y evaluación del volumen de trámites relacionados a becas que este manejaba para ese momento. En 2009 la Universidad de Buenos Aires (UBA) firma el primer convenio con CONICET para la utilización del SIGEVA y a partir de ese suceso varias instituciones, principalmente universitarias, realizan las firmas de convenios e implementación del sistema.

La Universidad Nacional del Noroeste de la Provincia de Buenos Aires (UNNOBA) a través de la Resolución del Rector 5398 (2012) implementa SIGEVA. La implementación inicial de la instancia SIGEVA-UNNOBA se realiza en conjunto entre los equipos técnicos de CONICET y la Prosecretaría TIC UNNOBA (PROTIC), correspondiéndole a esta última la responsabilidad de implementar las futuras actualizaciones del sistema.

En la UNNOBA actualmente se utiliza el SIGEVA para las presentaciones a convocatorias a proyectos de investigación y extensión y a becas de grado y posgrado y para la presentación de los informes de avances de los proyectos de investigación ya iniciados. Además, por medio de la Resolución del Rector 953 (2019) se estableció la obligatoriedad para todos los docentes de la universidad la carga y actualización de sus antecedentes en la instancia SIGEVA-UNNOBA. El acceso al sistema se realiza a través de la autenticación con la cuenta institucional, método utilizado en los demás sistemas de la universidad.

En el último tiempo la instancia SIGEVA-UNNOBA experimenta problemas de desempeño y estabilidad que afectan negativamente la interacción de los usuarios con el sistema. Debido a la importancia del SIGEVA en diversos procesos de las áreas de Investigación y Académica de la UNNOBA, la necesidad de la PROTIC de mejorar la performance del sistema y la habilitación establecida en el convenio aprobado mediante Resolución 5398 /2012 a introducir modificaciones al sistema, ha surgido el presente trabajo el cual consiste en realizar un relevamiento de la instancia SIGEVA-UNNOBA y, a partir del análisis de la

información recopilada, llevar a cabo la implementación y evaluación de mejoras, asentando lo desarrollado en un informe final.

2. Objetivos

Esta Práctica Profesional Supervisada (PPS) tiene como objetivo general establecido mejorar el desempeño, estabilidad y escalabilidad de la instancia SIGEVA-UNNOBA utilizando contenedores. Con el fin de alcanzar este objetivo general se plantearon los siguientes objetivos específicos:

- Reemplazar el antiguo servidor de aplicaciones Java por uno que cuente con mantenimiento activo y soporte técnico.
- Empaquetar la aplicación Sigeva y procesos auxiliares en contenedores.
- Reducir el tiempo fuera de servicio del sistema.
- Poder gestionar la escala del servicio.
- Facilitar el proceso de resolución de problemas de la instancia SIGEVA-UNNOBA mediante la gestión centralizada para el análisis de la misma.

3. Plan de Trabajo y Carga Horaria

CRONOGRAMA DE TAREAS										
ACTIVIDADES	***TIEMPO DE DURACIÓN (SEMANAS)***									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Análisis Instancia SIGEVA-UNNOBA (documentación, paquetes)	X	X								
Análisis de plataformas que usa la PROTIC (selección)		X	X							
Documento de propuestas de mejoras				X	X					
Implementación de mejoras (evaluación en test)					X	X	X			
Implementación en producción								X	X	
Elaboración informe final		X	X	X	X	X	X	X	X	X

La carga horaria se asignó de acuerdo a un periodo de 10 semanas (5 días a la semana) con 5 horas diarias en la franja horaria comprendida entre las 8 y las 13 hs.

4. Descripción de la PPS

4.1. Análisis de la Instancia SIGEVA-UNNOBA

4.1.1. Recopilación de documentación

En esta primera etapa del trabajo se hizo un relevamiento y recopilación de la documentación existente y disponible sobre el sistema SIGEVA y de la instancia SIGEVA-UNNOBA.

La PROTIC posee un sistema normativo que regula toda la información que gestiona (tanto interna como externa), para lo cual define tanto normas como procedimientos que detallan los pasos a seguir en cada caso. La gestión de tickets y tareas con la herramienta de trabajo colaborativo Egroup (EGroupware GmbH, 2022), servicio llamado Colab en la UNNOBA, se encuentra normalizada por este sistema normativo. Los tickets son solicitudes de requerimiento de un servicio a la PROTIC y las tareas son registros de las acciones para resolver un requerimiento originado en un ticket. Las tareas también pueden ser registros de las labores realizadas en la PROTIC con un fin determinado, el cual no se encuentra puntualmente asociado un ticket. Existen tickets y tareas relacionados con la instancia SIGEVA-UNNOBA, siendo generalmente el origen de los tickets eventualidades o problemas del sistema y las tareas las correspondientes al tratamiento estos tickets y, además, las referidas a la aplicación de actualizaciones o cualquier otro tipo de intervención, realizada o planificada, sobre la instancia.

Se recopilaron documentos técnicos del sitio oficial del SIGEVA entre los que se encuentran los requerimientos técnicos (CONICET, 2019), una guía de actualización de versiones (CONICET, 2016), un instructivo de configuración de procesos batch (CONICET, 2019) y una guía de migración de las bases de datos (CONICET, 2014).

4.1.2. Arquitectura y características del entorno

La arquitectura de la instancia SIGEVA-UNNOBA, representada en la figura 1, se encuentra constituida fundamentalmente por una máquina virtual (VM), la cual cuenta con un servidor apache que cumple las funciones de proxy reverso (interno) y ejecuta un webservice para validación de credenciales contra un servicio denominado login, un servidor de aplicaciones Glassfish (Oracle, s.f.) en el cual se ejecutan las aplicaciones que componen el SIGEVA y un servidor de base de datos MySQL. Adicionalmente a esta VM se encuentra un proxy reverso, que es el responsable de las consultas externas —es decir, desde internet o una red no administrativa de la universidad— y el mencionado servicio login encargado de la autenticación en los sistemas informáticos de la universidad y la gestión personal de las cuentas institucionales.

Las solicitudes originadas de una red de acceso interno de la universidad son atendidas directamente por el servidor apache, realizando un proxy pass hacia el servidor Glassfish que ejecuta el SIGEVA. Las solicitudes externas, en cambio, son manejadas por el proxy reverso público, quién realiza un proxy pass hacia la VM dónde dichas solicitudes son tratadas por el servidor Apache, de la misma forma que en el caso de las solicitudes internas.

La autenticación en el sistema SIGEVA se realiza mediante la utilización del webservice que se ejecuta en el servidor Apache, quién se encarga de llevar a cabo la validación de credenciales contra el servicio login.

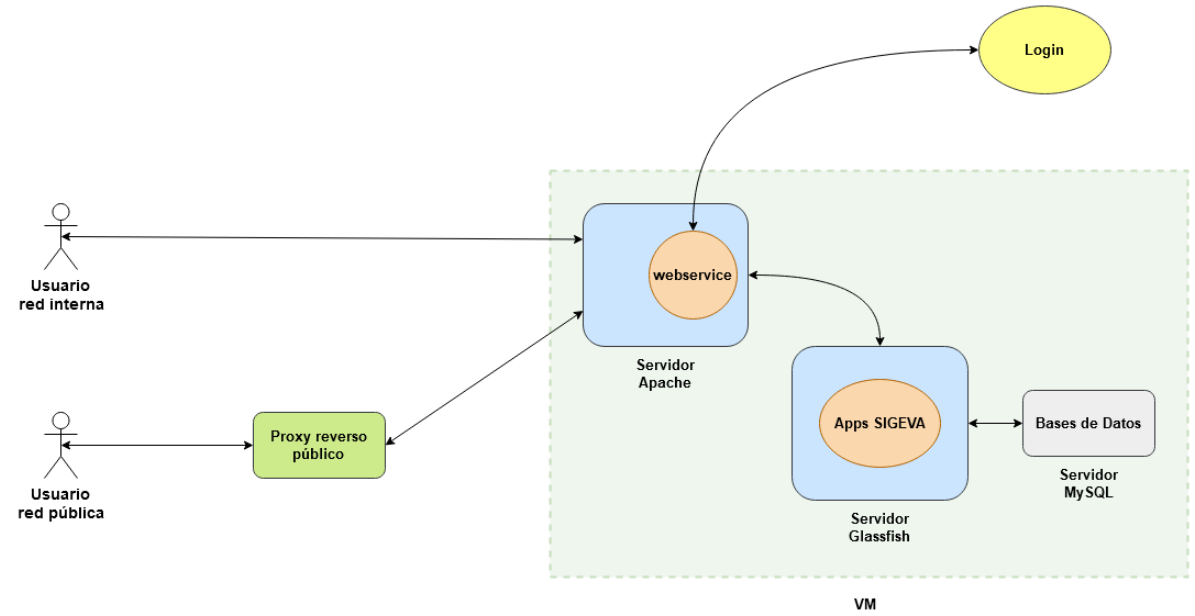

Figura 1. Arquitectura de la instancia SIGEVA-UNNOBA

En la tabla 1 se encuentran las características del entorno utilizado en la instancia SIGEVA-UNNOBA de producción. El ambiente de test es muy similar, variando en la cantidad de VCPUs asignados (2 unidades) y la url de acceso es sigeva.test.unnoba.edu.ar.

Memoria RAM	8 GB
VCPUs	4
Sistema Operativo	Debian 10
Servidor de aplicaciones	Glassfish 3.1.2.2
MySQL	Versión 5.6.30
Java	JDK 7u51
Almacenamiento	100 GB
Servidor SMTP	Servidor de correo configurado y funcionando.
URL de acceso	https://sigeva.unnoba.edu.ar

Tabla 1. Especificaciones del entorno SIGEVA-UNNOBA de producción.

En la documentación de requerimientos técnicos publicada por CONICET se encuentran las especificaciones de los entornos recomendados para la implementación del SIGEVA tanto para el ambiente de test como para el de producción, las cuales se pueden observar en la tabla 2.

Servidores	Uno para producción y uno para ambiente de test
Memoria RAM	Mínimo 8 GB
Sistema Operativo	Linux. Distribución recomendada CentOS
Servidor de aplicaciones	Glassfish (versión no especificada)
MySQL	Versión 5.6.27
Java	JDK 7
Almacenamiento	Mínimo 100 GB iniciales
Servidor SMTP	Servidor de correo saliente SMTP accesible desde la red local en que se instalará SIGEVA
URL de acceso	URL directa, ejemplo http://sigeva.institucion.edu.ar

Tabla 2. Requerimientos técnicos generales del SIGEVA especificados por el CONICET

Al comparar la tabla 1 con la tabla 2 se puede comprobar que los entornos de producción y test de la instancia SIGEVA-UNNOBA respètan las recomendaciones brindadas por el CONICET.

4.1.3. Detalles de la implementación y actualizaciones

Con la aprobación del convenio de uso del SIGEVA, por la resolución 5380/202, la UNNOBA procedió a la implementación del sistema. La instalación inicial de la instancia SIGEVA-UNNOBA fue llevada a cabo en un trabajo en conjunto entre el CONICET y la UNNOBA, de acuerdo a lo establecido en la mencionada resolución.

La PROTIC fue responsable de inicializar un entorno virtual siguiendo los requerimientos especificados por el CONICET. Una vez preparado el entorno, el equipo del CONICET realizó el proceso de instalación inicial, el cuál tiene como objetivo establecer y parametrizar un entorno de producción (o de test) del SIGEVA. Este proceso de instalación consistió de las siguientes tareas:

- Instalación y configuración de Java
- Instalación y configuración de Glassfish
- Instalación y/o configuración del servidor MySQL
- Copiado de carpetas de SIGEVA y configuración de ficheros
- Inicialización de las bases de datos de SIGEVA
- Instalación y configuración mínima de Apache, sólo orientada a la publicación directa del SIGEVA como servicio.

Luego del trabajo del equipo del CONICET, la PROTIC fue la responsable de configurar los registros DNS, el acceso externo por medio de un proxy reverso y la configuración del servidor de correo en el sistema.

El proceso de implementación inicial queda formalmente completado en el momento en que se verifica que:

- La aplicación es accesible desde Internet en la URL definida.
- SIGEVA realiza el envío de emails de manera correcta.
- Se puede realizar la sincronización de datos con SIGEVA-CONICET.

En lo referido a las actualizaciones, en la instancia SIGEVA-UNNOBA la responsabilidad de aplicarlas recae en la PROTIC. El proceso de actualización del sistema consiste generalmente en la actualización del código del SIGEVA como así también de la estructura de las bases de datos. Para la realización de estas tareas el CONICET incluye en cada paquete de versión las instrucciones y/o scripts para aplicar los cambios necesarios. Estos scripts están diseñados para realizar la actualización del SIGEVA en un entorno como el recomendado por CONICET, siendo necesario su revisión y/o modificación en cualquier otro escenario. La PROTIC registra, como se ha mencionado previamente, cada una de las tareas de actualización que se realizan.

4.1.4. Política de backup

El CONICET delega en cada institución adoptante del SIGEVA la responsabilidad de establecer la política y la realización de los backups, a excepción de la modalidad hosting. En el caso de la instancia SIGEVA-UNNOBA el proceso de backups es definido y realizado por la PROTIC..

En la documentación ofrecida por el CONICET se encuentran una serie de sugerencias para el proceso de backup, las cuales se encuentran resumidas en la tabla 3.

Backup diario	**Bases de datos:** ● eva ● eva duplicado ● sarauth **Directorio:** ● */usr/local/sigeva_files*
Al actualizar la versión	**El directorio completo de glassfish:** ● /usr/local/glassfish **Excepciones:** ● */usr/local/glassfish/glassfish/domains/domain1/applications/ejb-timer-service-app* ● */usr/local/glassfish/glassfish/domains/domain1/eclipseApps* ● */usr/local/glassfish/glassfish/domains/domain1/eclipseAppsTmp* ● */usr/local/glassfish/glassfish/domains/domain1/generated* ● */usr/local/glassfish/glassfish/domains/domain1/osgi-cache* ● */usr/local/glassfish/glassfish/domains/domain1/logs* En el caso del directorio “logs”, el archivo server.log debería guardarse. Se recomienda también la realización de un respaldo de las configuraciones de los componentes externos al SIGEVA pero que interaccionan con el mismo. Por ejemplo: ● Configuración del servidor SMTP ● Configuración del Apache ● Configuración del sistema operativo

Tabla 3. Sugerencias del CONICET sobre backups para el SIGEVA

Si bien la política de backup de la PROTIC queda reservada al ámbito interno de la misma, se puede hacer mención que el proceso de backup de la instancia SIGEVA-UNNOBA consiste en la realización de una copia diaria del volumen asociado a la VM sobre la cual está implementada, respaldando de esta forma la totalidad de los datos del sistema en concordancia con las sugerencias brindadas por el CONICET.

4.1.5. Problemas encontrados

La instancia SIGEVA-UNNOBA presenta problemas de desempeño y estabilidad, que son más notorios en las épocas de convocatorias abiertas en las cuales el sistema tiene un número de usuarios simultáneos superior a la media. Según los registros de la PROTIC, el sistema en estos periodos de convocatorias tiene un tiempo de respuesta elevado, pudiendo ser desde varios segundos hasta de minutos, en los peores casos. Además, es común que el servidor Glassfish deje de responder siendo necesario realizar un reinicio manual del mismo.

En el último tiempo se registró un aumento en la frecuencia en que Glassfish deja de responder. Ante esta situación, la PROTIC aplica una solución provisoria, la cual consiste en la definición de un crontab en la VM para reiniciar el servidor una vez por día, buscando mejorar la disponibilidad del sistema.

Analizando y probando el sistema se pudo constatar que el mismo presenta los inconvenientes documentados. Además, se detecta que la intervención realizada por medio de la definición del crontab para el reinicio diario del servidor Glassfish ocasiona que el primer usuario que ingrese al sistema luego de la ejecución de dicho crontab, tenga que esperar la compilación de los JSP de las aplicaciones del SIGEVA, proceso que demora de 2 a 5 minutos, según lo observado.

Algo a destacar en este punto es la existencia en los registros de tareas de la PROTIC de actividades de análisis e implementación de optimizaciones sobre el sistema, con foco en Glassfish, cuyos resultados no fueron satisfactorios ya que no se logró resolver la problemática existente.

4.2. Plataformas disponibles y utilizadas en la PROTIC

En la PROTIC se utiliza OpenStack (OpenInfra Foundation, 2022) como plataforma de aprovisionamiento y gestión de máquinas virtuales. Un gran número de los sistemas de la universidad se encuentran implementados sobre VM generadas en esa plataforma de virtualización. Los volúmenes utilizados en OpenStack, ya sean los de las VM o auxiliares para datos se encuentran en CEPH (Ceph Foundation, 2022), un sistema de almacenamiento distribuido libre, implementado en el datacenter de la PROTIC.

La PROTIC cuenta con un cluster kubernetes desplegado por medio de Rancher (SUSE Group, 2022) sobre nodos virtualizados. En este orquestador de contenedores se despliegan los sistemas contenerizados de la PROTIC. Para la persistencia de datos hace uso de CEPH por medio del driver correspondiente, permitiendo un aprovisionamiento dinámico de un volumen —Persistent Volume (PV) (Cloud Native Computing Foundation, 2022) en el ámbito de kubernetes—, el cual se asocia a un Persistent Volume Claim (PVC) (Cloud Native Computing Foundation, 2022) que utiliza la StorageClass (Cloud Native Computing Foundation, 2022) correspondiente al driver de CEPH, el cual es el encargado de relacionar el PV con el POD (Cloud Native Computing Foundation, 2022) que ejecuta el contenedor que lo utiliza para persistir los datos necesarios. Se utiliza Prometheus (Cloud Native Computing Foundation, 2022) como servidor de métricas y el stack Loki-Promtail (Grafana Labs, 2022) para la recopilación de logs de cada contenedor ejecutado en el clúster.

La herramienta Grafana (Grafana Labs, 2022) es utilizada como visualizador de los datos de Prometheus y Loki, en lo referido al cluster kubernetes, y para diversos datos de otros datasource utilizados en la PROTIC.

Las aplicaciones de la PROTIC desarrolladas en Java Platform, Enterprise Edition (Java EE) se ejecutan en entornos compuestos por el servidor de aplicaciones Glassfish sobre una VM. El acceso público por medio de internet de estas aplicaciones, como el de la mayoría de los sistemas de la PROTIC, se configura a través de un proxy reverso.

4.3. Propuesta de mejoras

4.3.1. Selección de las plataformas y tecnologías a utilizar

Teniendo en cuenta las plataformas y tecnologías disponibles en la PROTIC, una opción posible para una nueva arquitectura es la de empaquetar el SIGEVA y ejecutarlo mediante un Docker Compose (Docker Inc., 2022) en una VM provisionada en OpenStack. Una segunda opción es la de ejecutar el sistema contenerizado en el cluster kubernetes. En ambas opciones los datos se persisten en CEPH, ya sea en el volumen de la VM, o auxiliar, en la primera opción o haciendo uso del mecanismo de aprovisionamiento del cluster kubernetes para la segunda opción.

Se opta por la opción de utilizar el cluster kubernetes como plataforma de ejecución de los contenedores ya que la misma cuenta con una serie de características que contribuyen a solucionar los problemas de la instancia SIGEVA-UNNOBA. Kubernetes al ser un orquestador de contenedores permite realizar de forma automática procesos de gestión de la ejecución de los contenedores que en la otra opción se tendrían que hacer de forma manual o semiautomática (se pueden crear scripts para automatizarlos pero se complejizaba la propuesta de mejoras). Kubernetes puede de manera automática reiniciar la ejecución de un contenedor si detecta, por medio de las pruebas LivenessProbe (Cloud Native Computing Foundation, 2022) y ReadinessProbe (Cloud Native Computing Foundation, 2022), que el mismo no se encuentra funcionando de manera correcta. Además, Kubernetes puede mover el o los contenedores que se ejecutan en un nodo a otros nodos en caso que el nodo original presente alguna falla o se encuentre fuera de servicio y dispone de herramientas para gestionar la escala. En el cluster Kubernetes de la PROTIC se encuentran implementadas un conjunto de herramientas que posibilitan centralizar el análisis de logs y métricas de las aplicaciones desplegadas en el mismo, lo que facilita la resolución de problemas.

En lo referido al servidor de aplicaciones Java, considerando el amplio uso de Glassfish en la PROTIC y fundamentalmente en que el SIGEVA implementado está siendo ejecutado por ese tipo de servidor, se opta por seleccionar a Payara Server (Payara Foundation, 2022), que es un fork de Glassfish que cuenta con actualizaciones regulares y un soporte activo tanto de nivel comunitario como empresarial. Esta elección de Payara Server se fundamenta, además de lo ya expuesto, en que al ser un fork de Glassfish se pueden reutilizar la mayoría de las configuraciones existentes en la implementación original de la instancia SIGEVA-UNNOBA, lo que simplifica el trabajo a realizar y permite un ahorro de tiempo.

Con la utilización de Payara Server se busca resolver los problemas de desempeño ya que es un entorno de ejecución Java actualizado que cuenta con continuas revisiones. Al decidir empaquetar el sistema en contenedores se busca la mejora en la estabilidad y escalabilidad del mismo. La orquestación de estos contenedores en el clúster Kubernetes va a permitir gestionar la escala de manera rápida y simple por medio de las herramientas de escalado y determinación de cuotas de Kubernetes (Cloud Native Computing Foundation, 2022). La estabilidad se busca mejorar fundamentalmente haciendo uso de dos elementos: por un lado el sistema de pruebas de kubernetes, las mencionadas LivenessProbe y

ReadinessProbe, y por la propiedad de efimeridad de los contenedores, la cual consiste en la invariabilidad del entorno definido o, en otras palabras, que cualquier cambio que se produzca en un contenedor solo va a permanecer durante la instancia de ejecución del mismo.

El contenedor al ser efímero y el hecho de persistir por medio de volúmenes sólo los datos del SIGEVA, permite que en el caso de que se corrompan los archivos utilizados en el contenedor, ya sean de configuración, de Payara Server o cualquier otro, al ejecutar nuevamente el contenedor esos archivos van a estar tal cual fueron definidos en la imagen, lo que permite que el sistema vuelva a funcionar sin la presencia de archivos dañados. El uso del sistema de pruebas de Kubernetes posibilita, dependiendo el tipo de falla, que el sistema se recupere automáticamente sin la necesidad de la intervención de una persona y en un tiempo relativamente corto.

Para conseguir una gestión centralizada para la resolución de problemas se va a utilizar el stack Loki - Promtail como concentrador y recolector de logs respectivamente, Prometheus como servidor de métricas y Grafana para visualizar y analizar los datos obtenidos con las anteriores herramientas.

4.3.2. Arquitectura propuesta

Se propone utilizar una arquitectura en contenedores desplegados en Kubernetes. En un contenedor se empaqueta Payara Server quién se encarga de ejecutar las aplicaciones que componen al SIGEVA. En otro contenedor se configura el webservice de validación de credenciales. Para la base de datos se utiliza un contenedor MySQL. Cada uno de estos contenedores se ejecutan en POD diferentes.

Las solicitudes al sistema son controladas por Traefik mediante la definición de Ingressroutes en los cuales se especifican las reglas para direccionar estas solicitudes a los puertos necesarios en los POD correspondientes. Las solicitudes externas son controladas por el proxy reverso público, de igual manera que en la arquitectura original con la salvedad que el proxy pass se realiza contra el Traefik del clúster Kubernetes.

Dentro del clúster Kubernetes, Promtail se encarga de recolectar los logs de cada uno de estos contenedores y los envía, para su almacenamiento, a Loki. Las métricas de uso son

recolectadas por el Prometheus interno del clúster. El análisis centralizado de la instancia se realiza por medio de Grafana.

En la figura 2 se encuentra el diagrama de esta arquitectura propuesta.

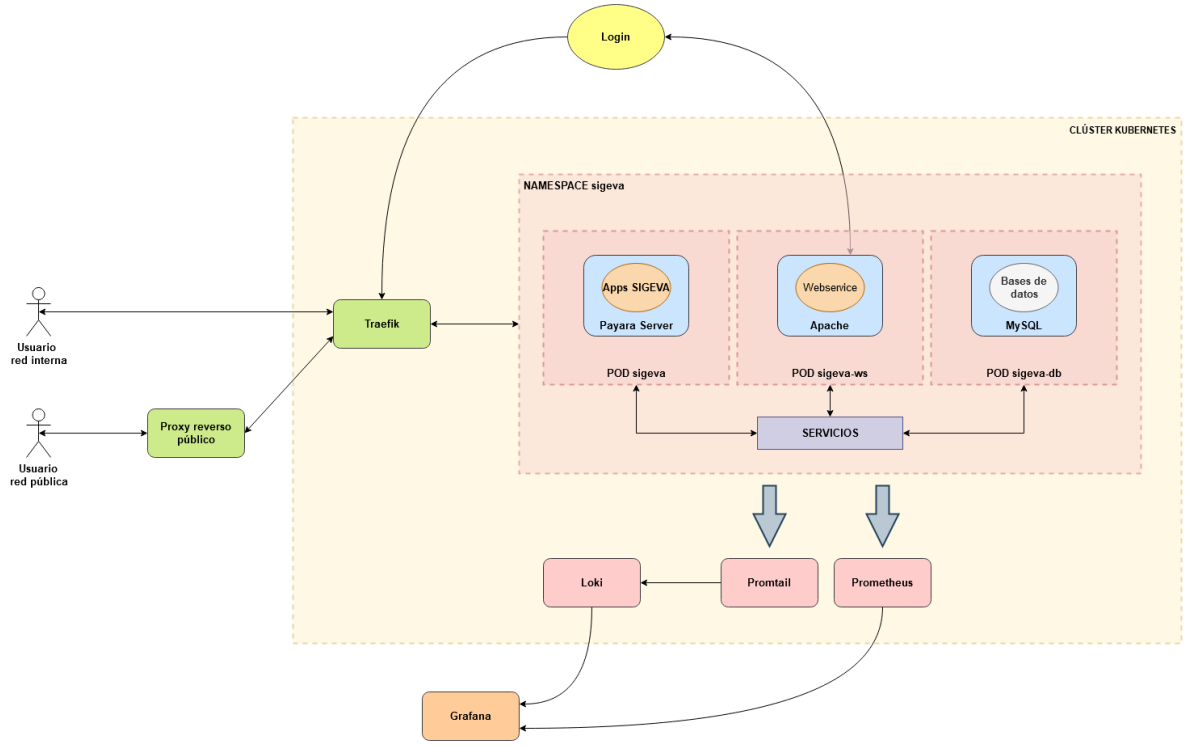

Figura 2. Diagrama de la arquitectura propuesta

4.4. Implementación de la propuesta de mejoras

4.4.1. Armado de la imagen del sistema

El armado de la imagen del contenedor que empaqueta a las aplicaciones del SIGEVA es el que mayor complejidad presenta ya que no solamente se debe configurar una instancia de Payara Server, con las consideraciones necesarias para su despliegue sobre Kubernetes, sino que también se deben realizar algunos ajustes en las configuraciones de las aplicaciones del SIGEVA.

Primeramente se definió una imagen base de Payara Server partiendo de la imagen oficial publicada por el desarrollador. En esta imagen se incorporó un script para posibilitar el seteo

de una contraseña para la consola de administración de Payara Server (asadmin) y un script para incorporar la funcionalidad de ejecutar comandos de la consola de administración de Payara Server (comandos asadmin), definidos por medio de un archivo, al momento de finalizar el despliegue de las aplicaciones. Además, se eliminan los archivos por defecto del docroot de Payara Server y se instala el paquete tzdata para poder configurar en el Deployment de Kubernetes la zona horaria correspondiente (America/Argentina/Buenos_Aires) mediante la definición de la variable de entorno TZ.

La imagen que empaqueta al sistema SIGEVA se construye utilizando la imagen base de Payara Server. En esta imagen se copiaron el archivo *.jar* correspondiente al conector MySQL, elemento necesario para el establecimiento de las conexiones de las aplicaciones con la base de datos, las bibliotecas brindados por el CONICET y los archivos que contienen los comandos asadmin que configurar el dominio de Payara Server. También, se sobreescribió el script que realiza el despliegue de las aplicaciones para garantizar que el mismo se realice en el orden correcto para evitar que se produzcan errores. Por último, se copiaron los archivos *.war* que contienen a cada una de las aplicaciones del SIGEVA.

Para la imagen del contenedor con el webservice de validación de credenciales se utiliza como base la imagen oficial de PHP sobre Apache. En esta imagen se copian los archivos del webservice y se instalan las dependencias necesarias.

El contenedor de las bases de datos utiliza la imagen oficial de MySQL y no se le realizaron modificaciones.

4.4.2. Modificaciones y configuraciones realizadas

Durante la etapa de construcción de la imagen del sistema se presentaron inconvenientes en el despliegue de las aplicaciones del SIGEVA en la arquitectura propuesta, por lo que fue necesario realizar modificaciones en los archivos *.war* que contienen a las aplicaciones del sistema para que el mismo funcione correctamente.

Uno de los problemas presentados fue que no se podía leer el archivo *apps.properties*, el cual contiene una serie de configuraciones generales de la aplicación en el formato de *clave=valor*, las cuales son necesarias para el funcionamiento del sistema. En este archivo se define el tipo de instancia (producción o test), el servidor smtp, las direcciones de emails de envío y recepción, datos correspondientes a la institución, el keystore y parámetros para

la sincronización con SIGEVA-CONICET, la conexión a la base de datos, entre otros parámetros del SIGEVA. En la VM original el archivo se encuentra ubicado en el directorio config dentro del directorio del dominio de Glassfish, siendo */usr/local/glassfish/glasssfish/domains/domain1/config/apps.properties* el path absoluto. En el contenedor con Payara Server el directorio config tiene el path absoluto */opt/payara/appserver/glassfish/domains/domain1/config*. Analizando los logs y los archivos de configuración de las aplicaciones se detecta el uso del path absoluto (correspondiente a Glassfish) para referenciar al archivo apps.properties en lugar del path relativo. Posiblemente el mismo también se encuentre hardcodeado en un método del sistema. Una manera práctica y sencilla de resolver este inconveniente, considerando que no se dispone del código fuente, es incluir en la imagen una copia del archivo apps.properties en el path original, es decir, en el directorio */usr/local/glassfish/glasssfish/domains/domain1/config/*. Esto se puede hacer copiando directamente el archivo a la imagen en su definición o montarlo al momento de la ejecución del contenedor, con la ventaja de no ser necesario construir una nueva imagen en caso de tener que modificar alguna configuración del sistema.

Otro de los problemas encontrados fue que Payara Server no podía encontrar los archivos *.tld* necesarios. Para solucionar este problema fue necesario definir en el archivo *web.xml*, luego del bloque *session-config*, de cada aplicación que usa archivos *.tld* un bloque *<jsp-config>* que contenga los *<taglib>* de cada uno de esos *.tld* requeridos. El código 1 es el que se debe incluir en el archivo web.xml correspondiente a la aplicación EVA. Además, se deben incorporar los archivos *struts-tiles.tld* y *displaytag.tld* en el *.war* de EVA en la carpeta *WEB-INF*. El archivo *struts-tiles.tld* se encuentra en *struts-tiles-1.3.10.jar* y el archivo *displaytag.tld* en *displaytag-conicet-1.2.jar*, estando ambos *.jar* ubicados dentro del *.war* de EVA en *WEB-INF/lib*.

```
<jsp-config>
  <taglib>
     <taglib-uri>
     http://displaytag.sf.net
     </taglib-uri>
     <taglib-location>
     /WEB-INF/displaytag.tld
     </taglib-location>
  </taglib>
  <taglib>
     <taglib-uri>
```

```
        http://struts.apache.org/tags-bean
        </taglib-uri>
        <taglib-location>
        /WEB-INF/struts-bean.tld
        </taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>
        http://struts.apache.org/tags-tiles
        </taglib-uri>
        <taglib-location>
        /WEB-INF/struts-tiles.tld
        </taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>
        http://struts.apache.org/tags-html
        </taglib-uri>
        <taglib-location>
        /WEB-INF/struts-html.tld
        </taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>
        http://struts.apache.org/tags-logic
        </taglib-uri>
        <taglib-location>
        /WEB-INF/struts-logic.tld
        </taglib-location>
    </taglib>
</jsp-config>
```

Código 1. Configuración de *<taglib>* para la aplicación EVA

Para la aplicación ADMAUTH se debe incorporar al *.war* el archivo *displaytag.tld* en la carpeta *WEB-INF* (tal como se hace para EVA) y es el código 2 el que se debe incorporar en en el archivo *web.xml* de la aplicación.

```
    <jsp-config>
        <taglib>
              <taglib-uri>
              http://displaytag.sf.net
              </taglib-uri>
              <taglib-location>
              /WEB-INF/displaytag.tld
              </taglib-location>
        </taglib>
    </jsp-config>
```

Código 2. Configuración de <*taglib*> para la aplicación ADMAUTH

La modificación del archivo *default-web.xml* fue también fue necesaria para que el sistema funcione en el contenedor. En este archivo se configura el dominio del servidor de aplicaciones (Glassfish originalmente y Payara Server en la solución propuesta). El archivo *default-web.xml* que se encuentra en la VM original contiene unas configuraciones definidas por el CONICET que son necesarias para que el SIGEVA funcione. La ausencia de estas configuraciones en el contenedor con Payara Server causaba errores en el ruteo entre las aplicaciones del SIGEVA. Para solucionar estos errores simplemente se copiaron cada una de las configuraciones definidas por el CONICET, en la instancia original, en las secciones correspondientes en el archivo *default-web.xml* de Payara Server. Este archivo default-web.xml, al igual que el de *apps.properties*, se puede copiar a la imagen o se puede montar durante la ejecución del contenedor, siendo esta la opción adecuada cuando se hace el despliegue en Kubernetes. En este informe no se detallan estas configuraciones debido a que el CONICET no las tiene accesible al público general, al menos en el momento de redacción del mismo.

Los archivos que contienen los comandos asadmin son tres: *preboot.commands* (comandos que se ejecutan antes del inicio de Payara Server), *postboot.commands* (comandos que se ejecutan luego del inicio de Payara Server) y *postdeploy.commands* (comandos que se ejecutan una vez terminado el despliegue de las aplicaciones). En este caso no es necesario ejecutar comandos asadmin antes del inicio de Payara Server por lo que se dejó en blanco el archivo *preboot.commands*. En *postdeploy.commands* se estableció que la aplicación por defecto es AUTH, ya que la misma es la encargada de gestionar el acceso al sistema. El archivo *postboot.commands* es el lugar en dónde se realizaron varias acciones, entre las que se encuentran la carga del conector y las bibliotecas que se copiaron a la imagen, la configuración de parámetros del protocolo HTTP y la definición de los recursos necesarios para las conexiones de las aplicaciones con las bases de datos.

El conjunto de aplicaciones del SIGEVA utiliza tres conexiones a las bases de datos. Cada una de estas conexiones se definen por medio de un *jdbc resource* el cual tiene asociado un *jdbc connection pool*. El código 3 es un ejemplo de los comandos asadmin utilizados para setear un *jdbc resource* y su correspondiente *jdbc connection pool* para cada una de las conexiones requeridas. Se debe reemplazar <POOLNAME> por el nombre del pool y <JDBCNAME> por el nombre del *jdbc resource*. En este código se indica que los valores

para las propiedades *URL*, *User* y *Password* se obtienen de variables de entorno con el formato *POOL_<POOLNAME>_<PROPERTY>*, siendo <PROPERTY> la propiedad a setear.

```
create-jdbc-connection-pool
--datasourceClassname=com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource
--resType=javax.sql.ConnectionPoolDataSource <POOLNAME>

set resources.jdbc-connection-pool.<POOLNAME>.property.URL=${ENV=POOL_<POOLNAME>_URL}
set
resources.jdbc-connection-pool.<POOLNAME>.property.Password=${ENV=POOL_<POOLNAME>_PASSWO
RD}
set resources.jdbc-connection-pool.<POOLNAME>.property.User=${ENV=POOL_<POOLNAME>_USER}

create-jdbc-resource --enabled=true --poolName=<POOLNAME> --target=domain <JDBCNAME>
create-resource-ref --enabled=true --target=server <JDBCNAME>
```

Código 3. Comandos asadmin para la definición de un *jdbc resource* y su correspondiente *jdbc connection pool*

4.4.3. Despliegue en Kubernetes

Para el despliegue de los contenedores necesarios se creó un namespace de nombre sigeva en el clúster Kubernetes de la PROTIC. Dentro de este namespace se definieron los objetos de Kubernetes encargados de gestionar el despliegue de los POD con sus respectivos contenedores.

El despliegue del contenedor que empaqueta las aplicaciones del SIGEVA se realizó por medio de un Deployment. En el mismo se indicó la imagen del contenedor a ejecutar, el puerto que utiliza la interfaz web de la consola de administración de Payara Server y el puerto en el cual el servidor atiende las solicitudes HTTP. También se definieron las variables de entorno que configuran las propiedades de las conexiones a las bases de datos, la que setea la contraseña de acceso a asadmin y la variable TZ para establecer la zona horaria; se configuraron las LivenessProbe y ReadinessProbe, la política de escalado, los límites y reservas de recursos (VCPUs y RAM) y el Pod Filesystem Group (Cloud Native Computing Foundation, 2022). Por último se montaron los archivos necesarios definidos en Secrets o ConfigMaps y el volumen para persistir los datos del sistema. En la figura 3 se observa el diagrama de montaje de estos recursos.

El archivo *apps.properties* se define en un Secret ya que contiene información sensible. El punto de montaje del Secret se realiza en los path

/opt/payara/appserver/glassfish/domains/domain1/config/apps.properties y */usr/local/glassfish/glasssfish/domains/domain1/config/apps.properties*.

Si bien el CONICET recomienda editar las configuraciones presentes en el archivo *apps.properties* solamente mediante la interfaz correspondiente en la aplicación, en esta arquitectura propuesta se realiza por medio de la modificación del Secret que define a dicho archivo. Esto se definió de esta manera para evitar el montaje de un volumen en el directorio config del servidor de aplicaciones (y así aprovechar la ya mencionada ventaja de la efemeridad de los contenedores). Al realizar este cambio, la interfaz de edición de estas configuraciones en la aplicación quedó prácticamente anulada ya que cualquier cambio que se realice por medio de la misma no se persiste ni se puede aplicar ya que requiere el reinicio del servidor, lo que provocaría la finalización de la ejecución del contenedor. Esta restricción del uso de la interfaz tiene un impacto mínimo ya que estas configuraciones rara vez se tienen que modificar y, en caso de ser necesario, es el personal técnico de la PROTIC el responsable de hacerlo.

El mecanismo de sincronización con SIGEVA-CONICET hace uso de un archivo del tipo Java KeyStore (JKS). Para montarlo, se creó un Secret a partir del archivo *keystore-sigeva-sincro.jks* presente en la VM original y se estableció el punto de montaje en el path */opt/payara/appserver/glassfish/domains/domain1/config/keystore-sigeva-sincro.jks*.

El archivo *default-web.xml* se lo definió en un ConfigMap y se estableció el punto de montaje en el path */opt/payara/appserver/glassfish/domains/domain1/config/default-web.xml*.

El SIGEVA almacena los archivos subidos al sistema en la carpeta *sigeva_files*. Entonces, para persistir estos archivos se montó un volumen CEPH en el path */opt/payara/sigeva_files*. Este path se define en la aplicación EVA, ya que desde el directorio de aplicaciones, esto es */opt/payara/appserver/glassfish/domains/domain1/applications/eva*, se utiliza el path *../../../../../../sigeva_files*, dando como resultado el path absoluto */opt/payara/sigeva_files*.

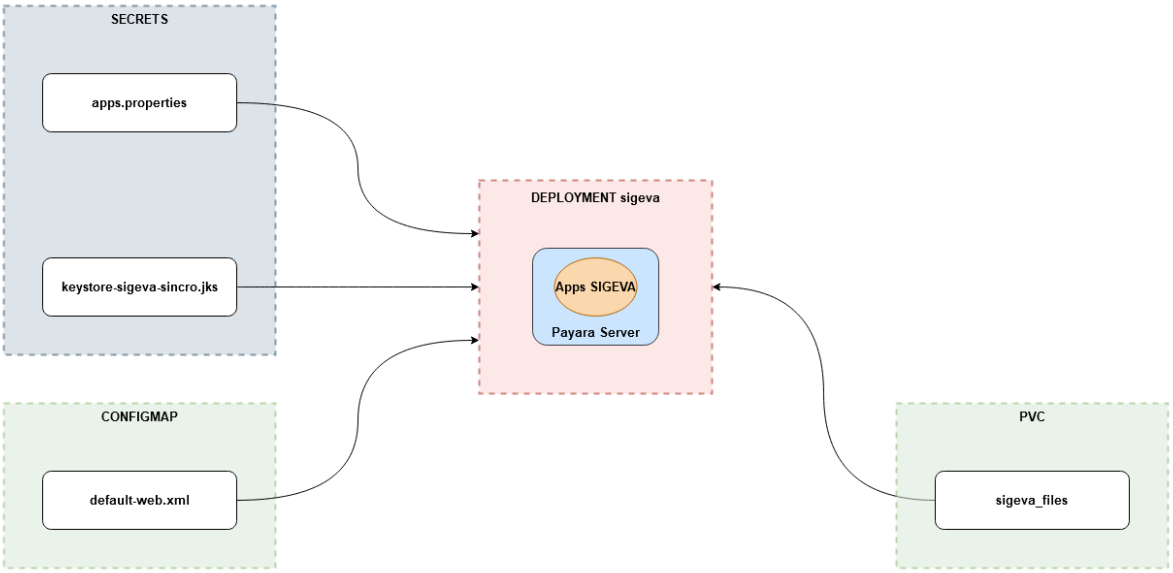

Figura 3 . Diagrama de Deployment Kubernetes para la imagen de las aplicaciones del SIGEVA.

El contenedor del webservice de validación de credenciales se desplegó por medio de un Deployment en el cual se indicó la imagen y el puerto utilizado. Para el contenedor de las bases de datos se empleó un StatefulSet (Cloud Native Computing Foundation, 2022), indicando la imagen a utilizar, el puerto mysql y se le montó un volumen CEPH para persistir los datos de las bases.

4.4.4. Evaluación en test

Al principio de la etapa de construcción de las imágenes las pruebas se realizaban en un entorno local, utilizando docker. Una vez que se consiguieron imágenes funcionales se configuró un ambiente de test en el clúster Kubernetes de la PROTIC, en el cual se probaron el funcionamiento de cada una de las imágenes construidas y las interacciones necesarias entre los contenedores.

Al momento de contar con imágenes estables del SIGEVA, se decidió configurar un ambiente de test (en el clúster Kubernetes) para que el área de Investigación probara el sistema e indique los resultados obtenidos. El resultado de esa prueba fue satisfactorio ya que desde el área de Investigación se informó el correcto funcionamiento de los módulos del SIGEVA, la ausencia de algún tipo de falla y, también, se destacó la fluidez en la

navegación por las secciones del mismo. Los detalles de esta prueba y las realizadas previamente en la PROTIC se asentaron formalmente en las tareas correspondientes en el servicio Colab.

4.5. Implementación en el ambiente de producción

Debido al éxito de las pruebas realizadas en el entorno de test, la PROTIC avaló la implementación de la arquitectura propuesta en el entorno de producción. Para llevar a cabo esta migración de manera ordenada y sin sobresaltos del SIGEVA de producción en VM a la arquitectura en contenedores, se definió y documentó un procedimiento a seguir, el cual consistió en una serie de pasos que se detallan a continuación.

Paso 1. Establecer fecha de migración y coordinación con el área de Investigación

Debido a que el SIGEVA va a estar fuera de servicio durante la migración se debe seleccionar un periodo adecuado para la realización de la misma. Para la implementación se optó por una semana en la cual no había ninguna convocatoria abierta y se realizó una comunicación con el área de Investigación para obtener el aval de la misma.

Paso 2. Bajar acceso de usuarios a la instancia de producción en la VM

Para evitar cualquier tipo de inconsistencia en los datos durante el proceso de migración es necesario dar de baja el acceso de los usuarios al sistema para que estos no puedan realizar cambios en sus respectivos bancos de datos que puedan quedar fuera de la sincronización entre la instancia antigua y la nueva.

El acceso público se dio de baja por medio del proxy reverso, haciendo que las solicitudes con servername “sigeva.unnoba.edu.ar” retornen un código HTTP STATUS 503. Como la cantidad de usuarios que acceden al sistema por la red interna es reducida, se los notificó para que no interactúen con el sistema mientras se realizaban las tareas de migración.

Paso 3. Copiar archivos “sigeva_files”

Se tienen que copiar al nuevo volumen los archivos del directorio *sigeva_files* de la instancia original. Esto se realizó desde la VM , montando el volumen CEPH definido por el

mecanismo de aprovisionamiento del clúster Kubernetes de la PROTIC y sincronizando los archivos con el comando *rsync -av*.

Paso 4. Crear los recursos kubernetes para el montaje de los archivos de configuración

Se definieron los Secrets para los archivos apps.properties y keystore-sigeva-sincro.jks y el ConfigMap para default-web.xml

Paso 5. Desplegar los contenedores en el clúster Kubernetes

Se definieron los Deployment para el contenedor de las aplicaciones del SIGEVA y para el contenedor del webservice de validación de credenciales. Además, se definió el StatefulSet para el contenedor MySQL de la base de datos.

Paso 6. Copiar los datos de la base de datos antigua a la nueva

Se realizó un dump de las bases de datos de la instancia antigua y se lo importó en el nuevo servidor MySQL.

Paso 7. Generar y aplicar los Ingressroutes necesarios

Se definieron y aplicaron los Ingressroutes para que Traefik pueda redirigir las solicitudes HTTP con servername “sigeva.unnoba.edu.ar” hacia el POD que ejecuta el contenedor del SIGEVA (o del webservice en el caso de las solicitudes relacionadas con la validación de credenciales).

Paso 8. Modificar el registro DNS

Se modificó el registro sigeva.unnoba.edu.ar en el DNS interno para que apunte hacia el Traefik del cluster kubernetes.

Paso 9. Comprobar el correcto funcionamiento de los procesos batch.

Se debe comprobar que la ejecución de los procesos batch se realice de manera correcta. Por cada ejecución de los procesos batch se almacena en los logs la fecha de inicio de la

misma, un detalle de lo realizado y si finalizó de manera correcta. El correcto funcionamiento se comprobó analizando estos logs, sin observar algún error producido.

Paso 10. Comprobar el funcionamiento del webservice de validación de credenciales

Se probó la autenticación del sistema y la misma se realizaba sin inconvenientes por lo que el webservice de validación de credenciales funcionaba correctamente.

Paso 11. Habilitar acceso en proxy reverso.

Para continuar con los pasos de la migración se debe permitir el acceso público. Para hacerlo se volvió a habilitar la configuración en el proxy reverso para que este realice el proxy pass de las peticiones con servername “sigeva.unnoba.edu.ar” hacia el Traefik del clúster Kubernetes.

Paso 12. Comprobar el correcto funcionamiento de la sincronización DESDE y HACIA SIGEVA-CONICET.

La comprobación de la sincronización de datos con SIGEVA-CONICET, al momento de realizar el presente trabajo, solo se puede hacer en el entorno de producción. Para probar la sincronización desde SIGEVA-CONICET a SIGEVA-UNNOBA se debe entrar a la instancia SIGEVA-UNNOBA, dirigirse al módulo banco de datos y utilizar la opción importar en alguna de las pestañas de datos disponibles. Para probar la sincronización desde SIGEVA-UNNOBA a SIGEVA-CONICET el procedimiento es similar, solo que en este caso se debe estar logueado en el SIGEVA-CONICET y al momento de importar se debe seleccionar la instancia SIGEVA-UNNOBA como origen de los datos.

Se realizaron varias pruebas desde ambos lados y todas funcionaron de manera correcta, sincronizando los datos correspondientes.

Paso 13. Configurar el backup del volumen de datos

Se configuró una ventana de backup vía snapshot de los volúmenes CEPH (tanto del que contiene al directorio *sigeva_files* como al de las bases de datos).

Paso 14. Notificar al área de Investigación la finalización de la migración

En este punto la instancia SIGEVA-UNNOBA se encuentra funcionando en la arquitectura propuesta. Se le informó al área de Investigación la finalización de las labores de migración y se le solicitó que realicen las pruebas que consideren necesarias. El área de Investigación realizó la evaluación pertinente del sistema e informó que el mismo se encontraba funcionando adecuadamente sin presentar algún tipo de inconveniente.

Paso 15. Asentar lo actuado

Se registró todo lo actuado en el marco de este procedimiento en la tarea correspondiente en el servicio Colab.

5. Conclusiones

En este trabajo se identificó un problema puntual en el área de la PROTIC el cual consistía en la degradación del rendimiento y estabilidad de la instancia SIGEVA-UNNOBA.

Partiendo del análisis de la situación inicial y la recopilación de documentación del sistema se logra elaborar una propuesta de mejoras, que se implementa de manera exitosa en un ambiente de test. Este resultado positivo permite obtener el aval para realizar la implementación en el ambiente de producción.

En la actualidad la instancia SIGEVA-UNNOBA cuenta con una arquitectura en contenedores, que se ejecuta en un cluster kubernetes, que permite escalabilidad y que logra subsanar los inconvenientes de estabilidad y desempeño existentes en el pasado, resaltando que estos cambios producidos en el ambiente productivo no han causado, al menos al momento de la redacción de este informe, algún tipo de fallas o problema de incompatibilidad. Otro punto a destacar es el uso de un servidor de aplicaciones que cuenta con actualizaciones periódicas y un soporte activo de la comunidad con la posibilidad de contratar un soporte de grado empresarial si así la situación a futuro lo amerite.

Hoy la navegación por las distintas secciones de la instancia SIGEVA-UNNOBA se realiza de manera fluida, no se producen ni los tiempos excesivos de espera de respuesta ni los períodos fuera de servicio que caracterizaban a la instancia, lo que agiliza la realización de trámites u otras operaciones y, en definitiva, mejora la experiencia de uso del sistema.

Como trabajo a futuro, la estrategia utilizada en la instancia SIGEVA-UNNOBA se puede generalizar a las demás aplicaciones de la PROTIC que usan Glassfish y así obtener mejoras en la gestión del rendimiento de las mismas.

6. Bibliografía

Ceph Foundation. (2022). Ceph. Ceph.io. Retrieved October 18, 2022, from https://ceph.io

Cloud Native Computing Foundation. (2022). Kubernetes Documentation. Retrieved October 18, 2022, from https://kubernetes.io/docs/home/

Cloud Native Computing Foundation. (2022). Prometheus - Monitoring system & time series database. Retrieved October 18, 2022, from https://prometheus.io/

CONICET. (2014). SIGEVA - Migración de Bases (4). https://sigeva.conicet.gov.ar/wp-content/uploads/2014/03/SIGEVA-Migraci%C3%B3n-Bases_v4.pdf

CONICET. (2016). SIGEVA - Actualización de Versiones (5). https://sigeva.conicet.gov.ar/wp-content/uploads/2016/09/SIGEVA-Guia-actualizacion-versiones_v5.pdf

CONICET. (2019). SIGEVA - Configuración de Batchs (6). https://sigeva.conicet.gov.ar/wp-content/uploads/2019/08/SIGEVA-Instructivo-configuraci%C3%B3n-Batch_v6.pdf

CONICET. (2019). SIGEVA - Requerimientos Técnicos (5). https://sigeva.conicet.gov.ar/wp-content/uploads/2019/04/SIGEVA-Requerimientos-tecnicos_v5.pdf

Docker Inc. (2022). Docker Compose. Retrieved October 19, 2022, from https://docs.docker.com/compose/

EGroupware GmbH. (2022). Groupware Software | Online Collaboration tools. Retrieved October 18, 2022, from https://www.egroupware.org

Grafana Labs. (2022). Grafana Loki OSS | Log aggregation system. Grafana. Retrieved October 18, 2022, from https://grafana.com/oss/loki/

Grafana Labs. (2022). Grafana OSS | Metrics, logs, traces, and more. Grafana. Retrieved October 18, 2022, from https://grafana.com/oss/grafana/

OpenInfra Foundation. (2022). OpenStack. OpenStack: Open Source Cloud Computing Infrastructure. Retrieved October 18, 2022, from https://www.openstack.org/

Oracle. (s.f.). GlassFish. Retrieved October 18, 2022, from https://javaee.github.io/glassfish/

Payara Foundation. (2022). Payara Server. Retrieved October 18, 2022, from https://www.payara.fish/products/payara-platform-community/

Resolución del Rector 5398 (2012). Aprobar el Convenio a suscribirse entre la Universidad Nacional del Noroeste de la Provincia de Buenos Aires, y el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET). UNNOBA, Pergamino, octubre de 2012.

Resolución del Rector 953 (2019). Establecer la obligatoriedad para todos los docentes de la UNNOBA de la carga y actualización de sus antecedentes en el Sistema Integral de Gestión y Evaluación (SIGEVA). UNNOBA, Pergamino, octubre de 2019.

SUSE Group. (2022). *Managed Kubernetes Cluster Operations*. Rancher. Retrieved October 18, 2022, from https://www.rancher.com/products/rancher

7. Acrónimos

Acrónimo	Descripción
CONICET	Consejo Nacional de Investigaciones Científicas y Técnicas
Java EE	Java Platform, Enterprise Edition
JDK	Java Development Kit
JKS	Java KeyStore
PPS	Práctica Profesional Supervisada
PROTIC	Prosecretaría TIC de la Universidad Nacional del Noroeste de la Provincia de Buenos Aires
PV	Persistent Volume
PVC	Persistent Volume Claim
SIGEVA	Sistema Integral de Gestión y Evaluación
UBA	Universidad de Buenos Aires
UNNOBA	Universidad Nacional del Noroeste de la Provincia de Buenos Aires
VM	Máquina virtual (Virtual Machine)

8. Agradecimientos

A todas las personas que formaron parte de esta etapa de mi vida.

A mi familia, en especial a mi madre y a mi padre, quienes me inculcaron los valores del estudio, por el gran apoyo, confianza y afecto que siempre me brindan.

A mi tío y mis abuelos y abuelas, a quienes siempre recordaré con mucho cariño.

A mis amigos y amigas por acompañarme durante todo este proceso.

A Javier por guiarme y aconsejarme durante el desarrollo de este trabajo.

A los integrantes de la PROTIC, en especial a Hugo y al equipo de infraestructura.

Simplemente ¡gracias!.