



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Herramientas para la implementación de prácticas en la
nube mediante Jupyter Notebooks

Trabajo Fin de Máster

Máster Universitario en Gestión de la Información

AUTOR/A: Sánchez Sancho, Jorge

Tutor/a: Moltó Martínez, Germán

CURSO ACADÉMICO: 2022/2023

Resumen

Normalmente, las materias relacionadas con la ingeniería informática realizan prácticas de laboratorio, que requieren el uso de cierto software que permita la resolución de estas. Este software requiere una instalación y configuración en los equipos de los estudiantes, que en algún caso puede ocasionar problemas de configuración.

Un patrón común durante la realización de las prácticas es el seguimiento por parte del alumno de un boletín en formato PDF que los docentes publican. Esto implica que los resultados obtenidos por el alumno tengan que ser anotados en otro tipo de soporte, ya sea físico o digital.

Este trabajo persigue la integración de los cuadernos Jupyter Notebooks para facilitar la realización de prácticas de laboratorio, especialmente en aquellos entornos que requieren el uso de recursos remotos de proveedores Cloud. Tiene el objetivo de transicionar de los tradicionales boletines de prácticas en formato PDF a un mecanismo en el que los boletines sean accesibles vía web, y permitir la ejecución de comandos remotos mediante credenciales delegadas por parte del alumno. Para ello, se estudiarán diferentes técnicas para exponer dichos boletines mediante JupyterLab, Google Colab o Binder.

Además, se analizarán las limitaciones de extensibilidad de estos boletines, para decidir su futura integración en asignaturas relacionadas con la docencia sobre proveedores Cloud, poniendo especial interés en Amazon Web Services.

Palabras clave: Nube, Jupyter, Amazon Web Services, Kubernetes

Summary

Normally, subjects related to computer engineering carry out laboratory practices, which require the use of certain software that allows them to be resolved. This software requires some type of installation and configuration on student computers, which in some cases, may cause configuration problems.

A common pattern during the process of completing practices is the student following a bulletin in PDF format that teachers publish. This implies that the results obtained by the student have to be recorded in another type of support, whether is a physical or digital one.

This project seeks for the integration of Jupyter Notebooks to simplify the process of completing laboratory practices, especially in those environments that require the use of remote resources from Cloud providers. The objective is to be able to transition from the traditional practice bulletins in PDF format to a mechanism in which the bulletins are accessible via the web and allow the execution of remote commands through delegated credentials for each student. To achieve this, different techniques will be studied to expose these bulletins using JupyterLab, Google Colab or Binder

In addition, the extensibility limitations of these bulletins will be analyzed, to decide their future integration in subjects related to teaching about Cloud providers, with a special interest in Amazon Web Services.

Keywords: Cloud, Jupyter, Amazon Web Services, Kubernetes

Tabla de contenidos

Índice de figuras	7
1. Introducción.....	10
1.1 Motivación	11
1.2 Objetivos	11
1.3 Impacto esperado	12
1.4 Estructura de la memoria	12
2. Situación actual.....	13
3. Proyecto Jupyter	14
3.1. Descripción de los cuadernos	15
3.2 Requisitos para su implementación en asignaturas	16
3.3 Pruebas para comprobar los requisitos	17
4. Herramientas relacionadas	21
4.1 JupyterLab	21
4.2 JupyterHub	22
4.3 Google Colab	23
4.4 Binder	24
5. Resultados obtenidos	26
5.1 JupyterLab	26
5.2 JupyterHub	27
5.3 Google Colab	28
5.4 Binder	29
5.5 Elección de la herramienta	29
6. Kubernetes	32
6.1 Introducción	32
6.2 Definición.....	32
6.3 Componentes	33
6.4 Pods.....	34
7. Amazon Web Services	36
7.1 Introducción	36
7.2 Servicios necesarios	37
7.2.1 Servicio de gestión de identidades	37

7.2.2	Servicio de resolución de nombres de dominio	37
7.2.3	Servicio de creación de instancias	38
8.	Casos de estudio	39
8.1	JupyterHub sobre una máquina en AWS	39
8.2	JupyterHub sobre una máquina física.....	44
8.3	JupyterHub sobre Kubernetes en una máquina física	47
8.4	Boletín de prácticas en un cuaderno.....	53
9.	Conclusiones	56
10.	Relación del trabajo realizado con los estudios cursados	57
11.	Trabajos futuros	58
12.	Referencias	59
13.	Anexo	61



Índice de figuras

Figura 1. Proyecto Jupyter con soporte a varios lenguajes	14
Figura 2. Productos desarrollados por el proyecto Jupyter	15
Figura 3. Instrucciones que inician la instalación de librerías.....	17
Figura 4. Fragmento de código para crear una conexión remota	18
Figura 5. Fragmento de código para mostrar una consola interactiva	18
Figura 6. Conexión con éxito a un equipo remoto	19
Figura 7. Comandos para comprobar el funcionamiento de la librería de Amazon	20
Figura 8. Logo de la herramienta JupyterHub.....	22
Figura 9. Recursos computacionales de la capa gratuita de Colab	23
Figura 10. Miembros de la federación de BinderHub	24
Figura 11. Interfaz de la implementación gratuita de Binder.....	25
Figura 12. Servidores ocupados de Binder	25
Figura 13. Interfaz Gráfica de JupyterLab.....	26
Figura 14. Interfaz gráfica de Google Colab	28
Figura 15. Aplicaciones replicadas en un clúster de Kubernetes	33
Figura 16. Esquema de comunicaciones entre pods.....	34
Figura 17. Esquema de un servicio basado en direcciones IP	35
Figura 18. Imagen del sistema operativo Ubuntu para servidores	40
Figura 19. Creación de un par de claves de seguridad.....	41
Figura 20. Creación de un grupo de seguridad para la instancia.....	42
Figura 21. Creación de la unidad de almacenamiento	42
Figura 22. Fragmento de código que inicia la instalación de JupyterHub	43
Figura 23. Configuración de la instancia concluido con éxito	43
Figura 24. Configuración de las carpetas de la máquina virtual.....	45
Figura 25. Configuración de los recursos de la máquina virtual.....	45
Figura 26. Configuración del almacenamiento virtual de la máquina.....	46
Figura 27. Selección de una instalación minimizada de Ubuntu	46
Figura 28. Selección de la instalación del servicio SSH	46
Figura 29. Comando de instalación de JupyterHub en local	47
Figura 30. Comando para iniciar la instalación de Microk8s	48
Figura 31. Comando para obtener los nodos del clúster	49
Figura 32. Información sobre los nodos del clúster	49
Figura 33. Comandos para instalar los servicios de Microk8s.....	49
Figura 34. Comando de instalación del gestor de paquetes	49
Figura 35. Activación del servicio de balanceo de carga	50
Figura 36. Instalación de los servicios de almacenamiento del clúster	50
Figura 37. Fichero de configuración del almacenamiento	51
Figura 38. Comando para aplicar una política de almacenamiento	51
Figura 39. Comandos para reparar el gestor de paquetes.....	52
Figura 40. Instrucción para instalar JupyterHub	52
Figura 41. Comando para localizar pods en un clúster	52



Figura 42. Obtención de las direcciones expuestas por el proxy	53
Figura 43. Comando que actualiza la configuración	53
Figura 44. Ejemplo de una celda de tipo Markdown	54
Figura 45. Ejemplo de una celda de tipo código	54
Figura 46. Visualización de una memoria en formato Markdown.....	55

1. Introducción

Los estudios universitarios son cada vez más demandados por los jóvenes como una formación superior que les permita desarrollar su carrera profesional. Esto es debido, principalmente, a que permiten certificar que una persona ha adquirido los conocimientos básicos para ejercer una profesión. Cada titulación universitaria cuenta con un plan de estudios que determina las materias que los alumnos tendrán que superar para obtener la titulación.

Las materias que forman parte de los mencionados planes de estudio están estructuradas en una parte teórica, en la que se desarrollan los conceptos, y una parte práctica, en la que el alumnado realiza una serie de pruebas de laboratorio para poner en práctica los conocimientos adquiridos. La parte práctica de una asignatura consta de un número determinado de sesiones de laboratorio, y por cada una de ellas, el docente publica un boletín a modo de enunciado o guía que el alumno debe seguir para completarlas. Una vez completadas las sesiones, suele ser necesario que el alumno realice la entrega de una memoria con los resultados obtenidos.

En titulaciones como la ingeniería informática, la parte práctica de las materias puede necesitar el uso de cierto software que permita su resolución. Esto puede llegar a ocasionar problemas de configuración o instalación en los equipos de los estudiantes que, en el peor de los casos, podría producir un borrado o pérdida accidental de los resultados obtenidos. Ante esta situación, surge la necesidad de encontrar una herramienta que facilite la realización de las prácticas mediante la integración del contenido multimedia de los boletines con la ejecución de código de tal forma que esté siempre accesible vía web. De esta forma, se conseguiría eliminar los problemas derivados de la instalación de software que puede ser potencialmente complejo.

Con el reciente aumento de popularidad en el uso de los servicios proporcionados por los proveedores de la nube, se valorará positivamente que la herramienta permita interactuar con ellos e incluso que permita la ejecución de comandos remotos mediante credenciales delegadas por parte del alumno, poniendo especial interés en la nube pública de Amazon conocida como Amazon Web Services.

Este trabajo analizará los cuadernos Jupyter Notebooks para determinar si es una herramienta adecuada para resolver los problemas planteados y para decidir su futura integración en materias relacionadas con la docencia sobre proveedores Cloud.

1.1 Motivación

En las asignaturas que forman parte del plan de estudios de titulaciones universitarias como la ingeniería informática, suele existir una separación entre el boletín que publica el docente con los enunciados de las prácticas y las sucesivas memorias que se deben entregar al finalizar cada una de las sesiones de laboratorio para demostrar el trabajo realizado e incluso mostrar los resultados obtenidos en cada una de ellas. A esta separación es necesario sumarle la problemática que puede darse cuando es requerido un determinado software que le sirva al estudiante para resolverlas en sus equipos pudiendo dar lugar a situaciones que dificulten la resolución de estas a tiempo.

Teniendo eso en cuenta, surge la necesidad de encontrar una herramienta que permita optimizar el proceso de resolución de las prácticas dando lugar a una transición de los boletines tradicionales a un formato en el que los boletines estén siempre accesibles vía web e incluso que permita la ejecución de comandos para su resolución. De esta forma se conseguiría evitar los problemas derivados de la instalación de software e incluso permitiría que el material estuviera siempre disponible. Por ello, este proyecto tiene la intención de realizar un estudio que permita determinar si es posible realizar una futura integración de los populares Jupyter Notebooks en asignaturas, en especial aquellas relacionadas con la ciencia de datos o con los proveedores de la nube pública.

1.2 Objetivos

El objetivo principal de este proyecto es realizar un estudio comparativo de las diferentes herramientas que permiten interactuar con los Jupyter Notebooks para determinar si es posible realizar una integración de dichos cuadernos en asignaturas universitarias relacionadas con la docencia sobre ciencia de datos o los proveedores cloud. Para lograr esto, será necesario superar los siguientes requisitos:

- Recabar información sobre el origen de los cuadernos, así como del proyecto que da soporte tanto a ellos como a las herramientas que permiten interactuar con ellos.
- Recopilar una lista de las principales herramientas que soportan los cuadernos.
- Determinar los requisitos necesarios que permitan determinar qué herramienta es la más adecuada para integrarse.
- Realizar un estudio comparativo de cada una de las herramientas para decidir cuál se adapta mejor a los requisitos para ser integrada.
- Realizar un breve estudio de las tecnologías que dan soporte a la herramienta candidata a ser implementada en asignaturas.

1.3 Impacto esperado

Con la realización de este proyecto se pretende dar con una herramienta que pueda integrarse en un futuro cercano en alguna de las asignaturas del plan docente de titulaciones afines a la informática, en especial en aquellas que estén relacionadas con la ciencia de datos, explotación de datos masivos, programación o incluso aquellas que estén relacionadas con los proveedores de la nube pública.

Es cierto que la implementación en determinadas asignaturas puede ser laborioso pero seguro que existe alguna asignatura que, con un poco de trabajo y dedicación, se podría incorporar a modo de prueba piloto. Si se lograra, se podrían dinamizar las prácticas e incluso solventar problemas de configuración en aquellos programas que usan los estudiantes para resolverlas.

1.4 Estructura de la memoria

El contenido de la memoria está organizado en apartados en los que se detallan los aspectos importantes del proyecto. En el primer apartado se realiza una descripción de la situación actual en la que se detalla la estructura que siguen las prácticas de las asignaturas relacionadas con la docencia universitaria, en especial aquellas titulaciones que forman a ingenieros.

En el siguiente apartado se describe el alcance que el proyecto Jupyter ha tenido con sus herramientas en el campo de la docencia, investigación y en ciencia de datos. Entre sus productos, se destaca la importancia de los cuadernos, así como de las herramientas que permiten interactuar con ellos. A continuación, se detallan cuáles son los requisitos necesarios para que se puedan implementar en asignaturas. Una vez especificados, se podrá realizar un análisis posterior de varias de las herramientas que permiten interactuar con este tipo de cuadernos con el fin de determinar cuál es la idónea para una futura integración.

Tras determinar cuál es la aplicación que mejor se adapta al entorno docente, se realiza una introducción a los conocimientos técnicos necesarios para entender el funcionamiento de una tecnología de orquestación de contenedores que permitirá hacer despliegues a gran escala. Incluso, se destina un apartado entero a realizar una breve introducción a los servicios que oferta Amazon en su nube pública para facilitar un futuro despliegue en su nube.

En los siguientes apartados, analizarán varios escenarios de despliegue para determinar las ventajas e inconvenientes de desplegar la herramienta en sus dos modalidades de funcionamiento tanto en una instalación local como una en la nube. Después, como último apartado, se realizan unas conclusiones sobre qué tipo de despliegue puede ser el más apropiado para una asignatura en el que se propondrán futuras mejoras para incrementar la experiencia de usuario.

2. Situación actual

En los campos relacionados con la explotación de datos masivos, ciencia de datos e incluso la investigación, suele ser necesario el uso de herramientas que permitan combinar diferentes elementos multimedia, como texto, imágenes o audio, con la posibilidad de ejecutar fragmentos de código para crear documentos que permitan recrear el mismo comportamiento, independientemente de donde se use. De esta forma, se crean documentos dinámicos con los que multitud de personas pueden interactuar para realizar tareas de descubrimiento, análisis o interpretación de resultados. La cualidad de estos documentos que más se valora, es la capacidad de reproducir un mismo comportamiento con los datos que contiene.

En las asignaturas que forman parte del plan de estudios de la ingeniería informática, en especial aquellas que están vinculadas con la ciencia de datos, la explotación de datos masivos o incluso aquellas que están relacionadas con los proveedores de la nube pública, podrían beneficiarse de este tipo de documentos. Desde hace tiempo, ha existido una separación entre los boletines que los docentes entregan a los alumnos, los programas que los estudiantes usan para resolver dichos boletines y las memorias que deben entregar. La adopción de estas nuevas herramientas podría beneficiar tanto a los docentes como a los estudiantes ya que permitirían eliminar la necesidad de instalar software o de preparar entornos de desarrollo que emplearían los estudiantes en las aulas de laboratorio. Se conseguiría, por tanto, evitar problemas de configuración o de instalación. Incluso, con un correcto mecanismo de copias de seguridad, se protegería a los estudiantes del borrado accidental de las prácticas.

Estas herramientas proporcionarían al profesor un lugar en el que pudiesen distribuir el contenido de las prácticas u otro material docente a los estudiantes para que ellos mismos comenzasen a trabajar sobre ellos. Sin duda una mejora que merece la pena estudiar para ver si sería viable su implementación en alguna asignatura.

3. Proyecto Jupyter

Es un proyecto [1] sin ánimo de lucro creado para desarrollar software de código abierto que permita dar soporte a la computación interactiva en el ámbito de la ciencia de datos y computación científica a través de todos los lenguajes de programación. Este proyecto se desarrolla de forma transparente en su repositorio oficial de GitHub para que la comunidad pueda participar de forma abierta tanto para incrementar su funcionalidad como para detectar errores. Sus orígenes se deben a su creador Fernando Pérez que es un desarrollador de software que en el año 2014 decidió partir de la base del proyecto IPython. En su afán por dar soporte a varios lenguajes de programación, ha conseguido hasta la fecha dar soporte en su proyecto a más de 40 lenguajes.



Figura 1. Proyecto Jupyter con soporte a varios lenguajes

Entre todos los productos que forman parte en este proyecto, cabe destacar la creación de su producto estrella que ha dado vida a los populares cuadernos conocidos por la comunidad como Jupyter Notebooks. Estos cuadernos destacan debido a que permiten la creación de documentos computacionales que pueden ser compartidos de forma verdaderamente sencilla entre usuarios. Se trata de un documento especial que suele requerir el uso de una herramienta que permita interactuar con ellos que se puede instalar de forma local en los equipos de los usuarios que, una vez puesta en marcha, ofrece una interfaz web para interactuar con ellos.

La experiencia de los desarrolladores que forman parte del proyecto ha permitido que a lo largo del tiempo se hayan podido lanzar diferentes herramientas que interactúan con los cuadernos. Cada una destaca por proporcionar tener un enfoque distinto a la hora de compartir los documentos e incluso en ofrecer soporte para la edición simultánea de varios usuarios en un mismo documento o la posibilidad de dar soporte a un gran número de usuarios en una misma instalación de la herramienta.

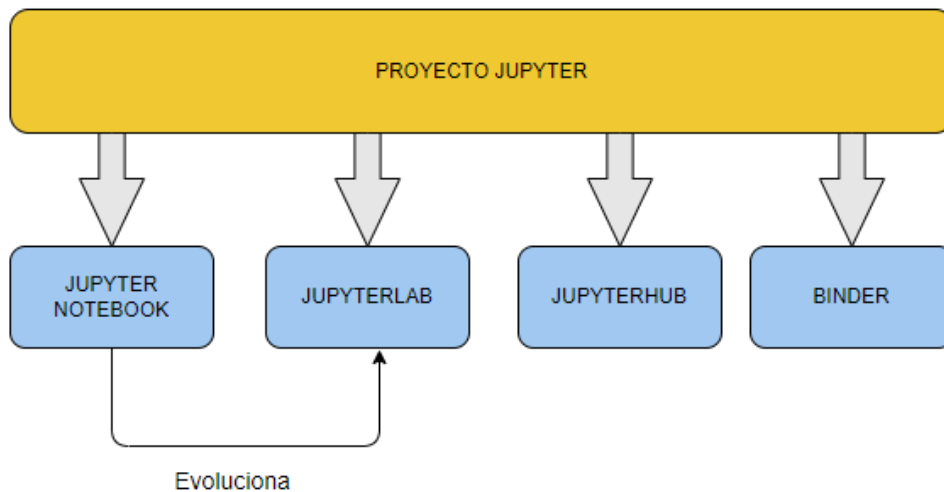


Figura 2. Productos desarrollados por el proyecto Jupyter

Como se puede observar en la figura inmediatamente superior, la primera herramienta que se lanzó al público recibió el mismo nombre que los cuadernos para facilitar que los usuarios asociasen los cuadernos a la herramienta que permitía interactuar con ellos. Observando el éxito de la herramienta, procedieron a desarrollar la siguiente versión de la misma, pero ofreciendo una interfaz renovada con un diseño modular que permitiese ampliar sus funcionalidades nativas mediante el uso de extensiones que se añaden a la herramienta. Después, a petición de la amplia comunidad de usuarios, lanzaron al público una nueva herramienta preparada para dar soporte de forma nativa a entornos multiusuario con el nombre de JupyterHub. Incluso crearon un proyecto dentro del proyecto Jupyter para crear una herramienta que permitiese acceder a los cuadernos desde una dirección web única para que varios usuarios pudiesen interactuar al mismo en un mismo cuaderno que recibió el nombre de Binder.

3.1. Descripción de los cuadernos

Los cuadernos que han sido creados por el proyecto Jupyter son un tipo de documento especial que permite combinar elementos multimedia como texto, audio, imágenes, o vídeos con fragmentos de código escritos en varios lenguajes de programación. Incluso permiten la ejecución de código al mismo tiempo que realiza una visualización de los resultados generados por la ejecución del mismo. De esta forma, se crea un documento completamente interactivo que permite ser modificado en cualquier momento. A pesar de admitir varios lenguajes de programación, el más empleado hasta la fecha es Python.

Por su versatilidad en el momento de combinar elementos, su popularidad continua en aumento en campos importantes como el análisis de datos, ciencia de datos e incluso campos tan relevantes como la investigación. En cuanto a su estructura, cabe destacar que el contenido de estos documentos se organiza en bloques que reciben el nombre de celdas dentro de la herramienta que interactúa con ellos. Cada una de las celdas puede ser de uno de los siguientes tipos:

- Texto explicativo que puede estar formateado con lenguajes de marcado.
- Fragmentos de código que podrá ser ejecutado las herramientas.
- Resultados obtenidos tras la ejecución del contenido de las celdas de código.

Es importante destacar que en el momento en el que se ejecuta una celda de código, se crea una nueva celda completamente nueva que mostrará el resultado de la operación justo en la celda inmediatamente inferior a la celda de código. Si la celda de resultados ya existe cuando se lanza a ejecución una celda de código, se refresca el contenido con el nuevo resultado obtenido.

Un aspecto positivo de utilizar este tipo de documentos es que tiene un formato que es ampliamente soportado por multitud de herramientas. Eso se traduce en que puede ser exportado a multitud de formatos diferentes entre los que destacan HTML, PDF, Latex o Markdown. Esta característica lo convierte en el documento perfecto para generar memorias o crear documentos que permitan reproducir un comportamiento concreto. Esta cualidad está bastante valorada dentro de la comunidad científica ya que permite generar documentos que reproducen el mismo comportamiento independientemente de donde se use el documento.

Estos cuadernos se pueden utilizar tanto con las herramientas oficiales que están en la página web del proyecto publicadas como con las herramientas de terceros que están siendo respaldadas por compañías que brindan soporte a los cuadernos. Ahora mismo, existen multitud herramientas que permiten interactuar con los cuadernos, pero las más empeladas por la comunidad de usuario son las siguientes:

- JupyterLab
- JupyterHub
- Google Colab
- Binder

3.2 Requisitos para su implementación en asignaturas

En este apartado se detallan los requisitos que se deben cumplir entre las herramientas y los cuadernos para que puedan ser implementados en asignaturas, especialmente en aquellas relacionadas con la docencia sobre los proveedores cloud que se centren en los servicios en la nube de Amazon, utilizado a modo de ejemplo para el desarrollo de este trabajo.

El primer requisito que se debe cumplir se centra en la necesidad de usar librerías que brinden la posibilidad de ejecutar comandos remotos en las máquinas desplegadas en la nube de Amazon, mediante el uso de conexiones SSH combinadas con credenciales, que se delegarán a los usuarios. Para cumplir con esta restricción, se ha valorado el uso de una librería nativa de Python que permita crear conexiones SSH seguras mediante el uso de consolas interactivas. La librería, conocida como Paramiko, ha sido seleccionada tanto por su simplicidad como por la extensa documentación que existe en su sitio web.

Como segundo requisito, la herramienta debe permitir hacer uso de la librería oficial de Amazon (AWS CLI) para poder interactuar de forma directa con todos los servicios disponibles en su nube. En principio, este requisito no debería ser complicado cumplirlo debido a que se encuentra disponible para varios lenguajes de programación. Su importancia recae en que permite realizar llamadas a la API de AWS para gestionar los recursos que cada alumno tiene disponibles en su cuenta de usuario de forma remota. El mantenimiento de la misma está siendo gestionado por la propia empresa que lanza de forma periódica actualizaciones que corrigen errores en su repositorio [2] oficial de GitHub. Similar a otras librerías, ésta también cuenta con una documentación bastante completa que contiene un listado completo con las instrucciones [5] que acepta tanto con la explicación como con un ejemplo para un mejor entendimiento.

Por último, aunque no menos importante, se valorará positivamente la capacidad de la herramienta para ser accedida desde cualquier navegador web en los equipos de los estudiantes con el fin de evitar que tengan que recurrir a la instalación de programas en sus equipos que puedan requerir de una configuración posterior.

3.3 Pruebas para comprobar los requisitos

Para determinar si las herramientas cumplen con los requisitos, se va a proceder a la creación de un conjunto determinado de pruebas que se repetirán en cada una de ellas para poder realizar una comparación completamente justa.

La primera de ellas consiste en la instalación de las librerías que permiten tanto crear conexiones seguras [3] mediante SSH como interactuar directamente con la API de la nube de Amazon [4]. Esta comprobación se realizará mediante la ejecución de dos comandos que inician el proceso de instalación de cada una de ellas. Si por algún motivo la propia herramienta mostrase un mensaje con un error, eso significaría que no son capaces de dar soporte a las librerías por lo que inmediatamente serían descartadas.

```
[ ] pip install paramiko
```

```
[ ] pip install awscli
```

Figura 3. Instrucciones que inician la instalación de librerías

En caso contrario, la herramienta devolvería un mensaje mostrando que la instalación ha sido exitosa. Demostrando que el entorno es capaz de ejecutar las instrucciones que incorpora la librería en su interior. Una vez comprobado que las herramientas pueden instalar las librerías requeridas, es necesario comprobar, en el caso de la librería que permite crear conexiones SSH, que es posible crear una consola interactiva que permita interactuar con equipos remotos. En este caso, se ha creado un pequeño fragmento de código que permite crear una consola que el usuario puede emplear para ejecutar instrucciones en un ordenador remoto.

```
[ ] import paramiko
import time

host = '*****'
port = 22
username = '*****'
password = '*****'
command = '****'

sshClient = paramiko.SSHClient()
sshClient.set_missing_host_key_policy(paramiko.AutoAddPolicy())
sshClient.connect(host, port, username, password)

channel = sshClient.get_transport().open_session()
channel.get_pty()
channel.invoke_shell()
channel.send("stty -echo\n")
```

Figura 4. Fragmento de código para crear una conexión remota

En la primera parte del código se importa la propia librería que se ha instalado junto con otra librería nativa que permitirá interactuar con instrucciones relacionadas con el tiempo. Tras la importación, se crea una zona para que el usuario pueda introducir los datos de conexión de la máquina junto con las credenciales necesarias iniciar sesión en ella. Después se crea una conexión segura con los datos que el usuario ha detallado, para crear una consola interactiva en las siguientes líneas de código.

```
while True:
    command = input()
    if command == 'exit':
        break

    channel.send(command + "\n")

    while True:
        if channel.recv_ready():
            output = channel.recv(1024)
            print (output.decode("utf8"))
        else:
            time.sleep(5)
            if not(channel.recv_ready()):
                break

sshClient.close()
```

Figura 5. Fragmento de código para mostrar una consola interactiva

En el fragmento de código aparece al final un bucle que, mientras no se introduzca la palabra de salida, seguirá ejecutando instrucciones en el equipo remoto para mostrar la


```
[ ] ! export AWS_CONFIG_FILE=/path/to/config_file  
  
[ ] !aws --version  
  
[ ] !aws sts get-session-token --duration-seconds 900  
  
[ ] !aws ec2 describe-instances  
  
[ ] !aws ec2 describe-instances \  
    --filters Name=instance-type,Values=m5.large  
  
[ ] !aws ec2 describe-instances help
```

Figura 7. Comandos para comprobar el funcionamiento de la librería de Amazon

Tras superar el proceso de inicio de sesión, se ha preparado un listado de instrucciones para comprobar que efectivamente el usuario está conectado. Entre las instrucciones que se muestran en la figura superior, se deben probar con especial interés aquellas que devuelven un listado de las instancias que se encuentran actualmente en ejecución junto con la variante que realiza la misma consulta, pero con unos filtros para que no devuelvan todas las máquinas. Si tras la ejecución de las dos instrucciones se devuelven dos listados, eso significa que la librería es completamente compatible tanto con los cuadernos como con la herramienta.

4. Herramientas relacionadas

En este apartado se realizará un análisis de las herramientas mencionadas en apartados anteriores para determinar si cumplen con los requisitos de integración en asignaturas. Es importante destacar que, aunque todas las herramientas que se mencionan en este estudio permiten interactuar con los cuadernos de forma similar, cada una realiza una implementación diferente que aporta características adicionales que pueden marcar la diferencia.

En primera instancia, se realizará una descripción de las herramientas para detallar las características que incorpora e incluso obtener información sobre su implementación. Una vez obtenida esta información, se pondrán a prueba para ver qué requisitos son los que cumplen cada una de forma individual. De esta forma, se podrá llegar a unos resultados que determinen la opción que se adapta mejor a los requisitos.

4.1 JupyterLab

JupyterLab es una herramienta de código abierto que forma parte del Proyecto Jupyter que ha sido creada para renovar su versión anterior conocida como Jupyter Notebook que ya permitía interactuar con los cuadernos. Se caracteriza principalmente por ofrecer una interfaz gráfica completamente renovada que se ha basado en los diseños web más actuales. La nueva interfaz es bastante similar a la anterior a excepción del panel lateral que muestra el sistema de archivos.

Cuenta con una estructura completamente modular que permite a los usuarios abrir varios cuadernos, archivos u otros componentes de la herramienta como si se tratasen de pestañas de un navegador dentro de la misma ventana ofreciendo de esta forma una experiencia mejorada similar a un entorno de desarrollo. Además, esta nueva versión incorpora otras novedades como el soporte para la colaboración en tiempo real, permitiendo que múltiples usuarios puedan trabajar conjuntamente en un mismo cuaderno de forma simultánea. Resulta de especial interés destacar que en esta versión se ofrece un soporte nativo a un gran número de formatos de archivo, así como el uso bajo demanda de terminales y consolas disponibles desde la propia interfaz.

Es importante remarcar que JupyterLab ha sido diseñado para ser extensible para que los usuarios tengan la capacidad de incrementar su funcionalidad mediante la creación de extensiones para personalizarse el entorno. Además, debido a su compatibilidad con el núcleo de Jupyter Notebook junto con sus ficheros, es posible realizar una transición de una herramienta a otra sin problemas.

4.2 JupyterHub

JupyterHub [6] es una herramienta de código abierto que forma parte del proyecto Jupyter y que ha sido creada para ofrecer un sitio web, ofreciendo entornos de desarrollo independientes entre sí, que permiten interactuar con los notebooks de Jupyter para un elevado número de usuarios concurrentes. Se trata de la versión para usuarios concurrentes de JupyterLab, que se ha detallado en la sección anterior.



Figura 8. Logo de la herramienta JupyterHub

El administrador del servidor sobre el que se ha desplegado la herramienta se encarga de realizar la configuración de los recursos [7] que se asignarán a los usuarios y de gestionar el acceso a los documentos, llegando incluso a implementar inicios de sesión avanzados [8] que se integren con terceros. Todas estas configuraciones e incluso otras que no se han mencionado, se realizan en un mismo fichero de configuración [9] que solo la persona con permisos puede modificar. Al final, es una herramienta que brinda una personalización bastante alta.

La herramienta está preparada para ser desplegada sobre máquinas, tanto físicas como virtuales, e incluso para estar en ejecución en los proveedores cloud. No obstante, antes de realizar un despliegue, conviene llevar a cabo un análisis que permita determinar el número de usuarios concurrentes, así como los requisitos de cómputo necesarios para dar soporte a todos los usuarios. Con estos datos, se puede tomar una decisión más acertada sobre qué tipo de instalación es la más adecuada.

La instalación reducida está pensada para un grupo concurrente, entre cero y cien usuarios, que se instala en un único servidor, ya sea físico o virtual, o puede estar alojado en un proveedor cloud. Mientras que la instalación completa está pensada para entornos que superen los cien usuarios simultáneos, cabe la posibilidad de estar desplegada sobre un número variable de servidores, mediante el uso de Kubernetes. Aunque esta opción es posible realizarla sobre máquinas físicas o virtuales, lo más habitual es hacerlo en los proveedores cloud.

4.3 Google Colab

Google Colaboratory [10] o Colab es un servicio gratuito [11] ofrecido por la empresa Google, que proporciona un entorno de ejecución para poder hacer uso de los Jupyter Notebooks desde cualquier navegador web, siendo la propia empresa la encargada de proporcionar los recursos computacionales que se encuentran en su nube. Según Google, la herramienta está pensada para la inteligencia artificial, ciencia de datos y la docencia, pero en la práctica es una versión de Jupyter Notebooks que se ejecuta directamente en la nube.

El único requisito para poder utilizar la herramienta es poseer una cuenta de usuario. Una vez obtenida ésta, el usuario puede comenzar a utilizar la herramienta en su capa gratuita, que le brinda acceso a unos recursos computacionales limitados, pero que pueden ser ampliados en cualquier momento mediante una suscripción de pago. Entre los recursos que brinda la herramienta se encuentra el almacenamiento, memoria, unidades de cómputo, unidades de procesamiento gráfico y las unidades de procesamiento tensor para la inteligencia artificial.

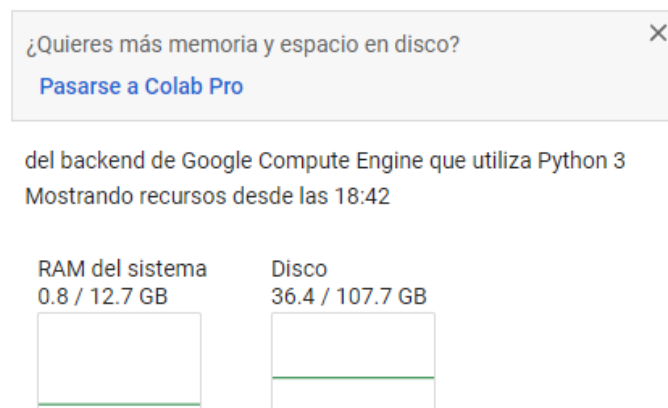


Figura 9. Recursos computacionales de la capa gratuita de Colab

La consecuencia directa de utilizar cuentas de Google es que el usuario puede compartir de forma sencilla los cuadernos con otros usuarios que también estén registrados, e incluso beneficiarse de la edición simultánea dentro de un mismo cuaderno. Ambas opciones pueden ser más seguras si el propietario de los cuadernos establece los permisos de acceso adecuados para cada uno de los usuarios con los que comparte los cuadernos.

La herramienta también destaca por su compatibilidad con varias fuentes de datos, al permitir que los usuarios puedan tener almacenados sus cuadernos en sus equipos, en repositorios de GitHub o incluso en el almacenamiento en la nube de la propia empresa conocido como Google Drive. No es de extrañar que este servicio sea compatible con otros servicios de la empresa, ya que siempre ha existido una fuerte compatibilidad dentro del ecosistema que ofrece.

4.4 Binder

Binder [12] es una herramienta que forma parte del proyecto Jupyter que permite crear un sitio web para compartir recursos computacionales con los usuarios con el objetivo de interactuar con los cuadernos. Su propósito es que todos los usuarios puedan disponer de un entorno para abrir cuadernos de tal forma que puedan ser compartidos mediante un enlace de colaboración. Esta característica permite que el usuario pueda compartir sus cuadernos con prácticamente cualquier persona que tenga acceso a un navegador con conexión a internet.

Como la herramienta se diseñó para dar soporte a un gran número de usuarios, se optó por hacer uso de los conocimientos adquiridos de sus aplicaciones hermanas para ser implementada con Kubernetes. Este diseño permite en gran medida realizar una gestión bastante eficiente de cada una de las sesiones de usuario que se crean. Como se pensó para estar siempre disponible en internet, se recomienda hacer un despliegue en los proveedores cloud, pero también es posible realizar uno sobre máquinas físicas propias.

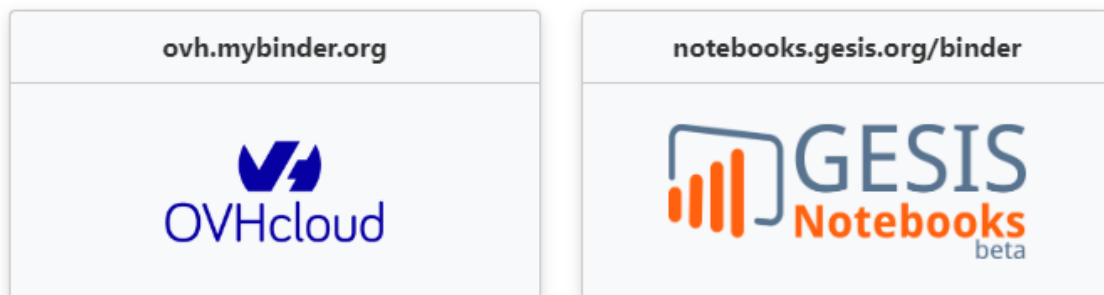


Figura 10. Miembros de la federación de BinderHub

En la actualidad, todavía existe una implementación gratuita [13] en la nube que está siendo mantenida por la propia comunidad de Binder. Todo el soporte computacional que recibe se obtiene de una federación [14] de despliegue en dos principales proveedores [15] cloud. Se trata de una federación que cualquiera puede donar dinero u ofrecer sus equipos para donar potencia de cómputo sobrante. No obstante, el objetivo de este despliegue no es más que el de servir como un servicio público para que los usuarios prueben la herramienta, al mismo tiempo que sirve para hacer una demostración de todas las funcionalidades que puede llegar a ofrecer.

Un aspecto bastante negativo de la herramienta es la necesidad de que los cuadernos se encuentren almacenados en repositorios públicos, como es el caso de GitHub, que esté configurado para ser público. Si los cuadernos no se encuentran en uno que sea estrictamente público, la herramienta no es capaz de acceder a ellos. Este factor se puede considerar grave cuando se emplean cuadernos que almacenan en su contenido credenciales de acceso como sería el caso en aquellas asinaturas que se conecten a recursos remotos.

The image shows the Binder interface for building and launching a repository. It features three input fields: 'GitHub repository name or URL' with a dropdown menu set to 'GitHub', 'Git ref (branch, tag, or commit)' set to 'HEAD', and 'Path to a notebook file (optional)' with a 'File' dropdown. A prominent orange 'launch' button is at the bottom.

Figura 11. Interfaz de la implementación gratuita de Binder

Como se puede apreciar en la figura inmediatamente superior, la interfaz de Binder es bastante sencilla para que el usuario sólo tenga la obligación de rellenar los campos con la información relacionada con el repositorio que contiene los cuadernos. Tras rellenar los campos, el servicio comienza a preparar el entorno interactivo para visualizar los cuadernos junto con el enlace que permite compartirlo con cualquier individuo.

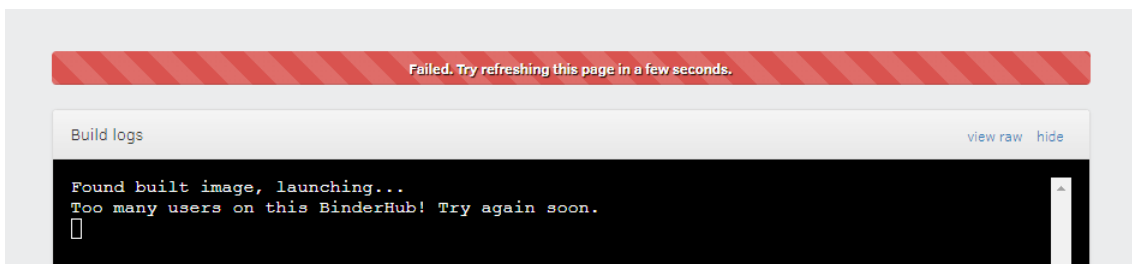


Figura 12. Servidores ocupados de Binder

Como se ha mencionado previamente, al ser un servicio que se ofrece de forma gratuita al público general, es posible que los recursos de la federación se encuentren saturados temporalmente por un alto número de usuarios conectados de forma concurrente. Es decir, que los servidores encargados de dar soporte a este despliegue estén saturados provocando que no puedan dar respuesta a las nuevas peticiones. En estas situaciones, cada vez que se intenta abrir un cuaderno mostrará un mensaje de error al usuario como el que se muestra en la figura anterior.

5. Resultados obtenidos

En este apartado se mostrarán los resultados obtenidos tras realizar una comprobación de los requisitos sobre cada una de las herramientas haciendo uso de las pruebas que se han detallado anteriormente.

5.1 JupyterLab

Siguiendo el mismo orden en el que se han presentado las aplicaciones, la primera que se ha analizado es JupyterLab. Lo primero que destaca es su interfaz web que sí ha sido renovada considerablemente respecto a su versión anterior al mostrar un diseño que recuerda a las páginas web actuales. Una característica bastante esperada ha sido la aparición de un panel lateral en el que se muestra el sistema de archivos. Esta incorporación permite encontrar los documentos que el usuario necesita para gestionar de forma considerablemente más rápida. No obstante, una de sus funciones más esperadas ha sido la posibilidad de tener varios cuadernos abiertos haciendo uso del nuevo sistema de pestañas.

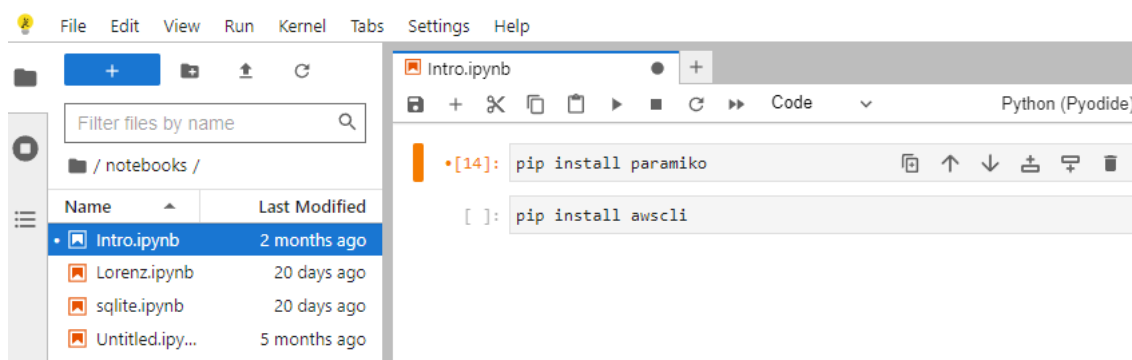


Figura 13. Interfaz Gráfica de JupyterLab

Tras un vistazo a las funcionalidades que se han incorporado, es el momento de realizar las pruebas de compatibilidad para determinar si es una candidata que implementar en alguna asignatura. La primera tanda de pruebas se ha realizado sobre la versión de demostración que se encuentra disponible en el sitio web de la herramienta y, cuenta con la peculiaridad de que se ejecuta en el propio navegador del alumno. Durante estas pruebas, la herramienta no ha sido capaz de superarlas, ya que no permitía realizar una instalación de las librerías. Cada vez que se intentaba instalar una de ellas, devolvía un mensaje con varios errores en el núcleo de la aplicación que provocaban retrasos que bloqueaban la instalación.

Para no descartar todavía la aplicación, se optó por realizar una instalación en el equipo del estudiante para determinar si era un problema de la versión online debido a que se ejecuta en el navegador. Tras finalizar el proceso de instalación, el acceso a la aplicación sigue siendo mediante un navegador web en el equipo del estudiante. Una vez se pudo

acceder de nuevo, se repitieron las pruebas y, en este caso, sí que permitían instalar las librerías e incluso superó todas las pruebas definidas.

No obstante, a pesar de haber superado todas las pruebas, esta aplicación no es la más adecuada debido a que para usarla sin problemas de funcionamiento es necesaria una instalación en el equipo del estudiante cuando lo que se busca es que esté disponible las 24 horas del día para todos los estudiantes vía web. De esta forma, sólo está disponible para el alumno que la tiene instalada en su equipo.

5.2 JupyterHub

JupyterHub ha sido la siguiente en analizarse para ver si es la candidata perfecta para ser integrada ya que es una de las aplicaciones más recomendadas por la comunidad de usuarios.

Antes de proceder con los resultados, es conveniente recordar que esta aplicación ha sido pensada para ser la versión para usuarios concurrentes de JupyterLab. No obstante, se puede realizar una instalación para un uso personal como para aquellos entornos en los que varios usuarios necesitan tener acceso. En la mayoría de casos se suele realizar una instalación en un equipo local, aunque también es posible hacerlo en un proveedor cloud. La elección de una opción u otra suele recaer en la disponibilidad de contar con equipos físicos para hacer la instalación.

Cuando un usuario se conecta a la herramienta, lo primero que puede observar es que la interfaz se parece considerablemente a su hermana pequeña. Cuenta con el mismo sistema de pestañas para tener abiertos varios cuadernos al mismo tiempo junto con el panel lateral que hace las veces de sistema de archivos. Incluso se dedica en la parte inferior de la ventana del navegador para indicar los recursos que se están empleando.

Un aspecto importante a la hora de utilizar esta herramienta es la gran cantidad de opciones de configuración que admite para realizar una customización prácticamente al detalle. Entre sus opciones de configuración cabe destacar la posibilidad de limitar los recursos que un usuario puede usar en sus cuadernos, el establecimiento de inicios de sesión avanzados con posibilidad de integrarse con terceros o la posibilidad de crear carpetas compartidas para que los docentes puedan almacenar el material en ellas con el fin de que los alumnos tengan acceso al material docente de forma sencilla.

En cuanto a las pruebas realizadas, JupyterHub ha sido capaz de superarlas sin ningún tipo de problema. La instalación de librerías se ha completado en un tiempo menor que con JupyterLab mientras que, con el resto de pruebas, no se ha notado ninguna diferencia. Teniendo en cuenta que ha superado las pruebas sin problemas y que es la herramienta que ofrece una gama configuraciones bastante amplia, se convierte por ahora en la mejor opción.

5.3 Google Colab

La siguiente herramienta que se ha analizado ha sido Google Colab para comprobar lo que puede llegar a ofrecer un servicio cuyo acceso es completamente gratuito para el público que consta de una implementación propia de la versión en la nube del proyecto Jupyter.



```
[ ] host = '*****'
    port = 22
    username = '*****'
    password = '*****'
    command = '****'

sshClient = paramiko.SSHClient()
sshClient.set_missing_host_key_policy(paramiko.AutoAddPolicy())
sshClient.connect(host, port, username, password)

channel = sshClient.get_transport().open_session()
channel.get_pty()
channel.invoke_shell()
channel.send("stty -echo\n")

while True:
    command = input()
    if command == 'exit':
        break
```

Figura 14. Interfaz gráfica de Google Colab

Lo que se encuentra un usuario nada más acceder al servicio es una interfaz que resulta bastante similar si es comparada con JupyterLab. Cuenta con un panel lateral bastante parecido que permite, entre otras cosas, acceder al sistema de archivos del servicio para gestionar los documentos disponibles. Una funcionalidad que resulta innovadora si se compara con el resto de herramientas analizadas, es la aparición en la parte derecha de unos gráficos que indican la cantidad de recursos disponibles para que el usuario pueda destinarlos a sus cuadernos. No es de extrañar que añadan esta funcionalidad ya que los usuarios empiezan con una capa gratuita que cuenta con unos recursos de cómputo que pueden ser fácilmente ampliados mediante una suscripción de pago. No obstante, no es la única funcionalidad de la barra lateral derecha, ya que ofrece, entre otras cosas, opciones de depuración de código para usuarios avanzados.

En cuanto a las pruebas que se han realizado, la herramienta las ha superado todas con éxito. Esto es debido principalmente a que cada usuario cuenta con un entorno de ejecución aislado en el que puede instalar todas las librerías que necesite para usarlas en sus cuadernos. El único aspecto que se puede considerar malo es que, si el usuario se desconecta, el entorno que el usuario tenía dedicado se borra tras un tiempo prudencial

para liberar recursos computacionales de la nube de Google. Esto implica que, cada vez que se conecte, debe usar los comandos que instalan las librerías para poder volver a utilizarlas.

5.4 Binder

Binder ha sido la última aplicación en ser analizada para determinar si cumple con los requisitos. Es importante destacar que las pruebas se han realizado sobre la versión que se encuentra disponible de forma gratuita en su sitio web. Se ha procedido de esta forma debido a que el objetivo de esa implementación es la de ofrecer un entorno real para que los usuarios comprueben la potencia de la herramienta, aunque también sería posible hacer una instalación en local para hacer las pruebas.

Lo primero que se encuentra un usuario cuando accede a la página web oficial es una interfaz de usuario bastante sencilla. Dicha interfaz se compone de un formulario que, una vez rellenados con los datos del cuaderno, inicia una ventana en el navegador que permite interactuar con el cuaderno.

En cuanto a las pruebas, Binder ha sido capaz de superarlas sin problema ninguno. Lo único que se ha podido apreciar es que tanto la instalación como la ejecución de código ha funcionado de una forma más lenta que en las otras herramientas. Esto podría estar relacionado con que Binder estuviera dando soporte a un gran número de usuarios en ese momento, por lo que no tendría muchos recursos disponibles.

5.5 Elección de la herramienta

La selección de la herramienta se realizará en base a los requisitos que se han definido en esta memoria e incluso estudiando aquellas características distintivas de las propias aplicaciones que se adaptan a entornos multiusuarios dentro de la docencia.

Uno de los requisitos más importantes que se han establecido, es que las herramientas no deben requerir la instalación de la misma en los equipos de los estudiantes con el fin de evitar problemas derivados de sus instalaciones. Teniendo esto en cuenta, la primera herramienta en ser descartada es JupyterLab, debido principalmente a que su uso está ligado a una instalación obligatoria en los equipos de los estudiantes. Con este descarte, la lista de candidatas se reduce, dando paso a aquellas que están diseñadas para dar un soporte multiusuario a través de un navegador web.

La siguiente candidata que se ha tenido en cuenta ha sido Google Colab debido a que es un servicio gratuito que ya se encuentra desplegado en la nube pública de Google. Uno de sus puntos fuertes es la capacidad de compartir cuadernos con otros usuarios para poder colaborar de forma simultánea en ellos. En este aspecto, se encuentran bastante avanzados al disponer de un buen sistema de gestión de permisos de usuario. De esta

forma, el dueño de un cuaderno puede limitar las acciones que otros usuarios pueden realizar sobre los cuadernos que comparte.

En términos de almacenamiento, la herramienta es bastante flexible al admitir varias formas de llevarlo a cabo. Permite utilizar documentos que se encuentren en el propio equipo del estudiante, que se encuentren en repositorios tanto públicos como privados e incluso, como era de esperar, que se encuentren en su servicio de almacenamiento en la nube conocido como Google Drive. Teniendo en cuenta todo lo mencionado sobre Colab, se posiciona como una de las opciones ganadoras.

La aplicación que intentaría quitarle el puesto a Colab sería Binder al ser otro servicio que también se ofrece de forma gratuita en la nube. Una diferencia considerable entre ambas es la disponibilidad del servicio. Mientras Colab está siempre disponible por el respaldo que tiene detrás al estar desplegada sobre uno de los mayores proveedores de la nube pública, Binder debe conformarse con estar casi siempre disponible. En algunos momentos, Binder suele experimentar problemas de disponibilidad porque los recursos computacionales de los que dispone no son suficientes para dar soporte al gran número de usuarios que se conectan. Esta diferencia hace que no sea tan buena opción como sí que lo es Colab.

Después, en temas de compartición de archivos, Colab permite realizar una asignación de permisos mientras que, en Binder, cualquier persona que reciba el enlace que actúa como una invitación, puede editar el documento como si fuese el propietario. Este es un comportamiento que no se adecúa a entornos docentes. Además, con estas desventajas comentadas, es necesario añadir que en Binder sólo es posible trabajar con cuadernos que se encuentren en repositorios públicos para que la aplicación los pueda abrir. Esto puede suponer problemas si el alumno necesita trabajar con documentos almacenados en su ordenador o si los cuadernos guardan credenciales de usuario. Incluso corre el riesgo de que cualquier persona modifique esos documentos sin contar con el permiso del dueño. Con todos estos puntos negativos respecto a Colab, Binder tiene que ser descartada a pesar de haber superado las pruebas con éxito.

JupyterHub se plantea ahora mismo como la única opción que puede quitar el puesto a Colab. Esto no sería nada extraño debido a que la herramienta ha sido diseñada para funcionar en entornos educativos siendo uno de los principales motivos, la facilidad que proporciona a los docentes para compartir el material [\[16\]](#) académico. Para ello dispone de tres formas de llevarlo a cabo.

- Usando un repositorio de GitHub como repositorio principal para publicar cada uno de los contenidos que el docente tenga que distribuir a los estudiantes. Con este método, cada alumno podría disponer del material en su carpeta personal. Permite que los alumnos obtengan siempre las versiones de los documentos actualizadas incluidos los posibles cambios que haga el profesor. Los alumnos sólo tendrían que pulsar un enlace que el docente proporciona para que todos los datos del repositorio se actualicen.
- Creando una carpeta compartida de sólo lectura para los alumnos. Esta es una buena opción si se cuenta con archivos grandes ya que evita que los usuarios se lo descarguen varias veces en su carpeta principal con los consecuentes problemas de almacenamiento. Con esta opción, sólo quedaría añadir un enlace simbólico para pudiera aparecer en la carpeta principal de cada usuario.

- Crear una carpeta compartida para que todos los usuarios puedan compartir cuadernos y otros ficheros. Sería interesante poder limitar el almacenamiento máximo en esa carpeta para no llenar el almacenamiento de la máquina sobre la que despliega. En este caso, también sería necesario crear un enlace simbólico para que aparezca en la carpeta principal de cada usuario.

Además, también ofrece otras opciones de configuración que permiten aumentar aún más la seguridad como la posibilidad de contar con un inicio de sesión avanzado con la posibilidad de integrarse con terceros, carpeta privada para cada uno de los usuarios o la posibilidad de limitar los recursos computacionales que puede usar un usuario. Se puede apreciar desde el primer momento que es una herramienta completamente apta para entornos docentes.

Un aspecto en el que también es superior es en la posibilidad de que el docente instale las librerías que emplearán sus alumnos. Con esto se consigue que los alumnos no se vean en la obligación de instalar las librerías cada vez que usen los cuadernos porque ya estará disponibles desde el primer momento. Además, como su diseño es modular, es posible instalar extensiones de terceros para ampliar todavía más sus funciones.

Como se puede apreciar, JupyterHub es superior en varios aspectos en los que Colab no puede competir, por lo que la candidata que mejor se adapta a los entornos docentes es sin duda, JupyterHub.



6. Kubernetes

En este apartado se va a realizar una introducción a los conocimientos teóricos básicos de Kubernetes para entender su funcionamiento. El propósito de este apartado es el de comprender los componentes internos de Kubernetes para poder hacer un despliegue correcto de una instancia de JupyterHub sobre un clúster de Kubernetes.

6.1 Introducción

Las aplicaciones que proporcionan soporte a un número variable de usuarios se suelen desarrollar para tener la capacidad de ejecutarse en un número variable de servidores de forma simultánea. De este modo, se intenta proporcionar una alta disponibilidad a los usuarios, al mismo tiempo que se gestiona de forma eficiente los recursos computacionales. El principal inconveniente que este tipo de aplicaciones tiene, es que puedan funcionar correctamente en unos servidores, pero que no sean capaces de hacerlo en otros, ya sea por problemas de configuración o incompatibilidad.

Docker es una herramienta que soluciona este problema porque permite ejecutar aplicaciones de manera contenerizada. Un contenedor [17] es un entorno de ejecución aislado creado especialmente para una aplicación que contiene todas las dependencias que necesita para funcionar. De esta forma, basta con replicarlo en otros servidores que utilicen Docker para obtener siempre el mismo comportamiento de la aplicación. Su principal diferencia respecto a una máquina virtual reside en que la máquina virtual tiene que cargar un sistema operativo completo mientras que los contenedores sólo necesitan cargar el núcleo del sistema operativo, eliminando así todos los componentes que son innecesarios.

Una consideración a tener en cuenta es que, si el número de servidores que tienen en ejecución contenedores aumenta de forma considerable, controlarlos de forma manual puede convertirse en una tarea ardua. Por este motivo, es necesario el uso de alguna herramienta específica que permita orquestar contenedores.

6.2 Definición

Kubernetes [18] es una herramienta de orquestación de contenedores de código abierto que permite controlar los aspectos relacionados con la implementación, escalado e incluso la administración de aplicaciones contenerizadas a través de un gran número de nodos o servidores. Entre sus principales beneficios cabe destacar que permite proporcionar alta disponibilidad a las aplicaciones de forma automática mediante la creación de réplicas de los contenedores en otros servidores. Esto se hace de forma transparente al usuario ya que es capaz de redirigir el tráfico que recibe una réplica de la

aplicación que no funciona a otra que funciona correctamente. Además, también permite escalar de forma automática una aplicación, en caso de que varíe el tráfico que recibe o porque se busca tener una mayor capacidad de cómputo para hacer frente a un incremento en el tráfico.



Figura 15. Aplicaciones replicadas en un clúster de Kubernetes

6.3 Componentes

Los componentes que forman parte de Kubernetes se pueden agrupar en dos grupos en función de las tareas que realizan para que todo funcione sin fallos. Los componentes que forman parte del plano de control son los encargados de tomar decisiones globales sobre un clúster de nodos entre los que se encuentran:

- El API Server se encarga de exponer una interfaz para que diferentes clientes puedan interactuar con Kubernetes.
- El Controller Manager es el responsable de monitorizar los nodos que están en funcionamiento para mantener un número adecuado de contenedores en cada
- El Scheduler está pendiente de los pods que no están asignados a ningún nodo para realizar un proceso de selección de nodo para entrar en ejecución.
- La base de datos etc es una base persistente y distribuida que almacena toda la información del clúster, incluyendo todas aquellas configuraciones internas.

Los componentes que forman parte de los nodos corren en cada uno de los nodos del clúster para mantener a los pods en funcionamiento.

- El agente Kubelet se mantiene siempre en ejecución con la misión de garantizar que los contenedores estén en ejecución en el pod correspondiente.
- Un servicio de Proxy que se encarga de gestionar el reenvío de conexiones entre los nodos que forman parte de un mismo clúster.
- El entorno de ejecución de contenedores es el software es el responsable de que los contenedores puedan entrar en ejecución.

6.4 Pods

Un pod es un conjunto formado por uno o más contenedores que comparten recursos computacionales tanto de red como de almacenamiento del nodo en el que se están ejecutando. Esto permite que todos los contenedores que se estén ejecutando dentro de un pod puedan ser accedidos desde otros nodos haciendo uso de la dirección IP que tiene asignada el pod. También existe la posibilidad de definir volúmenes específicos de almacenamiento para que sean compartidos entre los contenedores del pod.

Cuando los contenedores entran en ejecución, lo más habitual es que sean asignados a un único pod, pero no es una limitación. En situaciones avanzadas, puede ser de gran utilidad hacer uso de varios contenedores en un mismo pod para intentar repartir las tareas que realiza una aplicación.

Un aspecto importante a tener en cuenta es que los pods son volátiles, es decir, no están pensados para estar siempre en ejecución. Esta situación se puede dar cuando dentro de un nodo, un pod entra en un estado irrecuperable de error. Como no es posible recuperar su estado anterior, simplemente se elimina del clúster, liberando los recursos asignados para ser reemplazado por una réplica idéntica. Otra situación que podría darse es que se lanzase una nueva versión de una aplicación que provocase una destrucción de los pods antiguos para que sean reemplazados por los nuevos.

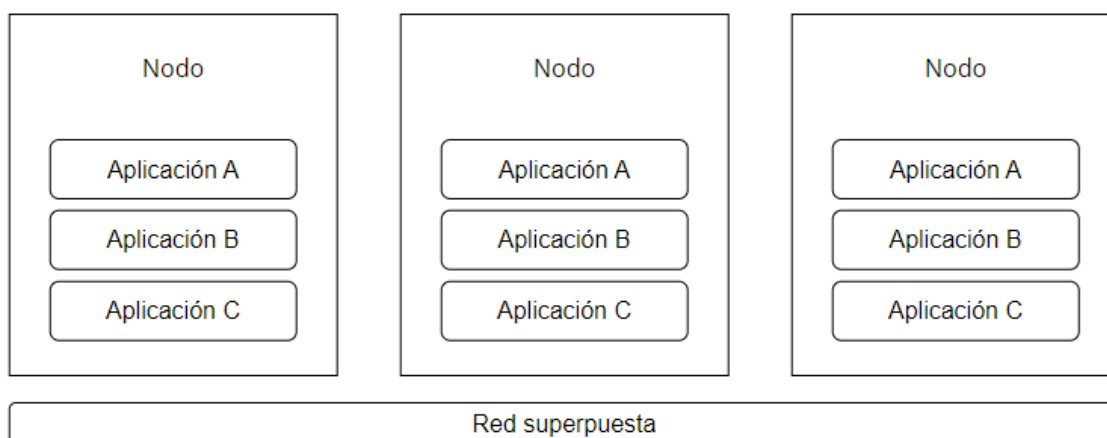


Figura 16. Esquema de comunicaciones entre pods

Las comunicaciones entre los pods se realizan mediante una red superpuesta que es compartida entre todos los nodos del clúster. De esta forma, un pod puede comunicarse con cualquier otro pod independientemente del nodo en el que se encuentre. Como se ha mencionado antes, los pods son volátiles, por lo que no es recomendable acceder entre ellos usando la dirección IP que tienen asignada porque en cualquier momento puede dejar de estar asignada, creando de esta forma fallos en las comunicaciones.

Para evitar problemas de comunicación, se suele hacer uso de unos servicios especiales que gestionan de forma eficiente la comunicación entre pods. Los más utilizados son los basados en direcciones IP o los que proporcionan un balanceador de carga.

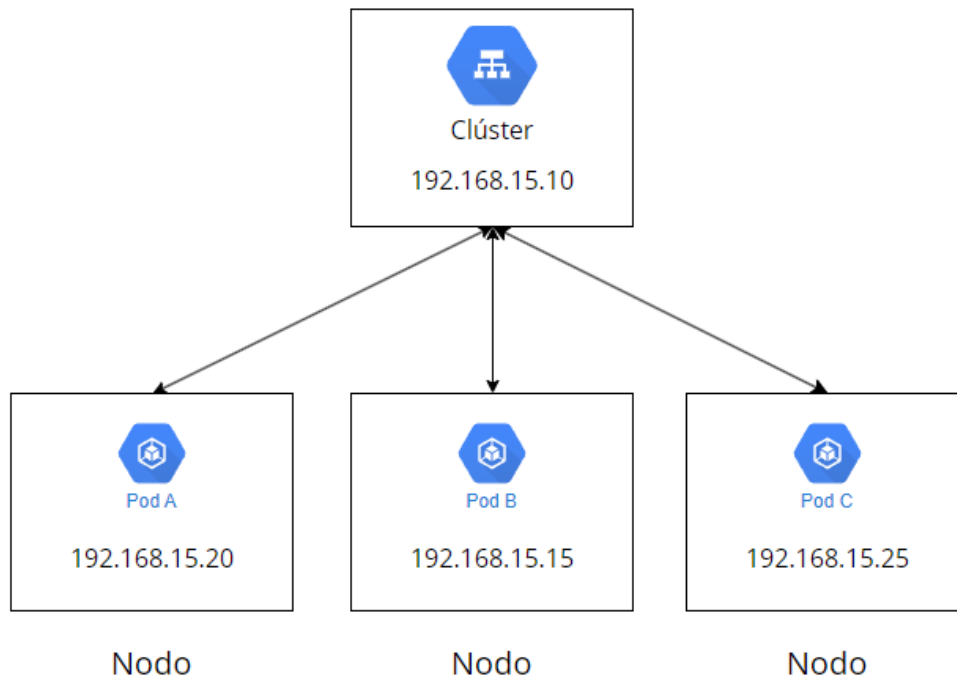


Figura 17. Esquema de un servicio basado en direcciones IP

El servicio basado en direcciones IP se caracteriza por tener una dirección IP asignada que nunca cambia. Esto permite que un pod se comunique directamente con el servicio en vez de comunicarse directamente con otro pod. El servicio es capaz de localizar internamente a los otros pods mediante un sistema de etiquetas que son asignadas a cada uno de los pods que forman parte de la misma aplicación.

Por otro lado, el servicio de balanceo de carga funciona de forma similar al servicio anterior, a diferencia de que permite establecer reglas para distribuir las peticiones de las aplicaciones entre los pods que estén registrados dentro del servicio con la misma etiqueta asignada.

7. Amazon Web Services

En este apartado se realizará una introducción teórica a los servicios en la nube que el gigante tecnológico Amazon ofrece al público. Después, se procederá a explicar cuáles son los servicios que se podrían utilizar para hacer un despliegue de JupyterHub.

7.1 Introducción

Amazon Web Services cuyas siglas son AWS, es la nube pública que está gestionada por Amazon. Se caracteriza por poner a disposición del público general, una colección de servicios de computación en la nube accesibles a través de internet. Actualmente es una de las plataformas más empleadas por las empresas por su gran variedad de servicios, cuya cifra asciende a más de 200, y por su modelo de facturación que se basa en el pago por el uso de sus servicios. Esto se traduce en una reducción de costes considerable.

Al tratarse de una nube pública, cualquier persona tiene la posibilidad de registrarse para activar una cuenta de usuario que le permita empezar a utilizar todos los servicios que ofrece dentro de su amplio catálogo. En cada una de las cuentas se ofrece un nivel o capa gratuita, que permite a los usuarios iniciar pruebas sin coste adicional de los servicios, siempre que no se superen los límites especificados de cada servicio para no salirse de la capa gratuita. No obstante, en caso de superar los límites de un servicio en su nivel gratuito, se aplicarán las tarifas estándar del servicio en función del uso que se le haya dado.

Dentro de la capa gratuita se pueden diferenciar tres tipos de ofertas, aquellas que ofrecen pruebas gratuitas a corto plazo, las que ofrecen pruebas gratuitas durante un año y las ofertas que son siempre gratuitas.

Los servicios que cuenten con una oferta de prueba a corto plazo podrán ser usados durante un periodo de tiempo determinado e incluso podría darse el caso de que sólo se pudieran utilizar una única vez de forma gratuita. Por otro lado, los servicios con una oferta gratuita durante un año permiten que los usuarios puedan usarlas sin coste con la condición de que no superen sus límites. En este tipo de ofertas, la activación se hace de forma automática en cuanto un usuario activa su cuenta. Mientras que los servicios que dispongan de una oferta siempre gratuita se podrán usar siempre que la cuenta de usuario siga activa y no se superen los límites de la oferta. Fuera de las ofertas, los servicios de la nube de Amazon siguen un modelo de facturación de pago por uso. Eso quiere decir que mientras no se usen sus servicios, no se facturará nada en la cuenta del usuario.

7.2 Servicios necesarios

En este apartado se detallarán los servicios que se pueden emplear para desplegar una instancia de JupyterHub en una máquina virtual en la nube de Amazon.

7.2.1 Servicio de gestión de identidades

Cognito [19] es un servicio que permite administrar identidades de usuario de una forma centralizada con varias opciones de seguridad. Este servicio permite entre otras cosas realizar un seguimiento de las operaciones de registro e inicio de sesión para llevar a cabo un control de acceso de los usuarios para aplicaciones tanto web como móviles. Su capa gratuita permite gestionar sin coste hasta 50.000 identidades de usuarios activos cada mes.

Este servicio se implementaría para llevar un control de los usuarios que pueden hacer uso de la herramienta JupyterHub. Si se optase por no implementar este servicio, se tendría que emplear el gestor de usuarios que proporciona de forma predeterminada JupyterHub.

7.2.2 Servicio de resolución de nombres de dominio

Route 53 [20] es un servicio web que proporciona un sistema de resolución de nombres de dominio para gestionar las peticiones DNS de los usuarios hacia las aplicaciones web. Su uso permitiría que los usuarios pudiesen acceder al entorno de JupyterHub con el uso de un alias sencillo para evitar tener que acceder usando la dirección IP pública que tiene asignada JupyterHub.

Desafortunadamente, este servicio no cuenta con ninguna oferta que permita ser usado en un nivel gratuito. En caso de usarlo, se realizaría una facturación en función de las zonas alojadas que se configuran, el número de consultas a las que diese una respuesta y por cada nombre de dominio que se registrase. Como no existe ninguna oferta sobre este servicio, se podría valorar el uso de otros servicios alternativos como lo podría ser Cloudflare que ofrece una capa gratuita que cuadra con el uso estimado que se le podría dar con JupyterHub.

7.2.3 Servicio de creación de instancias

EC2 [21] es un servicio que proporciona capacidad de cómputo escalable en la nube para la creación de instancias bajo demanda. Esto permite la creación de máquinas virtuales con la posibilidad de seleccionar las capacidades de cómputo deseadas como lo puede ser el procesador, el almacenamiento, la red e incluso el sistema operativo. En cuanto a la escalabilidad, se puede hacer de forma vertical, en la que se aumentan los recursos computacionales de las instancias o de forma horizontal, en la que se, a diferencia de la vertical, se agregan más instancias para satisfacer la demanda de peticiones.

Este servicio sí que cuenta con una capa gratuita que proporciona 750 horas de uso al mes de las instancias más pequeñas, siempre que se haga uso de los sistemas operativos Linux. Esta oferta tiene una duración de un año desde que se activa la cuenta.

No obstante, para incentivar el uso de este servicio, es posible reducir el coste de uso de las instancias si el propio usuario se compromete a utilizarlas durante periodos largos de tiempo que pueden ir del año hacia adelante. Con este tipo de compromisos, el coste de las instancias se puede llegar a reducir de forma considerable hasta alcanzar unos descuentos del 72% en los mejores casos.

Adicionalmente, también es posible alcanzar otro tipo de descuento al hacer uso de las instancias conocidas como instancias spot. En estos casos, los descuentos pueden llegar a alcanzar el 90% del coste habitual ya que aprovechan la capacidad de cómputo que sobra. Este tipo de instancias sólo están recomendadas para aplicaciones que puedan tolerar que las instancias se detengan y se vuelvan a poner en marcha. Esto es así porque en el momento que no exista capacidad sobrante de ciertas instancias, es bastante probable que la instancia spot desplegada se detenga hasta que EC2 vuelva a tener capacidad sobrante.

8. Casos de estudio

En apartados anteriores se ha determinado qué herramienta es la que se adapta a los entornos docentes al mismo tiempo que cumplían con los requisitos para ser integradas en asignaturas, especialmente en aquellas relacionadas con los proveedores cloud. Una vez se ha seleccionado la herramienta, es necesario determinar qué versión es la más adecuada para realizar una posible prueba piloto en una asignatura.

Para ello es necesario analizar una serie de situaciones que representen los casos más dados a ocurrir para estimar el número de usuarios concurrentes que la usarían. Otro aspecto que se tendrá que analizar es la posibilidad de desplegarse de forma local en alguno de los servidores del departamento correspondiente o en los servidores de AWS.

Anteriormente, se comentó que JupyterHub se puede implementar de forma reducida para un grupo concurrente de hasta cien usuarios en un único servidor mientras que, la versión completa, está pensada para más de cien usuarios concurrentes, siendo posible el uso de varios servidores para dar soporte. Teniendo esto en cuenta, se implementará la herramienta en su versión reducida tanto en una máquina virtual en la nube como en una máquina física. Para contemplar una situación con más de cien usuarios, se hará también una implementación de la versión completa en una máquina física. No se implementará la versión completa en la nube para evitar cargos por exceder los límites de la capa gratuita de AWS.

Por último, se realizará una demostración de un boletín de prácticas funcionando sobre JupyterHub para mostrar las ventajas de su uso.

8.1 JupyterHub sobre una máquina en AWS

En este apartado se pretende realizar un despliegue de JupyterHub sobre una máquina virtual en la nube de Amazon en su modalidad reducida. Para evitar incurrir en gastos adicionales, se configurará la máquina virtual para que cumpla los requisitos necesarios para operar dentro de la capa gratuita. Esto significa que la instancia contará con una única unidad de procesamiento, 1GB de memoria y hasta con 30GB de almacenamiento en volúmenes EBS, que es similar al almacenamiento en un disco duro tradicional.

Un paso previo al despliegue es la comprobación de los requisitos mínimos que exige la versión reducida para poder funcionar adecuadamente. En este aspecto es posible estar tranquilos porque la máquina virtual descrita anteriormente los cumple. Esto es debido a que se tuvieron en cuenta los requisitos de la capa gratuita de los proveedores cloud para establecerlos y que los usuarios los pudieran realizar despliegues para probar la herramienta.

No obstante, un requisito bastante importante para desplegar la versión reducida es el uso de un sistema operativo basado en Ubuntu con permisos de administrador en la máquina virtual sobre la que se hace el despliegue. En el sitio web de la herramienta se



recomienda el uso de la última versión estable con soporte para actualizaciones a largo plazo.

En la actualidad, Ubuntu es una de las distribuciones de Linux más utilizadas tanto por su versión de servidor, que no cuenta con interfaz gráfica u otros componentes que no son estrictamente necesarios para su funcionamiento, como por su versión de escritorio que cuenta con una popular interfaz junto con las herramientas que ofrece de forma nativa. La versión destinada a estar funcionando sin fallos en los servidores cuenta con unos requisitos menos demandantes si se comparan con su versión de escritorio. Esto es debido a que no tiene que dedicar recursos innecesarios para dar soporte a la interfaz o a los componentes adicionales que incorpora. Es una versión bastante optimizada que permite obtener el máximo rendimiento de los equipos. Por este motivo se recomienda su uso en aquellos equipos que cuentan con especificaciones reducidas. Teniendo esto en cuenta, se hará uso del sistema operativo Ubuntu para servidores en la versión 22.04 LTS que ofrece soporte para actualizaciones a largo plazo y que, además, entra dentro de la capa gratuita para hacer el despliegue.

Para comenzar con el despliegue, es necesario acceder a una cuenta que tenga acceso a los servicios en la nube de Amazon para poder visualizar el panel de administración de EC2 que será el que nos permita lanzar la instancia. Una vez dentro, se seleccionará la opción de lanzar instancia para comenzar con el asistente de configuración.

En dicho asistente deberemos indicar en el primer paso el nombre que identificará a la instancia cuando esté creada. En este caso se ha seleccionado JupyterHub-R que hace referencia al nombre de la herramienta seguido de la letra que indica que se trata de la versión reducida. En el siguiente paso el asistente solicita la selección de una imagen de sistema operativo, en la que se ha seleccionado la descrita anteriormente.

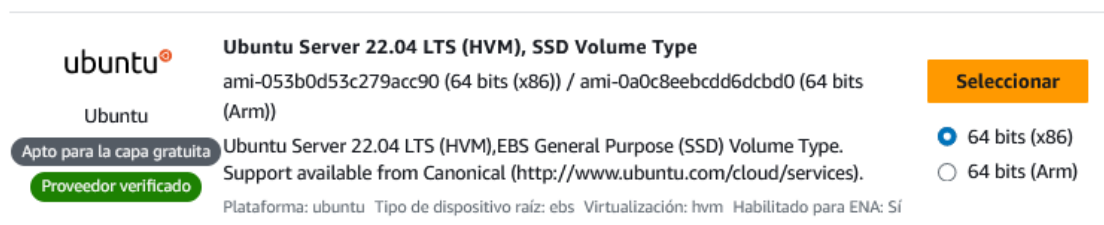


Figura 18. Imagen del sistema operativo Ubuntu para servidores

Una vez seleccionada, es posible visualizar una etiqueta en el asistente que indica que la imagen seleccionada es apta para ser usada en la capa gratuita. Lo mismo ocurre en el siguiente paso en el que solicita la elección de una máquina virtual para el despliegue. En este momento es necesario tener claro el tipo de instancia que se selecciona ya que, si la opción elegida no entra dentro de la capa gratuita, se incurrirá en gastos. Por ello, es necesario seleccionar la instancia que cumpla con los requisitos mencionados al principio de esta sección.

Durante el proceso de configuración, será necesario introducir la información para usar tanto un par de claves de seguridad como un grupo de seguridad. El par de claves sirve para permitir conectarse de forma segura mediante SSH a las instancias sin ser necesario la introducción de un usuario o contraseña. Si el usuario no cuenta con un par de claves, es posible hacerlo en el proceso de configuración de la instancia.

Par de claves

Un par de claves, compuesto por una clave privada y una clave pública, es un conjunto de credenciales de seguridad que se utilizan para demostrar su identidad cuando se conecta a una instancia.

Nombre

alucloud211-keypair

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves [Información](#)

RSA ED25519

Formato de archivo de clave privada

.pem
Para usar con OpenSSH

.ppk
Para usar con PuTTY

Etiquetas: *opcional*

No hay etiquetas asociadas a este recurso.

[Agregar nueva etiqueta](#)

Puede agregar hasta 50 etiquetas más.

Figura 19. Creación de un par de claves de seguridad

En su creación, es importante asignarle un nombre que permita identificarlas de forma unívoca prestando especial atención al formato de clave privada que generará. Se puede elegir entre dos formatos que corresponden a una de las dos herramientas más usadas para crear conexiones SSH.

En relación con el grupo de seguridad, es importante configurarlo correctamente ya que es el encargado de permitir o denegar conexiones entrantes o salientes de las instancias. Permite establecer reglas de conexión como si se tratase de un firewall reducido. Para garantizar el correcto funcionamiento de la herramienta, es necesario permitir el tráfico entrante que haga uso del protocolo tanto HTTP como su versión cifrada HTTPS para que los usuarios puedan acceder desde cualquier navegador web haciendo uso de los puertos estándar. También es recomendable habilitar el tráfico entrante que hace uso del protocolo SSH para permitir que los administradores o los usuarios avanzados puedan realizar tareas de mantenimiento o configuración en la máquina virtual.

Similar al par de claves de seguridad, el grupo de seguridad se puede crear previamente a la configuración de la instancia o en el mismo momento desde el propio asistente tal y como se muestra en la siguiente captura del proceso.

Firewall (grupos de seguridad) Información

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

Crear grupo de seguridad
 Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad - obligatorio

Este grupo de seguridad se agregará a todas las interfaces de red. El nombre no se puede editar después de crear el grupo de seguridad. La longitud máxima es de 255 caracteres. Caracteres válidos: a-z, A-Z, 0-9, espacios y _-/() #,@[]+= &; {}! \$*

Descripción - obligatorio Información

Reglas de grupos de seguridad de entrada

- ▶ Regla del grupo de seguridad 1 (TCP, 22, 0.0.0.0/0) Eliminar
- ▶ Regla del grupo de seguridad 2 (TCP, 80, 0.0.0.0/0) Eliminar
- ▶ Regla del grupo de seguridad 3 (TCP, 443, 0.0.0.0/0) Eliminar

Figura 20. Creación de un grupo de seguridad para la instancia

Previamente se había mencionado que la capa gratuita de este servicio permite el uso de hasta 30 GB de almacenamiento en volúmenes EBS, que equivalen a usar una unidad de disco de estado sólido con la misma capacidad. No obstante, como se va a hacer un despliegue sencillo para un entorno de pruebas, se configurará la instancia para hacer uso de un almacenamiento máximo de 8 GB a pesar de no alcanzar el límite del nivel gratuito.

▼ Configurar almacenamiento Información Avanzado

1x GiB Volumen raíz (Sin cifrar)

Los clientes que cumplan los requisitos de la capa gratuita pueden obtener hasta 30 GB de almacenamiento magnético o de uso general (SSD) de EBS ✕

Figura 21. Creación de la unidad de almacenamiento

El último paso de configuración antes de proceder a lanzar la instancia es la posibilidad de indicar a la instancia que ejecute unas instrucciones de código que permiten lanzar el proceso de instalación de JupyterHub sobre la máquina. Esto se indica en el apartado de detalles avanzados, en el recuadro destinado a introducir datos de usuario. En dicho recuadro, habrá que introducir el siguiente fragmento de código en el que se debe detallar el nombre del usuario administrador de JupyterHub. Si no se introduce este fragmento de código, será necesario conectarse mediante un cliente SSH a la instancia para lanzarlo de forma manual.

```
#!/bin/bash
curl -L https://tljh.jupyter.org/bootstrap.py \
| sudo python3 - \
--admin jupyteradmin
```

Figura 22. Fragmento de código que inicia la instalación de JupyterHub

Una vez rellenado el recuadro, se puede lanzar la instancia para iniciar tanto el proceso de instalación de Ubuntu como de la herramienta en su versión reducida. Si todo ha ido correctamente durante el proceso de configuración de la instancia, se mostrará un mensaje de confirmación en la pantalla con todas las acciones llevadas a cabo por el asistente. En dicho mensaje aparecerá un identificador que se puede usar para localizar la instancia mediante un número interno en el servicio EC2.

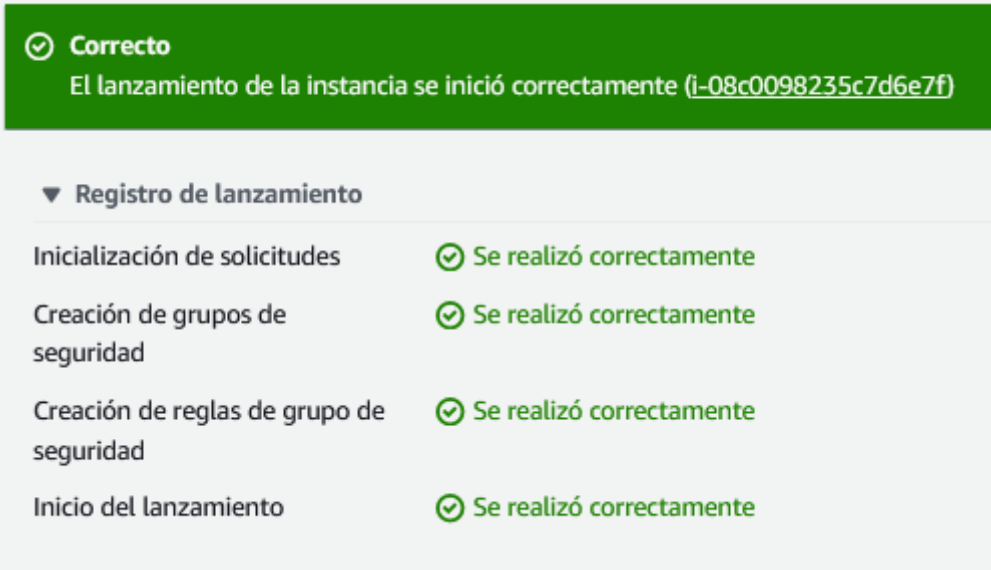


Figura 23. Configuración de la instancia concluido con éxito

Ahora es posible acceder al apartado del panel de EC2 que muestra las instancias que se encuentran actualmente en ejecución para localizar la que se está desplegando, usando el nombre asignado en el primer paso del asistente o mediante el identificador que se ha obtenido en el último paso.

Durante los siguientes minutos, tanto el proceso de instalación como de configuración puede llegar a durar hasta 10 minutos aproximadamente por lo que, una vez pasado una cantidad de tiempo prudencial, el servicio JupyterHub estará disponible. Para acceder basta con usar la dirección IP pública que tiene la instancia asignada para introducirla en la barra de un navegador web. Esta acción debería mostrar el portal de identificación del usuario. En este punto, basta con introducir el nombre de usuario, que tiene el rol de administrador, que se ha especificado en los datos del usuario.

8.2 JupyterHub sobre una máquina física

En el apartado anterior se realizó un despliegue de la herramienta sobre una máquina virtual que se encontraba ubicada en la nube de Amazon para determinar las posibles ventajas que tendría un despliegue en un proveedor cloud. Por ello, en este apartado se pretende repetir el mismo despliegue a diferencia de que la máquina estará ubicada en una máquina física con el objetivo de encontrar cuáles son las ventajas e inconvenientes de hacer un despliegue en local.

No obstante, al no contar con una máquina física para realizar el despliegue, se hará uso de una máquina virtual en el ordenador del estudiante que contará con los mismos requisitos computacionales que se emplearon en el apartado anterior. Eso significa que la máquina virtual contará con una única unidad de procesamiento, 1GB de memoria y con 8GB de capacidad de almacenamiento. Cabe destacar que el almacenamiento de la propia máquina virtual se estará respaldado por una unidad de estado sólido, el mismo tipo de memoria que se empleó en el apartado anterior en la máquina desplegada en la nube.

En relación con el sistema operativo, se hará uso de la misma versión de Ubuntu para desplegar una máquina idéntica que cumpla con los requisitos mínimos y para disponer de una versión que cuenta con soporte para actualizaciones a largo plazo. Se trata de la del sistema operativo Ubuntu en su versión 22.04 LTS para servidores que, al no contar con soporte para la interfaz gráfica de usuario, permite aprovechar todo el rendimiento del hardware disponible ya que cuenta únicamente con los componentes esenciales del sistema. Esta imagen del sistema operativo se puede obtener directamente desde el sitio web oficial de Ubuntu.

Para crear máquinas virtuales en un equipo, es necesario el uso de algún programa que permita la creación y gestión de máquinas virtuales. En la actualidad, existen multitud de aplicaciones que permiten la virtualización, tanto gratuitas como de pago. Para este caso, se ha decidido hacer uso de la herramienta conocida como VirtualBox que tiene a Oracle como empresa desarrolladora. Esta herramienta se ha elegido tanto por ser gratuita como por contar con una gran empresa detrás de ella que ofrece soporte a una gran comunidad de usuarios.

Para comenzar con el despliegue, es necesario contar con la aplicación de virtualización instalada en el equipo para iniciar el asistente que permite guiar al usuario en los pasos de configuración de la máquina. En el primero de ellos, el asistente solicita un nombre que permita identificar a la máquina, una carpeta en la que almacenar, entre otros, los discos virtuales junto con la ruta en la que se encuentra la imagen del sistema operativo

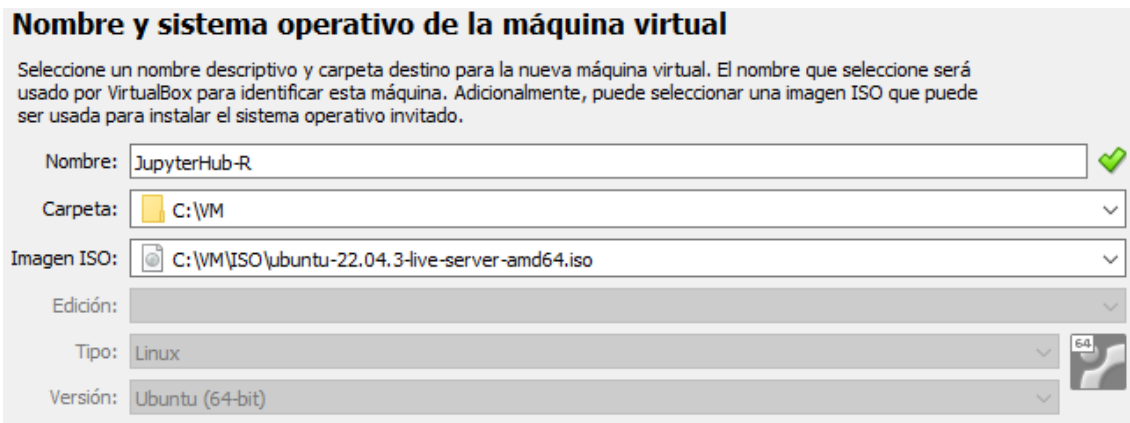


Figura 24. Configuración de las carpetas de la máquina virtual

El siguiente paso solicita la introducción de la cantidad de recursos computacionales que se le asignarán a la máquina. Concretamente, se debe especificar la cantidad de memoria junto con el número de unidades de cómputo. Es importante introducir los mismos datos que los empleados en el caso de estudio previo para poder llevar a cabo una comparación justa.

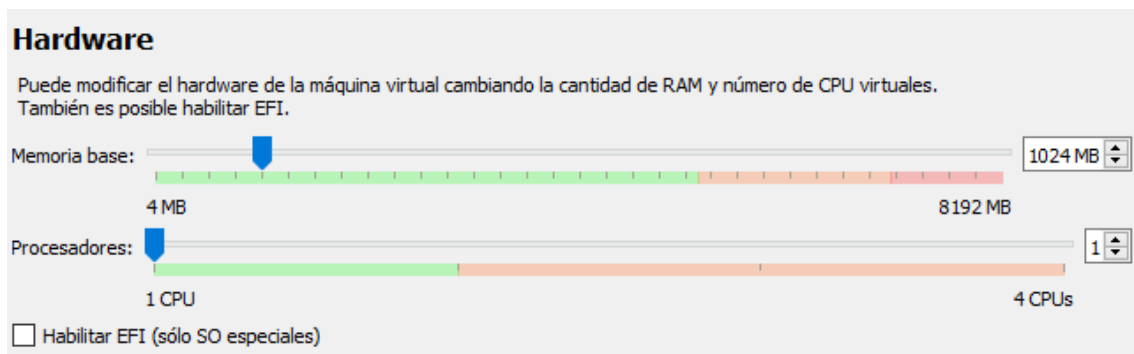


Figura 25. Configuración de los recursos de la máquina virtual

Tras rellenar dichos datos, el asistente solicitará que el usuario indique la cantidad de almacenamiento que tendrá disponible la máquina virtual, que al igual que en el caso anterior, se indicará que dispone de 8 GB de almacenamiento. Teniendo en cuenta que no se utiliza una gran cantidad de almacenamiento, se puede aumentar el rendimiento de la máquina en las tareas de lectura o escritura en disco si en el momento de creación de la máquina se marca la casilla que permite reservar el tamaño completo del disco. De esta forma, crea un disco virtual que ya ocupa el tamaño especificado. Si no se marca la casilla, se crearía un disco duro virtual que iría aumentando hasta llegar al tamaño máximo especificado conforme fuese necesario. Los progresivos aumentos que tendrían que hacerse provocarían retardos que se pueden evitar. No obstante, en sistemas donde no se cuenta con una gran cantidad de almacenamiento disponible, sería una buena opción no activarlo ya que, al no ocupar todo el espacio, puede ser utilizado para otras tareas.

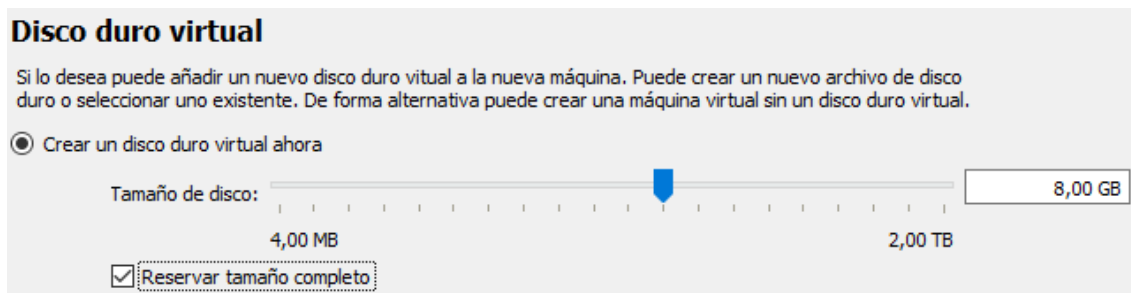


Figura 26. Configuración del almacenamiento virtual de la máquina

Una vez finalizado el asistente de configuración, la máquina virtual se pone en marcha para iniciar primero el proceso de instalación del sistema operativo. Es importante seleccionar durante el proceso de instalación del sistema operativo la opción que realiza una instalación minimizada con el fin de instalar el número mínimo de paquetes de software ya que, a posteriori, es posible instalarlos sin ningún tipo de problema.

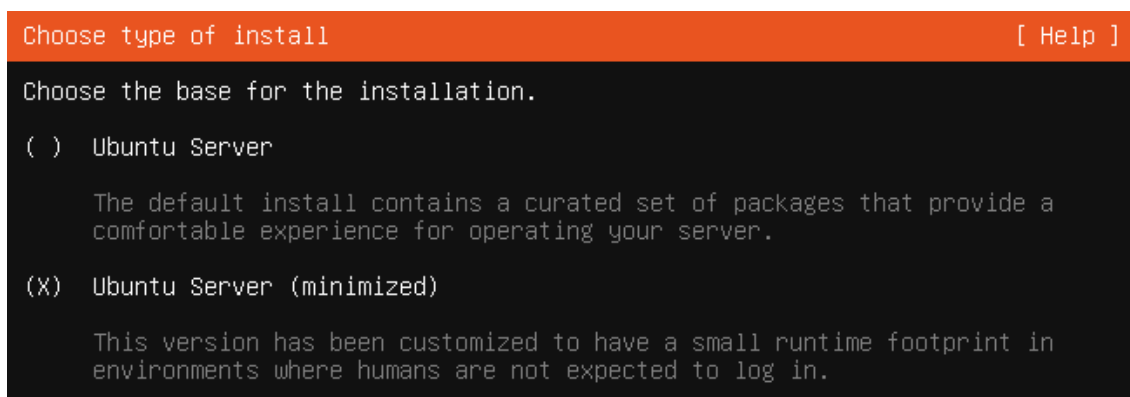


Figura 27. Selección de una instalación minimizada de Ubuntu

A pesar de poder visualizar el contenido de la máquina virtual en el equipo en el que se está ejecutando, es recomendable habilitar la conexión SSH de forma segura para poder realizar configuraciones remotas desde la misma red que está conectada la máquina. Se puede activar en el paso que pregunta sobre la configuración SSH. La identificación del usuario se puede hacer mediante el uso de unas credenciales válidas o mediante el uso de claves que pueden estar almacenadas en un repositorio de GitHub. Para este caso de estudio, es suficiente con las credenciales del administrador de la máquina por lo que no es necesario realizar ninguna importación de credenciales.

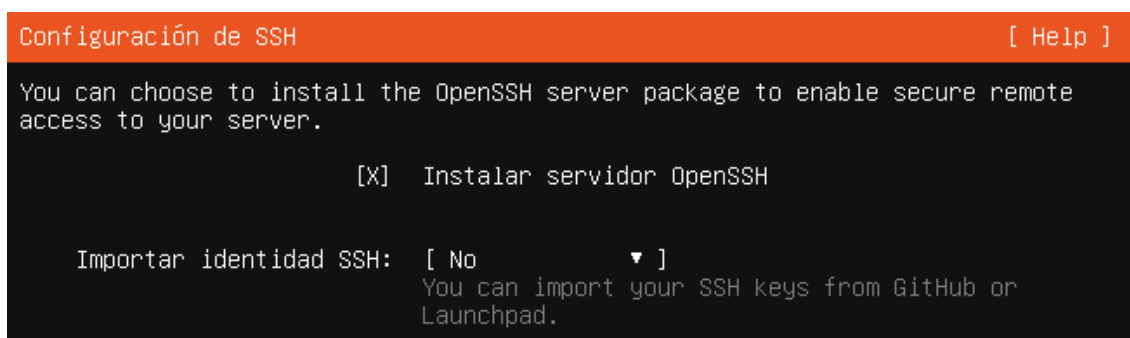


Figura 28. Selección de la instalación del servicio SSH

En uno de los apartados finales de la configuración del sistema, el asistente preguntará al usuario si desea realizar la instalación de algún paquete o componente de software que la amplia comunidad de usuarios utiliza. Como el objetivo es de realizar una instalación limpia con los componentes mínimos para el correcto funcionamiento, no será necesario seleccionar ningún componente.

Tras finalizar el proceso de configuración del sistema operativo, se puede verificar que es posible conectarse mediante SSH a la máquina. Esto se logra obteniendo en primer lugar la dirección IP que se le ha asignado a la máquina para usarla en la conexión SSH. Si todo se ha configurado correctamente, al usar un cliente de conexiones SSH, éste debería solicitar unas credenciales. Si se introduce tanto el usuario como la contraseña configurada en el asistente, debería ser posible conectarse a la máquina. Tras esta breve comprobación, es posible comenzar a instalar JupyterHub en su versión reducida mediante la ejecución del mismo comando empleado en el despliegue en AWS.

```
#!/bin/bash
curl -L https://tljh.jupyter.org/bootstrap.py \
| sudo python3 - \
--admin jupyteradmin
```

Figura 29. Comando de instalación de JupyterHub en local

Este comando hace uso de un fichero de configuración que se encuentra disponible en el repositorio oficial de la herramienta. No obstante, se podría descargar en la máquina virtual para modificarlo con nuevos parámetros de configuración si fuese necesario. De momento, se hará uso de la configuración por defecto y se indicará el nombre que usará el administrador de la herramienta. Cuando el comando de instalación entra en acción, es necesario tener en cuante que el proceso completo puede llegar a alcanzar una duración de 10 minutos, dependiendo de los recursos computacionales de la máquina o de la velocidad de conexión a internet. Tras esperar un tiempo prudencial, debería ser posible acceder a la herramienta desde cualquier navegador web introduciendo la dirección IP de la máquina siempre que el usuario se encuentre en la misma red.

8.3 JupyterHub sobre Kubernetes en una máquina física

En este apartado se pretende llevar a cabo un despliegue de JupyterHub en su versión completa mediante el uso de Kubernetes. Esta versión está pensada para entornos con un gran número de usuarios, pero también es apta para grupos más reducidos. Se diferencia principalmente de su versión reducida en que ha sido diseñada para usar la tecnología de Kubernetes. Esto ha producido que se pueda desplegar en un número de servidores que puede variar en función de la carga que generen los usuarios.

En el sitio web de la herramienta se recomienda comenzar el despliegue disponiendo de un clúster de ordenadores o servidores que tengan en ejecución Kubernetes con el fin de

simplificar el proceso de despliegue. Como esto no es siempre posible, en este caso de estudio se procederá a la creación de un clúster de Kubernetes desde cero que estará formado por un único ordenador. De esta forma, se crea la posibilidad de añadir en un futuro ordenadores conforme fuese necesario. No sólo se añadirían para ser utilizados por la herramienta sino para realizar despliegues de otras aplicaciones.

Como se ha mencionado en apartados anteriores, Kubernetes es una herramienta que permite la orquestación de contenedores de código abierto para que cualquier persona pueda editarlo o adaptarlo a sus necesidades. Eso ha permitido la aparición de diversas implementaciones de la herramienta entre las que destaca Microk8s. Esta distribución se caracteriza por estar desarrollada por la misma empresa que mantiene el sistema operativo Ubuntu. Se ha diseñado para poder disfrutar de una implementación a nivel empresarial de Kubernetes bastante ligera en términos de espacio en disco o memoria. Además, esta implementación permite instalar complementos tanto de repositorios de terceros como oficiales para ampliar sus funciones debido a la amplia comunidad de usuarios que desarrolla constantemente nuevos complementos.

Al no disponer de una máquina física para poder realizar el despliegue, también se hará uso de una máquina virtual en el equipo del estudiante que contará con los mismos recursos computacionales que se han empleado en los apartados anteriores. Por ello, la máquina virtual contará con una única unidad de procesamiento, 1GB de memoria y con 8GB de almacenamiento.

En relación con el sistema operativo elegido, también se hará uso de la misma versión de Ubuntu para servidores con el fin de desplegar una máquina idéntica que cumpla con los requisitos mínimos de la versión completa. Se trata de la del sistema operativo Ubuntu en su versión 22.04 LTS para servidores disponible desde el sitio web oficial de Ubuntu. Incluso se utilizará la misma aplicación para crear la máquina virtual dentro del equipo del estudiante.

Para no repetir el proceso de instalación del sistema operativo, se omite dicho proceso en este caso de uso ya que sería idéntico al efectuado en el apartado anterior. La única diferencia es que, en este despliegue, no es necesario ejecutar los comandos que inician la instalación de la versión reducida.

Una vez completado el proceso de instalación del sistema operativo, se dispondrá de una instalación de Ubuntu para servidores desde cero, lista para instalar la distribución de Microk8s. El proceso de instalación de MicroK8s es bastante sencillo ya que, basta con ejecutar el comando que inicia la instalación básica de la distribución. No obstante, antes de empezar con la instalación, es recomendable actualizar todos los paquetes de Ubuntu para garantizar que se implementa con todos los componentes de software actualizados, evitando de esta manera posibles problemas de seguridad.

```
sudo snap install microk8s --classic
```

Figura 30. Comando para iniciar la instalación de Microk8s

La duración de la instalación puede llegar a alcanzar una duración de 2 minutos en función de los recursos computacionales de la máquina o de la conexión a internet. Una vez concluido el proceso, se mostrará en la consola un mensaje de confirmación con la

versión que se ha instalado. Otra forma de comprobar que se ha instalado bien es consultando mediante línea de comandos, cuántos nodos forman parte del clúster con el siguiente comando.

```
microk8s.kubectl get nodes
```

Figura 31. Comando para obtener los nodos del clúster

El resultado debería ser que está formado por un único nodo que corresponderá con el nombre que tiene asignada la máquina. Entre otros datos, mostrará el tiempo que lleva en marcha junto con la versión de Kubernetes instalada, que deberá coincidir con la del comando anterior.

NAME	STATUS	ROLES	AGE	VERSION
jupyterhub-k	Ready	<none>	96m	v1.27.4

Figura 32. Información sobre los nodos del clúster

Conforme se añadiesen nodos al clúster para dar soporte a la demanda de los usuarios, se podría volver a lanzar el comando para identificar cada uno de los nodos que forman parte del clúster. Es importante tener en cuenta que para añadir o quitar nodos de un clúster es necesario realizarlo de forma manual con comandos o mediante scripts.

Con el clúster en funcionamiento, es necesario añadir una serie de complementos que serán necesarios para el correcto funcionamiento como es el servicio de resolución de nombres, el panel global que muestra información del clúster o el gestor de paquetes de Kubernetes que se puede instalar como complemento o como paquete de software. Para iniciar la instalación de los complementos es necesario usar los siguientes comandos:

```
microk8s enable dns // Enables DNS service
microk8s enable helm3 // Enables package manager for Kubernetes
microk8s enable dashboard // Enables main dashboard
```

Figura 33. Comandos para instalar los servicios de Microk8s

El gestor de paquetes de Kubernetes es una herramienta que proporciona una línea de comandos que permitirá realizar tareas como la instalación, actualización o gestión de las aplicaciones que se encuentran dentro de un clúster de Kubernetes. Esto permitirá tanto la instalación como la gestión de JupyterHub dentro del clúster. Esto es debido a que los paquetes que emplea el gestor son interpretados como abstracciones o plantillas que describen cómo deben instalarse los paquetes de software.

```
curl https://raw.githubusercontent.com/helm/helm/HEAD/scripts/get-helm-3 | bash
```

Figura 34. Comando de instalación del gestor de paquetes

La instalación del gestor de paquetes se puede realizar mediante la ejecución del código que se muestra en la figura inmediatamente superior. La duración de la instalación está por debajo del minuto por lo que, no se instala en ese periodo de tiempo, puede ser debido a un error en la dirección del enlace o en la conexión a internet.

Como se ha mencionado antes, los paquetes del gestor contienen plantillas que pueden determinar los recursos de Kubernetes que se instalarán junto con la configuración de despliegue. Ahora se puede proceder a la creación de un fichero de configuración para determinar la configuración inicial de JupyterHub. El fichero permite entre otras cosas expandir las funcionalidades, aumentar los recursos computacionales disponibles para los usuarios o cambiar los servicios de autenticación de los usuarios. Todas las configuraciones se pueden consultar en la guía oficial de la herramienta.

En las versiones más recientes de la herramienta, no es necesario especificar ninguna configuración en el fichero, pero sí es recomendable crear un fichero de configuración vacío para realizar futuras configuraciones. El fichero se puede almacenar en cualquier carpeta del sistema, pero se recomienda usar la carpeta personal del usuario que está realizando la instalación. El único requisito es que el fichero haga uso de la extensión yaml para que pueda ser interpretado por el gestor. Para tener los ficheros de instalación controlados, se creará una carpeta en del directorio principal del usuario que realiza la instalación.

Con el fin de gestionar la comunicación del usuario con el servicio de Kubernetes, es necesaria la creación de un balanceador de carga. Esto permitirá vincular una dirección disponible de la red, en la que está conectado el equipo, para que Kubernetes pueda dar una respuesta a todas las peticiones de los usuarios. Para ello, es recomendable usar un rango de red que se encuentre disponible como el que se muestra en la figura.

```
// Enables LoadBalancer for your kubernetes cluster
microk8s enable metallb:10.0.2.100-10.0.2.150
```

Figura 35. Activación del servicio de balanceo de carga

En este momento, sólo quedaría configurar el sistema de almacenamiento que usará el clúster para la persistencia de los archivos. El primer paso consiste en la instalación de los componentes necesarios tanto en la máquina virtual como para Microk8s. Para ello, se pueden ejecutar los siguientes comandos que aparecen en la guía de instalación.

```
sudo apt install open-iscsi
sudo systemctl enable iscsid.service

//Enables the community addons repository
microk8s enable community

//Enables the open-source storage solution for kubernetes
microk8s enable openebs
```

Figura 36. Instalación de los servicios de almacenamiento del clúster

Los dos comandos iniciales se encargan tanto de instalar el sistema de almacenamiento como de habilitar el componente de software que se acaba de instalar en el sistema. El siguiente paso sería informar a Microk8s para que tenga conocimiento de que dispone del sistema de archivos. Por ello, los siguientes comandos se encargan de habilitar el repositorio de la comunidad para obtener acceso a un número mayor de complementos, permitiendo habilitar después, el complemento de código abierto que permite gestionar el almacenamiento dentro de Kubernetes.

Una vez habilitado el complemento de almacenamiento, es necesario seguir el proceso de configuración que está detallado en la guía de instalación con el fin de especificar cómo debe ser usado para la persistencia de datos. El proceso consiste en la creación de un fichero de configuración con todos los parámetros necesarios para que Kubernetes use el almacenamiento local de la máquina.

```
## local-storage-dir.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local-storage-dir
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
    openebs.io/cas-type: local
    cas.openebs.io/config: |
      - name: StorageType
        value: hostpath
      - name: BasePath
        value: /kubernetes-storage
provisioner: openebs.io/local
reclaimPolicy: Delete
volumeBindingMode: WaitForFirstConsumer
```

Figura 37. Fichero de configuración del almacenamiento

Después, sólo quedaría indicarle a Kubernetes que utilice la política de almacenamiento que se acaba de definir en el fichero mediante el siguiente comando que le indica tanto la política que debe aplicar como la ruta del fichero que la contiene. Esto se puede hacer mediante el comando proporcionado en la guía.

```
microk8s.kubectl apply -f local-storage-dir.yaml
```

Figura 38. Comando para aplicar una política de almacenamiento

En este punto de la instalación, ya se dispone de todos los componentes necesarios para proceder con la instalación de JupyterHub. Para ello, es necesario notificar a Helm que dispone de un paquete que se utiliza a modo de repositorio que se empleará para hacer la instalación de JupyterHub. Después, es necesario indicarle que actualice todos los repositorios que tiene registrados para tener todos los paquetes de software al día.

```
helm repo add jupyterhub https://hub.jupyter.org/helm-chart/
helm repo update
```

Figura 39. Comandos para reparar el gestor de paquetes

Una vez se han completado las acciones de los comandos que se muestran en la figura superior, será necesario ubicar el directorio de la consola en la carpeta que contiene el fichero de configuración vacío que se ha creado en pasos anteriores. Así, una vez dentro del directorio, se puede ejecutar el siguiente comando para lanzar la instalación con el gestor de paquetes de Kubernetes.

```
helm upgrade --cleanup-on-fail \
  --install jupyterhub jupyterhub/jupyterhub \
  --namespace jhub \
  --create-namespace \
  --version=3.0.3 \
  --values config.yaml
```

Figura 40. Instrucción para instalar JupyterHub

En el fragmento de código que se muestra en la figura, se puede apreciar que se detalla un espacio de nombres completamente nuevo que permitirá después identificar a los pods que la herramienta desplegará dentro del clúster. Incluso se puede apreciar que el fichero de configuración se pasa como un parámetro. Es importante destacar que dicho fragmento de código es proporcionado por el sitio web de JupyterHub para realizar la instalación en un clúster. Es el comando por defecto que puede ser modificado para una customización más avanzada de la instalación.

Tras unos instantes, la herramienta estará completamente desplegada. En su proceso habrá creado dos pods en el clúster, uno de ellos corresponde con el hub mientras que el otro corresponde con el servicio de proxy que gestiona todas las conexiones entrantes de los usuarios. Una comprobación que se puede hacer es ejecutar el siguiente comando que se encarga de recuperar los pods que se han creado en un espacio de nombres dado que, en este caso, se detallará el que se ha definido para la herramienta.

```
kubectl get pod --namespace jhub
```

NAME	READY	STATUS	RESTARTS	AGE
hub-7d4dfth2ck-s77w8	1/1	Running	0	117m
proxy-9zk5bc4bp-9pdg	1/1	Running	0	117m

Figura 41. Comando para localizar pods en un clúster

El último paso consiste en determinar la dirección IP que el proxy expone para que los usuarios que se encuentren en la misma red puedan conectarse. Esto se realiza usando el siguiente comando que devuelve las direcciones tanto internas como externas.

```
kubectl --namespace jhub get service proxy-public
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
proxy-public	LoadBalancer	10.0.2.110	192.168.0.100	80:31916/TCP	118m

Figura 42. Obtención de las direcciones expuestas por el proxy

Cuando se realice un cambio en el fichero de configuración que se desea aplicar sobre el despliegue actual de JupyterHub, basta con editar el fichero de configuración inicial para ejecutar después el siguiente comando de Helm para actualizar los paquetes.

```
helm upgrade --cleanup-on-fail \  
  <helm-release-name> jupyterhub/jupyterhub \  
  --namespace jhub \  
  --version=<chart-version> \  
  --values config.yaml
```

Figura 43. Comando que actualiza la configuración

Se diferencia principalmente del comando de instalación en que no requiere la creación del espacio de nombre porque ya se ha creado ni la opción de instalación porque ya se encuentra instalado y en funcionamiento. Es necesario conocer que algunas opciones de configuración pueden incurrir en que los usuarios conectados sean expulsados de sus espacios de cómputo mientras se reinicia la aplicación.

8.4 Boletín de prácticas en un cuaderno

Una vez se dispone de un despliegue de JupyterHub operativo, se puede proceder a la transformación de un boletín de prácticas que ya ha sido distribuido a los estudiantes, en un cuaderno. En este tipo de documentos, la parte correspondiente a los enunciados debe incorporarse a las celdas de texto haciendo uso de Markdown [22] para poder producir posteriormente una memoria con la estructura adecuada. Con este lenguaje de marcado se consigue que cada uno de los elementos se muestre correctamente. Mientras que, la parte correspondiente al código, debe incluirse en celdas cuyo formato sea el de código para que los cuadernos sepan identificar que su contenido está formado por comandos.

La creación de un boletín dentro de la herramienta se haría seleccionando la opción de creación de nuevo cuaderno. Esto provoca la creación de una nueva pestaña en la parte derecha de la pantalla que contendrá un cuaderno vacío preparado para su edición. El siguiente paso consiste en la creación de la primera celda, que irá destinada a contener el primer enunciado del boletín. Esto se puede conseguir al desplazar el ratón al inicio del cuaderno para que aparezca la opción de añadir una nueva celda. Una vez creada, se tiene que indicar el tipo de datos que contendrá la celda que, en este caso, será de tipo Markdown.



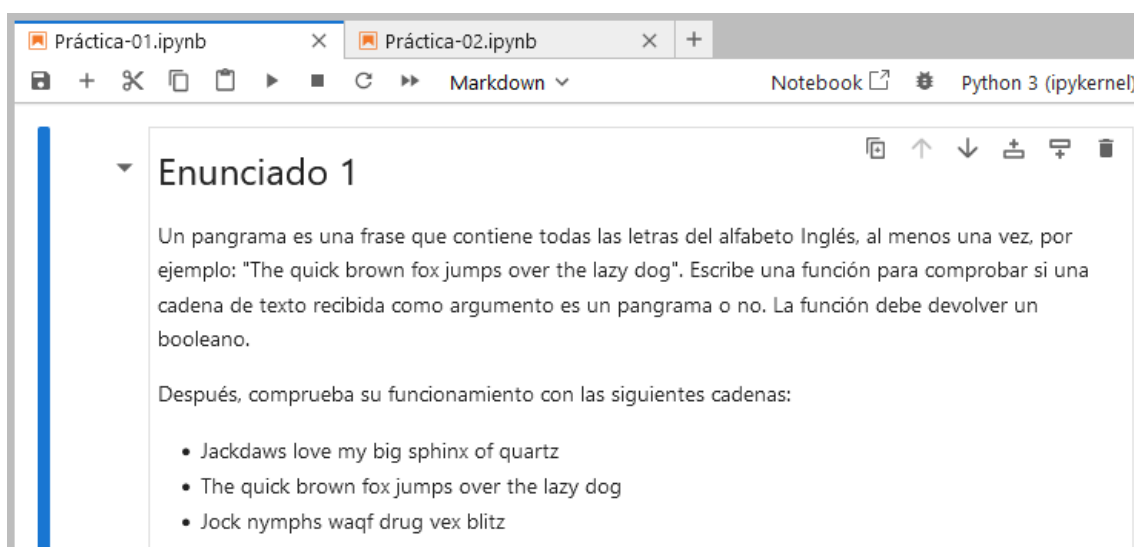


Figura 44. Ejemplo de una celda de tipo Markdown

Como se puede apreciar en la figura, se ha creado una celda para contener el enunciado que está formada por un título que menciona el nombre del apartado, un párrafo que contiene la descripción del ejercicio y una lista que contiene las opciones para que el propio alumno compruebe si es correcto el ejercicio. Este proceso se repetiría para cada uno de los enunciados del boletín con los elementos correspondiente que se tengan que añadir como fragmentos de código a modo de ejemplo para guiar al estudiante.

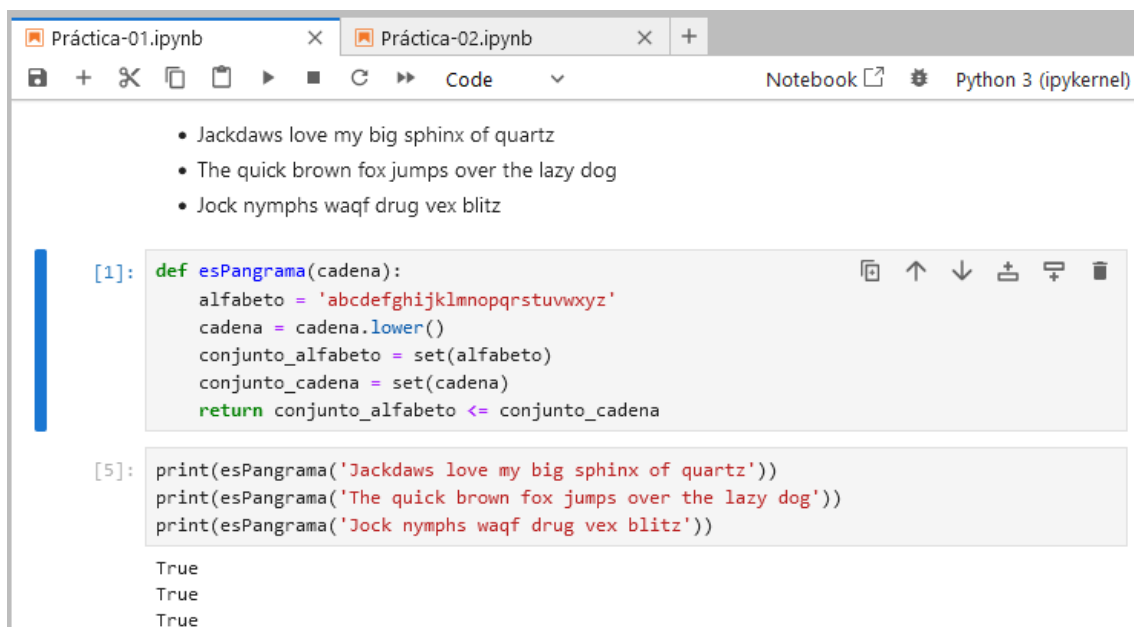


Figura 45. Ejemplo de una celda de tipo código

Desde la perspectiva del estudiante, podría editar el cuaderno añadiendo nuevas celdas de texto para incorporar sus justificaciones de los resultados obtenidos o mediante el uso de celdas de código para realizar todas las operaciones necesarias para resolver el apartado. En la figura superior, se puede apreciar la creación de dos nuevas celdas de código para la creación de una función que resuelva el ejercicio junto con la celda que contiene las instrucciones que permiten comprobar si está bien resultado. Mientras unas

celdas de código pueden devolver resultados como la última de la figura, otras pueden no hacerlo si la ejecución de estas no proporciona ninguna salida por consola.

Este proceso continuaría hasta que el alumno resolviese cada uno de los apartados que forman parte del boletín. Llegados a ese instante, el alumno podría generar su memoria seleccionando la opción de exportar dentro del menú de archivo para seleccionar uno de los formatos disponibles. En este caso, se procede a la exportación de un documento en formato Markdown que tendría un aspecto similar a la siguiente figura.

Enunciado 1

Un pangrama es una frase que contiene todas las letras del alfabeto Inglés, al menos una vez, por ejemplo: “The quick brown fox jumps over the lazy dog”.
Escribe una función para comprobar si una cadena de texto recibida como argumento es un pangrama o no. La función debe devolver un booleano.

Después, comprueba su funcionamiento con las siguientes cadenas:

- Jackdaws love my big sphinx of quartz
- The quick brown fox jumps over the lazy dog
- Jock nymphs waqf drug vex blitz

```
def esPangrama(cadena):  
    alfabeto = 'abcdefghijklmnopqrstuvwxyz'  
    cadena = cadena.lower()  
    conjunto_alfabeto = set(alfabeto)  
    conjunto_cadena = set(cadena)  
    return conjunto_alfabeto <= conjunto_cadena
```

Figura 46. Visualización de una memoria en formato Markdown

Como se puede apreciar, la transformación de los boletines a los cuadernos no es una labor complicada que se podría realizar con un poco de tiempo. Simplemente requiere la transformación del texto original junto con sus elementos multimedia a Markdown. En cuanto a la parte del alumno, es posible añadir tantas celdas de código como sean necesarias para resolver los apartados. Aunque la resolución del boletín mostrado es bastante sencilla, se pueden realizar tareas más elaboradas como las conexiones SSH mencionadas en la memoria o la ejecución de código remoto. La parte buena las celdas de código es que todos los resultados se pueden mostrar fácilmente en las memorias.

9. Conclusiones

Un factor importante que afectará a la experiencia de uso de la herramienta es el tipo de versión que se implementa debido al número de servidores que pueden dar soporte a las cargas de trabajo de los usuarios. En este aspecto, la versión reducida únicamente permite hacer uso de una máquina mientras que la versión completa puede hacer uso de un número de máquinas que varíe en el tiempo para adaptarse a la carga de trabajo sin que los usuarios perciban una pérdida de rendimiento. Esto provoca que la reducida pueda sufrir problemas de rendimiento si el número de usuarios aumenta. Este es uno de los motivos por los que es necesario realizar un análisis del número de alumnos que usarán la herramienta, así como del tipo de trabajos que realizarán.

En relación a las ampliaciones de recursos computacionales, en la versión completa que hace uso de Kubernetes, es posible dimensionarlos sin necesidad de detener el servicio mientras que, en la versión reducida, es necesario detener el servicio, redimensionar la máquina y volver a poner el servicio en marcha.

Después, es necesario reflexionar sobre la posibilidad de desplegar en la nube o en local debido a que cada opción tiene unas ventajas e inconvenientes. Si la opción elegida es la nube, cualquier usuario puede acceder mediante un navegador sin necesidad de hacer algún ajuste en los equipos de los estudiantes, pero los datos de la aplicación residirían en un tercero. Si se almacenan datos críticos como unas credenciales, es necesario preocuparse para evitar una posible filtración accidental. No obstante, si la opción implementada es en local, los datos permanecen en local sin necesidad de estar en un servicio de terceros, evitando así que intenten acceder a los datos. La parte mala de un despliegue de este tipo es que sólo estaría disponible en la red en la que se hace el despliegue. Es un arma de doble filo que puede considerarse bueno o malo en función del punto de vista. Desde una perspectiva de seguridad, es más seguro, pero se obliga a los estudiantes a acceder a los ordenadores de las aulas o, a usar una conexión VPN que les brinde acceso a la red del despliegue.

A estas dos consideraciones es necesario sumarle el análisis de costes que se debería hacer previo a la instalación. El uso de recursos en la nube suele incurrir en un coste mensual que se podría evitar si se dispone de equipamiento informático para hacer el despliegue.

Para concluir, se puede afirmar que se han cumplido los objetivos que se han marcado al comienzo del proyecto para el análisis de herramientas que permitan interactuar con los cuadernos. Finalmente, la herramienta seleccionada cumple con todos los requisitos para ser implementada en asignaturas relacionadas con la ciencia de datos o con los proveedores cloud. Además, la opción elegida es la que se adapta completamente a los entornos docentes tanto por todas las opciones de configuración como por la facilidad que proporciona a los docentes para distribuir el material a los estudiantes.

10. Relación del trabajo realizado con los estudios cursados

Los estudios cursados en el máster en gestión de la información han determinado en cierta medida el desarrollo de este proyecto.

El empleo de los cuadernos no es algo desconocido debido a que han sido empleados en asignaturas de programación que preparan al estudiante para realizar análisis de datos masivos con el fin de obtener un conocimiento nuevo que los datos no pueden mostrar. En dicha asignatura se interactuaba con los cuadernos mediante el uso de Colaboratory para aprender a manipular datos con varios lenguajes de programación. Este contacto previo con los cuadernos ha facilitado gran parte del análisis que se ha llevado en este proyecto.

Una asignatura que ha influido considerablemente en el proyecto ha sido la de Servicios en la Nube que proporciona una visión global de los proveedores de la nube pública con un gran foco en los servicios más empleados en la nube pública de Amazon. Durante las prácticas en esta asignatura, ha sido posible interactuar con los servicios directamente desde el panel de usuario. Esto ha proporcionado cierta tranquilidad a la hora de hacer los despliegues en AWS al haber usado previamente dichos servicios.

Otra asignatura que ha influido durante las pruebas que se han llevado a cabo ha sido la encargada de dar una introducción teórica a los sistemas de virtualización de sistemas. La asignatura ha permitido tomar decisiones sobre qué tipo de virtualización se debería hacer cuando se cuentan con recursos limitados en las máquinas o cómo algunas de las configuraciones pueden aumentar el rendimiento de los equipos virtualizados.

11. Trabajos futuros

Uno de los casos de estudio de este proyecto ha empleado Kubernetes para preparar un clúster de un único equipo para desplegar JupyterHub. Esto ha permitido realizar un primer contacto con la herramienta, pero no nos ha permitido ver todo su potencial. Un trabajo futuro podría ser la incorporación de otros equipos al clúster para ver cómo el rendimiento del despliegue aumenta, permitiendo dar soporte a más usuarios. Esto se podría hacer si el departamento de sistemas de la titulación contase con un despliegue en funcionamiento de Kubernetes con varios servidores agrupados en un clúster, lo que permitiría ver cómo se distribuyen los pods en cada uno de los servidores, e incluso observar cómo, en función del número de usuarios concurrentes junto con la carga que estarían generando a nivel computacional, apaga o enciende servidores para hacer una gestión eficiente de los recursos o a nivel energético.

Una implementación alternativa a la mencionada podría ser un despliegue empleando la nube pública de Amazon para crear un clúster de Kubernetes con máquinas virtuales o incluso mediante su servicio de Kubernetes. Ambas opciones no se han podido llevar a cabo debido a que superan los límites de la capa gratuita y por no disponer de tiempo suficiente para realizar pruebas de funcionamiento.

Como se pretende integrar la herramienta en una asignatura que guarde una relación estrecha con la nube pública de Amazon, podría ser de especial interés configurar la aplicación para que emplease Cognito como sistema de identificación de usuarios. Con la integración de este servicio se podría gestionar el acceso de los usuarios. Para ello, se podría realizar un registro de los estudiantes que cursan la asignatura en el servicio con el fin de gestionar sus accesos a JupyterHub.

12. Referencias

- [1] Proyecto Jupyter, Recuperado (2023 mayo 15) de https://es.wikipedia.org/wiki/Proyecto_Jupyter
- [2] Librería AWS, Recuperado (2023 mayo 25) de <https://github.com/aws/aws-cli/tree/v2>
- [3] Proyecto Python de Paramiko, Recuperado (2023 mayo 25) de <https://pypi.org/project/paramiko/>
- [4] Proyecto Python de AWSCLI, Recuperado (2023 mayo 25) de <https://pypi.org/project/awscli/>
- [5] Instrucciones AWSCLI, Recuperado (2023 mayo 25) de <https://docs.aws.amazon.com/cli/latest/index.html>
- [6] JupyterHub, Recuperado (2023 junio 6) de <https://jupyter.org/hub>
- [7] Customizing User Resources. Recuperado (2023 junio 6) de <https://z2jh.jupyter.org/en/latest/jupyterhub/customizing/user-resources.html#customizing-user-resources>
- [8] Authenticate using AWS Cognito. Recuperado (2023 junio 6) de <https://tljh.jupyter.org/en/latest/howto/auth/awscognito.html>
- [9] Configuration Reference. Recuperado (2023 junio 6) de <https://z2jh.jupyter.org/en/latest/resources/reference.html#helm-chart-configuration-reference>
- [10] Google Colab, Recuperado (2023 junio 8) de <https://colab.research.google.com/>
- [11] Das, Tamal. Google Colab: todo lo que necesitas saber, (2022, agosto 16). Recuperado de <https://geekflare.com/es/google-colab/>
- [12] Proyecto Binder, Recuperado (2023 junio 12) de <https://jupyter.org/binder>
- [13] BinderHub, Recuperado (2023 junio 12) de <https://mybinder.org/>
- [14] BinderHub Federation, Recuperado (2023 junio 12) de <https://mybinder.readthedocs.io/en/latest/about/federation.html>
- [15] Le Gal, Mael. MyBinder and OVH partnership (2019, julio 10). Recuperado de <https://blog.ovhcloud.com/mybinder-and-ovh-partnership/>
- [16] Share Data with your users, Recuperado (2023 junio 2) de <https://tljh.jupyter.org/en/latest/howto/content/share-data.html>
- [17] RedHat, El concepto de los contenedores, (2023, abril 6). Recuperado de <https://www.redhat.com/es/topics/containers>



- [18] RedHat, ¿Qué es Kubernetes?, (2020, marzo 27). Recuperado de <https://www.redhat.com/es/topics/containers>
- [19] Cognito, Recuperado (2023 julio 12) de <https://aws.amazon.com/es/cognito/>
- [20] Route 53, Recuperado (2023 julio 12) de <https://aws.amazon.com/es/route53/>
- [21] EC2, Recuperado (2023 julio 12) de <https://aws.amazon.com/es/ec2/>
- [22] Markdow, Recuperado (2023 septiembre 5) de <https://experienceleague.adobe.com/docs/contributor/contributor-guide/writing-essentials/markdown.html>

13. Anexo

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.			X	
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.			X	
ODS 10. Reducción de las desigualdades.			X	
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.			X	
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Actualmente, cuando se propone el desarrollo de un proyecto informático que requiere de cierta potencia de cómputo, se suelen plantear formas alternativas de desarrollar software de tal forma que consiga ser eficiente, no sólo en términos de tiempo, sino que también lo debe ser a nivel de eficiencia energética. Esto requerir un análisis previo al desarrollo que permita determinar la forma óptima de aprovechar los recursos computacionales disponibles para reducir la huella de carbono.

En el transcurso de este proyecto se han analizado dos formas distintas de desplegar una herramienta para ser usada por múltiples usuarios al mismo tiempo. Una contempla el uso de equipos disponibles que puedan estar desaprovechados por cualquier motivo con el fin de darles una segunda vida antes de finalizar su ciclo de vida. Esto permitiría reducir la cantidad de residuos informáticos que se desechan a diario. En cambio, el otro despliegue contempla el uso de los recursos de cómputo sobrantes de AWS. Así, se conseguiría aprovechar la energía sobrante de sus sistemas para la herramienta, en vez de desperdiciarse. Un beneficio directo de usar cualquiera de las dos versiones es la reducción del consumo eléctrico de la institución, una por aprovechar los equipos con una baja utilización y la otra por aprovechar la capacidad sobrante de la nube.

En relación con la reducción de las desigualdades, en especial en el ámbito docente, la herramienta propuesta permite garantizar un acceso idéntico a todos los usuarios con la finalidad de que ninguno tenga beneficios sobre otros. Al ser accedida a través de un navegador web, ninguno puede considerarse en inferioridad por contar con un equipo personal que no sea potente o que no cuente con las últimas tecnologías.