



GeeksHubs
academy _

Kafka Operativa

Introducción

- Operativa Básica con Kafka
 - Instalación y configuración entorno de trabajo
 - Instalación y configuración básica Kafka
 - Inicio del cluster
 - Cierre del cluster
 - Herramientas de Gestión de Topics
 - Posibilidades y Auto-configuración de topics
 - Compactación de históricos



Instalación y configuración entorno de trabajo



Instalación



Instalación

Crear máquina virtual



Nombre y sistema operativo

Seleccione un nombre descriptivo y una carpeta destino para la nueva máquina virtual y seleccione el tipo de sistema operativo que tiene intención de instalar en ella. El nombre que seleccione será usado por VirtualBox para identificar esta máquina.

Nombre:

Carpeta de máquina:

Tipo: 

Versión:

Instalación



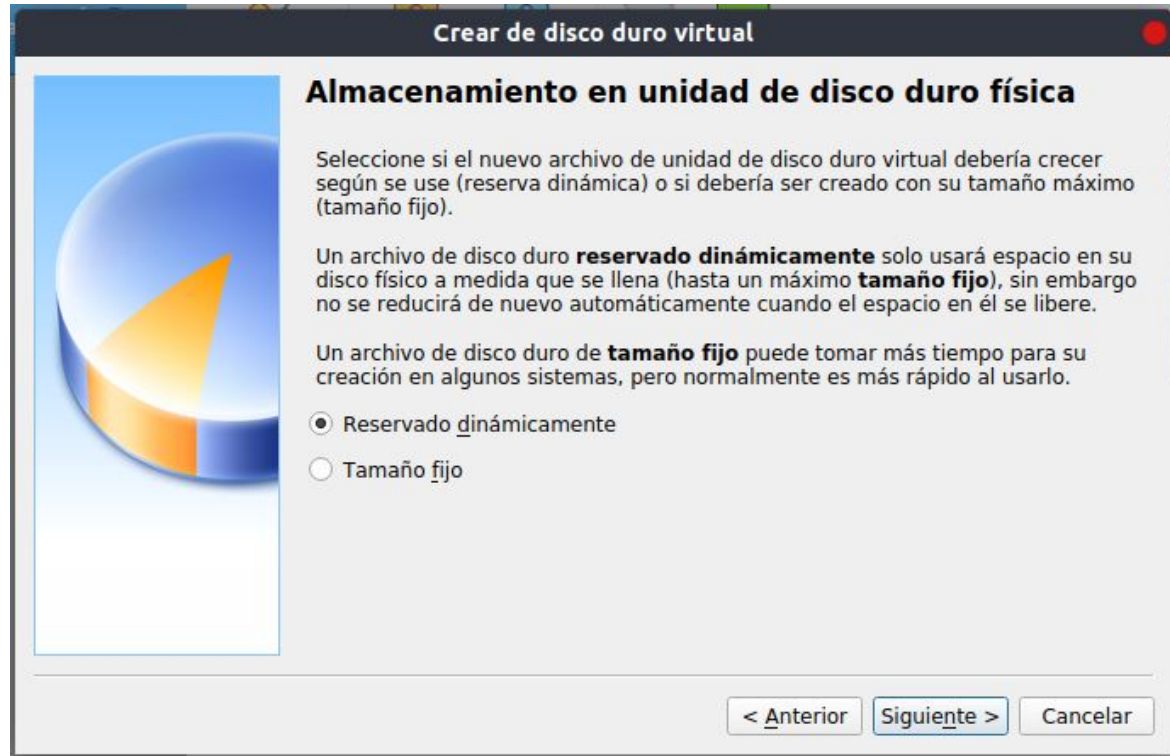
Instalación



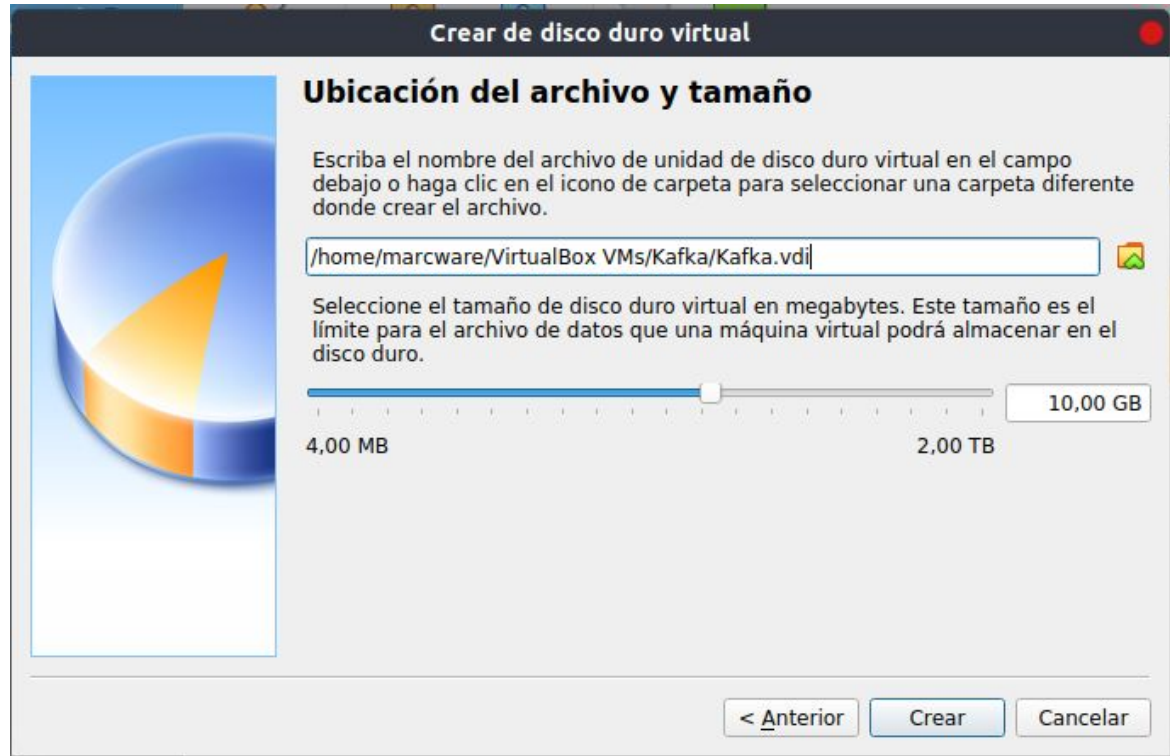
Instalación



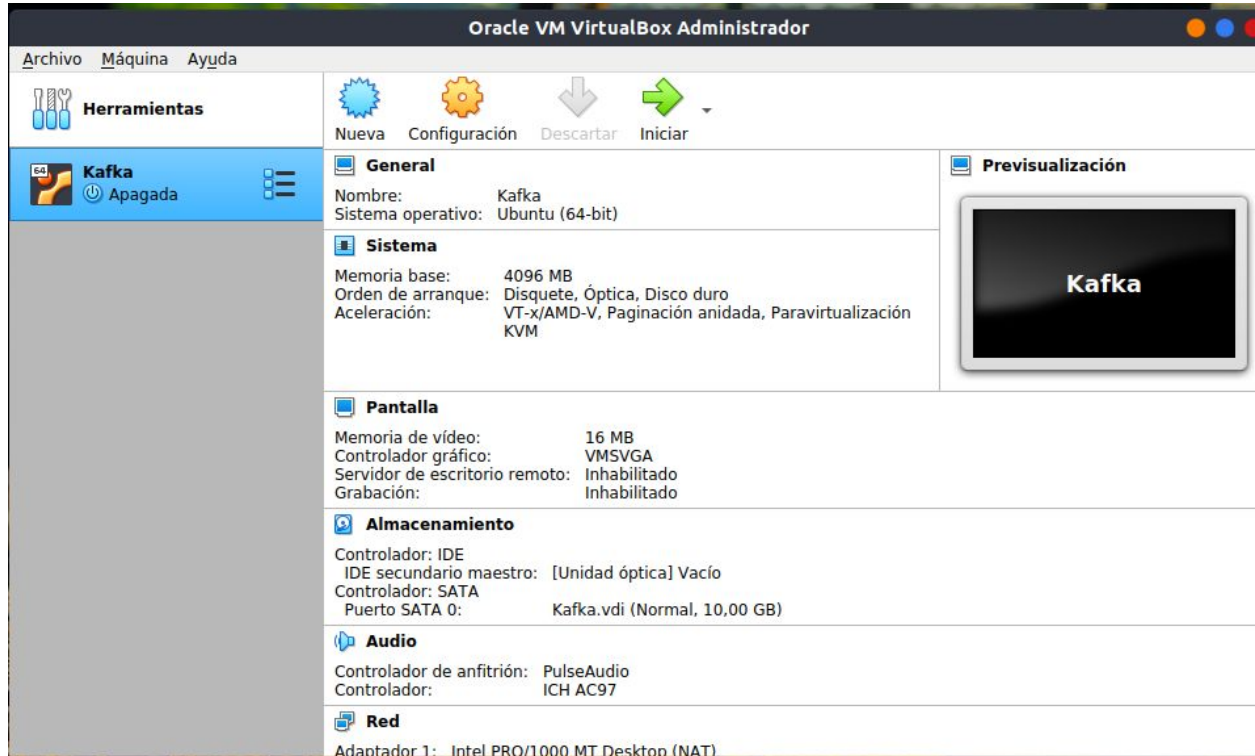
Instalación



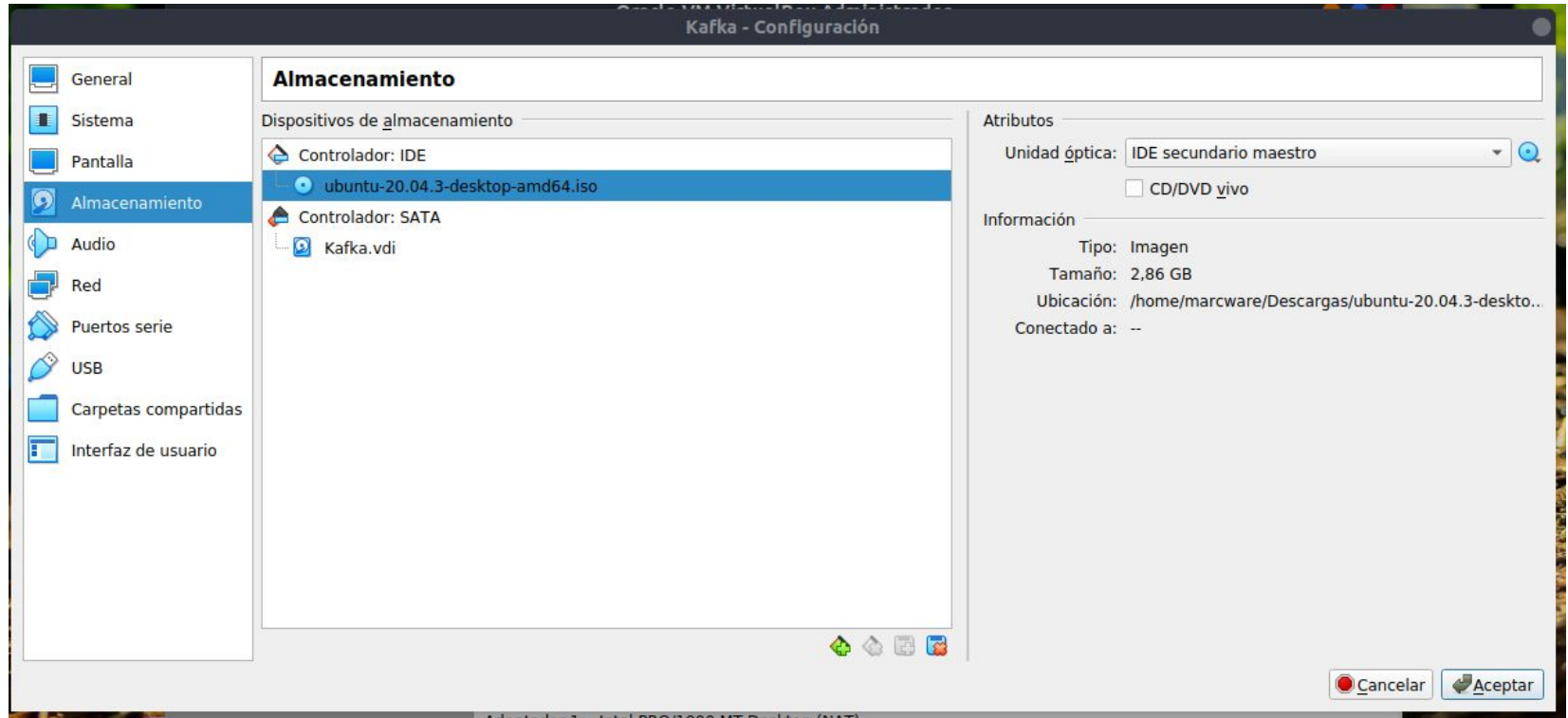
Instalación



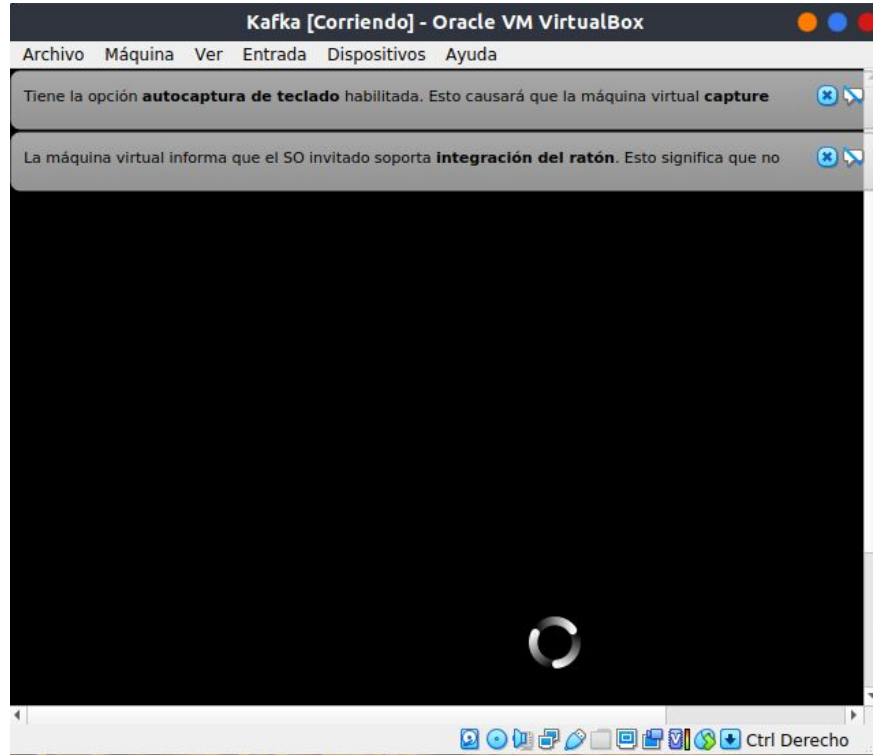
Instalación



Instalación



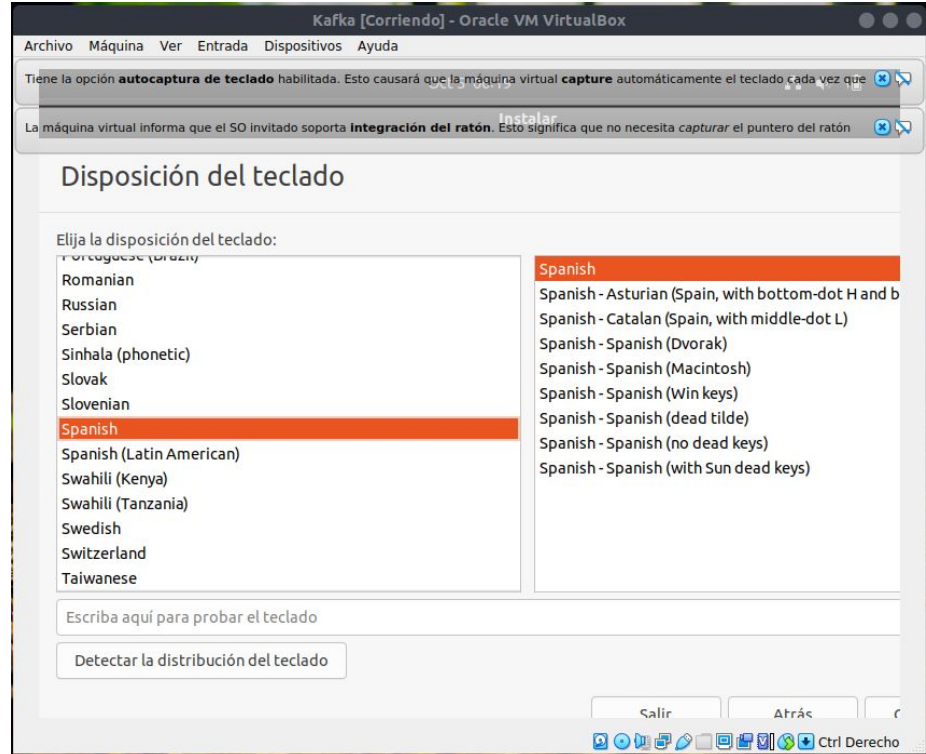
Instalación



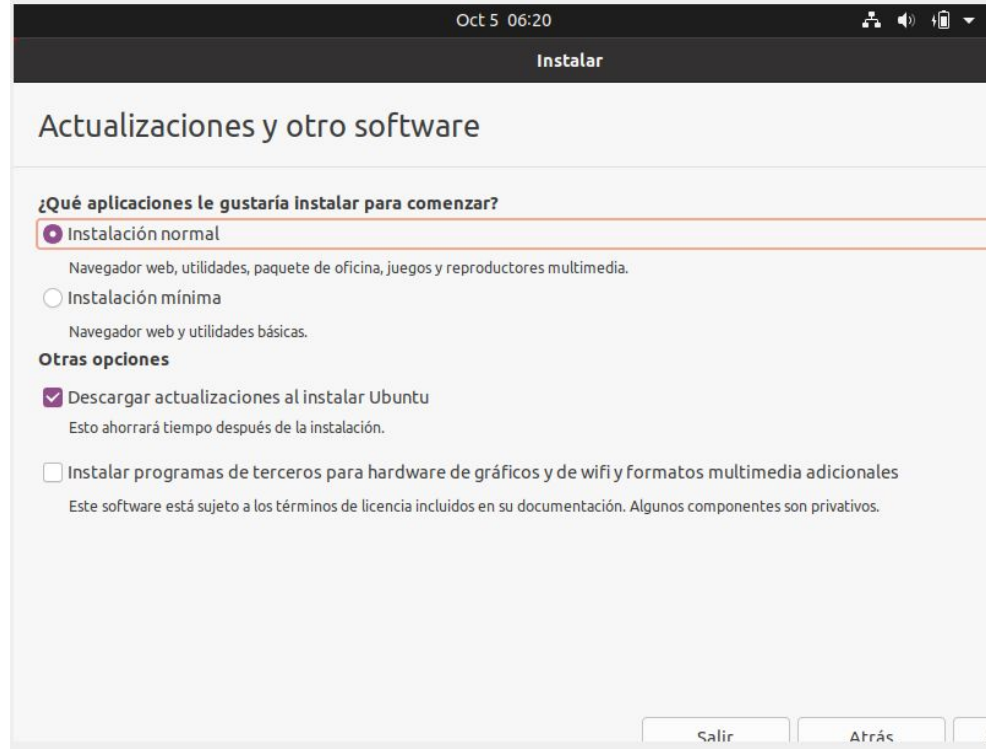
Instalación



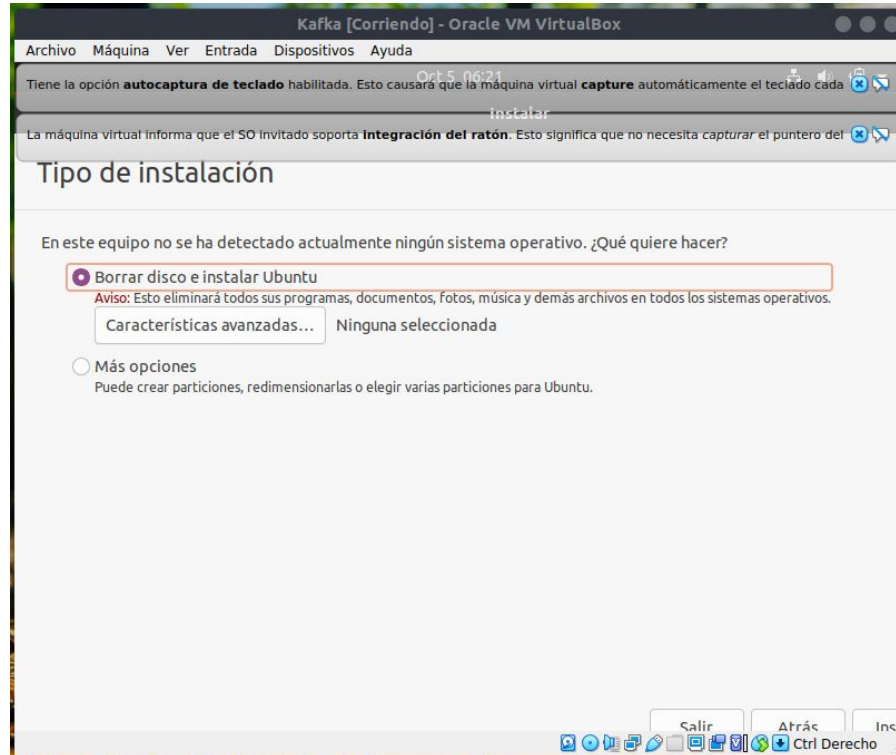
Instalación



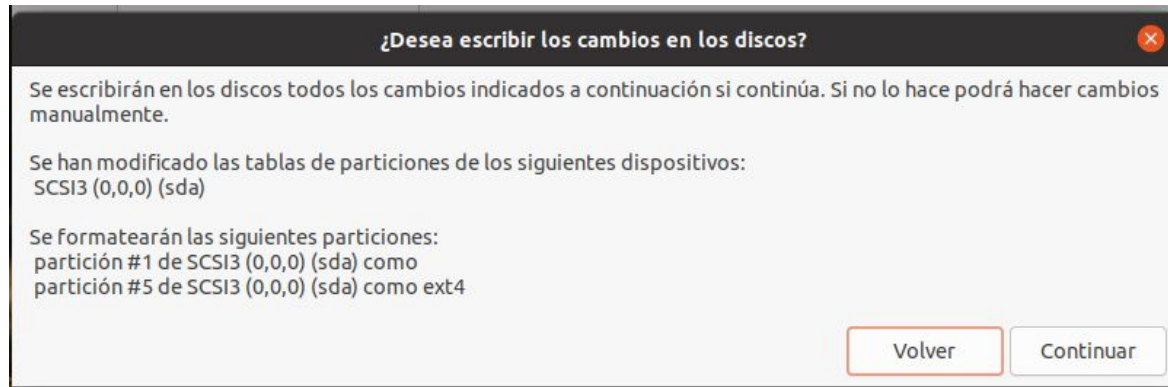
Instalación



Instalación



Instalación



Instalación



Instalación

Kafka [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Ora 5:08:23

Tiene la opción **autocaptura de teclado** habilitada. Esto causará que la máquina virtual **capture** automáticamente el teclado cada vez que se presiona una tecla.

Instalar

La máquina virtual informa que el SO invitado soporta **integración del ratón**. Esto significa que no necesita *capturar* el puntero del mouse.

¿Quién es usted?

Su nombre: ✓

El nombre de su equipo: ✓
El nombre que utiliza al comunicarse con otros equipos.

Elija un nombre de usuario: ✓

Elija una contraseña: **Contraseña corta**

Confirme su contraseña: ✓

☒ Iniciar sesión automáticamente

☐ Solicitar mi contraseña para iniciar sesión

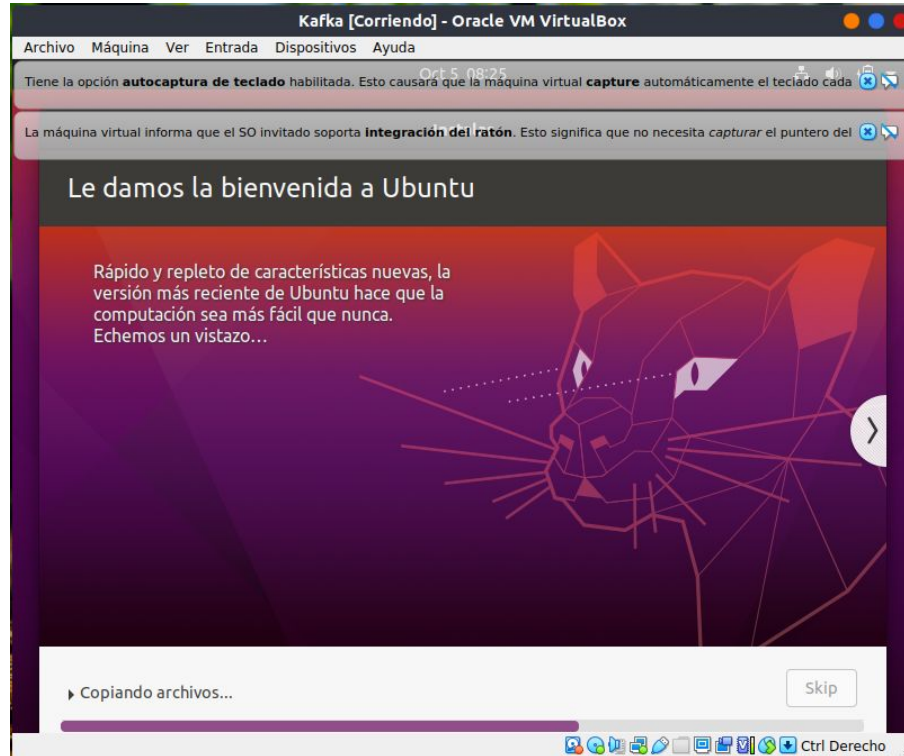
☐ Use Active Directory

You'll enter domain and other details in the next step.

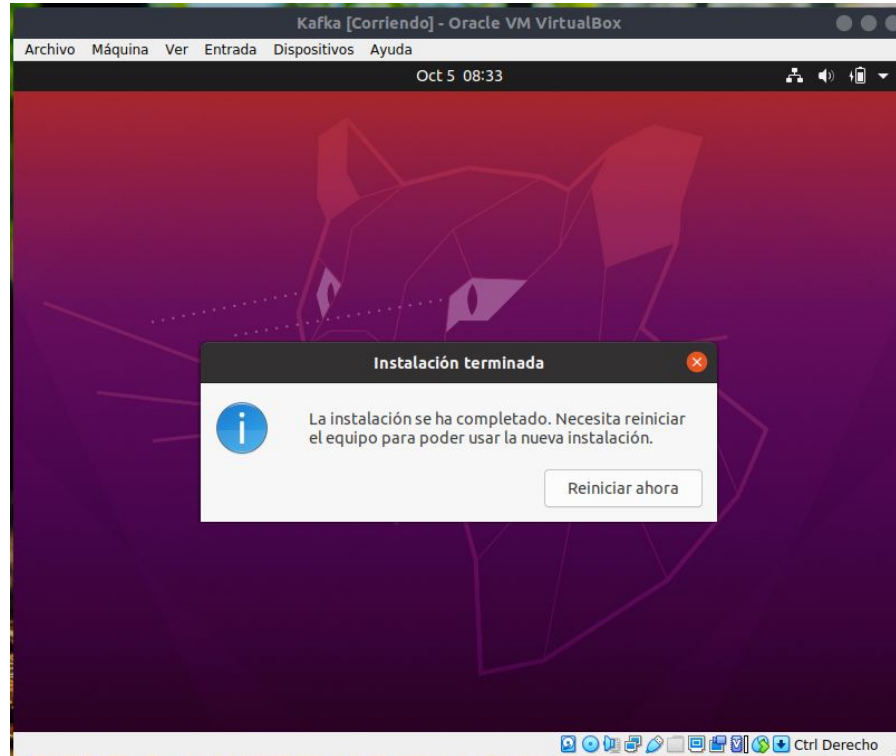
Atrás

Ctrl Derecho

Instalación



Instalación

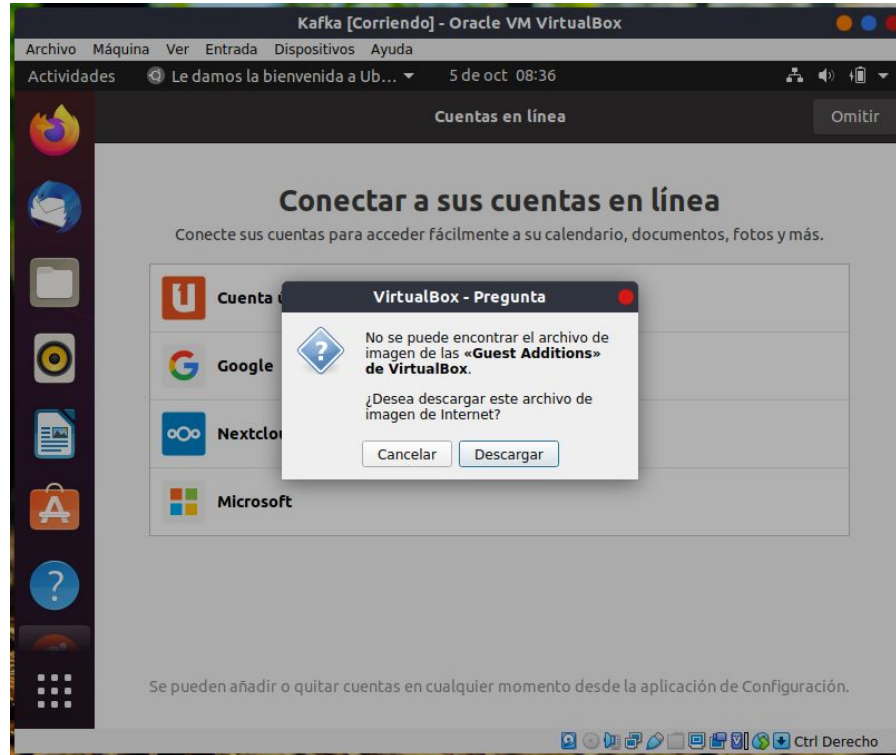


Instalación

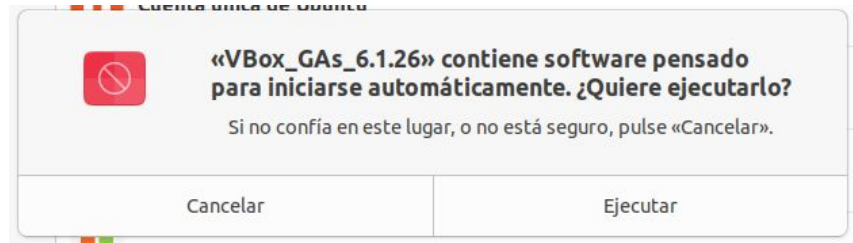
- `sudo apt update`
- `sudo apt upgrade`
- `sudo apt-get install build-essential gcc make perl dkms`
- `reboot`



Instalación



Instalación



Requisitos



Java

```
$ sudo apt install default-jre
```

```
kafka@kafka-VirtualBox:~/kafka$ java --version
```

```
openjdk 11.0.11 2021-04-20
```

```
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
```

```
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
```



ZooKeeper: Creando usuario

- `useradd zookeeper -m`
- `usermod --shell /bin/bash zookeeper`
- `passwd zookeeper`
 - Vamos a poner todos de contraseña zookeeper
- `usermod -aG sudo zookeeper`
- `sudo getent group sudo`



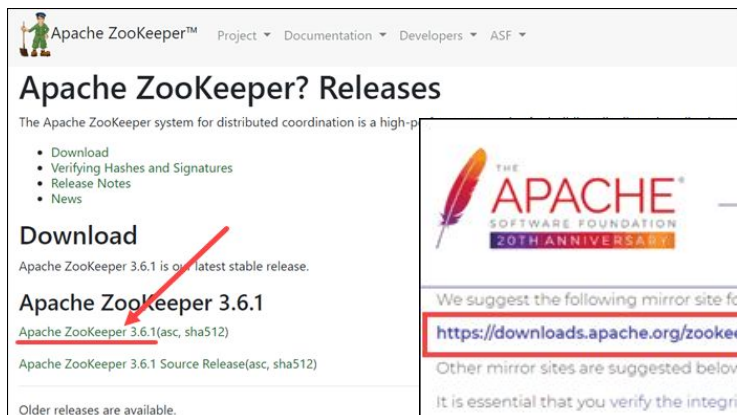
ZooKeeper: Creando Data Directory

- `sudo mkdir -p /data/zookeeper`
- `chown -R zookeeper:zookeeper /data/zookeeper`



ZooKeeper: Descarga

- http://zookeeper.apache.org/doc/current/zookeeperStarted.html#sc_Download



Apache ZooKeeper™ Project ▾ Documentation ▾ Developers ▾ ASF ▾

Apache ZooKeeper? Releases

The Apache ZooKeeper system for distributed coordination is a high-performance, fault-tolerant, open-source software library for maintaining a consistent view of replicated data across a collection of machines.

- Download
- Verifying Hashes and Signatures
- Release Notes
- News

Download

Apache ZooKeeper 3.6.1 is our latest stable release.

Apache ZooKeeper 3.6.1

[Apache ZooKeeper 3.6.1\(asc, sha512\)](#)

[Apache ZooKeeper 3.6.1 Source Release\(asc, sha512\)](#)

Older releases are available.



THE APACHE SOFTWARE FOUNDATION 20TH ANNIVERSARY

COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

Projects ▾ People ▾ Community ▾ License ▾ Sponsors ▾

We suggest the following mirror site for your download:

<https://downloads.apache.org/zookeeper/zookeeper-3.6.1/apache-zookeeper-3.6.1-bin.tar.gz>

Other mirror sites are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature ([.asc](#) file) or a hash ([.md5](#) or [.sha*](#) file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

HTTP

<https://downloads.apache.org/zookeeper/zookeeper-3.6.1/apache-zookeeper-3.6.1-bin.tar.gz>

ZooKeeper: Descarga

- `sudo wget`
<https://downloads.apache.org/zookeeper/zookeeper-3.6.1/apache-zookeeper-3.6.1-bin.tar.gz>
- `sudo tar -xvf apache-zookeeper-3.6.1-bin.tar.gz`
- `mv apache-zookeeper-3.6.1-bin zookeeper`
- `sudo mkdir /opt/zookeeper`
- `sudo mv zookeeper/ /opt/`
- `sudo chown -R zookeeper:zookeeper /opt/zookeeper`



ZooKeeper: Configuración

- `sudo nano /opt/zookeeper/conf/zoo.cfg`

```
tickTime = 2000  
dataDir = /data/zookeeper  
clientPort = 2181  
initLimit = 5  
syncLimit = 2
```

Guarda y sal del editor.



ZooKeeper: Encender servicio

- `sudo bin/zkServer.sh start`

Se debería mostrar que ZooKeeper se ha iniciado tras escribir el siguiente comando.

```
kafka@kafka-VirtualBox:/opt/zookeeper$ sudo bin/zkServer.sh start  
/usr/bin/java  
ZooKeeper JMX enabled by default  
Using config: /opt/zookeeper/bin/./conf/zoo.cfg  
Starting zookeeper ... STARTED
```



ZooKeeper: Conexión al servidor

Una vez iniciado el servicio, podemos conectarnos al servidor de ZooKeeper con el comando:

- `bin/zkCli.sh -server 127.0.0.1:2181`

Espera a que se confirme que estás conectado.

Para ver una lista de comandos disponibles, ejecuta `help`

Para cerrar la sesión escribe `quit`

Para detener el servicio ZooKeeper, ejecute el comando `bin/zkServer.sh stop`



ZooKeeper: Creación de un archivo de servicio del sistema

1. Crear un nuevo archivo zookeeper.service en un editor de texto de su elección:

- `sudo nano /etc/systemd/system/zookeeper.service`



ZooKeeper: Creación de un archivo de servicio del sistema

2. Pegar el siguiente contenido en el archivo:

```
[Unit]
Description=Zookeeper Daemon
Documentation=http://zookeeper.apache.org
Requires=network.target
After=network.target

[Service]
Type=forking
WorkingDirectory=/opt/zookeeper
User=zookeeper
Group=zookeeper
ExecStart=/opt/zookeeper/bin/zkServer.sh start /opt/zookeeper/conf/zoo.cfg
ExecStop=/opt/zookeeper/bin/zkServer.sh stop /opt/zookeeper/conf/zoo.cfg
ExecReload=/opt/zookeeper/bin/zkServer.sh restart /opt/zookeeper/conf/zoo.cfg
TimeoutSec=30
Restart=on-failure

[Install]
WantedBy=default.target
```



ZooKeeper: Creación de un archivo de servicio del sistema

3. Guardar y salir del archivo.

4. Recargar el servicio systemd ejecutando

- `systemctl daemon-reload`

5. A continuación, iniciar el servicio ZooKeeper y permitir que se inicie en el arranque:

- `systemctl start zookeeper`
- `systemctl enable zookeeper`

6. Verificar que el servicio está activo con el comando:

- `systemctl status zookeeper`



Instalación y configuración Kafka



Creación de un usuario para Kafka

Kafka ya tiene integrado el zookeeper, no hace falta tener los pases anteriores.

Instalando kafka tenemos ficheros de configuración de zookeeper



Creación de un usuario para Kafka

Kafka gestiona puede gestionar las solicitudes a través de la red, por ese motivo y para no comprometer el sistema y minimizar daños si el sistema es atacado.

Tendremos nuestro usuario root, otro usuario no root y el usuario para kafka.

```
$ sudo useradd kfk -m
```



Creación de un usuario para Kafka

Vamos a poner todos la contraseña kfk, como el nombre de usuario

```
$ sudo passwd kfk
```



Creación de un usuario para Kafka

Añadir el usuario a sudoers

```
$ sudo adduser kafka sudo  
$ su -l kafka
```



Descarga y extracción de archivos binarios de Kafka

Creamos una carpeta para descargar los binarios

```
$ mkdir ~/Downloads
```

Por si no tenemos instalado curls

```
$ sudo apt install curl
```

Descargamos los binarios

https://www.apache.org/dyn/closer.cgi?path=/kafka/3.0.0/kafka_2.13-3.0.0.tgz

Creamos un directorio para los fichero de Kafka

```
$ mkdir ~/kafka && cd ~/kafka
```

Extraemos los ficheros



Configuración del servidor de Kafka

Las opciones de configuración de Kafka se especifican en `server.properties`. Abra este archivo con nano o su editor favorito:

```
nano ~/kafka/config/server.properties
```

Agreguemos una configuración que nos permita eliminar temas de Kafka. Añada lo siguiente a la parte inferior del archivo:

```
delete.topic.enable = true
```



Creación de archivos de unidad systemd e inicio del servidor de Kafka

Cree el archivo de unidad para zookeeper:

```
$ sudo nano /etc/systemd/system/zookeeper.service
```

Introduzca la siguiente definición de unidad en el archivo creado anteriormente:

```
[Unit]
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=kafka
ExecStart=/home/kafka/kafka/bin/zookeeper-server-start.sh /home/kafka/kafka/config/zookeeper.properties
ExecStop=/home/kafka/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target
```



Creación de archivos de unidad systemd e inicio del servidor de Kafka

A continuación, cree el archivo de servicio de systemd para kafka:

```
$ sudo nano /etc/systemd/system/kafka.service
```

Introduzca la siguiente definición de unidad en el archivo creado anteriormente:

```
[Unit]
Requires=zookeeper.service
After=zookeeper.service

[Service]
Type=simple
User=kafka
ExecStart=/bin/sh -c '/home/kafka/kafka/bin/kafka-server-start.sh
/home/kafka/kafka/config/server.properties > /home/kafka/kafka/kafka.log 2>&1'
ExecStop=/home/kafka/kafka/bin/kafka-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target
```



Creación de archivos de unidad systemd e inicio del servidor de Kafka

Ahora que se definieron las unidades, inicie Kafka con el siguiente comando:

```
$ sudo systemctl start kafka
```

Para asegurarse de que el servidor se haya iniciado de forma correcta, consulte los registros de diario para la unidad kafka:

```
$ sudo journalctl -u kafka
```

El resultado debe ser similar al siguiente:

Output

```
Jul 17 18:38:59 kafka-ubuntu systemd[1]: Started kafka.service.
```

Ahora tiene un servidor Kafka que escucha en el puerto 9092.



Creación de archivos de unidad systemd e inicio del servidor de Kafka

Aunque iniciamos el servicio kafka, si reiniciáramos nuestro servidor este no se iniciaría de forma automática. Para habilitar kafka en el arranque del servidor, ejecute lo siguiente:

```
$ sudo systemctl enable kafka
```

Ahora que iniciamos y habilitamos los servicios, revisemos la instalación.



Pruebas de instalación

Publicaremos y consumiremos un mensaje “**Hello World**” para asegurarnos de que el comportamiento del servidor Kafka sea correcto. Para publicar mensajes en Kafka es necesario lo siguiente:

- Un *productor* que permita la publicación de registros y datos en temas.
- Un *consumidor* que lea mensajes y datos de temas.

Primero, cree un tema llamado TutorialTopic escribiendo:

```
$ ~/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1  
--partitions 1 --topic TutorialTopic
```



Pruebas de instalación

Puede crear un productor desde la línea de comando usando el script `kafka-console-producer.sh`. Se necesitan el nombre de host y el puerto del servidor de Kafka, y un nombre de tema como argumentos.

Publique la cadena "Hello, World" en el tema `TutorialTopic` escribiendo lo siguiente:

```
$ echo "Hello, World" | ~/kafka/bin/kafka-console-producer.sh --broker-list  
localhost:9092 --topic TutorialTopic > /dev/null
```



Pruebas de instalación

A continuación, puede crear un consumidor de Kafka usando el script `kafka-console-consumer.sh`. Como argumentos se necesitan el nombre de host y puerto del servidor de ZooKeeper, y un nombre de tema.

El siguiente comando consume mensajes de `TutorialTopic`. Tenga en cuenta el uso del indicador de `--from-beginning`, que permite el consumo de mensajes publicados antes de iniciar el consumidor:

```
$ ~/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic TutorialTopic --from-beginning
```

Si no hay problemas de configuración, verá `Hello, World` en su terminal.



Creación de Tópicos



Creación de Tópicos

Para poder comunicar la información necesitamos levantar un **Broker** con listas de información.

Es necesario crear listas de **tópicos**.

Usamos el siguiente comando :

```
bin/kafka-topic-sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic foo
```

Kakfa trabaja mediante **clusters**, es por ello que necesitamos desgranar la información por línea de comandos para levantar el tópico **'foo'**.

Una vez creemos el tópico, observaremos un mensaje en la consola tal que:

```
$ >> 'Created topic foo'
```



Validando Tópicos



Validando Tópicos

Si queremos saber cuántos tópicos tenemos levantados en nuestro entorno usaremos el siguiente comando:

```
bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

Seguidamente veremos la lista por la terminal.

```
$ >> foo
```



Inicio y cierre del cluster



Inicio del cluster

Teniendo un cluster de Kafka podemos ser escalables y dar servicio a todos nuestros usuario .

Conceptos sobre Cluster:

- **Cluster:** conjunto de servidores que se reparten la carga de trabajo
- **Nodo:** Servidor incluido dentro del Cluster
- **Partición:** El concepto de partición se refiere a dividir el trabajo o datos.
- **Réplica:** Se refiere a tener copias de particiones. El objetivo es que en caso de falla de un servidor no perder los datos.

Todo esto se considera alta disponibilidad y tolerancia a fallos.



Herramientas de Gestión de Topics



Herramientas de Gestión de Topics

Los temas se agregan y modifican utilizando la herramienta de temas:

```
> bin/kafka-topics.sh
```

Table 1. TopicCommand's Actions

Action	Description
<code>--alter</code>	Alters the number of partitions, replica assignment, and/or configuration of a topic or topics
<code>--create</code>	Creates a new topic
<code>--delete</code>	Deletes a topic or topics
<code>--describe</code>	Describes a topic or topics
<code>--list</code>	Lists available topics



Herramientas de Gestión de Topics

Table 2. TopicCommand's Options

Option	Description
<code>config</code>	A topic configuration override for the topic being worked on (as <code>name=value</code>)
<code>disable-rack-aware</code>	Disable rack aware replica assignment
<code>if-not-exists</code>	When creating a topic(s), the action will only execute if the topic does not already exist
<code>partitions</code>	<div>The number of partitions of the topic being created or altered<div><div>Warning</div><div>If partitions are increased for a topic that has a key, the partition logic or ordering of the messages will be affected.</div></div></div>
<code>replica-assignment</code>	A list of manual partition-to-broker assignments for the topic being created or altered
<code>replication-factor</code>	The replication factor for each partition of the topic being created
<code>topic</code>	A topic or topics to be altered , created , deleted , or listed and described (through getTopics)



Herramientas de Gestión de Topics

```
$ ./bin/kafka-topics.sh --create --zookeeper :2181 --replication-factor 1 --partitions 1 --topic my-topic
```

```
$ ./bin/kafka-topics.sh --list --zookeeper :2181 my-topic
```

```
$ ./bin/kafka-topics.sh \
```

```
--zookeeper :2181 \
```

```
--describe \
```

```
--topic my-topic
```



Posibilidades y Auto-configuración de topics



Posibilidades y Auto-configuración de topics

Aspectos básicos del registro

- `log.dirs=<Log_dir_path>`
 - Especifique el directorio o directorios donde desea almacenar los archivos de registro. Por ejemplo, `la/home/logs1`, `la/home/logs2`, `la/home/logs3`
- `num.partitions=4`
 - Especifique el número de particiones que se utilizan para procesar un tema. Especifique la mitad del número de procesadores físicos en el servidor. Por ejemplo, si tiene 8 procesadores, especifique 4.
- `num.recovery.threads.per.data.dir=1`
 - Especifique el número máximo de hebras que se utilizan para la recuperación de registro para cada directorio de datos.



Posibilidades y Auto-configuración de topics

Aspectos básicos del registro

- `log.index.size.max.bytes=154624`
 - Especifique el tamaño máximo del índice de desplazamiento en bytes.
- `log.index.interval.bytes=4096`
 - Especifique el intervalo en bytes cuando se añada una entrada al índice de desplazamiento.
- `message.max.bytes=1000000`
 - Especifique el tamaño máximo de mensajes que el servidor puede recibir.
- `auto.create.topics.enable=true`
 - Utilice esto para permitir la creación automática de temas en el servidor.



Instalación y configuración Con Cluster



Docker Cluster

DEMO TIME



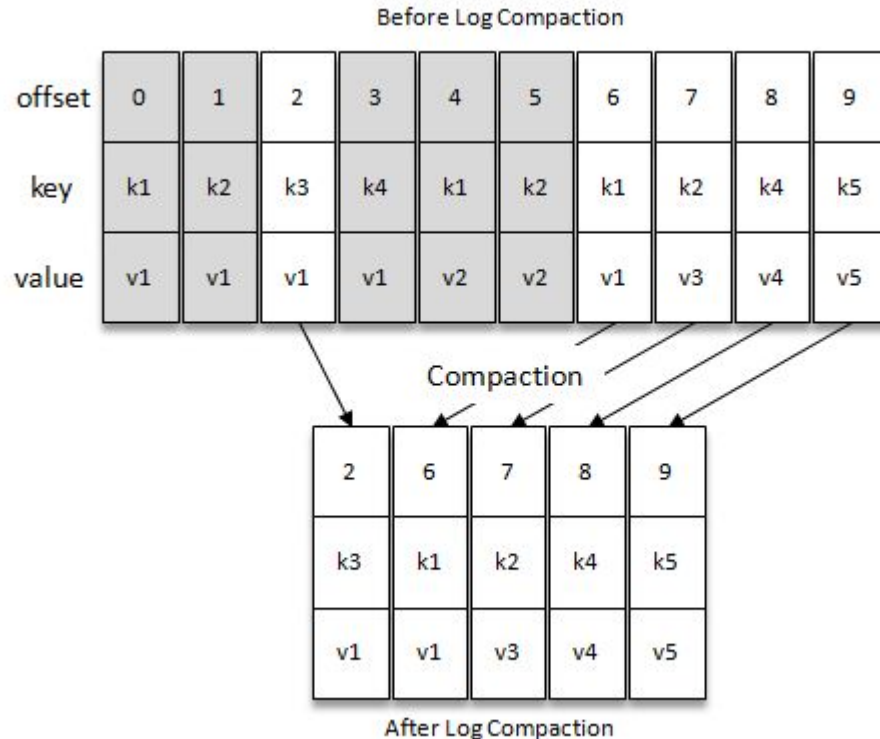
Compactación de históricos



Compactación de históricos

La compactación de registros en Kafka se refiere a una forma de limpiar los datos obsoletos proporcionados además de las reglas predeterminadas de eliminación de registros. Como se muestra en la figura a continuación, Log Compaction solo retiene la última versión para diferentes valores con la misma clave.

El proceso de eliminar versiones antiguas de entidades de temas compactados se denomina log cleaning.



Compactación de históricos

Nombre	Descripción
Tail	Esta parte del registro ya se ha limpiado y se garantiza que no tendrá duplicados
Head	Esta porción no ha sido limpiada todavía puede contener duplicados y se puede limpiar
Active segment	Se están escribiendo nuevos registros entrantes. No es elegible para compactación.

```
dirty ratio = (size of the head) / (size of head + tail)
```

Por defecto `min.cleanable.dirty.ratio` es 0.5.



Compactación de históricos

Offset	1	2	3	4	5	6	7	8	9	10
Key	K1	K2	K3	K2	K1	K3	K2	K1	K4	K1
Value	V1a	V2a	V3a	V2b	V1b	V3B	V2c	V1c	V4a	V1d



Compactación de históricos

Offset						6	7	8	9	10
Key						K3	K2	K1	K4	K1
Value						V3B	V2c	V1c	V4a	V1d

