

Manual: Instalación de Apache Cassandra con Docker

Prerrequisitos

1. Verificar que Docker Desktop esté instalado

- **Windows 10/11:** Descargar e instalar Docker Desktop desde docker.com
- **Requisitos mínimos:**
 - Windows 10 64-bit: Pro, Enterprise, o Education (Build 15063 o superior)
 - Habilitar Hyper-V y contenedores de Windows
 - Al menos 4 GB de RAM

2. Verificar instalación de Docker

Abrir **Command Prompt (CMD)** o **PowerShell** como administrador y ejecutar:

```
bash
```

```
docker --version
```

Deberías ver algo como: `Docker version 20.10.x`

Paso 1: Descargar la imagen de Cassandra

Descargar la imagen oficial de Cassandra

```
bash
```

```
docker pull cassandra:4.1.9
```

Explicación: Este comando descarga la imagen oficial de Apache Cassandra versión 4.1.9 desde Docker Hub. Esta imagen ya incluye Java y todas las dependencias necesarias.

Verificar que la imagen se descargó correctamente

```
bash
```

```
docker images
```

Deberías ver la imagen `cassandra` con tag `4.1.9`.

Paso 2: Crear y ejecutar el contenedor de Cassandra

Comando básico para crear el contenedor

bash

```
docker run --name my-cassandra -p 9042:9042 -d cassandra:4.1.9
```

Explicación de parámetros:

- `--name my-cassandra`: Asigna el nombre "my-cassandra" al contenedor
- `-p 9042:9042`: Mapea el puerto 9042 del contenedor al puerto 9042 del host
- `-d`: Ejecuta el contenedor en modo detached (segundo plano)
- `cassandra:4.1.9`: Especifica la imagen a usar

Comando avanzado (recomendado para producción)

bash

```
docker run --name my-cassandra \  
-p 9042:9042 \  
-p 7000:7000 \  
-p 7001:7001 \  
-p 7199:7199 \  
-p 9160:9160 \  
-v cassandra-data:/var/lib/cassandra \  
-e CASSANDRA_CLUSTER_NAME=MyCluster \  
-e CASSANDRA_DC=datacenter1 \  
-e CASSANDRA_RACK=rack1 \  
-d cassandra:4.1.9
```

Explicación de puertos:

- `9042`: Puerto CQL nativo (principal)
- `7000`: Puerto de comunicación inter-nodo
- `7001`: Puerto SSL para comunicación inter-nodo
- `7199`: Puerto JMX
- `9160`: Puerto Thrift (legacy)

Volumen persistente:

- `-v cassandra-data:/var/lib/cassandra`: Crea un volumen persistente para los datos

Paso 3: Verificar que el contenedor esté funcionando

Verificar el estado del contenedor

```
bash
```

```
docker ps
```

Deberías ver algo como:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
abc123def456	cassandra:4.1.9	"docker-entrypoint.s..."	30 seconds ago	Up 29 seconds	0.0.0.0:9042->9042/tcp	my-cassandra

Verificar los logs del contenedor

```
bash
```

```
docker logs my-cassandra
```

Indicadores de éxito:

- Buscar mensajes como: "Startup complete"
- "Listening for thrift clients..."
- "Starting listening for CQL clients"

Nota: Cassandra puede tardar 1-3 minutos en inicializarse completamente.

Paso 4: Verificar la instalación de Cassandra

Verificar la versión de Cassandra

```
bash
```

```
docker exec -it my-cassandra nodetool version
```

Salida esperada: `ReleaseVersion: 4.1.9`

Verificar el estado del nodo

```
bash
```

```
docker exec -it my-cassandra nodetool status
```

Salida esperada: Debería mostrar un nodo "UN" (Up Normal)

Acceder al shell CQL

```
bash
```

```
docker exec -it my-cassandra cqlsh
```

Dentro de CQLsh, verificar la versión:

```
sql
```

```
SELECT release_version FROM system.local;
```

Para salir de CQLsh:

```
sql
```

```
EXIT;
```

Paso 5: Comandos básicos de administración

Iniciar el contenedor (si está detenido)

```
bash
```

```
docker start my-cassandra
```

Detener el contenedor

```
bash
```

```
docker stop my-cassandra
```

Reiniciar el contenedor

```
bash
```

```
docker restart my-cassandra
```

Eliminar el contenedor

bash

`docker rm my-cassandra`

Ver información detallada del contenedor

bash

`docker inspect my-cassandra`

Paso 6: Pruebas básicas con CQL

Acceder a CQLsh

bash

`docker exec -it my-cassandra cqlsh`

Comandos CQL básicos para probar

sql

-- Crear un keyspace de prueba

```
CREATE KEYSPACE IF NOT EXISTS test_keyspace  
WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
```

-- Usar el keyspace

```
USE test_keyspace;
```

-- Crear una tabla de ejemplo

```
CREATE TABLE IF NOT EXISTS usuarios (  
  id UUID PRIMARY KEY,  
  nombre TEXT,  
  email TEXT,  
  fecha_creacion TIMESTAMP  
);
```

-- Insertar un registro

```
INSERT INTO usuarios (id, nombre, email, fecha_creacion)  
VALUES (uuid(), 'Juan Pérez', 'juan@example.com', toTimestamp(now()));
```

-- Consultar los datos

```
SELECT * FROM usuarios;
```

-- Listar todos los keyspaces

```
DESCRIBE KEYSPACES;
```

-- Salir

```
EXIT;
```

Paso 7: Conexión desde herramientas externas

Configuración para DataGrip (JetBrains)

- **Host:** localhost
- **Puerto:** 9042
- **Driver:** Apache Cassandra
- **Autenticación:** No auth (para instalación básica)

Configuración para otras herramientas

- **URL de conexión:** jdbc:cassandra://localhost:9042

- **Puerto CQL:** 9042
 - **Credenciales por defecto:** cassandra/cassandra (si se requiere)
-

Paso 8: Solución de problemas comunes

El contenedor no inicia

bash

Verificar logs para errores

`docker logs my-cassandra`

Verificar si el puerto está ocupado

`netstat -ano | findstr :9042`

Conexión rechazada

bash

Verificar que Cassandra esté completamente iniciado

`docker exec -it my-cassandra nodetool status`

Verificar que el puerto esté mapeado correctamente

`docker port my-cassandra`

Rendimiento lento

bash

Asignar más memoria al contenedor

`docker run --name my-cassandra -p 9042:9042 -m 4g -d cassandra:4.1.9`

Datos persistentes

bash

Crear un contenedor con datos persistentes

`docker run --name my-cassandra \
-p 9042:9042 \
-v cassandra-data:/var/lib/cassandra \
-d cassandra:4.1.9`

Paso 9: Comandos útiles para desarrollo

Backup básico

bash

Hacer snapshot

`docker exec my-cassandra nodetool snapshot`

Listar snapshots

`docker exec my-cassandra nodetool listsnapshots`

Monitoreo

bash

Ver estadísticas del nodo

`docker exec my-cassandra nodetool info`

Ver compactaciones en progreso

`docker exec my-cassandra nodetool compactionstats`

Limpieza

bash

Detener y eliminar contenedor

`docker stop my-cassandra`

`docker rm my-cassandra`

Eliminar imagen

`docker rmi cassandra:4.1.9`

Eliminar volumen (¡CUIDADO! Esto elimina todos los datos)

`docker volume rm cassandra-data`

Resumen de ventajas de usar Docker

✅ **Sin instalación de Java:** Docker incluye todo lo necesario ✅ **Aislamiento:** No interfiere con otras aplicaciones ✅ **Fácil limpieza:** Eliminar contenedor = limpieza completa ✅ **Portabilidad:** Funciona igual en Windows, Mac, Linux ✅ **Versiónado:** Fácil cambiar entre versiones de Cassandra ✅ **Desarrollo:** Ideal para pruebas y desarrollo local

Comandos de referencia rápida

bash

Descargar imagen

`docker pull cassandra:4.1.9`

Crear y ejecutar contenedor

`docker run --name my-cassandra -p 9042:9042 -d cassandra:4.1.9`

Verificar estado

`docker ps`

Acceder a CQL

`docker exec -it my-cassandra cqlsh`

Ver logs

`docker logs my-cassandra`

Detener

`docker stop my-cassandra`

Iniciar

`docker start my-cassandra`

Eliminar

`docker rm my-cassandra`

¡Listo! Ahora tienes Apache Cassandra 4.1.9 ejecutándose en Docker sin necesidad de instalar Java u otras dependencias en tu sistema Windows.