**MSSQL 2019 Server on Docker Container:**

Prerequisites 'OS requirements':

To install Docker Engine, you need the 64-bit version of one of these Ubuntu versions:

Ubuntu Groovy 20.10 Ubuntu Focal 20.04 (LTS) Ubuntu Bionic 18.04 (LTS) Ubuntu Xenial 16.04 (LTS) Docker Engine is supported on x86\_64 (or amd64), armhf, and arm64 architectures.

**NOTE: in this example we are using Ubuntu Bionic 18.04 (LTS)**

Install using the repository:

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

**# Update the apt package index and install packages to allow apt to use a repository over HTTPS:**

$ sudo apt-get update && sudo apt-get install

$ sudo apt-transport-https ca-certificates curl gnupg lsb-release

**# Add Docker’s official GPG key:**

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

**# Use the following command to set up the stable repository.**

To add the nightly or test repository, add the word nightly or test (or both) after the word stable in the commands below.

$ echo \

"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

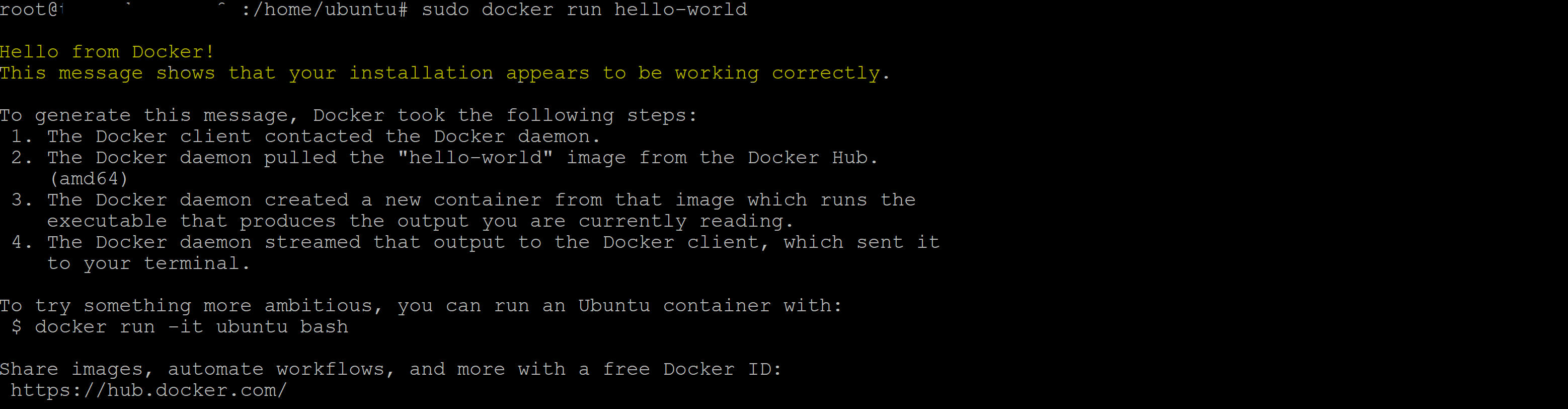
**# Now install Docker:**

$ sudo apt-get update

$ sudo apt-get install docker-ce docker-ce-cli containerd.io

**# Verify that Docker Engine is installed correctly by running the hello-world image.**

$ sudo docker run hello-world

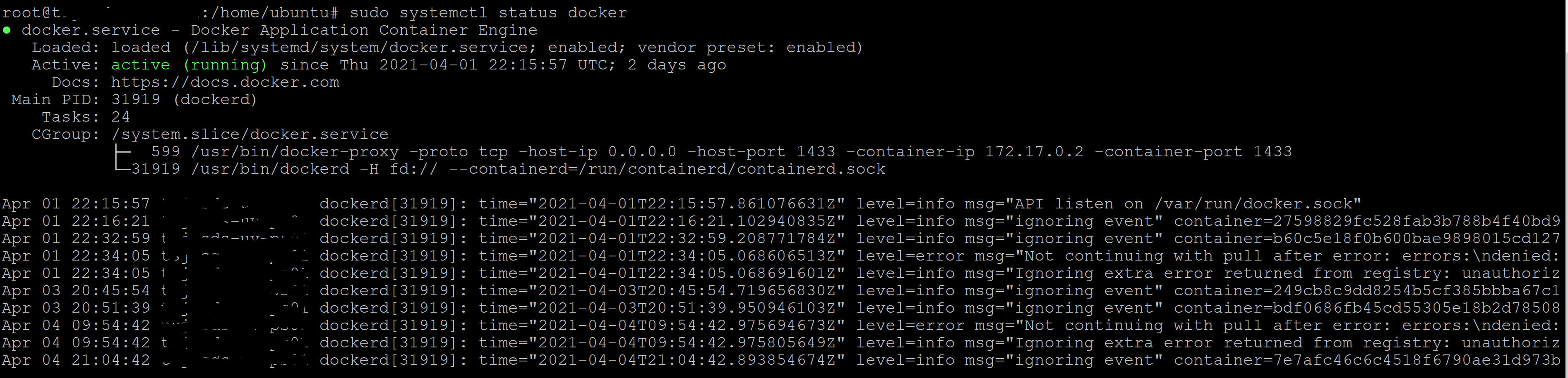


**# This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits**

**You can start Docker and check status:**

$ sudo systemctl start docker

$ sudo systemctl status docker



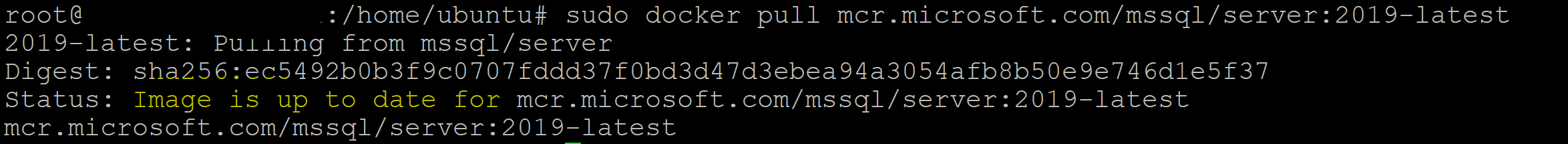
As you see above, I already installed MSSQL image and it’ running on port 1433 on localhost.

**# Pull and run the 2019 container image**

Before starting the following steps, make sure that you have selected your preferred shell (bash, PowerShell, or cmd) at the top of this article.

**# Pull the SQL Server 2019 Linux container image from Microsoft Container Registry.**

$ sudo docker pull mcr.microsoft.com/mssql/server:2019-latest



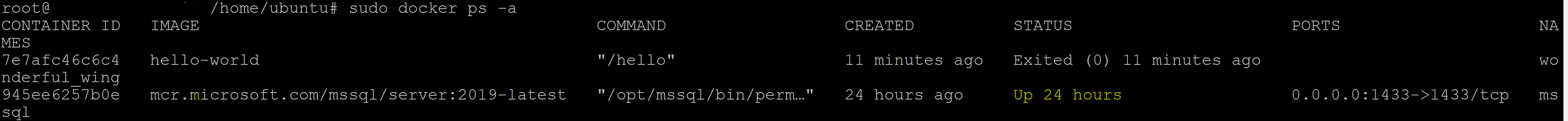
***As you see image is already downloaded and up to date.***

To run the container image with Docker, you can use the following command from a bash shell (Linux/macOS) or elevated PowerShell command prompt.

$ sudo docker run -e "ACCEPT\_EULA=Y" -e "SA\_PASSWORD=<pasteYourPasswordHere>" -p 1433:1433 --name mssql -h mssql -d mcr.microsoft.com/mssql/server:2019-latest

To view your Docker containers, use the docker ps command.

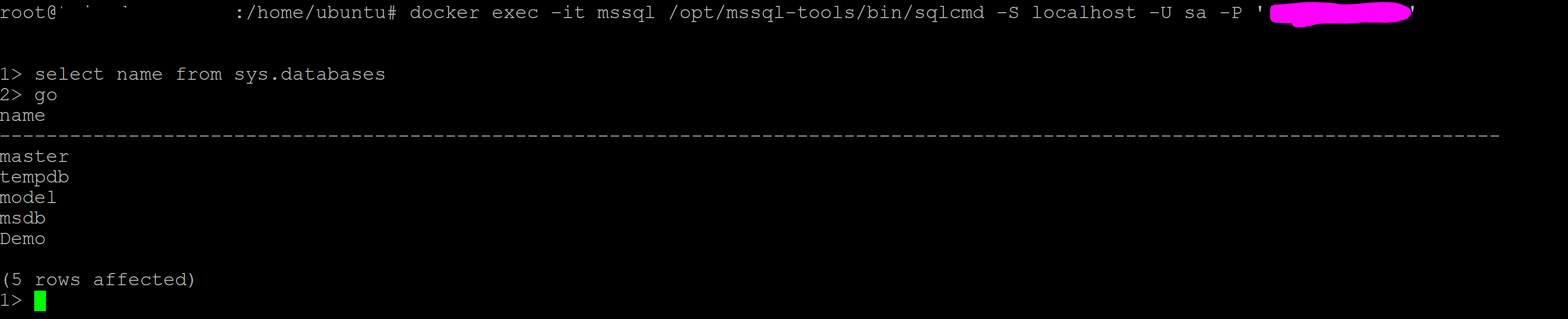
$ sudo docker ps -a



As you see, container is up for 24 hours and this is a quick way to check the status of your container.

# To access it via sqlcmd:

$ docker exec -it mssql /opt/mssql-tools/bin/sqlcmd -S localhost -U sa -P 'pasteYourPasswordHere'



And now, you’re inside the database!!

**# You can also access it through SSMS or even Azure Data Studio:**

