

---

Article

[Victor Sanner](#) · Apr 20, 2022 2m read

## Accessing an intersystems cache database inside a docker container + using Azure Functions

Hi,

I have wrote an article about how to install the intersystems cache driver in a Docker container, and then deploy it using Azure Functions:

[How to run a \(Python\) Azure Function as a Docker container & Deploy it using Bicep | Victor Sanner](#)

This might be useful to others, especially the dockerfile which I have copied below. This builds a debian docker container and installs the InterSystems Cache driver, which python can then use :)

To enable ssh & remote debugging on app service change the base image to the one below

```
# FROM mcr.microsoft.com/azure-functions/python:3.0-python3.6-appservice
FROM mcr.microsoft.com/azure-functions/python:3.0-python3.6
```

```
ENV AzureWebJobsScriptRoot=/home/site/wwwroot \
    AzureFunctionsJobHost__Logging__Console__IsEnabled=true
```

```
# Add ODBC Driver requirements
```

```
RUN apt-get update \
    && apt-get install unixodbc -y \
    && apt-get install unixodbc-dev -y \
    && apt-get install freetds-dev -y \
    && apt-get install freetds-bin -y \
    && apt-get install tdsodbc -y \
    && apt-get install --reinstall build-essential -y
```

```
# add pyodbc requirements
```

```
RUN apt-get install python-pyodbc -y
```

```
#copy the driver to container
```

```
COPY /Driver /usr/lib/intersystems/odbc
```

```
#run install script
```

```
RUN /usr/lib/intersystems/odbc/ODBCinstall
```

```
#copy the odbc configuration file to the container
```

```
COPY odbc.ini /etc/intersystemsodbc.ini
```

```
#make the odbc library aware that we have added a new driver
```

```
RUN odbcinst -i -d -f /etc/intersystemsodbc.ini
```

```
#symlink the odbc driver, not sure if its required
```

```
RUN ln -s /usr/lib/x86_64-linux-gnu/libodbc.so.2.0.0 /usr/lib/x86_64-linux-gnu/odbc/libodbc.so
```

## Accessing an intersystems cache database inside a docker container + using Azure Functions

Published on InterSystems Developer Community (<https://community.intersystems.com>)

---

```
#copy the Azure Functions project to the docker image
```

```
COPY . /home/site/wwwroot
```

```
#install the python libraries as you would usually do
```

```
RUN cd /home/site/wwwroot && pip install -r requirements.txt
```

I hope this saves some people a lot of time, because it was quite a challenge to build this!

[#Azure](#) [#Docker](#) [#Caché](#)

---

### Source

**URL:**<https://community.intersystems.com/post/accessing-intersystems-cache-database-inside-docker-container-using-azure-functions>