
COSMO CONSULT
Business-Software for People

agenda » Explain the problem Show the solution: Azure Retail Price API Tool for Querying Result analysis Dev container

CALCULATING AZURE COST BY USING THE PRICING API, A COMMAND LINE TOOL AND EXCEL

Tobias Fenster, April 20 2021

THE PROBLEM

CALCULATE AZURE COST IN DIFFERENT SCENARIOS

- Starting point:
  - A solution built on Azure services with some constant and some load-dependent parts
  - Price calculation for different configurations, e.g. small / medium / large
  - Different commitment types: pay as you go, reserved for one year, reserved for three years
  - Different usage types: dev/test vs. prod
- Calculate through the Azure pricing calculator: https://azure.microsoft.com/en-us/pricing/calculator

- 3 configurations x 3 commitment types x 2 usage types = 18 distinct calculations
- Not fun to create, even less fun to modify (what if we have to add or remove resources, try different options, find price differences in different regions, …)

THE SOLUTION

USE THE AZURE RETAIL PRICES API

- Unauthenticated API to give you access to Azure prices
- Very easy to get first results
- Some requests need unusual queries

CREATE A TOOL FOR QUERYING

- Direct REST calls are just not handy
- Store "configurations" and different variations of that
- Get the results in an easy to process format
- Basically only reading and transforming JSON, so JavaScript / TypeScript
- Deno invented by Ryan Dahl (see "10 things I regret about Node.js" at JSConf EU 2018) to fix Node.js problems
  - No node_modules!
  - Single binary
  - Combined runtime and package manager

ANALYZE THE RESULTS IN EXCEL

- You probably want to calculate anyway
- PowerQuery brings great new possibilities for transformation
- It's easy to get an overview and "tinker" with the numbers if needed

PUT DEVELOPMENT INTO A DEV CONTAINER

Now for the fancy stuff ;)

- Why? Standardized, separated dev environment with defined SDKs, tools etc.
- VS Code, but Linux – support for Windows containers is probably not coming short-term (Support Windows Containers · Issue #445 · microsoft/vscode-remote-release (github.com))
- Very good support on Windows 10 through Windows Subsystem for Linux (WSL2)
- Overall setup through devcontainer.json, container itself defined through Dockerfile

DEVCONTAINER.JSON

Now for the fancy stuff ;)

```
{
    "name": "Deno",
    "dockerFile": "Dockerfile",
    "settings": { "terminal.integrated.shell.linux": "/bin/bash" },
    "extensions": [
        "denoland.vscode-deno",
        "eamodio.gitlens",
        "humao.rest-client"
    ],
    "mounts": [
        "source=${localEnv:HOME}${localEnv:USERPROFILE},target=/host-home-
                    folder,type=bind,consistency=cached"
    ],
    "remoteUser": "vscode"
}
```

DOCKERFILE

Now for the fancy stuff ;)

```
FROM mcr.microsoft.com/vscode/devcontainers/base:debian-10

ENV DENO_INSTALL=/deno
RUN mkdir -p /deno \
    && curl -fsSL https://deno.land/x/install/install.sh | sh \
    && chown -R vscode /deno

ENV PATH=${DENO_INSTALL}/bin:${PATH} \
    DENO_DIR=${DENO_INSTALL}/.cache/deno

# [Optional] Uncomment this section to install additional OS packages.
# RUN apt-get update && export DEBIAN_FRONTEND=noninteractive \
#    && apt-get -y install --no-install-recommends <your-package-list-here>
```

Any

Questions?

Keep learning with Areopa!