Create and deploy SaaS applications using Azure App Service

Who am I

Site Reliability Engineer (SRE) @ **Warpnet**

Security Specialist @ **CJIB**

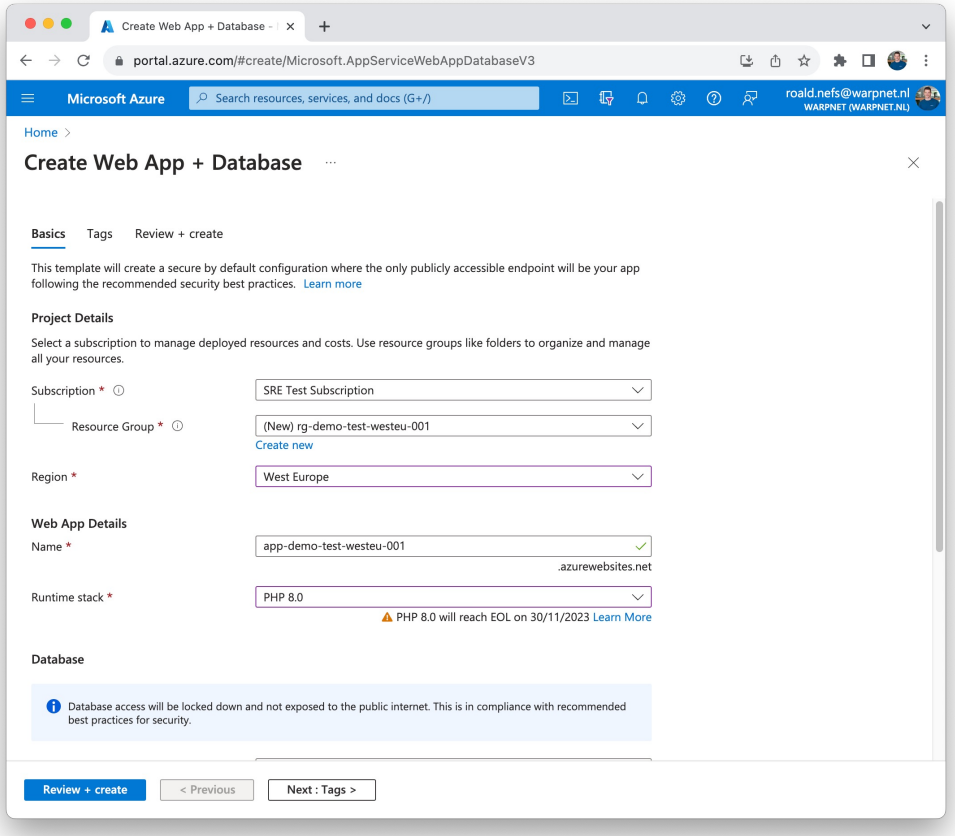
Prior Azure experience limited to AZ-900

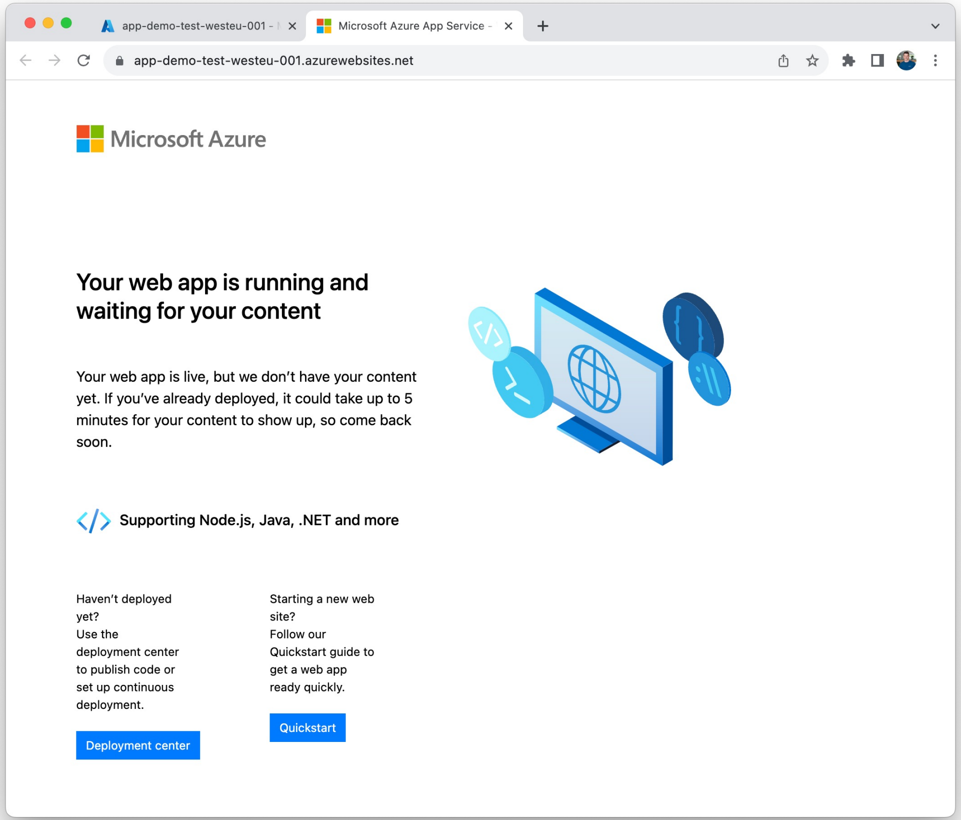
Let’s deploy a Laravel application

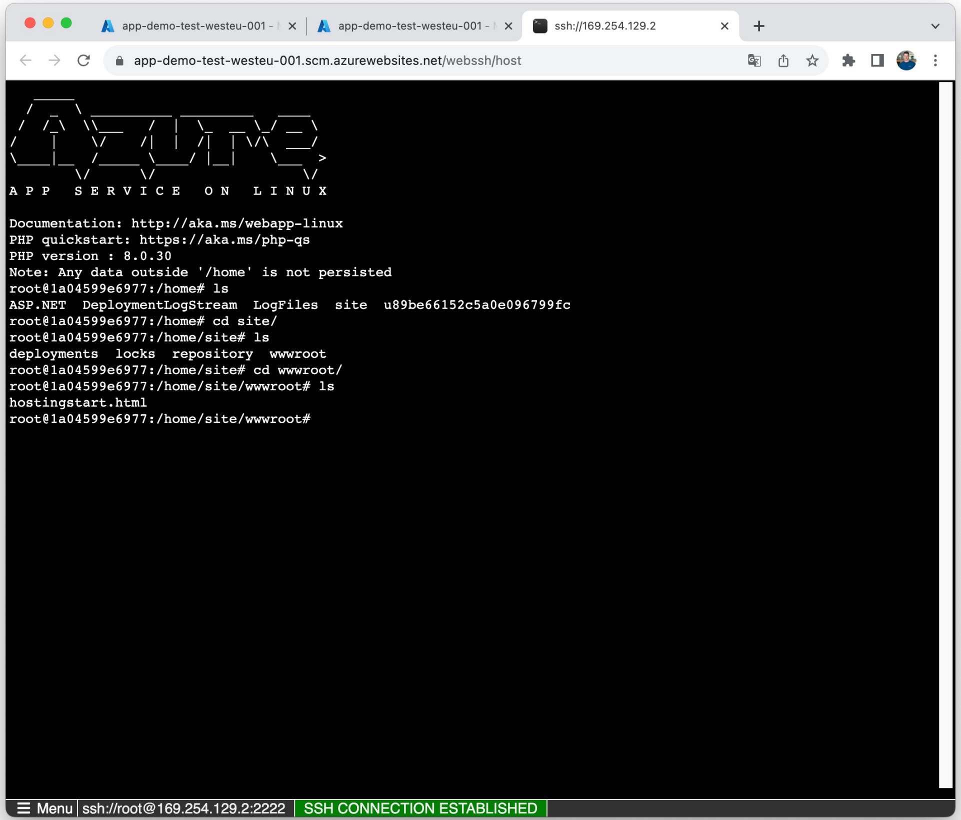
Create Web App + Database

Create **Web App + Database** on the PHP 8.0 runtime stack. This will result in the following resources:

- Resource group
- App Service plan
- App Service
- Virtual network
- Azure Database for MySQL – Flexible Server
- Private DNS zone




app-demo-test-westeu-001
Microsoft Azure App Service
app-demo-test-westeu-001.azurewebsites.net
Microsoft Azure
Your web app is running and waiting for your content
Your web app is live, but we don't have your content yet. If you've already deployed, it could take up to 5 minutes for your content to show up, so come back soon.
Supporting Node.js, Java, .NET and more
Haven't deployed yet? Use the deployment center to publish code or set up continuous deployment.
Deployment center
Starting a new web site? Follow our Quickstart guide to get a web app ready quickly.
Quickstart


app-demo-test-westeu-001
app-demo-test-westeu-001
ssh://169.254.129.2
app-demo-test-westeu-001.scm.azurewebsites.net/webssh/host
A P P   S E R V I C E   O N   L I N U X
Documentation: http://aka.ms/webapp-linux
PHP quickstart: https://aka.ms/php-qs
PHP version : 8.0.30
Note: Any data outside '/home' is not persisted
root@1a04599e6977:/home# ls
ASP.NET DeploymentLogStream LogFiles site u89be66152c5a0e096799fc
root@1a04599e6977:/home# cd site/
root@1a04599e6977:/home/site# ls
deployments locks repository wwwroot
root@1a04599e6977:/home/site# cd wwwroot/
root@1a04599e6977:/home/site/wwwroot# ls
hostingstart.html
root@1a04599e6977:/home/site/wwwroot#
Menu
ssh://root@169.254.129.2:2222
SSH CONNECTION ESTABLISHED

Database connectivity

Default Azure application settings do not match with Laravel, e.g.

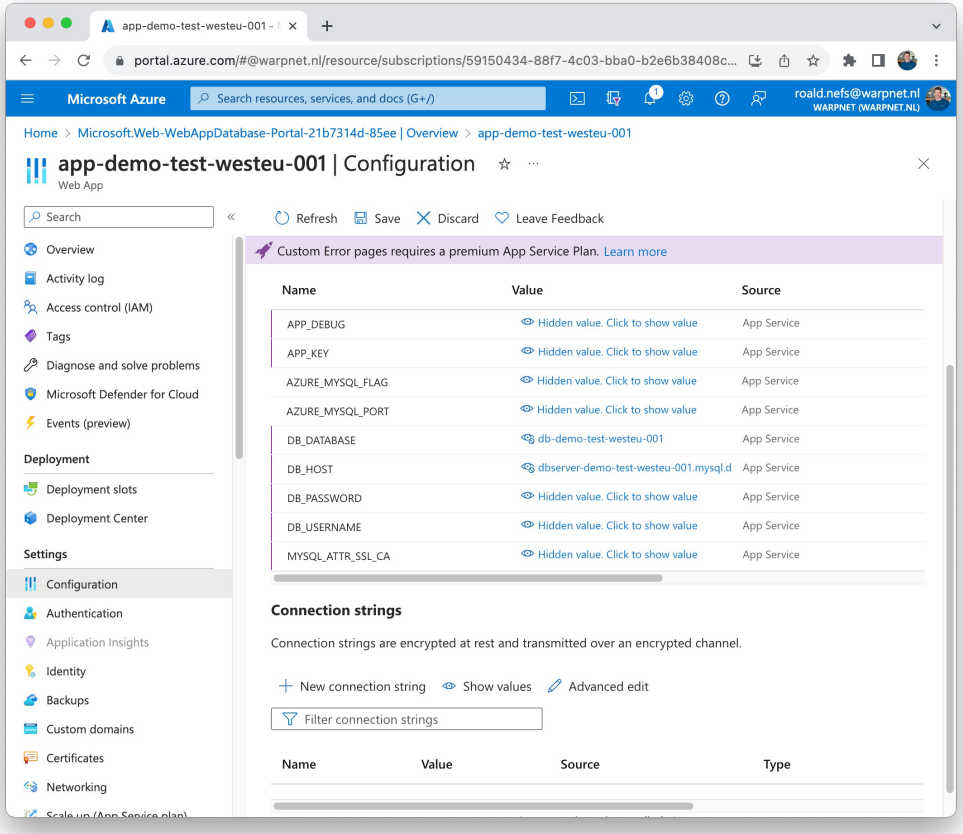
`AZURE_MYSQL_DBNAME` → `DB_DATABASE`

Add additional environment variables for Laravel:

`APP_DEBUG`

`APP_KEY`

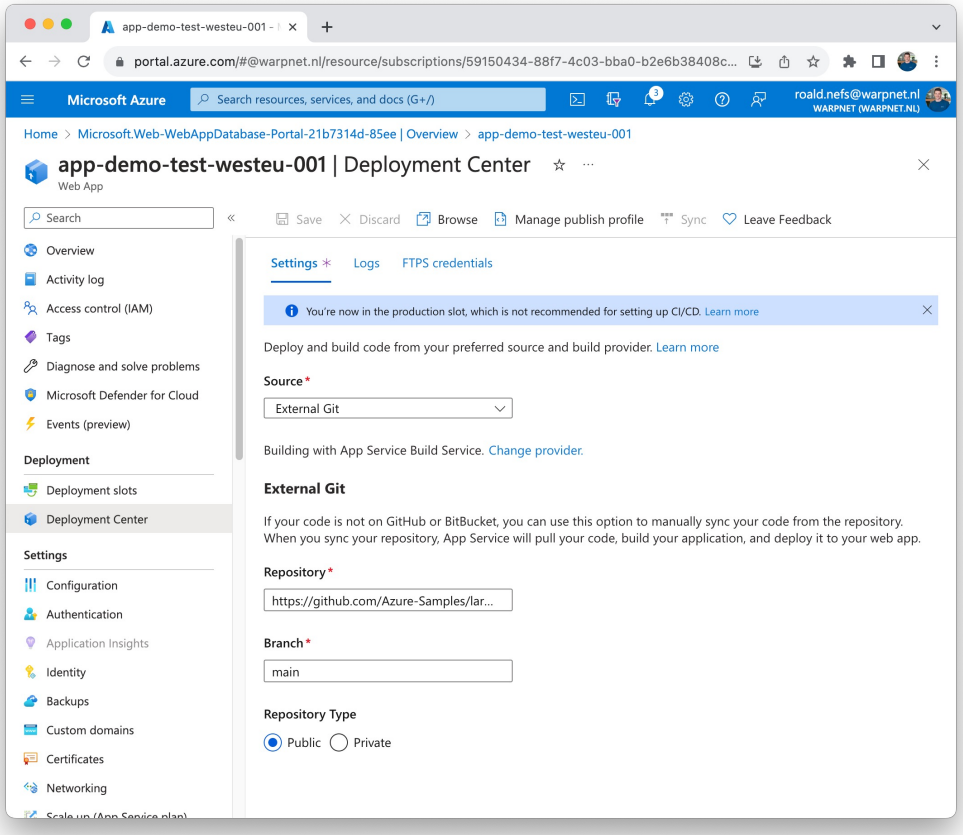
`MYSQL_ATTR_SSL_CA`



Deploy code

Let’s deploy the following Laravel project from a external repository:

`github.com/Azure-Samples/laravel-tasks`



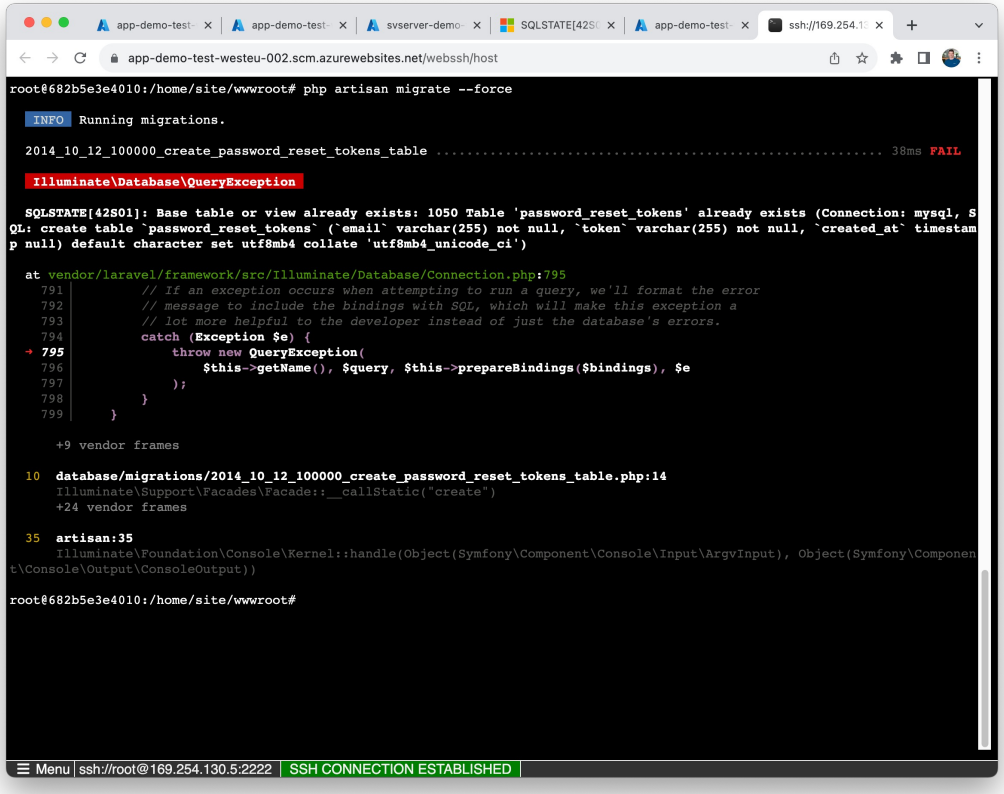
Takes a couple of minutes and ran into failing deployments 5/5 times…

Let’s try again

using PHP 8.2

Generate database schema

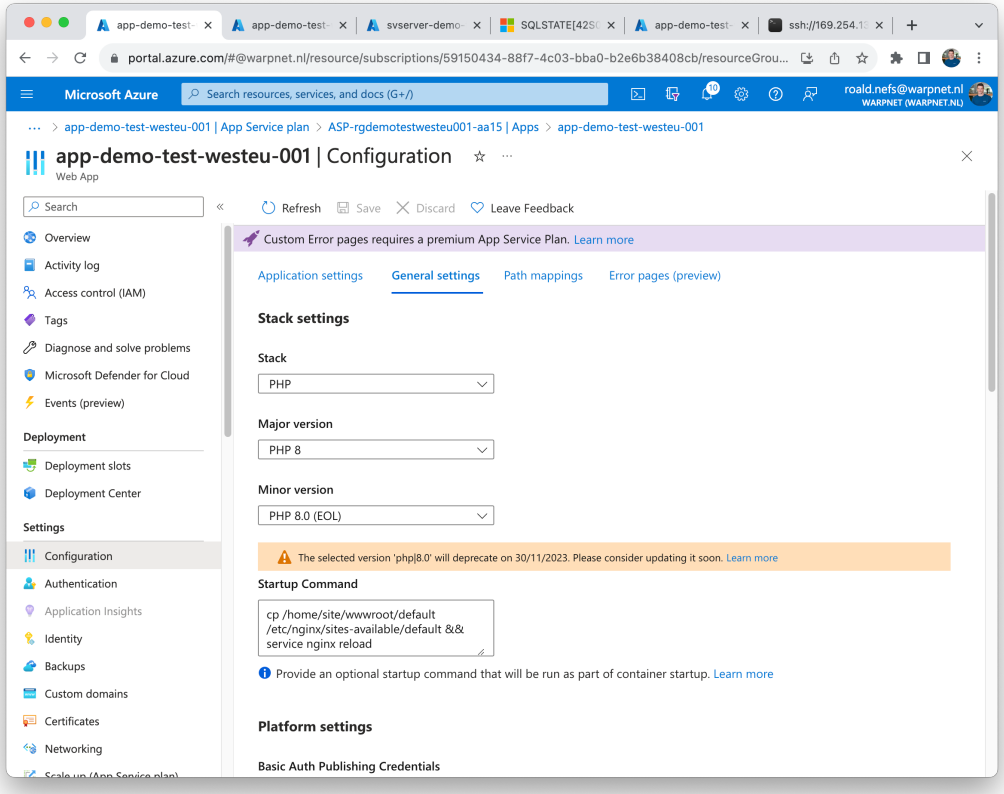
Login to the container using SSH to run the required database migrations.



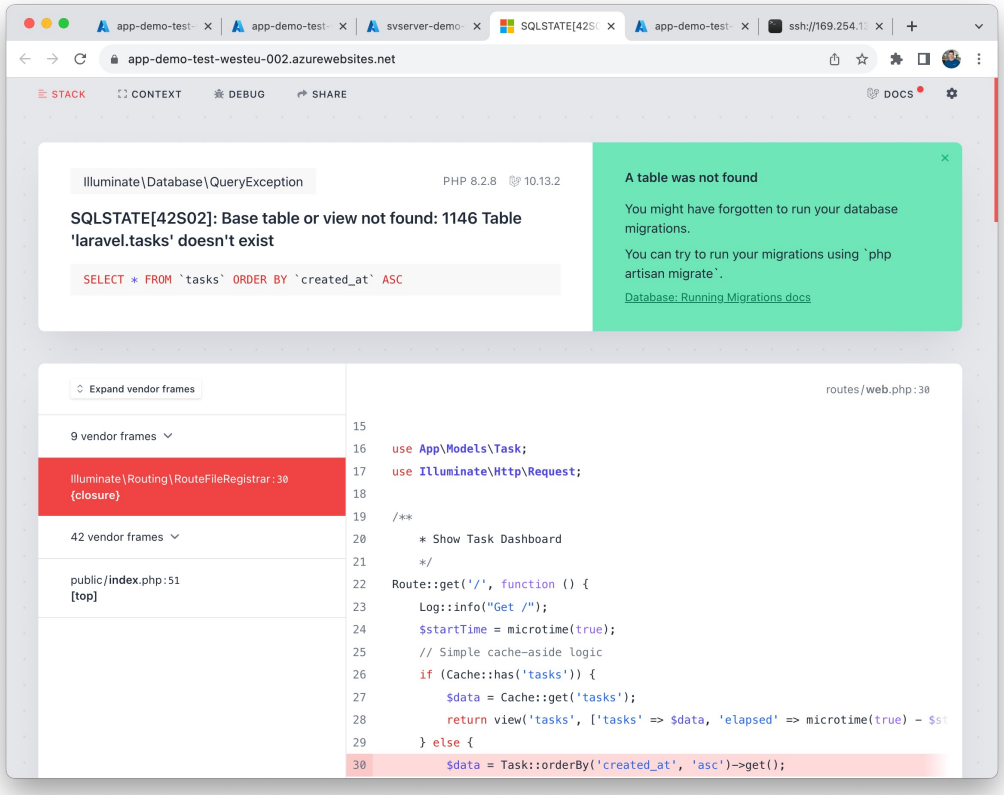
Change the site root

The Laravel application lifecycle begins in the `./public` directory. The example project includes a new Nginx configuration.

Simply update the startup command.



Browse the app…



Let’s deploy Keycloak

using the CLI

Create the resource group

Start with creating a resource group to store all required resources.

```
$ az group create \
  --location westeurope \
  --name rg-keycloak-test-westeu-001
```

Create the App Service Plan

Let’s create a App Service Plan based upon Linux and a cheap plan.

```
$ az appservice plan create \
  --resource-group rg-keycloak-test-westeu-001 \
  --name asp-keycloak-test-westeu-001 \
  --is-linux \
  --location westeurope \
  --sku B1
```

Create the Web App (container)

Create the Web App with the official container.

```
$ az webapp create \
  --name app-keycloak-test-westeu-001 \
  --plan asp-keycloak-test-westeu-001 \
  --resource-group rg-keycloak-test-westeu-001 \
  --deployment-container-image-name \
    quay.io/keycloak/keycloak:22.0.1
```

Let’s try again

By updating container start time limit

Let’s update the default `WEBSITES_CONTAINER_START_TIME_LIMIT` from **230** to **1800** (max) to be able to start the Keycloak container.

Create a PostgreSQL - Flexible Server

Let’s first create a PostgreSQL server to store the Keycloak data.

```
$ az postgres flexible-server create \
--resource-group rg-keycloak-test-westeu-001 \
--name psql-keycloak-test-westeu-001 \
--location westeurope \
--sku-name Standard_B1ms --tier Burstable \
--version 15 \
--public-access None \
--admin-user tigalahe20 \
--admin-password REDACTED
```

Create a PostgreSQL database

Create a new PostgreSQL database
for keycloak to use.

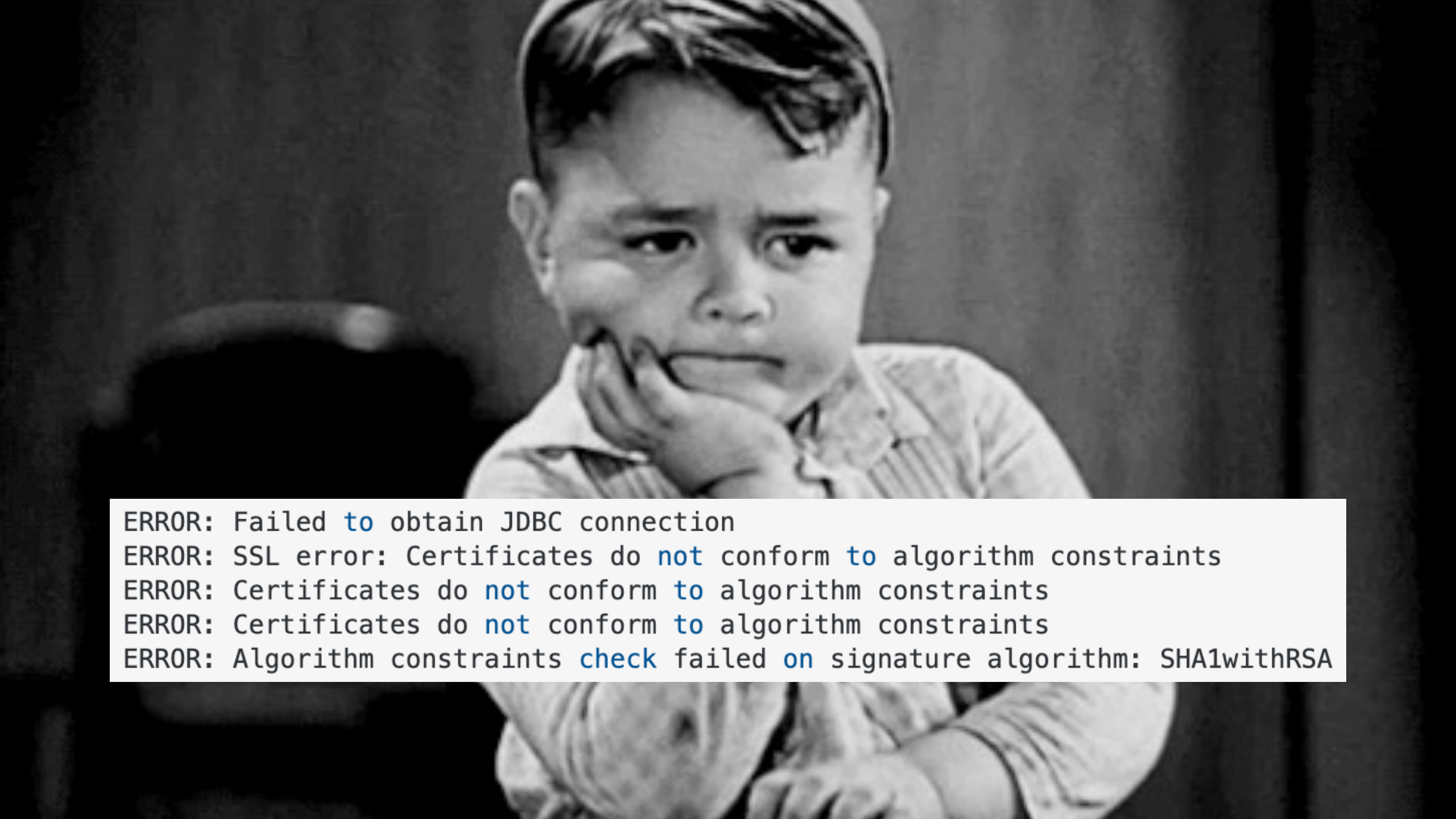
```
$ az postgres flexible-server db create \
  --resource-group rg-keycloak-test-westeu-001 \
  --server-name psql-keycloak-test-westeu-001
  --database-name keycloak
```

Update Keycloak settings

Update Keycloak settings by setting the required 12 application settings, e.g.:

```
KEYCLOAK_FRONTEND_URL
KEYCLOAK_ADMIN
KEYCLOAK_ADMIN_PASSWORD
KEYCLOAK_FRONTEND_URL
PROXY_ADDRESS_FORWARDING
DB_VENDOR
DB_ADDR
DB_USER
KC_PROXY
KC_HOSTNAME
```

```
$ az webapp config appsettings \
  set --resource-group rg-keycloak-test-westeu-001 \
  --name app-keycloak-test-westeu-001 \
  --settings @settings.json
```


ERROR: Failed to obtain JDBC connection
ERROR: SSL error: Certificates do not conform to algorithm constraints
ERROR: Certificates do not conform to algorithm constraints
ERROR: Certificates do not conform to algorithm constraints
ERROR: Algorithm constraints check failed on signature algorithm: SHA1withRSA

According to a random person on the internet

ThomasAunvik commented on Mar 9

Same issue for me regarding the certificate on Azure Postgresql.

It would seem that the Azure Database for Postgresql - ***Single Server***, is using the SHA256 DigiCertGlobalRootG2, for anyone using Single Server would not have this problem.

But for Flexible server, is still using the old SHA1 Root Certificate, causing this error.

☺ ❤️ 5

Let’s try again

using a custom Dockerfile

How to continue

Create a new Dockerfile that can be used to create a container including the needed certificate to allow the database connection.

```
FROM quay.io/keycloak/keycloak:22.0.1 as builder

# Enable health and metrics support
ENV KC_HEALTH_ENABLED=true
ENV KC_METRICS_ENABLED=true

# Configure PostgreSQL as the database vendor
ENV KC_DB=postgres

WORKDIR /opt/keycloak
RUN /opt/keycloak/bin/kc.sh build

FROM quay.io/keycloak/keycloak:latest
COPY --from=builder /opt/keycloak/ /opt/keycloak/
COPY DigiCertGlobalRootCA.crt.pem
/opt/keycloak/.postgresql/root.crt

ENV KC_DB=postgres
ENTRYPOINT ["/opt/keycloak/bin/kc.sh"]
```

Use custom Docker image

1. Upload the image to an existing Docker registry.
2. Update the container in the **Web App** configuration.

1. Create an Azure container registry.
2. Login to the new registry.
3. Push the custom Docker image.
4. Update the container in the **Web App** configuration.

KEYCLOAK

master

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

master

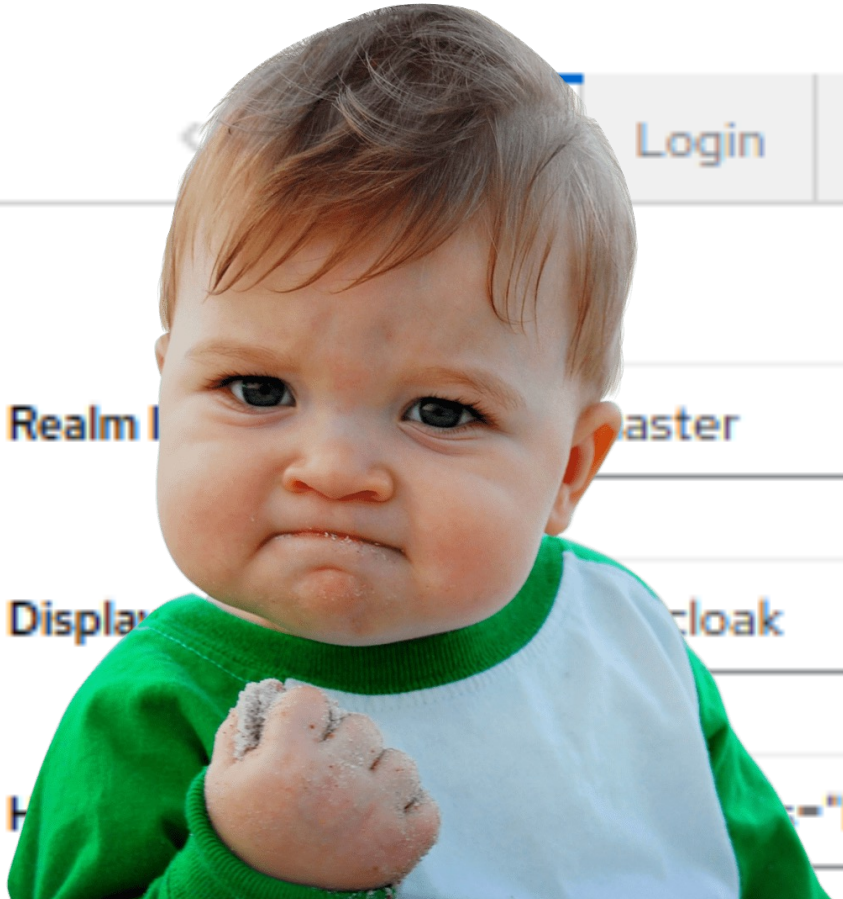
Realm settings are settings that control the options for users, application

Login | Email | Themes | Keys | Eve

Realm I aster

Displa cloak

H ="kc-logo-text">Keycloak</spa



Let’s recap