Research Web Apps on Azure

Martin Callaghan & Alex Coleman

Research Computing Group

1. Introductions
2. Communication and Research Outputs
3. Different types of App
4. Our pipeline
5. The finished app

Introductions

Who we are

- Martin Callaghan
- Alex Coleman

`https://github.com/ARCTraining/it_conf_2021_web_apps`

What we do

- Run the University High Performance Computing Service
- Support computational and data-focussed research
  - Writing grant proposals
  - Collaborations with research groups
  - Gateway to external collaborations
  - Researcher training
  - Computational, training and teaching consultancy
  - Code optimisation
  - Support and guidance on Cloud
  - GPU-accelerated Deep learning and AI support

Questions?

If you have any questions, pop then in the chat as we go and we'll deal with them at the right time.

Communication and Research Outputs

What is scholarly communication?

- Key part of every research project is disseminating outputs
- Or gathering data from study participants
- A Web App can be a part of this
- Most research data has no need to be kept *secure*

What's a Web App?

- For us, a tool that allows a researcher to share the outputs of a project
- Allows other researchers to interact with models or data
- Facilitates open and reproducible research
- An adjunct to a paper or poster

The tools

- Docker
- Shiny and R
- Streamlit (or Flask) and Python

Our pipeline

- Build a container of the app and it's framework
- Test locally
- Configure Azure Container Registry (ACR) and Azure App Service (AAS)
- Deploy local container to ACR
- Deploy from ACR to AAS

Different types of App

Completely self-contained

- Data is read-only in a flat file inside the container

Apps with a sidecar

- Database in a container ‘next to’ the App container
- Data usually read-only

Apps that use a native Cloud database

- Cosmos DB
- Azure SQL server
- Apps that need to read and write data

Our pipeline

Dockerise the App

- Create a Dockerfile
- Build the container locally
- Run it and check it works
- Over to Alex for a demo of this stage

Explaining the Dockerfile

Think of this as *infrastructure as code*

It describes the actions to build a Docker image from a pre-existing template

```
FROM rocker/shiny-verse

EXPOSE 3838

COPY /SpheroidAnalyseR/ /srv/shiny-server/

RUN install2.r --error \
    ggthemes \
    gridExtra \
    readxl \
```

Building the container locally

```
docker build -t myimage .
```

Run it locally to test

We create a **container** from the **image**

```
docker run --name mycontainer \ --rm -d -p 3838:3838
myimage
```

Check it works and then stop it.

```
docker stop mycontainer
```

Retag the container

```
docker tag myimage myreg.azurecr.io/myimage
```

Azure CLI

- Create Azure Container Registry (ACR)
- Create Azure App Service
- Deploy to ACR
- Create the App from the ACR image
- Over to Alex for a demo

The finished app

Here it is!

Link to this URL (it's in the chat):

Future work

- AAS has the concept of production and development apps
- Full integration with a ResOps (∼DevOps) CI/CD pipeline
- Research communications built and developed alongside the data and analysis