# Deploying the ‘ONFARM Soil Data’ tool.

## Step 0 - Grab the repo

Pull or fork <https://github.com/RKZill/R-app>

## Step 1 – Allocate Resources

Use **azure cloud shell** to make a resource group and container registry (or use existing if exists)  
**NOTE:** change ‘myapp-rg’ to be a fitting resource group name. Same with ‘mycontainerregistry’.

# Create a resource group

az group create --name myapp-rg --location eastus

# Create a container registry

az acr create --resource-group myapp-rg --name mycontainerregistry --sku Basic

## Step 2 – Create Pipeline

### Step 2.1

Sign into Azure DevOps and **create new pipeline**, on connect, **select GitHub YAML,** and select the repo pulled earlier.

### Step 2.2

In the **configure tab**, choose **Docker** (build and push an image to Azure container registry).

Step 2.3  
Select an **Azure subscription**, then select the **container registry** and create **an image name**. (Image name does not matter and is not viewable anywhere by users)

### Step 2.4

Add the “**latest**” tag to the pipelines.yml file, near the bottom.

A screenshot of a computer

Description automatically generated

### Step 2.5

Click **Validate**, then **Save and Run**, add a commit message, and click **Save.**

## Step 3 – Web App Basics

### Step 3.1

Find the container registry, go into **settings** **> Access Keys**, and enable “**Admin User**”

### Step 3.2

Sign into **Azure Portal** and under **App Services ,** click **Web App** and select **Create a New Web App**

Step 3.3  
In the create web app form, select a subscription, resource group, and name.

Step 3.4  
On **Publish** section, select “**Docker Container**” with **Linux** as OS option. Click **Next.**

## Step 4 – Web App Docker

In the docker tab, under **image source** choose **Azure Container Registry.** Access type should be **public,** and **image-and-tag** should be the **image name created earlier in step 2.3, along with the latest tag.**  Click **Review + Create**.

**Note:** The ‘latest’ tag will grab the latest push to a repo and automatically update and deploy the app.

## Step 5 – Finalization

On the web app, left side navigation menu there is a **configuration** option. Click it and click the **+ New Application Setting** button. The new setting should be called WEBSITES\_PORT with a value 3838 (the port docker uses). Back to main menu click **Restart**.

## Step 6 – Deploy again

Run the deployment once more.

## Step 7 – Using the App

The app can now be accessed through the website <APPNAME>.azurewebsites.net , or with an iframe.

<iframe src="https://ryan-r-app.azurewebsites.net/" height="1450px"></iframe>

**NOTE:** This iframe example uses test URL. This will be changed in the final deployment.  
**NOTE2:** Height is set to 1450px to work well with OSCIA main site, alter as needed.