

Demo – Azure Web App for Containers

All commands should work from within the

Prepare your environment

Setting up Azure Redis Cache can take a long time, so one better does this before doing a live demo. Even though Redis will show up quite quickly in the portal, it will take a while to be fully provisioned.

```
az group create -n azurevote -l northeurope
az redis create -g azurevote -n azurevote-redis -l northeurope --sku Basic --vm-size C0 --enable-non-ssl-port
```

Set up an App Service Plan and a Web App

```
# Create the plan
az appservice plan create -g azurevote -n azurevote-plan -l northeurope --is-linux

# Create the web app
az webapp create -g azurevote -p azurevote-plan -n azurevote-app -i xstof/azure-vote
```

Now we have provisioned a Web App for Containers which is based on the [xstof/azure-vote](https://github.com/Azure-Samples/azure-voting-app-redis/blob/master/azure-vote/Dockerfile-for-app-service) docker image. Alternatively, such image can be build yourself from the official repo using this Docker file: <https://github.com/Azure-Samples/azure-voting-app-redis/blob/master/azure-vote/Dockerfile-for-app-service>

Configure the Web App

Now that our container is configured, we'll need to provide it with some environment variables to that our frontend knows where to contact the Redis backend. This is done through a simple pair of "App Settings" in App Service. For this, please find your fully qualified redis name and primary key.

```
az webapp config appsettings set -g azurevote -n azurevote-app --settings REDIS=azurevote-redis.redis.cache.windows.net REDIS_PWD=replaceme
```

Now show in the Web App that these app settings are configured and replace the "replaceme" with the real primary key for Redis.

Recycle the Web App and show the end result by browsing the azure-vote site.