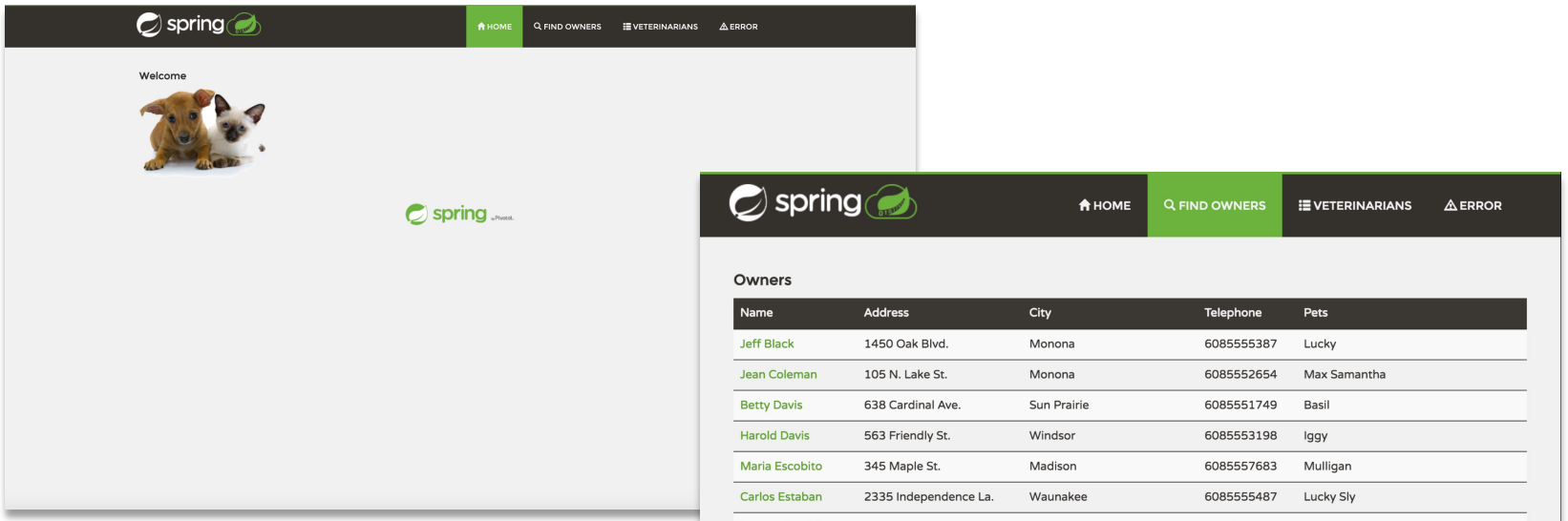
Lesson 2 of 4

Quickly deploy “PetClinic” to the cloud

The Existing Application



(You don’t actually need to change its code, we’re using what’s familiar.)

Deploy into Azure Spring Cloud

- Clone Github repository branch locally
- Build using maven build tool wrapper
- Quick set up with Azure CLI
- Create the application instance in Azure Spring Cloud
- Deploy Spring PetClinic JAR to Azure Spring Cloud via CLI

Clone the "initial-app" GitHub Branch to your workspace

git clone -b initial-app https://github.com/admiralappsec/workshop-testing.git

Build via Maven wrapper

cd workshop-testing

mvnw clean package -DskipTests -Denv=cloud

****Note –**
Make sure mvnw is executable, if an error pops up around execution, please use this command to make mvnw executable (while in the same directory as mvnw):
`*chmod +x mvnw*`

Quick Set-Up with Azure CLI

```
az login --service-principal -u "${AZURE_APPLICATION_ID}" -p
  "${AZURE_CLIENT_SECRET}" --tenant "${AZURE_TENANT_ID}";

az account set --subscription "${AZURE_SUBSCRIPTION_ID}";

az configure --defaults group="${AZURE_RESOURCE_GROUP_NAME}" spring-
  cloud="${AZURE_SP_SERVICE_NAME}";
```

Create the Application in Azure Spring Cloud

```
az spring-cloud app create --name ${APP_NAME} --instance-count 1 --is-public true --memory 2 --jvm-options='-Xms2048m -Xmx2048m' --enable-persistent-storage true
```

Deploy Spring PetClinic JAR to Azure Spring Cloud

```
az spring-cloud app deploy --name ${APP_NAME} \
        --jar-path ${APP_JAR}
```

****note – Make sure you are in the same directory as the JAR file when executing the above commands. The maven build will automatically put the deployable JAR inside the following directory:** `*workshop-testing/target*`

Once deployed, you can verify the successful deployment by entering the following command into your command line and taking the URL and entering this value into your browser:

```
az spring-cloud app show --name ${APP_NAME} | grep url
```