100 XP

Exercise - Deploy a monolithic application on App Service

10 minutes

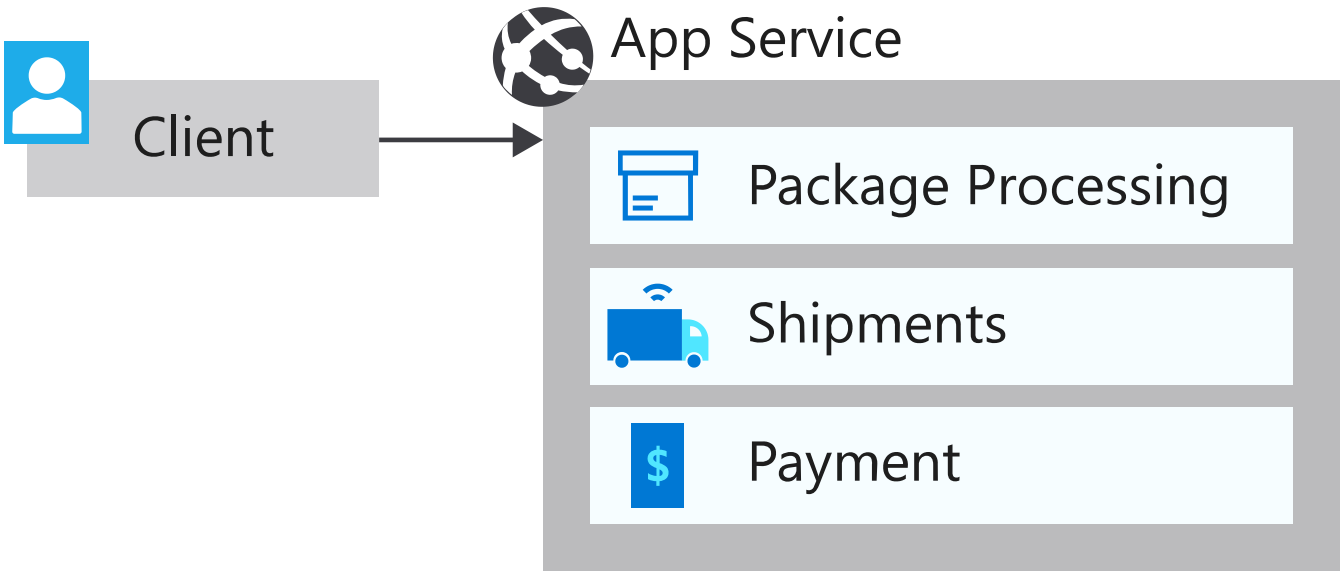
Sandbox activated! Time remaining: **1 hr 59 min**

You have used 1 of 10 sandboxes for today. More sandboxes will be available tomorrow.

We've talked a bit about monolithic and microservices architectures. Now let's deploy a monolithic application and dive further into how we can evolve it into a microservices architecture.

Fabrikam has currently added the drone service into their existing application.

In this exercise, we'll deploy a monolithic application on Azure App Service and enable Azure Monitor to get visibility into telemetry and application performance.



Deploy a monolithic application on Azure App Service

Let's start by deploying the application. First, we need to create the Azure resources to host the application.

1. Run this command to deploy the resources that are needed for this application. This deployment takes a few minutes.

Azure CLI | Copy

```
az deployment group create \
    --resource-group learn-bdae2072-cd38-4abb-9e0a-04f5d380f25a \
    --template-uri https://raw.githubusercontent.com/MicrosoftDocs/mslearn-
microservices-architecture/master/deployment/azuredeploy.json
```

2. Now that we have the resources created, let's deploy the application. First, run this command to pull down the source code from the sample repository.

Azure CLI | Copy

```
git clone https://github.com/MicrosoftDocs/mslearn-microservices-
architecture.git ~/mslearn-microservices-architecture
cd ~/mslearn-microservices-architecture/src/before
```

3. Run this command to zip up the application code, which we use to deploy to the app service.

Bash | Copy

```
zip -r DroneDelivery-before.zip .
```

4. Run this command to set a variable with the name of your app service.

Bash | Copy

```
APPSERVICENAME="$(az webapp list \
                    --resource-group learn-bdae2072-cd38-4abb-9e0a-
04f5d380f25a \
                    --query '[].name' \
                    --output tsv)"
```

5. Run this command to configure the app service to run a build as part of the deployment.

Azure CLI | Copy

```
az webapp config appsettings set \
    --resource-group learn-bdae2072-cd38-4abb-9e0a-04f5d380f25a \
    --name $APPSERVICENAME \
    --settings SCM_DO_BUILD_DURING_DEPLOYMENT=true
```

6. And now, run the following command to deploy the application to App Service. This deployment takes a few minutes to finish.

Azure CLI | Copy

```
az webapp deployment source config-zip \
    --resource-group learn-bdae2072-cd38-4abb-9e0a-04f5d380f25a \
    --name $APPSERVICENAME \
    --src DroneDelivery-before.zip
```

7. After the deployment finishes, confirm that the deployment was successful by visiting the website of your app service. Run this command to get the URL, and select it to open the page.

Bash Copy

```
echo https://$(az webapp config hostname list \
                --resource-group learn-bdae2072-cd38-4abb-9e0a-04f5d380f25a \
                --webapp-name $APPSERVICENAME \
                --query [].name \
                --output tsv)
```

Perform a load test against the application

Now let's test out the performance of the application in its monolithic architecture.

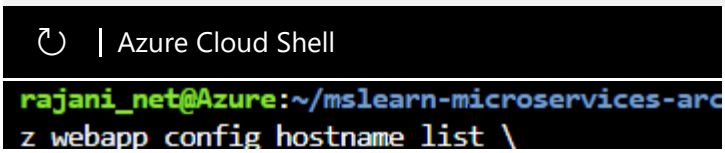
1. On the home page for your deployed application, select **Send Requests**. This action simulates the submission of 100 requests through the application.

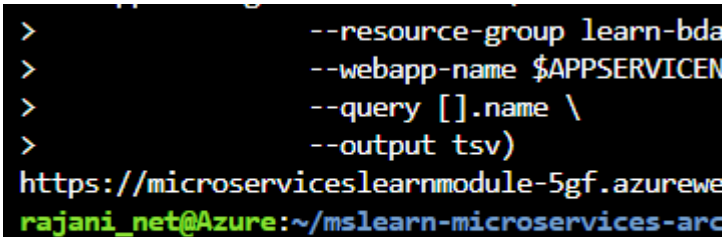
2. For the first request, you see a result of around 8 to 12 seconds to process 100 messages. If you refresh the page and resubmit if prompted, this number might drop by about half. It still takes around five seconds per request.

Fabrikam suspects that the monolithic architecture is preventing heavily used services from scaling, which leads to the poor performance seen here.

Next unit: Performance constraints of a monolithic application

Continue >




> --resource-group learn-bda
> --webapp-name $APPSERVICEN
> --query [].name \
> --output tsv)
https://microserviceslearnmodule-5gf.azurewe
rajani_net@Azure:~/mslearn-microservices-arc