

Troubleshoot service deployment

5 minutes

There are a lot of elements to a real-time service deployment, including the trained model, the runtime environment configuration, the scoring script, the container image, and the container host. Troubleshooting a failed deployment or an error when consuming a deployed service can be complex.

Check the service state

As an initial troubleshooting step, you can check the status of a service by examining its **state**:

Python

Copy

```
from azureml.core.webservice import AksWebservice

# Get the deployed service
service = AksWebservice(name='classifier-service', workspace=ws)

# Check its state
print(service.state)
```

ⓘ Note

To view the **state** of a service, you must use the compute-specific service type (for example **AksWebservice**) and not a generic **WebService** object.

For an operational service, the state should be *Healthy*.

Review service logs

If a service is not healthy, or you are experiencing errors when using it, you can review its logs:

Python


Copy

```
print(service.get_logs())
```


The logs include detailed information about the provisioning of the service, and the requests it has processed. They can often provide an insight into the cause of unexpected errors.

Deploy to a local container


Deployment and runtime errors can be easier to diagnose by deploying the service as a container in a local Docker instance, like this:

Python	 Copy
<pre>from azureml.core.webservice import LocalWebservice deployment_config = LocalWebservice.deploy_configuration(port=8890) service = Model.deploy(ws, 'test-svc', [model], inference_config, deployment_config)</pre>	

You can then test the locally deployed service using the SDK:

Python	 Copy
<pre>print(service.run(input_data = json_data))</pre>	

You can then troubleshoot runtime issues by making changes to the scoring file that is referenced in the inference configuration, and reloading the service without redeploying it (something you can only do with a local service):

Python	 Copy
<pre>service.reload() print(service.run(input_data = json_data))</pre>	

Next unit: Exercise - Deploy a model as a real-time service

[Continue >](#)