(../..)

This lab demonstrates how BOSH ***monitors*** its services and ***recovers*** them if they fail.

Setup

This lab assumes you have completed the Extract a Cloud Configuration and Scale a Release (../create-release) lab.

Confirm BOSH is set up correctly.

```
bosh target 192.168.50.4
bosh deployment articulate.yml
```

Check out `v2` of the `apps-platform-acceleration-bosh-code` repository.

Caution: the following command will erase any local changes you have made. If you would like to back up your work, commit it to a local branch.

```
cd apps-platform-acceleration-bosh-code
git checkout --force v2
```

Scale articulate back to a single instance. (If you completed the previous lab, you should have two instances running). Edit the `articulate.yml` manifest to deploy a single instance of the `articulate` instance group and so that it only has one ip address (`10.244.9.5`).

Once you have saved your changes, redeploy the application.

```
bosh -n deploy
```

When the deployment has finished, verify that you only have the single VM running, and make sure you get a successful response when `curl` ing its `/info` endpoint.

```
bosh vms
curl -H 'Accept: application/json' http://10.244.9.5:9000/info
```

Monit

When BOSH deploys a job, it uses a control script to start a process and `monit` to monitor that process.

Monit is most commonly configured to watch the PID of the running process.

Examine the monit configuration for your articulate job. From the root of `apps-platform-acceleration-bosh-code`:

```
cd articulate-release/jobs/articulate
cat monit
```

If the PID goes away (if the service dies), monit will attempt to restart it by running the command in the `start` stanza of the configuration.

Unresponsive Service

Articulate's monit ensures that articulate continues to respond to HTTP requests to the port on which the app has been configured to run.

If the app crashes or becomes unresponsive, monit will attempt to restart the service.

`ssh` into your articulate VM and watch the monit summary.

```
bosh ssh articulate 0
sudo -i
watch monit summary
```

Keep this terminal window open. `watch` will re-run `monit summary` every 2 seconds and print the results.

Open a new terminal window.

Kill the Java thread the articulate application is running in. Articulate has an endpoint that will do this. Send a curl request to that endpoint and watch monit detect the failure and bring the app back up.

```
curl http://10.244.9.5:9000/basics?doit=true
```

In the `watch` terminal window, you will see monit report that the service is `not monitored`, that it `Does not exist`, or both. The service will then return to its normal status of `running`.

Kill the thread again. This time, when monit reports that the service is no longer running, confirm the report by sending a request to its health endpoint.

```
curl http://10.244.9.5:9000/basics?doit=true
curl -H 'Accept: application/json' -i http://10.244.9.5:9000/health
```

Keep hitting the health endpoint until you receive a 200 response.

Monit has brought the job back up.

Dead Process

In the previous example, the web server died, but the Java process it was running in remained active.

What happens if the Java process itself dies? Let's find out.

Keep the terminal with the monit summary visible. In another window initiate an ssh session with the articulate VM.

```
bosh ssh articulate 0
```

Find the Java process running the articulate app and kill it. While it is down, watch monit detect the failure and bring the application back up.

NOTE: Replace the `$PID` variable in the `kill` command with the one you find.

```
sudo -i
ps aux | grep java
kill -9 $PID
```

In a third terminal window, `curl` articulate's health endpoint to verify it goes down and is brought back.

```
curl -H 'Accept: application/json' -i http://10.244.9.5:9000/health
```

Finally, confirm the process is down and confirm it has come back up using the `ps aux | grep java` command.

Close all terminal windows you opened for this part of the lab.

BOSH Health Monitor

In addition to monitoring processes running inside a VM using `monit`, BOSH monitors the VMs themselves using its built-in health monitor.

Automatic VM Resurrection

This time, you will watch the results of the `bosh vms` command from your *local* machine.

In a new terminal window, make sure you have the `watch` command installed. Install it via Homebrew (http://brew.sh/) if you do not.

```
[ ! `which watch` ] && brew install watch
```

Enter the following command in your terminal:

```
watch bosh vms
```

Garden Linux exposes an API with information about running containers. **From a new terminal window**, curl the `/containers` endpoint for the service running on port `7777` of your virtual machine.

```
curl –H "Accept: application/json" http://192.168.50.4:7777/containers
```

You should see something like the following:

```
{
  "Handles": [
    "[GUID–1]"
  ]
}
```

Copy the GUID of the VM containing your articulate application ([GUID-1]).

NOTE: If you have anything else deployed to your BOSH-lite instance, or if you have deployed more than one instance of the `articulate` app, you will see more than one GUID. Make sure you select the GUID for your articulate application the following steps.

Destroy the articulate VM by issuing a `DELETE` request to the garden service.

```
curl -X DELETE -i http://192.168.50.4:7777/containers/[GUID-1]
```

You should get back a response with an HTTP status of 200 and an empty JSON octet in the body.

Return to the terminal window that is watching the `bosh vms` command. What do you notice?

You should see the state of one of your articulate VMs change from `running` to `unresponsive agent` and eventually back to `running`. This process can take some time.

Confirm that the articulate service goes down and comes back up by curling the health endpoint after killing the VM.

```
curl -X DELETE -i http://192.168.50.4:7777/containers/[GUID-1]
watch "curl -H 'Accept: application/json' -i http://10.244.9.5:9000/he
alth"
```

NOTE: Automatic VM resurrection is turned `on` by default, but can be turned off in the deployment manifest or by running `$ bosh vm resurrection off`.

BOSH Cloud Check

BOSH Cloud Check can also bring back unresponsive or unhealthy VMs.

Open a terminal window and watch the status of your bosh vms:

```
watch bosh vms
```

In another window, turn off automatic VM resurrection and destroy the articulate container.

```
bosh vm resurrection off
curl -H "Accept: application/json" http://192.168.50.4:7777/containers
```

```
{
  "Handles": [
    "[GUID-1]"
  ]
}
```

```
curl -X DELETE -i http://192.168.50.4:7777/containers/[GUID-1]
```

Wait until you see that the status of the articulate VM has changed to `unresponsive agent`. Then run the BOSH cloud check.

```
bosh cck
```

Choose option 3 to "*Recreate VM for 'articulate/0 (some-guid)' and wait for processes to start*".

Watch the other window.

You should see the state of the articulate VM go from `unresponsive agent` to `failing` and finally to `running`.

`curl` the articulate app throughout this process to watch it come back up.

```
curl -X DELETE -i http://192.168.50.4:7777/containers/[GUID-1]
watch "curl -H 'Accept: application/json' -i http://10.244.9.5:9000/he
alth"
```

```
bosh cck
```

Return your BOSH director to its original state by turning automatic VM resurrection back on.

```
bosh vm resurrection on
```

course version: 1.5.3