Consul Template

In this lesson, we'll discuss Consul Templates.

Introduction

For each microservice, Apache httpd must have an entry in its configuration file. For this, Consul Template (https://github.com/hashicorp/consul-template) can be used.

In the example, the `00-default.ctmpl` file is used as a template to create the Apache httpd configuration. It is written in the Consul Templating Language (https://github.com/hashicorp/consul-template#templating-language). For each microservice, it creates an entry that distributes the load between the instances and redirects external requests to these instances.

The template

The essential part is:

```
{{range services}}

<Proxy balancer://{{.Name}}>
{{range service .Name}}  BalancerMember http://{{.Address}}:{{.P
ort}}
{{end}}
</Proxy>
  ProxyPass        /{{.Name}} balancer://{{.Name}}
  ProxyPassReverse /{{.Name}} balancer://{{.Name}}

{{end}}
```

- The Consul Template API functions are enclosed in `{{` and `}}`.
- For each service a configuration is generated with `{{range service}}`.
- It contains a reverse proxy that is configured with the `<Proxy>` element and the `Name` of the microservice.
- This element contains the microservice instances as `BalancerMember` with the `Address` and `Port` of each instance for distributing the load between the microservice instances.
- The end is formed by `ProxyPass` and `ProxyPassReverse` which, likewise, belong to the reverse proxy.

Starting the Consul Template

The Consul Template is started with the following section from the `Dockerfile` in the directory `docker/apache`:

```
CMD /usr/bin/consul-template -log-level info -consul consul:8500
 \\
  -template "/etc/apache2/sites-enabled/000-default.ctmpl:/etc/
  -template "/etc/apache2/sites-enabled/000-default.ctmpl:/etc/a
pache2/sites-enabled/000-default.conf:apache2ctl -k graceful"
```

Consul Template executes the command `apache2ctl -k graceful` if there are new services or if services have been removed and the configuration has therefore been changed. This causes Apache httpd to read the updated configuration and restart. However, open connections are not closed, they remain open until communication is terminated. If no Apache httpd is running yet, one will be started. Thus, Consul Template takes control of Apache httpd and ensures that one instance of Apache httpd is always running in the Docker container.

To do this, Consul Template must run in the same Docker container as Apache httpd. This contradicts the Docker philosophy that only one process should run in one container. However, this cannot be avoided in the concrete example because these two processes are so closely related.

Conclusion

Consul Template can ensure that a microservice can be reached from outside as soon as it has registered with Consul. To do this, the Apache httpd server does not need to know anything about the service discovery or Consul. It receives the information in its configuration and restarts.

In the next lesson, we'll study Consul and Spring Boot.

← Back

Routing: Apache httpd

Next →

Consul and Spring Boot

☑ Mark as Completed

Report an Issue