

Service discovery and KV store using consul and friends

In the below Assignment you will get to know Hashicorp consul which is a powerful tool used for service discovery ,Key value store and many more features:

<https://www.hashicorp.com/products/consul>

You will create a service that interacts with Consul.

CI CD jobs that packages and deploys your service in the form of Docker image and container.

And finally your service will have to work automatically with at most 3 clicks.

1st click for initializing your environment

2nd click for creating CI CD for your interacting service

3rd click for querying your service

Assignment tasks breakdown:

Stage 1 - Create A service

Create a service that interacts with Hashicorp Consul Key value store:

The service should have 2 functionalities:

- **Get consul key value** - retrieves a value from Consul KV store given a key.
Input (Key) / Output (Value)
- **Set consul key value** - Either updates an existing consul key value, or creates a new set of key value.
Input (Key,Value) / Output (Success/Fail)
- Your code should have the ability to be packaged as a **docker image** ,so make sure to prepare all necessary files for that matter in the same repo.
- Upload your service code into a public repo where it can be cloned and used.

The service can be written in any selected language of your preference.

Stage 2 - Create an environment startup script

Create a Script in any selected language of preference That will create our working environment
The script should:

- **Start a Hashicorp Consul server** container
- **Start a Hashicorp Consul client** container
- **Start a Jenkins server** container
- **2 CI CD jenkins jobs inside the Jenkins server**(will be detailed in the next stage)

Stage 3 - Create Continuous Integration (CI) and Continuous Deployment (CD)

The Jenkins You've started in the last stage should be populated with 2 Jenkins jobs:

- **CI jenkins Job** - Will *Clone* your public repository that contains your Service for stage 1, *create a docker image* out of it and save it on the local host for future use.
Input (repo URL), output (docker image)
 - **CD jenkins job** - will start the saved docker image created in the CI job, and will result with a running docker container on the localhost
Input (local docker image name) Output (Local running docker container)
- The above 2 jenkins jobs should exist automatically when you first start your jenkins server.

Stage 4 - Register your service to consul

After you have a running docker container of your service,
and you have a running consul server and client,
you will need to register your service with the consul client making it accessible as a consul service.



If you succeeded then you should be able to view your service in consul from the command line:

```
dig @127.0.0.1 -p 8600 homework.service.consul
;; ANSWER SECTION:
homework.service.consul. 0      IN      A       172.17.0.3
```

Or from the Consul UI <http://localhost:8500/>:

Services 2 total

service:name tag:name status:critical search-term

Service	Type	Health Checks ⓘ	Tags
consul		✓ 1	
counting		✓ 1	go

© 2019 HashiCorp Consul 1.5.2 Documentation

Stage 5 - Service functionality verification

Try to query your service running in docker on your local host in the following manner and make sure you get the expected result:

```
\Curl @127.0.0.1 -p 8600 homework.service.consul/get value?first-key'  
5  
'Curl @127.0.0.1 -p 8600 homework.service.consul/set value?first-key=first-value'  
Success
```

*** The assignment will be tested in the following manner:**

- You will provide us with your project that will include:
 - Link to a public repository you created and includes:
 - Executable startup script that creates the above environment on our localhost
 - Instructions on how to interact with the running Docker container and how to create a Get request to your service, and set request
 - We will need to see in the local consul cluster that was created the changes made by your service after we interacted with it.
 - We will verify your code and the outputs

Thank you very much for your patience and cooperation !

Best of luck !

Fyber Devops Team.

