Consul Tutorial

# Installation on Ubuntu 15.10:

* 1. Make sure *Screen* and *Unzip* modules are installed

$ sudo apt-get update

$ sudo apt-get install unzip screen

* 1. Start *Screen*

$ sudo screen -> press enter on welcome screen

* 1. In *screen* console

$ cd /usr/local/bin

* 1. Download the latest version of consul.zip

$ sudo wget <https://releases.hashicorp.com/consul/0.6.3/consul_0.6.3_linux_amd64.zip>

* 1. Unzip that

$ sudo unzip \*.zip

Clean-up the folder

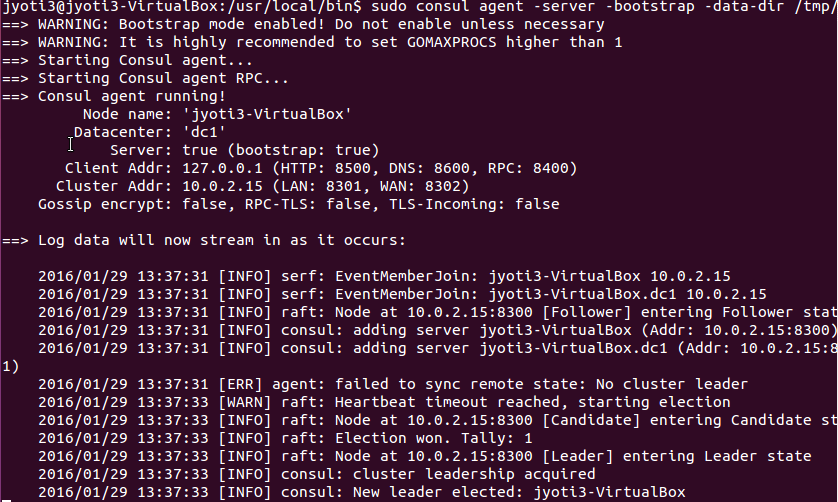
$ sudo rm \*.zip

* 1. Since it’s going to be first Consul server(in my example **172.16.121.104 is the consul bootstrap server**), it requires to be started in *bootstrap* mode

$ sudo consul agent -server -bootstrap -data-dir /tmp/consul

–server signifies the Server mode of consul agent

Output of this command will be something like following screenshot:

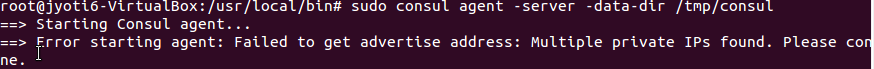


* 1. Now while running on other servers(in my example **172.16.121.100 and 172.16.121.103** **are consul node servers**), bootstrap mode need not be specified.

$ sudo consul agent -server -data-dir /tmp/consul

**Troubleshoot:**

In case, there are multiple network adapters are configured on machine and multiple private IP are failing the consul to start,

Provide advertise IP explicitly. Get the IP using

$ ifconfig

And then modify the consul command as following:

$ sudo consul agent -server -data-dir /tmp/consul -advertise 172.16.121.103

*Ref:* [*https://www.digitalocean.com/community/tutorials/an-introduction-to-using-consul-a-service-discovery-system-on-ubuntu-14-04*](https://www.digitalocean.com/community/tutorials/an-introduction-to-using-consul-a-service-discovery-system-on-ubuntu-14-04)

* 1. Ensure that node servers are joining the consul network: on Bootstar server(inmy example, 172.16.121.104), I screen console press ctrl+a+c to open another screen console.

In new console, provide the IPs of node servers joining separated by a blank space

$ Sudo console join 172.16.121.100 172.16.121.103

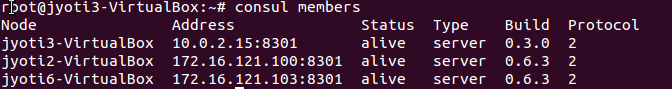
Output message should be as following:



* 1. Verify the members

$ sudo consule members

Output should be as following:



* 1. Equalize the priority

Since server1 (jyoti3-VirtualBox) was created as bootstrap it will not require other two node servers for decision making. In order to make it server with equal privileges in cluster, remove the bootstrap. On Server1, press ctrl-a-n to switch back to active consule terminal

End the consul session by tying ctrl-c

Re-start the consule session using regular start command (w/o using bootstrap attribute)

**$ sudo consul agent -server -data-dir /tmp/consul**

Again, go over to either of node servers, and type command to make this server join the cluster

OR, stay on the same server and switch over to another screen terminal (using ctrl-a-n) and re-join the other two nodes:

$ Sudo console join 172.16.121.100 172.16.121.103

* 1. **Installing a console client agent node**: build a new VM (in my case IP 172.16.121.102) and repeat the steps 1 through step 5.
  2. **Installing the UI on client node**:

**$ cd ~**

**$** sudo wget <https://releases.hashicorp.com/consul/0.6.3/0.6.3_web_ui.zip>

$ sudo unzip \*.zip

$ sudo rm \*.zip

* 1. Join the consul network as client:

$ sudo consul agent -data-dir /tmp/consul -client 172.16.121.102 -ui-dir /home/jyoti3/ -join <server nodes IP>

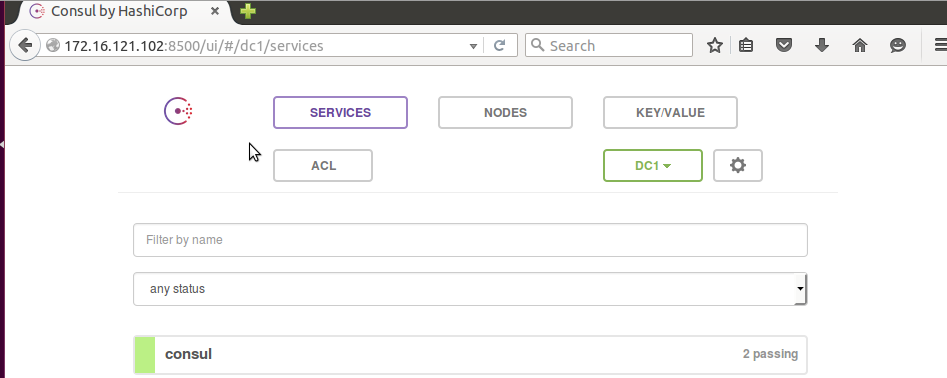
* 1. Now, log back on one of the Server nodes(in my example, jyoti3-virtualbox),

$ sudo consul members –rpc-addr=<consul client agent IP>

* 1. Access the UI – use client agent node’s IP(in my case 172.16.121.102) suffixed with :8500/ui

<http://172.16.121.102:8500/ui>

The dashboard screen should look like as following:



Troubleshoot:

In case, the consul service is not loading properly, check whether cluster members are electing Leader properly or not. You might be requiring to restart VMs and restart the cluster. Please refer this one link: <https://github.com/hashicorp/consul/issues/993>

* 1. **ADD the service**: in order to check other services and health check, let’s install a new service in the form of nginx

On the client node of cluster, in my example log in back to 172.16.121.102

1. Stop the consul agent

$ ctrl-c

1. Install nginx

$ sudo apt-get install nginx

1. Install curl

$ sudo apt-get install curl

1. Create config file to create service configurations

$ sudo mkdir ~/services

$ sudo gedit ~/services/web.json

Web.json should look like as following:

{

"service": {

"name": "web-server",

"port": 80,

"tags": ["nginx", "demonstration"],

"check": {

"script": "curl localhost:80 > /dev/null 2>&1",

"interval": "10s"

}

}

}

In the script above, Service section refers to the service itself, whereas, *check* section is making sure the curl command is executing successfully at the interval of 10 seconds.

Objective of this config is to make sure web server is running OK on port 80. If it’s not then curl command would start throwing error and that will be captured by health-check and will reflect in Consul UI.

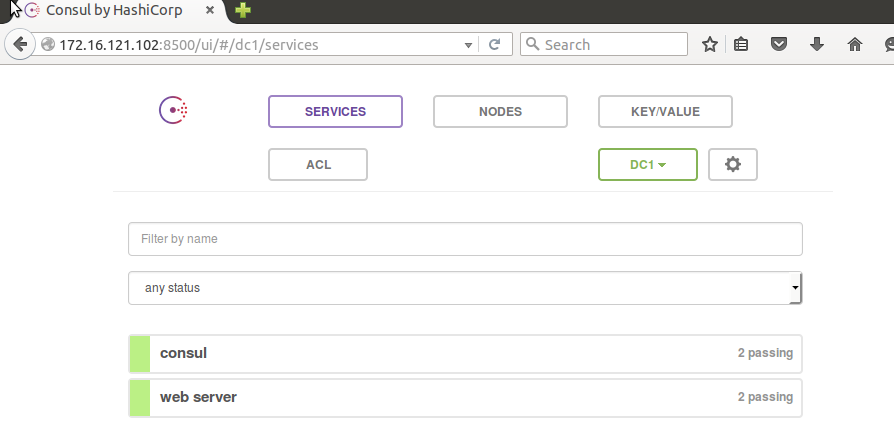
1. Re-join the cluster as client with additional specification of Service that we just added:

$ sudo consul agent -data-dir /tmp/consul -client 172.16.121.102 -ui-dir /home/jyoti3/ -join <server nodes IP> -config-dir /home/jyoti3/services

* 1. Now, go back to one of the server nodes and re-try the UI access

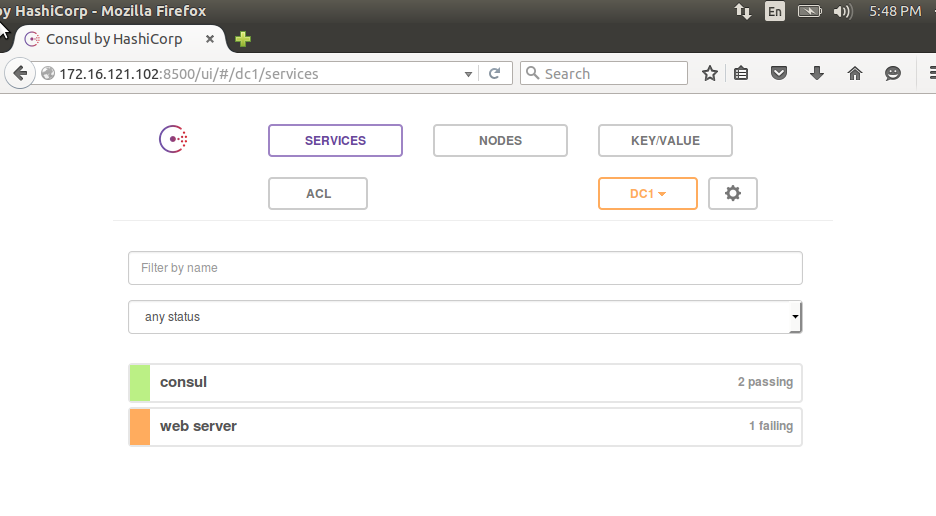
<http://172.16.121.102:8500/ui>

The output shall look like following:



Try by stopping the nginx service and re-checking the UI

$ sudo service nginx stop



* 1. Additional service configuration: in the event of having multiple services (nginx, mysql, rabbitMQ etc.) on same server, make sure the .json file with respective health check is saved in same folder ~/services. For an example of mysql check take a look at file *db.json*

{

"service": {

"name": "db-server",

"port": 3306,

"check": {

"name": "mysql\_check",

"script": "check\_mysql.sh",

"interval": "2s"

}

}

}

Make sure check\_mysql.sh is also kept at same location(~/services) or else specify the full path

Check\_mysql.sh

#!/bin/bash

nc -z localhost 3306

rt\_val=$?

#echo $rt\_val

if [ $rt\_val != 0 ]; then

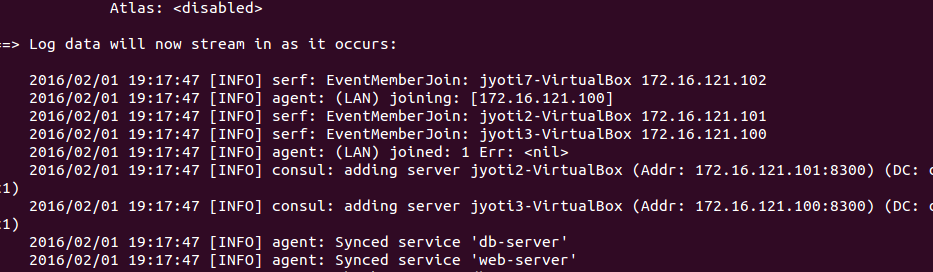
exit 3

else

exit 0

fi

Stop the client agent and restart, the output should show the new db service validation as well…



Now, go to Server node and refresh the UI

