

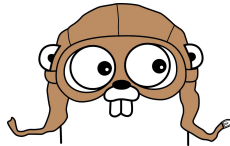
Consul

Justin Phelps @Linuturk



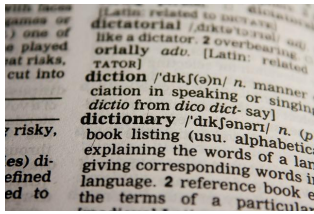
Talk about myself
Rackspace - DevOps Engineer in the DevOps group
Heat / Ansible / SaltStack
Bitcoin

What is Consul?



- From Hashicorp, makers of Vagrant
- github.com/hashicorp/consul
- Go binary and json config files
- Same binary provides server / agent (two faced)
- Built on gossip / Serf
- Service Definition / Discovery
- Health Checking
- Key/Value Store
- Multiple Datacenter

Service Discovery



- **Service Definition and Discovery:**
 - a. define/provide a service (by name), such as api or mysql, and
 - b. other clients can use Consul to discover providers of a given service.
 - c. Using either DNS or HTTP, applications can easily find the services they depend upon.
- SmartStack by AirBnB is comparable.

Health Checking



Nagios®



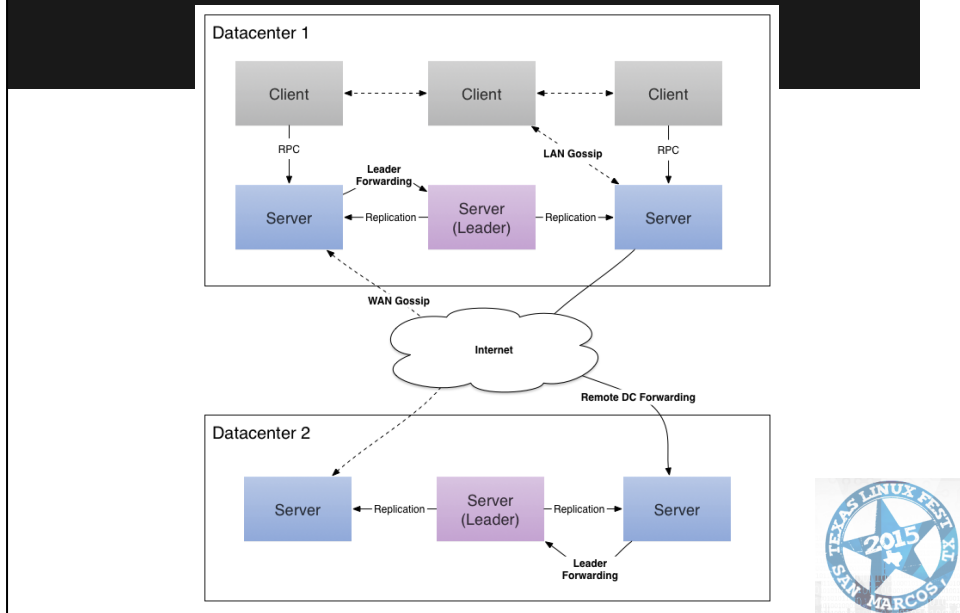
- **Health Checking:** Consul clients can provide any number of health checks,
 - a. either associated with a given service ("is the webserver returning 200 OK"),
 - b. or with the local node ("is memory utilization below 90%").
- This information can be used by an operator to monitor cluster health,
- used by the service discovery components to route traffic away from unhealthy hosts.
- Compatible with Nagios check scripts

Key/Value Store



- **Key/Value Store:** Applications can make use of Consul's hierarchical key/value store for any number of purposes, including
 - a. dynamic configuration,
 - b. feature flagging,
 - c. coordination,
 - d. leader election, and more.
- The simple HTTP API makes it easy to use.
- Comparable to etcd and Zookeeper

Multiple Datacenter



- **Multi Datacenter:** Consul supports multiple datacenters out of the box.
- This means users of Consul do not have to worry about building additional layers of abstraction to grow to multiple regions.

Follow Along

The **consul** folder at:

<https://github.com/Linuturk/saltstates>

The demo today:

<https://github.com/linuturk/consul-demo>



saltstates has init files and salt states for deploying consul

The demo today is built using docker. See README

Installing

```
001100  
011011  
000010
```

```
/usr/local/bin/consul  
/etc/consul/conf.d  
init/upstart script
```

The NGINX logo is displayed in a bold, green, sans-serif font.

1. Download and extract binary as `/usr/local/bin/consul`
2. Generate configuration files in `/etc/consul/conf.d/`
3. Create `init/upstart` script
4. Bootstrap your first server node.
5. Join a couple more server nodes. (3 to 5)
6. Join your clients.
7. Extract the web ui, ProxyPass using `nginx`.

Upstart Script

```
exec /usr/local/bin/consul agent -config-dir  
/etc/consul/conf.d/ > /var/log/consul.log
```

```
start on filesystem and static-network-up
```



All configuration files I'm showing are in the saltstates repo

Agent Configuration

```
/etc/consul/conf.d/(client|server).json
{
  "datacenter": "iad",
  "data_dir": "/opt/consul",
  "server": true,
  "ui_dir": "/usr/local/bin/consul_ui/dist",
  "start_join": ["10.208.232.22", "10.208.232.32"]
}
```



Walk through file.

Difference between agent and server is bold line.

Service Configuration

```
/etc/consul/conf.d/mysql.json
{
  "service": {
    "name": "mysql",
    "port": 3306,
    "check": {
      "name": "mysql_check",
      "script": "/usr/local/bin/check_mysql.sh",
      "interval": "1s"
    }
  }
}
```



Walk through file

Check Script

```
/usr/local/bin/check_mysql.sh
#!/bin/bash

nc -z localhost 3306
rt_val=$?

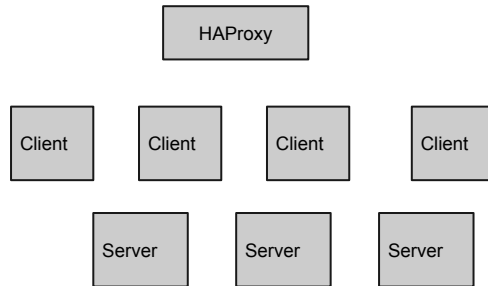
if [ $rt_val != 0 ]; then
    exit 3
else
    exit 0
fi
```



Simple check script.
Status determined by exit code.

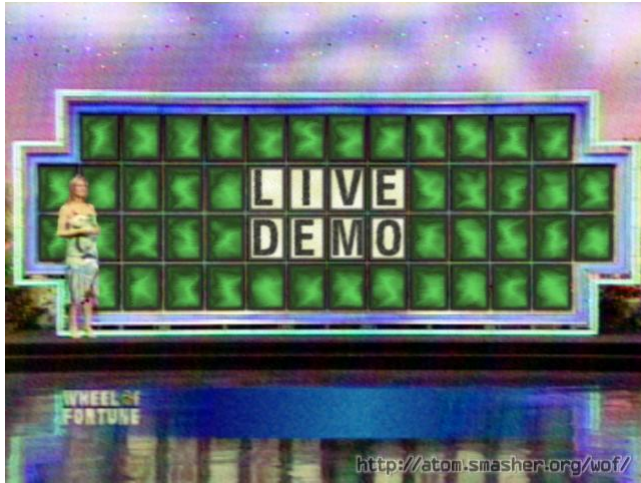
Demo

<https://github.com/linuturk/consul-demo>



Client running HA Proxy
consul-template reconfiguring haproxy based on "nginx" service
nginx running on port 80, proxypass to Consul webui on 8500
Servers are doing leader election and key value store.

Demo



1. Start cluster
2. Scale server to show joining
3. consul members
4. Web UI
5. haproxy stats
6. Scale clients and return to web UI and haproxy
7. Node leaving vs failure
8. DNS API example
9. HTTP API example

Encryption



API - TLS / SSL

Agent cross communication - The key must be 16-bytes, Base64 encoded.
consul keygen

DNS Forwarding



Consul defaults to port 8600.
Use local DNS to forward requests to localhost:8600
bind or dnsmasq

Watch out!



1. Don't lose your minimum of 3 server nodes per DC!
2. Write your own init scripts, no fancy packages.

Resources

<http://www.consul.io/>

#consul on irc.freenode.net

Slides and demo materials linked at:

www.onitato.com

