

*Transition to Microservices - Discover & Register
Applications.*

Consul in Enterprise

Nandhu Sriram
@sriramnandhu

TITANIUM SPONSORS



ADAPTIVETM
SOLUTIONS GROUP



VALOREM
CONSULTING

Platinum Sponsors



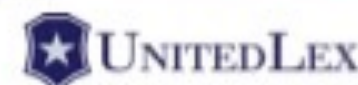
Octopus Deploy



Gold Sponsors



PocketCafe



Microservices

- ❖ What's the buzz behind?
- ❖ 12 factor application
- ❖ Benefits & drawbacks

The 12 Factors

- ❖ Codebase
- ❖ Dependencies
- ❖ Config
- ❖ Backing services
- ❖ Build, release, run
- ❖ Processes
- ❖ Port binding
- ❖ Concurrency
- ❖ Disposability
- ❖ Dev / prod parity
- ❖ Logs
- ❖ Admin processes

Consul?

- ❖ Tool for discovering, monitoring and configuring services in your infrastructure that is highly distributed and highly available.
- ❖ Consul provides Service Discovery.
- ❖ Consul provides a consistent view of your services and configuration.
- ❖ Consul can monitor and change your services topology based on Health monitoring.

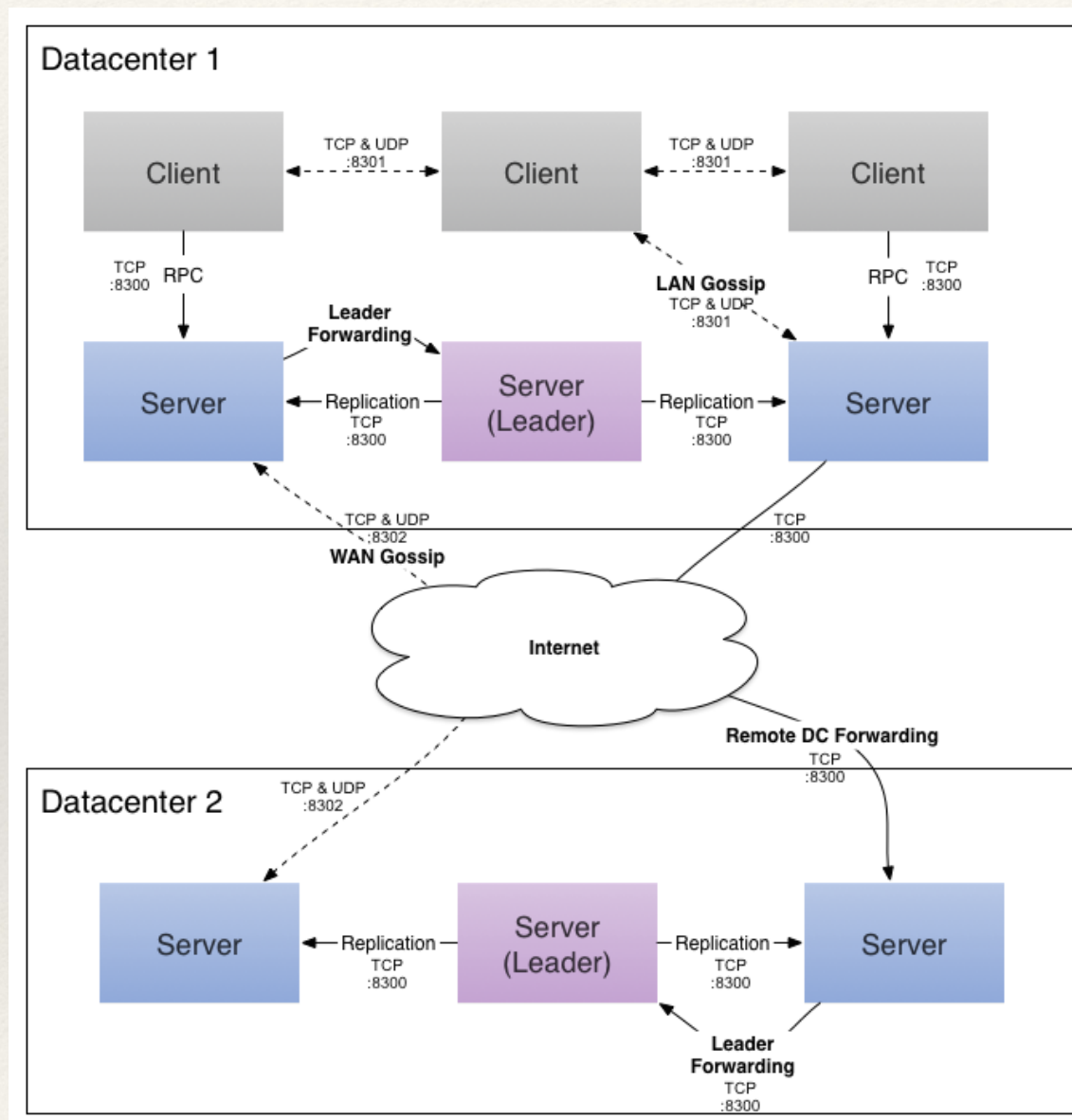
Consul Like

- ❖ Apache Zookeeper.
- ❖ Etcd.
- ❖ DNS server plus Consistent Key / Value Store + ZooKeeper + Nagios + ... = Consul.

Consul Components

- ❖ Health Checking: Consul clients can provide health checks, either associated with a given service ("is the web-server returning 200 OK"), or with the local node ("is memory utilization below 90%"). This information can be used by an operator to monitor cluster health, and it is used by the service discovery components to route traffic away from unhealthy hosts
- ❖ Key / Value Store: Applications can make use of Consul's hierarchical key / value store for dynamic configuration, feature flagging, coordination, leader election, and more.
- ❖ Service Discovery: Clients of Consul can provide a service, such as api or mysql, and other clients can use Consul to discover providers of a given service.

Consul Architecture



How Consul Works

- ❖ Every node that provides services to Consul runs a Consul agent. Running an agent is not required for discovering other services or getting / setting key / value data. The agent is responsible for health checking the services on the node as well as the node itself.
- ❖ The agents talk to one or more Consul servers in the cluster. The Consul servers are where data is stored and replicated. The servers themselves elect a leader. While Consul can function with one server, 3 to 5 is recommended to avoid failure scenarios leading to data loss. A cluster of Consul servers is recommended for each datacenter.
- ❖ Components of your infrastructure that need to discover other services or nodes can query any of the Consul servers or any of the Consul agents. The agents forward queries to the servers automatically
- ❖ Each datacenter runs a cluster of Consul servers. When a cross-datacenter service discovery or configuration request is made, the local Consul servers forward the request to the remote datacenter and return the result

Consul Agent

- ❖ Long running daemon on every member of Consul cluster.
- ❖ Client or server mode.
- ❖ Client runs on server hosting services.
- ❖ All nodes run an agent.
- ❖ Stays in sync, interface with REST and DHCP.


Consul Client

- ❖ Agent that forwards request to server.
- ❖ Mostly stateless.
- ❖ Client does LAN gossip (low traffic).
- ❖ Client runs on every hosting VM.

Consul Server

- ❖ Server is an agent as well.
- ❖ Server responsible for communicating with other nodes , elects and de-elects leader etc.
- ❖ Consul uses Serf, full gossip protocol to communicate with other nodes.

UI




SERVICES

NODES

KEY/VALUE

ACL

NYC3 ▾



Filter by name

any status ▾

EXPAND

consul9 passing

proxy4 passing

redis12 passing

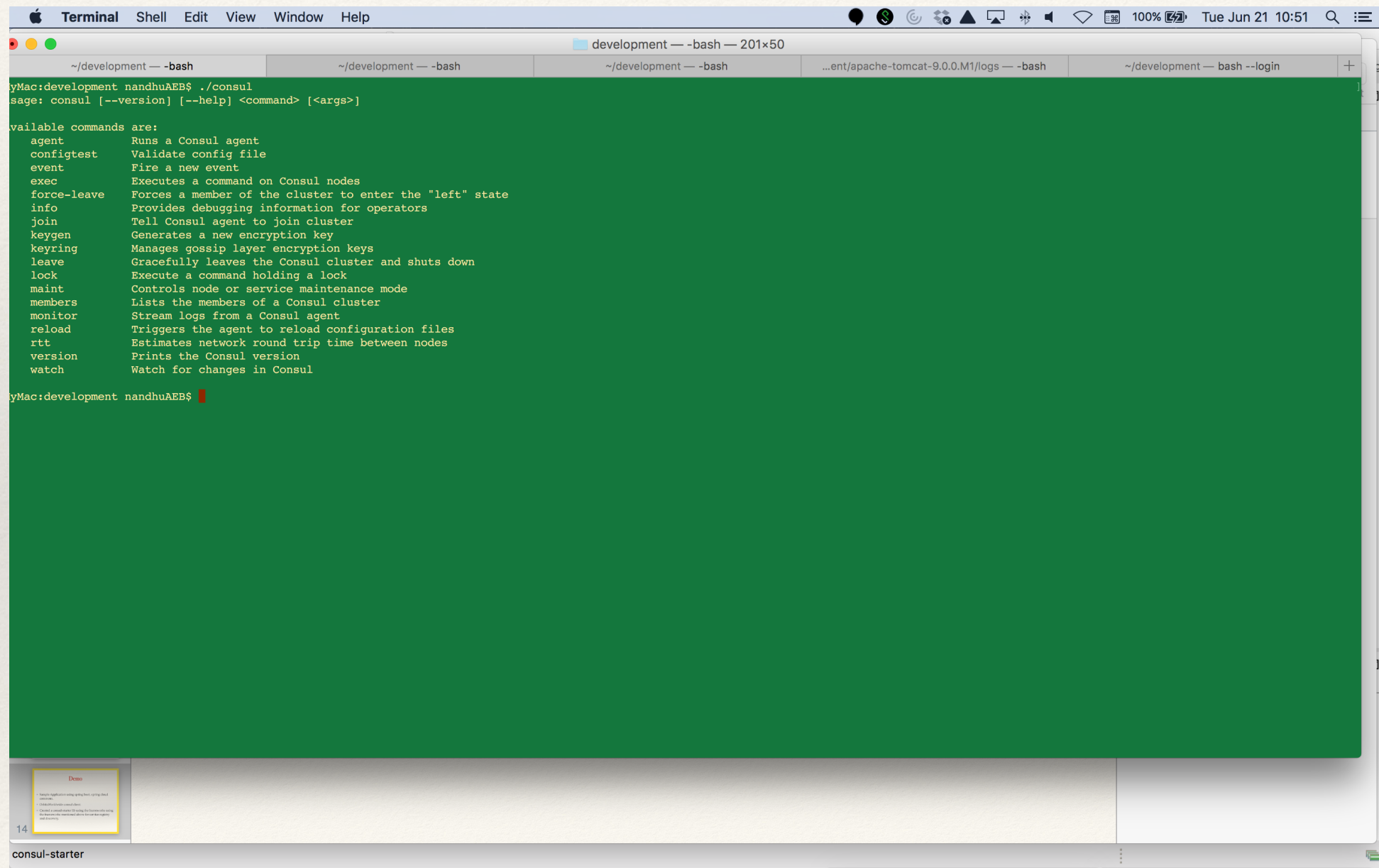
web12 passing

Consul command line

- ❖ Consul has a rich command line feature and fully functional rest api for registering , deregistering services.
- ❖ Starting a consul agent(server mode) `./consul agent -server -config-file ./consul_config.json -bind=0.0.0.0`.
- ❖ Config files can be stored locally and used to bootstrap config of consul.

```
{  
  "bootstrap": true,  
  "server": true,  
  "datacenter": "dc1",  
  "data_dir": "/consultmp/consul_new",  
  "log_level": "INFO",  
  "enable_syslog": true,  
  "acl_datacenter": "dc1",  
  "acl_default_policy": "allow",  
  "acl_down_policy": "extend-cache",  
  "ui_dir": "/Users/nandhuAEB/development/consulwebui/",  
  "acl_master_token": "secret",  
  "acl_token": "secret"  
}
```


Consul CLI



The screenshot shows a macOS Terminal window with the title bar "Terminal" and menu items "Shell", "Edit", "View", "Window", and "Help". The window contains a terminal session where the user has run the command `./consul`. The output displays the available commands for the Consul CLI. The terminal window has a green background and a white cursor. The status bar at the top of the terminal window shows "development — -bash — 201x50". The system status bar at the top of the screen shows the date and time as "Tue Jun 21 10:51".

```
Mac:development nandhuAEB$ ./consul
Usage: consul [--version] [--help] <command> [<args>]

available commands are:
agent          Runs a Consul agent
configtest     Validate config file
event          Fire a new event
exec           Executes a command on Consul nodes
force-leave    Forces a member of the cluster to enter the "left" state
info           Provides debugging information for operators
join           Tell Consul agent to join cluster
keygen         Generates a new encryption key
keyring        Manages gossip layer encryption keys
leave          Gracefully leaves the Consul cluster and shuts down
lock           Execute a command holding a lock
maint          Controls node or service maintenance mode
members        Lists the members of a Consul cluster
monitor        Stream logs from a Consul agent
reload         Triggers the agent to reload configuration files
rtt            Estimates network round trip time between nodes
version        Prints the Consul version
watch          Watch for changes in Consul

Mac:development nandhuAEB$
```

14

consul-starter

Consul Integration Webapps

- ❖ Easy integration with frameworks like Spring Boot, Drop-wizard.
- ❖ Consul integrates well with other Hashicorp products like Vault, Consul template etc.

Demo

- ❖ Sample Application using spring boot, spring cloud commons.
- ❖ OrbitzWorldwide consul client.
- ❖ Created a consul-starter lib using the frameworks using the frameworks mentioned above for service registry and discovery.

Reference

- ❖ <https://github.com/januslabs/consul-starter>.
- ❖ <https://consul.io>
- ❖ <http://12factor.net/>
- ❖ <http://microservices.io/>.