SERVICE DISCOVERY IN THE CLOUD

CONSUL @ SCALE14x
WHAT IS SERVICE DISCOVERY?

TWO MAIN COMPONENTS
I provide this service at this IP address and port.

Service registration

Can you tell me where to connect to this service?

Service discovery
WHERE WERE WE?

Intake Metrics Processed / second

External Proxy Mb/second

EC2 Nodes

add a graph
LATE 2014

• 4 year old codebase.

• Cutting apart our monolith.

• Rapid growth across the board.
OLD PRACTICES WERE FRAYING

We couldn’t do it the same way anymore.

Usually, you start off with some static configuration which gets you pretty far. Things get more complicated as you start deploying more services. With a live system, service locations can change quite frequently due to auto or manual scaling, new deployments of services, as well as hosts failing or being replaced.

Dynamic service registration and discovery becomes much more important in these scenarios in order to avoid service interruption.

HTTP://JASONWILDER.COM/BLOG/2014/02/04/SERVICE-DISCOVERY-IN-THE-CLOUD/
DISCOVERY WAS A HYBRID

- Chef searches. 30 minutes to update.
- Large numbers of manually managed IP addresses.
- There was nothing really wrong with it - but it was getting harder to manage.
"MO SYSTEMS. MO PROBLEMS." - THE NOTORIOUS B.I.G.

DISTRIBUTED SYSTEMS
Removing etcd and adding consul.

```bash
@@ -10,5 +10,5 @@
    cookbook 'openresty', git: 'https://github.com/darron/openresty-cookbook.git'
    cookbook 'gitreceive', git: 'https://github.com/darron/gitreceive-cookbook.git'
    cookbook 'sysdig', git: 'https://github.com/darron/sysdig-cookbook.git'
-    -cookbook 'etcd', git: 'https://github.com/spheromak/etcd-cookbook.git'
+    cookbook 'consul', git: 'https://github.com/darron/consul-cookbook.git'
    cookbook 'chef-sugar', git: 'https://github.com/sethvargo/chef-sugar'
```

Showing 6 changed files with 7 additions and 36 deletions.
OVERALL PLAN

NOVEMBER 2014

To provide a stable system to:

1. Register and provide a catalog of services that are available in the Datadog server cluster.
2. Make a distributed Key/Value store available on any machine in the Datadog server cluster.

This will allow us to:

1. Respond to individual component outages in seconds rather than minutes by integrating the service catalog health checks and configuration generation.
2. Build configuration files across the entire cluster in seconds by interacting with the Key Value store and Service Catalog in a simple and automated fashion.
3. Help to make the Datadog server cluster more resilient by leveraging the Service Catalog and Key Value store to automate additional failover and configuration tasks.
WHAT IS CONSUL?

**Service Discovery**
Consul makes it simple for services to register themselves and to discover other services via a DNS or HTTP interface. Register external services such as SaaS providers as well.

**Failure Detection**
Pairing service discovery with health checking prevents routing requests to unhealthy hosts and enables services to easily provide circuit breakers.

**Multi Datacenter**
Consul scales to multiple datacenters out of the box with no complicated configuration. Look up services in other datacenters, or keep the request local.

**Key/Value Storage**
Flexible key/value store for dynamic configuration, feature flagging, coordination, leader election and more. Long poll for near-instant notification of configuration changes.
WE WEREN'T SURE.

CAN IT HELP DATADOG?

Service discovery and configuration made easy. Distributed, highly available, and datacenter-aware.
STAGING

• ~100 nodes in total.

• 3 x m3.medium server nodes
  4GB of RAM - 3 ECU - 1 cpu core - SSD drives.
PHASE 1 PLAN

- Initial deploy
- Small amount of services.
- Minimal KV usage
- How will it act?
- Consul 0.4.1.
“MONITOR FIRST”
BEFORE PROD
HTTPS://BLOG.FROESE.ORG/PRESENTATIONS/
IT'S PROBABLY FINE

THIS IS FINE.

I'M OKAY WITH THE EVENTS THAT ARE UNFOLDING CURRENTLY.

THAT'S OKAY, THINGS ARE GOING TO BE OKAY.
DEPLOYED TO PROD LATE DECEMBER 2014.

desc 'Provision the first Consul bootstrap host'
task :consul_bootstrap, :az do |t, args|
  roles = BASE_ROLES + ['consul-bootstrap']
  size = IS_PROD ? 'm3.large' : 'm3.medium'
  az = args[:az] or pick_az
  provision(roles, 'consul-bootstrap', UBUNTU_1404_HVM_AMI, size, az, PROD_BACKEND, FIRST_EPHEMERAL)
  puts "Provisioned the first Consul bootstrap host."
end

desc 'Provision a Consul server host'
task :consul_server, :az do |t, args|
  roles = BASE_ROLES + ['consul-server']
  size = IS_PROD ? 'm3.large' : 'm3.medium'
  az = args[:az] or pick_az
  provision(roles, 'consul-server', UBUNTU_1404_HVM_AMI, size, az, PROD_BACKEND, FIRST_EPHEMERAL)
  puts "Provisioned a new Consul server host."
end
PROD

• 5 x m3.large server nodes
  7.5GB of RAM - 6.5 ECU
  2 cpu cores - SSD drives.

• Rapidly required us to spin up 2 more server nodes
  - it wasn’t stable at 3.
IT STABILIZED

AND ALL WAS WELL
ONE OF THE FIRST THINGS WE ADDED.

DATADOG SERVICE
DATADOG SERVICE

```json
{
  "service": {
    "name": "datadog",
    "tags": [
      "consul-server",
      "az-us-east-1c"
    ],
    "check": {
      "interval": "60s",
      "script": "/bin/true"
    }
  }
}
```
One of the first things we added:

**Datadog Service**
TOYED WITH AND DISABLED

CONSUL EXEC

```
[staging]darron@i-86402c2e:~$ consul exec -service datadog -tag consul-server w
  i-ccce987b: 23:51:24 up 52 days, 18:11, 0 users, load average: 0.00, 0.01, 0.05
  i-ccce987b: USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
  i-ccce987b:
  ==> i-ccce987b: finished with exit code 0
  i-bc13b90c: 23:51:24 up 52 days, 18:30, 0 users, load average: 0.04, 0.05, 0.05
  i-bc13b90c: USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
  i-bc13b90c:
  ==> i-bc13b90c: finished with exit code 0
  i-1dc56ea3: 23:51:24 up 52 days, 18:20, 0 users, load average: 0.00, 0.01, 0.05
  i-1dc56ea3: USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
  i-1dc56ea3:
  ==> i-1dc56ea3: finished with exit code 0
3 / 3 node(s) completed / acknowledged
[staging]darron@i-86402c2e:~$  
```
STRONGLY CONSISTENT KEY VALUE STORE AVAILABLE ON LOCALHOST WITH AN HTTP QUERY.

`GIT2CONSUL`

`HTTPS://GITHUB.COM/CIMPRESS-MCP/GIT2CONSUL`
Configuration we feed into the Consul KV store with git2consul. — Edit
HOW IT WORKS

GIT2CONSUL + CONSUL-CONFIG
UP AND TO THE RIGHT
MORE AND MORE USE
LEADERSHIP TRANSITIONS

PRETTY COMMON - MOSTLY HARMLESS
ALMOST 600 NODES

MAY 2015
WE’RE GETTING SERIOUS NOW

SERVICE REGISTRATION

Filter by name

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>bunk</td>
<td>24 passing</td>
</tr>
<tr>
<td>cassandra</td>
<td>60 passing</td>
</tr>
<tr>
<td>cole</td>
<td>2 passing</td>
</tr>
<tr>
<td>consul</td>
<td>5 passing</td>
</tr>
<tr>
<td>ctx-pshard0</td>
<td>4 passing</td>
</tr>
<tr>
<td>ctx-pshard1</td>
<td>6 passing</td>
</tr>
<tr>
<td>ctx-pshard2</td>
<td>6 passing</td>
</tr>
<tr>
<td>ctx-pshard3</td>
<td>6 passing</td>
</tr>
<tr>
<td>ctx-pshard4</td>
<td>4 passing</td>
</tr>
<tr>
<td>datadog</td>
<td>2068 passing</td>
</tr>
<tr>
<td>delancie-backend</td>
<td>1 passing</td>
</tr>
<tr>
<td>dogweb-backend</td>
<td>2 passing</td>
</tr>
</tbody>
</table>

**cassandra**

**TAGS**

- az-us-east-1b, az-us-east-1e, az-us-east-1a

**NODES**

- **i-9119fe7e** 10.186.190.71
  - Service 'cassandra' check: service:cassandra
  - Serf Health Status: serfHealth
  - 2 passing

- **i-f501a304** 10.95.168.48
  - Service 'cassandra' check: service:cassandra
  - Serf Health Status: serfHealth
  - 2 passing

- **i-60dcca9b** 10.35.179.159
  - Service 'cassandra' check: service:cassandra
  - Serf Health Status: serfHealth
  - 2 passing
CURL/HTTP LOOKUP

SERVICE DISCOVERY

```
[staging] darron@i-86402c2e:~$ curl -s http://127.0.0.1:8500/v1/catalog/service/consul | jq .
[
  {
    "Node": "i-1dc56ea3",
    "Address": "10.186.190.143",
    "ServiceID": "consul",
    "ServiceName": "consul",
    "ServiceTags": [],
    "ServiceAddress": "",
    "ServicePort": 8300
  },
  {
    "Node": "i-bc13b90c",
    "Address": "10.233.144.121",
    "ServiceID": "consul",
    "ServiceName": "consul",
    "ServiceTags": [],
    "ServiceAddress": "",
    "ServicePort": 8300
  },
  {
    "Node": "i-ccce987b",
    "Address": "10.47.151.240",
    "ServiceID": "consul",
    "ServiceName": "consul",
    "ServiceTags": [],
    "ServiceAddress": "",
    "ServicePort": 8300
  }
]```
SERVICE DISCOVERY

Dig command output:

[darron@i-86402c2e:~]$ dig consul.service.consul

; <<>> DiG 9.9.5-3ubunto0.6-Ubuntu <<> consul.service.consul
; global options: +cmd
; Got answer:
; -->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27619
; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

; QUESTION SECTION:
;consul.service.consul. IN A

; ANSWER SECTION:
consul.service.consul. 10 IN A 10.47.151.240
consul.service.consul. 10 IN A 10.233.144.121
consul.service.consul. 10 IN A 10.186.190.143

; Query time: 1 msec
; SERVER: 127.0.0.1#53(127.0.0.1)
; WHEN: Wed Jan 06 01:39:13 UTC 2016
; MSG SIZE  rcvd: 150
WOULD IT FLAP?

IN AND OUT OF THE SERVICE CATALOG

```json
{
"service": {
  "name": "datadog",
  "tags": [
    "consul-server",
    "az-us-east-1c"
  ],
  "check": {
    "interval": "60s",
    "script": "/bin/true"
  }
}
```
IN AND OUT OF THE SERVICE CATALOG

NO. IT DID NOT FLAP
WORRIED ABOUT SPEED

USING DNS
FRONTED CONSUL’S DNS RESOLVER

DNMSAQ

```
[staging] darron@i-86402c2e:/etc/dnsmasq.d$ cat 10-consul
server=/consul/127.0.0.1#8600
[staging] darron@i-86402c2e:/etc/dnsmasq.d$ sudo cat /etc/consul.d/ttl.json
{
    "dns_config": {
        "allow_stale": true,
        "service_ttl": {
            "*": "10s"
        }
    }
}
[staging] darron@i-86402c2e:/etc/dnsmasq.d$
```
CONSUL-TEMPLATE

CONSUL_DNS_BACKUP

(THE HOSTS FILE)
EVEN IN STAGING

NOT SUCCESSFUL
USE THE KV STORE TO DISTRIBUTE.

BUILT ON THE SERVER NODES
IT WORKS REALLY WELL
IT WAS A BIT HAIRY

WITHOUT RATE LIMITING
NONE AT ALL

NO LEADERSHIP
TRANSITIONS
"LET'S CLEAN THIS UP"

THE VERY NEXT DAY
CAUSING LEADERSHIP TRANSITIONS

READ-PRESSURE
CONSUL IS NEW

THE EDGE WAS A LITTLE BLOODY

THERE WAS VERY LITTLE REAL WORLD INFORMATION ABOUT IT.
DON'T DO THIS - WITH CONSUL 0.5.X

LOTS OF SMALL KEYS

- any_filter_array
  - enable any_filter_array everywhere
  - 4 months ago

- api_log_info_writes
  - [prod][api log] Disable info writes
  - 2 months ago

- appdirect_marketplace
  - [prod] enable appdirect endpoints
  - 3 months ago

- automate_filter_aws_tags
  - [Prod] Enabling EC2 Service checks for 40% of orgs
  - 5 months ago

- aws_auto_scaling_tags_fix
  - updates tree structure
  - 6 months ago

- aws_billing_linked_accounts
  - [AWS] Billing - add a feature flag to keep billing metrics related to...
  - 5 months ago

- aws_cloud_provider_tag
  - updates tree structure
  - 6 months ago

- aws_custom_cw_metrics_n...
  - updates tree structure
  - 6 months ago

- aws_ec2_check_spurious_t...
  - [ec2] check spurious tags for everyone
  - a day ago

- aws_ec2_custom_tags_issue
  - [AWS] enable workaround for missing ec2 custom tags
  - 2 months ago

- aws_ec2_filter_with_account...
  - updates tree structure
  - 6 months ago

- aws_ec2_metrics_all_stats
  - updates tree structure
  - 6 months ago

- aws_ecs_cluster_tags
  - Revert "Revert "[ECS] Turn cluster tags for everyone"
  - 5 months ago

- aws_elb_host_tags
  - updates tree structure
  - 6 months ago

- aws_elb_metrics_all_stats
  - [AWS] ELB - retrieve all stats for an org
  - 3 months ago

- aws_elb_new_hostname
  - Removed Campain 17774 and Groundwork 21953 from aws new host name list
  - 2 months ago

- aws_filter_aggregates
  - enabled aws_filter_aggregates for org 38736
  - 22 days ago
ONCE WE UNDERSTOOD
WE UPSIZED OUR VMS
INSTALLED LARGER SERVER NODES

THINGS QUIETED DOWN
THE DNSMASQ THING...

HOW DID IT WORK?
DNSMASQ HONORS THE CONSUL TTL ASKS ONCE EVERY 10 SECONDS
**WELL**

**IT RESPONDS QUICKLY**

<table>
<thead>
<tr>
<th>ssh</th>
<th>staging</th>
<th>darron@i-1dc56ea3:~</th>
<th>consul</th>
<th>10</th>
<th>IN</th>
<th>A</th>
<th>10.7.201.208</th>
<th>10.148.10.224</th>
<th>Rtt: 643.448μs</th>
<th>211.277μs</th>
<th>688.294μs</th>
<th>88.793μs</th>
<th>221.247μs</th>
<th>225.727μs</th>
<th>112.676μs</th>
<th>103.779μs</th>
</tr>
</thead>
<tbody>
<tr>
<td>bunk</td>
<td>service</td>
<td>consul.</td>
<td>consul</td>
<td>5</td>
<td>IN</td>
<td>A</td>
<td>10.148.10.224</td>
<td>10.7.201.208</td>
<td>Rtt: 211.277μs</td>
<td>223.807μs</td>
<td>688.294μs</td>
<td>88.793μs</td>
<td>221.247μs</td>
<td>225.727μs</td>
<td>112.676μs</td>
<td>103.779μs</td>
</tr>
<tr>
<td>bunk</td>
<td>service</td>
<td>consul.</td>
<td>consul</td>
<td>5</td>
<td>IN</td>
<td>A</td>
<td>10.7.201.208</td>
<td>10.148.10.224</td>
<td>Rtt: 211.277μs</td>
<td>223.807μs</td>
<td>688.294μs</td>
<td>88.793μs</td>
<td>221.247μs</td>
<td>225.727μs</td>
<td>112.676μs</td>
<td>103.779μs</td>
</tr>
<tr>
<td>bunk</td>
<td>service</td>
<td>consul.</td>
<td>consul</td>
<td>2</td>
<td>IN</td>
<td>A</td>
<td>10.7.201.208</td>
<td>10.148.10.224</td>
<td>Rtt: 223.807μs</td>
<td>221.247μs</td>
<td>688.294μs</td>
<td>88.793μs</td>
<td>221.247μs</td>
<td>225.727μs</td>
<td>112.676μs</td>
<td>103.779μs</td>
</tr>
<tr>
<td>bunk</td>
<td>service</td>
<td>consul.</td>
<td>consul</td>
<td>10</td>
<td>IN</td>
<td>A</td>
<td>10.148.10.224</td>
<td>10.7.201.208</td>
<td>Rtt: 688.294μs</td>
<td>223.807μs</td>
<td>688.294μs</td>
<td>88.793μs</td>
<td>221.247μs</td>
<td>225.727μs</td>
<td>112.676μs</td>
<td>103.779μs</td>
</tr>
<tr>
<td>bunk</td>
<td>service</td>
<td>consul.</td>
<td>consul</td>
<td>10</td>
<td>IN</td>
<td>A</td>
<td>10.7.201.208</td>
<td>10.148.10.224</td>
<td>Rtt: 688.294μs</td>
<td>223.807μs</td>
<td>688.294μs</td>
<td>88.793μs</td>
<td>221.247μs</td>
<td>225.727μs</td>
<td>112.676μs</td>
<td>103.779μs</td>
</tr>
</tbody>
</table>
DNSMASQ AND THE MAGICAL HOSTS FILE

HOW DID IT WORK?
People were a bit scared. Weren’t sure if we were going to continue.
They dropped out of the catalog.

Nodes were going deaf.
I was visiting the office & heard some grumbling.

But I could finally duplicate the problem.
HUGE THANKS TO JAMES AND ARMON FOR ALL THEIR HELP!

HASHICORP LENT A HAND
BUT THE KEY WAS QUAKE RELATED.

2 DEADLOCKS FIXED
FOR THE MOST PART
AND ALL WAS RIGHT
850 NODES - MOSTLY STABLE - BUT CAUTIOUS

OCTOBER 2015
CONSUL-CONFIG HELPED AS WE GREW
LIKE CONSUL EXEC BUT WITH SECURITY++

STARTED USING EVENTS

```
[staging]darron@i-86402c2e:~$ sudo cat /etc/consul.d/event-watch-apt-update.json

{
  "watches": [
    {
      "type": "event",
      "name": "apt-update",
      "handler": "sifter run -d true -e 'apt-get update'"
    }
  ]
}

[staging]darron@i-86402c2e:~$ consul event -service datadog -name apt-update
Event ID: 8ef0bb99-b7e4-fbc7-87c1-b82f76d78638
[staging]darron@i-86402c2e:~$  ```
RUN 1 OF N PROCESSES - WITH HOT SPARES

CONSUL LOCK

WAITING

WAITING

RUNNING
RUN 1 OF N PROCESSES - WITH HOT SPARES

CONSUL LOCK

WAITING

RUNNING

CRASHES
RUN 1 OF N PROCESSES - WITH HOT SPARES

CONSUL LOCK

description "consul lock example"

# Defaults set by kernel
limit nofile 1024 4096

emits consullock-up

start on runlevel [2345]
stop on runlevel [!2345]

env CONSUL_TOKEN=token-goes-here

exec consul lock -n 1 -token $CONSUL_TOKEN consullock /usr/local/bin/binary goes here

post-start exec initctl emit consullock-up

kill signal INT
WHEN THE LEADERSHIP TRANSITIONS GREW
SUPERSIZED OUR SERVERS
At 1200 nodes - it's like we (almost) turned them off.

C3.2xlarge did the trick.
2 SMALL OUTAGES LAST YEAR.
ONE FOR 3 MINUTES BECAUSE OF A PACKAGING PROBLEM
ONE FOR AN HOUR THAT WAS DOCUMENTATION AND “BROADCAST INPUT TO ALL PANES” RELATED
BTW - WE’RE HIRING

HTTP://JOBS.DATADOGHQ.COM/
WHAT HAVE WE LEARNED?

JANUARY 2016
It's your datacenter's backbone.

Consul is awesome.
MONITORING IS ESSENTIAL
TONS OF FIXES AND UPGRADES

UPGRADE TO 0.6.X

Below are the available downloads for the latest version of Consul (0.6.0). Please download the proper package for your operating system and architecture.

You can find the SHA256 checksums for Consul 0.6.0 online and you can verify the checksums signature file which has been signed using HashiCorp's GPG key. You can also download older versions of Consul from the releases service.

Download Consul Web UI

Mac OS X

32-bit | 64-bit
TASTES GREAT, LESS FILLING!

USES LESS MEMORY
CONSUL LOVES CPU

FEED IT ALL THE CPUS

LARGER CPU

ALL THE THINGS
Some Example Sizing

- m3.large ~300 nodes
- c3.xlarge ~500 nodes
- c3.2xlarge ~800 nodes
- As always YMMV.
- 0.6 is more efficient - might be able to be smaller nodes.
EMBRACE FAILURE

BUILD FOR IT - ADD RETRIES - BACKOFF - CIRCUIT BREAKERS

MAKES YOUR WHOLE SYSTEM MORE RESILIENT
DON'T DDOS YOURSELF

WATCH YOUR READ VELOCITY
RATHER THAN LOTS OF SMALL KEYS

USE FEWER AND LARGER KEYS

ESPECIALLY IF YOU’RE READING A LOT OF THEM AT ONCE
FEED IN DATA FROM THE OUTSIDE

LOCK DOWN PARTS OF THE KV STORE

ACLS ARE YOUR FRIEND
MAKE SURE THEY ONLY FIRE WHEN YOU WANT THEM TO

CONSUL WATCHES ARE POWERFUL

HTTPS://GITHUB.COM/DARRON/SIFTER
IF OUTPUT ISN'T UNIQUE

USE THE KV STORE TO MOVE THOSE FILES AROUND
That’s my last tip for today but we have one more thing.
Use the KV store to transport configuration files in & out both directions.
MAIN FEATURES

- 10MB Go binary
- Uploads and downloads files under 512KB
- Emits Dogstatsd metrics and Datadog Events
- Files sent == files delivered
- Doesn’t re-upload or re-deliver
- Very safe
- Runs commands after delivery
< 500MS TO DELIVER A FILE TO 1000 NODES

IT'S SUPER FAST

[kvexpress out events - Average time to write hosts file.

Avg of kvexpress.time over key:hosts, direction:out by host]
SPEED IN AND OUT
WHEN FILES UPDATE

CAN POST DIFFS
MEASURE ALL THE THINGS

METRICS FOR ALL
VERY QUICK DEMO
HTTPS://GITHUB.COM/DATADOG/KVEXPRESS
QUESTIONS?
SERVICE DISCOVERY IN THE CLOUD

CONSUL @ SCALE14x

THANKS!

DARRON@FROESE.ORG
@DARRON
GITHUB.COM/DARRON