CIRCUIT

(Re)discover your AEM

Presented by: Jakub Wądołowski (@jwadolowski)

ICF | OLSON + Adobe

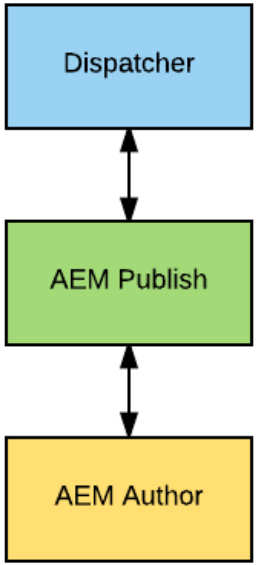
#CIRCUIT16

Environment complexity

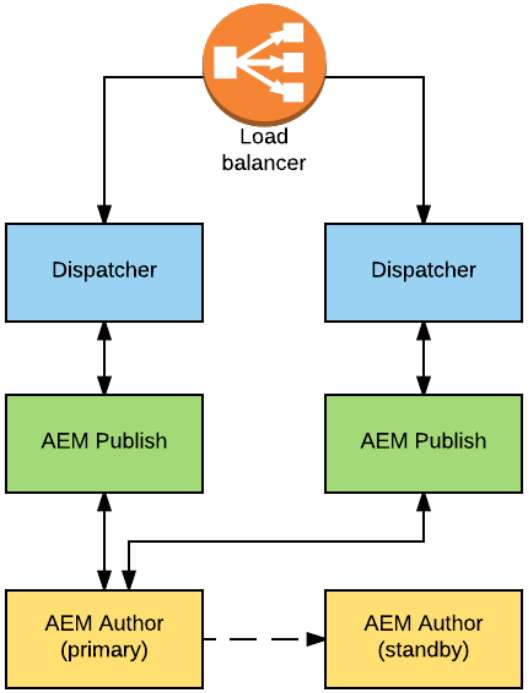
https://flic.kr/p/p2ZWXH

Complexity keeps growing

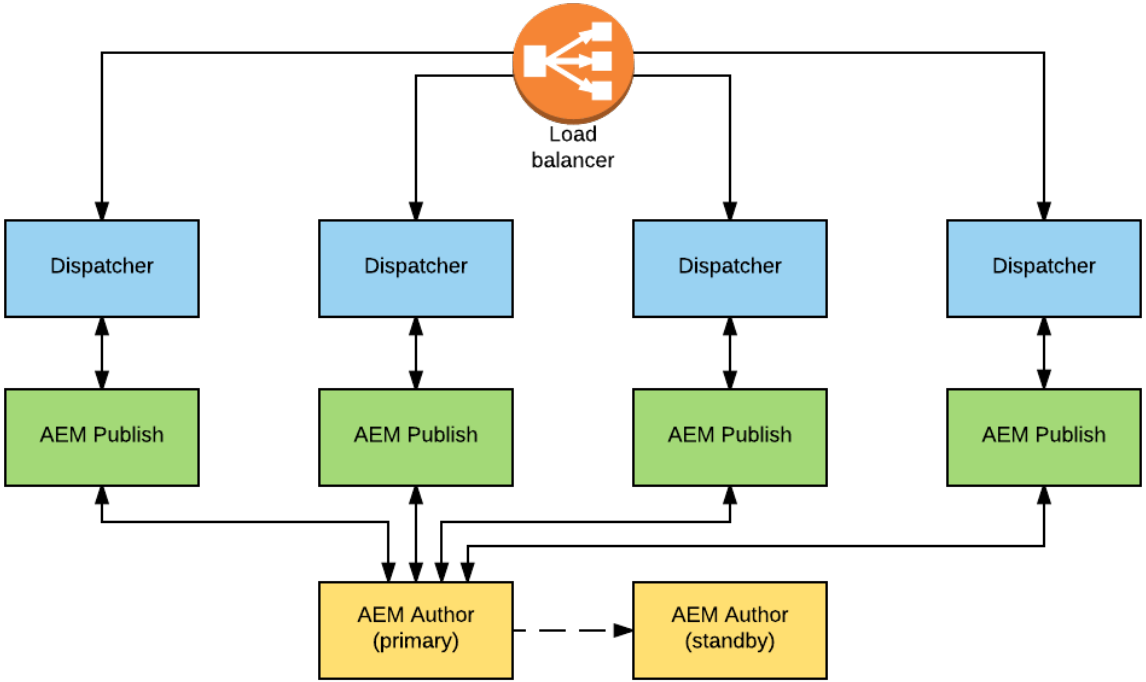
Start point



The next stage



Traffic volume keeps growing



Typical AEM project scope

- many environments (3+)
- dozens of IP addresses
- a few internal/external (micro)services
- hundreds of connections

Things get complicated quickly

https://flic.kr/p/bMbzcn

Hardcoding or manual configuration

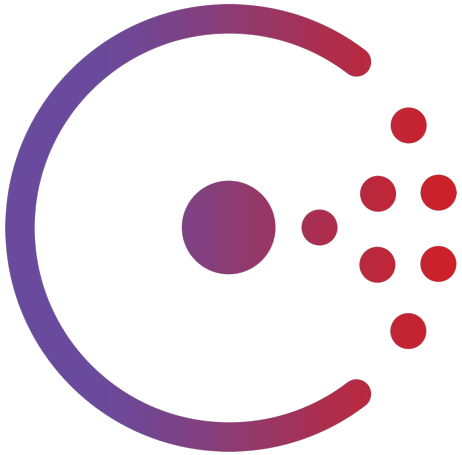
Solution research

What’s on the market

- Configuration management
  - Chef
  - Puppet
  - Ansible
- Service discovery
  - Consul
  - ZooKeeper
  - etcd
  - Doozer

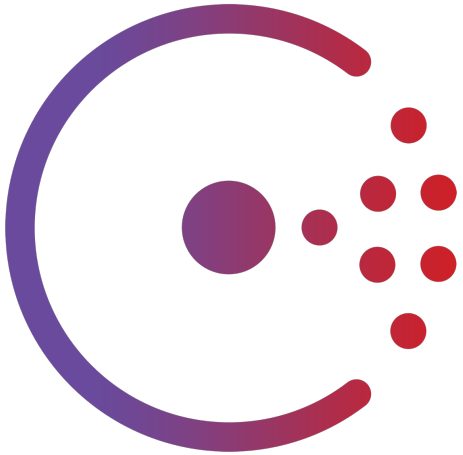
Why Consul? (1)

- Solves 4 basic problems
  - service discovery
  - load balancing
  - health checking
  - key-value configuration

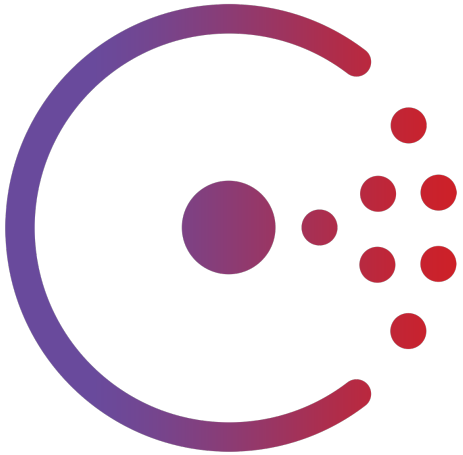


Why Consul? (2)

- Single Go binary
- Super simple deployment
- Datacenter aware
- Works everywhere (no multicast)
- Easy to operate



Consul architecture



- Agent installed on every server
- 2 types of agents
  - client
  - server (3 or 5 in each DC)
- Communication over gossip protocol
- Queries always go to local agent

SERVICES | NODES | KEY/VALUE | ACL | EU-WEST-1A

Filter by name | any status | EXPAND

- aem — 2 passing
- consul — 1 passing
- dispatcher — 1 passing
- sftp — 2 passing
- solr — 4 passing
- templating-engine — 2 passing
- tomcat — 2 passing

aem

TAGS

author, a1, publish, p1

NODES

[redacted] vagrant 10.0.2.15 — 1 passing

Serf Health Status serfHealth — passing

[redacted] vagrant 10.0.2.15 — 1 passing

Serf Health Status serfHealth — passing

SERVICES | NODES | KEY/VALUE | ACL | EU-WEST-1A

Filter by name | any status | EXPAND

[redacted] vagrant 10 services

[redacted] vagrant 10.0.2.15 DEREGISTER

SERVICES

aem author a1	:6102
aem publish p1	:6103
consul	:8300
dispatcher d1	:80
sftp No tags	:2211
solr master sm1	:8983
solr slave ss1	:8984
templating-engine author	:3002
templating-engine publish	:3003
tomcat tm1	:8080

CHECKS

```
$ consul members
Node            Address              Status  Type    Build  Protocol  DC
abcd-uat-a1     172.11.5.203:8301  alive   client  0.6.3  2         eu-west-1a
abcd-uat-adnx1  172.11.6.21:8301   alive   client  0.6.3  2         eu-west-1a
abcd-uat-dnx1   172.11.4.122:8301  alive   server  0.6.3  2         eu-west-1a
abcd-uat-ips1   172.11.4.6:8301    alive   client  0.6.3  2         eu-west-1a
abcd-uat-p1     172.11.5.62:8301   alive   server  0.6.3  2         eu-west-1a
abcd-uat-rl1    172.11.6.46:8301   alive   client  0.6.3  2         eu-west-1a
abcd-uat-sftp1  172.11.7.23:8301   alive   client  0.6.3  2         eu-west-1a
abcd-uat-sm1    172.11.5.157:8301  alive   client  0.6.3  2         eu-west-1a
abcd-uat-tcss1  172.11.4.196:8301  alive   server  0.6.3  2         eu-west-1a

$ consul members -wan
Node                       Address              Status  Type    Build  Protocol  DC
abcd-uat-dnx1.eu-west-1a   172.11.4.122:8302  alive   server  0.6.3  2         eu-west-1a
abcd-uat-dnx2.eu-west-1b   172.11.4.141:8302  alive   server  0.6.3  2         eu-west-1b
abcd-uat-p1.eu-west-1a     172.11.5.62:8302   alive   server  0.6.3  2         eu-west-1a
abcd-uat-p2.eu-west-1b     172.11.5.91:8302   alive   server  0.6.3  2         eu-west-1b
abcd-uat-tcss1.eu-west-1a  172.11.4.196:8302  alive   server  0.6.3  2         eu-west-1a
abcd-uat-tcss2.eu-west-1b  172.11.4.241:8302  alive   server  0.6.3  2         eu-west-1b
```

https://flic.kr/p/cwV8W7

Service registry

```
{
    "service": {
        "id": "aem_author",
        "name": "aem",
        "port": 6102,
        "tags": [
            "author",
            "a1"
        ]
    }
}
```

Discover!

https://flic.kr/p/CGmgy3

ICF | OLSON + Adobe

#CIRCUIT16

Services, not IP addresses!

HTTP API

- /v1/catalog/nodes
- /v1/catalog/node/<node>
- /v1/catalog/services
- /v1/catalog/service/<service>
- /v1/catalog/node/<node>
- /v1/catalog/register
- /v1/catalog/deregister

```
$ curl -s "http://localhost:8500/v1/catalog/services"
{
    "aem": [ "author", "a1", "publish", "p1" ],
    "dispatcher": [ "d1" ],
    "sftp": [],
    "solr": [ "sm1", "slave", "ss1", "master" ],
    "tomcat": [ "tm1" ]
}
```

```
$ curl -s "http://localhost:8500/v1/catalog/service/aem"
[
    {
        "Address": "10.0.2.15",
        "Node": "xyz-vagrant",
        "ServiceID": "aem_author",
        "ServiceName": "aem",
        "ServicePort": 6102,
        "ServiceTags": [ “author", “a1" ]
    },
    {
        "Address": "10.0.2.15",
        "Node": "xyz-vagrant",
        "ServiceID": "aem_publish",
        "ServiceName": "aem",
        "ServicePort": 6103,
        "ServiceTags": [ “publish", “p1" ]
    }
]
```

```
$ curl -s "http://localhost:8500/v1/catalog/service/aem?tag=author"
[
    {
        "Address": "10.0.2.15",
        "CreateIndex": 577,
        "ModifyIndex": 577,
        "Node": "xyz-vagrant",
        "ServiceAddress": "",
        "ServiceEnableTagOverride": false,
        "ServiceID": "aem_author",
        "ServiceName": "aem",
        "ServicePort": 6102,
        "ServiceTags": [
            "author",
            "a1"
        ]
    }
]
```

DNS API

[tag.]<service>.service[.datacenter].<domain>
<service>.service.<domain>

```
$ dig @localhost -p 8600 aem.service.consul

; <<>> DiG 9.8.3-P1 <<>> @localhost -p 8600 aem.service.consul
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2476
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

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

;; ANSWER SECTION:
aem.service.consul.     0       IN      A       172.18.5.62
aem.service.consul.     0       IN      A       172.18.5.203

;; Query time: 10 msec
;; SERVER: ::1#8600(::1)
;; WHEN: Sat Jul 23 13:53:48 2016
;; MSG SIZE  rcvd: 104
```

```
$ dig @localhost -p 8600 p1.aem.service.consul

; <<>> DiG 9.8.3-P1 <<>> @localhost -p 8600 p1.aem.service.consul
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3947
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;p1.aem.service.consul.         IN    A

;; ANSWER SECTION:
p1.aem.service.consul.    0    IN    A    10.0.2.15

;; Query time: 2 msec
;; SERVER: 127.0.0.1#8600(127.0.0.1)
;; WHEN: Sat Jul 23 14:44:59 2016
;; MSG SIZE  rcvd: 76
```

```
$ dig @localhost -p 8600 publish.aem.service.eu-west-1b.consul

; <<>> DiG 9.8.3-P1 <<>> @localhost -p 8600 publish.aem.service.eu-west-1b.consul
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36721
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;publish.aem.service.eu-west-1b.consul. IN A

;; ANSWER SECTION:
publish.aem.service.eu-west-1b.consul. 0 IN A    172.18.5.91

;; Query time: 2 msec
;; SERVER: ::1#8600(::1)
;; WHEN: Sat Jul 23 13:57:01 2016
;; MSG SIZE  rcvd: 108
```

```
$ dig @localhost -p 8600 a1.aem.service.consul SRV

; <<>> DiG 9.8.3-P1 <<>> @localhost -p 8600 a1.aem.service.consul SRV
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45677
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;a1.aem.service.consul.         IN    SRV

;; ANSWER SECTION:
a1.aem.service.consul.    0    IN    SRV    1 1 6102 xyz-vagrant.node.eu-west-1a.consul.

;; ADDITIONAL SECTION:
xyz-vagrant.node.eu-west-1a.consul. 0 IN A    10.0.2.15

;; Query time: 0 msec
;; SERVER: 127.0.0.1#8600(127.0.0.1)
;; WHEN: Sat Jul 23 22:50:14 2016
;; MSG SIZE  rcvd: 17
```

Consul + AEM

https://flic.kr/p/eE48CR

Where’s my publish?

OLSON + Adobe

#CIRCUIT16

Replication agents

- Number of publish instances and availability zones (DCs) is constant for each environment
- Publish to DC allocation: ID % AZ.length
- Chef's responsible for agent configuration

```
(1..node['xyz-webapp']['publish_count']).each do |id|
```

```
(1..node['xyz-webapp']['publish_count']).each do |id|
  domain = "p#{id}.aem.service.#{az[(id - 1) % az.length]}.consul"
  transport_uri = "http://#{domain}:6103/bin/receive?sling:authRequestLogin=1"
```

```
(1..node['xyz-webapp']['publish_count']).each do |id|
  domain = "p#{id}.aem.service.#{az[(id - 1) % az.length]}.consul"
  transport_uri = "http://#{domain}:6103/bin/receive?sling:authRequestLogin=1"

  cq_jcr "Author: /etc/replication/agents.author/publish#{id}}/jcr:content" do
```

```
  end
end
```

```
(1..node['xyz-webapp']['publish_count']).each do |id|
  domain = "p#{id}.aem.service.#{az[(id - 1) % az.length]}.consul"
  transport_uri = "http://#{domain}:6103/bin/receive?sling:authRequestLogin=1"

  cq_jcr "Author: /etc/replication/agents.author/publish#{id}/jcr:content" do
    path "/etc/replication/agents.author/publish#{id}/jcr:content"
    username node['cq']['author']['credentials']['login']
    password node['cq']['author']['credentials']['password']
    instance "http://localhost:#{node['cq']['author']['port']}"

    encrypted_fields %w(transportPassword)
    append false

    action :create
  end
end
```

```
(1..node['xyz-webapp']['publish_count']).each do |id|
  domain = "p#{id}.aem.service.#{az[(id - 1) % az.length]}.consul"
  transport_uri = "http://#{domain}:6103/bin/receive?sling:authRequestLogin=1"

  cq_jcr "Author: /etc/replication/agents.author/publish#{id}/jcr:content" do
    path "/etc/replication/agents.author/publish#{id}/jcr:content"
    username node['cq']['author']['credentials']['login']
    password node['cq']['author']['credentials']['password']
    instance "http://localhost:#{node['cq']['author']['port']}"
    properties(
      'jcr:primaryType' => 'nt:unstructured',
      'enabled' => 'true',
      'transportUri' => transport_uri,
      'transportUser' => 'admin',
      'transportPassword' => node['cq']['publish']['credentials']['password'],
      'cq:template' => '/libs/cq/replication/templates/agent',
      'sling:resourceType' => 'cq/replication/components/agent',
      'logLevel' => 'info'
    )
    encrypted_fields %w(transportPassword)
    append false

    action :create
  end
end
```

Agents on author

Test and Target (test_and_target)
This agent's replication settings are inherited from the cloud configuration attached to a campaign.

Agent is **disabled**. Replicating to [illegible]

Queue is **not active**

Replication agent (publish1)

Agent that replicates to p1.aem.service.eu-west-1a.consul

Agent is **enabled.** Replicating to **http://p1.aem.service.eu-west-1a.consul:6103/bin/receive?sling:authRequestLogin=1**

Queue is **idle**

Dispatcher Flush (flush)
Agent that sends flush requests to the dispatcher.

Agent is **disabled**. Replicating to [illegible]

Queue is **not active**

Agent is triggered when an offline reached

Default Agent (publish)
Agent that replicates to the default publish instance.

Agent is **disabled**. Replicating to [illegible]

Queue is **not active**

https://flic.kr/p/qm4gV8

DNS assembly

Transparent DNS

- Consul DNS interface listens on port 8600
- /etc/resolv.conf accepts just IPs
- **dnsmasq** to the rescue
  - DNS "proxy"
  - listens on localhost:53
  - DNS query routing
  - /etc/hosts on steroids

```
# dnsmasq logic
if domain ~ /\.consul$/
  dns_query(domain, "localhost:8600")
else
  internal_db.lookup(domain) || upstream_forward(domain)
end
```

dnsmasq - your new primary DNS server

```
# /etc/resolv.conf
nameserver 127.0.0.1
nameserver 8.8.8.8
nameserver 8.8.4.4
```

https://flic.kr/p/c42Qmm

Pick the right one

Flush agents

- 1:1 mapping between dispatcher and publish
- config wise each publish instance needs to be exactly the same

dnsmasq helps again

- Yet another domain name space: **.local**
- Chef renders dnsmasq config using Consul data
  - give me all dispatcher servers
  - pick the one that matches my ID (p1 => d1)
  - expose it under dispatcher.local

```
# /etc/dnsmasq.d/local_domains.conf
address=/dispatcher.local/172.19.9.122
address=/publish.local/172.19.9.62
address=/author.local/172.19.9.203
```

```
# Address of 1st dispatcher
[root@xyz-uat-p1 ~]# dig dispatcher.local +short
172.19.9.122

# Address of 2nd dispatcher
[root@xyz-uat-p2 ~]# dig dispatcher.local +short
172.19.9.141
```

Agents on publish

Dispatcher flush (flush1)

Sends flush requests to dispatcher

Agent is **enabled.** Replicating to **http://dispatcher.local/dispatcher/invalidate.cache**

Queue is **idle**

Agent is ignored on normal replication

Agent is triggered when receiving replication events

[illegible]

[illegible]

Undocumented features

https://flic.kr/p/bzeVcH

```
/renders
{
    /rend
    {
        /hostname "publish.local"
        /port "6103"

        /always-resolve "1"
    }
}
```

Health checks

https://flic.kr/p/8rhLYf

```
{
    "service": {
        "check": {
            "interval": "10s",
            "tcp": "localhost:8080",
            "timeout": "1s"
        },
        "id": "tomcat",
        "name": "tomcat",
        "port": 8080,
        "tags": [
            "tm1"
        ]
    }
}
```

Your front load balancer is the only one you'll ever need


https://flic.kr/p/aHfnmc
Prepared queries
ICF
OLSON +
Adobe
#CIRCUIT16

Prepared queries

- Complex service queries
- Returns set of healthy nodes that meets predefined criteria
- Example:
  - tomcat-cluster.query.consul:8080
  - use Tomcat from the same datacenter, as long as it’s healthy. Otherwise pick the nearest one.

Consul locks

https://flic.kr/p/p4xhXr

Consul locks

- Distributed locking
- Mutual exclusion or semaphore
- Configurable amount of holders (1 by default)

App rollouts

https://flic.kr/p/xYdtNW

ICF | OLSON + Adobe

#CIRCUIT16

App rollouts

- Practical use case: deployments
- Solves "at least one of publish must be always running" problem

Summary

- Services, not IP addresses
- Everything supports DNS
- Get rid of internal load balancers
- Simpler deployments
- Can’t imagine a project without Consul

THANK YOU!