



Deploy a Single Datacenter

Deploy a Single Datacenter

Objective 2a: Start and manage the Consul process

Objective 2b: Interpret a Consul agent configuration

Objective 2c: Configure Consul network addresses and ports

Objective 2d: Describe and configure agent join and leave behaviors

1

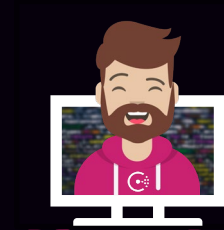
2

3

4

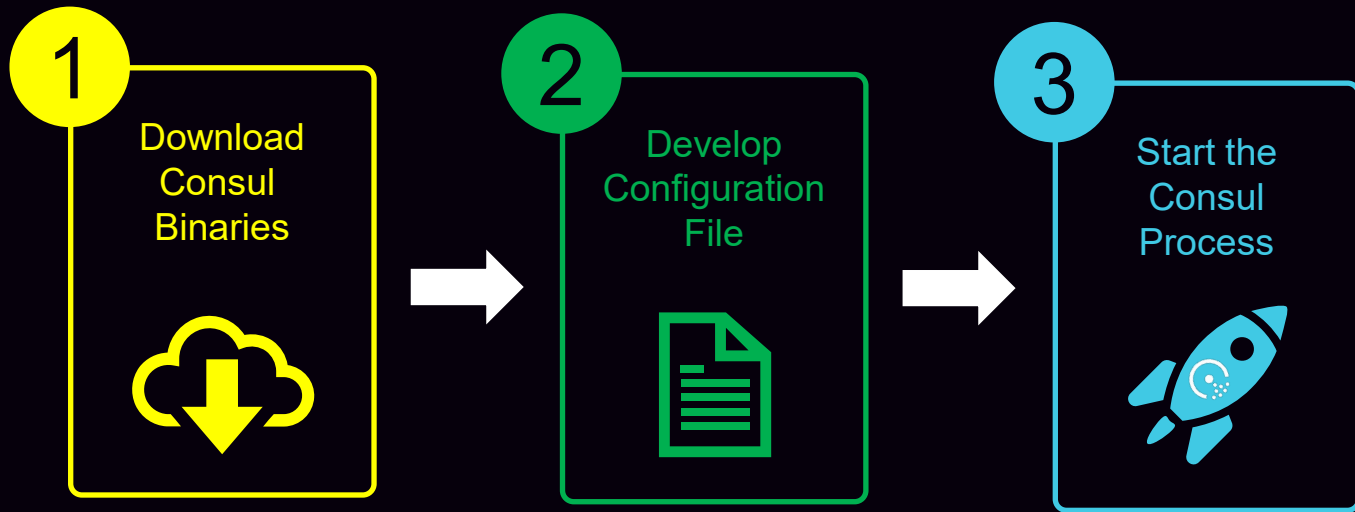
5

Difficulty Level



Starting the Consul Process

General Workflow to Start Consul



Starting the Consul Process

- Consul is started by running the `consul agent` command
 - `consul agent <flag>`
- The provided flags will dictate how Consul is configured

Terminal

```
$ consul agent -config-file=/opt/consul/config.hcl
```



Starting the Consul Process

- Consul agent is commonly started using a service manager
 - systemctl
 - Windows Service Manager
- Service manager will start the Consul agent using a command line
 - Command line can include flags or...
 - Configurations can be part of the configuration file (most common)



Starting the Consul Process

- Command-line using Consul command-line options:
 - `consul agent -<option> -<option> -<option>`

Terminal

```
$ consul agent -datacenter="aws" -bind="10.0.10.42" -data-dir=/opt/consul  
-encrypt=<key> -retry_join="10.0.10.64,10.4.23.98"
```



Starting the Consul Process

- Command-line using Consul configuration file
 - -config-file
 - -config-dir

Terminal

```
$ consul agent -config-file=/etc/consul/config.hcl
```



Starting the Consul Process

- Command line using configuration file
 - -config-file
 - single file → /etc/consul.d/config.hcl

Terminal

```
$ cd /etc/consul.d
$ ls
config.hcl
```

Service Manager

```
/usr/local/bin/consul agent -config-file=/etc/consul.d/config.hcl
```



Starting the Consul Process

- Command line using configuration directory
 - -config-dir
 - points to a directory → /etc/consul.d

Terminal

```
$ cd /etc/consul.d  
$ ls  
config.hcl  
metadata.hcl  
service.hcl
```

Helpful when you have
multiple files to be loaded

Service Manager

```
/usr/local/bin/consul agent -config-dir=/etc/consul.d/
```



Consul Server - Dev Mode

- Using `consul agent -dev` will start Consul in dev server mode

Terminal

```
$ consul agent -dev
```

- Useful for starting a Consul agent
 - All persistence options are turned off
 - Enables in-memory server
 - Connect is enabled (**will create a new root CA cert by default**)
 - gRPC port defaults to 8502



Never run Consul -dev mode in production



Manage the Consul Process

- Restarting the Consul process
 - Use the service manager

Terminal

```
$ systemctl restart consul
```

- Permanently removing the node
 - Gracefully remove node from Consul
 - Stop service

Terminal

```
$ consul leave  
$ systemctl stop consul
```



Manage the Consul Process

- Reloading the Consul configuration
 - Can modify certain configuration options

Terminal

```
$ consul reload
```

- Not all configuration options are reloadable (not an exhaustive list)
 - ACL Tokens
 - Checks
 - Log level
 - Node Metadata
 - Services
 - TLS Configuration
 - Watches



Interpret a Consul Agent Configuration

- Configuration file can be written in JSON or HCL
 - Defines the configuration for the Consul Agent (server & client)

Terminal

```
...  
  
"datacenter": "us-east-1",  
"client_addr": "0.0.0.0",  
"bind_addr": "10.11.11.11",  
"advertise_addr": "10.11.11.11",  
"bootstrap_expect": 5,  
"retry_join": ["provider=aws tag_key=Environment-Name tag_value=consul-cluster region=us-east-1"],  
"enable_syslog": true,  
"acl": {  
  "enabled": true,  
  "default_policy": "deny",  
}  
...
```



Interpret a Consul Agent Configuration

- Environment variables cannot be used to configure the Consul client
- Key Options in a SERVER configuration file:
 - `server` (**boolean**) – is this a server agent or not?
 - `datacenter` (**string**) – what datacenter to join
 - `node` (**string**) – unique name of agent (usually server name)
 - `join/retry_join/auto-join` (**string**) – what other servers/cluster to join
 - `client_addr/bind_addr/advertise_addr` (**string**) – what IP/interface to use for Consul communications
 - `log_level` (**string**) – level of logging (trace, debug, info, etc)
 - `encrypt` (**string**) – secret to use for encryption of Consul traffic (gossip)
 - `data-dir` (**string**) – provide a persistent directory for the agent to store state



Interpret a Consul Agent Configuration

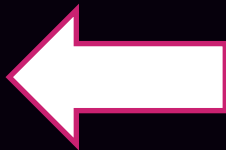
- Environment variables cannot be used to configure the Consul client
- Key Options in a SERVICE configuration file:
 - **name** (**string**) **<req>** – logical name of the service (web, app1, etc.)
 - **id** (**string**) – unique ID for this service – unique per agent (web-server-01, hcwebapp418, etc.)
 - **port** (**integer**) – what local port is the service running on? (80, 8080, 443)
 - **check** (**arguments**) – define arguments for health check



Configure Consul Network Addresses and Ports



eth0: 10.0.5.34
eth1: 10.0.3.88



Applications/Clients



Configure Consul Network Addresses and Ports

- DNS

- Port 8600 might work fine in your environment
- But others might lack the ability to send DNS traffic to a non-standard port (UDP 53)
- Ports below 1024 require to be run with root privileges
 - We do NOT want to run Consul as a root user
- We may need to set up forwarding using BIND or dnsmasq to forward requests received on 53 and forward to 8600

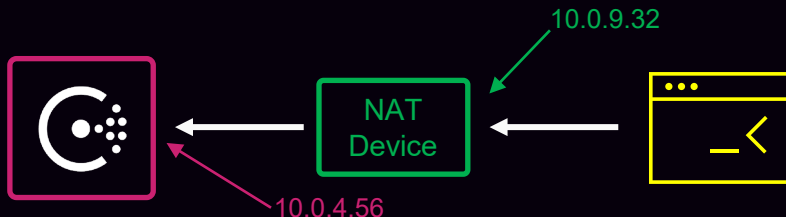


Configure Consul Network Addresses and Ports

- Consul API
 - **-bind** – interface that the Consul agent itself uses
 - **-advertise** – the interface that Consul tells other agents and clients to use when connecting to the local agent
- Useful for when Consul server agent nodes have multiple interfaces or if Consul is behind a NAT device

Consul Configuration

bind = 10.0.4.56
advertise = 10.0.9.32



Adding Servers

- Consul servers can “join” the cluster using multiple methods
- A Consul agent can join *any* node in the cluster
 - gossip will propagate the updated membership state across the cluster
- An agent that is already a member can join a different cluster
 - The two clusters will be merged into a single cluster



Adding Servers

- Multiple ways for an agent to join a cluster
 - Command line: `consul join <host>`

Terminal

```
$ consul join consul-node-a.example.com  
Successfully joined cluster by contacting 1 nodes.
```

IP Address works, too

- `<host>` can be any member of the cluster, client or server
- Generally used for testing or lab environment
- Manual join not recommended for production deployments (use agent config)



Adding Servers

- Multiple ways for an agent to join a cluster
 - Configuration file
 - `-join`
 - specify one or more agents to join (IPv4, IPv6, or hostnames)
 - If Consul is unable to join specified agents, agent startup will fail
 - `-retry_join`
 - specify one or more agents to join (IPv4, IPv6, or hostnames)
 - Will continue retrying until successful
 - Ideal for automated deployments or when agents may start random order



Adding Servers

- Multiple ways for an agent to join a cluster
 - Configuration file

Terminal

```
{  
  "bootstrap": false,  
  "bootstrap_expect": 3,  
  "server": true,  
  "retry_join": ["10.0.10.34", "10.0.11.72"]  
}
```

(agent join snippet)



Adding Servers

- Multiple ways for an agent to join a cluster
- Configuration file
 - Cloud Auto-join
 - Uses cloud meta-data to discover Consul nodes (tags)
 - AWS
 - Azure
 - GCP
 - Softlayer
 - Alibaba Cloud
 - Digital Ocean
 - Openstack
 - Scaleway
 - TencentCloud
 - Joyent Triton
 - vSphere
 - Packet
 - Linode
 - Kubernetes
- Requires credentials for authentication



Adding Servers

- Multiple ways for an agent to join a cluster
- Configuration file
 - Cloud Auto-join

Terminal

```
{  
  "bootstrap": false,  
  "bootstrap_expect": 3,  
  "server": true,  
  "retry_join": ["provider=aws tag_key=consul tag_value=true"],  
}
```



Removing Servers

- Command Line

- Consul leave triggers a graceful leave and shutdown
- It ensures that other nodes see the agent as “left” rather than “failed”

Terminal

```
$ consul leave  
Graceful leave complete
```

- For servers, a **consul leave** affects the raft peer-set, as Consul will reconfigure the cluster to have fewer servers



Listing Membership

- Determining the members of the cluster
 - Displays both servers and clients

Terminal

```
$ consul members
```

Node	Address	Status	Type	Build	Protocol	DC	Segment
consul-a	10.0.2.10:8301	alive	server	1.9.0	2	dc1	
consul-b	10.0.2.11:8301	alive	server	1.9.0	2	dc1	
web-app-01	10.0.8.9:8301	alive	client	1.8.6	2	dc1	



Deploy a Single Datacenter

Objective 2a: Start and manage the Consul process

Objective 2b: Interpret a Consul agent configuration

Objective 2c: Configure Consul network addresses and ports

Objective 2d: Describe and configure agent join and leave behaviors

1

2

3

4

5

Difficulty Level





END OF
SECTION

