

Consul



tutorialspoint

S I M P L Y E A S Y L E A R N I N G

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Consul is an important service discovery tool in the world of **Devops**. This tutorial covers in-depth working knowledge of Consul, its setup and deployment. This tutorial aims to help new user's setup consul, develop advanced knowledge on consul and learn some interesting projects built around consul. In the end, I hope the readers understand this tutorial and use consul for their daily work.

This tutorial will give you a quick start with Consul and make you comfortable with its various components.

Audience

This tutorial is prepared for the students, beginners as well as for intermediate Devops Practitioners to help them understand the basics to advanced concepts related to the Consul tool.

Prerequisites

Before you start doing practice with the examples given in this tutorial, it is being assumed that you already have a basic knowledge of Linux, Git, Golang, Docker and AWS (Amazon Web Services).

Copyright & Disclaimer

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents.....	ii
1. CONSUL – INTRODUCTION.....	1
Consul – Members and Agents.....	2
2. CONSUL – ARCHITECTURE.....	3
Raft Algorithm	4
Key Value Data.....	4
Types of Protocol	5
3. CONSUL – INSTALLATION.....	7
Installing Consul.....	7
Using the Command Line	8
Consul Template	9
4. CONSUL – WORKING WITH MICROSERVICES	13
Using Docker.....	13
Building Registrar for Service Discovery.....	15
Using rkt and Nomad	17
5. CONSUL – BOOTSTRAPPING & DNS.....	19
Automatic Bootstrapping.....	19
Manual Bootstrapping	20
Using DNS Forwarding	20
DNS Caching.....	23

6. CONSUL – QUERYING NODES.....	25
Using Dig.....	25
Using the Monitor Command.....	26
Using the Watch Command	26
By Registering External Services.....	28
7. CONSUL – FAILOVER EVENTS	30
Single Cluster Failure.....	30
Jepsen Testing.....	33
Multiple Cluster Failure.....	33
Taking Snapshots	36
8. CONSUL – USING CONSUL UI	39
Consul UI Setup.....	39
Features of Consul UI	40
9. CONSUL – USING CONSUL ON AWS	44
Features of AWS	44
AWS Deployment.....	44
Using the AWS	45

1. Consul – Introduction

Consul is a Hashicorp based tool for discovering and configuring a variety of different services in your infrastructure. It is based and built on Golang. One of the core reasons to build Consul was to maintain the services present in the distributed systems. Some of the significant features that Consul provides are as follows.

- **Service Discovery:** Using either DNS or HTTP, applications can easily find the services they depend upon.
- **Health Check Status:** It can provide any number of health checks. It is used by the service discovery components to route traffic away from unhealthy hosts.
- **Key/Value Store:** It can make use of Consul's hierarchical key/value store for any number of purposes, including dynamic configuration, feature flagging, coordination, leader election, etc.
- **Multi Datacenter Deployment:** Consul supports multiple datacenters. It is used for building additional layers of abstraction to grow to multiple regions.
- **Web UI:** Consul provides its users a beautiful web interface using which it can be easy to use and manage all of the features in consul.






Service Discovery

Service discovery is one of the most important feature of Consul. It is defined as the detection of different services and network protocols using which a service is found. The usage of service discovery comes in as a boon for distributed systems. This is one of the main problems, which are faced by today's large-scale industries with the advancement of distributed systems in their environment.

Comparison with Etcd and Zookeeper

When we look at other service discovery tools in this domain, we have two popular options. Some major players in the software industry have been using it in the past. These tools are **Etcd** and **Zookeeper**.

Let us consider the following table for comparing different aspects of each tool. We will also understand what each one of them uses internally.

Properties	Consul	Etcd	Zookeeper
User Interface	Available		
RPC	Available	Available	
Health Check	HTTP API	HTTP API	TCP
Key Value	3 Consistency modes	Good Consistency	Strong Consistency
Token System	Available		
Language	Golang	Golang	Java

Consul – Members and Agents

Consul members can be defined as the list of different agents and server modes using which a consul cluster is deployed. Consul provides us with a command line feature using which we can easily list all the agents associated with consul.

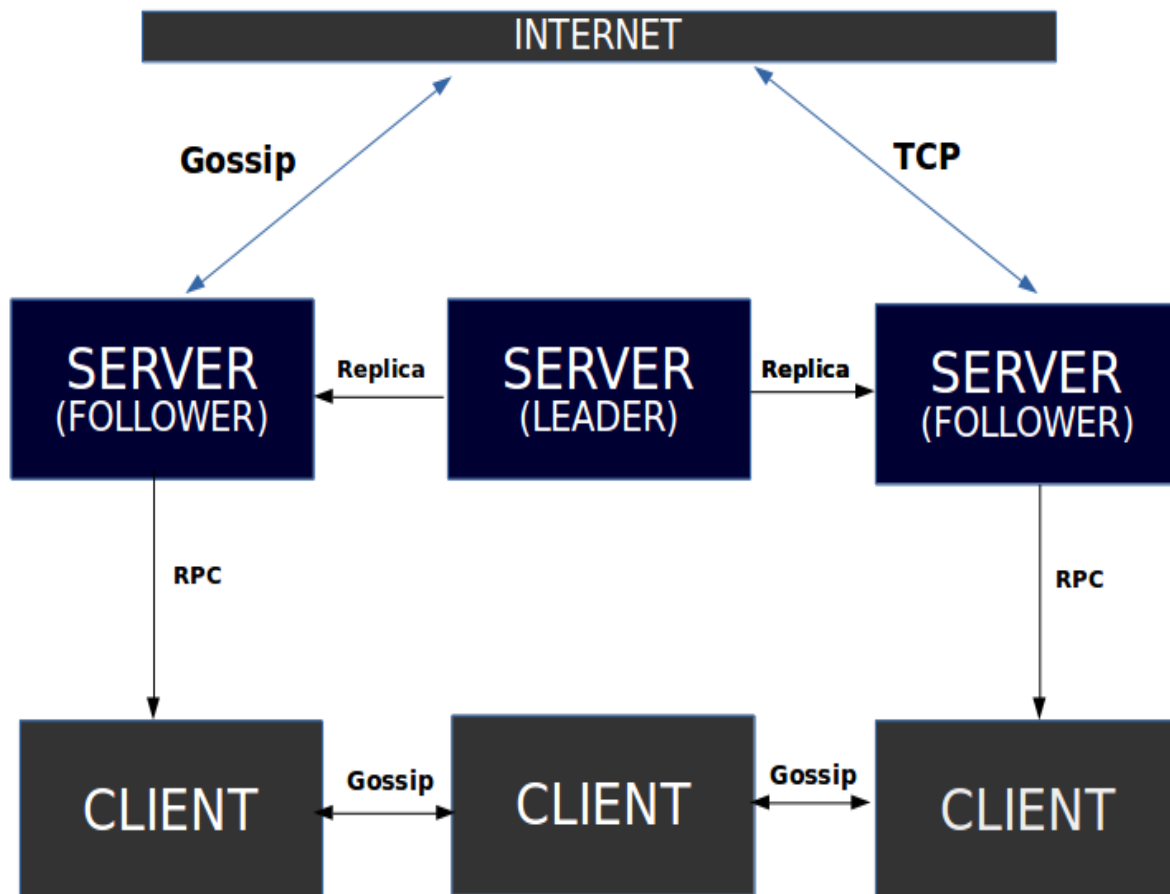
Consul agent is the core process of Consul. The agent maintains membership information, registers services, runs checks, responds to queries, etc. Any agent can be run in one of two modes: **Client** or **Server**. These two modes can be used according to their role as decided when using consul. The consul agent helps by providing us information, which is listed below.

- **Node name:** This is the hostname of the machine.
- **Datacenter:** The datacenter in which the agent is configured to run. Each node must be configured to report to its datacenter.
- **Server:** It indicates whether the agent is running in server or client mode. Server nodes participate in the consensus quorum, storing cluster state and handling queries.
- **Client Addr:** It is the address used for client interfaces by the agent. It includes the ports for the HTTP, DNS, and RPC interfaces.
- **Cluster Addr:** It is the address and the set of ports used for communication between Consul Agents in a cluster. This address must be reachable by all other nodes.

In the next chapter, we will understand the architecture for Consul.

2. Consul – Architecture

The architecture diagram for consul working in one datacenter can be best described as shown below:



As we can observe, there are three different servers, which are managed by Consul. The working architecture works by the using raft algorithm, which helps us in electing a leader out of the three different servers. These servers are then labelled according to the tags such as **Follower** and **Leader**. As the name suggests, the follower is responsible for following the decisions of the leader. All these three servers are further connected with each other for any communication.

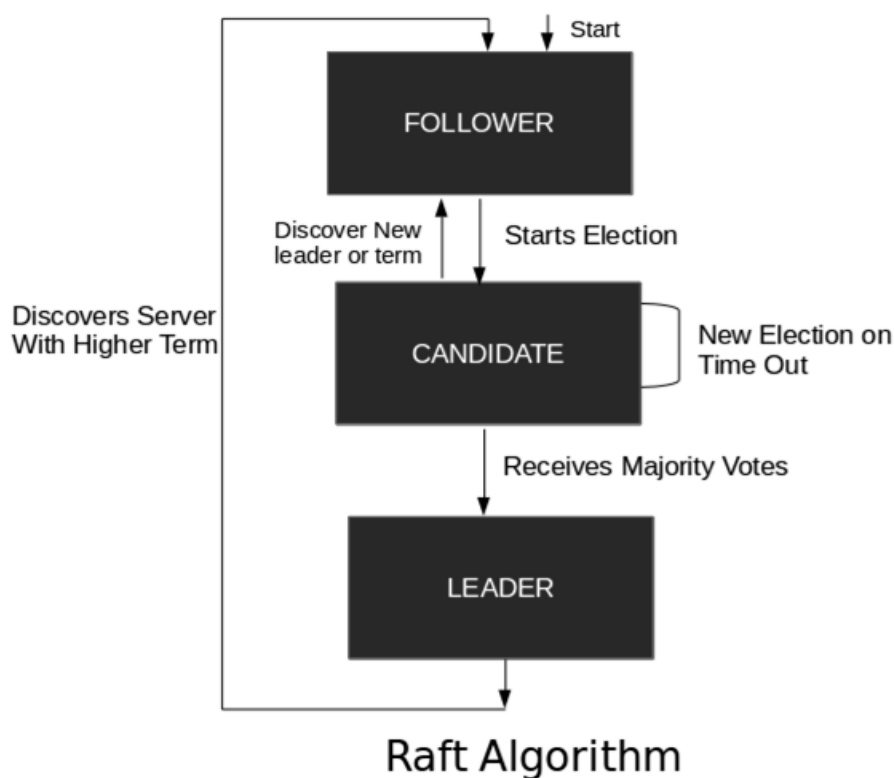
Each server interacts with its own client using the concept of RPC. The Communication between the Clients is possible due to **Gossip Protocol** as mentioned below. The Communication with the internet facility can be made available using TCP or gossip method of communication. This communication is in direct contact with any of the three servers.

Raft Algorithm

Raft is a consensus algorithm for managing a replicated log. It relies on the principle of **CAP Theorem**, which states that in the presence of a network partition, one has to choose between consistency and availability. Not all the three fundamentals of the CAP Theorem can be achieved at any given point of time. One has to tradeoff for any two of them at the best.

A **Raft Cluster** contains several servers, usually in the odd number count. For example, if we have five servers, it will allow the system to tolerate two failures. At any given time, each server is in one of the three states: **Leader**, **Follower**, or **Candidate**. In a normal operation, there is exactly one leader and all of the other servers are followers. These followers are in a passive state, i.e. they issue no requests on their own, but simply respond to requests from leaders and the candidate.

The following illustration describes the workflow model using which the raft algorithm works:



Key Value Data

Since the Consul's version 0.7.1, there has been an introduction of separate key value data. The KV command is used to interact with the Consul's key-value store via the command line. It exposes top-level commands for **Inserting**, **Updating**, **Reading** and **Deleting** from the store. To get the Key/Value object store, we call the KV method available for the consul client –

```
kv := consul.KV()
```


The **KVPair Structure** is used to represent a single key/value entry. We can view the structure of Consul KV Pair in the following program.

```
type KVPair struct {
    Key          string
    CreateIndex  uint64
    ModifyIndex  uint64
    LockIndex    uint64
    Flags        uint64
    Value        []byte
    Session      string
}
```

Here, the various structures mentioned in the above code can be defined as follows:

- **Key** – It is a slash URL name. For example – sites/1/domain.
- **CreateIndex** – Index number assigned when the key was first created.
- **ModifyIndex** – Index number assigned when the key was last updated.
- **LockIndex** – Index number created when a new lock acquired on the key/value entry
- **Flags** – It can be used by the app to set the custom value.
- **Value** – It is a byte array of maximum 512kb.
- **Session** – It can be set after creating a session object.

Types of Protocol

There are two types of protocol in Consul, which are called as –

- Consensus Protocol and
- Gossip Protocol

Let us now understand them in detail.

Consensus Protocol

Consensus protocol is used by Consul to provide Consistency as described by the CAP Theorem. This protocol is based on the Raft Algorithm. When implementing Consensus protocol, the Raft Algorithm is used where raft nodes are always in any one of the three states: Follower, Candidate or Leader.

Gossip Protocol

The gossip protocol can be used to manage membership, send and receive messages across the cluster. In consul, the usage of gossip protocol occurs in two ways, **WAN** (Wireless Area Network) and **LAN** (Local Area Network). There are three known libraries, which can implement a Gossip Algorithm to discover nodes in a peer-to-peer network:

- **teknek-gossip** – It works with UDP and is written in Java.
- **gossip-python** – It utilizes the TCP stack and it is possible to share data via the constructed network as well.
- **Smudge** – It is written in Go and uses UDP to exchange status information.

Gossip protocols have also been used for achieving and maintaining a distributed database consistency or with other types of data in consistent states, counting the number of nodes in a network of unknown size, spreading news robustly, organizing nodes, etc.

Remote Procedure Calls

The RPC can be denoted as the short form for Remote Procedure Calls. It is a protocol that one program uses to request a service from another program. This protocol can be located in another computer on a network without having to acknowledge the networking details.

The real beauty of using RPC in Consul is that, it helps us avoid the latency issues which most of the discovery service tools did have some time ago. Before RPC, Consul used to have only **TCP** and **UDP** based connections, which were good with most systems, but not in the case of distributed systems. RPC solves such problems by reducing the time-period of transfer of packet information from one place to another. In this area, [GRPC](#) by Google is a great tool to look forward in case one wishes to observe benchmarks and compare performance.

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>