



# KPLABS Course

HashiCorp Certified: Consul Associate

Security

ISSUED BY

Zeal

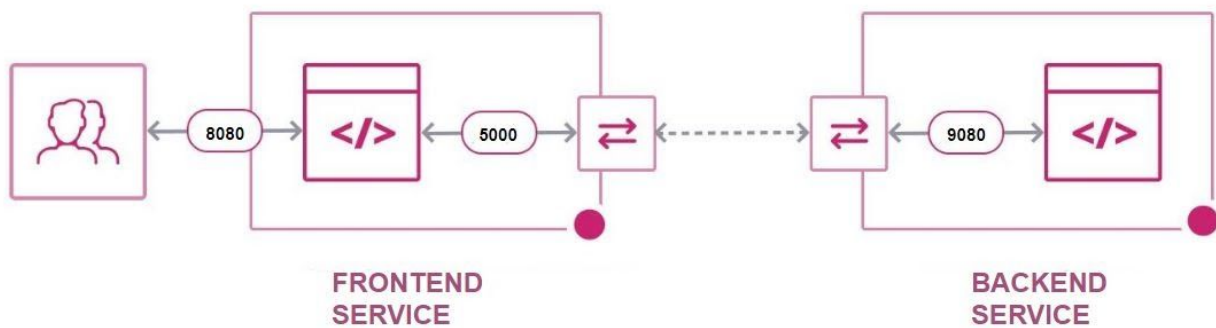
REPRESENTATIVE

[instructors@kplabs.in](mailto:instructors@kplabs.in)

# Module 1: Overview of Consul Connect

## 1.1 Overview of Consul Connect

Consul Connect provides service-to-service connection authorization and encryption using mutual Transport Layer Security (TLS).



## 1.2 Sample Use-Case

Frontend Service wants to communicate with Backend Service.

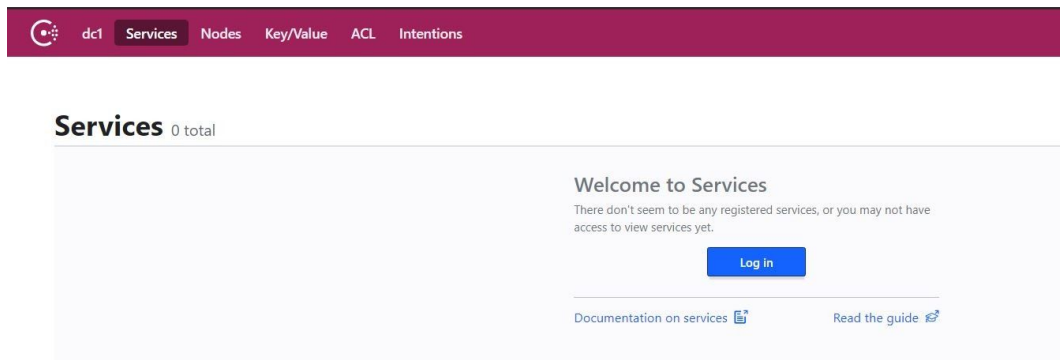
Additional Requirements:

- Should be over TLS (encrypted communication).
- Should have required level of authorization.

## Module 2: Consul ACLs

One of the biggest challenges as of now in Consul is the lack of Authentication.

To overcome this, Consul uses Access Control Lists (ACLs) to secure access to the UI, API, CLI, service communications, and agent communications.



### Step 1: Enable ACL in Consul

To enable ACLs, add the following ACL parameters to the agent's configuration file and then restart the Consul service.

```
acl = {  
  enabled = true  
  default_policy = "deny"  
  enable_token_persistence = true  
}
```

### Step 2: Create a Bootstrap Token

It is important to have one token with unrestricted privileges in case of emergencies.

This will also allow you to quickly get started.

```
[root@consul-01 consul.d]# consul acl bootstrap
AccessorID:      c9e3a6dd-5b4a-de5f-fe6b-dfe8e01bb031
SecretID:        f23a368f-5b75-b832-7733-591b7d8eef6c
Description:     Bootstrap Token (Global Management)
Local:           false
Create Time:     2020-11-14 12:58:55.315441221 +0000 UTC
Policies:        00000000-0000-0000-0000-000000000001 - global-management
```

### Important Note

Using the token on the command line with the `-token` flag is not recommended, instead, you can set it as an environment variable once.

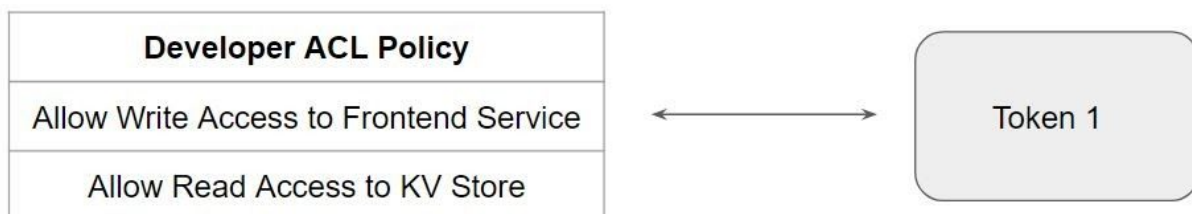
CONSUL\_HTTP\_TOKEN

## Module 3: Understanding ACL Roles

### 3.1 ACL System in Consul

The ACL is Capability-based, relying on tokens that are associated with policies to determine which fine-grained rules can be applied.

There are two primary components of the ACL system: ACL Policies & ACL Tokens



### 3.2 Overview of Rules

Rules are composed of a resource, a segment (for some resource areas), and a policy disposition.



### 3.3 Actions for Rules

Following actions can be determined while writing a rule:

Action	Description
read	allow the resource to be read but not modified.
write	allow the resource to be read and modified.
deny	do not allow the resource to be read or modified.
list	allows access to all the keys under a segment in the Consul KV

### 3.4 Scope for ACL Rules

Following resources are available for constructing rules:

Resource	Description
acl	Operations for managing the ACL system ACL API
agent	Utility operations in the Agent API, other than service and check registration
event	Listing and firing events in the Event API
key	Key/value store operations in the KV Store API
keyring	Keyring operations in the Keyring API
node	Node-level catalog operations in the Catalog API, Health API, Prepared Query API, Network Coordinate API, and Agent API
operator	Cluster-level operations in the Operator API, other than the Keyring API
etc	Includes query, service and session

## Module 4: Anonymous Tokens


The anonymous token is used when a request is made to Consul without specifying a bearer token.

The anonymous token's description and policies may be updated but Consul will prevent this token's deletion.

[← All Tokens](#)

### Edit Token

AccessorID 00000000-0000-0000-0000-000000000002

Token  anonymous

Scope global

## Module 5: Enabling ACLs on Agent

When you enable ACLs with a “deny” based approach, by default requests will be denied.

This applies even at the agent level.

```
consul[71676]: 2020-11-16T07:43:49.921Z [WARN] agent: Coordinate update blocked by A
consul[71676]: 2020-11-16T07:44:09.118Z [WARN] agent: Coordinate update blocked by A
consul[71676]: 2020-11-16T07:44:21.626Z [WARN] agent: Node info update blocked by AC
consul[71676]: 2020-11-16T07:44:29.209Z [WARN] agent: Coordinate update blocked by A
consul[71676]: 2020-11-16T07:44:51.352Z [WARN] agent: Coordinate update blocked by A
consul[71676]: 2020-11-16T07:45:13.343Z [WARN] agent: Coordinate update blocked by A
```

### Step 1: Create Policy for Agent Token

Create the following policy for agent token

```
node_prefix "" {
  policy = "write"
}
service_prefix "" {
  policy = "read"
}
```

### Step 2: Add token in Configuration

Add the agent token within the configuration

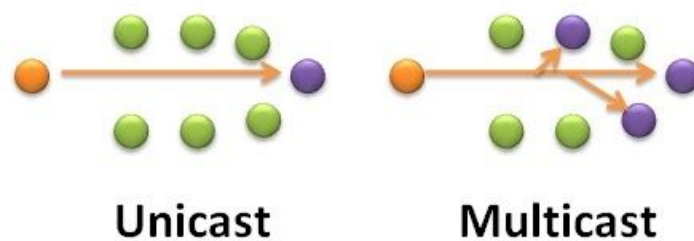
```
acl = {
  enabled = true
  default_policy = "deny"
  enable_token_persistence = true
  tokens {
    "agent" = "34c522d7-bce0-c27d-fc2e-69dc83e15487"
  }
}
```

# Module 6: Overview of Gossip Protocol

## 6.1 Overview of Unicast and MultiCast

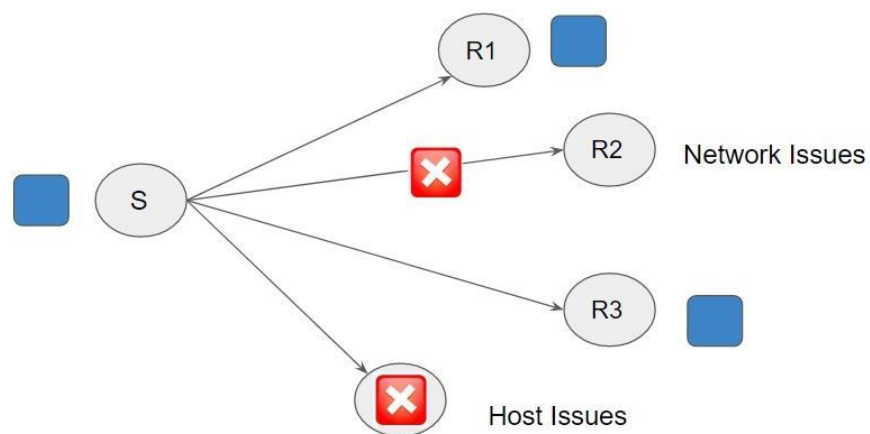
A Unicast transmission/stream sends IP packets to a single recipient on a network.

Multicast transmission sends IP packets to a group of hosts on a network



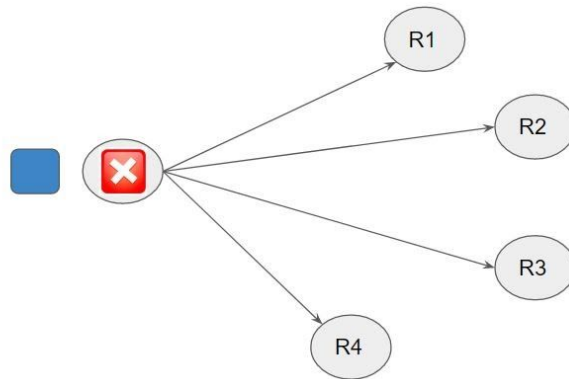
## 6.2 Multi-Cast Challenges

Suppose a sender wants to send a message to a group of hosts. The hosts might not receive the message due to various issues like network connectivity, the host being down, and so on.



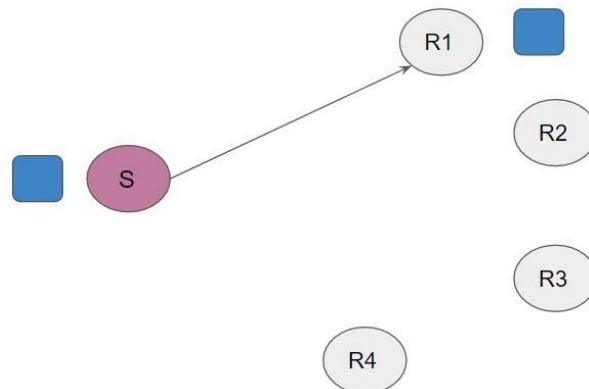


It can also be possible that the sender of the message went down.

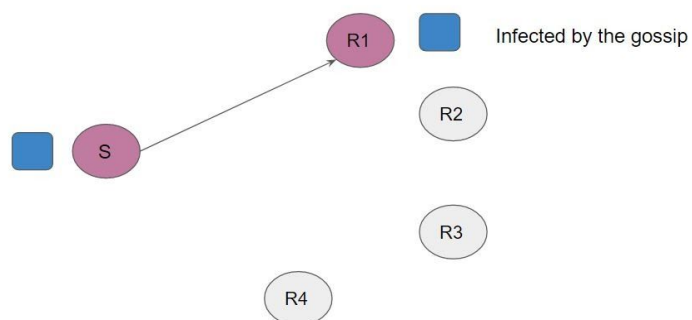


### 6.3 Gossip Protocol

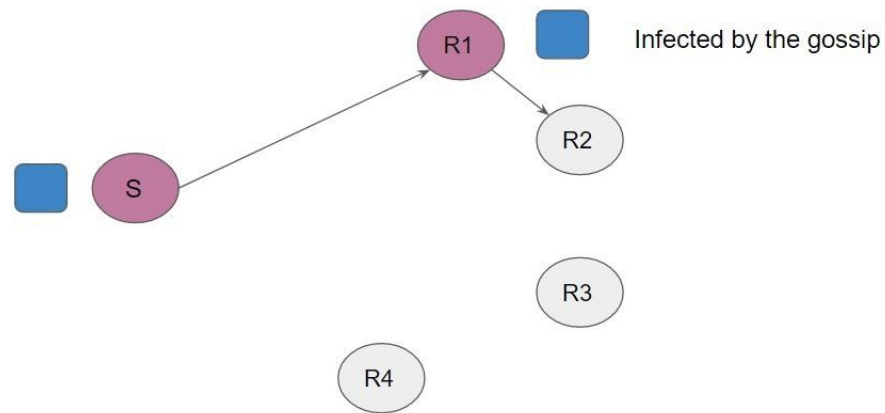
Data is periodically transmitted to random targets. In the below case, the sender has sent the message to R1.



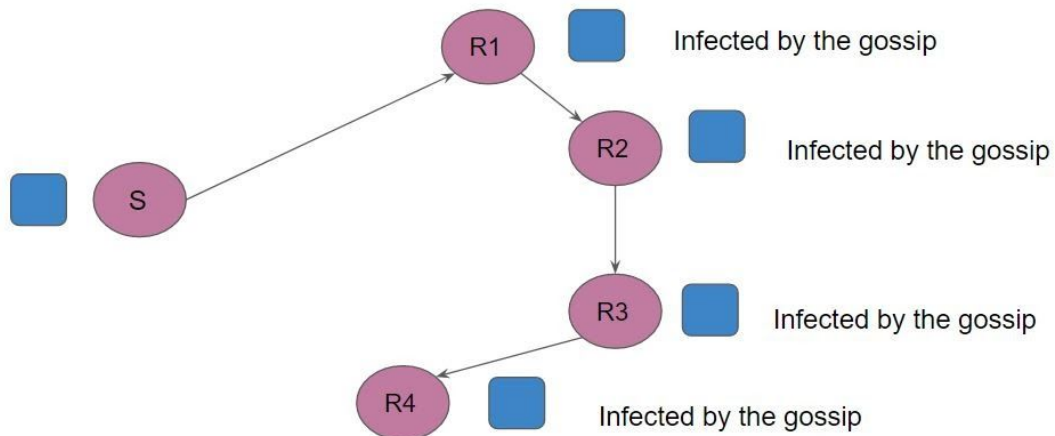
Once R1 receives the message, it is referred to as infected by the gossip.



Once R1 receives the message, it chooses random targets and sends out copies of the message.



In the final stage, all the hosts will have the data.

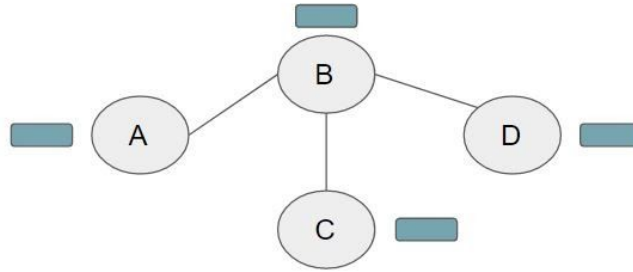


## 6.4 Understanding Gossip Protocol

A gossip protocol is a procedure or process of computer peer-to-peer communication

The amount of overhead involved is not that high when compared to a non-gossip scenario.

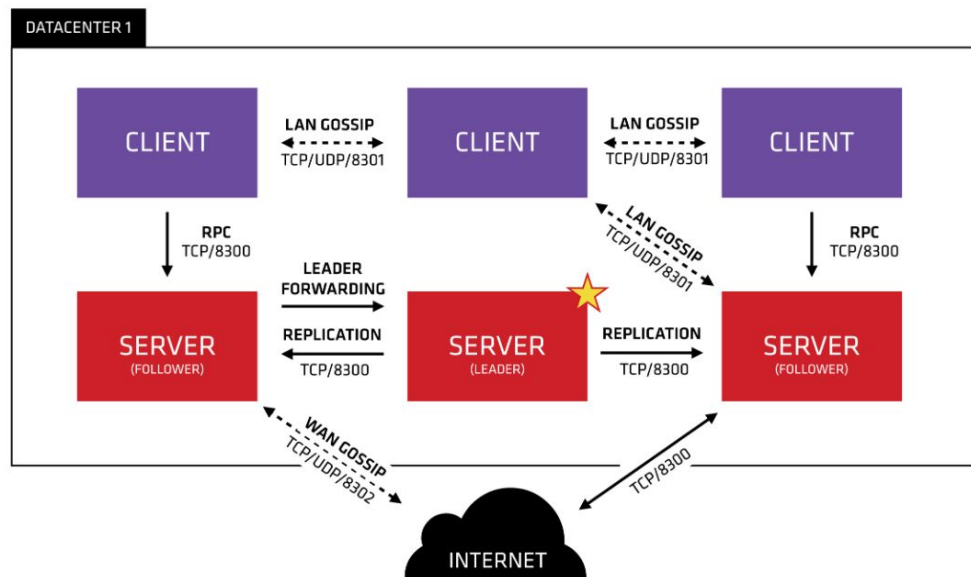
Some ad-hoc networks have no central registry and the only way to spread common data is to rely on each member to pass it along to their neighbors.



## Module 7: Gossip Encryption in Consul

### 7.1 Consul and Gossip Protocol

Consul uses a gossip protocol to manage membership and broadcast messages to the cluster



### 7.2 Challenge with Plain Text Data

By default, the data would be in plaintext and it is possible to capture the network packets and retrieve the information.

```

23:54:56.295177 IP (tos 0x0, ttl 62, id 21957, offset 0, flags [DF], proto UDP (17), length 180)
134.209.154.246.amberon > consul-01.amberon: [udp sum ok] UDP, length 152
0x0000: 4500 00b4 55c5 4000 3e11 93ca 86d1 9af6 E...U.@.>.....
0x0010: 9f41 91a0 206d 206d 00a0 e9b7 0282 a750 .A...m.m.....P
0x0020: 6179 6c6f 6164 da00 8201 84aa 4164 6a75 ayload.....Adju
0x0030: 7374 6d65 6e74 cbbf 1c6a 34cb fc78 dba5 stment...j4..x..
0x0040: 4572 726f 72cb 3fb2 6863 53ca 9a97 a648 Error?.hcS....H
0x0050: 6569 6768 74cb 3f1f 3860 b1cf e438 a356 eight?.8`...8.V
0x0060: 6563 98cb bf0d cd0a 4ee7 df3f cb3f 1460 ec.....N..?..`
0x0070: 6fea 7b72 24cb bf2e 6d61 3029 00b0 cbbf o.{r$...ma0)....
0x0080: 058f 3c60 5d43 94cb 3f12 a934 98b5 9dc3 ..<`]C...?.4....
0x0090: cbbf 31f6 ddb9 045e e6cb bf2b 9cda 8468 ..1....^....+...h
0x00a0: 2181 cb3f 0fb7 332f 826e c8a5 5365 714e !..?..3/.n..SeqN
0x00b0: 6fcd 031d o...

```

### 7.3 Enabling Encryption

As part of security, it is important to enable gossip encryption.

Two steps:

- Generate a cryptographic key.
- Add key within the configuration file.

#### Step 1: Generate Cryptographic Key

We can easily generate a key with the consul keygen command

```

[root@consul-01 ~]# consul keygen
6k527qHgwLNPdai7Yuu9nmKr11Z6je+4H6JQ7NZwjYg=

```

#### Step 2: Add Key in Configuration File

Add the encryption key parameter to the agent configuration file

```

[root@consul-01 ~]# cat consul.hcl
data_dir = "/etc/consul.d/consul-dir"
bind_addr = "159.65.145.160"
client_addr = "0.0.0.0"
bootstrap_expect = 1
node_name = "consul-server"
ui = true
server = true
encrypt = "tpnV5YZa56x9Gc0nC40b/nPdcG2pus3xZei/oMhFHow="

```

## 7.4 Configuring Gossip for Existing Datacenter

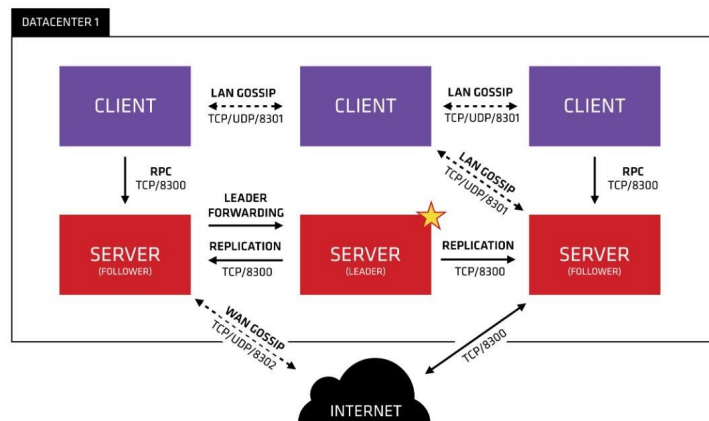
Gossip encryption can also be enabled on existing data centers but requires several extra steps.

The additional configuration of the agent configuration parameters, `encrypt_verify_incoming` and `encrypt_verify_outgoing` is necessary.

```
data_dir = "/etc/consul.d/consul-dir"
bind_addr = "159.65.145.160"
client_addr = "0.0.0.0"
bootstrap_expect = 1
node_name = "consul-server"
ui = true
server = true
encrypt = "tpnV5YZa56x9Gc0nC40b/nPdcG2pus3xZei/oMhFHow="
encrypt_verify_incoming = true,
encrypt_verify_outgoing = true
```

## 7.5 Important Note

- TCP and UDP protocol can be used for Gossip.
- Port Number: 8301



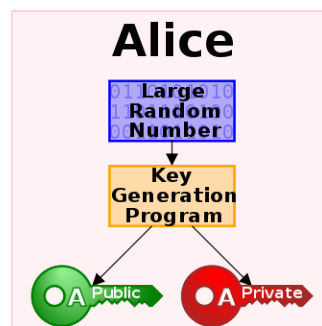
# Module 8: Introduction to Asymmetric Key Encryption

Asymmetric cryptography uses public and private keys to encrypt and decrypt data.

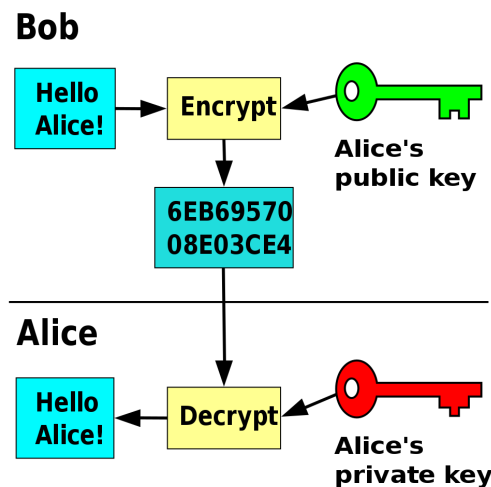
One key in the pair can be shared with everyone; it is called the public key. The other key in the pair is kept secret; it is called the private key.

Either of the keys can be used to encrypt a message; the opposite key from the one used to encrypt the message is used for decryption.

## Step 1: Generation of Keys



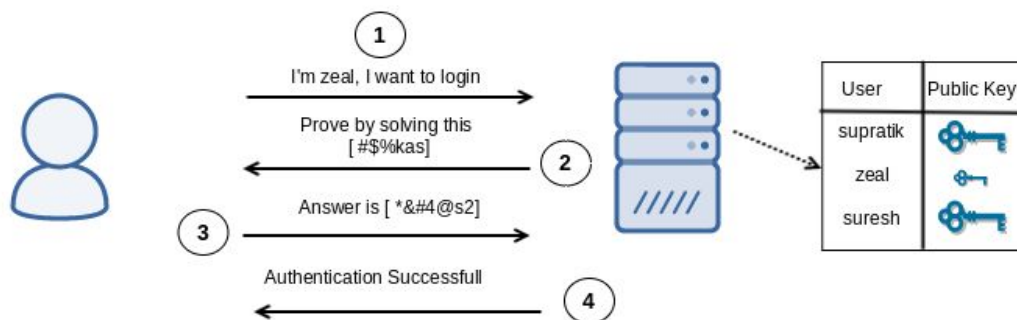
## Step 2: Encryption and Decryption



## 8.1 Use-Case of Asymmetric Key Encryption

### Step 1

User zeal wants to log in to the server. Since the server uses a public key authentication, instead of taking the password from the user, the server will verify if the User claiming to be zeal actually holds the right private key.



### Step 2

The server creates a simple challenge,  $2+3=?$  and encrypts this challenge with the Public Key of the User and sends it back to the User. The challenge is sent in an encrypted format.

### Step 3:

Since the user zeal holds the associated private key, he will be able to decrypt the message and compute the answer, which would be 5. Then, he will encrypt the message with the private key and send it back to the server.

### Step 4:

The server decrypts the message with the user's Public Key and checks if the answer is correct. If yes, then the server will send an Authentication Successful message and the user will be able to log in.

## 8.2 Protocols

Because of the advantage that it offers, Asymmetric key encryption is used by a variety of protocols.

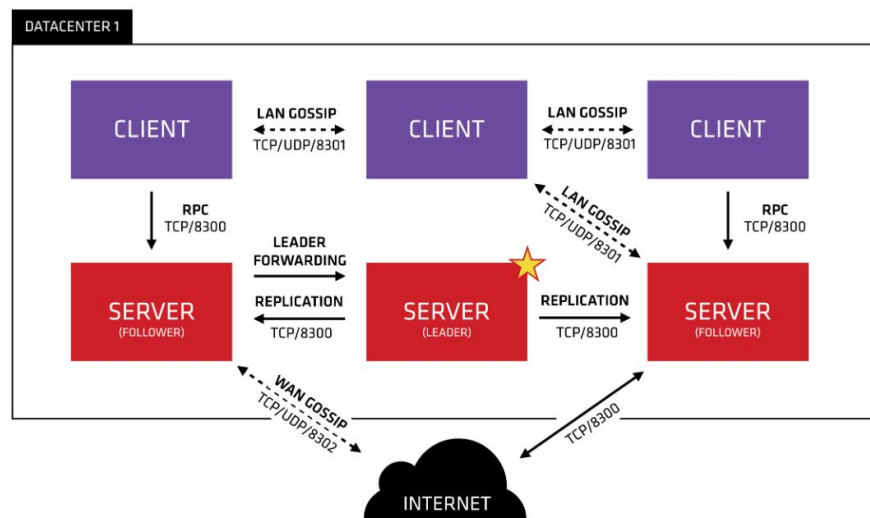
Some of these include:

- PGP
- SSH
- Bitcoin
- TLS
- S/MIME

## Module 9: RPC Encryption with TLS

### 9.1 Consul & RPC

Consul Client and Server communicate over RPC on port 8300.



### 9.2 Challenges with Plain Text Data

By default, the data would be in plaintext and it is possible to capture the network packets and retrieve the information.



```

10:51:50.059116 IP (tos 0x0, ttl 64, id 17638, offset 0, flags [DF], proto TCP (6),
    consul-01.tmi > consul-01.55355: Flags [P.], cksum 0x623a (incorrect -> 0x185c),
    ions [nop,nop,TS val 3504529536 ecr 3504529536], length 80
    0x0000: 4500 0084 44e6 4000 4006 93ca 9f41 91a0 E...D.@.@...A..
    0x0010: 9f41 91a0 206c d83b 1405 92eb 7efa 9eef .A...l.;....~...
    0x0020: 8018 0156 623a 0000 0101 080a d0e2 e080 ...Vb:.....
    0x0030: d0e2 e080 83a5 4572 726f 72a0 a353 6571 .....Error..Seq
    0x0040: 12ad 5365 7276 6963 654d 6574 686f 64b0 ..ServiceMethod.
    0x0050: 5374 6174 7573 2e52 6166 7453 7461 7473 Status.RaftStats
    0x0060: 83ab 4c61 7374 436f 6e74 6163 74a1 30a9 ..LastContact.0.
    0x0070: 4c61 7374 496e 6465 7867 a84c 6173 7454 LastIndexg.LastT

```

### 9.3 Enabling Encryption

As part of security, it is important to enable RPC encryption.

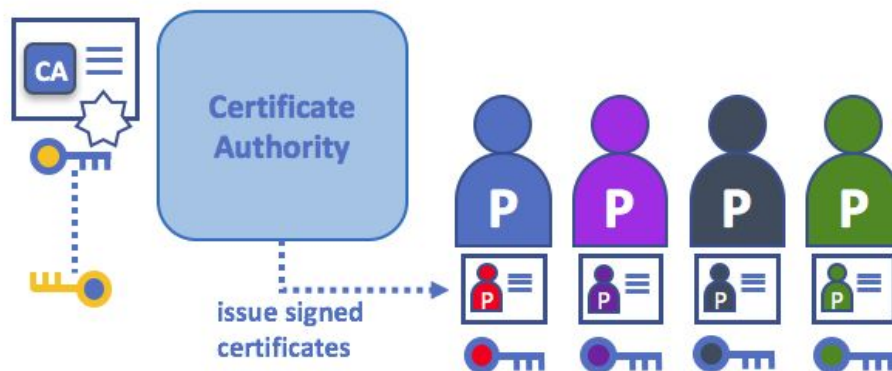
Three steps:

- Initialize In-Built CA
- Create Server Certificates.
- Configure Servers and Clients.

### 9.4 Certificate Authority

Certificate Authority is an entity that issues digital certificates.

The key part is that both the receiver and the sender trusts the CA.



## 9.5 Overview Steps Involved

### Step 1: Initialize Built-In CA

You can use the in-built CA provided by the Consul or make use of 3rd party private CAs.

```
[root@consul-01 consul.d]# consul tls ca create  
==> Saved consul-agent-ca.pem  
==> Saved consul-agent-ca-key.pem
```

### Step 2: Create Server Certificates

You can create the server certificates easily using in-built CA.

```
[root@consul-01 consul.d]# consul tls cert create -server  
==> WARNING: Server Certificates grants authority to become a  
server and access all state in the cluster including root keys  
and all ACL tokens. Do not distribute them to production hosts  
that are not server nodes. Store them as securely as CA keys.  
==> Using consul-agent-ca.pem and consul-agent-ca-key.pem  
==> Saved dc1-server-consul-0.pem  
==> Saved dc1-server-consul-0-key.pem
```

### Step 3: Configure Server & Clients

Add appropriate configuration parameters within the configuration file to make use of certificates.

```
verify_incoming = true,  
verify_outgoing = true,  
verify_server_hostname = true,  
ca_file = "consul-agent-ca.pem",  
cert_file = "/etc/consul.d/dc1-server-consul-0.pem",  
key_file = "/etc/consul.d/dc1-server-consul-0-key.pem",  
auto_encrypt {  
  allow_tls = true  
}
```

Server Configuration

```
verify_incoming = false,  
verify_outgoing = true,  
verify_server_hostname = true,  
ca_file = "consul-agent-ca.pem",  
auto_encrypt = {  
  tls = true  
}
```

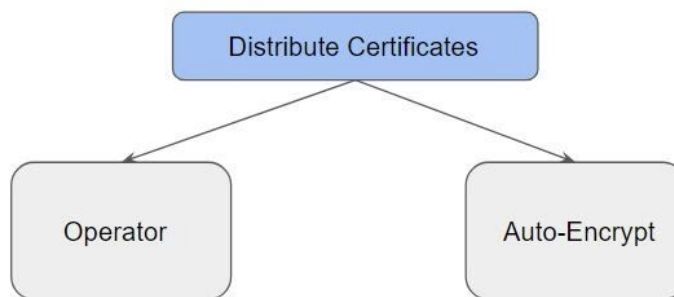
Client Configuration

## 9.6 Methods for Distributing Certificates

There are two methods for distributing client certificates: operator and auto encryption

With auto-encryption, you can configure the Consul servers to automatically distribute certificates to the clients.

The operator method is recommended if you need to use a third-party CA



## **Module 10: Overview of API**

API stands for an application programming interface.

It is generally used for inter-communication between multiple software.

With the API, the exact structure of request and response is documented upfront and is likely to remain the same throughout time.

### Use-Case

James wants to build a weather report application. Since it needs a weather report for all countries, he wonders where he can get all the data from. OpenWeatherMap has all this data and thus James decides to integrate his application fetch data from OpenWeatherMap database.

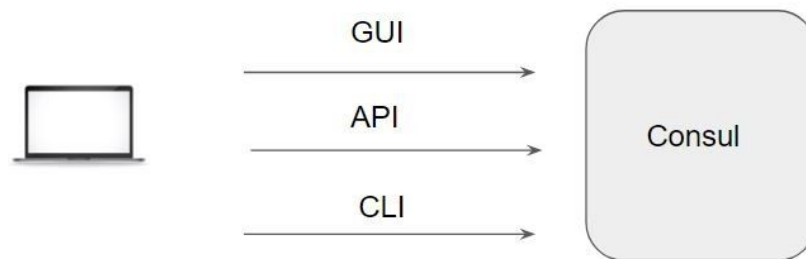
Now question is, how will he parse all this data?

# Module 11: HTTP API in Consul

## 11.1 Overview of Consul Interface

There are multiple ways to connect with Consul:

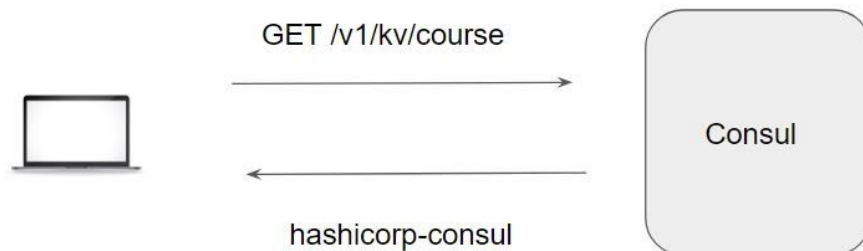
GUI, API as well as CLI



## 11.2 HTTP API

The main interface to Consul is a RESTful HTTP API.

All API routes are prefixed with /v1/



## 11.3 Important Pointers

Depending on the resource on which operation needs to be performed, the endpoint changes.

The API documentation provides extensive information about the same.

By default, the output of all HTTP API requests is minimized JSON. If the client passes pretty on the query string, formatted JSON will be returned.

## 11.4 Authentication

When authentication is enabled, a Consul token should be provided to API requests using the X-Consul-Token header or with the Bearer scheme in the authorization header.

```
curl \
  --header "X-Consul-Token: <consul token>" \
  http://127.0.0.1:8500/v1/agent/members
```

```
curl \
  --header "Authorization: Bearer <consul token>" \
  http://127.0.0.1:8500/v1/agent/members
```