SALT SYNDIC

- Phần cơ bản nhất hoặc điển hình cho mô hình Salt bao gồm 1 Master node controlling 1 nhóm các Minion nodes.

- 1 kiểu Node trung gian, gọi là Syndic, sử dụng để tăng tính linh hoạt và tăng khả năng mở rộng mô hình Salt thay vì chỉ sử dụng mô hình cơ bản.

- Syndic node có thể được coi là 1 node minion đặc biệt. Syndic node bao gồm salt-syndic daemon và salt-master daemon chạy trên cùng 1 hệ thống. Salt-master daemon chạy trên Syndic node kiểm soát group minion tại tầng dười, salt-syndic daemon để kết nối master node tại tầng cao, đôi khi được gọi là Master of Master

- Salt-syndic daemon chuyển tiếp các publication và event giữa Master node và local salt-master daemon. Nó cho phép master node điều khiển các Minion node được kết nối tới salt-master daemon đang chạy trên Syndic node.

CONFIGURING THE SYNDIC

- Để setup Salt Syndic, cần phải các hình syndic node và master node trên cùng 1 hệ thống. Nếu Master node được lưu tại 10.10.0.1, config sẽ như sau:

Tại syndic node:

# /etc/salt/master

syndic\_master: 10.10.0.1 # may be either an IP address or a hostname

và

# /etc/salt/minion

# id is shared by the salt-syndic daemon and a possible salt-minion daemon

# on the Syndic node

id: my\_syndic

Tại master node:

# /etc/salt/master

order\_masters: True

- “syndic\_master” option sử dụng để nói với Syndic node sẽ tìm Master ở đâu giống như cách “master” option tell minions node nơi tìm Master node.

- “id” option sẽ được sử dụng bởi “salt-syndic” daemon để xác định với master node. Nếu nó ko được set, giá trị hostname hoặc ip address sẽ được sử dụng.

- “order\_masters” option config Master node gửi thêm các thông tin cần thiết để Syndic nodes kết nối trực tiếp tới nó.

Note:

+ Mỗi syndic sẽ cung cấp 1 file\_root riêng và files sẽ ko tự động gửi về Master node

CONFIGURING THE SYNDIC WITH MULTIMASTER

- Syndic với Multimaster cho phép kết nối syndic tới nhiều Master, cung cấp thêm 1 tầng dự phòng trong syndic config.

- Higher level master cần được cấu hình trong multimaster config.

- Trên syndic, “syndic\_master” option được lưu tại list các higher level master.

- Vì mỗi syndic được kết nối tới mỗi master, jobs gửi từ bát kỳ master sẽ được forward tới các minion đã được kết nối tới mỗi syndic. Nếu giá trị “master\_id” được set trong master config trên higher level master, kết quả của job sẽ được trả lại master tạo ra request.

- Event/jobs ko có master\_id sẽ được trả lại bất kỳ master nào có sẵn.

RUNNING THE SYNDIC

- salt-syndic daemon được chia ra tình nhưng tiến trình riêng biệt, needs to be started in addition to the salt-master daemon running on the Syndic node.

- Bắt đầu salt-syndic daemon giống như các bắt đầu 1 tiền salt khác.

- Master node bằng nhiều cách thấy được Syndic node giống như Minion node. Chi tiết hơn, Master sẽ cần accept Syndic Minion key giống bất kỳ 1 node minion khác.

On the Syndic node:

# salt-syndic

or

# service salt-syndic start

On the Master node:

# salt-key -a my\_syndic

- Master node có thể kiểm soát được các minion node được kết nối tới Syndic. Các kết nối giữa Syndic và minion sẽ không bị ảnh hưởng khi master node kiểm soát, điều kiển các minion

Note:

Only the Syndic key will be listed in the Master node's key registry but this also means that key activity between the Syndic's Minions and the Syndic does not encumber the Master node. In this way, the Syndic's key on the Master node can be thought of as a placeholder for the keys of all the Minion and Syndic nodes beneath it, giving the Master node a clear, high level structural view on the Salt cluster.

On the Master node:

# salt-key -L

Accepted Keys:

my\_syndic

Denied Keys:

Unaccepted Keys:

Rejected Keys:

# salt '\*' test.ping

minion\_1:

True

minion\_2:

True

minion\_4:

True

minion\_3:

True

TOPOLOGY

- Master node (node mà chính nó ko phải là syndic nodemà nó là higher level Master) sẽ chạy “salt-master” daemon và tùy chỉnh “salt-minion” daemon.

- Syndic node sẽ chạy “salt-syndic” và “salt-master” daemin và tùy chọn các salt-minion daemon

- Minion node cần chạy “salt-minion” daemon.

- Khi “salt-master” daemon đưa ra 1 lệnh, lệnh sẽ gửi tới Syndic và Minion được kết nối trực tiếp đến. Minion node sẽ xử lý command. Trên syndic node, “salt-syndic” daemon sẽ chuyển tiếp các cấu lệnh tới tiến trình master (chính tại nó) để gửi thông điệp tới minion được kết nối tới.

- Khi event hoặc job return data được sinh ra bởi salt-minion daemin, nó sẽ được tổng hợp bởi “salt-master” daemon kết nối tới. Node master đó sẽ chuyển tiếp data tới “salt-syndic” daemon, sau đó “salt-syndic” sẽ gửi tiếp đến “salt-master” daemon tầng trên cho đến khi node master chính, ban hành lệnh.

SYNDIC WAIT

- “syndic\_wait” là master config file setting, nó sử dụng để định nghĩa ra số thời gian salt-minion chờ node syndics trước khi bỏ qua node đó, mặc định 5s.

- “syndic\_wait” setting cần thiết bởi higher-level master không biết chính xác các node bên dưới syndic. Higher-level master có list các node minions riêng, vì thế, salt client sẽ ko biết phải đợi bao lây trước khi tất cả các kết quả được trả lại. Ví thế, giá trị syndic\_way option sẽ định nghĩa ra thời gian chờ tất cả kết quả của các minion.

Note

To reduce the amount of time the CLI waits for Minions to respond, install a Minion on the Syndic or tune the value of the syndic\_wait configuration.

SYNDIC CONFIG OPTIONS

id: Syndic id (shared by the salt-syndic daemon with a potential salt-minion daemon on the same system)

syndic\_master: Master node IP address or hostname

syndic\_master\_port: Master node ret\_port

syndic\_log\_file: path to the logfile (absolute or not)

syndic\_pidfile: path to the pidfile (absolute or not)

syndic\_wait: time in seconds to wait on returns from this syndic

e