MULTI MASTER TUTORIAL

- Multi master system cho phép sử dụng nhiều Salt Master, tạo điều kiện cho Salt Minion kết nối tới.

- Khi sử dụng multi-master, tất cả master đều có thể điều kiên minion, gửi các cmd xuống minion.

Note

If you need failover capabilities with multiple masters, there is also a MultiMaster-PKI setup available, that uses a different topology MultiMaster-PKI with Failover Tutorial

- Master không chia sẻ các thông tin, việc xác thực các minion cần được cấu hình tay trên từng master. Việc share file trên từng server cũng cần thực hiện tay, để “file\_roots” giữa các master giống nhau.

- Minion data cache bao gồm grain, pillar, mine. Chúng được cache trên Salt Master. Mặc định, Salt sử dụng localfs cache module, cũng có thể sử dụng các external data store để thay thế

- Sử dụng minion cache module cho phép data được lưu trữ trên salt master về

Using a pluggable minion cache modules allows for the data stored on a Salt Master about Salt Minions to be replicated on other Salt Masters the Minion is connected to.

SUMMARY OF STEPS

- Các bước triển khai

1. Tạo mới 1 master server (master phụ)

2. Copy primary master key tới master phụ

3. Chạy master phụ

4. Cấu hình minion kết nối tới master phụ

5. Restart minion

6. Accept key trên master phụ.

PREPPING A REDUNDANT MASTER

- Chuẩn bị master (phụ).

- Nếu master phụ đã chạy => stop.

- Yêu cầu duy nhất cho master phụ là giữa các master phải sử dụng chung 1 private key vì tất cả minion chỉ chập nhận 1 khóa được cung cấp từ master.

- Đường dẫn mặc định của master key pair tại “/etc/salt/pki/master/”.

- Lấy private key “master.pem”, copy nó đến cùng 1 thư mục trên master phụ.

- Làm tương tự với “master.pub”.

Note:

+ Trong trường hợp hiện tại, chưa có bất kỳ 1 minion nào được join vào master phụ, nên việc xóa các chửng chỉ hiện có trên master phụ ko xảy ra lỗi.

+ Không có giới hạn các master phụ (tạo tùy thích)

- Khi đã thay đổi master key, master phụ có thể chạy.

CONFIGURE MINIONS

- Vì minion cần biết rõ các master sẽ kết nối tới, ta sẽ thêm master mới vào minion config.

- Update trong file config minion

master:

- saltmaster1.example.com

- saltmaster2.example.com

- restart minion

Note

If the ipc\_mode for the minion is set to TCP (default in Windows), then each minion in the multi-minion setup (one per master) needs its own tcp\_pub\_port and tcp\_pull\_port.

If these settings are left as the default 4510/4511, each minion object will receive a port 2 higher than the previous. Thus the first minion will get 4510/4511, the second will get 4512/4513, and so on. If these port decisions are unacceptable, you must configure tcp\_pub\_port and tcp\_pull\_port with lists of ports for each master. The length of these lists should match the number of masters, and there should not be overlap in the lists

- Sau khi minion restart, nó sẽ check lại master gốc và cả master phụ vừa tạo.

Note:

+ Minion có thể tự động phát hiện lỗi tại server và thử reconnect lại sau khi gặp lỗi. Để kích hoặc chức năng này, set “master\_alive\_interval” trong minion config và chỉ rõ thời gian check lại liên kết.

SHARING FILES BETWEEN MASTERS

- Salt không tự động share files giữ các master. 1 số lượng file cần được share hoặc nên được chia share.

MINION KEYS

- Minion key có thể được accepte đơn gian thông qua salt-key trên tất cả master. Key có thể được accepted, delete, rejected trên 1 master mà ko được tự động update trên master phụ. Nó cần được quản lý rõ ràng trên tất cả các master hoặc share thư mục /etc/salt/pki/master/{minions,minions\_pre,minions\_rejected}.

Note:

+ Khi chia sẻ /etc/salt/pki/master, việc share master.pem có thể tạo ra 1 lỗ hổng bảo mật nghiêm trọng.

FILE\_ROOTS:

- “file\_roots” content cần được đồng bộ, hoặc rõ ràng giưa các master. Không tạo ra các xung đột giữa cấu hình các file\_roots giữa các server.

- Đề nghị giữ đồng bộ giữa các master thông qua các phần mêm hỗ trợ như git hoặc thông qua data storge.

Important

If using gitfs/git\_pillar with the cachedir shared between masters using GlusterFS, nfs, or another network filesystem, and the masters are running Salt 2015.5.9 or later, it is strongly recommended not to turn off gitfs\_global\_lock/git\_pillar\_global\_lock as doing so will cause lock files to be removed if they were created by a different master.

PILLAR\_ROOTS

- Pillar root cần được đồng bộ giống như file\_roots.

MASTER CONFIGURATIONS

- Các Master thường có nhưng cấu hình mở rộng như:

external\_auth

publisher\_acl

peer

peer\_run

- Cần được đồng bộ giưa các master

e