Getting Started: Standard MCollective Deployment

Summary

- Cách bước để deploy standard Mcollective environment.

The Standard MCollective Deployment

Architecture and Configuration

- Về cơ bản, đây là kiến trúc và config conventions tạo ra standard Mcollective deployment:

+ Middleware và connector: ActiveMQ

-- One ActiveMQ server (expandable later)

-- CA-verified TLS

-- No extra subcollectives

-- One ActiveMQ user account/password for all MCollective traffic

+ Security plugin: SSL (authentication only)

+ Credentials:

-- Certificate/key pair for each client user (for both connector and security plugins)

-- One shared certificate/key pair for all servers (for security plugin)

-- Pre-existing Puppet certificate/key pair on each server (for connector plugin)

-- -- Alternately, you can use the shared server certificate/key for both the connector and security plugins; unique certificates aren’t strictly necessary.

+ Authorization: ActionPolicy plugin

Security Model

Application Level

With the SSL security plugin, each client user has a unique key pair and all servers share a single key pair. Each server node holds a collection of all authorized client public keys.

* When clients issue requests, they sign the payload and TTL with their private key; servers do the same when they send replies. This strongly identifies individual clients and identifies servers as a group. (An authorized server could theoretically impersonate another authorized server; in the common use cases for MCollective, this isn’t a significant concern.)
* Servers will reject requests signed by any key that isn’t in their collection of authorized clients. This acts as coarse-grained client authorization. (Note especially that the shared server key pair cannot be used to send requests, which is an advantage over the weaker PSK security plugin.)
* The ActionPolicy plugin allows fine-grained client authorization at the per-action level. (This relies on the client authentication provided by the SSL security plugin.)
* Servers also check the signature of the request payload and TTL, which protects against message tampering and replay attacks.

Application Level

With the SSL security plugin, each client user has a unique key pair and all servers share a single key pair. Each server node holds a collection of all authorized client public keys.

* When clients issue requests, they sign the payload and TTL with their private key; servers do the same when they send replies. This strongly identifies individual clients and identifies servers as a group. (An authorized server could theoretically impersonate another authorized server; in the common use cases for MCollective, this isn’t a significant concern.)
* Servers will reject requests signed by any key that isn’t in their collection of authorized clients. This acts as coarse-grained client authorization. (Note especially that the shared server key pair cannot be used to send requests, which is an advantage over the weaker PSK security plugin.)
* The ActionPolicy plugin allows fine-grained client authorization at the per-action level. (This relies on the client authentication provided by the SSL security plugin.)
* Servers also check the signature of the request payload and TTL, which protects against message tampering and replay attacks.

Steps to Deploy

Các bước cần làm để deploy MCollective:

+ Create and collect credentials

+ Deploy and configure middleware

+ Install MCollective (on both servers and admin workstations)

+ Configure servers

+ Configure clients

+ Deploy plugins

- Các bước trên không phải 100% các vấn đề, nhưng nó là các trật tự giúp việc tiếp cận trở nên dễ nhất.

Step 1: Create and Collect Credentials

- Credentials là area lớn nhất trong shared global config Mcollective. Hoàn thành nó trước khi chuyển sang bước tiếp theo

- Standard deployment sử dụng những chứng chỉ:

|  |  |
| --- | --- |
| Credential | Used By: |
| ActiveMQ username/password | Middleware, servers, clients |
| CA certificate | Middleware, servers, clients |
| Signed certificate and private key for ActiveMQ | Middleware |
| Signed certificate and private key for each server | Servers |
| Signed certificate and private key for each user | Clients (both parts), servers (certificate only) |
| Shared server public and private key | Servers (both parts), clients (public key only) |

Ý nghĩa:

- ActiveMQ ↔ MCollective traffic sử dụng CA-signed X.509 certificates cho việc encryption và xác minh. Cũng giống như chứng chỉ trong puppet.

+ Không giống puppet, chúng ta không sử dụng certificate DN/CN cho authentication — the CA verification độc nhất, tránh cho việc bị tấn công man-in-the-middle attacks.

- SSL security plugin (trên server và client) sử dụng RSA public/privates key pair (hoắc 1 số key khác có thể đọc được bởi ssl) sử dụng cho việc chứng thực, ủy quyền, và ký các message. public portion is flexible: it can be either a raw RSA key, or a signed SSL certificate. The plugin identifies keys by filename.

- Nếu sử dụng Puppet, có thể re-use các chứng chỉ có sẵn.

- Trên Mcollective node, tất cả chứng chỉ phải thuộc dạng “.pem”, trên middleware, 1 số dạng đặc biệt có thể đc sử dụng

Walkthrough / Checklist

- Đảm bảo hoàn thành các phần về chứng chỉ để sử dụng cho các bược sau. Các bước bên dưới sử dụng khi đã cài đặt puppet.

Note:

+ lấy các đường dẫn về chứng chỉ trong puppet agent và puppet master

puppet agent --configprint certdir,privatekeydir

puppet master --configprint certdir,privatekeydir

- PASSWORD: Do: Quyết đinh chi user connecting tới ActiveMQ; sử dụng tên “mcollective”. Tạo passwd cho user.

- CA: Đã có sẵn từ PuppetCA, nó luôn lưu trữ tại “$certdir/ca.pem”.

- ACTIVEMQ CERT: có thể sử dụng lại từ puppet agent, hoặc sinh chứng chỉ mới bằng CA puppet master: “puppet cert generate activemq.example.com”. Tìm chứng chỉ trong “$certdir/<NAME>.pem” và “$privatekeydir/<NAME>.pem”

- SHARED SERVER KEYS: Trên CA puppet master, sinh ra chứng chỉ mới “puppet cert generate mcollective-servers”. Tìm lại trong: “$certdir/mcollective-servers.pem” và “$privatekeydir/mcollective-servers.pem”.

- SERVER CERTS: Đã có sẵn, trên mỗi server đều có chứng chỉ riêng (puppet agent node or puppet master node)

- Client certs: tiếp tục tạo client credentials khi muốn thêm 1 admin user:

+ first admin user — yourself, có thể sinh ra chứng chỉ trên CA puppet master với “puppet cert generate <NAME>” và lấy nó trong “$certdir/<NAME>.pem” và “$privatekeydir/<NAME>.pem”. Xóa CA’s copy của private key khi retrieved it.

+ For future admin users, you need to build a process for issuing and distributing credentials

Step 2: Deploy and Configure Middleware

- Các bước dưới sử dụng puppet để hoàn thành

- Sử dụng template cho activemq.xml file và using the java\_ks resource type cho keystores

Bước 1: Cài đặt ActiveMQ 5.5 trên ActiveMQ server. Nếu sử dụng dòng fedora hoặc như RHEL, add thêm package repo, nếu không có vào trang chỉ để install activemq.

# rpm -Uvh <https://yum.puppetlabs.com/puppet5/puppet5-release-el-7.noarch.rpm>

Bước 2: Cách