

クラウド基盤構築演習

第二部: Eucalyptusによるクラウド基盤構築

第15回:トラブルシューティングとトラブル事例

ver1.1 2012/07/20



目次

- 問題の切り分けと状況確認
- インストール時のミスによるトラブル事例
- コンポーネントが動かないトラブル事例
- インスタンスが起動できないトラブル事例
- インスタンスに接続できないトラブル事例



問題の切り分けと状況確認

問題を切り分ける際の観点

- 本当に問題か？
 - 問題発見者の認識違いによる報告でないかを疑う
 - 利用者の手順ミスでないかを疑う
- 今まで動いていたか？/最初っから動いていなかったか？
 - 今まで動いていたのに問題が発生した場合は、原因となる事柄が問題発生前にあるため、問題発生時および発見時から時間を遡りつつ調査を行なう
 - 最初っから動いていない場合は、構築時の手順なりを疑う必要がある
- 環境内が原因か？/環境外が原因か？
 - 場合によっては環境外に原因があったりする場合もある
- 常に発生するか？/2回に1回の頻度か？/再現性が低いか？/再現性ゼロか？
 - 発生頻度の特性によって原因が判明する場合あり
 - 再現性がゼロの場合は再現させるための条件が不足している場合あり
- 環境起因か？/ハードウェア起因か？/ソフトウェア起因か？
 - ハードウェア起因の場合は、実機を見たほうが早い場合も…

状況確認をする際の注意点 (1)

- 状況確認する際には、確認したことを全て記録に残すこと
 - 問題を調査する人間が認識間違いや勘違いを起すと更に問題がややこしくなるため、常に確認したことをエビデンスとして残すこと。
 - エビデンスを残すことにより他者への報告が可能となる。
(逆を言うとエビデンスのない報告は信憑性が低くなる)
 - エビデンスは基本として一次情報から残す
 - コマンドの出力結果はリダイレクトして残したり、端末エミュレータの機能を使ってログに出力する
 - ログファイルはそのままコピーするなど
 - 画面のスクリーンショットも有効
 - 中途半端に手打で転記するのは好ましくない。どうしても転記しないと無理な場合はカメラで写真を撮ること

状況確認をする際の注意点 (2)

- 環境負荷の低いものから確認をすること
 - コマンドを叩かなくても確認できることがあれば、それから確認する
 - 実機のLEDとか、監視ソフトやログサーバ側でのログ確認など
 - 既に実環境にログインしている端末などがあれば、そこで確認する
 - トラブルによってはログイン処理自体捌けなくて反応が返ってこない場合もある
- 低レイヤーから確認していくこと
 - トラブルの内容から考えて直接関係がないと思っても低レイヤーの情報は基本情報として確認すること
 - まず最初にディスクの空き容量、メモリの消費量、CPUの負荷率、I/O Waitの確認、ネットワークの使用量など
 - 次にプロセス数、ファイルディスクリプタの使用量など
 - ファイルシステムによってはi-nodeの数も重要

状況確認をする際の注意点 (3)

- アプリケーションは調査対象外のアプリケーションも確認すること
 - 対象アプリケーション → 依存アプリケーション → 直接関係しないアプリケーションの順でよいので確認すること
 - Eucalyptusの場合は例えば以下の順
 - 対象コンポーネント
 - 対象外コンポーネント
 - 依存アプリケーション
 - apache, postgresql, libvirt, kvm/xen/VMware, iptables, AoE/iSCSI, LVM2, DHCP, MTAなどなど
 - 直接関係しないアプリケーション
 - dnsmasq, SELinux, OpenStack(あまり無いとは思う構成だが)

インストール時のハマりどころ (概要)

- 設定ミス
 - 使える機能が多い分、MANAGEDモードが一番ハマり易い
- 時刻同期の必要性
 - 各コンポーネント(各サーバ)の時刻が大幅にズれていると思わぬエラーが発生する可能性あり
 - 時刻同期ができていないと、各コンポーネントのログを突き合わせるときに地獄をみる
- 以外と忘れがちなSMTPの設定
 - CLCはユーザ作成時にEucalyptusのadminユーザ宛てにメールを送信（送信しないように選択することも可能だが）
 - 利用者がCLCのWeb管理画面からユーザ作成の申請をした場合にもメールを送信
- インストール前の環境設計が重要
 - 利用度を想定してハードウェアのスペックを考える
 - ネットワークの設計も蔑ろにしないこと
- デプロイ系のツールを使わずに手作業で構築する場合には…
 - 手順書で間違い易い部分などは適宜注釈などを追記する
 - チェックシートなどを準備して、逐次チェックしながら作業する



インストール時のミスによるトラブル 事例

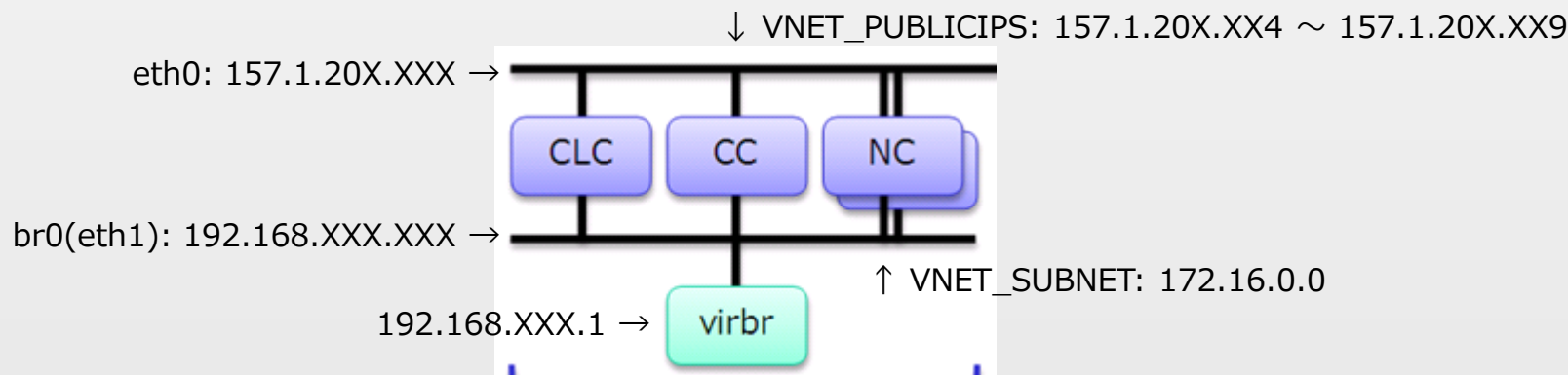
設定ミスの例 (前回の演習から)

- eucalyptus.confへの設定ミス
 - VNET_PUBLICIPSが"157.1.207.14-157.1.207.19"に設定されているホストが多数あり
 - 第10回の手順書自体がわかり辛かった点も...

191	VNET_PUBLICIPS	(コメント) your-free- public-ip-1 your-free- public-ip-2 ...	(コメント外す) 別紙を参照	#VNET_ADDRSERVERNET="32" #VNET_DNS="your-dns-server-ip" ↓ VNET_MODE="MANAGED-NOVLAN" VNET_PRIVINTERFACE="eth1" VNET_PUBINTERFACE="eth0" VNET_PUBLICIPS="157.1.207.14-157.1.207.19" VNET_SUBNET="172.16.0.0"
196	VNET_SUBNET	(コメント) 192.168.0.0	(コメント外す) 172.16.0.0	
197	VNET_NETMASK	(コメント)	(コメント外す)	

設定ミスの例 (前回の演習から)

- eucalyptus.confへの設定ミス
 - VNET_PUBLICIPSが"192.168.1.120-192.168.1.220"とか
 - 192.168.1.~のIPアドレスは今回の環境ではPublicIP側ではない
 - VNET_SUBNETが"10.1.0.0"とか
 - VNET_DNSが"192.168.1.10"とか
 - VNET_MODEが"MANAGEe"とか
 - NCのVNET_PUBINTERFACEが"eth0"とか



設定ミスの例 (前回の演習から)

- eucalyptus.confへの設定ミス
 - VNET_PUBLICIPSの設定漏れ
 - VNET_PUBLICIPSのコメントは外されていたが、値が設定されていなかった…
 - このパラメータに無効な文字列が設定されている場合、PublicIPは127.0.0.1が設定される
 - 設定ファイルの1/3が削除されていた例も…
 - そのためインスタンスの起動要求を受けるNCが動作できずにインスタンスが起動しない状態になる
- そもそも論で言えば、CCなりNCが起動時にパラメータチェックをして明かに不正な値が設定されている場合にはエラーを吐いて起動しないようにするのが筋だが、そこまでの機能は現バージョンでは有していない。

設定ミスの例 (前回の演習から)

■ SELinuxへの設定ミス

- “disabled”が“disable”となっていたり…
- SELINUX=disabledとするところがSELINUXTYPE=disabledとなっていたり
 - ↓この設定ミスの場合はOSが起動しなくなる

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled  - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls      - Multi Level Security protection.
SELINUXTYPE=disabled
```

設定ミスの例 (前回の演習から)

- euca_confでのコンポーネントの登録ミス
- オプションおよび値の間違い

Walrus用とCluster用のオプションが混在

```
# euca_conf --register-walrus --partition walrus --host 157.1.207.31 ¥  
> --component cluster0  
register-walrusを指示しているが、--hostの値はCCのIPアドレス
```

CCの登録後にSCでなく再度別のコンポーネント名としてCCが重複登録

```
# euca_conf --register-cluster --partition jp-east-1a --host 157.1.207.151 ¥  
> --component cluster0  
# euca_conf --register-cluster --partition jp-east-1a --host 157.1.207.151 ¥  
> --component storage0
```

CCと通信するNICのIPアドレス以外のIPアドレスを指定してしまった例

```
# euca_conf --register-nodes "157.1.207.42 157.1.207.43"
```

設定ミスの例 (前回の演習から)

- euca_confでのコンポーネントの登録ミス
 - 登録漏れ
 - 手順書の判り難さからWalrusの登録漏れが目立った
 - 多重登録
 - 基本的には登録済みのコンポーネントを一度deregisterしてからでないと再度登録しては駄目
- コンポーネントのインストール漏れ
 - Walrusがインストールされていない環境があった



コンポーネントが動かないトラブル事例

コンポーネントの生存確認

■ コマンドラインで確認する方法

Walrusの確認方法

```
# euca_conf --list-walruses
```

WALRUS	walrus	walrus0	157.1.207.30	ENABLED {}
--------	--------	---------	--------------	------------

CCの確認方法

```
# euca_conf --list-clusters
```

CLUSTER	jp-east-1a	cluster0	157.1.207.31	ENABLED {}
---------	------------	----------	--------------	------------

SCの確認方法

```
# euca_conf --list-scs
```

STORAGECONTROLLER	jp-east-1a	storage0	157.1.207.31	ENABLED {}
-------------------	------------	----------	--------------	------------

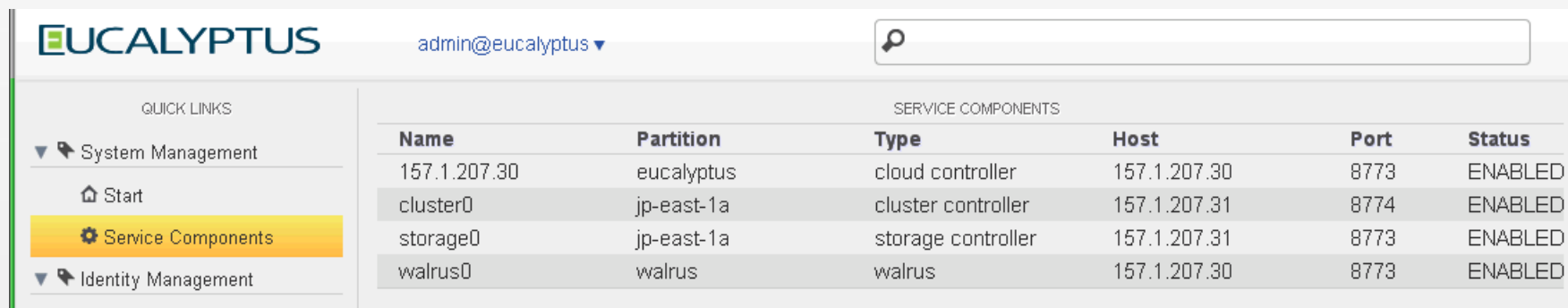
NCの確認方法 (生存確認とは言えないが...)

```
# euca_conf --list-nodes
```

NODE	192.168.3.100	cluster0	i-192D43E3
NODE	192.168.3.101	cluster0	

コンポーネントの生存確認

■ WebUIで確認する方法

The screenshot shows the Eucalyptus web interface. On the left is a sidebar with 'QUICK LINKS' including 'System Management', 'Start', 'Service Components' (highlighted in yellow), and 'Identity Management'. The main area is titled 'SERVICE COMPONENTS' and contains a table with the following data:

Name	Partition	Type	Host	Port	Status
157.1.207.30	eucalyptus	cloud controller	157.1.207.30	8773	ENABLED
cluster0	jp-east-1a	cluster controller	157.1.207.31	8774	ENABLED
storage0	jp-east-1a	storage controller	157.1.207.31	8773	ENABLED
walrus0	walrus	walrus	157.1.207.30	8773	ENABLED

コンポーネントが動かない事例

- iptablesの設定が活きている
 - ディストリビューションによってデフォルトのiptablesの設定は異なるが、ほとんどの場合はポートが閉じられている
 - それにより、各コンポーネントが通信で使用するポート番号は遮断された状態となり、コンポーネント間が通信できずに、動いていない状態となる
- SELinuxが有効になっている
 - SELinuxが有効になっているとコンポーネントが正しく動作できないため、場合によってはコンポーネント自体が起動しない状態となる
- コンポーネントの認証情報が正しくない
 - /var/lib/eucalyptus/keys/配下にある認証情報が古い情報になっているか、もしくは存在しない場合があり、コンポーネント間の通信が成功しない
- ハードディスクの空き容量がない
 - ハードディスクの空き容量がないため、ログ出力が行なえずコンポーネントが動作しない、もしくはコンポーネントが起動できない



インスタンスが起動できないトラブル事例

インスタンスが起動できない事例

■ マシンリソースが不足している場合

■ euca-describe-availability-zones verboseの結果、freeの項目が0になっている場合

■ NCの物理リソース(CPU、Memory、HDD)のうちどれか1つでもリソース不足になれば起動できない

```
# euca-describe-availability-zones verbose
AVAILABILITYZONE      jp-east-1a      157.1.207.31 arn:euca:eucalyptus:jp-east-1a:cluster:cluster0/
AVAILABILITYZONE      | - vm types    free / max    cpu    ram    disk
AVAILABILITYZONE      | - m1.small     0001 / 0006    1     128     2
AVAILABILITYZONE      | - c1.medium    0001 / 0006    1     256     5
AVAILABILITYZONE      | - m1.large     0000 / 0002    2     512    10
AVAILABILITYZONE      | - m1.xlarge    0000 / 0002    2    1024    20
AVAILABILITYZONE      | - c1.xlarge    0000 / 0000    4    2048    20
```

インスタンスが起動できない事例

- 指定したVM Typesのスペックよりマシンイメージが大きい場合
 - VM Typesのdiskの値よりマシンイメージのほうが大きい場合は起動できない
 - マシンイメージのサイズ + Swapディスクのサイズ(デフォルト値は512MB) + EKIのサイズ + ERIのサイズの合計がVM Typesのdiskの値を越える場合は起動できない

$$\text{ephemeral} = \text{VM TypesのDisk} - (\text{swap} + \text{emi} + \text{eki} + \text{eri})$$

VM Typesの Disk Size { ephemeral領域
swap領域
OS領域

インスタンスが起動できない事例

- Walrusからマシンイメージがダウンロードできない場合
 - NCからWalrusに到達できない場合は当然ながらインスタンスを起動できない
 - Walrus上に復号処理されたマシンイメージが存在しておらず、起動要求があった際に復号処理をするが、NCの試行回数(10回)以内に復号処理が完了せずに、NCがマシンイメージの取得を諦める場合
- Walrusに登録されたマシンイメージに問題がある場合
 - カーネルイメージを登録(BundleImage)する際に--kernel trueを指定し忘れた
 - RAMディスクイメージを登録する際に--ramdisk trueを指定し忘れた
 - イメージを登録する際にアーキテクチャの指定を間違えた。(例えばx86_64のイメージなのにi386とか)



インスタンスに接続できないトラブル事例

インスタンスに接続できない事例

- iptablesが設定されている
 - 前述したようにCCとNCのiptablesは無効にしておかないとコンポーネント間の通信ができなかったり、インスタンスを起動しても通信できなかったりする
- セキュリティグループの設定をし忘れている
 - 「そんな馬鹿な」と思うかもしれないが、実際に利用していると設定し忘れることがよくある
- キーペア間違い
 - 一見同じ名前のキーペアだが、よくよくフィンガープリントを比較してみると違うキーペアだったりする
 - 不安な場合は、以下のように比較して確認すること

```
[root@clc ~]# euca-describe-keypairs
KEYPAIR key01    f0:f0:89:e5:ca:fb:87:30:c7:e8:8f:52:2c:b6:a6:7c:8f:4a:75:1e
[root@clc ~]# head -n 1 .euca/admin/keypairs/key01.pem
KEYPAIR key01    f0:f0:89:e5:ca:fb:87:30:c7:e8:8f:52:2c:b6:a6:7c:8f:4a:75:1e
```

インスタンスに接続できない事例

- .ssh/known_hosts内の情報と接続しようとしているインスタンスの情報が違う場合
 - 以下のようにssh-keygenコマンドで情報を削除

```
[root@clc ~]# ssh -i .euca/admin/keypairs/key01.pem 157.1.207.34
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
29:15:e5:e5:6e:36:c0:e9:bc:2c:87:72:1e:62:61:51.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending key in /root/.ssh/known_hosts:3
RSA host key for 157.1.207.34 has changed and you have requested strict checking.
Host key verification failed.
[root@clc ~]# ssh-keygen -R 157.1.207.34
/root/.ssh/known_hosts updated.
Original contents retained as /root/.ssh/known_hosts.old
```

インスタンスに接続できない事例

- キーペアの秘密鍵のパーミッションが正しくない
 - 秘密鍵のパーミッションは0400か0600と設定すること

```
[root@clc ~]# ssh -i .euca/admin/keypairs/key01.pem 157.1.207.34
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!            @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0666 for '.euca/admin/keypairs/key01.pem' are too open.
It is recommended that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: .euca/admin/keypairs/key01.pem
root@157.1.207.34's password:
```

- キーペアのインジェクションに失敗した場合
 - EucalyptusやOpenStackはインスタンスの起動前にマシンイメージをマウントしてキーペアの公開鍵を/root/.ssh/authorized_keysファイルに直接追加する。ただし、Eucalyptusではマシンイメージをext2/ext3と想定しているため、マシンイメージがそれ以外のフォーマットである場合は公開鍵を追加できない