

クラウドインフラ構築特論

グループ 2 Combu

小宮・宮崎・蔡・劉

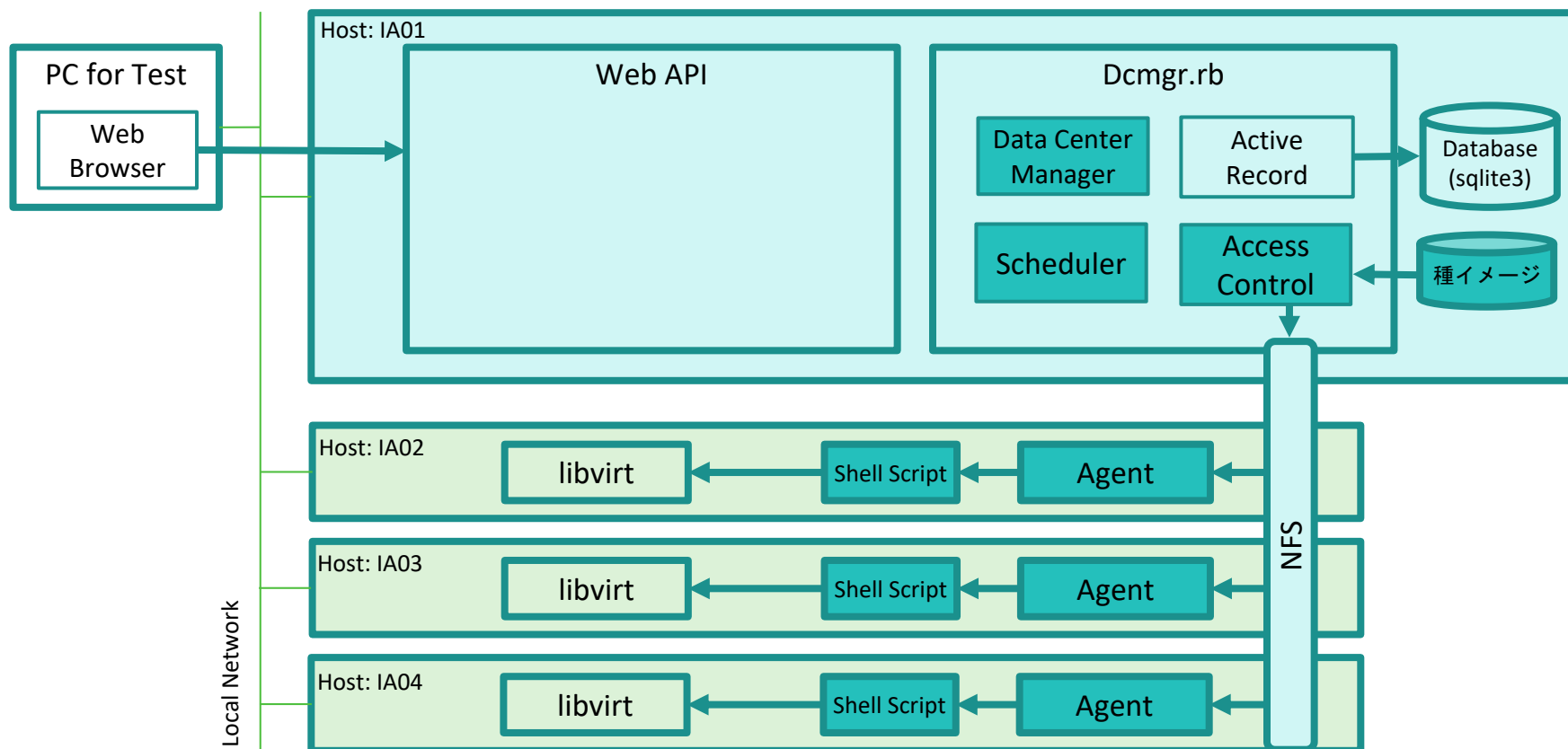
2017/11/21



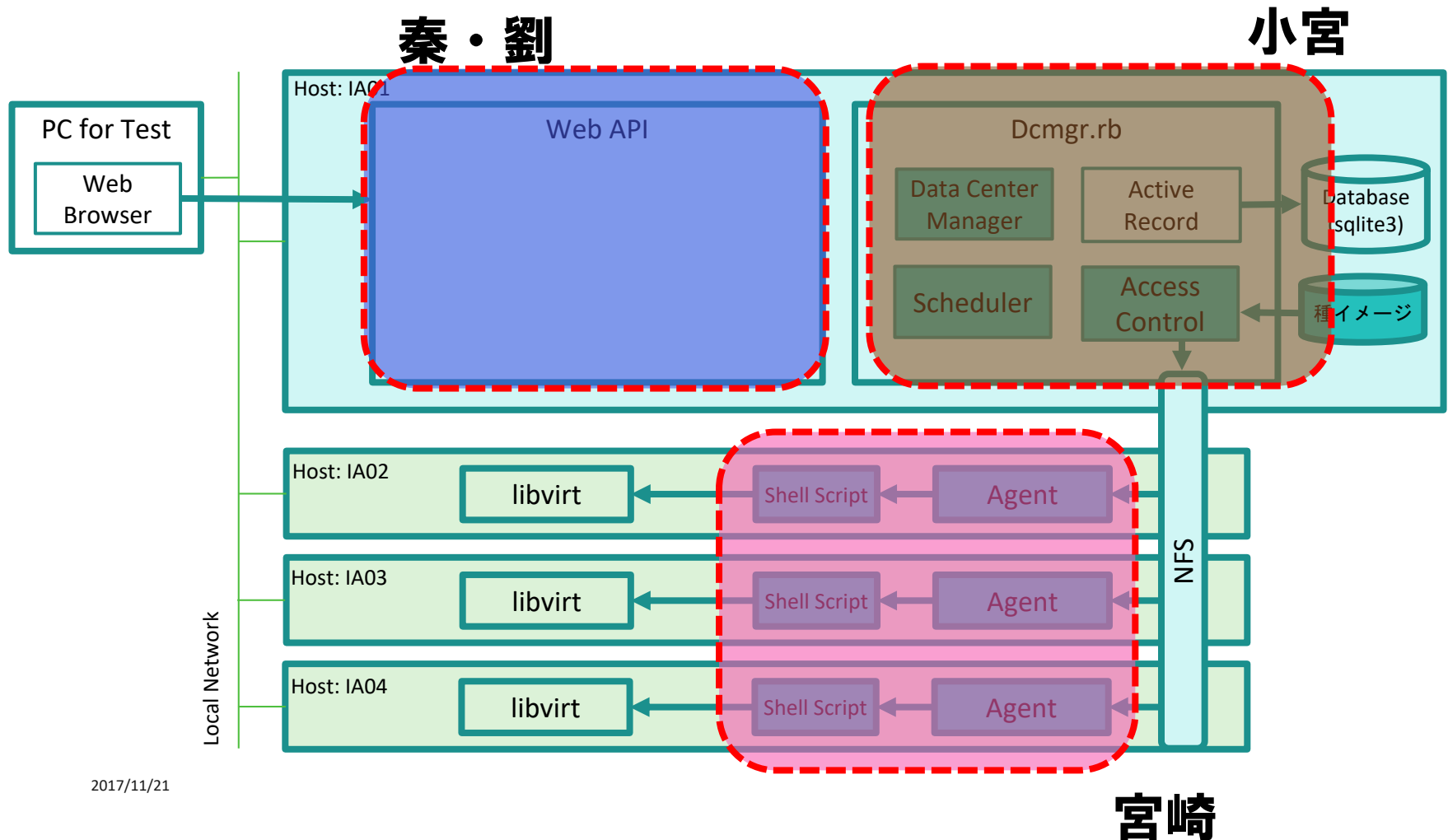
Agenda

- システム全体の配置と担当範囲
- VMの状態遷移
- 工夫したところ（各自）
- ソースコード
 - <https://github.com/RyutoHata/Combu>
- Appendix
 - 設定資料など

システムの配置



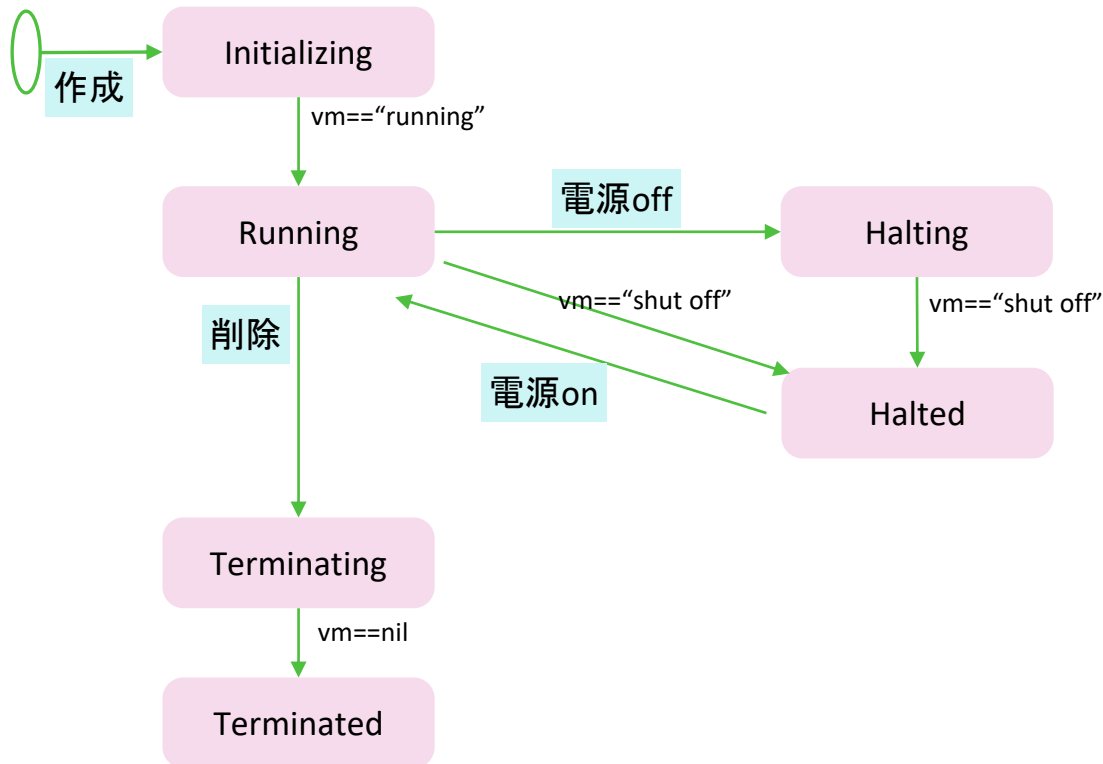
システムの配置 (担当範囲)



WEB API

- GET / ホーム画面を表示する
- GET /vm/new VM作成フォームを表示する
- POST /vm VMを作成
- GET /vm VMリストを表示する
- PUT /vm/:id VMの電源操作（ON or OFF）
- DELETE /vm/id VMを削除する

VMの状態遷移



Agentで定期的に
'virsh list -all'
の結果をDcmgrに
通知する。

userがvmにログイン
して' shutdown' を
してもhaltedの状態
に移行できる。

工夫したところ (秦)

- 管理画面からvmの電源操作や削除が行えるようにした

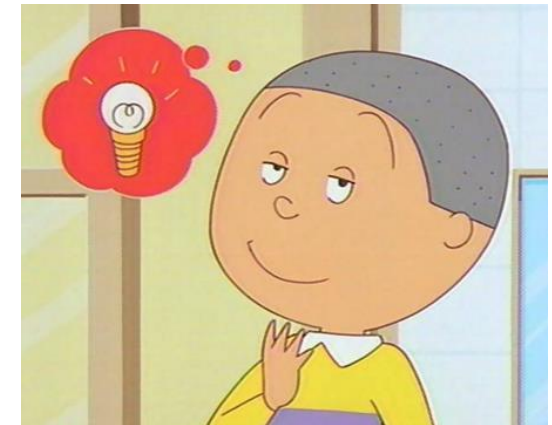
VM 一覧

[新規作成](#)

VM ID	CPU 数	メモリ(MB)	SSH公開鍵	サーバー	IPアドレス	ステータス	削除	電源
1	1	2	lkasjdlfajsoidsx	IA02	192.168.0.20	Terminated	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
2	3	5	asdjflksadjflga	IA03	192.168.0.21	Halting	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
3	1	8	aaaaaaaaaaaaaaaaaaaa	IA04	192.168.0.22	Terminated	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
4	1	4	asjdlfrlkjcxz	IA02	192.168.0.23	Initilizing	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
5	3	4	adsfjlreiodsa	IA04	192.168.0.24	Initilizing	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
6	5	3	asldjroilsd	IA02	192.168.0.25	Initilizing	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
7	4	43	asdlfjlwafkjdsfzx	IA03	192.168.0.26	Initilizing	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
8	1	1	asdjflksajfds	IA04	192.168.0.27	Initilizing	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
9	2	4	dsafdsfdfsdfsds	IA04	192.168.0.20	Initilizing	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行
10	1	4	sdlkjlfcxvjoirew	IA02	192.168.0.22	Initilizing	削除する	<input type="radio"/> ON <input type="radio"/> OFF 実行

DELETE '/vm/:id'

PUT '/vm/:id'



工夫したところ (劉)

- フロントの部分は秦さんと作成。
- 面白要素

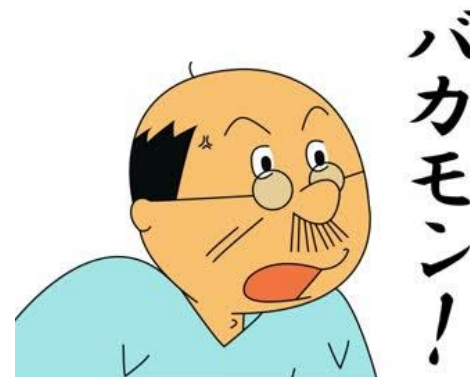
チーム：Combu デス

次に進みたい場合 → 以下の2つの画像をどれかクリックしてね～



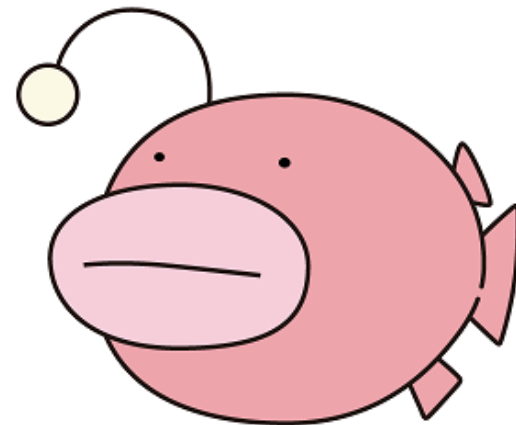
工夫したところ (小宮)

- NFS上のファイルのやり取りで、キューの動作を簡単に実装しました。
- Schedulerの割り当てアルゴリズム
” 要求されたCPUとMemoryの割り当てが可能なホストの中からCPUの空きが最も多いものを選択する。”
- Agentから`virsh list -all`の結果を定期的にUpdateしてもらい、DatabaseのVMの状態を更新するようにしたことで、ユーザーによるVMのShutdownによる状態変化も検出可能にしました。
- Web APIとdcmgrのInterfaceを、JSON形式の文字列で定義したことにより、単体でのテストがやりやすくなった。結果、結合テストで発見されるバグが少なくスムーズに進んだ。



工夫したところ (宮崎)

- 動作ごとに細分化しました。
- Agentをpython2.7とシェルスクリプトで書いています。
- エラー表示は考えないことにしました。
- Agentでは置かれているファイルを少ない番号から読み込み、中に書いてあるスクリプトを直に出力するように簡素に使っています。
- 起動中のVMを表示するために専用で定期的にファイルを出力するようになってます。





実装したかったこと



全データの削除の実装
外部からのVMへの接続
オートスケールの実装





提 供



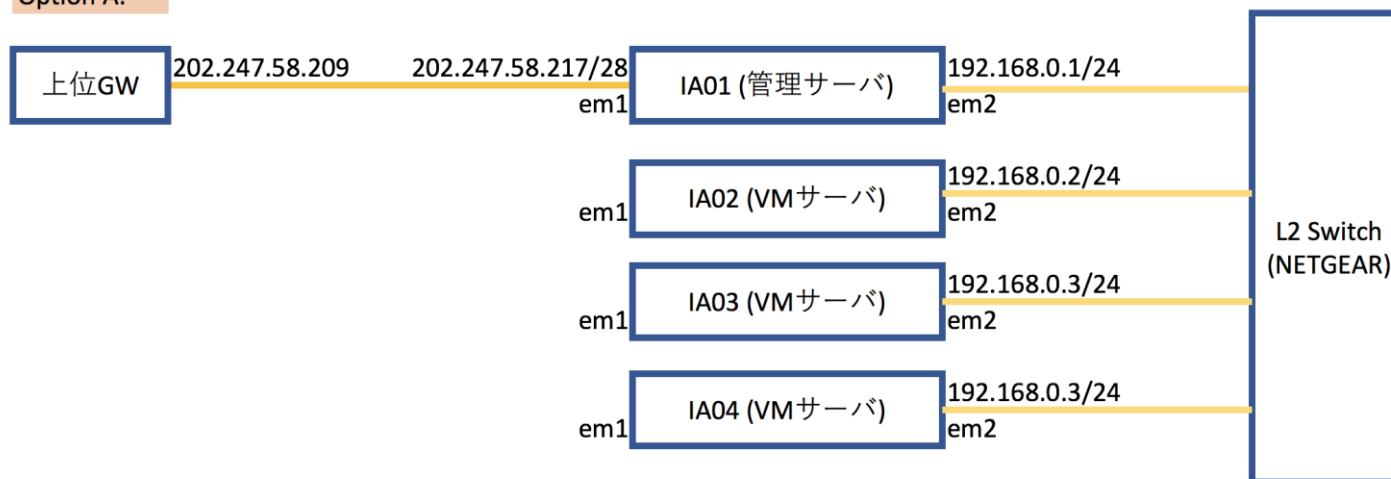
御覧の皆様



Appendix

ネットワーク構成

Option A:



VM起動スクリプト

[Start.sh]

```
#!/bin/sh
# $1 : VM Domain Name
# $2 : Number of VCPU
# $3 : Memory Size (MB)
# $4 : IP_address
# $5 : VM Hostname(ex:'hoge hoge')
# $6 : SSH-Key
#
KHOME=/home/kvm/disk/$1
SEED=/mnt/ia01/seed
mkdir $KHOME
cp $SEED/disk.qcow2 $KHOME
/usr/bin/truncate -s 10m $KHOME/metadata_drive; sync;
parted $KHOME/metadata_drive < $SEED/parted_procedure.txt
MAPPER=`kpartx -av $KHOME/metadata_drive | cut -d' ' -f3`
udevadm settle
echo $MAPPER
mkfs -t vfat -n METADATA /dev/mapper/$MAPPER
mkdir $KHOME/md_mount
/bin/mount -t vfat /dev/mapper/$MAPPER $KHOME/md_mount
cp $SEED/metadata_master/* $KHOME/md_mount
echo $5 > $KHOME/md_mount/hostname
echo 'IPADDR="%$4%" >> $KHOME/md_mount/ifcfg-ens3
echo $6 > $KHOME/md_mount/id_rsa.pub
/bin/umount -l $KHOME/md_mount
virt-install --name $1 --vcpus $2 --memory $3 ¥
--disk $KHOME/disk.qcow2,format=qcow2 ¥
--disk $KHOME/metadata_drive ¥
--network bridge=br0 ¥
--mac=RANDOM ¥
--vnc --noautoconsole --import
```

パラメータとして、
ドメイン名、CPUの数
メモリ量、IPアドレス、
ホスト名、SSH=KEY をもらう。

種イメージをコピー
metadataの作成

metadataにホスト名、
IPアドレス、SSHKeyを書き込み

種イメージと、metadriveを使って
VMを起動

VM削除スクリプト

[terminate.sh]

```
#!/bin/sh
```

```
KHOME=/home/kvm/disk/$1
```

```
SEED=/mnt/ia01/seed
```

```
virsh shutdown $1
```

```
virsh undefine $1
```

```
rm -rf $KHOME
```

パラメータとして、
VMドメイン名をもらう

VMの停止と削除

VMのフォルダを削除