em (Environment Manager)

Development Manual

Table of Contents

TODO 3

Release Procedure 5

# TODO

* Overlays should get rid of is-overlay-applied() – we’re not using it and complicates things unnecessarily. The overlays should be idempotent.
* I noticed I can create two instances with the same ID/name – there should be a protection against that.
* **Risky usage of the instance provisioning private key.** In order to allow instances to ssh into the NFS server for various reasons (read the uname, etc), I am currently bundling em-provisioning-private-key.pem with the tar pushed to the instance. This is not OK, as the private key should not be published. I must replace this mechanism with an “environment key pair” that is generated on a per-environment basis. Search for “appending instance provisioning key file” and “ssh-ing into the NFS server”
* **NFS is not started by the overlay.** Currently the nfs server is not started by the nfs-server overlay, I had troubles with starting the server and running export –a, so I had to stop and then start the server. This is awkward. I need to fix immediate NFS server startup.

Features Requests during Usage

• Ranges: p001-p005.

• Need: em status --<state> to filter only the nodes in the specified state. Also we want alphabetically ordered nodes.

• Understand why this is and get rid of it:

The authenticity of host '52.11.144.156 (52.11.144.156)' can't be established.

RSA key fingerprint is 9e:96:48:53:93:c8:0d:41:c3:dc:69:84:4a:7c:32:6c.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '52.11.144.156' (RSA) to the list of known hosts.

Last login: Mon Jun 8 12:15:46 2015 from 207.114.215.130

TODO

• Procedure to automatically create the file server node.

• ec2-user is not guaranteed to have the same uid across the nodes. Best if I create my own consistent users.

• to-deplete. Unify “em” and “amazon” libraries. The unintegrated amazon remnants are maintained in src/main/bash/to-deplete.

• em create c001-c005 (ranges)

• Remove private keys from the GitHub repository. Find a better solution to keep the environment identity. Currently it’s in GitHub and publicly accessible. I should get the private/public key from the NFS server and push them to the newly burned instances. Get rid of the private key from GitHub and change it on f01.

• Understand the implications of exposing my Amazon keys via usage of ec2 tools on shared instances. Create an “em” account? Currently my secret key is exposed in the environment of the ec2-user on f01.

• Investigate cloud-init – it can do some configuration work. For example, it accepts modules to perform package update. “Defining userdata for instances in AWS seems really useful for doing all kinds of bootstrap-type actions.”

• em.shlib java() function should configure the external Java process for a quick startup.

• Document filters: name=b01, name=b01|b02|b03|b04

• Storage Management

o Difference between “standard” and “gp2” volume type. It seems that new instances are created with standard volumes.

o Resolve the lsblk problem – format those disks.

• Hundreds of NFS clients hitting the servers?

* Sometimes during the creation procedure we get this (meaning there’s a race condition between the creation of the instance and attempting to set a tag on it)

nombp1:em ovidiu$ em create nfs01

creating instance nfs01 ...

Client.InvalidInstanceID.NotFound: The instance ID 'i-b4615942' does not exist (Service: AmazonEC2; Status Code: 400; Error Code: InvalidInstanceID.NotFound; Request ID: 71c1317e-8fa2-45bf-b032-027aa7ac7b2d)

[error]: the instance i-b4615942 was created but failed to set tag Name=nfs01

# Release Procedure

cdem  
mvn clean install

./install [--zip-only]