em 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**
  + Difference between “standard” and “gp2” volume type. It seems that new instances are created with standard volumes.
  + Resolve the lsblk problem – format those disks.
* Hundreds of NFS clients hitting the servers?