

patron d'un script bash bien construit:

```
#!/bin/bash

."$dirname$0)/common/ec2"

usage(){
    echo "${BASENAME} [-h] -B <bucketname> <instance-id> [<instance-id>...]"
    log "-a script to terminate all provided instances and the delete"
    log "the associated bucket."
}

delete_instances() {
...
}

empty_bucket() {
...
}

free_ip() {
...
}
```

-En tête du script #!
-Fonction en bash function my_function() {}
-variable interne à la fonction

```
    complex_function() {
        #....
        local status="${?}"
        if[${status} -ne 0] ; then
            log "a command faile in $FUNCNAME with status ${status}"
        fi
        #....
    }
```

-fonction usage(){}
-libraires
-unify_args()
-remote_open()

<http://www.linuxjournal.com/content/bash-associative-arrays>
http://fr.wikipedia.org/wiki/Programmation_proc%C3%A9durale
<http://pmd.sourceforge.net/cpd.html>
<http://tldp.org/LDP/abs/html/internalvariables.html>
<http://puppetlabs.com/blog/speed-up-ssh-by-reusing-connections>

Plus loins

-protéger les variables

- Overquoting
- variable global en lecture seul
- tester la nullité des variables et des arguments
- Gérer les arguments
 - getopt
- gestion des erreurs
 - gérer les erreurs dans les commandes imbriquées
- créer un fichier temporaire
- processus et PID

Linux magazine n°192 avril 2016