LOG

Table of Contents

Command Cheat

```
cat tonFichier | awk '{gsub("\[|:.* ","",$4) tab[$4" - "$1]++} END {for(i in tab)
print i "occurence: "tab[i]}'
```

gsub("\[|:.* ","",$4) : je substitus [:. en RIEN et j'affiche $4 je crée un tableau et j'affiche les resultats sans duplication avec leur occurence sur l'ensemble des sorties

- rechercher que les messages d'erreurs avec AWK

  ```
  awk '/.err>/ {print}' /var/log/auth.log
  ```

Commande Linux:

Manipulation de text : cut et awk

```
    cut -f 1 -d : /etc/passwd
    awk -F : {'print $1' } /etc/passwd
> les deux commandes renvoi le même resultat
```

Logback file

- Logback (Logback est l'un des frameworks de journalisation les plus utilisés dans la communauté Java. Il remplace son prédécesseur, Log4j. Logback offre une implémentation plus rapide que Log4j, offre plus d'options de configuration et plus de flexibilité dans l'archivage des anciens fichiers journaux.)

Les niveaux possibles sont, par ordre de priorité: TRACE, DEBUG, INFO, WARN et ERROR. Chaque niveau a une méthode correspondante que nous utilisons pour enregistrer un message à ce niveau.

> logger.debug("Hello la terre"); > rootLogger.setLevel(Level.ERROR); > logger.warn("This message is not logged because WARN < ERROR."); > logger.error("This is logged.");

Nous voyons ces messages lorsque nous exécutons cet extrait:

```
20:44:44.241 [main] DEBUG monWorkspace.logback - Hello la terre
20:44:44.243 [main] DEBUG monWorkspace.logback - This message is logged because DEBUG
== DEBUG.
20:44:44.243 [main] ERROR monWorkspace.logback - This is logged.
```

Catalina.out file

- Catalina.out fichier générer par catalina.sh, script pour le lancer Tomcat. Catalina.out contient juste tous les ecritures de sortie standard java system.out et system.err.

Rotation de **catalina.out** Il suffit de: . Supprimer la capture swallowoutout = mettre à true swallowOutput dans Server.xml . ou empecher l'application de se connecter sur la sortie standard system.out & system.err

Garbage collector file

- gc.out Garbage Collector - overheadLimitExceed, sous class de java.Lang.VirtualMachineError. Le fichier gc.out est généré quand la limite memoire est atteinte c'est à dire que la JVM a passé trop de temps effectuer un nétoyage de la mémoire.

AQ: - utilisation de ressource systeme

CE QU'ON VEUT: Digeste sans tag Debug

ATTENDU - une format de sortie Minfi et ELK

RESULTAT

Rotation et Archivage des LOG?

Rotation

Utiliser service **logrotate** (linux): Point fort: - archive et supprime les anciens fichiers - compression des fichiers Point faible - trop de quantité de données read and write (usure de disque) - consomation en resource

- prendre en compte la taille des fichiers logs
- utilisation de logrotate (/etc/logrotate.conf)

```
# see "man logrotate" for details
# rotate log files weekly
weekly
```

```
# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog
```

```
# keep 4 weeks worth of backlogs
rotate 4
```

```
# create new (empty) log files after rotating old ones
create
```

```
# uncomment this if you want your log files compressed
# compress
```

```
# packages drop log rotation information into this directory
include /etc/logrotate.d
```

```
# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}
```

```
/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}
```

```
# system-specific logs may be configured here
```

Archivage de log

- Existe il un dépot distant pour archiver ou on l'archive sur la VM? Git ou VM à la limite si sur git, il faut un cleanUP au bout de certain temps, exemple 1 mois
- à quel frequence? par semaine (logrotate)
- cleanup au bout de combien de temps?

Audit de LOG Objectifs - Adapter les fichiers logs d'application aux recommandations - Faciliter l'exploitation des logs de chaque backends - Avoir un monitoring en temps réel sur Kibana **Évaluation des besoins** - Minimum de fichier log dont application.log et access.log - Le contenu des logs doit être digest dans le but de les exploiter facilement - Rotation de log backend tous les X heures / jours - Cleanup des archives **Recommandations sur le projet**

Amélioration

Recommandation

Impacte sur la qualité - Amélioration de la performance et l'utilisation des ressources mémoire. - Standard des fichiers logs **Conclusion**