

Dicas para Desenvolvedores

O link para o treinamento gravado é: <https://web.microsoftstream.com/video/7f617153-fb95-4cb2-a769-8c78de7408b6>

- Acessar a máquina na qual deseja fazer a limpeza por meio do Session Manager (Console AWS Systems Manager Session Manager)
- Executar **'sudo su'** para acessar como root
- Executar **'df -h'** caso queira verificar como está cada filesystem
- Caso queira ver qual diretório está consumindo mais espaço, execute no /home: **'du -k | sort -n'** ou **'du -hd1 | sort -n'**
- No caso das máquinas do Online/Extração (SW7) o que geralmente ocupa a máquina são os core dumps – verificar a existência de core dumps de todos usuários com o comando **'ls */site/*/tmp/**/core*'** – Para ver todos os core dumps da máquina: **'find /home -regextype sed -regex ".*core\.[0-9]*"'**
- Remover os arquivos com o comando **'rm -f */site/*/tmp/**/core*'**
- Executar **'df -h'** novamente para verificar os filesystems

TOME CUIDADO AO EXECUTAR 'rm -rf' COM USUARIO ROOT!!!

- Em algumas situações quando se copiam programas feitos no Windows para o Linux, alguns caracteres indesejáveis surgem, gerando o erro abaixo.
- Supondo que o script escrito em Shell tenha sido feito no Windows com o nome de `teste.sh`. Ao tentar executá-lo no Linux, a mensagem de erro surgirá:

`./teste.sh: line 2: $'\r': command not found`

- Esse `\r` é o problema. E isso pode ser resolvido por meio da instalação do `dos2unix` com o comando abaixo:

`sudo apt install dos2unix`

- Após a instalação, basta digitar:

`dos2unix teste.sh`

- Para remover os caracteres estranhos. Dessa forma, é possível utilizar o seu script sem problemas.

- Para quem não tem o `dos2unix` instalado (e não quer instalá-lo):
 - Para fazer isso para todos os arquivos no diretório atual, execute:

```
for i in *;do if [[ -f $i ]]; then sed -i 's/\r$//' "$i"; fi; done
```

Valgrind é uma ótima ferramenta que auxilia na detecção de memory leaks no sistema, para executar ele é necessário colocar a linha do valgrind antes da chamada do engine da aplicação.

Exemplo de como utilizar o Valgrind:

- Dentro de `~/POS/tasks/acqsvr_crt_537` foi colocada a linha exatamente antes da chamada do engine: `valgrind --leak-check=full --error-limit=no --suppressions=<arquivo com supressões> --log-file=<arquivo com resultado da análise>`, colocando o arquivo de supressões `minimal.supp` e colocando o nome desejado do log.

Ajuste de Timeout de resposta na transação:

- `~/site/POS/cfg/istparam.cfg`: Se necessário alterar os timeout de banco de dados (`database.lag_alert_ms`, `restart_on_timeout_byapp_time`)
- `~/sw-bin/go`: Alterar o parâmetro `-T` da chamada do jar
- Se necessário alterar o timeout do banco Oracle/Postgres usado pela aplicação, Verificar a classe `connection` dentro da `gen-lib`.

O log irá apontar várias métricas além dos leaks, como leitura/gravação inválida de memória, valores não inicializados, entre outros, porém se o foco é o memory leak, as informações importantes vem após "HEAP SUMMARY", que é aonde começa o rastreamento da memória da aplicação.

Link das EAKs envolvidas até o momento com o rastreamento de memory leaks do Online:

- <http://jira/browse/EAK-6225>
- <http://jira/browse/EAK-6876>
- <http://jira/browse/EAK-7085>
- <http://jira/browse/EAK-7197>

(Inicializa o repo, não precisa ser feito toda vez)






```
terraform init '-input=false' '-backend=true' '-backend-config=bucket=redecad-terraform' '-backend-config=key=SW7/<id repo>.tfstate' '-backend-config=region=us-east-1'
```

Pegando o id do repositório, na página inicial do repositório:



lambda-pos-region-replication

Project ID: 10316

 **6 Commits**  **3 Branches**  **1 Tag**  **778 KB Files**  **12.9 MB Storage**

(Valida se o código do terraform tem algum erro gritante)

Terraform validate

(Planeja tudo que será executado quando for aplicado, irá mostrar oque irá criar, modificar ou excluir)

```
terraform plan -detailed-exitcode '-input=false' '-var-file=./_variables/_defaults.tfvars' '-var-file=_variables/dev.tfvars'
```

(Aplica toda a estrutura de terraform na AWS de desenvolvimento, eu prefiro fazer essa etapa pela pipeline por questão de facilidade de registro do que está aplicado ou não;)

```
terraform apply -auto-approve '-input=false' '-var-file=./_variables/_defaults.tfvars' '-var-file=_variables/dev.tfvars'
```

Dicas: [SW Sandbox Online](#)

- [\[Git\] Merge para Master](#)
- [Angular](#)
- [Cursos de Git](#)
- [Transferir informação do BD para Issuer](#)