

# Monitorizar WildFly con CLI y sus habilidades de bash

---

En realidad, hay muchas herramientas libres y de código abierto para el seguimiento del servidor de aplicaciones JBoss / JBoss. En este artículo vamos a aprender a crear una cáscara de monitoreo usando tan poco como **jboss cli** y **bash** y **awk**.

En algunos casos hay que supervisar uno o dos atributos clave de la configuración del servidor de aplicaciones y recibir una alerta si el atributo ha alcanzado un valor crítico. En tal escenario, no es digno de instalar una herramienta compleja, especialmente si usted tiene algunas secuencias de comandos básicos shell de Linux.

Vamos a aprender cómo controlar algunos atributos clave, tales como las fuentes de datos disponibles y la memoria de almacenamiento dinámico libre en el servidor de aplicaciones.

Vamos a empezar desde los **CLI commands**. Aquí es cómo controlar el número de **conexiones disponibles** en una piscina DB:

```
/jboss-cli.sh -c
"/subsystem=datasources/data-source=MYDS/statistics=pool:read-attribute
(name="AvailableCount") "
{
    "outcome" => "success",
    "result" => "100"
}
```

Por lo tanto, lo que se quiere capturar es el valor de "número". Con tan poco como algunas habilidades **awk** que podemos hacer lo siguiente:

```
availpool=$(./jboss-cli.sh -c
"/subsystem=datasources/data-source=MYDS/statistics=pool:read-attribute(name
="AvailableCount")" | awk '/result/{gsub("\\"", "", $3); print $3}')

if [ $availpool -le 5 ]
```

```

then

    echo "Database Connections Pool getting low! Reaming connections :
    $availpool" | mail -s "JBoss AS Alert" adminATadmin.com

fi

```

Muchas veces se puede calcular por simple aritmética de otras métricas Por ejemplo, si desea ser informado acerca de **memory free** en el servidor, usted tiene para calcular: **max memory - memory** utilizada como sigue:

```

[standalone@127.0.0.1:9990 /]
/core-service=platform-mbean/type=memory:read-attribute(name=heap-memory-usage)
{
    "outcome" => "success",
    "result" => {
        "init" => 67108864L,
        "used" => 158278728L,
        "committed" => 341835776L,
        "max" => 477102080L
    }
}

```

Tendremos que variables para este propósito:

```

heapused=$(./jboss-cli.sh -c
"/core-service=platform-mbean/type=memory:read-attribute(name=heap-memory-usage)" | grep "used" | awk '{print $3}' | sed 's/L, //' )

heapmax=$(./jboss-cli.sh -c
"/core-service=platform-mbean/type=memory:read-attribute(name=heap-memory-usage)" | grep "max" | awk '{print $3}' | sed 's/L, //' )

freememory=$((heapmax - heapused))

if [ $freememory -le 128000000 ]

then

    echo "JVM Heap memory getting low: Remaining: $freememory bytes" |
    mail -s "JBoss AS Alert" adminATadmin.com

fi

```

En el guión anterior, estamos enviando una alerta por correo si la cantidad de memoria de almacenamiento dinámico libre cae por debajo de 128 MB. Pan comido !

Tenga en cuenta que hemos asumido que el guión `jboss-cli.sh` va a conectar automáticamente a la dirección de bucle de retorno. Si no desea incluir la dirección IP y el puerto de la interfaz de gestión en el guión, se sugiere introducir el valor por defecto para que en **`jboss- cli.xml`**