



Cybermen - SPC
Public Edition, version 2.5.

Guía de Instalación y configuración

Fecha de generación: 02/01/2015

Autor: Eduardo O. Frigerio.
Copyright: 2014 - Eduardo O. Frigerio

Contenido

CONTENIDO	2
INTRODUCCIÓN.....	3
DE LA EDICIÓN PÚBLICA (THE PUBLIC EDITION)	3
DEL IDENTIFICADOR CLPACKAGEINFO	3
COMO EMPEZAR	5
REQUERIMIENTOS PREVIOS	5
REQUERIMIENTOS DE HARDWARE Y DE SISTEMA OPERATIVO.	5
CONFIGURACIÓN DE JAVA SE	5
INSTALACIÓN Y CONFIGURACIÓN DE CYBERMEN - SPC.....	6
INSTALACIÓN	6
<i>Recomendación</i>	7
CONFIGURACIÓN DE UN CLÚSTER.....	7
IDENTIFICACIÓN DE LOS NODOS	7
PARÁMETROS DEL SISTEMA.....	8
AJUSTES A NIVEL DE JVM.....	8
DATA LOGGING, REGISTRO Y CONFIGURACIÓN	9
<i>Logging in cluster (LMonitor),</i>	10
CONFIGURACIÓN DEL CANAL DE COMUNICACIONES	11
<i>El parámetro jgroups.bind_addr</i>	11
CONFIGURACIÓN DEL SERVICIO SHUTDOWN	12
<i>.period</i>	12
<i>.topCurrentThread</i>	12
<i>.runFinalization</i>	12
<i>.topFreeMemory</i>	13
<i>.timeShutdown</i>	13
<i>.monitorUse</i>	13
<i>.timeOfWaitWithoutConnection</i>	13
<i>Recomendación importante</i>	14
SERVICIO DEPLOY, SECURITY CONTEXT & SECURITY POLICY	14
VOCABULARIO.....	15

Introducción

Cybermen - SPC (Services Platform in Cluster) es una plataforma de servicios y clúster de procesamiento distribuido, orientado principalmente al desarrollo de aplicaciones de supervisión, control y adquisición de datos (SCADA) y control distribuido (DCS).

Cybermen - SPC de por sí, no hace a las aplicaciones SCADA o DCS, pero constituye una base para el desarrollo de este tipo de soluciones, basándose para las mismas en una arquitectura de servicios colaborativos (SOA) y operando en un clúster de procesamiento.

Por estar basado en Java SE, Cybermen - SPC no solo es cross-platform (opera en cualquier sistema operativo que soporte java) sino que al mantenerse dentro de Standard Edition y operar con un número acotado de dependencias externas, funciona con bajos requerimientos de hardware. Tanto, que sus nodos de procesamiento son capaces de operar en hardware ARM v7 con soporte para Java SE Embedded

De la edición pública (the public edition)

Existen tres distribuciones distintas de Cybermen - SPC, una de grado empresarial (the enterprise edition), otra de nivel físico / controlador eléctrico (the hardware edition), y finalmente una de conocimiento público (the public edition)

La edición pública constituye el mínimo común denominador de las tres distribuciones, además de la base de estandarización para el desarrollo de servicios en Cybermen.

Esto significa que cualquier servicio construido y probado en la edición pública debería funcionar correctamente en la edición empresarial, y en la hardware edition.

Siempre que compartan el mismo número de versión del CLPackageInfo ([ver referencia al tema en esta guía](#)) y que el hardware / OS lo permita.

Del identificador CLPackageInfo

Al problema normal de incompatibilidad entre versiones que suele tener cualquier software / API, en el caso de Cybermen - SPC, y al tratarse de un clúster con N instancias / instalaciones individuales, se le agrega la complejidad de tener que actualizar cada una de las instalaciones sin "romper" la integridad del clúster.

A esto se le debe sumar además el hecho de que es posible crear "clúster compuestos" por nodos de las distintas distribuciones (enterprise, hardware y public edition) y que los mismos operen de forma normal.

Para lidiar con esta complejidad, es que se establece un número de identificación único del kernel de base.

Esto significa que todas las distribuciones y versiones de los nodos instalados, de los clientes de Cybermen, de las herramientas de administración y de desarrollo, solo serán visibles dentro del mismo clúster cuando posean el mismo número de versión del CLPackageInfo.

Dicho de otra forma, de introducirse cambios menores en la plataforma que no rompan compatibilidad hacia atrás, este identificador permanecerá inalterado después de un update de las instalaciones, y el nodo actualizado, seguirá permaneciendo al mismo clúster

Por el contrario, si el update rompe compatibilidad hacia atrás el identificador cambiará y los nodos actualizados conformarán un clúster separado e independiente de la versión anterior.

Como empezar

Requerimientos previos

Usted debe contar con suficiente espacio de almacenamiento para instalar el JRE (Java Runtime Environment) correspondiente a su sistema operativo, los nodos de procesamiento de Cybermen - SPC que sean necesarios (este parámetro está condicionado por la arquitectura de despliegue y cantidad de maquinas que se desee utilizar) y las aplicaciones a ser utilizadas dentro de cada nodo.

Antes de instalar uno o más nodos de Cybermen - SPC se debe contar con una instalación activa de java.

Desde que Cybermen - SPC está basado en Java SE puede operar en cualquier sistema operativo que soporte Java SE.

Requerimientos de hardware y de sistema operativo.

Teóricamente, los nodos de Cybermen - SPC deberían ser capaces de operar en cualquier arquitectura / sistemas operativos con soporte para Java SE o Java SE Embedded versiones 7.0 o superior.

En la práctica Cybermen - SPC ha sido probado en las siguientes configuraciones.

Sistema operativo.	Arquitectura	JRE
Raspbian OS.	ARM11	OpenJDK v7.x (os provide)
Raspbian OS.	ARM11	Oracle Java SE Embedded v7.x
Windows 7 o superior	x86	Oracle JDK i586 v7.x
Windows 7 o superior	x64	Oracle JDK x64 v7.x
Ubuntu	x86	OpenJDK (os provide)
Ubuntu	x86	Oracle JDK i586 v7.x
Ubuntu	X64	OpenJDK (os provide)
Ubuntu	X64	Oracle JDK x64 v7.x

Configuración de Java SE

Como mencionado anteriormente, usted debe contar con una instalación activa del JRE 7.0 o superior antes de poder instalar uno o más nodos de Cybermen - SPC.

En arquitectura x64 usted puede instalar el JRE de 32 o de 64 bit conforme a sus requerimientos.

No obstante, y dado que cada nodo de Cybermen - SPC opera en esencia como un servidor en sí mismo, es recomendable instalar el JDK correspondiente a su plataforma (esto permite utilizar la versión de HotSpot optimizada para servidores).

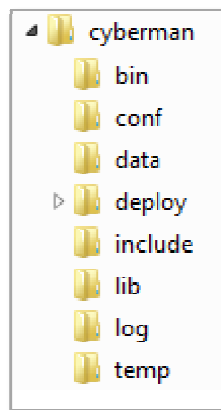
Nota: Ver también la sección "[Ajustes a nivel de JVM](#)" en esta guía de instalación.

Instalación y configuración de Cybermen - SPC

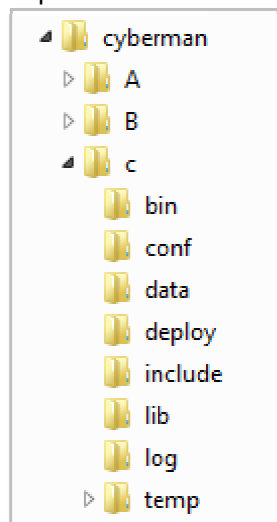
Instalación

Desempaque el archivo Cybermen.zip en el directorio correspondiente utilizando para ello la herramienta provista por su plataforma.

Al ejecutar esta acción deberá obtener un árbol de directorios con la siguiente estructura.



Esta instalación, corresponde a un nodo de procesamiento de un clúster, debiendo repetir la operación tantas veces como nodos deseen instalarse en el mismo equipo.



Al tratarse de un clúster de procesamiento (y no de un servidor), Cybermen - SPC requiere de no menos de 2 nodos para poder operar.

No obstante es posible que con solo 2 nodos de procesamiento se produzcan detenciones o demoras en los servicios si es que uno de estos nodos es detenido, o el servicio de Shutdown está en ejecución dentro de uno de ellos ([ver configuración de Shutdown en esta guía](#)), por lo que se recomienda no conformar un clúster con menos de tres nodos de procesamiento.

Recomendación

En equipos antiguos, hardware pobre, o de sistemas incluidos (como es el caso de la mayoría de los procesadores ARM o el resto de las arquitecturas soportadas por [Java SE Embedded](#)).

No es recomendable instalar más de un nodo en la misma máquina.

Configuración de un clúster

Cybermen - SPC permite instalar más de un clúster de procesamiento dentro de la misma red, e incluso instalar nodos correspondientes a distintos clústeres en el mismo hardware

Para poder distinguir entre un clúster y otro es necesario asignarle un nombre a cada uno.

Esto se hace desde el archivo "cyberconfig.xml " de la carpeta "conf" por medio del parámetro "cybermen.clusterName "

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE properties (View Source for full doctype...)>
<properties version="1.0">
  <entry key="cybermen.clusterName">cybermen</entry>
  <entry key="cybermen.fileHandler.pattern">'.yyyy-MM-dd</entry>
</properties>
```

Por defecto, todos los nodos pertenecen al clúster "cybermen", siempre que este parámetro no sea alterado.

Identificación de los nodos

Cada nodo de procesamiento en un clúster de Cybermen - SPC posee un ID que lo identifica frente al resto del clúster y ante sí mismo.

Este identificador se utiliza además para evitar que se levanten dos o más instancias de ejecución de la misma instalación, evitando de esta forma los problemas de corrupción en ejecución relacionados a tener más de una instancia de ejecución del mismo servidor de aplicaciones.

De no especificarse como parámetro de configuración, un nodo de procesamiento generará el ID como el hashCode resultante de los datos de la maquina y del directorio sobre el cual se encuentra instalado.

No obstante (y dado que este identificador es empleado en varias operaciones) es recomendable especificar el ID como nombre único del nodo, siguiendo una regla o convención lógica, por medio del parámetro "cybermen.installationID"

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE properties (View Source for full doctype...)>
<properties version="1.0">
  <entry key="cybermen.clusterName">cybermen</entry>
  <entry key="cybermen.fileHandler.pattern">'.yyyy-MM-dd</entry>
  <entry key="cybermen.installationID">Alfa</entry>
</properties>
```

Esto se hace desde el archivo "cyberconfig.xml " de la carpeta "conf" por medio del parámetro "cybermen.installationID "

Parámetros del sistema

Adicionalmente Cybermen permite agregar System Property (ver métodos [getProperty en la clase java.lang.System de Java SE](#)) simplemente agregando las mismas en el archivo "cyberconfig.xml".

Estos parámetros pueden ser requeridos tanto por librerías de terceros como por servicios de Cybermen, pidiendo personalizar los mismos en cada nodo de procesamiento de forma individual.

Ajustes a nivel de JVM

Modo de ejecución, cantidad de memoria mínima o máxima, o cualquiera de los [flags de arranque de la JVM](#) deben especificarse en el archivo "jvm-options.xml" de la carpeta "conf" por medio del parámetro "node"

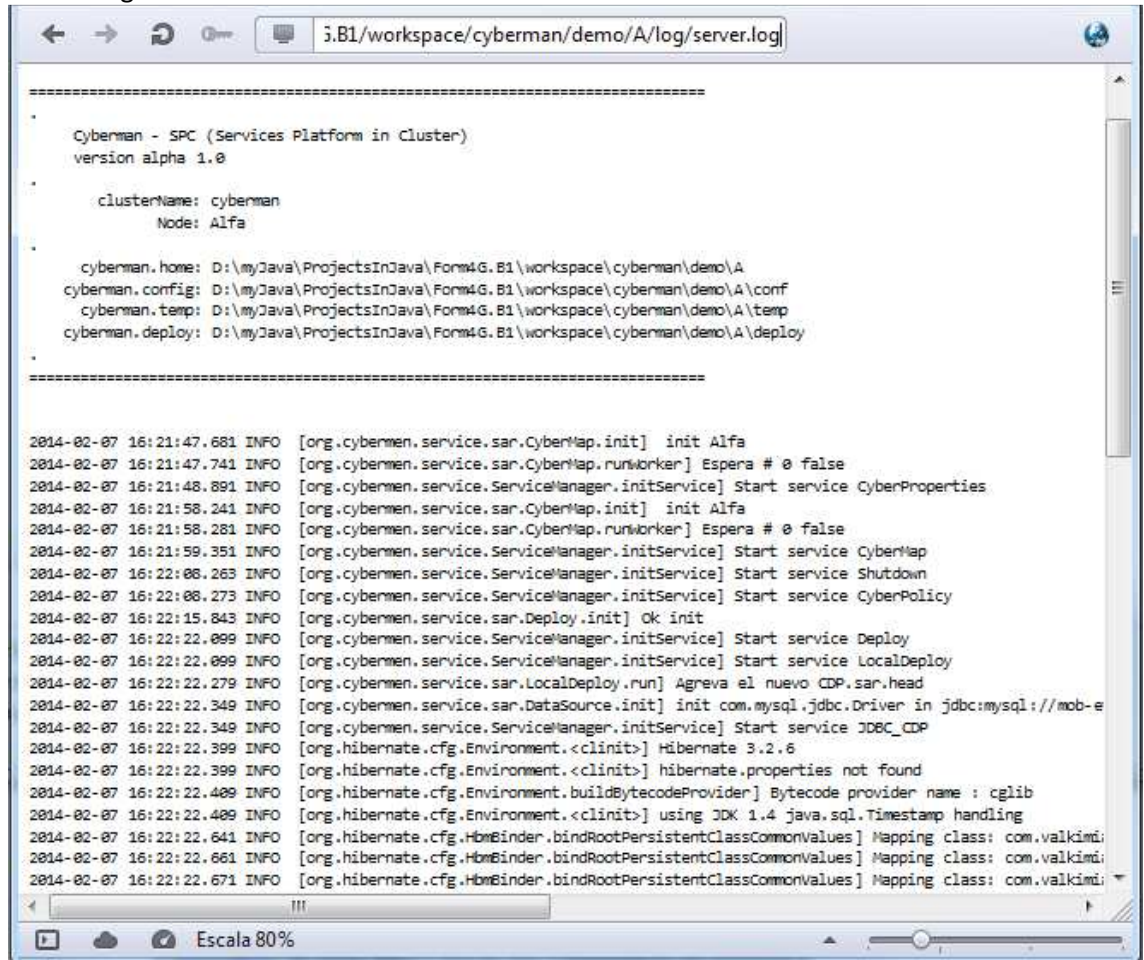
```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE properties (View Source for full doctype...)>
<properties version="1.0">
  <entry key="node">-server -Xms128m -Xmx512m -Xincgc -
Djava.net.preferIPv4Stack=true</entry>
</properties>
```

Nota: Esta forma de configurar los flags de la JVM se establece desde la versión 2.3 de Cybermen - SPC, con miras a reducir y estandarizar el numero de script de arranque de cada plataforma / os soportado, establecer una base en común entre las distintas distribuciones (enterprise, public y hardware edition), como también con miras a una futura herramienta centralizada de gestión, administración y monitoreo de los nodos.

Data Logging, registro y configuración

Cyberman - SPC emplea como mecanismo de data logging el "[Java Logging Technology](#)" provisto por Java SE desde la versión 5.0

El archivo de registro de log se genera dentro del directorio "log" con el nombre "server.log"



```
=====
Cyberman - SPC (Services Platform in Cluster)
version alpha 1.0

  clusterName: cyberman
    Node: Alfa

  cyberman.home: D:\myJava\ProjectsInJava\Form4G.B1\workspace\cyberman\demo\A
  cyberman.config: D:\myJava\ProjectsInJava\Form4G.B1\workspace\cyberman\demo\A\conf
  cyberman.temp: D:\myJava\ProjectsInJava\Form4G.B1\workspace\cyberman\demo\A\temp
  cyberman.deploy: D:\myJava\ProjectsInJava\Form4G.B1\workspace\cyberman\demo\A\deploy
=====

2014-02-07 16:21:47.681 INFO [org.cyberman.service.sar.CyberMap.init] init Alfa
2014-02-07 16:21:47.741 INFO [org.cyberman.service.sar.CyberMap.runworker] Espera # 0 false
2014-02-07 16:21:48.891 INFO [org.cyberman.service.ServiceManager.initService] Start service CyberProperties
2014-02-07 16:21:58.241 INFO [org.cyberman.service.sar.CyberMap.init] init Alfa
2014-02-07 16:21:58.281 INFO [org.cyberman.service.sar.CyberMap.runworker] Espera # 0 false
2014-02-07 16:21:59.351 INFO [org.cyberman.service.ServiceManager.initService] Start service CyberMap
2014-02-07 16:22:08.263 INFO [org.cyberman.service.ServiceManager.initService] Start service Shutdown
2014-02-07 16:22:08.273 INFO [org.cyberman.service.ServiceManager.initService] Start service CyberPolicy
2014-02-07 16:22:15.843 INFO [org.cyberman.service.sar.Deploy.init] Ok init
2014-02-07 16:22:22.099 INFO [org.cyberman.service.ServiceManager.initService] Start service Deploy
2014-02-07 16:22:22.099 INFO [org.cyberman.service.ServiceManager.initService] Start service LocalDeploy
2014-02-07 16:22:22.279 INFO [org.cyberman.service.sar.LocalDeploy.run] Agrega el nuevo CDP.sar.head
2014-02-07 16:22:22.349 INFO [org.cyberman.service.sar.DataSource.init] init com.mysql.jdbc.Driver in jdbc:mysql://mob-e
2014-02-07 16:22:22.399 INFO [org.cyberman.service.ServiceManager.initService] Start service JDBC_CDP
2014-02-07 16:22:22.399 INFO [org.hibernate.cfg.Environment.<clinit>] Hibernate 3.2.6
2014-02-07 16:22:22.399 INFO [org.hibernate.cfg.Environment.<clinit>] hibernate.properties not found
2014-02-07 16:22:22.409 INFO [org.hibernate.cfg.Environment.buildBytecodeProvider] Bytecode provider name : cglib
2014-02-07 16:22:22.409 INFO [org.hibernate.cfg.Environment.<clinit>] using JDK 1.4 java.sql.Timestamp handling
2014-02-07 16:22:22.641 INFO [org.hibernate.cfg.HbmBinder.bindRootPersistentClassCommonValues] Mapping class: com.valkimi
2014-02-07 16:22:22.661 INFO [org.hibernate.cfg.HbmBinder.bindRootPersistentClassCommonValues] Mapping class: com.valkimi
2014-02-07 16:22:22.671 INFO [org.hibernate.cfg.HbmBinder.bindRootPersistentClassCommonValues] Mapping class: com.valkimi
=====
```

Cyberman - SPC también ofrece un mecanismo de rotación de log. Esto es que por final del día creará un archivo ".log" conteniendo el log de ejecución de ese día y especificando en el nombre del archivo LOG año mes y día del log.

El archivo "server.log", entonces, pasa a contener únicamente la información de registro correspondiente a las operaciones desde las cero horas del día en curso.

Para alterar la latencia del mecanismo de rotación de log, se deberá cambiar el parámetro "cybermen.fileHandler.pattern" del archivo "cyberconfig.xml" en el directorio "conf"

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE properties (View Source for full doctype...)>
<properties version="1.0">
<entry key="cybermen.clusterName">cybermen</entry>
<entry key="cybermen.fileHandler.pattern">'.yyyy-MM-dd</entry>
<entry key="cybermen.installationID">Alfa</entry>
</properties>
```

En el ejemplo anterior el mecanismo de rotación y compresión, por final del día 10/01/2014, generará un archivo "server.2014.10.01.log".

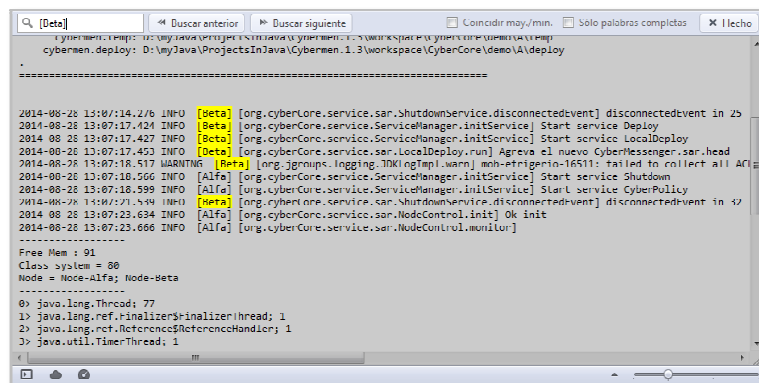
Para alterar tanto la latencia como el formato de fecha en el nombre, se deben utilizar los formatos de fecha / hora válidos para la clase [SimpleDateFormat](#) de Standard Edition.

Logging in cluster (LMonitor),

Cybermen - SPC ofrece un mecanismo por medio del cual es posible monitoriar todo el clúster de procesamiento desde una única consola.

Esto es ejecutando el comando "monitor" del directorio "/bin".

Además de mostrar en consola la información de cada nodo del clúster, identificando el origen del registro de log.

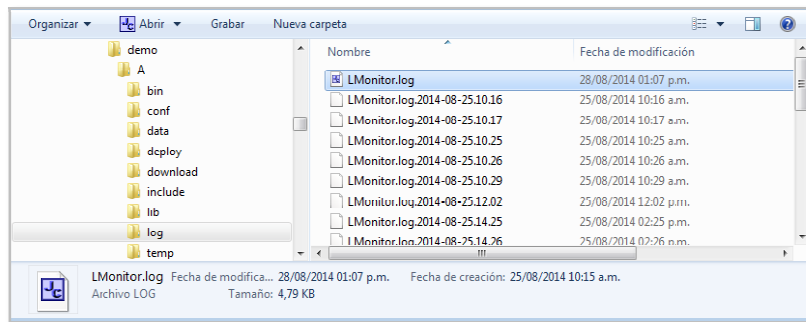


```
[Beta] [org.cyberCore.service.sar.ShutdownService.disconnectedEvent] disconnectedEvent in 2b
2014-08-28 13:07:17.424 INFO [Beta] [org.cyberCore.service.ServiceManager.initService] Start service Deploy
2014-08-28 13:07:17.427 INFO [Beta] [org.cyberCore.service.ServiceManager.initService] Start service LocalDeploy
2014-08-28 13:07:17.453 INFO [Beta] [org.cyberCore.service.sar.LocalDeploy.run] Agrega el nuevo CyberMessenger.sar.head
2014-08-28 13:07:18.517 WARNING [Beta] [org.jgroups.logging.JMXImpl.warn] mnh-efrigaria-16511: failed to collect all ATs
2014-08-28 13:07:18.566 INFO [Alfa] [org.cyberCore.service.ServiceManager.initService] Start service Shutdown
2014-08-28 13:07:18.599 INFO [Alfa] [org.cyberCore.service.ServiceManager.initService] Start service CyberPully
2014-08-28 13:07:21.444 INFO [Beta] [org.cyberCore.service.sar.ShutdownService.disconnectedEvent] disconnectedEvent in 4
2014-08-28 13:07:23.634 INFO [Alfa] [org.cyberCore.service.sar.NodeControl.init] Ok init
2014-08-28 13:07:23.666 INFO [Alfa] [org.cyberCore.service.sar.NodeControl.monitor]

Free Mem : 91
Class system = 88
Node = Node-Alfa; Node-Beta

0> java.lang.Thread: 77
1> java.lang.ref.Finalizer$FinalizerThread: 1
2> java.lang.ref.Reference$ReferenceHandler: 1
3> java.util.TimerThread: 1
```

Dicha información es almacenada en el directorio "log" en el archivo "LMonitor.log"



Nótese que este registro adopta la configuración de rotación del archivo de LOG, especificada como parámetro de configuración del nodo (el parámetro "cybermen.fileHandler.pattern" del archivo "cyberconfig.xml" en el directorio "conf").

Recomendación,

Dado que para lograr la registración de los log de un clúster en un solo repositorio, es necesario transferir esta información por medio de la red, y que los registros de evento (en general) suelen ser bastante verbosos (particularmente cuando hay problemas).

Se recomienda limitar el número de instancias del monitor, con el fin de no generar tráfico de red excesivo y acotar el alcance del "efecto-observador de Jordan" o el "efecto Hawthorne" (según sea el caso).

Configuración del canal de comunicaciones

Cybermen - SPC utiliza [JGroups](#) v3.2.12.Final como toolkit de mensajería fiable para las comunicaciones multipunto.

Por defecto Cybermen - SPC emplea la pila protocolar UDP por lo que solo son visibles entre sí aquellos nodos de un clúster que se encuentren dentro del mismo segmento de red.

Esta pila protocolar se encuentra especificada en el archivo "jgroups.xml" del directorio "conf" y se corresponde a la entregada por JGroups para la [pila protocolar UDP](#), tal como figura en [su documentación](#).

Si se desea alterar esta pila protocolar por cualquiera de las otras soportadas por JGroups, se deberá alterar el archivo "jgroups.xml" conforme a las especificaciones de la documentación de JGroups correspondiente a la versión 3.2.12.Final

El parámetro jgroups.bind_addr

Desde la pila protocolar UDP, Si un nodo de Cybermen está conectado a más de una red al mismo tiempo, es necesario modificar el script de arranque agregando el parámetro "-Djgroups.bind_addr=127.0.0.1" indicando la dirección IP de la placa de red, sobre la cual deberá canalizarse el tráfico UDP.

Configuración del servicio Shutdown

Un problema común en el desarrollo de aplicaciones es el conocido como degradación de la JVM, este proceso ocurre por errores de programación, sean estos de las aplicaciones, en los servidores, o en librerías de terceros.

El problema consiste en que la JVM consume cada vez mas memoria, o cada vez mas procesador, o ambas cosas, de manera regular y progresiva.

Dependiendo de la naturaleza del error este proceso puede tardar horas, días, meses o incluso años.

Pero (inevitablemente), será necesario reiniciar para liberar recursos.

Al tratarse de un clúster de procesamiento y no de un servidor, Cybermen - SPC ofrece la opción de identificar este problema de manera preventiva dentro de un nodo de procesamiento, reiniciar el nodo de forma segura y sin afectar a los servicios en ejecución.

Ver la "Guía de desarrollo de aplicaciones" para entender este punto.

La forma de configurar esto es por medio del servicio Shutdown en el descriptor de arranque (archivo "cyber-service.xml " en el directorio "conf").

```
<entry key="Shutdown.classtName"> org.cyberCore.service.sar.ShutdownService</entry>
<entry key="Shutdown.topCurrentThread">400</entry>
<entry key="Shutdown.topFreeMemory">5</entry>
<entry key="Shutdown.timeShutdown">24:00:00,000</entry>
<entry key="Shutdown.runFinalization">00:20:00,000</entry>
<entry key="Shutdown.timeOfWaitWithoutConnection">00:05:00,000</entry>
<entry key="Shutdown.monitorUse">false</entry>
<entry key="Shutdown.period">5000</entry>
```

Se pueden especificar cualquiera de los parámetros que se describen a continuación.

.period

Periodo de latencia.

En el ejemplo anterior, este parámetro está fijado en 5000 milisegundos.

Eso significa que el servicio de Shutdown verificará el estado del nodo con una latencia de 5 segundos

.topCurrentThread

Cantidad máxima de thread en ejecución.

En el ejemplo anterior, este parámetro está fijado en 400 thread.

Eso significa que si en un nodo de procesamiento se supera los 400 thread, el nodo se reiniciará de forma segura automáticamente

.runFinalization

System.currentTimeMillis().

En el ejemplo anterior, este parámetro está fijado en 20 minutos.

Eso significa que cada 20 minutos se forzará la ejecución del garbage collector.

.topFreeMemory

Cantidad mínima de memoria libre.

En el ejemplo anterior, este parámetro está fijado en 5 MB.

Eso significa que si en un nodo de procesamiento, la JVM tiene menos de 5 megabit de memoria libre disponible. El servicio de Shutdown primero forzará la ejecución del garbage collector tratando de liberar memoria, y si no lo logra, entonces reiniciará de forma segura automáticamente el nodo.

.timeShutdown

Tiempo de cierre.

En el ejemplo anterior, este parámetro está fijado en 40 minutos.

Eso significa que los nodos de un clúster se reiniciarán de forma secuencial con intervalos de 40 minutos aun cuando no se detecten fallas en los mismos.

.monitorUse

Monitoreo de estado.

En el caso de estar activo (=true), el servicio publicará en el log de ejecución información sobre el estado general del nodo en cada ciclo de revisión (parámetro "period").

- Cantidad de memoria libre.
- Cantidad total de thread activos.
- Cantidad de thread por grupo o familia.
- ID de los nodos activos en el clúster

.timeOfWaitWithoutConnection

Tiempo sin conexión al clúster.

En el ejemplo anterior, este parámetro está fijado en 5 minutos.

Eso significa que si un nodo esta desconectado del resto del clúster durante un periodo de tiempo mayor a 5 minutos, el mismo se reiniciara de manera segura.

Aclaración

En algunos escenarios de despliegue, empleando la pila protocolar UDP de JGroups, y ante un evento de falla de red prolongado, se ha descubierto que no siempre JGroups logra restablecer la comunicación sobre el canal.

Tanto los servicios distribuidos como remotos de Cybermen, poseen la capacidad de registrar este evento, la duración del mismo, y actuar en consecuencia.

Por lo que este parámetro del servicio de Shutdown, se conserva como un recurso de última instancia.

Recomendación importante

Esto es solo una aspirina.

Shutdown permite mitigar los efectos del problema de degradación de la JVM pero no lo soluciona. Por lo que se deberá estar atento a la ejecución de los reinicios de parte de este servicio.

La solución del problema de degradación (como siempre) pasa por corregir el código defectuoso que lo está provocando.

Servicio Deploy, security context & security policy

Como mencionado en la "Guía de desarrollo de aplicaciones" el servicio Deploy, es el empleado desde las herramientas de desarrollo para las tareas de despliegue y remoción de un descriptor de servicio o de un paquete de servicios con su descriptor (archivos SAR) dentro de un clúster de procesamiento.

También (y conforme a lo mencionado en dicha guía) al tratarse de un servicio del tipo "RemoteInvokerService" se debe especificar un dominio de seguridad, y una política para dicho dominio, quienes son los responsables tanto de la autenticación de clientes externos como de la verificación de los roles de este cliente y su correlación con los especificados para el acceso al servicio o la invocación de un método de este.

En el caso del servicio Deploy, un cliente deberá poseer el rol de "developer" o el de "administrator" para acceder al servicio. y los de "deploy", "undeploy", "load", para poder emplear los métodos asociados a cada uno de estos roles.

El servicio de Deploy y su política de seguridad están declarados en el descriptor de arranque (archivo "cyber-service.xml " en el directorio "conf").

```
<entry key="CyberPolicy.classtName">org.cyberPacket.service.security.UsersRolesLogin</entry>
<entry key="CyberPolicy.usersProperties">login-users.properties</entry>
<entry key="CyberPolicy.rolesProperties">login-roles.properties</entry>

<entry key="Deploy.classtName">org.cyberPacket.service.sar.Deploy</entry>
<entry key="Deploy.security-domain">CyberPolicy</entry>
<entry key="Deploy.depends">NodeControl</entry>
<entry key="Deploy.period">5000</entry>
```

Por defecto, el servicio Deploy , emplea una política del tipo "UsersRolesLogin" y cuyos parámetros de autenticación de usuario y de roles por usuario, se encuentran declarados en los archivos "login-users.properties" y "login-roles.properties" de la carpeta "conf".

Adicionalmente se encuentra el servicio NodeControl para las tareas de "reset" y "shutdown" dentro de la misma política de seguridad.

```
<entry key="NodeControl.classtName">org.cyberCore.service.sar.NodeControl</entry>
<entry key="NodeControl.security-domain">CyberPolicy</entry>
```

Ver la sección LoginService de la "Guía de desarrollo de aplicaciones" para obtener más información.

Vocabulario

DCS (Distributed Control System): [Sistema de Control Distribuido](#)

SCADA (Control And Data Acquisition): [Supervisión, Control y Adquisición de Datos](#)

SOA (Service Oriented Architecture): [Arquitectura Orientada a Servicios](#)

Java SE ([Java Platform, Standard Edition](#)):

[Java SE Embedded](#)

HotSpot: información de referencia, "[JVM options: -client vs -server](#)"

Embedded system: [Sistemas incluidos.](#)

[Raspbian OS:](#)

JGroups: [toolkit y protocolo de mensajería multipunto.](#)

[Java Logging Technology](#)

[SimpleDateForma](#)