

The logo consists of a solid red triangle in the top-left corner, with a series of thin, white, nested L-shaped lines forming a staircase pattern along its hypotenuse.

RED HAT CONSULTING

# JBOSS CLI

The logo consists of a solid red triangle in the bottom-right corner, with a series of thin, white, nested L-shaped lines forming a staircase pattern along its hypotenuse.

**RED HAT CONSULTING**

# **OBJETIVOS DE ESTA SESIÓN**

---

Introducción a la Línea de Comando de Interfase (CLI) de JBoss

- JBoss EAP6 managment interface

**RED HAT CONSULTING**

# **INTERFASES DE ADMINISTRACIÓN**

---

Introducción a la Línea de Comando de Interfase de JBoss

- API nativa de administración (CLI)
- API HTTP de administración (JON, Web, REST)
- Interfases topológicas de administración.

# API NATIVA DE ADMINISTRACIÓN

---

Punto de entrada para los clientes de administración que se basan en el protocolo nativo de EAP para integrarse con la capa de administración.

- Usan un protocolo binario abierto, basado en JBoss Remoting.
- Usan un API estilo RPC, para describir y ejecutar operaciones contra el dominio administrado, o un servidor standalone.
- API de administración "detyped".

Usado por la Command Line Interfaces (CLI) para administrar operaciones.

Tiene capacidades de integración para un amplio rango de otros clientes.

Ejemplo: Un cliente personalizado Java-based.

RED HAT CONSULTING

# API NATIVA DE ADMINISTRACIÓN

---

Configuración en standalone.xml o un controlador de host (host.xml).

Ejemplo: (host.xml con puerto por defecto)

```
<management-interfaces>
  <native-interface security-realm="ManagementRealm">
    <socket interface="management" port="9999" />
    ...
  </native-interface>
</management-interfaces>
```

RED HAT CONSULTING

# API HTTP DE ADMINISTRACIÓN

---

Punto de entrada para los clientes de administración que se basan en el protocolo HTTP de EAP para integrarse con la capa de administración.

- Protocolo codificado JSON.
- API de estilo RPC "detyped" para describir y ejecutar operaciones contra el dominio administrado o servidor standalone.

Capacidades de integración para un amplio rango de clientes

- Consola Web (GWT).
- Plugin EAP 6 JON.
- e.g. Web / Rich Client.

# API HTTP DE ADMINISTRACIÓN

---

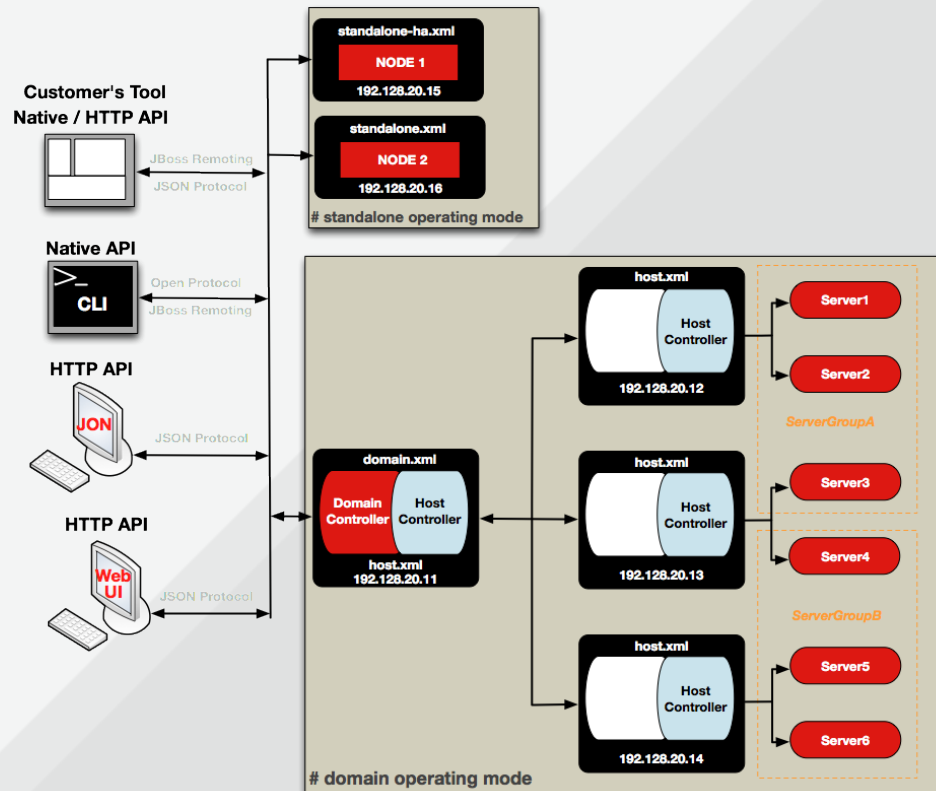
Configuración en standalone.xml o un controlador de host (host.xml).

Ejemplo: (host.xml con puerto por defecto)

```
<management-interfaces>
  <http-interface security-realm="ManagementRealm">
    <socket interface="management" port="9990"/>
    ...
  </http-interface>
</management-interfaces>
```

	Acceso local a la consola de administración del dominio
	Acceso remoto a la consola de administración del dominio
http://hostname:9990/management	Expone el API HTTP de administración del dominio (REST/JSON)

# TOPOLOGÍA DE INTERFASES DE ADMINISTRACIÓN





# REPRESENTACIÓN DEL MODELO DINÁMICO -DMR

---

Sintaxis "detyped": basada en JSON, mas potente y compacto que XML.

```
<subsystem xmlns="urn:jboss:domain:logging:1.1">
<console-handler name="CONSOLE">    <level name="INFO"/>    <formatter>    <pattern-formatter
pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n"/>    </formatter>    </console-handler>
... remainder ...
```

```
"logging" => {
"console-handler" => {
"CONSOLE" => { "autoflush" => true,
"encoding" => undefined,
"filter" => undefined,
"formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
"level" => "INFO",
"target" => "System.out"
}},
... remainder ...
```

# ADMINISTRACIÓN JBOSS EAP 6: JBOSS-CLI

---

- Cliente interactivo
- Conecta a la instancia standalone o al controlador de dominio.
- Sintaxis basada en \*nix shell, recursos como si fueran archivos.
- Comando contextual y recurso tab-completion.
- Operaciones de composición de alto nivel.
- Cambios persistentes.

```
[standalone@localhost:9999 /] batch
[standalone@localhost:9999 / #] jms-queue add --queue-address=tradeQueue --entries=queue/tradeQueue
#1 jms-queue add --queue-address=tradeQueue --entries=queue/tradeQueue
[standalone@localhost:9999 / #] deploy ~/jboss-as-login.war
#2 deploy ~/jboss-as-login.war
[standalone@localhost:9999 / #] undeploy jboss-as-login.war
#3 undeploy jboss-as-login.war
[standalone@localhost:9999 / #] jms-queue remove --queue-address=tradeQueue
#4 jms-queue remove --queue-address=tradeQueue
[standalone@localhost:9999 / #] run-batch
The batch executed successfully.
[standalone@localhost:9999 /] █
```

# HERRAMIENTA CLI

---

Se ejecuta con `./bin/jboss-cli.sh`

Parámetros:

- `--connect` ejecuta el proceso CLI y lo conecta al controlador.
- `--controller=<ip>:<port>`.
- `--gui` ejecuta un GUI CLI.
- `--file=<cli.script.file>` ejecuta un script en modo no interactivo.
- `--help`.
- `--command=<command1,command2>` ejecuta comandos en modo NI.

Comandos

- `quit`
- `[node_path]:operation_name (parameter_list) {header_list}`

# HERRAMIENTA CLI

---

## Características

- tab-completion (autocompletación por tab).
- Comando "cd" para navegar a través del árbol de recursos.
- Comando historial (.jboss-cli-history).
- Modo de lotes (atomic script)
  - batch ... run-batch
  - list-batch ... edit-batch-line ... remove-batch-line ...  
... holdback-line ... discard-batch

## Ejemplos

```
/profile=default/subsystem=logging/console-handler=CONSOLE:read-attribute(name=level)  
:read-resource(recursive=true,include-runtime=true)
```

## HERRAMIENTA CLI - VENTAJAS

---

- Permite la generación de scripts para automatizar tareas de administración
- permite modificar de configuración de JBoss EAP sin editar el XML directamente, además asegura que los cambios se hagan correctamente
- Si una configuración falla o es incorrecta, el API realiza rollback inmediatamente
- Permite la generación de snapshot de las configuración general, permite puntos de restauración.

RED HAT CONSULTING

# PARA HACER LA VIDA MAS FÁCIL

---

Una colección de scripts de JBOSS cli de ejemplo

**01-DS:** Agregar Driver, Agregar/Remover Datasources.

**02-ConnectionFactory:** Agregar/Remover CF.

**03-JMS:** Agregar/Remover Queue y Topic

**04-Bean-Pool:** Creacion de Beanpool

**05-Logging:** Agregar Appender(Size, Peridic), Loggers y registro en Root Logger

**06-Deployments:** Deploy/Undeploy Aplicativo

**07-DeploymentOverlay:** creacion de Overlay

**08-Modules:** instalacion de modulo jboss

**09-Server-groups:** Agregar/Remover ServerGroup, Configuración JVM

**10-Admin:** Start/Shutdown/Restart ServerGroup y Servers

**11-interfaces:** configuracion de Binding de interfaces

**12-Patch:** parchado de hosts

**13-Vault:** Creacion de vault para codificar atributos

**14-Server:** Agregar/Remover Servidores



redhat®

THANK YOU!



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/redhat](https://www.youtube.com/redhat)



[twitter.com/RedHatNews](https://twitter.com/RedHatNews)



[plus.google.com/+RedHat](https://plus.google.com/+RedHat)