Deploy and undeploy with the CLI

Versión 5

Creado por Alexey Loubyansky el 02-may-2011 4:02. Modificado por última vez el 02-may-2011 11:49 por Alexey Loubyansky.

Deployment and undeployment in the CLI is done with the corresponding commands: deploy and undeploy. Each of these operations may consist of a few steps.
E.g., first step to deploy an application is to upload its content to the server's deployment repository and then, second, is to enable (or actually deploy) it.
To undeploy an enabled (or already deployed and running) application, first, it has to be disabled (i.e. stoped) and then, second, if necessary, its content can be removed from the server's deployment repository. If the application is disabled but not removed from the server's deployment repository, it can be deployed later but just enabling it.

The scenarios below demonstrate how deploy and undeploy commands and its arguments handle these tasks.

Standalone mode

First, we need to make sure there is a standalone AS7 instance is running and the CLI is connected to its controller, e.g.

```
{code}Connected to standalone controller at localhost:9999
[standalone@localhost:9999 /]{code}
```

Now to deploy an application, all that's necessary is to type in deploy and the path to the package (the tab-completion will help to navigate the filesystem), e.g.

```
{code}[standalone@localhost:9999 /] deploy ../../../../testsuite/smoke/target/deployments/
'test-deployment.sar' deployed successfully.{code}
```

You can make sure that the application is deployed by checking deployments in the standalone.xml

```
{code}<deployments>
    <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="af4edd
</deployments>{code}
```

To undeploy this application its name has to be passed in as the argument to undeploy (if you type in undeploy, the tab-completion will list all the deployed applications and help complete the deployment name), e.g.

```
{code}[standalone@localhost:9999 /] undeploy test-deployment.sar
Successfully undeployed test-deployment.sar.{code}
```

Now if you check the standalone.xml you'll see that the deployment is gone from the list.

Based on this simple example you can see that the deploy command actually automatically performs two steps: uploads the content and enables it. And the undeploy command, by default, disables the application and removes its content from the repository.

If necessary though, you can just upload the content to the repository w/o enabling it. The deploy command has --disabled switch for that, e.g.

```
{code}[standalone@localhost:9999 /] deploy ../../../../testsuite/smoke/target/deployments/
'test-deployment.sar' deployed successfully.{code}
```

Which will result in the following in the standalone.xml

```
{code}<deployments>
    <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="18a29b
</deployments>{code}
```

Then to enable the deployment you'll need to specify just its name as the argument (if you press the tab key after '--name=' it'll help you autocomplete the value)

```
{code}[standalone@localhost:9999 /] deploy --name=test-deployment.sar
'test-deployment.sar' deployed successfully.{code}
```

```
{code}<deployments>
    <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="18a29b
</deployments>{code}
```

Now, when undeploying, If you don't want to remove the content but just disable (stop) the running application then you can use --keep-content switch, e.g.

```
{code}[standalone@localhost:9999 /] undeploy test-deployment.sar --keep-content
Successfully undeployed test-deployment.sar.{code}
```

If you press the tab key after the deployment name, it'll autocomplete the --keep-content argument. And if you now look into the standalone.xml, you'll see

```
{code}<deployments>
   <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="af4eddd
```

```
</deployments>{code}
```

If you want to re-deploy an already deployed application (whether enabled or disabled) and replace its content with the new version, the deploy command will fail unless you specify --force argument, e.g.

```
{code}[standalone@localhost:9999 /] deploy ../../../../testsuite/smoke/target/deployments/
'test-deployment.sar' is already deployed (use --force to force re-deploy).
[standalone@localhost:9999 /] deploy ../../../../testsuite/smoke/target/deployments/test-d
'test-deployment.sar' re-deployed successfully.{code}
```

Removal of a disabled deployment is not different from an enabled deployment, e.g.

```
{code}[standalone@localhost:9999 /] undeploy test-deployment.sar
Successfully undeployed test-deployment.sar.{code}
```

TODO: the command result messages could be more specific...

If you check the standalone.xml you'll see that the deployment was removed from the server's deployment repository.

Domain mode

Make sure the domain is up and the CLI is connected to its controller, e.g.

```
{code}
Connected to domain controller at localhost:9999
[domain@localhost:9999 /]{code}
```

The main difference in deploying and undeploying in the domain mode is that you have to specify the server groups to which these operations should apply. In case of deploy, you can choose between two arguments to specify the server groups:
--all-server-groups - to apply to all the available server groups;
--server-groups - to apply the operation to the spicific list of server groups separated by commas.

E.g.

```
{code}
[domain@localhost:9999 /] deploy ../../../../testsuite/smoke/target/deployments/test-deplo
'test-deployment.sar' deployed successfully.{code}
```

If you now looking into the domain.xml, you'll see something like this

```
{code}
    <deployments>
        <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="af
    </deployments>
    <server-groups>
        <server-group name="main-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
            <deployments>
                <deployment name="test-deployment.sar" runtime-name="test-deployment.sar"
            </deployments>
        </server-group>
        <server-group name="other-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
            <deployments>
                <deployment name="test-deployment.sar" runtime-name="test-deployment.sar"
            </deployments>
        </server-group>
    </server-groups>{code}
```

The corresponding undeploy command (i.e. the one that would undo the just performed deploy command) is

```
{code}
[domain@localhost:9999 /] undeploy test-deployment.sar --all-relevant-server-groups
Successfully undeployed test-deployment.sar.{code}
```

Notice that undeploy uses --all-relevant-server-groups instead of --all-server-groups. --all-relevant-server-groups means to undeploy the application from all the server groups in which the deployment is enabled (which might be a subset of all the available server groups).
And if you check the domain.xml now, you'll see that the deployment was removed from the domain deployment repository.

The following command is an equivalent to the deploy command using --all-server-groups above (assuming main-server-group and other-server-group are all the available server groups)

```
{code}
[domain@localhost:9999 /] deploy ../../../../testsuite/smoke/target/deployments/test-deplo
'test-deployment.sar' deployed successfully.{code}
```

BTW, the tab-completion will help completing the value for --server-groups. You can check and make sure that the domain.xml looks the same as above after the deploy.

Now, suppose, we want to undeploy the application from only one server group, e.g.

```
{code}
[domain@localhost:9999 /] undeploy test-deployment.sar --server-groups=other-server-group
Successfully undeployed test-deployment.sar.{code}
```

Note, since the application is still supposed to be enabled in the main-server-group --keep-content must be specified, otherwise the command will fail. You can now check the domain.xml.

```
{code}
    <deployments>
        <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="af
    </deployments>
    <server-groups>
        <server-group name="main-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
            <deployments>
                <deployment name="test-deployment.sar" runtime-name="test-deployment.sar"
            </deployments>
        </server-group>
        <server-group name="other-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
        </server-group>
    </server-groups>{code}
```

As you can see, the deployment is not just disabled but completely removed from the other-server-group. At the moment it is handled this way to keep the implementation of deploy/undeploy simpler.

To deploy the application to the other-server-group again you need to pass its name as the value of the --name argument to the deploy command, e.g.

```
{code}
[domain@localhost:9999 /] deploy --name=test-deployment.sar --server-groups=other-server-g
'test-deployment.sar' deployed successfully.{code}
```

You can check the domain.xml to make sure that the deployment is back in the other-server-group.

Removing content from the domain deployment repository which is not referenced from any of the available server groups doesn't require specification of server groups. E.g.

```
{code}
[domain@localhost:9999 /] undeploy test-deployment.sar --all-relevant-server-groups --keep
Successfully undeployed test-deployment.sar.{code}
```

```
{code}
    <deployments>
        <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="af
    </deployments>
    <server-groups>
        <server-group name="main-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
        </server-group>
        <server-group name="other-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
        </server-group>
    </server-groups>{code}
```

```
{code}
[domain@localhost:9999 /] undeploy test-deployment.sar
Successfully undeployed test-deployment.sar.{code}
```

Note, this is possible only the deployment isn't referenced from any of the available server groups, otherwise the command will fail. If you want to remove the deployment from the deployment repository in this case you'll have to specify either --all-relevant-server-groups or list them in --server-groups.
If you now check the domain.xml you'll see that the deployment has been removed from the domain deployment repository.

It is also possible to just upload the content to the domain's deployment repostiory w/o enabling it in any of the available server groups. E.g.

```
{code}[domain@localhost:9999 /] deploy ../../../../testsuite/smoke/target/deployments/test
'test-deployment.sar' deployed successfully.{code}
```

After that, the domain.xml will look like this

```
{code}<deployments>
        <deployment name="test-deployment.sar" runtime-name="test-deployment.sar" sha1="40
    </deployments>
    <server-groups>
        <server-group name="main-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
        </server-group>
        <server-group name="other-server-group" profile="default">
            <jvm name="default">
                <heap size="64m" max-size="512m"/>
            </jvm>
            <socket-binding-group ref="standard-sockets"/>
        </server-group>
    </server-groups>{code}
```

85646 Vistas Categorías: Etiquetas: cli, deploy, undeploy, as7

4 Comentarios

- Red Hat
- Site Help:
- FAQ
- Report a problem