Integrated Dell Remote Access Controller Configuration Cloning Script

This Dell Technical White Paper provides information about iDRAC configuration cloning scripts.

Author(s)

Henry Gbedemah

Sharad Naik

May 2012| Rev 1.0

Contents

Figures

Executive summary

This white paper describes how to use configuration cloning scripts to easily configure a large number of iDRACs.

Introduction

The Integrated Dell Remote Access Controller (iDRAC) is a dedicated service processor that enables out-of-band remote management services on Dell Enterprise class servers. It is an embedded platform running Linux and Busybox.

The iDRAC is capable of remotely managing its host system—even when the host operating system crashes, it performs tasks like re-starting and updating software on the host system. Fully configuring an iDRAC to harness its complete management capabilities consists of setting approximately 500 attributes. This activity can become very tedious and time consuming in a large data center, which may consist of several servers hosting iDRACs.

iDRAC cloning scripts are provided to help administrators who routinely have to configure a large number of iDRACs to do so more easily. Using these scripts requires Python 2.7 and wsmancli to be installed on the client machine. The client machine should have network connectivity to both the source iDRAC and the target iDRACs.

Environmental Check Point

Before you begin, it is recommended to prepare the following items:

1. Verify you have both a source and target iDRAC that are accessible through the network and reachable by WSMAN.
2. Check to see if the openwsman command line tool is installed (Linux machine required) and Python 2.7 is installed. For more information, seethe Python release site .
3. Make sure Python is installed on the system (minimum version 2.7).
4. Download the scripts and XML files from Dell Tech Center. The scripts relevant to this document are:
    - [pullattr.py]
    - [pushattr.py]
    - [pulliDRAC.sh]
    - [pushiDRAC.sh]
    - [IDRAC0.01.xml]

Figure 1: Pictorial View of the Environment

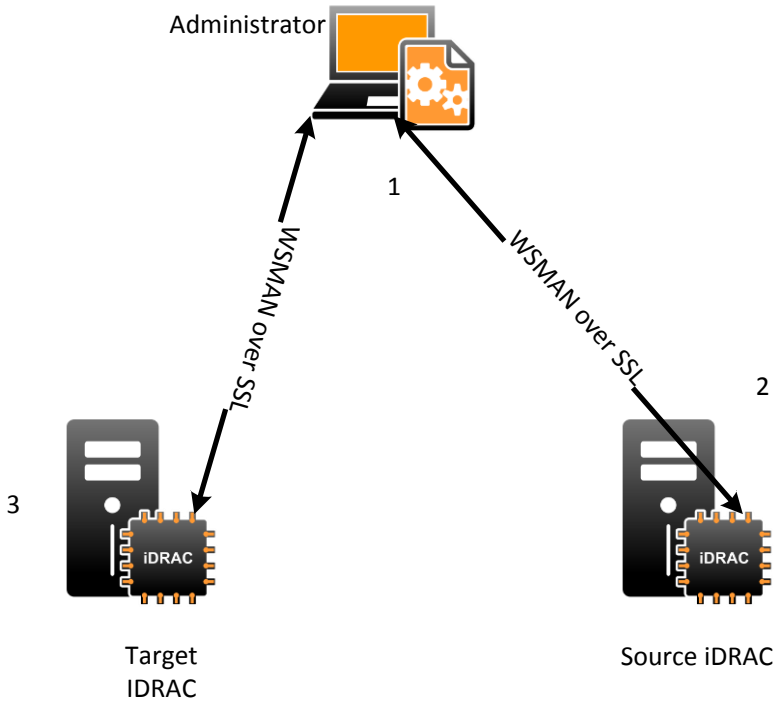

Figure 1 shows a pictorial view of the environment. It starts with the Administrator running scripts to send WS-MAN commands through an SSL connection to a source system, which has a baseline configuration to collect iDRAC configuration attributes. The administrator then runs more scripts to use the collected configuration to configure the target iDRAC.

To clone the baseline iDRAC:

1. Copy the attributes and values from a baseline iDRAC.
2. Stage the attributes, making required changes.
3. Push the attributes to the target iDRAC.

Copying Attributes

The first step in cloning iDRAC attributes is to copy a known configuration from a pre-configured iDRAC. This step is achieved by running *pulliDRAC.sh*, with iDRAC as the argument. For example:

```
[pulliDRAC.sh idrac]
```

```
IP address (Enter for default: )  ? [iDRAC IP]

User name (Enter for default: )  ? [USER NAME]

User password (Enter for default: )  ? [PASSWORD]
```

This script pulls all iDRAC attribute values into a file named *xxx.xxx.xxx.xxx*_idrac.ini where *xxx.xxx.xxx.xxx* is the IP address of the source iDRAC. The contents of the .ini file are in the format Attributename=Attributevalue_pair. The script invokes a WS-MAN enumerate command for all iDRAC attributes, which in in turn formats the attributes into a list of AttributeName=AttributeValue pairs in the resulting .ini file. The list is ordered according to the display order of the attributes represented in the attribute registry.

For example:

```
Building the Order Attributes Template File...

Getting the Attributes …

Creating the .ini file …

File Created: 192.168.0.120_idrac.ini

Adding Attributes to the file…
```

Staging Attributes

The next step is to scrub the attribute values that you want to change—for example, user names and passwords. Even though user names are populated with the programmed user names from the source iDRAC, these may not be the users you require on your target iDRAC. For that reason, user names must be scrubbed. This also applies to IP addresses and other attributes that make a system unique. Passwords are not copied; instead, a place holder of ****** is put in the .ini file to maintain the integrity of all passwords. So, at a minimum, the root password should be changed before applying the changes to the target iDRAC.

Pushing Attributes

The final step is pushing the iDRAC configuration to a target iDRAC. This is achieved by invoking the pushIDRAC.sh shell script, which prompts for required input . The attributes are pushed to the target iDRAC by using the ApplyAttributes method of the iDRAC Card service class. This method creates a job in the background to apply all iDRAC attribute values sent.

For example:

```
[pushiDRAC.sh]
```

```
IP address (Enter for default: )  ? [iDRAC IP]

User name (Enter for default: )  ? [USER NAME]

User password (Enter for default: )  ? [PASSWORD]

Config attributes(.ini) file, to be pushed:

 (ENTER for default: ) ? 192.168.0.120_idrac.ini
```

Sample output:

```
NOW setting attributes…

Config job id:JID_466921314998 for:iDRAC.Embedded.1 has been created.

iDRAC.Embedded.1

        RAC006:AttributeValue cannot be changed for ReadOnly Attribute Name
Info.1#Product

   RAC017:Job created to apply the attribute value

   RAC002:General failure IPv6.1#Address1

   RAC024:VLAN is Disabled so cannot set VLAN Priority NIC.1#VlanPriority

Checking job status:

Job:JID_466921314998 completed

All job execution complete
```

Conclusion

When the AttributeName AttributeValue pairs are pushed out to the target iDRAC, the iDRAC stages the attributes to be set, and reports whether particular attributes are included in the job to be created.

Reasons why a name value pair may not be included in the job are:

- RAC006: AttributeValue cannot be changed for the read-only attribute <AttributeName>
- RAC022: The user name is not configured, so the user cannot be enabled and the password cannot be set

After the job runs, there are two possible outcomes:

- All the attributes that were ok in the staged set above are set correctly

- Some of the attributes fail

The job results are reported as “All job execution completed” or “Update task successfully completed for one or more attributes, but with errors!”