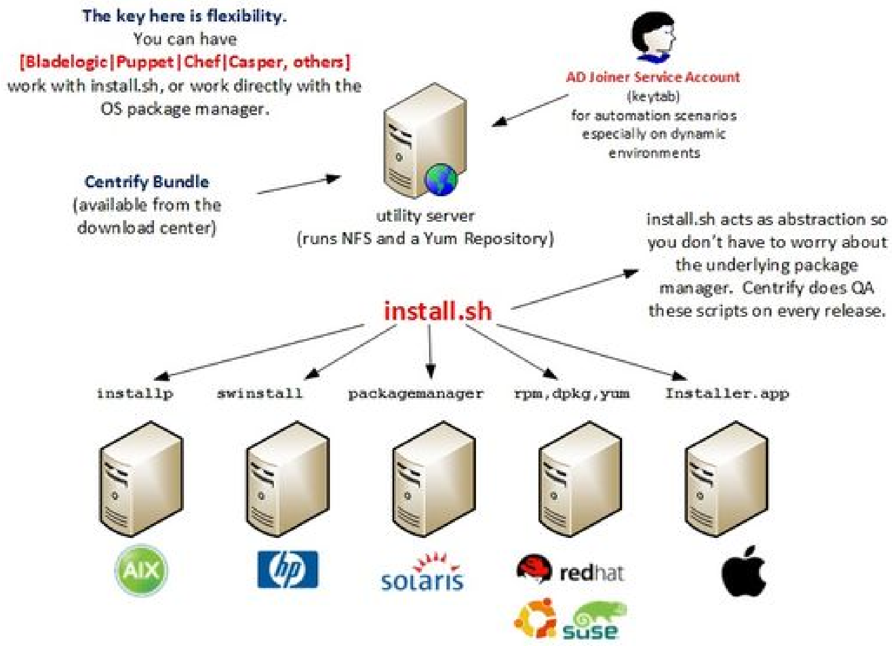
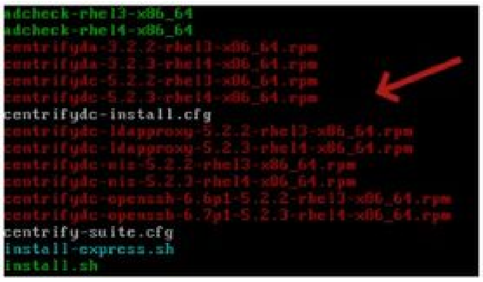
[HOWTO] Orchestration Basics - Setting up your own Centrify repository for RHEL and derivatives

The foundation to many deployment or orchestration tools is to have private hosted repositories of source installation packages. Centrify offers native packages for all the supported platforms. In addition, Centrify also offers install.sh; this script and the answer files can abstract the underlying package manager for UNIX, Linux or Mac systems.


The key here is flexibility.
You can have
[Bladelogic|Puppet|Chef|Casper, others]
work with install.sh, or work directly with the OS package manager.
AD Joiner Service Account
(keytab)
for automation scenarios especially on dynamic environments
Centrify Bundle
(available from the download center)
utility server
(runs NFS and a Yum Repository)
install.sh acts as abstraction so you don't have to worry about the underlying package manager. Centrify does QA these scripts on every release.
install.sh
installp
swinstall
packagemanager
rpm,dpkg,yum
Installer.app
AIX
hp
solaris
redhat
suse

Below are the context of the tarball for x64 RHEL systems. Note that all Centrify Packages are delivered as RPMs. There are helper files like centrify-suite.cfg. In additon, remember, you can safely install Centrify and bake it in your image, it won't be active until you join the Domain. Important: You can't just press install and forget. Make sure the platform is supported, otherwise you'll have to use the package manager directly to bypass checks (or the --no_os_check) option.



```
adcheck-rhel3-x86_64
adcheck-rhel4-x86_64
centrifyda-3.2.2-rhel3-x86_64.rpm
centrifyda-3.2.3-rhel4-x86_64.rpm
centrifydc-5.2.2-rhel3-x86_64.rpm
centrifydc-5.2.3-rhel4-x86_64.rpm
centrifydc-install.cfg
centrifydc-ldapproxy-5.2.2-rhel3-x86_64.rpm
centrifydc-ldapproxy-5.2.3-rhel4-x86_64.rpm
centrifydc-nis-5.2.2-rhel3-x86_64.rpm
centrifydc-nis-5.2.3-rhel4-x86_64.rpm
centrifydc-openssh-6.6p1-5.2.2-rhel3-x86_64.rpm
centrifydc-openssh-6.7p1-5.2.3-rhel4-x86_64.rpm
centrify-suite.cfg
install-express.sh
install.sh
```

This means that you can use an NFS Server, a Samba Server, a Web Server or your package manager in conjunction with the Centrify bits to deploy software easily across your enterprise. Alternatively, the Yellow-Dog Updater Modified (YUM) (and APT) provide a simple to set-up and robust package manager that can be used primarily with RedHat and derivative platforms.

Centrify Repo

For existing customers with Support Portal access, Centrify offers Yum, APT and SuSe repos. For instructions on how to set up, look here: https://www.centrify.com/support/customer-support-portal/repos

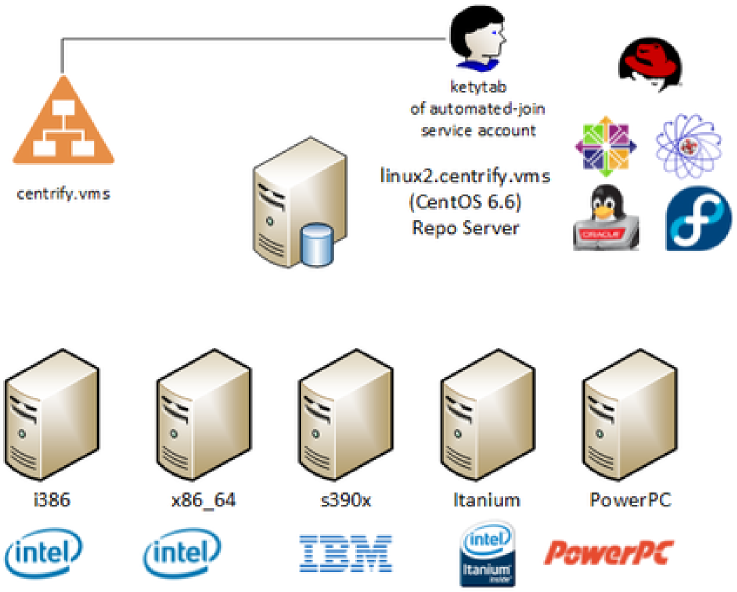
Disclaimers

- This article provides a "quick basic configuration"; in a true deployment you have to account for high-availability, replication, security, package integrity, supported platforms, supported versions, change control, etc.
- All names, logos and trademarks used in this articles correspond to their existing owners.

What is required?

- A RHEL-based system with enough storage for the Centrify RPM packages for each platform (or just for the subset you need to support).
- If using web as the delivery mechanism, the web server (Apache) has to be set and configured accordingly.

Example Diagram



In my mock organization, there are different types of RHEL derivatives, including RedHatEnterprise, Fedora, Oracle, Scientific Linux, etc, all running on different architectures, including Intel/AMD (32 and 64 bit) , zLinux IBM s390, Itanium and IBM Power processors.

Implementation Steps

Verify pre-requisites. I'm planning to use http as the transport for my repo.

```
# Check if Apache is installed
$ sudo  yum list installed | grep httpd
httpd.x86_64            2.2.15-47.el6.centos
httpd-tools.x86_64      2.2.15-47.el6.centos

# If not present
$ sudo yum install httpd
$ sudo chckconfig httpd on
$ sudo service httpd start
```

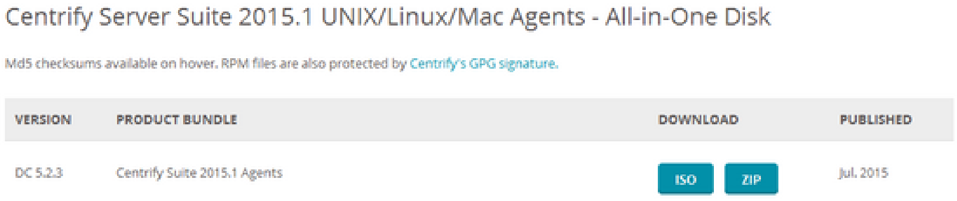
Install the createrepo package

```
$ sudo yum install createrepo
```

Download the Centrify Bundle for the platforms to be supported.

Example data - In my mock organization, I will be supporting the bits for 2014.1 and 2015.1. This organization has a policy to only deploy maintenance releases.

1. Go to the Centrify Customer Center and Navigate to the Downloads Section.
2. Identity the bundle for 2015.1 and 2014.1:



Centrify Server Suite 2015.1 UNIX/Linux/Mac Agents - All-in-One Disk

Md5 checksums available on hover. RPM files are also protected by Centrify's GPG signature.

VERSION	PRODUCT BUNDLE	DOWNLOAD	PUBLISHED
DC 5.2.3	Centrify Suite 2015.1 Agents	ISO ZIP	Jul. 2015

The bundle is available in ISO or ZIP formats.

3. Copy and unzip the bundle in a staging server. Note that ALL the platforms are present. We are only interested in the .TGZ files that contain the word "rhel" in the name.

```
centrify-suite-2015.1-rhel4-i386.tgz      <= this is for x86 (Intel or AMD)
centrify-suite-2015.1-rhel4-ia64.tgz      <= this is for Itanium
centrify-suite-2015.1-rhel4-ppc.tgz       <= this is for Power
centrify-suite-2015.1-rhel4-x86_64.tgz    <= this is for x64 (Intel or AMD)
centrify-suite-2014.1-rhel5-s390x.tgz     <= this is for zLinux
```

4. Gunzip and untar all those files (use tar xzvf [package].tgz). You will see RPMs for all the software included. Here's a description of all packages:
   CentrifyDA is the audit agent, only used with Enterprise Edition or Privilege Service.
   CentrifyDC is adclient; can be used in Express mode or Zone mode (licensed)
   CentrifyDC is absolutely the only package required in most scenarios. Any other package, and you're in the land of "you must know what you're doing"
   CentrifyDC-ldapproxy is the Centrified OpenLDAP Server - required to support filers or older apps over LDAP interfaces
   CentrifyDC-nis is another proxy. It exists to support legacy NIS scenarios. Needless to say, this is a transitional package, using NIS is a bad idea and will lead to audit comments.
   CentrifyDC-openssh is our provided version of OpenSSH; since SSH has matured to provide PAM, GSSAPI and even Kerberos support, 80% of the time is not needed, however, if you have an older UNIX or are in a complex AD, this package can be your friend.

Example data - If we follow the example, there should be 5 RPM packages for each of the different architecture. However, since there's variability on the s390 package releases, I ended up with 38 packages.

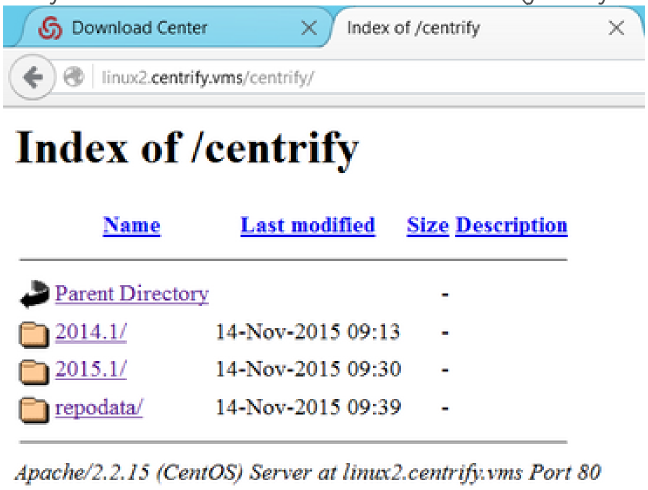
Tip: use the `rpm -qpil` command to inspect them. You'll see that they are very well docummented.

Copy the Centrify RPMs to the target location and verify access

In my over-simplistic example, we'll piggy-back on the default website.

1. Create a folder under /var/www/html
   ```
   $ sudo mkdir /var/www/html/centrify
   ```
2. Copy the RPMs to the folder.
   ```
   $ cd /path/to/centrify/staging
   $ sudo mv * /var/www/html/centrify
   ```
3. Set the proper permissions in the folder
   ```
   chmod -R ugo+rX /var/www/html/centrify
   ```

4. Verify that the files are accessible via the web server (you may have to check the firewall settings)



Index of /centrify

Name	Last modified	Size	Description
Parent Directory		-	
2014.1/	14-Nov-2015 09:13	-	
2015.1/	14-Nov-2015 09:30	-	
repodata/	14-Nov-2015 09:39	-	

Apache/2.2.15 (CentOS) Server at linux2.centrify.vms Port 80

Create your Repository

1. Use the createrepo command to create the repository

```
$ sudo createrepo --database /var/www/html/centrify
```

2. Update the repository database

```
$ sudo createrepo --update /var/www/html/centrify
Spawning worker 0 with 38 pkgs
Workers Finished
Gathering worker results
```

Create your repository's configuration file

In this simple configuration, we ended-up with this file (centrify.repo):

```
[centrify]
name=centrify
baseurl=http://linux2.centrify.vms/centrify
enabled=1
gpgcheck=0
```

Note: We are adding a keytab from a *least privilege* AD user that can only perform the join (or leave). This will ensure that we don't need to put any passwords, keys or hashes in our provisioning script. For more information on how to create the AD account and corresponding keytab, see this article.

Verify that your Repository is working

1. Log in to a test system, copy or create the centrify.repo file in the /etc/yum.repos.d folder

```
$ sudo cp /path/to/centrify.repo /etc/yum.repos.d/centrify.repo
```

2. Checking package availability and Metadata

```
$ sudo yum install centrifydc
No package centrifydc available.
  * Maybe you meant: CentrifyDC
# This verifies that our metadata is OK.
```

3. More information about CentrifyDC (the base package) - output truncated.

```
$ sudo yum info CentrifyDC
Available Packages
Name        : CentrifyDC
Arch        : i386
Version     : 5.2.3
Release     : 429
Size        : 25 M
Repo        : centrify
Summary     : Centrify DirectControl Agent
```

```
URL         : http://www.centrify.com/
License     : Copyright (C) 2004-2015 Centrify Corporation
Description : RPM to install Centrify DirectControl on Linux x86 platforms.

Name        : CentrifyDC
Arch        : x86_64
Version     : 5.2.3
Release     : 429
Size        : 34 M
Repo        : centrify
Summary     : Centrify DirectControl Agent
URL         : http://www.centrify.com/
License     : Copyright (C) 2004-2015 Centrify Corporation
Description : RPM to install Centrify DirectControl on Linux x86_64 platforms.
```

Note that I can see that the lastest version is available for two platforms that may apply to my system.

4. Let's verify the integrity of dependencies (LDAP Proxy depends on DirectControl)

```
$ repoquery --requires CentrifyDC-ldapproxy
/bin/sh
CentrifyDC <= 5.2.3-999
CentrifyDC >= 5.2.3-000
/bin/sh
CentrifyDC <= 5.2.3-999
CentrifyDC >= 5.2.3-000
```

5. Let's install DirectControl

```
$ sudo yum install CentrifyDC

Resolving Dependencies
--> Running transaction check
---> Package CentrifyDC.x86_64 0:5.2.3-429 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package             Arch            Version            Repository         Size
================================================================================
Installing:
 CentrifyDC          x86_64          5.2.3-429          centrify           34 M

Transaction Summary
================================================================================
Install       1 Package(s)

Total download size: 34 M
Installed size: 87 M
Is this ok [y/N]:
```

Putting it All Together

Ultimately, there are 3 operations to onboard a Centrify system in AD.

- Get the package and install it [we should be good here]
- Verify that the system is ready to join (OS check, Perl, inspect DNS, check for communicaton with AD)
  [read below note #1]
- Use the adjoin command to activate DirectControl and join AD.
  [read below note # 2]

Note # 1: With YUM you have a reliable way to get the packages across for multiple RHEL derivatives; however, your logic should include adcheck in the mix if you're not using an enterprise image. Ideally you would always do QA and use supported platforms and have a standard DNS configuration that includes the ability to get an authoritative response from a Domain controller. If that's the case, and there are no firewalls in-between, adjoin should just work.

Note # 2: We have covered adjoin extensively; keep in mind that it is Kerberized and it won't require a password to work. Its counterpart (adleave) will work the same way, and it's essential for cleanup and releasing of licenses (otherwise they will count for 45 days against your usage).

The ultimate automation script should contain just 3 lines:

1. yum install CentrifyDC
2. kinit to authenticate to authorized AD user
3. adjoin

In my example:

```
$ yum install CentrifyDC
$ env KRB5_CONFIG=/temp/krb5.conf /usr/share/centrifydc/kerberos/bin/kinit
-kt /temp/ad-joiner.keytab ad-joiner
$ adjoin --zone Global --container "ou=servers,ou=centrifyse"
--computerrole "PCI Systems" centrify.vms
```

Verification Video (5 minutes, 10 seconds)

Appendix: Flipping the script - deprovisioning

In elastic environments, decommisioning a system (or 'Terminating' in AWS lingo) has a Centrify implication; it has to do with proper Active Directory hygiene and licensing purposes(*). The proper way to leave the domain is to use the remove option of the adleave command. Based on my example, If I wanted to leave the domain and uninstall Centrify here's the sequence:

1. kinit to an account that is authorized to remove the computer from the domain
2. use the adleave -r command
3. optional: use yum erase CentrifyDC

In my example:

```
$ env KRB5_CONFIG=/temp/krb5.conf /usr/share/centrifydc/kerberos/bin/kinit
-kt /temp/ad-joiner.keytab ad-joiner
$ adleave --remove
$ yum erase CentrifyDC
```

(*) If you don't use the "--remove" option with adleave, you are creating an orphaned object in the zone and a computer object that is disabled. It takes 45 days for the Centrify consoles to consider this system as **inactive**; inactive systems don't count against your Centrify license counts. You can run the Analyze tool to find and clean orphaned and tombstoned objects.