CumulusRDF Quickstart

Part I: VM-Images for Demonstration and Performance Tests

Virtual Machine Types

A CumulusRDF Test & Demo environment is required. It will be created for Virtual Box, VMWare, and Amazon EC2 server instances. The embedded Cassandra and Tomcat servers are used for demos and smaller tests, all deployed on one single host, running CentOS. An extended version of the CumulusRDF server will connect to an existing Apache Cassandra cluster, or later as soon as an HBase connector is available, to a Hadoop Cluster, running HBase on multiple nodes. Such a distributed environment allows real world use-cases and performance tests to prepare production deployments.

Name	Scope	Status
User / Demo Image *UDI*	Run all demos and tests in Pseudo Distributed mode.	**ready**
Dev Image *DEVI*	Use relevant development tools in an Pseudo Distributed Environment.	**in progress**
Extended User / Demo Image *EUDI / EDEVI*	Show real life Use-Cases and do Performance Tests / Scalability Analysis in a real cluster setup.	
Production Image	Deploy a tested version for production.	Out of scope

Tab 1: Different image types are provided for demonstration, developers and real world applications. A production image is out of the scope of this document.

Overview

The VM is created with "**packer**"and needs a "**kickstart script**" (CentOS version). Suse Studio can be used as an alternative beside the local build procedure. This allows a profile on the public server without a local build environment, which is important, especially if download bandwidth is limited.

The CumulusRDF server image is created in two steps: (I) Host creation and (II) project bootstrap. Hardware setup, disc layout and basic OS setup (including network configuration) is done first. The cumulus-rdf svn repository is checked out during the first start. A **bootstrap script** is executed to finalize the build procedure.

Maven is used to build the software packages and to start the local servers for a demo-setup, which means the compilation and the deployment of the latest version (or a manually chosen one) is done before Apache Tomcat and Apache Cassandra servers are

started locally. The extended version will check accessabillity of the configured storage layer and also indicate the state.

Standard Build Procedure via Packer

Here you find, how packer is installed. We need an image-descriptor file for each image we want to build.

Image Name	Disc Layout	Packages
UDI	clearpart --all --drives=sda ignoredisk --only-use=sda part /boot --fstype=ext2 --asprimary --size=500 part / --fstype=ext4 --asprimary --size=20480 part /home --fstype=ext4 --asprimary --size=10480 part swap --asprimary --size=2048 bootloader --location=mbr --driveorder=sda --append="nomodeset rhgb quiet"	%packages @base @core
DEVI		-
EUDI	-	-
EDEVI	-	-

Tab 2: Different image types use different disc layouts and specific packages, according to our virtual machine image requirements.

Packer needs an ISO file, e.g. the net-install CD and a checksum for verification. The ISO image is locally cached and reused, but the URL based network installer has no cache, so we provide the installation image for network installation locally on an NFS server and we use a two step image creation procedure. Without an NFS server we have to download all the CentOS packages directly.

We have to do the following two steps:

- Create an image via Net-Installer (online download, or from local NFS server) initiated via ISO image (local copy).
- Create an image from first image with more customization.

Packer requires a recent version of VMWare Player, VMWare Workstation, or VMWare Fusion (Version 4.x is not working) So we start with VirtualBox and EC2 images, before we build VMWare images later as well.

List of required files and tools

Two vm-description files for packer are shown here. `CentOS-UDI.json` is used for step one and `CentOS-UDI-FINAL.json` creates the final version of a User and Demo Image (UDI).

CentOS-UDI.json

```
{
    "builders": [
        {
            "boot_command": [
                "<esc>",
                "<wait>linux ks=http://{{.HTTPIP }}:{{ .HTTPPort}}/centos-ks-UDI.txt<enter>"
            ],
            "boot_wait": "5s",
            "disk_size": 50000,
            "guest_os_type": "RedHat_64",
            "headless": false,
            "http_directory": "./.httpfiles",
            "iso_checksum": "939fd1d87c11ffe7795324438b85adfb",
            "iso_checksum_type": "md5",
            "iso_url": "http://mirror.simwood.com/centos/6.5/isos/x86_64/CentOS-6.5-x86_64-netinstall.iso",
            "ssh_password": "root",
            "ssh_username": "root",
            "type": "virtualbox-iso",
            "vboxmanage": [
                ["modifyvm", "{{.Name}}", "--memory", "4096"],
                ["modifyvm", "{{.Name}}", "--cpus", "2"]
            ],
            "vm_name": "packer-centos-6.5-64bit-v2-UDI",
            "shutdown_command": "shutdown -P now"
        }
    ]
}
```

CentOS auto installation is based on “Kickstart” scripts. Packer provides this script via an embedded HTTP server which serves dat from:

./.httpfiles/centos-ks-UDI.txt

```
text

# Use network installation
url --url="http://mirror.centos.org/centos/6/os/x86_64"

install

# setup the network with DHCP
network --device=eth0 --bootproto=dhcp

lang en_US.UTF-8

keyboard us

#cdrom
lang en_US.UTF-8

keyboard us

timezone --utc Europe/London

#rootpw  --iscrypted $$nndedddewfooofcerd3r434
rootpw "root"

# Cloudera-Manager does not like selinux
selinux --disabled
```

```
# Custom user added
user --name=cumulus --groups=users --password=rdf

authconfig --enableshadow --passalgo=sha512 --enablefingerprint

firewall --service=ssh

# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --all --drives=sda

ignoredisk --only-use=sda

part /boot --fstype=ext2 --asprimary --size=500
part / --fstype=ext4 --asprimary --size=20480
part /home --fstype=ext4 --asprimary --size=10480
part swap --asprimary --size=2048

bootloader --location=mbr --driveorder=sda --append="nomodeset rhgb quiet"

# packages that will be installed, anything starting with an @ sign is a yum package group.
%packages
@base
@core

%end
%post --log=/root/my-post-log
exec < /dev/tty3 > /dev/tty3
chvt 3

echo
echo "################################"
echo "# Running Post Configuration   #"
echo "################################"
# prevent future yum updates pulling down & install new kernels (and breaking VMware & video drivers).
echo "exclude=kernel*" >> /etc/yum.conf
# update the system
yum update -y
# reboot
shutdown -r now
```

CentOS-UDI-FINAL-ovf.json

```
{
    "builders": [
        {
            "boot_wait": "5s",
            "headless": false,
            "ssh_password": "root",
            "ssh_username": "root",
            "ssh_wait_timeout": "30m",
            "type": "virtualbox-ovf",
            "source_path": "output-virtualbox-iso/packer-centos-6.5-64bit-v2-UDI.ovf",
            "vboxmanage": [
                ["modifyvm", "{{.Name}}", "--memory", "4096"],
                ["modifyvm", "{{.Name}}", "--cpus", "2"]
            ],
            "vm_name": "packer-centos-6.5-64bit-UDI-v4",
            "shutdown_command": "sudo shutdown -P now"
        }
    ],
    "provisioners": [
          {
                          "type": "shell",
                          "inline": ["svn checkout http://cumulusrdf.googlecode.com/svn/trunk
/home/cumulus/cumulus-rdf-readonly"]
          },       {
                          "type": "shell",
                          "inline": ["chown cumulus:users -R /home/cumulus/cumulus-rdf-readonly"]
          },
          {
                          "type": "shell",
                          "inline": ["chmod 777 /home/cumulus/cumulus-rdf-readonly/src/vm/bootstrap.sh"]
          },
          {
                          "type": "shell",
                          "inline": ["/home/cumulus/cumulus-rdf-readonly/src/vm/bootstrap.sh"]
          },
          {
                          "type": "shell",
                          "inline": ["yum install mc -y"]
          }
          ]
}
```

Build Procedure

We run packer with the following command:

```
$ ./packer validate CentOS-UDI.json
Template validation failed. Errors are shown below.
Errors validating build 'virtualbox-iso'. 1 error(s) occurred:
* Output directory 'output-virtualbox-iso' already exists. It must not exist.
$ mv output-virtualbox-iso/ output-virtualbox-iso-v1/
```

In order to rerun packer, one has to clean artefacts from previous runs. A packer description file should be validated before the build is started.

```
$ ./packer validate CentOS-UDI.json
Template validated successfully.
```

Now we can build a first temporary image:

```
$ ./packer build CentOS-UDI.json
virtualbox-iso output will be in this color.

==> virtualbox-iso: Downloading or copying Guest additions checksums
    virtualbox-iso: Downloading or copying:
http://download.virtualbox.org/virtualbox/4.3.10/SHA256SUMS
==> virtualbox-iso: Downloading or copying Guest additions
    virtualbox-iso: Downloading or copying:
http://download.virtualbox.org/virtualbox/4.3.10/VBoxGuestAdditions_4.3.10.iso
==> virtualbox-iso: Downloading or copying ISO
    virtualbox-iso: Downloading or copying:
http://mirror.simwood.com/centos/6.5/isos/x86_64/CentOS-6.5-x86_64-netinstall.iso
==> virtualbox-iso: Starting HTTP server on port 8318
==> virtualbox-iso: Creating virtual machine...
==> virtualbox-iso: Creating hard drive...
==> virtualbox-iso: Creating forwarded port mapping for SSH (host port 4077)
==> virtualbox-iso: Executing custom VBoxManage commands...
    virtualbox-iso: Executing: modifyvm packer-centos-6.5-64bit-v2-UDI --memory 4096
    virtualbox-iso: Executing: modifyvm packer-centos-6.5-64bit-v2-UDI --cpus 2
==> virtualbox-iso: Starting the virtual machine...
==> virtualbox-iso: Waiting 2s for boot...
==> virtualbox-iso: Typing the boot command...
==> virtualbox-iso: Waiting for SSH to become available...
```

The second image is created with:

```
$ ./packer build CentOS-UDI-FINAL-ovf.json
```

The bootstrap script is executed automatically at the end of build procedure:

```
#############################################################
# Install Maven3
# Download the current maven version from the prescribed repository:
# check the version, maybe a more recent version is available ...
wget http://www.eng.lsu.edu/mirrors/apache/maven/maven-3/3.1.1/binaries/apache-maven-3.1.1-bin.tar.gz

# Extract the archive to the maven home directory: /usr/local/
cp apache-maven-3.1.1-bin.tar.gz /usr/local
rm apache-maven-3.1.1-bin.tar.gz
cd /usr/local
tar -zxvf apache-maven-3.1.1-bin.tar.gz

# Create a sym link..
ln -s apache-maven-3.1.1 maven

# Define environment variables

echo 'export M2_HOME=/usr/local/apache-maven-3.1.1' >> /home/$USER/.bashrc
echo 'export PATH=${M2_HOME}/bin:${PATH}' >> /home/$USER/.bashrc
```

```
# Execute the environment changes with the command, ... and test mvn
source /home/$USER/.bashrc
chmod 777 /usr/local/apache-maven-3.1.1/bin/mvn
/usr/local/apache-maven-3.1.1/bin/mvn -version

# Build the current version of CumulusRDF and load all maven artefacts
cd /home/cumulus/cumulus-rdf-readonly
/usr/local/apache-maven-3.1.1/bin/mvn -DskipTests clean compile package cassandra:stop cassandra:start tomcat7:run
```

After some more minutes one can login as user cumulus (password: rdf) and start the embedded Apache Cassandra and Apache Tomcat servers via the following maven command:

```
cumulus$ cd cumulus-rdf-readlony

cumulus$ mvn -DskipTests clean package cassandra:stop cassandra:start tomcat7:run
```

This maven build takes some minutes for the first time it is started. In future executions it is much faster, because many libraries are loaded only during initial run. Now it is time to start a browser to open the CumulusRDF dashboard (http://127.0.0.1:9090/cumulus).

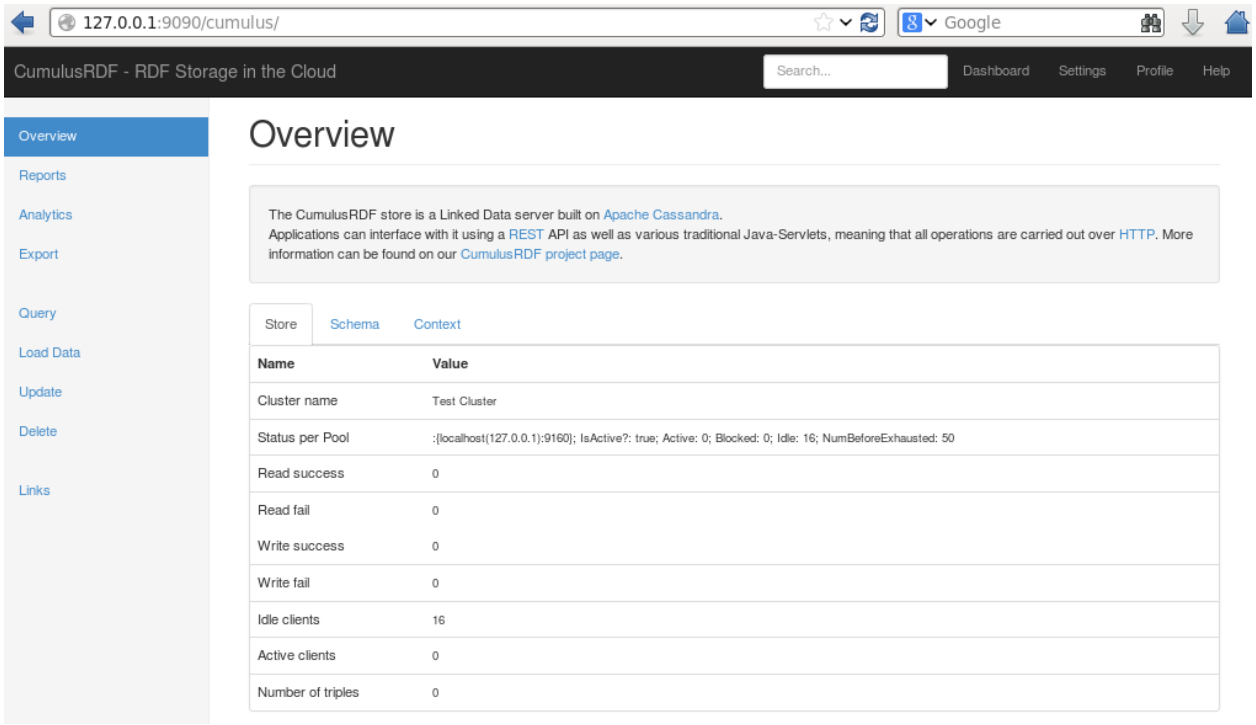

Fig 1: CumulusRDF dasboard in firefox on http://127.0.0.1:9090/cumulus.

Congratulations, you have a UDI image now for some local CumulusRDF tests!

Next Steps: We have to present some examples to illustrate CumulusRDF usage.