
Zenoss ZCA Utils Documentation

Release 1.0

ZCA

July 18, 2016

1	Converting Zenoss VMWare Virtual Appliance to VirtualBox	1
1.1	Purpose	1
1.2	Instructions	1
1.2.1	Linux	1
1.2.2	Windows	3
2	Zenpack Egg to Github	5
2.1	Purpose	5
2.1.1	Assumptions	5
2.1.2	Setup Your Zenoss Box To interact With Your GitHub Account	6
2.1.3	Do the Conversion	6
3	Fast Track an Alpha 4 Build on RHEL or Derivative	9
3.1	Purpose	9
3.2	Requirements/Recommendations	9
3.3	Run It	9
4	Description	11
5	Indices and tables	13

Converting Zenoss VMWare Virtual Appliance to VirtualBox

Author David Petzel

Date 02/18/2012

Contents

- *Converting Zenoss VMWare Virtual Appliance to VirtualBox*
 - *Purpose*
 - *Instructions*
 - * *Linux*
 - *Scripted*
 - *Manual*
 - * *Windows*
 - *Scripted*
 - *Manual*

1.1 Purpose

So you can use the VMWare Appliance Zenoss Inc. provides under VirtualBox

1.2 Instructions

1.2.1 Linux

Scripted

If you want to save yourself some copy/paste time, a script is available that should automatically handle most of the steps below. The following short set of commands should get you up and running. If you have an issue with the script, you should be able to fall back on the manual steps, or maybe even fix the script and let me know the changes so I can incorporate them:

```
cd /tmp
wget https://raw.githubusercontent.com/ZCA/zenoss_zca_utils/master/source/scripts/vmware_appliance_to_virtualbox
chmod +x vmware_appliance_to_virtualbox.sh
./vmware_appliance_to_virtualbox.sh
```

At this point the VM should be converted and booting up. Once it has finished booting, login to the console and run the following commands *from within* the virtual machine:

```
cd /tmp
wget https://raw.githubusercontent.com/ZCA/zenoss_zca_utils/master/source/scripts/vmware_appliance_to_virtualbox
chmod +x replace_tools_with_additions.sh
./replace_tools_with_additions.sh
```

Manual

- Download the existing VMware Image:

```
cd /tmp
wget http://downloads.sourceforge.net/project/zenoss/zenoss-alpha/4.1.70-1427/zenoss-4.1.70-1427
```

- Unzip the downloaded zip file:

```
unzip zenoss-4.1.70-1427-x86_64.vmware.zip
```

- Create a new VirtualBox VM:

```
VM_NAME="Zenoss_4.1_Appliance"
VM_BASE_PATH=/VMs
sudo mkdir $VM_BASE_PATH
sudo chmod 777 $VM_BASE_PATH
VBoxManage createvm --name $VM_NAME --basefolder $VM_BASE_PATH --register
```

- Move the VMDK file over to the VM's directory:

```
mv zenoss-4.1.70-1427-x86_64.vmdk $VM_BASE_PATH/$VM_NAME/
```

- Change Settings on the newly created VM:

```
VBoxManage modifyvm $VM_NAME --ostype RedHat_64 --memory 2048 --nic1 nat --nictype1 82545EM --io
```

- Attach the VMDK file to the VM:

```
VBoxManage storagectl $VM_NAME --name "SCSI Controller" --add scsi --controller LsiLogic
VBoxManage storageattach $VM_NAME --storagectl "SCSI Controller" --type hdd --port 0 --medium $V
```

- Some Optional Port forwards I find useful:

```
VBoxManage controlvm $VM_NAME natpf1 "SSH,tcp,,8022,,22"
VBoxManage controlvm $VM_NAME natpf1 "ZOPE,tcp,,8080,,8080"
```

- Add a DVD/CD Drive (At a minimum you'll need this for installing Guest Additions):

```
VBoxManage storagectl $VM_NAME --name "IDE Controller" --add ide --controller PIIX4
VBoxManage storageattach $VM_NAME --storagectl "IDE Controller" --type dvddrive --port 1 --device
```

- Power on the new Virtual Machine:

```
VBoxManage startvm $VM_NAME
```

- Once the VM has started up, log into the console and Remove VMWare Tools:

```
vmware-uninstall-tools.pl
```

- Use the VirtualBox documentation to install VirtualBox Guest Additions

1.2.2 Windows

Scripted

Maybe some day

Manual

Lets get our powershell on...All commands are run in a power shell prompt

- Setup some Variables:

```
$buildNumber = "4.1.70-1434"
$arch = "x86_64"
$baseFileName = "zenoss-$buildNumber-$arch"
$zipFileName = "$baseFileName.vmware.zip"
$zipFileDownloadUrl = "http://downloads.sourceforge.net/project/zenoss/zenoss-alpha/$buildNumber/$baseFileName.zip"

$VM_NAME="Zenoss_Appliance_$buildNumber"
$VM_BASE_PATH="\VMs"
```

- Download the existing VMware Image:

```
cd \temp
$webclient = New-Object System.Net.WebClient
echo "Going to Download File. This will take a long time without output. Be Patient"
$webclient.DownloadFile($zipFileDownloadUrl,"$pwd\$zipFileName")
```

- Unzip the downloaded zip file:

```
$shell_app=new-object -com shell.application
$zip_file = $shell_app.namespace((Get-Location).Path + "\"$zipFileName")
$destination = $shell_app.namespace((Get-Location).Path)
$destination.Copyhere($zip_file.items())
```

- Create a new VirtualBox VM:

```
if ((Test-Path -path $VM_BASE_PATH) -ne $True){New-Item $VM_BASE_PATH -type directory}
VBoxManage createvm --name $VM_NAME --basefolder $VM_BASE_PATH --register
```

- Move the VMDK file over to the VM's directory:

```
mv $baseFileName\$baseFileName.vmdk $VM_BASE_PATH\$VM_NAME\
```

- Change Settings on the newly created VM:

```
VBoxManage modifyvm $VM_NAME --ostype RedHat_64 --memory 2048 --nic1 nat --nictype1 82545EM --ioapic
```

- Attach the VMDK file to the VM:

```
VBoxManage storagectl $VM_NAME --name "SCSI Controller" --add scsi --controller LsiLogic
VBoxManage storageattach $VM_NAME --storagectl "SCSI Controller" --type hdd --port 0 --medium $VM_NAME\$baseFileName.vmdk
```

- Add a DVD/CD Drive (At a minimum you'll need this for installing Guest Additions):

```
VBoxManage storagectl $VM_NAME --name "IDE Controller" --add ide --controller PIIX4
VBoxManage storageattach $VM_NAME --storagectl "IDE Controller" --type dvddrive --port 1 --device $VM_NAME\ISO\GuestAdditions.iso
```

- Some Optional Port forwards I find useful:

```
VBoxManage controlvm $VM_NAME natpf1 "SSH,tcp,,8022,,22"  
VBoxManage controlvm $VM_NAME natpf1 "ZOPE,tcp,,8080,,8080"
```

- Power on the new Virtual Machine:

```
VBoxManage startvm $VM_NAME
```

- Once the VM has started up, log into the console (root/zenoss) and Remove VMWare Tools:

```
vmware-uninstall-tools.pl
```

- While still logged into the console, install VirtualBox guest additions:

```
yum -y install bzip2 make gcc  
mkdir /media/ga  
mount /dev/cdrom /media/ga  
/media/ga/VBoxLinuxAdditions.run
```

- Reboot for good measure:

```
reboot
```

Zenpack Egg to Github

Author David Petzel

Date 02/18/2012

Contents

- *Zenpack Egg to Github*
 - *Purpose*
 - * *Assumptions*
 - * *Setup Your Zenoss Box To interact With Your GitHub Account*
 - * *Do the Conversion*
 - *Automated*
 - *Manual*

2.1 Purpose

There are times when someone might develop a ZenPack in the comfort of their own environment and later wish to contribute it back to the community. In these case the original author maybe unable, unwilling, or just not interested in making the ZenPack source available on GitHub as is being recommended by Zenoss Inc.

There has been some discussions around allowing the ZCA to accept the standalone egg file and will handle getting the ZenPack into GitHub. This document will outline the steps to take the egg and get into GitHub under the ZCA organization.

Its worth noting that you *do* need to do the conversion on a machine running zenoss. I'd suggest you get the git client installed on your zenoss machine. I personally think its easiest to also setup an SSH key pair under the zenoss user on my zenoss development instance and link that key to my account.

2.1.1 Assumptions

- Your running all these commands on your zenoss box
- You have configured your zenoss box to work with your GitHub account. This is a one time setup per box, regardless of how many packs you convert. If you have not already done this, see the section *Setup Your Zenoss Box To interact With Your GitHub Account*
- You are converting a fictional ZenPack named ZenPacks.community.gitify which was supplied to you as ZenPacks.community.gitify-1.0-py2.6.egg

- You have saved the egg file in your /tmp/ directory on your zenoss box

2.1.2 Setup Your Zenoss Box To interact With Your GitHub Account

Ensure you have the git client installed on your Zenoss box. You'll also need to setup an SSH key pair. I'd suggest setting up a key pair under your zenoss user login. GitHub has a pretty good document that covers this so I won't re-invent the wheel. <http://help.github.com/linux-set-up-git/>

I'd suggest you do make sure to setup your GitHub token, as it will allow us to do everything in conversion process from the command line on our Zenoss box, without having to drop out to the web UI to create the repo on GitHub when the time comes. Additionally setting up that API should allow a script to be written at a future date which can ease all of the steps below.

2.1.3 Do the Conversion

Automated

Save the egg file in /tmp/ and run the following:

```
wget https://raw.github.com/ZCA/zenoss_zca_utils/master/source/scripts/gitify_zenpack/gitify_zenpack.sh
sh gitify_zenpack.sh <eggfile_name>
```

Manual

The first thing we need to do is install the egg:

```
zenpack --install=ZenPacks.community.gitify-1.0-py2.6.egg
```

Now, switch the pack to development mode. This step is also discussed in the Developers guide:

```
cp $ZENHOME/Products/ZenModel/ZenPackTemplate/* $ZENHOME/ZenPacks/ZenPacks.community.gitify-1.0-py2.6
```

Open a zendmd shell:

```
zendmd
```

Run the following snippet in a zendmd shell. What I found was that after doing the switch to development mode, the setup.py that gets created has an empty name field, which of course makes sense given it came from a template, however with that blank name, the relocate fails as it tries to use the setup.py with the blank name. The following snippet, populates the setup.py based on the info that was contained in the original egg:

```
for pack in dmd.ZenPackManager.packs():
    if pack.id == "ZenPacks.community.gitify":
        pack.writeSetupValues()
        exit()
```

Next relocate the files outside of the zenoss directory:

```
cp -r $ZENHOME/ZenPacks/ZenPacks.community.gitify-1.0-py2.6.egg /tmp/ZenPacks.community.gitify
zenpack --link --install=/tmp/ZenPacks.community.gitify
```

Now lets initialize a git repo. This will be a local git repo and in no way tied to GitHub yet:

```
cd /tmp/ZenPacks.community.gitify
git init
```

You should see something that looks like:

```
Initialized empty Git repository in /tmp/ZenPacks.community.gitify/.git/
```

Now that we have an empty git repo, lets setup a few default files. Of note, we are going to pull down a standard .gitignore file supplied by Zenoss, as well as creating our base README.rst file which GitHub wants and will also serve as the file in which we will document the ZenPack:

```
wget https://raw.githubusercontent.com/zenoss/Community-ZenPacks-SubModules/master/.gitignore
touch README.rst
```

Now we commit all the stuff we just did (This still won't result in interactions with GitHub):

```
git add .
git commit -a -m 'Initial Commit - Post EGG Extraction'
```

Make your local git repo aware of the version on GitHub (no actual interaction occurs yet):

```
git remote add origin git@github.com:ZCA/ZenPacks.community.gitify
```

Now we actually create the repo on GitHub. You can do this in the Web UI or using the API:

```
github_user=`git config --global github.user`
github_key=`git config --global github.token`
curl -k -F "login=$github_user" -F "token=$github_key" -i https://github.com/api/v2/json/repos/create
```

You will know this works based on the response. You'll see some JSON indicating success. Now its time to push everything up to GitHub:

```
git push -u origin master
```

You can now remove the pack from your installation:

```
cd /tmp
zenpack --remove=ZenPacks.community.gitify
```

That should just above cover it. You can test by checking out the new git repo into a separate directory and doing a development install:

```
mkdir /tmp/install_test
cd /tmp/install_test
git clone git://github.com/ZCA/ZenPacks.community.gitify.git
zenpack --link --install=ZenPacks.community.gitify
zenpack --list
zenpack --remove=ZenPacks.community.gitify
```

Fast Track an Alpha 4 Build on RHEL or Derivative

Author David Petzel

Date 02/18/2012

Contents

- *Fast Track an Alpha 4 Build on RHEL or Derivative*
 - *Purpose*
 - *Requirements/Recommendations*
 - *Run It*

3.1 Purpose

In order to take some of the tedious process around building new Zenoss Core Alpha 4 builds I put together a small shell script to handle all the mundane tasks. Its nothing fancy, but should hopefully save you some time in tracking down various downloads and such

The script is completely dependant on the fact that you are running it on a clean/fresh RHEL derivative. Its not going to work on another distro like Ubuntu.

3.2 Requirements/Recommendations

The following requirements must be met:

- You must be running on a RHEL derivative.
- Ensure you meet all other requirements as outlined in the official installation guide.
- The machine you are running this on will need access to various internet sites.
- You should start with a minimal install to avoid any dependency issues
- You might want to disable iptables, or at least open up 8080 so you access the web ui after instalation is complete

3.3 Run It

Just run the following to get the ball rolling:

```
wget -N https://raw.githubusercontent.com/ZCA/zenoss_zca_utils/master/source/scripts/4x_alpha_builder/4x_alpha_build_$HOSTTYPE.sh
sh 4x_alpha_build_$HOSTTYPE.sh
```

Description

A collection of various utilities, scripts, etc maintained by ZCA members

Indices and tables

- `genindex`
- `modindex`
- `search`