

System of Making Images portable within xCAT

vallard@sumavi.com

Use Case

We want to create a system of making xCAT images more portable so that they can be shared and prevent people from reinventing the wheel. While every install is unique there are some things that can be shared among different sites to make images more portable. In addition, creating a method like this allows us to create snap shots of images we may find useful to revert to in different situations.

Importing

1. User downloads a tarball from somewhere. (Sumavi.com will be hosting many of these)
2. User runs: `xcatimport <tarball>`
3. The `xcatimport` command fills out the `osimage` table and other tables, and populates file directories with appropriate files from the tarball.
4. The user runs: `nodeset image=<image name>` and the node is able to install.

Exporting

1. The user defines the appropriate entries in the OS image table
2. The user runs: `xcatexport <tarballname>`
3. A tarball is placed in the current directory or one specified by tarball name. This image can now be used on other systems.

How it works

You can only export/import one image at a time. Each tarball will have the following simple structure:

```
manifest.cfg  
<files>
```

manifest.cfg

The manifest.cfg will be analogous to a kickstart file where it tells xCAT how to store the items:

Manifest.cfg:

```
name <name> # required. Eg: My_Image
os <os> #required. Eg; centos5.4
arch x86_64 #required
profile compute # required. Eg: hpc_image
method <install|netboot|statelite> # required
initrd <filename> # required for netboot|statelite
template <filename> # required for install
rootimg <filename> # required for netboot|statelite
kernel <filename> # required for netboot|statelite
media <required> # required for install
extra <filename:path> # optional
extra <filename:path> # optional
```

disk install example

For a kickstart install the following files would be included in the tarball:

```
manifest.cfg
foo.tmpl
```

The manifest.cfg file would look like:

```
name Foo_Image
os rhels5.4
arch x86_64
profile foo
method install
template foo.tmpl
media required
```

The foo.tmpl would be copied into /install/custom/rhels5.4/x86_64/foo.tmpl and the appropriate entries would then be placed in the osimage table. The user will then be notified that they need to run copycds for this to work. Provided they have done so, they can now run:

```
nodeset n01-n10 osimage=Foo_Image
```

Then the node will install.

Exporting requires that the osimage table be filled out properly:

```
"Foo_Image", "foo", "linux", "install", "Linux", "centos5.4", "centos5.4", "x86_64", , ,
```

If an entry is not found, then xcatexport will signal an error.

Stateless example

For a stateless image example the following files will be included in the tarball:

```
manifest.cfg
rootimg.gz
initrd.gz
kernel
```

The manifest.cfg file would look like:

```
name Foo_Stateless
os rhels5.4
arch x86_64
profile foo
method netboot
initrd initrd.gz
kernel kernel
rootimg rootimg.gz
```

Now the directory /install/netboot/centos5.4/x86_64/foo will be created and the kernel, initrd.gz, and rootimg.gz files will be copied here. The rootimg.gz will also be expanded. The osimage table will be filled out and the user can now run
nodeset n01-n10 osimage=Foo_Image

Then the node will netboot. Note that the installation media is not required in this example.

Also notice that the pkglist and the exlist files are not copied by default. They can be added with the extras options if desired.

Design Considerations:

1. What about postscripts?

Postscripts are not tied to images, they are tied to node groups. In my opinion postscripts are dead dogs if you're flipping OSES all the time. If you want to copy postscripts you can use the extra tag in the manifest:

```
extra my_cool_postscript:/install/postscripts
```

or:

```
xcatexport Foo_Image -
extra=/install/postscripts/my_cool_postscript:/install/postscripts
```