Operating system images with bootc

Production Line

The bootc project lets you use a container-based workflow to create bootable operating system images. In a few simple steps, we show you how to use Podman to create ready-to-run images. By Thorsten Scherf

Container images generally only provide the components you need to run a specific application inside a container and do not include a Linux kernel, a bootloader, firmware, or similar components. Many workflows already in place create and manage container images for applications, typically with a zoo of tools that create an image, check it for vulnerabilities, and then publish it in a container store. This workflow would also be helpful for bootable system images – which is exactly where two projects, bootc [1] and bootc Image Builder [2], enter the scene.

Containers for Operating Systems

These bootc projects let you use the same workflow you have in place for application containers to create operating system containers. All you need is a container or Dockerfile along with the bootc Image Builder service to create a bootable Open Container Initiative (OCI) image that you can start on a host or virtual machine.

An image created in this way also comes with the Linux kernel, which is loaded in the normal way at boot time. Therefore, the basic operating system does not reside in the container and a systemd process with a PID of 1 is available in the normal way. Transactional updates let you keep the system up to date. The updates create additional layers on the system on top of the base image. You just need to create an updated bootc base image, with no need for a new disk image. In this way, the update process is identical to other OSTree-based systems.

You can use the `bootc-image-builder` container tool to create bootable disk images with the aid of the Podman container runtime. This process generates a disk image from a base image. Alternatively, you can also use the graphical Podman Desktop [3] tool, which simplifies the container handling and comes with a selection of extensions – including some for bootc – to help create bootable container images. I will primarily be focusing on the bootc extension in this article.

Generating Images in a Terminal Window

To begin, I'll first take a look at the process of generating disk images in

Thorsten, Is the spacing at the start of the highlighted line of code correct – it's currently not marked as a continuation of the previous line. Thank you, AV

a terminal window from a standard bootc image. Of course, you always have the options of extending this bootc image to include additional components or making other adjustments in advance with the use of a container file or Dockerfile. You then publish the customized image in a store, which is also necessary when, for example, updates become available for the image you are using. You can use either the RAW, QCOW2, VMDK, ISO, or Amazon Machine Image (AMI) formats for the disk image. In any case, you always need a bootc base image first. This special container image contains the kernel and the systemd service along with other components for a bootable system. Base images are available for Fedora, CentOS, and Red Hat Enterprise Linux systems. The following example shows that the components are included in the image:

```
# podman run --rm ➥
  -it quay.io/centos-bootc/centos-bootc:➥
   stream9 rpm ➥
  -q kernel systemd podman bootc
kernel-5.14.0-467.el9.aarch64
systemd-252-37.el9.aarch64
podman-5.1.0-1.el9.aarch64
bootc-0.1.11-2.el9.aarch64
```

Because this kind of image does not contain a user account by default, you can simply add one by editing the configuration file in TOML format and assign an SSH key and a group directly to the user:

```
[[customizations.user]]
name = "tscherf"
password = "Secret23"
key = "ssh-rsa AAA ... tscherf@domain.com"
groups = ["wheel"]
```

In this example, I used `centos-bootc:stream9` as the bootc base image and passed it into the `bootc-image-builder` service. The completed disk image uses the QCOW2 format. If everything goes well, the new image will be available in the `output` directory – but don't hold your breath. You can then launch the build process for the CentOS Stream 9 image itself, as in Listing 1.

Booting the Image

To boot the disk image you just created for test purposes, you can use a virtualization tool such as libvirt. This example,

```
# virt-install ➥
  --name bootc ➥
  --memory 4096 ➥
  --vcpus 2 ➥
  --disk qcow2/disk.qcow2 ➥
  --import
```

uses `virt-install`, which is based on libvirt.

Podman Desktop

The Podman Desktop graphical front end is an alternative to the build process in a terminal. The tool provides a bootc extension and then automatically takes care of generating a new bootable container image for you. To do so, press the *Build* button and then simply select the desired bootc base image from the pull-down menu along with the desired image format. The tool then automatically launches the Image Builder Service in a container and creates the new image there.

Of course, you can also adapt the bootc base image to your own requirements up front before you create a bootable disk image from it. This example,

```
FROM quay.io/centos-bootc/centos-bootc:➥
     stream9
RUN dnf -y install httpd && ➥
    systemctl enable httpd && ➥
    mv /var/www /usr/share/www && ➥
    fed -ie 's,/var/www,/usr/share/www,'
  /etc/httpd/conf/httpd.conf
RUN rm -rf /usr/share/httpd/noindex
COPY index.html /usr/share/www/html
EXPOSE 80
```

sets up a simple container file for running a web server in a bootc container.

Conclusions

Container images for apps are nothing new, but what about bootable operating systems? Armed with bootc container images and a little help from the bootc Image Builder service, you can quickly create bootable container images for your choice of operating system. You can then use your container-based workflow for both your applications and your operating systems. You will find more information about this new deployment method for operating systems on Red Hat's image mode page [4]. ■

Info

[1] bootc: [https://github.com/containers/bootc]

[2] bootc Image Builder: [https://github.com/osbuild/bootc-image-builder]

[3] Podman Desktop: [https://podman-desktop.io]

[4] RHEL image mode: [https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux/image-mode]

The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Experience group. He is a regular speaker at various international conferences and writes a lot about open source software.

Listing 1: Build Process

```
# podman run --rm -it --privileged --pull=newer --security-opt label=type:unconfined_t -v $(pwd)/config.toml:/config.toml -v $(pwd)/output:/output -v
  /var/lib/containers/storage:/var/lib/containers/storage quay.io/centos-bootc/bootc-image-builder:latest --type qcow2 --local quay.io/centos-bootc/
  centos-bootc:stream9
```