

Create custom images suitable for deploying systems using Red Hat Enterprise Linux image builder

by
Yogesh Chavan
Karan Rai
Nenad Peric

Index

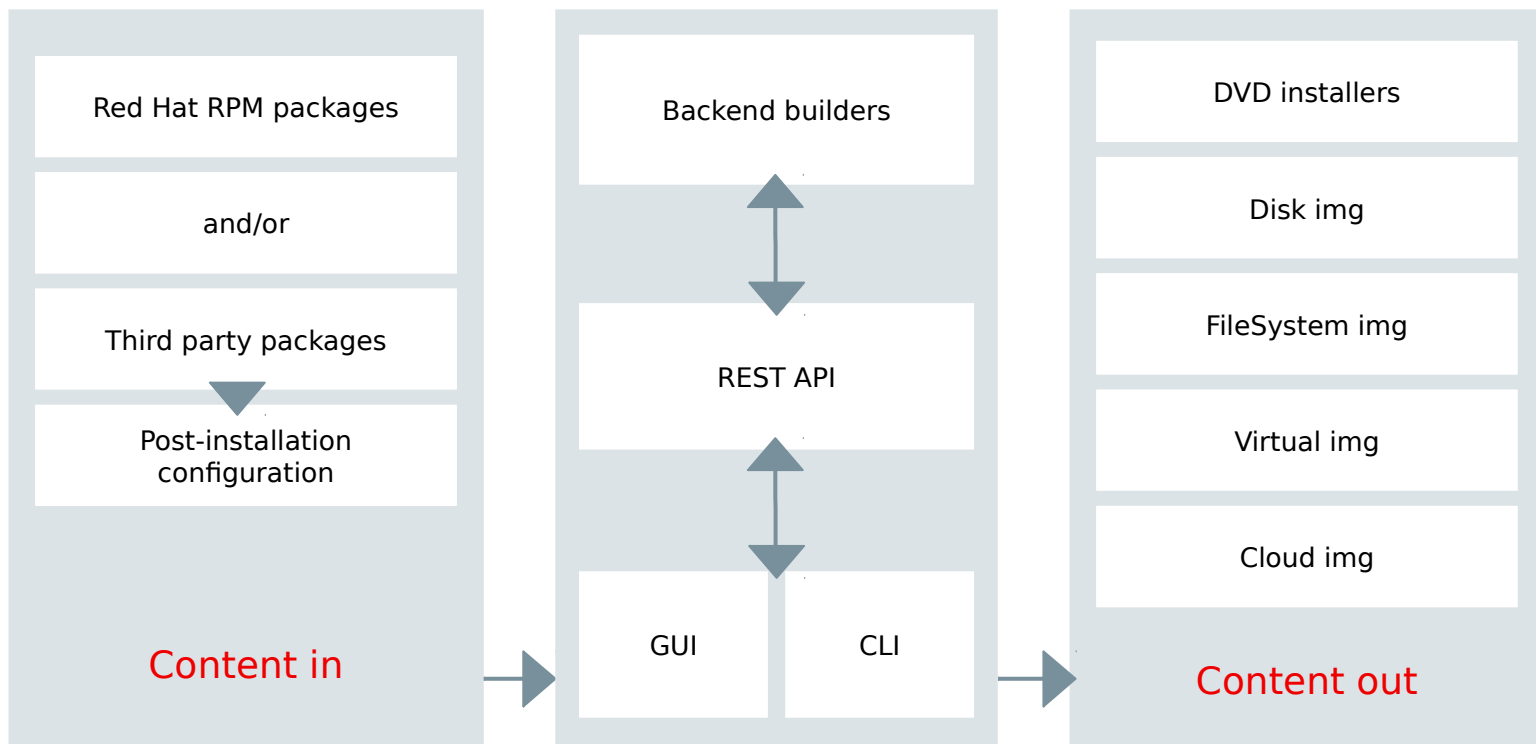
- Image builder
 - What?
 - Why?
 - How?
- Lab

- **What** is Image Builder ?

What is Image Builder?

- Image Builder is an image-building tool. Output image can be DVD installer, disk img, filesystem img, virtual img or cloud image.
- Introduced in RHEL 7.6 and RHEL 8
- Used to create custom deployable images (Customization via rpm packages selection, post-install configuration, etc.)
- Create images in a variety of formats for deployment to a variety of environments
- Customize images for third-party packages and updated RHEL Errata content

What is Image Builder?



• **Why** Image Builder ?

Why Image Builder?

Before Image Builder:

- Creating customized RHEL images was unsupported
- Clients and partners often requested the ability to customize
- Customization grew in importance for cloud environments

Why Image Builder?

- Provides an End-user with the ability to create supported custom RHEL images according to their needs
- Reduces deployment and configuration time on public cloud services
- Can be used to create images for deployment in a disconnected environment
- Output images can be configured for custom repositories (diverge from the Red Hat Content Delivery Network defaults)
- Provides package selection and configuration from a user-friendly web UI in the RHEL 8 web console
- Allows users to save and alter image configuration to create multiple replicas later

Why Image Builder?

- Supported output image formats:
 - Live ISO (.iso)
 - Raw disk (.img)
 - File system (.img)
 - Tarball (.tar.xz)
 - QCOW2 for KVM, Red Hat Virtualization, Red Hat Satellite, and Red Hat CloudForms
 - AMI (Amazon Web Services®)
 - VHD (Microsoft® Azure®)
 - VMDK (VMware® vSphere® Hypervisor)
 - QCOW2 for OpenStack

- **How** to use Image Builder ?

How to use Image Builder?

Two known front-ends

- Command line tool
- GUI via Web Console plugin

How to use Image Builder?

Command line tool

- Actual command : *composer-cli*
- Get help about it : *composer-cli -h*
- Currently it has some functions that are only available in the CLI (e.g. post-install configuration)

How to use Image Builder?

GUI via Web Console plugin

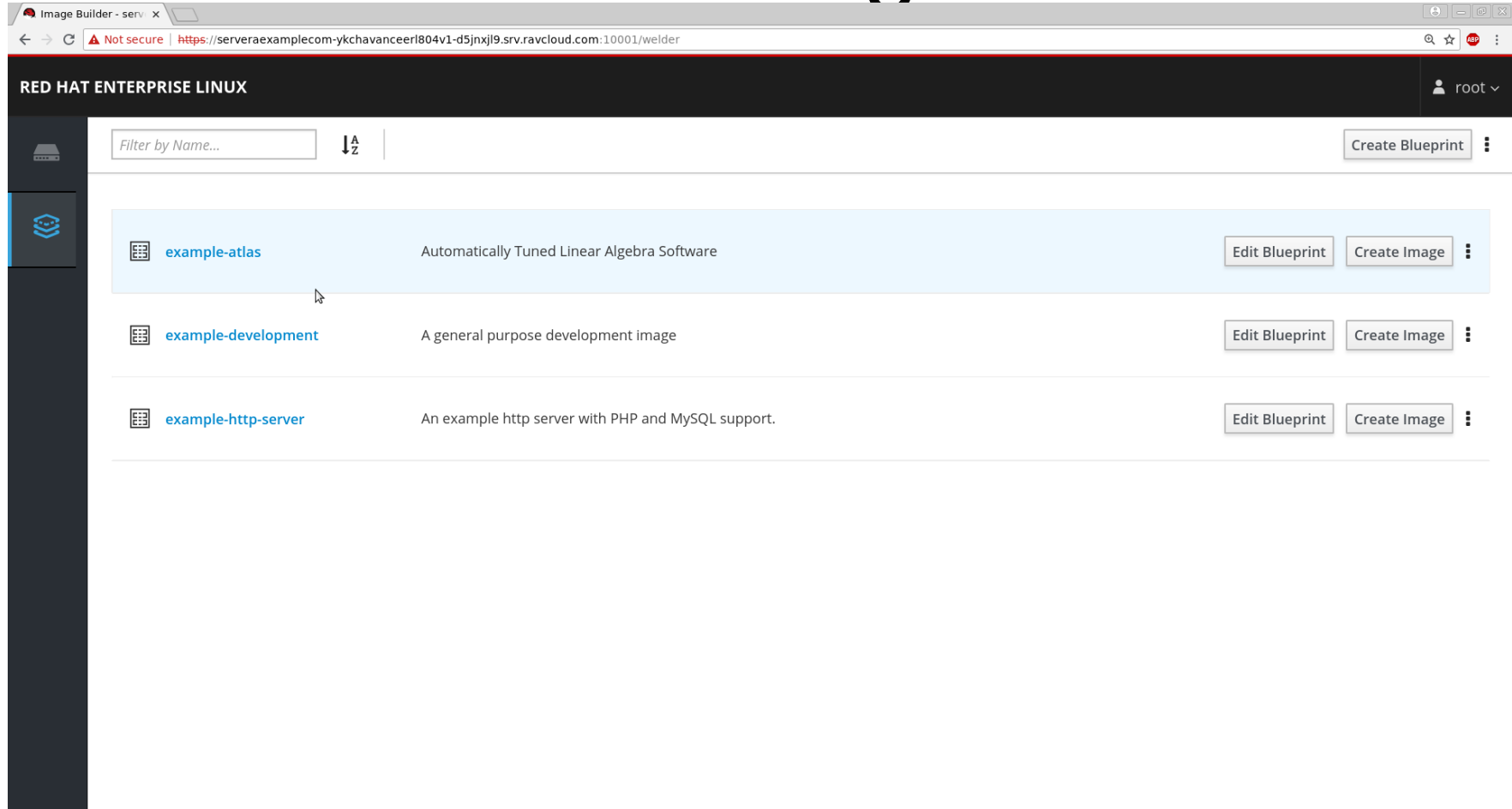
- Wait a minute... What is Web Console?
- Web Console
 - Web based system management tool. Available on all installations except - minimal installation

How to use Image Builder?

Web Console plugin

- Actual plugin name : **cockpit-composer**
- Allows using Image Builder remotely from a web interface
- Does not require having the GUI packages installed on the RHEL system
- Currently the Image Builder functions available in the web console are more limited than in the CLI

How to use Image Builder?



The screenshot displays the Red Hat Image Builder web interface in a browser. The browser's address bar shows a URL starting with 'https://serveraexamplecom-ykchavanceerl804v1-d5jnxjl9.srv.ravcloud.com:10001/welder'. The interface has a dark header with 'RED HAT ENTERPRISE LINUX' and a user profile 'root'. A sidebar on the left contains icons for 'Images' and 'Blueprints'. The main content area features a search bar 'Filter by Name...' and a 'Create Blueprint' button. Below this, a list of blueprints is shown:

Blueprint Name	Description	Actions
example-atlas	Automatically Tuned Linear Algebra Software	Edit Blueprint, Create Image
example-development	A general purpose development image	Edit Blueprint, Create Image
example-http-server	An example http server with PHP and MySQL support.	Edit Blueprint, Create Image

How to use Image Builder?

Blueprint

- a list of preselected components (RPM packages) that form a template for a custom image
- Create multiple images in multiple supported formats from the same blueprint
- A blueprint saves a record of the inputs and instructions for an image build

Talk is cheap.
Show me the Lab.

Lab Index

- Lab1 - Installation
- Lab2 - Create blueprint and image
- Lab3 - Test image with virt-install
- Lab4 - Customize blueprint
- Lab5 - Test output image

Lab - setup

- Red Hat Enterprise Linux 8 installed with Web Console
- Web Console enabled using command
 - *systemctl enable cockpit.socket*
- Installed **virt-viewer** and **virt-install** for testing output image

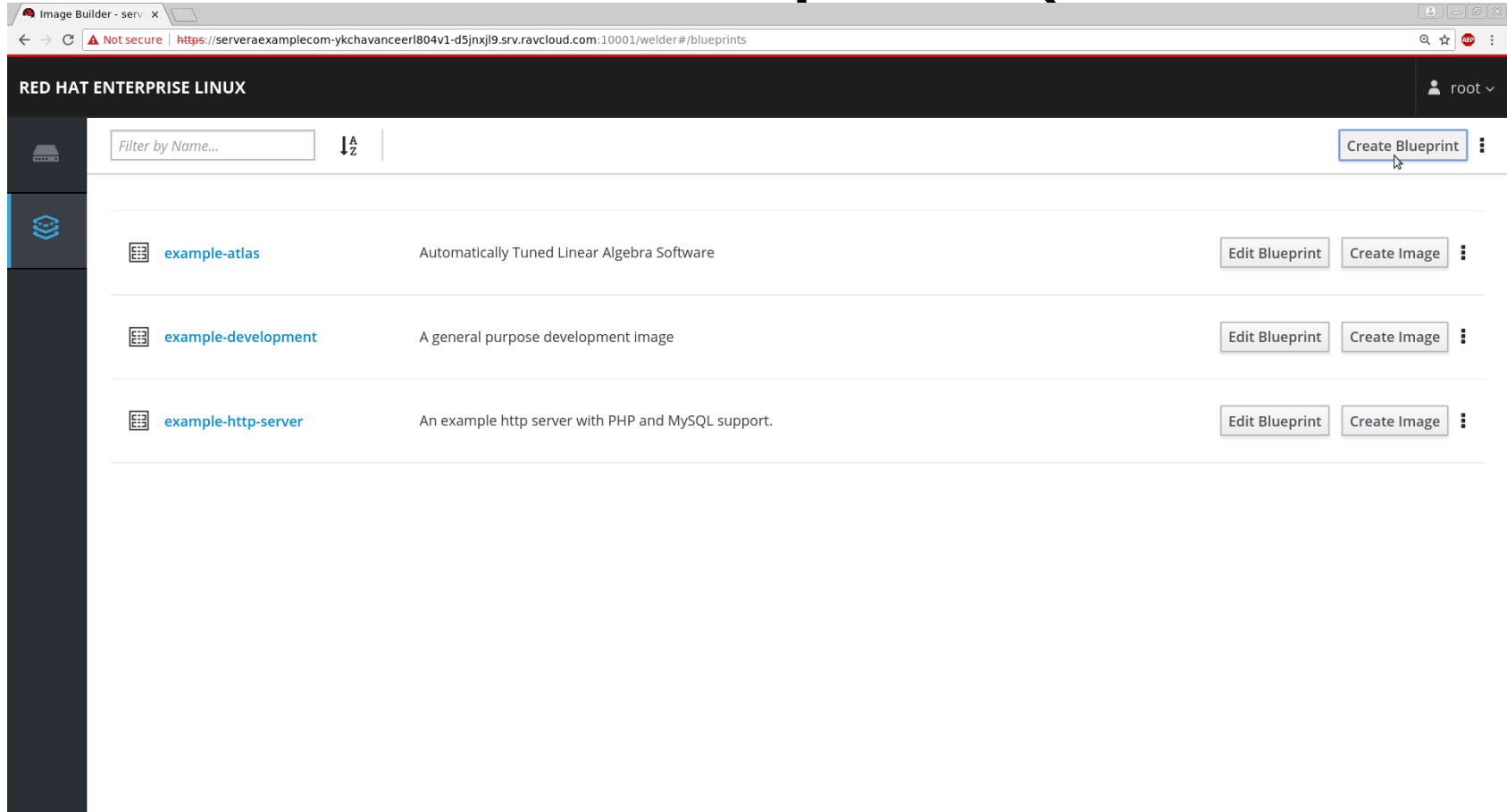
Lab Setup

- *Access Web Console via*
 - *<server hostname>:9090* OR
 - *<server ip>:9090*
 - Username - lab-user
- Become root
 - *sudo -i*




Lab1 - Installation

- Install Image Builder with the CLI and the web console plugin:
 - *yum install lorax lorax-composer composer-cli cockpit-composer*
- Enable and start composer service
 - *systemctl enable --now lorax-composer.socket*
- Restart cockpit service to load newly installed plugin
 - *systemctl restart cockpit.service*
 - Note: Above command disconnects Web Console. Reconnect.

Lab2 - Create blueprint (summit19)



The screenshot shows a web browser window with the URL `https://serveraexamplecom-ykachavanceer1804v1-d5jnxjl9.srv.ravcloud.com:10001/welder#/blueprints`. The page title is "RED HAT ENTERPRISE LINUX" and the user is logged in as "root". A sidebar on the left contains icons for a server and a blueprint. The main content area has a search bar labeled "Filter by Name..." and a "Create Blueprint" button. Below this, there is a table of blueprints:

Blueprint Name	Description	Actions
 example-atlas	Automatically Tuned Linear Algebra Software	Edit Blueprint Create Image ⋮
 example-development	A general purpose development image	Edit Blueprint Create Image ⋮
 example-http-server	An example http server with PHP and MySQL support.	Edit Blueprint Create Image ⋮

Lab2 – Select packages

Image Builder - serv x

Not secure | <https://serveraexamplecom-ykchavanceerl804v1-d5jnxjl9.srv.ravcloud.com:10001/welder#/edit/composer-test>

RED HAT ENTERPRISE LINUX

root

[Back to Blueprints](#) > [composer-test](#) > Edit Blueprint

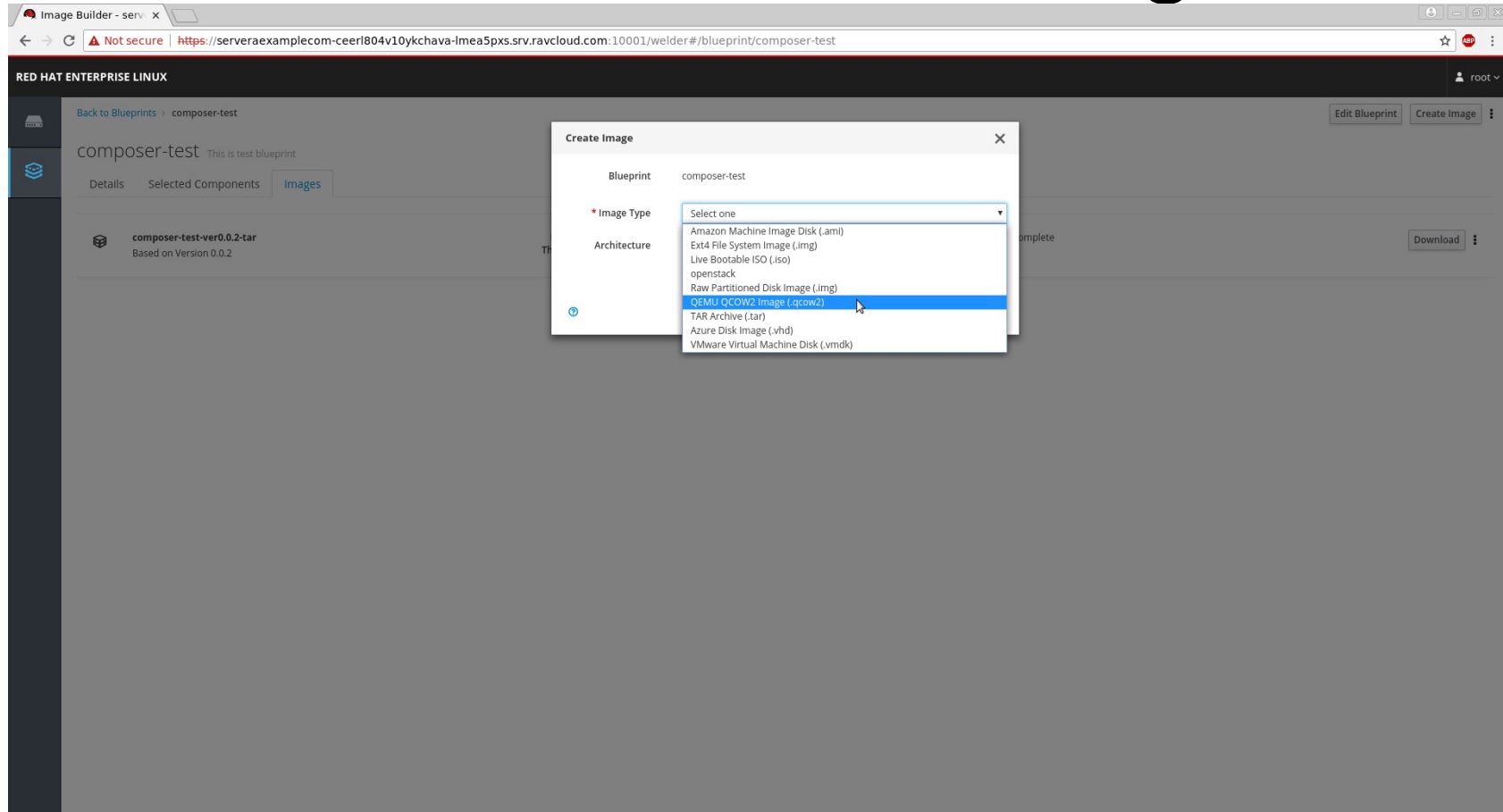
Commit Discard Changes Create Image

composer-test

Available Components	Blueprint Components
<input type="text" value="bison"/> Name: bison Clear All Filters 1 - 2 of 2 < 1 of 1 > <div><div> Select components in this list to add blueprint.</div><div><div> bison A GNU general-purpose parser generator</div><div> bison-runtime Runtime support files used by Bison-generated parsers</div></div><div><div> Add Component Version 3.0.4 Release 10.el8</div><div></div></div></div>	<div><div>Name Filter by Name...</div><div>↓ A Z ↺ ↻</div></div> <div><h3>Add Blueprint Components</h3><p>Browse or search for components, then add them to the blueprint.</p></div>

<https://serveraexamplecom-ykchavanceerl804v1-d5jnxjl9.srv.ravcloud.com:10001/cockpit/b8db3e96bbb3526e7c9b4b997e7c2b66a64b78ef1a07c0aa3df79a495bd83c9/welder/index.html#>

Lab2 – Create image



Lab3 – Test image with virt-install

- Run this command to get GUID of image
 - *composer-cli compose list*
- **Get GUI of the server**
- Command to test image with virt-viewer
 - *virt-install --name RHEL8Lab2 --memory 2048 --vcpus 2 --os-variant rhel8.0 --import --disk /var/lib/lorax/composer/results/<GUID number for that image>/disk.qcow2*

Problem – No **cloud-init**

Images created with Image Builder in the web console:

- Have their root account locked for security purposes
- By default, do not have any other users configured
- Currently can not have a user added using just the web console (requires the CLI)
- This results in images that have no way to log in
- This is not a problem where cloud-init is available

Lab4 – Get blueprint config file

- **Go back to Web Console of the server**
- Download a copy of the blueprint configuration file from Image Builder
 - *composer-cli blueprints save <blueprint-name>*
- You will get file `<blueprint-name>.toml`
- Blueprint configuration files follow TOML (Tom's Obvious, Minimal Language) format, which uses key/value pairs. For more information, see [TOML on GitHub] (<https://github.com/toml-lang/toml>).

Lab4 – Edit blueprint file

- Append following content to <blueprint-name>.toml file
 - `[[customizations.user]]`
 - `name = "myuser"`
 - `password = "mypassword"`
 - `groups = ["users", "wheel"]`

Lab4 – Push and verify config file

- Push the revised blueprint configuration file back to Image Builder:
 - *composer-cli blueprints push* <blueprint-name.toml>
- Verify that your changes appear in the configuration file
 - *composer-cli blueprints show* <blueprint-name>

Lab5 – Test image with virt-install

- Run this command to get GUID of image
 - *composer-cli compose list*
- **Get GUI of the server**
- Command to test image with virt-viewer
 - *virt-install --name RHEL8Lab2 --memory 2048 --vcpus 2 --os-variant rhel8.0 --import --disk /var/lib/lorax/composer/results/<GUID number for that image>/disk.qcow2*
- *virt-viewer will be fired*

Lab5 – Test output image

Inside virt-viewer VM

- Login as `myuser:mypassword`
- Check if you are member of groups **users** and **wheel** by command
 - *id*
- Can you sudo?
 - *sudo -i*
- Do you have your specified package installed already?
 - *rpm -qa | grep <package-name>*

More user configuration options

- `[[customizations.sshkey]]`
- `user = "root"`
- `key = "<public SSH key>"`
- `shell = "/usr/bin/bash"`
- `uid = 1001`
- `gid = 1001`

Documentation

- Chapter 28 : Building custom system images with composer
 - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/installation_guide/chap-composer-x86
- Chapter 6: Building custom system images with Composer
 - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/installing_and_deploying_rhel/building-custom-system-images-with-composer_graphical-installation