Red Hat
Training and
Certification

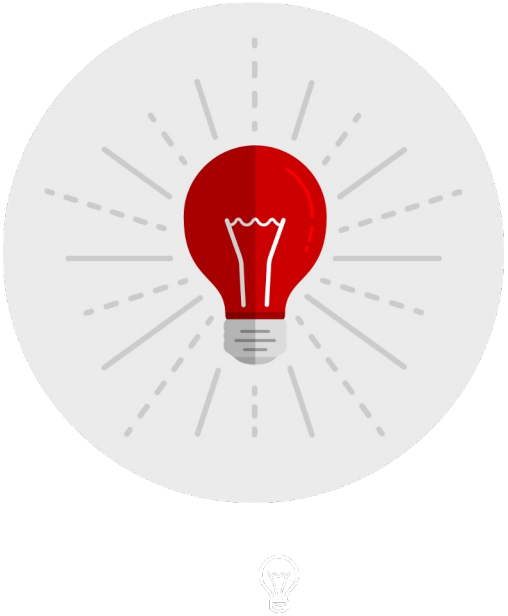
RED HAT
OPEN INNOVATION LABS

DevOps Culture and Practice Enablement

Containers, Registries, OpenShift (Kubernetes) Overview
Ansible Overview

Red Hat

Topics

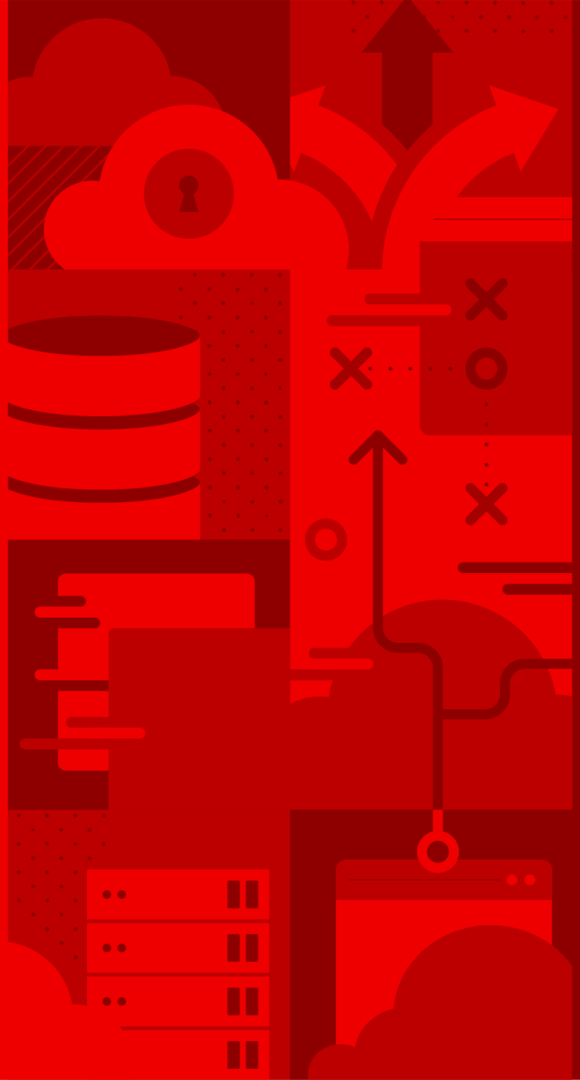


Containers
What are containers?

Registries (Image Registries)
What are container images and image registries?

OpenShift / Kubernetes
What exactly is OpenShift and Kubernetes?

Ansible
What exactly is Ansible?



Containers

Red Hat

Container

A container is basically a running, virtualized application in a self-contained package. Containers are based on a **container image** which consists of the entire runtime environment including:

- The application
- All libraries and dependencies
- Additional binaries needed by the application
- Configuration files needed to run the application

https://www.redhat.com/en/topics/containers

Red Hat

Containers

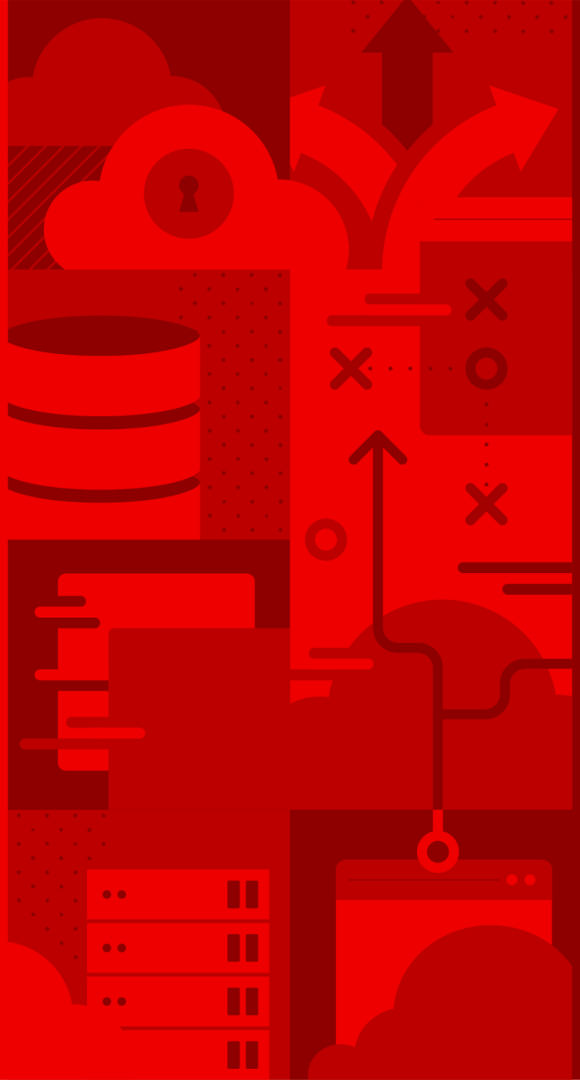
Julia Evans
@b0rk

containers aren't magic

These 15 lines of bash will start a container running the fish shell. Try it!
(download this script at bit.ly/containers-arent-magic)

```
wget bit.ly/fish-container -O fish.tar       # 1. download the image
mkdir container-root; cd container-root      #
tar -xf ../fish.tar                          # 2. unpack image into a directory
cgroup_id="cgroup_$(shuf -i 1000-2000 -n 1)" # 3. generate random cgroup name
cgcreate -g "cpu,cpuacct,memory:$cgroup_id"  # 4. make a cgroup &
cgset -r cpu.shares=512 "$cgroup_id"         #    set CPU/memory limits
cgset -r memory.limit_in_bytes=1000000000 \  #
      "$cgroup_id"                           #
cgexec -g "cpu,cpuacct,memory:$cgroup_id" \  # 5. use the cgroup
    unshare -fmuipn --mount-proc \           # 6. make + use some namespaces
    chroot "$PWD" \                          # 7. change root directory
    /bin/sh -c "                             #
        /bin/mount -t proc proc /proc &&     # 8. use the right /proc
        hostname container-fun-times &&      # 9. change the hostname
        /usr/bin/fish"                       # 10. finally, start fish!
```

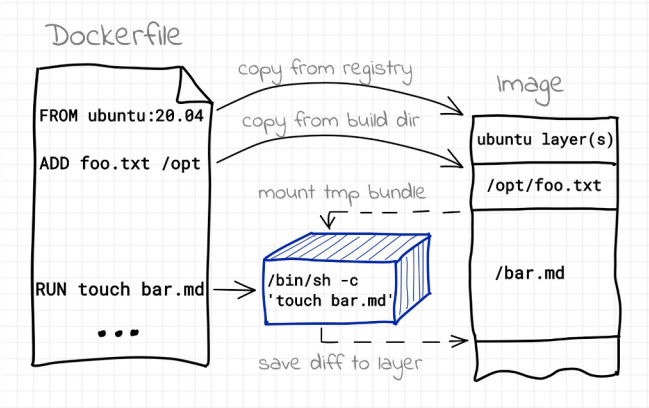
Red Hat

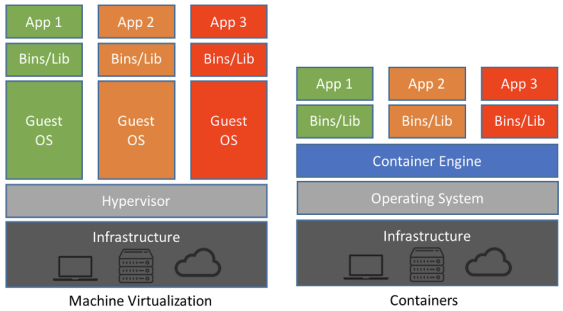


Registries

Container Image

A **container image**, in its simplest definition, is a file which is pulled down from a Registry Server and used locally as a mount point when starting Containers. Container images can be built based on instructions called Dockerfiles.

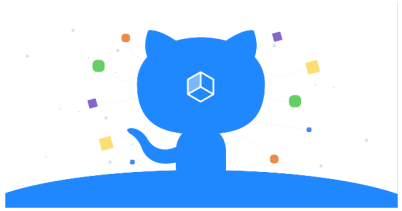




https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction

Red Hat

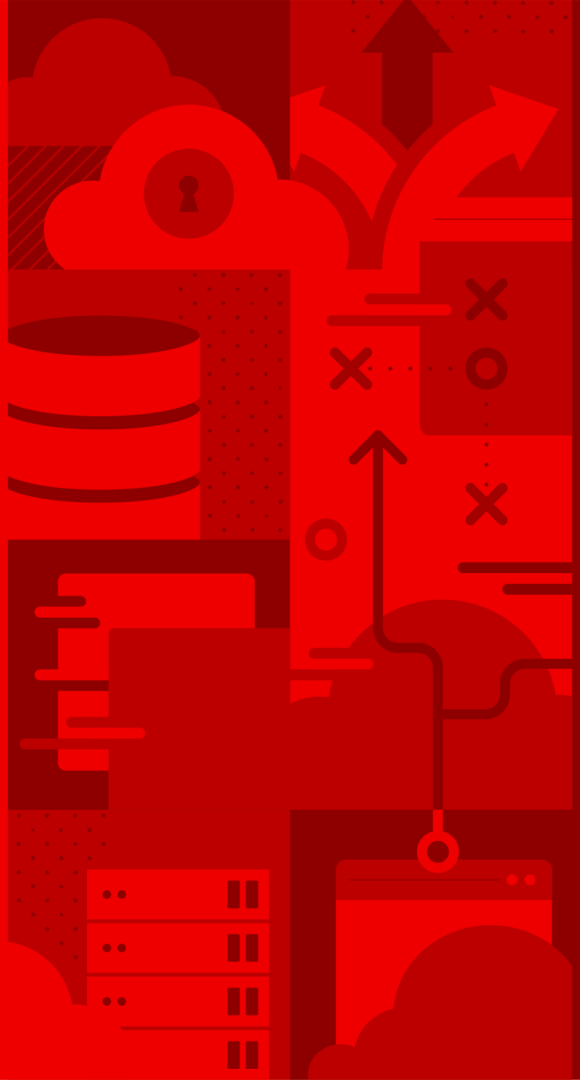
Container Image Registries



A container registry is **a repository, or collection of repositories, used to store container images for Kubernetes, DevOps, and container-based** application development.

https://www.redhat.com/en/topics/cloud-native-apps/what-is-a-container-registry#:~:text=A%20container%20registry%20is%20a,and%20container%2Dbased%20application%20development

Red Hat



OpenShift & Kubernetes

Red Hat

OpenShift and Kubernetes

kubernetes

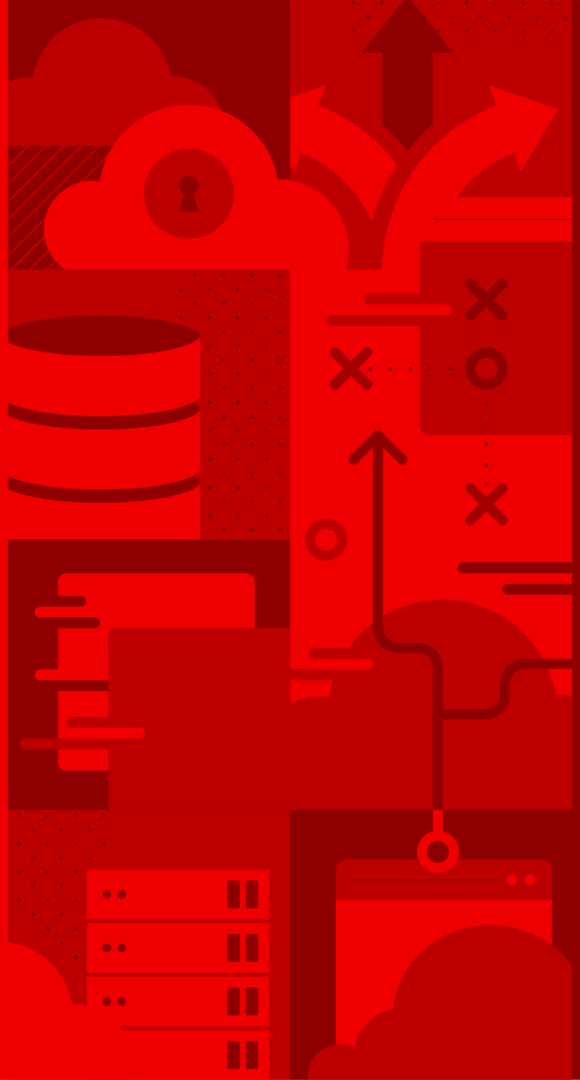
Kubernetes is an open-source container-orchestration system for automating computer application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF). Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation

https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Red Hat® OpenShift® is an enterprise-ready Kubernetes container platform with full-stack automated operations to manage hybrid cloud, multi-cloud, and edge deployments. Extends existing Kubernetes framework with:

- Routes
- Dashboards (Unified UI)
- Monitoring/Metrics/Logging
- Integrated Developer workflows supporting CI/CD pipelines and S2I

https://www.redhat.com/en/technologies/cloud-computing/openshift



Ansible

Ansible Introduction & Overview

Ansible – An automation language leveraging modules to be used in one or more tasks on managed systems. Most Ansible automation leverages and Ansible playbook which is a YAML formatted file containing Ansible directives.

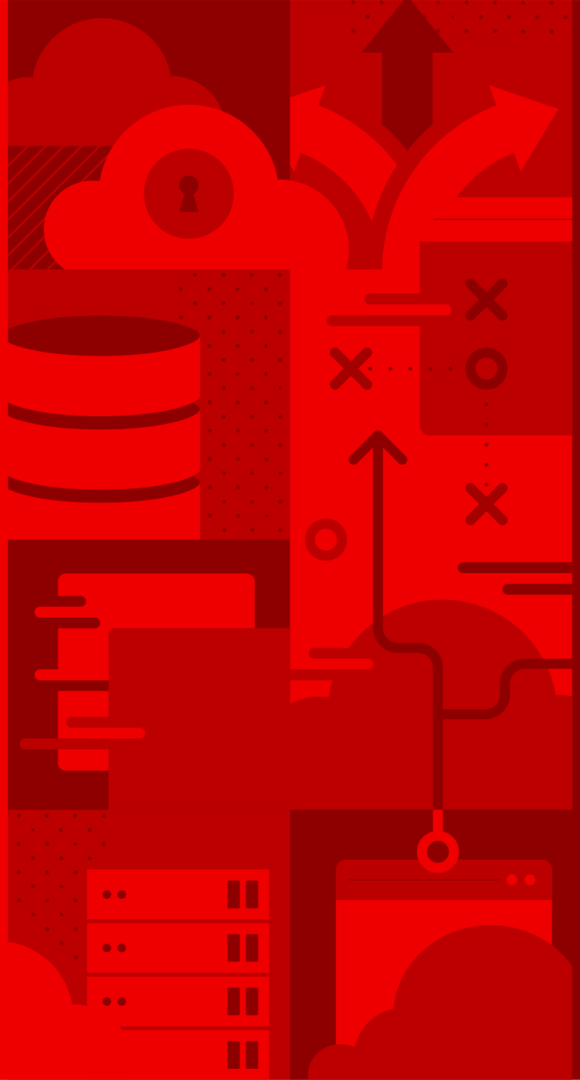
Ansible modules – Components used by Ansible tasks and playbooks which are generally implemented and developed in Python. Ansible modules work with certain system utilities and are optimized to be leveraged as a declarative automation language and provide idempotency.

Ansible ad-hoc commands – A way of executing a single Ansible task quickly that relies on a single Ansible module to perform the tests/changes of the task.

Ansible Roles– Curated, independent components of Ansible allowing re-use of common tasks. Roles can consist of variables, tasks, handlers, files, and templates.

Ansible Components & Commands

- **ansible.cfg** – Configuration file for running the **ansible** and **ansible-playbook** commands.
- **Inventory** – Inventory file used by the **ansible** and **ansible-playbook** commands identifying managed hosts/nodes and also contains Ansible inventory variables.
- **ansible** – Command used to perform/execute Ansible ad-hoc commands against a managed node.
- **ansible-playbook** – Command used to execute and run Ansible playbooks
- **ansible-galaxy** – Command to create or utilize Ansible roles. Many of these roles are published on http://galaxy.ansible.com
- **Playbook** – Collection of Ansible tasks organized into one or more Ansible plays.
- **Task** – Collection/list of Ansible modules arranged into instructions. Each task utilizes an Ansible module to perform a given action.
- **Ansible Module** – Specific module (small program generally implemented in Python) which perform the commands and executes the program to get the desired state of a given task.



Ansible Playbooks

Playbook Structure

```
---
- name: Ansiblize Managed Hosts
  hosts: localhost
  vars_files:
    - variables.yml

  tasks:
    - name: Create Ansible User
      debug:
          msg: This will use the USER  module

  handlers:
    - name: Restart SSHD
      debug:
          msg: This sill use the SERVICE or SYSTEMD
```


Sample Playbook

```yaml
---
- name: Register Hosts to Satellite
  hosts: rhel83_test
  vars_files:
    - sat_vars.yml
  vars:
    activation_key: rhhi-vm
    lifecycle_env: Library
    Organization_Name: MichetteTech
    Sat_User_Name: sat_user_name
    Sat_User_PW: sat_password
    SatURL: https://sat6.michettetech.com

  tasks:
    - name: Prepare System for Satellite Registration - Get Cert
      get_url:
        url: "{{ SatURL }}/pub/katello-ca-consumer-latest.noarch.rpm"
        dest: /tmp/katello-latest.rpm
        validate_certs: no

    - name: Install Katello rpm
      yum:
        name: /tmp/katello-latest.rpm
        state: latest
        disable_gpg_check: yes
```

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

https://www.facebook.com/RedHatServices

https://twitter.com/RedHat_Services

E-Mail: tmichett@redhat.com

Red Hat