System Containers

Concept, Creation and Usage

Jason Brooks
Project Atomic

Yu Qi (Jerry) Zhang
Project Atomic

github.com/yuqi-zhang/flock

Today's Topics

1. Concept
   a. Background on the Atomic Host
   b. Why System Containers
   c. How System Containers Work
2. Usage
   a. Running a System Container
   b. Updates/Rollbacks and Other Features
   c. Use-Cases and Existing Containers
3. Creation
   a. How to configure necessary files
   b. Build a hello-world system container

Concept

Atomic Host

- A lightweight, immutable platform for running container applications
- Optimized for Kubernetes and Openshift
- Aggregated software units: tested and shipped as a whole
- What if you want to de-couple host and services?
- What if you want to add new services?
- What if you want a smaller base?

System Containers

- Systemd services as runc containers
- Uses:
    - Atomic CLI to manage
    - OSTree to store
    - Skopeo to push/pull
    - Systemd for lifecycle management
- Read-only and host-specific

Why SysContainers

- Can run pre-docker/cri-o services as if they were traditional binary-on-fs services
- Does not require a running container engine
- Can utilize the existing atomic host and ostree
- Easily switch versions
- Provides the usual benefits of bundling and isolation for a consistent experience

What’s inside

- Follows oci format
- Services and commands in containers
- Image layers are stored as ostree branches
- A hardlinked checkout is created during install
- Image is read-only

What’s inside

- **config.json.template**:
  - template OCI config for runc
- **manifest.json**:
  - default values for configuration variables
- **service.template**:
  - unit file for systemd
- **tmpfiles.template**:
  - config file for systemd-templates

Comparison to Docker

- Similarities
  - Follows oci format
  - Concept of layers
  - Uses runc as the container runtime
  - Non-conflicting
- Differences
  - Systemd as lifecycle management
  - Generates specific files on the host
  - Pre-defines mounts in config
  - Does not require a running daemon

Usage

Running a Container

- Pulling an image
- Installing the image
- Starting the service
- Checking status
- Stopping the service
- Uninstalling the Container

Other Functionality

- Image/container commands
- Installation options
- Updating a container
- Rolling back a container
- Running a command in a container

Example: etcd/flannel

- demo/workshop

Example: docker/cri-o

- demo/workshop

Creation

Files

- Checkout location:

  /var/lib/containers/atomic/${NAME}.0

  - The filesystem: ../rootfs
  - Template and config files ../*.json/conf/service
- Using mounts and exports/hostfs
- Systemd tmpfiles

Building

- As a Docker image
- Using system-buildah

Example: hello-world

- Demo with a hello-world image

Kube/Origin

Openshift-Ansible

- Can use system containers for origin!
  - openshift_use_openvswitch_system_container=True
  - openshift_use_node_system_container=True
  - openshift_use_master_system_container=True
  - openshift_use_etcd_system_container=True
  - system_images_registry=”docker.io”
- Can also use system containerized docker
  - openshift_docker_use_system_container=True
  - Uses /etc/docker/container-docker.json for config
  - Service is “container-engine”

What are your questions?

Find us on #atomic and
atomic-devel@projectatomic.io