Atomic, with and without Atomic

Jan Pazdziora
Sr. Principal Software Engineer
Identity Management Special Projects, Red Hat
jpazdziora@redhat.com

6th February 2016

Applications vs. system services

Applications	System services
Provide end-user or externally-facing functionality	Configure system Support other programs
Often network-enabled services	Targetting local system
Multiple instances possible	Single instance on any system
Data and interfaces isolated	Well-known paths for sharing with other services
Can run on any host	Bound to a particular host
For example:	
DNS server for a domain	Caching nameserver

System daemons and services

- System services enhance the behaviour of the base operating system:
  - network configuration
  - system time sync
  - logging
  - caching DNS service
  - authentication
  - ...

Processes on minimal system

```
  1 ?     Ss    0:02 /usr/lib/systemd/systemd --switched-root --system --deserial
597 ?     Ss    0:00 /usr/lib/systemd/systemd-journald
616 ?     Ss    0:00 /usr/sbin/lvmetad -f
637 ?     Ss    0:00 /usr/lib/systemd/systemd-udevd
754 ?     S<Lsl 0:00 /usr/sbin/dmeventd -f
758 ?     S<sl  0:00 /sbin/auditd -n
788 ?     Ss    0:00 /usr/lib/systemd/systemd-logind
789 ?     Ssl   0:00 /usr/sbin/NetworkManager --no-daemon
816 ?     S     0:00  \_ /sbin/dhclient -d -q -sf /usr/libexec/nm-dhcp-helper -pf
794 ?     Ssl   0:00 /usr/sbin/gssproxy -D
801 ?     Ss    0:00 /usr/sbin/rpc.gssd
892 ?     Ss    0:00 /usr/sbin/sshd -D
896 ?     Ss    0:00 /usr/sbin/crond -n
902 tty1  Ss+   0:00 /sbin/agetty --noclear tty1 linux
903 ttyS0 Ss+   0:00 /sbin/agetty --keep-baud 115200 38400 9600 ttyS0 vt22
```

Typical workflow

- Install package(s)
- Edit config or run some setup tool
- Enable service (to be run automatically after reboot)
- Start the service
- On Fedora:

```
# dnf install -y chrony
# vi /etc/chrony.conf
# systemctl enable chronyd
# systemctl start chronyd
```

```
# dnf install -y freeipa-client
# ipa-client-install [params]     # will enable and start sssd service
# systemctl status sssd
```

- Configuration management systems might be used; underneath, they typically do something like shown above.

SSSD

- System Security Services Daemon.
- External user identities, authentication, authorization.
  - Users from multiple domains (FreeIPA, Active Directory, generic LDAP servers).
  - Host-based access control, group policy object enforcement.
  - sudo rules, SELinux contexts for users, ...
  - Failover, caching for speed or off-line use.
- The `ipa-client-install` for IPA-enrollment of machines.
  - Host record created in IPA server.
  - Configures SSSD, fetches `krb5.keytab`, tweaks PAM, ...

Atomic platform

- Minimal, one-size-fits-all system.
- Targeted at applications running in docker containers.
- Read-only (except `/etc`, `/var`), booted from images (ostree).
  - Images are built from standard packages (rpms).
  - Packages only installed in image build-time.
- Upgrade done by booting new image.
- Packages (rpms) can be seen on the system.
- But no `dnf`. And system is read-only anyway.

System services on Atomic

- How does admin add system services?
  - Build and maintain derived ostree images?
    - Not likely.
  - There is docker daemon running on Atomic.
    - It is meant for running application containers.
    - We can try to run system services with it as well.

System service in a container

- Dockerfile:

```
FROM fedora:23
RUN dnf install -y freeipa-client ... && dnf clean all
```

- Build:

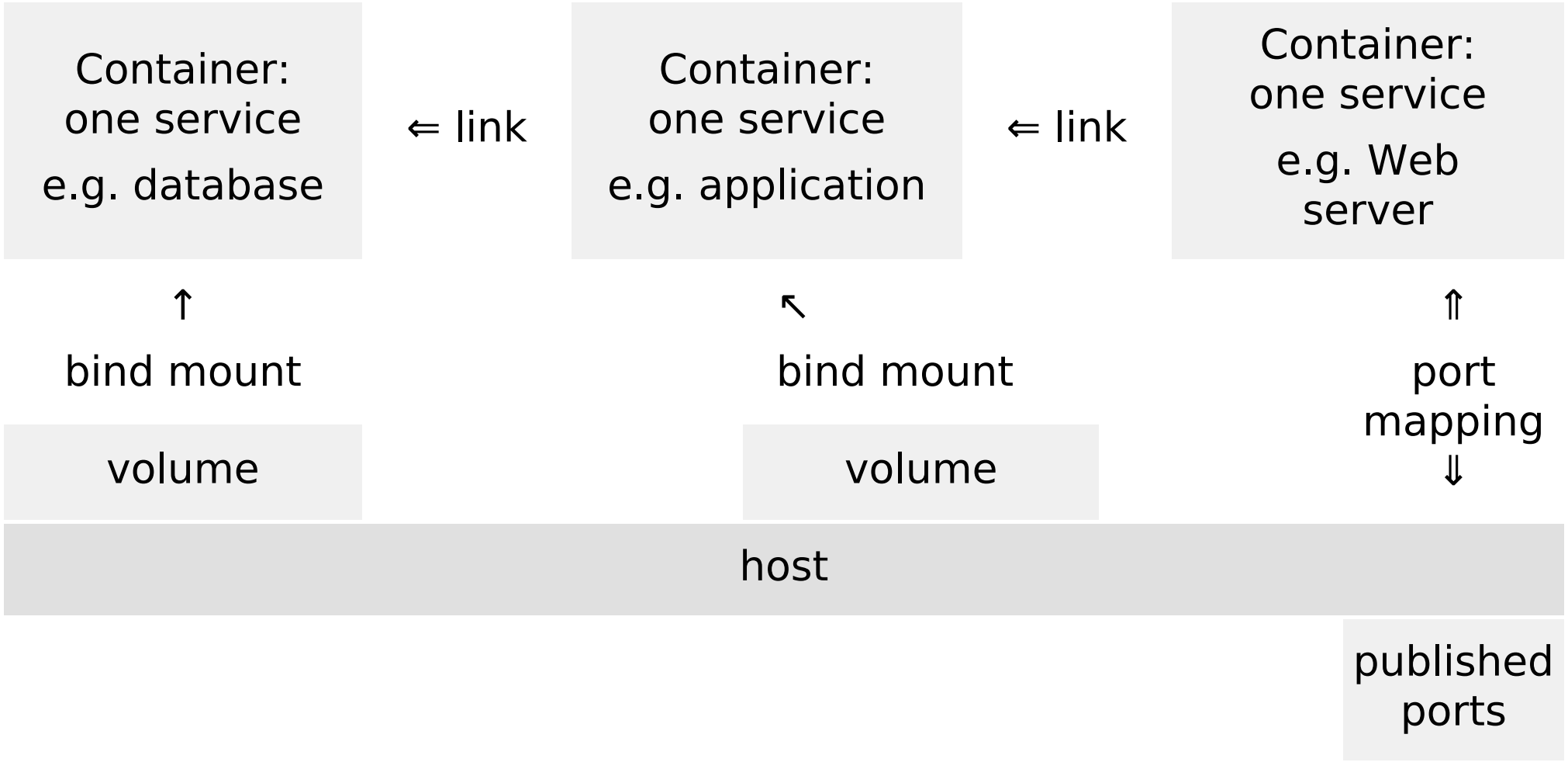
```
# docker build -t sssd .
```

- Run container:

```
# docker run sssd ...
```

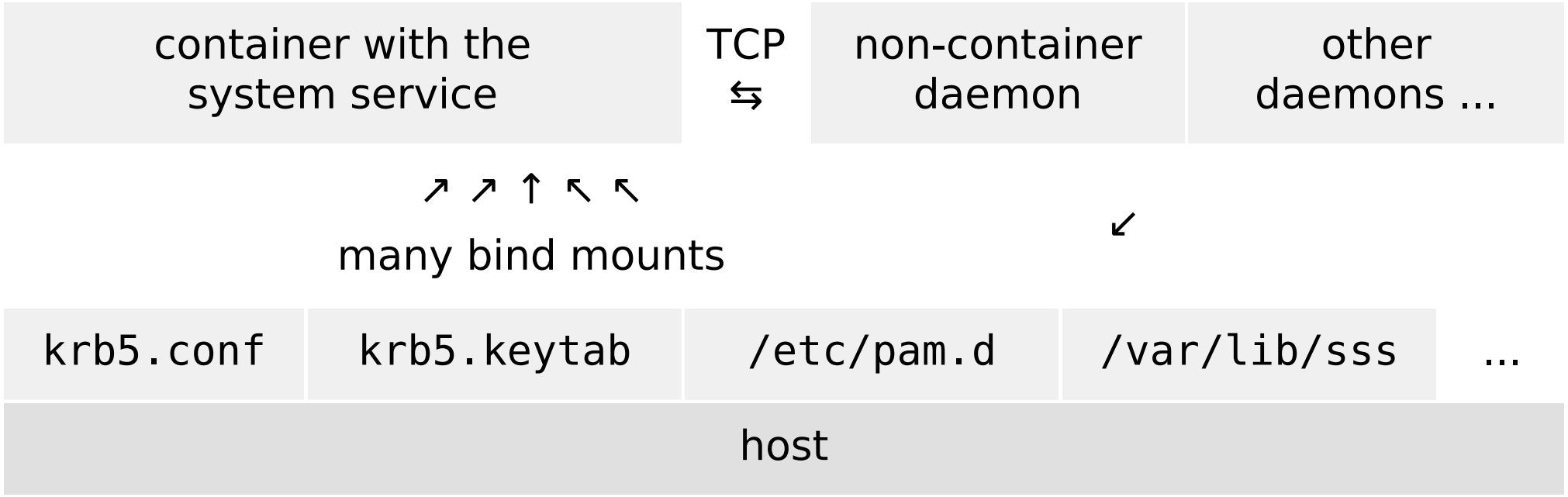
- What do we run in the container? `/usr/sbin/sssd`?
- We need to IPA-enroll the machine first (run `ipa-client-install`)
  - Can't run it in build-time (because it establishes host identity).
  - And it's not installed on the host.
- Where is configuration stored?

Typical containerized application



- One or few bind mounts per container: `docker run -v $DATA:/data`

Setup needed for system services



- Dozen of bind mounts

SSSD container mount points

```
# docker run ...  -v /etc/ipa/:/etc/ipa/:ro \
  -v /etc/krb5.conf:/etc/krb5.conf:ro \
  -v /etc/krb5.keytab:/etc/krb5.keytab:ro \
  -v /etc/nsswitch.conf:/etc/nsswitch.conf:ro \
  -v /etc/openldap/:/etc/openldap/:ro \
  -v /etc/pam.d/:/etc/pam.d/:ro \
  -v /etc/passwd:/etc/passwd.host:ro \
  -v /etc/pki/nssdb/:/etc/pki/nssdb/:ro \
  -v /etc/ssh/:/etc/ssh/:ro \
  -v /etc/sssd/:/etc/sssd/:ro \
  -v /etc/systemd/system/sssd.service.d:/etc/systemd/system/sssd.service.d:ro \
  [...]
  -v /etc/sysconfig/sssd:/etc/sysconfig/sssd:ro \
  -v /var/cache/realmd/:/var/cache/realmd/ \
  -v /var/run/dbus/system_bus_socket:/var/run/dbus/system_bus_socket \
  -v /var/lib/authconfig/last/:/var/lib/authconfig/last/:ro \
  -v /var/lib/ipa-client/sysrestore/:/var/lib/ipa-client/sysrestore/:ro \
  -v /var/lib/samba/:/var/lib/samba/ \
  -v /var/lib/sss/:/var/lib/sss/ \
  -v /var/log/sssd/:/var/log/sssd/ sssd ...
```

Defining the run parameters

- Labels in Dockerfile:

```
LABEL RUN 'docker run -d --privileged --net=host \
  --name ${NAME} -e NAME=${NAME} -e IMAGE=${IMAGE} \
  -v /etc/ipa/:/etc/ipa/:ro \
  -v /etc/krb5.conf:/etc/krb5.conf:ro \
  -v /etc/krb5.keytab:/etc/krb5.keytab:ro \
  -v /etc/nsswitch.conf:/etc/nsswitch.conf:ro \
  -v /etc/openldap/:/etc/openldap/:ro \
  [...]
  ${IMAGE} /bin/run.sh'
```

- Tool that can use the labels: `/usr/bin/atomic`.

```
# atomic run fedora/sssd
docker run -d --net=host --name sssd -e NAME=sssd -e IMAGE=sssd -v [...]
2311d245efffa5292c9c2ca141161528cb582264d2a01923ba9bd30503831f1f
#
```

The setup phase

- Many of the bind-mounted files don't exist on typical host.
- Label for container installation (initial deployment):

```
LABEL INSTALL 'docker run --rm=true --privileged --net=host \
  -v /:/host -e NAME=${NAME} -e IMAGE=${IMAGE} -e HOST=/host \
  ${IMAGE} /bin/install.sh'
```

```
# atomic install fedora/sssd
docker run --rm=true --privileged --net=host -v /:/host -e NAME=sssd [...]
```

- Any additional arguments are appended.
- Could we pass parameters to `ipa-client-install`?

```
# atomic install fedora/sssd --password one-time-secret
docker run --rm=true [...] sssd /bin/install.sh --password one-time-secret
```

IPA-enrollment with container

- The `atomic` tool nicely separates initial setup from runs.
- In `INSTALL`, `-v /:/host` gives access to the full host filesystem.
- In `install.sh` we copy in existing configuration from host, run service setup tool (`ipa-client-install`), and copy result back out to host:

```
HOST=${HOST:-/host}
( cd "$HOST" && while read f ; do
    if [ -e "$f" ] ; then cp --parents -rp -t / "$f" ; fi
done ) < /etc/host-data-list
[...]
ipa-client-install -U "$@"
[...]
xargs cp --parents -rp -t "$HOST" < /etc/host-data-list
chroot "$HOST" restorecon -ri -f - < /etc/host-data-list
```

- Realmd also supported, for example for joining host to Active Directory:

```
# atomic install fedora/sssd realm join [...]
```

Containerized systemd service

- On Atomic, the `/usr/lib/systemd/system` is read-only.
- But `/etc/systemd/system` is free to be used:

  sssd.service.template:

```
[Service]
ExecStart=/usr/bin/atomic run --name=${NAME} ${IMAGE}
ExecStop=/usr/bin/atomic stop --name=${NAME} ${IMAGE}
Type=oneshot
RemainAfterExit=yes
```

  install.sh:

```
sed "s%\${IMAGE}%${IMAGE:-sssd}%g; s%\${NAME}%${NAME:-sssd}%g;" \
  /etc/sssd.service.template > $HOST/etc/systemd/system/$NAME.service
chroot $HOST systemctl daemon-reload
```

- After `atomic install fedora/sssd`, `systemctl ... sssd` work.
- Containerized system services behave like normal services.
- Some (client) shared libraries are needed in ostree image.

Issues

- `--privileged` and SELinux rules for interoperability with `spc_t` needed.
  - Support for `--security-opt label:type:sssd_t` would be nice.
- When the docker daemon goes down, containers go down.
  - `--restart=always`? `--restart=unless-stopped`?
  - ```
    [Install]
    WantedBy=docker.service
    ```
- Command-line tools for the containerized services cannot be deployed.
  - `/usr/local/bin` is read-only.
  - Could heavy use of `/etc/alternatives` help?
- Do we want to split the INSTALL-time and RUN-time containers?
    ```

Working with `atomic`

- It can manage the ostree on the Atomic host.
- But it also understands LABELs in docker images.
  - Directives to run the container come with the container image.
- This feature is not dependent on Atomic platform.
- Atomic being read-only is a bit hard to develop on.
- Containerized services can be developed on normal operating systems.
  - And run there as well.
  - With configuration (and data) on host in standard locations, it's possible to switch between native and containerized services.

```
# atomic install fedora/sssd --migrate
```

- Hence: `atomic`, not just on Atomic.

Conclusion

- Software for system services can come in container images.
- The `atomic` tool understands LABELs in docker images.
- It distinguishes INSTALL and RUN invocations.
- It can be used both on Atomic host, and on normal operating system.
- In INSTALL, we can create systemd service on the host.
- Multiple setup modes are possible with parameters to INSTALL phase.
  - Including migration of existing configuration.

- With SSSD, Kerberos authentication, access control by external users, and centralized sudo rule management is now possible on Atomic.

References & feedback

- www.projectatomic.io
- github.com/fedora-cloud/Fedora-Dockerfiles/tree/master/sssd
- hub.docker.com/r/fedora/sssd/
- adelton.com/docs/docker/fedora-atomic-sssd-container
- adelton.com/docs/docker/complex-application-in-container

- www.devconf.cz/feedback/281
- jpazdziora@redhat.com