# Containers without a Container Manager, with systemd Linux Piter 2017, St. Petersburg, Russia

November 2017

Containers without a Container Manager, with systemd

(日) (日) (

Containers without a Container Manager, with systemd

◆□ > ◆□ > ◆三 > ◆三 > ○ = ○ ○ ○ ○

... Resource Bundling,

Containers without a Container Manager, with systemd

《曰》 《聞》 《臣》 《臣》

æ

### ... Resource Bundling, Sandboxing,

・ロン ・四と ・ヨン ・ヨン Containers without a Container Manager, with systemd

æ

### ... Resource Bundling, Sandboxing, Delivery

・ロト ・ 日 ・ ・ ヨ ・ ・ ヨ ・ Containers without a Container Manager, with systemd

臣

# Containers? ... Resource Bundling, Sandboxing, Delivery Let's focus on the first two!

▲□ ▶ ▲ □ ▶ ▲ □ ▶

Resource Bundling

Containers without a Container Manager, with systemd

◆□ ▶ ◆圖 ▶ ◆ 臣 ▶ ◆ 臣 ▶ ○ 臣 ○

#### RootDirectory=

Containers without a Container Manager, with systemd

<ロ> (四) (四) (三) (三) (三) (三)

RootDirectory= RootImage=

> ・ロ・ ・ 日・ ・ 田・ ・ 日・ Containers without a Container Manager, with systemd

æ

Images for RootImage=?

・ロト ・日下・ ・日下 Containers without a Container Manager, with systemd

< 三→

æ

#### Images for RootImage=?

Discoverable GPT,

・ロト ・ 日 ・ ・ ヨ ・ ・ ヨ ・ Containers without a Container Manager, with systemd

臣

Images for RootImage=? Discoverable GPT, unambiguous GPT or MBR,

• • = •

э

Images for RootImage=?

Discoverable GPT, unambiguous GPT or MBR, raw file system

• • 3 • •

э

Images for RootImage=? Discoverable GPT, unambiguous GPT or MBR, raw file system For example, create it with mkosi

・ 同 ト ・ ヨ ト ・ ヨ ト

Images for RootImage=? Discoverable GPT, unambiguous GPT or MBR, raw file system For example, create it with mkosi dm-verity, LUKS, ...

・ 同 ト ・ ヨ ト ・ ヨ ト

#### RootImage=/RootDirectory=

▲掃♪ ▲ 置♪ Containers without a Container Manager, with systemd

< ∃ >

臣

RootImage=/RootDirectory= Fancy chroot()

Containers without a Container Manager, with systemd

< A

• • = •

RootImage=/RootDirectory= Fancy chroot() Inherits the same problems

Containers without a Container Manager, with systemd

< 同 > < 三 > < 三 >

#### MountAPIVFS=

Containers without a Container Manager, with systemd

◆□ > ◆□ > ◆三 > ◆三 > 三 の へ ⊙

How to share data?

Containers without a Container Manager, with systemd

◆□ ▶ ◆圖 ▶ ◆ 臣 ▶ ◆ 臣 ▶ ○ 臣 ○

How to share data? BindPaths=, BindReadOnlyPaths=

・ロト ・回ト ・ヨト ・ヨト

æ

How to share data? BindPaths=, BindReadOnlyPaths= RuntimeDirectory=, StateDirectory=, CacheDirectory=, LogsDirectory, ConfigurationDirectory=

< 同 > < 三 > < 三 >

How to share data? BindPaths=, BindReadOnlyPaths= RuntimeDirectory=, StateDirectory=, CacheDirectory=, LogsDirectory, ConfigurationDirectory=

Also nice to keep bundles self-contained, no need for tmpfiles.d/

・ 同 ト ・ ヨ ト ・ ヨ ト

#### Containers without a Container Manager, with systemd

◆□ > ◆□ > ◆三 > ◆三 > 三 の へ ⊙

Not at all: use PrivateUsers=, to disconnect the tables



臣

Not at all: use PrivateUsers=, to disconnect the tables

nss-systemd synthesizes user entries for both root and nobody, the user IDs always needed, and always defined the same way.

・ 同 ト ・ ヨ ト ・ ヨ ト

Not at all: use PrivateUsers=, to disconnect the tables

nss-systemd synthesizes user entries for both root and nobody, the user IDs always needed, and always defined the same way.

Missing: how to make sure the host and the bundle environment can share the service's own user identity?

(日) (日) (日)

#### Sandboxing

Containers without a Container Manager, with systemd

#### Established way to sandbox UNIX services:

・ロト ・回ト ・ヨト ・ヨト Containers without a Container Manager, with systemd

æ

#### Established way to sandbox UNIX services: UNIX users

・ロト ・回ト ・ヨト ・ヨト Containers without a Container Manager, with systemd

臣

## Established way to sandbox UNIX services: UNIX users DynamicUser=



臣

# Established way to sandbox UNIX services: UNIX users DynamicUser=

Also nice to keep bundles self-contained, no need for sysusers.d/

Containers without a Container Manager, with systemd

< 同 > < 三 > < 三 >

# Established way to sandbox UNIX services: UNIX users DynamicUser=

Also nice to keep bundles self-contained, no need for sysusers.d/ No artifacts

Containers without a Container Manager, with systemd

<日本<br />
<</p>

#### RemoveIPC=

Containers without a Container Manager, with systemd

◆□ > ◆□ > ◆三 > ◆三 > 三 の へ ⊙

RemoveIPC= PrivateTmp=

Containers without a Container Manager, with systemd

RemoveIPC= PrivateTmp= No artifacts

Containers without a Container Manager, with systemd

PrivateDevices=

Containers without a Container Manager, with systemd

◆□ > ◆□ > ◆三 > ◆三 > 三 の へ ⊙

PrivateDevices= PrivateNetwork=

PrivateDevices= PrivateNetwork= IPAddressAllow=/IPAddressDeny=

> ・ロト ・ 日 ・ ・ 日 ・ ・ 日 ・ Containers without a Container Manager, with systemd

æ

ProtectKernelTunables=

Containers without a Container Manager, with systemd

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

ProtectKernelTunables= ProtectKernelModules=

イロト イヨト イヨト イヨト

3

ProtectKernelTunables=
ProtectKernelModules=
ProtectControlGroups=

・ロト ・回ト ・ヨト ・ヨト

æ

## SystemCallFilter=

Containers without a Container Manager, with systemd

<ロ> (四) (四) (三) (三) (三) (三)

SystemCallFilter= Now with system call groups!

Containers without a Container Manager, with systemd

▲圖 → ▲ 国 → ▲ 国 →

臣

RestrictAddressFamilies=

## Containers without a Container Manager, with systemd

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

RestrictAddressFamilies=

SystemCallArchitectures=



・ロト ・聞ト ・ヨト ・ヨト

3

RestrictAddressFamilies= SystemCallArchitectures= RestrictNamespaces=

▲圖→ ▲ 国→ ▲ 国→

臣

RestrictAddressFamilies= SystemCallArchitectures= RestrictNamespaces= LockPersonality=

・ 回 ト ・ ヨ ト ・ ヨ ト

æ

RestrictAddressFamilies= SystemCallArchitectures= RestrictNamespaces= LockPersonality= MemoryDenyWriteExecute=

・ 回 ト ・ ヨ ト ・ ヨ ト

э

RestrictAddressFamilies= SystemCallArchitectures= RestrictNamespaces= LockPersonality= MemoryDenyWriteExecute= RestrictRealtime=

・ 同 ト ・ ヨ ト ・ ヨ ト

э

RestrictAddressFamilies= SystemCallArchitectures= RestrictNamespaces= LockPersonality= MemoryDenyWriteExecute= RestrictRealtime= KeyringMode=

3

Containers without a Container Manager, with systemd

▲□ ▶ ▲ □ ▶ ▲ □ ▶

Iterate through bundle images/dirs in /var/lib/portables, extract relevant unit files, make them available on the host

Iterate through bundle images/dirs in /var/lib/portables, extract relevant unit files, make them available on the host

Extend them with RootImage=/RootDirectory=, plus sandboxing options, via .service.d/ drop-ins

高 ト イ ヨ ト イ ヨ ト

Iterate through bundle images/dirs in /var/lib/portables, extract relevant unit files, make them available on the host

Extend them with RootImage=/RootDirectory=, plus sandboxing options, via .service.d/ drop-ins

Result: bundled images, containing service code, made available locally easily like native services

・ 戸 ト ・ ヨ ト ・ ヨ ト

Missing: pidns? hidepid?

Containers without a Container Manager, with systemd

・日本 ・モト ・モト

臣

Missing: pidns? hidepid? systemctl purge?

< A

• • = •

• 3 >

Open for everybody

・ロト ・ 日 ・ ・ ヨ ・ ・ ヨ ・ Containers without a Container Manager, with systemd

æ

## Open for everybody Use externally managed network namespace

Containers without a Container Manager, with systemd

< A

Image: A image: A

э

Open for everybody Use externally managed network namespace Use externally managed eBPF programs Open for everybody Use externally managed network namespace Use externally managed eBPF programs Use externally managed console TTY

• • = •

That's all, folks!

・ロ・ ・聞・ ・ヨ・ ・ヨ・ Containers without a Container Manager, with systemd

Ξ.