Atomic Rollbacks

Automatic rollbacks

See greenboot for information on automatic rollbacks and how to integrate without your bootloader.

Manual rollbacks

To manually rollback, an older entry can be selected via GRUB bootloader or in the case of an Android bootloader, a slot switch may be triggererd using an AB switching tool. This may be useful for testing purposes.

Rollbacks

```
                        +------------------------------------------------+
+------------------+    |                                                |
|                  |    |                                                |
|                  |    |                                                |
|                  |    |      (ostree:0) latest     (multi-user.target) |
|                  |    |                                                |
| Bootloader       |--->+ root                                           |
|                  |    |                                                |
|                  |    |      (ostree:1) latest - 1 (multi-user.target) |
|                  |    |                                                |
|                  |    |                                                |
+------------------+    |                                                |
                        +------------------------------------------------+
```

Bootloaders have multiple boot entries to choose from after upgrade. On rollback, the bootloader will boot the “latest - 1” version, rather than the latest version of the OS.

Alternate rollback techniques

Below is an alternate technique to traditional AB switching that can be used. On rollback, an alternative boot target is used, rather than booting as default.target.

```
                        +------------------------------------------------+
+------------------+    |                                                |
|                  |    |                                                |
|                  |    |                                                |
|                  |    |      (ostree:0) latest     (multi-user.target) |
|                  |    |                                                |
| Bootloader       |--->+ root                                           |
|                  |    |                                                |
|                  |    |      (ostree:1) latest - 1 (rescue.target)     |
|                  |    |                                                |
|                  |    |                                                |
+------------------+    |                                                |
                        +------------------------------------------------+
```

In this case, instead of rolling back to an older version, we also boot into an alternate systemd boot target. Here we will describe how you can put togther an alternate systemd boot target, using the built-in rescue.target as an example.

Below is a rescue.service file, it essentially executes systemd-sulogin-shell rescue when this service is activated.

rescue.service:

```
#  SPDX-License-Identifier: LGPL-2.1-or-later
#
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it
#  under the terms of the GNU Lesser General Public License as published by
#  the Free Software Foundation; either version 2.1 of the License, or
#  (at your option) any later version.

[Unit]
```

```
Description=Rescue Shell
Documentation=man:sulogin(8)
DefaultDependencies=no
Conflicts=shutdown.target
After=sysinit.target plymouth-start.service
Before=shutdown.target

[Service]
Environment=HOME=/root
WorkingDirectory=-/root
ExecStartPre=-/usr/bin/plymouth --wait quit
ExecStart=-/usr/lib/systemd/systemd-sulogin-shell rescue
Type=idle
StandardInput=tty-force
StandardOutput=inherit
StandardError=inherit
KillMode=process
IgnoreSIGPIPE=no
SendSIGHUP=yes
```

Below is a rescue.target file, it is reached once rescue.service is complete.

rescue.target:

```
#  SPDX-License-Identifier: LGPL-2.1-or-later
#
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it
#  under the terms of the GNU Lesser General Public License as published by
#  the Free Software Foundation; either version 2.1 of the License, or
#  (at your option) any later version.

[Unit]
Description=Rescue Mode
Documentation=man:systemd.special(7)
Requires=sysinit.target rescue.service
After=sysinit.target rescue.service
AllowIsolate=yes
```

This is a simple bash script, it checks whether `ostree admin status -D` is `not-default` and if it is, it notifies systemd to alternatively boot into rescue.target.

In the happy path, when we have booted the latest version `ostree admin status -D` would output `default`.

ostree-rollback-to-rescue:

```
#!/usr/bin/bash

set -ex

if [ "$(ostree admin status -D)" = "not-default" ]; then
  exec systemctl --no-block isolate rescue.target
fi
```

This is a systemd service file that runs ostree-rollback-to-rescue early in the boot sequence, it is essential that this service is run early to ensure we don't execute a full boot sequence, hence options `DefaultDependencies=no` and `Before=` are used.

ostree-rollback-to-rescue.service

```
[Unit]
Description=OSTree rollback to rescue
DefaultDependencies=no
OnFailure=emergency.target
OnFailureJobMode=replace-irreversibly
After=initrd-root-fs.target initrd-fs.target initrd.target boot.mount
Before=cryptsetup.target integritysetup.target remote-fs.target slices.target swap.target veri

[Service]
Type=oneshot
ExecStart=/usr/sbin/ostree-rollback-to-rescue

[Install]
WantedBy=sysinit.target
```