Flatcar Container Linux

flatcar.org

Saturday 16:30 - HS3, "System Administration" track

"Day 2 Ops"

Linux for Kubernetes and Container Workloads

Sunday 15:15 – HS5, "Cloud" track

Flatcar Linux: what's new in this Container OS?

github.com/flatcar/Flatcar

→ README.md

Flatcar

Designed for

Containers

Infra automation

Security

with declarative provisioning

Flatcar Container Linux

- Image-based OS with A/B updates for read-only /usr partition
- Declarative provisioning with Ignition on first boot to configure the instance from the initrd (systemd units, files, SSH keys, disks...)
- Vendor neutral and community oriented

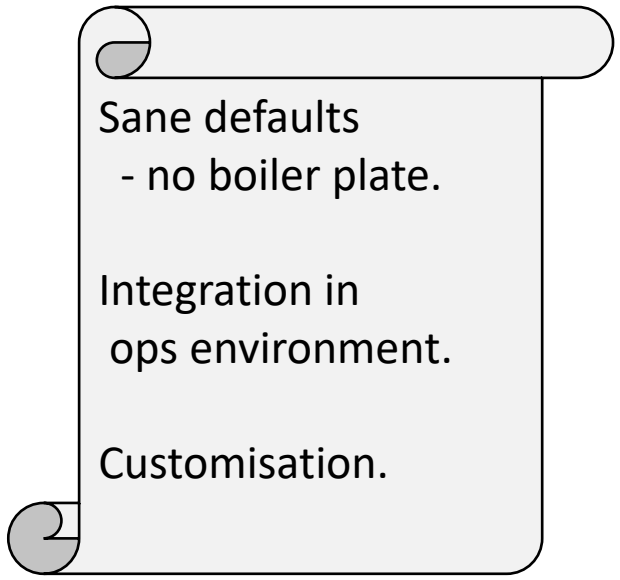
Flatcar Container Linux

Documentation:

https://flatcar.org/docs/

Flatcar automates Node deployments

Configure -> Deploy -> Operate



Butane-config.yaml

```
Custom data,
User data,
Http, or
[i]PXE
```

ignition.json

Automated
- Self-configuration
- Unattended updates

Declarative configuration, before provisioning

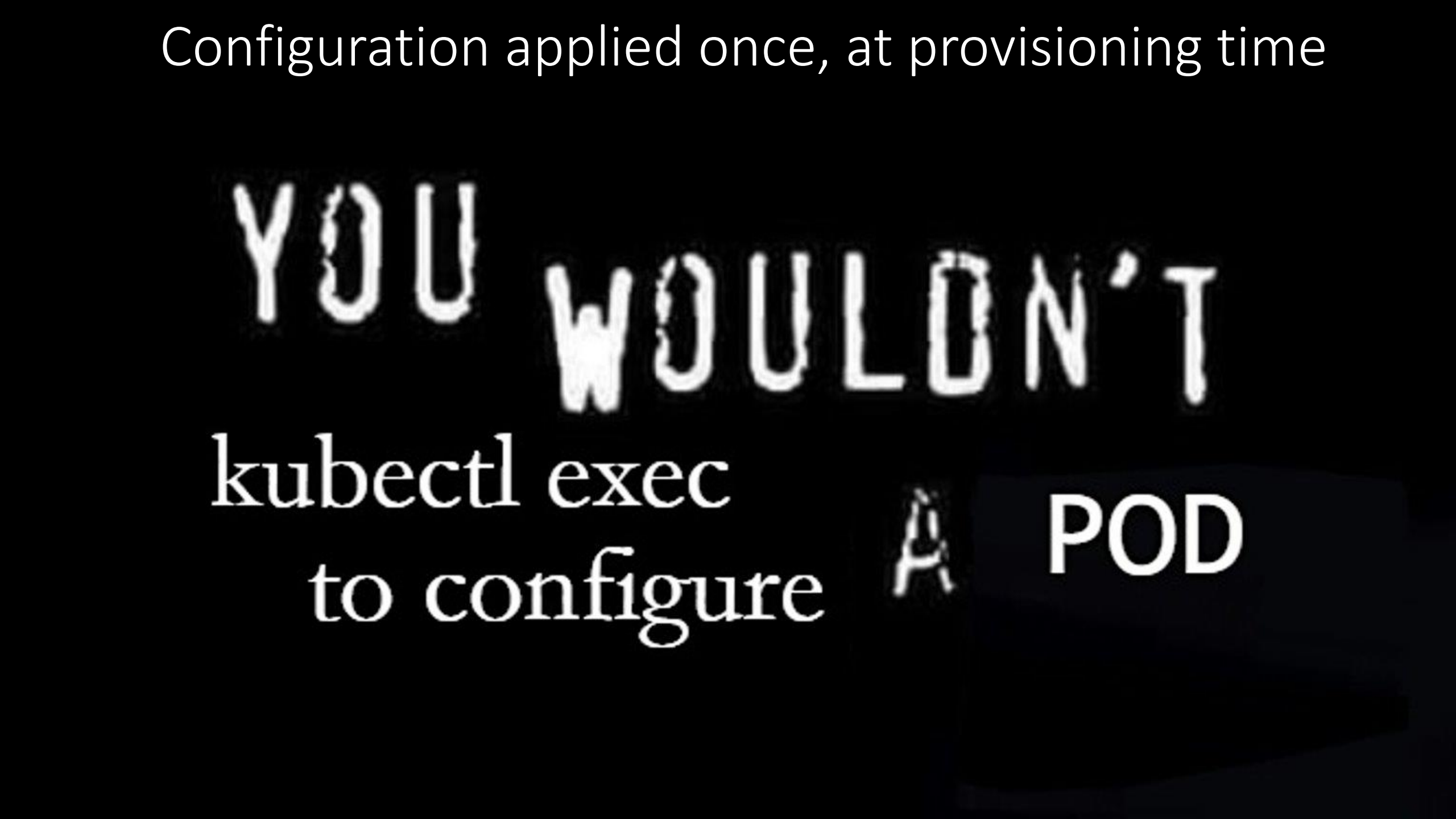

```
variant: flatcar
version: 1.0.0

passwd:
  users:
    - name: caddy
      no_create_home: true
      groups: [ docker ]

storage:
  files:
    - path: /srv/www/html/index.html
      mode: 0644
      user:
        name: caddy
      contents:
        inline: |
          <html><body align="center">
          <h1>Hallo FrOSCon 2024!</h1>
          <img src="froscon_logo_print_color.png" alt="FrOSCon logo"
          </body></html>
    - path: /srv/www/html/froscon_logo_print_color.png
      mode: 0644
      user:
        name: caddy
      group:
        name: caddy
      contents:
        local: froscon_logo_print_color.png

systemd:
  units:
    - name: update-engine.service
      mask: true
    - name: Froscon-demo-webserver.service
      enabled: true
      contents: |
        [Unit]
        Description=FrOSCon example static web server
        After=docker.service
        Requires=docker.service
        [Service]
        User=caddy
        TimeoutStartSec=0
        ExecStartPre=-/usr/bin/docker rm --force caddy
        ExecStart=/usr/bin/docker run -i -p 80:80 --name caddy \
                  -v /srv/www/html:/usr/share/caddy \
                  docker.io/caddy caddy file-server \
                  --root /usr/share/caddy --access-log
        ExecStop=/usr/bin/docker stop caddy
        Restart=always
        RestartSec=5s
        [Install]
        WantedBy=multi-user.target
```

Configuration applied once, at provisioning time



Booth Demos

Saturday, 13:00 - Sunday, 13:00

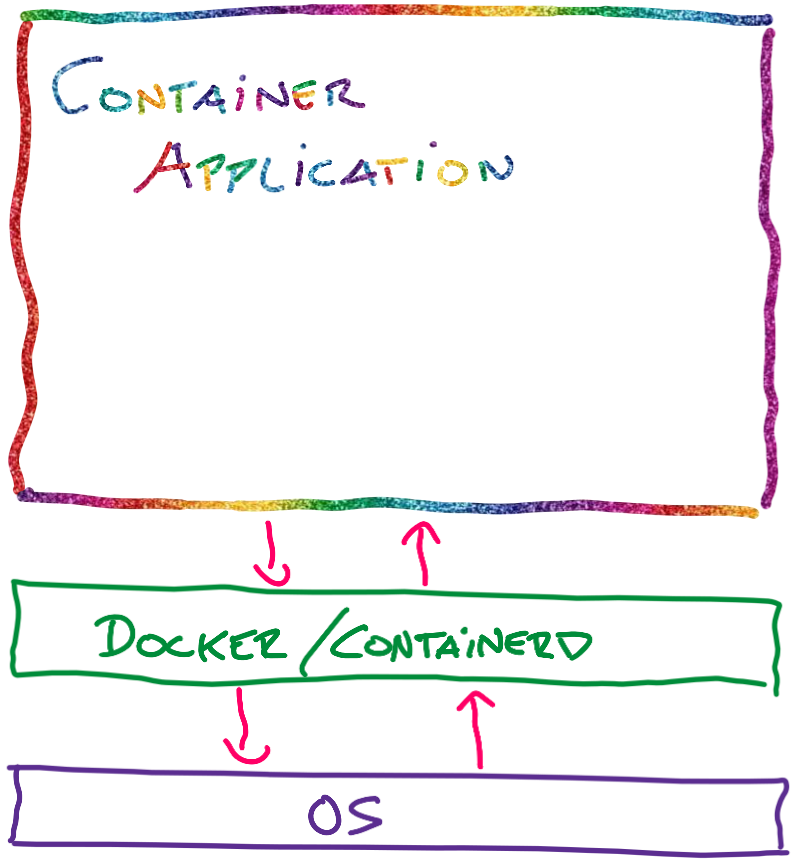
Deploy a Jitsi Server On Demand!

Quickstart guide:

Deploy NGINX in a local QEMU VM

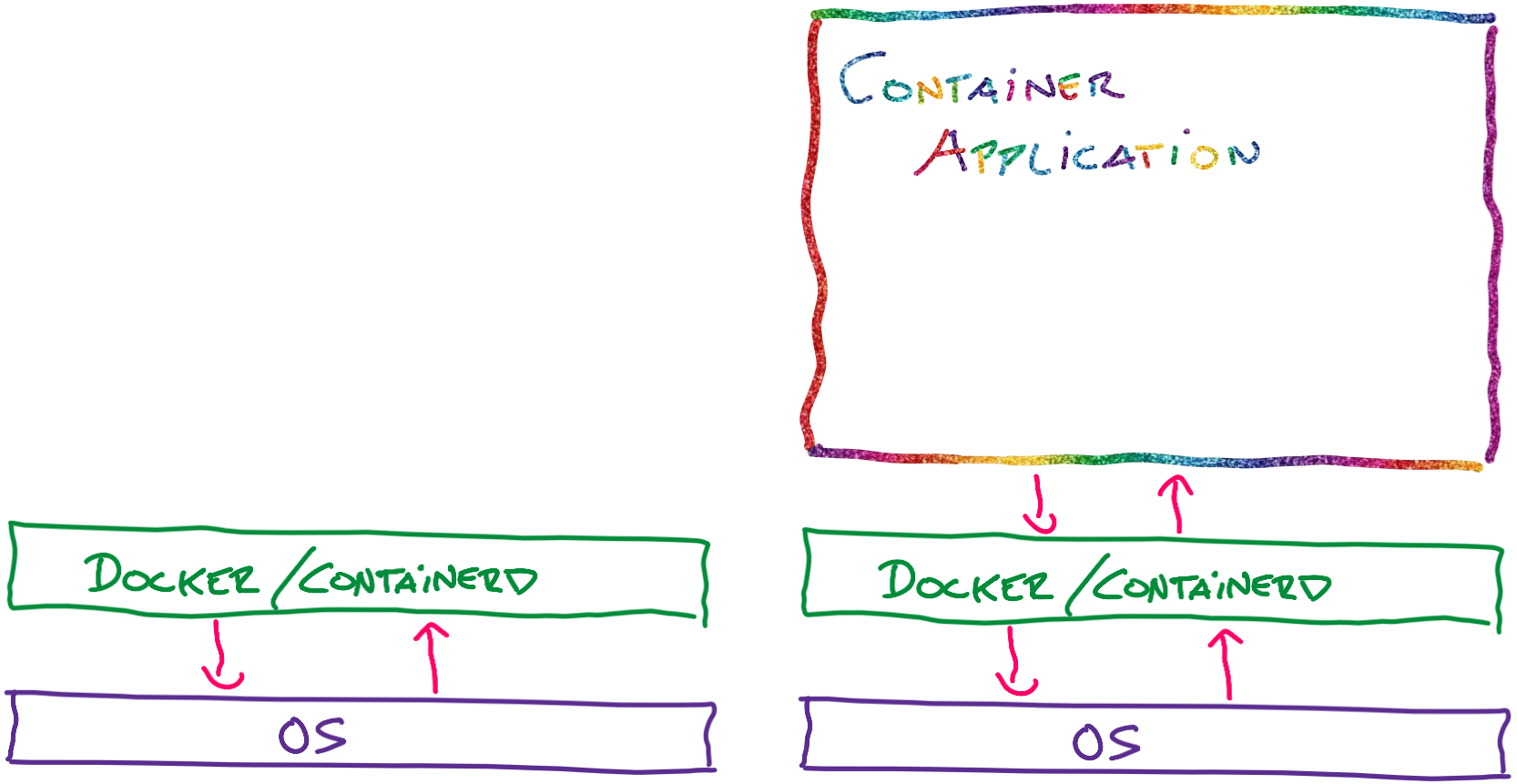
flatcar.org/docs/latest/installing/

Automated, Atomic In-Place Updates



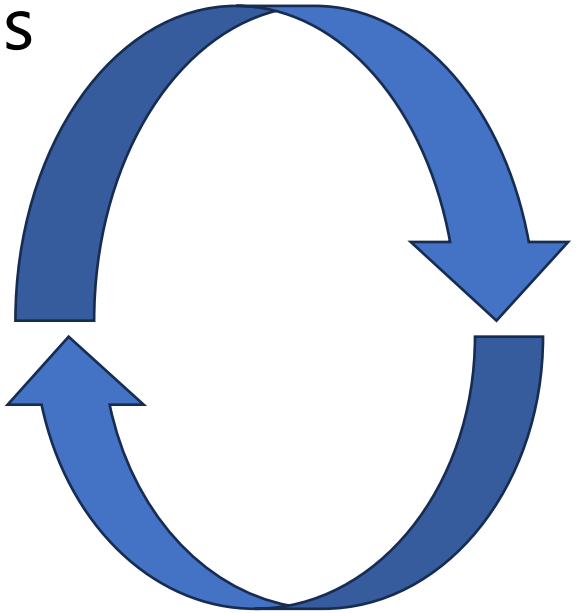
Automated, Atomic In-Place Updates

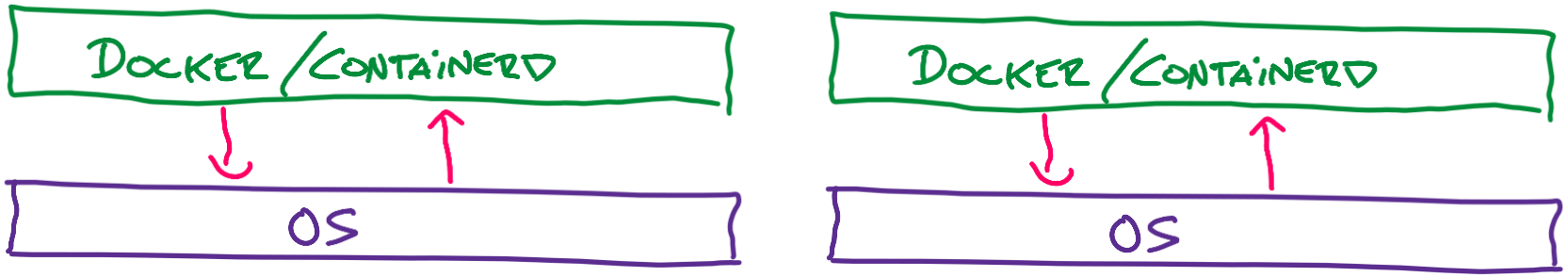
1. Stage



Automated, Atomic In-Place Updates

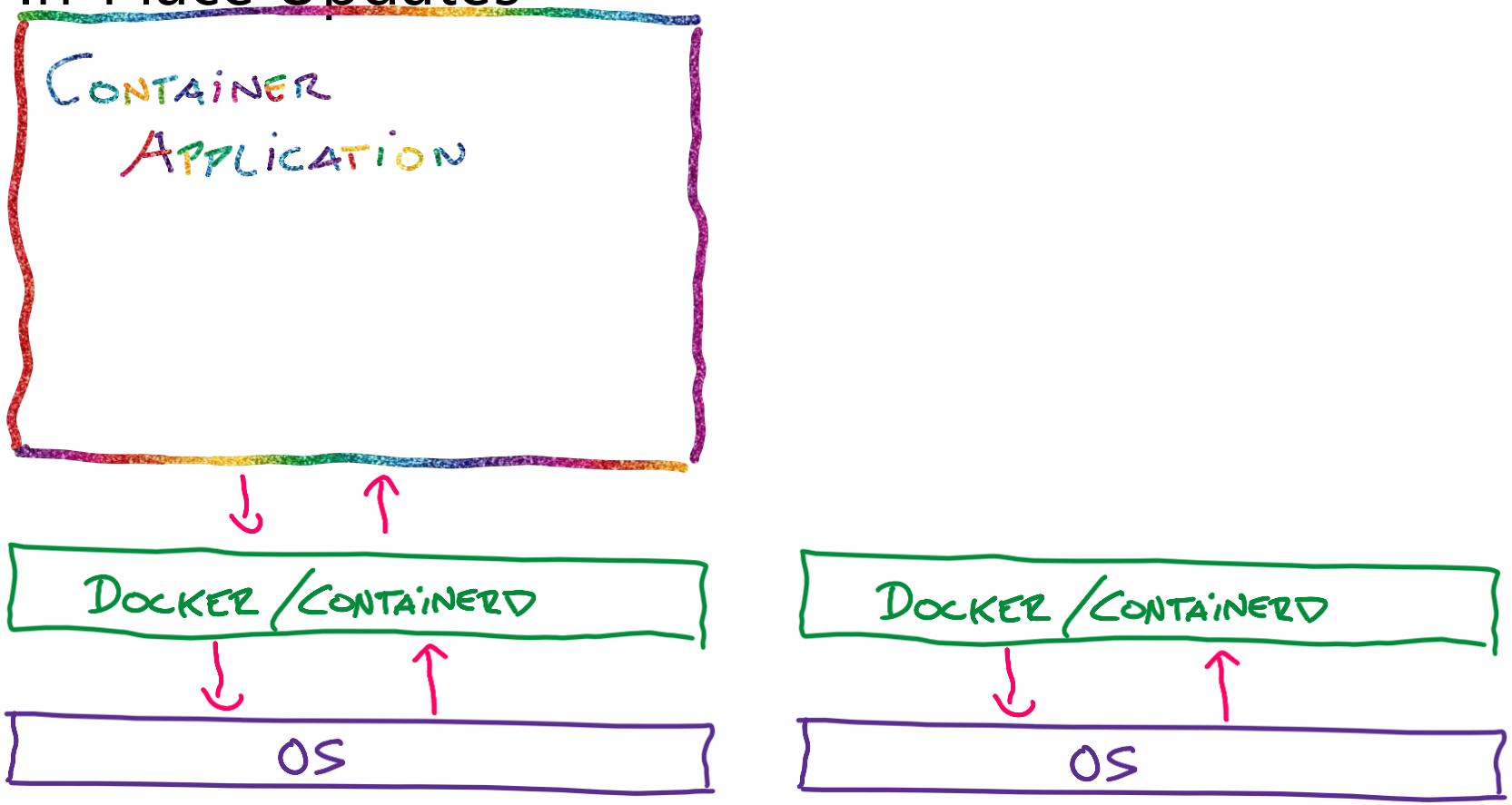
1. Stage
2. Activate





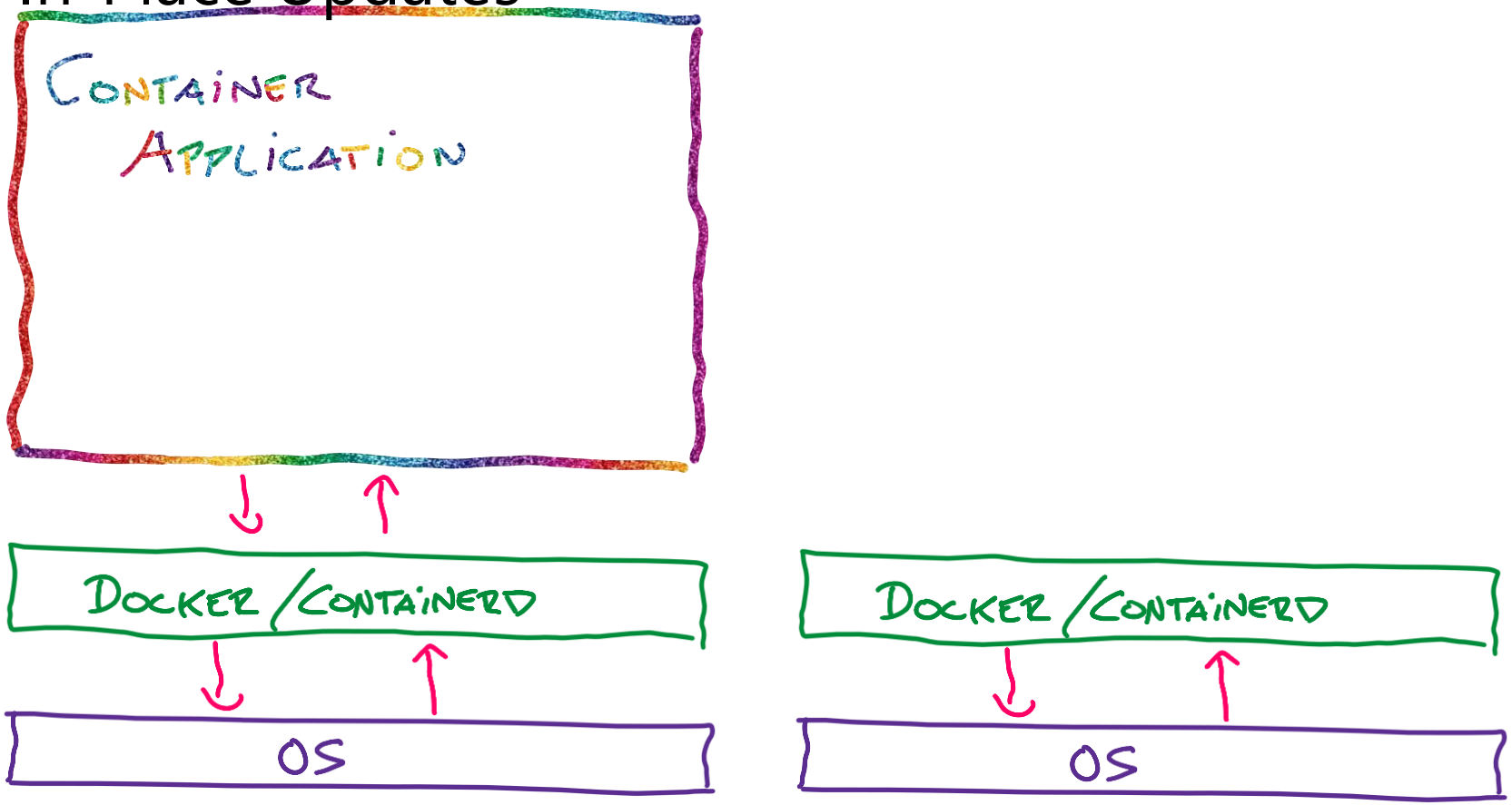
Automated, Atomic In-Place Updates

1. Stage
2. Activate
3. Done



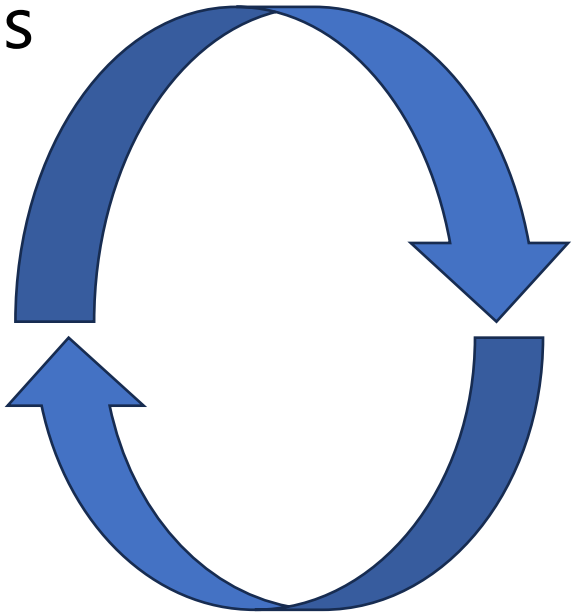
Automated, Atomic In-Place Updates

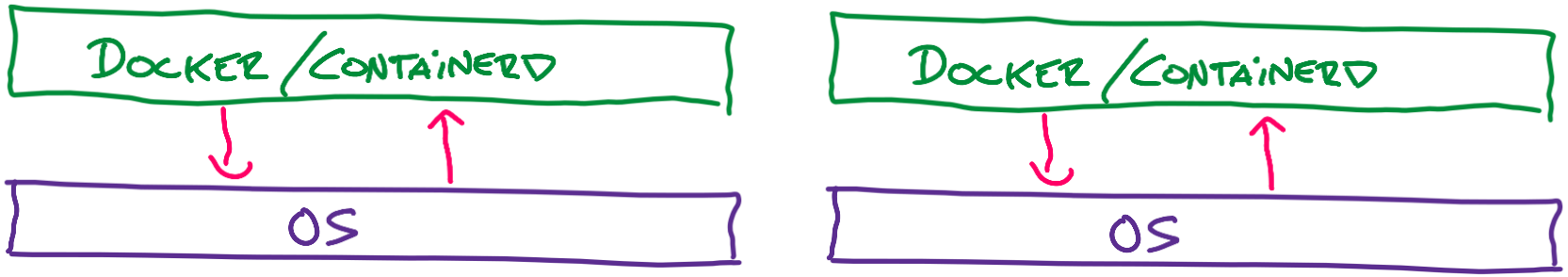
1. Stage
2. Activate
3. Done?



Automated, Atomic In-Place Updates

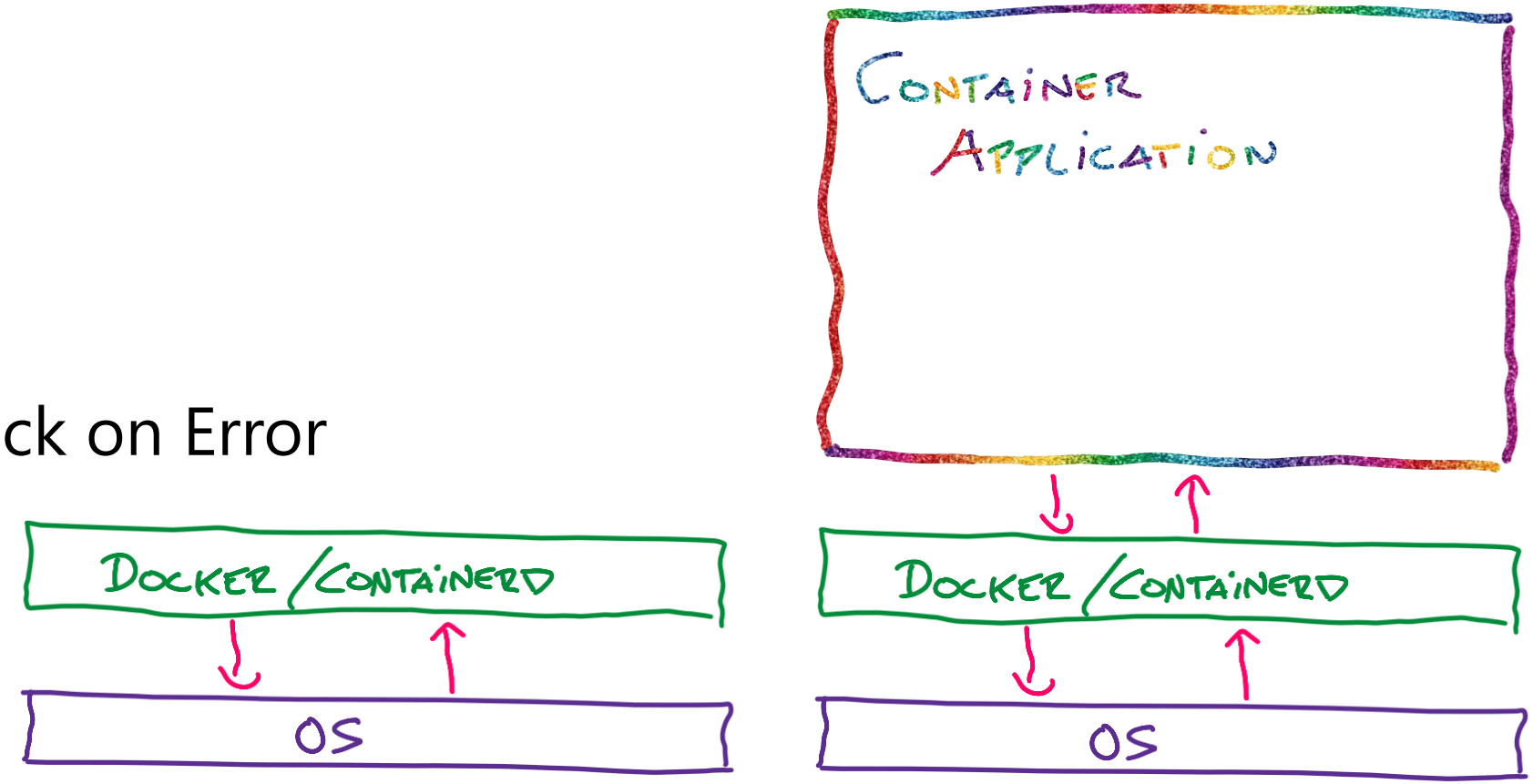
1. Stage
2. Activate
3. Done?
4. Atomic Roll-Back





Automated, Atomic In-Place Updates

1. Stage
2. Activate
3. Done?
4. Atomic Roll-Back on Error



Flatcar Container Linux

Terraform examples:

github.com/flatcar/flatcar-terraform/

Flatcar Container Linux

Deploy a Jitsi server, fully automated:

github.com/flatcar/jitsi-server

The Community's
Container Linux