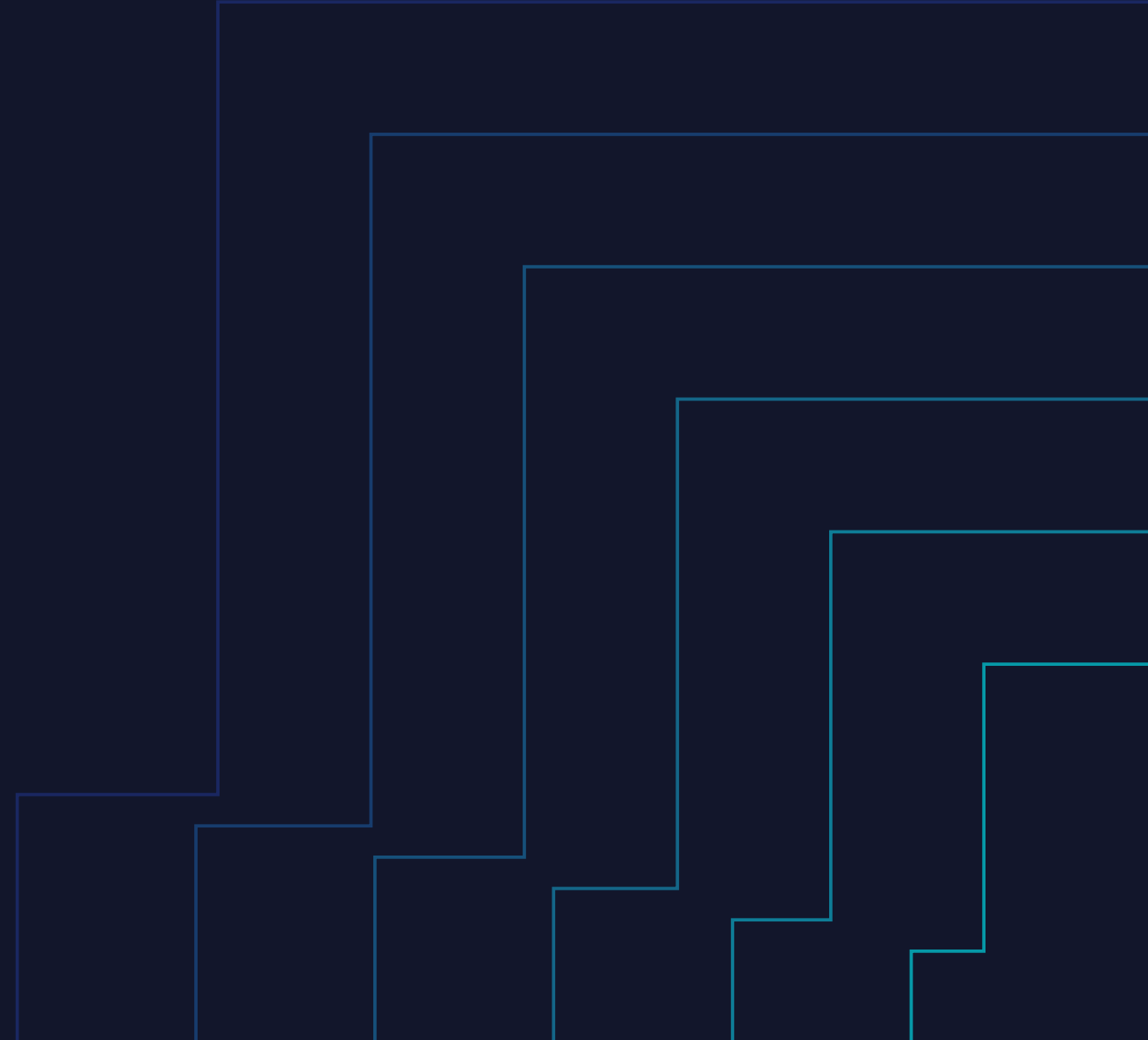




An overview of Ignition

frOScon – Sankt Augustin, Germany

20th of August 2022



Hallo 🙋



Mathieu Tortuyaux (@tormath1)

Software Engineer, Microsoft

Provisioning



With Ignition, it's declarative

```
{
  "ignition": {
    "version": "3.3.0"
  },
  "passwd": {
    "users": [
      {
        "groups": [
          "docker"
        ],
        "name": "caddy",
        "noCreateHome": true
      }
    ]
  },
  "storage": {
    "files": [
      {
        "group": {
          "name": "caddy"
        },
        "path": "/src/www/html/brand-logo.svg",
        "user": {
          "name": "caddy"
        },
        "contents": {
          "source": "https://www.flatcar.org/media/brand-logo.svg"
        },
        "mode": 420
      },
      {
        "group": {
          "name": "caddy"
        },
        "path": "/srv/www/html/index.html",
        "user": {
          "name": "caddy"
        },
        "contents": {
          "source":
```

With Ignition, it's all or nothing

- [Boot or die - asciinema](#)

With Ignition, it runs only once

- Boot once - asciinema

With Ignition, config can be generated

```

variant: flatcar
version: 1.0.0
passwd:
  users:
    - name: caddy
      no_create_home: true
      groups:
        - docker
storage:
  files:
    - path: /src/www/html/brand-logo.svg
      contents:
        source: https://www.flatcar.org/media/brand-logo.svg
        mode: 0644
        user:
          name: caddy
        group:
          name: caddy
    - path: /srv/www/html/index.html
      mode: 0644
      user:
        name: caddy
      group:
        name: caddy
      contents:
        inline: |
          <html><body align="center">
            <h1>Hello world!</h1>
            
          </body></html>
  filesystems:
    - path: /srv/www
      device: /dev/disk/by-partlabel/USR-B
      format: ext4
      wipe_filesystem: true
      with_mount_unit: true
systemd:
  units:
    - 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

```

butane < butane.yml > ignition.json

```

{
  "ignition": {
    "version": "3.3.0"
  },
  "passwd": {
    "users": [
      {
        "groups": [
          "docker"
        ],
        "name": "caddy",
        "noCreateHome": true
      }
    ]
  },
  "storage": {
    "files": [
      {
        "group": {
          "name": "caddy"
        },
        "path": "/src/www/html/brand-logo.svg",
        "user": {
          "name": "caddy"
        },
        "contents": {
          "source": "https://www.flatcar.org/media/brand-logo.svg"
        },
        "mode": 420
      },
      {
        "group": {
          "name": "caddy"
        },
        "path": "/srv/www/html/index.html",
        "user": {
          "name": "caddy"
        }
      }
    ]
  }
}

```

With Ignition, config can be generated

```

</body></html>
filesystems:
  - path: /srv/www
    device: /dev/disk/by-partlabel/USR-B
    format: ext4
    wipe_filesystem: true
    with_mount_unit: true
systemd:
units:
```

butane < butane.yml > ignition.json

```
{
  "contents": "# Generated by Butane\n[Unit]\nBefore=local-fs.target\nRequires=systemd-fsck@dev-disk-by\\x2dpartlabel-USR\\x2dB.service\nAfter=systemd-fsck@dev-disk-by\\x2dpartlabel-USR\\x2dB.service\n\n[Mount]\nWhere=/srv/www\nWhat=/dev/disk/by-partlabel/USR-B\nType=ext4\n\n[Install]\nRequiredBy=local-fs.target",
  "enabled": true,
  "name": "srv-ww.mount"
},
{
```


Cool, it works locally – what about cloud infra ?

- Metadata server
- Kernel command line parameters

User data

Pass a script, configuration file, or other data that will be accessible to your applications **throughout the lifetime of the virtual machine**. Don't use user data for storing your secrets or passwords. [Learn more about user data for VMs](#) ↗

Enable user data



User data *

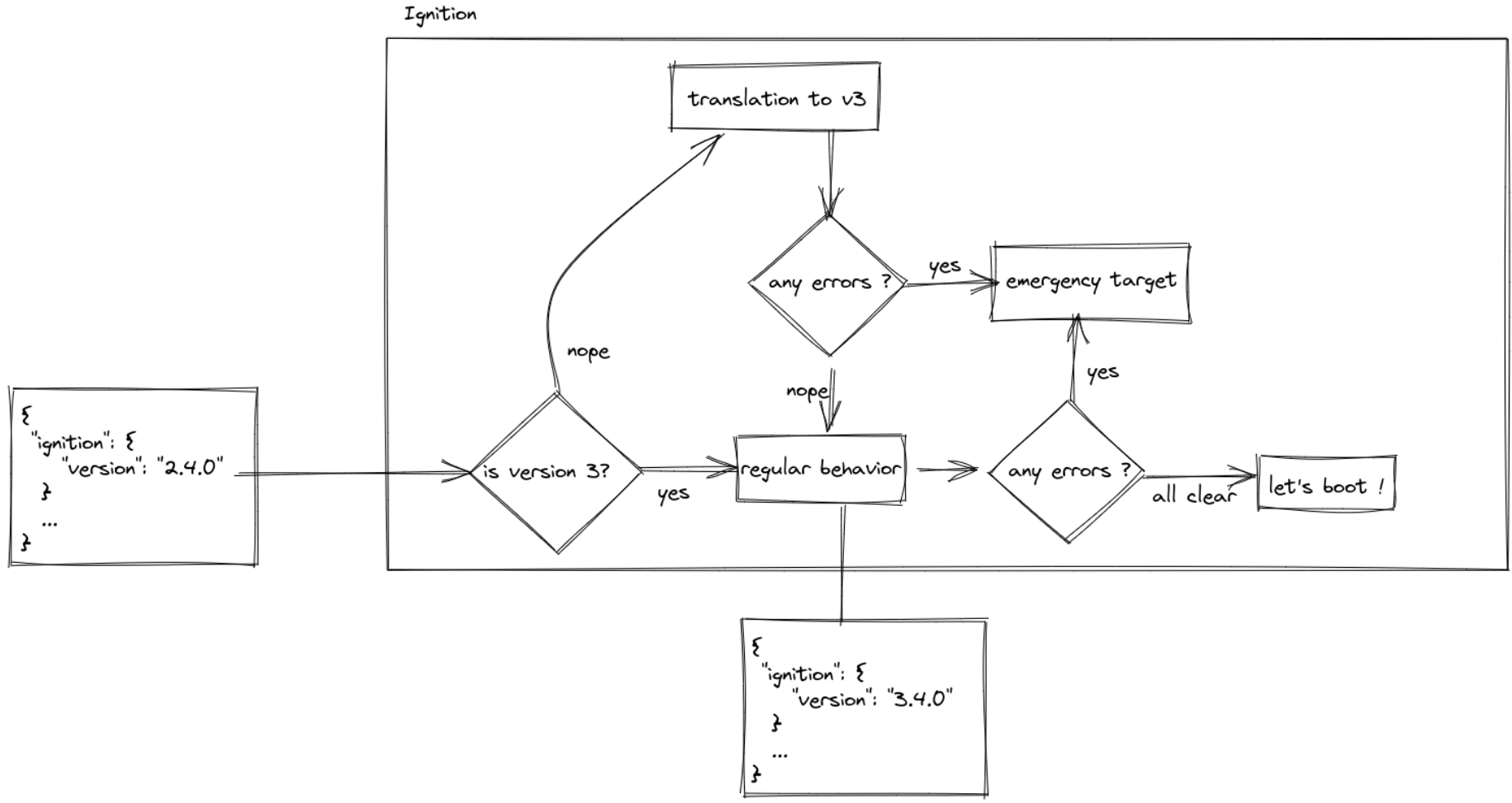
```
{
  "ignition": { "version": "3.0.0" },
  "storage": {
    "files": [{
      "path": "/etc/hostname",
      "mode": 420,
      "overwrite": true,
```

Ignition spec

- v1
- v2.x.y
- v3.x.y

```
f
  "ignition": {
    "version": "3.3.0"
  },
  "passwd": {
    "users": [
      {
        "groups": [
          "docker"
        ],
        "name": "caddy",
        "noCreateHome": true
      }
    ]
  },
  "storage": {
    "files": [
      {
        "group": {
          "name": "caddy"
        },
        "path": "/src/www/html/brand-logo.svg",
        "user": {
          "name": "caddy"
        },
        "contents": {
          "source": "https://www.flatcar.org/media/brand-logo.svg"
        },
        "mode": 420
      },
      {
        "group": {
          "name": "caddy"
        },
        "path": "/srv/www/html/index.html",
        "user": {
          "name": "caddy"
        }
      }
    ]
  }
}
```

Ignition on Flatcar



Ignition on Flatcar

```
44 PATCHES=(
45     "${FILESDIR}/0001-sed-s-coreos-flatcar-linux.patch"
46     "${FILESDIR}/0002-mod-add-ign-converter-dependency.patch"
47     "${FILESDIR}/0003-sum-go-mod-tidy.patch"
48     "${FILESDIR}/0004-vendor-go-mod-vendor.patch"
49     "${FILESDIR}/0005-config-v3_4-convert-ignition-2.x-to-3.4-exp.patch"
50     "${FILESDIR}/0006-internal-prv-cmdline-backport-flatcar-patch.patch"
51     "${FILESDIR}/0007-provider-qemu-apply-fw_cfg-patch.patch"
52     "${FILESDIR}/0008-config-3_4-test-add-ignition-2.x-test-cases.patch"
53     "${FILESDIR}/0009-internal-disk-fs-ignore-fs-format-mismatches-for-the.patch"
54     "${FILESDIR}/0010-VMware-Fix-guestinfo-.config.data-and-.config.url-v.patch"
55     "${FILESDIR}/0011-config-version-handle-configuration-version-1.patch"
56     "${FILESDIR}/0012-config-util-add-cloud-init-detection-to-initial-pars.patch"
57     "${FILESDIR}/0013-Revert-drop-OEM-URI-support.patch"
58     "${FILESDIR}/0014-internal-resource-url-support-btrfs-as-OEM-partition.patch"
59     "${FILESDIR}/0015-internal-exec-stages-disks-prevent-races-with-udev.patch"
60 )
61
```

Ignition workflow

- Initramfs
- fetch-offline [-> fetch] [-> kargs] -> disks -> mount -> files

Kudos

- @bgilbert
- Fedora CoreOS folks
- Flatcar Alpha and Beta users



Thank you

Mathieu Tortuyaux (@tormath1)

Software Engineer, Microsoft