Getting Started with Fedora CoreOS
A Hands-On Workshop
Dusty Mabe - @dustymabe

Timothée Ravier - @siosm

Nasir Hussain - @_nasirhm_
Fedora CoreOS - Emerging Fedora Edition

- Came from the merging of two communities:
  - CoreOS Inc’s Container Linux
  - Project Atomic’s Atomic Host
- Incorporates Container Linux
  - Philosophy
  - Provisioning Stack
  - Cloud Native Expertise
- Incorporates Atomic Host
  - Fedora Foundation
  - Update Stack
  - SELinux Enhanced Security
Features: Automatic Updates

- Fedora CoreOS features Automatic Updates by default
  - Automatic updates → Reliable updates
    - Extensive tests in automated CI pipelines
    - Several update streams to preview what’s coming
      - Users run various streams to help find issues
    - Managed upgrade rollouts over several days
      - Halt the rollout if issues are found
  - For when things go wrong
    - rpm-ostree rollback can be used to go back
    - future: automated rollback
      - based on user specified health checks
Multiple Update Streams

• Offered update streams with automatic updates
  ○ next - experimental features, Fedora major rebases
  ○ testing - preview of what’s coming to stable
    ■ point in time snapshot of Fedora stable rpm content
  ○ stable - most reliable stream offered
    ■ promotion of testing stream after some bake time

• Goals
  ○ Publish new releases into update streams every two weeks
  ○ Find issues in next/testing streams before they hit stable
Features: Automated Provisioning

- Fedora CoreOS uses Ignition to automate provisioning
  - Any logic for machine lifetime is encoded in the config
    - Very easy to automatically re-provision nodes
  - Same starting point whether on bare metal or cloud
    - Use Ignition everywhere as opposed to kickstart for bare metal and cloud-init for cloud
Ignition: Details

Ignition configs
- Declarative JSON documents provided via user data
- Runs exactly once, during the initramfs stage on first boot
- Can write files and systemd units, create users and groups, partition disks, create RAID arrays, format filesystems
- If provisioning fails, the boot fails (no half provisioned systems)
- Ignition configs are machine-friendly (JSON), currently spec v3

Writing Configs
- Fedora CoreOS Config Transpiler to translate to Ignition spec
  - Configs are Human friendly (YAML)
  - Ignition semantics, plus sugar for common operations
  - Transpiler catches common errors at build time

```json
{
  "ignition": {
    "config": {},
    "timeouts": {},
    "version": "3.0.0"
  },
  "passwd": {
    "users": [
      {
        "name": "core",
        "passwordHash": "$6$43y3tkl...",
        "sshAuthorizedKeys": [
          "key1"
        ]
      }
    ],
    "storage": {},
    "systemd": {}
  }
}
```
Features: Cloud Native & Container Focused

- Software runs in containers
  - podman or moby engine container runtimes
- Ready for clustered deployments
  - Spin up 100 nodes and have them join a cluster
    - Ignition configs used to automate cluster join
  - Spin down nodes when no longer needed
  - Spin up nodes again when load increases
- Offered on (or for) a plethora of cloud/virt platforms
  - Alibaba, AWS, Azure, DigitalOcean, Exoscale, GCP, Openstack, Vultr, VMWare, QEMU/KVM
Features: OS Versioning & Security

- Fedora CoreOS uses rpm-ostree technology
  - “Like git for your Operating System”
    - 32.20200615.2.0 - 86c0246
    - A single identifier tells you all software in that release
  - Uses read-only filesystem mounts
    - Prevents accidental OS corruption (rm -rf)
    - Prevents novice attacks from modifying system
- SELinux enforcing by default
  - Prevents compromised apps from gaining further access
What’s Next

• More Cloud Platforms
• Multi-arch support (aarch64, ppc64le, s390x)
• More FCCT human friendly helper functions
• Host extensions (more reliable package layering)
• More/improved documentation
• Tighter integrations with OKD
Workshop

• Composed of 5 Tutorials
  • Initial Setup
    ■ Setup instructions that must be completed first
  • Enabling autologin and custom hostname
    ■ Write your first Ignition config and start a Fedora CoreOS instance
  • Starting a service on first boot
    ■ Learn how to start a custom script via a systemd unit on the first boot
  • SSH access and starting containers
    ■ Learn how to automatically start a container on first boot and to SSH in
  • Testing Fedora CoreOS updates
    ■ Learn how updates are handled and how to rollback if needed
Workshop Options

• Execute on your own, ask questions when you have them
  • Self Guided
• Ride along with us
  • On the video stream we’ll share our screen and run through the workshop
  • Please interrupt us and ask questions when you have them
    ■ This is intended to be a learning experience

https://docs.fedoraproject.org/en-US/fedora-coreos/tutorials/
Get involved!

- Web: https://getfedora.org/coreos
- Issues: https://github.com/coreos/fedora-coreos-tracker/issues
- Forum: https://discussion.fedoraproject.org/c/server/coreos
- Mailing list: coreos@lists.fedoraproject.org
- IRC: freenode #fedora-coreos
Thank you!