The CentOS Ecosystem

Karanbir Singh

CentOS

redhat.

The Red Hat relationship

Everything is a SIG

Abstract the CentOS Project away from the CentOS Linux distro :: So we can do more and we can be wider inclusive.

The Goals

- Be a great environment for people to develop, adapt and adopt tools, code, projects, apps in and around the Infrastructure demographics. This should include the traditional established roles, as well as create opportunities to evolve into the emerging work roles, eg: containers and cloud.

- Provide the ecosystem the tools, support and resources they need to be successful with the CentOS project. Make our success and relevance be directly linked to the wider ecosystem. Driven by user stories and not vendor / provider ones.

How are we approaching these goals ?

Stream 1 : Availability

- More reliable mirror network for existing install base
- Extend mirror network to operate over IPv6
  - Mirrorlist -> done, Self Mirrors -> soon
- Updated media, evolving to run Monthly (with a full CI run)
- More media variants
- Boot.centos.org
- Early access to builds, (buildlogs.centos.org)
- ABRT and Debugging in the public space

Stream 1 : Availability

- Alternative Media / Formats
  - Cloud roles
    - Aws / gce / rackspace / linode etc
    - Hosting vendors / vps instance vendors
  - Containers
    - Docker (both /_/centos and /u/centos)
    - Base Images & app layered images

Stream 1 : Availability

Whatever format, delivery mechanish, runtime environment, bootstrap process – the baseline content is always in sync.

Need: Ability to operate outside the constraints of the specific workload.

Stream 2 : Extend Availability

- Complementing the existing expectations :
  - ARMv7
  - Aarch64:ARMv8
  - i686
- Expanding the existing expectations:
  - Power8 le/be
  - Power7
  - s390x

Regardless of where you want to run your workload, CentOS should provide a good-enough platform within the constraints of that environment.

Lets look at the Layered ecosystem

Stream 3: Facilitate Layers

- Ability to build on top of the CentOS Linux platform
  - Towards a component goal
  - Directly to satisfy a User Story
- Dont worry about the user base skew
- Low barrier to entry
  - Including signup
  - Content injection
  - Licensing requirements
  - Sponsorship requirements

Stream 3: Facilitate Layers

Complete Buildsystem for end to end requirements

Ability to consume via layers other peoples work in the same environ

Ability to contribute into layered repos and scratch builds

- Git.centos.org : version control
- - sync with distro
- - branch from distro or others
- - completely own your branch

- Cbs.centos.org : koji buildsys
- - lots of cpu, disk, ram
- - local mirrors
- - integrated with git.centos.org
- - automation capabilities

Stream 3: Facilitate Layers

- Testing capabilities via CI.CentOS.org:
- - 768 cores, 3 TiB of ram, 128 SSD based physical machines
- - ability to consume test instances from bare metal, including groups
- - Test iteratively over base + additional layers
- - Lots of reporting options (Jenkins driven)

- Content delivery via centos.org
- - build time repos, cbs.centos.org/repos/
- - testing / eval content via buildlogs.centos.org
- - production content via mirror.centos.org
- - not limited to existing media types, additional formats welcome

Early Success stories

- Base Layer
  - Qemu
  - Libvirt
  - Libguestfs
  - Xen
- Component Layer
  - Gluster FS
  - Ovirt
- App Layer
  - OpenStack
  - Atomic Project

Early Success stories

- Developer wins
  - Nightly builds for docker / kubernetes allows projects like Mesos to develop/test against CentOS Linux. Optionally consuming for their own dep chains either:
    - In Distro versions
    - Upstream Stable versions
    - Upstream nightly versions
- CI / CD pipelines
  - Tests to validate docs, regression suites on a dep chain change, and being able to link like a matrix to build developer and user confidence in a specific stack: eg: GlusterFS doing CI against both itself, the distro, and an evolving set of dep chains (like libvirt / qemu etc).

Early Success stories

- OpenStack : Tested both ways, delivered with confidence.
  - RDO and Khaleesi runs nightly. Packages are only pushed on pass.
  - CentOS Linux snapshots are tested nightly, No package pushes.
  - The docs and use case validations:
    - On every CentOS Linux rpm update (dep or not)
    - On every openstack package update
    - Run as a stack, rpms both ways only pushed on complete success
  - Result: Using the CentOS Linux Base + CentOS Linux Openstack tested updates stack, the user should never see content that hasent passed a pre-defined test set.

Early Success stories

- Atomic App: A complete application in closed loop, containers with real dep and in dep tracking.
  - Single definition
  - Able to use definable registries to satisfy deps
  - Dep and in dep tracking of components, so relationships can be identified and acted-on (eg. Upward escalation of a security issue in an underlying dep container)
  - Delivered itself as a CentOS hosted container (docker for now, but others are easy to add)
  - Tested, managed, facilitated within the CentOS CI infra (Community container pipeline).

Our focus remains to curate the existing user base, the user mindset and to help create a manageable bridge for those looking to investigate and trial new emerging technologies.

Getting in Touch

- K Singh
- Twitter.com/kbsingh
- Tel: +44 207 009 4455
- Email: kbsingh@centos.org
- Kbsingh on irc.freenode.net