Building Windows Images for Firefox CI

How SRE at Mozilla builds Windows Images for use in the continuous integration of Mozilla Firefox

Speaker Name

Jonathan Moss
Site Reliability Engineer at Mozilla
Github: https://github.com/jwmoss

POWERSHELL EST 2019 SATURDAY

RTPSUG

Thank you to our sponsors!

System Frontier

TEKsystems

Own change

IRONMAN SOFTWARE

Chocolatey

Problem Statements

- How do we build windows machine images quickly?
- How do we ensure the windows machine images are tested before they’re used in continuous integration?
- How do we manage secrets when building machine images in Azure or other clouds?
- How do we keep costs low?
- How do we document what’s installed on machine images?
- How do we make it easy to contribute?

Packer

- Tool used to build images for multiple platforms from a single source configuration.
- Integrates with PowerShell.
- Build images in Azure, GCP, and AWS.

Puppet

- Open source configuration as code language written in ruby.
- Often deployed in a client/server environment.
- We use a “master-less” implementation.
  - Link: https://github.com/mozilla-platform-ops/ronin_puppet

Continuous Integration for Firefox at Mozilla

Firefox's continuous integration (CI) system is made up of three parts.

First there's Taskcluster, a task execution framework developed by Mozilla. Taskcluster is capable of building complex, scalable and highly customizable CI systems. Taskcluster is more like a set of building blocks that can be used to create CI systems, rather than a fully-fledged CI system itself.

The second part is the Firefox CI instance of Taskcluster. This is the Taskcluster deployment responsible for running most of Mozilla's CI needs. This component is comprised of a Kubernetes cluster to run the Taskcluster services (maintained by SRE Services), some customizations to support Taskcluster on hg.mozilla.org and access control in the ci-configuration repo (maintained by Release Engineering).

The third part is Taskgraph. Taskgraph is a Python library that can generate a DAG of tasks and submit them to a Taskcluster instance. This is the component that most Gecko and Mozilla developers will interact with when working with tasks, and will be the focal point of the rest of this documentation.

Source: https://firefox-source-docs.mozilla.org/taskcluster/index.html

Software used within Continuous Integration at Mozilla:

- Python
- PowerShell
- Golang
- HashiCorp Configuration Language (HCL)
- Javascript

Let’s look at some code!

Questions?

References

- Worker-Images: https://github.com/mozilla-platform-ops/worker-images
- Azure Packer Builder: https://developer.hashicorp.com/packer/integrations/hashicorp/azure/latest/components/builder/arm
- Pester: https://pester.dev/docs/quick-start
- Firefox: https://www.mozilla.org/en-US/firefox/
- Treeherder: https://github.com/mozilla/treeherder
- Taskcluster: https://github.com/taskcluster/taskcluster
- Building Firefox on Windows: https://firefox-source-docs.mozilla.org/setup/windows_build.html
- Ronin Puppet: https://github.com/mozilla-platform-ops/ronin_puppet