WIKIMEDIA FOUNDATION

Puppet

Taavi Väänänen
SRE, Wikimedia Cloud Services
Wikimedia Hackathon 2024, Tallinn

Topics this presentation will cover

1. What is Puppet, exactly?
2. How is it used at Wikimedia?
3. How can I make Puppet changes when I need to?

Wikimedia Site Reliability Engineering (SRE)

- 50+ people
- Responsible for running Wikimedia's sites and services used by the general public (including MediaWiki and all associated services) reliably, securely, and with high performance
- We run our own hardware on ~~6~~ 7 colocation sites around the world

WIKIMEDIA FOUNDATION

01 What is Puppet?

What is Puppet?

- Declarative software configuration management tool
- Uses node *facts* to compile a *catalog* on the server, which then gets applied by the agent
- Configured using a Ruby-inspired custom DSL
- Puppet is not:
  - a software deployment tool (use Apt, Scap3, or Helm instead)
  - a command orchestration tool (use Cumin/Spicerack instead)

Example

```
# SPDX-License-Identifier: Apache-2.0
# @summary Installs and configures all the custom Toolforge CLIs
# @param web_domain domain under which all tool webservices are exposed
class profile::toolforge::bastion::toolforge_cli (
  Stdlib::Fqdn $web_domain = lookup('profile::toolforge::web_domain', {default_value =>
'toolforge.org'}),
) {
  package { [
    'toolforge-cli',
    'toolforge-builds-cli',
    'toolforge-envvars-cli',
    'toolforge-jobs-framework-cli',
    'toolforge-webservice',
  ]:
    ensure => installed,
  }

  $harbor_domain = "${::wmcs_project}-harbor.wmcloud.org"
  $cli_config = {
    'build' => {
      'dest_repository' => $harbor_domain,
      'builder_image'   => "${harbor_domain}/toolforge/heroku-builder-classic:22",
```

Example

```
# SPDX-License-Identifier: Apache-2.0
# @summary Installs and configures all the custom Toolforge CLIs
# @param web_domain domain under which all tool webservices are exposed
class profile::toolforge::bastion::toolforge_cli (
  Stdlib::Fqdn $web_domain = lookup('profile::toolforge::web_domain', {default_value =>
'toolforge.org'}),
) {
  package { [
    'toolforge-cli',
    'toolforge-builds-cli',
    'toolforge-envvars-cli',
    'toolforge-jobs-framework-cli',
    'toolforge-webservice',
  ]:
    ensure => installed,
  }

  $harbor_domain = "${::wmcs_project}-harbor.wmcloud.org"
  $cli_config = {
    'build' => {
      'dest_repository' => $harbor_domain,
      'builder_image'   => "${harbor_domain}/toolforge/heroku-builder-classic:22",
```

Advanced features

- PuppetDB
  - Allows using data or resources from one node on another
- External Node Classifier (ENC)
  - Used by the Horizon Puppet integration

WIKIMEDIA FOUNDATION

02 Using Puppet at Wikimedia

Wikimedia environments

- Two main environments:
  - “Production”
    - 2,000+ physical servers, 250+ VMs
  - Cloud VPS
    - ~900 VMs
- Out of scope here:
  - Fundraising
  - Wikimedia Enterprise

operations/puppet.git

- Very active repo: ~99,000 non-merge commits since 2011 (~21 commits/day for the past 12,5 years)
  - Compare: mediawiki/core.git has ~90,000 non-merge commits since 2003 (~13 commits/day for the past 21 years)
- Merge access restricted to `ops` group (aka “global `root`”)
- Being slowly Apache-2.0 licensed

Structure

- Modules
  - Manage an individual technology
- Profiles
  - Use resources and modules to manage a specific technology stack
- Roles
  - Use profiles to manage a complete system with a specific task
  - Each host should have exactly one role applied

Testing tools: PCC

- Generates a diff in the generated catalog with the production branch and the given commit
- To operate, either:
  - Add a `Hosts:` trailer to the commit message, and comment `check experimental`
  - Use the `./utils/pcc` tool included in puppet.git
- Will not catch syntax errors in config files, etc.

Resources only in the new catalog

- File[/var/log/kubernetes/]
- File[/etc/kubernetes/audit-policy.yaml]

Resources only in the old catalog

- File[/etc/kubernetes/infrastructure-users]

Resources modified

- Class[K8s::Apiserver]
  Parameters differences:

```
--- Class[K8s::Apiserver].orig
+++ Class[K8s::Apiserver]

+    audit_policy => audit-policy-modify-pods.yaml
```

- File[/etc/default/kube-apiserver]
  Content differences:

```
--- /etc/default/kube-apiserver.orig
+++ /etc/default/kube-apiserver
@@ -6,6 +6,11 @@
 #
 DAEMON_ARGS="--admission-control-config-file=/etc/kubernetes/admission-config.yaml \
   --allow-privileged=true \
+ --audit-log-compress \
+ --audit-log-maxbackup=10 \
+ --audit-log-maxsize=100M \
+ --audit-log-path=/var/log/kubernetes/audit.log \
+ --audit-policy-file=/etc/kubernetes/audit-policy.yaml \
   --authorization-mode=Node,RBAC \
   --client-ca-file=/etc/kubernetes/pki/wikikube_staging__kube-apiserver_server.chain.pem \
   --disable-admission-plugins=PersistentVolumeClaimResize,StorageObjectInUseProtection \
```

Testing tools

- Rspec
  - Unit testing for individual Puppet classes
- Pontoon
  - Tries to make Puppet in Cloud VPS behave more like wikiland
- dcl
  - Container-based setup to allow testing changes on a local development machine

Dealing with secrets

- Hiera, `secret()`
- Private repository, and labs/private.git

Help, I'm confused

- https://wikitech.wikimedia.org/wiki/Puppet/Coding_and_style_guidelines (aka: [[Puppet coding]])
- #wikimedia-sre on irc.libera.chat
  - (for Cloud VPS specific questions/issues, #wikimedia-cloud)

- These slides: https://people.wikimedia.org/~taavi/presentations/2024-hackathon-puppet.pdf

taavi@wikimedia.org

Hiera

```
$ sudo puppet lookup --explain --compile --node
deployment-deploy03.deployment-prep.eqiad1.wikimedia.cloud profile::apt::purge_sources
Searching for "profile::apt::purge_sources"
  Global Data Provider (hiera configuration version 5)
    Using configuration "/etc/puppet/hiera.yaml"
    Hierarchy entry "Http Yaml"
      URI "https://puppet-enc.cloudinfra.wmcloud.org/v1/deployment-prep/node
           /deployment-deploy03.deployment-prep.eqiad1.wikimedia.cloud"
        Original uri: "https://puppet-enc.cloudinfra.wmcloud.org/v1/%{::wmcs_project}/node/
                       %{facts.networking.fqdn}"
       No such key: "profile::apt::purge_sources"
    Hierarchy entry "cloud hierarchy"
      Path "/srv/puppet_code/environments/production/hieradata/cloud/eqiad1/deployment-prep
           /hosts/deployment-deploy03.yaml"
        Original path: "cloud/%{::wmcs_deployment}/%{::wmcs_project}/hosts
                        /%{facts.networking.hostname}.yaml"
         No such key: "profile::apt::purge_sources"
      Path "/srv/puppet_code/environments/production/hieradata/cloud/eqiad1
           /deployment-prep/common.yaml"
        Original path: "cloud/%{::wmcs_deployment}/%{::wmcs_project}/common.yaml"
        Found key: "profile::apt::purge_sources" value: true
```