A Continuation of Devops
Policy as Code

March 2019

Gareth Rushgrove


@garethr
Docker

This talk

What to expect

- **A little history**
  Infrastructure, APIs and devops
- **Parallels with security**
  Security as policy management
- **Security tool examples**
  How can tools facilitate sharing and collaboration

A little history

“The API is the product”

Todd Sampson, way back in 2008

Infrastructure as code

A banner for lots of tools and approaches

Just sysadmins solving problems

From adhoc to software

```
$ sudo apt-get install some-package
$ nano /etc/some-config-file.ini
...
$ nano /etc/some-other-config-file.xml
...
$ sudo service start some-service
```

```
class { 'apache':
  default_vhost => false,
}

apache::vhost { 'vhost.example.com':
  port    => '80',
  docroot => '/var/www/vhost',
}
```

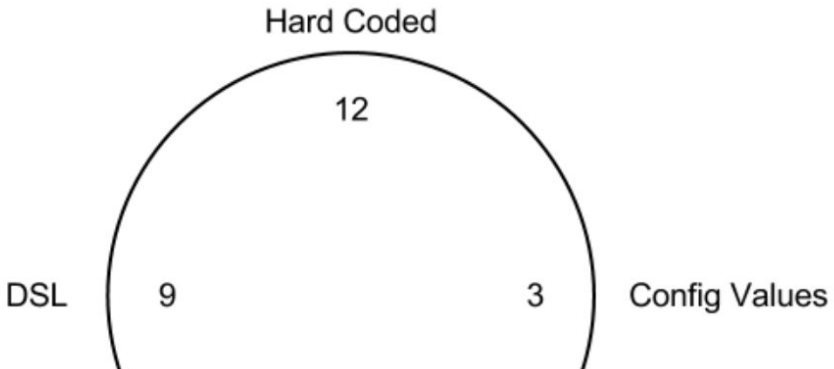
DSLs and the configuration clock

MONDAY, MAY 07, 2012

The Configuration Complexity Clock

When I was a young coder, just starting out in the big scary world of enterprise software, an older, far more experienced chap gave me a stern warning about hard coding values in my software. "They *will* have to change at some point, and you don't want to recompile and redeploy your application just to change the VAT tax rate." I took this advice to heart and soon every value that my application needed was loaded from a huge .ini file. I still think it's good advice, but be warned, like most things in software, it's good advice *up to a point*. Beyond that point lies pain.

Let me introduce you to my 'Configuration Complexity Clock'.



Enter Devops

CHEF BLOG

What Devops Means to Me

By John Willis July 16, 2010

I have been in IT operations for over 30 years and I have seen a lot of good and bad ideas come and go. I have been following the Devops movement for over a year now and I have kept my eyes very focused to the Devops radar. To me Devops is one of the most interesting ideas that I have seen in the last 30 years. Like any new idea, Devops is creating some controversy and there are many discussions about what is wrong with Devops. In fact, many of the recent questions about Devops sound very similar to some of the questions that were being asked about Clouds two years ago. A lot of those old Cloud questions have now been answered; however, many still remain open. I believe this is exactly the path that the recent “Devops” questions are following. I have spoken and interviewed a lot of the thought leaders in the Devops movement. Here are some of my thoughts based on

- Culture
- Automation
- Measurements
- Sharing

Still the best distillation of devops

Co-evolution of tools and practice

Advancement in one begets the other in sociotechnical systems

“Other people’s computers”

Towards well defined APIs

Why all the fuss?

24x

faster recovery from failures

3x

lower change failure rate

22%

less time spent on unplanned work and rework

50%

less time remediating security issues.

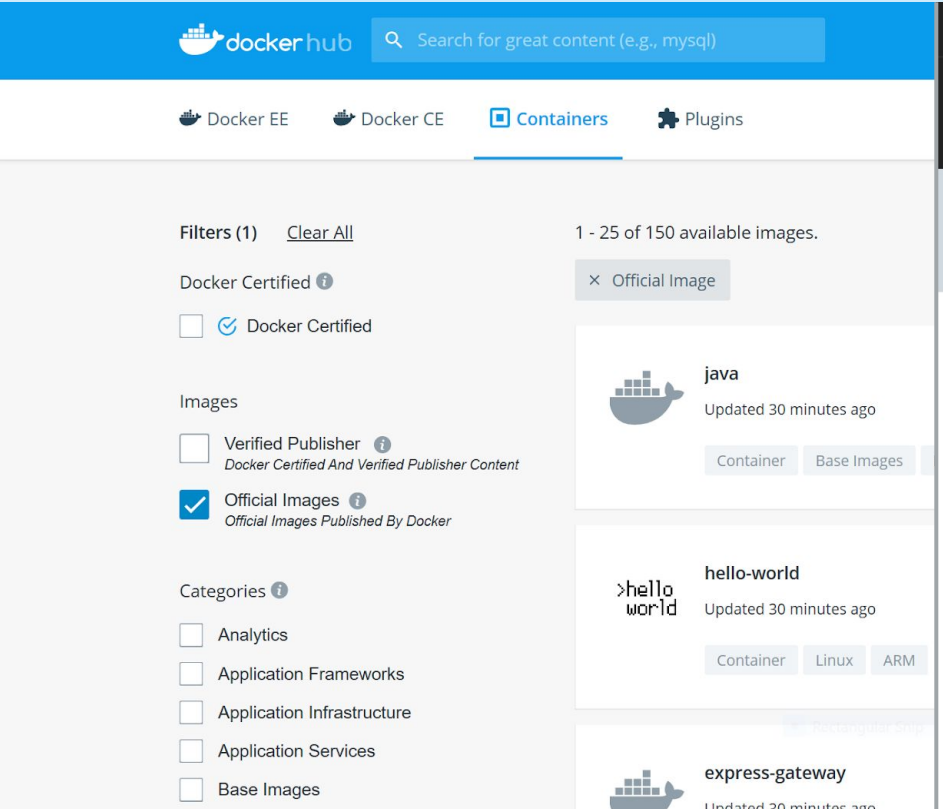
From puppet State of Devops report 2017

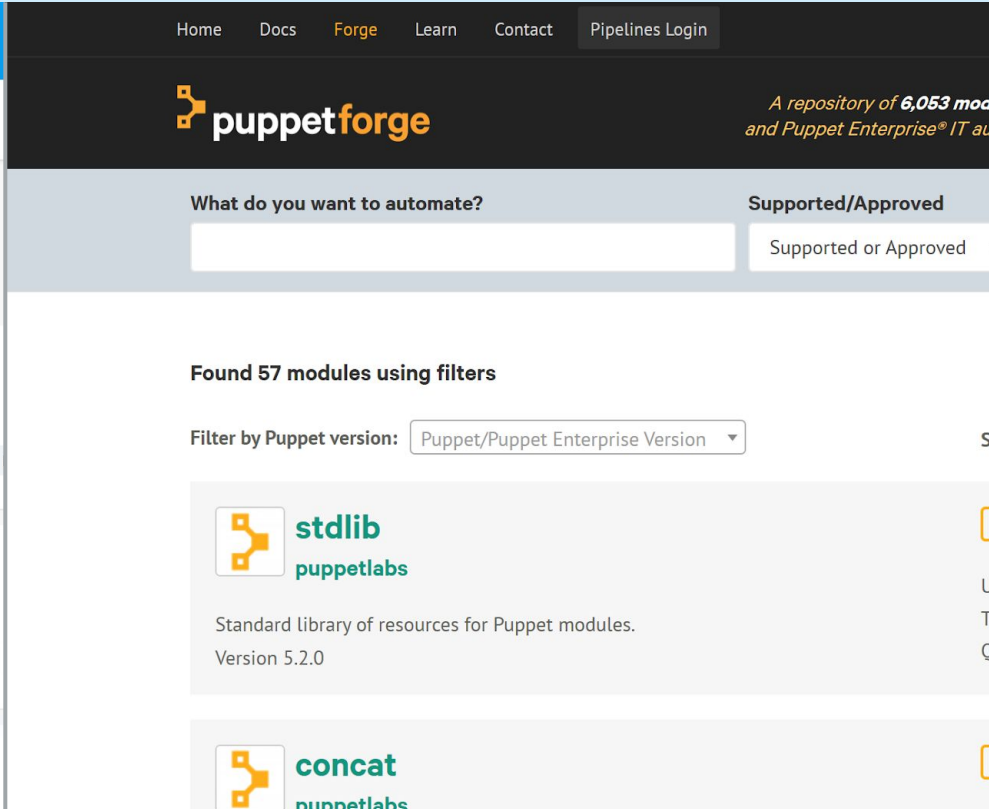
What did we learn?

Not everyone needs to be an expert

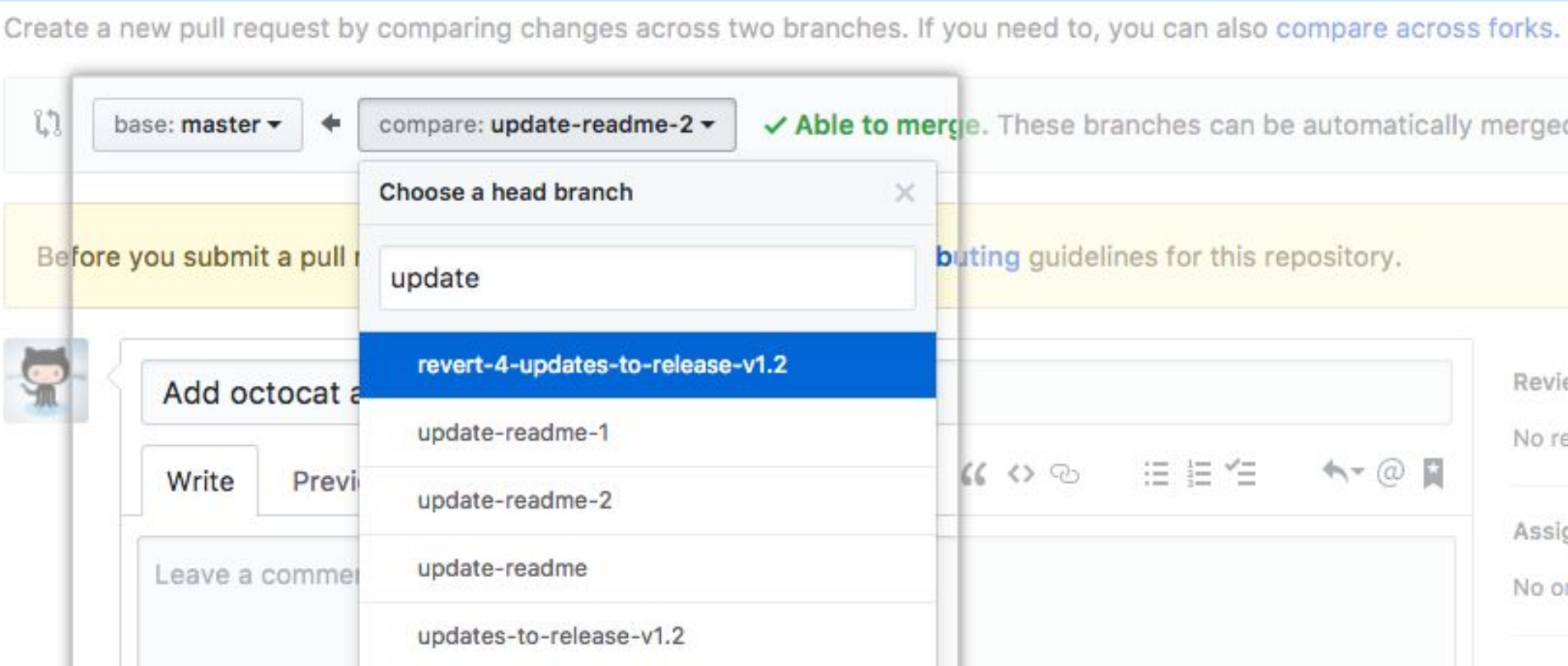
Content reuse scales

The utility of a marketplace





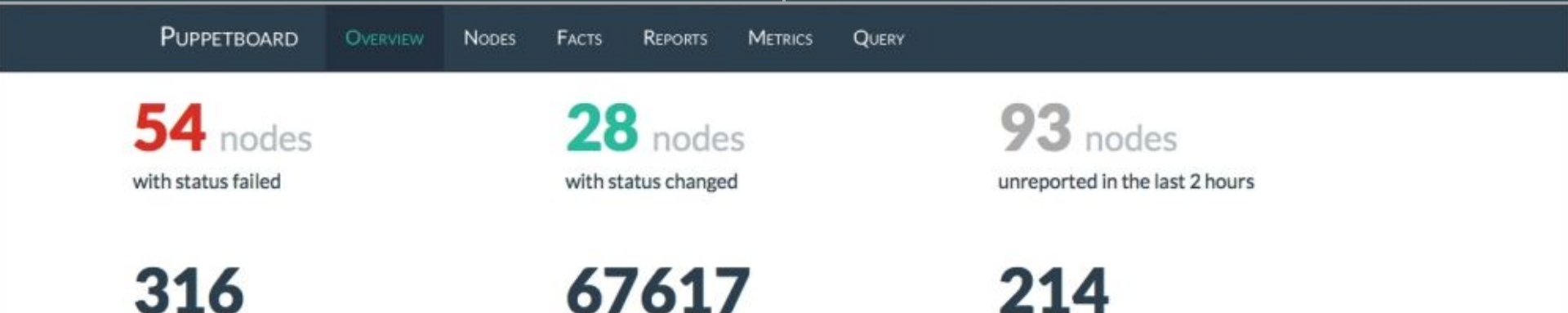
Version control as change control



Shared tooling emerges

```
$ puppet-lint /etc/puppet/modules
foo/manifests/bar.pp - ERROR: trailing
whitespace found on line 1
apache/manifests/server.pp - WARNING: variable
not enclosed in {} on line 56
...
```

```
require 'chefspec'

describe 'file::delete' do
  let(:chef_run) { ChefSpec::SoloRunner.new(platform: 'ub

  it 'deletes a file' do
    expect(chef_run).to delete_file('/tmp/explicit_action
    expect(chef_run).to_not delete_file('/tmp/not_explici
  end
end
```



The importance of community

Parallels with security

Lots of spreadsheets

And lots of manual processes

Silos abound

“Low performers take weeks to conduct security reviews and complete the changes identified.”

From DORA DEVOPS RESEARCH & ASSESSMENT Accelerate State of Devops report

“Probably the security teams would rather the policy docs not be published? Or doesn’t make sense to OSS it”

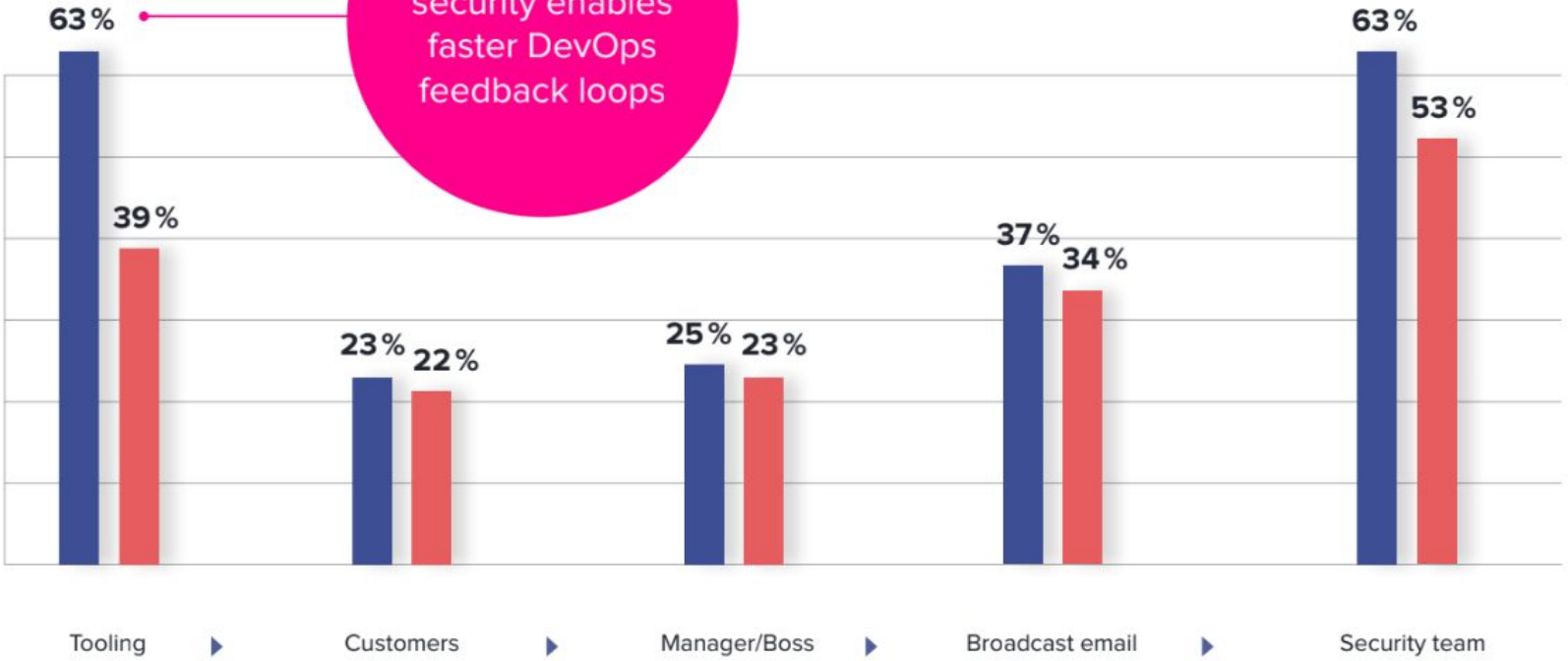
Vincent Janelle, @randomfrequency

“The only way to really ensure software security is to put automated security controls in the pipelines”

Juanjo Torres, BBVA

From sonatype DevSecOps Community Survey 2019

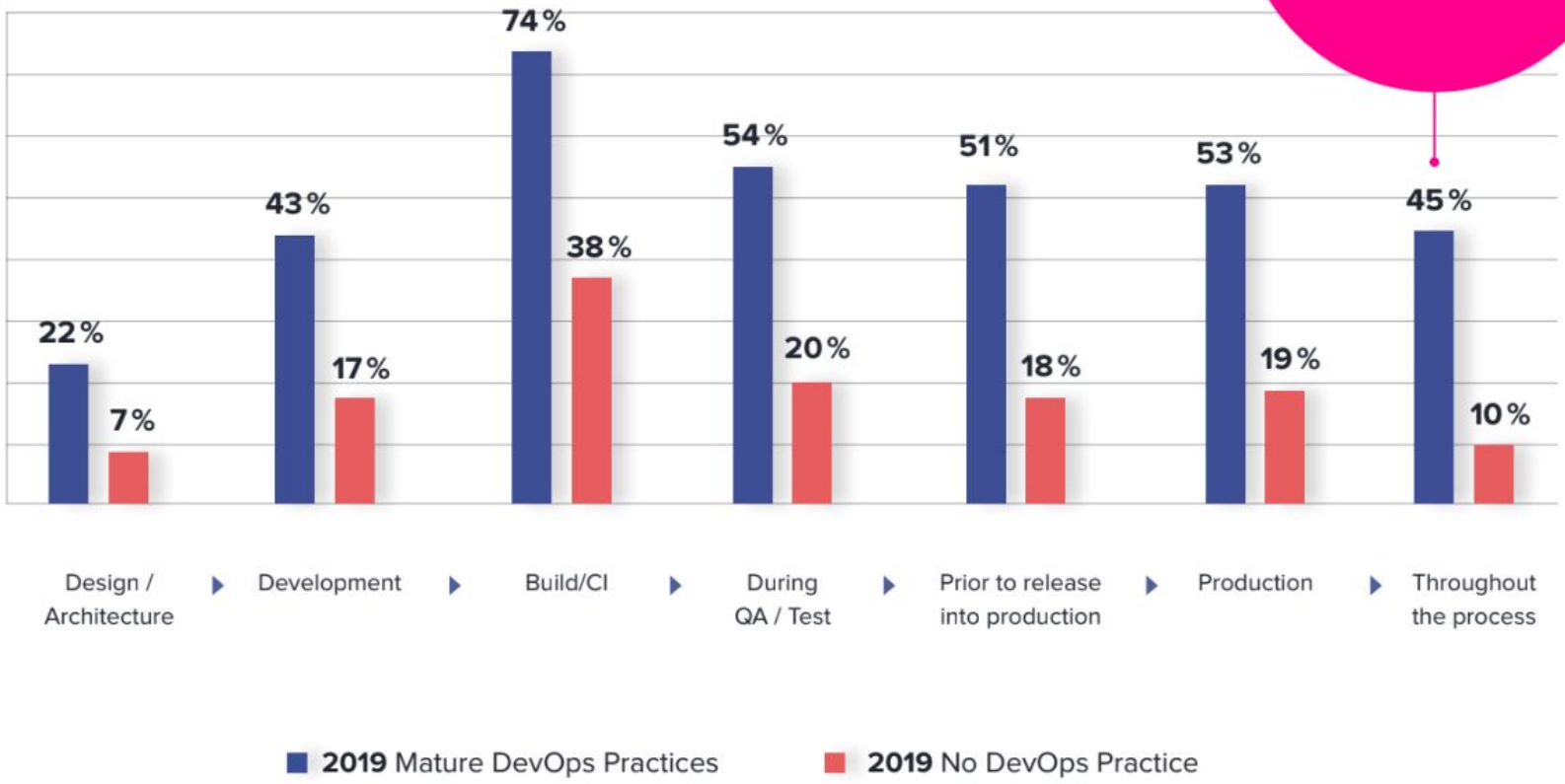
Automating security enables faster DevOps feedback loops



2019 Mature DevOps Practices

2019 No DevOps Practice

Mature DevOps practices are 350% more likely to integrate automated security.



Security automation is not new

Neither was using code to manage servers, or automated deployments or working across silos

“Elite performers build security in and can conduct security reviews and complete changes in days.”

From DORA DEVOPS RESEARCH & ASSESSMENT Accelerate State of Devops report

Security as policy management

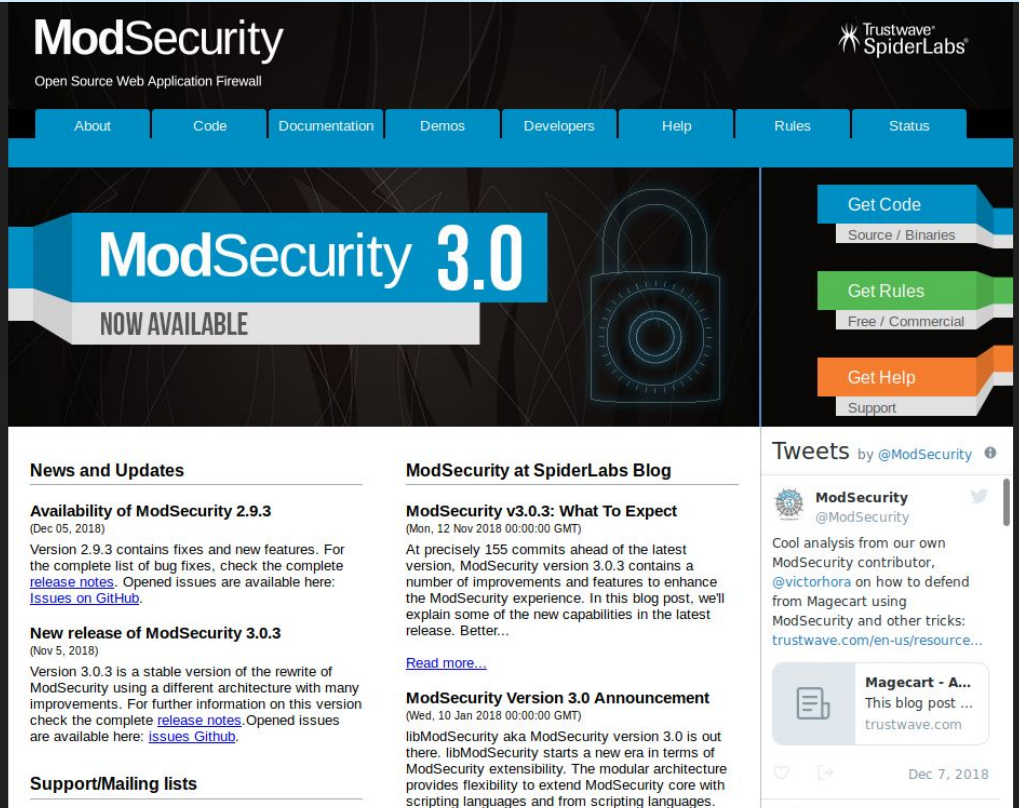
Part of security is the definition and implementation of controls

How do we get to policy as code?

By which we mean controls which are machine readable and machine enforceable

Security tooling examples

ModSecurity: Web Application Firewall



Write application firewall rules in code

```
# User login password
SecRule REQUEST_FILENAME "@endsWith /wp-login.php" \
     "id:9002100,\
     phase:2,\
     pass,\
     t:none,\
     nolog,\
ctl:ruleRemoveTargetByTag=CRS;ARGS:pwd"
```

OWASP Core Rule Set

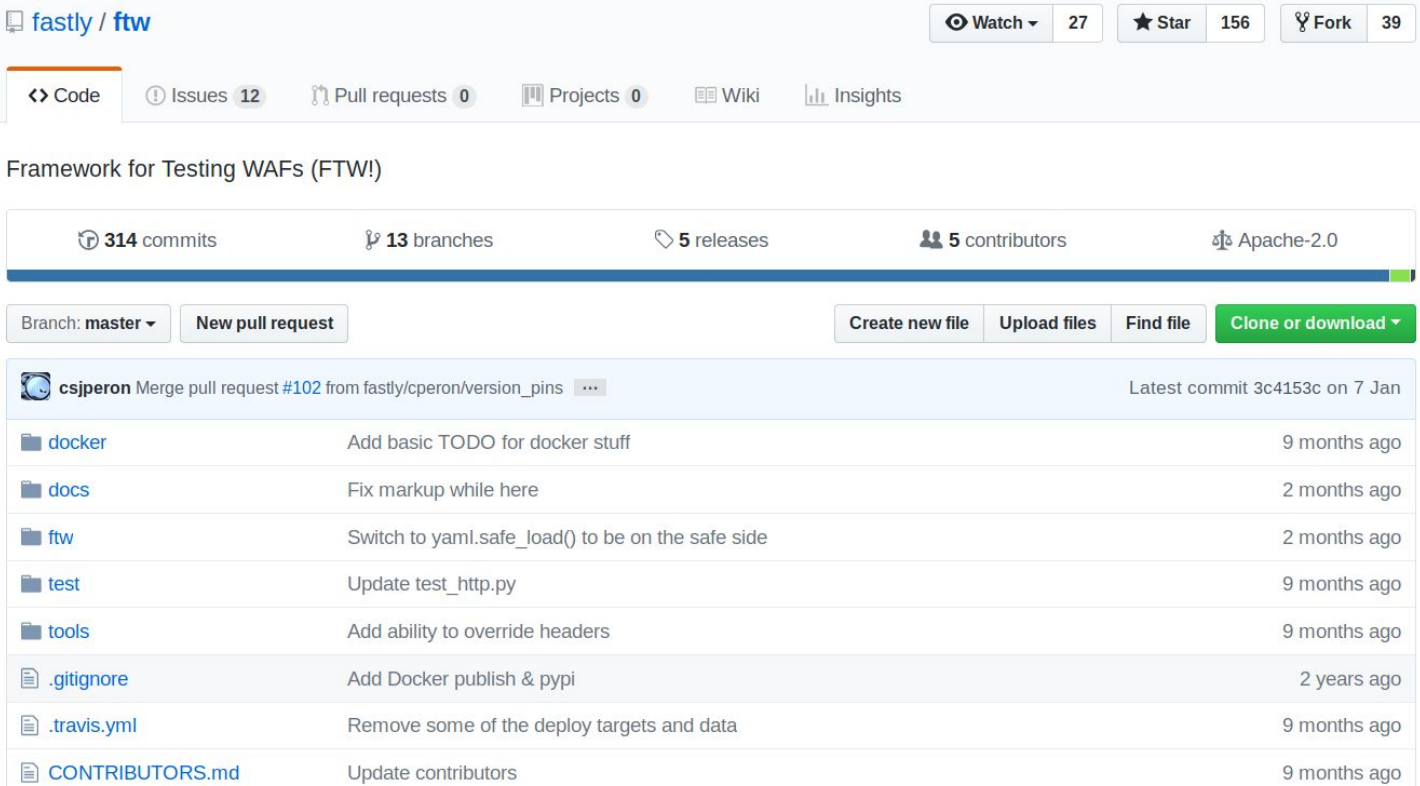
The **OWASP ModSecurity Core Rule Set (CRS)** is a set of generic attack detection rules for use with **ModSecurity** or compatible web application firewalls. The CRS aims to protect web applications from a wide range of attacks, including the OWASP Top Ten, with a minimum of false alerts. The CRS provides protection against many common attack categories, including:

- SQL Injection (SQLi)
- Cross Site Scripting (XSS)
- Local File Inclusion (LFI)
- Remote File Inclusion (RFI)
- PHP Code Injection
- Java Code Injection **New in CRS 3.1!**
- HTTPoxy
- Shellshock
- Unix/Windows Shell Injection
- Session Fixation
- Scripting/Scanner/Bot Detection
- Metadata/Error Leakages

Current version: 3.1.0 Nov 28, 2018

Latest blog posts:

Some ecosystem tooling

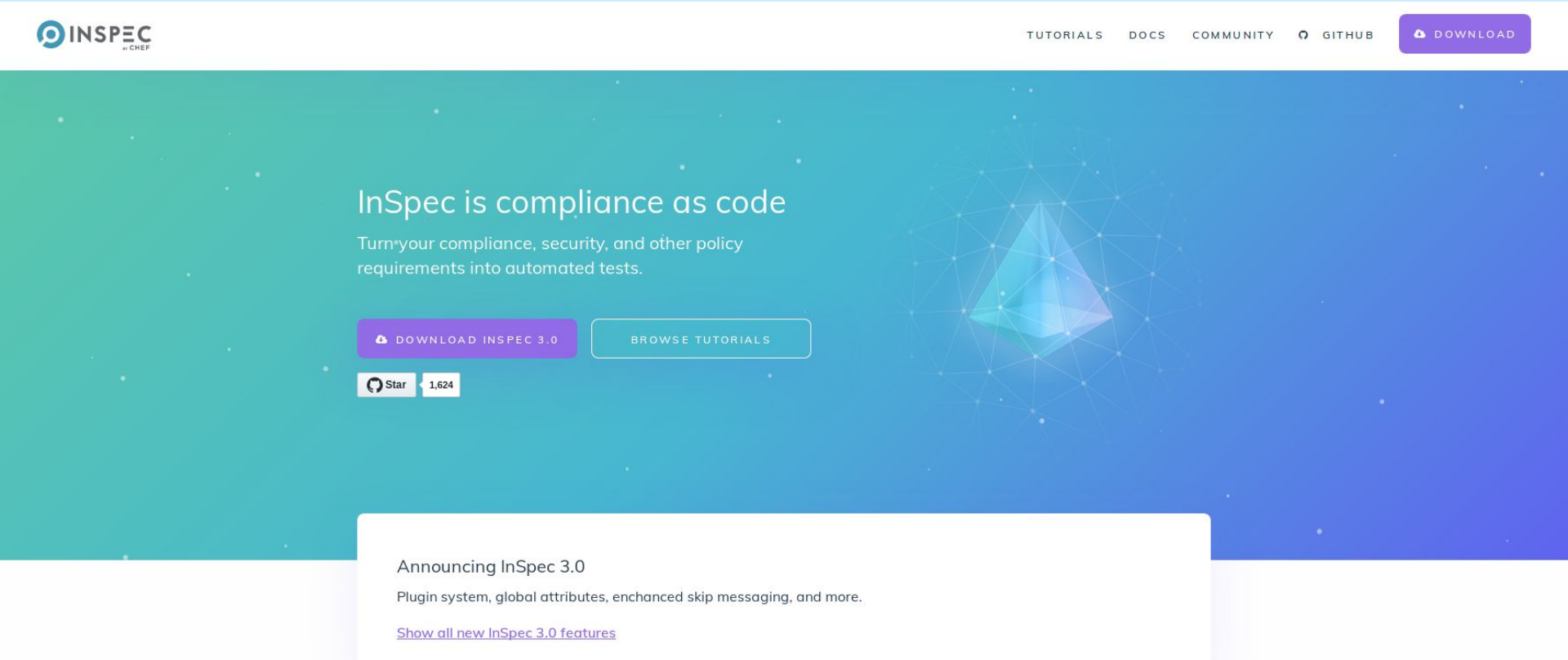


But...

Some observations about ModSecurity

- ✘ A somewhat terse DSL
- ✘ Terse *may* be an understatement
- ✔ Some shared content, but no community sharing
- ✘ Tied to Apache, and more recently Nginx
- ✘ Rule based vs heuristic based

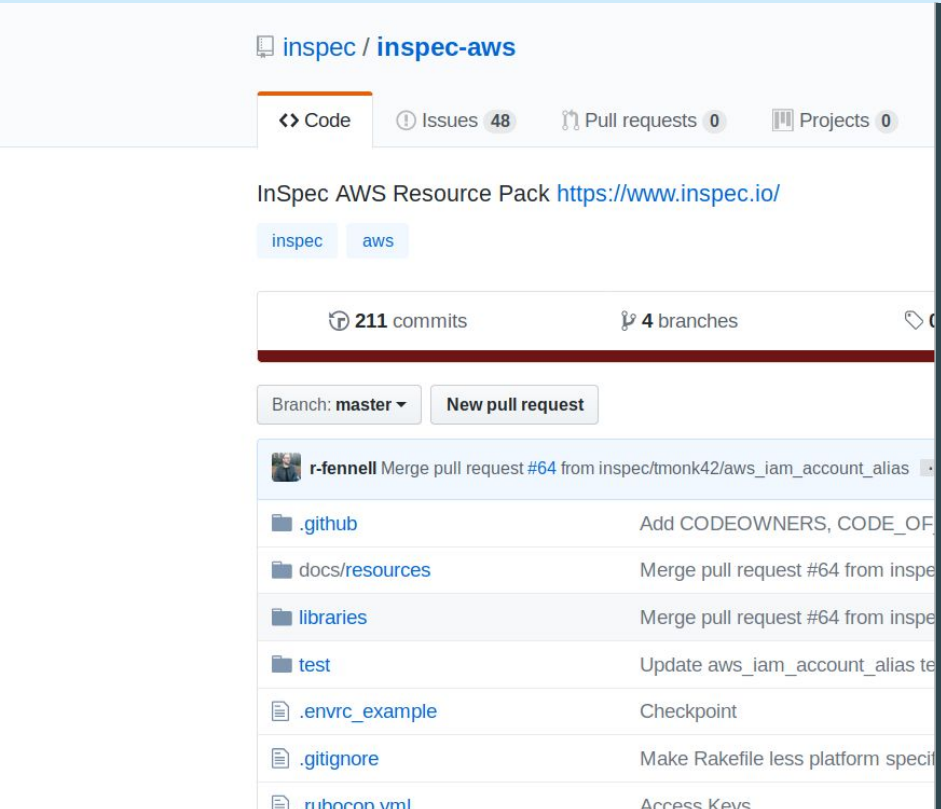
Inspec: compliance as code



Helpers for writing controls with rspec

```
control 'cis-ubuntu-lts-5.4.4' do
  impact 0.7
  title 'Ensure default user umask is 027 or more restrictive'
  desc 'The default umask determines the permissions of files created by users.'
  describe file('/etc/bash.bashrc') do
    its('content') { should match /^umask 027/ }
  end
  describe file('/etc/profile') do
    its('content') { should match /^umask 027/ }
  end
end
```

Extended for other types of policy




```
describe aws_eks_cluster('my-eks') do
  it { is_expected.to exist }
  expect(subject.status).to eq 'ACTIVE'
  expect(subject.subnet_counts).to be > 1
end


describe aws_s3_bucket('test_bucket') do
  it { is_expected.to exist }
  it { is_expected.not_to be_public }
end
```

A supermarket of shared profiles

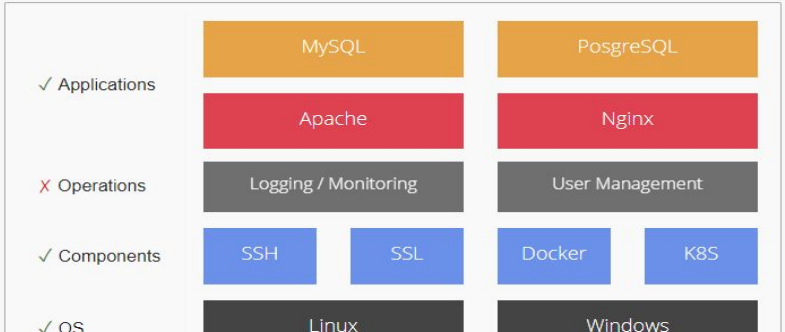
```
$ inspec supermarket profiles

———————————————————— Available profiles: ————————————————————

• Ansible Fashion Police brucellino/ansible-fashion-police
• apache2-compliance-test-tthompson thompsontelmate/apache2-compliance-test-tthompson
• Apache DISA STIG som3guy/apache-disa-stig
• Black Panther brucellino/black-panther
• chef-alfresco-inspec-mysql alfresco/chef-alfresco-inspec-mysql
• chef-alfresco-inspec-tomcat alfresco/chef-alfresco-inspec-tomcat
• chef-client-hardening sliim/chef-client-hardening
• CIS Distribution Independent Linux Benchmark dev-sec/cis-linux-benchmark
• CIS Docker Benchmark dev-sec/cis-docker-benchmark
• CIS Kubernetes Benchmark dev-sec/cis-kubernetes-benchmark
• CVE-2016-5195 ndobson/cve-2016-5195
• DevSec Apache Baseline dev-sec/apache-baseline
• DevSec Linux Baseline dev-sec/linux-baseline
• DevSec Linux Patch Baseline dev-sec/linux-patch-baseline
```

A community building content

devops + security
Server Hardening Automation

Overview

DevSec Hardening Framework Baselines



Easy to use without expertise

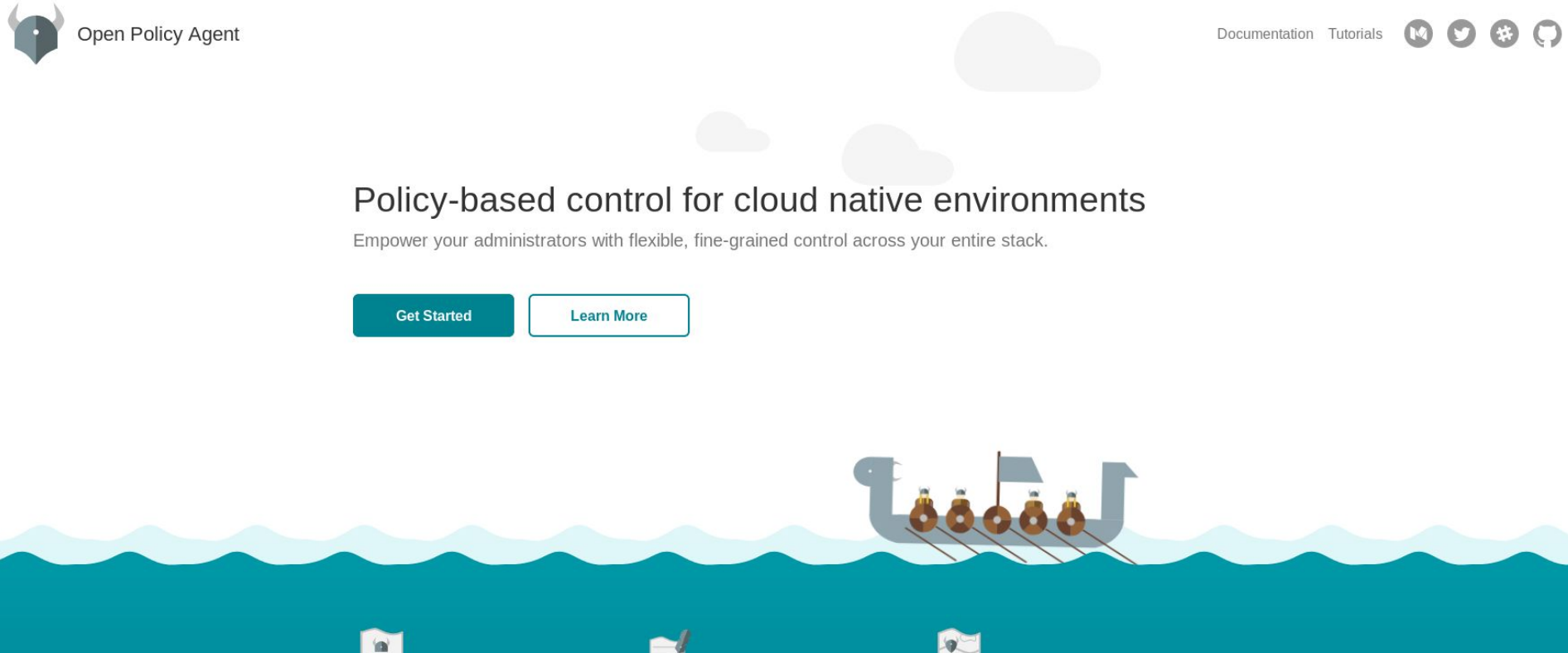
```
$ inspec supermarket exec dev-sec/linux-baseline

     ×  Kernel Parameter kernel.core_pattern value should match /^\/.*/
     expected "|/usr/share/apport/apport %p %s %c %d %P" to match /^\/.*/
     Diff:
     @@ -1,2 +1,2 @@
     -/^\/.*/
     +"|/usr/share/apport/apport %p %s %c %d %P"

  ✔  sysctl-32: kernel.randomize_va_space
     ✔  Kernel Parameter kernel.randomize_va_space value should eq 2
  ✔  sysctl-33: CPU No execution Flag or Kernel ExecShield
     ✔  /proc/cpuinfo Flags should include NX


Profile Summary: 25 successful controls, 28 control failures, 1 control skipped
Test Summary: 67 successful, 42 failures, 2 skipped
```

But...

Some observations about Inspec

- ✘ Ruby and programming language fashion
- ✔ High-quality shared content
- ✔ Chef supermarket as a central repository
- ✘ No tools for non-programmers

Open Policy Agent



Open Policy Agent allows you to express policies in a high-level declarative language that promotes safe, fine-grained logic.

Prohibit changes to AWS IAM rules

```
package terraform.analysis

import input as tfplan

default authz = false
authz {
    not touches_iam
}

touches_iam {
    all := instance_names["aws_iam"]
    count(all) > 0
}

# list of all resources of a given type
instance_names[resource_type] = all {
    resource_types[resource_type]
    all := [name |
```

Block images from other registries

```
package admission

import data.k8s.matches

deny[{
     "id": "container-image-whitelist",          # identifies type of violation
     "resource": {
       "kind": "pods",                           # identifies kind of resource
       "namespace": namespace,                   # identifies namespace of resource
       "name": name                            # identifies name of resource
     },
     "resolution": {"message": msg},         # provides human-readable message to display
}] {
     matches[["pods", namespace, name, matched_pod]]
     container = matched_pod.spec.containers[_]
     not re_match("^registry.acmecorp.com/.+$", container.image)
     msg := sprintf("invalid container registry image %q", [container.image])
}
```

Test Kubernetes Helm charts

```
deny[msg] {
  input.kind = "Deployment"
  not input.spec.template.spec.securityContext.runAsNonRoot = true
  msg = "Containers must not run as root"
}
```

```
$ helm opa CHART
Processing file deployment.yaml
Violations:
- Containers must not run as root
Processing file ingress.yaml
Processing file service.yaml
===
Result: Chart is not compliant
```

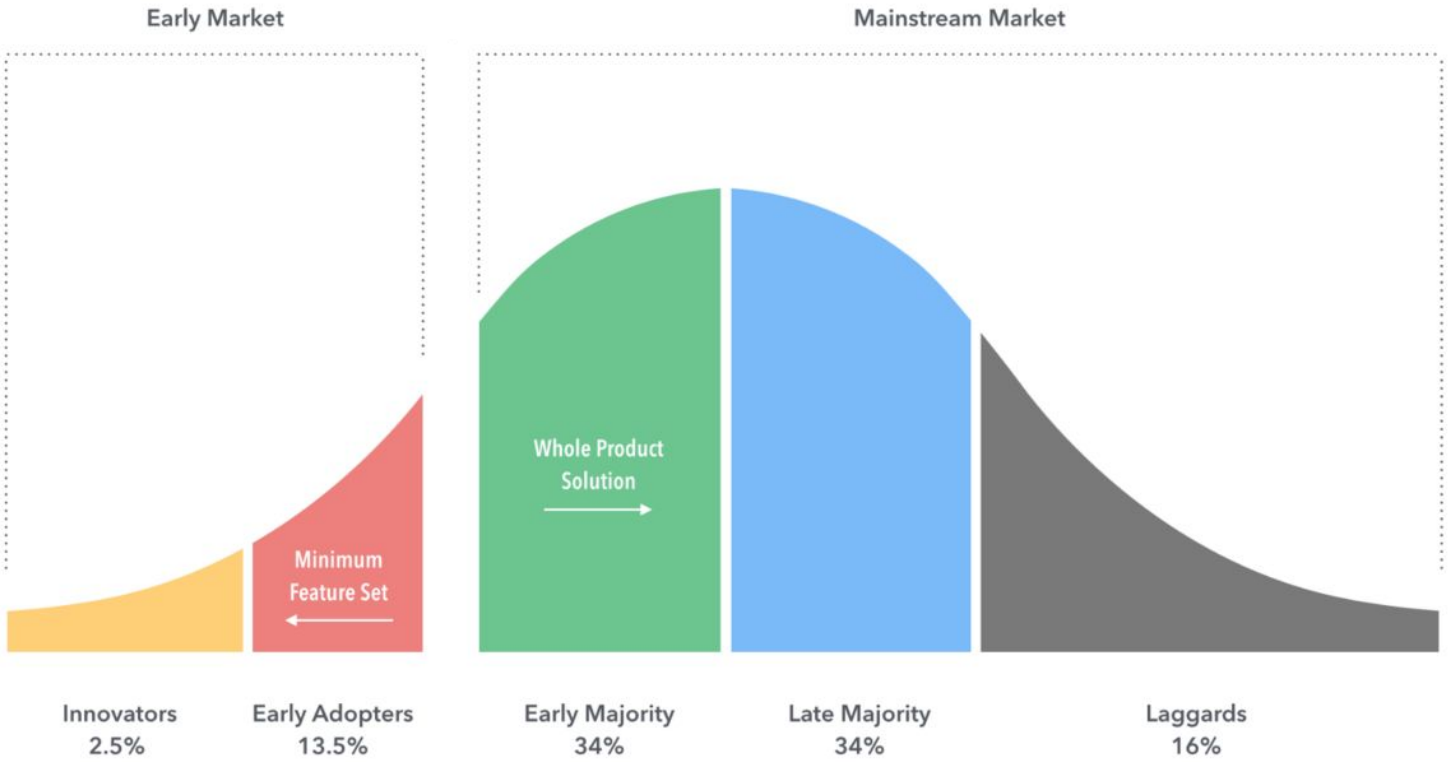
But...

Some observations about Open Policy Agent

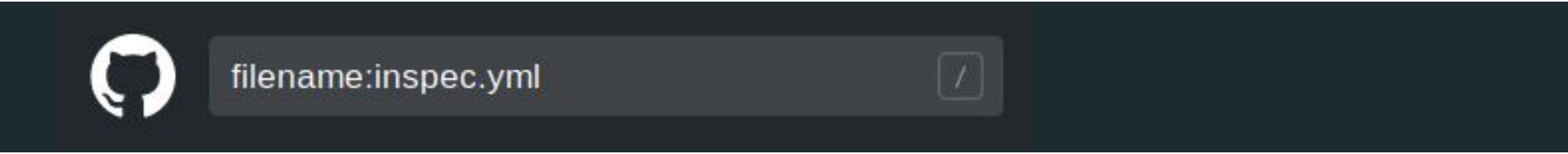
- New
- ✔ Built-in tools for testing
- ✔ Widely applicable to different problems
- ✘ Limited examples outside use with Kubernetes
- ✘ No built-in sharing or central repository (yet)

Conclusions

Crossing the chasm



A way to go still



Puppet manifests	1.4million
Dockerfiles	1.16million
Compose files	229,000
Helm Charts	36,000

ModSecurity configs	3207
Inspec profiles	1736
.rego files	361

Policy as code is a powerful idea

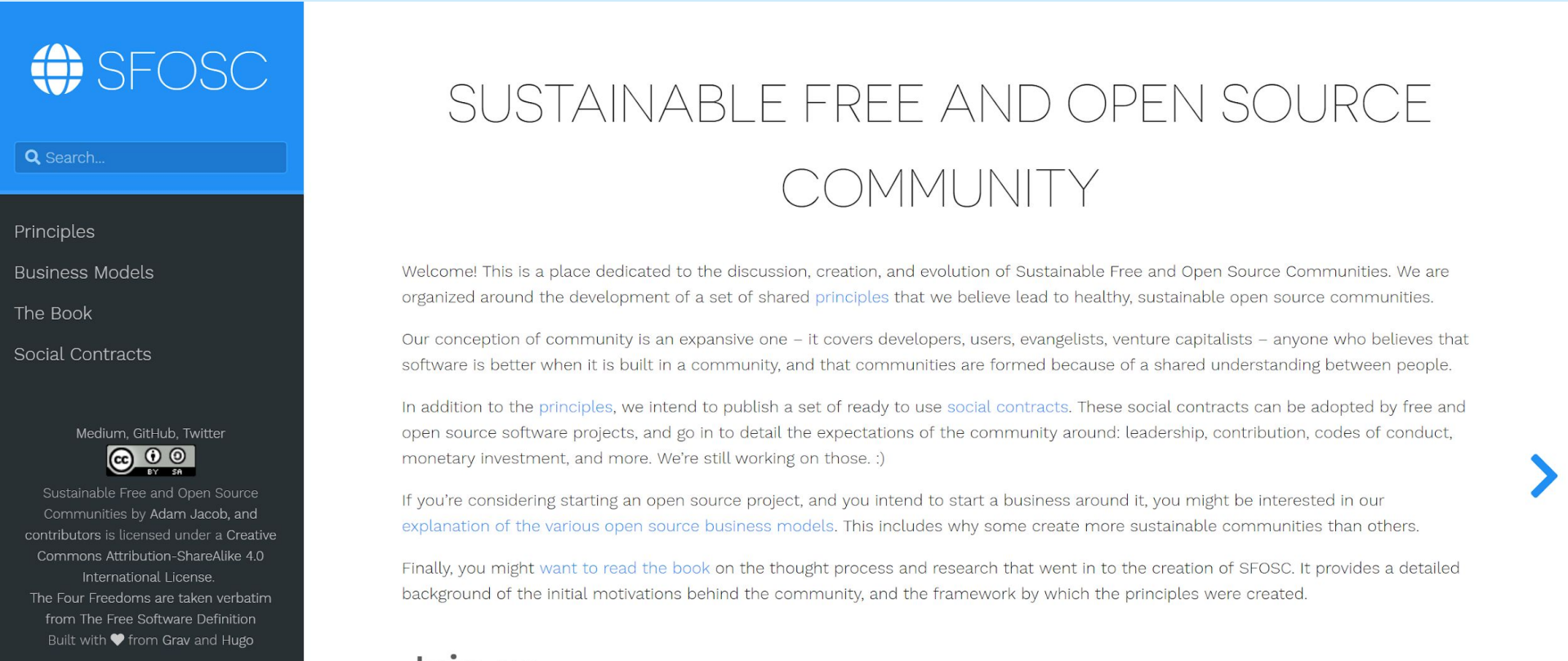
But we’re not there yet in terms of tools and ecosystems

For tool builders

Build for community

Don’t just write code, think about enabling an ecosystem

Follow Adam and SFOSC



For end users

Build for sharing

Blog posts, examples, tools, talks, everything helps

Put this in your own context

Emphasise sharing, reuse and community when adopting new tools and practices in your own organisation

Thanks

and any questions?