



Take control of your “DevOps Dumping Ground”



Cloud adoption has shifted the center of gravity for automation

Today

- Model/task-driven automation
 - OSs, hardware, apps, VMs, hypervisor
 - Configuration management is king
-

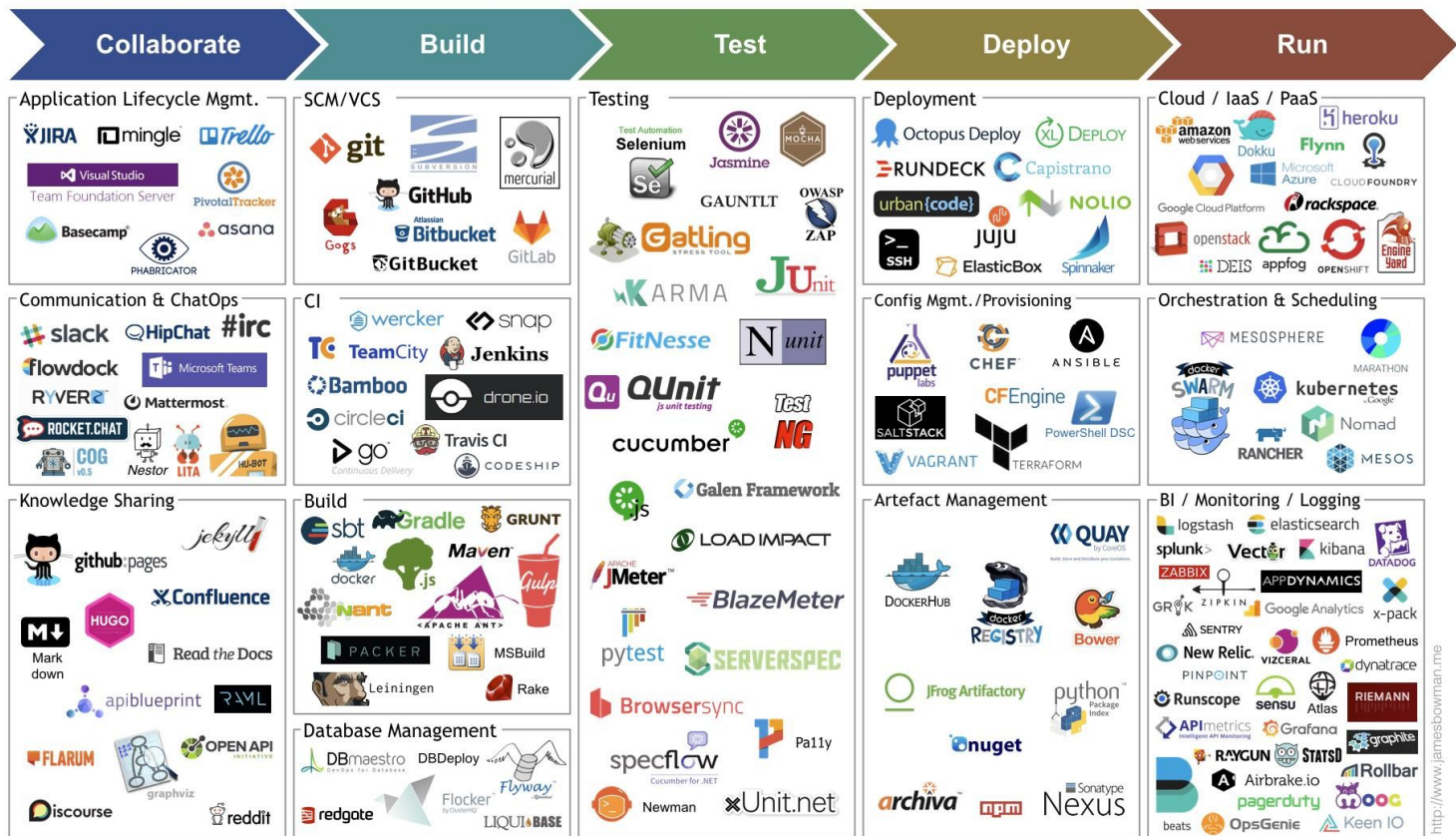
Tomorrow

- Event-driven orchestration
 - APIs, services, serverless, K8s, container platforms
 - Configuration management is a smaller part of the infra stack
-

On-prem

Cloud

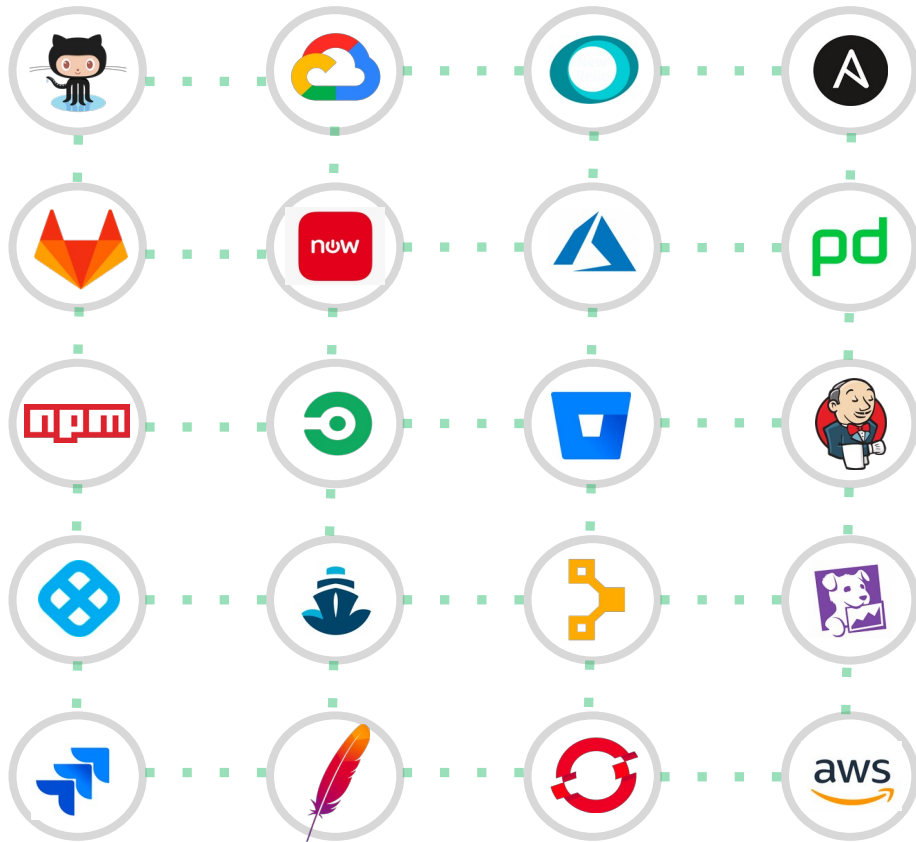
Lesson 1: CD is Hard. Really Hard.



source: jamesbowman.me

<http://www.jamesbowman.me>

Lesson 2: Events Are Everywhere.



Lesson 3: 'Ware The DevOps Dumping



source: tirefi.re

Listen to events
from the services
that you already use

Cloud events

Incidents

Tickets

Git events

Monitoring



Triggers

Secrets

Workflows

AI/ML/data

Orchestrate actions
across downstream
tools and services

K8s

Serverless

Cloud storage

CI/CD Tools

Config Mgmt



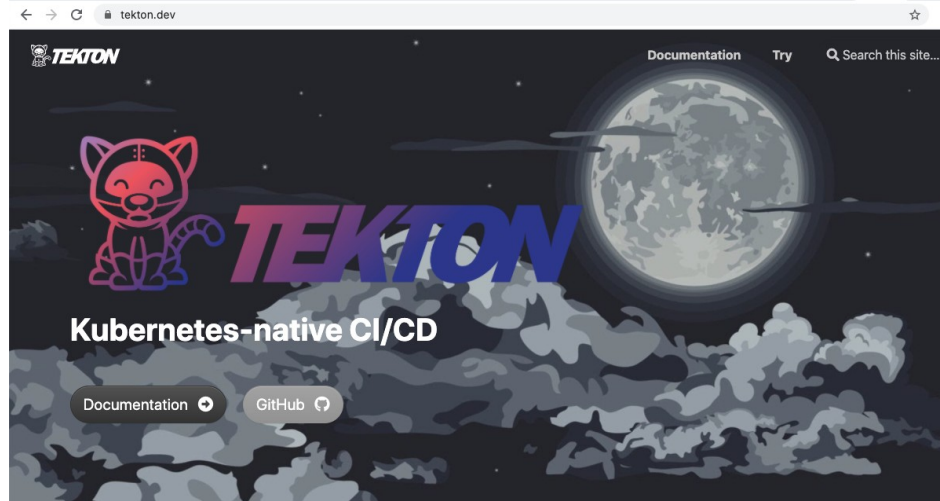
TEKTON

Tekton

- Kubernetes-native CI/CD
- Powerful, open-source workflow engine
- *Tasks* are individual, containerized steps
- *Pipelines* stitch together tasks into a workflow

Tekton pipelines help us build and traverse a graph of the Actions and Queries that make up the workflows for Relay!

tekton.dev



Tekton is a powerful and flexible open-source framework for creating CI/CD systems, allowing developers to build, test, and deploy across cloud providers and on-premise systems. **Get started with interactive tutorials.**

```
tasks:
- name: build-go-app-task
  taskRef:
    name: build-go-app-task
  ...

# And apply it to your Kubernetes cluster as well:
kubectl apply -f pipeline.yaml
# And it is done :)

# You can run your pipeline manually, or trigger it every time an
# event arrives, such as one from a GitHub webhook
# Tekton has a dashboard and a CLI as well
# Learn more about Tekton at
# tekton.dev/docs
# or play with one of our interactive tutorials at
# tekton.dev/try
```



Standardization

Tekton standardizes CI/CD tooling and processes across vendors, languages, and deployment environments. It works well with Jenkins, Jenkins X, Skaffold, Knative, and



Built-in best practices

Tekton lets you create CI/CD systems quickly, giving you scalable, serverless, cloud native execution out of the box.



Maximum flexibility

Tekton abstracts the underlying implementation so that you can choose the build, test, and deploy workflow based on your
 runtime requirements.

Relay Taxonomy

Trigger steps are based on cloud events, git events, monitoring alerts, tickets, and incidents – like a Splunk alert, a Cloudwatch alert, a GitHub PR being merged, etc.

Action steps are reusable, modular, and composable-- things like destroying old EC2 instances, provisioning environments with Terraform, or clearing unattached EBS volumes.

Query steps let you break out of a workflow to wait for an external process like manual approval or a callback from another system.

The screenshot displays the AWS Step Functions console for a workflow named 'ec2-grim-reaper'. The workflow is defined in the 'Graph' view, showing a sequence of three steps: an 'Automated trigger' (schedule), an 'AWS Step' (describe-instances), and a 'Step' (filter-instances). The right-hand panel provides details for the 'Scheduled trigger', including the next kickoff time and the connection to the 'my-aws-account'.

Workflows > ec2-grim-reaper

ec2-grim-reaper

Reap all untagged instances

NEW No runs yet **Created at** 6/23/2020, 4:45:09 PM

Graph **Code** **Run history** **Settings**

Automated trigger

schedule

Repeats Every hour
Cron 0 * * * *

Step

describe-instances

Image aws-ec2-step-instan...
Secrets NONE
Parameters 1

Step

filter-instances

Image core:latest-python
Secrets NONE
Parameters NONE

Scheduled trigger

This workflow uses a scheduled trigger to start automatically. [Configuring a scheduled trigger.](#)

Next kickoff: 6/23/2020, 5:00:00 PM
Repeats: Every hour
Cron: 0 * * * *

Connections

Connections are how you authenticate with the integrations used by the workflow. They are stored globally for re-use in other workflows. [Learn more.](#)

my-aws-account

Secrets

Secrets are stored securely in our vault. Each secret is local to a specific workflow. [Learn more about our secrets.](#)

No secrets specified

[Add secret](#)

Parameters

Parameters dynamically pass external data to workflow step. You can also provide them manually for manual runs. [Learn more about parameters.](#)

Workflows-as-Code-YAML

Parameters prompt the user for data input at runtime to customize the behaviour of the workflow. Values can be supplied interactively, via API, or on CLI.

Secrets are stored securely server-side using Hashicorp Vault, referenced in a workflow using the "!!Secret" tag.

Output tags enable data from previous steps to be used by ones that come later.

```
1  description: Automatically clean up EC2 instances that do not have the correct tags
2
3  parameters:
4    region:
5      description: The AWS region to run in
6      default: us-east-1
7    terminationDateTag:
8      description: The name of the tag to use for determining the termination date
9      default: termination_date
10   lifetimeTag:
11     description: The name of the tag to use for determining the lifetime
12     default: lifetime
13   dryRun:
14     description: True if this workflow should only print the resources it would delete
15     default: 'true'
16
17  steps:
18    - name: describe-instances
19      image: projectnebula/ec2-describe-instances
20      spec:
21        aws: &aws
22        region: !Parameter region
23        accessKeyID: !Secret aws.accessKeyID
24        secretAccessKey: !Secret aws.secretAccessKey
25    - name: filter-instances
26      image: projectnebula/core:latest-python
27      spec:
28        terminationDateTag: !Parameter terminationDateTag
29        lifetimeTag: !Parameter lifetimeTag
30        instances: !Output [from: describe-instances, name: instances]
31        inputFile: workflow/filter-instances.py
32    - name: terminate-instances
33      when:
34        - !Fn.equals [!Parameter dryRun, 'false']
35      image: projectnebula/ec2-terminate-instances
36      spec:
37        aws: *aws
38        instanceIDs: !Output [from: filter-instances, name: instanceIDs]
```


Open-source Integration Ecosystem

A library of useful integrations that cover the most common use cases and external services with sample workflows and well-documented steps that let you remix and combine however you want.

Low-friction interoperability if you have existing containers that you want to use in Relay workflows, they can drop right in alongside purpose-built integrations.

Easy step authoring if you've built something that other people might find useful or want to contribute to existing steps

**github.com/relay-
integrations**



Github

GitHub is the predominant Git repository hosting service. Host your private or public git repositories here. I'm telling you things you already know.

<https://github.com/relay-integrations/github>

Overview

README

Coming soon features




Our library of integrations and workflows is under active development by the Relay team. Features marked as "coming soon" are intended to provide you with an idea of our product roadmap. Please reach out to us with any [feedback or suggestions](#) you have.

Workflows (2)

Do more with Github. Relay helps you connect Github with other apps to put your DevOps on auto pilot.

Deploy to Kubernetes with Helm



When a PR is merged, upgrade helm chart on kubernetes cluster



»

Provision infrastructure with Terraform


When a PR is merged, run Terraform.



COMING SOON

Triggers (6)

Relay listens for events and uses them to automatically trigger the rest of your workflow.



When a new PR is raised

Triggers when a new pull request is raised against a defined repo.

[View on Github](#)

DEMO

puppetlabs/relay-core

Kubernetes-based execution backend for running workflows on top of Tekton.

The screenshot shows the GitHub repository page for `puppetlabs/relay-core`. The repository has 44 pull requests, 6 stars, and 1 fork. The main branch is `development` with 2 branches and 0 tags. A recent pull request #159 by `hunner` is highlighted, merged 2 days ago. The repository contains a file tree with folders like `cmd`, `examples`, `manifests/resources`, `pkg`, `scripts`, `tests`, and `vendor`, and files like `.gitignore`, `.travis.yml`, `CODEOWNERS`, `CONTRIBUTING.md`, `LICENSE`, `README.md`, `go.mod`, `go.sum`, and `tools.go`. The `README.md` file is open, showing the title `Relay Core` and the text: "This repository contains the Kubernetes-based execution backend for [Relay](#)." The `Installation` section mentions providing a Helm chart or Kustomizations in the future. The `Requirements` section is also visible.

Search or jump to... [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

`puppetlabs/relay-core` [Unwatch](#) 44 [Star](#) 6 [Fork](#) 1

[Code](#) [Pull requests](#) [Actions](#) [Security](#) [Insights](#) [Settings](#)

[development](#) 2 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

hunner Merge pull request #159 from puppetlabs/patch/latest-cil-deploy ✓ c334ae7 2 days ago 534 commits

cmd	Add labels to the metrics descriptor	23 days ago
examples	Prepare for open sourcing	28 days ago
manifests/resources	Propagate tenant changes to dependent objects	11 days ago
pkg	Minor updates to event key handling	3 days ago
scripts	Update cil deploy to latest and fix image tag secret	2 days ago
tests	Minor updates to event key handling	3 days ago
vendor	Fix vendor	24 days ago
.gitignore	Prepare for open sourcing	28 days ago
.travis.yml	Prepare for open sourcing	28 days ago
CODEOWNERS	Prepare for open sourcing	28 days ago
CONTRIBUTING.md	Prepare for open sourcing	28 days ago
LICENSE	Prepare for open sourcing	28 days ago
README.md	Clarify project description in README	28 days ago
go.mod	Fix vendor	24 days ago
go.sum	Prepare for open sourcing	28 days ago
tools.go	WIP: Remove dependency on old k8s.io/code-generator, auto-gener...	4 months ago

README.md

Relay Core

This repository contains the Kubernetes-based execution backend for [Relay](#).

Installation

In the near future, we hope to provide a Helm chart or Kustomizations to let you easily deploy this project to your own cluster. In the meantime, please reach out to us on [Slack](#) in the `#relay` channel and we'll help you out!

Requirements

About

The Kubernetes-based execution backend for Relay

[relay.sh](#)

[Readme](#)

[Apache-2.0 License](#)

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 7

Languages

Go 99.1% Other 0.9%



Available now!

Sign up at <https://relay.sh/> !

Join us in Slack for Q&A: #relay channel

Check out our Github Repos:

Integrations: **[github.com/relay-integrations](https://github.com/puppetlabs/relay-integrations)**

Relay core: **github.com/puppetlabs/relay-core**

Thank you!

Melissa Sussmann | melissa.sussmann@puppet.com

Kenaz Kwa | kenaz@puppet.com