Shifting Left With Cloud Native CI/CD

QCon San Francisco 2019
Christie Wilson
Lead Engineer @ Google
Tekton Co-Creator

@bobcatwilson
Make Mistakes!
Rockstars
Heroes
Ninjas
...
Janitors
Can’t have success without failure
Cloud Native CI/CD can make failure easier and cheaper
Agenda

- What is Cloud Native?
- What is Cloud Native CI/CD?
- What is shifting left?
- What is Tekton?
- Demo!
- What’s next for Tekton?
What is Cloud Native?
What is Cloud Native?

- Microservices in containers
- Dynamically orchestrated
- Optimized resource utilization

Images / Containers

Kubernetes
wut?
Images/Containers

- A binary and all of its dependencies
- Containers share an OS
- Run as resource isolated processes

Non-cloud native equivalent:

```bash
~ docker run hello-world
```

```bash
~ pip install hello-world
~ python my-hello-world.py
```
Kubernetes

- Platform for managing containers
- Abstracts away the underlying hardware
  - Computing
  - Networking
  - Storage

Non-cloud native equivalent:

- People + wikis
- Build your own!
- Machine under your desk
A Few Kubernetes Concepts
Pod + Node

- **Node** = machine you run stuff on
  - Could be a physical machine or VM
- **Pod** = 1 or more containers
  - Run on the same pod

Non-cloud native equivalent:

- Node -> Host machine
- Pod -> Bunch of stuff you run together
  - e.g. My python process + supervisord + some proxy service
Yet Another Markup Language
Moar complicated
2008 complexity
Cloud Native complexity
Cloud Native complexity

There is a lot to grok here and it’s constantly changing!

- Start with a webserver
- End up with Kubernetes + Istio + Knative + Helm + Spinnaker*

* **Istio**: Stuff for services to talk to each other
  **Helm**: Kubernetes package manager
  **Knative**: Serverless platform on Kubernetes
  **Spinnaker**: Control plane for deployment
Agenda

- What is Cloud Native?
- What is Cloud Native CI/CD?
- What is shifting left?
- What is Tekton?
- Demo!
- What’s next for Tekton?
What is CI/CD?
What is CI?

Continuous Integration

Literally integrating code together continually
Software Supply Chain

- How does my code get from source to running in prod?
- What pieces are involved?
- CI/CD = build, test + deploy
Cloud Native
CI/CD

The Definitive* Definition
* aka defined by me

1. Serverless
2. Specs and standards
3. Reusable components
4. Infrastructure agnostic
5. Config as code
Cloud Native CI/CD

1. Serverless
2. Specs and standards
3. Reusable components
4. Infrastructure agnostic
5. Config as code
Serverless CI/CD

- Scaling up and down resources as needed
- You specify what you want to run, freed of the much of responsibility to manage the underlying resources
Serverless CI/CD

- Don’t need to waste resources
- No need for monolithic execution
Cloud Native CI/CD

1. Serverless
2. Specs and standards
3. Reusable components
4. Infrastructure agnostic
5. Config as code
Specs & Standards

Doesn’t have to be perfect, just good enough & widely used

- Part of what makes k8s so cool is that it’s a spec we can all agree on
Specs & Standards

Infinite extensibility!

- Any k8s resource can be manipulated (e.g. with controllers, admission webhooks)
- Anything built on k8s can be manipulated with k8s tools
Cloud Native CI/CD

1. Serverless
2. Specs and standards
3. Reusable components
4. Infrastructure agnostic
5. Config as code
Reusable components

Do we need another Slack notification plugin?

- Write it once, use it again and again
- Kubernetes is itself a building block, that enables the creation of more building blocks
Cloud Native CI/CD

1. Serverless
2. Specs and standards
3. Reusable components
4. Infrastructure agnostic
5. Config as code
Use your own infra

Infrastructure agnostic

- A kubernetes pod is a kubernetes pod
- If you can deploy to prod kube, it should be possible to deploy to your kube
- Use the same config!
Use your own infra

Shift left

- The sooner you fail the better!
- Perform testing earlier in your lifecycle
- Why not do it before the code is pushed!
Use your own infra

Parity

- Make the complexity of kubernetes worth it!
- Pre-kube, it was nearly impossible for me to setup a production like environment on my own machine
Cloud Native CI/CD

1. Serverless
2. Specs and standards
3. Infrastructure agnostic
4. Reusable components
5. Config as code
Config as Code

Even the YAML

```
steps:
- name: unit-test
  image: golang:${{inputs.params.version}}
  workingdir: /workspace/src/${{inputs.params.package}}
  command:
    - /bin/bash
  args:
    - -c
    - "go test ${{inputs.params.flags}} ${{inputs.params.packages}}"
  env:
    - name: GOPATH
      value: /workspace
    - name: GOOS
      value: "${{inputs.params.GOOS}}"
    - name: GOARCH
      value: "${{inputs.params.GOARCH}}"
    - name: G011MODULE
      value: "${{inputs.params.G011MODULE}}"
```
Learning & Debugging

Fail easily!

- As our systems get more complicated, we need to be able to learn faster
- We need to be able to look at what the system are doing
Debugging = Learning
The faster you can learn, the faster you can deliver value
The most important skill to have as a programmer is the ability to teach yourself new things effectively and efficiently.

You're going to be constantly growing and picking up new technologies. The ability to do that is more important than any individual tool or technology.
How do we debug?

● By poking and prodding the thing
● Looking at what it’s doing
● Changing it, seeing what happens
Agenda

- What is Cloud Native?
- What is Cloud Native CI/CD?
- What is shifting left?
- What is Tekton?
- Demo!
- What’s next for Tekton?
Shifting Left
Software development

- REQUIREMENTS
- DESIGN
- DEV
- TEST
- DEPLOY

TESTING INCREASES
Cost of defects
Shift Left

Fail earlier!

There WILL be defects
- Test earlier!
- Catch failures earlier!
Shifting Left with Cloud Native
Agenda

- What is Cloud Native?
- What is Cloud Native CI/CD?
- What is shifting left?
- What is Tekton?
- Demo!
- What’s next for Tekton?
Tekton is Cloud Native CI/CD

Tekton is a Cloud Native CI/CD building block!
The CDF

CD.Foundation

Jenkins

Spinnaker

Tekton
Contributors from:

- Google
- CloudBees
- Red Hat
- IBM
- Salesforce
- Puppet
- … and more!

New contributor friendly!
Porcelain vs Plumbing
Toilet vs. Plumbing Underneath
Who is Tekton for?

Now!
Developers building CI/CD systems

Now!
End users that want DIY

Watch this Space
All users that want a catalog of reusable CI/CD
CRDs
Custom Resource Definitions

- Extending kubernetes with custom types
- Controllers act on Resources
- = CI/CD platform on Kubernetes
Steps

- Is actually a container spec (k8s type)
- Container image +
  - Environment variables
  - Arguments
  - Volumes
  - etc.
Task CRD

- New CRD
- Sequence of steps
- Run in sequential order
- Run on the same k8s node
Pipeline CRD

- Express Tasks order
  - Sequentially
  - Concurrently
  - (Graph)
- Execute Tasks on different nodes
- Link inputs and outputs
Runtime CRDs

- Instances of Pipeline/Task:
  - PipelineRun
  - TaskRun
- PipelineResource
  - Runtime info like image registry, git repo, etc.
Tekton Pipeline CRDs
PipelineResources + Software Supply Chain

- Typed artifacts moving through your supply chain
But I want to make PipelineRun whenever I open a Pull Request?
Tekton Triggers CRDs
<table>
<thead>
<tr>
<th>Cloud Native CI/CD</th>
<th>Tekton</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serverless</td>
<td>Serverless execution</td>
</tr>
<tr>
<td>Specs and standards</td>
<td>Tekton API</td>
</tr>
<tr>
<td>Infrastructure agnostic</td>
<td>Containers are the building block</td>
</tr>
<tr>
<td>Reusable components + Config as Code</td>
<td>Tasks, Pipelines, PipelineResources, TriggerTemplates, TriggerBindings, EventListeners</td>
</tr>
</tbody>
</table>
Agenda

- What is Cloud Native?
- What is Cloud Native CI/CD?
- What is shifting left?
- What is Tekton?
- Demo!
- What’s next for Tekton?
Tekton Demo

- CI/CD config living alongside the code
- Ran it on my cluster!
- Used kubernetes tools
- Reproducible
- Serverless execution
Agenda

- What is Cloud Native?
- What is Cloud Native CI/CD?
- What is shifting left?
- What is Tekton?
- Demo!
- What’s next for Tekton?
Tekton: Roadmap

2019 Q3
● Tekton Triggers 0.1

2020 Q1
● Tekton Pipelines Beta

2020 +
● Catalog!
● Manual approvals, notifications, other cool advanced CI/CD stuff
Join in!

- tekton.dev
- github.com/tektoncd/pipeline
- github.com/tektoncd/community
- CDF Summit at Kubecon NA (Nov 18)
  - cds2019.sched.com
  - events.linuxfoundation.org/events/kubecon-cloudnativecon-north-america-2019/co-located-events/
- @tektoncd on Twitter!
Thanks!

Software we think we should be making

Nailed it!

Software we actually make
Sources

- What is serverless: [martinfowler.com/articles/serverless.html](martinfowler.com/articles/serverless.html)
- Building a Debugging Mindset: [www.slideshare.net/InfoQ/building-a-debugging-mindset](www.slideshare.net/InfoQ/building-a-debugging-mindset)
- Learning is the most important skill: [twitter.com/ASpittel/status/1101165138361479169](twitter.com/ASpittel/status/1101165138361479169)
- Tekton: [github.com/tektoncd/pipeline](github.com/tektoncd/pipeline)