Hacking for Fun & Profit
The Kubernetes Way

Tal Peretz, Demi Ben-Ari @ Panorays
Some important things

● **What I’m not:** A Docker / Kubernetes Expert

● **What you won’t be after this talk:** A Docker / Kubernetes Expert

● What you will be after this talk?
  
  ● **Happier people** (Because I’ve stopped talking)
  
  ● You’ll know what was our problem and our way of solution
  
  ● You’ll know where to search and learn more things
  
  ● The answer to the “What’s the meaning of life?” (42)
About Us

**Demi Ben-Ari, Co-Founder & VP R&D @ Panorays**

- Google Developer Expert
- **Co-Founder of Communities:**
  - “Big Things” - Big Data, Data Science, DevOps
  - Google Developer Group Cloud
  - Ofek Alumni Association

**In the Past:**

- Sr. Data Engineer - Windward
- Team Leader & Sr. Java Software Engineer,
  Missile defence and Alert System - “Ofek” – IAF
About Us

Tal Peretz, Data Scientist @Panorays

- B.Sc Math & Computer Science
- MBA in entrepreneurship, innovation and technology

In the Past:

- Backend Developer for Air Traffic Control System
- Founded IAF Data Science Team
Mapping the World’s Cyber Posture
A breach to even the smallest 3rd party may cause a cyber typhoon in the industry.

- PNI Data Breach – Photo Services Affected – By Thomas George
- Geekwire, databreaches.net, Amateurphotographer.co.uk, scmagazine.com
It’s Not Only Your IT Vendors

3rd Party vendors flow data into company’s systems

Providers hold information about customers / employees

Consultants hold sensitive information of the company

“We’re seeing third party risk management show up as one of the top three board agenda items”

- T.R. Kane, cybersecurity and privacy partner at PwC, 2016
Panoramic Dynamic Ratings
for 3rd party Suppliers

360° full perimeter overview
Cyber gaps from the hacker point of view

Dynamic ratings
24/7 monitoring and alerting upon attack surface change

Installation free
No installation on customer or 3rd party vendor
So Basically what do we do? (Previous Situation)

- Every VM running would imitate the whole reconnaissance phase of the hackers lifecycle.
- Parallelism is being done through firing up more VMs.
- Built an internal orchestration system to launch all of the scans via Cron & Bash.
- All of the servers are running on Google Cloud Platform.
The Problem
What’s the hardest problem in Software Engineering?

\[ P = \text{NP?} \]
Naming Things

When you try to choose a meaningful variable name.
What’s the biggest problem in Software Engineering?

- Naming Things
SIMPLE STEPS
TO THE SOLUTION
Step #1 - Appoint a CNO

- **Chief Naming Officer** - your go to guy for all of the hardest problems
Step #2 - Define the problem and abstractions

- Parallelizm happening in the manner of a company (VMs being launched).
- Scan and evaluation process is not transparent.
- Server utilization is low.
- Had to build an internal orchestration system via Cron & Bash.
  - (Think how fun is that…)
- How do you monitor all of this?
- Need to control it all via an easy API
We’ve created a “Microlith”
In the beginning...

#!/bin/bash
In the beginning...
In the beginning...
Problems

- Manual
- Sequential
- Wasteful
- Inflexible
The Transporter

a Dynamic Workflow Engine
Built for Running Kubernetes Jobs According to a Predefined Workflow.
The Transporter

- Flexible and Efficient
- Parallel
- Automated
Overview
A bit about Kubernetes

- Greek for “Helmsman”; also the root of the words “governor” and “cybernetic”.
- Manages container clusters
- Inspired and informed by Google’s experience and an internal system (Borg)
- Supports multiple cloud and bare-metal environments
- 100% Open source, written in Go
- Manage applications, not Machines
Kubernetes Terminology

- Deployment
- Service
- ReplicaSet
- Pod
- Volume

- Label
- Selector
- ConfigMap
- Secret

- DaemonSet
- Stateful Set
- Job
- Liveness Probe
- Readiness Probe
RULES:
The Deal is the deal.
No names.
Never open the package.
Never make a promise you can't keep.
The Deal is The Deal

- Jobs
- Phases
- Workflows

Sequential
- wake up
- coffee
- code

Parallel
- drive
- sneeze

Parallel
- eat
- watch

Sequential
- shower
- brush
- sleep
The Deal is The Deal

- Jobs
- Phases
- Workflows

**Parallel**
- discover
- map
- fetch

**Sequential**
- detect
- aggregate
- save
Never Make a Promise You Can’t Keep

- Retries
- Schedule
- Timeouts
Under The Hood

Workflow

Worker

Task_1
Task_2
Task_3
...
Task_{n-2}
Task_{n-1}
Task_n

Q_1
Q_2
Q_3
...
Q_{n-2}
Q_{n-1}
Q_n

Worker_1
Worker_2
Worker_3
...
Worker_{n-2}
Worker_{n-1}
Worker_n

KubeJob_1
KubeJob_2
KubeJob_3
...
KubeJob_{n-2}
KubeJob_{n-1}
KubeJob_n
Almost Never Make a Promise You Can’t Keep

- Failure
- Notifications
Never Open The Package Container

- Docker Images of Jobs on GCR
- Kubernetes Jobs
No Names

- UUID

\( (([A-Za-z0-9][-A-Za-z0-9_.]*)?[A-Za-z0-9])?') \)

Max Chars: 63

- Labels, La

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Type</th>
<th>Pods</th>
<th>Namespace</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>website_purpose_speculator_bot_0b6131f9</td>
<td>🟢 OK</td>
<td>Job</td>
<td>1/1</td>
<td>the-transporter</td>
<td>flow-staging-k8s</td>
</tr>
</tbody>
</table>
So What’s On Our Cluster?

- The Transporter Service
- Workers Deployments
- Redis
- KubernetesJobs triggered by the transporter
Demo
What’s Next?

- Workflows Monitoring
- ConfigMap for Versions
- Asset level Parallelization $\sim O(n_{\text{assets}} \times job_{\text{slowest}})$ $\sim O(job_{\text{slowest}}) \times 100-1,000,000$
Conclusions

- If you have a possibility -> Don’t implement distributed systems
- Kubernetes is a great container orchestration tool
- Installing it on bare metal is not that fun - but also possible
- “Perfect” is the enemy of “Working” / “Giving Value”
Questions
Thank You