

# **Successful Kubernetes Development Workflows**

Ellen Körbes



Microsoft Visual Basic [design]

File Edit View Run Debug Options



Window Help

Color Palette

Debug

Ctrl+B

Menu Design

Ctrl+M

Procedures

F2

Project

F4

Properties

Toolbox

Data Manager

Report Designer

List Box Example

Name to add

IstClient

Remove

1680 1215 x 495

CONTROLS.MAK

View Form View Code

BUTTON.F frmButton

CHECK.FR frmCheck

LISTBOX.F frmListBox

MAIN FRM frmMain

Properties

cmdRemove CommandButt

&Remove

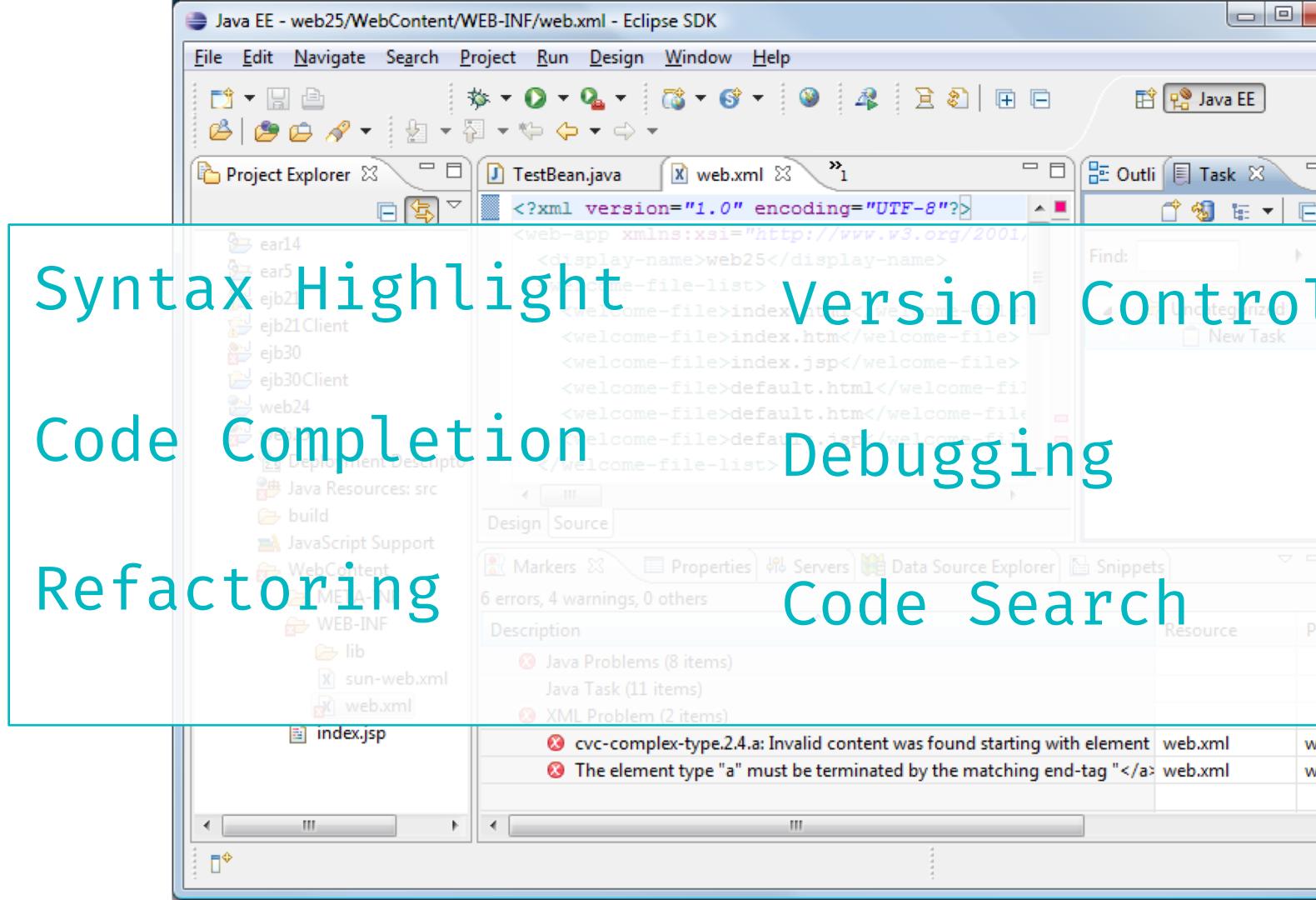
LISTBOX.FR M

Object: cmdRemove

Proc: Click

```
Sub cmdRemove_Click ()  
Dim Ind As Integer  
Ind = lstClient.ListIndex  
If Ind >= 0 Then  
    lstClient.RemoveItem Ind  
    lblDisplay.Caption = lstClient.ListCount  
Else  
    Beep      ' Should never occur, because Remove is always disabled
```

' Get index.  
' Make sure list  
' Remove it from  
' Display number



**Borland C++ - hello**

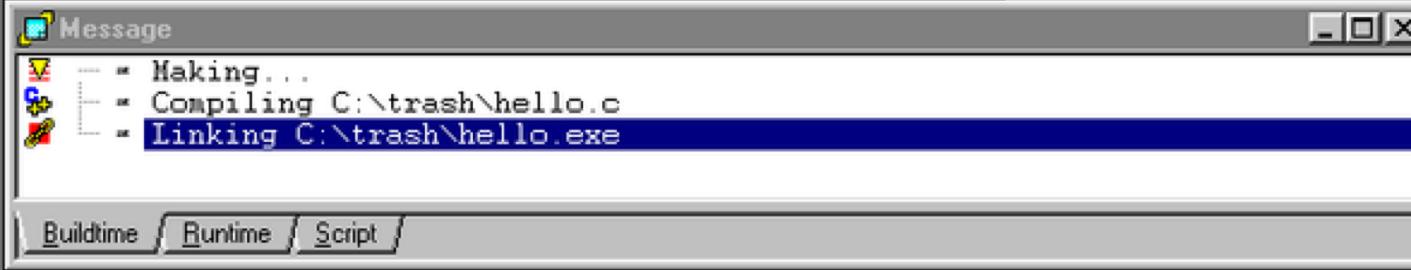
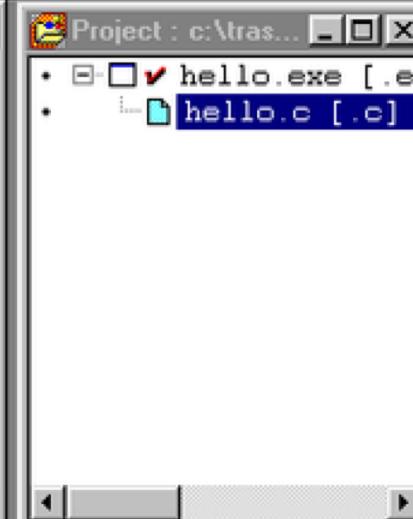
File Edit Search View Project Script Tool Debug Options Window Help



C:\trash\hello.c

```
#include <stdio.h>

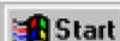
int
main(void)
{
    puts("hello world");
    return 0;
}
```



7:14

Insert

12:17:14 PM



Borland C++ - hello



12:17 PM

IntelliJ IDEA File Edit Search View Go To Code Analyze Refactor Build Run Tools Version Control Window DB Navigator Help Fri 6:20 AM

Augsburg\_450-451\_CCompiler - [/Users/mikehubbartt/Documents/eclipse-workspace/Augsburg\_450-451\_CCompiler] - [Augsburg\_450-451\_CCompiler] - .../src/MyScanner.java ...

Project /Users/mikehubbartt/... src MyScanner

View as: Project

1: Project Augsburg\_450-451\_CCompiler (/Users/mikehubbartt/...)

- .idea
- .settings
- src
  - CCompiler
  - MyScrapbook.jpage
  - MyParser
  - MyScanner
  - ScannerList
  - SemanticAnalyzer
  - SpecialException
  - SymbolTable
  - SymTableEntry
  - TheToken
  - TokenList
- .classpath
- .project
- Augsburg\_450-451\_CCompiler.iml
- helooo

External Libraries

2: Commander

3: Ant Build

4: Hibernate Tools

5: Maven Projects

6: TODO

ScannerList.java SemanticAnalyzer.java SpecialException.java  
CCompiler.java MyScanner.java MyParser.java TokenList.java

```
* <i>Assignment 1 - CSC450</i><p>
* #author Mike Hubbartt
*
public class MyScanner implements TokenList, ScannerList {
    private PushbackReader stdin;
    private PrintWriter stdout;
    private File infile, outfile;
    private int lineno = 1, fileExtend;
    private HashMap<String, Integer> KeyList;
    private ArrayList<Integer> theStream;
    String parsestatus = "", filename, outputname;

    public MyScanner(String filename) throws IOException, SpecialException {
        // Retrieve the source code file
        keyHashMap();
        fileExtend = filename.indexOf('.');
        infile = new File(filename);
        if (infile.exists() && infile.canRead()) { // Create the output files if they don't already exist
            stdin = new PushbackReader(new FileReader(infile));
            filename = filename.substring(0, fileExtend); // This allows variable length filename extensions
            outfile = new File(filename + '.' + 'o'); // This adds '.o' to the end of the filename
            outputname = filename + '.' + 's'; // set name for code generation output file
            if (!outfile.exists() || outfile.canWrite()) {
                stdout = new PrintWriter(new BufferedWriter(new FileWriter(outfile)));
            } else
                throw new SpecialException("Bad output file: " + filename + ".out");
        } else
            throw new SpecialException("Bad input file: " + filename);
    } // End of MyScanner()

    public void CreateOutput() throws IOException, SpecialException {
        // Create an output file & populate the data structure with tokens created during the scanner phase
        TheToken tokValue;
        tokValue = lexan();
        ArrayList<Integer> theStream = new ArrayList<>();

        // Create the output file header content
        stdout.println("// C Compiler Project");
        stdout.println("// CSC450-451 WINTER/SPRING 2010");
        stdout.println("// Date Scan Output File Created: " + new Date());
    }
}
```

9:1 Insert MacRoman

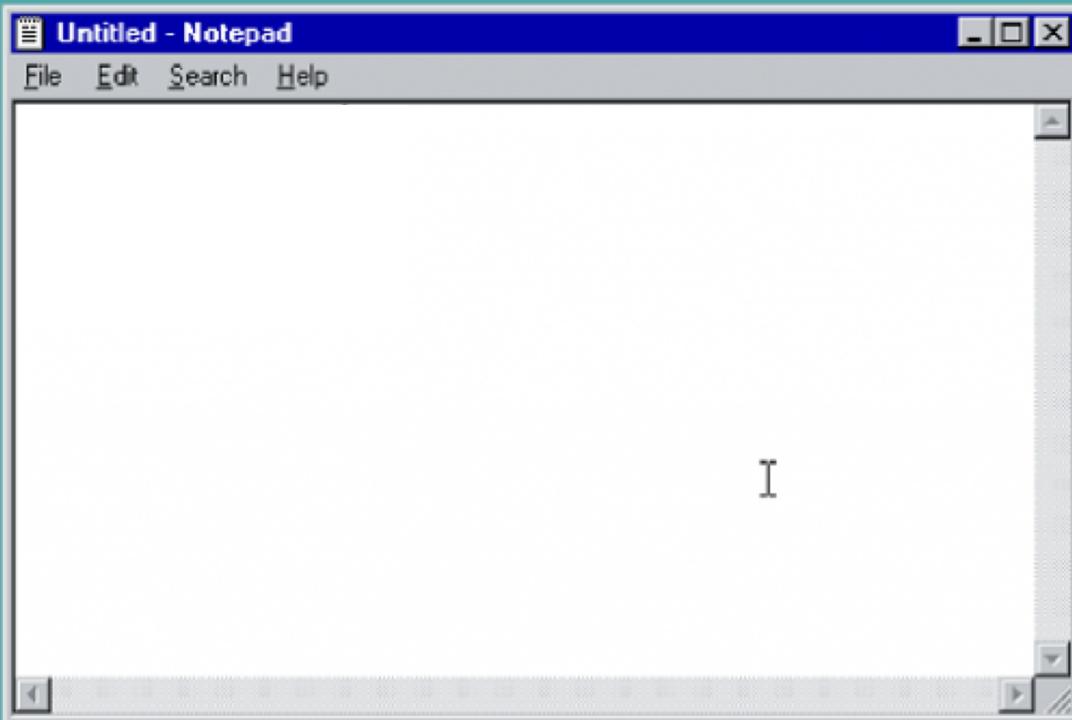
84M of 126M



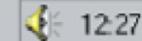
My Computer



Recycle Bin



Untitled - Notepad



12:27





```
kubectl run --restart=Always          # creates  
deployment  
  
kubectl run --restart=Never           # creates pod  
  
kubectl run --restart=OnFailure       # creates job
```

The Problem:

Containers and Kubernetes are  
incredible!



The Problem:

Containers and Kubernetes are  
incredible!

...except for the **development workflow**.



## \$ whoami

- **Ellen Körbes.**
- At Tilt, previously Garden. Both focused on Kubernetes DevEx.
- Constantly listening to developers' issues.
- Constantly working to solve them.



## Developer Relations

- [l@tilt.dev](mailto:l@tilt.dev)
- [@ellenkorbes](https://twitter.com/ellenkorbes)
- they/them
- [#tilt@slack.k8s.io](mailto:#tilt@slack.k8s.io)
- [@dex.dev](https://dex.dev)

Successful Kubernetes Development Workflows

# The Problem Set

## The Problem Set

Development  
Clusters

Managing  
Configuration  
Files

Feedback Loop  
Automation

Cluster Context  
Sharing

Debugging

Successful Kubernetes Development Workflows

# The Protagonists

Successful Kubernetes Development Workflows

# The Protagonists



Datadog

Cloud monitoring  
SaaS provider.

Engineering team:  
**~800** devs.

# The Protagonists



Datadog

Cloud monitoring

SaaS provider.

Engineering team:  
**~800** devs.



unu

Electric scooters

manufacturer in  
Berlin.

Development team  
has **~25** engineers.

# The Protagonists



Datadog

Cloud monitoring  
SaaS provider.

Engineering team:  
**~800** devs.



unu

Electric scooters  
manufacturer in  
Berlin.

Development team  
has **~25** engineers.



Minspace

Creative learning &  
gamification agency.

**Very tiny!**

**Four** engineers!

## Successful Kubernetes Development Workflows

# The Protagonists



Datadog

Cloud monitoring

SaaS provider.

Engineering team:  
**~800** devs.



unu

Electric scooters  
manufacturer in  
Berlin.

Development team  
has **~25** engineers.

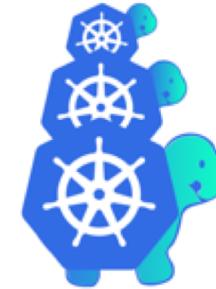


Minspace

Creative learning &  
gamification agency.

**Very tiny!**

**Four** engineers!



Cluster API

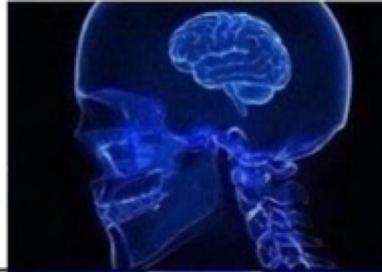
Use a cluster to  
create, configure,  
and manage other  
clusters.

**230+** contributors.

*Very. Weird. Workflow!*

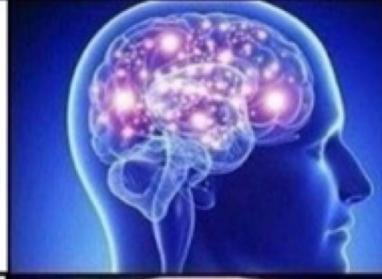
# RUN YOUR APP

---



## RUN YOUR APP IN A CONTAINER

---



**RUN YOUR CONTAINERIZED  
APP IN CONTAINERS  
ON A VIRTUAL MACHINE**

---



**USE A CONTAINERIZED APP  
RUNNING IN CONTAINERS ON  
A VIRTUAL MACHINE TO RUN  
A VIRTUAL MACHINE TO  
RUN YOUR CONTAINERIZED  
APP IN MORE CONTAINERS**

---



The Problem

# Development Clusters

no dev cluster ==



# Concerns

**Local** Cluster:

- Can the whole app fit a laptop's **RAM**?
- Which **type**? There's Minikube, kind, Microk8s, etc.
- Double-click setup
- Feedback bottleneck:
  - 1. Compute
  - 2. Network

# Concerns

**Remote** Cluster:

- Cash money dollars.
- Requires **more infra & setup** out of the box.
- No double-click setup!
- Feedback bottleneck:
  - 1. Network
  - 2. Compute

# Datadog

- **300+ services**—won't fit a laptop
- Self-managed cluster on **public cloud**
- **Separate namespaces** per team or per developer
- Wrapper tools for provisioning
- At first devs use staging services, which are cloned when working on them
- Option to add debugging tools

# Minspace

- Microk8s on Linux
- Mostly **Docker for Mac**
- **Docker Compose** for unit/integration tests... because Kubernetes in CI.
- Local clusters mirror prod, except for e.g. Mongo, replication

Successful Kubernetes Development Workflows

# KUBERNETES ON THE LAPTOP IS FINE!!!!111

...when you do it right.

# unu

- **Docker for Mac** (Troublesome!)
- Run everything **locally**
- Hitting **limits!**
- Solution: **optional services**

# Cluster API

- Concept: **management clusters**
- Use **kind** as the local dev cluster
- Why? Quick & **easy to tear** down
- Specific development **on every cloud**

# Takeaway

**Small** companies? **Local** cluster.

**Big** companies? **Remote** cluster.

Local clusters are easier to start, and  
companies migrate to remote once things don't  
fit a laptop anymore.

The Problem

# Managing Configuration Files

manually editing yaml ==



# Why?

Consider a simple Kubernetes app. YAML files:

- Deployment
- Service
- PersistentVolume
- StatefulSet
- Ingress

**It goes on...**

# Datadog

- **Helm** templates
- Different values for different environments (1 node vs. 100)
- **One dev writes the YAML** the first time...
- ...everyone else just `tilt up`.

# Minspace

- Services follow a **common pattern**
- **Helm templating** creates YAML
- Helm is further **automated with Tilt**

# unu

- Services follow a **common pattern**
- **Helm templating** creates YAML
- Helm is further **automated with Tilt**
- Semi-custom Tilt/Bash YAML generator.

Multi-layered Helm values file so people can override values per service, env, or locally.

# Cluster API

- **Convention** for all provider projects:  
Provider-specific JSON.
- User-specific Tilt settings on `tilt-settings.json` **overlays** on top of **defaults**.
- **Kustomize** templating
- For development everything is extremely **uniform**.

```
apiVersion: v1
kind: Service
metadata:
name: {{ template "fullname" . }}
labels:
    chart: "{{ .Chart.Name }}-{{ .Chart.Version }}"
spec:
type: {{ .Values.service.type }}
ports:
- port: {{ .Values.service.externalPort }}
    targetPort: {{ .Values.service.internalPort }}
    protocol: TCP
    name: {{ .Values.service.name }}
selector:
    app: {{ template "fullname" . }}
```

```
image:
repository: software/todo
tag: 1.0.0
nullPolicy: IfNotPresent
apiVersion: v1
kind: Service
metadata:
name: software
labels:
chart: "mychart-0.1.0"
spec:
type: ClusterIP
ports:
- port: 80
    targetPort: 80
    protocol: TCP
    name: nginx
selector:
app: software
...
```

```
apiVersion: v1
kind: Service
metadata:
name: {{ template "fullname" . }}
labels:
    chart: "{{ .Chart.Name }}-{{ .Chart.Version }}"
spec:
type: {{ .Values.service.type }}
ports:
- port: {{ .Values.service.externalPort }}
    targetPort: {{ .Values.service.internalPort }}
    protocol: TCP
    name: {{ .Values.service.name }}
selector:
    app: {{ template "fullname" . }}
```

```
image:
repository: software/todo
tag: 1.0.0
nullPolicy: IfNotPresent
apiVersion: v1
kind: Service
metadata:
name: software
labels:
chart: "mychart-0.1.0"
spec:
type: ClusterIP
ports:
- port: 80
    targetPort: 80
    protocol: TCP
    name: nginx
selector:
app: software
...
```

```
apiVersion: v1
kind: Service
metadata:
name: {{ template "fullname" . }}
labels:
    chart: "{{ .Chart.Name }}-{{ .Chart.Version }}"
spec:
type: {{ .Values.service.type }}
ports:
- port: {{ .Values.service.externalPort }}
    targetPort: {{ .Values.service.internalPort }}
    protocol: TCP
    name: {{ .Values.service.name }}
selector:
    app: {{ template "fullname" . }}
```

```
image: software/todo
repository: software/todo
tag: 1.0.0
nullPolicy: IfNotPresent
apiVersion: v1
kind: Service
metadata:
name: software
labels:
chart: "mychart-0.1.0"
spec:
type: ClusterIP
ports:
- port: 80
    targetPort: 80
    protocol: TCP
    name: nginx
selector:
app: software
...
```

# Takeaways

**Everyone uses a templating solution.**

Big companies sometimes roll their own.

Almost everyone uses **Helm** templates.

**IDE-like:** Values → Tilt → Helm → Kubernetes.

The Problem

# Feedback Loop Automation

# What?

Roughly, we want the following operations:

- docker build
- docker push
- kubectl apply

...to be done automatically.

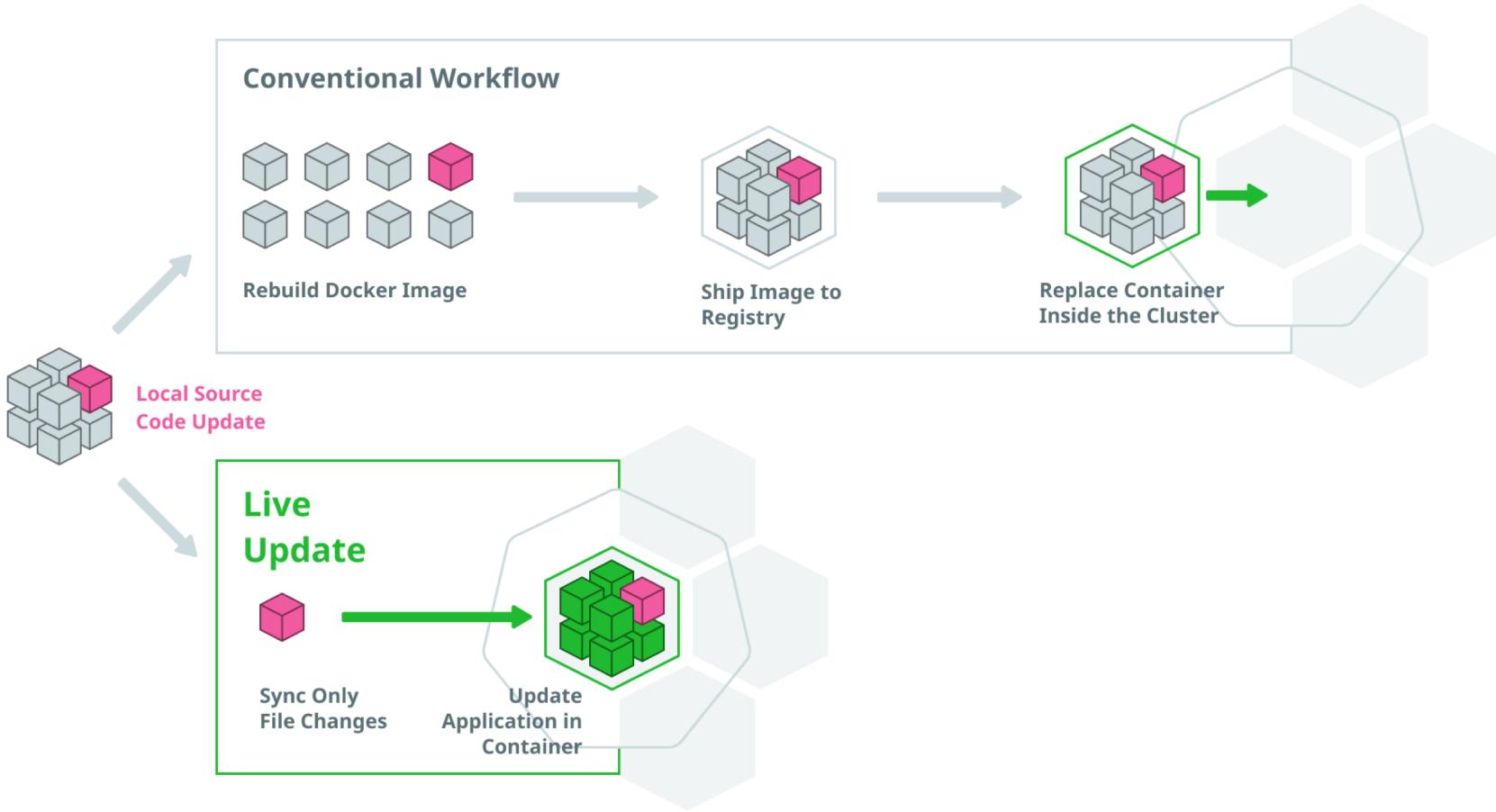
manual app update ==



# Why?

Developer cognitive load:

- Developers like to **stay focused**
  - **# of operations** per code change?
  - **Time** from change to new process?
- Custom workflow automation
- Onboarding





L  
@l\_korbes



Assuming your team uses **#Kubernetes**:

How long does it take between changing a line of code, and that code running in your development cluster? (Mark the closest answer.)

Feel free to share details of your setup 😊

30 minutes

22.6%

1 minute

21.6%

2 seconds

4.9%

**Results**

**50.9%**

1,931 votes · Final results

3:38 PM · Nov 29, 2019 · [Twitter Web App](#)

# Datadog

- Rolling out **Tilt**, currently at ~40%.
- **CI image** pulled locally
- **Build locally** inside CI image
- Tilt wraps Helm
- Easily discoverable **buttons in Tilt**:
  - Get dependencies
  - DB migrations

```
13   Tilt      x localhost:3000/ x |+
14   ← → C ⌂ 0 0 localhost:10350
15
16 Logs          Alerts ⚡
17 muxer        Applying Filter: Rectangler
18 object-detector 10.1.49.1 - - [21/Apr/2020 15:43:06] "POST /model
19 /predict?threshold=0.7 HTTP/1.1" 200 -
20
21 flush-database Web Trigger · flush-database
22
23 flush-database STEP 1/1 - Running command: [sh -c curl
24 http://localhost:8080/flush] (in "/home/l/go/src
25 /github.com/windmilleng/pixeltilt/simple")
26           % Total     % Received % Xferd  Average Speed   Time
27           Time      Time Current
28
29 flush-database
30           Spent    Left  Speed
31 100      9  100      9   0    0  1800      0 ---:---:---
32
33 flush-database
34           Dload  Upload  Total
35
36 flush-database
37           Spent    Left  Speed
38 100      9  100      9   0    0  1800      0 ---:---:---
39
40 flush-database
41           DONE IN: 0.02s
42
43 flush-database
44
45 flush-database
46
47 Create a Snapshot
48 ALL
49 (Tiltfile) 7m ago 0.0s
50 glitch 12m ago 0.3s
51 red 53m ago 0.1s
52 rectangler 2h ago 1.5s
53 storage 2h ago 1.7s
54 muxer 8m ago 0.3s
55 max-object-detector 2h ago 0.2s
56 frontend 58m ago 0.1s
57 flush-database <5s ago 0.0s
```

The screenshot shows the Tilt CLI interface running a Go application. The left pane displays logs from various components like muxer, object-detector, and flush-database. The right pane shows a timeline of events with their timestamps and durations. A red arrow points to the most recent 'flush-database' event, which occurred less than 5 seconds ago.

# unu

- unu inspired Tilt's **extensions** feature!
- **Tilt + tons of automation**, such as:

# unu

- unu inspired Tilt's **extensions** feature!
- **Tilt + tons of automation**, such as:
  - Internal Traefik proxy
  - TLS management
  - Vault integration
  - Tracing support
  - Sharded, replicated mongo cluster
  - Prometheus alerts
  - Live reload for Grafana dashboards (!)
  - *Special thanks: David Rubin, who wrote the first third-party Tilt extension!*

# Minspace

Minspace:

- **Tilt**, specifically for dev → prod parity
- Had tons of Tilt **hacks that** eventually became **native features**

# Cluster API

- User-specific **Tilt** settings on `tilt-settings.json` **overlays** on top of **defaults**
- Very complex Tilt **automation** e.g. cert. management functionality
- Used to build the Go binary in the container, with a full toolchain in the dev image, now **building binaries locally**

# Takeaways

- Pattern:
  - Uniform **services** fit a common **structure**, and allow for recycling configs, live reload settings, etc
  - Devs **automate everything** e.g. unu's service discovery, Traefik, etc.

The Problem

# Cluster Context Sharing

# What?

- “Hey, can you check out this error in my app? It’s in a cluster halfway across the world, and I’m not sure which of the 25 services is causing the error. Also we’re 7 time zones apart so can you look at it during the night and get back to me in the morning?”

copy&paste logs ==



# Datadog

- Devs work in public, shareable namespaces
- Use wrapper tool to switch namespaces
- **Any dev can access another dev's namespace**

# Minspace

- No need for high-tech when it's a small team sharing the same office!

# unu

- No need for high-tech when it's a small team sharing the same office!
- **Heavy use of automation** makes the need for this very rare.

# Cluster API

- Snapshots!

TIKT cloud SNAPSHOT

Shared by 33 weeks 5 days ago

ALL	—
● (Tiltfile)	8mo
● server	8mo
● client	8mo
● proto	8mo

LOGS PREVIEW ALERTS Snapshot shared by maiamcc

POD STATUS: Running POD ID: helloworld-server-65957d6757-j7mn8

Restarting

1 changed: [helloworld/helloworld.pb.go]

Rebuilding: server  
→ Updating container(s): 3b650bff4c  
Will copy 1 file(s) to container(s): 3b650bff4c  
- '/Users/maia/code/go/src/github.com/windmilleng/local\_resource\_example/helloworld.pb.go' --> '/app/helloworld/helloworld.pb.go'  
RUNNING: CGO\_ENABLED=0 GOOS=linux GOARCH=amd64 go install /app/greeter\_server/...  
RUNNING: /app/restart.sh  
Restarting  
→ Container 3b650bff4c updated!

0 errors 0 warnings LAST EDIT: server • helloworld/helloworld.pb.go 4/4 running

» COLLAPSE

# Takeaways

- **Big** companies: You have a **namespace** and your colleague logs into it
- **Middle** ground: **Snapshots**
- **Small** companies: analog solutions

The Problem

# Debugging!

# Datadog & unu

- No debuggers. Output to **logs**, then use **tracing**.
- Datadog uses Datadog for metrics & traces
- **Auto-instrumentation** helps!
- unu uses Jaeger

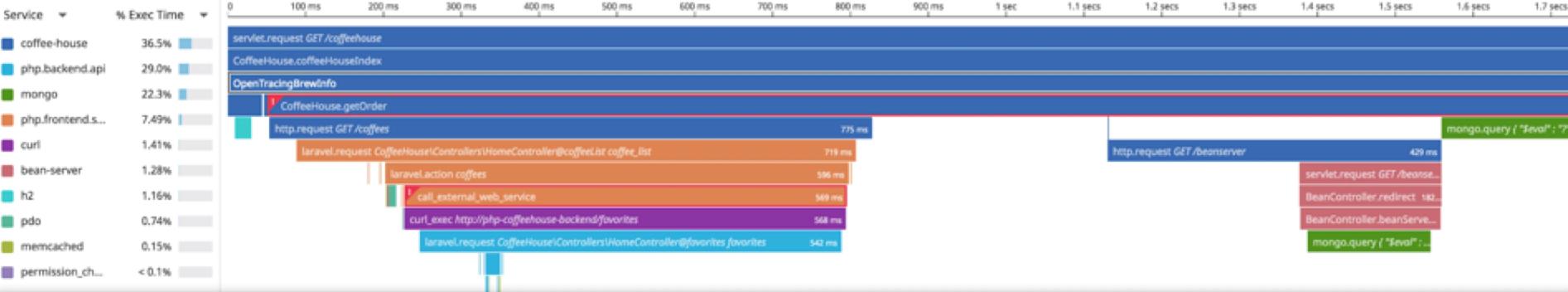
Q Search APM

> Services > coffee-house > servlet.request > GET /coffeehouse > 1592247487685740042

## coffee-house | GET /coffeehouse

Feb 7, 4:13 pm 1.85 s GET http://java-coffeehouse:8080/coffeehouse 200 OK

Flame Graph Span List (59)



## coffee-house | OpenTracingBrewInfo OpenTracingBrewInfo

demo.coffeehouse.dev 1.85 s (99.5% of total time)

Span Metadata Host Info Logs

trace\_id Q trace\_id:1592247487685740042

DATE ↑ SERVICE HOST

Feb 07 16:13:16.000	coffee-house	demo.coffeehouse.dev
GET /api/auth/	(10.8.4.7) - 200 OK	Authentication successful
Feb 07 16:13:16.000	coffee-house	demo.coffeehouse.dev
Monitor thread successfully connected to server with description ServerDescription address=mongodb:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[3, 4, 17]}, minWireVersion=0, maxWireVersion=5, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=null, roundTripTimeNanos=1277564		
Feb 07 16:13:17.000	coffee-house	demo.coffeehouse.dev
java.lang.InterruptedIOException: Thread interrupted for external calls timeout	500	
Feb 07 16:13:18.000	coffee-house	demo.coffeehouse.dev
GET http://java-coffeehouse:8080/coffeehouse completed with status code 200 in 1845 ms		

# Minspace & Cluster API

Minspace:

- **Remote debugging:** IDE connects to node remote debugging; Tilt exposes the ports

Cluster API:

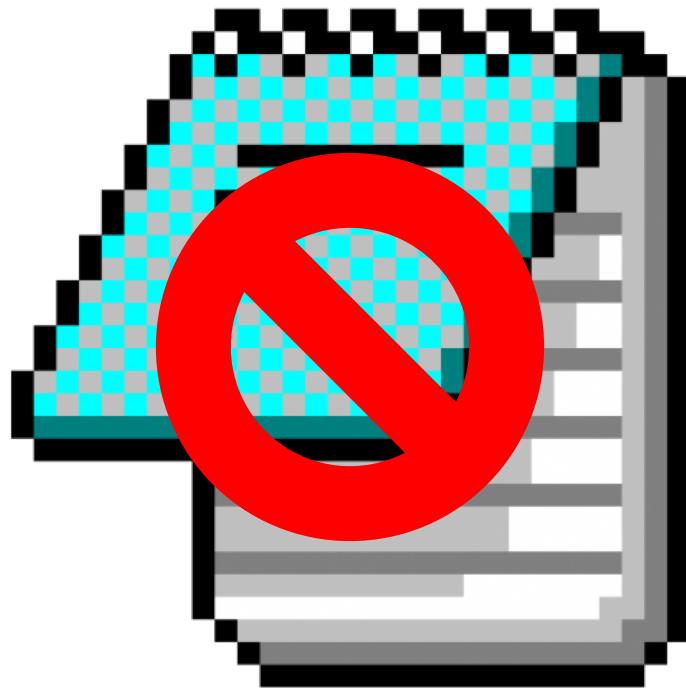
- Debugging: **Printlines**—no support for debuggers
- **Quick feedback loop** means this is fine

# Takeaway

- Integration with **tracing** (Datadog & unu) and **debugging** tools (Minspace) is still rare but growing!

Successful Kubernetes Development Workflows

**notepad.exe is not a development environment**



Successful Kubernetes Development Workflows

# Thank You!

# dex.dev

- Educational resource for **developer experience** in the containers & Kubernetes world
  - Videos
  - Articles
  - Livestreams
- Sign up! 😊

```
dex(developer.experience);
```

**Developer Experience needs a perspective shift.**



Writing code for containers & Kubernetes sucks. But it doesn't have to. We'll show you how.

# Ellen Körbes



## Developer Relations

- l@tilt.dev
- @ellenkorbes
- they/them
- #tilt@slack.k8s.io
- [dex.dev](#)

## Featured:

- Datadog [datadoghq.com](#)
- unu [unumotors.com](#)
- Mindspace [mindspace.net](#)
- Cluster API [cluster-api.sigs.k8s.io](#)