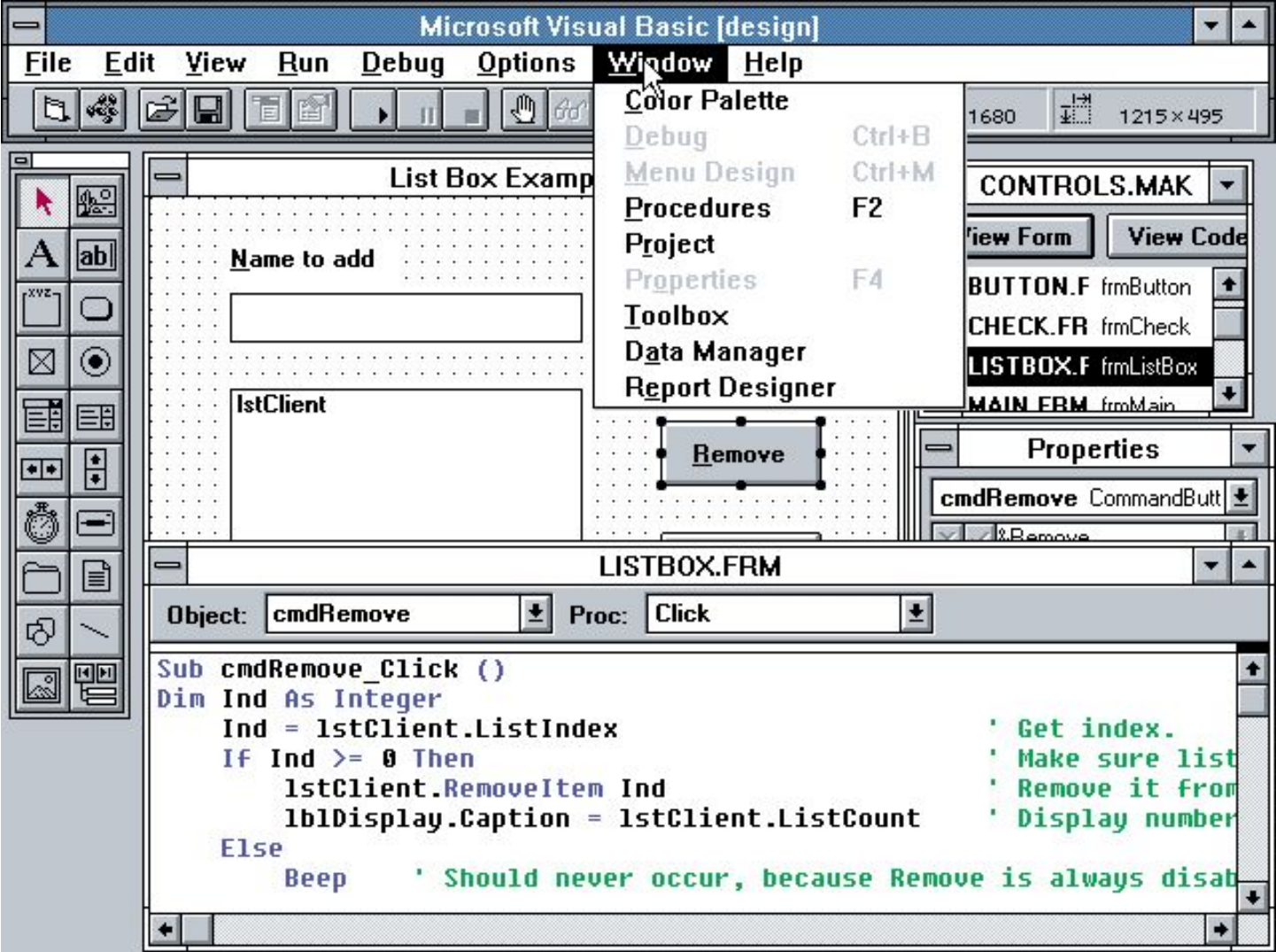
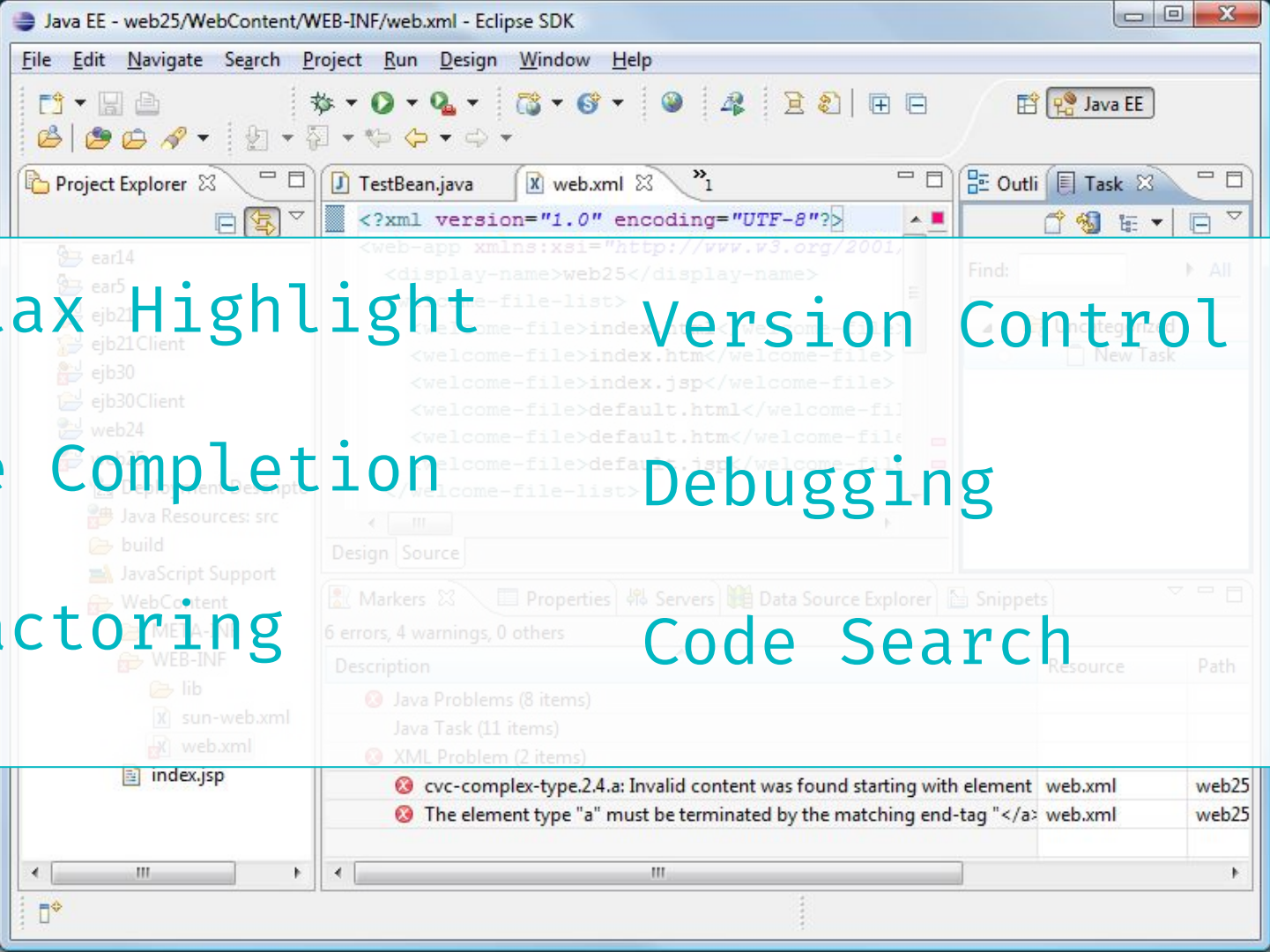


Successful Kubernetes Development Workflows

Ellen Körbes







Syntax Highlight Version Control

Code Completion Debugging

Refactoring Code Search



C:\trash\hello.c

```
#include <stdio.h>

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

Project : c:\tras...

- hello.exe [.exe]
- hello.c [.c]

Message

```
--- Making...
--- Compiling C:\trash\hello.c
--- Linking C:\trash\hello.exe
```

Buildtime

Runtime

Script

7:14

Insert

12:17:14 PM

Project /Users/mikeh...

View as: Project

- Augsburg_450-451_CCompiler (/Users/...)
 - .idea
 - .settings
 - src
 - CCompiler
 - MyCScrapbook.jpape
 - MyParser
 - MyScanner
 - ScannerList
 - SemanticAnalyzer
 - SpecialException
 - SymbolTable
 - SymTableEntry
 - TheToken
 - TokenList
 - classpath
 - project
 - Augsburg_450-451_CCompiler.iml
 - helloo
- External Libraries

6: TODO

```

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

 * <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('.'); // Open the source file and remove the period and extension
        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());
    }
}

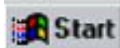
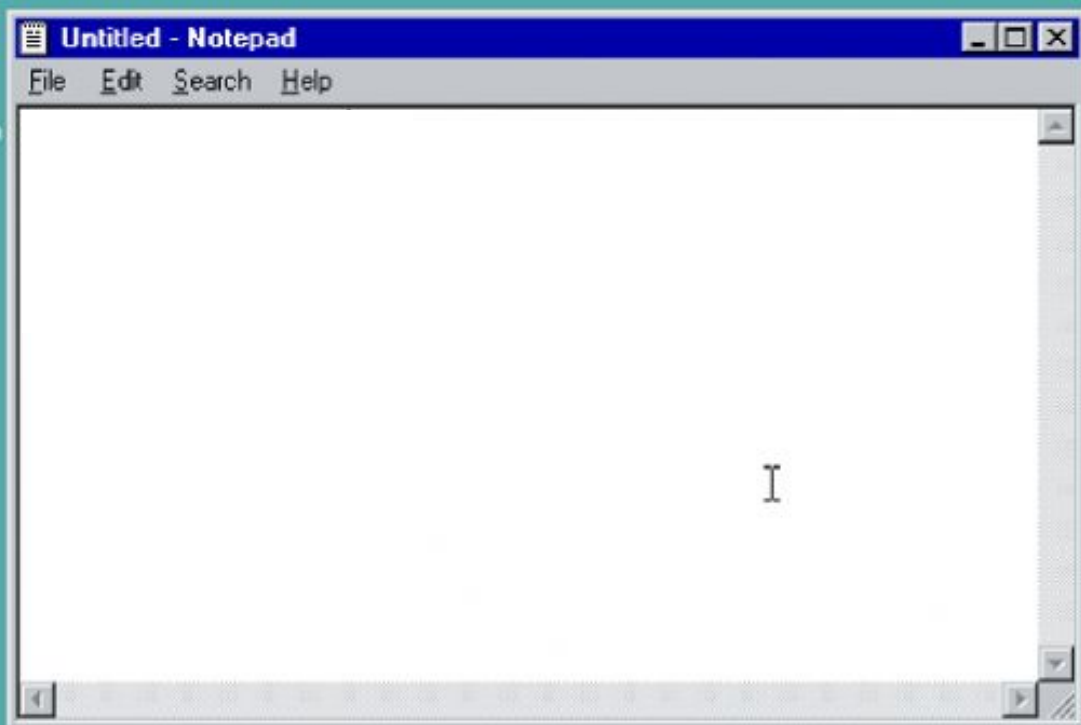
```



My Computer



Recycle Bin



Start

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

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

Successful Kubernetes Development Workflows

The Problem Set



@ellenkorbes

The Problem Set

Development
Clusters

Feedback Loop
Automation

Managing
Configuration
Files

Cluster Context
Sharing

Extras

Successful Kubernetes Development Workflows

The Protagonists



@ellenkorbes

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.



MindSPACE

Creative learning &
gamification agency.

Very **tiny!**

Four engineers!

The Protagonists



Datadog

Cloud monitoring
SaaS provider.

Engineering team:
~**800** devs.



unu

Electric scooters
manufacturer in
Berlin.

Development team
has ~**25** engineers.

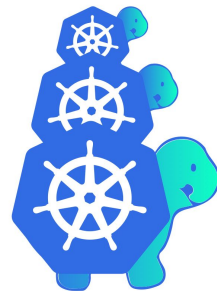


Mindspace

Creative learning &
gamification agency.

Very **tiny!**

Four engineers!



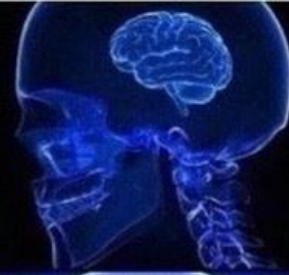
Cluster API

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**



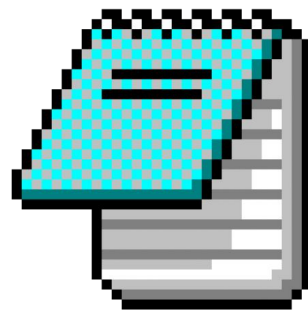
@ellenkorbes

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



Mindspace

- 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.

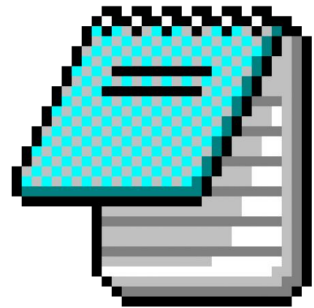
Local clusters are easier to start, and companies migrate to remote once things don't fit a laptop anymore.



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`.



Mindspace

- 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
  pullPolicy: 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
  pullPolicy: 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
...
```



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?

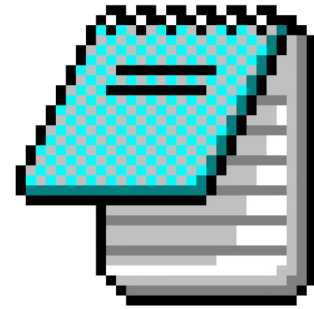
Roughly, we want the following operations:

- docker build
- docker push
- kubectl apply

...to be done automatically.



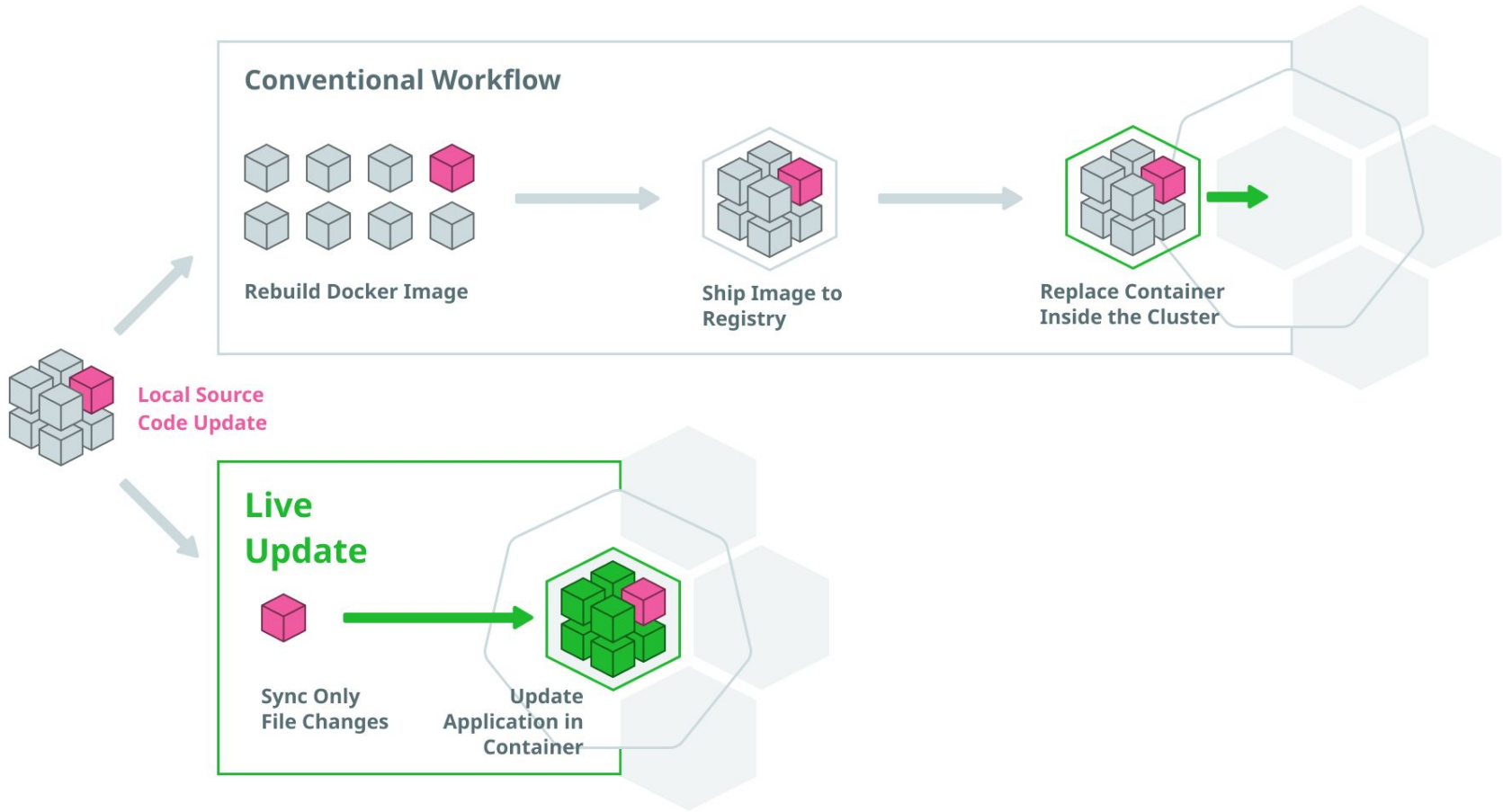
manual app update ==



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 😊



1,931 votes · Final results

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



@ellenkorbes

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



The screenshot shows the Tilt web interface. On the left, a 'Logs' panel displays the output of a 'flush-database' service. The logs show a 'Web Trigger' event and a 'STEP 1/1' command execution. A table summarizes the transfer statistics:

% Total	% Received	% Xferd	Average Speed	Time
100	9	100	9	0
---	---	---	---	---
1800				

Below the table, the logs indicate 'Flushed!' and 'DONE IN: 0.02s'. On the right, a list of services is shown with their status and age. A red arrow points to the 'flush-database' service at the bottom of the list.

- (Tiltfile) 7m ago 0.0s
- glitch 12m ago 0.3s
- red 53m ago 0.1s
- rectangler 2h ago 1.5s
- storage 2h ago 1.7s
- muxer 8m ago 0.3s
- max-object-detector 2h ago 0.2s
- frontend 58m ago 0.1s
- flush-database <5s ago 0.0s

At the bottom of the interface, it shows '0 errors', '0 warnings', 'LAST EDIT: -', and '9/9 running'.



@ellenkorbes

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!*



Mindspace

Mindspace:

- **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.

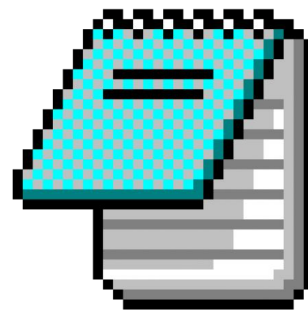
The Problem

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**

Mindspace

- 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!

TILT cloud SNAPSHOT Shared by 33 weeks 5 days ago

LOGS PREVIEW ALERTS Snapshot shared by maia MCC

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

```
Rebuilding
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/hellowor
/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!
```

ALL —

- (Tiltfile) 8mo
- server 8mo
- client 8mo
- proto 8mo

» COLLAPSE

✖ 0 errors ▲ 0 warnings LAST EDIT: server ▶ helloworld/helloworld.pb.go 4/4 running 🟢

Takeaways

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

The Problem

Extras!



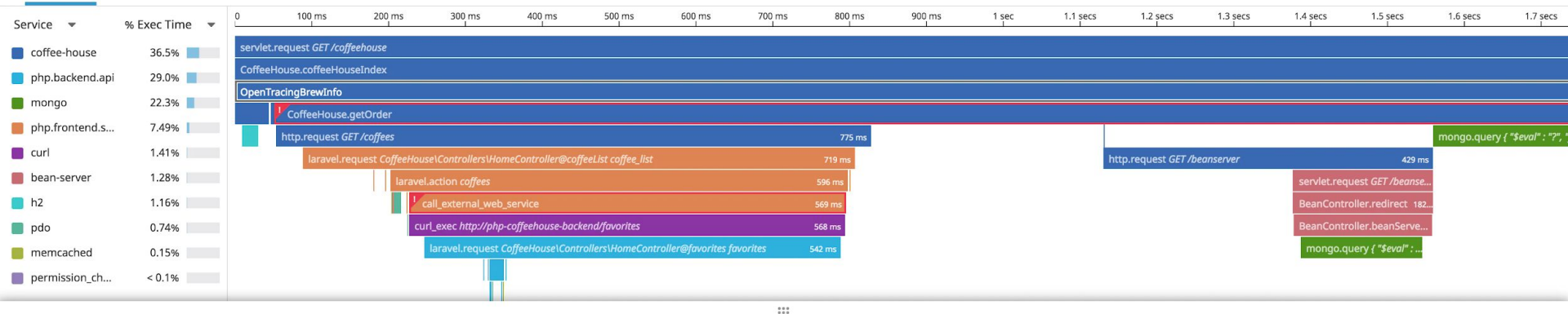
Datadog & unu

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

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: 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.InterruptedExceptio: 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		

Mindspace & Cluster API

Mindspace:

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

Cluster API:

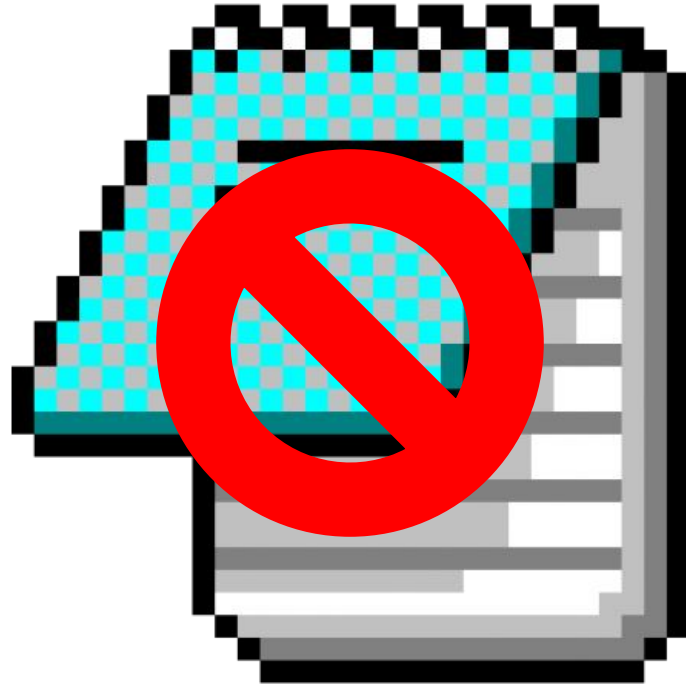
- Debugging: **Printlines**—no support for debuggers
- **Quick feedback loop** means this is fine

Takeaway

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

Successful Kubernetes Development Workflows

notepad.exe is not a development environment



Successful Kubernetes Development Workflows

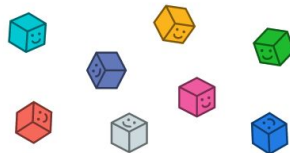
Thank You!

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](#)

