



DevOps for Defense

February 2020

Life of a Microservice @Shipt

Vadim Uchitel

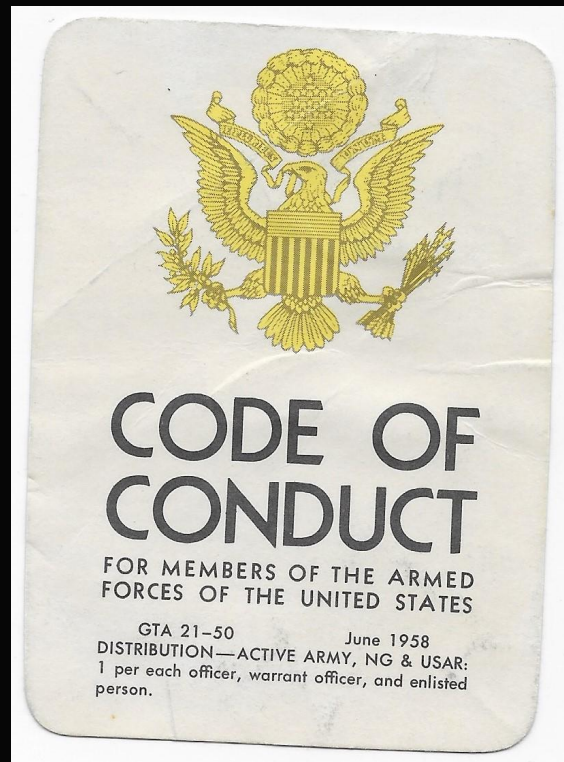
Sponsored by:



<https://devopsfordefense.org>
<https://www.meetup.com/DevOps-for-Defense/>
<https://github.com/jondavid-black/DevOpsForDefense>
devopsfordefense@gmail.com
<https://twitter.com/devops4defense>

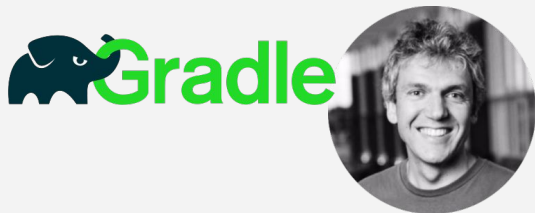
DevOps for Defense Meetup: Code of Conduct

- UNCLASSIFIED ONLY!!!!
- Treat each other with respect and professionalism.
- Do not talk about private, sensitive, or proprietary work.
- Do talk about your experiences, needs, desires to improve work in our domain.
- Do share your thoughts.
- Do learn from others.
- Do respect & tip your bartenders!



What's Next for DevOps for Defense?

March 5th



Hans Dockter

CEO Gradle Inc.

Powerful Automation & Insight

April 2nd



Open source complete CI/CD
toolchain out-of-the-box.

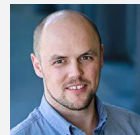
Beyond

(Still coordinating & planning,
but here's what we're thinking.)



Nicolas M. Chaillan

USAF Chief Software Officer -
Bringing DevSecOps DoD-wide



Dr. Mik Kersten

CEO Tasktop -
Author of Project to Product



Hack-a-thon?

Opportunity to put our
DevOps learning into
practice.

Provide us feedback so we can tailor to your needs.

DevOps for Defense Community Challenge

Donate Time 1 Day a Month

<https://girlswhocode.com/>

Emeka Barclay Marshall
Language Arts | Liberty Middle School
Apple Teacher
Google Certified Educator
Microsoft Innovative Educator
Flipgrid Certified Educator



“Knowledge is power. Information is liberating. Education is the premise of progress, in every society, in every family.” -Kofi Annan

Website: caffeinatedteacher.weebly.com

Twitter: [@teacheremeka](https://twitter.com/teacheremeka)



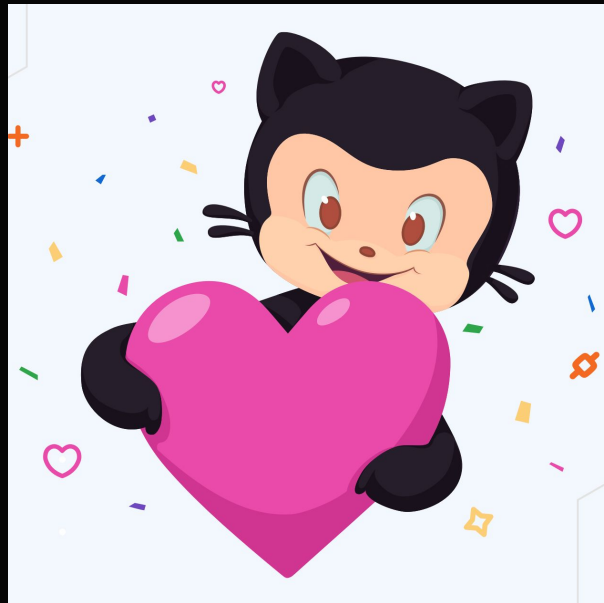
Help Drive Positive Change

Support Girls Who Code through smiles.amazon.com

Consider Subscribing & Supporting



Subscribe to
DevOps for Defense
on **YouTube**



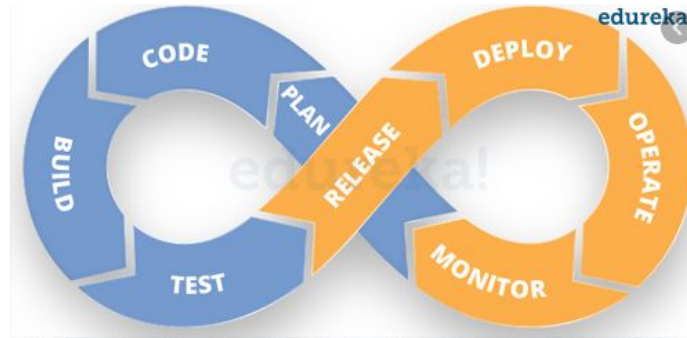
Consider supporting us through GitHub Sponsors.
Multiple levels of support. 100% Match!

<https://github.com/jondavid-black>

**No fees! - All donations directly support the
future of DevOps for Defense meetup!**

Life of a microservice @Shipt

Narrated by Vadim Uchitel

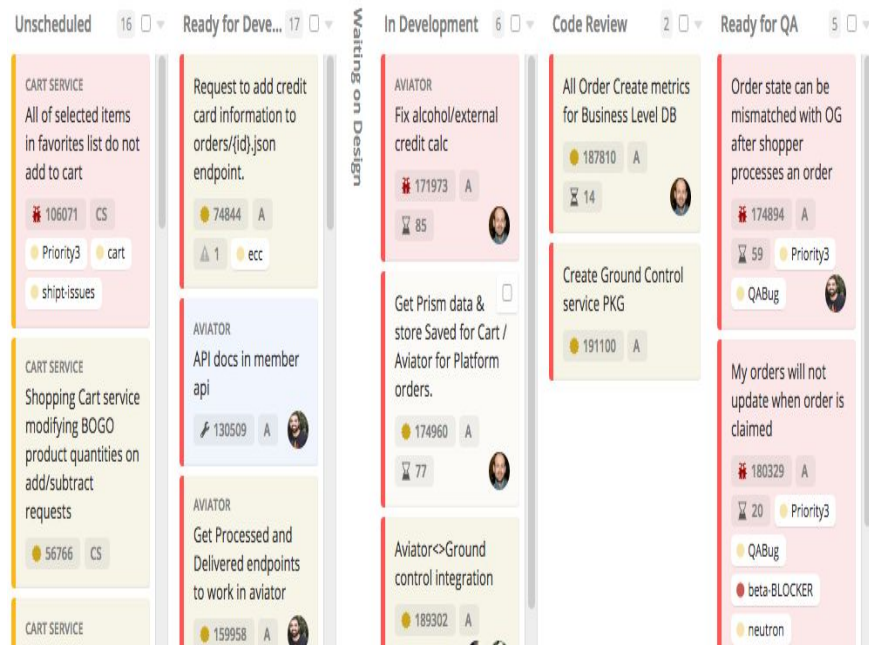


About Narrator (<https://www.linkedin.com/in/vadim-uchitel-2037b91/>)

- CFDRC / ESI-CFD (10 years)
-> Deployed code every 12 months
- SAIC / Leidos (11 years)
-> Deployed code every 3 - 6 months
- Shipt (2.5 years)
-> Deploy code several times per day

Plan. Clubhouse (JIRA / Kanban ...)










- Creating a card in Clubhouse starts the development process
- Automatic integration with Github
- All status changes are posted to Email / Slack

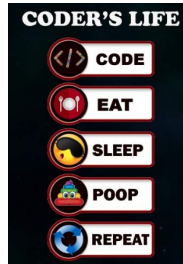




Let's write code and commit to Github

- Commit code
- Open Pull Request (PR)
- Require 1 or 2 reviews
- Require unit tests (code coverage should go up)
- Require CI to pass
- At Shipt we use Golang (Go), programming language developed by Google

 All checks have passed 4 successful checks		Hide all checks
	 codecov/patch — 100.00% of diff hit (target 84.76%)	Details
	 codecov/project — 84.79% (+0.03%) compared to 229b1b1	Details
	 continuous-integration/drone/pr — Build is passing	Details
	 continuous-integration/drone/push — Build is passing	Details

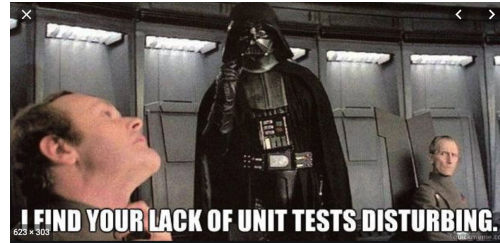


Unit Tests

- You can write code faster without unit tests
- You will pay for it in the long term, since making changes to code without unit tests will lead to bugs

-> Pay now or pay more later? Your call

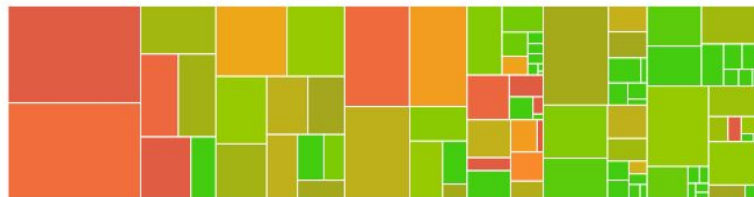
-> At Shipt we require ~85% coverage and use <http://codecov.io/> to track it



codecov bot commented 2 days ago

Codecov Report

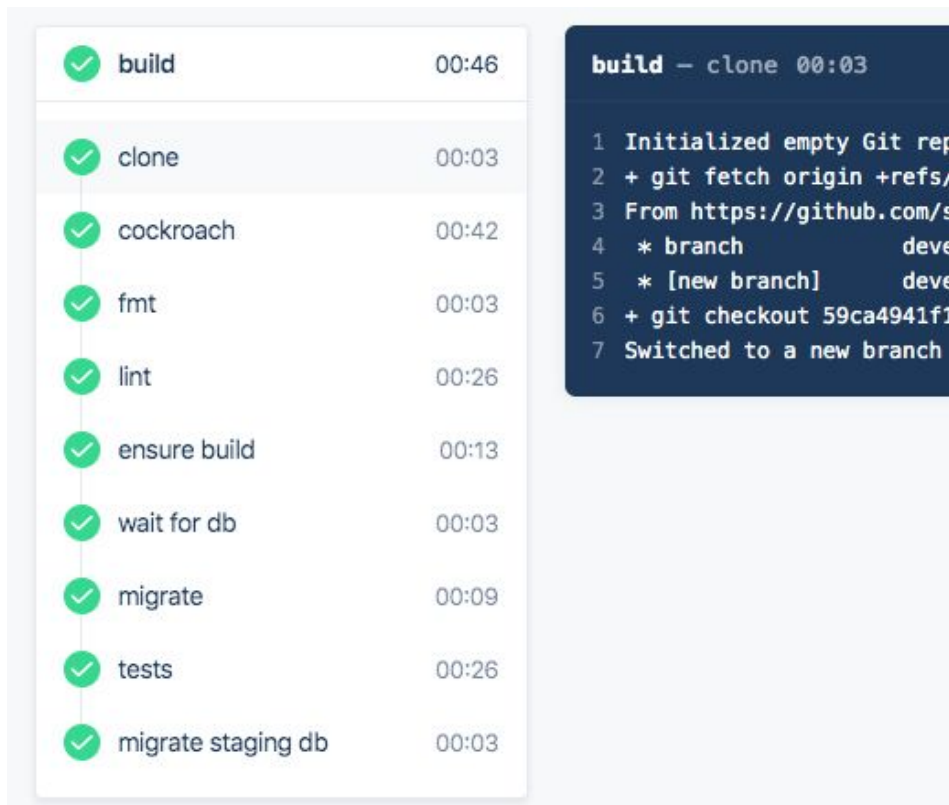
Merging #1309 into development will increase coverage by 0.03% .
The diff coverage is n/a .



Coverage Diff			
##	development	#1309	+/-
=====			
+ Coverage	84.76%	84.79%	+0.03%
=====			

Continuous Integration (CI). Drone (drone.io)

- Drone vs Jenkins
<https://rancher.com/drone-vs-jenkins>
- “Drone is a wonderful piece of CI software. It has quickly become a very popular choice for wanting to get up and running quickly, looking for a simple container-native CI solution.”
- YAML driven and designed for quick setup of unit tests & building docker containers
- Less powerful than Jenkins with its multiple plugins, but easier to use



The screenshot displays a Drone CI build interface. On the left, a vertical list of 11 build steps is shown, each with a green checkmark icon, the step name, and its duration. The steps are: build (00:46), clone (00:03), cockroach (00:42), fmt (00:03), lint (00:26), ensure build (00:13), wait for db (00:03), migrate (00:09), tests (00:26), and migrate staging db (00:03). On the right, a dark blue terminal window shows the output of the 'clone' step, which includes commands to initialize a Git repository, fetch the origin, and checkout a new branch.

Step	Duration
build	00:46
clone	00:03
cockroach	00:42
fmt	00:03
lint	00:26
ensure build	00:13
wait for db	00:03
migrate	00:09
tests	00:26
migrate staging db	00:03

```
build - clone 00:03

1 Initialized empty Git repository in .
2 + git fetch origin +refs/heads/*:refs/remotes/origin/*
3 From https://github.com:drone-io/drone
4 * branch            develop
5 * [new branch]      develop -> origin/develop
6 + git checkout 59ca4941f1
7 Switched to a new branch 'develop'
```

Don't repeat Phoenix. Build Staging environment

- “Staging environments are made to test codes, builds, and updates to ensure quality under a production-like environment before application deployment.”
- Always have a way to mimic production environment. Otherwise, **you will fail**
- At Shipt, Staging infrastructure (but not data) matches Production



Integration Tests

- Integration tests automatically run after each code deploy to Staging
- Integration tests (executed by Drone) call multiple services in Staging environment, validate responses, and notify about failures (via Slack or Email)
- Since services are developed by multiple engineers / teams, integration tests ensure that overall system still operates correctly after the changes



Load Tests

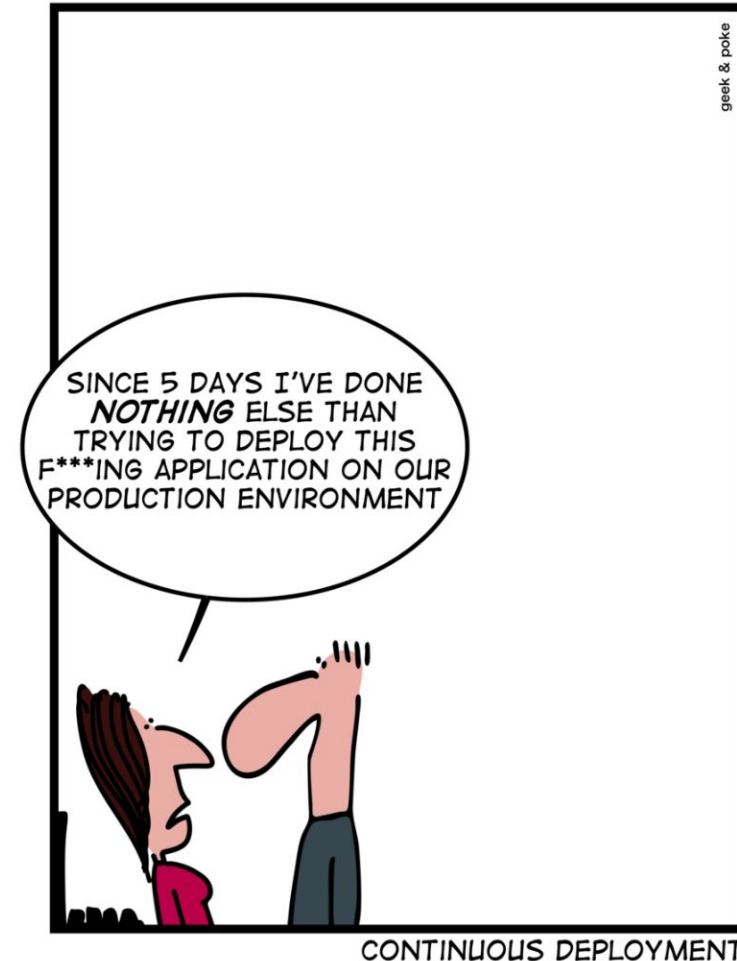
- **Load testing** is performed to determine a system's behavior under both normal and anticipated peak **load** conditions
- At Shipt we use Locust
<https://github.com/locustio/locust>
to perform load testing with 10x average production load
- Running load tests allow engineers to sleep better on weekends when system load is at its peak



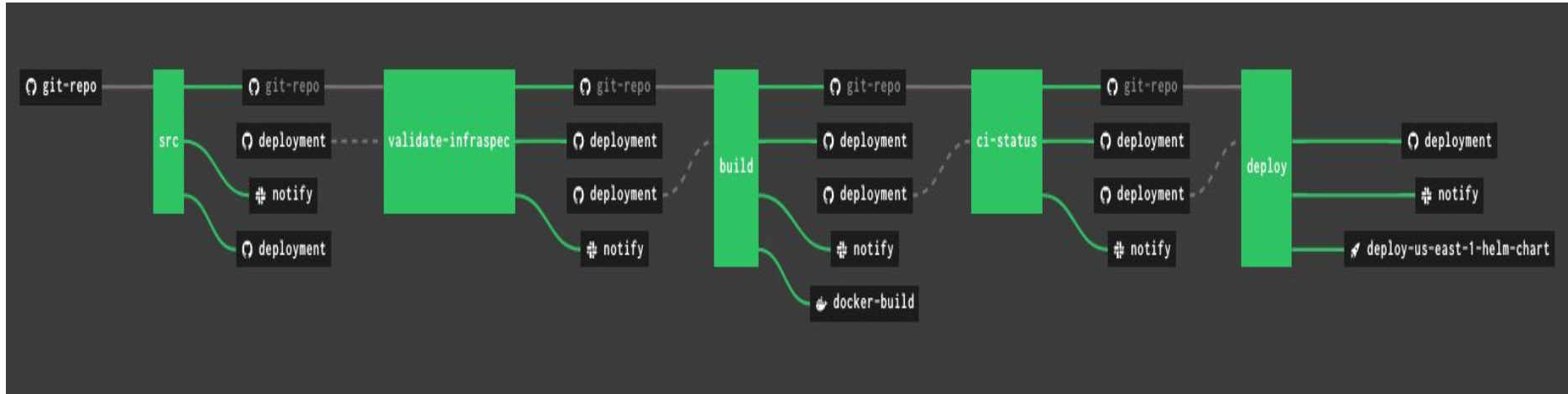
Continuous Deployment (CD)

Continuous Deployment is a software development practice in which every code change goes through the entire pipeline and is put into production, automatically, resulting in many production **deployments** every day.

SIMPLY EXPLAINED



Continuous Deployment(CD). Concourse <https://concourse-ci.org/>



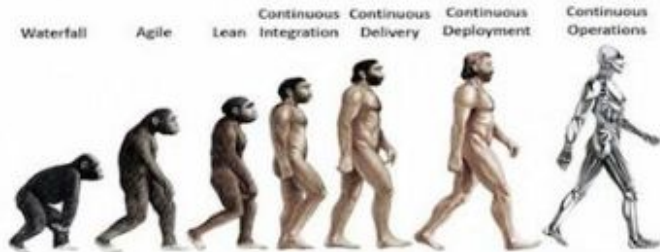
- Concourse provides easy-to-use process and visualization of the deployment pipeline and allows to rerun different parts of the pipeline in the case of the failure
- All pipeline failures & successes are immediately reported to Slack / Email
- Unit tests (run via Drone) are automatically executed and failure will stop the deploy
- Pipeline run (typically takes several minutes) results in code being operational and in-use by Customers
- Pipeline automatically starts after Github code merge to Master branch

Infrastructure as Code

- **All Shipt infrastructure runs on AWS**
- “Infrastructure As Code” allows to provision (as well as delete) necessary cloud resources to run the microservice
- Internal spec (aka Infrspec) <https://shipt.tech/infrspec-declarative-microservices-557639222c7c>
 - converts YAML file into Terraform script
- Terraform (terraform.io)
 - Allows to specify what your infrastructure should be via scripts
 - Automatically provisions and deletes necessary cloud infrastructure.
 - Works for AWS, GCP, Azure
- New alternative - Pulumi (pulumi.com)
 - Similar to Terraform, but allows to specify your cloud resources in multiple programming languages (Go, JS, Python)

Service is in production Are we done?

DevOps Movement



WHY SHOULD I
**TEST MY
CODE**

WHEN USERS
DO IT FOR ME


**IN
PRODUCTION**

Error Monitoring and Crash reporting

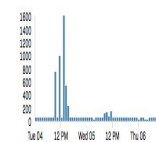


- All critical errors and crashes are reported to Rollbar (rollbar.com) and notifications being sent to Slack / Email
- Rollbar integration with Github allows fast triage and identification of the issue
- All important queues have alerts attached with OpsGenie and Slack alerts

Top 10 items in last 24 hours

24hr Trend	Count	IPs	Title
	4	1	#1860 BadRequest: Status Code: 500: Put https://shipkit-service-us-east-1.s3.amazonaws.com/rollbar/rollbar-go/rollbar-go.log
	3	1	#2133 MappedCategoriesEmpty: Mapped Categories empty
	1	1	#2132 MappedCategoriesEmpty: Mapped Categories empty
	154	1	#1858 Timeout: Timeout: Error posting order to provider: CircuitBreakerOpen...
	138	1	#1859 Timeout: Timeout: Status Code: 503
	85	1	#2112 GetFlightPlanFailed: ItemNotFound
	71	1	#1857 MemberShipCheckError: Errors: 1 launch Thank you for signing up for Shipkit...

Hourly Error/Critical Occurrences

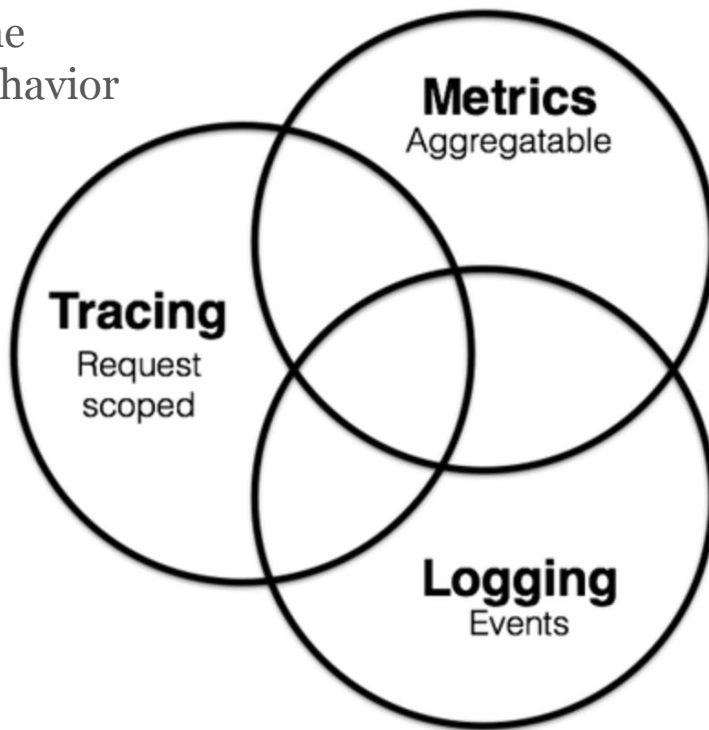


Traceback (most recent call last):

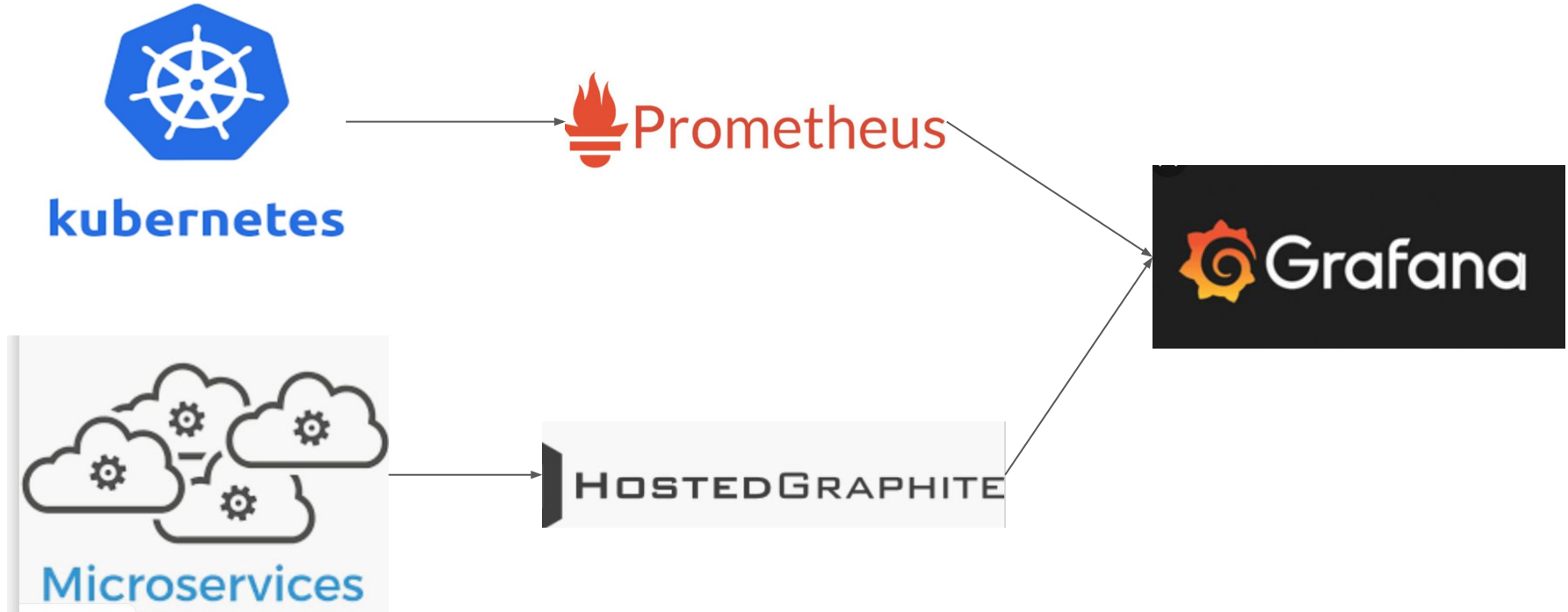
```
1 File "vendor/github.com/rollbar/rollbar-go/rollbar-go", line 444 in rollbar-go.log
2 File "vendor/github.com/shipkit-tofu/logger/logger-go", line 147 in logger.sendRollbar
3 File "vendor/github.com/shipkit-tofu/logger/logger-go", line 87 in logger.trackError
4 File "vendor/github.com/shipkit-tofu/logger/logger-go", line 138 in logger.(*InfoLog).Error
5 File "vendor/github.com/shipkit-tofu/logger/logger-go", line 268 in logger.Error
6 File "internal/controller/helpers.go", line 287 in controller.getFlightPlan
7 File "internal/controller/admin/controller.go", line 77 in controller.AdminController.Get
8 File "/usr/local/go/src/net/http/server.go", line 1995 in http.HandlerFunc.ServeHTTP
9 File "vendor/github.com/shipkit-tofu/server/middleware/logging.go", line 98 in middleware.AttrPath.func1
10 File "/usr/local/go/src/net/http/server.go", line 1995 in http.HandlerFunc.ServeHTTP
11 File "vendor/github.com/gorilla/mux.go", line 212 in mux.(*Router).ServeHTTP
12 File "vendor/github.com/gorilla/mux.go", line 54 in handlers.(*Cors).ServeHTTP
13 File "vendor/github.com/urfave/negroni/negroni.go", line 46 in negroni.Wrap.func1
14 File "vendor/github.com/urfave/negroni/negroni.go", line 29 in negroni.HandlerFunc.ServeHTTP
15 File "vendor/github.com/urfave/negroni/negroni.go", line 38 in negroni.middleware.ServeHTTP
16 File "vendor/github.com/shipkit-tofu/server/middleware/tracing.go", line 39 in middleware.(*TracingMiddleware).ServeHTTP
17 File "vendor/github.com/urfave/negroni/negroni.go", line 38 in negroni.middleware.ServeHTTP
18 File "vendor/github.com/urfave/negroni/negroni.go", line 193 in negroni.(*Recovery).ServeHTTP
19 File "vendor/github.com/urfave/negroni/negroni.go", line 38 in negroni.middleware.ServeHTTP
20 File "vendor/github.com/shipkit-tofu/server/middleware/logging.go", line 66 in middleware.(*LoggingMiddleware).ServeHTTP
21 File "vendor/github.com/urfave/negroni/negroni.go", line 38 in negroni.middleware.ServeHTTP
22 File "vendor/github.com/urfave/negroni/negroni.go", line 96 in negroni.(*Negroni).ServeHTTP
23 File "/usr/local/go/src/net/http/server.go", line 2774 in http.HandlerFunc.ServeHTTP
24 File "/usr/local/go/src/net/http/server.go", line 1878 in http.(*Conn).serve
25 File "/usr/local/go/src/runtime/asm_amd64.s", line 1337 in runtime.goset
```

Observability

Observability is what is required in order to gain understanding of the application and infrastructure behavior in a production environment

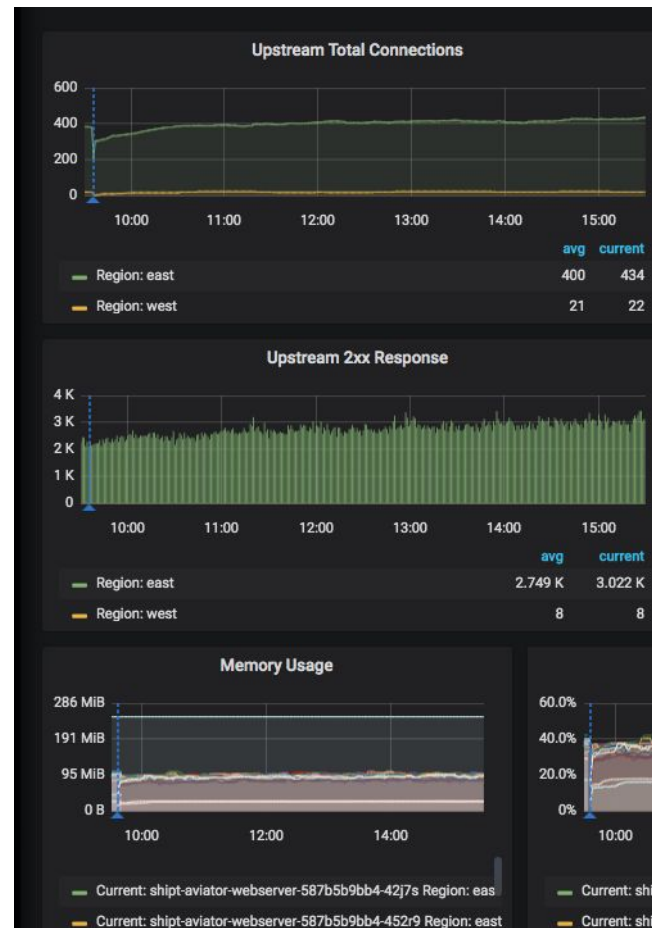


Metrics types (container & service levels)



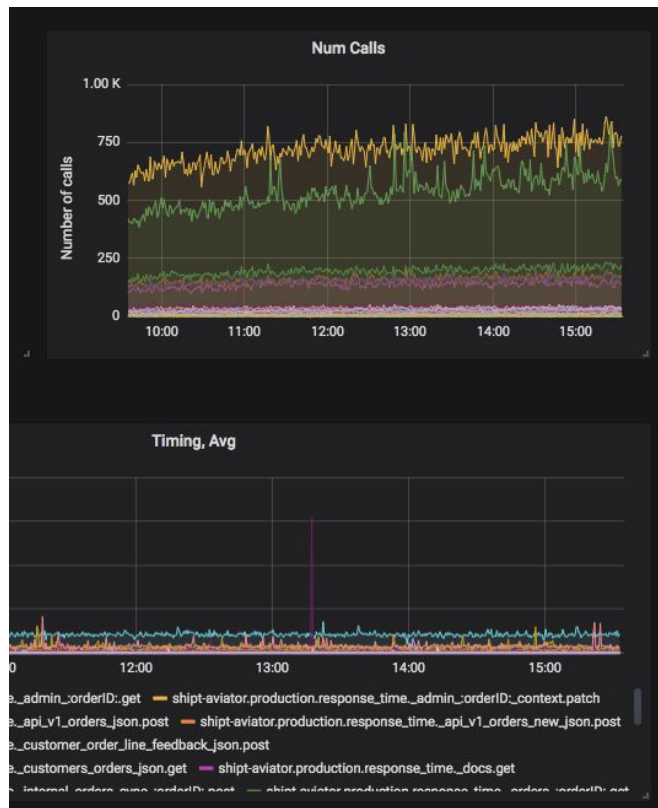
Container-level metrics

- Container-level telemetry provides information about container health wrt to Hardware and Network
- Data provided by:
 - AWS Load Balancers
 - Docker containers
 - K8s tools



Service-level metrics

- Service level telemetry provides business-specific information that is relevant to a service.
- Examples:
 - Latency data
 - Specific business functions, such as number of XYZ created
 - Number of calls and latency data to access datastores



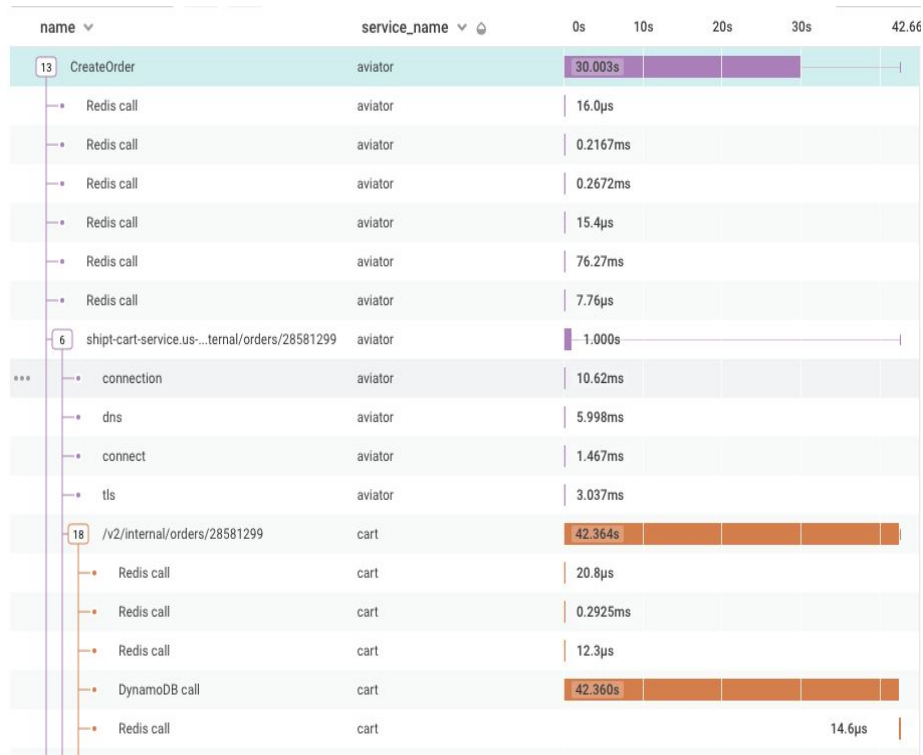
Logging (Scalyr / Splunk)

- Logs (typically taken from container stdout streams or output files) are combined into a single tool for easy searching
- Typically the most useful tool in case of an outage
- Always write your code with ability and control to have as much logging as possible

```
10:02:06.210 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 396640 Order has already been cancelled]
10:02:09.826 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: 215c
10:02:09.826 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 396640 Order has already been cancelled]
10:02:12.633 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: f01b
10:02:12.633 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 396646 Order has already been cancelled]
10:02:31.051 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: bd1a
10:02:31.051 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 396640 Order has already been cancelled]
10:02:39.204 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: 5919
10:02:39.204 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 340776 Order has already been cancelled]
10:02:45.270 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: 6bb7
10:02:45.270 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 396646 Order has already been cancelled]
10:02:51.191 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: 9d30
10:02:51.191 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 235997 Order has already been cancelled]
10:02:55.024 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: 46a4
10:02:55.024 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 235997 Order has already been cancelled]
10:03:01.328 app-production-east-1 shipt-aviator-webserver OfferServiceCall[curl -X 'GET' -H 'X-Cloud-Trace-Context: 35ea
10:03:01.328 app-production-east-1 shipt-aviator-webserver ClaimIgnore[28567190 396640 Order has already been cancelled]
```


Tracing / Distributed Tracing (Honeycomb)

- A trace represents a single user's journey through an entire stack
- Allows to track inefficiencies in the system and understand interaction between services
- Tracing tools provide UI to see how request goes through the system



Close the loop and learn from failures

- No blame Post-Mortems / Retrospectives
 - Each incident is followed by no blame post-mortem and creation of action items and new Clubhouse cards
- Always dedicate time to repay tech debt

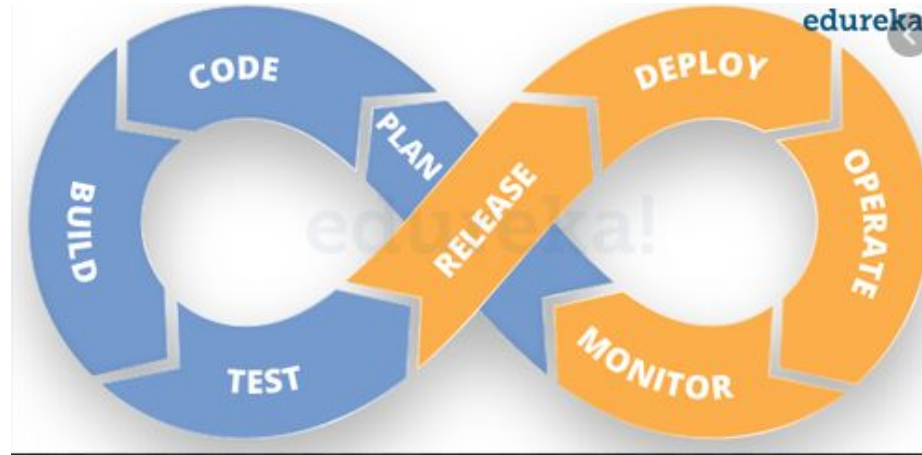
It's Not Your Fault

(Blameless)

post-mortems



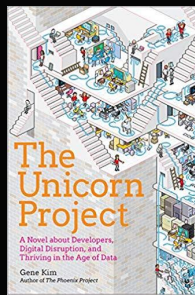
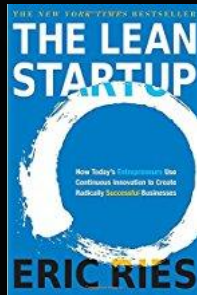
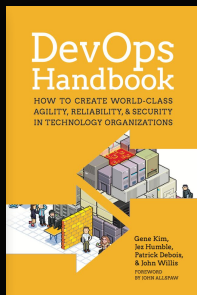
Questions?



DevOps Resources

<https://devopsfordefense.org/resources/>

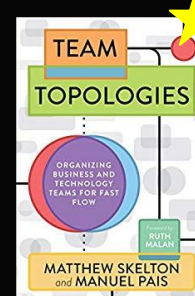
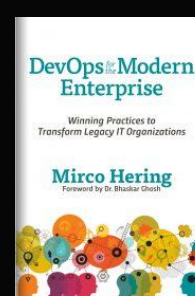
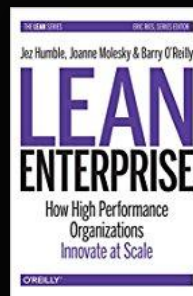
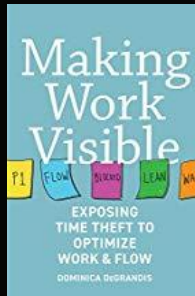
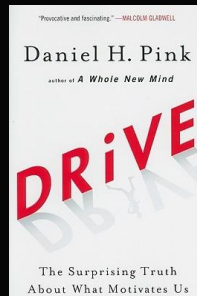
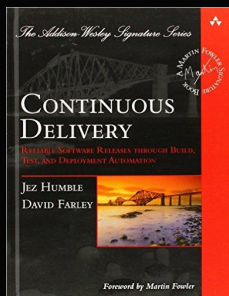
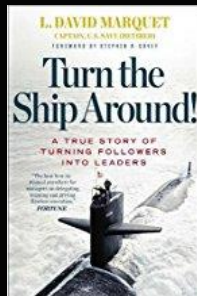
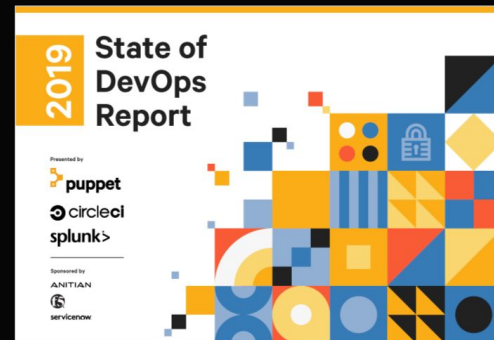
Books / Publications:



<https://www.meetup.com/DevOps-for-Defense/>
<https://github.com/jondavid-black/DevOpsForDefense>
devopsfordefense@gmail.com

Conference Presentations (YouTube):

- DevOps Enterprise Summit (DOES)
- IT Revolution
- Velocity
- GoTo



Group Exercise: Lean Coffee

Book Club: The Unicorn Project

1. Each table has a facilitator.
2. The facilitator has a short introduction.
3. Everyone write down questions or topics for discussion on the subject. Place them in the middle of the table.
4. The group votes on each question or topic by placing a dot on the card. 3 votes per person.
5. Cards with most dots goes first. Set a timer for 5 minutes and discuss.
6. After 5 minutes, either vote (thumbs up/down) to keep going or move on to the next card.



Suggested Topics: “Team Interactions”, “Better Tools”, “#NoMeetingFriday”