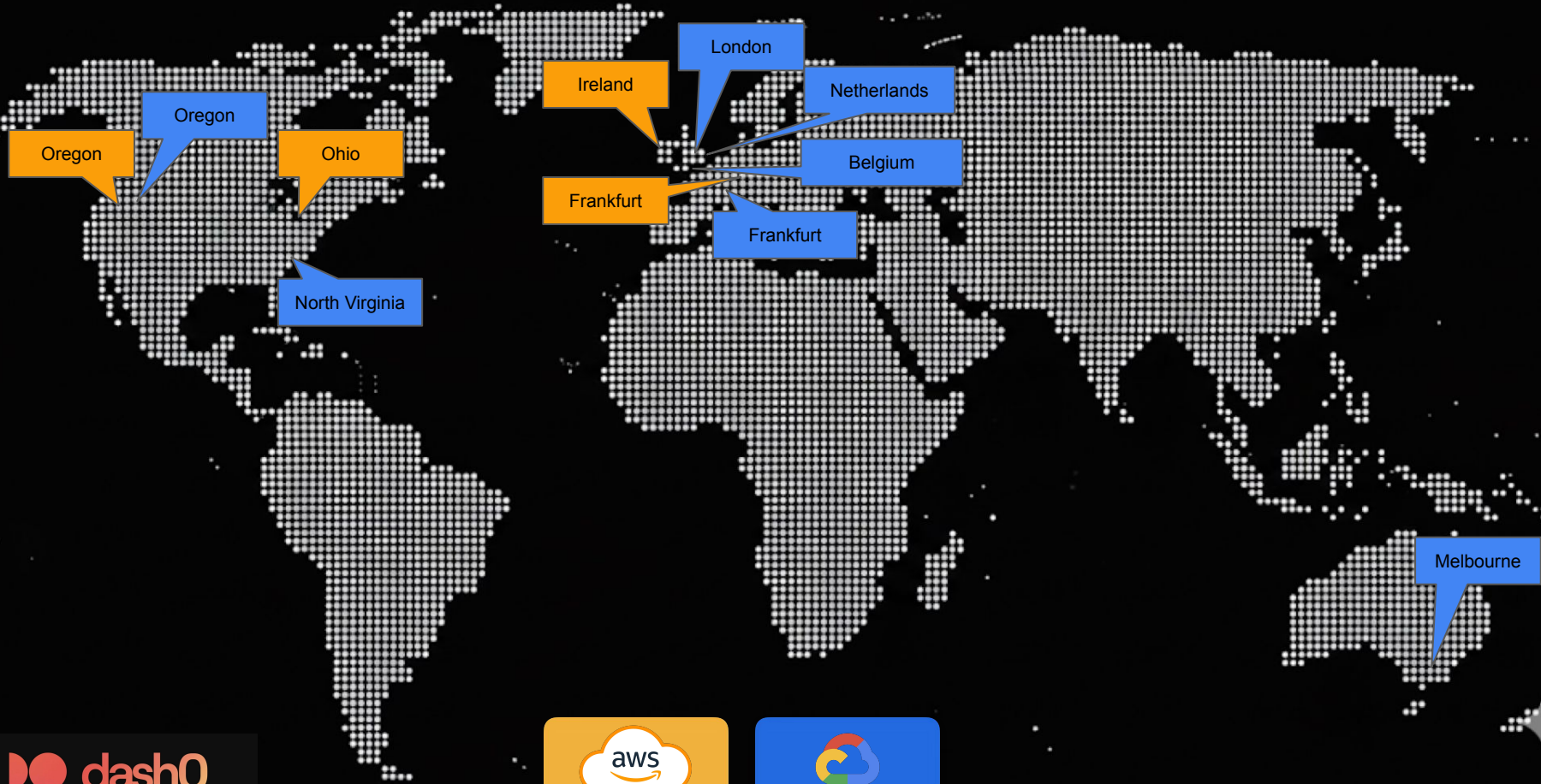


Quality Gates in Production

How We Turn OpenTelemetry Signals
into Deployment Decisions

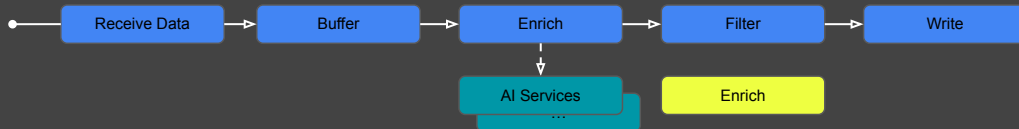


Dash0 Production Cloud Infrastructure 2025



- Home
- Agent0
- Dashboards
- Integrations
- Monitoring
- Resources
 - Map
 - Table
- Websites
- Synthetics
- Alerting
 - Failed checks
 - Checks
 - Notifications
- Telemetry
 - Logging
 - Metrics
 - Explorer
 - Query Builder
 - Tracing
 - Web Events

Ingestion Pipeline for each OpenTelemetry Signal (Logs, Metrics, Traces, Web Events, Web Spans)



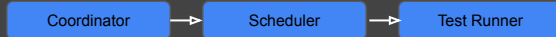
- GoLang
- Python
- Java

Alerting Pipeline



- NodeJS
- GoLang

Synthetics Pipeline



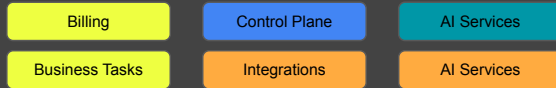
- GoLang

Frontend



- NodeJS
- GoLang
- NextJS

Others



- NodeJS
- GoLang
- Python
- Java

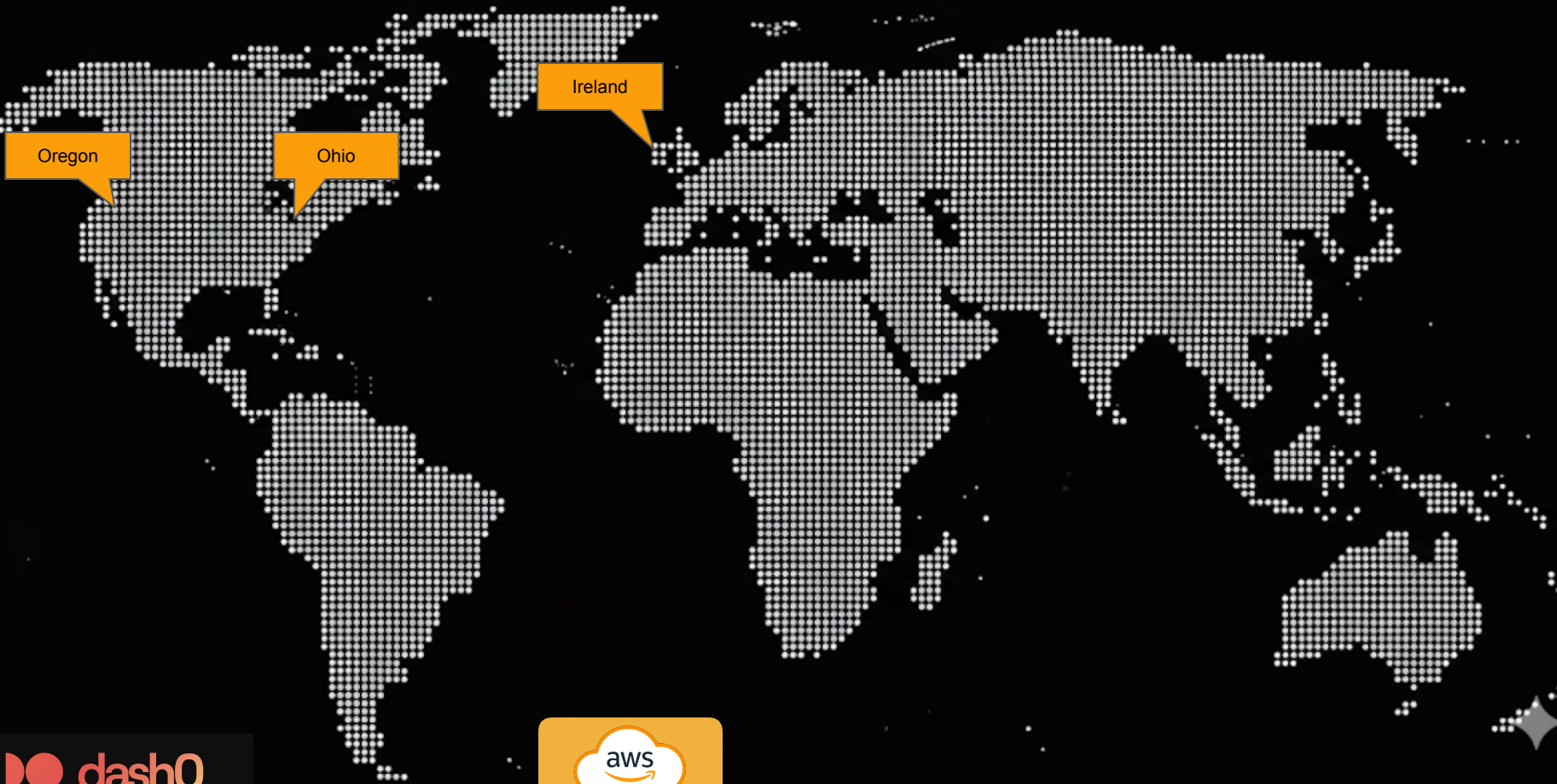
- As a startup **Speed is critical**. Release early and often
- We all want **Fast Deployments** and **Reliable Platforms**
- As an observability platform we need to be more **reliable** than the platforms of our customers that depend on us

- Gap between CI passing and successful production
 - **Backend engineers** (Unit tests, integration tests)
 - **Front End engineers** (Unit tests, E2E tests)
 - **SRE / Platform engineers** (Alerts, Synthetic Checks)



The Deployment Paradox

Dash0 Cloud Infrastructure 2024



01

CI/CD Pipeline 2024

Unit tests, integration tests, acceptance tests

ci-arm.yaml
on: push

Matrix: Test Product

13 jobs completed
Show all jobs

Matrix: Build Collector Images

1 job completed
Show all jobs

Release Helm Charts 3m 14s

Deploy Dev app.dash0... 6m 26s

Update docker-compose.y... 5s

Helm Tests

Test Helm Charts 1m 29s

Matrix: Build Images

11 jobs completed
Show all jobs

Matrix: Test Infra

3 jobs completed
Show all jobs

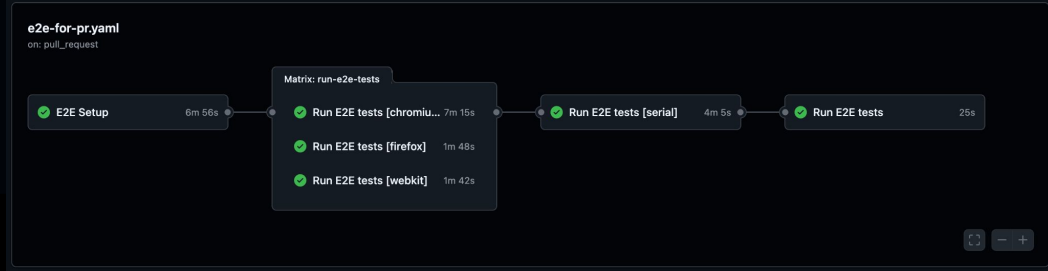


AWS ECR

OpenAPI Spec

Generate types 43s

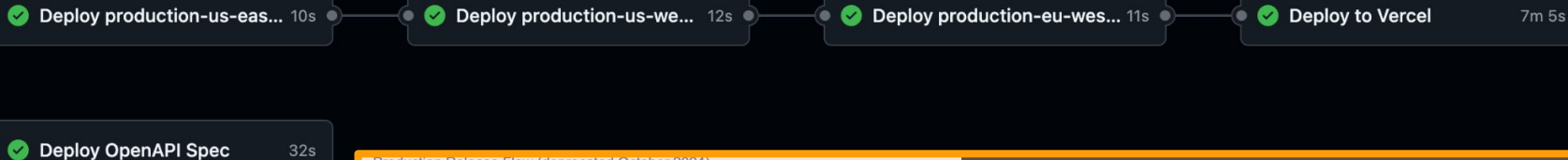
Lint GitHub Actions 14s



Production Deployment until October 2024

deploy.yaml

on: workflow_dispatch



Production Release Flow (deprecated October 2024)

[Walkthrough videos](#)

[Step 1: Coordinate with the team](#)

[Step 2: Check development environment](#)

[Step 2.1: Check logs](#)

[Step 2.2: Check traces](#)

[Step 2.3: Check alert Slack channels](#)

[Step 2.4: Final notification](#)

[Step 3: Run ClickHouse migrations \(if applicable\)](#)

[Step 4: "Approve and deploy" tagged code to production environment](#)

[Step 4.1: Open <https://github.com/dash0hq/dash0/actions/workflows/deploy.yaml>](#)

[Step 4.2: Select "Run workflow"](#)

[Step 4.3: Select the tag you want to deploy from, i.e. v1.1.108](#)

[Step 4.4: Open the new workflow](#)

[Step 4.5: Click "Review deployment" and approve deployment to global region](#)

[Step 4.6: Click "Review deployment" and approve deployment to us-west-2 region](#)

[Step 4.7: Click "Review deployment" and approve deployment to eu-west-1 region](#)

[Step 5: Slack Notifications](#)

[Step 6: Check production](#)

[Step 6.1: Check logs](#)

[Step 6.2: Check traces](#)

[Step 6.3: Check Slack alert channels](#)

[Step 6.4: Check production app](#)

[Step 7: Check ArgoCD environment files](#)

[Step 8: Communication](#)

[Rollback production deployment](#)

- Press a **single button**
- Deployment fully automated
- Follow checklist

Only 1-2 deployments per week

02

CI/CD Pipeline 2025

Production Deployment since October 2024

ci-arm.yaml

on: push



Matrix: Test Product

✔ 24 jobs completed

Show all jobs

Matrix: Test Infra

✔ 3 jobs completed

Show all jobs

✔ Test Helm Charts 4m 8s

✔ Generate types 2m 4s

✔ Require Passing Components 21s

Matrix: Build Collector Images

✔ 1 job completed

Show all jobs

Matrix: Build Images

✔ 15 jobs completed

Show all jobs

✔ Release Helm Charts 8m 39s

✔ Deploy Dev app.dash0-... 9m 8s

✔ Update docker-compose.y... 9s



Quality Gates

ci-arm.yaml

on: push



GitHub
Wait Timer

Matrix: QualityGate check Arg...

- ✔ QualityGate check ArgoC... 17s
- ✔ QualityGate check ArgoC... 21s
- ✔ QualityGate check ArgoC... 16s

Check that all components have new tag



Quality Gates

ci-arm.yaml

on: push



✔ QualityGate check Checkly... 3s

Check basic functionality

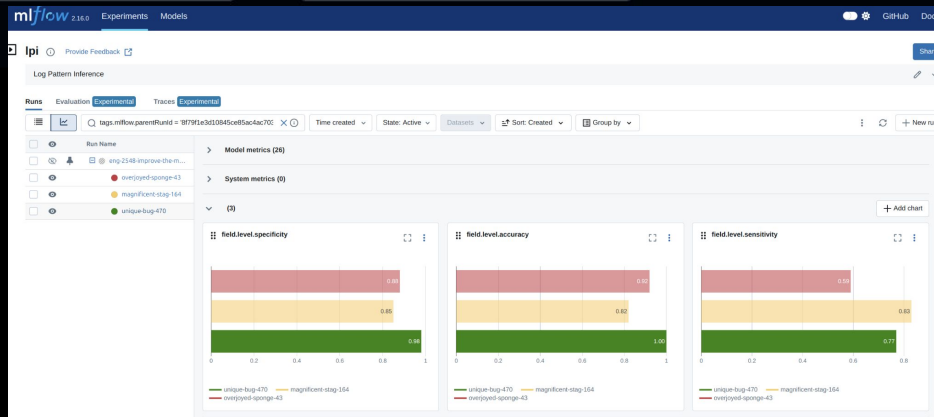
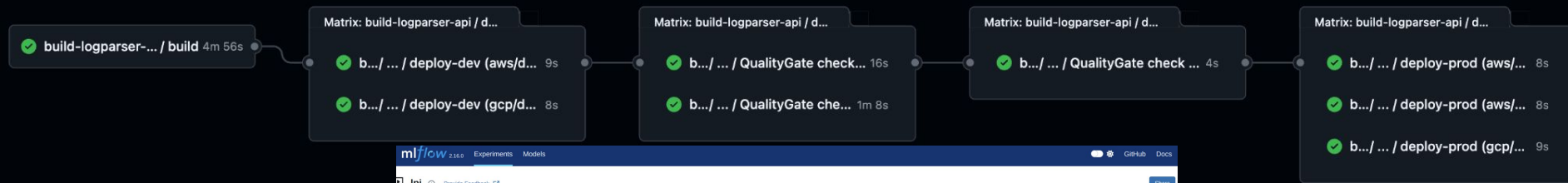
- @playwright/test
- Heartbeats



03

CI/CD Pipeline for **AI Services**

Quality Gates

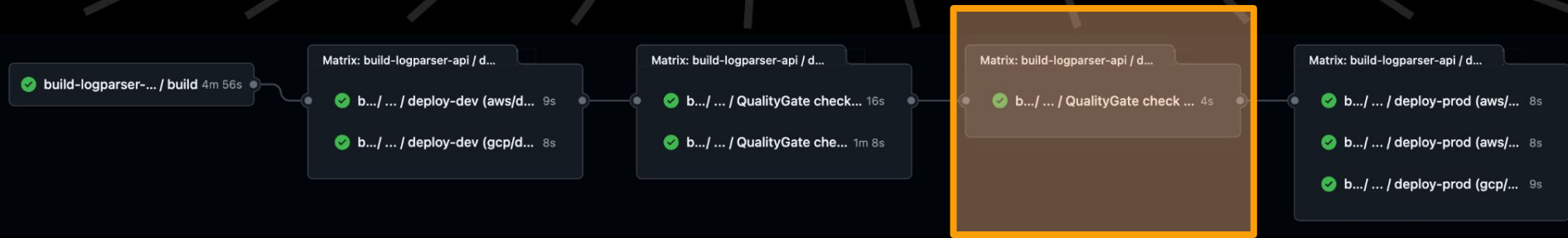


mlflow™

MLflow for managing machine learning lifecycle

- Version control for models
- Logs training runs - hyperparameters, metrics, code versions, and artifacts
- Compare results (i.e. log parsing)
 - High **specificity**: does it generate false positives
 - Max **accuracy**: is it correct
 - Good **sensitivity**: does it catch all cases

Quality Gates



Rule

Services Tracing Logging Metrics

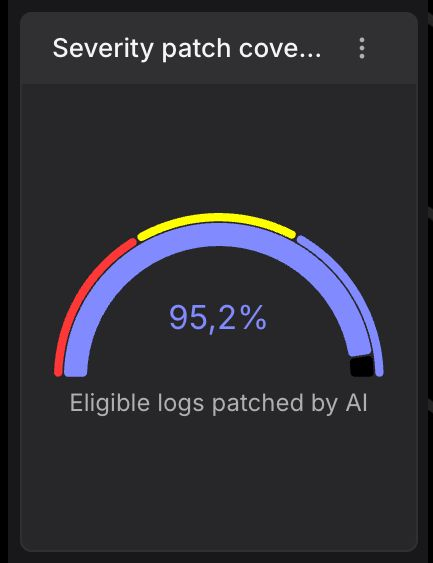
Name sre - ai-log-parser-api - Severity patch coverage % below threshold

Expression

```
1 sum by (k8s_cluster_name) (increase(processor_dash0logparser_processed_records_total{patched="true"}[10m]))
2 /
3 sum by (k8s_cluster_name) (increase(processor_dash0logparser_processed_records_total{}[10m])) < $_threshold
4
```

Thresholds

CRITICAL	0.85	🗑️
DEGRADED	0.9	🗑️

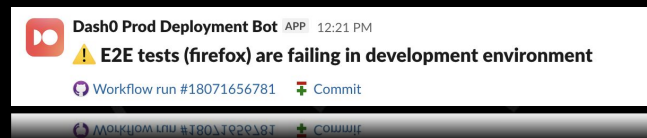


04

Lessons Learned

What is working well

- Production deployments are **not scary** anymore
 - All existing tests are used deployment decisions
- **End2End** tests are more stable
 - fixed quicker
- **Alert Rules** are tested and verified in development
 - better quality of alerts
- 5-10 prod deployments per day
 - Features and bug fixes
- **Staggered rollouts** by environments
 - Using regular alerts to verify production stability
- Migrated from GitHub Runners to Depot.dev
 - Instances cost ½ price
 - No rounding up to 1 min
 - Utilize **matrix builds** as much as possible



What is working well



Matthias Today at 7:01 AM

Hi 🙌 Is somebody already taking care about the failing quality gate regarding our prod deployment?



16 replies



Matthias Today at 7:02 AM

It would be nice to have a prod deployment today to ship check rule sharing 😊



joschi Today at 7:07 AM

Be the change you want to see in the world (and in the main branch)!



What is “mostly” working well

End2End tests cause most pain

- Expensive to run (last check in pipeline)
- Sequential execution for 3rd party APIs (rate limiting)
- **Retry Strategy**: 3 retries for regular tests, 5 retries for serial tests
- Generous Timeouts:
 - WebKit/Safari: 60 seconds
 - Screenshots: 60 seconds
- **Fixed Viewport**:
 - Consistent 1600x800 viewport across all browsers

Sometimes E2E tests are just trolling us ...

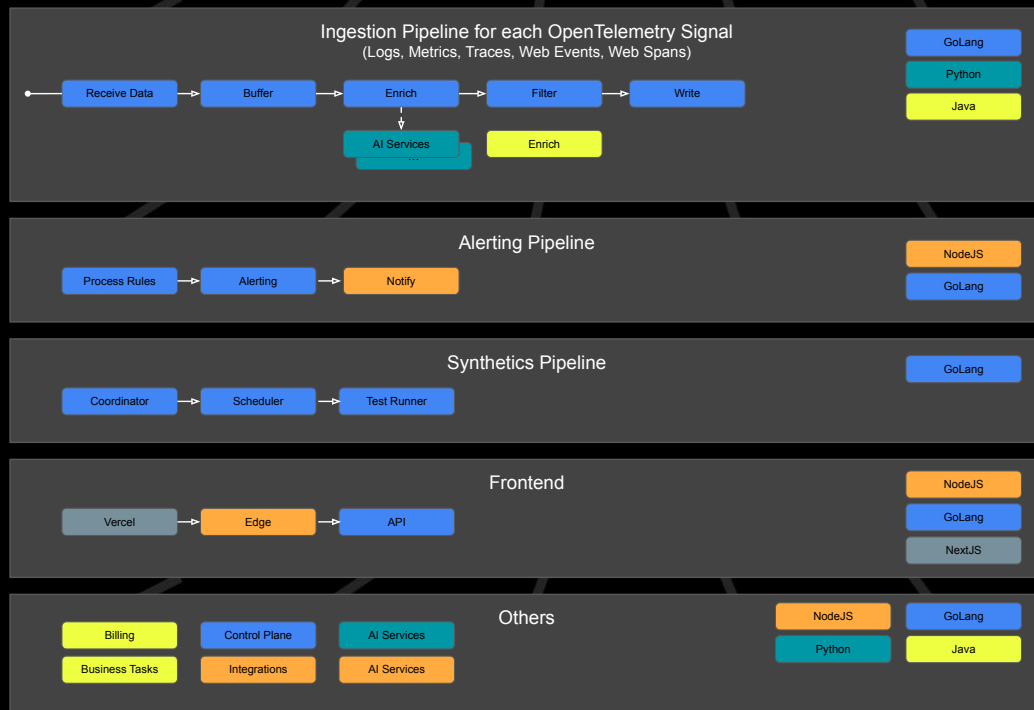
Matrix: run-e2e-tests

✘	Run E2E tests [chromi...	8m 20s
✘	Run E2E tests [firefox]	9m 31s
✔	Run E2E tests [webkit]	11m 20s

What is not working well

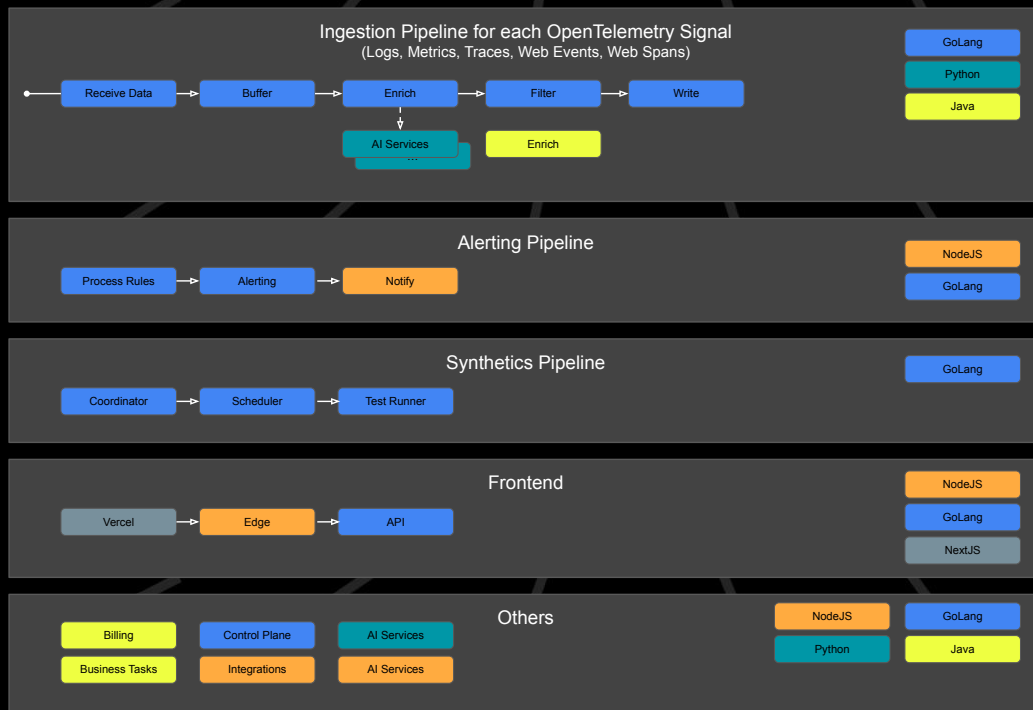
Status Quo
October 2025

Always update
everything



What is not working well

Currently evaluating tools like `nx` or `turborepo` if they would solve that problem



Only build and deploy when code has changed

Only build and deploy when code has changed

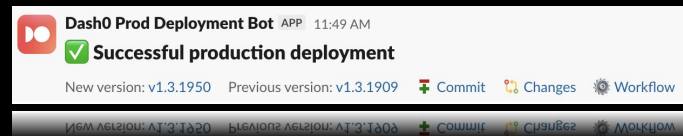
Only build and deploy when code has changed

Only build and deploy when code has changed

Only build and deploy when code has changed

Key recommendations

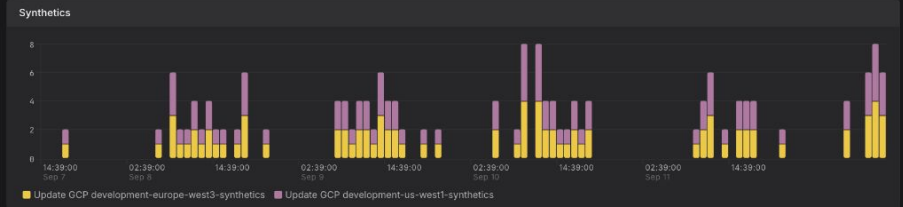
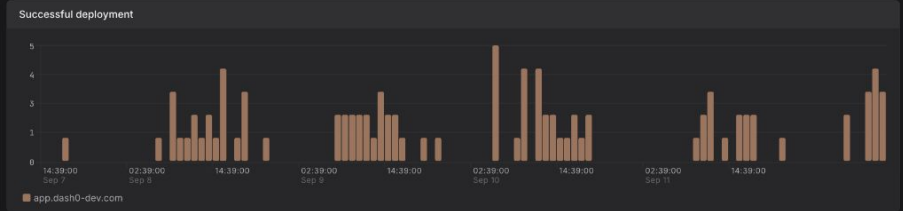
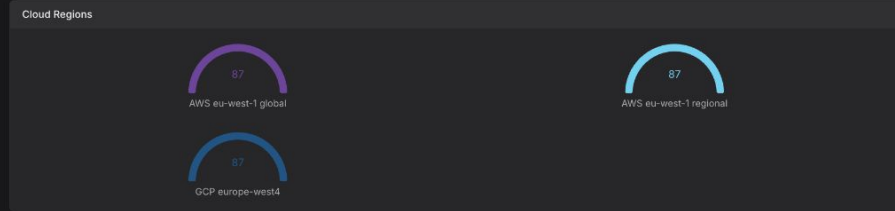
- Integrate existing test suites into deployment pipeline
 - Make use of what you already have
 - Then improve in iterations
- Keep development and production environments **consistent**
- Alert **configuration-as-code**
- **Production readiness checks** for new components. Must have:
 - Owner
 - Alert configuration (PrometheusAlerts)
 - ... with meaningful instructions
 - Dashboards configuration (PercyDashboard)
- Make deployments trivial
 - Automate deployment decisions
 - Inform team about deployments automatically



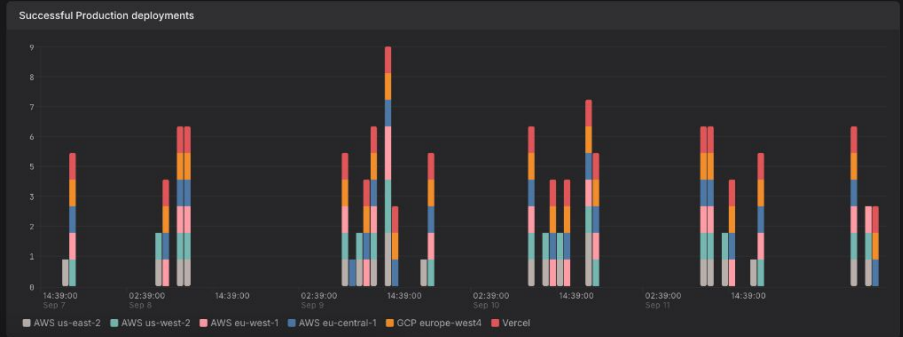
+ Add variable

+ Add panel Add group Share Edit as JSON Settings

Development Cloud Regions



Production Cloud Regions



Environments / Configure production-auto-deployment

Deployment protection rules

Configure reviewers, timers, and custom rules that must pass before deployments to this environment can proceed.

Required reviewers

Specify people or teams that may approve workflow runs when they access this environment.

Wait timer

Set an amount of time to wait before allowing deployments to proceed.

15

minutes

Enable custom rules with GitHub Apps [Preview](#)

[Learn about existing apps](#) or [create your own protection rules](#) so you can deploy with confidence.

The End

Thank you for your attention.

If you want to chat you can find me at the Dash0 booth or

<https://www.linkedin.com/in/marcelbirkner/>

