
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

OPN306

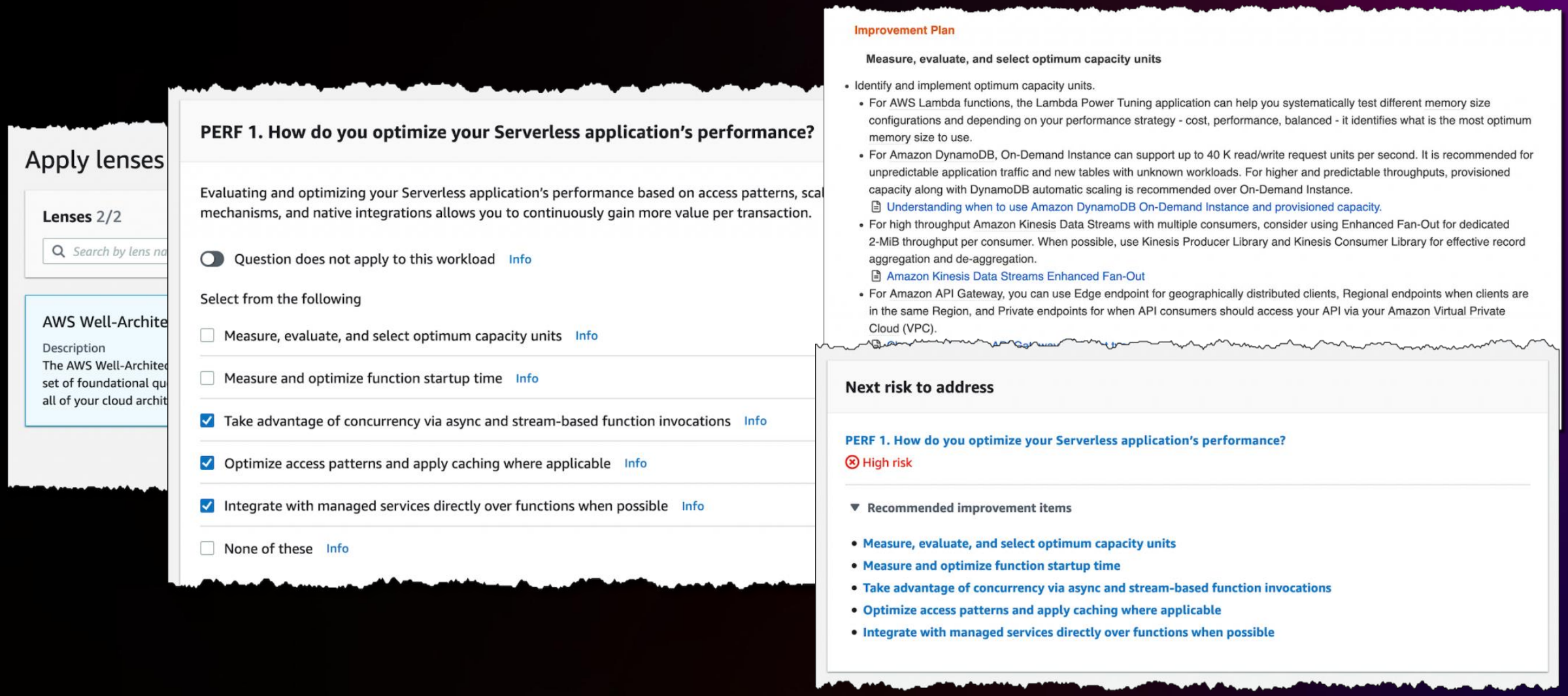
AWS Lambda Powertools: Lessons from the road to 10 million downloads

Heitor Lessa (he/him)

Principal Solutions Architect, Developer Acceleration (DevAx)
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

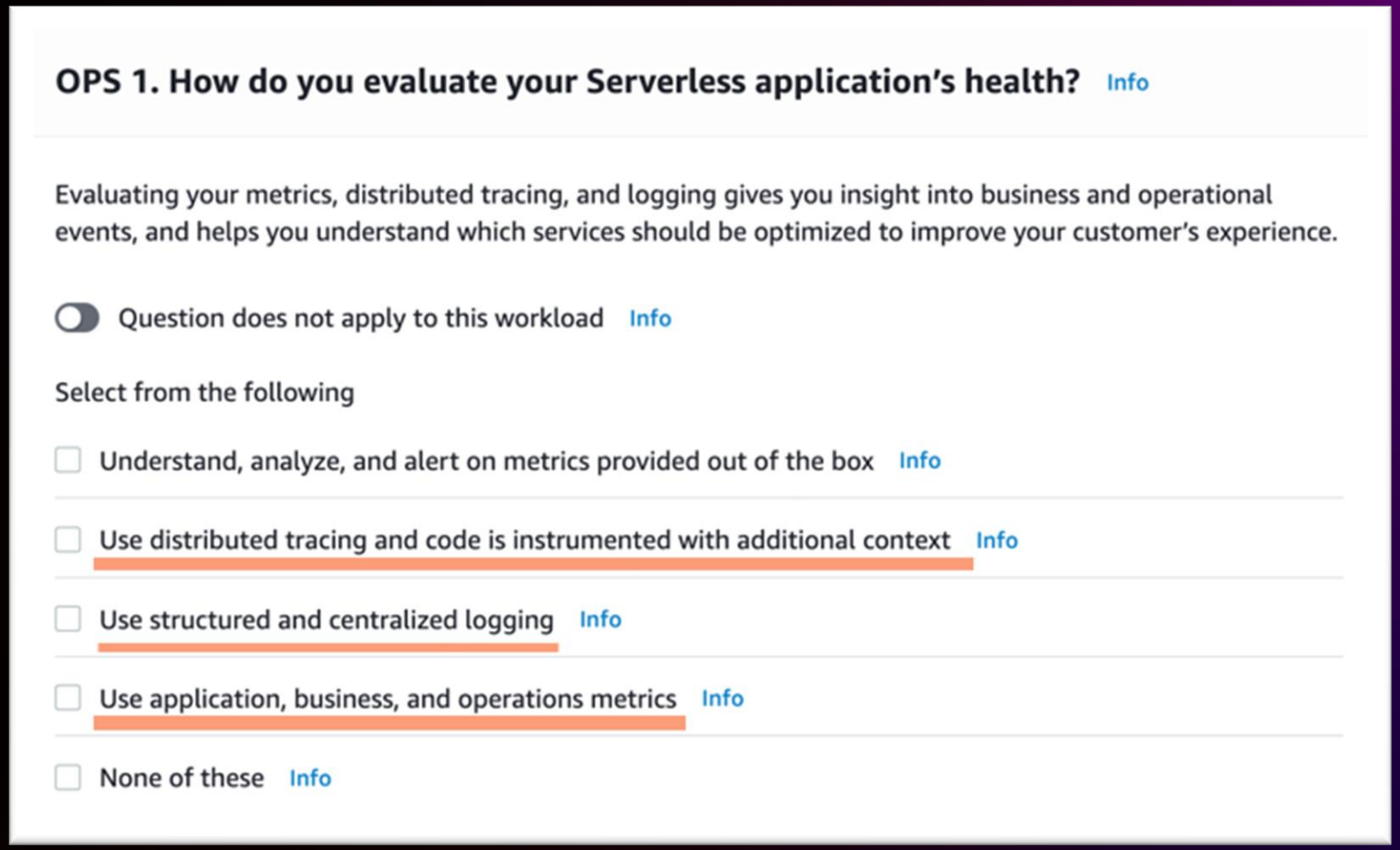
Serverless Lens in AWS WA Tool



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless Lens in AWS WA Tool

Most common risks



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Findings come in late



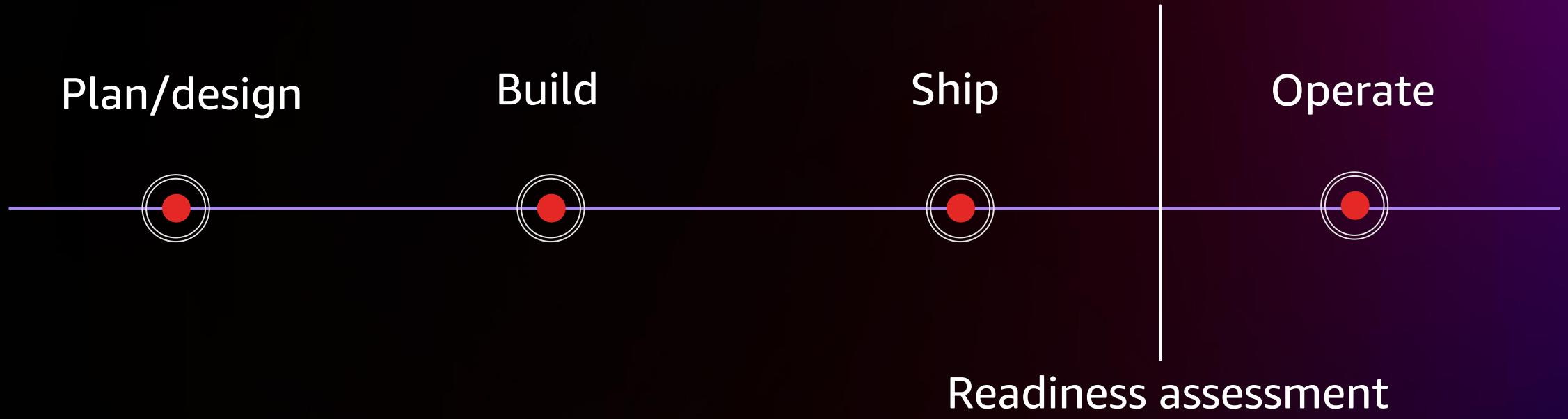
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Findings come in late



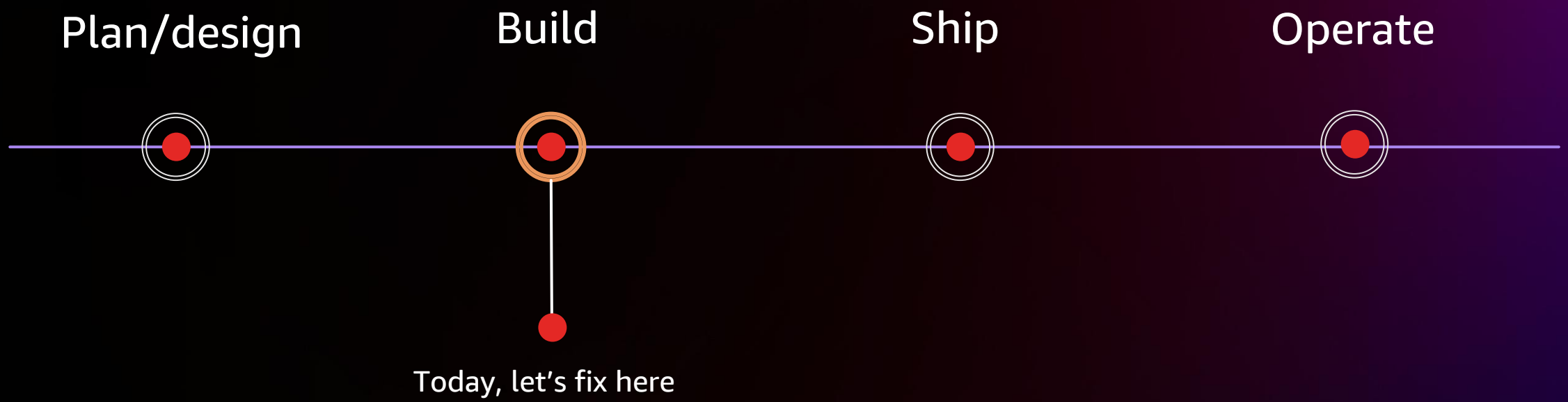
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Findings come in late



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Findings come in late



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda Powertools



- Logger
- Metrics
- Tracer
- + Many more

A toolkit to ease
leading practices adoption

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda Powertools

OPS1 solution

```python
from aws_lambda_powertools import Logger, Metrics, Tracer
from aws_lambda_powertools.logging import correlation_paths
from aws_lambda_powertools.metrics import MetricUnit
from aws_lambda_powertools.utilities.typing import LambdaContext

tracer = Tracer()
logger = Logger()
metrics = Metrics()

def decode_cart_data(payload: dict) -> dict: ...

@metrics.log_metrics(capture_cold_start_metric=True)
@tracer.capture_lambda_handler
@logger.inject_lambda_context(correlation_id_path=correlation_paths.API_GATEWAY_REST)
def lambda_handler(event: dict, context: LambdaContext):
    metrics.add_metric(name="CartCheckedOut", unit=MetricUnit.Count, value=1)

    cart_state = decode_cart_data(event)

    tracer.put_annotation(key="CustomerId", value=cart_state.get("customer_id", ""))
    metrics.add_metadata(key="cart_details", value=cart_state)

    ...

```

OPS 1: How do you evaluate your Serverless application's health?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

OPN306

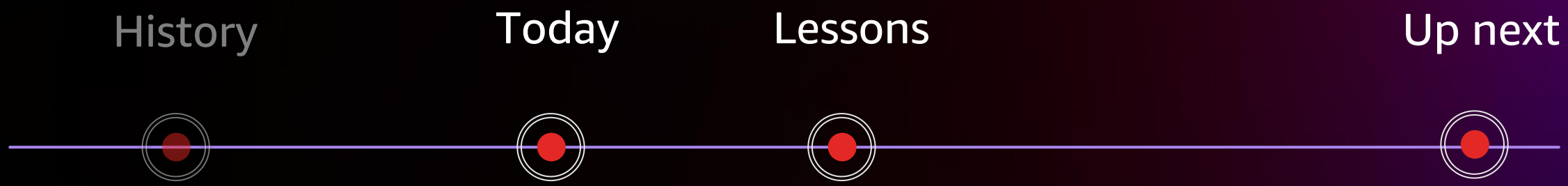
AWS Lambda Powertools: Lessons from the road to 10 million downloads

Heitor Lessa (he/him)

Principal Solutions Architect, Developer Acceleration (DevAx)
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Today

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Today: Serverless Lens mapping (51%)

Powertools assistance

Question	Practice
SEC 1	Scope access based on identity's metadata
SEC 2	Design smaller, single-purpose functions
SEC 2	Use temporary credentials between resources and components
SEC 3	Store secrets that are used in your code securely
REL 1	Use mechanisms to protect nonscalable resources
REL 2	Manage transaction, partial, and intermittent failures
REL 2	Manage duplicate and unwanted events
REL 2	Orchestrate long-running transactions
OPS 1	Use distributed tracing, and code is instrumented with additional context
OPS 1	Use structured and centralized logging
OPS 1	Use application, business, and operations metrics
OPS 2	Use infrastructure as code (IaC) and stages isolated in separate environments
OPS 2	Use configuration management
PERF 1	Measure and optimize function startup time
PERF 1	Take advantage of concurrency via async and stream-based function invocations
PERF 1	Optimize access patterns and apply caching where applicable
COST 1	Minimize external calls and function code initialization
COST 1	Optimize logging output and its retention
COST 1	Use cost-aware usage patterns in code
Total	19

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Today: Scale in numbers

6.5 billion

invocations MoM

161%

growth YoY

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Today: How did we get here?

Our customers grew

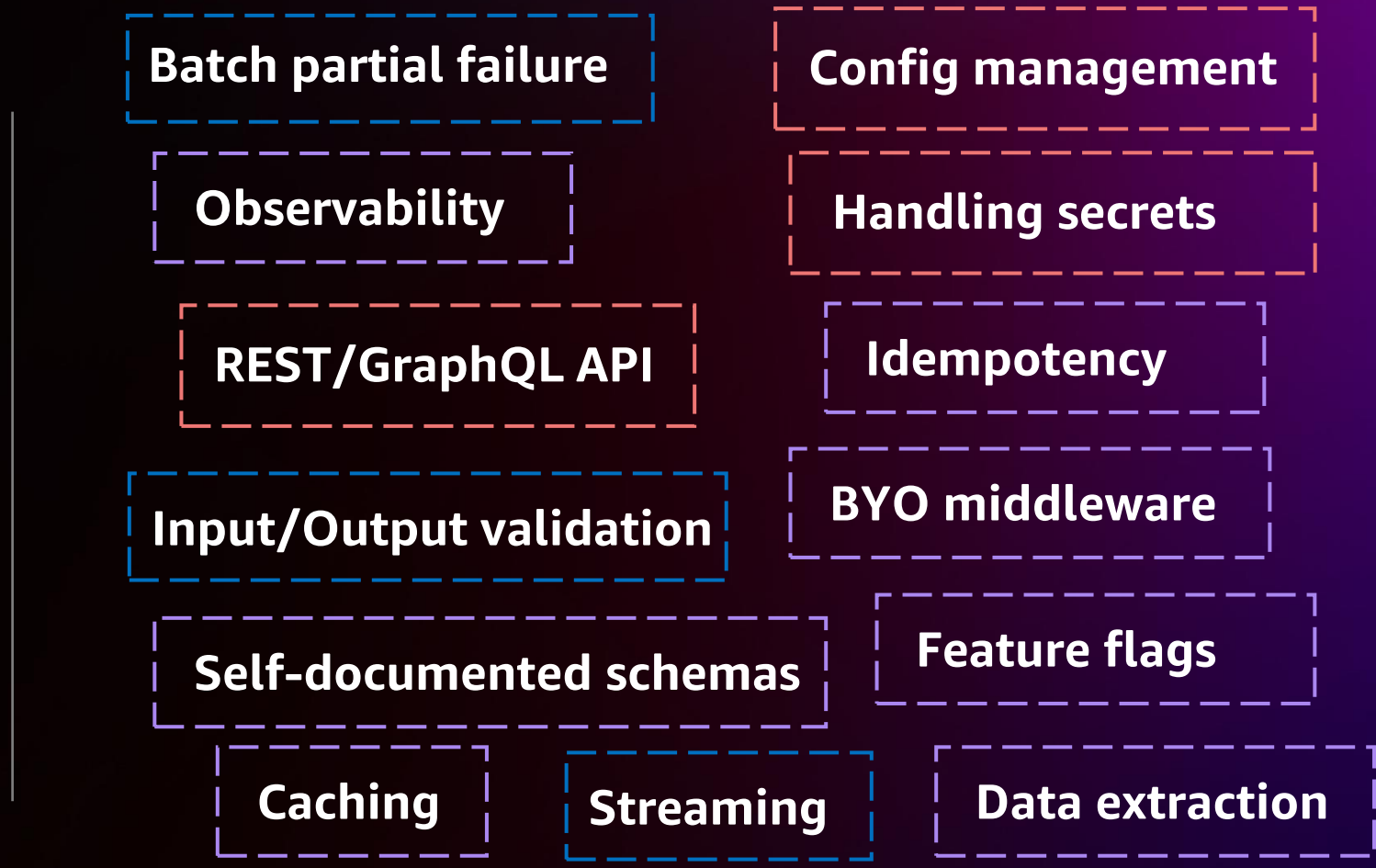
Observability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Today: How did we get here?

Our customers grew

The community grew too



~40% came from the community

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Today: How did we get here?

Our customers grew

The community grew too

Developers	APIs, event-driven architecture
Data engineers	ETL, data validation
Platform eng.	Automation, abstractions
Security eng.	AppSec, guardrails

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons: I am because we are

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release notes

Challenge

What's Changed

- chore(core): enable wildcard import for core top-level modules by @[redacted] in #1517
- chore(ci): migrate E2E tests to CDK CLI and off Docker by **@heitorlessa** in #1501
- feat(idempotency): support methods with the same name (ABCs) by including fully qualified name in v2 by **@leandrodamascena** in #1535
- feat(layers): add support for publishing v2 layer by **@rubenfonseca** in #1558
- docs(v2): document optional dependencies and local dev by **@heitorlessa** in #1574

New Contributors

- @[redacted] made their first contribution in #1517

Full Changelog: `v1.29.1...v1.30.1`

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release notes

Lesson

Summary

This release adds support for **(1)** Lambda context in Batch processing, **(2)** Context sharing support in REST and GraphQL APIs, and **(3)** Consolidate features to make prototyping local/non-prod environments easier. Bug fixes and other minor improvements are at the bottom.

```python
from typing import Optional

from aws_lambda_powertools.utilities.batch import (BatchProcessor, EventType,
                                                   batch_processor)
from aws_lambda_powertools.utilities.data_classes.sqs_event import SQSRecord
from aws_lambda_powertools.utilities.typing import LambdaContext

processor = BatchProcessor(event_type=EventType.SQS) # or DynamoDB, Kinesis

# lambda_context will be populated at runtime
def record_handler(record: SQSRecord, lambda_context: Optional[LambdaContext] = None): ...

@batch_processor(record_handler=record_handler, processor=processor)
def lambda_handler(event, context: LambdaContext):
    return processor.response()
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release notes

Lesson

New context support in Event Handler (REST and GraphQL)

> **Docs**

You can now inject contextual information before or during event routing. This is the precursor to Middleware support in Event Handler. This unlocks two common use cases:

- You split routes using `Router()` feature, and you need to pass data from `App()` without running into potential circular import issues
- You have custom logic to run before every event, and want to expose the result to any registered route with minimal effort

Thanks to @[redacted] for the feature request, and @[redacted], @[redacted], and **@walmsles** for the discussion to get to an optimal UX.

UX

> **Entrypoint: app.py**

```python
import todos

from aws_lambda_powertools.event_handler import APIGatewayRestResolver
from aws_lambda_powertools.utilities.typing import LambdaContext

app = APIGatewayRestResolver()
app.include_router(todos.router)


def lambda_handler(event: dict, context: LambdaContext) -> dict:
    app.append_context(is_admin=True)  # arbitrary number of key=value data
    return app.resolve(event, context)
```

> **todos.py**

```python
import requests
from requests import Response

from aws_lambda_powertools.event_handler.api_gateway import Router


ENDPOINT = "https://jsonplaceholder.typicode.com/todos"

router = Router()

@router.get("/todos")
def get_todos():
    is_admin: bool = router.context.get("is_admin", False)
    todos = {}

    if is_admin:
        todos: Response = requests.get(ENDPOINT)
        todos.raise_for_status()
        todos = todos.json()[:10]

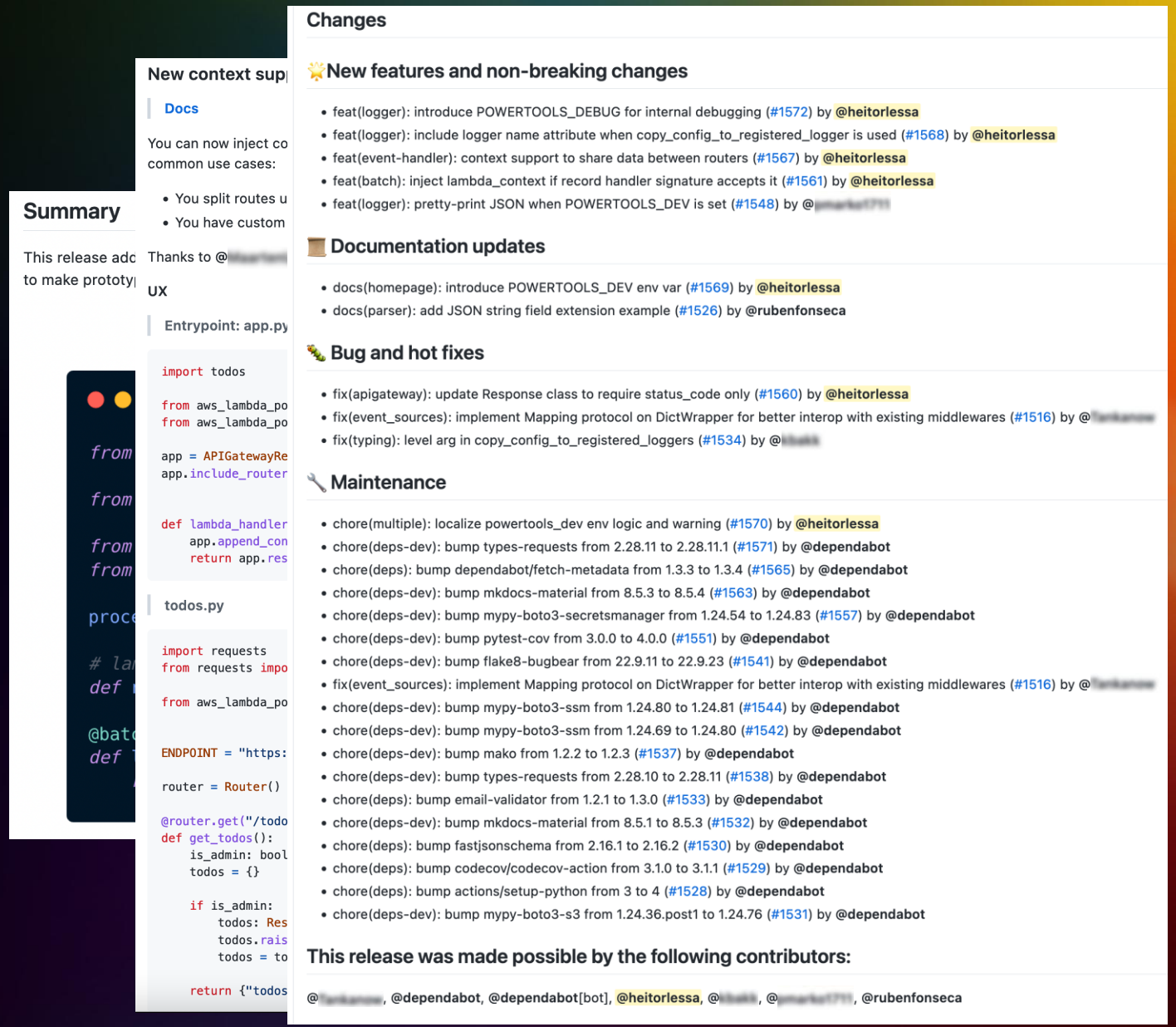
    return {"todos": todos}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release notes

Lesson



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Ambiguity

Challenge

- Language idioms?
- Can I run on Fargate too?
- What about X dependency?
- Breaking changes?
- When to cut your losses?
- Best vs. emerging vs. leading practice
- When too many opinions are harmful

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons



Ambiguity

Lesson

Tenets
AWS Lambda only
Eases the adoption of best practices
Keep it lean
We strive for backwards compatibility
We work backwards from the community
Progressive

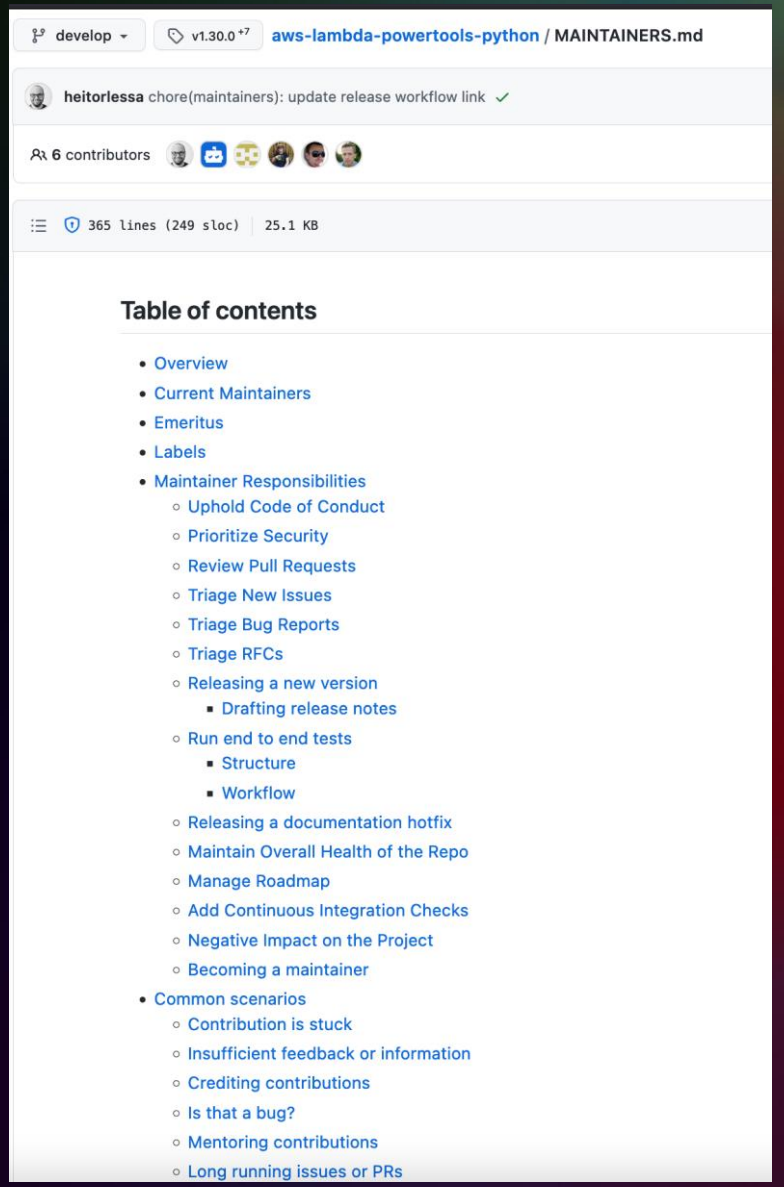
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Ambiguity

Lesson


develop v1.30.0 +7 aws-lambda-powertools-python / MAINTAINERS.md

heitorlessa chore(maintainers): update release workflow link

6 contributors

365 lines (249 sloc) 25.1 KB

Table of contents

- Overview
- Current Maintainers
- Emeritus
- Labels
- Maintainer Responsibilities
  - Uphold Code of Conduct
  - Prioritize Security
  - Review Pull Requests
  - Triage New Issues
  - Triage Bug Reports
  - Triage RFCs
  - Releasing a new version
    - Drafting release notes
  - Run end to end tests
    - Structure
    - Workflow
  - Releasing a documentation hotfix
  - Maintain Overall Health of the Repo
  - Manage Roadmap
  - Add Continuous Integration Checks
  - Negative Impact on the Project
  - Becoming a maintainer
- Common scenarios
  - Contribution is stuck
  - Insufficient feedback or information
  - Crediting contributions
  - Is that a bug?
  - Mentoring contributions
  - Long running issues or PRs

Thanks to our OpenSearch friends

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Ambiguity

Lesson



Labels

These are the most common labels used by maintainers to triage issues, pull requests (PR), and for project management:

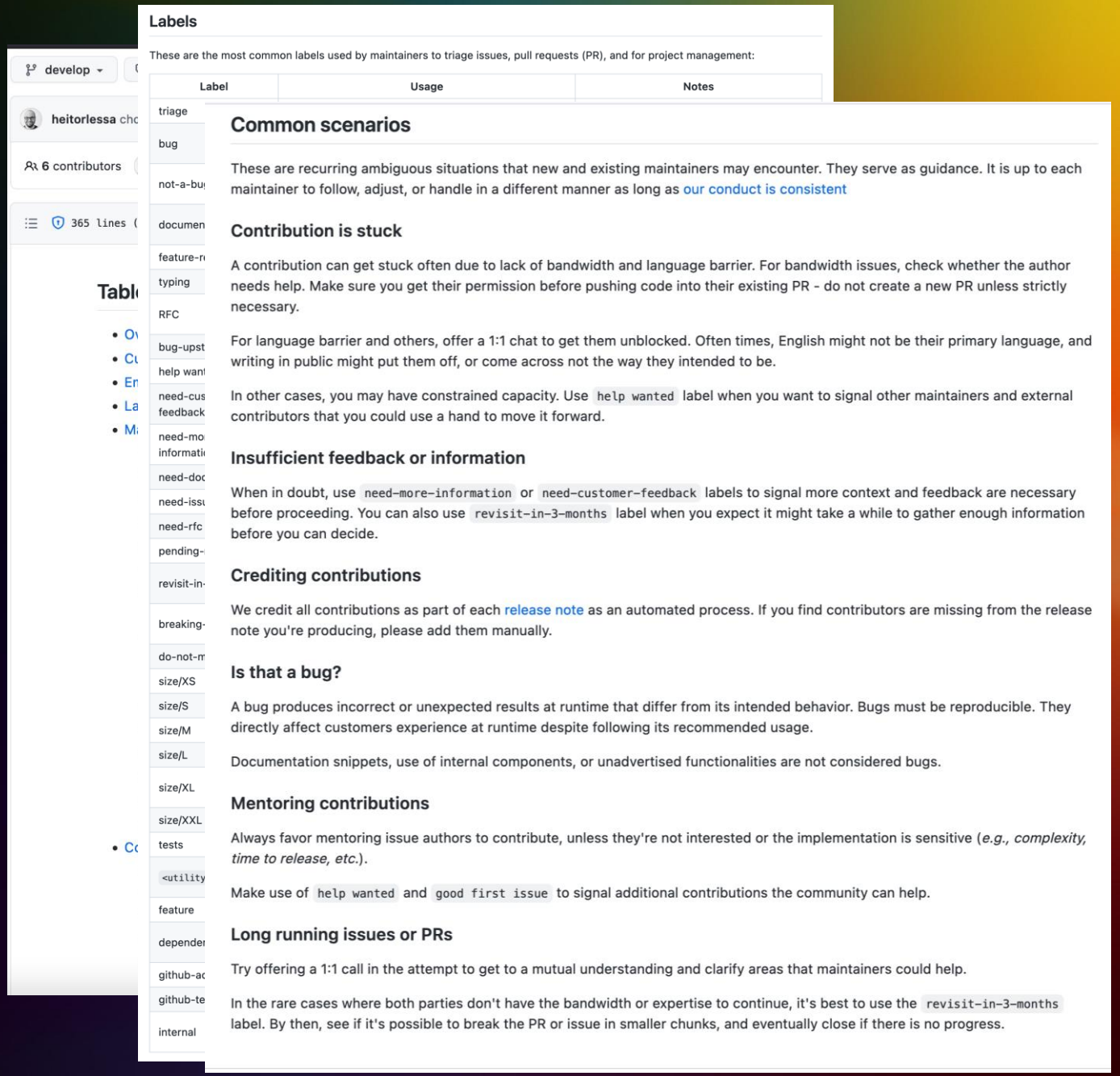
Label	Usage	Notes
triage	New issues that require maintainers review	Issue template
bug	Unexpected, reproducible and unintended software behavior	PR/Release automation; Doc snippets are excluded;
not-a-bug	New and existing bug reports incorrectly submitted as bug	Analytics
documentation	Documentation improvements	PR/Release automation; Doc additions, fixes, etc.;
feature-request	New or enhancements to existing features	Issue template
typing	New or enhancements to static typing	Issue template
RFC	Technical design documents related to a feature request	Issue template
bug-upstream	Bug caused by upstream dependency	
help wanted	Tasks you want help from anyone to move forward	Bandwidth, complex topics, etc.
need-customer-feedback	Tasks that need more feedback before proceeding	80/20% rule, uncertain, etc.
need-more-information	Missing information before making any calls	
need-documentation	PR is missing or has incomplete documentation	
need-issue	PR is missing a related issue for tracking change	Needs to be automated
need-rfc	Feature request requires a RFC to improve discussion	
pending-release	Merged changes that will be available soon	Release automation auto-closes/notifies it
revisit-in-3-months	Blocked issues/PRs that need to be revisited	Often related to `need-customer-feedback`, prioritization, etc.
breaking-change	Changes that will cause customer impact and need careful triage	
do-not-merge	PRs that are blocked for varying reasons	Timeline is uncertain
size/XS	PRs between 0-9 LOC	PR automation
size/S	PRs between 10-29 LOC	PR automation
size/M	PRs between 30-99 LOC	PR automation
size/L	PRs between 100-499 LOC	PR automation
size/XL	PRs between 500-999 LOC, often PRs that grown with feedback	PR automation
size/XXL	PRs with 1K+ LOC, largely documentation related	PR automation
tests	PRs that add or change tests	PR automation
`<utility>`	PRs related to a Powertools utility, e.g. `parameters`, `tracer`	PR automation
feature	New features or minor changes	PR/Release automation
dependencies	Changes that touch dependencies, e.g. Dependabot, etc.	PR/ automation
github-actions	Changes in GitHub workflows	PR automation
github-templates	Changes in GitHub issue/PR templates	PR automation
internal	Changes in governance, tech debt and chores (linting setup, baseline, etc.)	PR automation

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Ambiguity

Lesson



develop

heitorlessa cho

6 contributors

365 lines (

Tabl

- Ov
- Cu
- En
- La
- Ma
- Co

Labels

These are the most common labels used by maintainers to triage issues, pull requests (PR), and for project management:

Label	Usage	Notes
triage		
bug		
not-a-bu		
documen		
feature-r		
typing		
RFC		
bug-upst		
help want		
need-cus feedback		
need-mo informati		
need-doc		
need-issu		
need-rfc		
pending-		
revisit-in-		
breaking-		
do-not-m		
size/XS		
size/S		
size/M		
size/L		
size/XL		
size/XXL		
tests		
<utility		
feature		
dependen		
github-ac		
github-te		
internal		

Common scenarios

These are recurring ambiguous situations that new and existing maintainers may encounter. They serve as guidance. It is up to each maintainer to follow, adjust, or handle in a different manner as long as our conduct is consistent

Contribution is stuck

A contribution can get stuck often due to lack of bandwidth and language barrier. For bandwidth issues, check whether the author needs help. Make sure you get their permission before pushing code into their existing PR - do not create a new PR unless strictly necessary.

For language barrier and others, offer a 1:1 chat to get them unblocked. Often times, English might not be their primary language, and writing in public might put them off, or come across not the way they intended to be.

In other cases, you may have constrained capacity. Use `help wanted` label when you want to signal other maintainers and external contributors that you could use a hand to move it forward.

Insufficient feedback or information

When in doubt, use `need-more-information` or `need-customer-feedback` labels to signal more context and feedback are necessary before proceeding. You can also use `revisit-in-3-months` label when you expect it might take a while to gather enough information before you can decide.

Crediting contributions

We credit all contributions as part of each release note as an automated process. If you find contributors are missing from the release note you're producing, please add them manually.

Is that a bug?

A bug produces incorrect or unexpected results at runtime that differ from its intended behavior. Bugs must be reproducible. They directly affect customers experience at runtime despite following its recommended usage.

Documentation snippets, use of internal components, or unadvertised functionalities are not considered bugs.

Mentoring contributions

Always favor mentoring issue authors to contribute, unless they're not interested or the implementation is sensitive (*e.g., complexity, time to release, etc.*).

Make use of `help wanted` and `good first issue` to signal additional contributions the community can help.

Long running issues or PRs

Try offering a 1:1 call in the attempt to get to a mutual understanding and clarify areas that maintainers could help.

In the rare cases where both parties don't have the bandwidth or expertise to continue, it's best to use the `revisit-in-3-months` label. By then, see if it's possible to break the PR or issue in smaller chunks, and eventually close if there is no progress.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Contribution

Challenge

- Bugfix?
- Discussion?
- New features?
- What about docs?
- Sample apps?
- hacktoberfest?
- Case study?
- Automation improvements?
- What is a contribution after all?

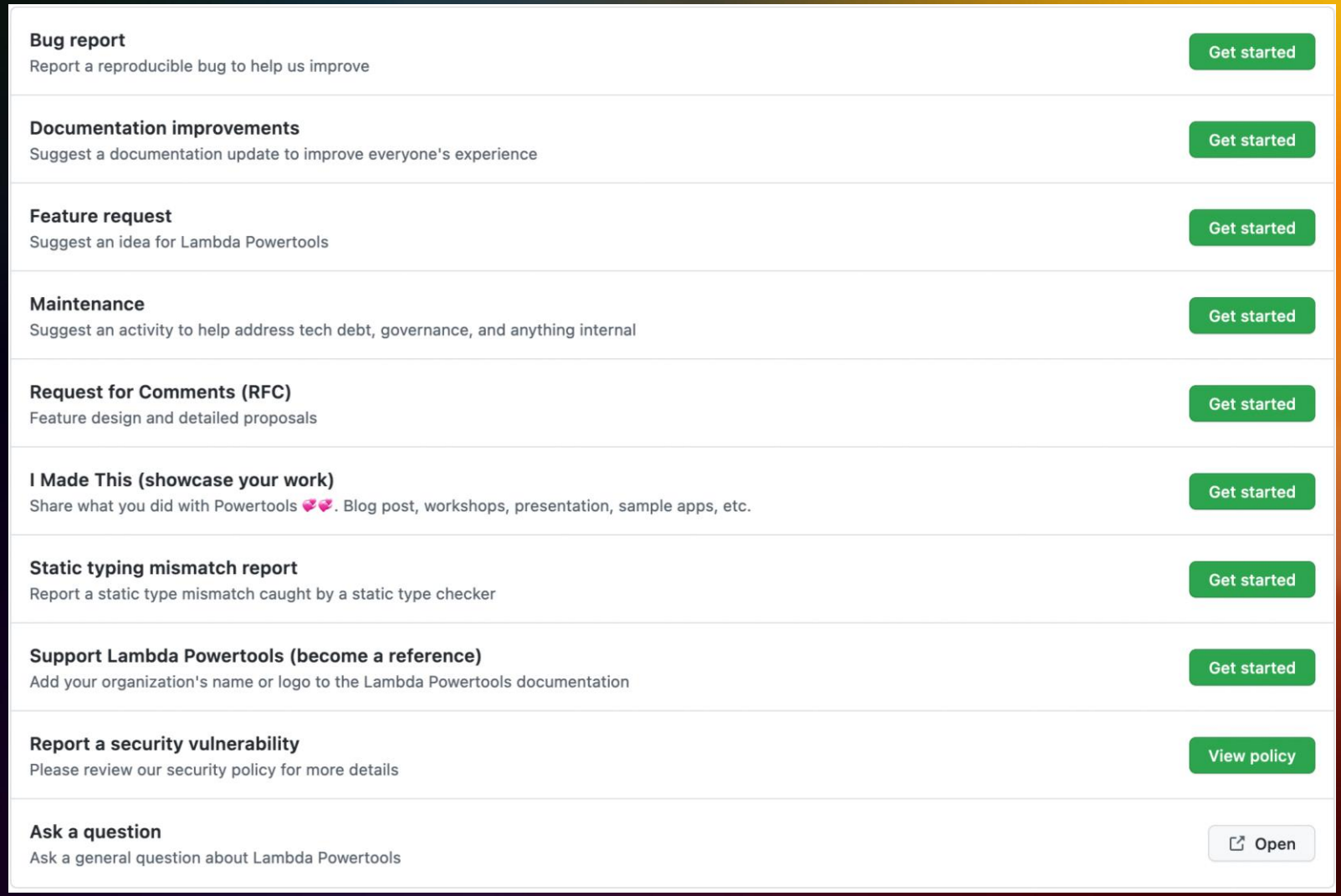
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Contribution

Lesson



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons



Contribution

Lesson

pre-commit

+

make pr

Shortened **feedback loop** and consistency at scale

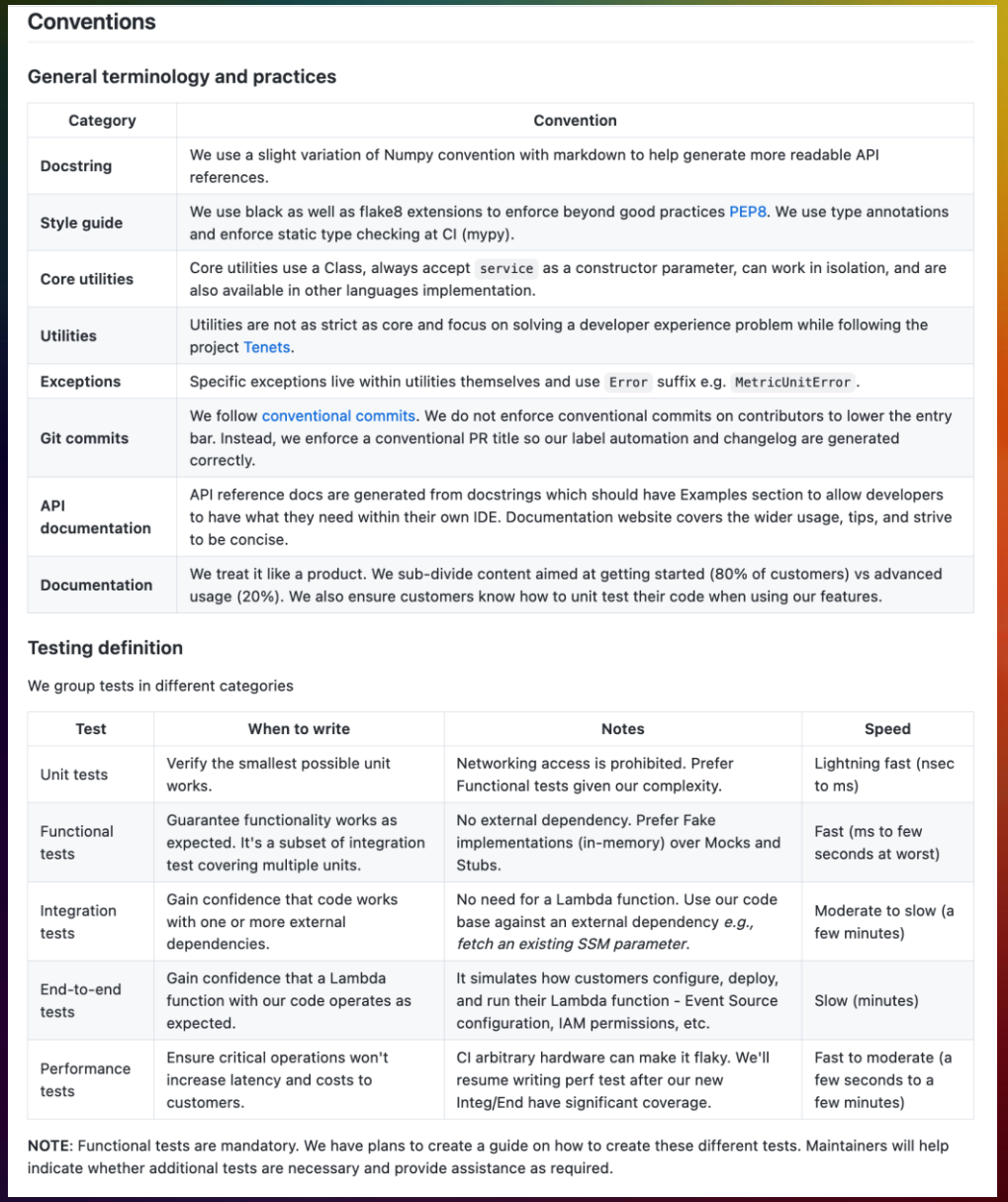
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Contribution

lesson



Conventions

General terminology and practices

Category	Convention
Docstring	We use a slight variation of Numpy convention with markdown to help generate more readable API references.
Style guide	We use black as well as flake8 extensions to enforce beyond good practices PEP8. We use type annotations and enforce static type checking at CI (mypy).
Core utilities	Core utilities use a Class, always accept `service` as a constructor parameter, can work in isolation, and are also available in other languages implementation.
Utilities	Utilities are not as strict as core and focus on solving a developer experience problem while following the project Tenets.
Exceptions	Specific exceptions live within utilities themselves and use `Error` suffix e.g. `MetricUnitError`.
Git commits	We follow conventional commits. We do not enforce conventional commits on contributors to lower the entry bar. Instead, we enforce a conventional PR title so our label automation and changelog are generated correctly.
API documentation	API reference docs are generated from docstrings which should have Examples section to allow developers to have what they need within their own IDE. Documentation website covers the wider usage, tips, and strive to be concise.
Documentation	We treat it like a product. We sub-divide content aimed at getting started (80% of customers) vs advanced usage (20%). We also ensure customers know how to unit test their code when using our features.

Testing definition

We group tests in different categories

Test	When to write	Notes	Speed
Unit tests	Verify the smallest possible unit works.	Networking access is prohibited. Prefer Functional tests given our complexity.	Lightning fast (nsec to ms)
Functional tests	Guarantee functionality works as expected. It's a subset of integration test covering multiple units.	No external dependency. Prefer Fake implementations (in-memory) over Mocks and Stubs.	Fast (ms to few seconds at worst)
Integration tests	Gain confidence that code works with one or more external dependencies.	No need for a Lambda function. Use our code base against an external dependency *e.g., fetch an existing SSM parameter.*	Moderate to slow (a few minutes)
End-to-end tests	Gain confidence that a Lambda function with our code operates as expected.	It simulates how customers configure, deploy, and run their Lambda function - Event Source configuration, IAM permissions, etc.	Slow (minutes)
Performance tests	Ensure critical operations won't increase latency and costs to customers.	CI arbitrary hardware can make it flaky. We'll resume writing perf test after our new Integ/End have significant coverage.	Fast to moderate (a few seconds to a few minutes)

NOTE: Functional tests are mandatory. We have plans to create a guide on how to create these different tests. Maintainers will help indicate whether additional tests are necessary and provide assistance as required.

Expose conventions and definitions early in the process

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Tech writing

Challenge

Like coding, it's another competency . . . that can be learned

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Tech writing

Lesson

Co-authorship & editing mechanisms

Line editing, copywriting, 1:1, etc.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Tech writing

Lesson

Strategic Writing for UX by Torrey Podmajersky

Voice chart, style guide, diverse audience, etc.

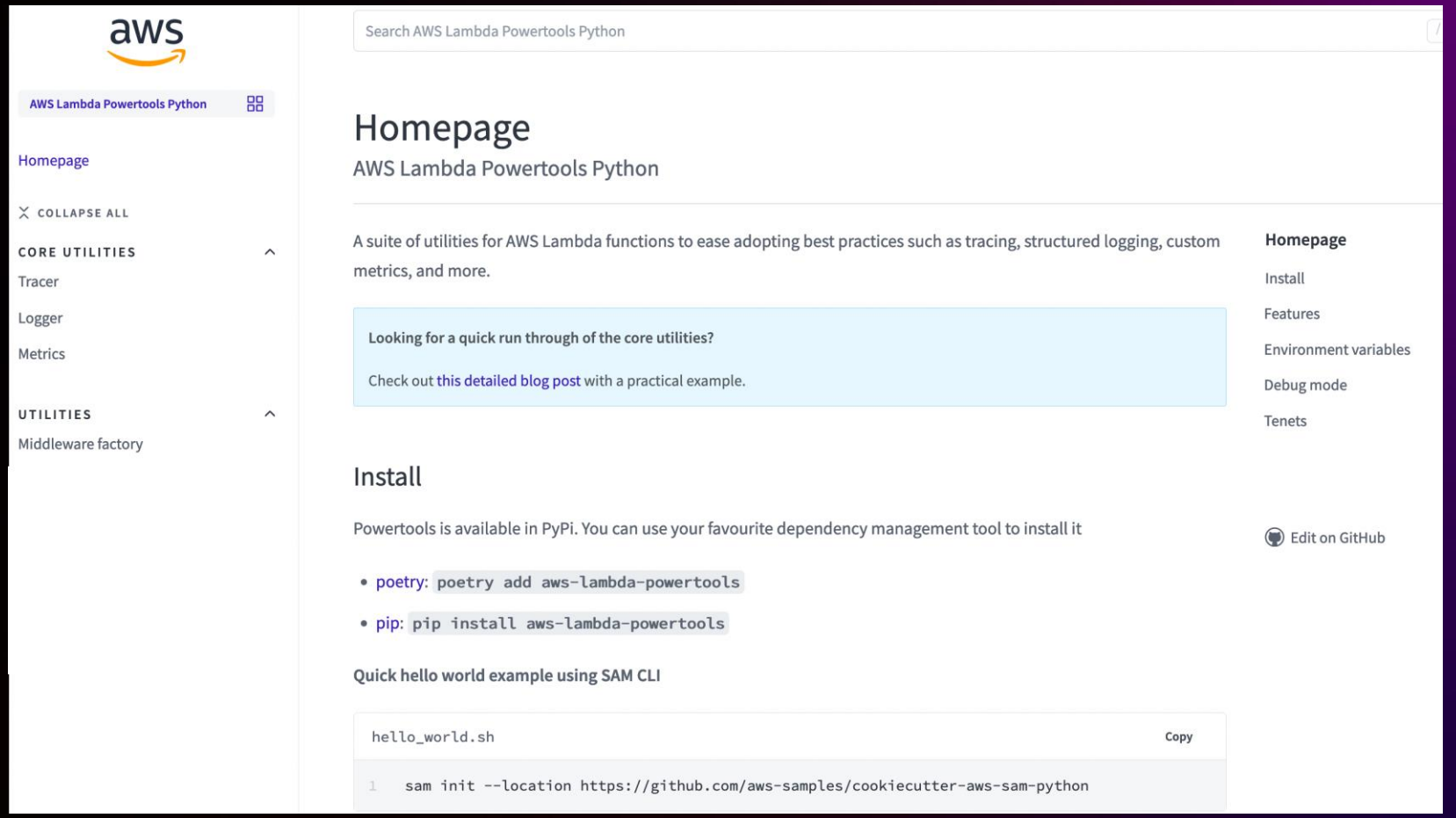
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Docs growth

Challenge



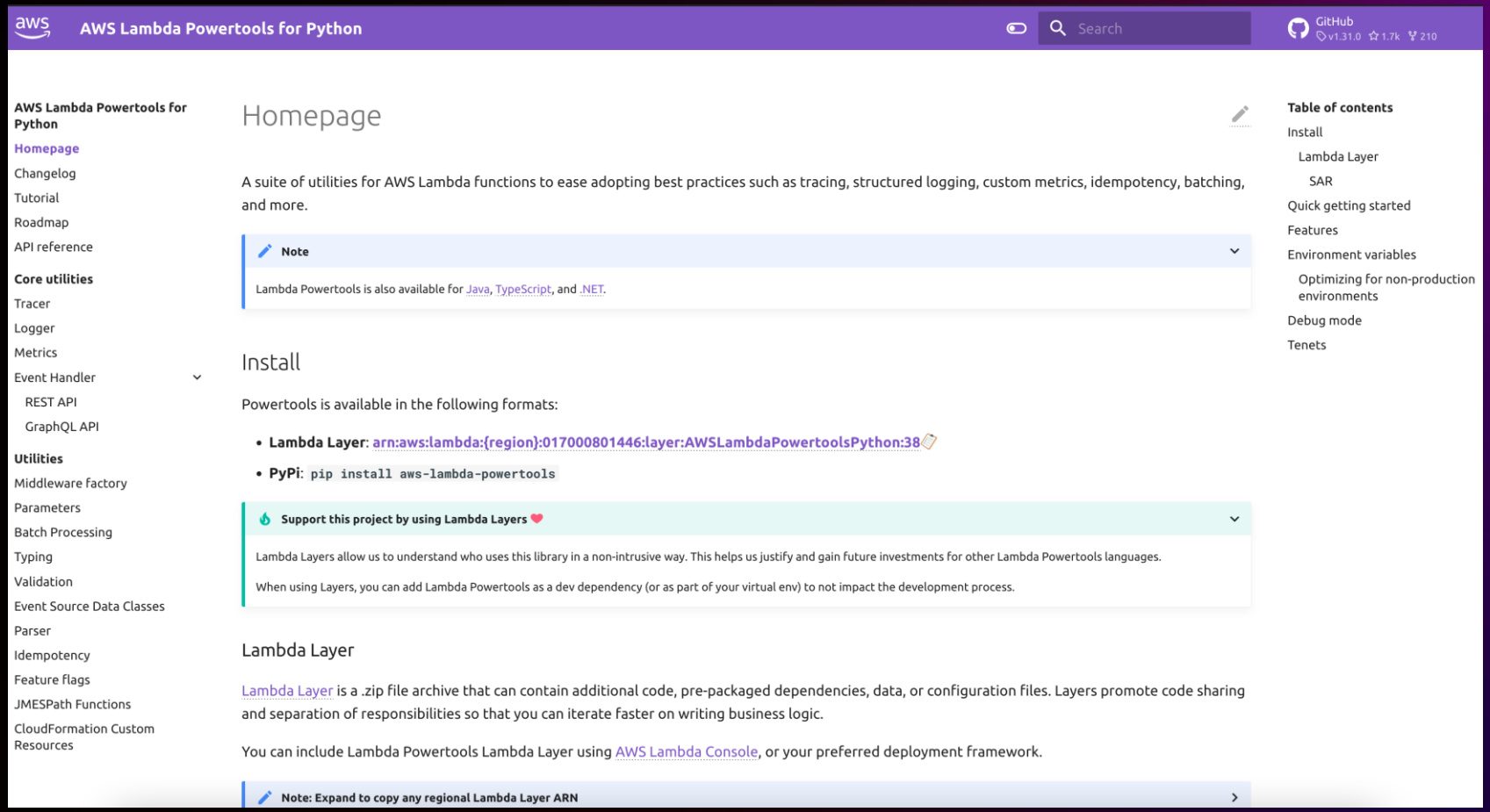
Gatsby Apollo theme 💔

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Docs growth

Challenge



mkdocs-material ♡

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Examples growth

Challenge

app.py | hello_event.json | response.json

```python
from aws_lambda_powertools import Logger, Tracer
from aws_lambda_powertools.logging import correlation_paths
from aws_lambda_powertools.event_handler import APIGatewayRestResolver

tracer = Tracer()
logger = Logger()
app = APIGatewayRestResolver()

@app.get("/hello")
@tracer.capture_method
def get_hello_universe():
    return {"message": "hello universe"}

# You can continue to use other utilities just as before
@logger.inject_lambda_context(correlation_id_path=correlation_paths.API_GATEWAY_REST)
@tracer.capture_lambda_handler
def lambda_handler(event, context):
    return app.resolve(event, context)
```

From a few to 300+

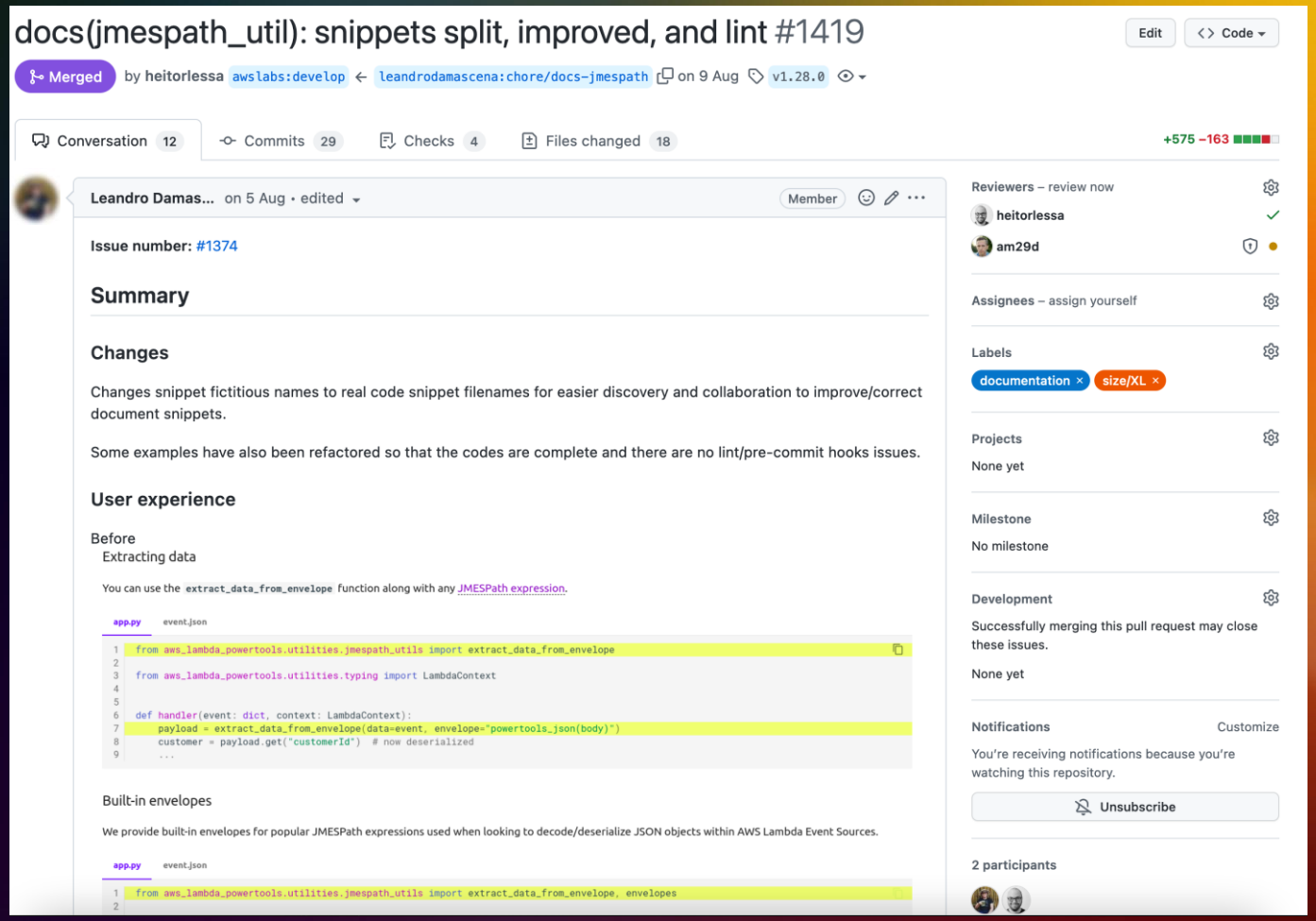
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Examples growth

Lesson



Gradual lint and realistic uplift

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Consistency

Challenge

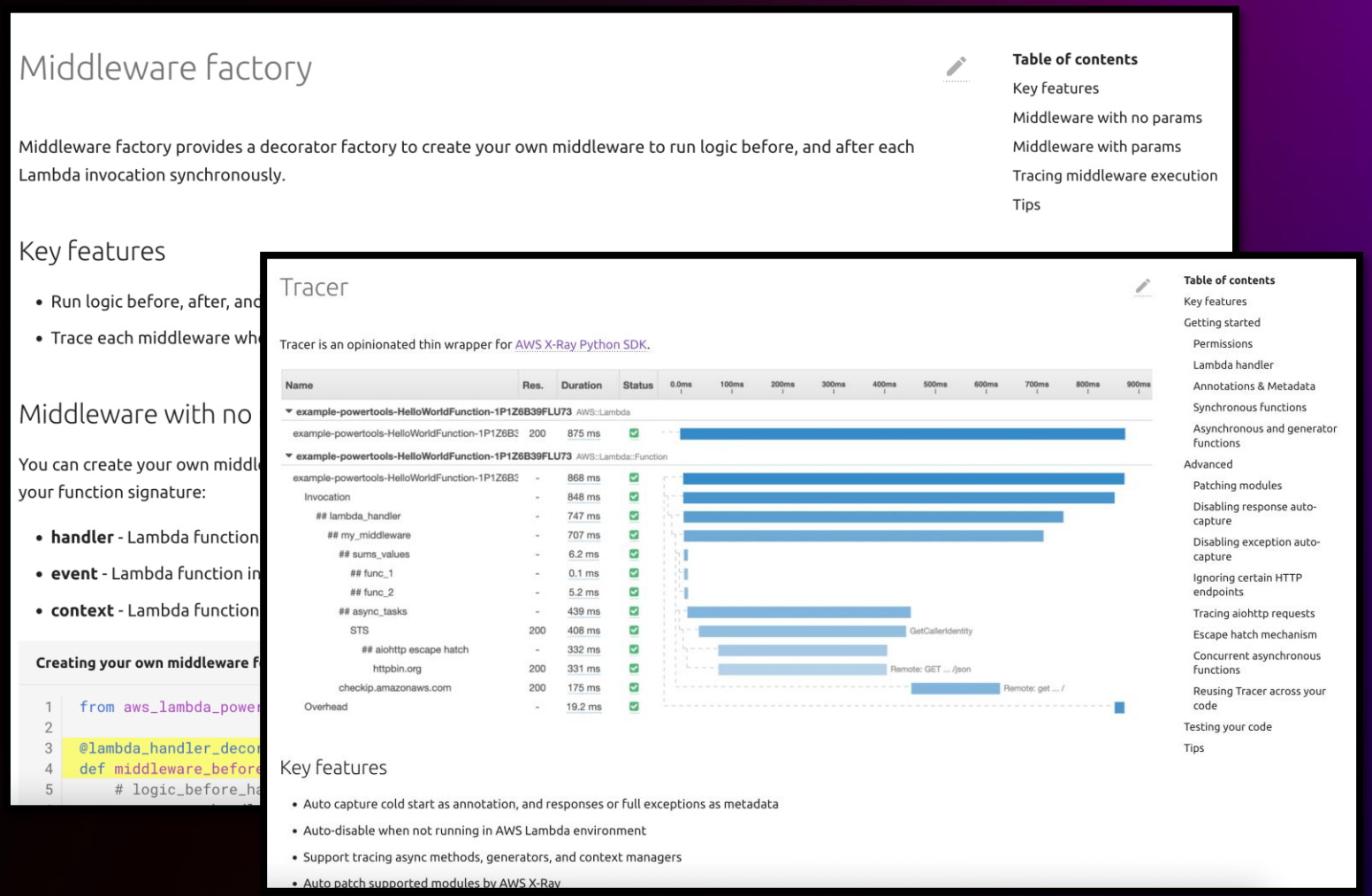


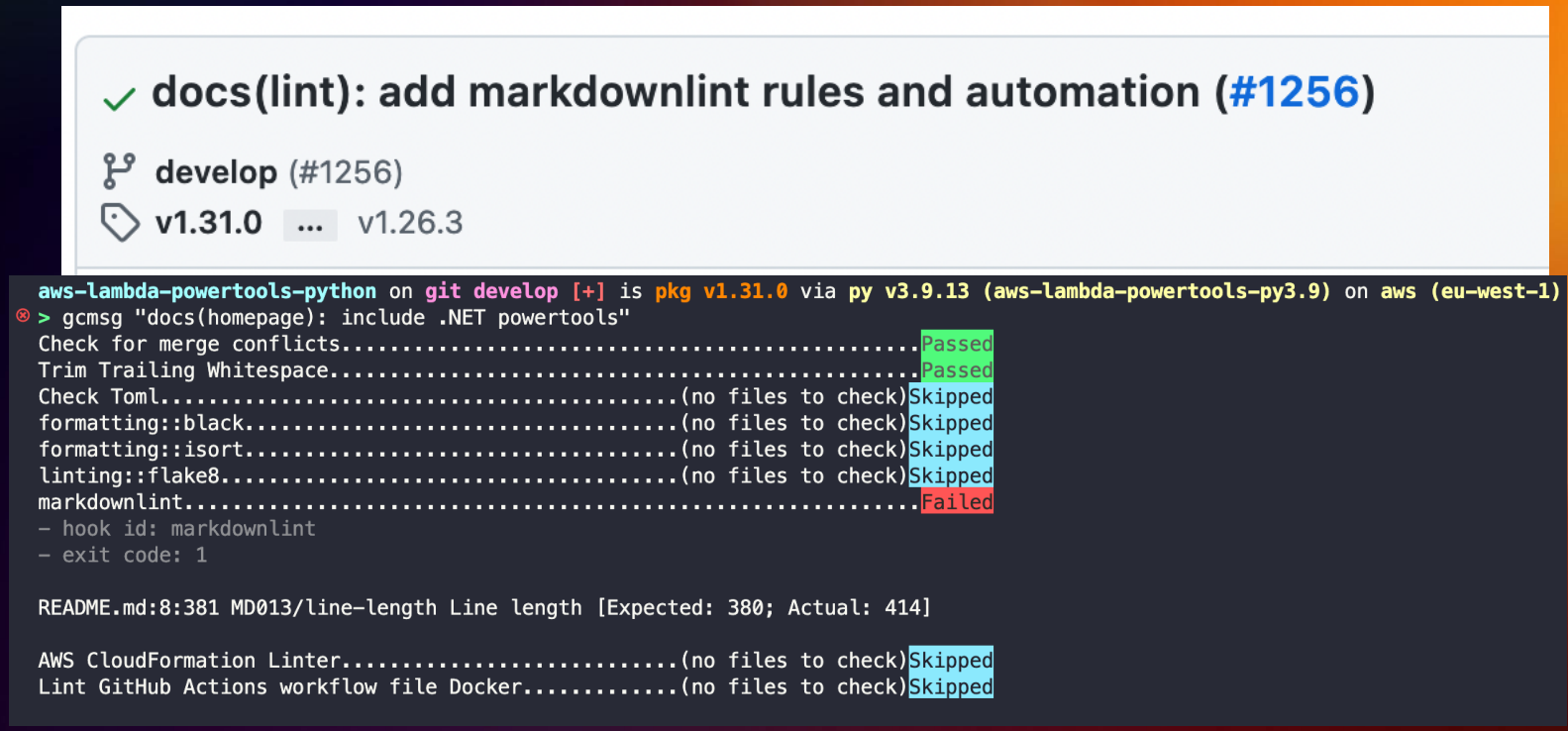
Table of contents inconsistency

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Consistency

Lesson



Gradual and sustainable consistency

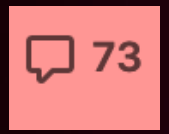
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Challenge

“Why don’t you just”

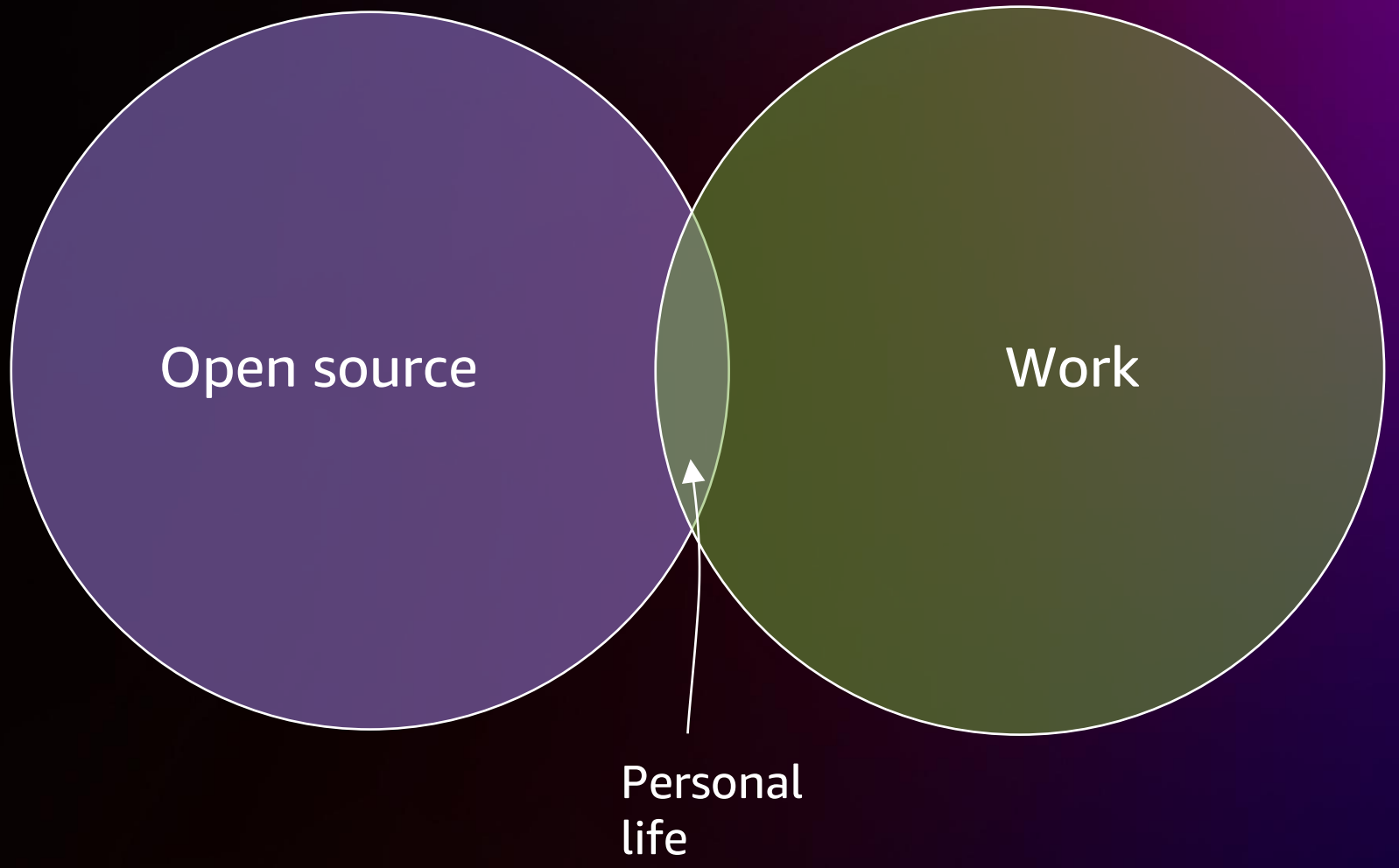


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Challenge

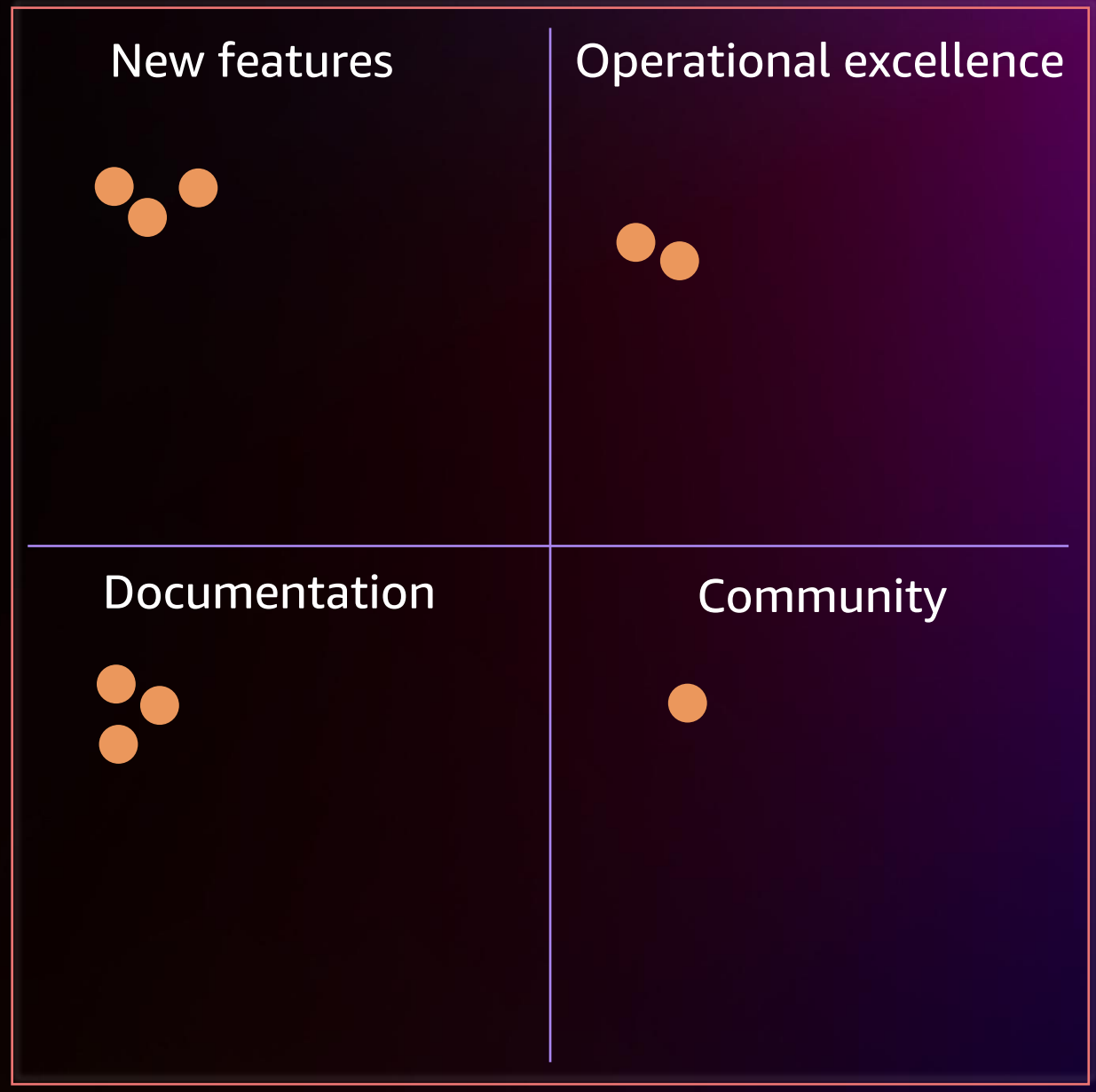


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Challenge

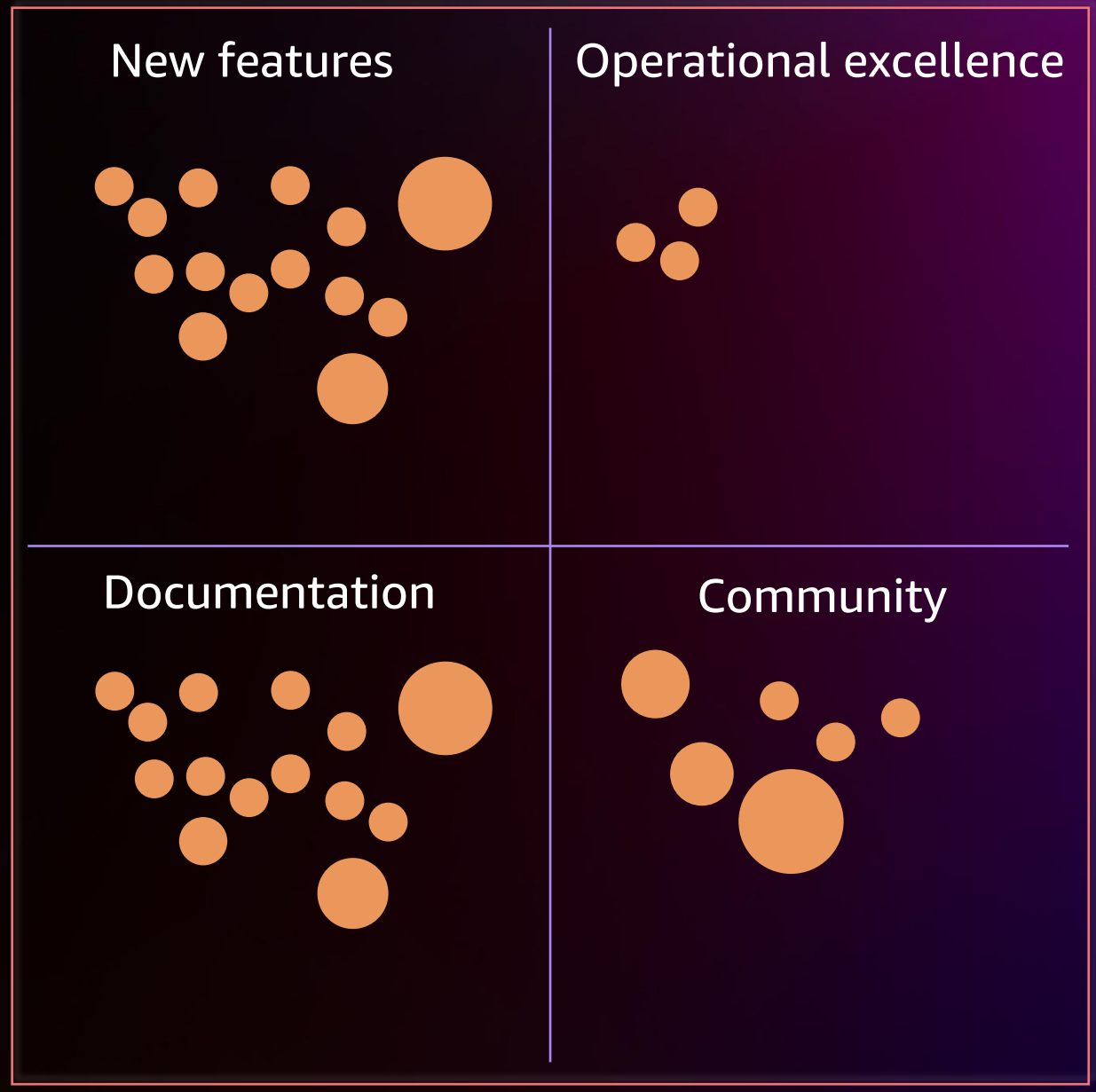


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Challenge



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Challenge



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Lesson



Working in Public

The Making and Maintenance of Open Source Software

by Nadia Eghbal

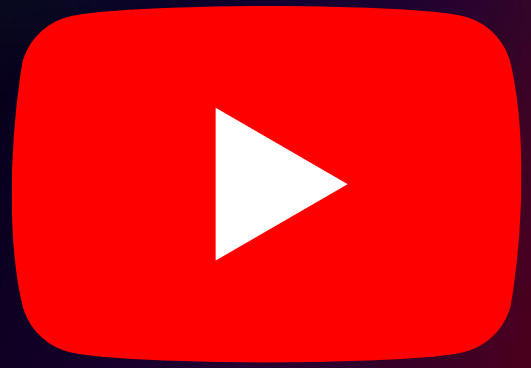
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Lesson



The Hard Parts of Open Source

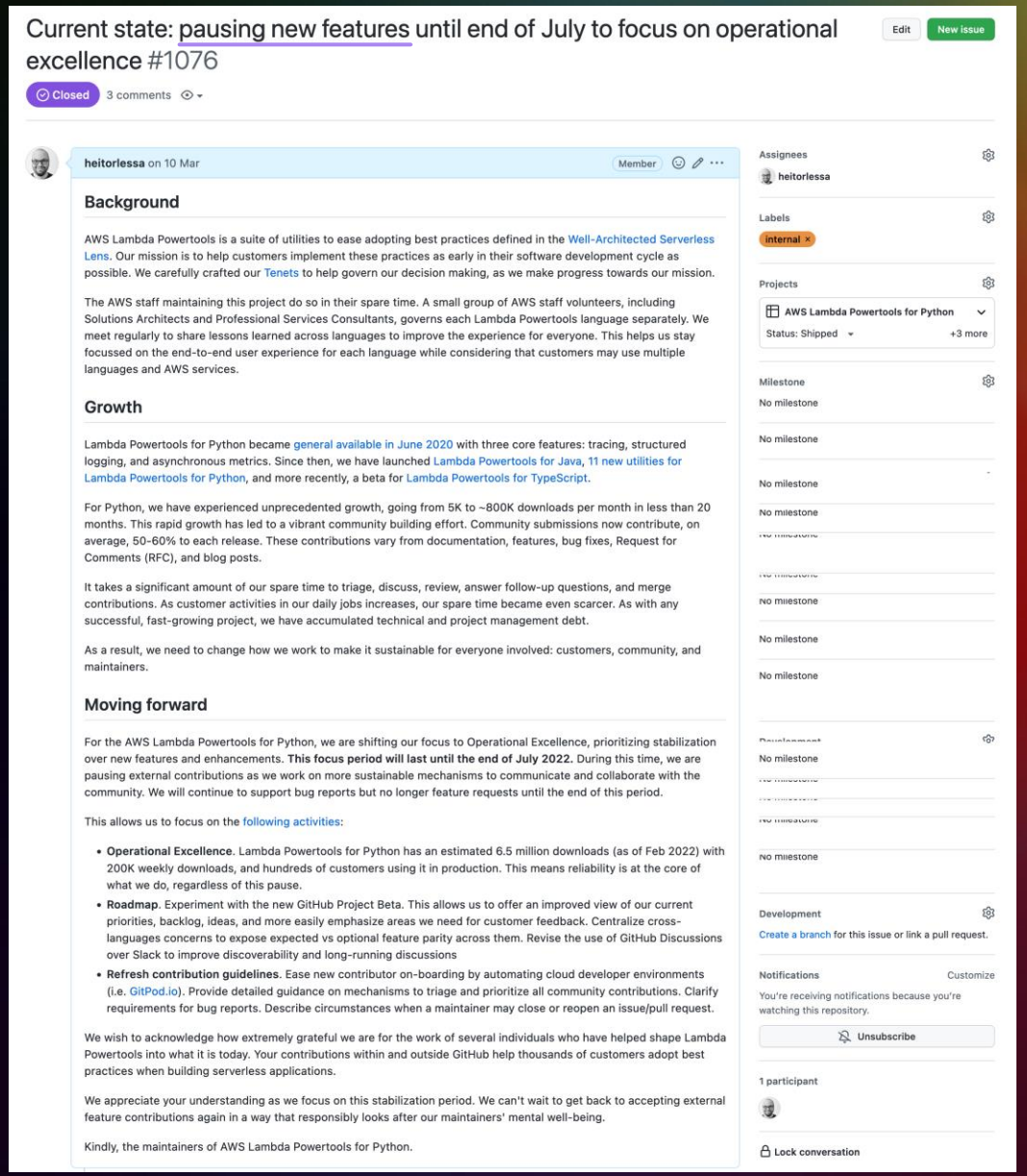
“Everything is done in haste, at fever pitch.”

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Lesson



Current state: pausing new features until end of July to focus on operational excellence #1076

Edit | New issue

Closed | 3 comments

heitorlessa on 10 Mar — Member

Background

AWS Lambda Powertools is a suite of utilities to ease adopting best practices defined in the Well-Architected Serverless Lens. Our mission is to help customers implement these practices as early in their software development cycle as possible. We carefully crafted our Tenets to help govern our decision making, as we make progress towards our mission.

The AWS staff maintaining this project do so in their spare time. A small group of AWS staff volunteers, including Solutions Architects and Professional Services Consultants, governs each Lambda Powertools language separately. We meet regularly to share lessons learned across languages to improve the experience for everyone. This helps us stay focussed on the end-to-end user experience for each language while considering that customers may use multiple languages and AWS services.

Growth

Lambda Powertools for Python became general available in June 2020 with three core features: tracing, structured logging, and asynchronous metrics. Since then, we have launched Lambda Powertools for Java, 11 new utilities for Lambda Powertools for Python, and more recently, a beta for Lambda Powertools for TypeScript.

For Python, we have experienced unprecedented growth, going from 5K to ~800K downloads per month in less than 20 months. This rapid growth has led to a vibrant community building effort. Community submissions now contribute, on average, 50-60% to each release. These contributions vary from documentation, features, bug fixes, Request for Comments (RFC), and blog posts.

It takes a significant amount of our spare time to triage, discuss, review, answer follow-up questions, and merge contributions. As customer activities in our daily jobs increases, our spare time became even scarcer. As with any successful, fast-growing project, we have accumulated technical and project management debt.

As a result, we need to change how we work to make it sustainable for everyone involved: customers, community, and maintainers.

Moving forward

For the AWS Lambda Powertools for Python, we are shifting our focus to Operational Excellence, prioritizing stabilization over new features and enhancements. **This focus period will last until the end of July 2022.** During this time, we are pausing external contributions as we work on more sustainable mechanisms to communicate and collaborate with the community. We will continue to support bug reports but no longer feature requests until the end of this period.

This allows us to focus on the following activities:

- **Operational Excellence**. Lambda Powertools for Python has an estimated 6.5 million downloads (as of Feb 2022) with 200K weekly downloads, and hundreds of customers using it in production. This means reliability is at the core of what we do, regardless of this pause.
- **Roadmap**. Experiment with the new GitHub Project Beta. This allows us to offer an improved view of our current priorities, backlog, ideas, and more easily emphasize areas we need for customer feedback. Centralize cross-languages concerns to expose expected vs optional feature parity across them. Revise the use of GitHub Discussions over Slack to improve discoverability and long-running discussions
- **Refresh contribution guidelines**. Ease new contributor on-boarding by automating cloud developer environments (i.e. GitPod.io). Provide detailed guidance on mechanisms to triage and prioritize all community contributions. Clarify requirements for bug reports. Describe circumstances when a maintainer may close or reopen an issue/pull request.

We wish to acknowledge how extremely grateful we are for the work of several individuals who have helped shape Lambda Powertools into what it is today. Your contributions within and outside GitHub help thousands of customers adopt best practices when building serverless applications.

We appreciate your understanding as we focus on this stabilization period. We can't wait to get back to accepting external feature contributions again in a way that responsibly looks after our maintainers' mental well-being.

Kindly, the maintainers of AWS Lambda Powertools for Python.

Assignees: heitorlessa

Labels: internal

Projects: AWS Lambda Powertools for Python — Status: Shipped +3 more

Milestone: No milestone

Development: Create a branch for this issue or link a pull request.

Notifications: Customize — You're receiving notifications because you're watching this repository. — Unsubscribe

1 participant

Lock conversation

Ops excellence

Roadmap

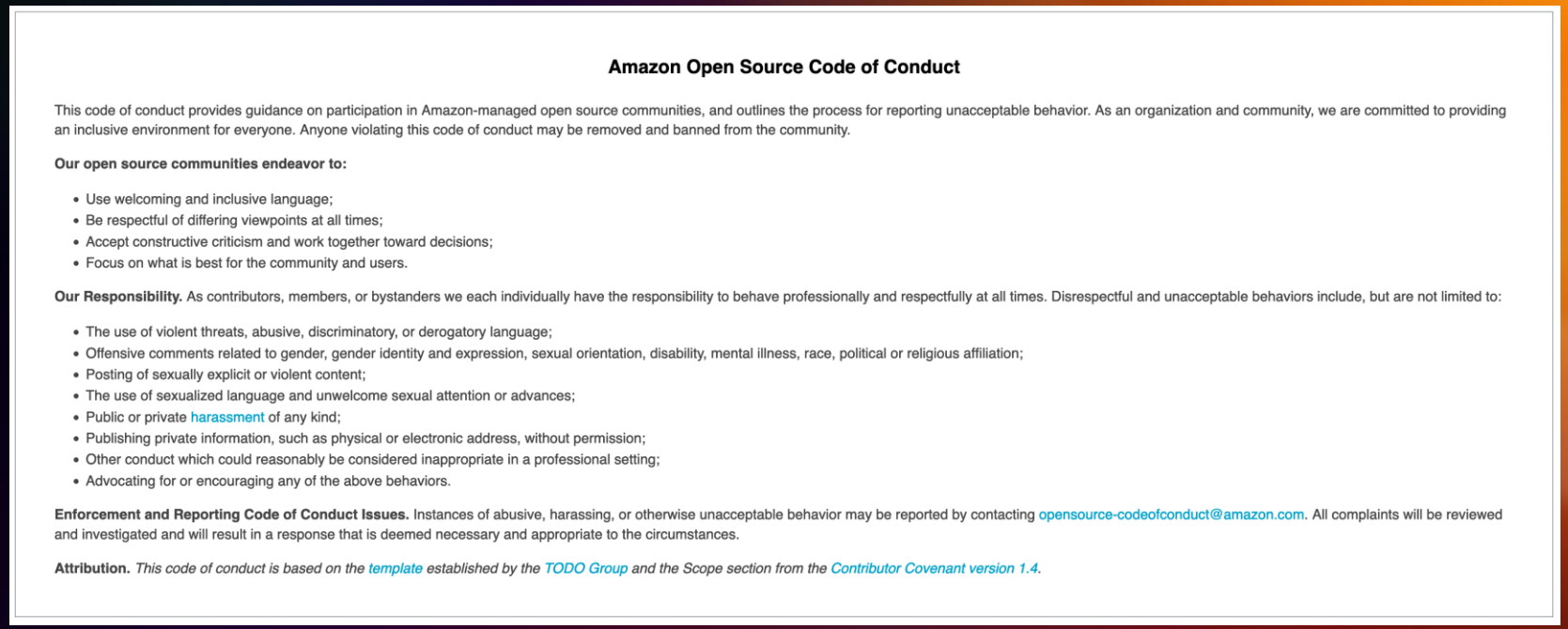
Contribution guidelines

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Lesson



Amazon Open Source Code of Conduct

This code of conduct provides guidance on participation in Amazon-managed open source communities, and outlines the process for reporting unacceptable behavior. As an organization and community, we are committed to providing an inclusive environment for everyone. Anyone violating this code of conduct may be removed and banned from the community.

Our open source communities endeavor to:

- Use welcoming and inclusive language;
- Be respectful of differing viewpoints at all times;
- Accept constructive criticism and work together toward decisions;
- Focus on what is best for the community and users.

Our Responsibility. As contributors, members, or bystanders we each individually have the responsibility to behave professionally and respectfully at all times. Disrespectful and unacceptable behaviors include, but are not limited to:

- The use of violent threats, abusive, discriminatory, or derogatory language;
- Offensive comments related to gender, gender identity and expression, sexual orientation, disability, mental illness, race, political or religious affiliation;
- Posting of sexually explicit or violent content;
- The use of sexualized language and unwelcome sexual attention or advances;
- Public or private harassment of any kind;
- Publishing private information, such as physical or electronic address, without permission;
- Other conduct which could reasonably be considered inappropriate in a professional setting;
- Advocating for or encouraging any of the above behaviors.

Enforcement and Reporting Code of Conduct Issues. Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting opensource-codeofconduct@amazon.com. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances.

Attribution. *This code of conduct is based on the template established by the TODO Group and the Scope section from the Contributor Covenant version 1.4.*

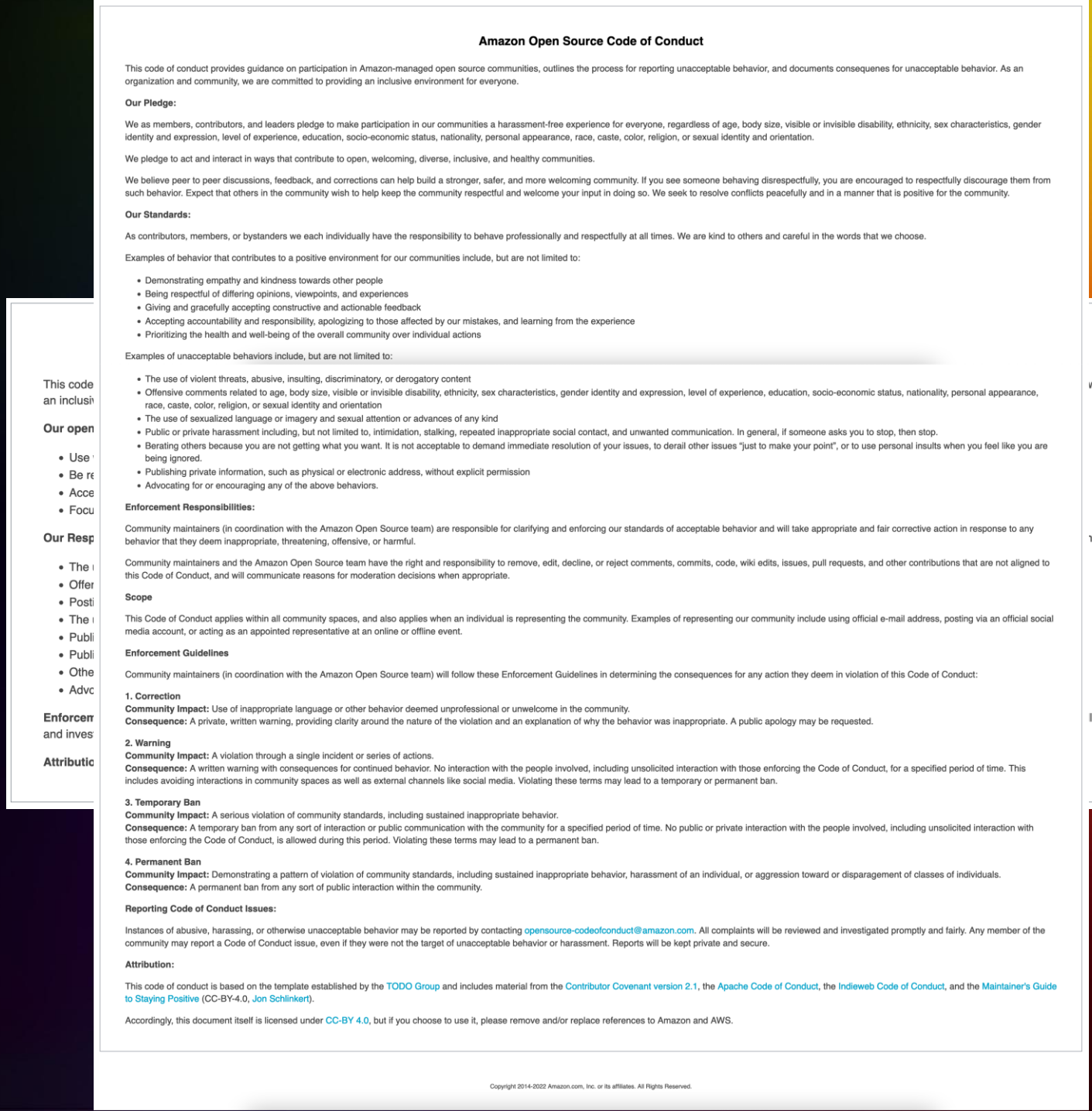
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Balancing act

Lesson



Amazon Open Source Code of Conduct

This code of conduct provides guidance on participation in Amazon-managed open source communities, outlines the process for reporting unacceptable behavior, and documents consequenes for unacceptable behavior. As an organization and community, we are committed to providing an inclusive environment for everyone.

Our Pledge:

We as members, contributors, and leaders pledge to make participation in our communities a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, caste, color, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to open, welcoming, diverse, inclusive, and healthy communities.

We believe peer to peer discussions, feedback, and corrections can help build a stronger, safer, and more welcoming community. If you see someone behaving disrespectfully, you are encouraged to respectfully discourage them from such behavior. Expect that others in the community wish to help keep the community respectful and welcome your input in doing so. We seek to resolve conflicts peacefully and in a manner that is positive for the community.

Our Standards:

As contributors, members, or bystanders we each individually have the responsibility to behave professionally and respectfully at all times. We are kind to others and careful in the words that we choose.

Examples of behavior that contributes to a positive environment for our communities include, but are not limited to:

- Demonstrating empathy and kindness towards other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive and actionable feedback
- Accepting accountability and responsibility, apologizing to those affected by our mistakes, and learning from the experience
- Prioritizing the health and well-being of the overall community over individual actions

Examples of unacceptable behaviors include, but are not limited to:

- The use of violent threats, abusive, insulting, discriminatory, or derogatory content
- Offensive comments related to age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, caste, color, religion, or sexual identity and orientation
- The use of sexualized language or imagery and sexual attention or advances of any kind
- Public or private harassment including, but not limited to, intimidation, stalking, repeated inappropriate social contact, and unwanted communication. In general, if someone asks you to stop, then stop.
- Berating others because you are not getting what you want. It is not acceptable to demand immediate resolution of your issues, to derail other issues "just to make your point", or to use personal insults when you feel like you are being ignored.
- Publishing private information, such as physical or electronic address, without explicit permission
- Advocating for or encouraging any of the above behaviors.

Enforcement Responsibilities:

Community maintainers (in coordination with the Amazon Open Source team) are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community maintainers and the Amazon Open Source team have the right and responsibility to remove, edit, decline, or reject comments, commits, code, wiki edits, issues, pull requests, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is representing the community. Examples of representing our community include using official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

Enforcement Guidelines

Community maintainers (in coordination with the Amazon Open Source team) will follow these Enforcement Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction
Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.
Consequence: A private, written warning, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning
Community Impact: A violation through a single incident or series of actions.
Consequence: A written warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban
Community Impact: A serious violation of community standards, including sustained inappropriate behavior.
Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban
Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.
Consequence: A permanent ban from any sort of public interaction within the community.

Reporting Code of Conduct Issues:

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting opensource-codeofconduct@amazon.com. All complaints will be reviewed and investigated promptly and fairly. Any member of the community may report a Code of Conduct issue, even if they were not the target of unacceptable behavior or harassment. Reports will be kept private and secure.

Attribution:

This code of conduct is based on the template established by the TODO Group and includes material from the Contributor Covenant version 2.1, the Apache Code of Conduct, the Indieweb Code of Conduct, and the Maintainer's Guide to Staying Positive (CC-BY-4.0, Jon Schlinkert).

Accordingly, this document itself is licensed under CC-BY 4.0, but if you choose to use it, please remove and/or replace references to Amazon and AWS.

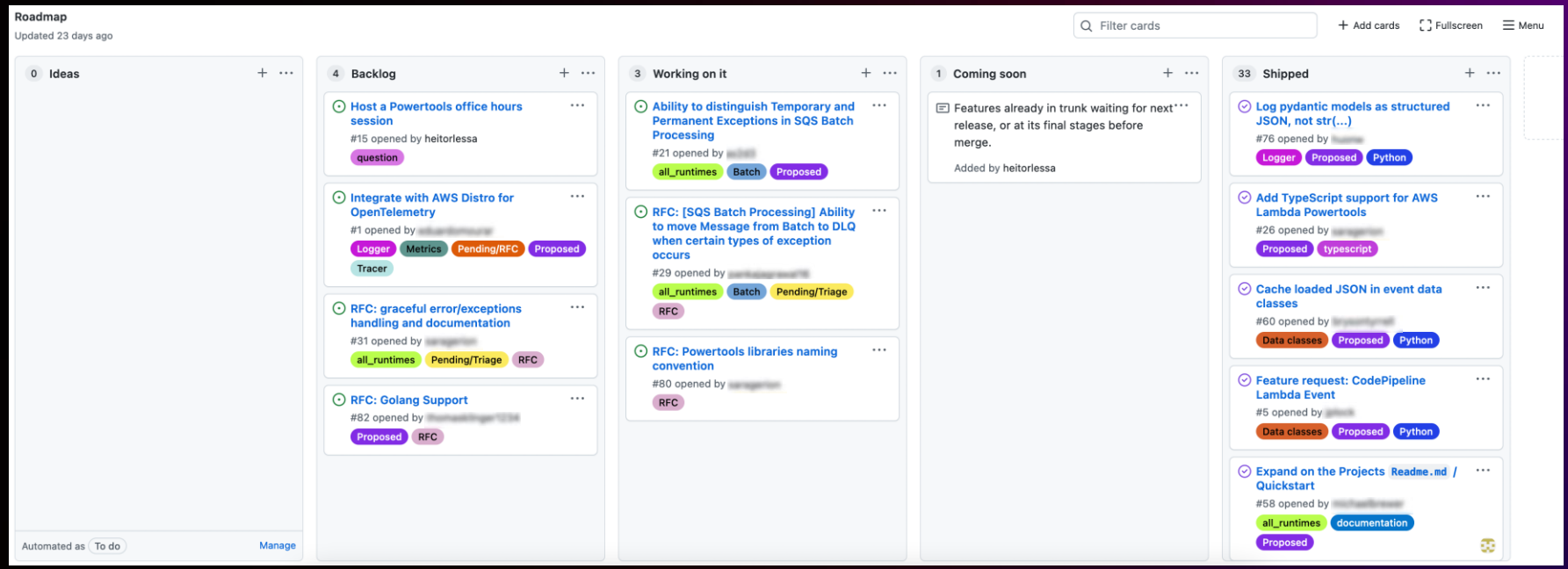
Copyright 2014-2022 Amazon.com, Inc. or its affiliates. All Rights Reserved.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Transparency

Challenge

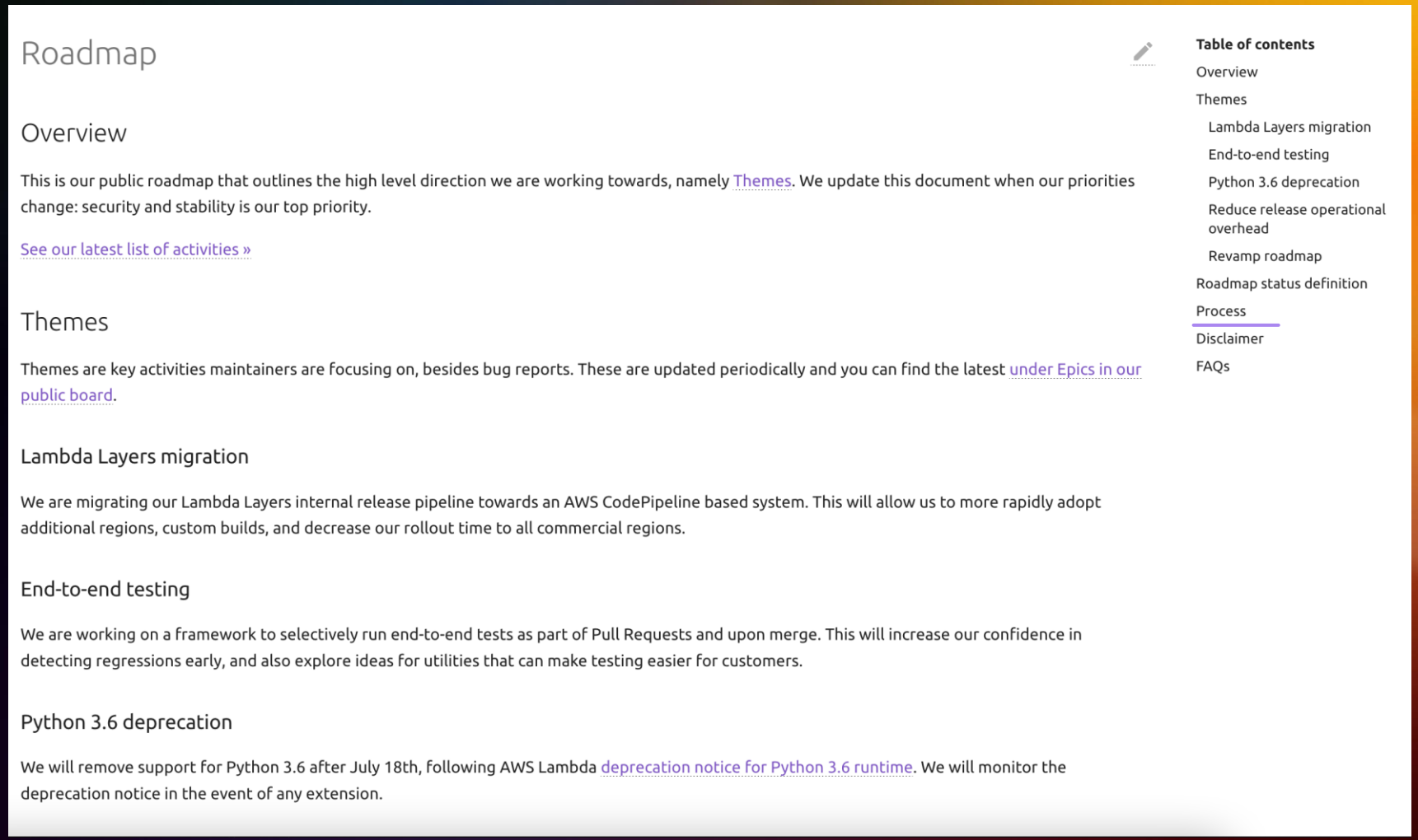


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Transparency

Lesson



Roadmap

Table of contents
- Overview
- Themes
  - Lambda Layers migration
  - End-to-end testing
  - Python 3.6 deprecation
  - Reduce release operational overhead
  - Revamp roadmap
- Roadmap status definition
- Process
- Disclaimer
- FAQs

Overview

This is our public roadmap that outlines the high level direction we are working towards, namely Themes. We update this document when our priorities change: security and stability is our top priority.

See our latest list of activities »

Themes

Themes are key activities maintainers are focusing on, besides bug reports. These are updated periodically and you can find the latest under Epics in our public board.

Lambda Layers migration

We are migrating our Lambda Layers internal release pipeline towards an AWS CodePipeline based system. This will allow us to more rapidly adopt additional regions, custom builds, and decrease our rollout time to all commercial regions.

End-to-end testing

We are working on a framework to selectively run end-to-end tests as part of Pull Requests and upon merge. This will increase our confidence in detecting regressions early, and also explore ideas for utilities that can make testing easier for customers.

Python 3.6 deprecation

We will remove support for Python 3.6 after July 18th, following AWS Lambda deprecation notice for Python 3.6 runtime. We will monitor the deprecation notice in the event of any extension.

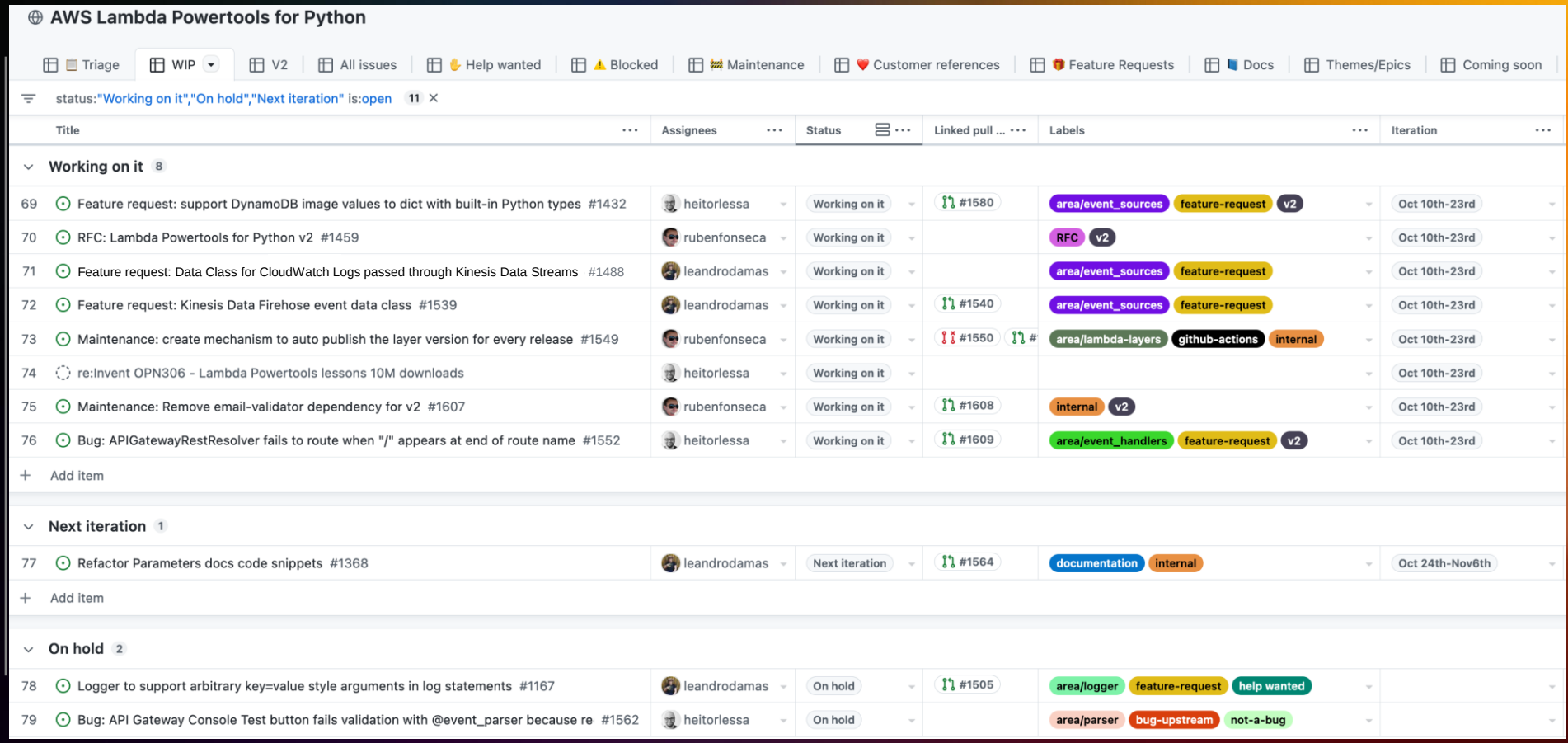
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Transparency

Lesson



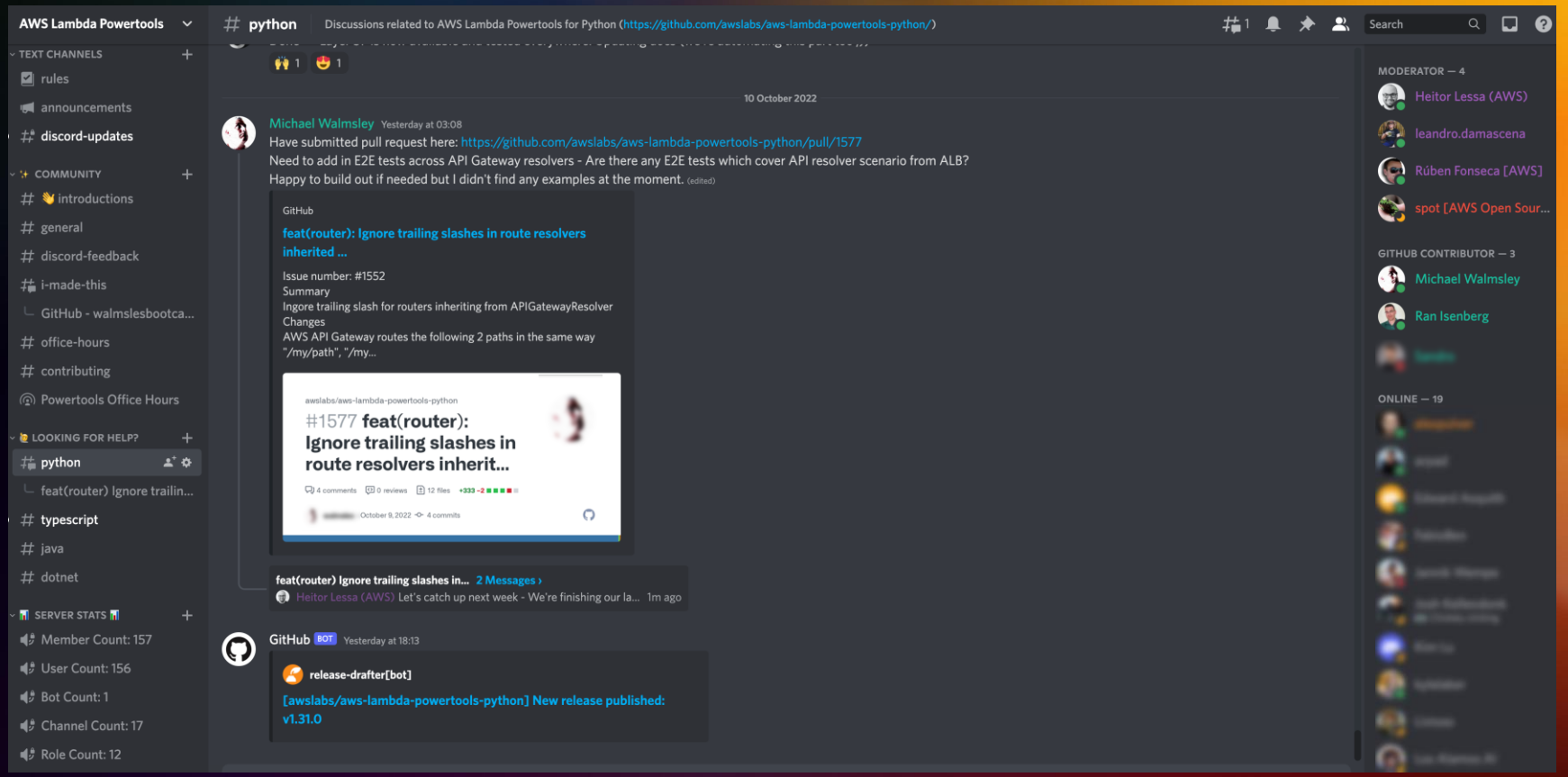
Roadmap and daily operations revamped

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Transparency

Lesson



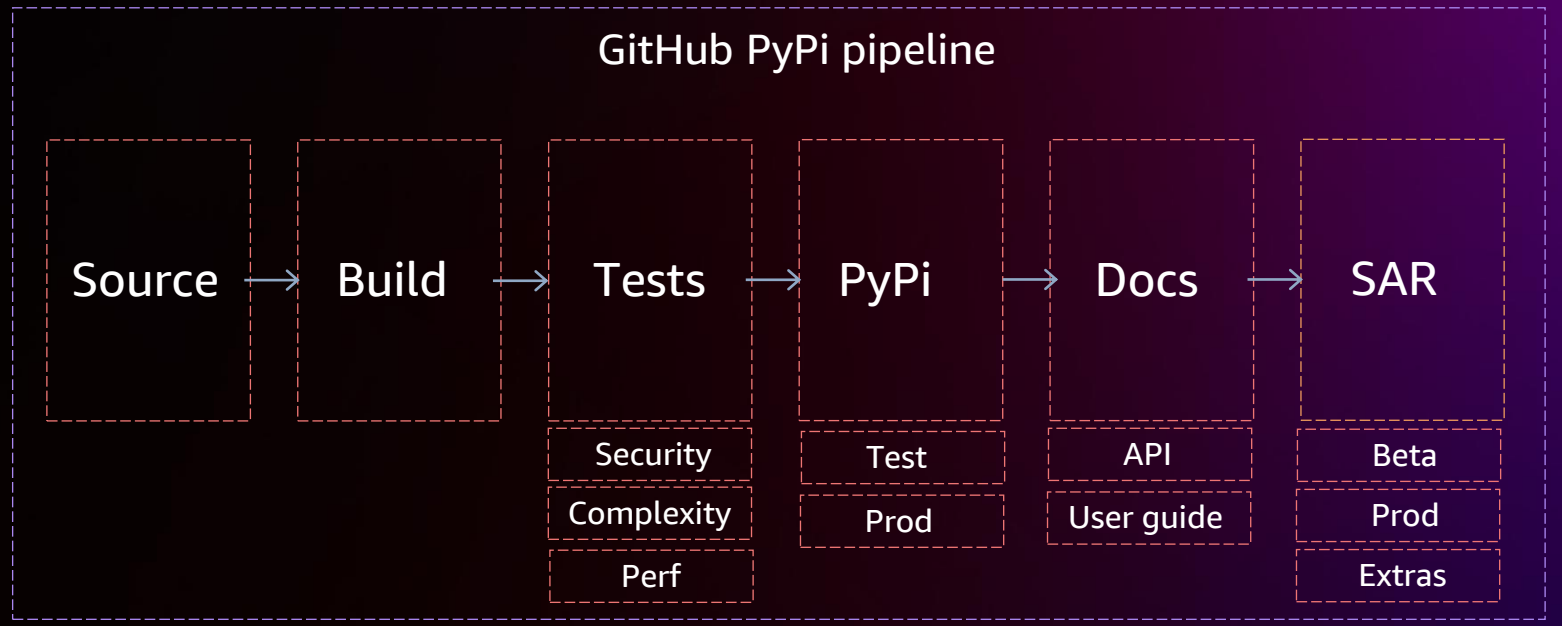
Migration to Discord led to a healthier community

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release overhead

Challenge

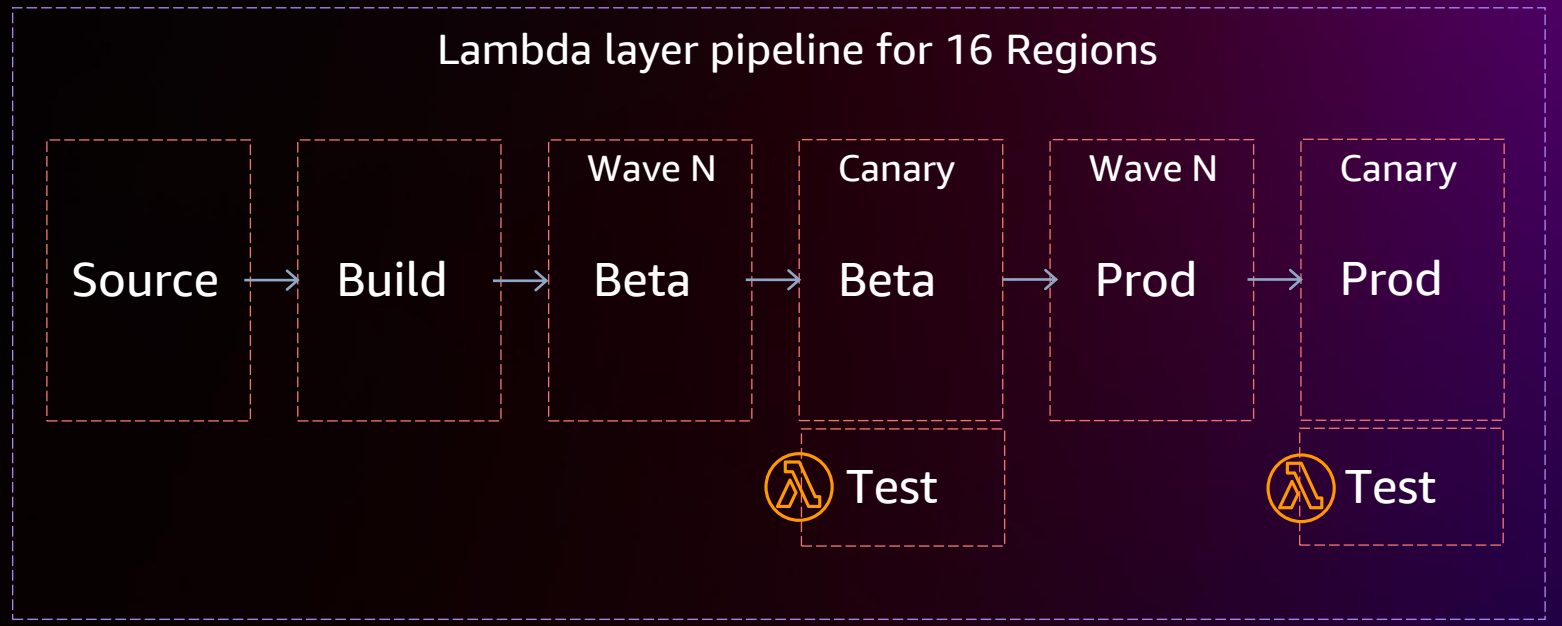


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release overhead

Challenge



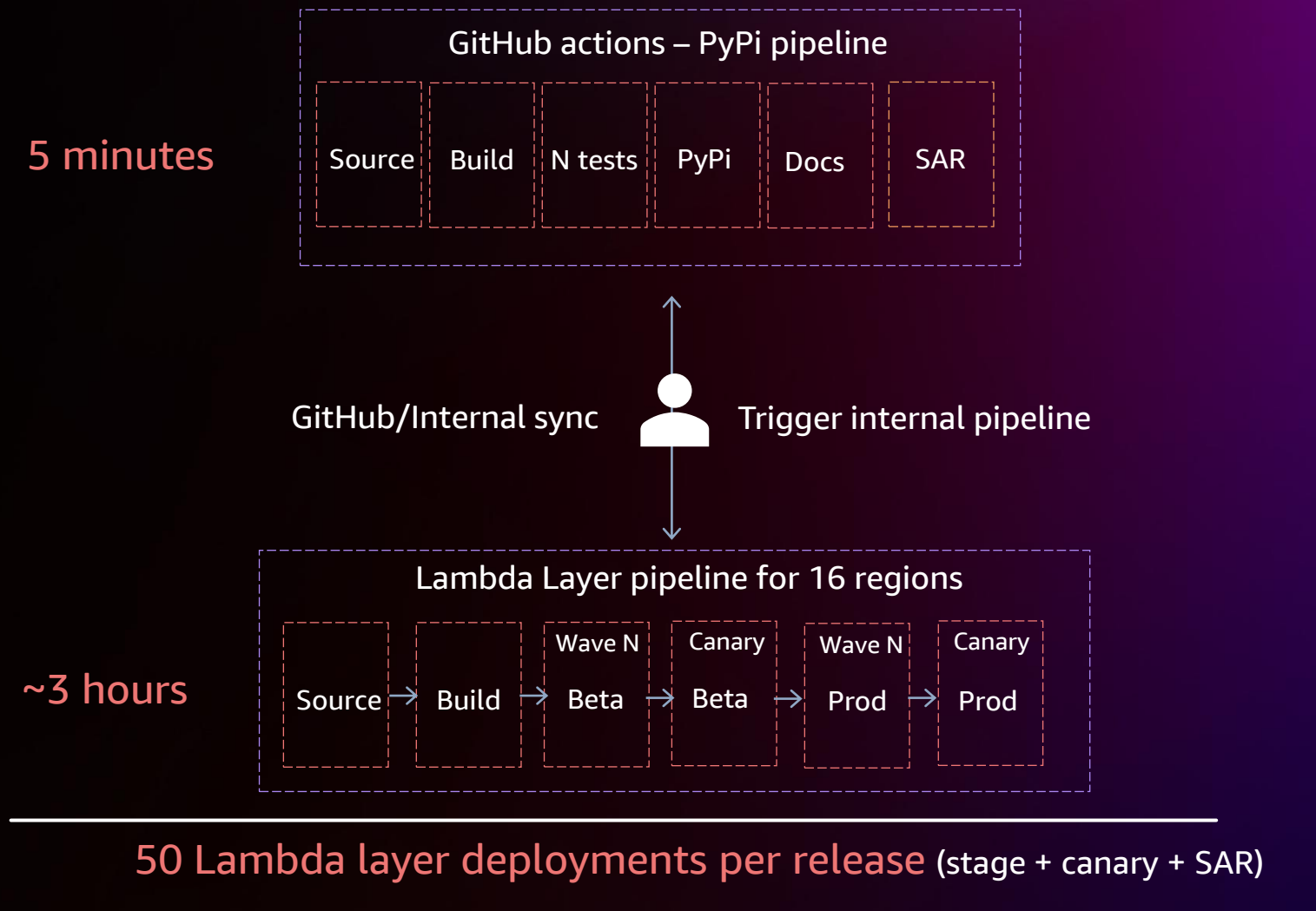
Internal pipeline was disconnected from GitHub processes

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release overhead

Challenge



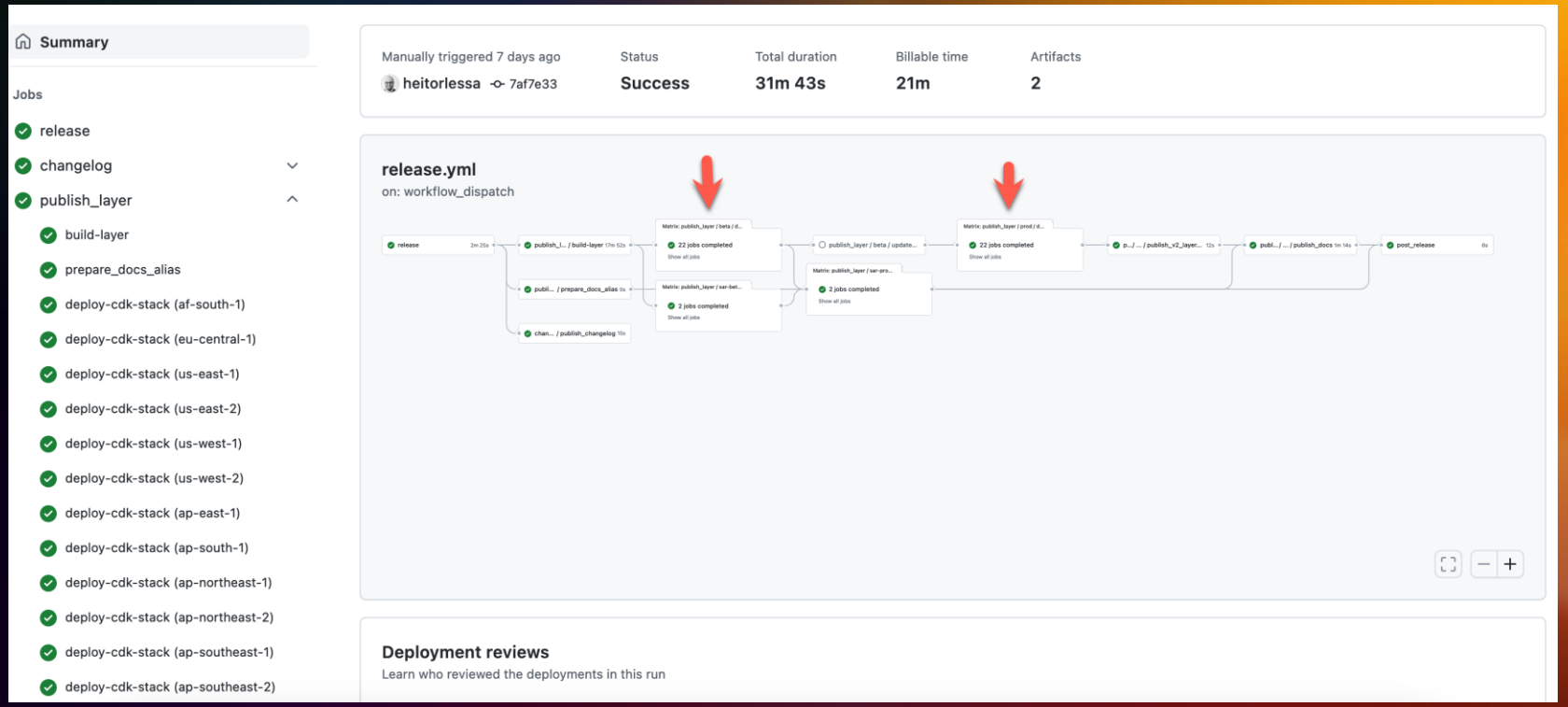
50 Lambda layer deployments per release (stage + canary + SAR)

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Release overhead

Lesson



178 Lambda Layer deployments per release (stage + canary + SAR)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

End-to-end test

Challenge

- Speed?
- Frequency?
- Run a single test/infra?
- Step-through debugging?
- Cost?
- Parallelization?
- Isolated test runs?
- Entry bar for infrastructure as code?

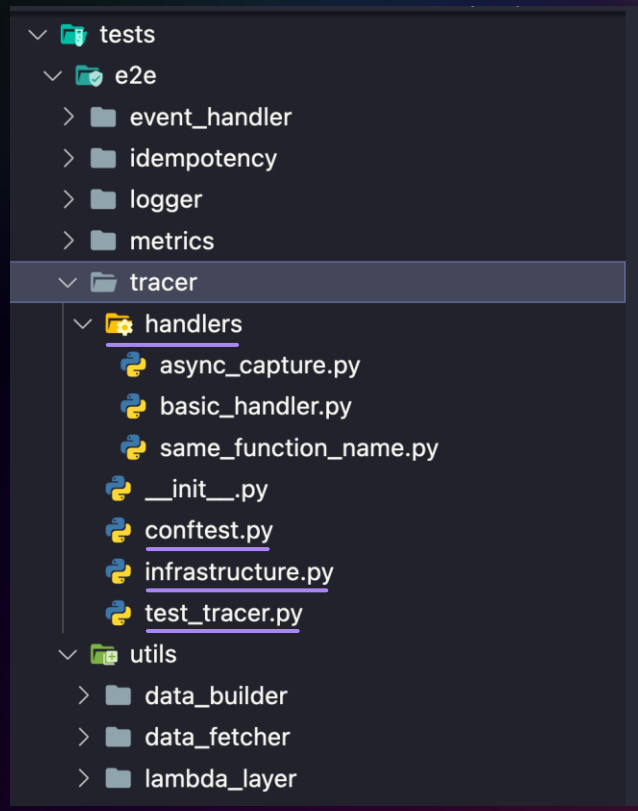
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

End-to-end test

Lesson




```python
import pytest

from tests.e2e.tracer.infrastructure import TracerStack


@pytest.fixture(autouse=True, scope="module")
def infrastructure():
    """Setup and teardown logic for E2E test infrastructure


    Yields
    ------
    Dict[str, str]
        CloudFormation Outputs from deployed infrastructure
    """
    stack = TracerStack()
    try:
        yield stack.deploy()
    finally:
        stack.delete()
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

End-to-end test

Lesson

```python
# tests/e2e/tracer/infrastructure.py
from tests.e2e.utils.infrastructure import BaseInfrastructure


class TracerStack(BaseInfrastructure):
    def create_resources(self) -> None:
        self.create_lambda_functions()

```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

End-to-end test

Lesson

```python
# tests/e2e/idempotency/infrastructure.py
from aws_cdk import CfnOutput, RemovalPolicy
from aws_cdk import aws_dynamodb as dynamodb
from aws_cdk.aws_dynamodb import Table

from tests.e2e.utils.infrastructure import BaseInfrastructure


class IdempotencyDynamoDBStack(BaseInfrastructure):
    def create_resources(self):
        table = self._create_dynamodb_table()

        env_vars = {"IdempotencyTable": table.table_name}
        functions = self.create_lambda_functions(function_props={"environment": env_vars})

        table.grant_read_write_data(functions["TtlCacheExpirationHandler"])
        table.grant_read_write_data(functions["TtlCacheTimeoutHandler"])
        table.grant_read_write_data(functions["ParallelExecutionHandler"])

    def _create_dynamodb_table(self) -> Table:
        table = dynamodb.Table(
            self.stack,
            "Idempotency",
            removal_policy=RemovalPolicy.DESTROY,
            partition_key=dynamodb.Attribute(name="id", type=dynamodb.AttributeType.STRING),
            time_to_live_attribute="expiration",
            billing_mode=dynamodb.BillingModePAY_PER_REQUEST,
        )

        CfnOutput(self.stack, "DynamoDBTable", value=table.table_name)

        return table
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

End-to-end test

Lesson

```python
def test_lambda_handler_trace_is_visible(basic_handler_fn_arn: str, basic_handler_fn: str):
    # GIVEN
    service = data_builder.build_service_name()
    handler_name = basic_handler.lambda_handler.__name__
    handler_subsegment = f"## {handler_name}"
    handler_metadata_key = f"{handler_name} response"

    method_name = f"basic_handler.{basic_handler.get_todos.__name__}"
    method_subsegment = f"## {method_name}"
    method_metadata_key = f"{method_name} response"

    trace_query = data_builder.build_trace_default_query(function_name=basic_handler_fn)

    # WHEN
    event = json.dumps({"service": service})
    _, execution_time = data_fetcher.get_lambda_response(lambda_arn=basic_handler_fn_arn, payload=event)
    data_fetcher.get_lambda_response(lambda_arn=basic_handler_fn_arn, payload=event)

    # THEN
    trace = data_fetcher.get_traces(start_date=execution_time, filter_expression=trace_query, minimum_traces=2)

    assert len(trace.get_annotation(key="ColdStart", value=True)) == 1
    assert len(trace.get_metadata(key=handler_metadata_key, namespace=service)) == 2
    assert len(trace.get_metadata(key=method_metadata_key, namespace=service)) == 2
    assert len(trace.get_subsegment(name=handler_subsegment)) == 2
    assert len(trace.get_subsegment(name=method_subsegment)) == 2

```

Data fetchers are not customer-facing (yet!)

aws

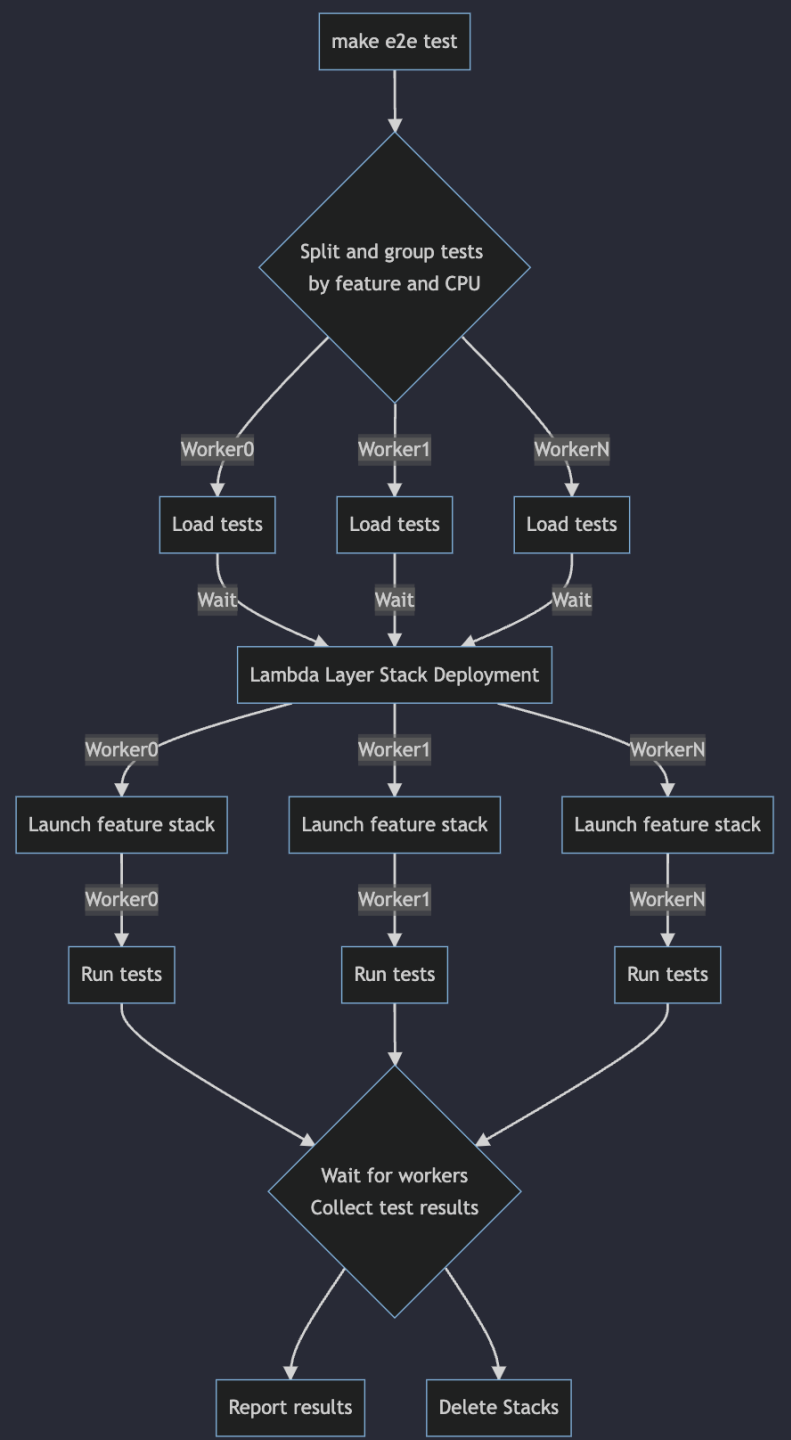
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

End-to-end test

Lesson

Parallelization workflow



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Impact

Challenge

Stars?

Downloads?

Forks?

# of case studies?

# of language bindings?

How can we track unobtrusively?

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Impact	Lesson
Lambda layer	Unobtrusive tracking
Community	~40% of Powertools features
Belonging	Customer and Partner’s feedback

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons

Impact

Lesson



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up next: What the near future holds

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up next

Feature parity

Python



Java



TypeScript



.NET



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up next

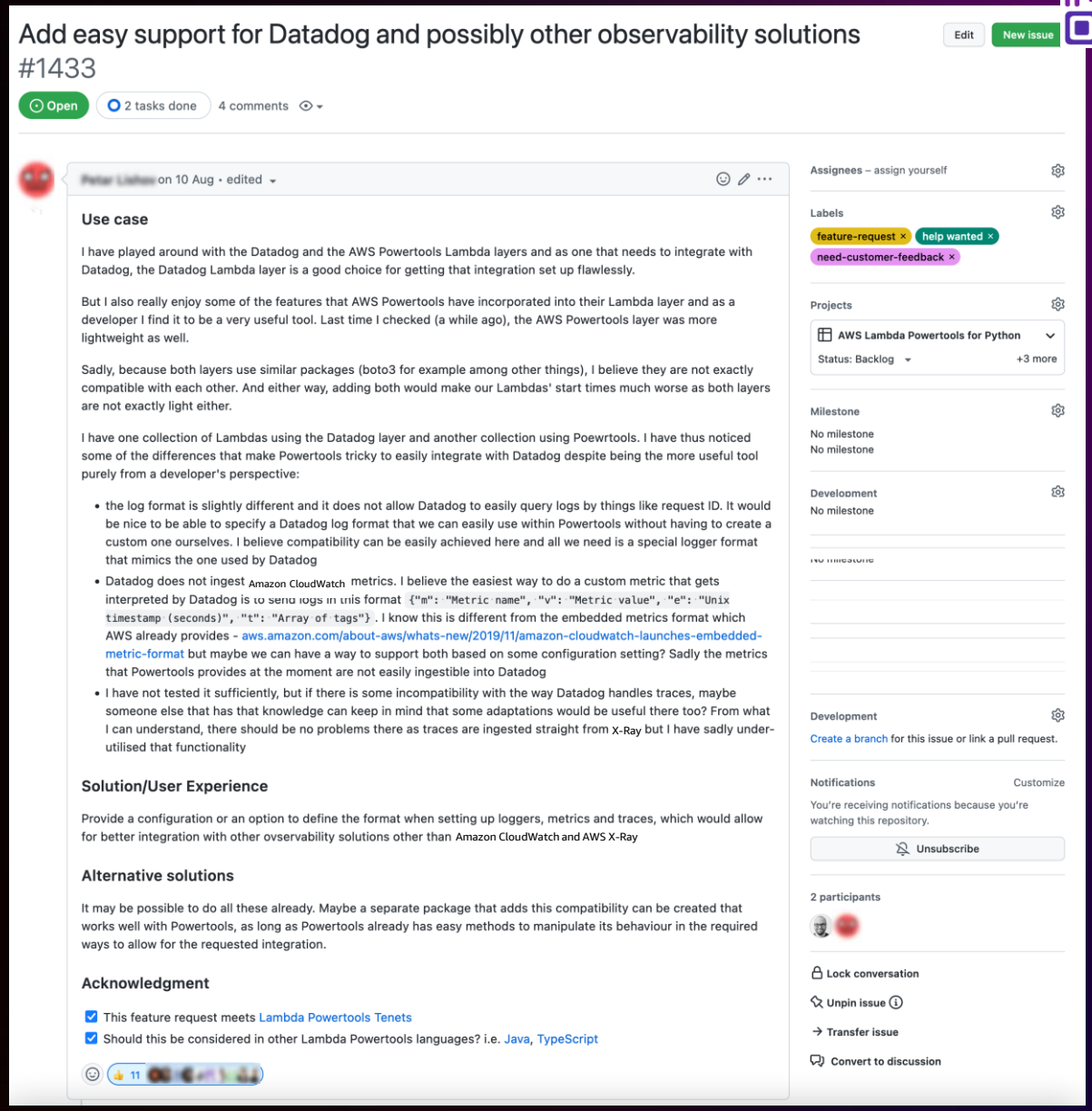
Feature parity

	Python	Java	TypeScript	.NET
Tracer	x	x	x	x
Logger	x	x	x	x
Metrics	x	x	x	x
Event handler (REST)	x			
Event handler (GraphQL)	x			
Idempotency	x	x	In-progress	In-progress
Parameters	x	x	In-progress	
Batch processing	x	x		
Validation	x	x		
Event source data classes	x	x		x
Parser	x			
Feature flags	x			
Streaming	x			
JMESPath (data extraction)	x	x		
Amazon SQS large message handling		x		

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up next

Observability providers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up next

Event source JSON schema



activeMQEvent.json	feat(data-classes): ActiveMQ and RabbitMQ support (#770)	12 months ago
albEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
apiGatewayAuthorizerRequestEvent.json	feat(data-classes): authorizer for http api and rest api (#620)	14 months ago
apiGatewayAuthorizerTokenEvent.json	feat(data-classes): authorizer for http api and rest api (#620)	14 months ago
apiGatewayAuthorizerV2Event.json	feat(data-classes): authorizer for http api and rest api (#620)	14 months ago
apiGatewayProxyEvent.json	feat(parser): Support for API GW v1 proxy schema & envelope (#403)	2 years ago
apiGatewayProxyEventPrincipalId.json	fix(event-sources): Pass authorizer data to APIGatewayEventAuthorizer (...	10 months ago
apiGatewayProxyEvent_noVersionAuth.json	fix(parser): Make ApiGateway version, authorizer fields optional (#532)	15 months ago
apiGatewayProxyV2Event.json	feat(data-classes): authorizer for http api and rest api (#620)	14 months ago
apiGatewayProxyV2IamEvent.json	feat(parser): add support for API Gateway HTTP API #434 (#441)	17 months ago
apiGatewayProxyV2LambdaAuthorizerEvent.json	feat(parser): add support for API Gateway HTTP API #434 (#441)	17 months ago
appSyncAuthorizerEvent.json	feat(data-classes): AppSync Lambda authorizer event (#610)	14 months ago
appSyncAuthorizerResponse.json	feat(data-classes): AppSync Lambda authorizer event (#610)	14 months ago
appSyncDirectResolver.json	feat(data-classes): AppSync Resolver Event (#323)	2 years ago
appSyncResolverEvent.json	feat(data-classes): AppSync Resolver Event (#323)	2 years ago
cloudWatchDashboardEvent.json	feat(event_sources): add CloudWatch dashboard custom widget event (#1474	28 days ago
cloudWatchLogEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
codePipelineEvent.json	feat(data-classes): support for code pipeline job event (#416)	17 months ago
codePipelineEventData.json	feat(data-classes): support for code pipeline job event (#416)	17 months ago
cognitoCreateAuthChallengeEvent.json	feat(data_classes): Cognito custom auth triggers	2 years ago
cognitoCustomMessageEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
cognitoDefineAuthChallengeEvent.json	feat(data_classes): Cognito custom auth triggers	2 years ago
cognitoPostAuthenticationEvent.json	test(data_classes): correct some of the unit tests	2 years ago
cognitoPostConfirmationEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
cognitoPreAuthenticationEvent.json	test(data_classes): correct some of the unit tests	2 years ago
cognitoPreSignUpEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
cognitoPreTokenGenerationEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
cognitoUserMigrationEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
cognitoVerifyAuthChallengeResponseEvent.json	feat(data_classes): Cognito custom auth triggers	2 years ago
connectContactFlowEventAll.json	feat(data-classes): Add connect contact flow event (#304)	2 years ago
connectContactFlowEventMin.json	feat(data-classes): Add connect contact flow event (#304)	2 years ago
dynamoStreamEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
eventBridgeEvent.json	feat: include new replay-name field in parser and data_classes	2 years ago
kafkaEventMsk.json	feat(data-classes): add KafkaEvent and KafkaEventRecord (#1485)	28 days ago
kafkaEventSelfManaged.json	feat(parser): add KafkaMskEventModel and KafkaSelfManagedEventModel (#...	28 days ago
kinesisFirehoseKinesisEvent.json	feat(parser): add KinesisFirehoseModel (#1556)	yesterday
kinesisFirehosePutEvent.json	feat(parser): add KinesisFirehoseModel (#1556)	yesterday
kinesisStreamEvent.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
lambdaFunctionUrlEvent.json	fix(event_sources): add test for Function URL AuthZ (#1421)	2 months ago
lambdaFunctionUrlIAMEvent.json	fix(event_sources): add test for Function URL AuthZ (#1421)	2 months ago
rabbitMQEvent.json	feat(data-classes): ActiveMQ and RabbitMQ support (#770)	12 months ago
s3Event.json	feat(trigger): data class and helper functions for lambda trigger eve...	2 years ago
s3EventDecodedKey.json	test(general): Add missing tests for parser	2 years ago
s3EventGlacier.json	feat: Add S3 lambda event support to Parser utility #224	2 years ago
s3ObjectEventIAMUser.json	feat(data-classes): Add S3 Object Lambda Event (#353)	2 years ago
s3ObjectEventTempCredentials.json	feat(data-classes): Add S3 Object Lambda Event (#353)	2 years ago
sesEvent.json	fix(data-classes): Add missing SES fields and (#1045)	6 months ago
snsEvent.json	Feature: Add sns notification support to Parser utility #206	2 years ago
snsSqsEvent.json	feat: add support for SNS->SQS protocol (#272)	2 years ago
sqsEvent.json	feat: Advanced parser utility (pydantic)	2 years ago

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up next: Long-term

Lambda Powertools

Beyond Lambda — AWS Fargate, AWS Glue, etc.

Testing — Event Builder, data fetcher (E2E test)

/You tell us/ — Making your experience smoother

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Summary

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Summary

Key lessons learned

Customers grew	So did the community
Balancing act	Ops excellence is business as usual
Empathy	Practice and refine what we preach

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources



All links here

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Heitor Lessa (he/him)
@heitor_lessa



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.