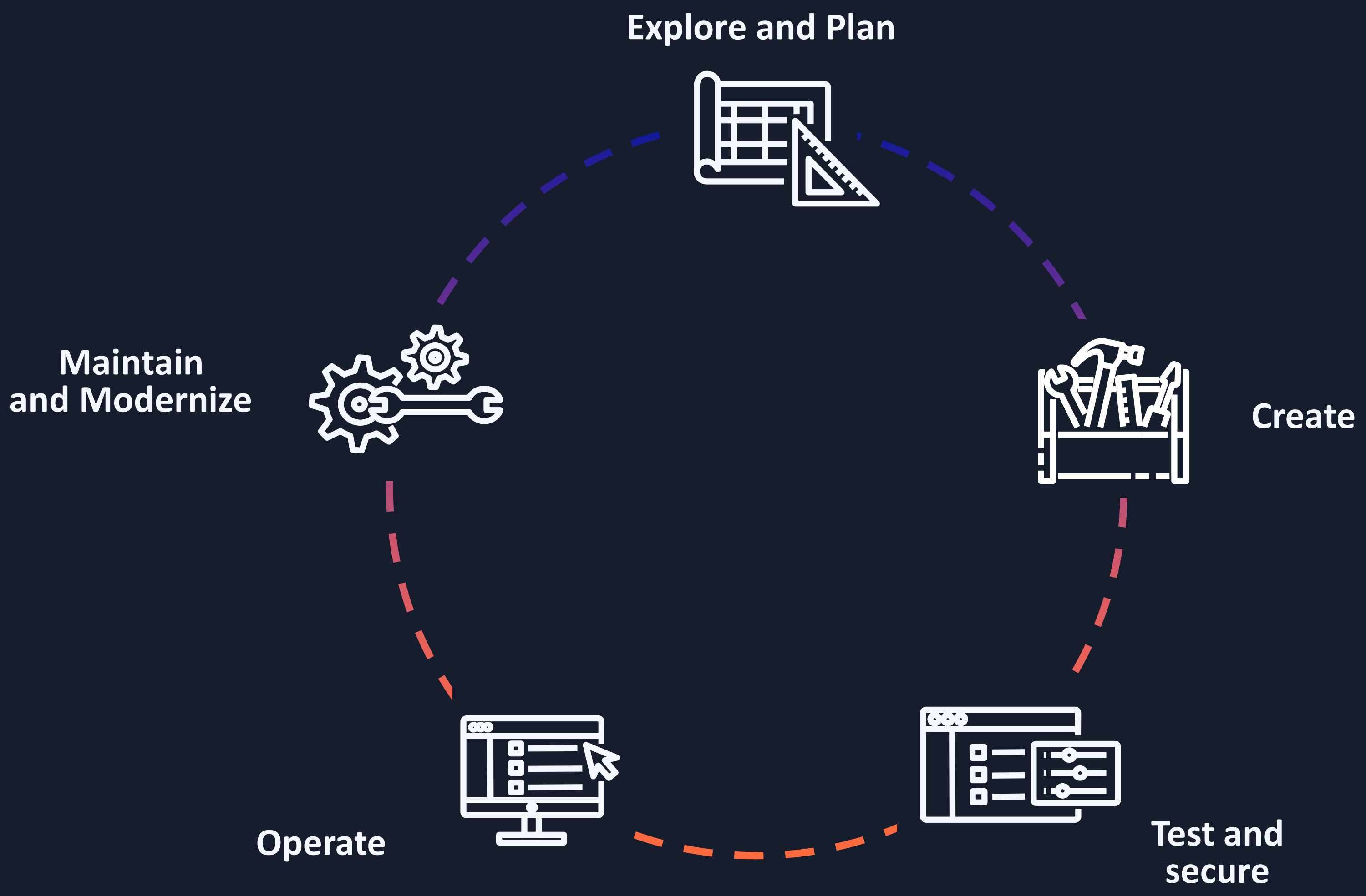
Bringing AI to Agile software development with Amazon Q Developer

José Nunes

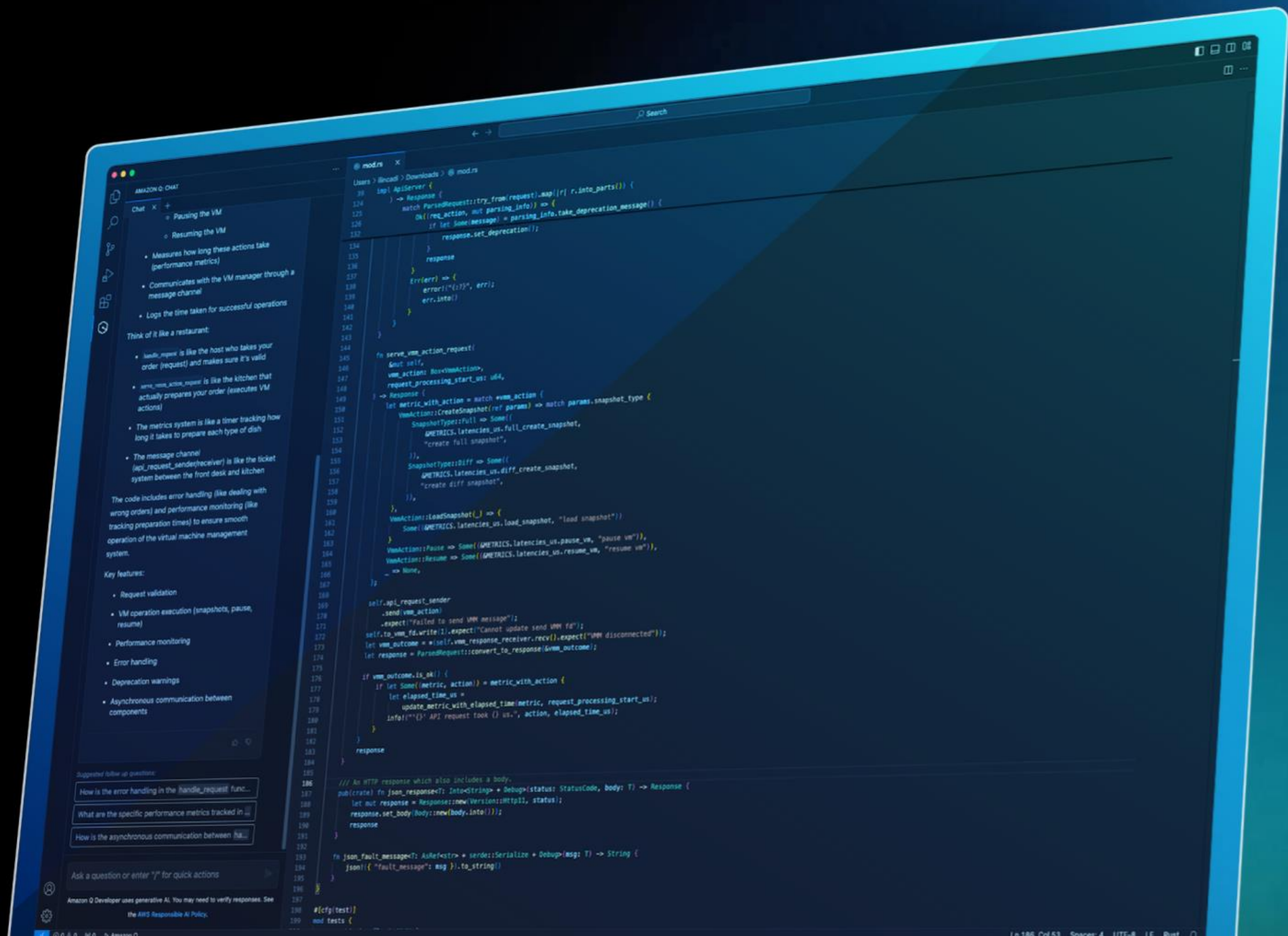
Solutions Architect
Amazon Web Services

Where are developers spending time in the SDLC?



Amazon Q Developer

The most capable generative AI-powered assistant for software development.



- Pausing the VM
- Resuming the VM
- Measures how long these actions take (performance metrics)
- Communicates with the VM manager through a message channel
- Logs the time taken for successful operations

Think of it like a restaurant:

- handle_request is like the host who takes your order (request) and makes sure it's valid
- serve_vmm_action_request is like the kitchen that actually prepares your order (executes VM actions)
- The metrics system is like a timer tracking how long it takes to prepare each type of dish
- The message channel (api_request_sender/receiver) is like the ticket system between the front desk and kitchen

The code includes error handling (like dealing with wrong orders) and performance monitoring (like tracking preparation times) to ensure smooth operation of the virtual machine management system.

Key features:

- Request validation
- VM operation execution (snapshots, pause, resume)
- Performance monitoring
- Error handling
- Deprecation warnings
- Asynchronous communication between components

Suggested follow up questions:

How is the error handling in the handle_request func...

What are the specific performance metrics tracked in ...

How is the asynchronous communication between ha...

Ask a question or enter "/" for quick actions

Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy.

```
impl ApiServer {
    ) -> Response {
        match ParsedRequest::try_from(request).map(|r| r.into_parts()) {
            Ok((req_action, mut parsing_info)) => {
                if let Some(message) = parsing_info.take_deprecation_message() {
                    response.set_deprecation();
                }
                response
            }
            Err(err) => {
                error!("{:?}", err);
                err.into()
            }
        }
    }

    fn serve_vmm_action_request(
        &mut self,
        vmm_action: Box<VmmAction>,
        request_processing_start_us: u64,
    ) -> Response {
        let metric_with_action = match *vmm_action {
            VmmAction::CreateSnapshot(ref params) => match params.snapshot_type {
                SnapshotType::Full => Some((
                    &METRICS.latencies_us.full_create_snapshot,
                    "create full snapshot",
                )),
                SnapshotType::Diff => Some((
                    &METRICS.latencies_us.diff_create_snapshot,
                    "create diff snapshot",
                )),
            },
            VmmAction::LoadSnapshot(_) => {
                Some((&METRICS.latencies_us.load_snapshot, "load snapshot"))
            }
            VmmAction::Pause => Some((&METRICS.latencies_us.pause_vm, "pause vm")),
            VmmAction::Resume => Some((&METRICS.latencies_us.resume_vm, "resume vm")),
            _ => None,
        };

        self.api_request_sender
            .send(vmm_action)
            .expect("Failed to send VMM message");
        self.to_vmm_fd.write(1).expect("Cannot update send VMM fd");
        let vmm_outcome = *(self.vmm_response_receiver.recv().expect("VMM disconnected"));
        let response = ParsedRequest::convert_to_response(&vmm_outcome);

        if vmm_outcome.is_ok() {
            if let Some((metric, action)) = metric_with_action {
                let elapsed_time_us =
                    update_metric_with_elapsed_time(metric, request_processing_start_us);
                info!("'{}' API request took {} us.", action, elapsed_time_us);
            }
        }
        response
    }
}

/// An HTTP response which also includes a body.
pub(crate) fn json_response<T: Into<String> + Debug>(status: StatusCode, body: T) -> Response {
    let mut response = Response::new(Version::Http11, status);
    response.set_body(Body::new(body.into()));
    response
}

fn json_fault_message<T: AsRef<str> + serde::Serialize + Debug>(msg: T) -> String {
    json!({ "fault_message": msg }).to_string()
}
}

#[cfg(test)]
mod tests {
```

Amazon Q Developer specialized agents can automate tasks across the SDLC

Software Development Agents write and implement entire application features in minutes

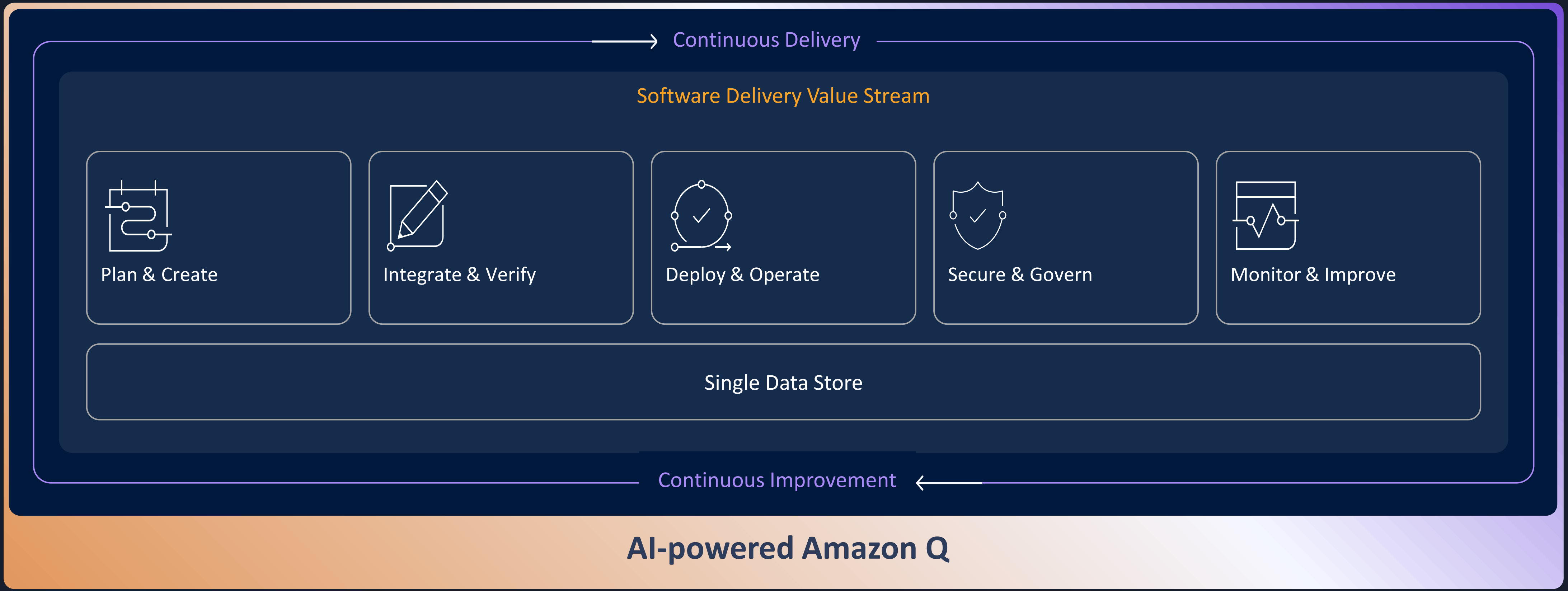
Unit Test agent generates and add tests to the project, helping improve code quality, fast.

Documentation agent generates and updates README files, create data-flow diagrams, and keep projects fully documented.

Code review agent catches bugs and security vulnerabilities, or misconfigurations

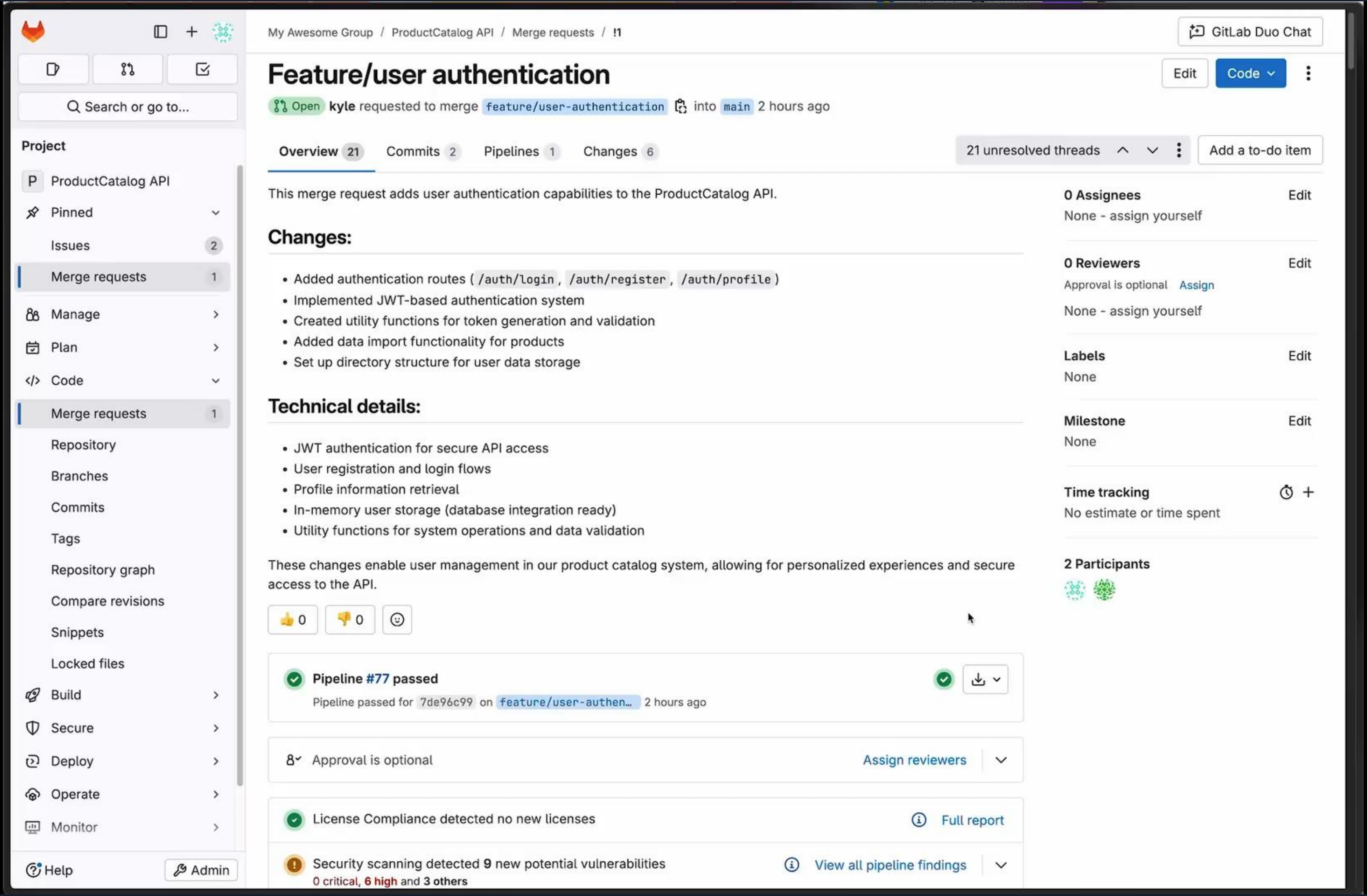
Transformation agents accelerate large-scale enterprise workload migration and modernizations

Integration with a comprehensive DevSecOps Platform



Vulnerability resolution example

GENERATE A MERGE REQUEST THAT ADDRESSES A VULNERABILITY



Feature/user authentication

Open **kyle** requested to merge `feature/user-authentication` into `main` 2 hours ago

This merge request adds user authentication capabilities to the ProductCatalog API.

Changes:

- Added authentication routes (`/auth/login` , `/auth/register` , `/auth/profile`)
- Implemented JWT-based authentication system
- Created utility functions for token generation and validation
- Added data import functionality for products
- Set up directory structure for user data storage

Technical details:

- JWT authentication for secure API access
- User registration and login flows
- Profile information retrieval
- In-memory user storage (database integration ready)
- Utility functions for system operations and data validation

These changes enable user management in our product catalog system, allowing for personalized experiences and secure access to the API.

Pipeline #77 passed
Pipeline passed for `7de96c99` on `feature/user-authen...` 2 hours ago

Approval is optional — Assign reviewers

License Compliance detected no new licenses — Full report

Security scanning detected 9 new potential vulnerabilities — View all pipeline findings
0 critical, **6 high** and **3 others**

0 Assignees Edit
None - assign yourself

0 Reviewers Edit
Approval is optional Assign
None - assign yourself

Labels Edit
None

Milestone Edit
None

Time tracking
No estimate or time spent

2 Participants

Agents are transforming Amazon’s internal systems

10,000+	4,500+	$260M
APPS MIGRATED	DEVELOPER YEARS SAVED	ANNUAL SAVINGS

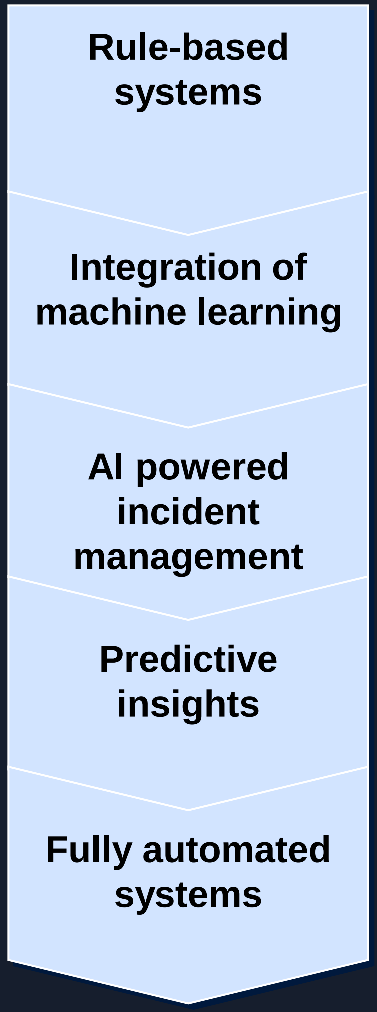
First major cloud provider to announce

ISO 42001

accredited certification for AI

Enter AIOps

The expected evolution of AIOps


Rule-based systems
Integration of machine learning
AI powered incident management
Predictive insights
Fully automated systems

- **Rule-based systems** trigger automation
- **ML models** for anomaly detection
- **Ensemble of techniques (rule-based, ML, AI, gen AI)** for dynamic incident correlation, root cause analysis, and remediation suggestions
- **Predictive capabilities** to forecast future disruptions and capacity needs
- **The holy grail** – systems auto detect issues that are brewing and self heal without human intervention

What challenges can AIOps help with?

Too much data

Ever-increasing volume and diversity of telemetry data

What changed?

Difficult to identify what specifically changed over time

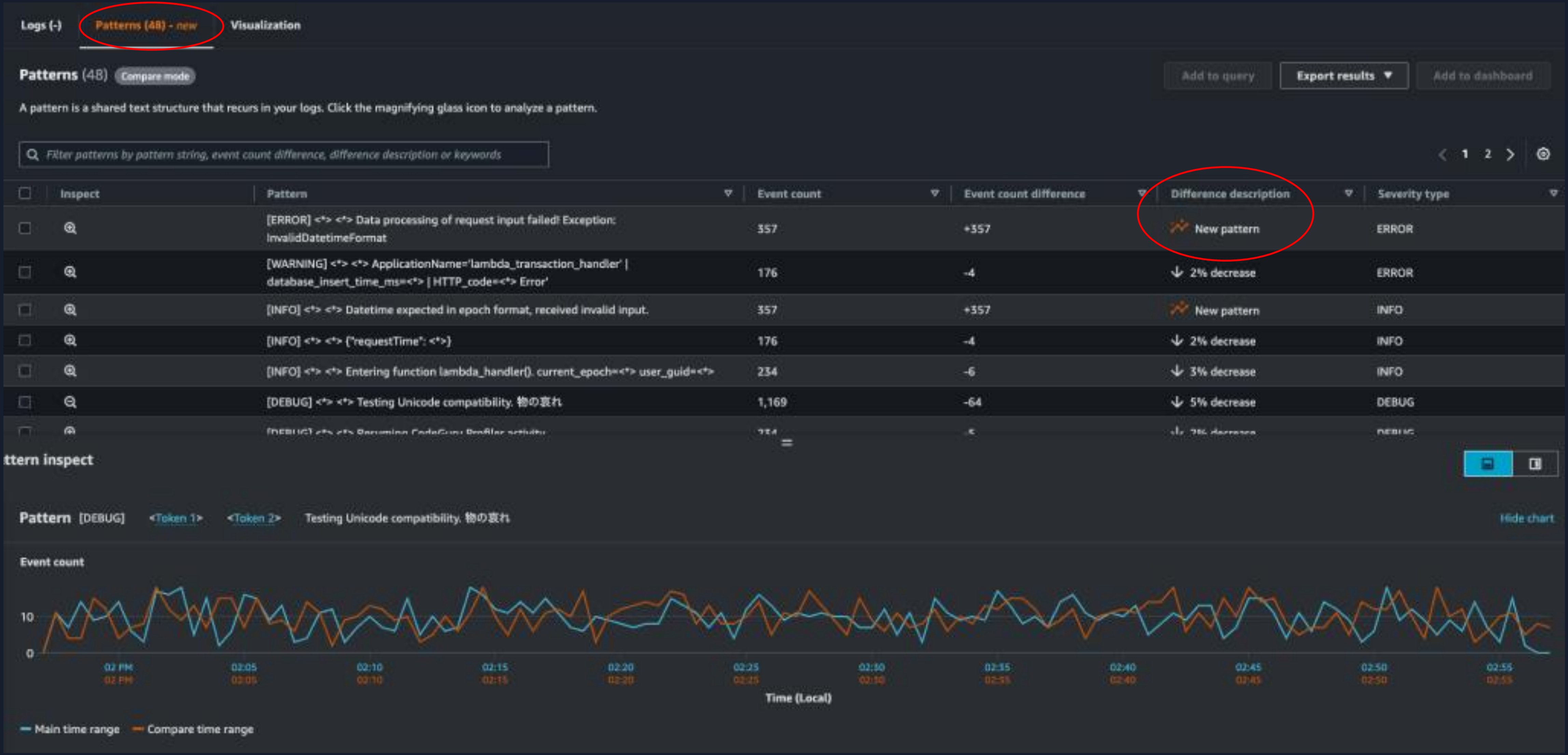
Proactive detection

Identifying unusual changes in your application

Unknown unknowns

Monitoring for unforeseen issues

Compare mode in CloudWatch Logs Insights



Logs (-)
Patterns (48) - new
Visualization
Add to query
Export results ▼
Add to dashboard
Filter patterns by pattern string, event count difference, difference description or keywords
1 2

Patterns (48) Compare mode

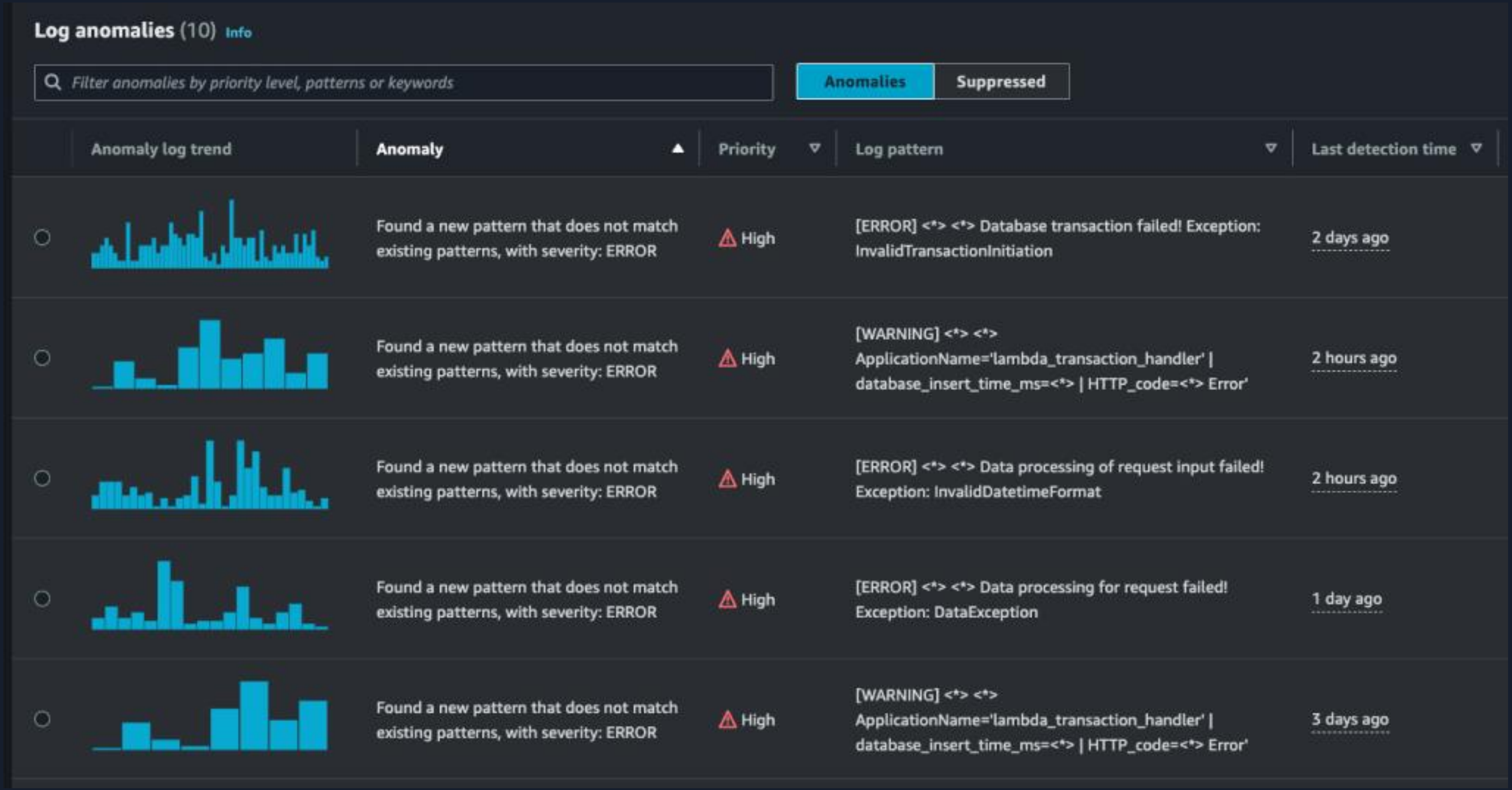
A pattern is a shared text structure that recurs in your logs. Click the magnifying glass icon to analyze a pattern.

□	Inspect	Pattern	Event count	Event count difference	Difference description	Severity type
□		[ERROR] <*> <*> Data processing of request input failed! Exception: InvalidDatetimeFormat	357	+357	New pattern	ERROR
□		[WARNING] <*> <*> ApplicationName='lambda_transaction_handler' | database_insert_time_ms=<*> | HTTP_code=<*> Error'	176	-4	↓ 2% decrease	ERROR
□		[INFO] <*> <*> Datetime expected in epoch format, received invalid input.	357	+357	New pattern	INFO
□		[INFO] <*> <*> {"requestTime": <*>}	176	-4	↓ 2% decrease	INFO
□		[INFO] <*> <*> Entering function lambda_handler(). current_epoch=<*> user_guid=<*>	234	-6	↓ 3% decrease	INFO
□		[DEBUG] <*> <*> Testing Unicode compatibility. 物の哀れ	1,169	-64	↓ 5% decrease	DEBUG
□		[DEBUG] <*> <*> Resuming CodeGuru Profiler activity	234	-5	↓ 2% decrease	DEBUG

ttern inspect

Pattern [DEBUG] <Token 1> <Token 2> Testing Unicode compatibility. 物の哀れ

Always-on anomaly detection



Log anomalies (10) Info

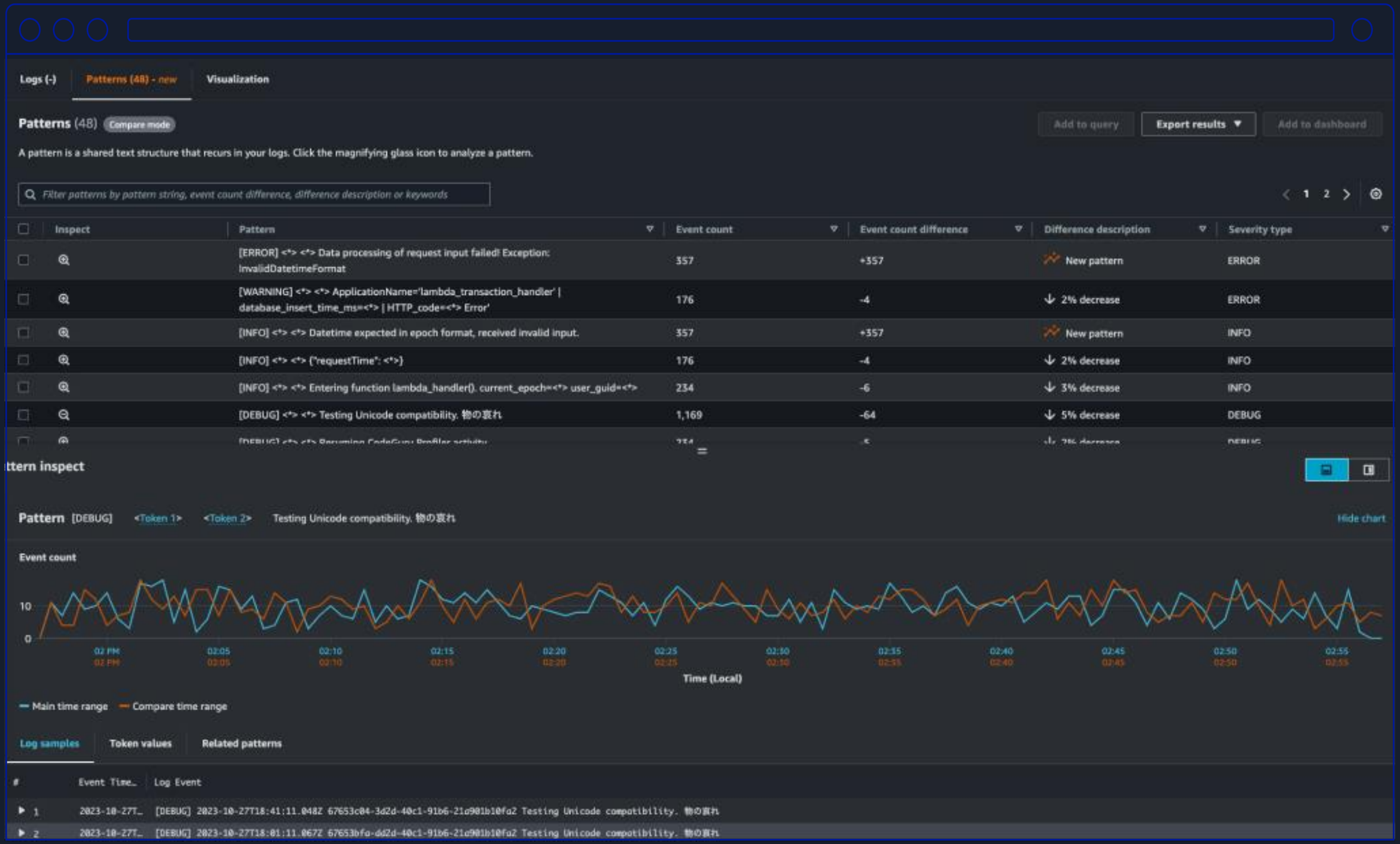
	Anomaly log trend	Anomaly ▲	Priority ▽	Log pattern ▽	Last detection time ▽
○		Found a new pattern that does not match existing patterns, with severity: ERROR	⚠ High	[ERROR] <*> <*> Database transaction failed! Exception: InvalidTransactionInitiation	2 days ago
○		Found a new pattern that does not match existing patterns, with severity: ERROR	⚠ High	[WARNING] <*> <*> ApplicationName='lambda_transaction_handler' | database_insert_time_ms=<*> | HTTP_code=<*> Error'	2 hours ago
○		Found a new pattern that does not match existing patterns, with severity: ERROR	⚠ High	[ERROR] <*> <*> Data processing of request input failed! Exception: InvalidDatetimeFormat	2 hours ago
○		Found a new pattern that does not match existing patterns, with severity: ERROR	⚠ High	[ERROR] <*> <*> Data processing for request failed! Exception: DataException	1 day ago
○		Found a new pattern that does not match existing patterns, with severity: ERROR	⚠ High	[WARNING] <*> <*> ApplicationName='lambda_transaction_handler' | database_insert_time_ms=<*> | HTTP_code=<*> Error'	3 days ago

ML-powered logs pattern analysis and logs anomaly detection

Use **patterns** view to visualize recurring patterns when querying your logs

Compare mode helps answer “what changed” over time

Always-on **anomaly detection:** proactive notification of emerging issues



Patterns (48)

A pattern is a shared text structure that recurs in your logs. Click the magnifying glass icon to analyze a pattern.

Filter patterns by pattern string, event count difference, difference description or keywords

□	Inspect	Pattern	Event count	Event count difference	Difference description	Severity type
□	🔍	[ERROR] <*> <*> Data processing of request input failed! Exception: InvalidDatetimeFormat	357	+357	New pattern	ERROR
□	🔍	[WARNING] <*> <*> ApplicationName='lambda_transaction_handler' | database_insert_time_ms=<*> | HTTP_code=<*> Error'	176	-4	↓ 2% decrease	ERROR
□	🔍	[INFO] <*> <*> Datetime expected in epoch format, received invalid input.	357	+357	New pattern	INFO
□	🔍	[INFO] <*> <*> {"requestTime": <*>}	176	-4	↓ 2% decrease	INFO
□	🔍	[INFO] <*> <*> Entering function lambda_handler(). current_epoch=<*> user_guid=<*>	234	-6	↓ 3% decrease	INFO
□	🔍	[DEBUG] <*> <*> Testing Unicode compatibility. 物の哀れ	1,169	-64	↓ 5% decrease	DEBUG
□	🔍	[DEBUG] <*> <*> Resuming CodeGuru Profiler activity	234	-5	↓ 2% decrease	DEBUG

Pattern [DEBUG] <Token 1> <Token 2> Testing Unicode compatibility. 物の哀れ

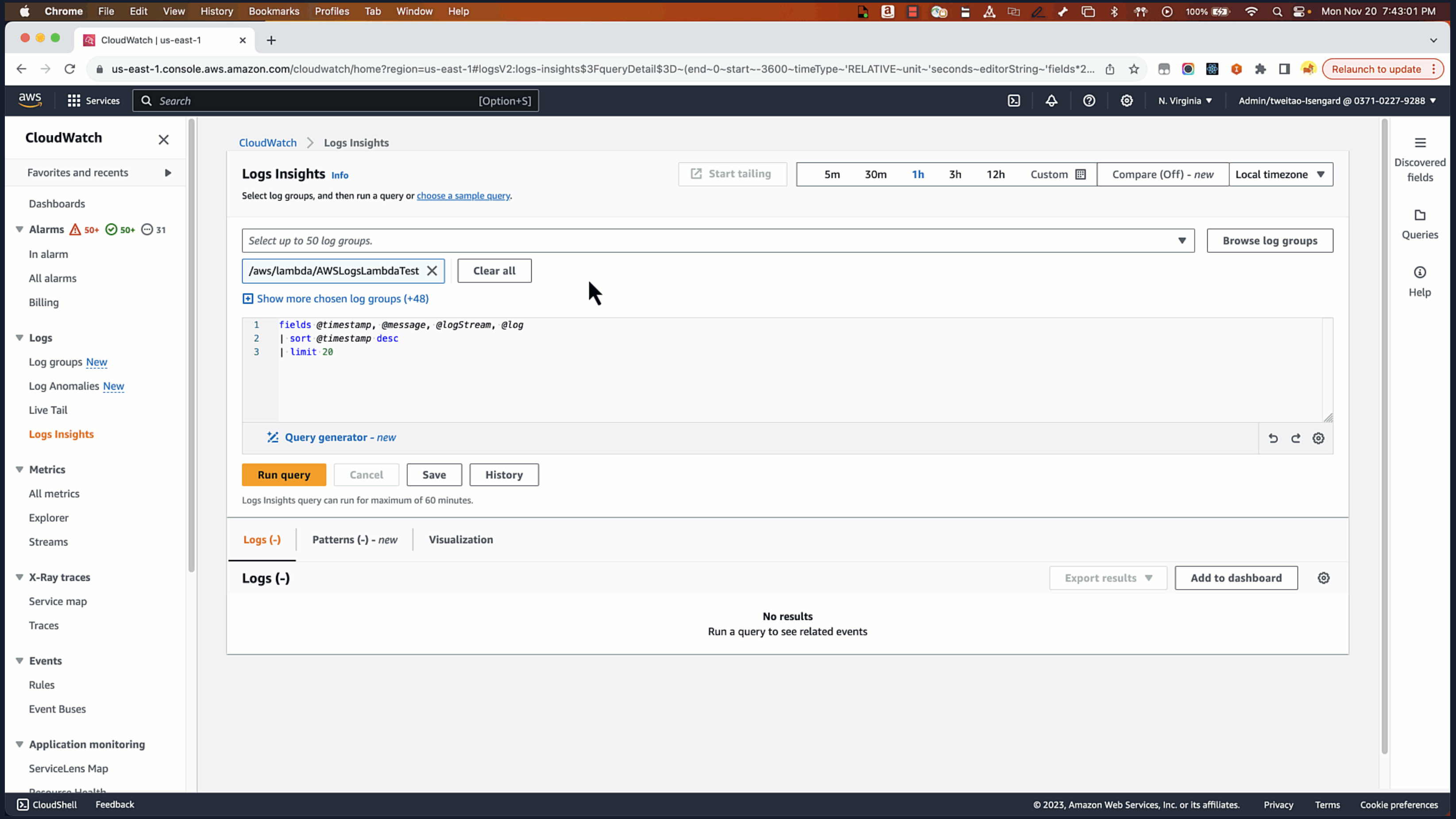
```
#     Event Time…   Log Event
▸ 1   2023-10-27T…  [DEBUG] 2023-10-27T18:41:11.048Z 67653c04-3d2d-40c1-91b6-21a901b10fa2 Testing Unicode compatibility. 物の哀れ
▸ 2   2023-10-27T…  [DEBUG] 2023-10-27T18:01:11.067Z 67653bfa-dd2d-40c1-91b6-21a901b10fa2 Testing Unicode compatibility. 物の哀れ
```

AI-powered natural language query generation

Easy getting started: Generate queries to interact with your Logs and Metrics by asking questions in natural language

Develop query expertise: Provides line by line explanation of the generated query to help you learn the syntax

Iterative deep dives: Update existing queries with natural language instructions for guided query iteration



Select log groups, and then run a query or choose a sample query.

Select up to 50 log groups.

/aws/lambda/AWSLogsLambdaTest

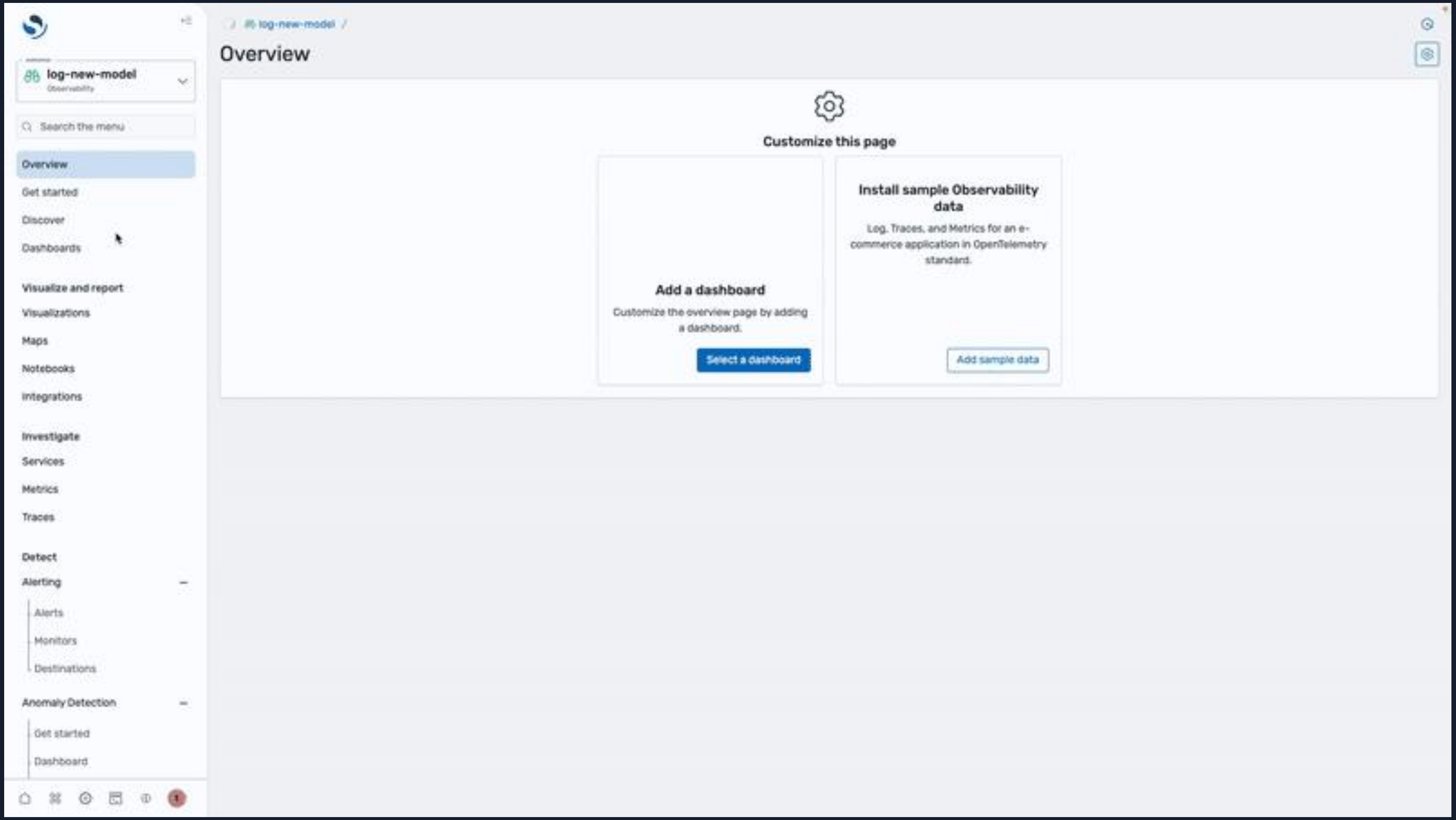
```
fields @timestamp, @message, @logStream, @log
| sort @timestamp desc
| limit 20
```

Logs Insights query can run for maximum of 60 minutes.

Logs (-)

No results
Run a query to see related events

Operation analytics made fast, accessible and intelligent



Overview

Customize this page

Add a dashboard

Customize the overview page by adding a dashboard.

Install sample Observability data

Log, Traces, and Metrics for an e-commerce application in OpenTelemetry standard.

Challenges in large-scale operations

- Lack of data
- Information overload
- Alarm and tool fatigue
- Data correlation
- Remediation and prevention

Amazon Q Developer Operational Investigations

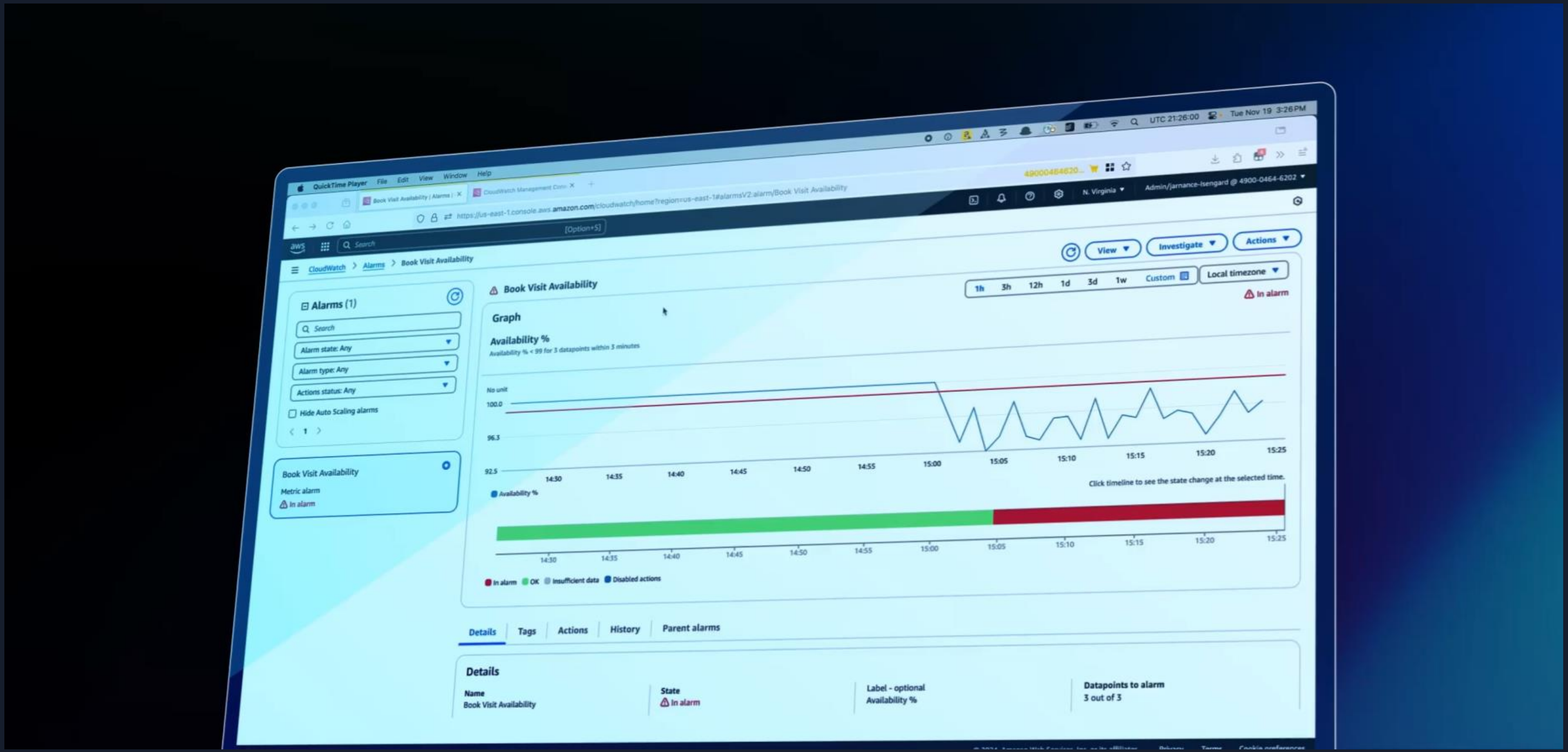
- Localize issues in a distributed environment

- Guided operational troubleshooting so that manual guesswork is removed

- Prioritized list of root cause hypotheses (diagnoses)

- Recommended remediation and optimization actions

Investigate and remediate issues



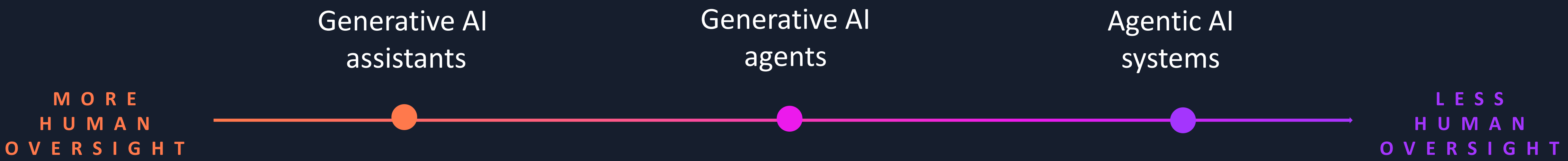
Details

Name	State	Label - optional	Datapoints to alarm
Book Visit Availability	In alarm	Availability %	3 out of 3

- Get meaningful investigation hypotheses and guidance for remediation
- Streamline investigation summaries and updates
- Trigger investigations from alarms, metrics,or manually chatting with Q

What’s next?

Evolving into Agentic AI



MORE HUMAN OVERSIGHT

Generative AI assistants

- Follow a set of rules
- Automate repetitive tasks

Generative AI agents

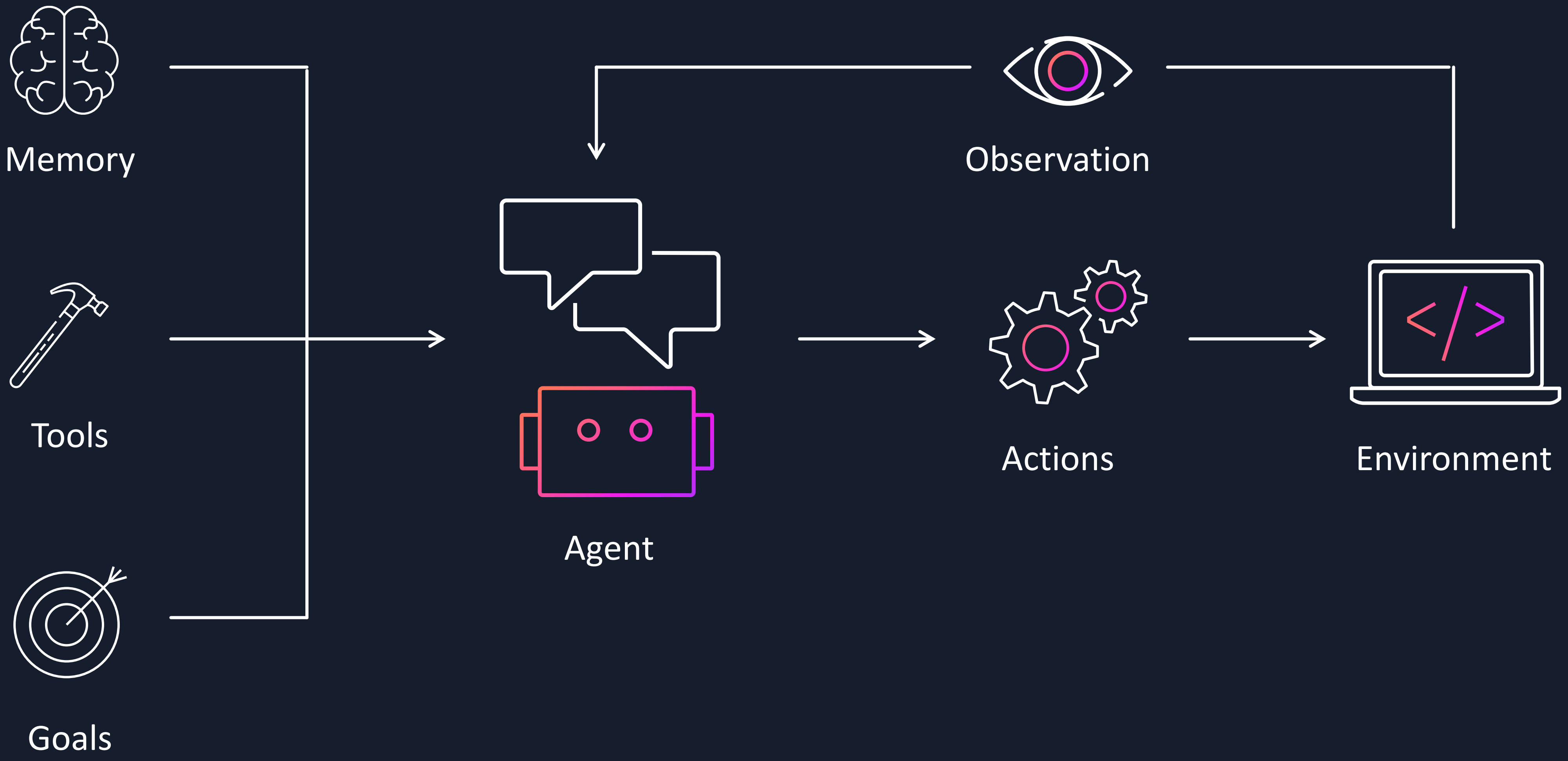
- Achieve a singular goal
- Address broader range of tasks
- Automate entire workflows

Agentic AI systems

- Fully autonomous
- Multi-agent systems
- Mimic human logic and reasoning

LESS HUMAN OVERSIGHT

Agentic AI for full automation



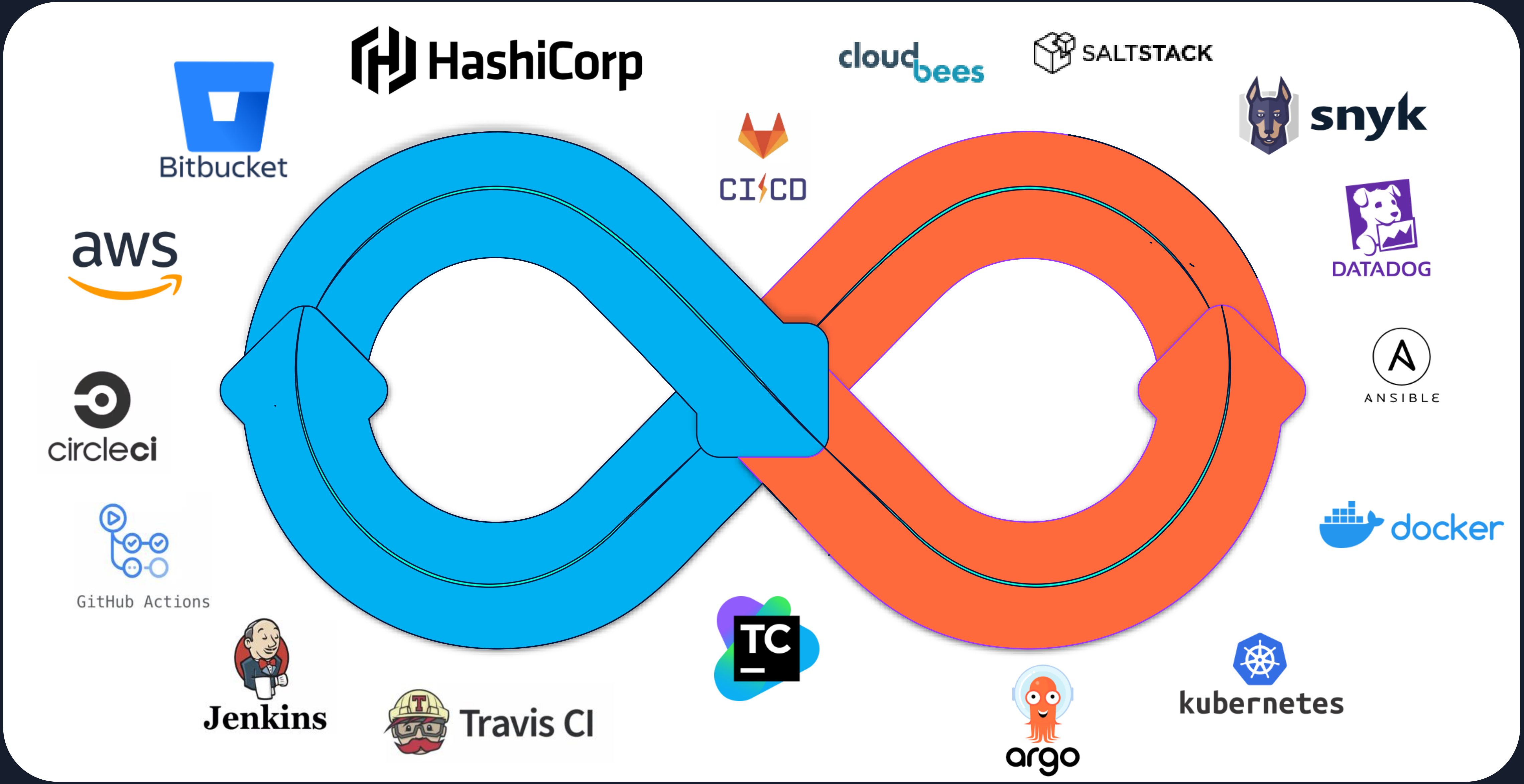
Open Source Protocols

Model Context Protocol

Introduced by Anthropic, MCP provides a standardized way to connect AI models to different data sources and AI-powered tools.

- MCP servers available for commonly used platforms (e.g. GitLab, GitHub, BitBucket)
- Run MCP servers and clients on AWS using open source MCP libraries
- Open-source AWS MCP servers for code assistants
- Amazon Q Developer CLI includes MCP support

Integrate with popular CI/CD ecosystem



Thank you

José Nunes

Solutions Architect
Amazon Web Services