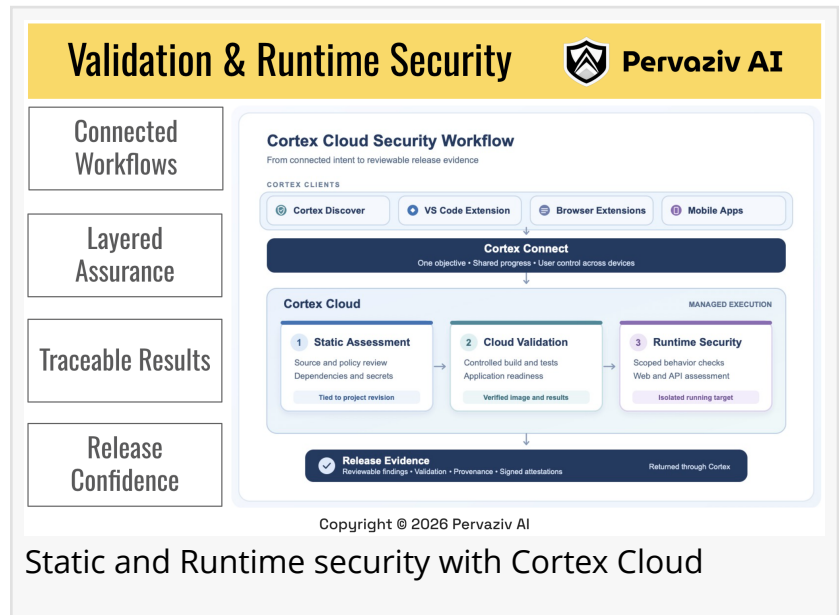


Pervaziv AI Extends Cortex Cloud with Static and Runtime Security Across the Software Lifecycle

New Cortex Cloud workflow links source, dependency and secret analysis with isolated builds, offline tests, runtime assessment and release evidence.

SAN FRANCISCO, CA, UNITED STATES, September 23, 2026 / EINPresswire.com/ -- [Pervaziv AI](#) today announced the [Cortex](#) Cloud Security Workflow, extending its managed AI execution environment with a connected path for static analysis, software validation, runtime security assessment and release evidence.



The new workflow is designed for a software development landscape in which AI coding agents are moving beyond suggestions and increasingly participating in implementation, testing, remediation and longer running engineering tasks. As that responsibility grows, security cannot sit only at the beginning or end of the development process. Teams need a way to understand not only what changed in source code, but also what was built, what actually ran, what the application exposed, which checks were performed and whether the resulting evidence belongs to the exact release candidate under review.

“

Security cannot be limited to scanning the finished code. It must be a continuous process that also accounts for the execution environment, including configurations, permissions, network restrictions.”

Irina Calciu

[Cortex Cloud](#) now brings those steps into one managed sequence. An eligible project can move from a fixed source revision through static assessment, validation planning, a qualified image build, offline tests, runtime verification, bounded security assessment and release evidence. Each stage is connected to the project and artifacts that came

before it, giving engineering and security teams a clearer path from an AI assisted change to

evidence they can inspect.

That distinction is central to the release. A scan result on its own can be useful, but it often leaves important questions unanswered. Which revision produced the finding? Was the application able to build? Did its planned tests pass? Did the same image become healthy in the runtime environment? What routes were actually reachable? Did the security result come from that same validated artifact?

The Cortex Cloud Security Workflow is designed to keep those questions connected.

Executive Commentary

"AI coding agents are moving from suggestions to execution, so security has to move with them," said Anoop Jaishankar, Founder and CEO of Pervaziv AI. "The important question is no longer just what the code looks like. It is what was built, what actually ran, what the application exposed, and whether every result belongs to the same release candidate. Cortex Cloud connects that evidence into one workflow, making security part of execution rather than a scan at the end."

Irina Calciu, Founding Systems Architect at Pervaziv AI, said, "AI coding agents are taking on greater responsibility across software development. In this rapidly evolving landscape, security cannot be limited to scanning finished code. It must be a continuous process that also accounts for the execution environment, including configurations, permissions, and network restrictions. Cortex uses dedicated cloud execution environments to isolate execution and perform project-specific builds, dependency checks, and security scans. While no single scan can guarantee that an AI-generated change is perfectly secure, Cortex provides customers with a repeatable process to help them evaluate every code change."

Security Becomes Part of Managed Execution

Cortex Cloud launched as a managed execution destination for eligible Cortex work. It gave users a choice between working locally and moving appropriate tasks into a durable cloud workspace, while keeping the user, project state, workflow progress and result connected.

The Cortex Cloud Security Workflow extends that foundation rather than creating a separate security product or disconnected scanning path.

Before application code runs, Cortex Cloud can examine project metadata, source and available repository evidence to understand how the project should be validated and where visible risks may exist. The workflow can consider manifests, lockfiles, relevant continuous integration configuration, container definitions and available test evidence. Static analysis, dependency checks and secret scanning provide a view of risks that can be identified before runtime.

Cortex then creates a deterministic validation plan tied to the source revision and the evidence used to create it. Planning does not execute repository code. Unsupported or ambiguous configurations are surfaced rather than silently guessed, helping preserve a clear boundary between what Cortex knows from the project and what would require additional input or support.

The result is a security process grounded in a specific project state, not a generic scan detached from the software that will eventually run.

Validate Before Assessing Runtime Behavior

Runtime security results become more useful when the target has first been established as the intended application.

Cortex Cloud uses a dedicated build stage to create and qualify an immutable validation image. The workflow checks that the source evidence and approved plan still correspond to the build, applies controls to the build context, evaluates the container definition against Cloud policy, and records the identity and provenance of the resulting image.

The image is then passed into an ephemeral validation environment as a digest checked artifact.

Planned tests run in an offline container. Cortex Cloud records the outcome, duration, exit status, bounded output and digests of the complete output streams. Failures can be classified so teams can distinguish a product test failure from an unavailable dependency, a resource limit or an environment problem.

A passed test run is required before the workflow moves into runtime verification.

At that stage, Cortex Cloud starts the application under a narrow declared contract. The workflow can use a defined port, an HTTP or TCP readiness check, and bounded startup and observation periods. It records whether the application became ready, remained healthy, exited unexpectedly or encountered relevant resource conditions.

This sequence matters because a runtime scan against an application that never started correctly can create noise rather than confidence. Cortex Cloud establishes what is running before asking what that running application exposes.

Four Views of the Running Application

Once the validated image is healthy, the Cortex Cloud Security Workflow can apply several complementary forms of runtime assessment.

Scoped discovery maps resources that are reachable on the running application within a fixed target and request budget. This helps identify the routes the application actually exposes in the assessment environment.

Passive web analysis observes responses and identifies issues that can be detected without an active attack sequence.

Curated HTTP checks evaluate the running target against a restricted set of supported security checks.

API examples and coverage testing exercise supported behavior described by the application's API specification, providing another view of how the declared interface compares with observed behavior.

These approaches answer different questions. Source inspection can reveal a risky code pattern. A dependency check can identify a vulnerable component. A runtime assessment can reveal an unexpected reachable resource, an unsafe response or behavior that becomes visible only after the application starts. API testing can show where declared behavior and observed behavior do not align.

The runtime assessment remains deliberately bounded. It runs against a fresh instance of the validated image on an internal network without published host ports or external network access. Supported assessment types and quick or standard profiles define the scope of work. Results can then be normalized, redacted where needed, fingerprinted and correlated into a single assessment with severity counts and discovery evidence.

A finding remains evidence for review. It is not permission for Cortex Cloud to edit the repository or deploy a change.

Static and Runtime Evidence in One Release View

One of the broader goals of the release is to reduce the gap between security tools that inspect source and tools that observe behavior.

Those views are complementary, not interchangeable.

Static analysis can identify patterns and weaknesses visible in code. Secret scanning can surface exposed credentials or sensitive values. Dependency analysis can identify risks in the software materials used to build an application. Repository and build policy checks can identify whether the project is suitable for controlled validation. Runtime assessment can show what the built application actually exposes and how it behaves within the assessment scope.

Cortex Cloud ties those results to a shared release identity.

A later result cannot simply point to a similar image, a newer repository state or an unrelated validation attempt. The workflow checks the relationship among the owner, workspace, source revision, plan, build, validation run and runtime specification before the next stage can proceed.

That creates a more useful review model. Instead of reconciling a source result from one commit, a test report from another build and a runtime scan from a different image, teams can review evidence associated with the same release candidate.

A clean assessment also has a precise meaning. It means the supported checks found no issue within their defined scope. It does not mean every endpoint, configuration, input or attack path has been tested, and it is not a guarantee that the software is perfectly secure.

This also changes how teams can think about AI assisted security operations. Instead of treating generation, scanning, testing and runtime review as unrelated checkpoints owned by different tools, Cortex Cloud can preserve the identity of the work as it moves through each stage. Developers still decide what to accept. Security teams still determine whether additional investigation is required. Release owners still decide whether evidence is sufficient for their process. The platform's role is to make those decisions better informed by keeping the relevant artifact, environment, assessment scope and resulting evidence connected rather than asking people to reconstruct that relationship after the fact.

Evidence That Follows the Release

The workflow is designed to return more than a pass or fail result.

A release evidence package can include static analysis, dependency and secret scan results associated with the candidate under review. It can include normalized build records, test results, runtime verification and security assessment records. It can preserve the source revision, validation plan, configuration, image and artifact digests that connect those stages.

Runtime findings can be returned with discovery resources, severity counts and the constraints under which the assessment ran. The release evidence can also include a software bill of materials, release provenance, artifact integrity checks and signed attestations that can be verified separately.

The objective is traceability.

A reviewer should be able to follow a claim back to the artifact that was actually assessed and understand how that artifact reached the point of assessment. The workflow also helps prevent a common release problem: assembling reassuring evidence from different revisions, images or attempts into one status that appears more coherent than the underlying process really was.

For organizations adopting more AI assisted development, that traceability becomes increasingly important. Faster code generation can increase the volume of changes moving through review. Security teams need evidence that can keep pace without reducing the review process to a larger stream of isolated findings.

Connected to Cortex Across Work Surfaces

The security workflow also extends the broader Cortex architecture Pervaziv AI has been building across user surfaces and execution environments.

Cortex Connect maintains continuity of intent, progress and control across supported mobile, browser and Visual Studio Code experiences. A developer can begin an investigation close to the code, follow progress elsewhere and return to the project for review without rebuilding the objective from scratch.

Cortex Cloud provides the managed execution environment for eligible work that should run remotely.

Cortex Discover adds a dedicated agentic AI browser with live page awareness, selected tab context, visual understanding and supported browser actions. An issue noticed while reviewing documentation, a pull request, a dashboard or another web application can become part of a connected Cortex workflow and, when eligible, move into Cortex Cloud for project aware validation.

The latest Cortex inference cache architecture addresses a different concern: reducing repeated processing when eligible context, prompt prefixes or exact responses can be safely reused under their respective validity rules.

The new security workflow adds another layer to that progression. It asks what evidence should be produced when AI assisted work reaches build, validation and runtime assessment.

The roles remain distinct. Cortex Connect keeps the work connected. Cortex Discover brings browser context into the experience. Cortex Cloud provides managed execution. Inference caching improves eligible reuse. The Cortex Cloud Security Workflow adds connected security and release evidence to the work that runs there.

From Finding to Reviewable Decision

Security teams rarely need another isolated list of findings. They need enough context to decide what a finding means.

Was the target healthy? Which revision was assessed? Which image was running? What static

checks had already been performed? What runtime scope was used? Was the application reachable only inside the assessment environment? Which evidence is complete, and what still needs deeper testing?

Cortex Cloud is designed to make those questions easier to answer from the workflow itself.

The release also reflects a broader direction for Cortex. Pervaziv AI has been separating the responsibilities required to move Enterprise AI from answers toward dependable work. Specialized models and agents provide different forms of intelligence. Routing connects requests with models, current information and engineering skills. Planning structures larger objectives. Connect preserves continuity. Discover brings the browser into the execution experience. Cloud provides a durable managed place for eligible work to run. Inference caching reduces repeated processing where reuse remains valid.

Security now becomes part of that execution path.

This is particularly important as coding agents take on larger tasks. The useful unit of review is no longer only the text an AI model generated. Teams increasingly need to evaluate the resulting software system: its source, dependencies, build, tests, runtime behavior, reachable surface and evidence trail.

Cortex Cloud creates a controlled place to perform that evaluation while preserving the relationship among those stages.

From Code to Runtime Evidence

The Cortex Cloud Security Workflow is not built around the claim that one tool can prove software is secure. It is built around a more practical goal: make security assessment continuous, traceable and connected to the software that actually ran.

That means inspecting what can be known before execution, validating the project and image before runtime assessment, observing the application within a bounded environment, and returning evidence that refers back to the same release candidate.

For developers, that can make security feedback easier to relate to the change under review. For security teams, it can provide clearer context around findings and assessment scope. For engineering leaders, it can create a more repeatable process for understanding how AI assisted changes were validated before release.

Cortex Cloud began by giving eligible AI work a durable managed place to run. The new security workflow makes that environment more useful for the next question enterprises are asking as AI coding agents take on more responsibility: not only whether the work completed, but what evidence shows that the resulting software was built, tested and assessed under known

conditions.

That is the next step in the Cortex Enterprise AI Control Layer: connected intent, managed execution and security evidence that follows the work from source to runtime.

Pervaziv AI

Pervaziv AI

[email us here](#)

Visit us on social media:

[LinkedIn](#)

[Instagram](#)

[YouTube](#)

[X](#)

[Other](#)

This press release can be viewed online at: <https://www.einpresswire.com/article/944355056>

EIN Presswire's priority is source transparency. We do not allow opaque clients, and our editors try to be careful about weeding out false and misleading content. As a user, if you see something we have missed, please do bring it to our attention. Your help is welcome. EIN Presswire, Everyone's Internet News Presswire™, tries to define some of the boundaries that are reasonable in today's world. Please see our Editorial Guidelines for more information.

© 1995-2026 Newsmatics Inc. All Right Reserved.