GitHub Advanced Security (GHAS) on Azure DevOps Feature Matrix

This document helps answer the question **"Is this GitHub security feature available in GitHub Advanced Security for Azure DevOps?"**

The sections, feature names, and ordering follow the GHAS-on-GHES feature matrix as closely as possible. Core capabilities and relevant cloud-only features are included where they help explain Azure DevOps support.

Last reviewed: November 11, 2026. Scope: **Azure DevOps Services / Azure Repos Git**, not Azure DevOps Server or TFVC. Support is based exclusively on public product documentation and published release notes.

Contents

- How do I read this document?
- Release notes
- Secret Protection
- Code Security
  - Code scanning
  - Copilot Autofix and AI security
  - Dependabot alerts
  - Dependabot updates
  - Dependency graph, dependency review and snapshot submission API
- Security Overview
- Administration
- Dependencies

How do I read this document?

Each row represents a feature. There is only one support column: **Azure DevOps**.

Cell	Meaning
✅	Generally available.
☑️	Public preview, including limited public preview.
🔒	Private preview.

Cell	Meaning
Empty	No documented native Azure DevOps support established for this feature.

Empty cells are intentionally left blank, including features available on GitHub but not documented on Azure DevOps. A blank is not a prediction or proof that a capability is technically impossible. In particular, version-dependent CodeQL CLI functionality is not automatically equivalent to a supported Azure DevOps product integration.

Every **Feature** link points to public Azure DevOps documentation. For a feature without documented ADO support, the link points to the closest relevant ADO capability or API documentation; it does **not** imply that the missing feature is described or supported there. Notes distinguish native equivalents, narrower implementations, and unverified details.

The dependency-scanning rows compare equivalent security outcomes: Azure DevOps calls its capability **dependency scanning**, not Dependabot. Third-party extensions and customer-built automation do not count as native support.

Release notes

- CodeQL default setup GA: Sprint 276.
- PR security status checks GA: Sprint 275.
- Dependency default setup and work-item linking GA: Sprint 264.
- Copilot Autofix limited public preview: Sprint 281.

Secret Protection

Secret scanning identifies credentials in repository content and history. Push protection blocks supported secrets before they enter the repository.

- Secret scanning documentation.
- Supported secret patterns.
- Advanced Security REST API documentation.

Feature	Azure DevOps	Notes
Partner patterns / supported secret types	✅	Use the ADO pattern table; detection, push protection, and validity-check coverage differ by secret type.
Non-provider patterns	✅	Available through the **Other** confidence filter.
User-space secret scanning		No personal-repository/user-space counterpart established; ADO repositories belong to projects.

Feature	Azure DevOps	Notes
Push protection bypass	✅	Contributor bypass uses `skip-secret-scanning:true` in the commit message; an alert is still generated. This is not delegated approval.
Detect secrets in discussions and PR titles, bodies, and comments		Documented scanning scope is repository source content/history.
Push protection on web-interface commits	✅	Explicitly documented for the Azure Repos web interface.
Push protection on file uploads		Web-commit protection is documented; a separate file-upload coverage guarantee was not established.
Audit log events for non-provider patterns		
Create a push protection bypass through REST API		
Organization-level enablement of non-provider patterns		Detection exists; a separately configurable organization-wide non-provider-pattern toggle was not established.
REST API for secret scanning scan history		
Delegated alert dismissal for secret scanning		Dismissal permissions exist; a request/reviewer/approval queue was not established.
Secret scanning completion events through audit log and webhooks		Validity-status service hooks are documented, but do not establish scan-completion event parity.
Base64-encoded GitHub token detection		No explicit encoded-token support statement established.
Push protection bypass request management through REST API		
Free secret risk assessment for organization/security administrators		
Configurable push protection patterns		No documented selection of which provider patterns receive push protection.
Secret scanning in wikis		No explicit Azure DevOps wiki coverage established.
Secret scanning REST fields: first location detected and additional locations		ADO APIs have their own schema; GitHub field-level compatibility was not established.
Secret scanning validity checks	✅	Automatic for supported types; also supports on-demand verification and validation-status filtering.
Secret scanning alert assignment		Azure Boards work-item linking is separate from a native alert-assignee field.
Enterprise-level push protection delegated bypass		
Push protection exemptions for apps, teams, and roles		

Feature	Azure DevOps	Notes
Sort bypass and alert-dismissal approval requests by date		
Custom validity states for custom-pattern alerts through REST API		
Repository-level push protection exemptions for trusted actors		
Secret scanning across full Git history and all branches	✅	A unique credential is consolidated into one alert across its occurrences.
Secret push protection for Git pushes	✅	Native Azure Repos Git enforcement for supported high-confidence patterns.
Custom secret regex patterns		No documented custom-pattern configuration.
Dry runs and push protection for custom secret patterns		
AI detection of unstructured secrets / generic passwords		Non-provider structured patterns do not establish AI password detection.
Public monitoring for leaked enterprise credentials		No ADO-native equivalent established.

Code Security

Code scanning

Code scanning uses CodeQL to identify vulnerabilities. Azure DevOps also accepts findings from third-party tools.

- Code scanning documentation.
- Initialize CodeQL task reference.
- Publish third-party SARIF results.

Feature	Azure DevOps	Notes
CodeQL toolcache / version management	✅	Agent/task-managed rather than a fixed appliance-bundled version; automatic installation can obtain the latest CodeQL version.
Kotlin language support	✅	Uses the Java/Kotlin analyzer (`java`); requires the appropriate build configuration.
Swift language support	✅	Advanced setup; Swift is not supported by ADO default setup.
CodeQL threat models for detecting relevant alerts		Not separately established in the reviewed ADO configuration documentation.
Code scanning merge protection	✅	ADO branch-policy status checks, not GitHub rulesets; new/all high-and-critical checks.

Feature	Azure DevOps	Notes
CodeQL threat-model configuration in default setup		
CodeQL model packs for organizations/repositories		No matching managed organization/default-setup integration established.
CodeQL scans Java projects without a build	✅	Default setup uses `none` build mode; advanced setup also exposes build-mode configuration.
CodeQL scans C# projects without a build	✅	
CodeQL dependency caching for advanced setup		No ADO-specific equivalent of the managed caching integration established.
Zstandard archives for CodeQL tools		Archive support is CLI/version-dependent, not a separately documented ADO feature.
Delegated alert dismissal for code scanning		ADO permission-based dismissal is not approval-based dismissal.
Code scanning alert-related audit log events		Exact event coverage was not established.
Development section to track alert resolution progress		Work-item linking exists, but an equivalent alert development/progress section was not established.
CodeQL scanning of GitHub Actions workflows		The ADO language/task documentation does not establish this integration.
Rust language support	✅	Listed in default setup and the current Initialize CodeQL task reference.
CodeQL scans C/C++ projects without a build	✅	C/C++ is included in ADO default setup, which uses `none` build mode.
Code scanning alert assignment		Native assignment to a human owner was not established.
Periodic code scanning of inactive repositories	✅	Scheduled default-branch scans provide this outcome; no new commit is required to trigger the weekly schedule.
Incremental CodeQL analysis for pull requests		No ADO integration statement established; do not infer support from CLI releases alone.
Code scanning alerts linked to work-management items	✅	Azure Boards work items rather than GitHub Issues. Programmatic linking is also documented in Sprint 268.
CodeQL data extensions for custom barriers and sanitizer models		CLI/query-pack-version dependent; not separately established in ADO documentation.
CodeQL analysis of composite action metadata files		No ADO-specific support statement established.
CodeQL static analysis and repository alerts	✅	
Licensed standalone CodeQL CLI analysis	✅	For repositories covered by a paid GHAzDO license; publish SARIF to ADO.

Feature	Azure DevOps	Notes
Third-party code scanning / SARIF ingestion	✅	
Custom queries, query suites, and configuration files	✅	Advanced setup exposes query-suite and configuration-file inputs.
Automatic CodeQL installation on self-hosted agents	✅	`enableAutomaticCodeQLInstall`.
CodeQL default setup without YAML changes	✅	Automatic language detection and an initial scan when enabled at organization/project level.
PR-triggered CodeQL default setup		Documented default setup scans the default branch only. Use advanced setup/build validation for PR analysis.
Go and Swift in CodeQL default setup		Both require advanced setup on ADO.
PR annotations for code scanning	✅	Requires configured build validation and appropriate baseline scans.
Configurable severity thresholds and tool-specific merge rules		ADO's built-in checks cover new/all high-and-critical alerts, not an arbitrary matching threshold/ruleset configuration.
Configurable default-setup agent pool and scan schedule	✅	Organization settings support Microsoft-hosted, self-hosted, and Managed DevOps Pools; weekly schedule selection.

Copilot Autofix and AI security

These additional rows capture relevant capabilities beyond the GHES matrix's code-scanning release list.

Preview access caveat: the current how-to labels Autofix public preview; Sprint 281 calls it **limited public preview** and states that additional participants are not being accepted. Treat the feature as public preview with restricted/uncertain enrollment, not GA.

Feature	Azure DevOps	Notes
Copilot Autofix for code-scanning alerts	☑️	Alert-driven remediation: the coding agent explores repository context, refines changes, and opens a fix PR. Preview-access caveat above applies.
Autofix for third-party code-scanning alerts	☑️	Supported alongside CodeQL alerts.
Autofix for dependency vulnerabilities	☑️	Not deterministic scheduled Dependabot security/version updates.
Repository custom instructions and remediation skills	☑️	Documented `.github` and `.agents` instructions/skill locations.
Automatic inline Autofix suggestions on existing PR alerts		Documented alert-driven fix-PR generation does not establish automatic inline PR suggestions.

Feature	Azure DevOps	Notes
Multiple alerts fixed in a single Autofix PR		
Custom Autofix agent pools		Current Autofix prerequisites specify the Microsoft-hosted **Azure Pipelines** pool. This differs from configurable CodeQL scan pools.
AI repository scanning beyond CodeQL-supported languages		
AI application/agent scanning for prompt injection		

Dependabot alerts

The native ADO equivalent for vulnerability detection is **dependency scanning**. It finds vulnerable direct and transitive dependencies using GitHub-reviewed advisories. This does not imply the availability of the Dependabot service or all its workflows.

- Dependency scanning documentation.
- Supported package ecosystems.

Feature	Azure DevOps	Notes
Vulnerable function-call / reachability information		Dependency paths and root dependencies are documented; vulnerable function-call reachability was not established.
Dependabot auto-triage rules		No native rule-based auto-triage equivalent established.
CVSS 4.0 schema for advisory scoring		CVSS-based severity is documented; explicit ADO CVSS 4.0 presentation/parity was not established.
EPSS information for vulnerability prioritization	✅	ADO displays **EPSS percentile**, not an identical advisory-score UI.
EPSS filtering in dependency alerts	✅	Percentile range filter; metadata is not available for every advisory.
Dependabot-specific metrics page		ADO Risk/Coverage/Alerts views exist; an equivalent dedicated metrics page was not established.
Dependency vulnerability alert assignment		Work-item linking is separate from native alert assignment.
Delegated alert dismissal for dependency alerts		
Alerts for known malicious package versions	🔒	Current ADO documentation labels malware findings **private preview**.
Vulnerability alerts for direct and transitive dependencies	✅	Uses GitHub-reviewed advisories from the GitHub Advisory Database.

Feature	Azure DevOps	Notes
Root-dependency and dependency-path remediation context	✅	Helps locate the direct dependency responsible for transitive exposure.
Development/runtime dependency scope	✅	Scope badges and filters when the result includes scope metadata.
Branch, pipeline, package, and severity filters	✅	The alerts hub does not display scans completed on PR branches.
Automatic closure when a vulnerable component is removed/fixed	✅	Based on latest relevant pipeline results.
Dependency-scanning default setup	✅	Injects the scanning task into suitable pipelines; a pipeline must still run.
Pipeline-independent dependency scanning		The documented ADO setup requires pipeline execution.
PR dependency-vulnerability annotations and merge checks	✅	Requires build validation and baseline scanning; high/critical checks do not constitute full dependency review.

Dependabot updates

No native deterministic Dependabot security/version-update service was established in the reviewed ADO product documentation. The rows below deliberately retain the feature granularity of the GHES matrix, with empty support cells. **Agentic dependency Autofix is a separate public-preview capability**, listed above.

Feature links point to ADO dependency-remediation documentation for context, not to documentation claiming these Dependabot features exist on ADO.

Feature	Azure DevOps	Notes
Dependabot security-update PRs		
Scheduled Dependabot version-update PRs		
Dependabot support for devcontainer.json		
Dependabot access to Cargo private registries		
Pause scheduled update jobs after repeated failures		
Grouped security updates		
Private-registry support for target-branch configuration		
Updates for pnpm workspace catalogs		
Updates for .NET SDK global.json		

Feature	Azure DevOps	Notes
Version updates for Bun dependencies		
Version updates for Docker Compose dependencies		
Version updates for uv dependencies		
Organization-wide Dependabot access grants		
Version updates for Helm dependencies		
Custom update frequencies using cron expressions		
Gradle lockfile updates		
Minimum package-age configuration		
Cross-ecosystem grouped update PRs		
Central management of private-registry configurations		
Version updates for vcpkg and Rust toolchains		
Updates for private Go modules		
Version updates for Conda dependencies		
OIDC authentication for private registries		
Security updates for uv / Python		
Version updates for OpenTofu		
Version updates for Bazel		
Version updates for Julia		
Version updates for pre-commit hooks		
Organization-level OIDC for Dependabot/code-scanning registries		
Multiple private registries per ecosystem		
Version updates for Deno		

Feature	Azure DevOps	Notes
Version updates for sbt		
Version/security updates for Poetry v2		
Cross-organization access to internal/private repositories		
Version updates for the Nix ecosystem		
Swift package updates in Xcode projects		

Dependency graph, dependency review and snapshot submission API

ADO dependency scanning stores component snapshots and dependency relationships. That underlying data is **not evidence of a complete dependency-inventory UI, dependency-review API, or SBOM-export feature**.

Feature	Azure DevOps	Notes
Package URLs in generated SBOMs		Native SBOM export and package-URL coverage were not established.
Component snapshots and dependency relationships	✅	Updated by dependency-scanning pipeline runs.
Repository dependency-inventory UI		A vulnerability alert list is not a complete dependency inventory.
Dependency graph REST API		Alert APIs do not establish a dependency-graph API equivalent.
Dependency snapshot submission API		No GitHub-style general-purpose snapshot-submission API established.
Native SBOM export through UI/API		Separate pipeline-based SBOM tools do not count as native export.
Dependency review showing added, removed, and changed PR dependencies		PR vulnerability annotations are available, but a full dependency-review diff was not established.
Merge blocking for new high/critical dependency vulnerabilities	✅	ADO branch-policy status checks provide this narrower prevention outcome.
Dependency-review license allow/deny policies		
Enterprise-wide open-source license-policy management		

Security Overview

Security overview summarizes security risk and scanning coverage across projects and repositories **within an Azure DevOps organization**.

- Security Overview documentation.

Feature	Azure DevOps	Notes
Historical enablement-trends dashboard		Coverage shows enablement/scanning status; historical enablement-trend parity was not established.
Enterprise-level secret-scanning metrics and enablement trends		No multi-organization enterprise equivalent established.
Security Overview group by tool		Tool filtering is documented; an equivalent group-by-tool dashboard was not established.
Security Overview filter by security tool	✅	
CodeQL pull-request alerts metrics view		
SAST vulnerability summary in Security Overview	✅	Risk table filtered to code scanning; not an identical GitHub-specific dashboard.
Prevention, detection, and remediation metrics dashboards		Risk counts do not establish full prevention/bypass/MTTR dashboard parity.
CSV export from the CodeQL pull-request alerts metrics view		General ADO CSV export exists, but not an established equivalent PR-metrics view.
Security Overview accessibility enhancements		No matching feature-specific release established; this is not a statement that ADO is inaccessible.
Filter bar for delegated bypass requests		
Overview includes PR-analysis CodeQL alerts on non-default branches		Documented Risk counts use default-branch findings.
Organization-wide Risk, Coverage, and Alerts views	✅	Cross-project/repository views in organization settings.
Organization-level Risk and Coverage rollup metrics	🔒	Explicitly labeled private preview in current documentation.
Cross-repository alert search and tool-specific filters	✅	
CSV export of risk, coverage, and alerts	✅	Alerts export is limited to the first **1,000** filtered alerts.
Shareable filtered remediation views / security campaigns	✅	ADO calls shared URL-filtered alert views **security campaigns**.
Managed campaigns with owners, notifications, and progress workflow		Shared filtered views do not establish the managed campaign lifecycle available on GitHub.

Feature	Azure DevOps	Notes
Campaign-level bulk remediation with Copilot		

Administration

ADO supports organization/project/repository enablement and its own permissions and APIs. These should not be equated automatically with GitHub security configurations or GitHub-specific roles.

Feature	Azure DevOps	Notes
Enable security features at scale	✅	Organization/project bulk enablement; not a reusable named security-configuration object.
Enterprise-level security configurations		No multi-organization reusable configuration/enforcement equivalent established.
Security configurations for archived repositories		
Advanced repository selection by security-feature status		Coverage filters exist, but matching configuration-assignment selection was not established.
Security manager role assigned directly to individual users		ADO has assignable permissions; no identical GitHub security-manager role established.
Standalone Code Security and Secret Protection products	✅	Bundled arrangements also remain documented.
"Not set" option in reusable security configurations		
Security configurations selecting default/advanced CodeQL setup		Both setup modes exist; a reusable security-configuration object was not established.
Enterprise Security Manager role		
Code scanning default setup overriding Actions policies		GitHub Actions policy is not an ADO control. ADO has its own pipeline/task prerequisites.
Audit log events for Dependabot configuration changes		
Security configurations setup wizard		No corresponding named configuration workflow established.
Enterprise-wide disablement for private/internal repositories		ADO enablement is organization/project/repository scoped, not a multi-organization enterprise control.
Automatic enablement for future repositories/projects	✅	

Feature	Azure DevOps	Notes
Granular alert/settings permissions	✅	View alerts, manage/dismiss alerts, and manage settings.
Advanced Security alert and enablement REST APIs	✅	ADO-specific endpoints and schemas, not GitHub API compatibility.
Programmatic alert-to-work-item linking	✅	
Secret validity-status service hooks	✅	This does not imply identical coverage for every GitHub webhook/audit event.

Dependencies

The following requirements apply to the ADO capabilities that are documented as available or in preview.

Feature	License required?	Azure Pipelines / agent capacity required?	Notes
Security Overview	No, to access the overview	No	Available in organization settings even without enabled repositories; findings/coverage depend on the enabled products.
Secret scanning	Secret Protection or GHAS bundle	No	Native repository/history scanning.
Secret push protection	Secret Protection or GHAS bundle	No	Native Git/web-commit checks for supported patterns.
Validity checks	Secret Protection or GHAS bundle	No	Automatically included for supported types.
Dependency scanning	Code Security or GHAS bundle	Yes	Default setup injects a task into suitable pipelines; a pipeline must execute.
CodeQL default setup	Code Security or GHAS bundle	Yes, agent capacity	No YAML changes; managed scheduled scans. Pool/schedule configuration is separate from Autofix execution.
CodeQL advanced setup	Code Security or GHAS bundle	Yes for documented pipeline tasks	Standalone licensed CLI analysis with SARIF publishing is also supported.

Feature	License required?	Azure Pipelines / agent capacity required?	Notes
Third-party SARIF findings	Code Security or GHAS bundle	Yes for the documented Publish task	Scanner execution and third-party licensing are separate.
PR security annotations/status checks	Relevant security product	Yes	Requires configured scanning/build validation and baseline scans of relevant branches.
Copilot Autofix	GHAS / Code Security entitlement; preview access	Yes, Microsoft-hosted Azure Pipelines pool	Requires a parallel job. Fix generation consumes separately Azure-billed AI credits; GitHub Copilot budgets do not govern this meter.