

Wiz IaC Scanning

Last updated by | Timi Pere | Oct 6, 2025 at 4:12 PM CDT

Wiz IaC scanning analyzes your Terraform, CloudFormation, Kubernetes manifests, and other IaC templates to detect:

- Misconfigurations
- Secrets exposure
- Vulnerabilities
- Compliance violations

Wiz Code IaC Scanning & CI/CD Integration

Part 1: Setup Wiz Code in Visual Studio Code

1. Install Wiz Code Extension

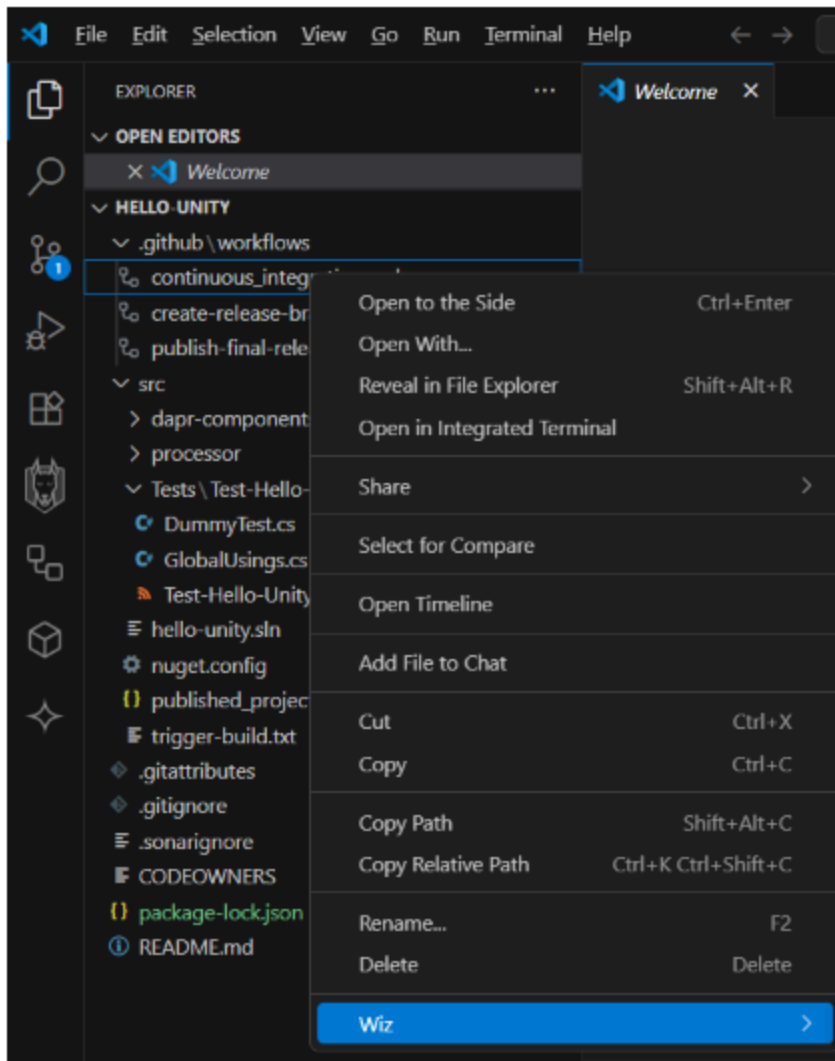
- Open **Visual Studio Code**.
- Go to the **Extensions panel** (`Ctrl+Shift+X`).
- Search for “**Wiz Code**” and install the extension by WizCloud.

2. Authenticate Wiz Code

- Open the **Command Palette** (`Ctrl+Shift+P`).
- Type and select `wiz: Authenticate` .
- A browser window will open — log in with your **Wiz credentials**.
- Once authenticated, your IDE is linked to your Wiz account.

Part 2: Scan IaC Files or Folders

1. Right-Click to Scan



- In the **Explorer panel**, right-click on a file or folder.
- Select **“Scan with Wiz”**.
- Wiz will analyze the code for:
 - Misconfigurations
 - Secrets
 - Vulnerabilities
 - Sensitive data

2. View Results

- Findings will appear in the **Wiz Findings Panel**.

- You can click on each issue to view details and suggested fixes.

The image shows the Wiz IaC Scanning interface. On the left is a sidebar with icons for file explorer, search, graph, alerts (1), image scan, policies, and a cube. The main panel is titled 'WIZ' and shows a list of findings. The first finding is 'IAM password policy should be enabled for account line 1' from the file '.github/workflows/publish-final-release.yml'. This finding is highlighted with a blue bar. Below it, the 'FINDING DETAILS' section is expanded, showing the finding title, severity (Low), filename, expected state, found state, remediation instructions, and ignored policies. A red circle highlights the star icon in the sidebar, and a red line underlines the 'FINDING DETAILS' header.

WIZ

CODE SECURITY FINDINGS

.github/workflows/publish-final-release.yml 1

IAM password policy should be enabled for account line 1

IMAGE SECURITY FINDINGS

No image security findings to display.

Verify scan policies & extension settings, then run a Wiz image scan to populate findings.

FINDING DETAILS

IAM password policy should be enabled for account

Low

Filename

c:\Users\10396379\OneDrive - BD\snyk-import\hello-unity\.github\workflows\publish-final-release.yml:1 [CLOUD_FORMATION]

Expected

IAM Account Password Policy should be defined with strong password requirements

Found

IAM Account Password Policy is not defined

Remediation Instructions

None

Ignored Policies

- Default IaC policy - Finding excluded by policy configuration

Part 3: CI/CD Integration with GitHub Actions







1. Install Wiz CLI (for pipeline use)

- For Windows: use the cmd: `npm install -g wiz-cli`
- For macOS with Homebrew installed, use the cmd: `brew install --cask wizcli`
- Add it to your system path.

Reference doc - [Scan Code in VS Code](#) | [Wiz Docs](#)

2. Create Wiz Service Account

- Go to Wiz Console → **Settings > Access Management > Service Accounts**.
- Since there is an existing integrations as below, there is no need to create a new service account:
 - DevOps_IaC_Scan (Azure DevOps CI)
 - Terraform_IaC_Scan (Custom Integration)

 DevOps - IaC - Scan_a89b176e-dc30... Integration	 DevOps - IaC - Scan Azure DevOps CI	TGS create:security_scans, ...	Active
 stspowerbi Custom Integration (GraphQL API)	-	STS read:endpoint_attack_...	Active
 Terraform_IaC_Scan Custom Integration (GraphQL API)	-	All Projects read:security_scans, cr...	Active
 Wiz_567e3261-dde6-4f16-b7ae-6d5... Integration (internal)	 Wiz Automated Platform Action	All Projects read:all, write:issue_sta...	Active

These integrations already have the necessary permissions like `create:security_scans`, which is exactly what's needed for IaC scanning via Wiz CLI or Wiz Code. These integration can be referenced in the CI/CD pipeline or Wiz CLI config.

3. Add GitHub Actions Workflow

Create `.github/workflows/wiz-scan.yml` in your repo:



```

name: Wiz IaC Scan

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  wiz-scan:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Install Wiz CLI
        run: |
          curl -sSL https://wiz.io/install.sh | bash

      - name: Run Wiz IaC Scan
        run: |
          wizcli iac scan --path ./ --output scan-results.json

      - name: Upload Scan Results
        uses: actions/upload-artifact@v3
        with:
          name: wiz-scan-results
          path: scan-results.json

```

4. Optional: Fail Build on Critical Issues

Add a step to fail the pipeline if critical issues are found:

```

- name: Fail on Critical Findings
  run: |
    jq '.findings[] | select(.severity == "CRITICAL")' scan-results.json && exit 1 || exit 0

```



Reference: [Scan IaC files with Wiz CLI](#) | [Wiz Docs](#)

Part 4: Enable CSPM in Wiz

1. Connect Cloud Accounts

- Go to Wiz Console → **Settings > Cloud Integrations**.
- Choose Azure or AWS.
- Use agentless API-based integration.

2. Enable Configuration Rules

- Customize or use built-in rules (CIS, PCI-DSS, etc.).
- Use **Rego** for custom policies.

3. Enable Compliance Monitoring

- Go to **Compliance > Frameworks**.
- Select and schedule scans for SOC2, HIPAA, etc.

4. Use Security Graph

- Visualize attack paths and prioritize risks.

Part 5: Wiz Policies to Configure

- Terraform Misconfiguration Detection
- Secrets in Code
- Cloud Resource Exposure
- RBAC Misconfigurations
- Compliance Frameworks

Reference: [Policies](#) | [Wiz Docs](#)