Qualys IaC Security Integration with GitHub

In the current continuous integration and continuous deployment (CICD) environment, the security scans are conducted on cloud resources after deployment. As a result, you secure your cloud resources post deployment to respective Cloud accounts.

With an introduction of Infrastructure as Code (IaC) security feature by Qualys CloudView, you can now secure your IaC templates before the cloud resources are deployed in your cloud environments. The IaC Security feature will help you shifting cloud security and compliance posture to the left, allowing evaluation of cloud resource for misconfigurations much early during development phase.

CloudView offers an integration with GitHub to secure Git repositories using a GitHub actions, that can be used to scan your IaC templates from GitHub repositories. It continuously verifies security misconfigurations against CloudView security controls and displays the misconfigurations for each run. You have a continuous visibility of security posture of your IaC Templates at GitHub repositories and plan for remediation. Follow this guide for more details.

For supported templates, other integrations, and features of Cloud IaC Security, refer to CloudView User Guide and CloudView API User Guide.

Scanning IaC Templates at GitHub

The GitHub integration allows you to perform IaC scans at the GitHub repositories on the pull and push requests. We provide you with a GitHub actions template and options that can be configured to run based on various triggers.

You can perform IaC scan on either of the following:

- the entire repository for the branch where the manual/scheduled event was performed.

- the templates that were newly added to the branch.

The results are generated within GitHub that provide you with proactive visibility into the Cloud security by scanning the templates residing in GitHub repositories.

Let us see the quick workflow:

Pre-requisite

Configure Environment Variables

Configure GitHub Actions

Trigger Scan

Understanding Scan Output

Pre-requisite

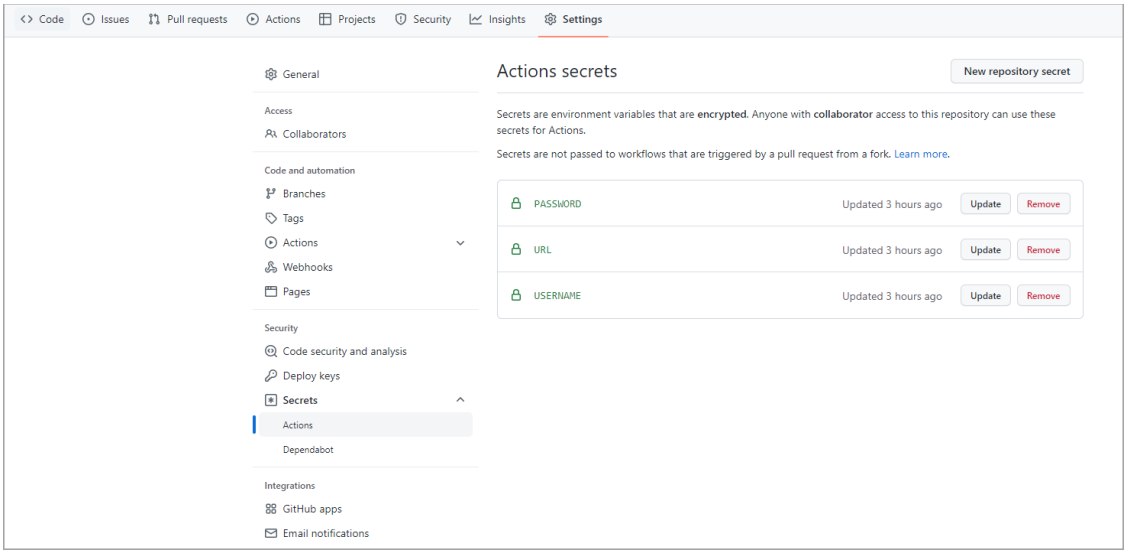
Ensure that you have valid subscription of Qualys CloudView (Cloud Security Assessment) app.

Before you trigger IaC scans in GitHub, ensure that you configure environment variables that are used in the actions.

Self-hosted runners must use a Linux operating system and have Docker installed to run this action.

Configure Environment Variables

On GitHub console, go to your organization > **Setting** > **Secrets** > **Actions**. Provide the required details for actions secrets.



Variable	Description
URL	Qualys platform URL. To know about your Qualys platform URL, click here.
USERNAME	Qualys username
PASSWORD	Qualys password

Configure GitHub Actions

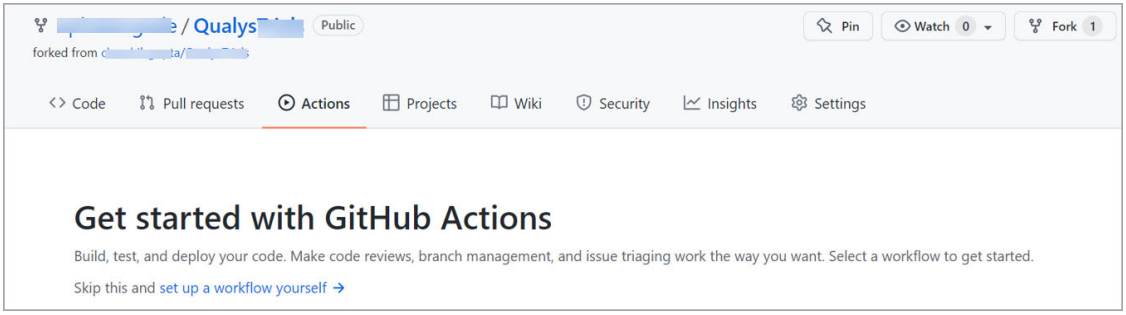
You can use the Qualys GitHub action template from GitHub marketplace to scan the repository.

It will then execute on every action such as pull request, push request, manual trigger, and scheduled job.

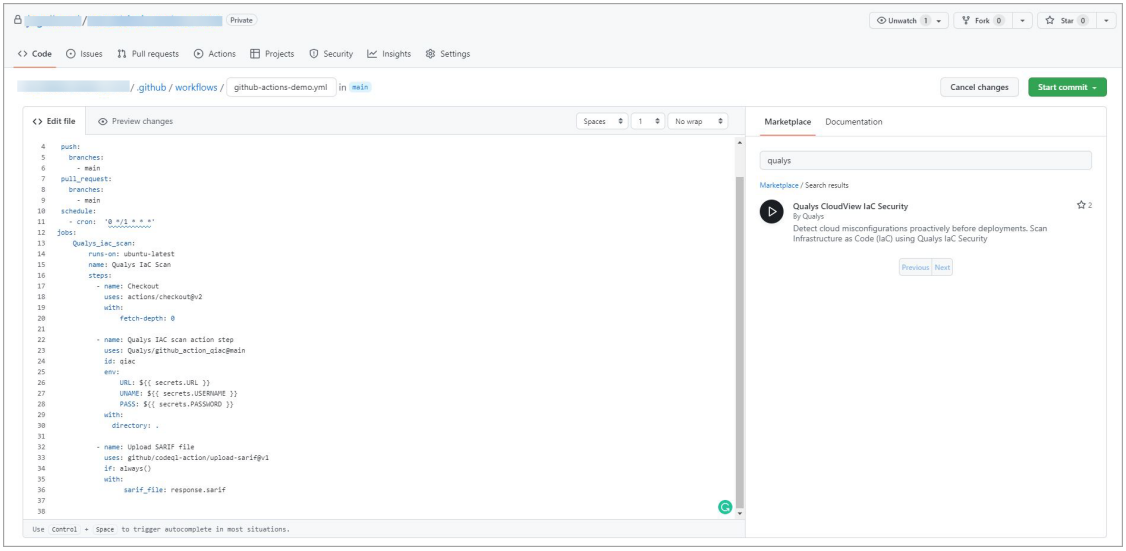
To add the Qualys GitHub action in your repository:

1. In GitHub, navigate to your repository, and click **Actions**.

2. In the **Actions** tab, click **set up a workflow yourself**.



3. In the **Marketplace**, enter qualys to search for the Qualys CloudView IaC Security template.

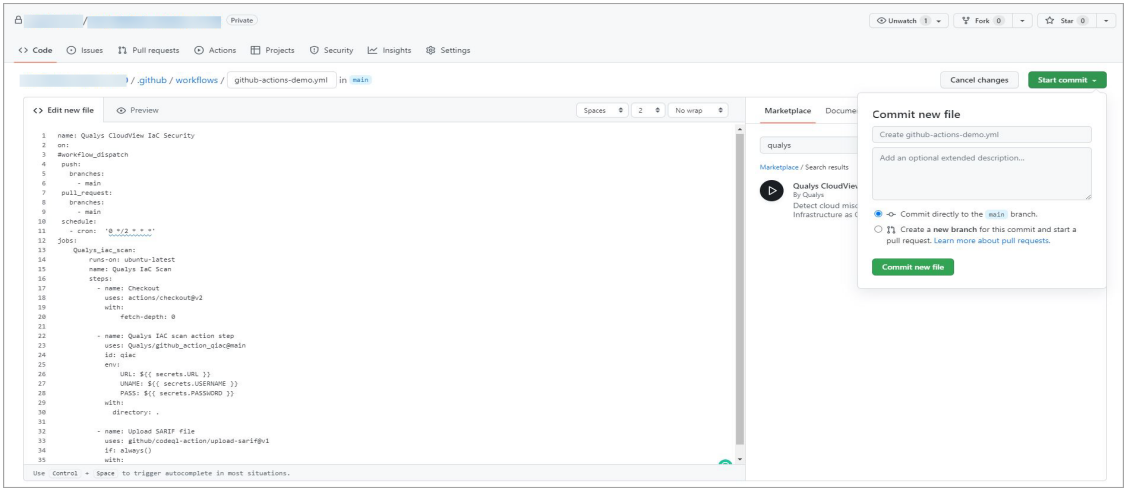


4. Click the Qualys CloudView IaC Security to view the template.

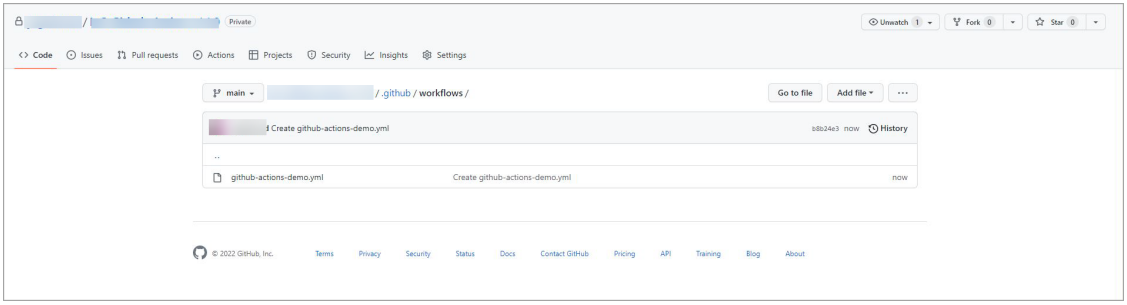
5. Copy the template and add it to the steps section in the .yml file. You can override the existing steps section or append with the contents of the template.

6. Click **Start commit**.

7. Click **Commit new file**.



The file will be committed to your repository. You can view the file in the repository, under the workflows.



Note: The GitHub actions should include the `actions/checkout` step before the scan action. Else, the scan action does not have access to the IaC files to be scanned.

Trigger Scan

Once you have configured the GitHub actions, you can trigger a scan in the following ways:

Trigger Scan (Automatically)

Trigger Scan (Manually)

Trigger Scan (Scheduled)

Trigger Scan (Automatically)

The IaC scan is automatically triggered on every pull request and push request event. Once the GitHub actions is configured, it is automatically executed, and the scan is triggered with every push request and pull request.

In case of push or pull request event, the scan scope is limited only to the changed or newly-added files.

Here is the example for a push request.

```
name: Qualys IAC Scan
on:
  push:
    branches:
      - main
jobs:
    Qualys_iac_scan:
        runs-on: ubuntu-latest
        name: Qualys IaC Scan
        steps:
          - name: Checkout
            uses: actions/checkout@v2
            with:
                fetch-depth: 0

          - name: Qualys IAC scan action step
            uses: Qualys/github_action_qiac@main
            id: qiac
            env:
                URL: ${{ secrets.URL }}
                UNAME: ${{ secrets.USERNAME }}
                PASS: ${{ secrets.PASSWORD }}
```

Here is the example for a pull request.

```
name: Qualys IAC Scan
on:
  pull_request:
    branches:
```

```
      - main
jobs:
    Qualys_iac_scan:
        runs-on: ubuntu-latest
        name: Qualys IaC Scan
        steps:
          - name: Checkout
            uses: actions/checkout@v2
            with:
                fetch-depth: 0

          - name: Qualys IAC scan action step
            uses: Qualys/github_action_qiac@main
            id: qiac
            env:
                URL: ${{ secrets.URL }}
                UNAME: ${{ secrets.USERNAME }}
                PASS: ${{ secrets.PASSWORD }}
```

Trigger Scan (Manually)

You could manually trigger a scan for the entire repository by using the following script.

```
name: Qualys IAC Scan
on: workflow_dispatch
jobs:
    Qualys_iac_scan:
        runs-on: ubuntu-latest
        name: Qualys IaC Scan
        steps:
          - name: Checkout
            uses: actions/checkout@v2
            with:
                fetch-depth: 0

          - name: Qualys IAC scan action step
            uses: Qualys/github_action_qiac@main
            id: qiac
            env:
                URL: ${{ secrets.URL }}
                UNAME: ${{ secrets.USERNAME }}
                PASS: ${{ secrets.PASSWORD }}
            with:
              directory: 'path of directory to scan (optional)'
```

If the path is provided in the directory attribute, the scan is limited to the specified directory. If the path is not provided, the entire repository will be scanned.

Trigger Scan (Scheduled)

You can schedule the IaC scans to be executed at a scheduled time on a hourly, daily, or weekly basis by using the GitHub actions. Use the cron notation to configure the schedule time.

```
name: Qualys IAC Scan
on:
  schedule:
    - cron:  '*/5 * * * *'
jobs:
    Qualys_iac_scan:
        runs-on: ubuntu-latest
        name: Qualys IaC Scan
        steps:
          - name: Checkout
            uses: actions/checkout@v2
            with:
                fetch-depth: 0

          - name: Qualys IAC scan action step
            uses: Qualys/github_action_qiac@main
            id: qiac
            env:
                URL: ${{ secrets.URL }}
                UNAME: ${{ secrets.USERNAME }}
                PASS: ${{ secrets.PASSWORD }}
            with:
              directory: 'path of directory to scan (optional)'
```

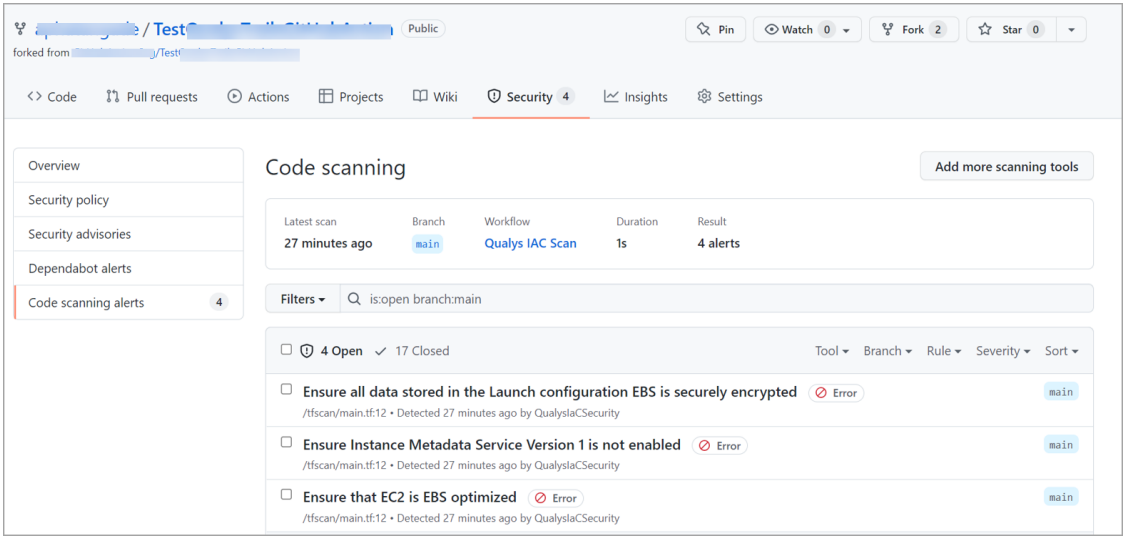
If the path is provided in the directory attribute, the scan is limited to the specified directory. If the path is not provided, the entire repository will be scanned.

Upload SARIF File on GitHub

You can upload the scan results to GitHub in a SARIF file format by using the following actions:

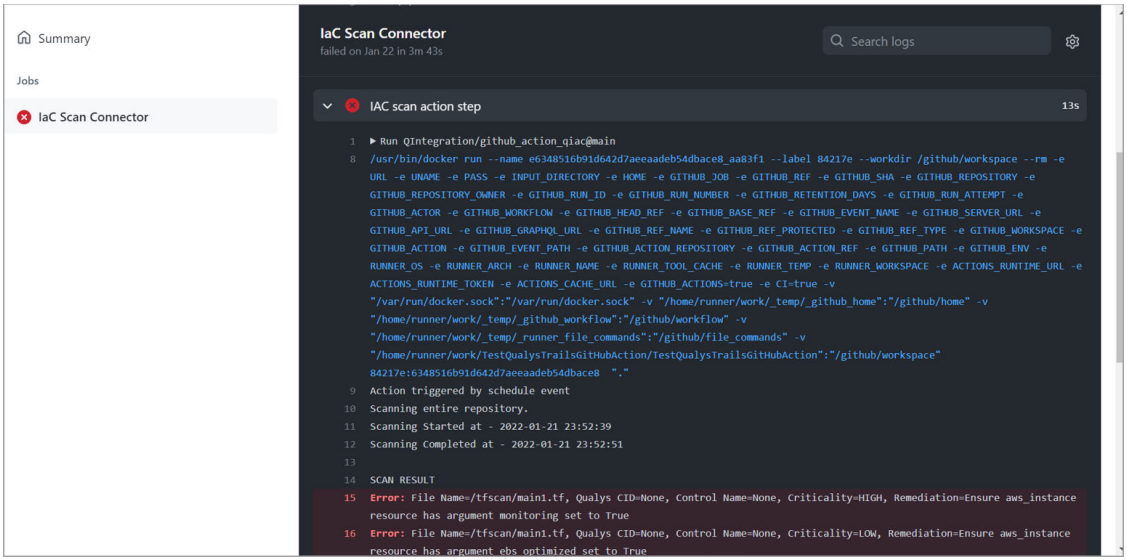
```
name: Qualys IAC Scan
on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main
  schedule:
    - cron:  '*/5 * * * *'
jobs:
    Qualys_iac_scan:
        runs-on: ubuntu-latest
        name: Qualys IaC Scan
        steps:
          - name: Checkout
            uses: actions/checkout@v2
            with:
                fetch-depth: 0

          - name: Qualys IAC scan action step
            uses: Qualys/github_action_qiac@main
            id: qiac
            env:
                URL: ${{ secrets.URL }}
                UNAME: ${{ secrets.USERNAME }}
                PASS: ${{ secrets.PASSWORD }}
            with:
              directory: 'path of directory to scan (optional)'

          - name: Upload SARIF file
            uses: github/codeql-action/upload-sarif@v1
            if: always()
            with:
                 sarif_file: response.sarif
```

The results are displayed in the **Security** tab > **Code scanning alerts**.



Understanding Scan Output

Once the IaC scan is completed, GitHub shows scan output in annotations.



For details on elements in the output format, refer to Secure IaC section in CloudView API User Guide.