Qualys IaC Security Integration with Bitbucket

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 Bitbucket to secure Git repositories using a pipeline script, that can be used to scan your IaC templates from Bitbucket repositories. It continuously verifies security misconfigurations against CloudView security controls and displays the failed checks for each run. You have a continuous visibility of security posture of your IaC Templates at Bitbucket Pipeline and plan for remediation. Follow this guide for more details.

Scanning IaC Templates at Bitbucket

The Bitbucket integration allows you to perform IaC scans at the Bitbucket repositories on the pull and push requests. We provide you with a pipeline script 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 Bitbucket pipeline output that provide you with proactive visibility into the security of your IaC templates residing in Bitbucket repositories.

Let us see the quick workflow:

Pre-requisite
Configure Environment Variables
Configure Pipeline Script
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 Bitbucket, ensure that you configure environment variables that are used in the script.
Configure Environment Variables

On Bitbucket console, go to Repository > Repository Setting > Repository Variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUALYS_URL</td>
<td>Qualys platform URL. To know about your Qualys platform URL, click here.</td>
</tr>
<tr>
<td>QUALYS_USERNAME</td>
<td>Qualys username</td>
</tr>
<tr>
<td>QUALYS_PASSWORD</td>
<td>Qualys password</td>
</tr>
</tbody>
</table>

Configure Pipeline Script

We provide you with a pipeline script that you can use in the repository. The pipeline script should copied from QIntegration/bitbucket_pipelines@main to the default pipeline (bitbucket-pipelines.yml) in your Bitbucket repository. It will then execute the script on every action such as pull request, push request, manual trigger, and scheduled job.

**Note:** You can copy the same file into the repository or just add the Qualys IaC scan Bitbucket step into the existing file (bitbucket-pipelines.yml). Add Qualys IaC scan Bitbucket step at the top of all steps.
Contents of Pipeline Script (bitbucket-pipelines.yml)

```yaml
image: qualys/qiac_security_cli

pipelines:
  custom: # defines that this can only be triggered manually or by a schedule
    qualys: # The name that is displayed in the list in the Bitbucket Cloud GUI
      - step:
          script:
            - export ScheduleBuildTrigger=true
            - sh /home/qiac/bitbucket.sh $ScheduleBuildTrigger
  default:
    - step:
        name: Qualys
        caches:
          - pip
        script:
          - export ScheduleBuildTrigger=false
          - sh /home/qiac/bitbucket.sh $ScheduleBuildTrigger

Note:
- The pipeline script runs on Qualys qiac docker image from docker hub.
- Configure ScheduleBuildTrigger to true if you want the pipeline script to trigger as per your schedule.
- Refer to the pipeline script.

Trigger Scan

Once you have configured the pipeline script, 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 action such as pull request, push request. Once the script is configured, the script is automatically executed, and the scan is triggered with every push request and pull request. With every such action, the new files that were added to the branch that performed pull operation are scanned.
**Trigger Scan (Manually)**

You could manually trigger a scan for the entire repository.

1. On Bitbucket console, go to Repository > Pipeline > Run Pipeline.

![Run Pipeline dialog box](image)

The Run Pipeline dialog box is displayed.

2. Select the branch on which you want to trigger the scan from Branch drop-down.
3. Select **custom: qualys** from the Pipeline drop-down.
4. Click **Run**.

The scan is initiated on all the files in the selecting branch of your repository. To scan all the files in the repository, select the trunk branch.
Trigger Scan (Scheduled)
You could schedule the IaC scans to be executed at a scheduled time on a hourly, daily, or weekly basis.

1. On Bitbucket console, go to Repository > Pipeline > Schedule.

The Create a schedule dialog box is displayed.
2. Select the branch on which you want to trigger the scan from Branch drop-down.
3. Select custom: qualys from the Pipeline drop-down.
4. Select the frequency at which you want the IaC scan to be automatically triggered. You could choose from Hourly, Daily or Weekly options. You could also configure the time at which the scan should be triggered.
5. Click Create.

The scan is initiated on all the files in the selecting branch of your repository. To scan all the files in the repository, select the trunk branch.

For more information on Bitbucket pipeline triggers, refer to https://support.atlassian.com/bitbucket-cloud/docs/pipeline-triggers/
Understanding Scan Output

The build fails if there is a misconfiguration in the template file. Bitbucket shows output in tabular format in the build log.

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