

Qualys VSCode Extension for IaC Security

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 as a VSCode extension 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.

The Qualys IaC Security VSCode extension empowers DevOps teams to build Infrastructure as Code (IaC) scans into their existing CI/CD processes. By integrating scans in this manner, cloud misconfigurations are detected and remediated earlier in the SDLC to catch and eliminate security flaws.

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

Pre-requisite

Ensure that you have the required subscription and permissions as stated below.

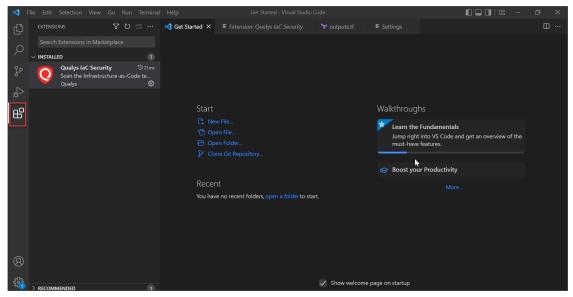
- Visual Studio Code version 1.64.0 or higher.
- · Valid subscription for Qualys CloudView (Cloud Security Assessment) app.
- Enabled API access and a role is assigned with all the necessary permissions.

Install the Qualys IaC VSCode Extension

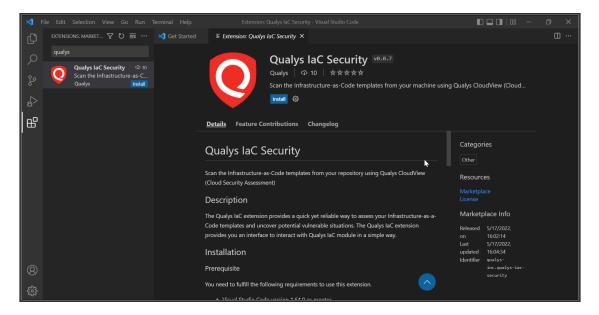
You can install the Qualys IaC VSCode extension from VSCode Extension Marketplace. Follow the below steps to set up the extension.

1. To install the extension from the VSCode marketplace, open VSCode.

2. Click the last icon on the left side of the page.



- 3. Enter Qualys in the search bar to search for all the Qualys extensions.
- 4. Click the Qualys IaC VSCode extension in the extensions list.
- 5. Click **Install** to install the extension in your VSCode. You can see the installed extension in the **Installed** tab when you navigate to **Organization Settings** > **Extension**.

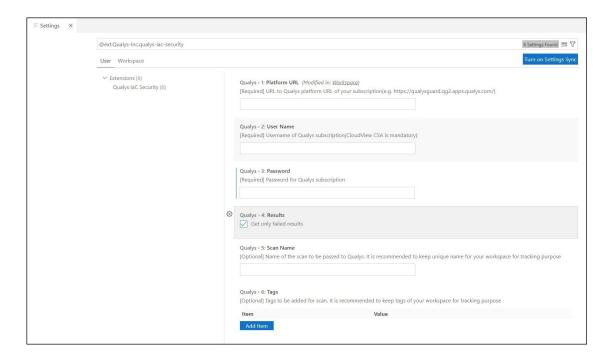


The installation is now complete.

Getting Started with the Extension

To get started with the Qualys IaC Security Visual Studio Code extension,

- 1 Open the command palette with Ctrl + Shift + P,
- 2 Type Qualys to see all the available Qualys commands. Select Qualys IaC configuration to bring up the settings page.
- 3 Fill in the input fields such as Platform URL, Username, Password, etc.



Available Commands

The following commands are available from the command palette.

- Qualys IaC Scan
 - Performs a scan on a Terraform file opened on Visual Studio Code editor
- Qualys IaC configuration
 - Adjust Qualys IaC configuration for a particular project

Run A Scan

You can use the Qualys IaC Security extension as a pre-deployment task in your project pipeline. After installing, you can see the Qualys IaC Scan option when you open the context menu.

To run a simple static scan, choose the Qualys IaC Scan option from context menu or use shortcut key (Ctrl + Shift + Q).

```
* azurerm_sql_server.tf ×
azurerm_sql_server > 😭 azurerm_sql_server.tf
  1 resource "azurerm_resource_group" "example" {
  2
       name
                 = "database-rg"
  3
       location = "West Europe"
  4
      resource "azurerm_storage_account" "example" {
                                                                                     Change All Occurrences
                                                                                                              Ctrl+F2
  8
        resource_group_name
                                 = azurerm_resource_group.example.name
                                                                                     Source Action...
        account_tier
  9
                                 = azurerm_resource_group.example.location
                                = "Standard"
 10
                                                                                     Cut
                                                                                                               Ctrl+X
        account_replication_type = "LRS"
 11
                                                                                                               Ctrl+C
                                                                                     Сору
 12
 13
                                                                                                              Ctrl+V
                                                                                     Paste
      resource "azurerm_sql_server" "example" {
 14
                                                                                     Command Palette...
                                                                                                          Ctrl+Shift+P
 15
                                     = "mssqlserver"
 16
        resource_group_name
                                    = azurerm resource group.example.name
                                                                                                          Ctrl+Shift+Q
                                                                                     Qualys IaC Scan
 17
        location
                                     = azurerm_resource_group.example.location
 18
        version
                                     = "12.0"
 19
        administrator login
                                  = "mradministrator"
        administrator login password = "thisIsDog11"
 20
 21
        # The filebase64sha256() function is available in Terraform 0.11.12 and later
        # For Terraform 0.11.11 and earlier, use the base64sha256() function and the file() function:
 22
 23
        # source_code_hash = "${base64sha256(file("lambda_function_payload.zip"))}"
        extended_auditing_policy {
 24
                                                   = azurerm storage account.example.primary blob endpoint
 25
         storage endpoint
          storage_account_access_key
                                                   = azurerm_storage_account.example.primary_access_key
 26
          storage_account_access_key_is_secondary = true
 27
 28
          retention_in_days
 29
 30
 31
        tags = {
 32
          environment = "production"
 33
 34
```

Qualys IaC Scan Result

After the scan is complete, you can view output for details on the job execution.

The **Summary** displays the details of the Terraform file that is scanned, errors (failures), scan time, and job details.

```
☆ iam_policy_document.tf ×

first > iam-policies > * iam_policy_document.tf
  data "aws_iam_policy_document" "testdoc" {
  2
       statement {
         sid = "
         actions = ["sts:assumerole", "sts:getsessiontoken"]
  4
         resources = ["arn:aws:iam:*:*:myuser"]
  7
      }
  8
      data "aws_iam_policy_document" "testdoc1" {
 10
      statement {
 11
         sid = ""
 12
          effect = "Allow"
 13
           actions - ["sz.CraataRuckat" "iam.UndataLoginDrofila"]
 1/
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
File Name : iam policy document.tf
Scan launched successfully. Scan ID: 16899371-cf08-4326-b8f9-d6e1b0be07f5
Fetching the scan status with scan ID: 16899371-cf08-4326-b8f9-d6e1b0be07f5
The scan status is: FINISHED
Result Summary
  Check Type
               Passed | Failed | Failed Stats
                                                         Skipped
                                                                   Parsing Errors
                                                                   0
  terraform
                        1
                                 high=1,low=0,medium=0
Terraform Checks
  Control Id | Control Name
                                                                    Criticality
                                                                                 Result
                                                                                          Resource
                 Ensure no IAM policies documents allow "*" as a
                                                                                  FAILED
                                                                                            aws iam policy document.testdoc3
                 statements actions
Remediation
  Control Id
               Remediation
                 Remove any "*" Statement actions form the policy
  304
                 document for the aws_iam_policy_document
```