



Quick Start Guide

Accurics SaaS Solution

www.accurics.com

Contents

- Welcome to Accurics** 3
 - Accurics protects hybrid and multi-cloud environments with the following capabilities:.....3
- How Accurics works** 4
- Prerequisites for using Accurics** 5
- Setting up the Accurics Account** 6
 - Sign up.....6
- Set up the Accurics environment** 7
- Integrate with a Cloud provider** 8
 - Integrating with AWS8
 - Integrating with Azure 11
 - Integrating with Google Cloud 12
 - Integrate with Code Repository 13
 - Step 1: Authorize Accurics to access your repositories 14
 - Step 2: Select the terraform repository 15
 - Step 3: Entering TF plan input variables as part of the configuration steps..... 15
 - Step 4: Terraform State file information 16
 - Step 5: Selecting the compliance policy 17
 - Congratulations! Your first environment is now ready to use. 18
- Initiating a cloud scan**..... 19
- Accurics CLI** 20
 - Downloading the Accurics CLI..... 20
- Installing the Accurics CLI on your local machine** 22
 - MAC 22
 - Linux..... 22
 - Windows 22
- Scan your terraform files using Accurics CLI** 23
- Viewing and Analyzing the scan results** 25
- Setting up IaC Scan for On-premises Repository** 27
 - 1. Create An On-premises Bot 27
 - 2. Download Bot Deployment File 28
 - 3. Create OAuth Application on the GitHub Enterprise Server 28
 - 4. Set up the Bot 28
 - 5. Create an Environment for On-Prem Repo 29
 - Customer Support..... 31

Welcome to Accurics

As more and more organizations move to the cloud, it is becoming increasingly important to identify changes to the cloud resources after deploying them through the Infrastructure as Code (IAC) code. Any change to a cloud resource is a potential security policy violation of the cloud security best practices. Accurics helps you identify such resource drifts and violations both in the IAC code and the resources deployed on the cloud and then helps you fix the violations.

Increase in cloud deployments creates issues with consistency. Technologies such as Docker, Terraform, Kubernetes, and OpenFaaS manage infrastructure through code and reduce manual errors, making it challenging to maintain governance across the cloud stack.

Accurics aims to protect the full cloud-native stack throughout the DevOps life cycle, from when it's defined in code through the life cycle of infrastructure employed in production. Accurics' solution scans code such as Terraform, Kubernetes YAML, Dockerfile, and OpenFaaS YAML to detect and remediate misconfigurations, policy violations, and potential breach paths before cloud infrastructure is provisioned.

It also monitors infrastructure deployed across AWS, Azure, and Google Cloud Platform to alert to production changes that could introduce security drift. You can watch a 2 min video about Accurics which is located [here](#)

Accurics protects hybrid and multi-cloud environments with the following capabilities:

Full Stack Visibility: Visualizes the real-time topology in code and cloud across a full stack, including serverless, container, platform and infrastructure technologies.

Infrastructure as Code Security: Continuously scans infrastructure code such as Terraform, Ansible, Kubernetes YAML, Dockerfile and OpenFaaS YAML for misconfigurations, vulnerabilities, policy violations, and potential breach paths before cloud infrastructure is provisioned.

Cloud Posture Management: Continuously monitors production cloud deployments for changes that introduce misconfigurations, policy violations, and potential breach paths.

Drift Detection: Continuously assesses the posture of a cloud deployment and flags any drifts from the posture defined through code.

Posture Restoration: If a drift is due to a legitimate change, code can be updated to reflect the change, and if it introduces risks, cloud can be restored to the last known secure posture.

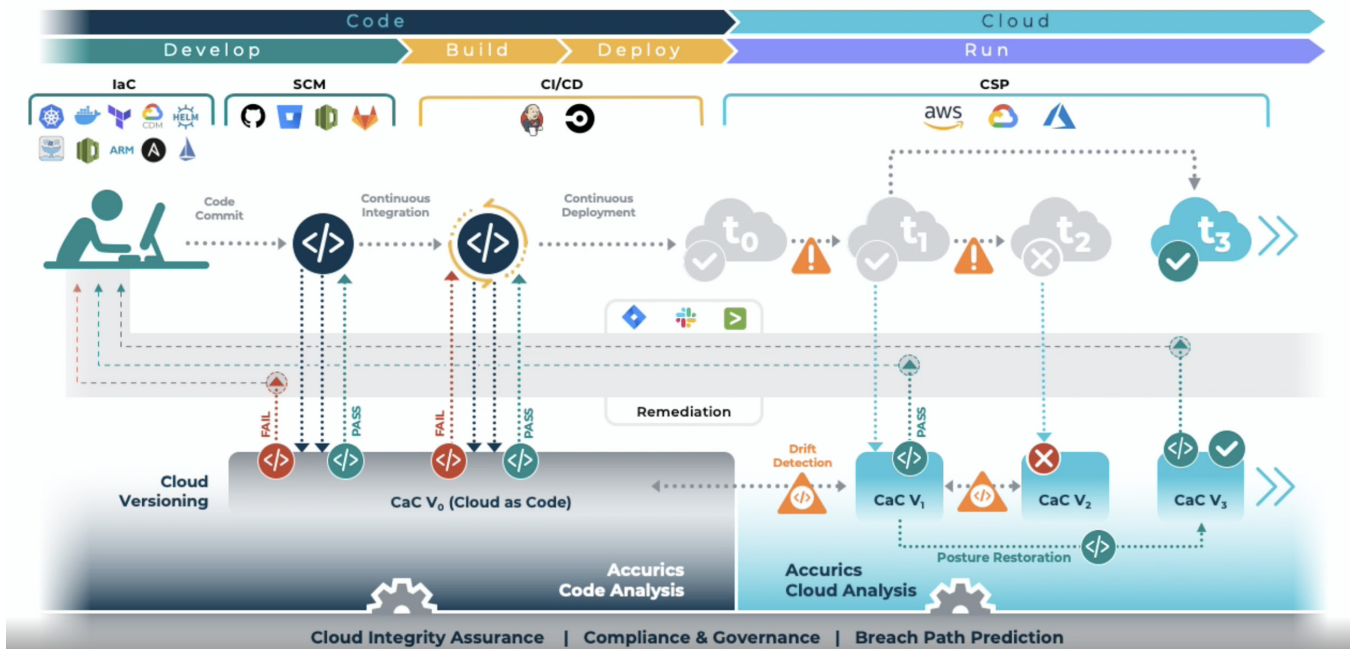
Remediation: Resolves issues via integrations with alert management mechanisms such as Slack, JIRA, Splunk, webhooks and email.

How Accurics works

Accurics seamlessly integrates into your DevOps lifecycle and scans infrastructure as code such as Terraform, Kubernetes YAML at pre-commit or post-commit level.

There are various stages where Accurics integrates into your devOps lifecycle to provide end to end cloud security, these stages are:-

- 1. Pre-Commit Stage:** at this stage, Accurics CLI can be downloaded from your Accurics tenant that can scan your Terraform code at the developer machine even before the code is checked in to a source code management system (Github, Bitbucket, Gitlab)
- 2. Post-Commit Stage:** Similar to pre-commit stage, Accurics can directly integrate into your respective SCM tools CM such as Github/Bitbucket or Gitlab, and can scan your infrastructure as code (IAC)
- 3. Build Stage / CI/CD:** Accurics provides guardrails against any unresolved misconfigurations or vulnerabilities to be pushed into your runtime environment. Accurics integrates with CI/CD tools like Jenkins/CircleCI pipelines to monitor the builds and prevent misconfigurations going into runtime by stopping the build.
- 4. Run-time/Cloud Service Provider:** Accurics scans your cloud for analyzing security risks & compliance violations without installing any agents into your runtime infrastructure. It also monitors infrastructure deployed across AWS, Azure, and GCP to alert any changes in production that could introduce cloud posture drift.



Prerequisites for using Accurics

This section explains what you need before setting up Accurics solution, and how Accurics seamlessly integrates with your devops lifecycle.

1. Access to cloud provider accounts: If you are trying to set up Accurics, you would need to access your cloud accounts. Accurics provides a command line script to create a READ-ONLY permission role. [See Step 3: Integrate with AWS Cloud provider](#)
2. Access to code repositories such as Github, Bitbucket or Gitlab: In order to perform the infrastructure as code scan, you will need to authorize Accurics to be able access the [See Step 4: Integrate with Code Repository](#)
3. Terraform code location in your repositories and the input variables that you are using to run your TF plan. These variables are part of your variable.tf file in your root directory. An example of terraform plan command is given below

```
terraform plan -var key_name=terraform-customer01 -var
public_key_path=terraform-customer01.pub

variable "key_name" {
  description = "Desired name of AWS key pair"
  default = "terraform-customer01"
}

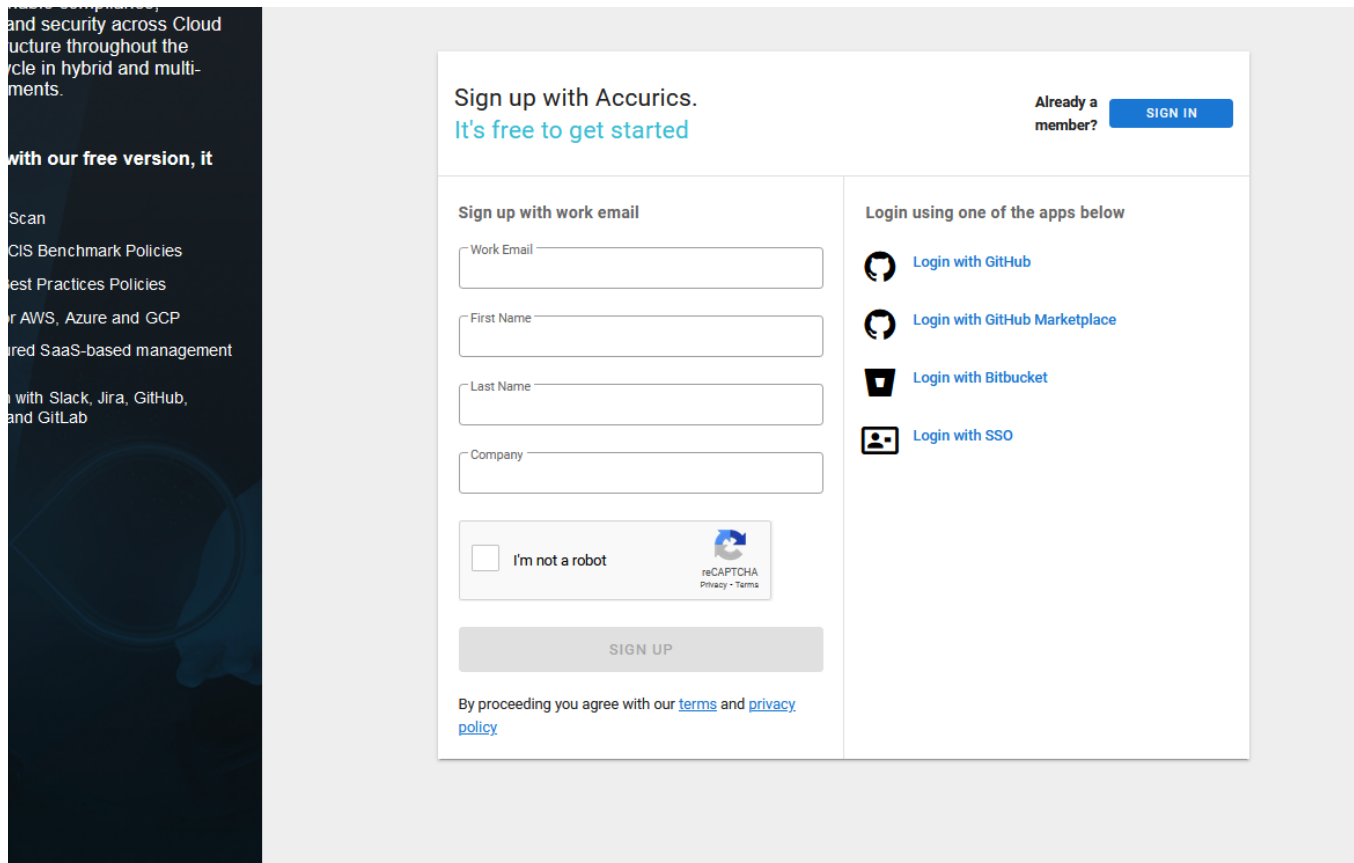
variable "aws_region" {
  description = "AWS region to launch servers."
  default = "ap-south-1"
}
```

4. Location of Terraform State file and READ ONLY access to the state files: Optionally, you can provide Accurics access to the Terraform state files. The presence of the state files ensures that the mapping of resources in the state file with the resources on the cloud is more accurate. You can add the location of the state file as part of code repository configuration setup. [See Integrate with Code Repository.](#)

Setting up the Accurics Account

Sign up

Sign up using Github, Bitbucket or a work email account at <https://app accurics.com/login>. If you are using your work email to create a login, you will receive an email with the activation link to the email provided. Clicking that link will take you to the "Set Up Password" screen. Once you have set the password, you will be logged into the product console.



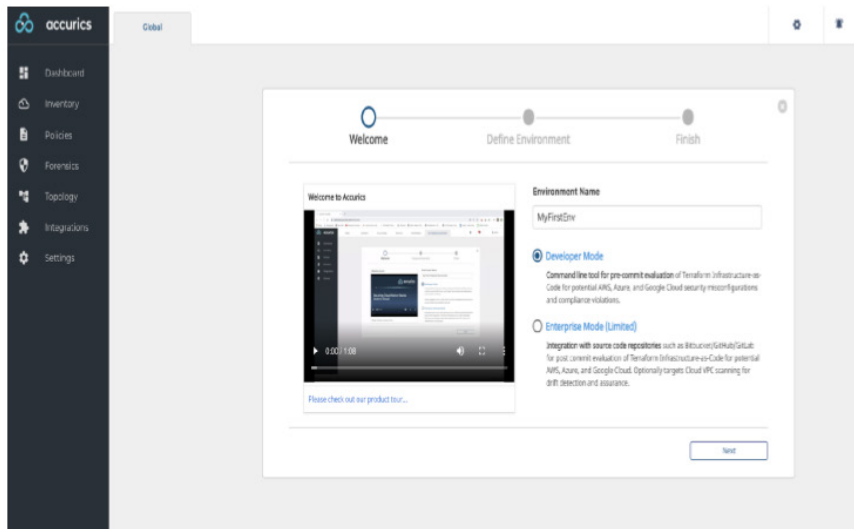
Set up the Accurics environment

Once you are logged into the console, you will be prompted to select the “Mode”. You would also need to provide the name of the environment.

Accurics is available to use in two different modes:

1. **Developer mode:** requires the use of Accurics CLI to scan IAC Code.
2. **Enterprise mode:** provides built-in integration with multiple source code repositories.

Step 2:
Please provide a name to your env and select “Enterprise Mode”



Integrate with a Cloud provider

Select Enterprise Mode to integrate Accurics with a Cloud Provider: AWS, Azure, or GCP.

Integrating with AWS

Perform the following steps to connect to your AWS account to scan the AWS resources.

1. Provide Region name of the target AWS account.
2. Access to your target AWS account

Accurics will require login credentials to access the metadata of the cloud resources and services in your target AWS account. There are two different options to authenticate

- a. **Option 1: Role ARN and external ID** When third parties such as Accurics require access to your organization’s AWS resources, you can use roles to delegate access to them.

With IAM roles, you can grant these third parties access to your AWS resources without sharing your AWS security credentials. Instead, the third party can access your AWS resources by assuming a role that you create in your AWS account.

Role ARN is a unique identifier for the IAM role that can be used to access the target AWS cloud account.

External ID is optional but it depends on how the trust policy is set up for the IAM role. To use an external ID, update a role trust policy with the external ID of your choice. Then, when someone uses the AWS CLI or AWS API to assume that role, they must provide the external ID. For more information, see [How to use an external ID when granting access to your AWS resources to a third party](#).

Configure Cloud (Optional)

Configure your Cloud for scanning to find vulnerabilities and drifts between your IaC code and Cloud.

Configure Cloud for Scan

Select a region

▼ Option 1: Role ARN and External ID

> Creating a Role ARN

Role ARN

External ID

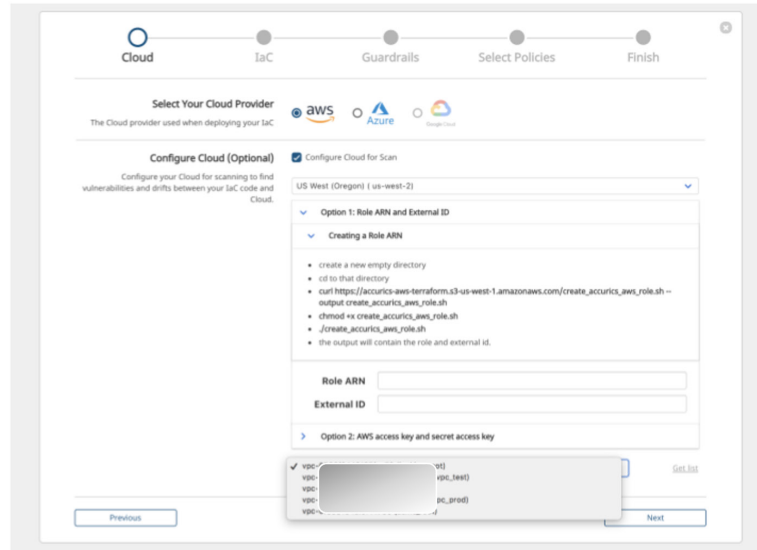
> Option 2: AWS access key and secret access key

Select a vpc
 ▼

Creating a Role ARN & External ID: Accurics provides the automation script linked on the setup wizard that can be executed via AWS CLI to create a Role ARN & External ID. The script creates an IAM role with read-only access in the specified AWS account for accurics to read the resource configuration.

Please note that AWS CLI will require “Admin” privileges to be able to successfully execute the script.

Step 3:
Select your cloud
provider, user
credentials &
select the VPC for
the scanning

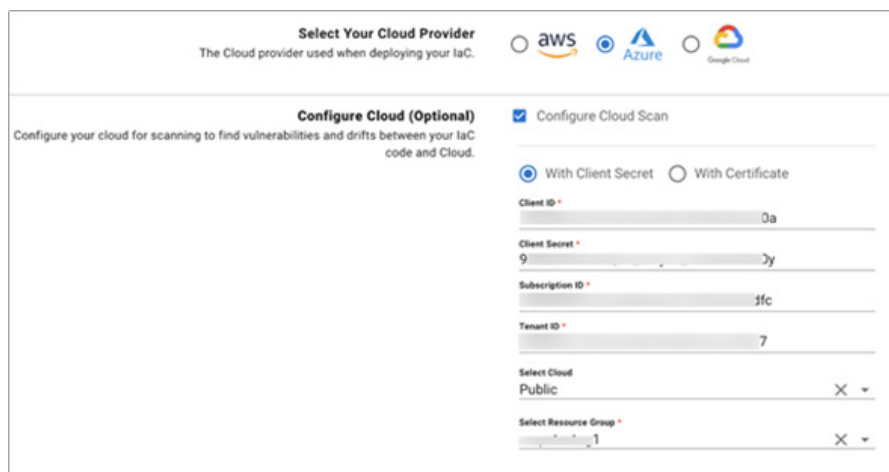


Integrating with Azure

You must set up the Accurics environment for scanning your Azure cloud resources.

On the Cloud Configuration page:

- click **Azure** as the Cloud Provider
- select **Configure Cloud Scan**
- specify the Client ID, Client Secret, Subscription ID, and the Tenant ID
- select cloud as **Public**



The screenshot shows the 'Configure Cloud (Optional)' section of the Accurics interface. At the top, there is a 'Select Your Cloud Provider' section with radio buttons for AWS, Azure (selected), and Google Cloud. Below this, the 'Configure Cloud (Optional)' section is active, with a checkbox for 'Configure Cloud Scan' checked. Underneath, there are two radio buttons: 'With Client Secret' (selected) and 'With Certificate'. The form contains several input fields: 'Client ID' with the value '0a', 'Client Secret' with the value '9...y', 'Subscription ID' with the value 'ffc', and 'Tenant ID' with the value '7'. At the bottom, there are two dropdown menus: 'Select Cloud' set to 'Public' and 'Select Resource Group' set to '1'.

Integrating with Google Cloud

You must set up the Accurics environment for scanning your Google Cloud resources.

Select Your Cloud Provider
The Cloud provider used when deploying your IaC.

aws Azure Google Cloud

Configure Cloud (Optional)
Configure your cloud for scanning to find vulnerabilities and drifts between your IaC code and Cloud.

Configure Cloud Scan

Select a region*
US West (S. California) (us-west2) ▾

Service Account Credentials JSON* (Please upload your service account credentials file.)

Upload file
 Upload Certificate File ×

On the Cloud Configuration page:

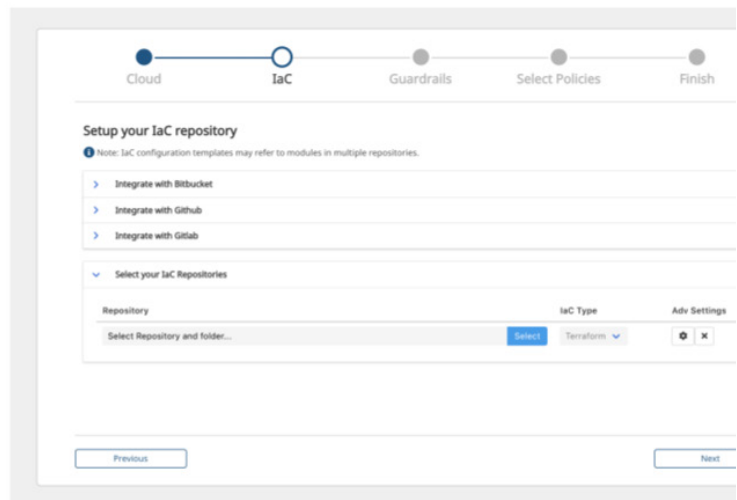
- click Google Cloud as the Cloud Provider
- select Configure Cloud Scan
- select a region
- Upload the Google Service Account Key JSON file.

Integrate with Code Repository

In addition to monitoring your cloud configuration, Accurics will also scan the Infrastructure as Code files in your code repository. When you Remediate a policy violation found in your code repository, Accurics will help remediate the violation by opening a Jira ticket with the Pull Request for a file that fixes the issue and mapping the violation back to the source code.

Currently, Accurics integrates with Github, Bitbucket & Gitlab. Accurics also integrates with your hosted source code repositories, which are behind your enterprise firewall. We will provide the steps in separate documents if requested.

Step 4:
Connect to your repository provider and select the IAC repository from the drop-down

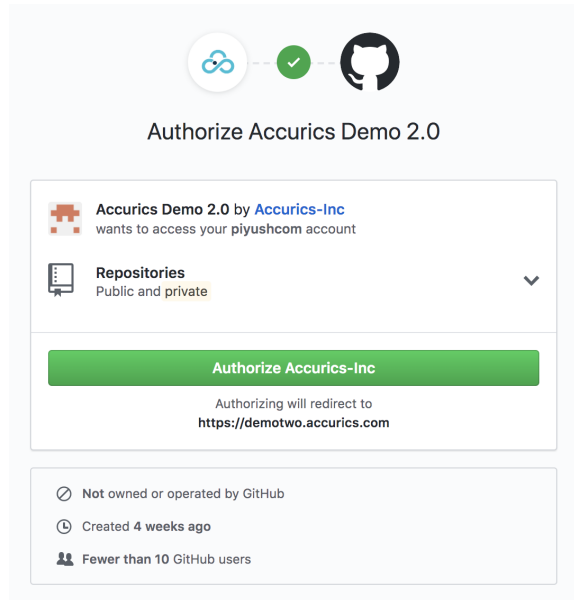


In order to integrate Accurics with Github, Bitbucket, or Gitlab, perform the following steps:

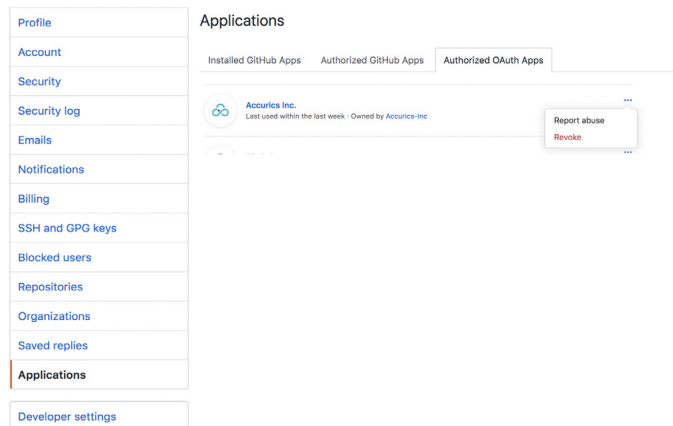
Step 1: Authorize Accurics to access your repositories

Select your respective repository provider and click "Connect". Let's use GitHub in this example. Accurics will be using Oauth 2.0 as a mode of authorization.

Click on "Authorize Accurics-Inc". This operation will add an Accurics app to your Github account with "READ ONLY" privileges.

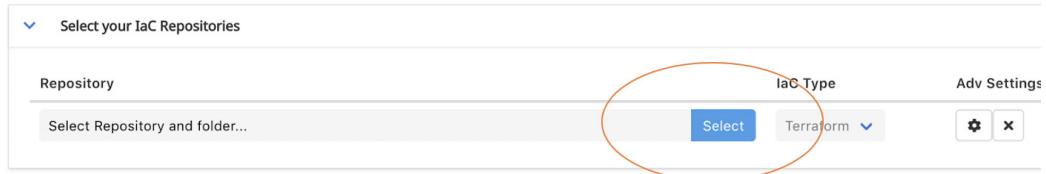


You can revoke the authorization anytime by going to your Github account and clicking on settings section.



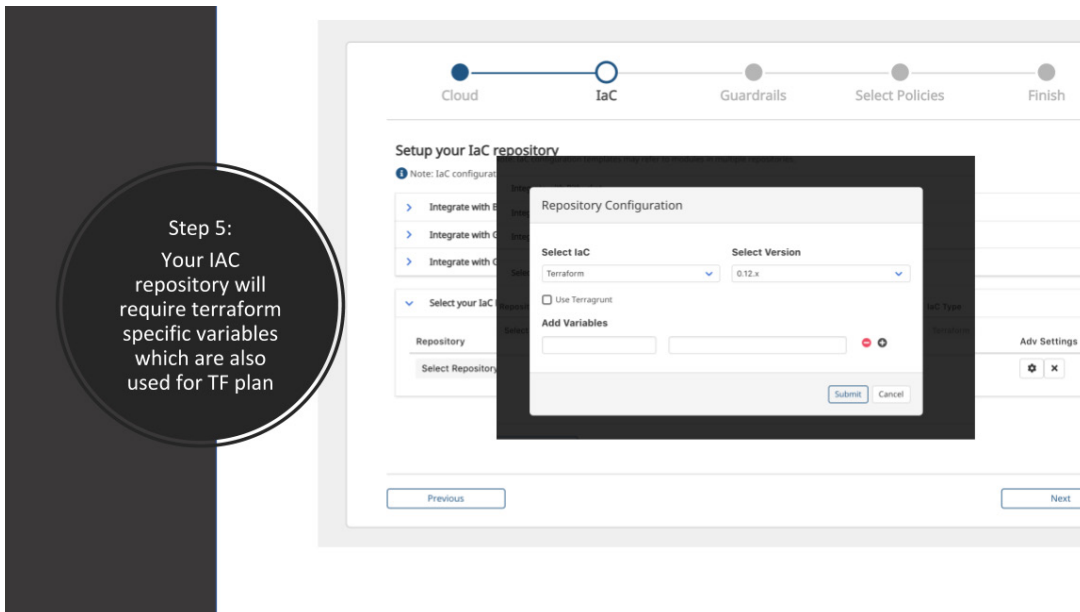
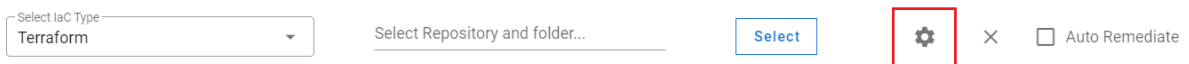
Step 2: Select the terraform repository

Once you have authorized Accurics to access your account, next step is to select the terraform code repository from the drop down as shown in the below screenshot.

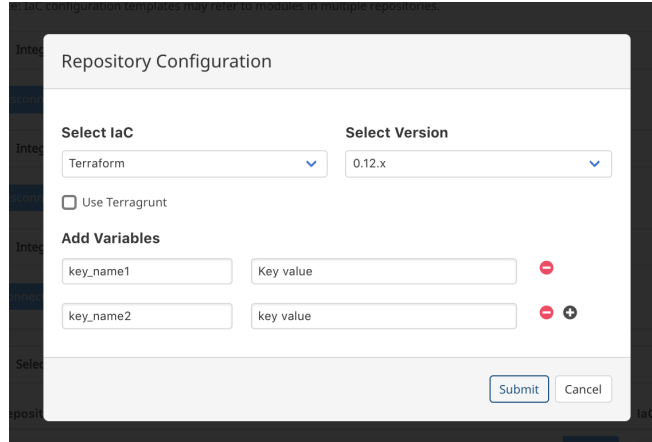


Step 3: Entering TF plan input variables as part of the configuration steps

Once you have selected the terraform code repository, please select the "Advanced Settings" option that is next to the repository location. Please refer to the below screenshot



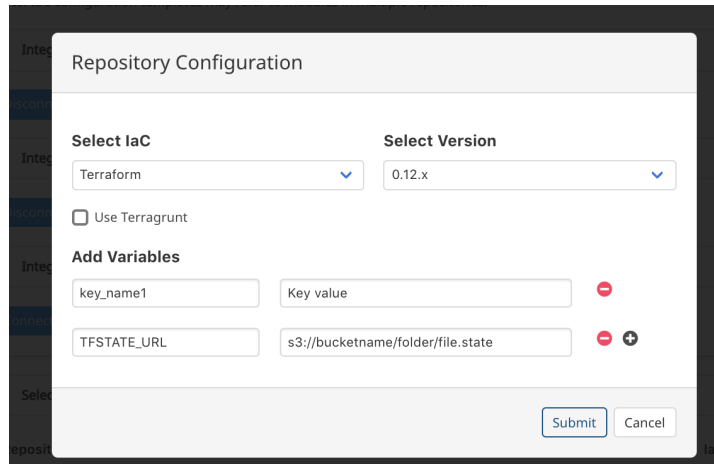
This will open a popup window requesting information about terraform code version and input variables that you are using to run your terraform plan. You will need to add all the variables, their key name and key value in the input boxes given in the popup window.



Step 4: Terraform State file information

Accurics may require access to terraform state files to get resource attribute information in cases when your resources do not have a name tag. This is to ensure that all cloud resources are accurately mapped to the terraform code even if resources do not have tags.

Note that the role ARN you are providing should have access to this S3 bucket resource.



Step 5: Selecting the compliance policy

This is the last step to set up Accurics to be able to scan and analyze the cloud resources for any misconfiguration vulnerabilities.

Select any of the following policies and you are done.



Select the security policy you would like to use for the assessment

<input type="checkbox"/> Accurics AWS EKS Security Best Practices 	<input type="checkbox"/> Accurics Network Security for AWS 	<input type="checkbox"/> Accurics Security Best Practices for AWS
<input type="checkbox"/> AWS CIS Benchmark Policy 	<input type="checkbox"/> GDPR Readiness Best Practices for AWS 	<input type="checkbox"/> HIPAA Best Practices for AWS
<input type="checkbox"/> NY DFS for AWS 	<input type="checkbox"/> PCI DSS Best Practices for AWS 	<input type="checkbox"/> SOC2 Best Practices for AWS

Once you have selected the desired compliance policy, review the provided information for the accuracy and click "Finish". This step will create an Accurics environment and will redirect you to the environment dashboard.



Please review your environment configurations

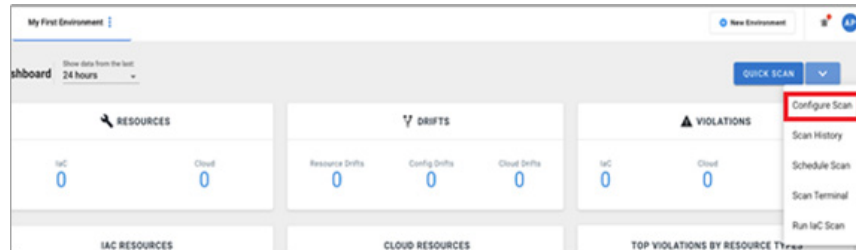
Cloud Provider The Cloud provider used when deploying your IaC		
Cloud Details The Cloud details that will be used when scanning for violations and drifts	Region	us-west-2
	Access Key	Not configured
	Secret Key	Not configured
	Role ARN	arn:aws:iam::3[redacted] Role
	External ID	Not configured
	VPC	Not configured
Repositories The repositories that will be scanned	Not configured	
Policies The policies that will be run against your IaC repositories and Cloud	Not configured	

Congratulations! Your first environment is now ready to use.

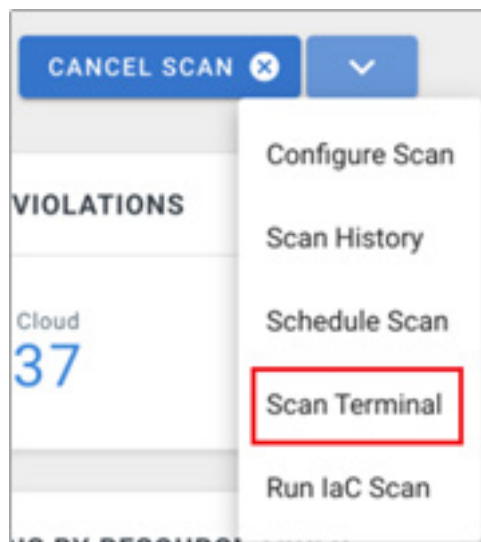
Now you can go to the environment dashboard and initiate an on-demand cloud scan. See [Initiating a cloud scan](#)

Initiating a cloud scan

1. Go to your Accurics environment.
2. Select **Configure Scan** from the QUICK SCAN menu.



3. On the **Scan Options** page, select the resource types to scan, and then click **Run Scan**.
4. To see the scan progress in real time, click **Scan Terminal** from the menu shown in Step 1.



Example of an ongoing scan:

Resource	Total Scanned	Status
Disks	3	COMPLETED
Network Interfaces	3	COMPLETED
Virtual Machine	3	COMPLETED
Virtual Network	2	COMPLETED
Network Security Groups	1	COMPLETED
Analysis Services	0	COMPLETED
Application Security Groups	0	WAITING
Resource Groups	0	WAITING

Accurics CLI

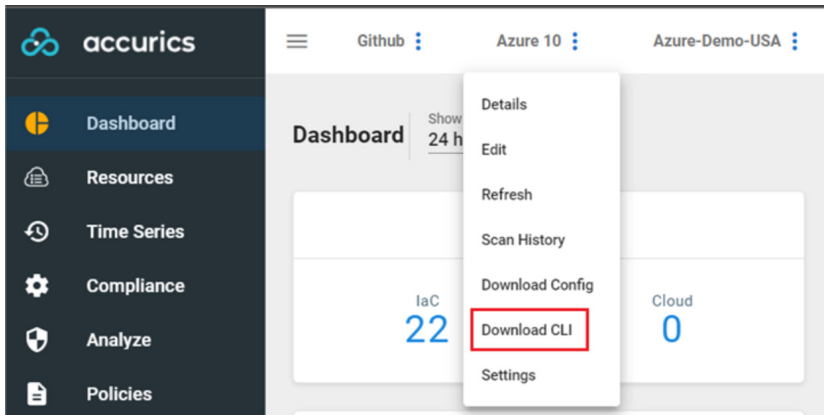
You can use the Accurics CLI to scan code on your local machine.

Downloading the Accurics CLI

You must download the Accurics CLI before you can use it

To download the Accurics CLI:

1. Login to the Accurics console.
2. Click **Download CLI** from the environment setup to scan the IAC repository used for CI/CD builds.



The compressed file contains:

- The Accurics executable
- A Config file containing the environment details

Installing the Accurics CLI on your local machine

The procedure to install the Accurics CLI is different for each operating system.

MAC

Simply run the following command to install the Accurics CLI

```
brew install accurics
```

Linux

Copy the accurics CLI to your usr/local/bin directory

```
cp accurics /usr/local/bin/
```

Provide executable access to the file

```
chmod +x accurics
```

Copy the config.zip file to the folder where you run “terraform plan” or “terragrunt plan” and then extract then file

```
cp config.zip <terraform folder path>  
unzip config.zip
```

Windows

Copy the config.zip file to the folder where you run “terraform plan” or “terragrunt plan” and then extract then file

```
cp config.zip <terraform folder path>  
unzip config.zip
```

Scan your terraform files using Accurics CLI

Perform the following steps to scan your terraform files. On Windows, use "accrucs.exe" instead of "accurics" while running the commands.

1. Run the following command to initialize Terraform:

```
accurics init
```

3. Use the following commands to run your Terraform/Terragrunt plan:

Terraform plan

```
accurics plan -config=<config file name>
```

Terragrunt plan

```
accurics tgplan -config=<config file name>  
accurics tgplanall -config=<config file name>
```

4. Use the following command to run your CFT plan:

```
accurics plan cf -config=<config file name> -  
templateFile=<template file name>[mandatory] -  
paramFile=<paramfile file name> -stackName=<stack name>
```

Examples:

```
accurics plan cf -config=config -  
templateFile=template.json  
-paramFile=params.json stackName=mystack  
  
accurics plan cf -config=config -templateFile=template.json  
  
accurics plan cf -config=config -  
templateRepo=. (To recursive scan all directories)
```

Note: *templateFile* is mandatory, whereas *paramFile* and *stackName* are optional.

Here is a sample output of the Accurics CLI:

```
MacBook-Pro acqa-repo1-aws-tf12-part1 % accurics plan -config=config_acqa-repo1-aws-tf12-part1
2021/01/15 18:33:44 runPlan...
2021/01/15 18:33:44 [plan -out=1610715824491.out]

2021/01/15 18:34:40 Running Accurics analysis...
/Users/spat/iDrive/Accurics/source_codes/acqa-repo1-aws-tf12-part1
2021/01/15 18:34:40 mapping terraform resources to source code...
2021/01/15 18:34:40 Repo Root Path... /Users/spat/iDrive/Accurics/source_codes/acqa-repo1-aws-tf12-part1
2021/01/15 18:34:40 Current working directory ... /Users/spat/iDrive/Accurics/source_codes/acqa-repo1-aws-tf12-part1
2021/01/15 18:34:40 getting source code for all the resources present in '/Users/spat/iDrive/Accurics/source_codes/acqa
2021/01/15 18:34:40 getting source code for all the resources present in '/Users/spat/iDrive/Accurics/source_codes/acqa
m/lgallard/terraform-aws-codebuild.git'
2021/01/15 18:34:40 resources to source code mapping done!
2021/01/15 18:34:40 Creating dependency graph...
2021/01/15 18:34:40 GetDotFileUsingGraph Directory: /Users/spat/iDrive/Accurics/source_codes/acqa-repo1-aws-tf12-part1
2021/01/15 18:34:43 Using configuration file:- config_acqa-repo1-aws-tf12-part1

-----

Accurics successfully scanned the repository! Following is the summary - for details visit Accurics Web Console.

{
  "resources": 48,
  "violation": 4,
  "low": 0,
  "medium": 0,
  "high": 4,
  "native": 1,
  "inherit": 3,
  "drift": 0,
  "iacdrift": 0,
  "clouddrift": 0
}

-----

MacBook-Pro acqa-repo1-aws-tf12-part1 % echo $?
1
MacBook-Pro acqa-repo1-aws-tf12-part1 %
```

Accurics CLI returns "1" if high severity violations are detected, and "0" when there are no high severity violations.

The scan results, including the number of resources, violations, etc., are sent to the Accurics console.

Viewing and Analyzing the scan results

Once the code & cloud scan is completed, the dashboard will refresh to show you the summary of findings.

Following are the few key indicators that you can review to make sure code & cloud scan has discovered all the expected resources correctly.

1. IAC resources:

These are the resources that Accurics has discovered from the provided terraform repository. Please note that sometimes terraform code may have resources that can't be directly mapped to the cloud resources due to multiple reasons such as

- Terraform specific resources such as Null resources
- Resources attributes are defined as resources, but in the cloud these resources are part of an AWS resource. For example, in terraform security group and security group rule are two separate resources, but in the AWS cloud, security group rules are part of the security group resource.

Accurics will continue to provide more granular information on such resources to have you focus on key resources.

2. Cloud resources

These are the resources that Accurics has discovered from the provided target AWS account and the VPC.

Clicking on the number, it will take you to the "Inventory" view to see more granular information about these resources.

3. IAC Violations

IAC violations is the number highlighting all the compliance and governance issues found in the resource configurations in your terraform code.

4. Cloud Violations

Cloud violations is the number highlighting all the compliance and governance issues found in the cloud resource configurations. You can find more detailed information in the "Forensics" view" on the left navigation.

5. Resource drifts

Resource drifts are the number of resources that couldn't be mapped to the IAC terraform code provided at the time of environment setup. There can be multiple reasons why these resources couldn't be mapped with the resources defined in your IAC code. Some of the reasons could be as follows:

Brownfield resources: These are the resources that may have been added/created in the cloud account directly and not provisioned through your terraform code.

Untagged resources: If resources do not have appropriate tags

You can find more detailed information in the "Inventory" view" on the left navigation.

6. Configuration drifts

These are resources that have different configurations in cloud & IAC code and have drifted from the single source of truth which is terraform code.

It is very important to resolve these drifts as they are breaking the immutability of your infrastructure and continue to introduce more violations.

7. These drifts can be good or bad but they will introduce cloud posture drifts. Accurics aims to eliminate these drifts to keep infrastructure secure.

Setting up IaC Scan for On-premises Repository

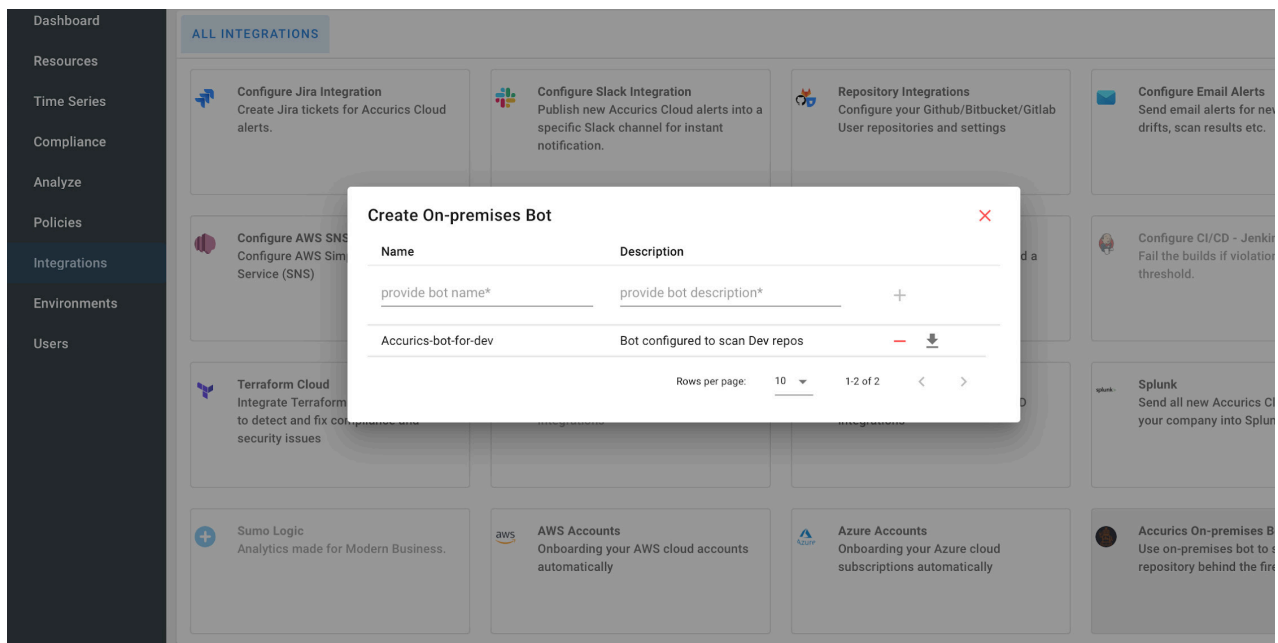
If you have your Code repositories deployed behind the firewall, you can use Accurics On-premises Bots to connect to the repository. The Accurics Bot Scans the Repository within the firewall-bound network and sends processed data to Accurics Services for reporting on Accurics Cloud Console.

TCP port 9022 should be accessible by Accurics host on-premises setup. GitHub Enterprise host should be able to access the Bot host and vice-versa.

Here are the steps required to configure an On-prem Accurics Bot

1. Create An On-premises Bot

In Accurics Console, Visit Integrations -> Accurics On-premises Bot and create one



2. Download Bot Deployment File

Accurics Bot is a set of services hosted in Docker containers. The machine where the bot will be deployed need to have Docker run-time installed. The machine should also have Docker-compose installed.

Download the Bot Deployment file by clicking the download button next to the created bot. This is a deployment YML

3. Create OAuth Application on the GitHub Enterprise Server

Accurics Bot authenticates with GitHub Enterprise server using OAuth app. Create an OAuth app for Accurics Bot (Settings->Developer Settings->OAuth App).

Use following URL as callback URL in the OAuth APP:

<http://<Bot-IP-Address>:9022/v1/auth/oauth/github/callback>

Bot-IP-Address is the IP Address of the machine where Bot will be deployed.

Record the Client ID and Secret. These will be used later to configure the Bot.

4. Set up the Bot

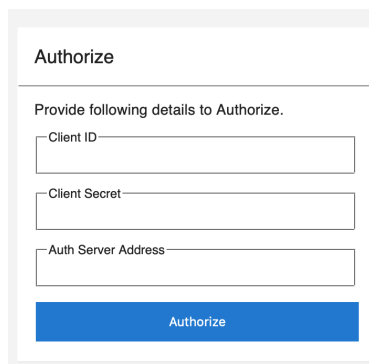
- a. Copy the configuration file to the virtual machine where the Bot needs to be hosted
- b. Login to the virtual machine through the shell terminal, go to the directory where the configuration file resides and run the following command,

```
$ docker-compose -f <configuration_file_name> up
```

- c. Wait for Bot to setup and then follow the instructions displayed on the terminal to complete OAuth Application Authorization
- d. Launch web browser on the machine and visit the URL

```
http(s)://<bot_host_ip>:9022
```

The browser will display the following web page to enter the GitHub App credentials.



Authorize

Provide following details to Authorize.

Client ID

Client Secret

Auth Server Address

e. Enter the following

- i. Client ID: Client ID of the OAuth Application
- ii. Client Secret: Client Secret of the OAuth Application
- iii. Auth Server Address: IP Address of the Github Enterprise Server

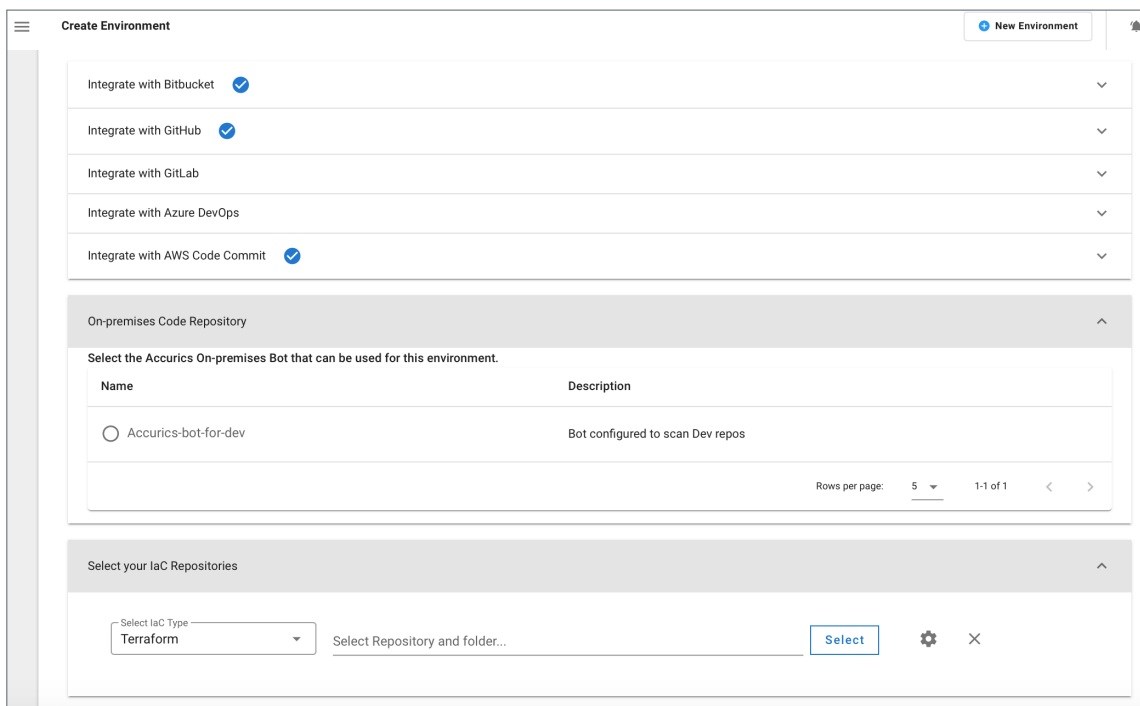
Click Authorize to authorize the Bot with the GitHub Server.

GitHub server will ask for user credentials as part of the OAuth based authentication and after successful authentication, the Bot will display a success message.

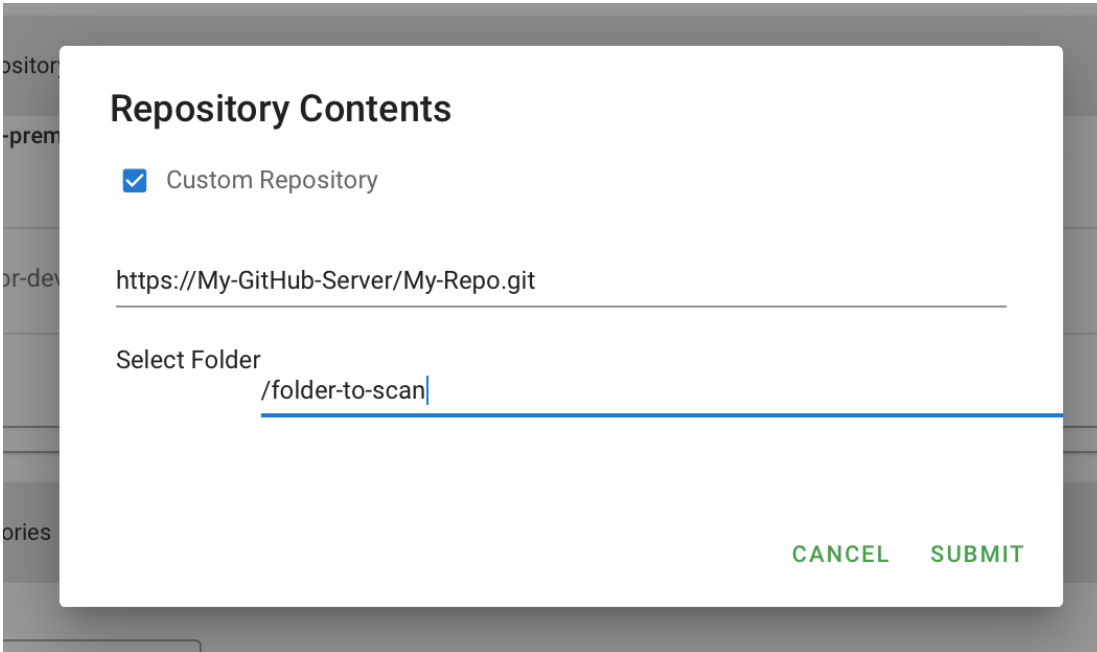
Your bot is authenticated with the GitHub server now and is ready to accept scan requests.

5. Create an Environment for On-Prem Repo

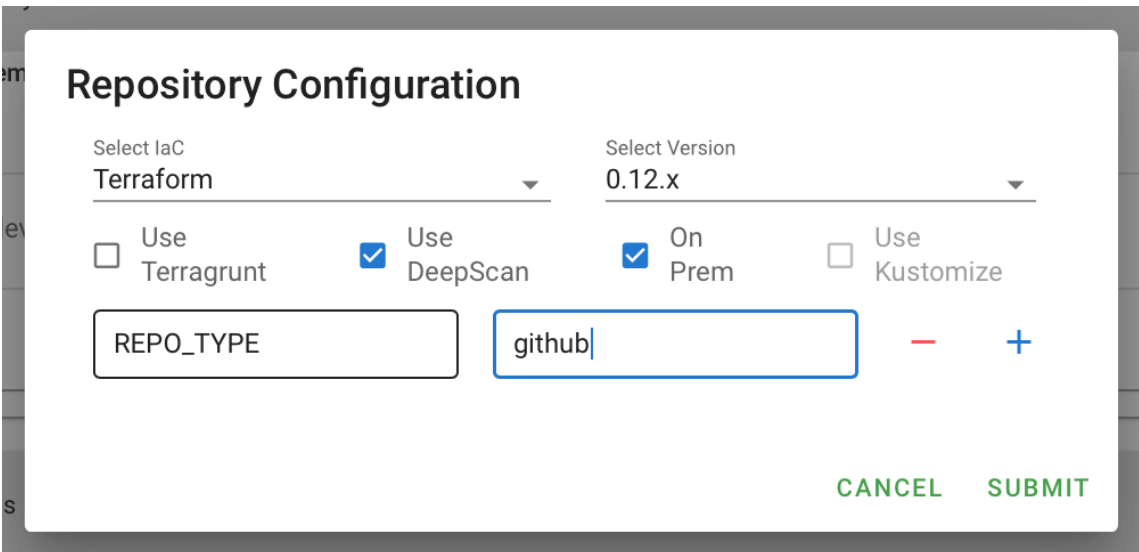
Visit Accurics Cloud Console and create an Environment.



Expand the On-premises Code Repository and select the Bot you just created. Click Select to enter Repository Contents.



Choose the option, Custom Repository and enter the URL of the Repo you want to scan. Select the settings for Repo and choose On-Prem option as displayed in the image below.



Follow the instructions to create the Environment. Once the Environment is created, click the Quick Scan option to initiate the IaC Scan. With Quick Scan, Accurics Console creates a Job for the Bot to scan the IaC. Bot picks up the job, scans IaC and sends results back to the console. You can check the violations and drifts in the console as before.



Thank you for choosing Accurics!

If you have any questions or need more information, please visit our website, submit a query via email, or give us call.

Customer Support

+1 833-466-8825 (1-833-IMMUTBL)

<https://app.accurics.com/login>

<https://www.accurics.com/contact-us/>