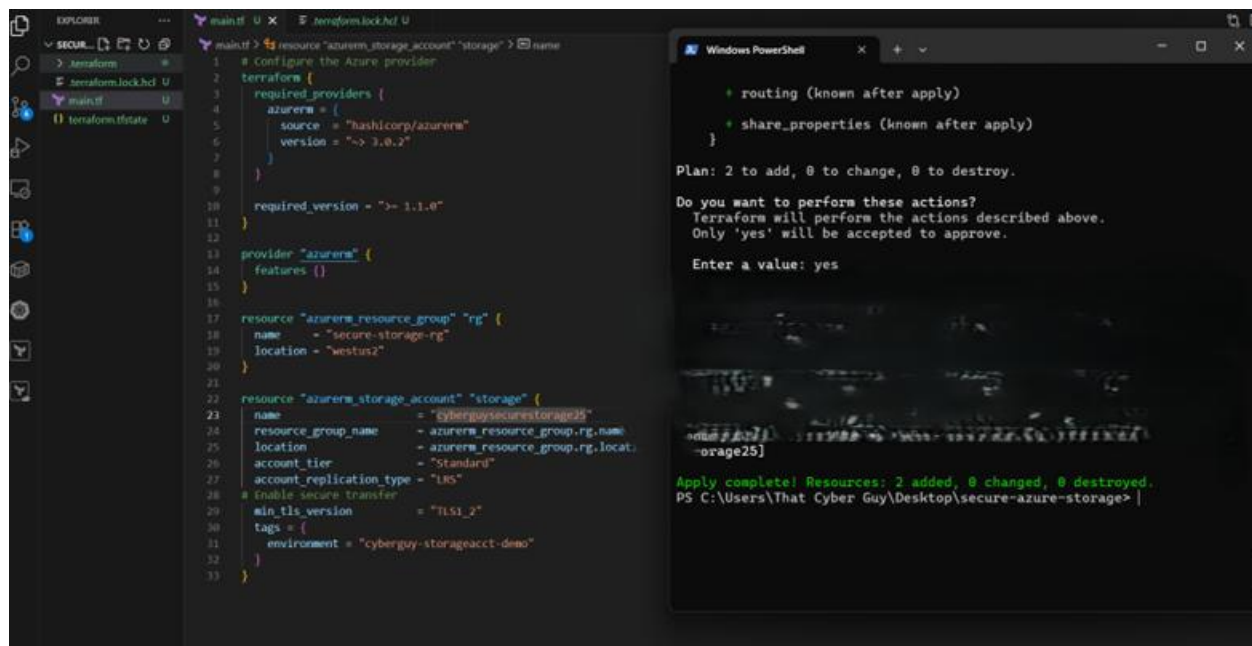


Use **Terraform** to create an **Azure Storage Account**, enforce a custom **Azure Policy** (Block Public Network Access), enable **diagnostic logging**, and manage everything through **GitHub pull requests**.

Workflow Overview:

1. Write Terraform configuration to define Azure resources.
2. Push code to GitHub and open a pull request.
3. GitHub Actions runs Terraform plan to preview changes.
4. Merge pull request to finalize changes.
5. Azure Portal confirms successful deployment.



The screenshot displays a development environment with two main windows. The left window, titled 'main.tf', shows a Terraform configuration file for creating an Azure Storage Account. The right window, titled 'Windows PowerShell', shows the output of the 'terraform apply' command.

```
1 # Configure the Azure provider
2 terraform {
3   required_providers {
4     azurerm = {
5       source = "hashicorp/azurerm"
6       version = "~> 3.0.2"
7     }
8   }
9   required_version = ">= 1.1.0"
10 }
11
12 provider "azurerm" {
13   features {}
14 }
15
16 resource "azurerm_resource_group" "rg" {
17   name     = "secure-storage-rg"
18   location = "westus2"
19 }
20
21 resource "azurerm_storage_account" "storage" {
22   name                = "cyberguysecurestorage25"
23   resource_group_name = azurerm_resource_group.rg.name
24   location             = azurerm_resource_group.rg.location
25   account_tier         = "Standard"
26   account_replication_type = "LRS"
27   # Enable secure transfer
28   min_tls_version      = "TLS1_2"
29   tags = {
30     environment = "cyberguy-storageacct-demo"
31   }
32 }
33
```

The PowerShell window shows the following output:

```
+ routing (known after apply)
+ share_properties (known after apply)
}
Plan: 2 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\That Cyber Guy\Desktop\secure-azure-storage>
```

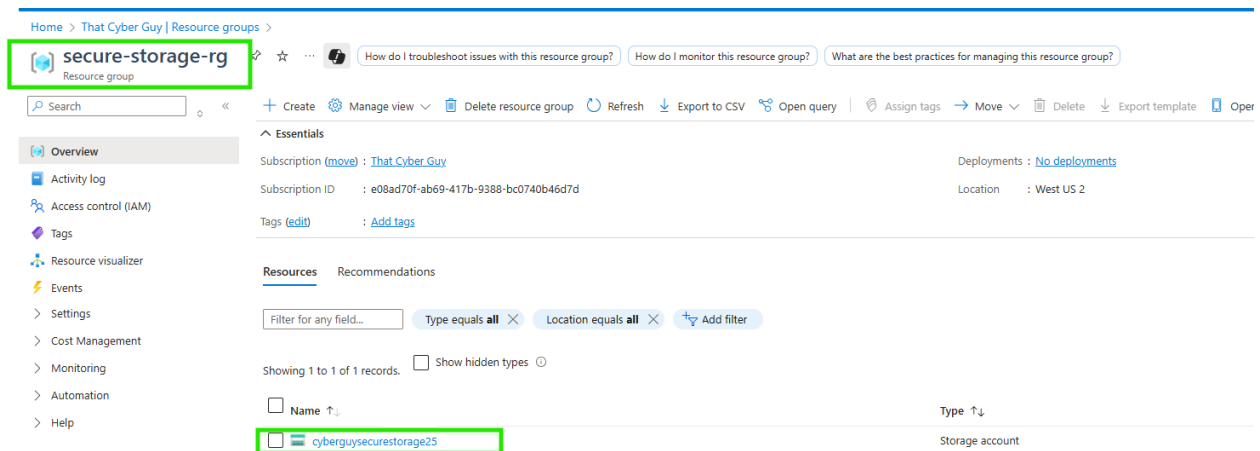


Figure 1.0. Creating the resource group and naming the resource using Terraform.

Public network access is enabled from all networks, which poses a high risk of data exposure or unauthorized access.

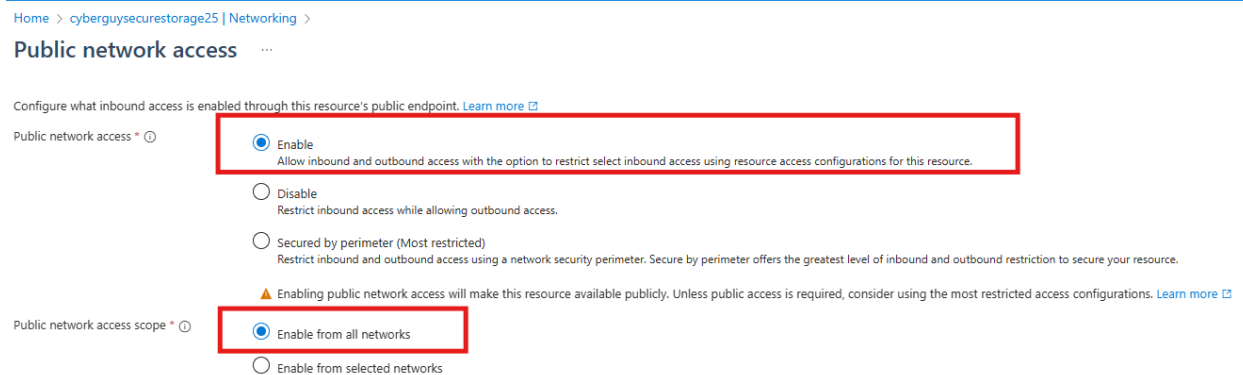


Figure 2.0. Public network access enabled from all networks.

To address this issue, I used Azure Policy to create a custom policy that enforces the 'Storage accounts should restrict network access' rule. This ensures that only trusted networks can access our storage resources, reducing exposure to public internet threats and aligning with the principle of least privilege.

Home > Policy | Definitions

Deny public network access on Storage Accounts >

Assign policy

Basics

Parameters

Remediation

Non-compliance messages

Review + create

Scope

Scope *

That Cyber Guy/secure-storage-rg

Learn more about setting the scope

Exclusions

Optionally select resources to exclude from the policy assignment.

Resource selectors

Expand

Using resource selectors, you can further refine this assignment's applicability by targeting specific subsets of resources. Expand to learn more.

Basics

Policy definition *

Deny public network access on Storage Accounts

Overrides

Expand

Using overrides, you can change the effects or referenced versions of definitions for all or a subset of resources evaluated by this assignment. Expand to learn more.

Assignment name *

Deny public network access on Storage Accounts

Description

This policy denies storage accounts that have public network access enabled

Policy enforcement

Enabled

Previous

Next

Review + create

Figure 3.0. Policy definition.

Deny public network access on Storage Accounts

Policy definition

[Assign policy](#) [Edit definition](#) [Duplicate definition](#) [Delete definition](#)

Essentials

Name	: Deny public network access on Storage Accounts	Definition location	:
Description	: This policy denies storage accounts that have public network access enabled.	Definition ID	:
Available Effects	: Deny	Type	:
Category	: That Cyber Guy Demo	Mode	:

Definition

Assignments (1)

Parameters (0)

```
9      "category": "That Cyber Guy Demo",
10      "createdBy": "",
11      "createdOn": "2025-08-10T01:05:06.084471Z",
12      "updatedBy": null,
13      "updatedOn": null
14    },
15    "version": "1.0.0",
16    "parameters": {},
17    "policyRule": {
18      "if": {
19        "allOf": [
20          {
21            "field": "type",
22            "equals": "Microsoft.Storage/storageAccounts"
23          },
24          {
25            "field": "Microsoft.Storage/storageAccounts/publicNetworkAccess",
26            "equals": "Enabled"
27          }
28        ]
29      },
30      "then": {
31        "effect": "Deny"
32      }
33    }
34  }
```

Figure 3.1. Custom policy definition that denies public access to the storage account.

After the custom policy shown in Figures 3 and 3.1 is applied, users **cannot** enable public network access for any storage accounts.

Home > cyberguysecurestorage25 | Networking >

Public network access

Configure what inbound access is enabled through this resource's public endpoint. [Learn more](#)

Public network access

☒ Enable
Allow inbound and outbound access with the option to restrict select inbound access using resource access configurations for this resource.

☐ Disable
Restrict inbound access while allowing outbound access.

☐ Secured by perimeter (Most restricted)
Restrict inbound and outbound access using a network security perimeter. Secure by perimeter offers the greatest level of inbound and outbound restriction to secure your resource.

⚠ Enabling public network access will make this resource available publicly. Unless public access is required, consider using the most restricted access configurations. [Learn more](#)

Public network access scope

☒ Enable from all networks

☐ Enable from selected networks

[Save](#) [Cancel](#)

Failed to save resource settings

Resource 'cyberguysecurestorage25' was disallowed by policy. Reasons: 'All Storage account should NOT have Public Network Access enabled.' 'Public Network Access Should NOT be enabled for storage account'. See error details for policy resource IDs.

[Help me troubleshoot](#)

Figure 3.2. Policy enforcement.

Created a new *logging.tf* file to provision a Log Analytics resource that will serve as the destination for storage account logs.

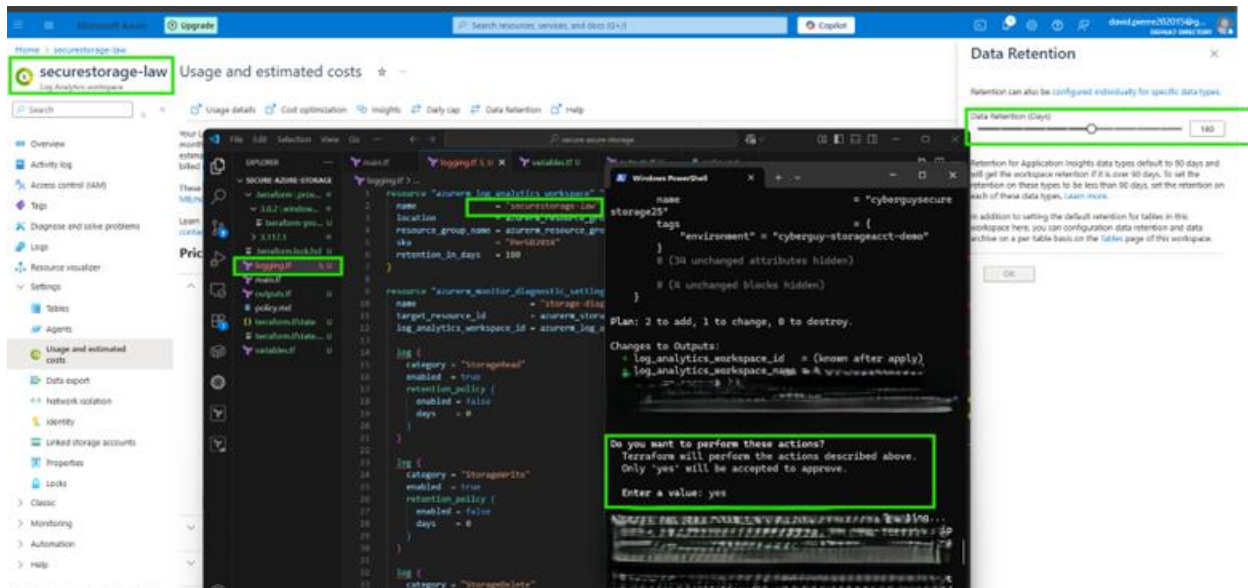


Figure 4.0. Log analytics creation.

L Blob storage logs are now forwarded to Log Analytics, where they are retained for 180 days.
Diagnostic setting

Save Discard Delete Feedback

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

A more flexible, faster, and robust way to collect metrics is in preview! Click [here](#) to configure platform metrics collection from microsoft.storage/storageaccounts/blobservices to storage account, event hubs, and Log Analytics workspace. [Learn more.](#)

Diagnostic setting name: CyberSIEM

Logs

Category groups

☒ audit ☐ allLogs

Categories

☒ Storage Read

☒ Storage Write

☒ Storage Delete

Destination details

☒ Send to Log Analytics workspace

Subscription

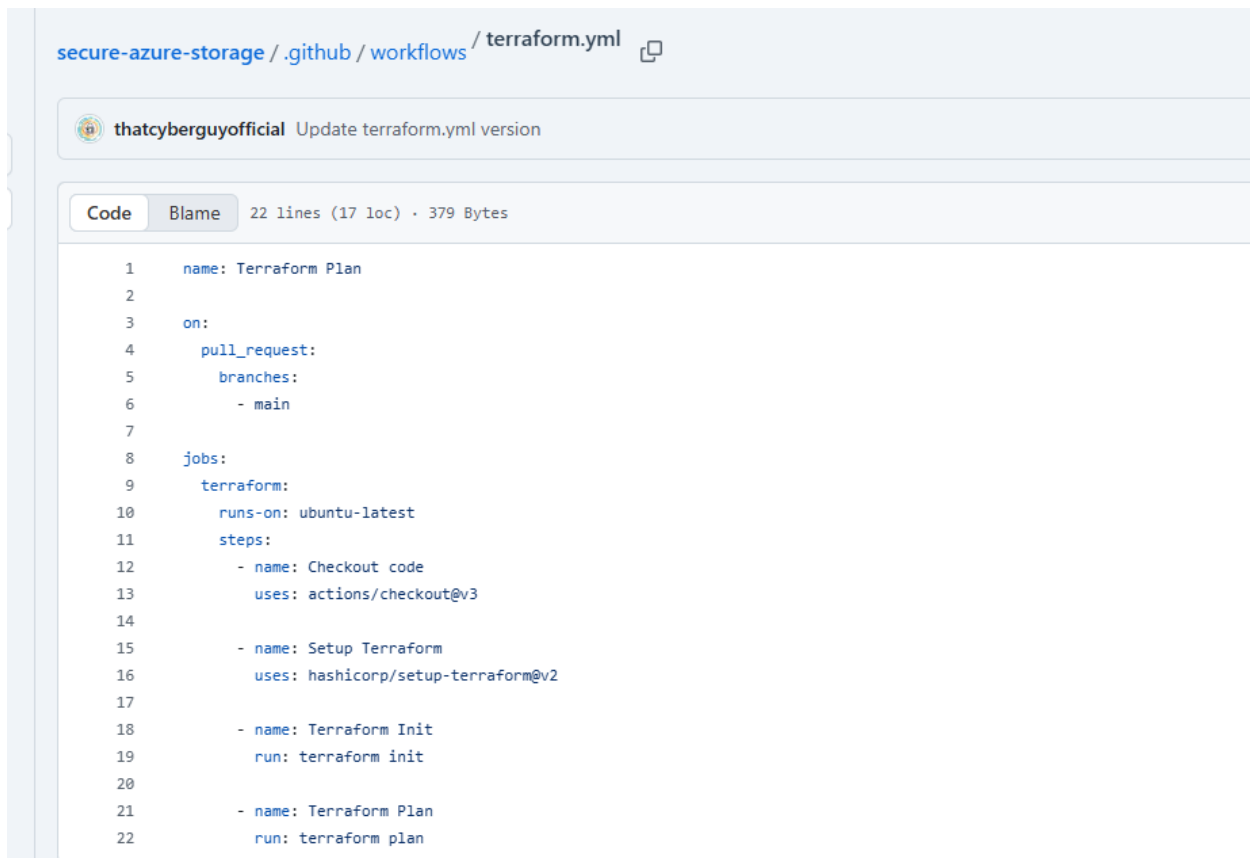
That Cyber Guy

Log Analytics workspace

securestorage-law (westus2)

☐ Archive to a storage account

Figure 5.0. Log Analytics workspace configured as the destination for storage account logs.



```
1  name: Terraform Plan
2
3  on:
4    pull_request:
5      branches:
6        - main
7
8  jobs:
9    terraform:
10     runs-on: ubuntu-latest
11     steps:
12       - name: Checkout code
13         uses: actions/checkout@v3
14
15       - name: Setup Terraform
16         uses: hashicorp/setup-terraform@v2
17
18       - name: Terraform Init
19         run: terraform init
20
21       - name: Terraform Plan
22         run: terraform plan
```

Figure 6.0. Terraform plan..

This project demonstrates a complete Infrastructure-as-Code (IaC) workflow using **Terraform** to provision resources in **Microsoft Azure**, integrated with **GitHub Actions** for automated CI/CD.