Checkov POC — Terraform IaC Security Scanning Integration

1 Executive Summary

This document outlines a Proof of Concept (POC) we built to explore how Checkov — an open-source IaC security scanner — can be integrated into Terraform pipelines using GitHub Actions. The goal was to automate security checks, enforce compliance, and improve infrastructure security posture for a new client project.

2 Project Background

We recently got onboarded to a new client project where our task is to implement automated security scanning in their Infrastructure-as-Code (IaC) workflows. To prepare, we built a POC using Terraform, Checkov, and GitHub Actions. This helped us understand Checkov's capabilities and how it fits into CI/CD pipelines.

3 What is Checkov?

Checkov is a static analysis tool developed by Bridgecrew (Palo Alto Networks) that scans IaC templates for misconfigurations and compliance violations.

Why it's useful:

- Supports AWS, Azure, GCP, Kubernetes
- Enforces standards like CIS, NIST, SOC2
- Allows custom policies and skip rules
- Integrates easily with CI/CD
- Outputs JSON/SARIF for auditability

4 POC Architecture & Workflow

Workflow Overview:

Code Push → GitHub Actions Triggered → Terraform Init & Validate → Checkov Scan → Soft Fail if Violations → JSON Report Generated → Annotations Displayed → Terraform Plan & Apply

Tools Used:

- Terraform
- Checkov
- GitHub Actions
- AWS

5 Implementation Details

Environment Setup:

- Installed Terraform & Checkov locally
- Configured AWS IAM user
- Added GitHub secrets for AWS credentials

Folder Structure:

terraform-aws-ec2-s3/

```
├── main.tf
├── variables.tf
├── outputs.tf
├── .checkov.yaml
├── .github/workflows/ci-cd.yml
└── README.md
```

6 Key Code Snippets

Terraform (main.tf):

```
provider "aws" {
  region = var.region
}

resource "aws_s3_bucket" "insecure_bucket" {
  bucket = var.bucket_name
}

resource "aws_instance" "insecure_instance" {
  ami = var.ami_id
  instance_type = var.instance_type
}
```

Checkov Configuration (.checkov.yaml):

```
soft-fail: true
framework: terraform
skip-check:
  - CKV_AWS_145
quiet: true
output: cli
```

GitHub Actions Workflow (ci-cd.yml):

Run Checkov (CLI + JSON) → Upload report as artifact → Display inline annotations → Run Terraform Plan & Apply

Checkov in Workflow (ci-cd.yml): GIVES OUTPUT IN CLI (Workflow logs)

```
- name: Run Checkov (CLI)
        uses: bridgecrewio/checkov-action@v12
        with:
          directory: .
          soft_fail: false
          output_format: cli
```

Checkov in Workflow (ci-cd.yml): GIVES OUTPUT IN JSON FORMAT

```
- name: Run Checkov (JSON)
        uses: bridgecrewio/checkov-action@v12
        with:
          directory: .
          soft_fail: false
          output_format: json
          output_file_path: checkov-report.json
```

7️⃣ Customizations & Enhancements

Feature	Description
Soft Fail Mode	Allows pipeline to continue despite warnings
Custom .checkov.yaml	Tailored scan behavior
JSON Output	Structured report for audit
Artifact Upload	Stores scan results
Inline Annotations	Highlights issues in PR

💬 Conclusion

This POC helped us validate Checkov's ability to automate IaC security scanning, detect issues early, and improve infrastructure security posture. It's a solid foundation for rolling out secure, scalable IaC practices for our client.