Checkov IaC security scanner

CONCEPTS

Checkov scans infrastructure-as-code (Terraform, Kubernetes, Dockerfiles, CloudFormation, CI configs) for misconfigurations without deploying anything.

IaC files tf, yaml, Dockerfile
↓ `checkov -d`
Framework runners auto-detected
↓ `policies`
Checks CKV_* IDs
↓ `-o`
Report pass / fail / skip

Every check has an ID like CKV_AWS_20; you use these IDs to select, skip and suppress checks.

SETUP

`pip install checkov`
Install with pip

`checkov --version`
Show the version

`checkov --help`
List all options

`checkov --list`
List available checks

SCANNING

`checkov -d <dir>`
Scan a directory recursively

`checkov -f <file>`
Scan a single file

`checkov -f tfplan.json --repo-root-for-plan-enrichment <dir>`
Scan a Terraform plan

FRAMEWORKS

`checkov -d <dir> --framework <name>`
Run only these frameworks

`checkov -d <dir> --skip-framework <name>`
Leave frameworks out

`checkov -d <dir> --framework secrets`
Hunt for hard-coded secrets

CHOOSING CHECKS

`checkov -d <dir> -c <id>,<id>`
Run only these checks

`checkov -d <dir> --skip-check <id>,<id>`
Skip these checks

`checkov -d <dir> --external-checks-dir <dir>`
Load custom Python or YAML checks

OUTPUT

`checkov -d <dir> --compact`
Hide code snippets

`checkov -d <dir> --quiet`
Show only failed checks

`checkov -d <dir> -o <format>`
Pick an output format

`checkov -d <dir> -o json`
JSON report

`checkov -d <dir> -o sarif`
SARIF for code scanning

`checkov -d <dir> -o junitxml`
JUnit report for CI

`checkov -d <dir> -o cli -o json --output-file-path console,<file>`
Print and save in one run

EXIT CODES AND CI

`checkov -d <dir> --soft-fail`
Always exit 0

`checkov -d <dir> --hard-fail-on <id|sev>`
Fail only on these checks

BASELINES

`checkov -d <dir> --create-baseline`
Save current findings as a baseline

`checkov -d <dir> --baseline <file>`
Report only new failures

SUPPRESSING FINDINGS

Terraform: inside the resource

```
resource "aws_s3_bucket" "site" {
  #checkov:skip=CKV_AWS_20:public static site
}
```