datree.io

Packs and Rules

Custom Rules

Create custom rules based on any or all of these options

- Structured File Properties
- Files & Directories
- Commit Meta-Data
- Github Settings (branch, repository, PR)

Built-in Rules

Or choose from our built-in list of curated rules

Dev Best Practices

Prevent secrets files from leaking into source code
Avoid security breach by ensuring generated secrets keys are excluded from the source code
Read More

Prevent pushing dependencies directories to source code
Don't increase the repository size by unintentionally pushing the project dependencies dir
Read More

Ensure all packages have a pinned version in the manifest
Pulling packages without setting their version can introduce breaking changes
Read More

Ensure separation of personal config files from source code
Including personal config files may introduce security risks if you expose details about your machine's setup
Read More

Prevent the usage of unlicensed OSS dependencies
The default copyright laws apply for dependencies without a license, meaning you cannot modify or redistribute them without explicit permission from the copyright holder

Prevent the usage of GPL type licensed OSS dependencies
Using strong copy-left licensed (e.g. GPL type) dependencies can legally enforce you to make your entire project source code publicly available

Docker

Prevent pulling images from public registry
For security purposes, it is recommended to pull only verified images from a private registry
Read More

Ensure images have a pinned version
Pulling an image without setting its version can introduce breaking changes
Read More

Ensure all apt-get / yum packages have a pinned version
Pulling packages without setting their version can introduce breaking changes
Read More

Ensure LABEL maintainer property exists
Indicates who is responsible for the container for better code ownership
Read More

Ensure USER property exist and it's not root
Indicates who is responsible for the container for better code ownership
Read More

Ensure HEALTHCHECK property exist in all config files
Specify health check API allows you to determine the container readiness in complex environments
Read More

Travis CI

Ensure Node builds are installing dependencies via `npm ci`
The command `npm ci` is significantly better than a regular `npm install` as it meant to be used in CI process
Read More

Ensure notification step exist s in the build process
Notifications will keep engineers updated on the build status, allowing for quick iterations
Read More

Ensure Travis CI configuration is set for services
Achieve consistency by making sure all services are treated the same way and have a proper CI flow.

Serveless

Prevent deployments of untagged resources on AWS
Custom tags make it easier to track resources and the AWS spending on Lambda functions
Read More

Ensure there are (at least) 2 subnets when setting VPC
Mitigates the risk of IP exhaustion in the subnet and limits the blast radius to Lambda functions
Read More

Ensure CloudFormation deploy role is configured
Pass a dedicated deployer role to CloudFormation to apply the principle of least privilege to the deployment pipeline to reduce security risks
Read More

Prevent permissive IAM role statements
Follow the principle of least privilege and grant functions the minimal amount of access they need to reduce security risks
Read More

Git

Ensure code owners defined for project
Define people or teams responsible for a project or files inside the project to automatically request reviews from the pre-defined code owners
Read More

Prevent out-of-date pull requests
Ensures all branches are synced and up to date with the target branch code changes
Read More

Ensure a .gitignore file included in project
Avoid unwanted files to be committed to remote repository
Read More

SOC2 \ Change Control

Ensure pull request title link to a Jira ticket
Create a logical link between Jira (planning system) and the code changes for better traceability on each pull request
Read More

Ensure commit message link to a Jira ticket
Create a logical link between Jira (planning system) and the code change for better traceability on each commit
Read More

Prevent unrecognized committers and authors
race any line of code in the entire codebase to its true author and committer by highlighting users with misconfigured local git client
Read More

Ensure pull request template is set
The template will prepopulate the description field to standardize the information and tasks needed to be completed for any code change
Read More

CircleCI

Ensure the use of caching mechanism when installing `npm` dependencies
Caching is one of the most effective ways to make jobs faster on CircleCI by reusing the data from expensive fetch operations from previous jobs
Read More

Ensure build Docker images have a pinned version
Pulling Docker images without setting its version can introduce breaking changes
Read More

Prevent the usage of CircleCI 1.0 deprecated version
Since August 31, 2018, CircleCI 1.0 is depracted and users should use CircleCI 2.0 instead
Read More

Ensure all orbs have a pinned version
Pulling orbs without setting their version can introduce breaking changes

Ensure CircleCI configuration is set for services
Achieve consistency by making sure all services are treated the same way and have a proper CI flow

Open-Source

Ensure all tasks on the task list are completed
Task lists in the pull request descriptions are incredibly useful for project coordination and keeping track of important tasks
Read More

Ensure commit message subject does not exceed 50 characters
Shorter subject lines are more readable and encourage the author to think for a moment about the most concise way to explain the code change
Read More

Ensure Angular's commit message convention is followed
For better collaboration across committers, ability to leverage git utilities and use of automation tools
Read More

Ensure OSS mandatory files (License, README and Code of Conduct) are included
Include these mandatory files so others will be more likely to contribute to your projects

Ensure Codecov tests coverage is reported
Test coverage is an important indicator of software quality and an essential part of software maintenance

Prevent empty pull request description
Pull request descriptions help the reviewer understand the reasoning behind the code changes and speed up the review process for everyone

TypeScript

Ensure ESLint file exists when TypeScript is used
A linter helps detect errors during development and not during runtime and ESLint is de facto the standard linter for TypeScript repositories
Read More

Ensure that "use strict" mode is used (`strict` is set)
Strict mode makes it easier to write "secure" JavaScript code by turning previously accepted "bad syntax" into real errors
Read More

Prevent the usage of (deprecated) TSLint config file
Since January 2020, TSLint is deprecated and TypeScript users should use ESLint instead
Read More

Prevent TypeScript from compiling node modules source code
Exclude the node_module path to keep performance at a high level. This way, the TypeScript compiler will know the files in this path are not part of your source code
Read More

Ensure the eslint-plugin package is included when TypeScript is used
The package `@typescript-eslint/eslint-plugin` is a required plugin to enable linting on TypeScript files

Ensure parser package is included when TypeScript is used
The package `@typescript-eslint/parser` is a required plugin for ESLint to understand (parse) TypeScript's syntax

Technologies Partners

Among our integrations

- Microsoft
- JFrog
- sonatype
- Checkmarx