IaC for Security and Speed Cheatsheet: 8 best practices for IaC adoption

snyk

1. Share IaC code in a version control system

A durable history of changes allows experimentation across a team of users: you can undo mistakes, and review history to learn why changes were made that you've since forgotten.

2. Run tests early and often to detect flaws in your IaC

You can and should use every quick method of detecting flaws as early in development as possible.

- Snyk IaC tests IaC code and plan output to look for security misconfigurations.
- Tools like `terratest` apply configurations and validate that what happened matches what was expected.

3. Only automated systems deploy changes

Pipelines should be the only way that changes get made if you want predictable tests. No more failures because a colleague made a "temporary" change!

4. Actions performed in the pipeline are idempotent

To have confidence in your pipeline, you want to be able to predict its behavior and assert your desired outcome. If your results are unpredictable, it's tough to fully automate testing and deployment.

5. Apply IaC frequently, even if it has not changed

Re-applying your IaC reconciles your desired state. If every run of our pipeline is predictable AND our pipelines are the only way changes are made, then reapplying IaC state should be easy.

6. Maintain an IaC component bill of materials

Having a record of the versions of IaC and app components that are tested together aids reproducibility. If human error destroys an entire environment, including the CI server, you can rebuild exactly what was there.

7. Sign IaC changes so authorship can be proven

Logging into git provides access control, but not a true record of authorship. Configure your git system to enforce the use of signing keys to provide proof of identity.

8. Promote changes through environments

Along with IaC, you'll need to test your app code, service interactions, and more, so you'll likely need multiple environments that are chained together. Only when all "lower" environments pass their tests do the "higher" environments, and ultimately production, get deployed.

Authors

Jim Armstrong
Sr. Product Marketing Director at Snyk
jim.armstrong@snyk.io
@jdarmstro

ENGINEER BETTER ›

EngineerBetter
contact@engineerbetter.com
@engineerbetter

Learn About Snyk IaC