finmid

KEYCLOAK DevDay 2025

Integration Testing of Keycloak Terraform Modules

Darmstadt 06.03.2025

sre@finmid.com

Hello 👋

A short introduction and explaining the problem statement that drives us through the presentation.

1

Use Cases

Discovering the use cases that let us come up with the presented requirements and the problem solution.

2

Test Concepts

Exploring our core testing concepts and approaches that form the foundational parts of our testing stack.

3

Demo 🚀

Slides are boring. Good that we have prepared a demo that shows the concepts right in the code.

4

Bartosz Bednarek

Head of Engineering
finmid

@bbednarek
@bartoszbednarek

Fabian Böttcher

Site Reliability Engineer
finmid

@Cakasim

***finmid** is the financial infrastructure provider to today's most popular platforms.*

With just a few lines of code, finmid enables any B2B software platform to offer financing to their customers at scale, driving revenue and customer retention.

*** founded 2021 in Berlin**

finmid

Integration Testing of Keycloak Terraform Modules

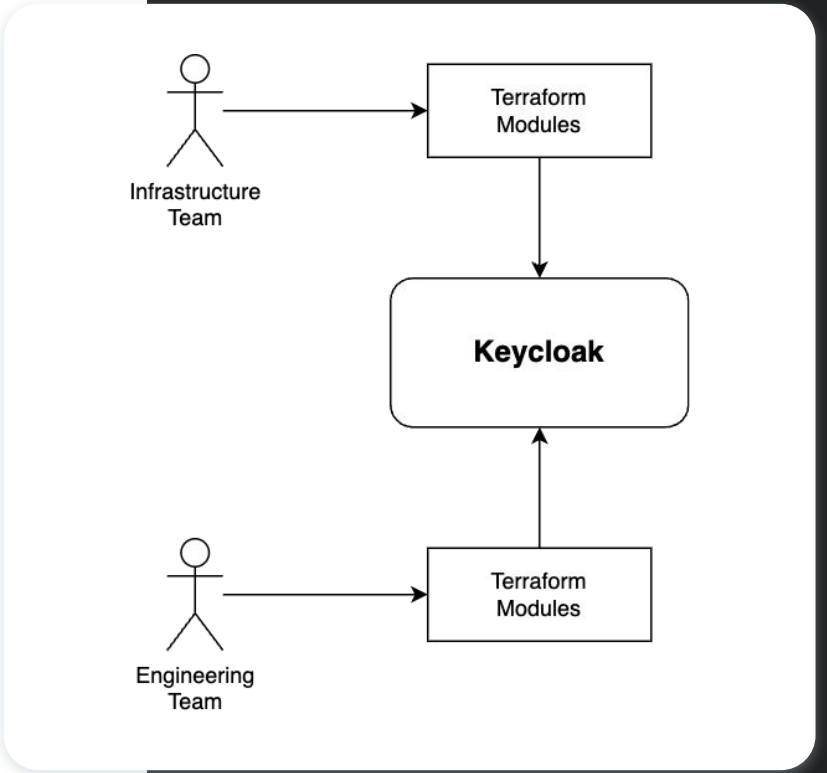
How to ensure Keycloak Terraform modules work as expected

sre@finmid.com

How stakeholders interact with Keycloak.

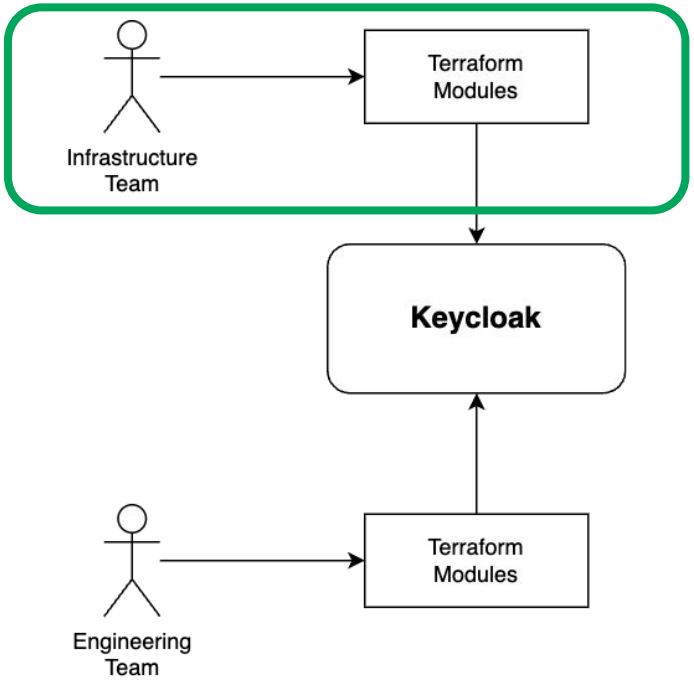
#1 The **Infrastructure Team** provisions and configures foundational aspects of Keycloak using Terraform modules.

#2 The **Engineering Team** utilizes Keycloak to authenticate users and applications; they create OIDC clients using Terraform modules.



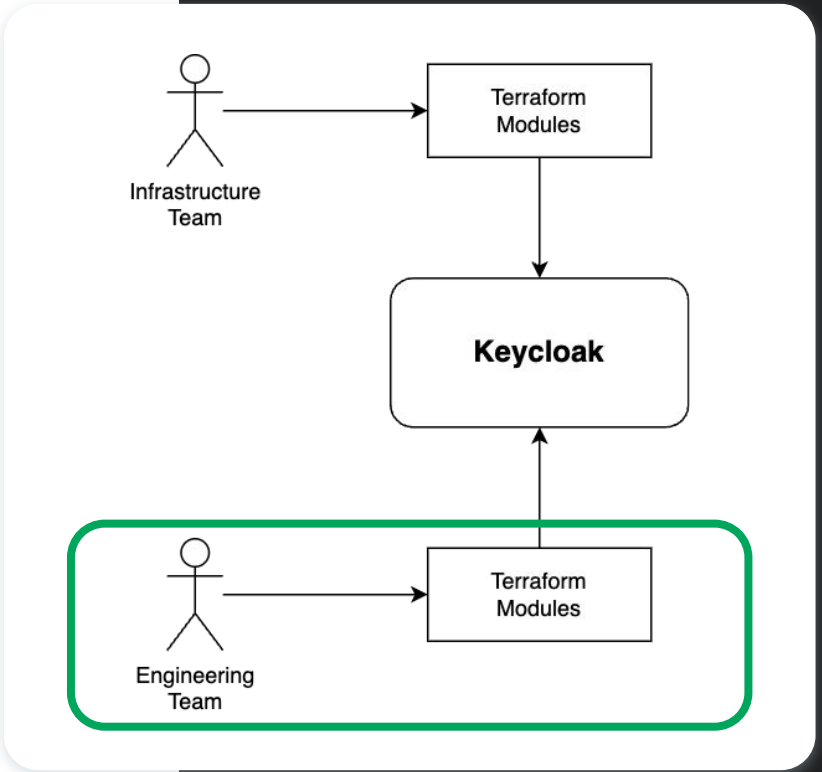
#1
How the Infrastructure Team uses Terraform Modules

- Create and configure realms
- Set up identity providers (e.g., Google)
- Configure authentication flows
- Additional realm level components



#2
How the Engineering Team uses Terraform Modules

- Abstraction for working with Keycloak
- Create clients to **authenticate users** of Keycloak protected applications
- Create clients to **authenticate services** that access the Keycloak Admin API



Engineering Team
Expectations

“We should be able to provision Keycloak resources like OIDC clients on a self-service basis, in a safe and reliable way without being blocked by the Infrastructure Team.”

A smart developer

Infrastructure Team
Expectations

“We should focus on innovation and improvements, support in important matters and not wasting time in setting up common infrastructure for teams.”

A motivated SRE

How to match these expectations.

- See Terraform modules as products that allow engineers to interact with infrastructure in a safe and predictable way.
- Interacting with infrastructure becomes a self-service for common tasks and engineers are unblocked.
- **This requires thorough testing** which covers relevant configurations and ensures the modules provision the expected state of resources.

finmid

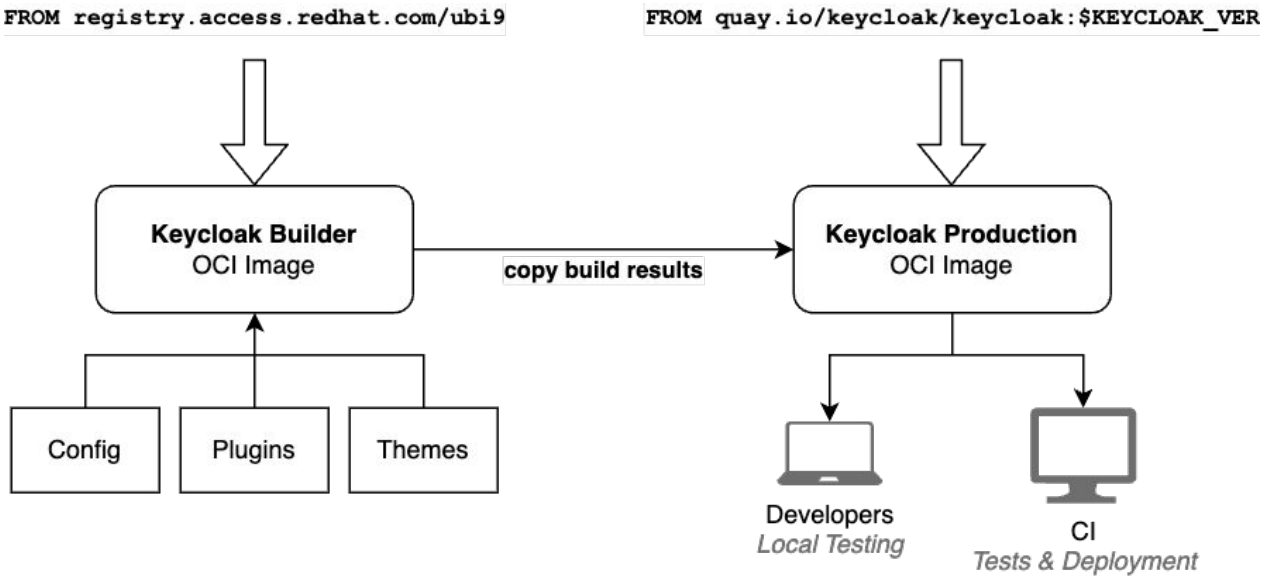
Integration Testing of Keycloak Terraform Modules

How to test Keycloak Terraform modules

sre@finmid.com

How we build Keycloak

to run our tests and deploy to production



Testing strategies we use
to test Keycloak Terraform modules

Integration Tests

- Spin up a production-like setup locally or in the CI.
- Apply the infrastructure code like in production.
- Verify expectations using the Keycloak Admin API.

Browser Tests

- Using Playwright to run tests in a web browser.
- Helps to avoid regression and detect broken flows.
- Especially useful to test changes in login flows.

How we run Keycloak
locally in isolation

Implemented multiple JUnit 5 **test extensions** to run local Keycloak as close as possible to production environment.

JUnit 5

Running set of Testcontainers to **mock** all of the **external dependencies** like APIs of peripheral services.

Testcontainers

Customized PostgreSQL testcontainer which **reflects production** configuration including same extensions.

Database

How we run Keycloak
locally in isolation

APIs of external services or third-parties are mocked using MockServer. Emulating different scenarios helps to test error cases like slow or error responses.

MockServer

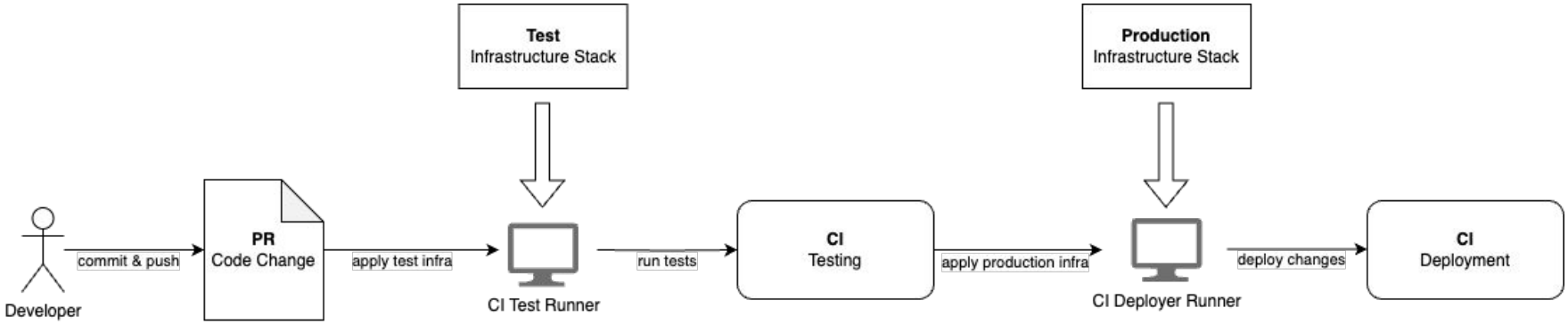
By using the great Terraform provider from Michael Parker, we are configuring all components of Keycloak using code; the same code is used to setup the local env.

Terraform

Terragrunt allows us to workaround common pitfalls of Terraform and manage dependencies between modules easily.

Terragrunt

How we deploy a change
from testing to production



Demo: Test use cases
for application and user authentication

Authenticating Apps

Learn how we test application authentication provisioned by our **service account client module** using the Keycloak Admin API to verify set expectations.

Authenticating Users

Learn how we test user authentication flows provisioned by our OIDC **standard flow client module** leveraging our ephemeral test setup and web browser tests.

How this setup helped us
to run Keycloak with more confidence

- Safe and controllable **interface** to Keycloak **for** everyone, **decoupling** engineers from the infra team
- Provides **regression** testing capabilities using different **testing strategies** while maintaining tests at **low cost**
- Configuration in the code, enables **auditability**, **review** and **rollback** process
- **Predictable upgrades** because of a testing setup that reflects production as much as possible

