

Doing SAP BTP Platform Automation like a boss ...

with Terraform and btp CLI

Experts:

Christian Lechner, Product Manager BTP Platform Automation

Vipin VijayKumar, Development Manager Terraform





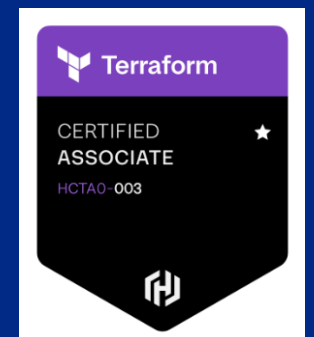
Hi, I am Christian

Product Manager Platform Automation (btp CLI & Terraform)

❤️ Developer Experience

E: christian.lechner@sap.com

L: <https://www.linkedin.com/in/christian-lechner-inthecloud/>





Hi, I am Vipin

Lead for the Terraform provider for SAP BTP. All things Open Source.

E: vipin.vijaykumar@sap.com

L: <https://www.linkedin.com/in/vipin-vijaykumar-0a24a61a/>

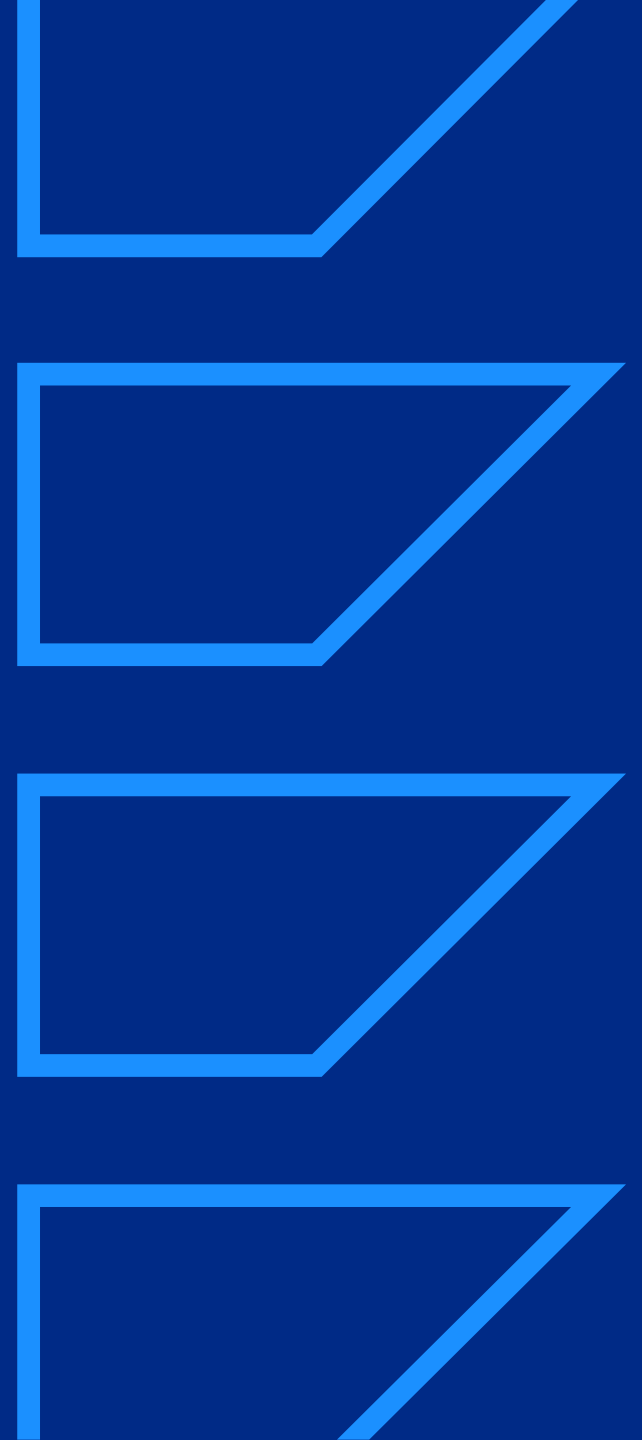
Agenda

01 Why should I do Platform Automation?

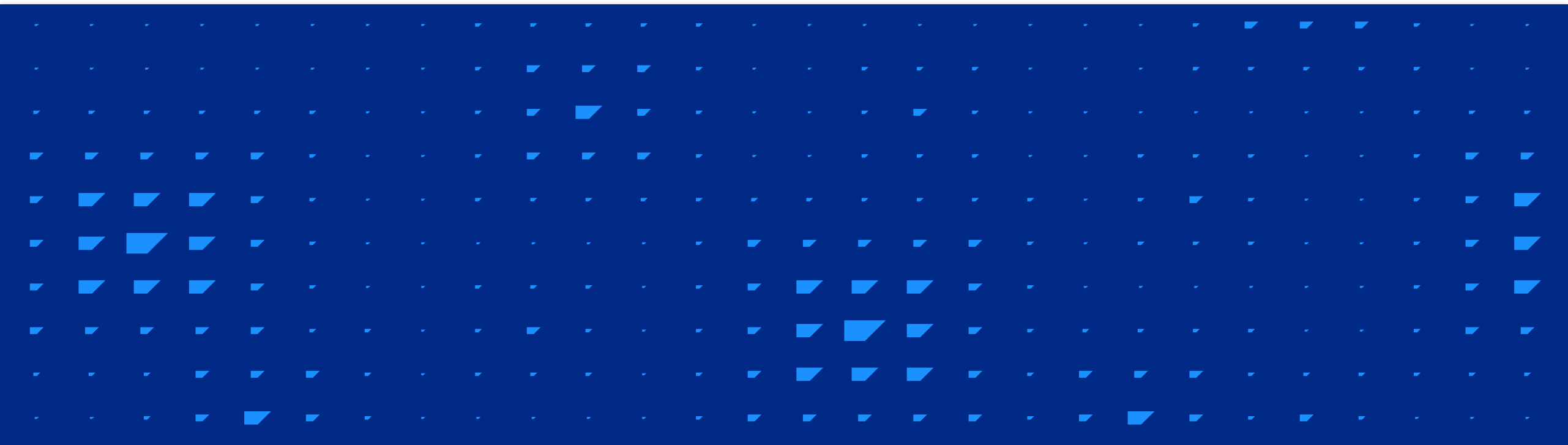
02 Starting fresh aka Greenfield

03 Fifty Shades of Brownfield

04 It's 2026, can AI help me?



Why should I do Platform Automation?



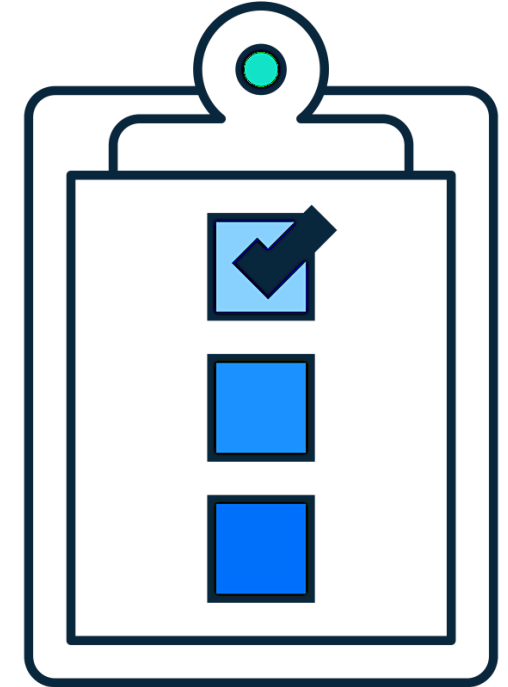
Managing SAP BTP – Status Quo

Manual Runbooks

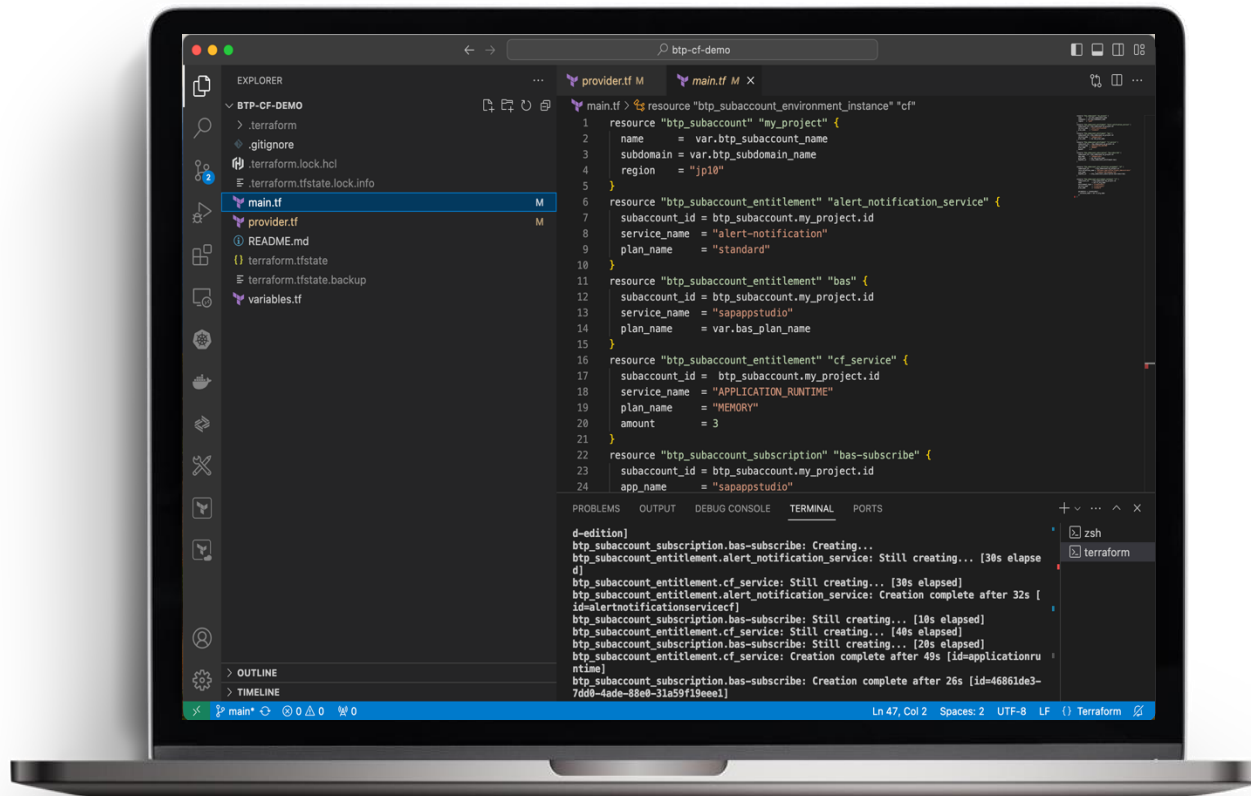
- Word documents
- Wiki pages
- Jira Tickets
- ...

Scripts/One-Off Tools

- Customer specific scripts using CLIs
- API call sequences via tools like Postman or in apps
- ...



What is Terraform?



Terraform as Infrastructure as Code tool enables you to **provision** and **manage** of infrastructure **through code** instead of through manual processes.

Infrastructure is defined **declaratively**

Central storage of **state**

Terraform - Benefits

Consistency & Repeatability

Environments are defined declaratively, eliminating “snowflake” setups and ensuring stages (Dev, QA, Prod) are identical

Compliance & Governance

Codified infrastructure supports policy enforcement, auditability, and security guardrails, helping with compliance

Resilience & Reliability

Automated recovery and consistent deployments reduce downtime, improving user experience and business continuity

Scalability of Operations

A small team can operate large fleets of infrastructure efficiently, freeing talent to focus on innovation rather than firefighting



In addition ... make your developers happy

Provide *self-services* for the development teams to *provision curated setups*

→ Internal Developer Platform

What will we cover?

Terraform Providers

Provider/Tool	Latest Version
Terraform Provider for SAP BTP (docs)	v1.24.0
Terraform Provider for Cloud Foundry (docs)	v1.16.0
Terraform Provider for SAP Cloud Connector (docs)	v1.5.0
Terraform Provider for SAP Cloud Identity Services (docs)	v0.7.0-beta1
Terraform Provider for SAP BTP Services (docs)	v0.1.0-beta1

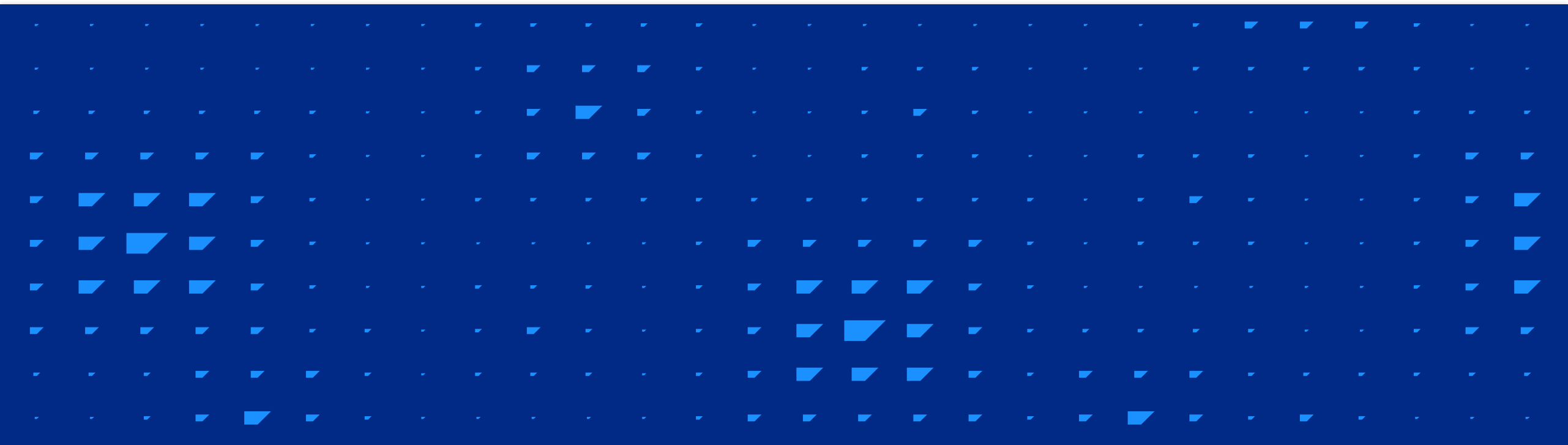
Terraform Modules

Module	Latest Version
Terraform Module for Role Collection Assignments on SAP BTP (docs)	v1.0.1

Tools

Tool	Latest Version
Terraform Exporter for SAP BTP (docs)	v1.6.0

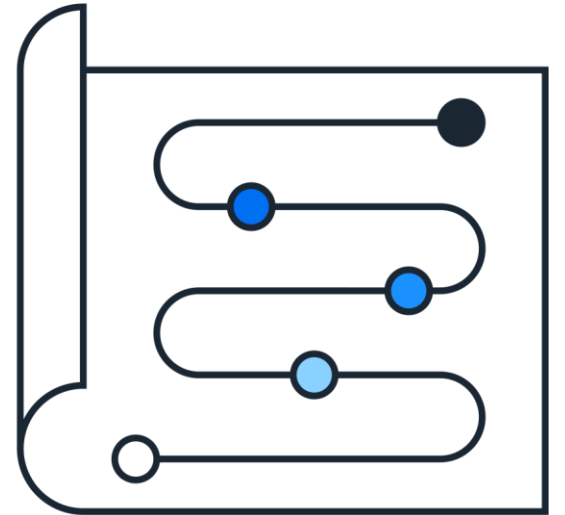
Starting fresh aka Greenfield



Setup is a journey

- Infrastructure provisioning and management is an **iterative process**
- In most cases there **no** “right” or “wrong”
- **Accept** that the setup **will change** – the first try won’t fit forever
- Setup doesn’t need cover **everything**

→ ***Start small and grow as you go***



Don't start with the Terraform Code!

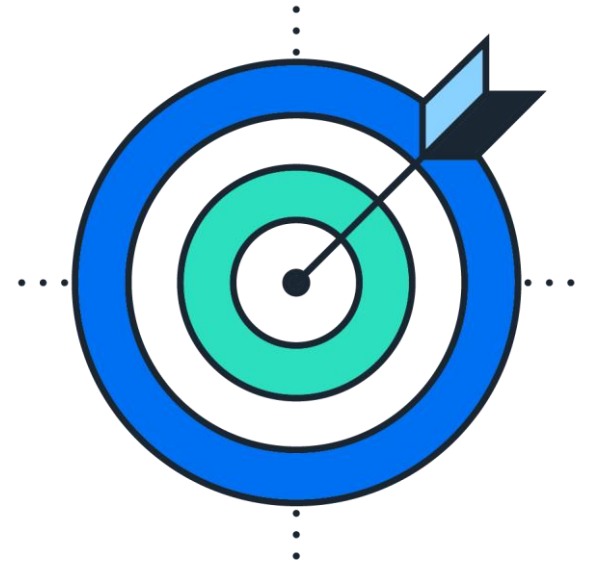
Define your **target picture**

- Who is **responsible** for which parts on SAP BTP?
- Who has which **authorizations** on SAP BTP?
- How should the **account structure** look like?
- **Basics Governance**: naming conventions, labels

→ SAP BTP Administrator's Guide:

<https://help.sap.com/docs/btp/btp-admin-guide/btp-admin-guide>

Samples: <https://github.com/SAP-samples/btp-admin-guide-samples>



Scenario

The **platform team** is responsible for the **subaccounts**, **entitlements**, **trust configuration** and the **environment instance**

Each **subaccount** corresponds to a **stage** (Dev, Test, Prod)

Naming conventions and **labels** are defined (e.g., cost center, project responsible etc.)

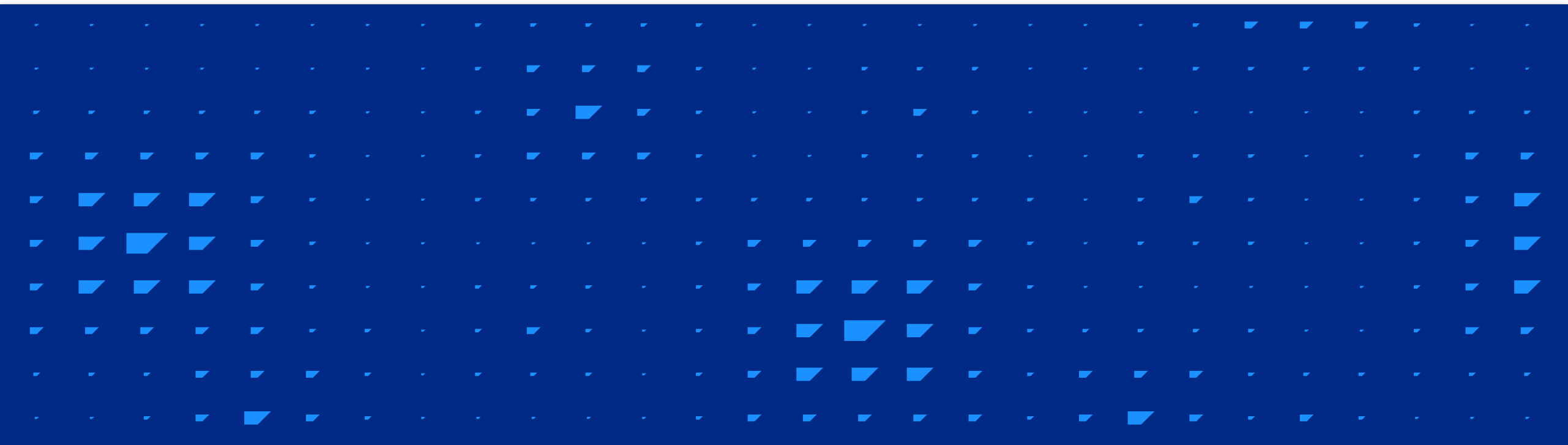
Entitlements are defined per stage, split in required and optional



Let's code



Fifty Shades of Brownfield



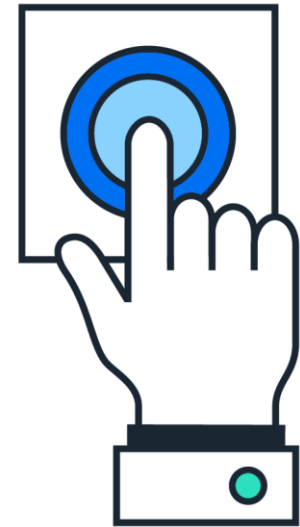
Brownfield Scenario

SAP BTP is already **in use**

The **setup** was mostly done via **ClickOps** (SAP BTP Cockpit) or boosters

Maybe some **accounts** are **partially** managed by Terraform

Where to start?



Import via Terraform

Terraform **standard** functionality

It **works** 😊

No semantic knowledge (explicit dependencies)

Nice enhancement via **Terraform queries**

Import via Terraform - Limitations

Every Terraform provider has **limitations** with regards to imports

Some resources might **not support** import at all due to the underlying API

Some resources might **have restrictions** due to the underlying API

These **limitations are documented** in the provider repositories
(e.g., <https://github.com/SAP/terraform-provider-http/blob/main/guides/IMPORT.md>)

Brownfield - Flow

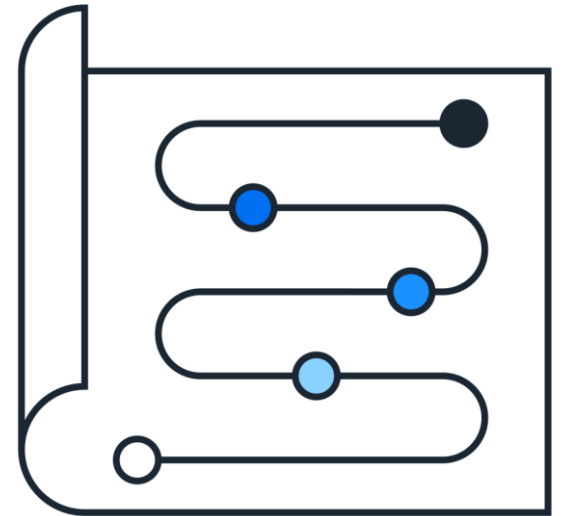
Iterative approach is **king**

Analyze + **Clean Up** (btp CLI): can resources be removed?

In case of *partially managed*: **Determine unmanaged** resources
→ Terraform State + Terraform Query

Import desired resources
→ Terraform Query / Terraform Exporter for SAP BTP

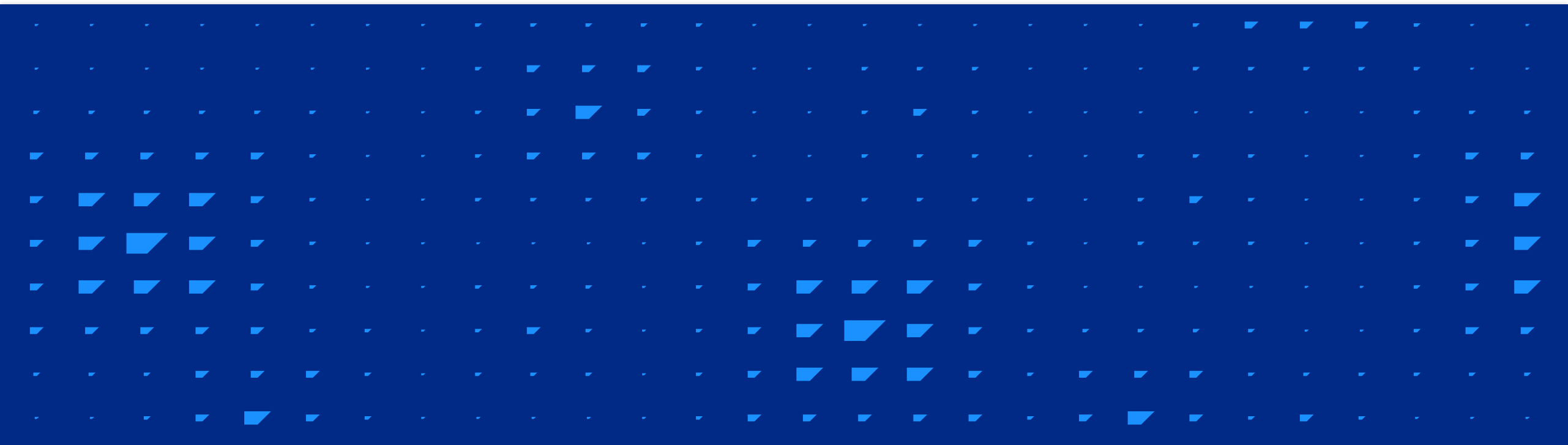
If applicable: **Align**



Let's code



It's 2026, can AI help me?



Before we even start ...

LLMs are fundamentally **probabilistic**, autoregressive **models** that **predict the most likely next token** (word or sub-word). They generate **non-deterministic outputs** by sampling from probability distributions over their vocabulary.

Halucinations are an integral part of the architecture

Although LLMs have a **version**, they are **not “pinned”** even when specified, the I/O parameters are preserved

→ **Keeping this in mind, they can be really useful!**

Before we even start ...

**Ensure automated tests,
guardrails and monitoring
are in place**

Developing Terraform Code

Tools like Claude Code are already good in writing Terraform code

Use **Model Context Protocol** (MCP) Servers

- <https://github.com/hashicorp/terraform-mcp-server>
- <https://github.com/opentofu/opentofu-mcp-server>

Use agent skills (don't forget **evals**)

- <https://github.com/hashicorp/agent-skills>
- <https://github.com/antonbabenko/terraform-skill>

Provide **organization specifics, guidelines, best practices and examples** via markdown files and linting rules



Code Reviews

Integrate agents into **code review flow**

Context is king

Add **skills** to the flow to support the agents

Add **linting** and **tests** to the flow (tflint, terraform test)

Be cautious with auto-approvals – keep a human in the loop



Troubleshooting

Let tools like Claude Code explain results of 'terraform plan', detected drifts

If you run into errors, copy the error to the agent and ask how to fix it

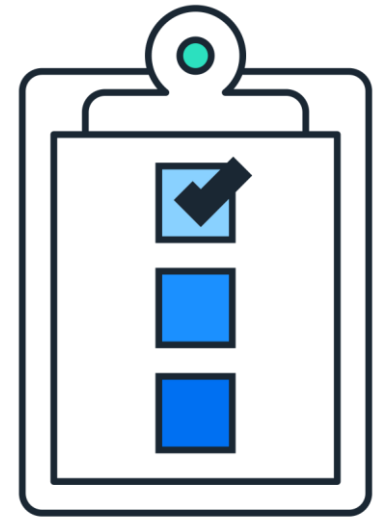
➔ Even if the tools does not find the right solution it might bring in a “perspective” that helps you

ALWAYS make sure to identify the root cause

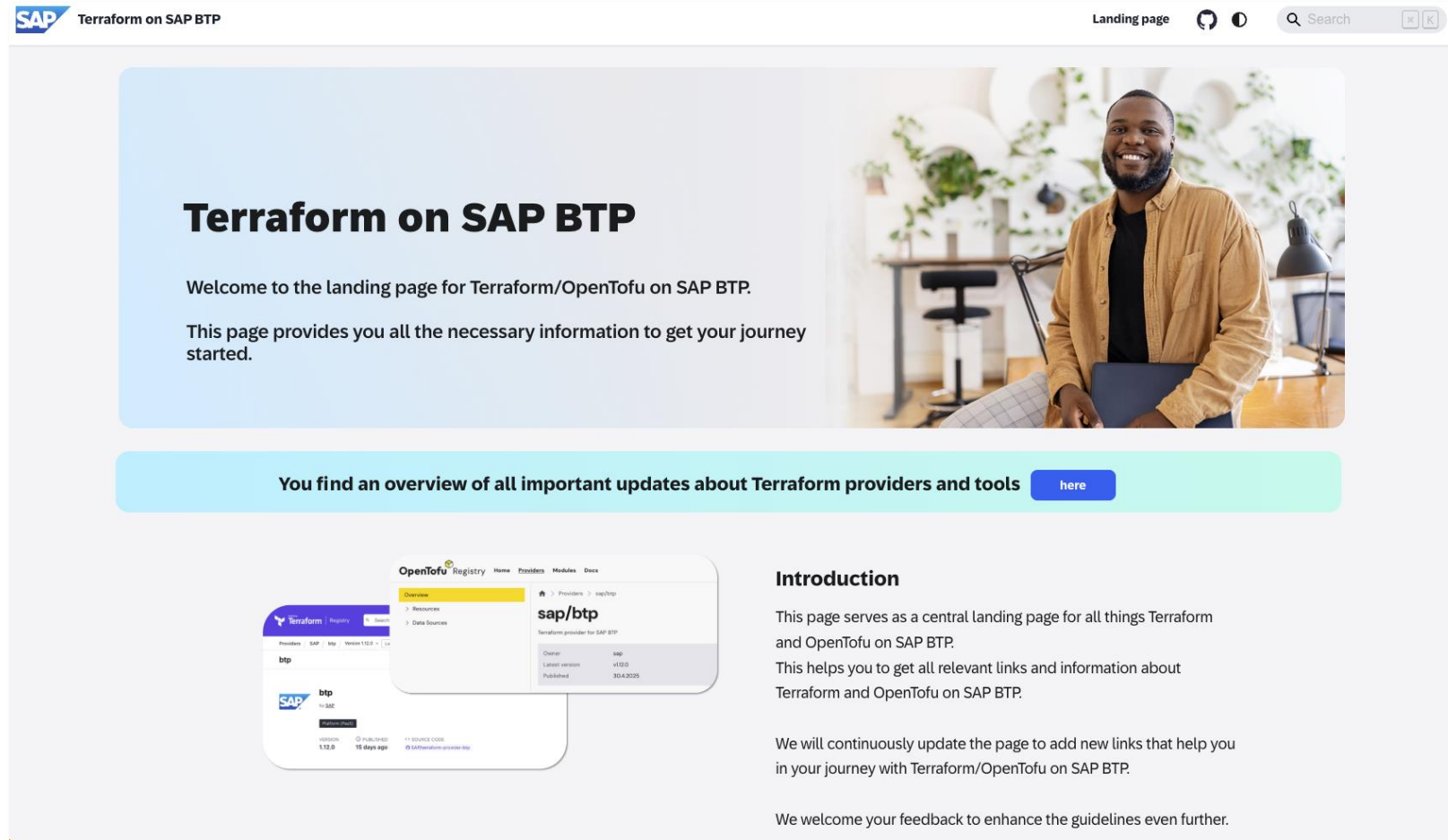


tl;dr

- ✓ **Terraform** and **btp CLI** are powerful tools to automate your platform setup based on industry standards
- ✓ Side effects: **automation** also improves **governance, compliance and developer experience**.
- ✓ No matter **where you are** in your journey, you can get **started with platform automation**
- ✓ **AI** can **support** you, but keep its **shortcomings in mind**
And: it will "just" act as an **amplifier**



Where to get started



The screenshot shows the landing page for Terraform on SAP BTP. The page has a light blue header with the SAP logo and the text "Terraform on SAP BTP". On the right side of the header, there are links for "Landing page", a GitHub icon, and a search bar. The main content area features a large blue box on the left with the title "Terraform on SAP BTP" and a welcome message. To the right of this box is a photograph of a smiling man in a yellow shirt sitting at a desk with a laptop. Below the blue box is a light blue banner with a call to action. At the bottom left, there are two overlapping screenshots of the OpenTofu Registry and the Terraform provider page for SAP BTP. To the right of these screenshots is an "Introduction" section with two paragraphs and a final line of text.

Terraform on SAP BTP

Welcome to the landing page for Terraform/OpenTofu on SAP BTP.

This page provides you all the necessary information to get your journey started.

You find an overview of all important updates about Terraform providers and tools [here](#)

OpenTofu Registry

Providers > **sap/btp**

Overview

Resources

Data Sources

sap/btp

Terraform provider for SAP BTP

Owner: SAP

Latest version: v1.0.0

Published: 30.03.2025

Introduction

This page serves as a central landing page for all things Terraform and OpenTofu on SAP BTP.

This helps you to get all relevant links and information about Terraform and OpenTofu on SAP BTP.

We will continuously update the page to add new links that help you in your journey with Terraform/OpenTofu on SAP BTP.

We welcome your feedback to enhance the guidelines even further.

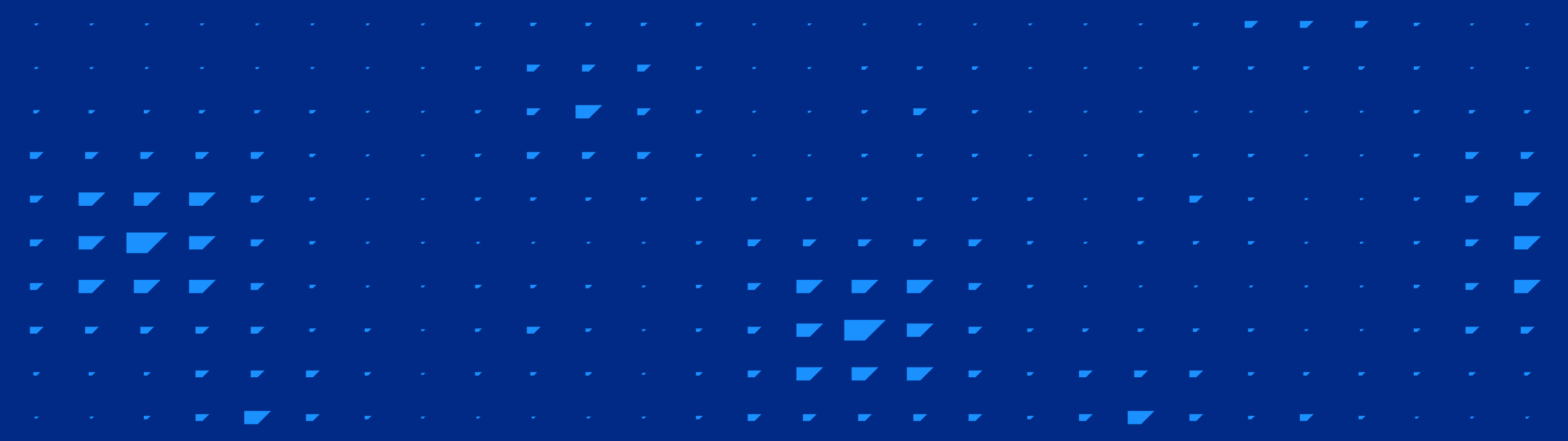
<https://sap-docs.github.io/terraform-landingpage-for-btp/>

Your perspective matters

Join the conversation

You ask. We answer.





Thank you.