Cloud Velocity: Accelerating Innovation through Smarter DevOps | Azure

Intellus Summit 2025

20/05/2025

Ramprakash Balasubramanian

Thomas Neuser

etex inspiring ways of living

Speakers

Ramprakash Balasubramanian

Data platform engineer

Etex Group

Thomas Neuser

Data platform engineer consultant

Intellus Group

etex inspiring ways of living

Agenda

- Introduction on Etex
- As-was situation
- Technical introduction
- Set up the basis
- Acceleration through reusability
- Where are we now?
- Future aspirations

etex inspiring ways of living

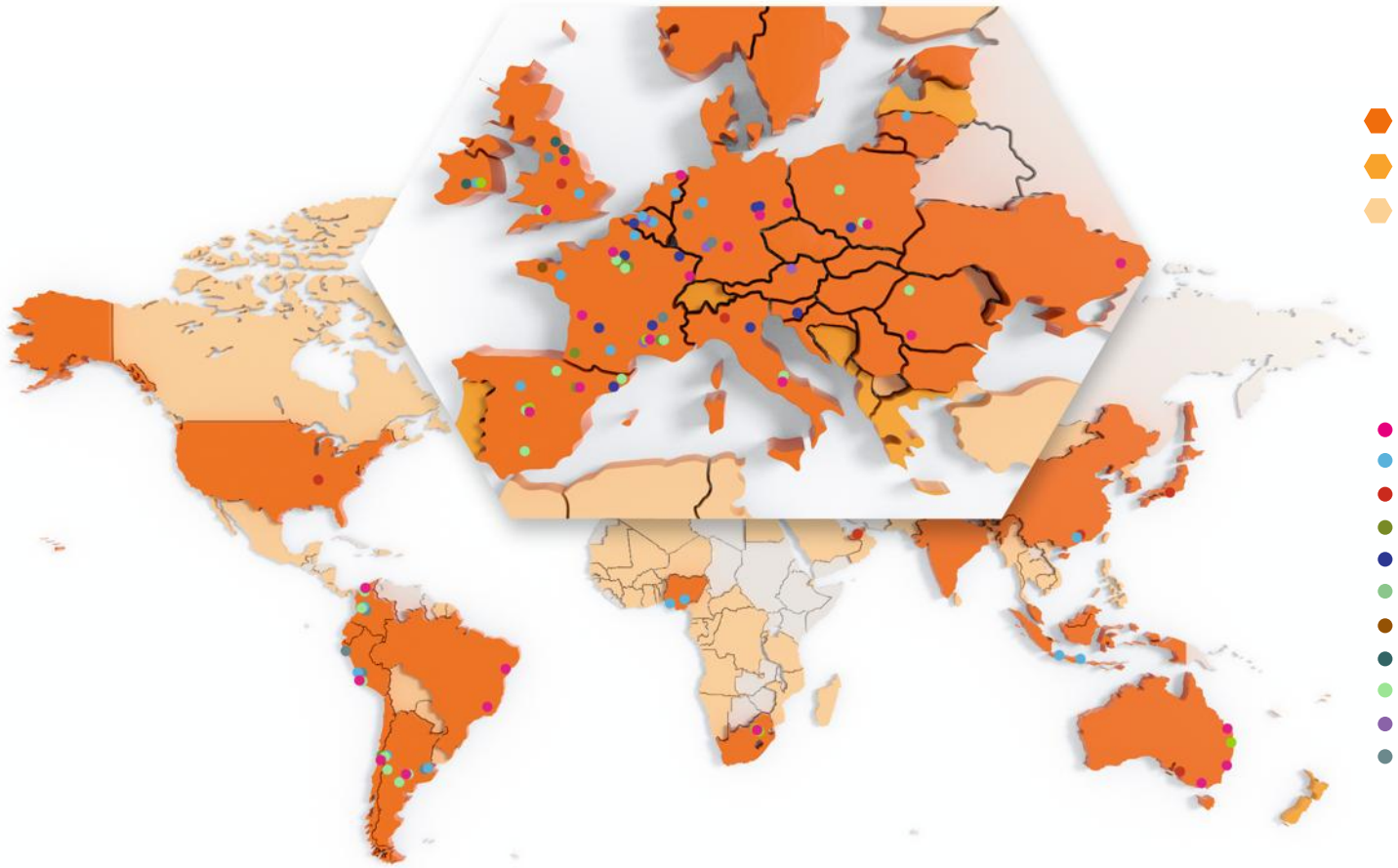
Introduction on Etex

etex inspiring ways of living

etex
inspiring ways
of living

An international industrial company

Selective presence • Local anchorage • Major employer



- Operating country
- Other country with teammates
- Export country

- Plasterboard
- Fibre Cement
- Passive fire Protection & High performance insulation
- Quarries
- Insulation
- Plaster & compounds
- Timber framing
- Steel framing
- Metal profiles
- ITC
- Other

> 13,500
Teammates

45
Operating Countries

> 160
Sites: plants, quarries, offices

5
Innovation & Technology Centres

EUR 3,714
Million revenue

As soon as the Russian invasion of Ukraine started, Etex took a firm position by looking at all options to exit Russia, which is currently ongoing. This was carried out by putting Etex's Russian activities on a standalone basis and stopping fibre cement export activities there.

CEDRAL DURLOCK EQUITONE

Eternit Gyplac Kalsi

PLADUR Promat SINIAT

Superboard URSA

etex inspiring ways of living

Our main brands

Across divisions

Foundation Team Introduction

The Etex MDP Foundational Core Principles

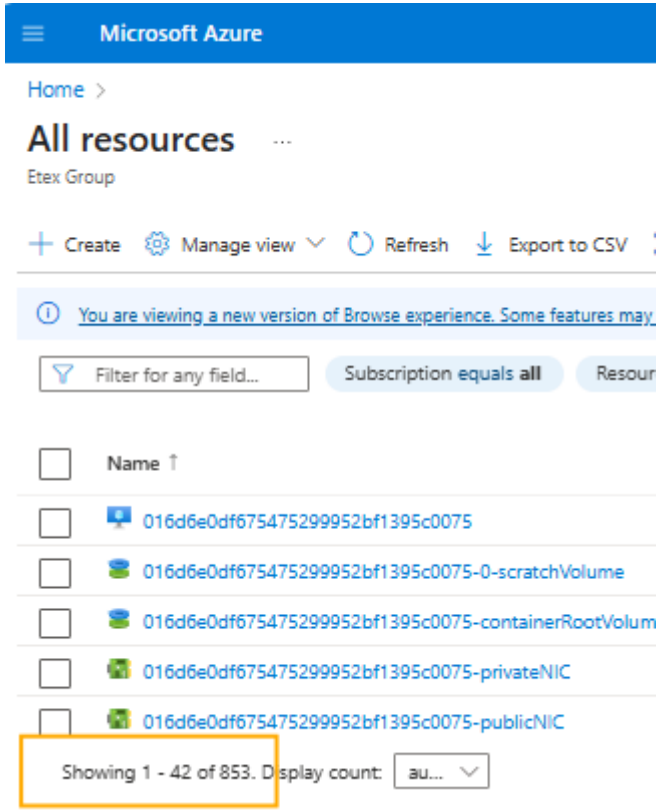
- **Cost-efficient, standardized infrastructure** that prevents duplication and waste
- **A secure, well-governed environment** that ensures compliance and trust
- **Self-service and automation** that enable teams to move faster and focus on insights
- A **manufacturing-driven approach to data** that treats it as a product

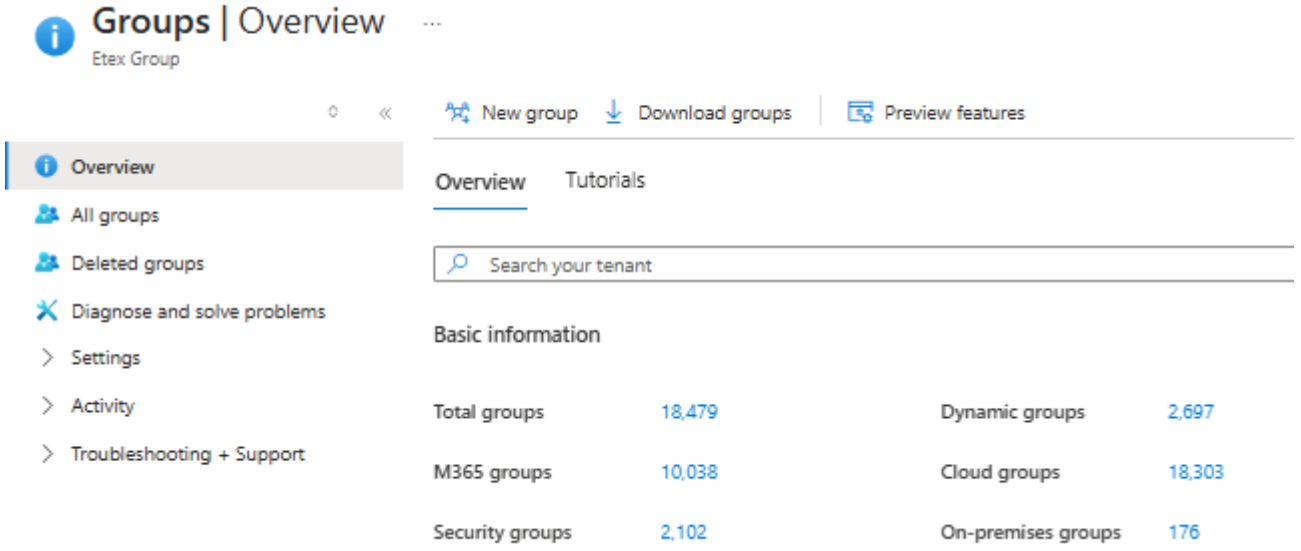
etex inspiring ways of living

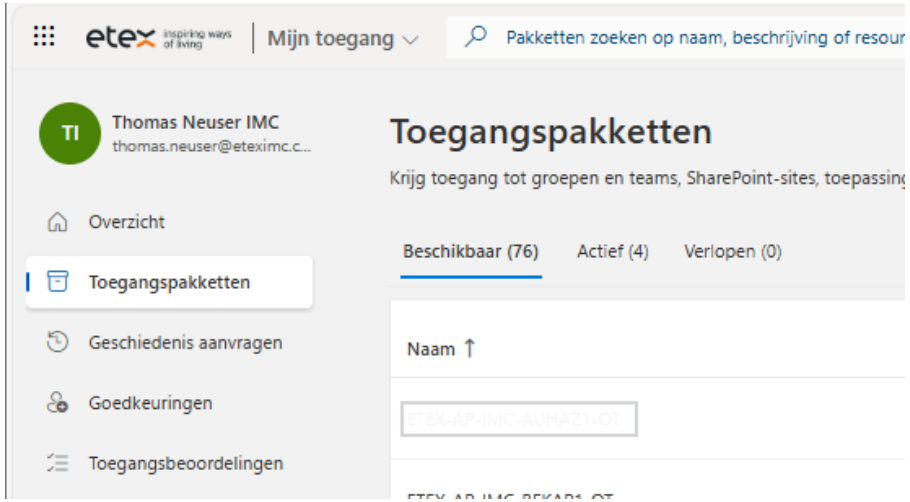
As-was situation

etex inspiring ways of living

No greenfield deployment…







Different Personas

Data Engineer

How can I effectively deploy my data products?

Data Scientist

How can I effectively deploy my Data Science applications?

Analytics Engineer

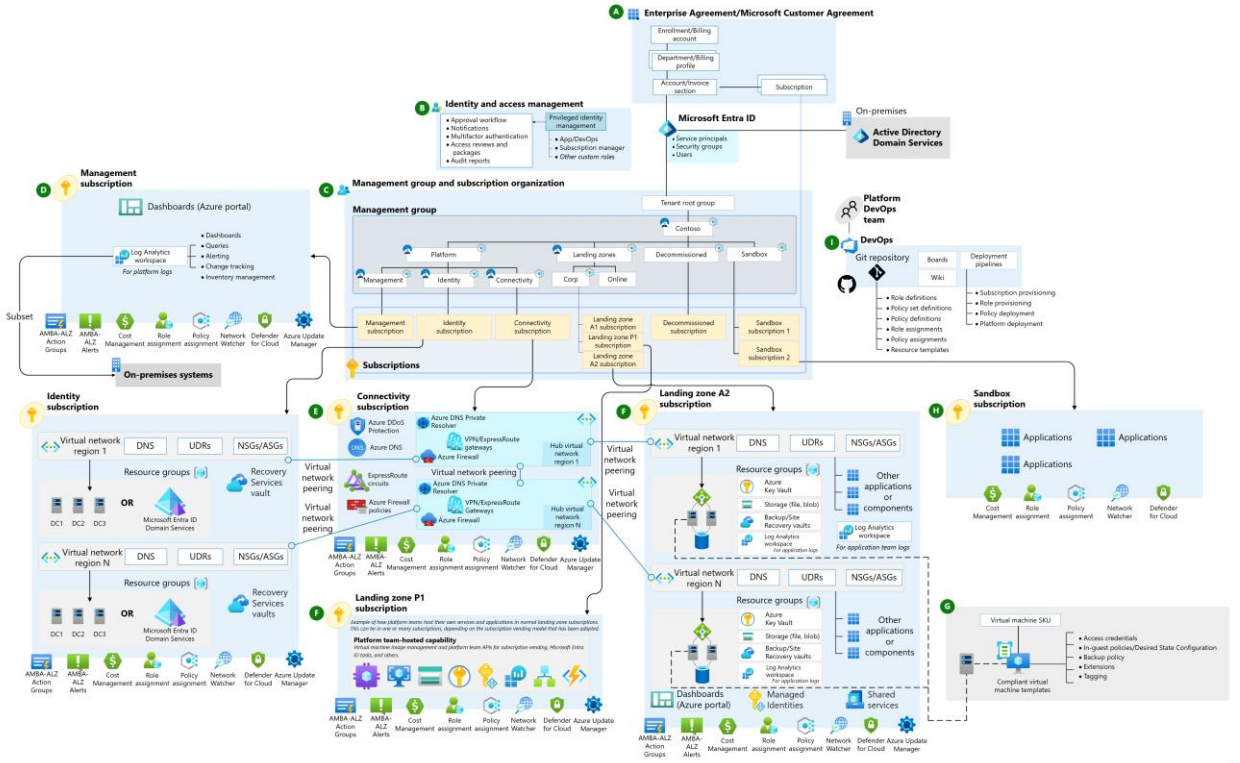
How do I create and deploy a self-service data product?

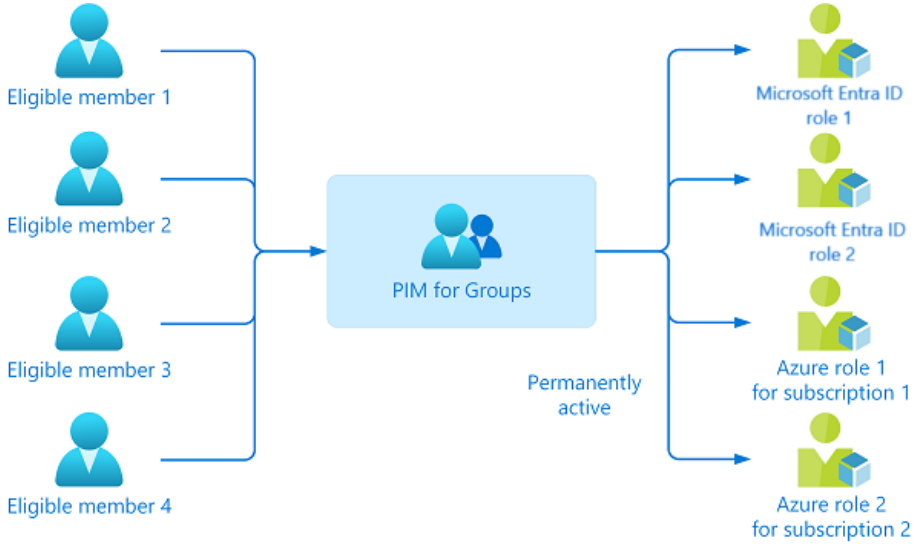
Business Consumer

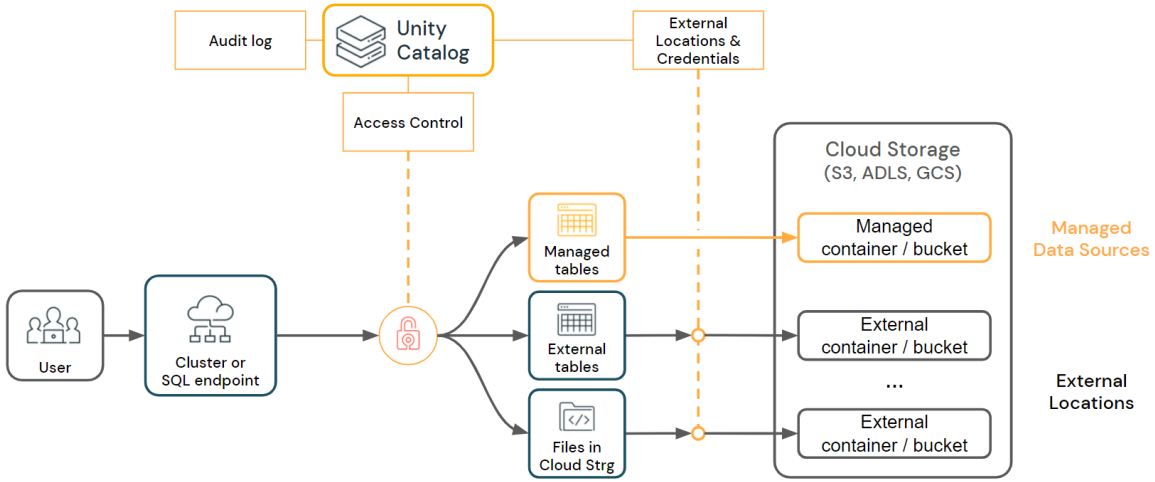
What Data Can I access?
Who do I need to ask for additional access?

etex inspiring ways of living

A lot of ideas…







etex inspiring ways of living

Solution: MDP Framework 2.0

The idea behind MDP framework 2.0 is:

- To streamline the access management
- To avoid duplicate users or groups
- To follow LPA (Least Privilege Access)
- To separate access groups based on the personas, domain and access types
- Get rid of click-ops
- Standardized resource deployment across the various environments

etex inspiring ways of living

Technical introduction

etex inspiring ways of living

Cloud Velocity: Accelerating Innovation through Smarter DevOps

etex inspiring ways of living

Cloud Velocity: Accelerating Innovation through Smarter DevOps

etex inspiring ways of living

Cloud Velocity: Accelerating Innovation through Smarter DevOps

- Cloud platform by Microsoft
- Hosts several services that are used by data engineers/analysts/scientist to perform data operations

etex inspiring ways of living

Cloud Velocity: **Accelerating Innovation** through Smarter DevOps

Terraform

Azure Pipelines

etex inspiring ways of living

Cloud Velocity: **Accelerating Innovation** through Smarter DevOps

- Infrastructure (IaC) framework by HashiCorp
- Multi-cloud/multi-services
- IaC: declarative state of your infra environment within code

etex inspiring ways of living

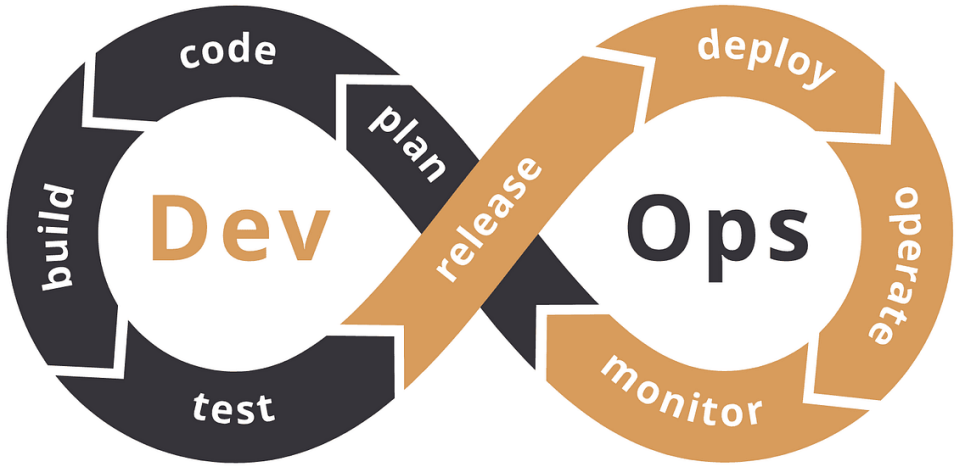
Cloud Velocity: **Accelerating Innovation** through Smarter DevOps

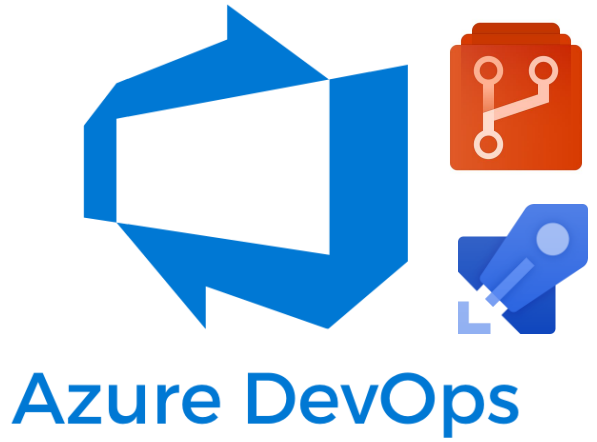
Azure Pipelines

- Part of the Azure DevOps services suite
- Used for automation of validation and deployments
- “EaC”: Everything as Code for reusability

etex inspiring ways of living

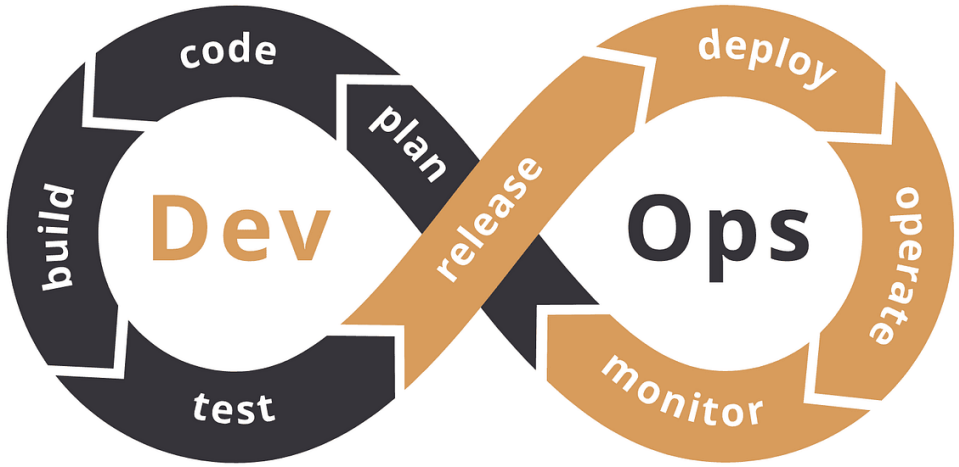
Cloud Velocity: Accelerating Innovation through **Smarter DevOps**





etex inspiring ways of living

Cloud Velocity: Accelerating Innovation through **Smarter DevOps**



- Bridges Development and Operations, enabling faster & more reliable software delivery
- Automated & standardized workflows, consistent integration, testing & deployment

etex inspiring ways of living

Cloud Velocity: Accelerating Innovation through **Smarter DevOps**

- Azure DevOps
- Suite consisting of multiple services for development & project management
- Azure Repos
- Azure Pipelines
- Azure Boards
- Test Plans
- Artifacts

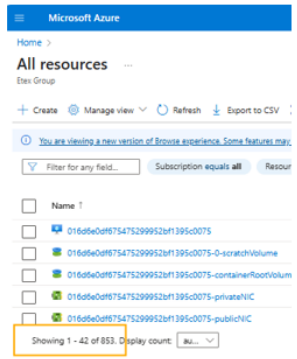
etex inspiring ways of living

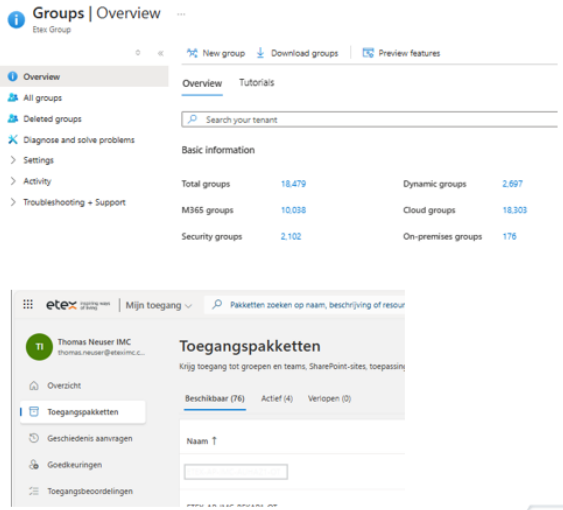
Set the basis

etex inspiring ways of living

Set the basis

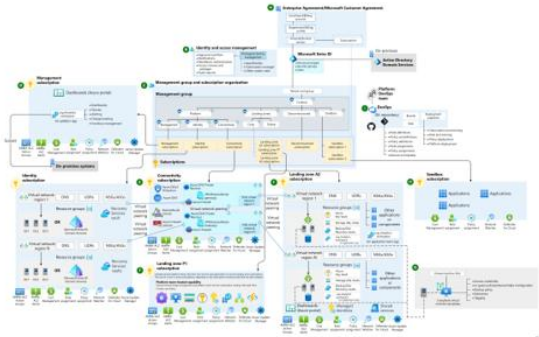
No greenfield deployment…

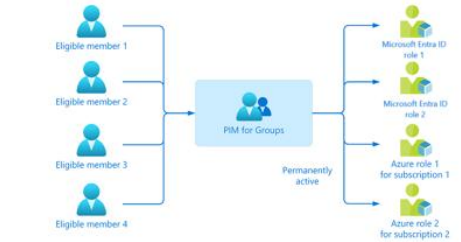


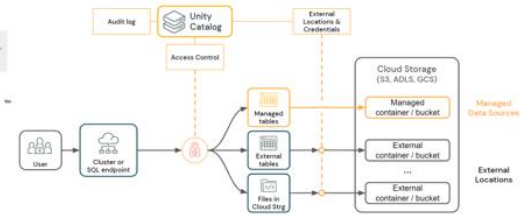


10 I

A lot of ideas…







12 I

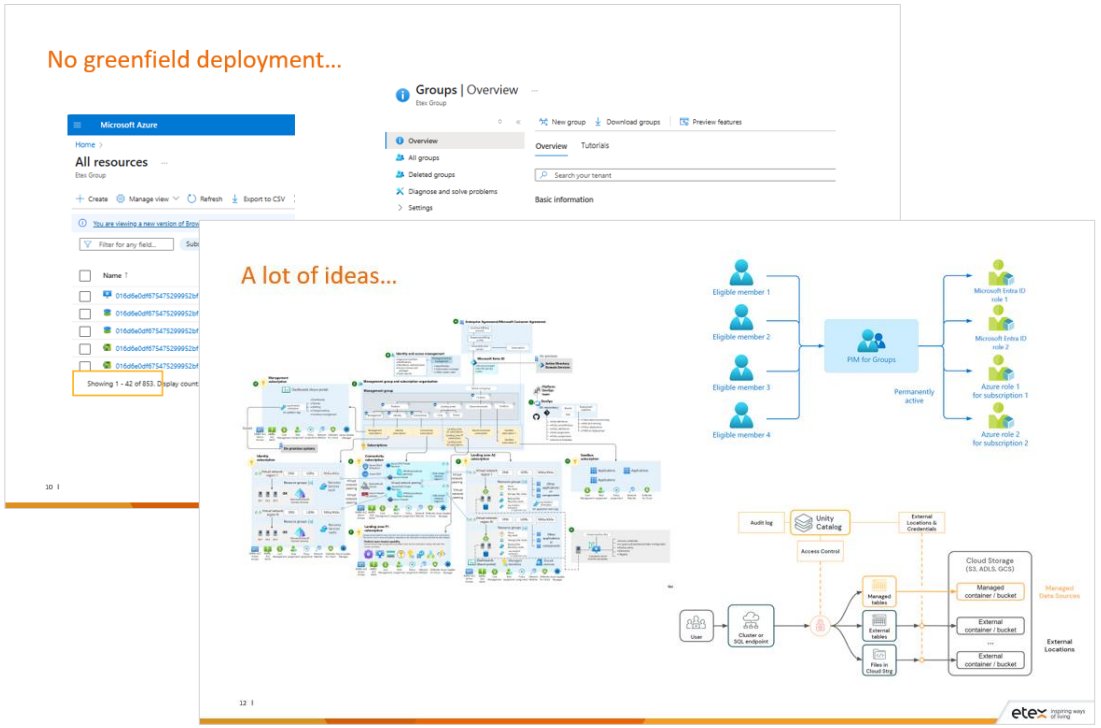
etex inspiring ways of living

Set the basis

Work is needed in different areas with possible technical scope overlap

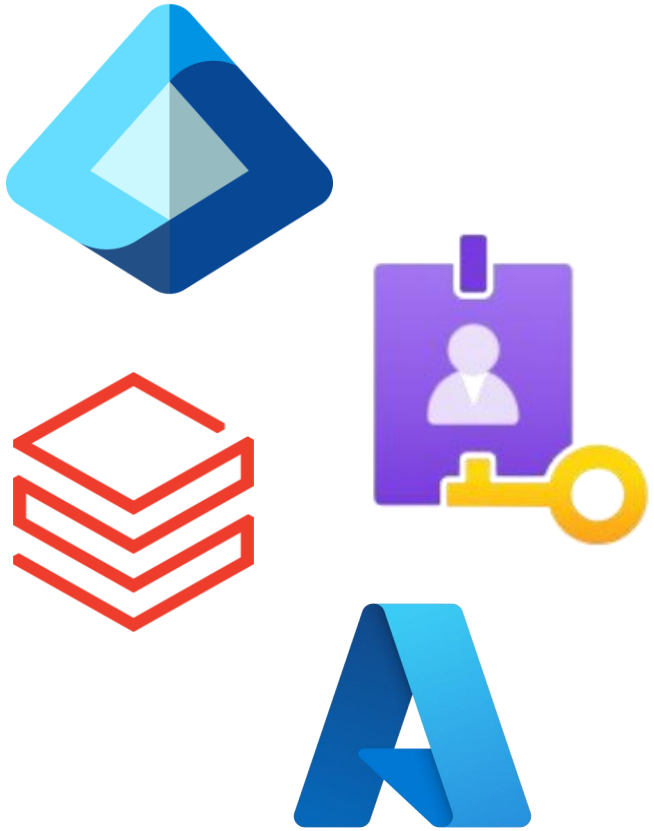
The required work was split up in use cases

To enable/support deployments for those use cases, **a base layer is needed**



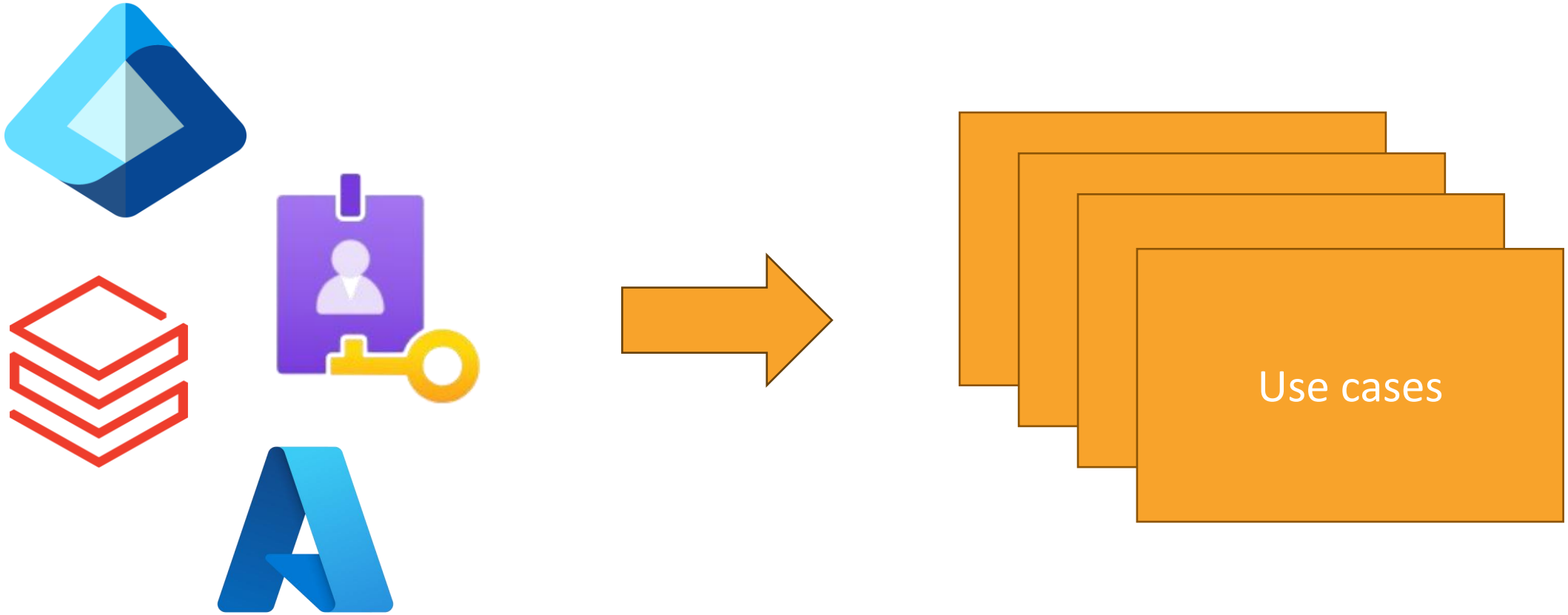
etex inspiring ways of living

Set the basis: conceptual idea



etex inspiring ways of living

Set the basis: conceptual idea



etex inspiring ways of living

Set the basis: conceptual idea

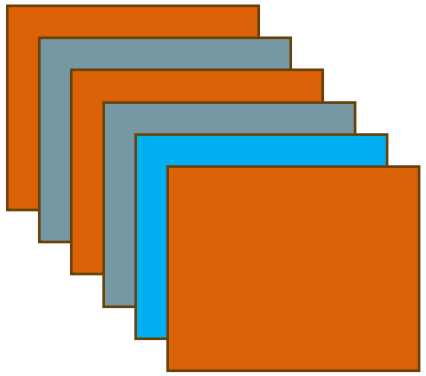
- We need multiple Azure resources across different environments
- We need security groups and policies to manage and govern permissions for different teams
- We need Unity Catalog objects(catalogs, schemas, etc.) for different teams

etex inspiring ways of living

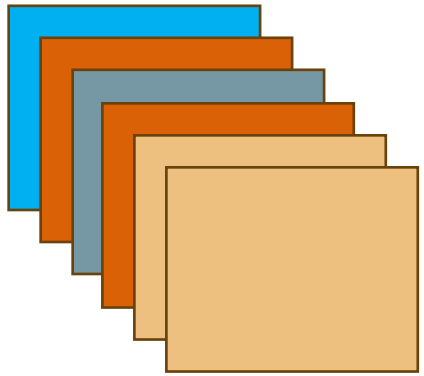
Set the basis: conceptual idea

etex inspiring ways of living

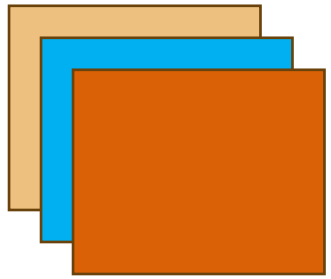
UC 1



UC 2

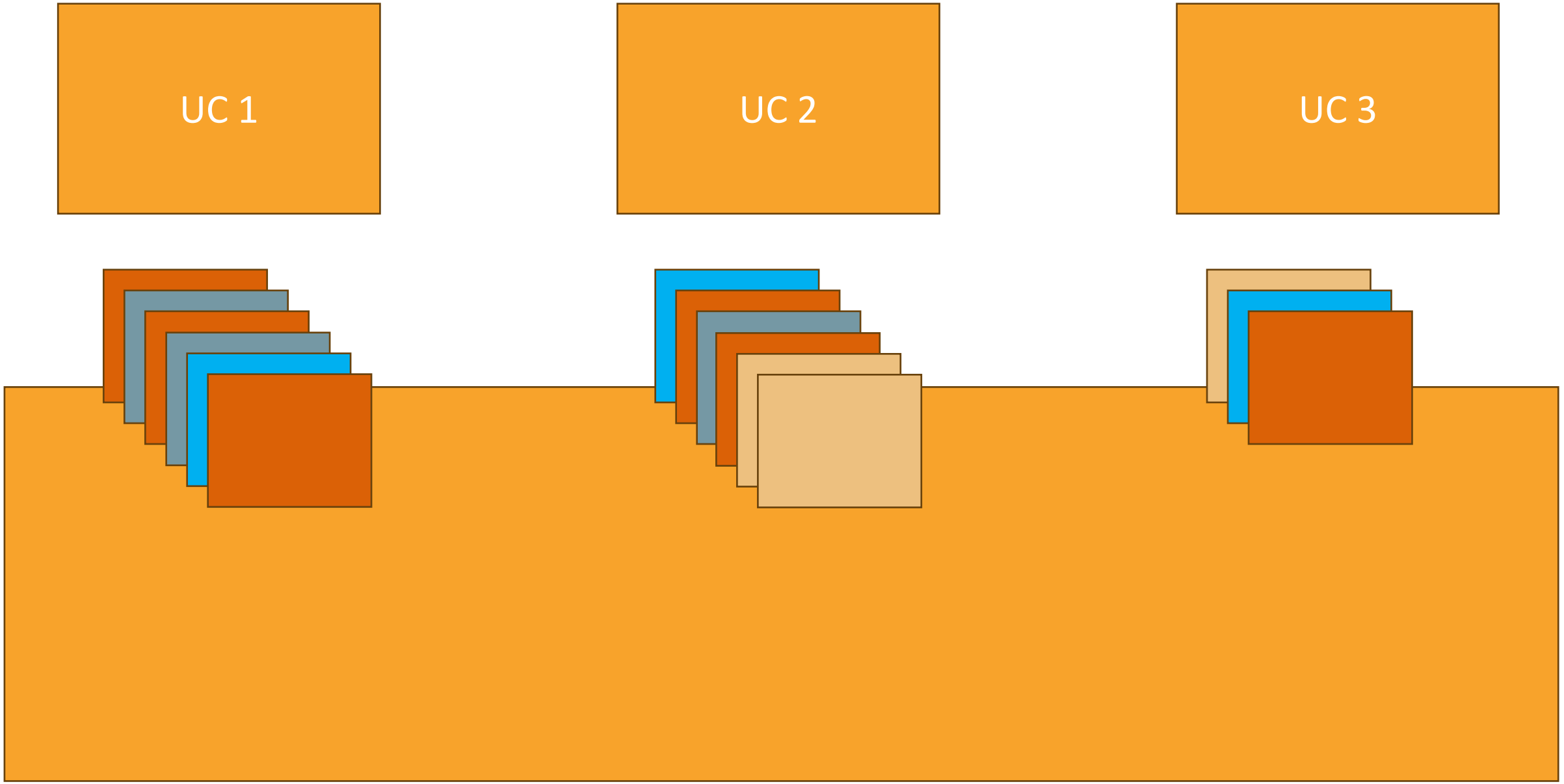


UC 3



Resource within Azure, Databricks, Unity Catalog, Entra ID, …

etex inspiring ways of living



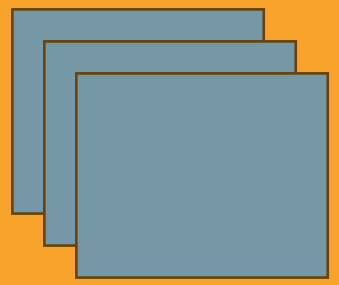
etex inspiring ways of living

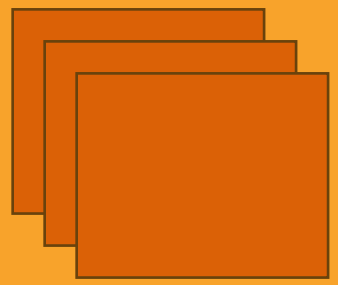
UC 1

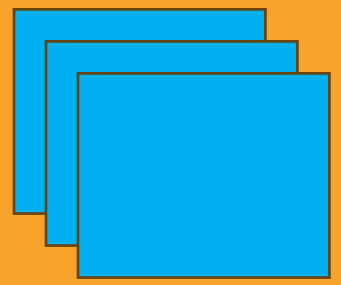
UC 2

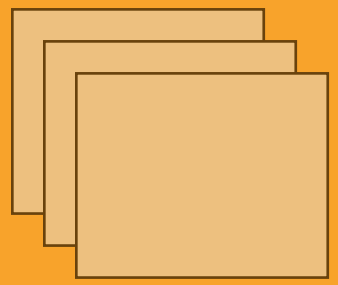
UC 3

Base layer with Terraform modules









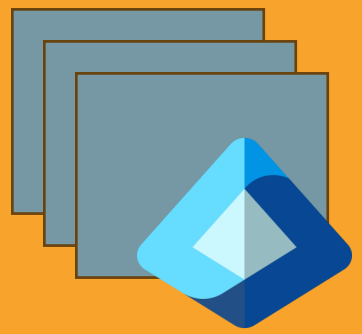
etex inspiring ways of living

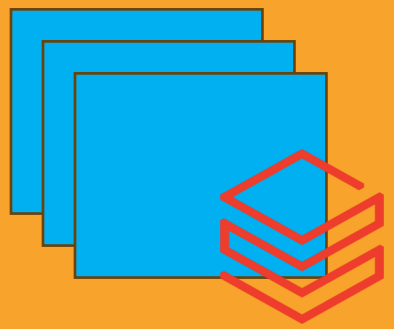
UC 1

UC 2

UC 3

Base layer with Terraform modules





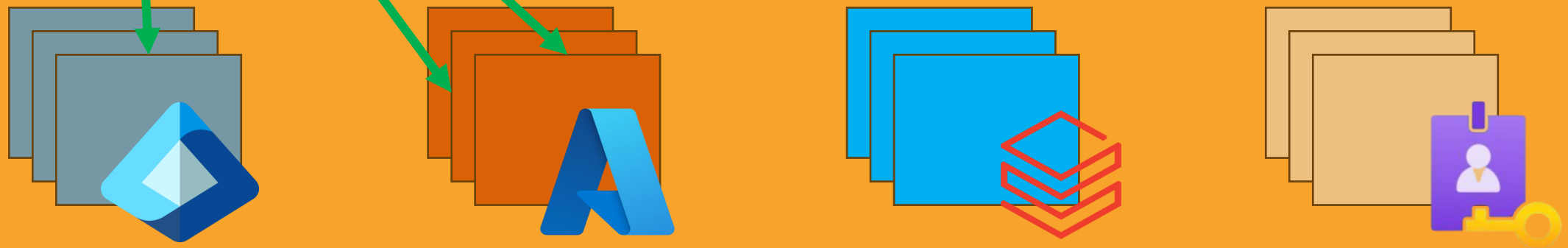
etex inspiring ways of living

UC 1

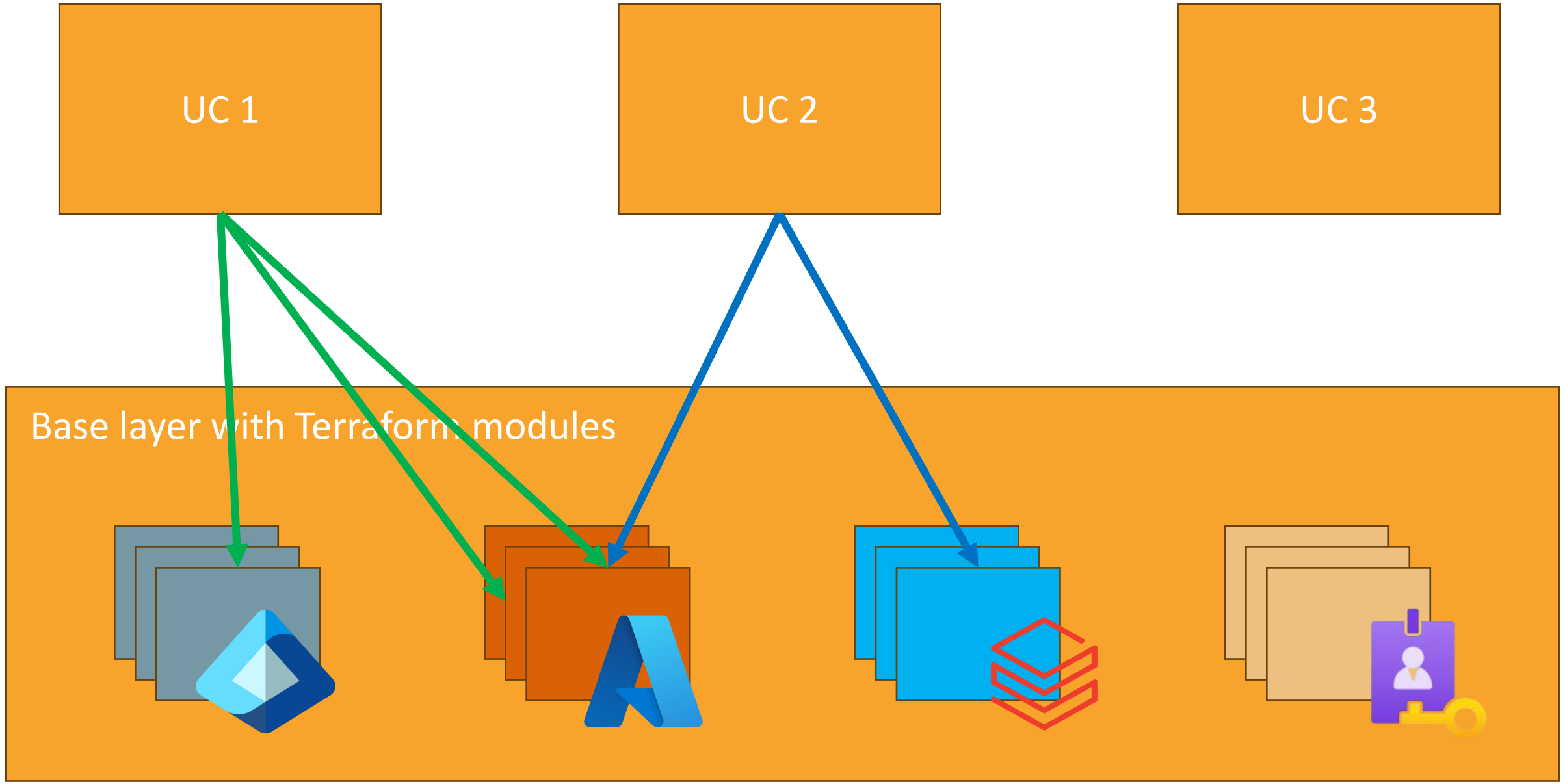
UC 2

UC 3

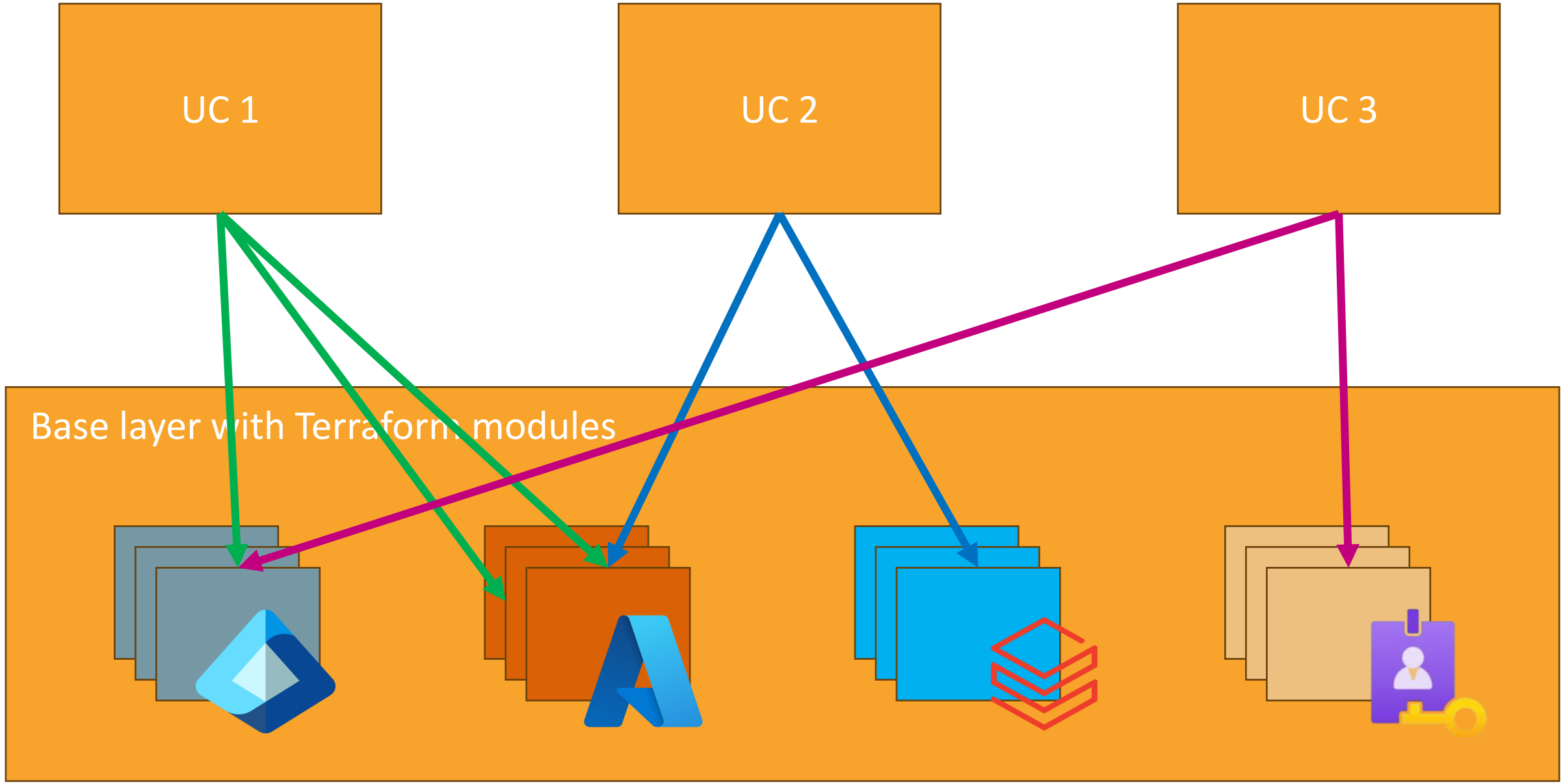
Base layer with Terraform modules



etex inspiring ways of living

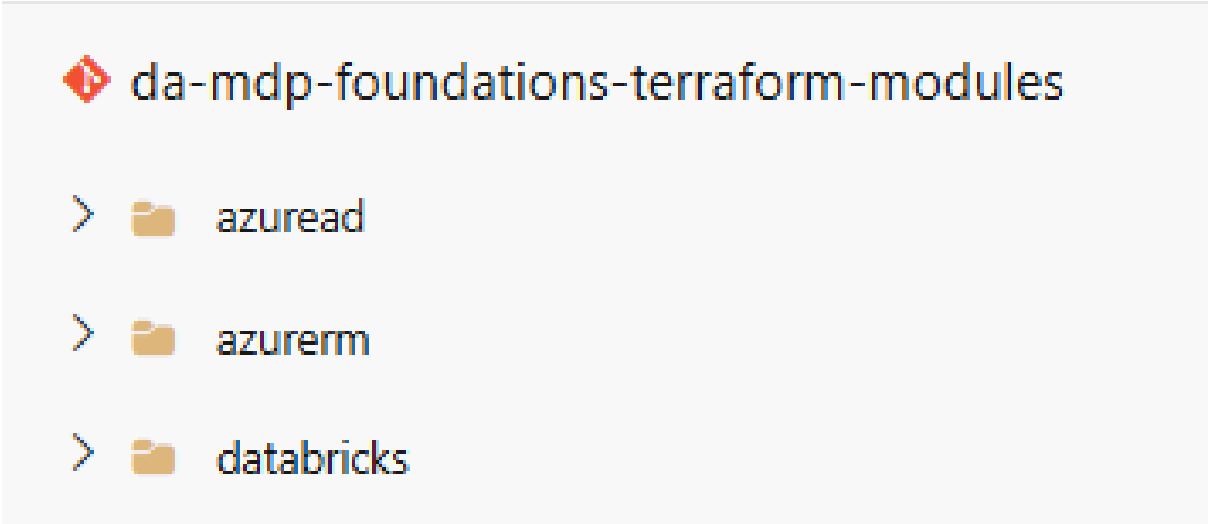


etex inspiring ways of living



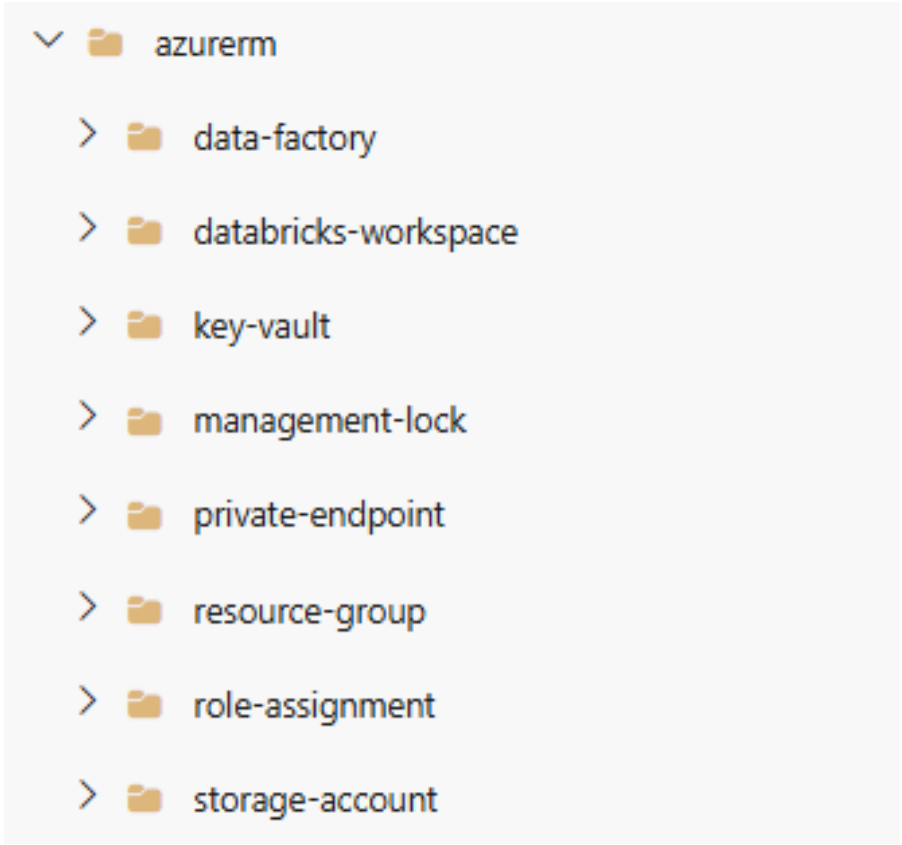
etex inspiring ways of living

In practice



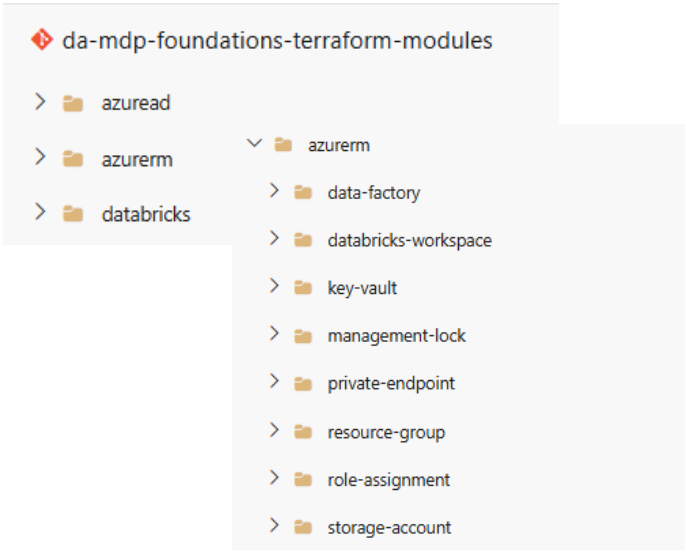
etex inspiring ways of living

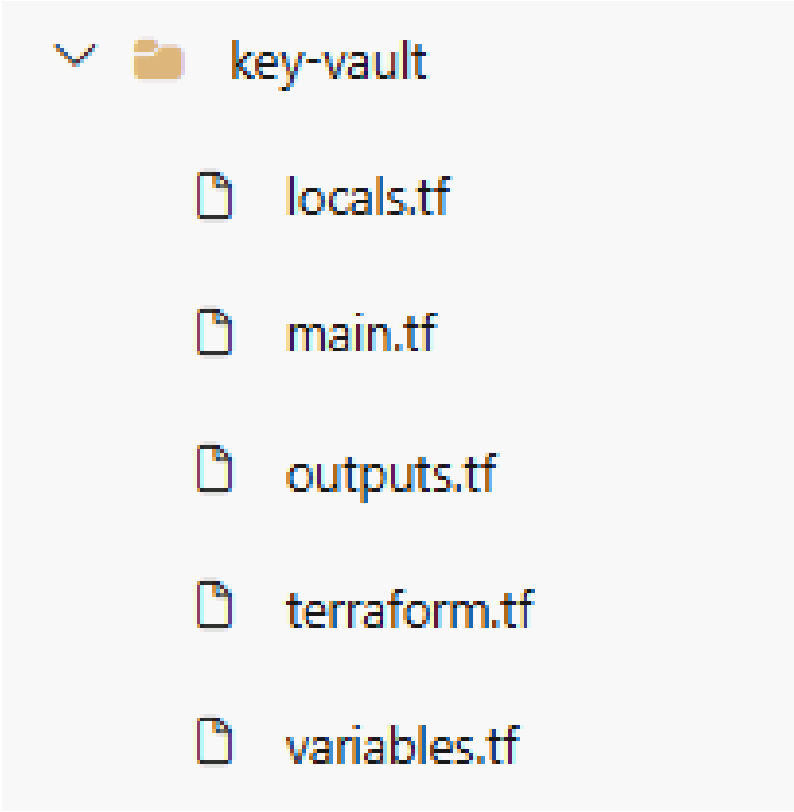
In practice



etex inspiring ways of living

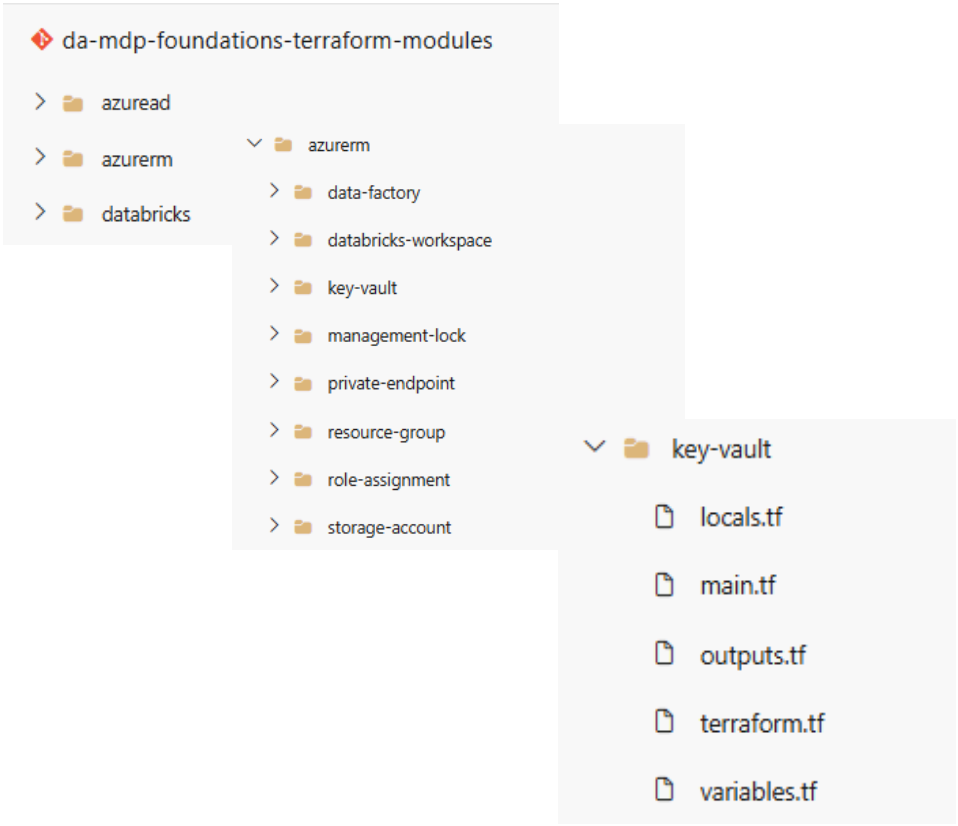
In practice





etex inspiring ways of living

In practice




```hcl
resource "azurerm_key_vault" "key_vault" {
  name                       = var.name
  location                   = var.location
  resource_group_name        = var.resource_group_name
  tenant_id                  = var.tenant_id
  purge_protection_enabled = var.purge_protection_enabled

  sku_name                       = var.sku_name
  public_network_access_enabled = var.public_network_access_enabled
  enable_rbac_authorization     = var.enable_rbac_authorization

  dynamic "network_acls" {
    for_each = var.default_action_kv_fw == "Deny" ? [1] : []
    content {
      bypass                     = var.bypass_kv_fw
      default_action             = var.default_action_kv_fw
      ip_rules                   = var.ip_rules_fw
      virtual_network_subnet_ids = var.subnet_ids_fw
    }
  }
}

resource "azurerm_key_vault_access_policy" "access_policy" {
  count                   = length(var.access_policies)
  key_vault_id            = azurerm_key_vault.key_vault.id
  tenant_id               = var.access_policies[count.index].tenant_id
  object_id               = var.access_policies[count.index].object_id
  secret_permissions      = var.access_policies[count.index].secret_permissions
  key_permissions         = var.access_policies[count.index].key_permissions
  certificate_permissions = var.access_policies[count.index].certificate_permissions
}

resource "azurerm_key_vault_secret" "secret" {
  count        = length(var.secrets)
  name         = var.secrets[count.index].name
  value        = var.secrets[count.index].placeholder
  key_vault_id = azurerm_key_vault.key_vault.id
  lifecycle {
    ignore_changes = [value]
  }
}

```


etex inspiring ways of living

Module versioning

- Modules are used by different use cases at the same time
- Still, development is possible when new module features are needed for a certain use case
- Use case need to refer to a certain instance of the module, so that its use remains stable

→ Module **versioning** is needed

etex inspiring ways of living

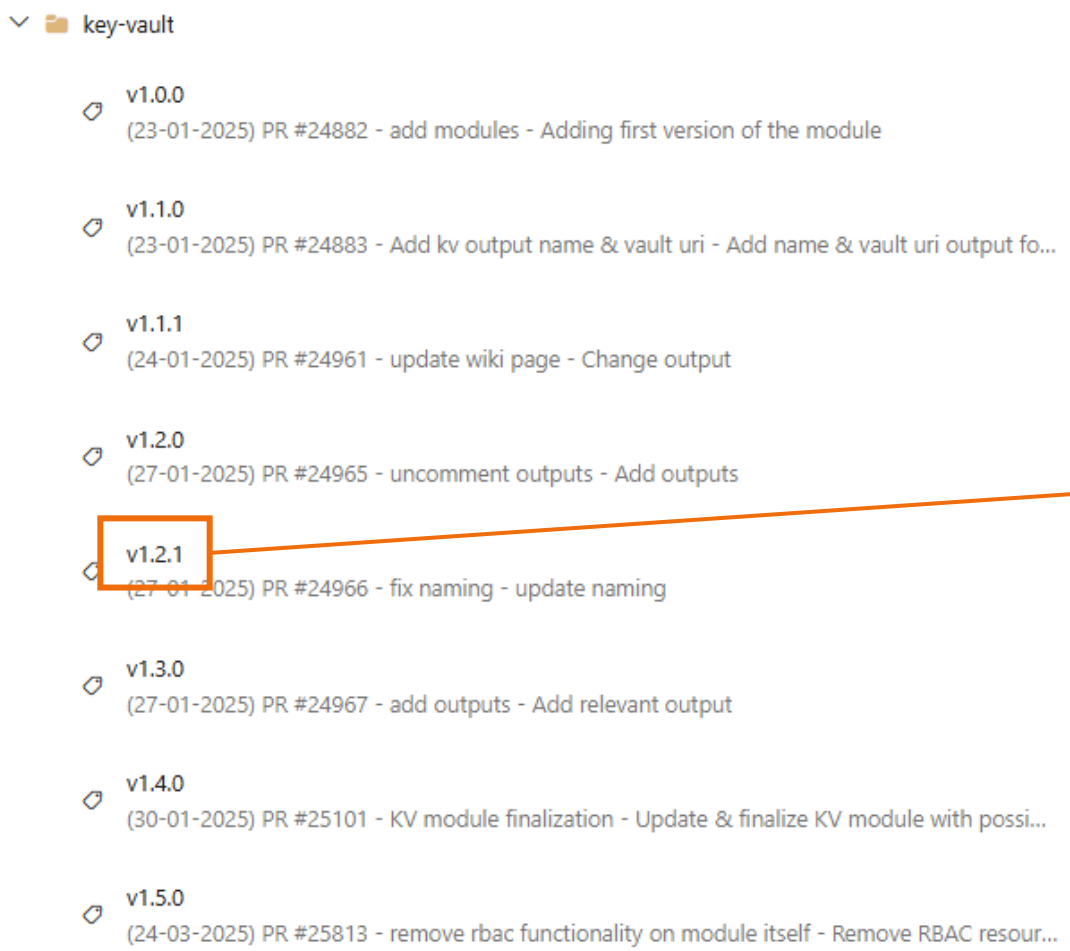
Module versioning

key-vault

- **v1.0.0**
  (23-01-2025) PR #24882 - add modules - Adding first version of the module
- **v1.1.0**
  (23-01-2025) PR #24883 - Add kv output name & vault uri - Add name & vault uri output fo...
- **v1.1.1**
  (24-01-2025) PR #24961 - update wiki page - Change output
- **v1.2.0**
  (27-01-2025) PR #24965 - uncomment outputs - Add outputs
- **v1.2.1**
  (27-01-2025) PR #24966 - fix naming - update naming
- **v1.3.0**
  (27-01-2025) PR #24967 - add outputs - Add relevant output
- **v1.4.0**
  (30-01-2025) PR #25101 - KV module finalization - Update & finalize KV module with possi...
- **v1.5.0**
  (24-03-2025) PR #25813 - remove rbac functionality on module itself - Remove RBAC resour...

etex inspiring ways of living

Module versioning



Semantic versioning:
v[major].[minor].[patch]

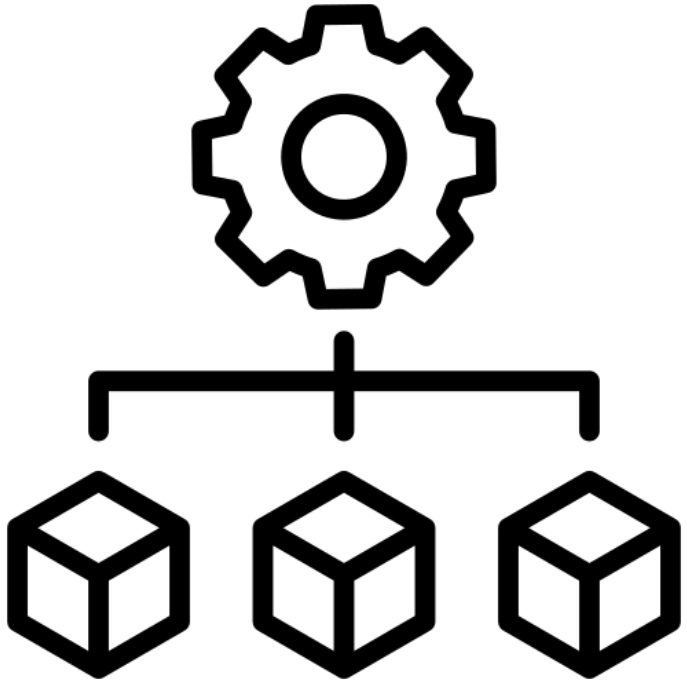
etex inspiring ways of living

Module versioning

```
module "mdp_key_vaults" {
  for_each            = local.input_map_mdp_key_vaults
  source              = "git::https://etexit.visualstudio.com/Modern%20data%20platform/_git/da-mdp-foundations-terraform-modules?ref=azurerm/key-vault/v1.5.0"
  name                = each.key
```

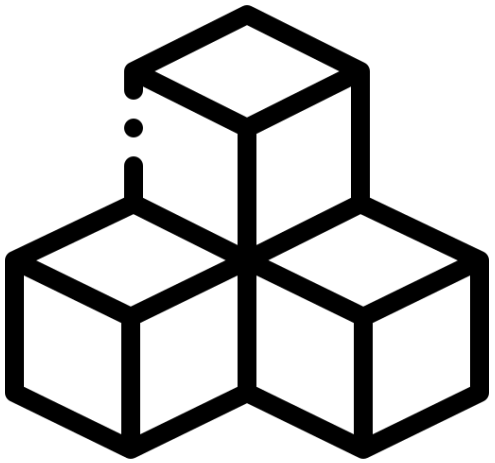
etex inspiring ways of living

Automation for modules



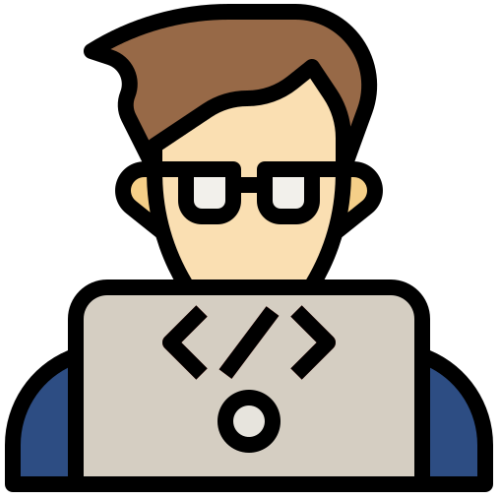
etex inspiring ways of living

Automation for modules



etex inspiring ways of living

Automation for modules



etex inspiring ways of living

Automation for modules

etex inspiring ways of living

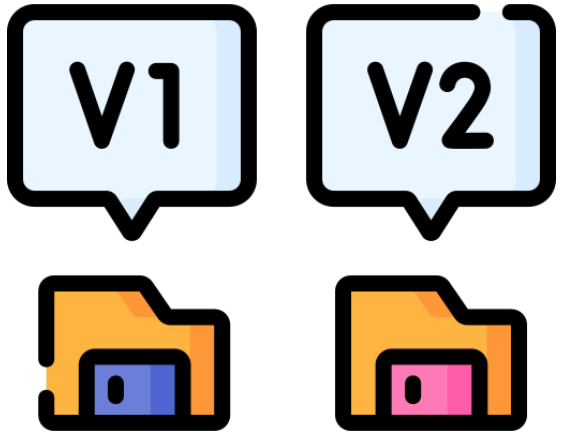
Automation for modules

etex inspiring ways of living

Automation for modules

etex inspiring ways of living

Automation for modules



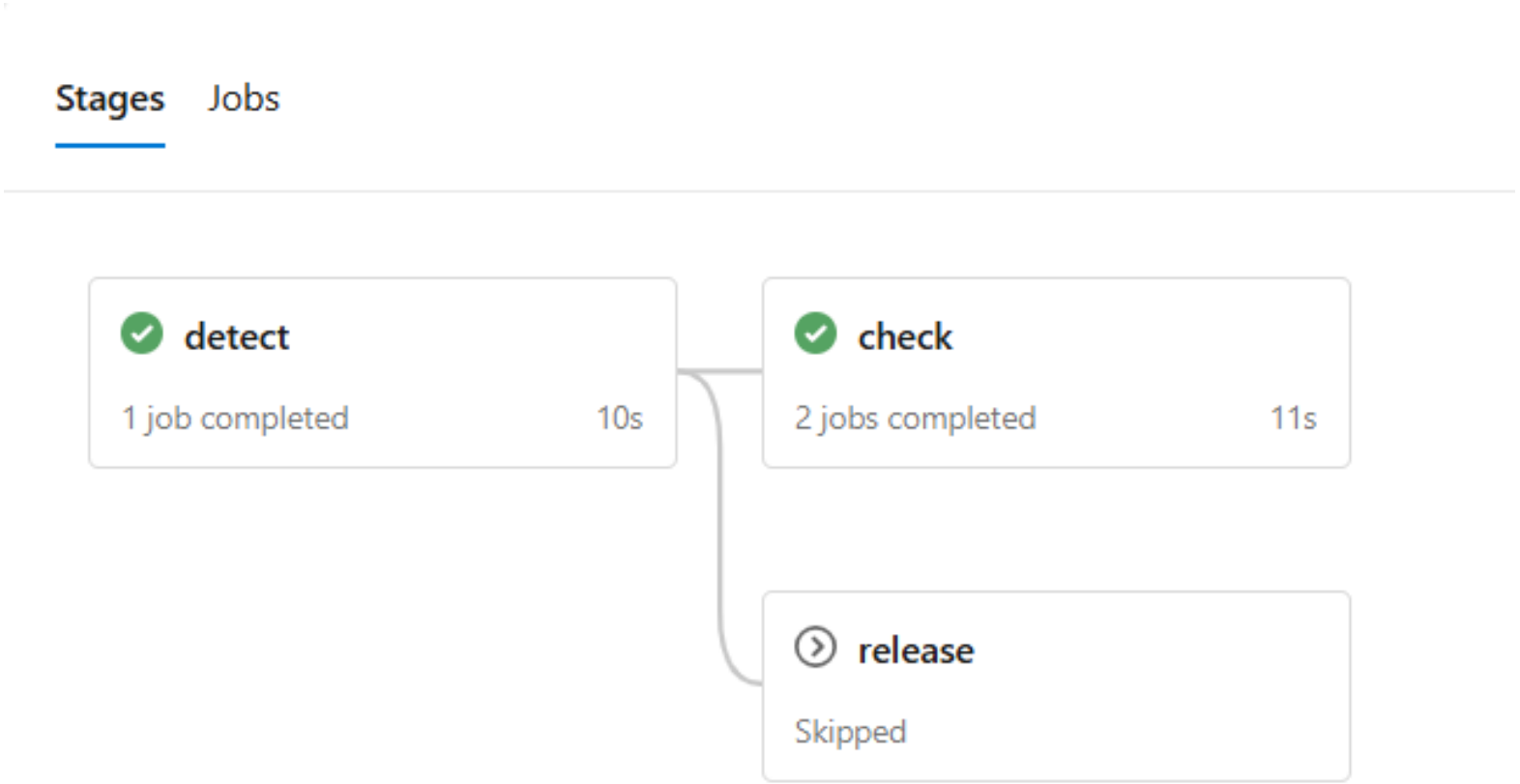
etex inspiring ways of living

Automation for modules

etex inspiring ways of living

Module versioning pipelines

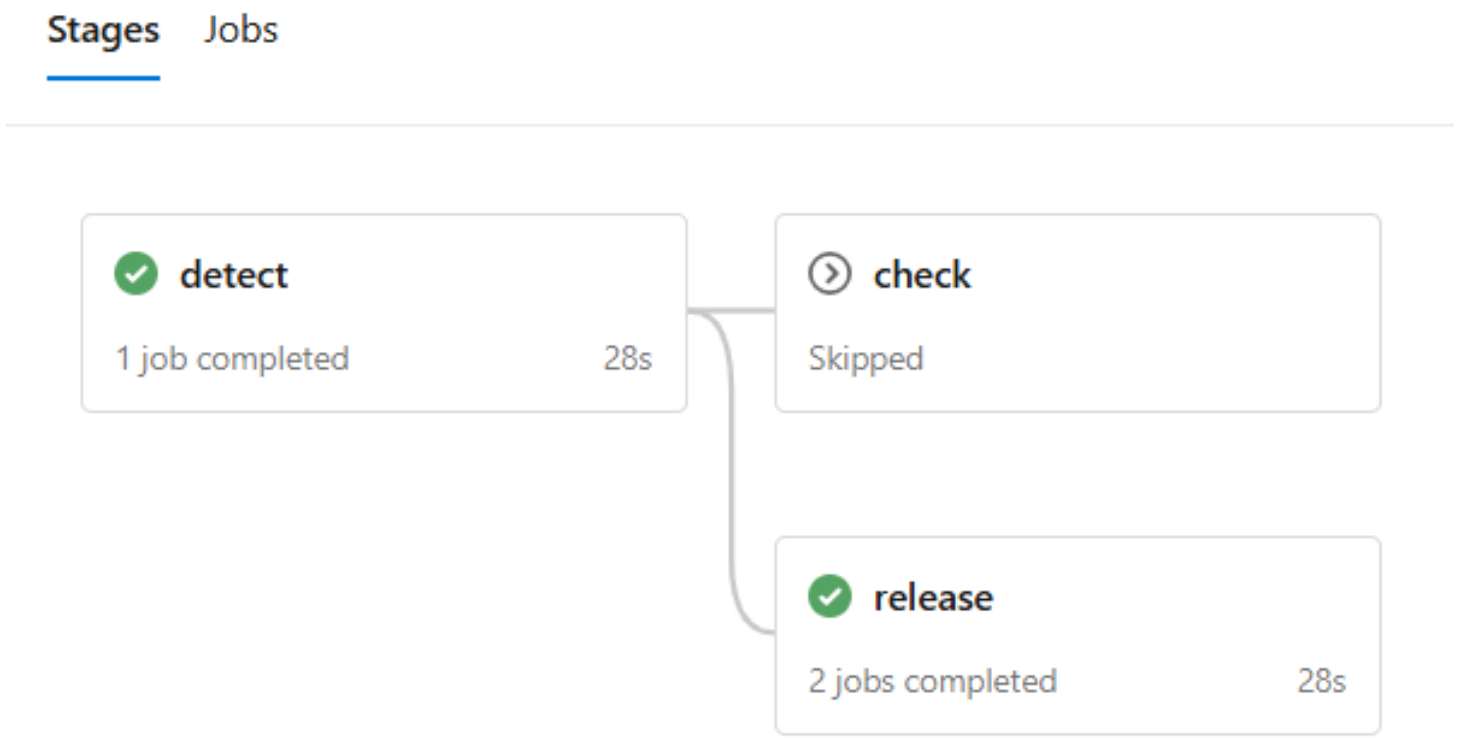
Module pipeline: validation part



etex inspiring ways of living

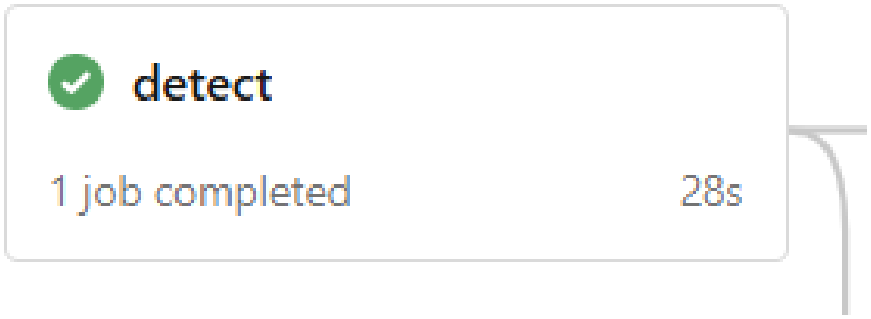
Module versioning pipelines

Module pipeline: deployment part



etex inspiring ways of living

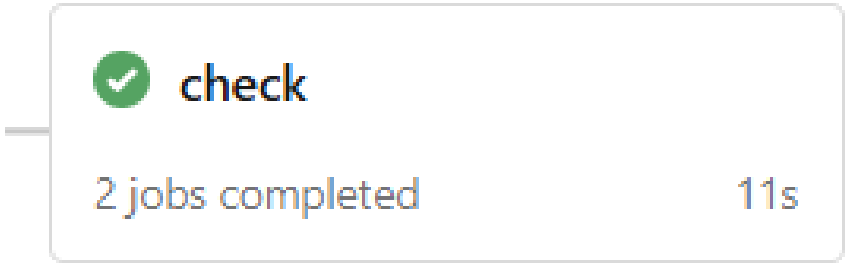
Module versioning pipelines: detect



- Check PR labels on update type (major/minor/patch)
- Check which module has been created/changed
- Create release plan

etex inspiring ways of living

Module versioning pipelines: check



- Perform linter check
- Perform TF format check
- Perform TF validate check

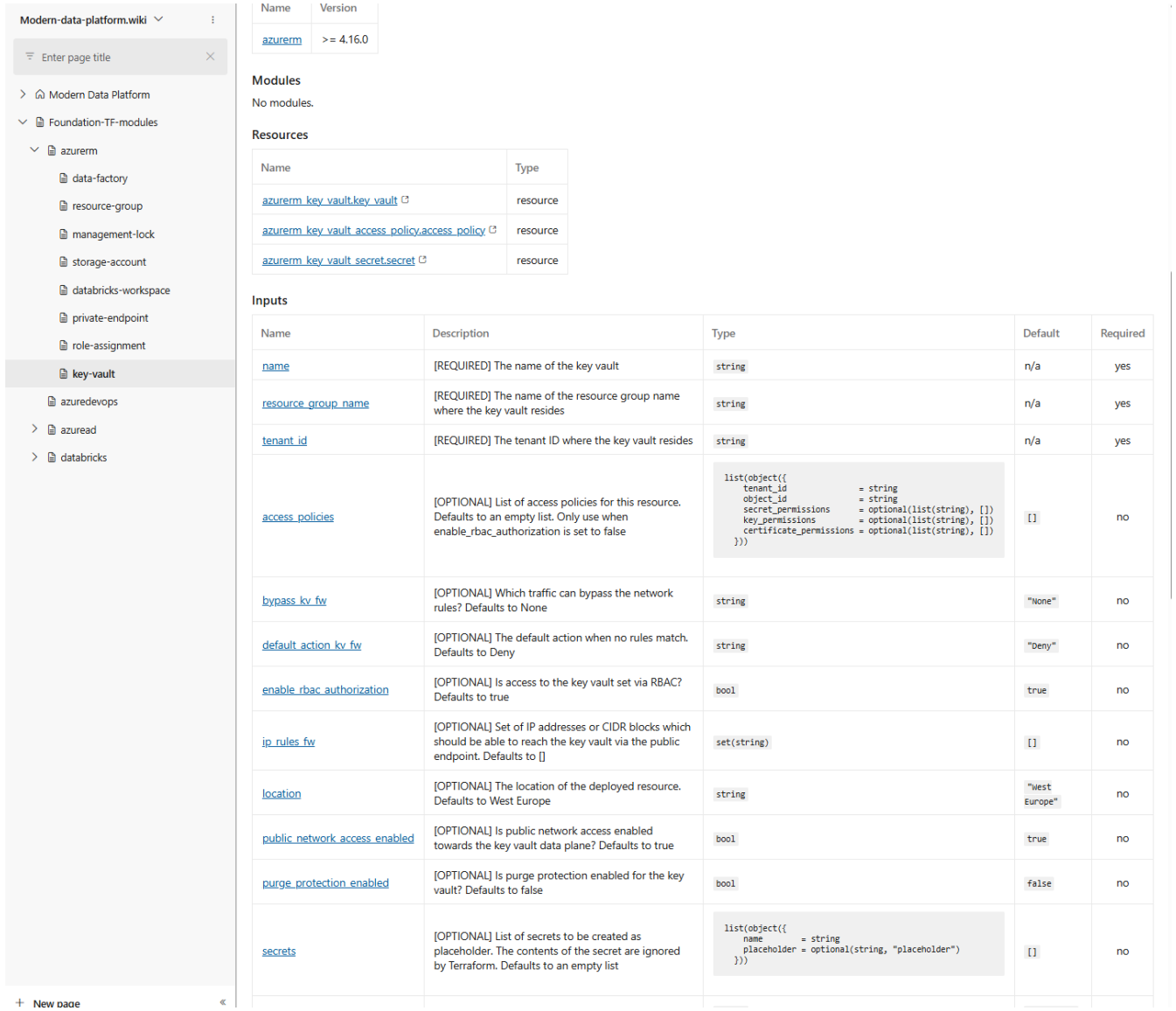
etex inspiring ways of living

Module versioning pipelines: release

- Tag and publish new module version
- Generate and publish version documentation in WIKI

etex inspiring ways of living

Documentation



Name	Version
azurerm	>= 4.16.0

Modules

No modules.

Resources

Name	Type
azurerm_key_vault.key_vault	resource
azurerm_key_vault_access_policy.access_policy	resource
azurerm_key_vault_secret.secret	resource

Inputs

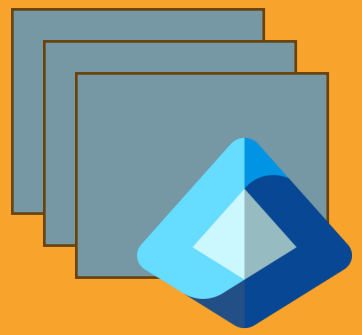
Name	Description	Type	Default	Required
name	[REQUIRED] The name of the key vault	`string`	n/a	yes
resource_group_name	[REQUIRED] The name of the resource group name where the key vault resides	`string`	n/a	yes
tenant_id	[REQUIRED] The tenant ID where the key vault resides	`string`	n/a	yes
access_policies	[OPTIONAL] List of access policies for this resource. Defaults to an empty list. Only use when enable_rbac_authorization is set to false	`list(object({ tenant_id = string object_id = string secret_permissions = optional(list(string), []) key_permissions = optional(list(string), []) certificate_permissions = optional(list(string), []) }))`	`[]`	no
bypass_kv_fw	[OPTIONAL] Which traffic can bypass the network rules? Defaults to None	`string`	`"None"`	no
default_action_kv_fw	[OPTIONAL] The default action when no rules match. Defaults to Deny	`string`	`"Deny"`	no
enable_rbac_authorization	[OPTIONAL] Is access to the key vault set via RBAC? Defaults to true	`bool`	`true`	no
ip_rules_fw	[OPTIONAL] Set of IP addresses or CIDR blocks which should be able to reach the key vault via the public endpoint. Defaults to []	`set(string)`	`[]`	no
location	[OPTIONAL] The location of the deployed resource. Defaults to West Europe	`string`	`"West Europe"`	no
public_network_access_enabled	[OPTIONAL] Is public network access enabled towards the key vault data plane? Defaults to true	`bool`	`true`	no
purge_protection_enabled	[OPTIONAL] Is purge protection enabled for the key vault? Defaults to false	`bool`	`false`	no
secrets	[OPTIONAL] List of secrets to be created as placeholder. The contents of the secret are ignored by Terraform. Defaults to an empty list	`list(object({ name = string placeholder = optional(string, "placeholder") }))`	`[]`	no

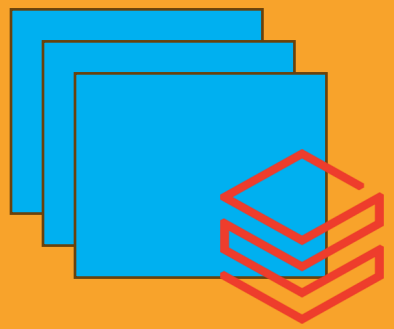
etex inspiring ways of living

Acceleration through reusability

etex inspiring ways of living

Base layer with Terraform modules





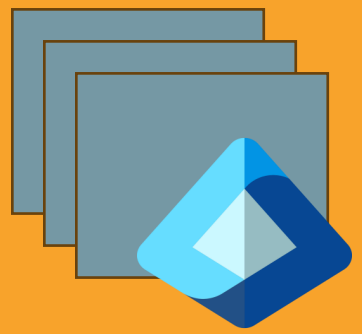
etex inspiring ways of living

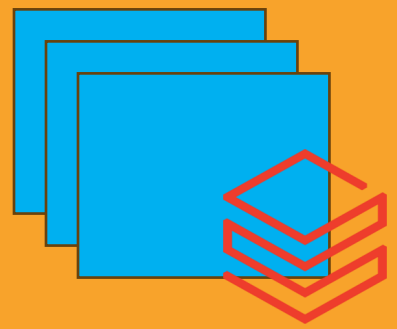
UC 1

UC 2

UC 3

Base layer with Terraform modules





etex inspiring ways of living

UC 1

UC 2

UC 3

etex inspiring ways of living

Translating areas of work into use cases: use case identification

- Determination of scope: what will be deployed/managed by IaC?
- Break scope down in smaller parts of resources/services belonging together: **use cases**
- Grouping depends on:
  - Target platform, practical integration/implementation
  - Deployment strategy

etex inspiring ways of living

Use cases

etex inspiring ways of living

Use cases

iam

azure-resources

rbac

dbx-account-settings

databricks-compute

unity-catalog

databricks-security

etex inspiring ways of living

Use cases

iam

azure-resources

rbac

dbx-account-settings

databricks-compute

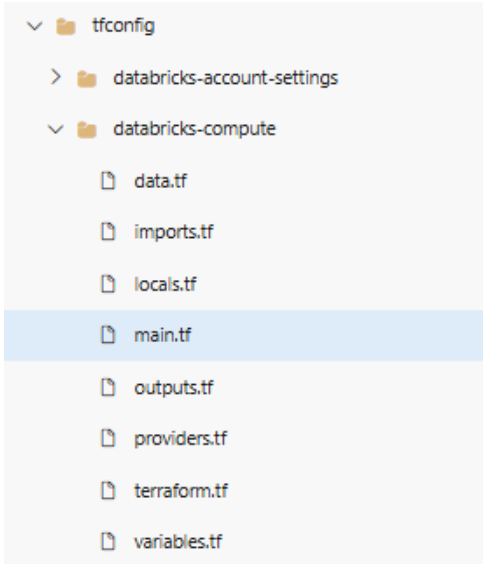
unity-catalog

databricks-security

…

etex inspiring ways of living

In practice



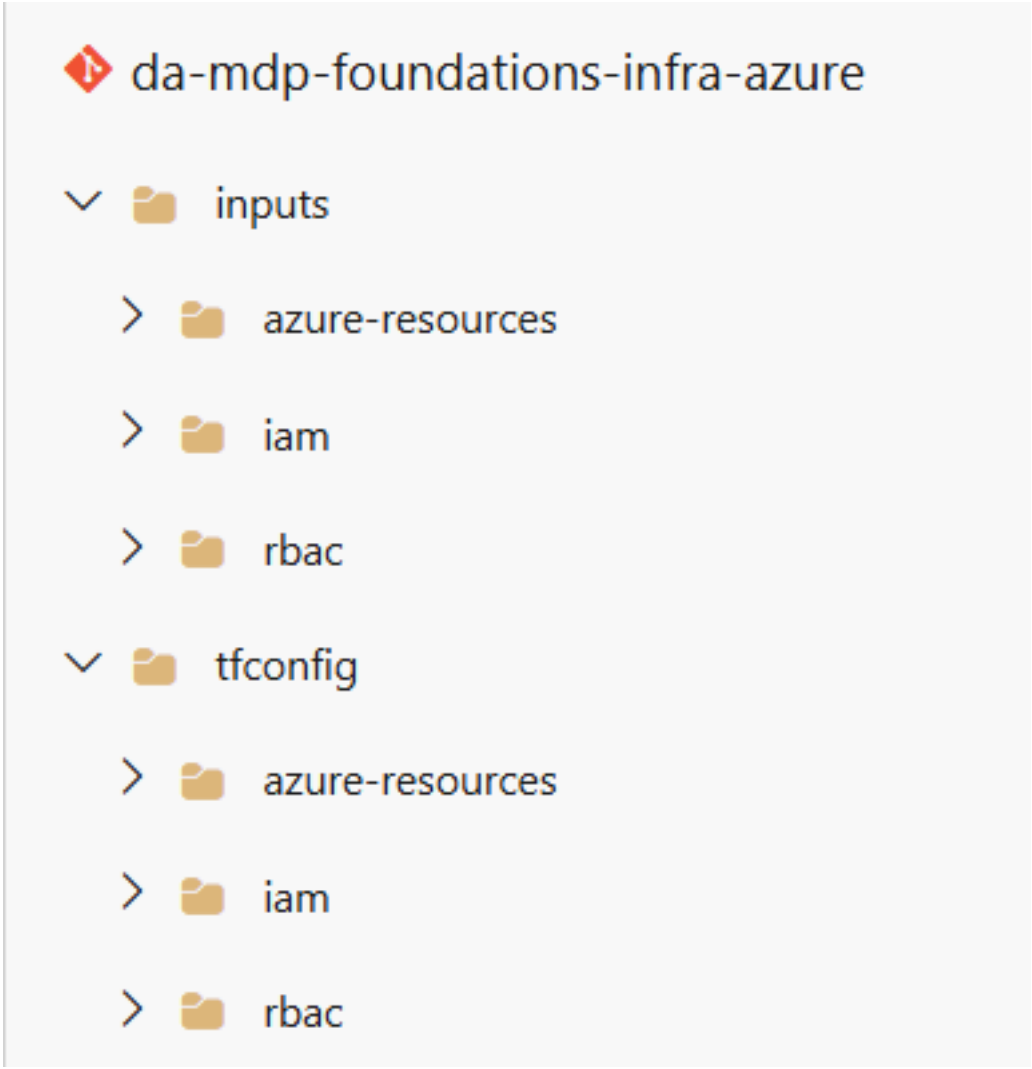

```hcl
module "sql_warehouses" {
  for_each          = local.input_map_sql_warehouses
  source            = "git::https://etexit.visualstudio.com/Modern%20data%20platform/_git/da-mdp-foundations-terraform-modules?ref=databricks/sql-warehouse/v1.1.1"
  name              = each.key
  auto_stop_mins    = each.value.auto_stop_mins
  cluster_size      = each.value.cluster_size
  enable_photon     = each.value.enable_photon
  min_num_clusters  = each.value.min_num_clusters
  max_num_clusters  = each.value.max_num_clusters
  tags = merge(
    var.general_tags,
    each.value.tags
  )
  enable_serverless_compute = each.value.enable_serverless_compute
  warehouse_type            = each.value.warehouse_type
  sql_wh_permissions        = each.value.sql_wh_permissions
}

module "interactive_clusters" {
  for_each                        = local.input_map_interactive_clusters
  source                          = "git::https://etexit.visualstudio.com/Modern%20data%20platform/_git/da-mdp-foundations-terraform-modules?ref=databricks/interactive-cluster/v1.1.0"
  name                            = each.key
  spark_version                   = each.value.spark_version
  spark_version_long_term_support = each.value.spark_version_long_term_support
  enable_autoscale                = each.value.enable_autoscale
  node_type                       = each.value.node_type
  driver_node_type                = each.value.driver_node_type
  min_num_workers                 = each.value.min_num_workers
  max_num_workers                 = each.value.max_num_workers
  num_workers                     = each.value.num_workers
  auto_stop_mins                  = each.value.auto_stop_mins
  tags = merge(
    var.general_tags,
    each.value.tags
  )
  cluster_permissions = each.value.cluster_permissions
}
```


etex inspiring ways of living

Use cases: input vs. config

- Inputs: *changes what will be deployed*
- Config: *changes how it will be deployed*



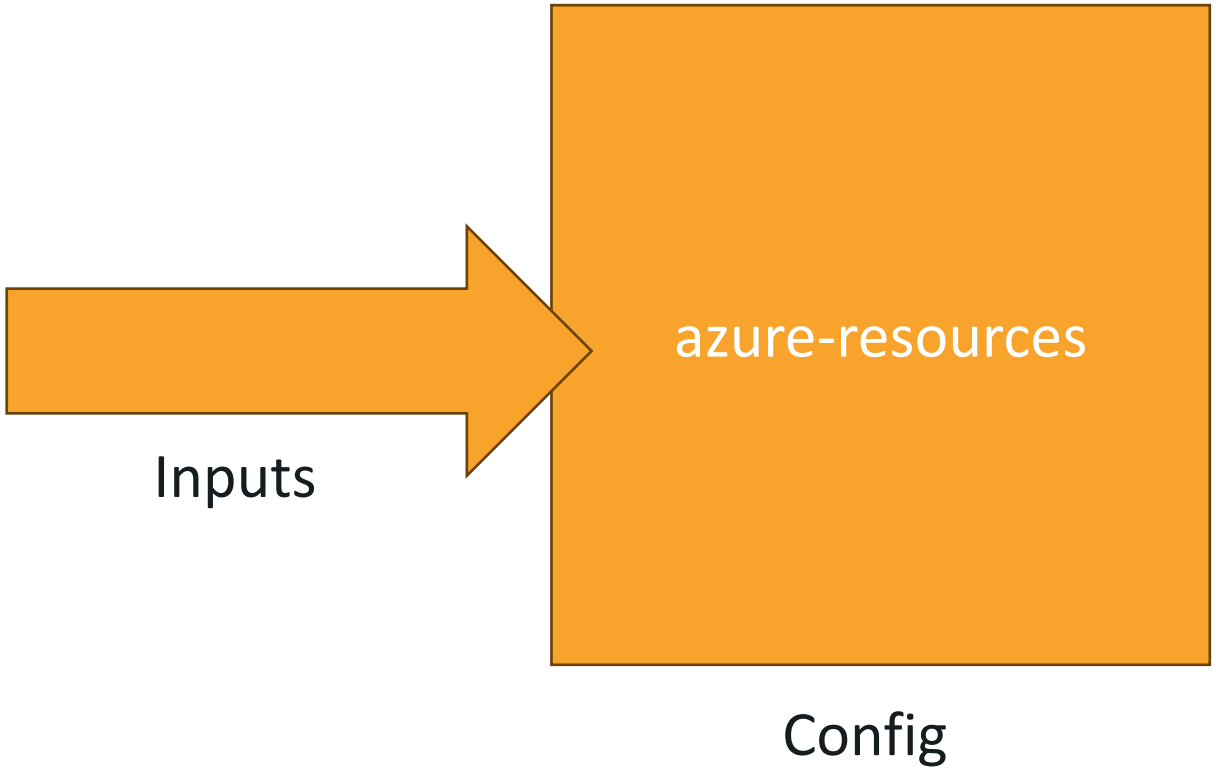
etex inspiring ways of living

Use cases

Config

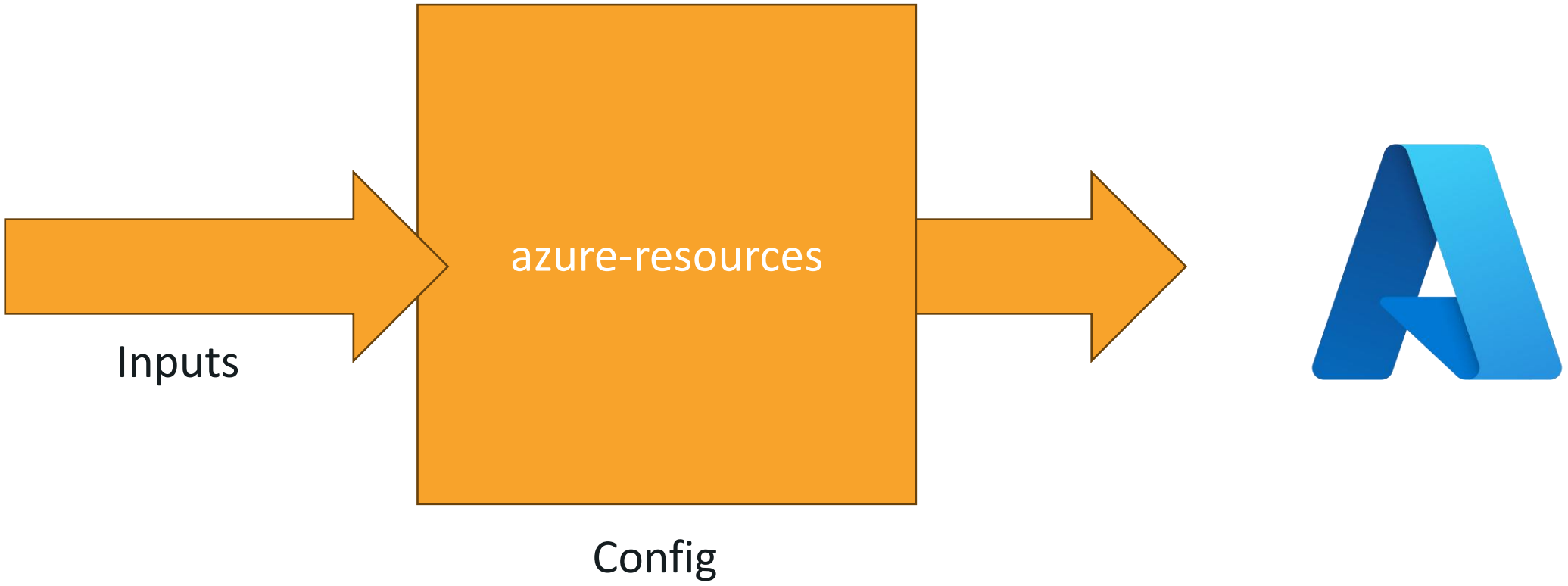
etex inspiring ways of living

Use cases



etex inspiring ways of living

Use cases



etex inspiring ways of living

Use cases: automation

Per use case:

- Automated deployments
- Inter-use case independent deployments
- Versioning

etex inspiring ways of living

Use cases: versioning

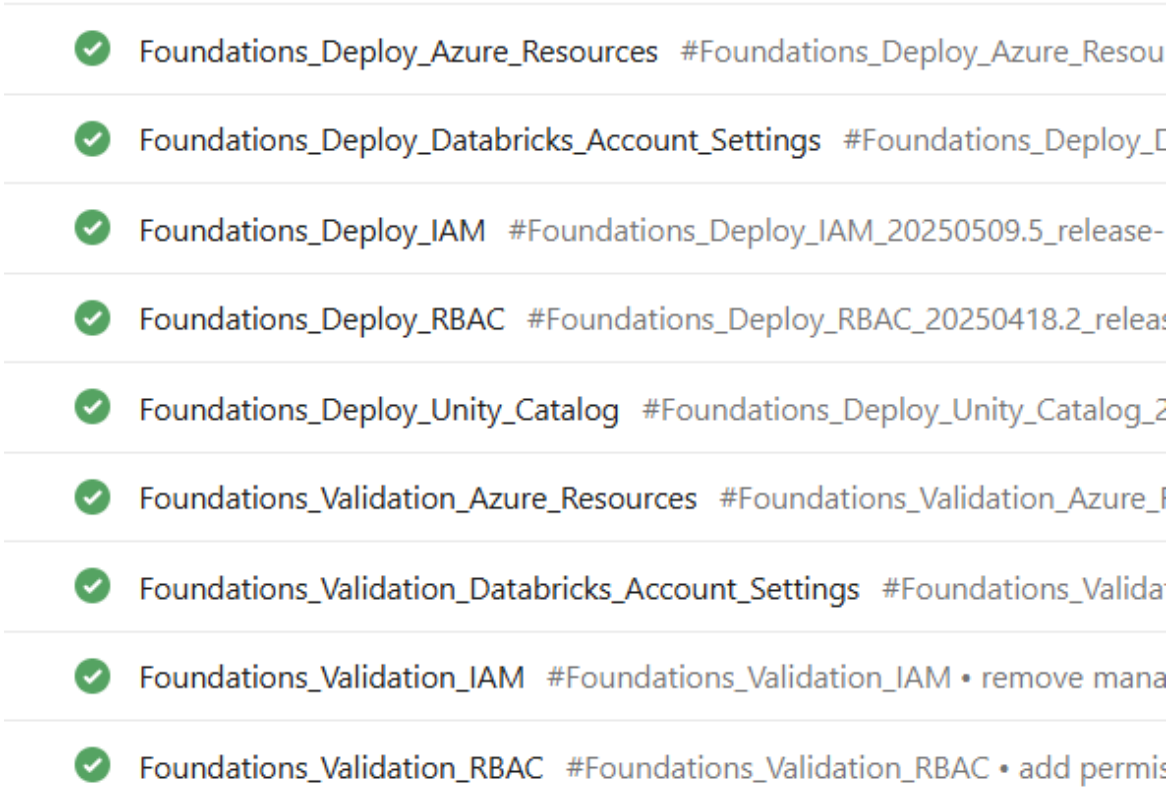
- Modules: *semantic versioning*
- Use cases: other approach for versioning

etex inspiring ways of living

Use cases: independent automated deployments

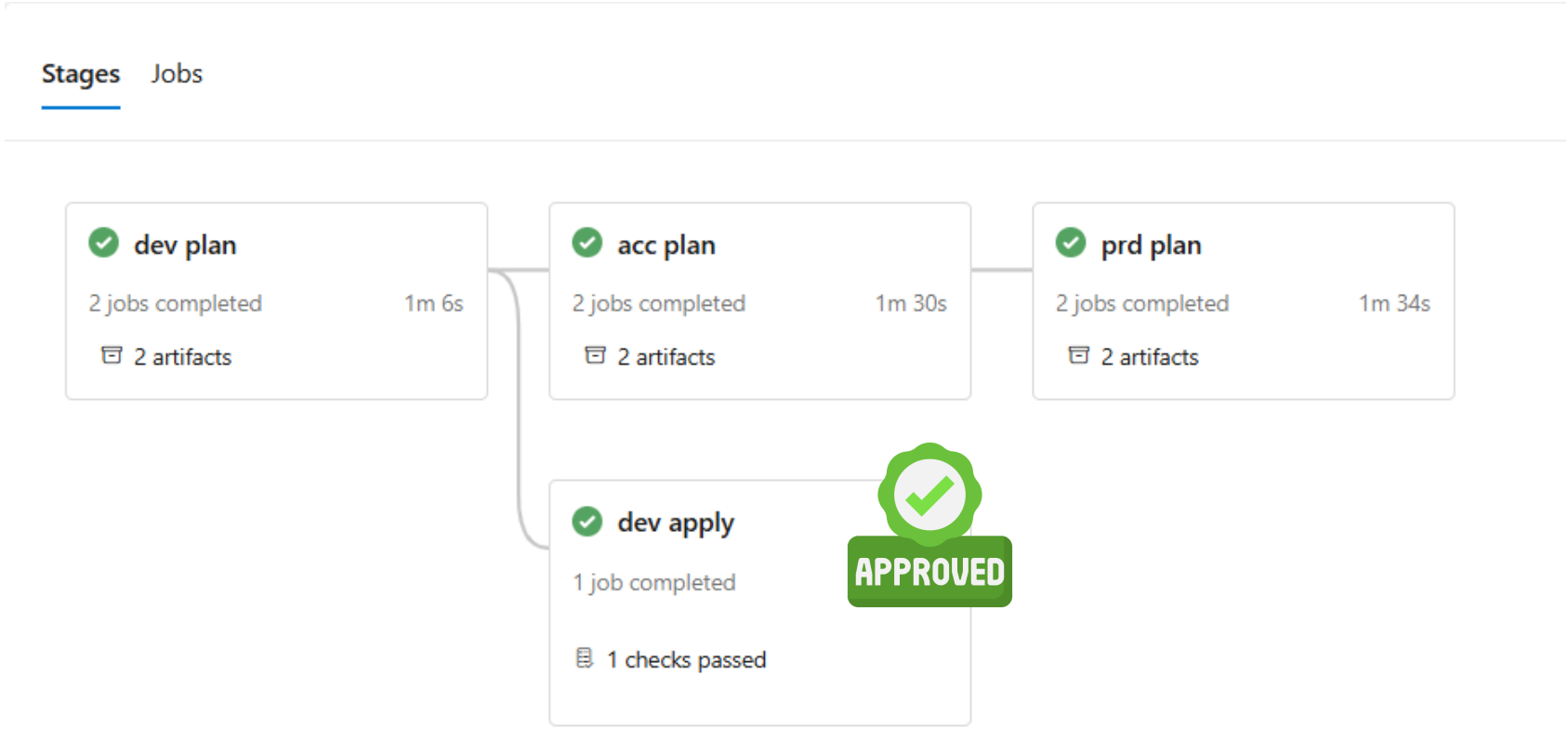
- Each use case:
  - *Validation pipeline*
  - *Deployment pipeline*

- Pipelines are triggered based on:
  - Branch, PR
  - File path



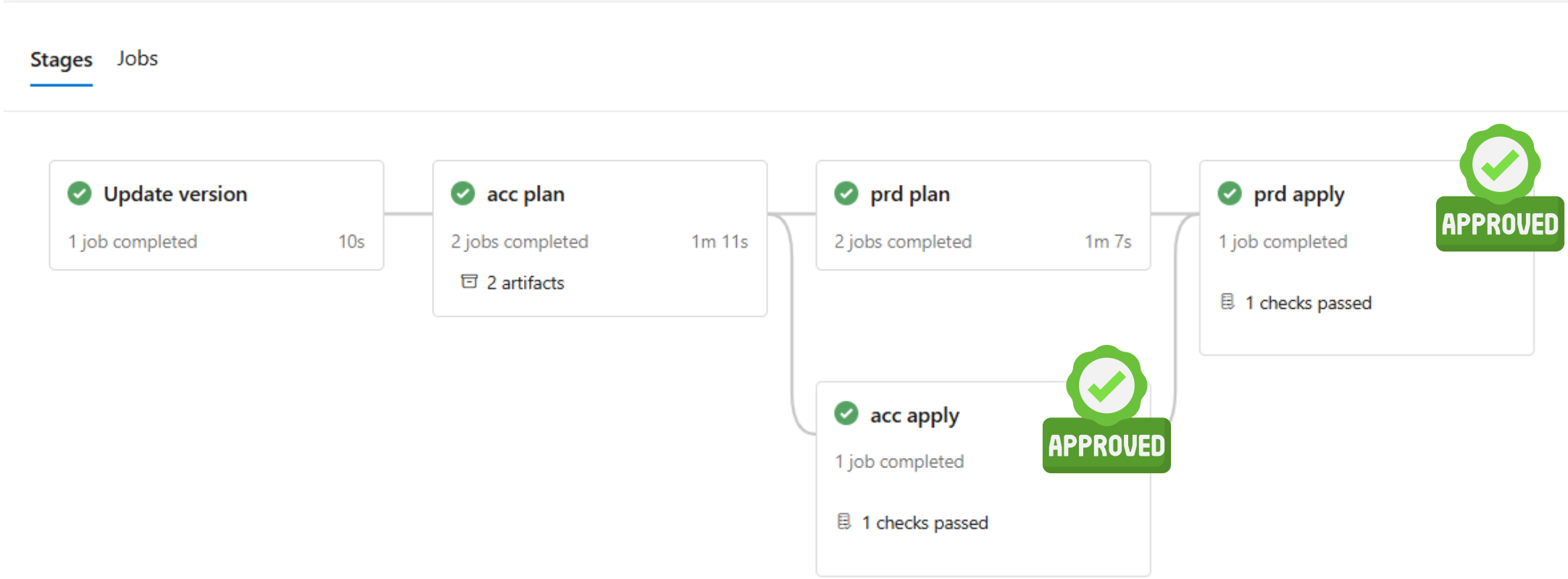
etex inspiring ways of living

Use cases: validation pipeline



etex inspiring ways of living

Use cases: deployment pipeline



etex inspiring ways of living

Where are we now?

etex inspiring ways of living

Where are we now?

- Baseline (framework) is set up
  - Modules
  - Use cases

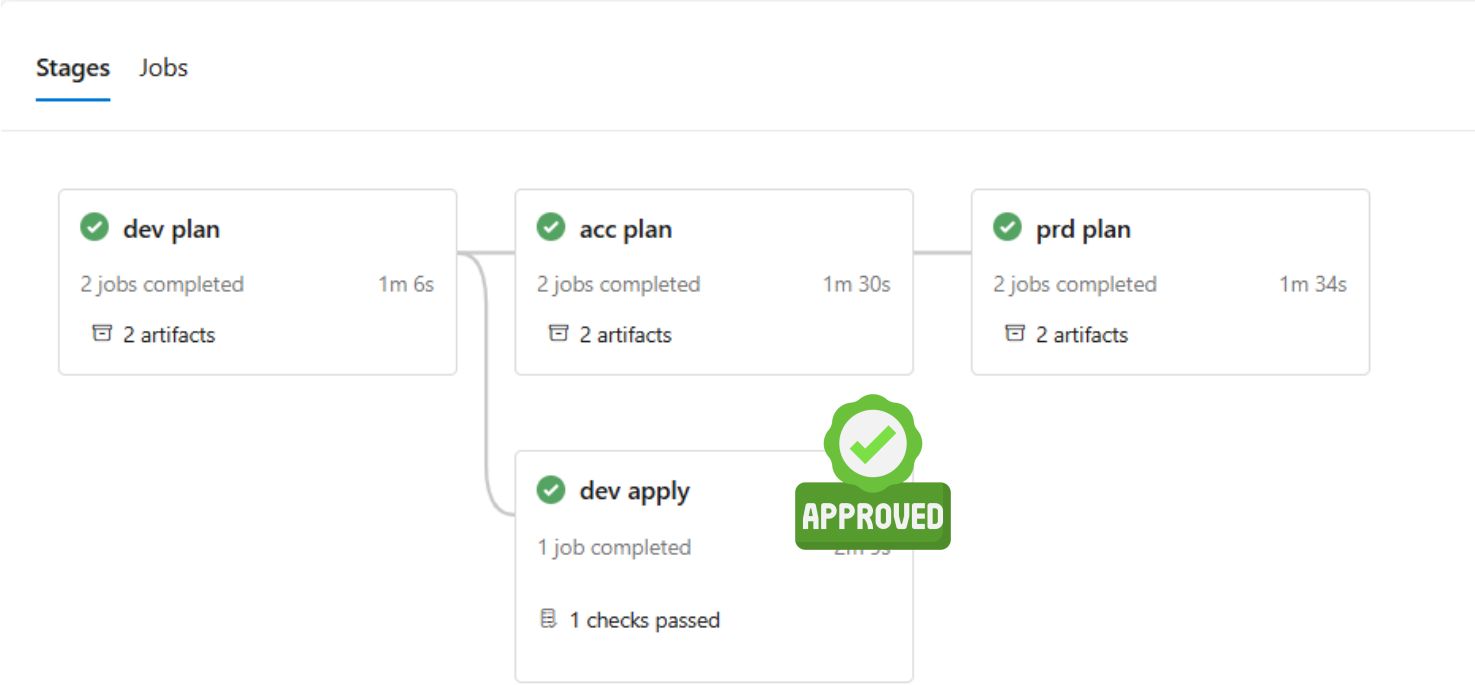
- Foundations team is now using framework
  - Extra resources → via inputs
  - New use cases / architecture changes → via config

- Standardize way of working now that initial development is done

etex inspiring ways of living

Standardized way of working

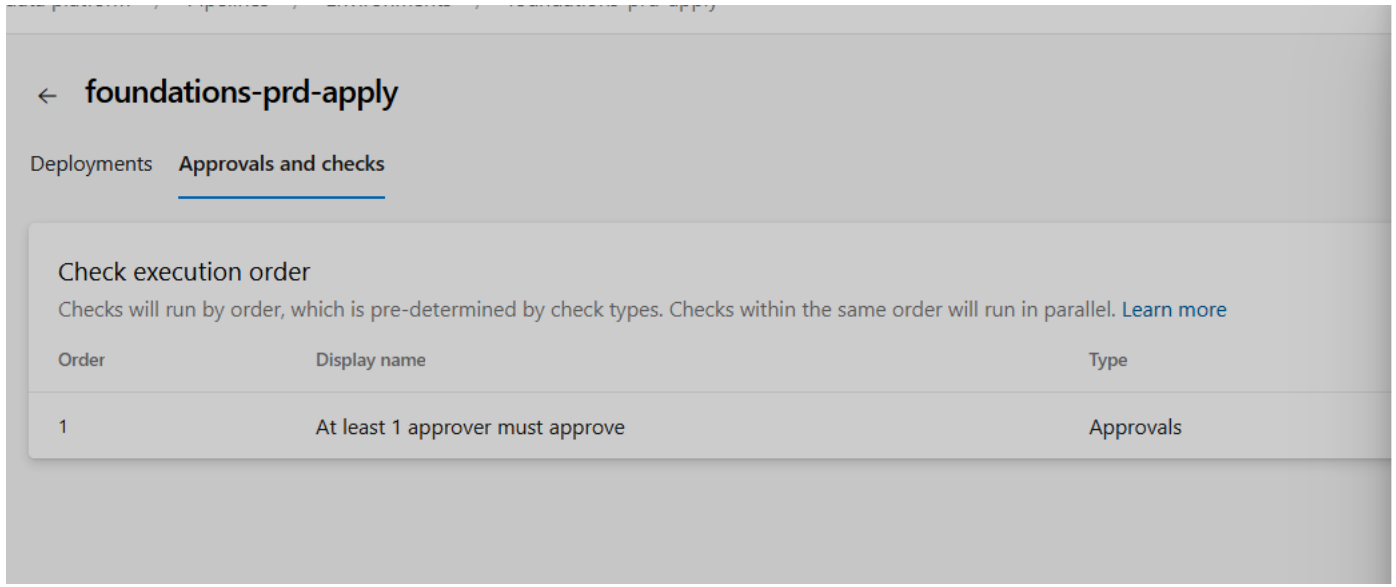
- Having *guardrails* in place for development
  - Deployment approvals via environments

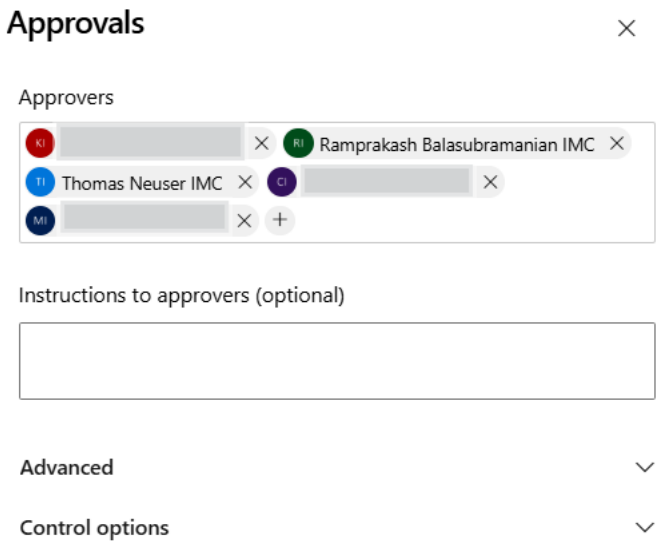


etex inspiring ways of living

Standardized way of working

- Having *guardrails* in place for development
  - Deployment approvals via environments

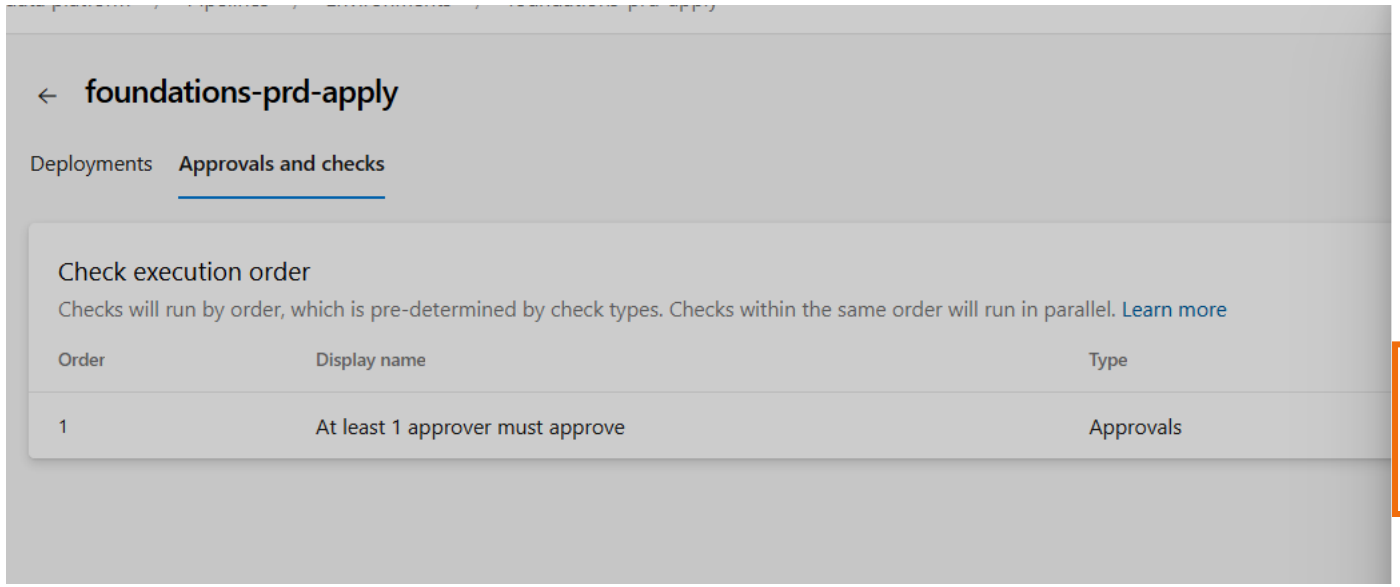


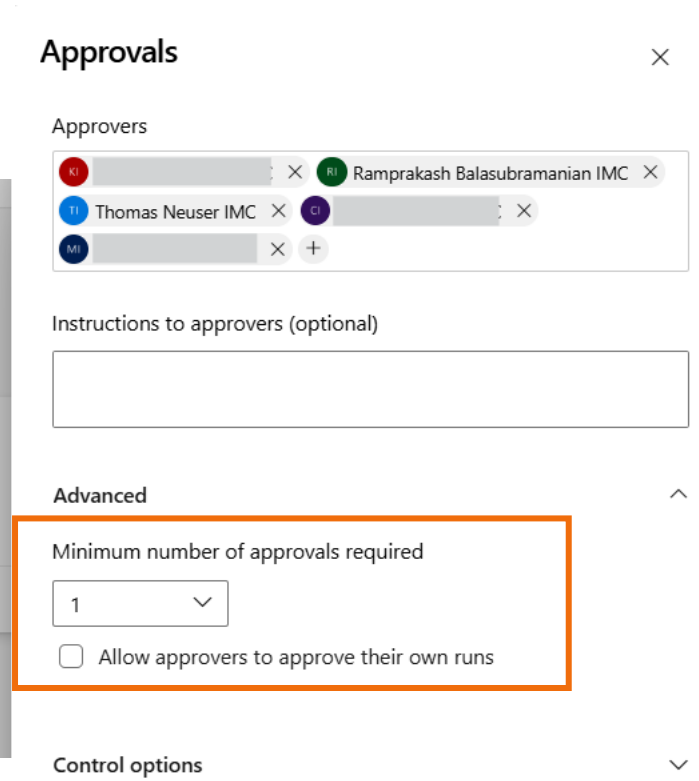


etex inspiring ways of living

Standardized way of working

- Having *guardrails* in place for development
  - Approvals via environments

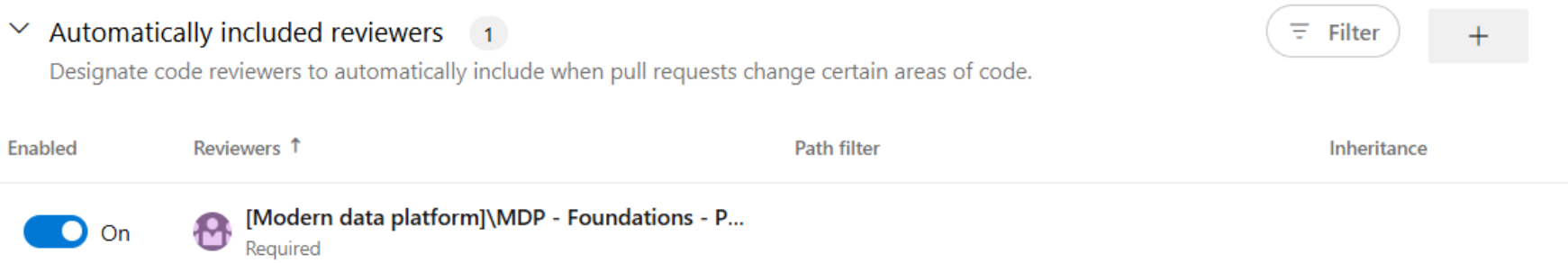




etex inspiring ways of living

Standardized way of working

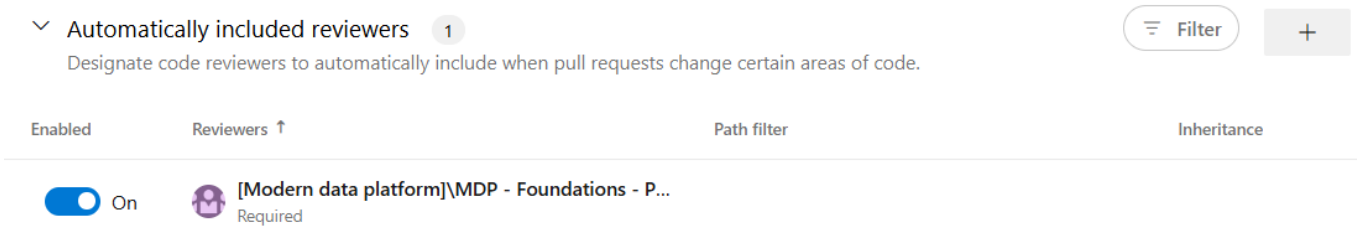
- Having *guardrails* in place for development
  - Pull request validation checks via policies: *Required reviewers*

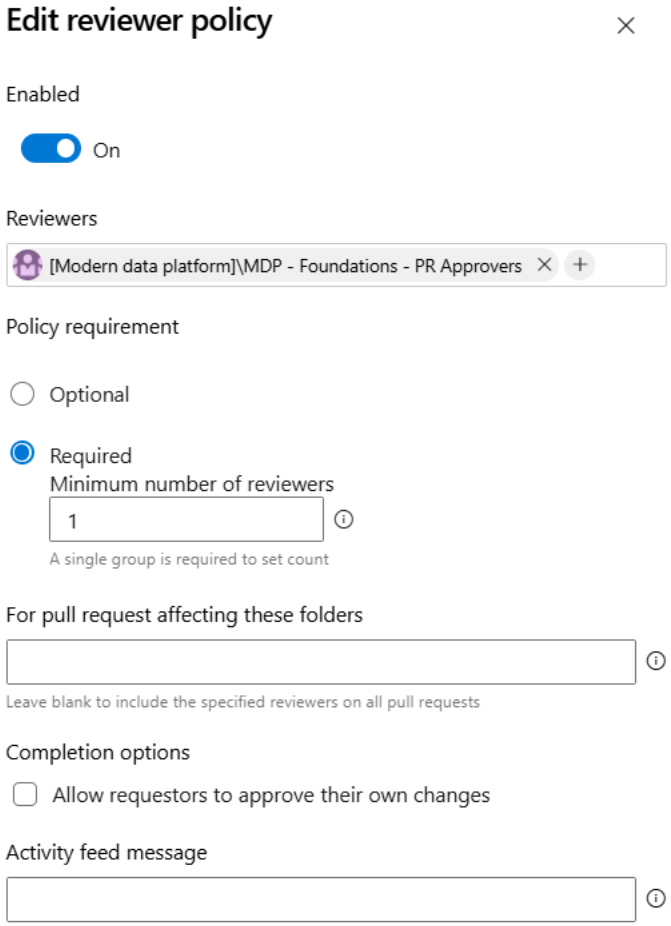


etex inspiring ways of living

Standardized way of working

- Having *guardrails* in place for development
  - Pull request validation checks via policies: *Required reviewers*

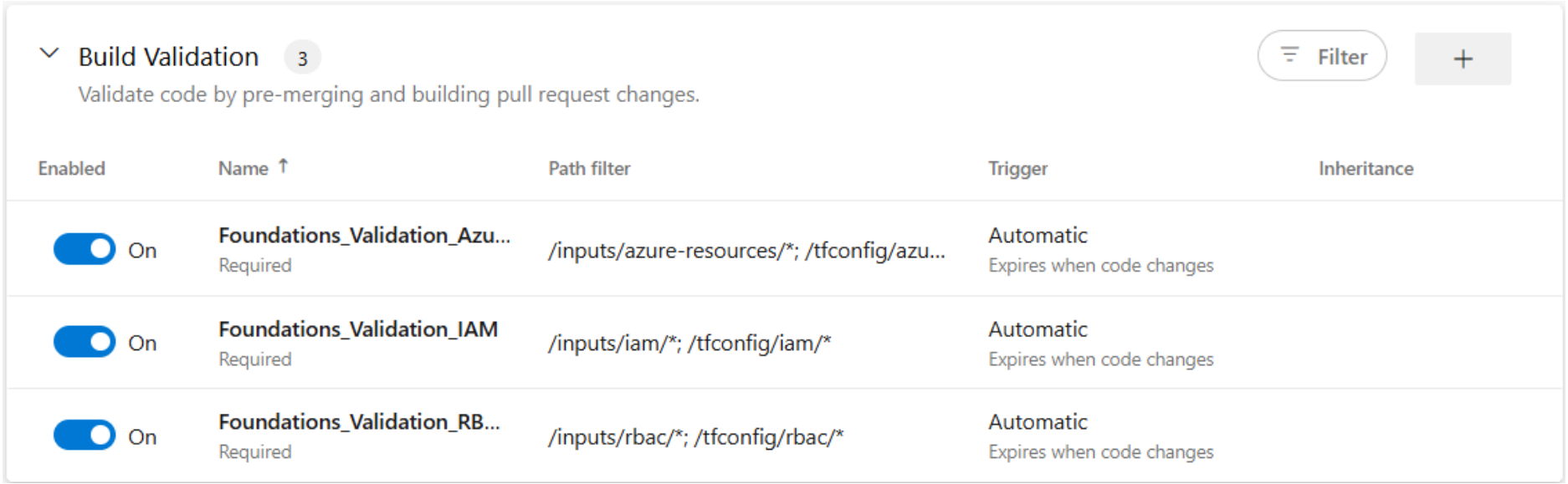




etex inspiring ways of living

Standardized way of working

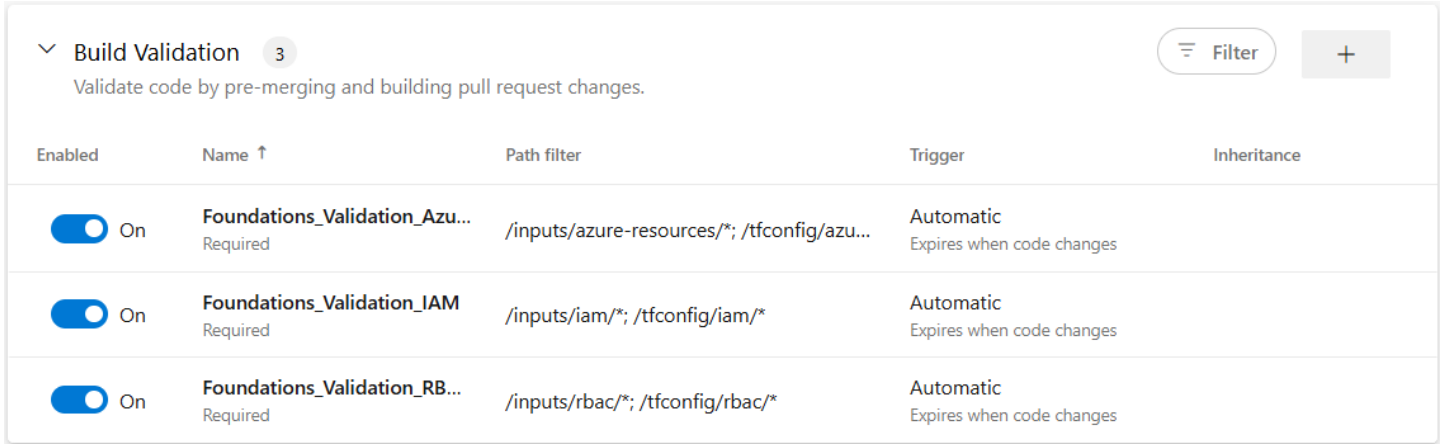
- Having *guardrails* in place for development
  - Pull request validation checks via policies:
    Build validation pipelines

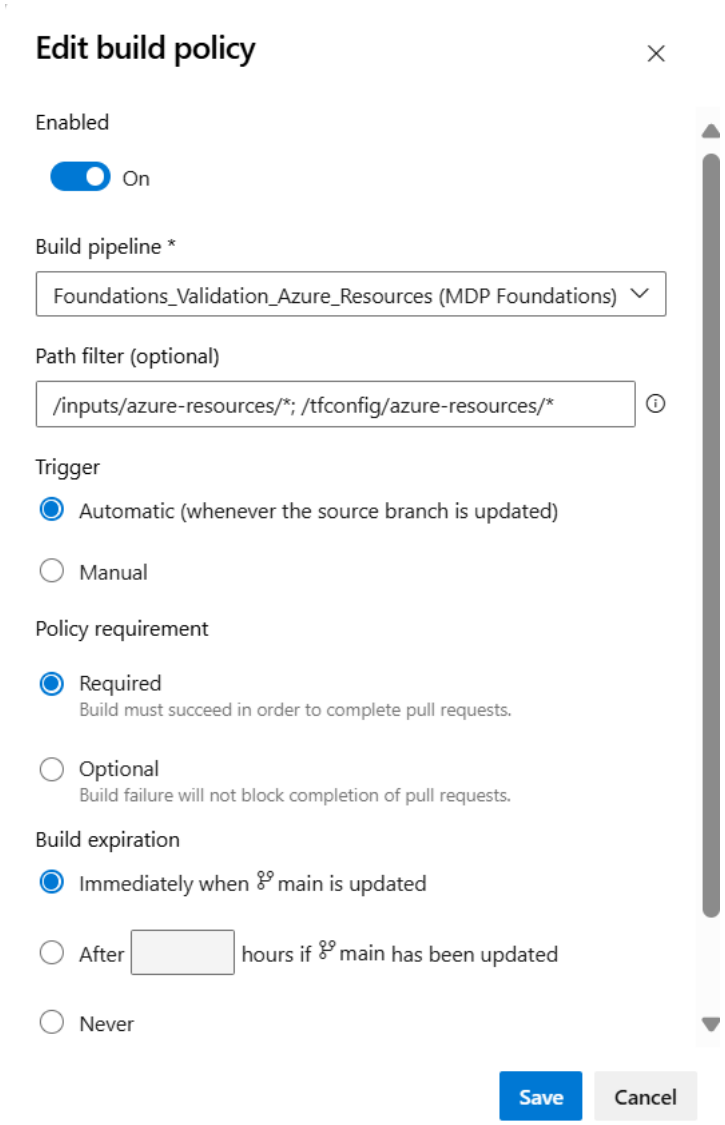


etex inspiring ways of living

Standardized way of working

- Having *guardrails* in place for development
  - Pull request validation checks via policies: *Build validation pipelines*





Solution impact

etex inspiring ways of living

Impact & added value

- Clean-up and added governance to all related platforms
  - Enforcing naming convention
  - Govern permissions
  - Get rid of “click-ops”
  - Standardization across deployments

- Acceleration of deployments because of the base layer with TF modules

etex inspiring ways of living

Future aspirations

etex inspiring ways of living

Future aspirations

- Further incorporation of the already existing platform components into IaC
- Expand the scope of managed IaC resources (new use cases, new platforms)
- Move towards self-service Infra-as-Code (2 phases)

etex inspiring ways of living

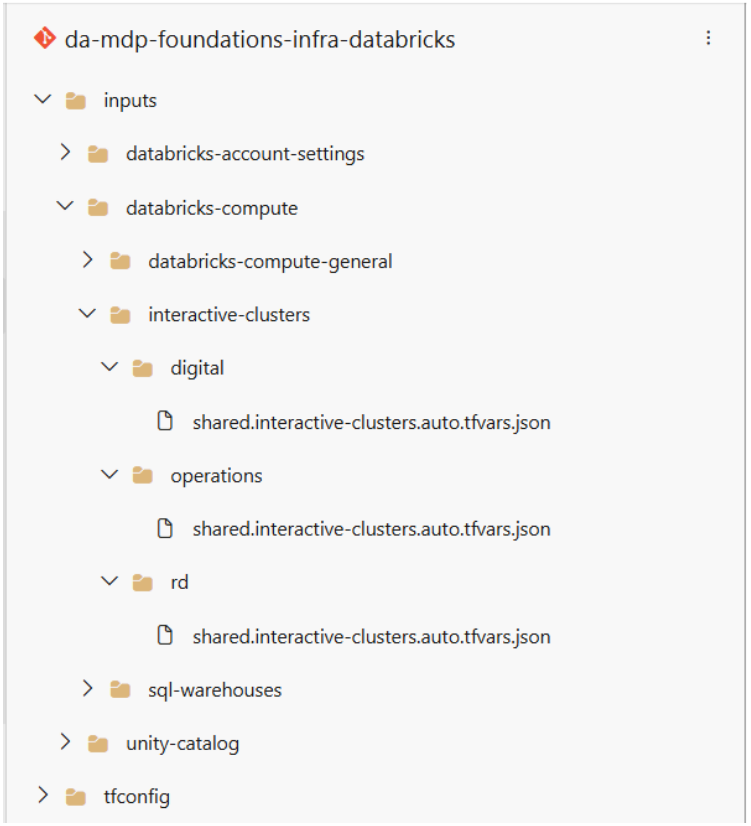
Future aspirations

- Further incorporation of the already existing platform components into IaC
- Expand the scope of managed IaC resources (new use cases, new platforms)
- **Move towards self-service Infra-as-Code (2 phases)**

etex inspiring ways of living

Self-service Infra-as-Code (phase 1): goal

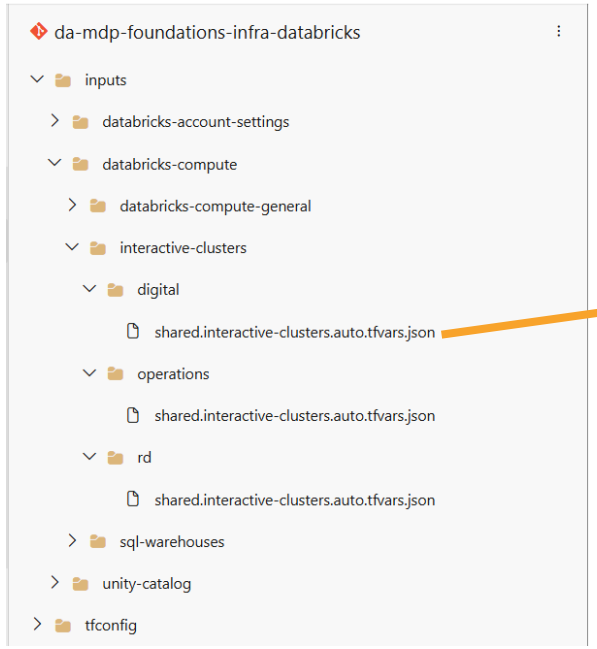
Enable technical teams to tweak IaC inputs via our repository



etex inspiring ways of living

Self-service Infra-as-Code (phase 1): goal

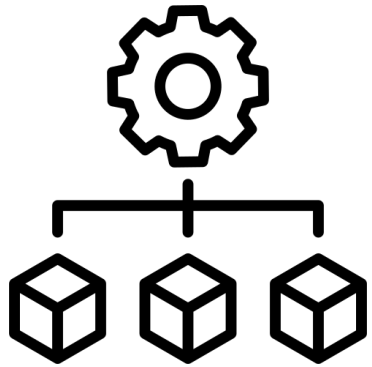
Enable technical teams to tweak IaC inputs via our repository




```
{
    "interactive_clusters": [
        {
            "domain": "dig",
            "subdomain": "placeholder",
            "use_case": "pbi",
            "index": "01",
            "spark_version": null,
            "spark_version_long_term_support": true,
            "enable_autoscale": true,
            "node_type": "Standard_DS3_v2",
            "driver_node_type": "Standard_DS3_v2",
            "min_num_workers": 1,
            "max_num_workers": 2,
            "num_workers": 1,
            "auto_stop_mins": 15,
            "cluster_permissions": [
                {
                    "principal_type": "GROUP",
                    "principal_name": "GROUP_NAME",
                    "permission_level": "CAN_ATTACH_TO"
                }
            ],
            "tags": {
                "UseCase": "Placeholder",
                "Domain": "Placeholder",
                "Subdomain": "Placeholder",
                "Description": "Placeholder",
                "BusinessOwner": "Placeholder"
            }
        }
    ]
}
```

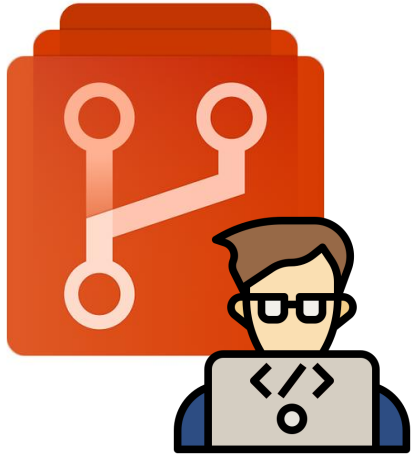

etex inspiring ways of living

Self-service Infra-as-Code (phase 1): flow



etex inspiring ways of living

Self-service Infra-as-Code (phase 1): flow



etex inspiring ways of living

Self-service Infra-as-Code (phase 1): flow

etex inspiring ways of living

Self-service Infra-as-Code (phase 1): flow

etex inspiring ways of living

Self-service Infra-as-Code (phase 1): downsides

- Teams have to be technical and code savvy to add resources
- Foundations team have to check the code for possible mistakes, involves (a lot of) manual work

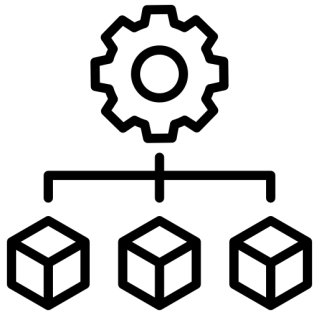
etex inspiring ways of living

Self-service Infra-as-Code (phase 2): goal

Enable (technical) teams to tweak IaC inputs via Service desk/GUI application

etex inspiring ways of living

Self-service Infra-as-Code (phase 2): flow



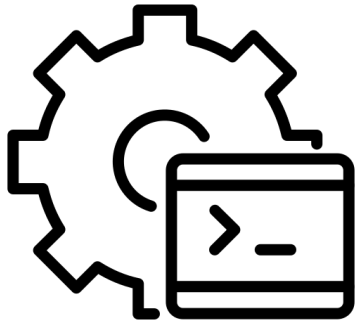
etex inspiring ways of living

Self-service Infra-as-Code (phase 2): flow

TOPdesk

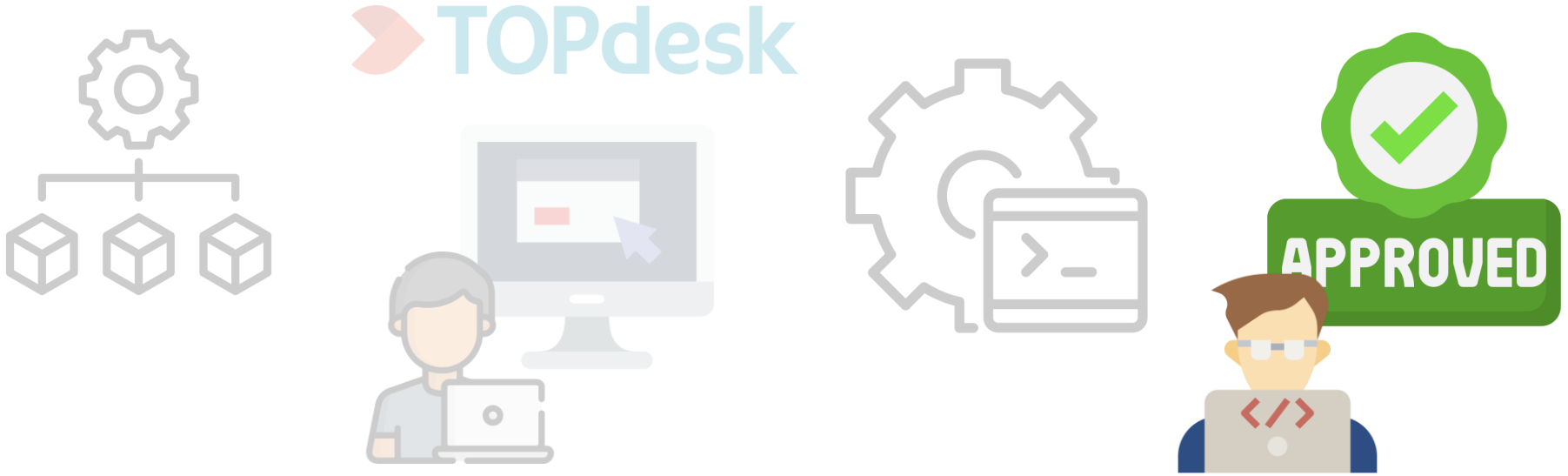
etex inspiring ways of living

Self-service Infra-as-Code (phase 2): flow



etex inspiring ways of living

Self-service Infra-as-Code (phase 2): flow



etex inspiring ways of living

Self-service Infra-as-Code (phase 2): flow

etex inspiring ways of living

Self-service Infra-as-Code (phase 2): upsides

- No direct code tweaking is needed anymore
- Addition of needed code is handled automatically and put in the right place
- Less error prone
- PR review becomes less work intensive, validation can happen within the application's GUI

etex inspiring ways of living

That’s a wrap, your thoughts?

etex inspiring ways of living

Thank you

Ramprakash Balasubramanian

Thomas Neuser

etex inspiring ways of living