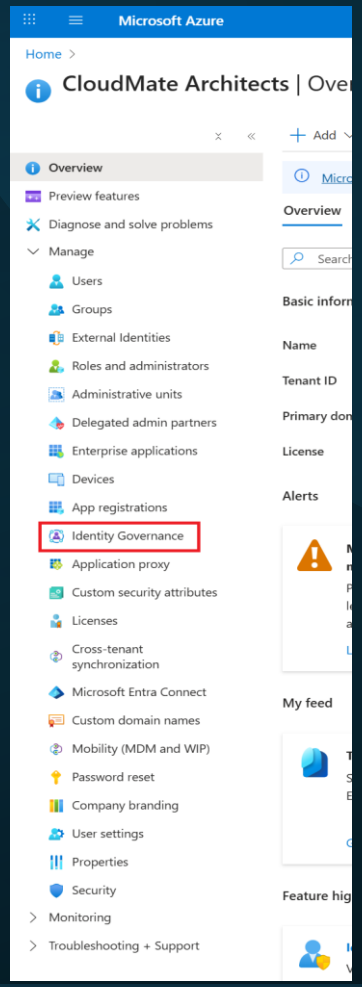
Build a Secure Access Catalog in Microsoft Entra

Implement governance foundation with Microsoft Entra Entitlement Management

PerparimLabs

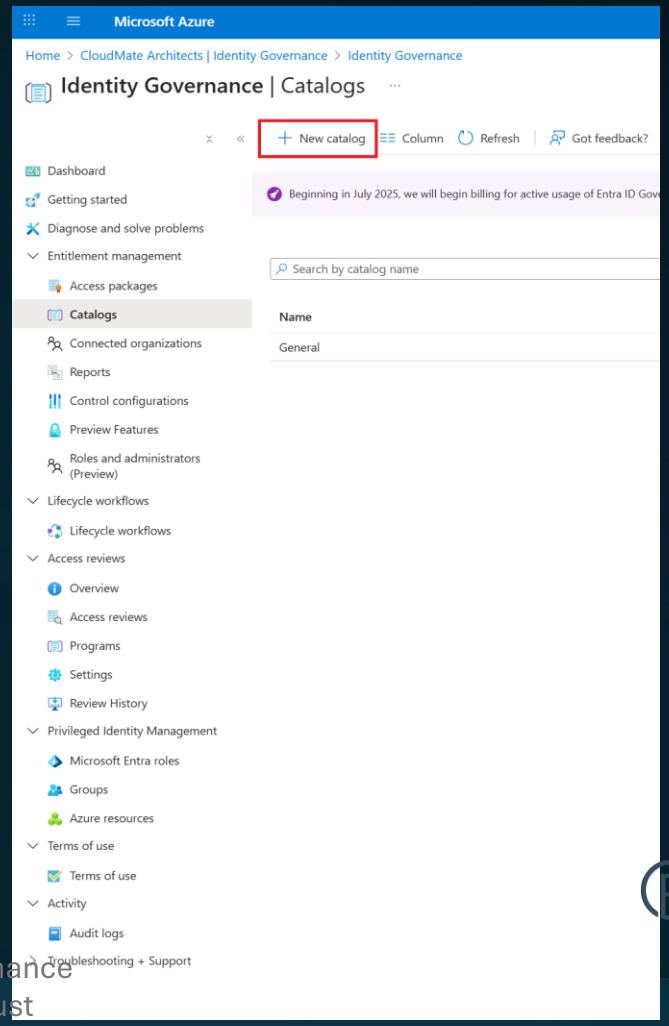
Open Microsoft Entra Admin Center

Navigate to **Identity Governance → Entitlement Management**



Architect Insight: Entitlement Management automates identity lifecycle governance — replacing static group membership with policy-based access that expires automatically.

Create New Catalog

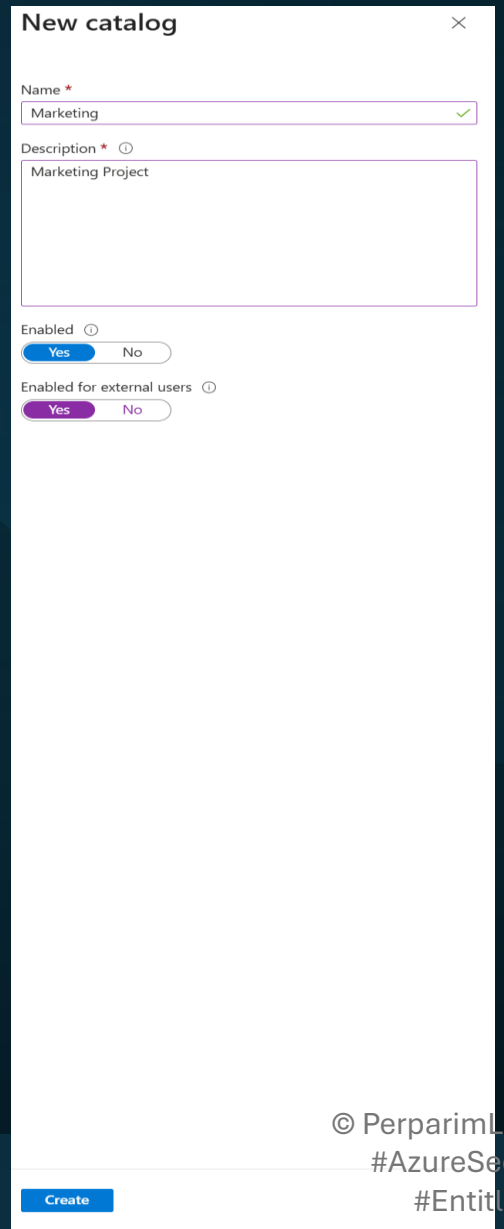


PerparimLabs

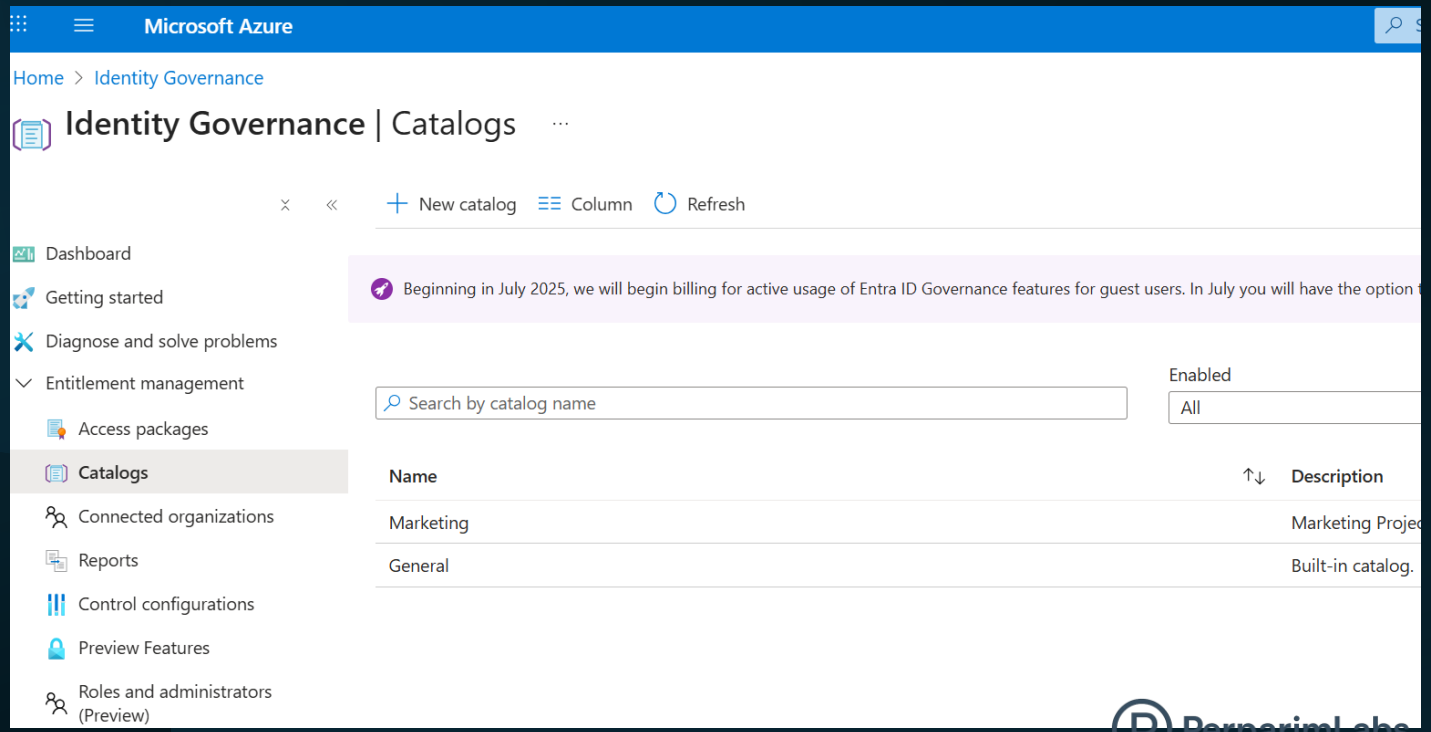
Name the Catalog

Why It Matters: Catalogs act as containers for resources and access packages.
They enforce governance boundaries — each catalog can have its own set of owners, policies, and connected resources.

Example:
Marketing Project
— describes business scope

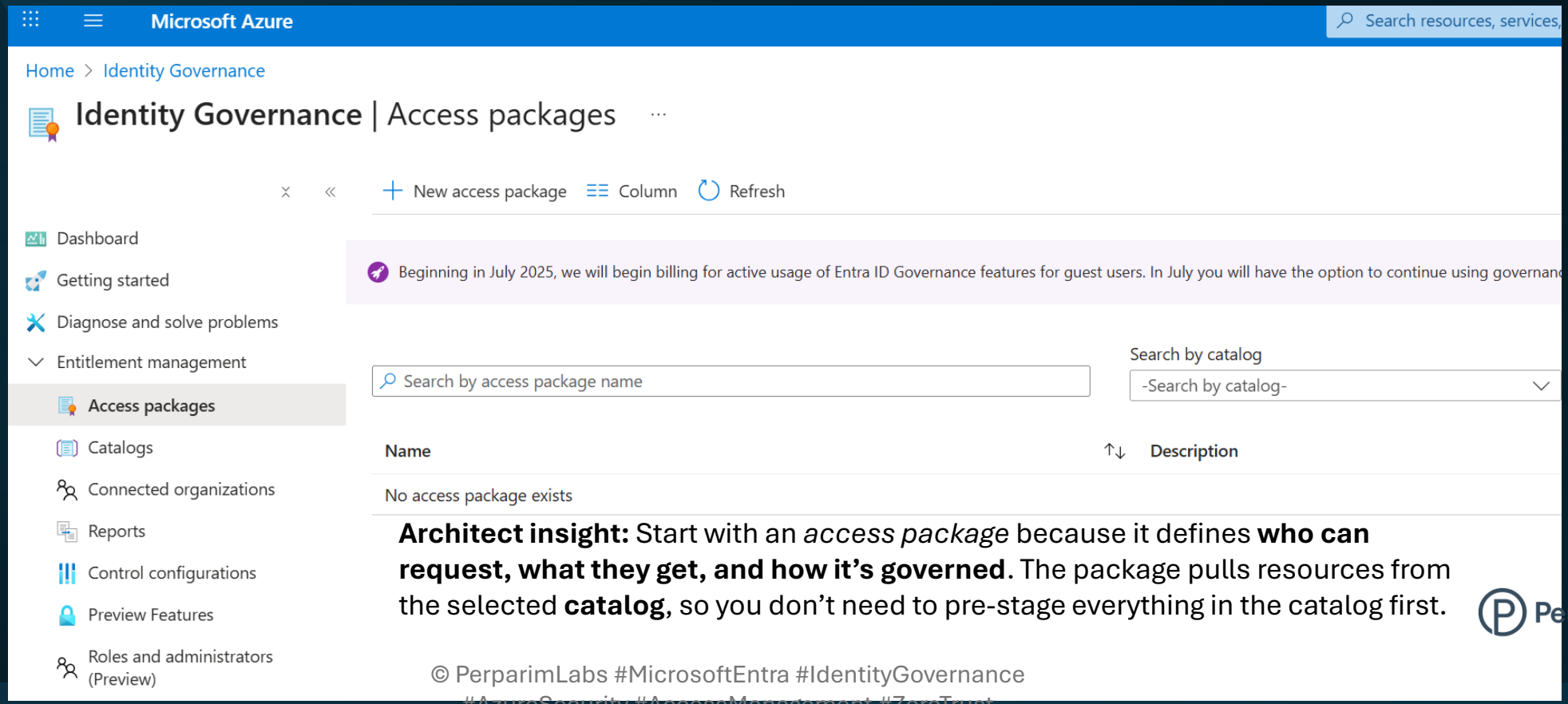


Created the **Marketing** catalog that will host all project resources.



Open Access packages → New access package

Start a new package that will pull resources from the **Marketing** catalog.



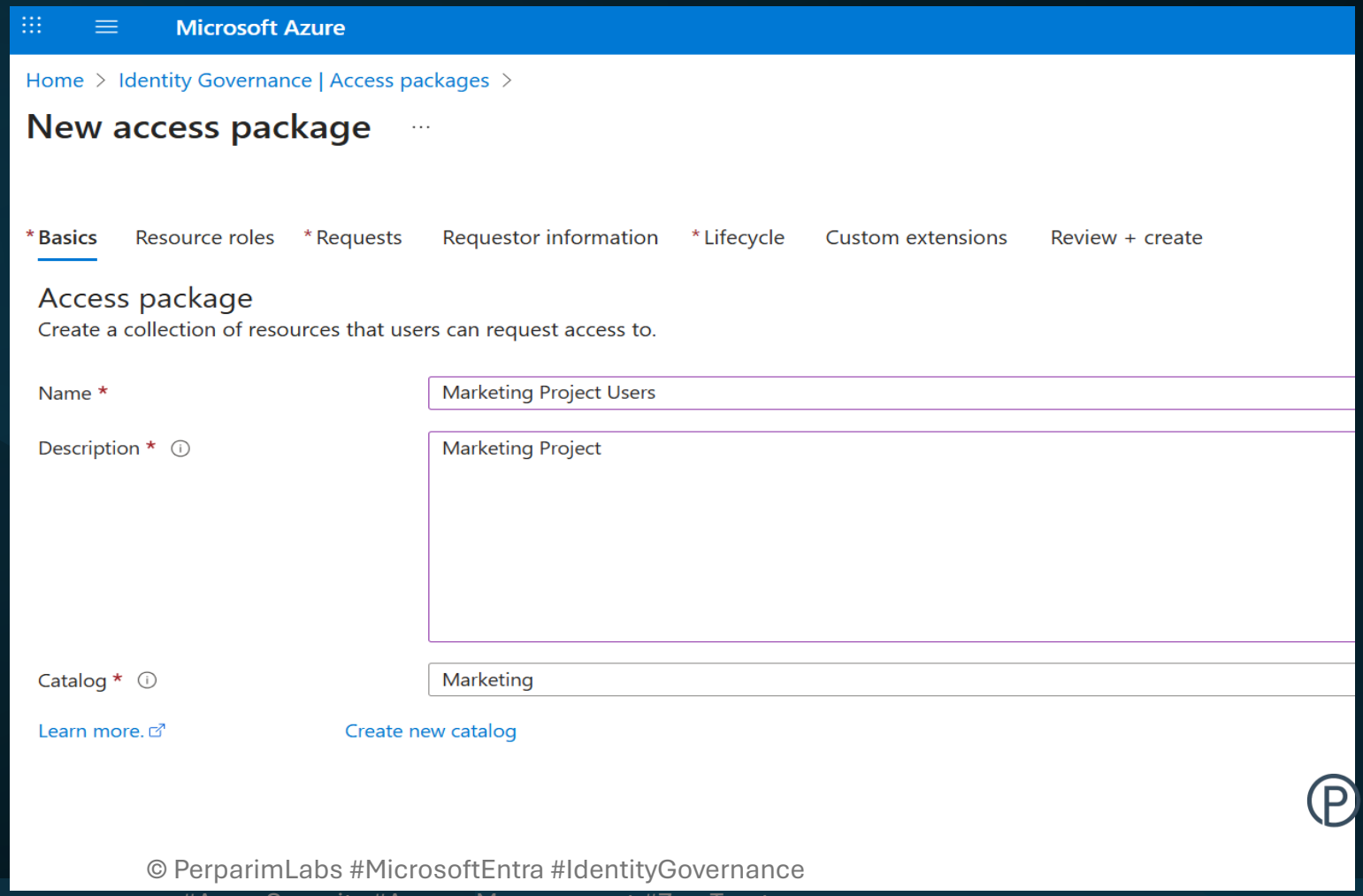
Architect insight: Start with an *access package* because it defines **who can request, what they get, and how it's governed**. The package pulls resources from the selected **catalog**, so you don't need to pre-stage everything in the catalog first.

PerparimLabs

Name, description, and select **Marketing** catalog

Package name **Marketing Project Users**; catalog **Marketing**.

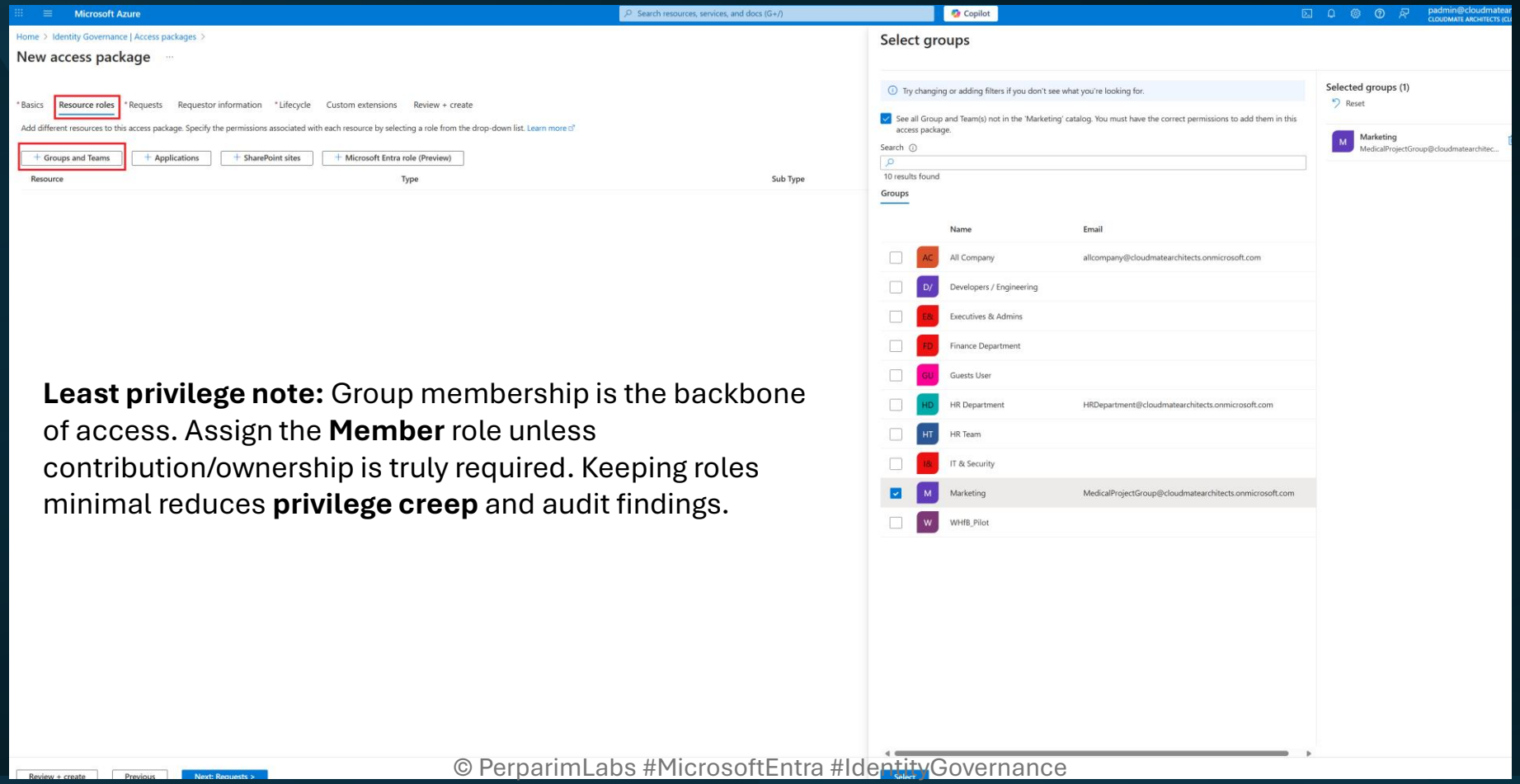
Why it matters: The **catalog** sets the governance boundary (owners, resources, policies). The package name/description should reflect **business purpose** so approvers/auditors instantly understand *why access exists*.



PerparimLabs

Add Group: **Marketing** (Member)

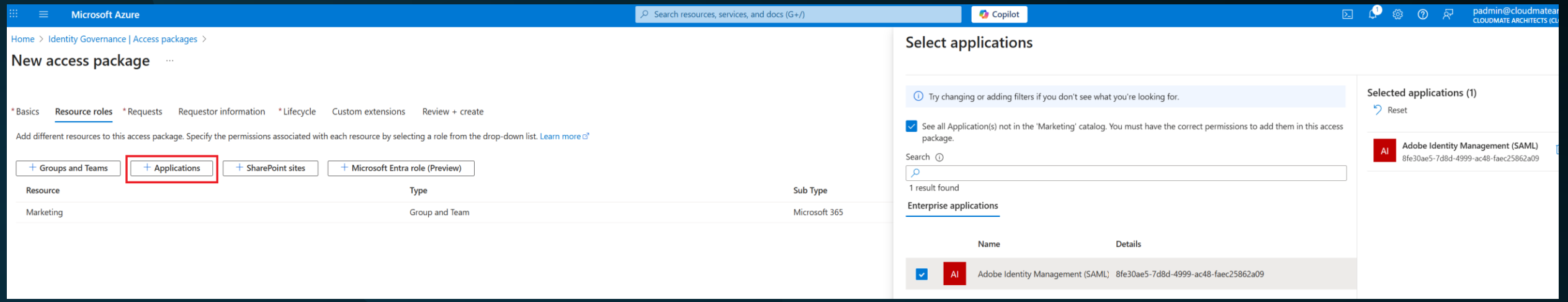
Added **Marketing** M365 Group, role **Member**.



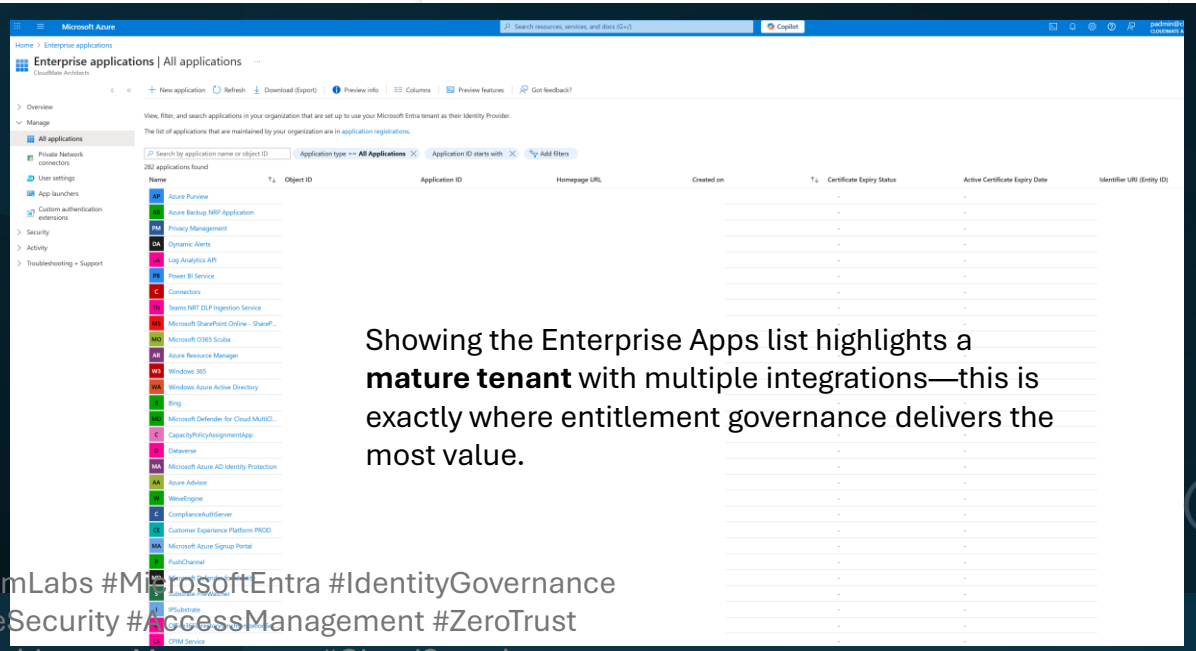
Least privilege note: Group membership is the backbone of access. Assign the **Member** role unless contribution/ownership is truly required. Keeping roles minimal reduces **privilege creep** and audit findings.

Add Application: **Adobe Identity Management (SAML)**

Added SAML app with role **User**.



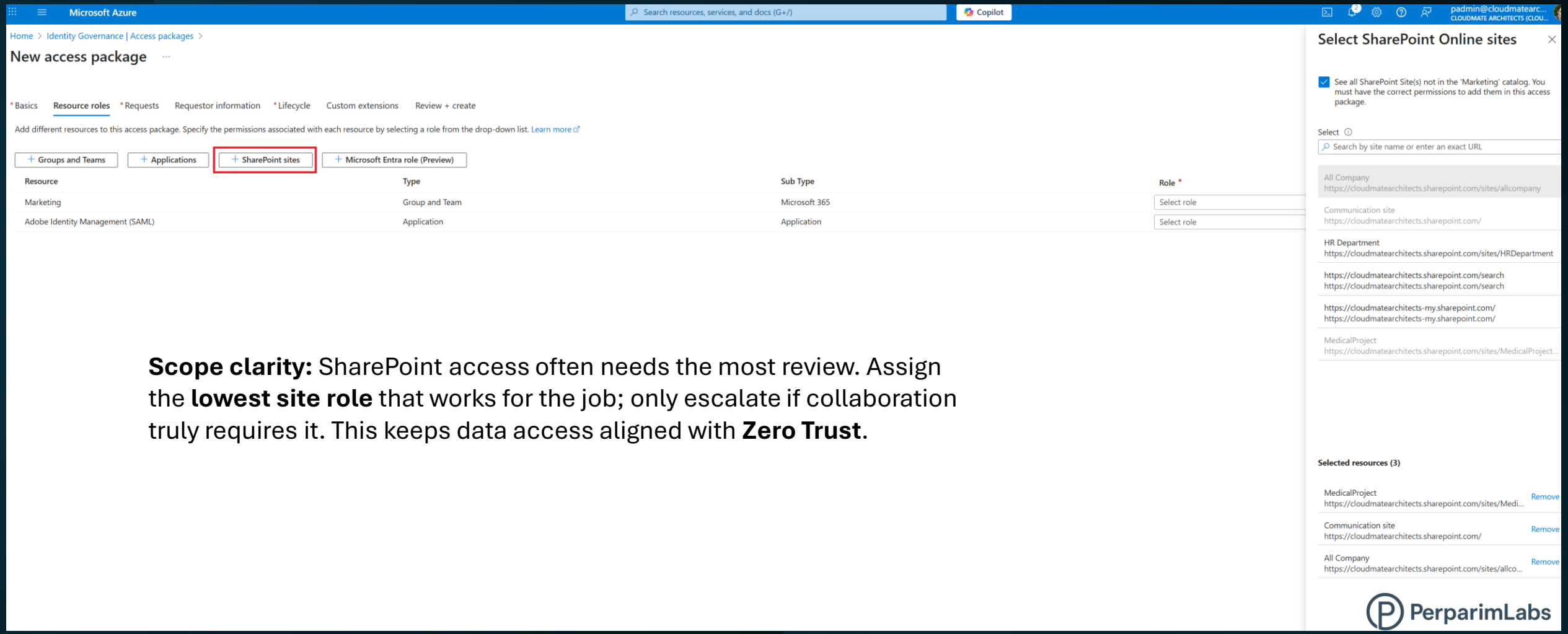
Governance tip: Adding **enterprise apps** to the package brings **SSO + app roles** under the same lifecycle controls (expiration, reviews). If the app supports **SCIM**, user provisioning/de-provisioning can be fully automated.



Showing the Enterprise Apps list highlights a **mature tenant** with multiple integrations—this is exactly where entitlement governance delivers the most value.

Add SharePoint sites

Added **MedicalProject**, **Communication site**, **All Company** with their default member roles.

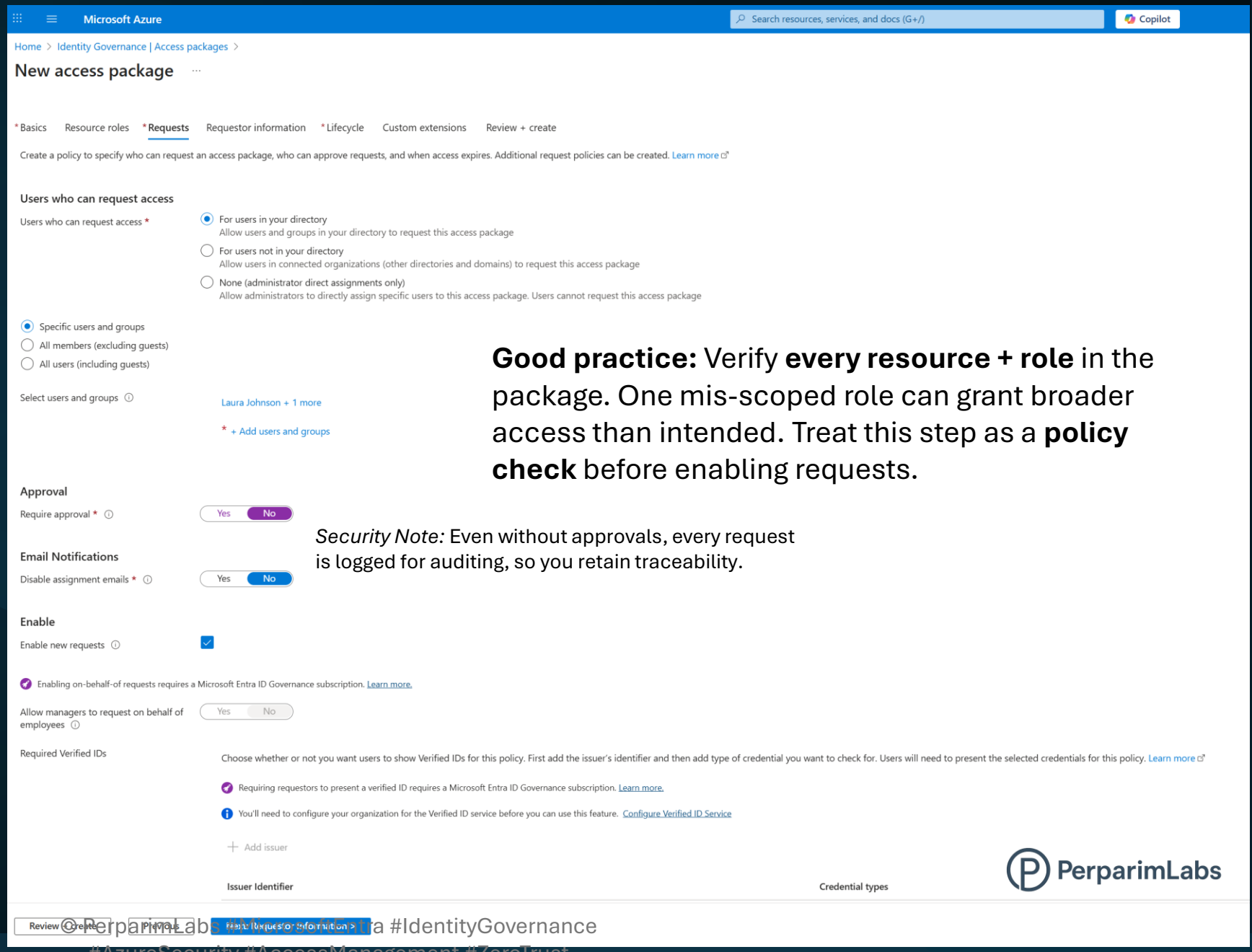


Scope clarity: SharePoint access often needs the most review. Assign the **lowest site role** that works for the job; only escalate if collaboration truly requires it. This keeps data access aligned with **Zero Trust**.

Who Can Request Access

- Defined users and groups allowed to request access to the package.
- Selected: **Perparim Abdullahu** and **Laura Johnson**
- Approval requirement: **No approval needed**
- This setup enables eligible users to self-request access while maintaining governance visibility.

Risk vs. friction: Limiting requestors (specific users/groups) maintains **Segregation of Duties**. Skipping approvals is fine for **low-risk** packages; for sensitive apps/data, enable **manager or app owner approval**.

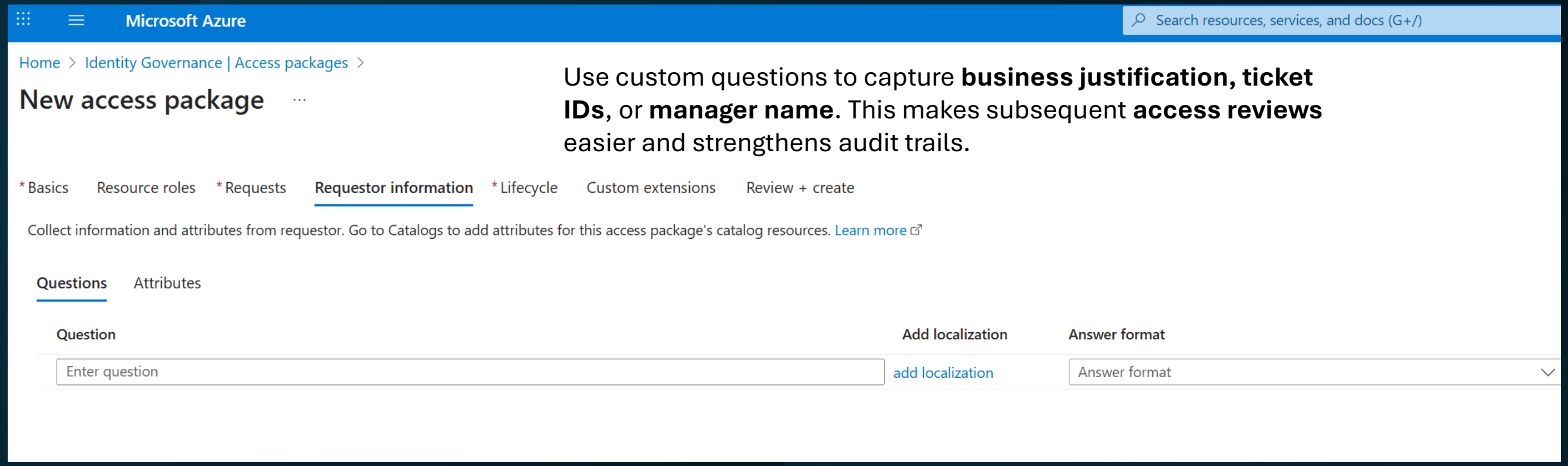


Good practice: Verify **every resource + role** in the package. One mis-scoped role can grant broader access than intended. Treat this step as a **policy check** before enabling requests.

Security Note: Even without approvals, every request is logged for auditing, so you retain traceability.

Requestor Information (Optional Questions)

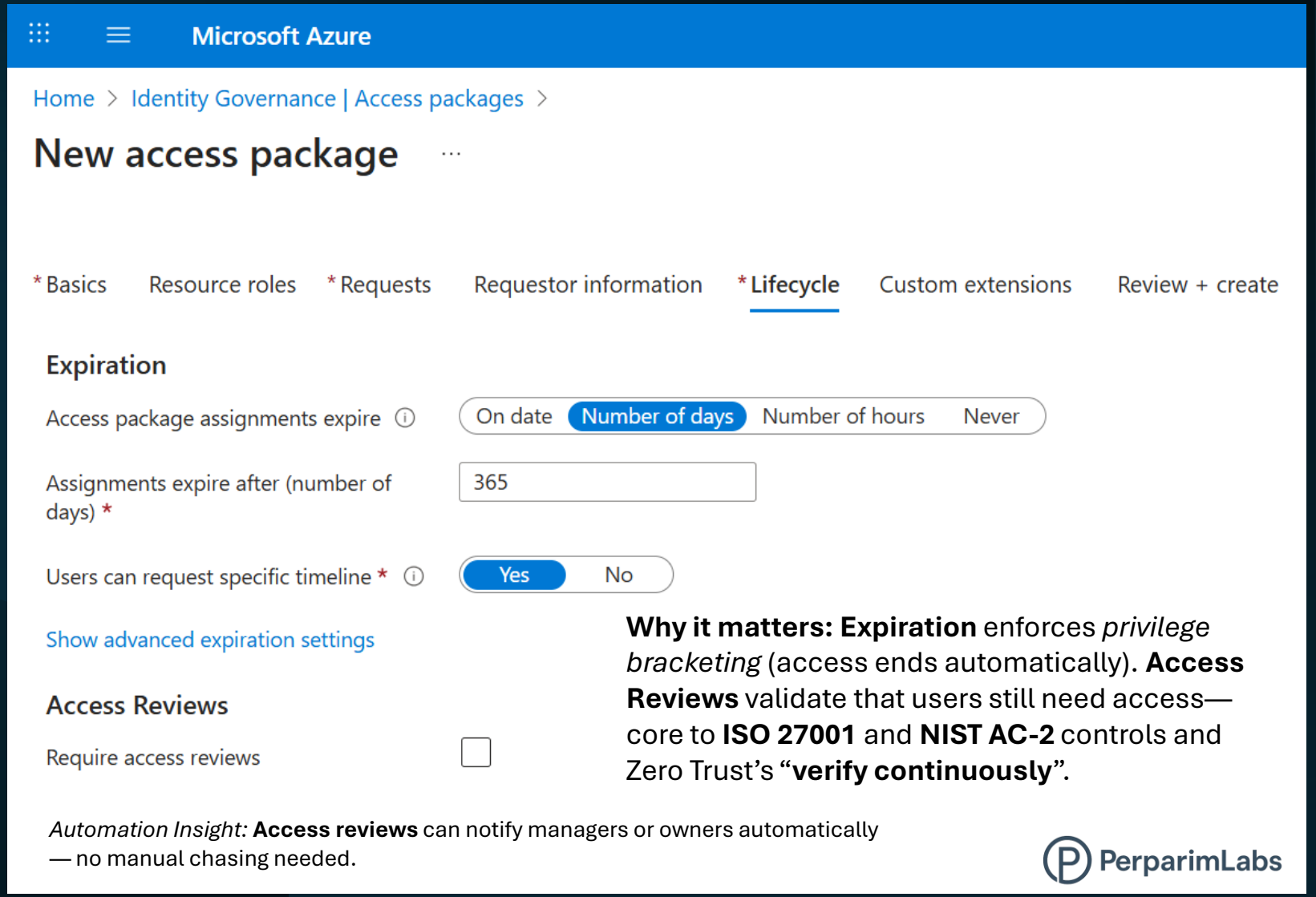
Here administrators can define additional attributes or custom questions to collect justification or business context from requesters.
In this demo, no custom questions were configured.



PerparimLabs

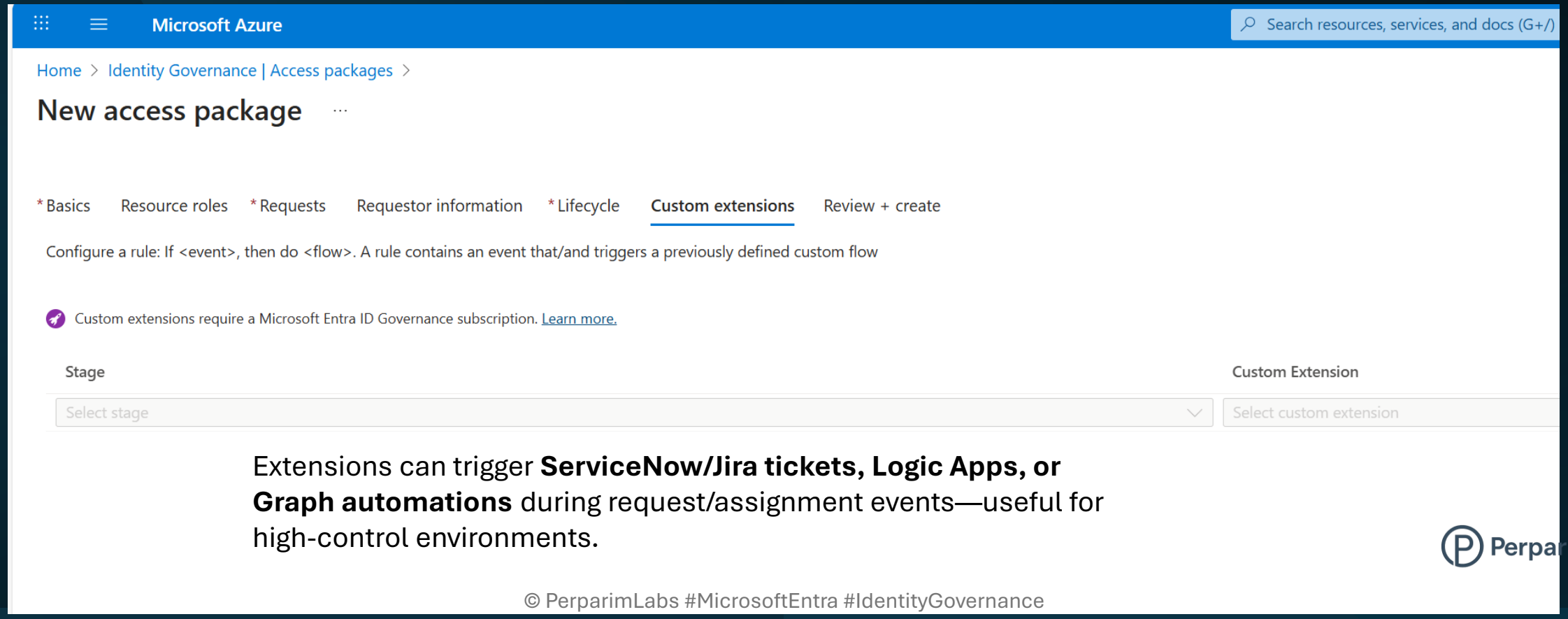
Lifecycle and Access Reviews

Configured lifecycle settings for ongoing access management.
Access reviews can be enabled to periodically verify if users still need access to the package — ensuring least-privilege and compliance.
In this configuration, no access review policy was set.



Custom Extensions

Custom extensions allow administrators to automate external workflows such as ServiceNow approvals or ticket generation.
This step was skipped in this demo.

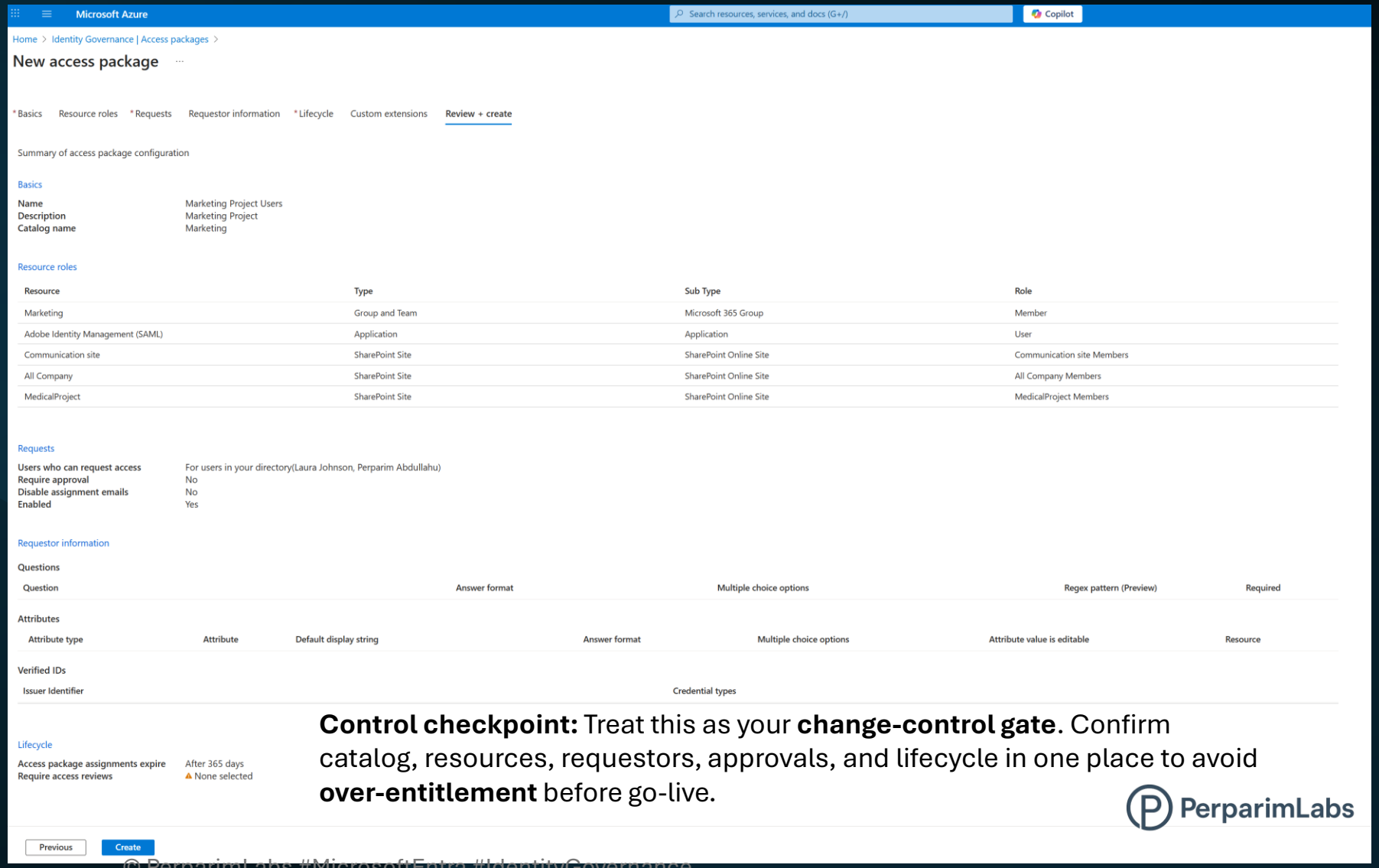


Extensions can trigger **ServiceNow/Jira tickets, Logic Apps, or Graph automations** during request/assignment events—useful for high-control environments.

PerparimLabs

Review and Confirm Configuration

Reviewed all configuration details for the *Marketing Project Users* access package, including catalog, resources, request settings, and lifecycle. Confirmed settings and created the package successfully.

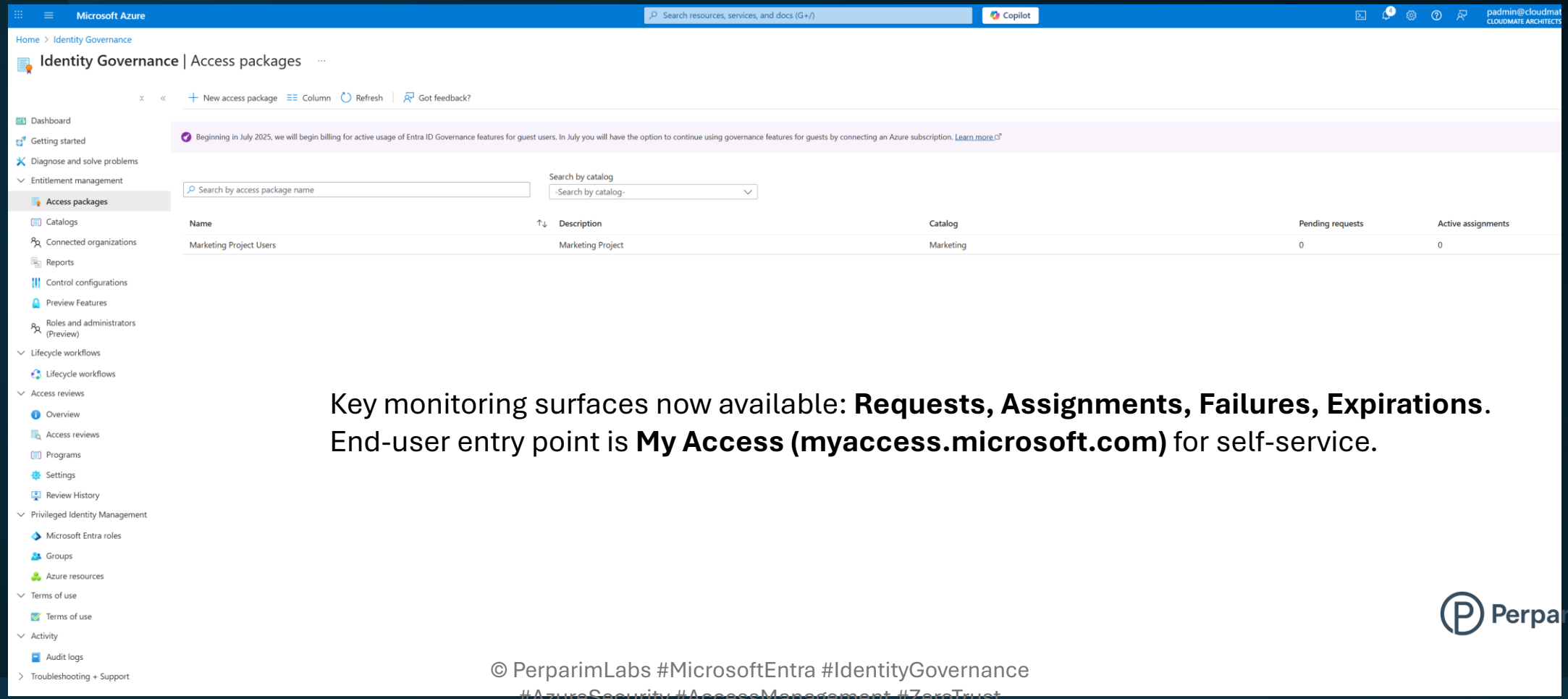


Control checkpoint: Treat this as your **change-control gate**. Confirm catalog, resources, requestors, approvals, and lifecycle in one place to avoid **over-entitlement** before go-live.

PerparimLabs

Access Package Deployed Successfully

The *Marketing Project Users* access package now appears under the Access Packages list.
It's ready for users to request access through the *My Access Portal*.



Key monitoring surfaces now available: **Requests, Assignments, Failures, Expirations**.
End-user entry point is **My Access (myaccess.microsoft.com)** for self-service.

PerparimLabs