Data Community Conference Switzerland 2025

Level up your Cost Observability with the new FinOps standard FOCUS and Microsoft Fabric

Alexandre PARÈS

www.data-community.ch

Agenda

What's FinOps?

Data consumed by the FinOps practice

What's FOCUS, the new FinOps standard?

FOCUS dataset ingestion and automation

FOCUS with MS Fabric, a great combo

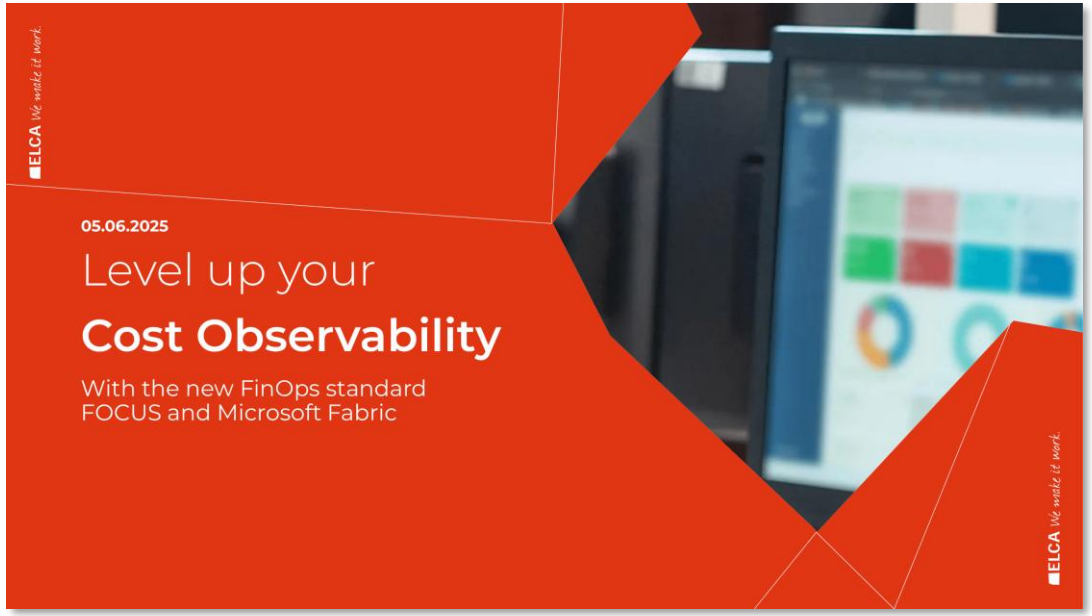
What's next?

ELCA We make it work.
ELCA We make it work.

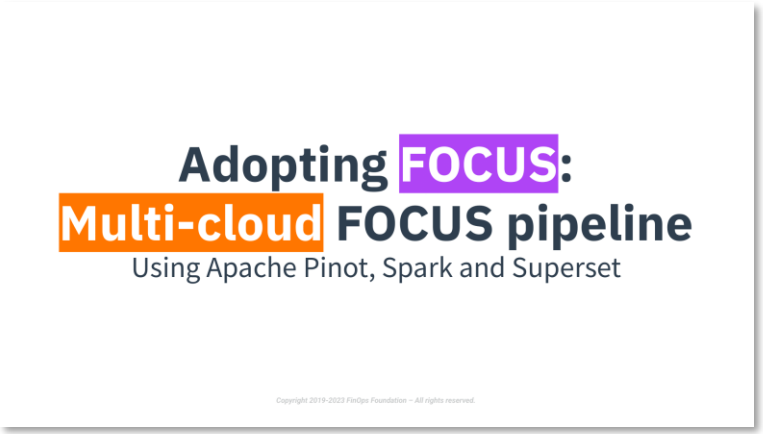
Who am I?

- 7 years of Cloud
- 4+ years of FinOps
- Engineer (MSc)
- Instructor
- Skydiving

(2025) Silicon Chalet Meetup: Oracle, Geneva



(2025) Azure Switzerland Bootcamp: Bern



(2025) Swiss FinOps Meetup: AWS, Zürich

(2024) Silicon Chalet Meetup: ELCA, Pully

(2023) May FinOps Summit: FinOps Foundation

What is FinOps?

ELCA We make it work.

FinOps Framework

By the FinOps Foundation

ELCA We make it work.

FinOps is an operational framework and cultural practice which

maximizes the business value of cloud and technology

enables timely data-driven decision making

creates financial accountability

through collaboration between engineering, finance, and business teams.

Copyright 2019-2025 FinOps Foundation – All rights reserved

FinOps is an operational framework and cultural practice which

maximizes the business value of cloud and technology

enables timely data-driven decision making

creates financial accountability

through collaboration between engineering, finance, and business teams.

Copyright 2019-2025 FinOps Foundation – All rights reserved

FinOps Framework

finops.org/framework

FinOps is an operational framework and cultural practice which **maximizes the business value of cloud and technology**, enables timely data-driven decision making, and creates financial accountability through collaboration between engineering, finance, and business teams.

Principles

- Teams need to collaborate
- Business value drives technology decisions
- Everyone takes ownership for their technology usage
- FinOps data should be accessible, timely, and accurate
- FinOps should be enabled centrally
- Take advantage of the variable cost model of the cloud

Business Strategy ↔ FinOps ↔ Technology Strategy

Scopes give context for Practitioners to apply the FinOps Framework.

- Public Cloud
- SaaS
- Data Center
- Licensing
- AI
- Custom

Core Personas are always engaged in a FinOps practice.

- Engineering
- FinOps Practitioner
- Finance
- Leadership
- Procurement
- Product

Allied Personas support a FinOps practice.

- ITFM
- ITAM
- ITSM
- Security
- Sustainability

Domains are the outcomes of a FinOps practice & **Capabilities** describe how to achieve them.

Understand Usage & Cost

- Data Ingestion
- Allocation
- Reporting & Analytics
- Anomaly Management

Quantify Business Value

- Planning & Estimating
- Benchmarking
- Forecasting
- Unit Economics
- Budgeting

Optimize Usage & Cost

- Architecting for Cloud
- Rate Optimization
- Cloud Sustainability
- Workload Optimization
- Licensing & SaaS

Manage the FinOps Practice

- FinOps Practice Operations
- Policy & Governance
- FinOps Assessment
- FinOps Tools & Services
- FinOps Education & Enablement
- Invoicing & Chargeback
- Onboarding Workloads
- Intersecting Disciplines

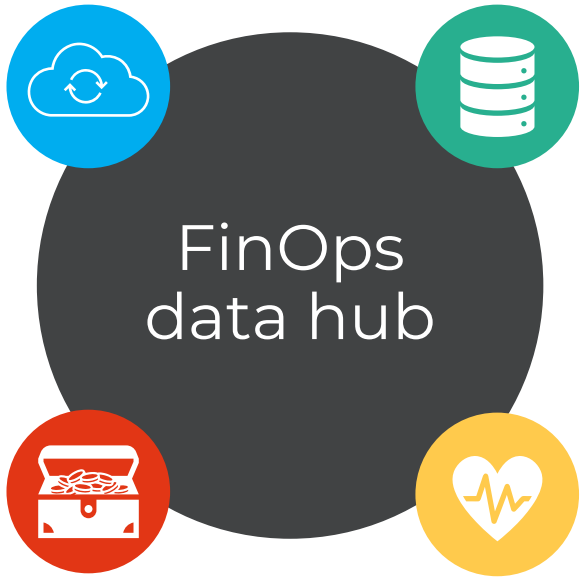
FinOps Foundation

FinOps Foundation Framework 2019 - 2025. Use of the FinOps Framework requires attribution to the FinOps Foundation under the cc by 4.0 license.

What data does the FinOps practice rely on?

ELCA We make it work.

The FinOps Practice relies on multiples data sources



Cloud Service Providers

Cost and Usage Reports (FOCUS),
Reservations & recommendations,
Cloud Budgets

CMDB

Owners, IT Assets,
Projects, Relations, etc.

Finance & Procurement

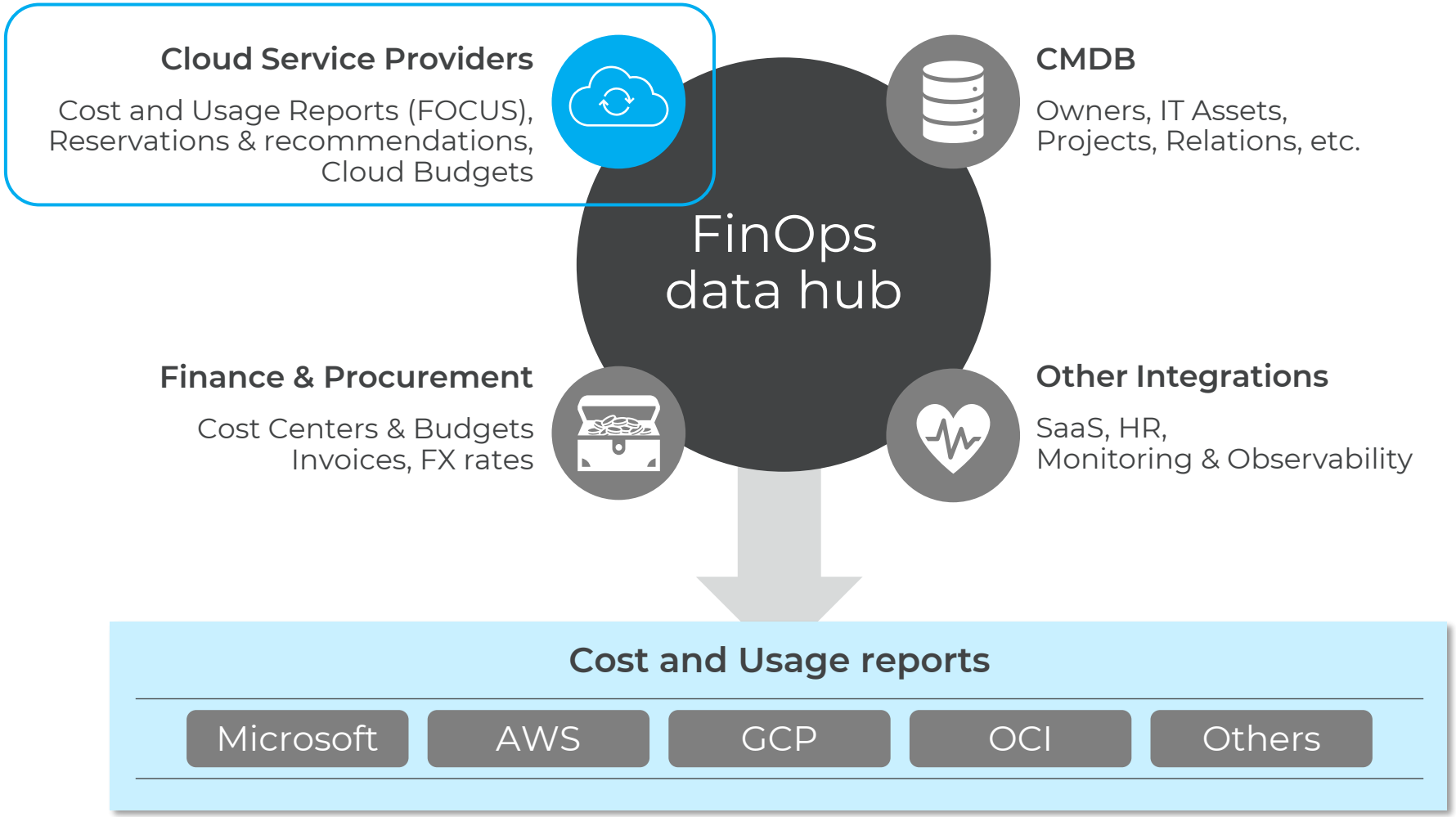
Cost Centers & Budgets
Invoices, FX rates

Other Integrations

SaaS, HR,
Monitoring & Observability

ELCA We make it work.

The FinOps Practice relies on multiples data sources



ELCA We make it work.

Do we need n integrations for n providers?

ELCA We make it work.

ELCA We make it work.

Cloud Service Providers have their own proprietary billing exports

Microsoft Azure

- Cost and usage (actual)
- Cost and usage (amortized)

AWS

- Cost and Usage Report (Legacy CUR)
- Cost and Usage Report (CUR) 2.0

Google Cloud

- Standard usage cost
- Detailed usage cost

Oracle Cloud

- Cost reports

For clarity, only exports related to costs and usages are displayed (carbon footprint, reservations, committed use discounts, etc. aren't displayed)

ELCA We make it work.

What is FOCUS?

FOCUS is an open-source technical specification for cloud billing data.

In plain language, it is **the unifying format for billing data.**

FOCUS normalizes billing data across vendors, which reduces time and effort needed to begin FinOps analysis.

Example: **What does "Billed Cost" mean?**

Prior to FOCUS, each cloud provider had a different definition:

AWS: lineItem/NetUnblendedCost

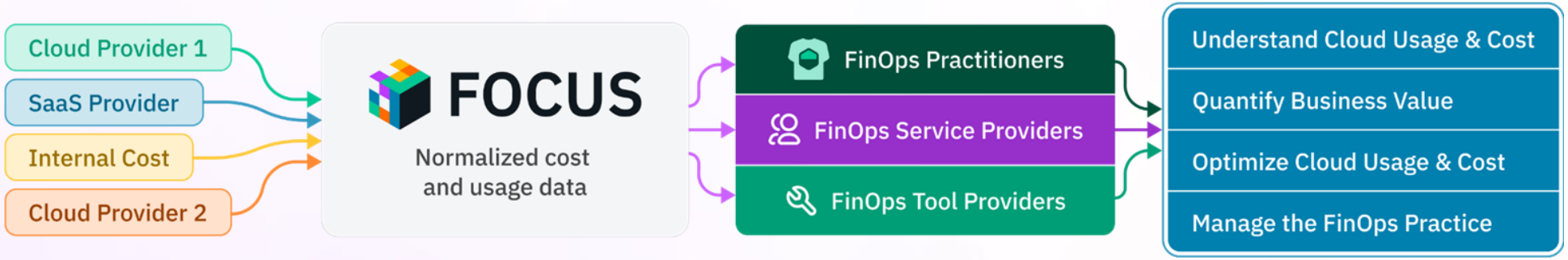
Azure: CostInBillingCurrency (from Actual dataset)

GCP: cost + credits.amount

OCI: cost/myCostOverage

Copyright 2019-2025 FinOps Foundation – All rights reserved

FOCUS: The Unifying Format for Billing Data



Billing data from vendors is generated...

...in the FOCUS format...

...for ingestion by Practitioners...

...to achieve FinOps Domains.

Copyright 2019-2025 FinOps Foundation – All rights reserved

How does FOCUS help?

FOCUS normalizes billing data, which reduces time and effort to start FinOps analysis.

Copyright 2019-2025 FinOps Foundation – All rights reserved

ELCA We make it work.

FOCUS Support*

FOCUS™

About | Specification | Adoption | Get Involved | Get Certified | Get Started

Getting Started with FOCUS™ Datasets

Download your data in the FOCUS format:

	FOCUS™ Version Supported	Conformance Gap Report	
Microsoft (Azure)	v1.2	✓	→
Google Cloud	v1.0	×	→
AWS	v1.0	✓	→
Oracle Cloud	v1.0	×	→
Tencent Cloud	v1.0	✓	→
Alibaba Cloud	v1.0	✓	↗
Databricks	v1.2	×	
Grafana	v1.2	×	

focus.finops.org/get-started

*As of 2025-11-05

ELCA We make it work.

FOCUS Support*

Provider	FOCUS Version & Status
✅ Microsoft Azure	1.0-preview (preview) 1.0/1.0r2 (GA) 1.2-preview
✅ AWS	1.0-preview 1.0 (GA)
⚠️ GCP	1.0 BigQuery view (GA) 1.0 BigQuery immutable dataset (private preview)
✅ OCI	1.0 (GA)
❌ Databricks	1.2 System table (closed private preview)

*As of 2025-11-05

ELCA We make it work.

FOCUS v1.0 columns

Account
- Billing Account ID
- Billing Account Name
- Sub Account ID
- Sub Account Name

Charge Origination
- Invoice Issuer
- Provider
- Publisher

Pricing
- Pricing Category
- Pricing Quantity
- Pricing Unit

SKU
- SKU ID
- SKU Price ID

Billing
- Billed Cost
- Billing Currency
- Consumed Quantity
- Consumed Unit
- Contracted Cost
- Contracted Unit Price
- Effective Cost
- List Cost
- List Unit Price

Location
- Availability Zone
- Region ID
- Region Name

Service
- Service Category
- Service Name

Charge
- Charge Category
- Charge Class
- Charge Description
- Charge Frequency

Commitment Discount
- Commitment Discount Category
- Commitment Discount ID
- Commitment Discount Name
- Commitment Discount Status
- Commitment Discount Type

Resource
- Resource ID
- Resource Name
- Resource Type
- Tags

Timeframe
- Billing Period End
- Billing Period Start
- Charge Period End
- Charge Period Start

Source: FOCUS Column Library – The FinOps Foundation

ELCA We make it work.

FOCUS v1.2 columns

Account
- Billing Account ID
- Billing Account Name
- Billing Account Type
- Sub Account ID
- Sub Account Name
- Sub Account Type

Charge Origination
- Invoice ID
- Invoice Issuer
- Provider
- Publisher

Pricing
- Pricing Category
- Pricing Currency
- Pricing Currency Contracted Unit Price
- Pricing Currency Effective Cost
- Pricing Currency List Unit Price
- Pricing Quantity
- Pricing Unit

Timeframe
- Billing Period End
- Billing Period Start
- Charge Period End
- Charge Period Start

Billing
- Billed Cost
- Billing Currency
- Consumed Quantity
- Consumed Unit
- Contracted Cost
- Contracted Unit Price
- Effective Cost
- List Cost
- List Unit Price

Location
- Availability Zone
- Region ID
- Region Name

Resource
- Resource ID
- Resource Name
- Resource Type
- Tags

SKU
- SKU ID
- SKU Meter
- SKU Price Details
- SKU Price ID

Capacity Reservation
- Capacity Reservation ID
- Capacity Reservation Status

Charge
- Charge Category
- Charge Class
- Charge Description
- Charge Frequency

Commitment Discount
- Commitment Discount Category
- Commitment Discount ID
- Commitment Discount Name
- Commitment Discount Quantity
- Commitment Discount Status
- Commitment Discount Type
- Commitment Discount Unit

Service
- Service Category
- Service Name
- Service Subcategory

Source: FOCUS Column Library – The FinOps Foundation

Let’s automate the FOCUS™ exports

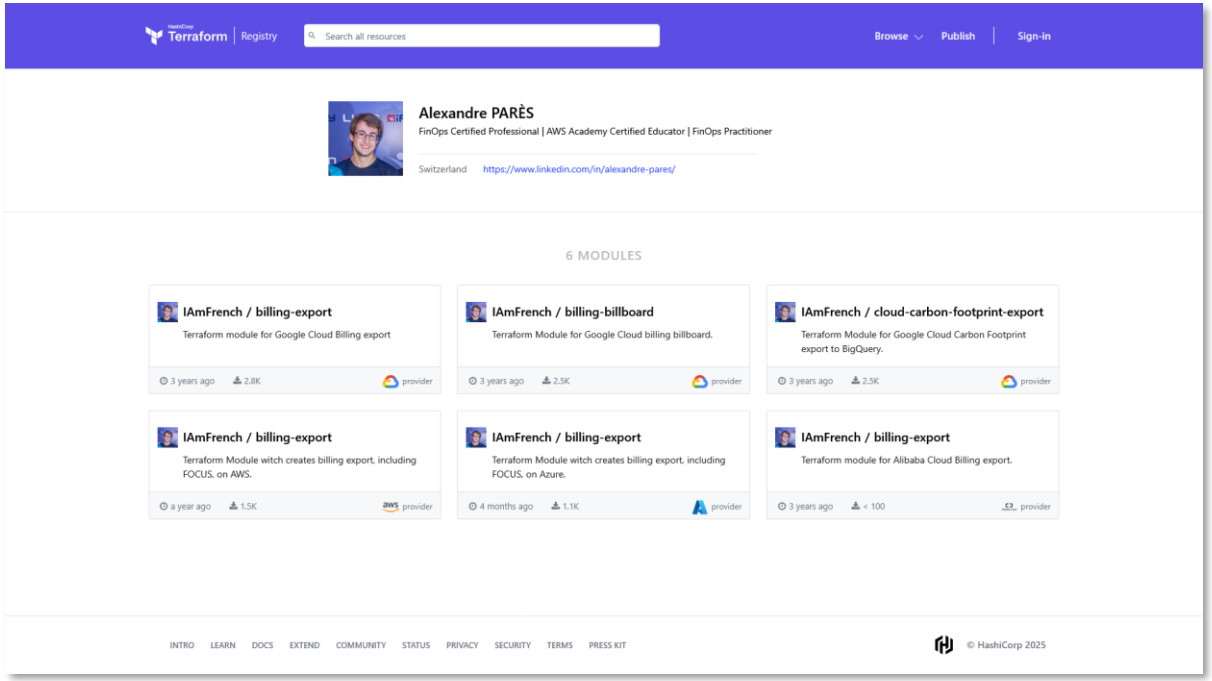
ELCA We make it work.

Getting started with FOCUS™: Create the exports

AWS | Azure | OCI | Other

- Manual, via the Web Console / Portal
- Automated, using IaC

ℹ️ Best practice to use Infrastructure as Code (IaC)



Terraform Modules to create billing exports – Terraform Registry

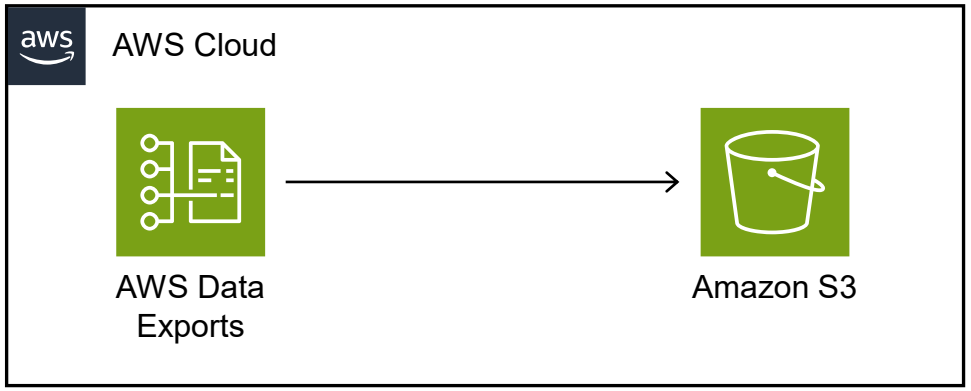
ELCA We make it work.

Getting started with AWS & FOCUS

AWS | Azure | OCI | Other

```
# FOCUS billing export for AWS
module "aws_billing_export" {
  source  = "IAmFrench/billing-export/aws"
  version = "1.0.4"

  # Name of the S3 bucket to create exports in
  s3_bucket_name = "focusexport1a2b3c4d"
  # Type of the export
  export_type = "FOCUS"
  # Version of the export
  export_version = "1.0"
  # Name of the export
  export_name = "focus-v1-0-export"
  # Prefix of the export
  export_s3_prefix = "focus/v1.0/1234556789"
}
```

Terraform Module to create a FOCUS billing exports on AWS – GitHub



ELCA We make it work.

Getting started with Azure and FOCUS

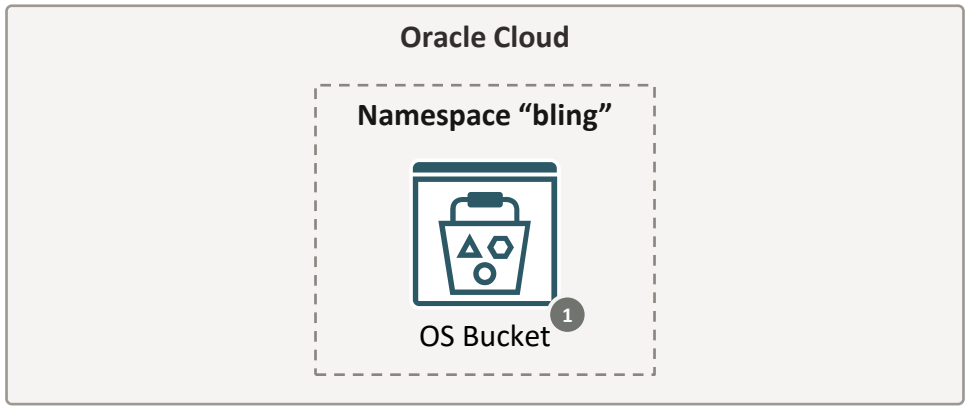
AWS | **Azure** | OCI | Other

```
# FOCUS billing export for Azure
module "azurerm_billing_export" {
  source = "IAmFrench/billing-export/azurerm"
  version = "1.1.1"

  storage_account_name = "focusexport1a2b3c4d"

  export_type          = "FOCUS"
  export_version       = "1.0r2"
  export_scope_and_id = {
          scope        = "billing-account"
          id           = "123456789"
  }
  export_name = "focus-export-for-billing-account-123456789"
  export_directory     = "v1.0r2/billing_account_123456789"
  [...]
}
```

Terraform Module to create a FOCUS billing exports on Azure – GitHub

Create a FOCUS export for OCI using IaC

AWS | Azure | **OCI** | Other

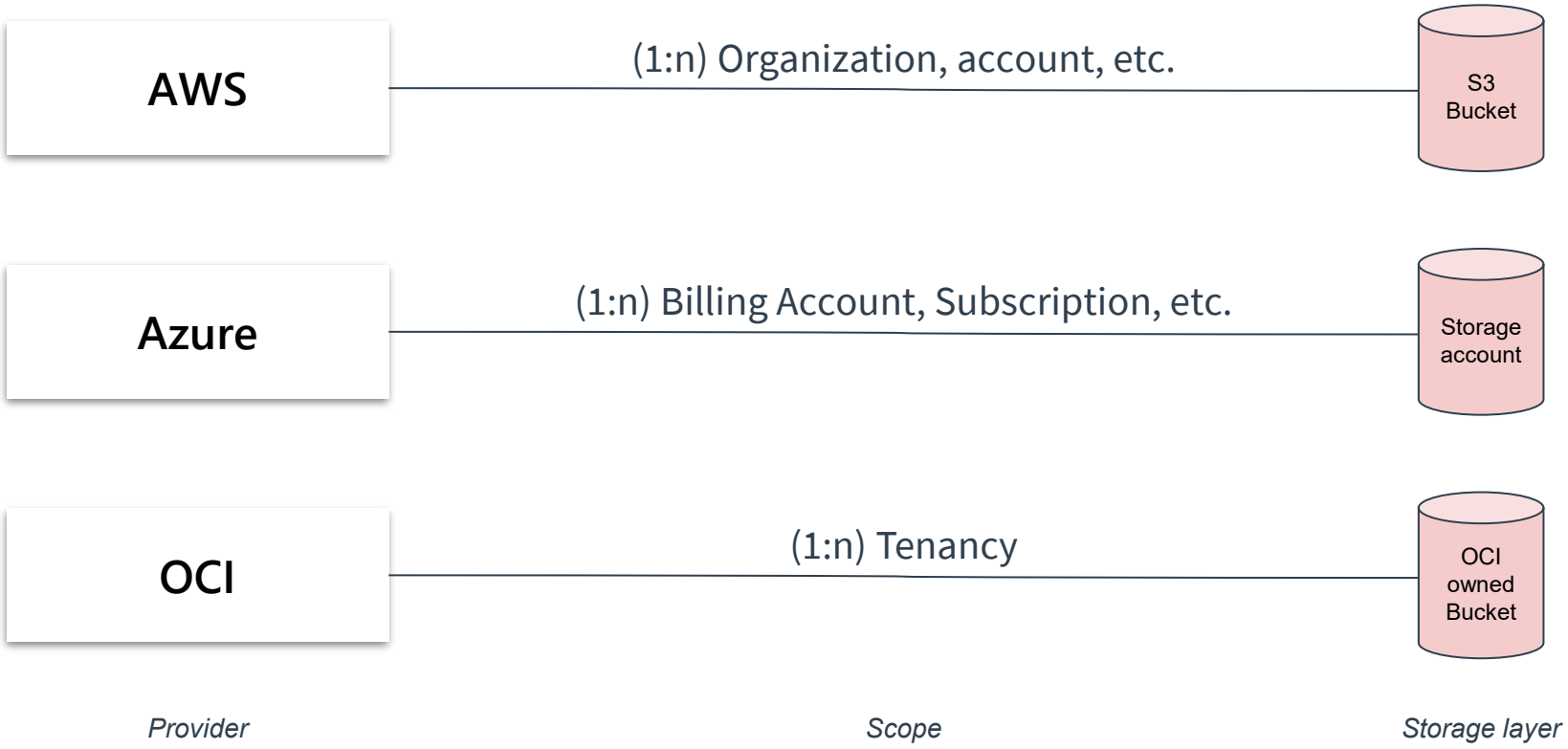


1
```
OS Bucket region = Tenancy Home Region
Bucket name = Tenancy OCID (ocid1.tenancy.oc1..aaaaaa…)
Prefix = “FOCUS Reports”
```

✅ FOCUS export is available with no additional configuration

ELCA We make it work.

You can create multiples FOCUS exports per provider!



ELCA We make it work.

Now what?

ELCA We make it work.

Multiple File structures

AWS

```
focus/v1.0/123456789/focus-v1-0-export/data
├── BILLING_PERIOD=2024-01
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
├── BILLING_PERIOD=2024-02
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-0002.snappy.parquet
├── BILLING_PERIOD=2024-03
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
├── BILLING_PERIOD=2024-04
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
...
```

Azure

```
v1.0r2/billing_account_123456789
├── 20240101-20240131/21232f24-e863-446d-a196-87cf094b9b53
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240201-20240229/67ebeeb3-6d5d-4b3d-a8a5-7c9fa8aef388
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240301-20240331/239313a6-a1c0-4a9b-9d54-7a548c953d85
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_1_000n.snappy.parquet
├── 20240401-20240430/c965e5d5-f647-4b2d-a371-3da8ee9bae4d
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
...
```

OCI

```
FOCUS Reports
├── 2024
│    ├── 07
│    │  ├── 01
│    │  │   ├── 000100000123456-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123567-00001.csv.gz
│    │  ├── 02
│    │  │   ├── 000100000123678-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123789-00001.csv.gz
│    │  ├── ...
│    ├── 08
│    │  ├── 01
│    │  │   ├── 000100000123890-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123512-00001.csv.gz
...
```


ELCA We make it work.

Multiple File formats

AWS

```
focus/v1.0/123456789/focus-v1-0-export/data
├── BILLING_PERIOD=2024-01
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
├── BILLING_PERIOD=2024-02
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-0002.snappy.parquet
├── BILLING_PERIOD=2024-03
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
├── BILLING_PERIOD=2024-04
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
...
```

parquet (snappy)
csv (gzip)

Azure

```
v1.0r2/billing_account_123456789
├── 20240101-20240131/21232f24-e863-446d-a196-87cf094b9b53
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240201-20240229/67ebeeb3-6d5d-4b3d-a8a5-7c9fa8aef388
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240301-20240331/239313a6-a1c0-4a9b-9d54-7a548c953d85
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_1_000n.snappy.parquet
├── 20240401-20240430/c965e5d5-f647-4b2d-a371-3da8ee9bae4d
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
...
```

parquet (snappy)
csv

OCI

```
FOCUS Reports
├── 2024
│    ├── 07
│    │  ├── 01
│    │  │   ├── 000100000123456-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123567-00001.csv.gz
│    │  ├── 02
│    │  │   ├── 000100000123678-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123789-00001.csv.gz
│    │  ├── ...
│    ├── 08
│    │  ├── 01
│    │  │   ├── 000100000123890-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123512-00001.csv.gz
...
```

csv (gzip)

ELCA We make it work.

Multiple FOCUS versions (1.0-preview, 1.0, 1.0r2, etc.)

AWS

```
focus/v1.0/123456789/focus-v1-0-export/data
├── BILLING_PERIOD=2024-01
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
├── BILLING_PERIOD=2024-02
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-0002.snappy.parquet
├── BILLING_PERIOD=2024-03
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
├── BILLING_PERIOD=2024-04
│    ├── focus-v1-0-export-0001.snappy.parquet
│    ├── ...
│    └── focus-v1-0-export-000n.snappy.parquet
...
```

1.0-preview
1.0

Azure

```
v1.0r2/billing_account_123456789
├── 20240101-20240131/21232f24-e863-446d-a196-87cf094b9b53
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240201-20240229/67ebeeb3-6d5d-4b3d-a8a5-7c9fa8aef388
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240301-20240331/239313a6-a1c0-4a9b-9d54-7a548c953d85
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_1_000n.snappy.parquet
├── 20240401-20240430/c965e5d5-f647-4b2d-a371-3da8ee9bae4d
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
...
```

1.0
1.0r2

OCI

```
FOCUS Reports
├── 2024
│    ├── 07
│    │  ├── 01
│    │  │   ├── 000100000123456-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123567-00001.csv.gz
│    │  ├── 02
│    │  │   ├── 000100000123678-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123789-00001.csv.gz
│    │  ├── ...
│    ├── 08
│    │  ├── 01
│    │  │   ├── 000100000123890-00001.csv.gz
│    │  │   ├── ...
│    │  │   └── 000100000123512-00001.csv.gz
...
```

No options to select the dataset version

ELCA We make it work.

How to automate FOCUS dataset retrieval?

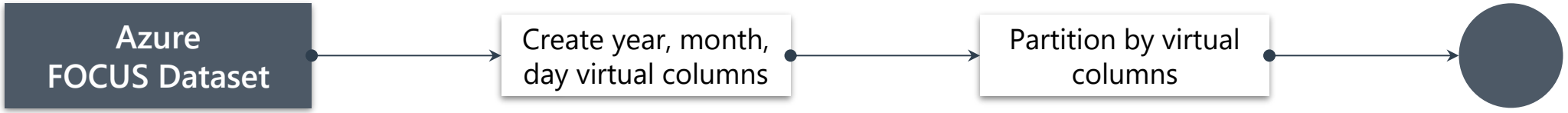
ELCA We make it work.

ELCA We make it work.

ELCA We make it work.

Step 1: Normalize the file structure and file format

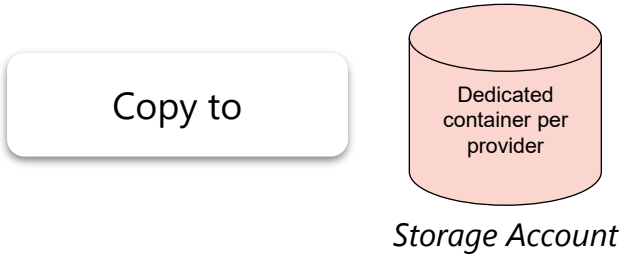
Example with an Azure dataset



For all files of a specific month

```
v1.0r2/billing_account_123456789
├── 20240101-20240131/21232f24-e863-446d-a196-87cf094b9b53
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240201-20240229/67ebeeb3-6d5d-4b3d-a8a5-7c9fa8aef388
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
├── 20240301-20240331/239313a6-a1c0-4a9b-9d54-7a548c953d85
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_1_000n.snappy.parquet
├── 20240401-20240430/c965e5d5-f647-4b2d-a371-3da8ee9bae4d
│    ├── part_0_0001.snappy.parquet
│    ├── ...
│    └── part_n_0001.snappy.parquet
...
```

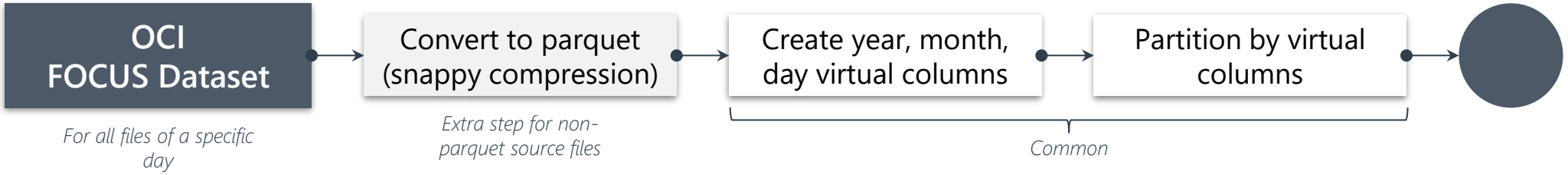
```
year=2024
├── month=1
│    ├── day=1
│    │    ├── file1.snappy.parquet
│    │    ├──  ...
│    │    └── filen.snappy.parquet
│    ├── day=2
│    │    ├── file1.snappy.parquet
│    │    ├──  ...
│    │    └── filen.snappy.parquet
│    ├── ...
├── month=2
│    ├── day=1
│    │    ├── file1.snappy.parquet
│    │    ├──  ...
│    │    └── filen.snappy.parquet
...
```



ELCA We make it work.

Step 1: Normalize the file structure and file format

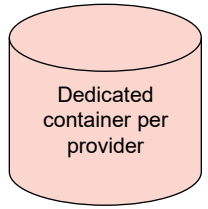
Example with an OCI dataset




```
FOCUS Reports
├── 2024
│   ├── 07
│   │   ├── 01
│   │   │   ├── 000100000123456-00001.csv.gz
│   │   │   ├── ...
│   │   │   └── 000100000123567-00001.csv.gz
│   │   ├── 02
│   │   │   ├── 000100000123678-00001.csv.gz
│   │   │   ├── ...
│   │   │   └── 000100000123789-00001.csv.gz
│   │   ├── ...
│   ├── 08
│   │   ├── 01
│   │   │   ├── 000100000123890-00001.csv.gz
│   │   │   ├── ...
│   │   │   └── 000100000123512-00001.csv.gz
...
```

```
year=2024
├── month=1
│   ├── day=1
│   │   ├── file1.snappy.parquet
│   │   ├ ...
│   │   └── filen.snappy.parquet
│   ├── day=2
│   │   ├── file1.snappy.parquet
│   │   ├ ...
│   │   └── filen.snappy.parquet
│   ├── ...
├── month=2
│   ├── day=1
│   │   ├── file1.snappy.parquet
│   │   ├ ...
│   │   └── filen.snappy.parquet
...
```

Copy to



Storage Account

Step 2: (trust but) Verify your FOCUS datasets

Example with an OCI dataset

```
# Display "native" formats of DateTime columns
duckdb -c "select BillingPeriodEnd, BillingPeriodStart, ChargePeriodEnd, ChargePeriodStart from
read_csv('0001000001234567-00001.csv.gz', types={'BillingPeriodEnd': 'VARCHAR', 'BillingPeriodStart':
'VARCHAR'}) limit 1"
```

BillingPeriodEnd varchar	BillingPeriodStart varchar	ChargePeriodEnd varchar	ChargePeriodStart varchar
2024-08-31T23:59:59.999Z	2024-08-01T00:00:00.000Z	2024-08-16T01:00Z	2024-08-16T00:00Z

2.12.4. (ChargePeriodEnd) Content constraints

Constraint	Value
Column type	Dimension
Feature level	Mandatory
Allows nulls	False
Data type	Date/Time
Value format	Date/Time Format

3.3.4. (Date/Time) Requirements

[...]

- *Date/time values format MUST be aligned with ISO 8601 standard [...]*
- *Use two-digit hours (HH), minutes (mm), and seconds (ss).*

[...]

Source: FOCUS 1.0 Specification - GitHub

ELCA We make it work.

Step 2: (trust but) Verify your FOCUS datasets

Example with an OCI dataset

```
# Display "native" formats of DateTime columns
duckdb -c "select BillingPeriodEnd, BillingPeriodStart, ChargePeriodEnd, ChargePeriodStart from
read_csv('0001000001234567-00001.csv.gz', types={'BillingPeriodEnd': 'VARCHAR', 'BillingPeriodStart':
'VARCHAR'}) limit 1"
```

BillingPeriodEnd varchar	BillingPeriodStart varchar	ChargePeriodEnd varchar	ChargePeriodStart varchar
2024-08-31T23:59:59.999Z	2024-08-01T00:00:00.000Z	2024-08-16T01:00Z	2024-08-16T00:00Z

2.6.3 (BillingPeriodEnd) Description

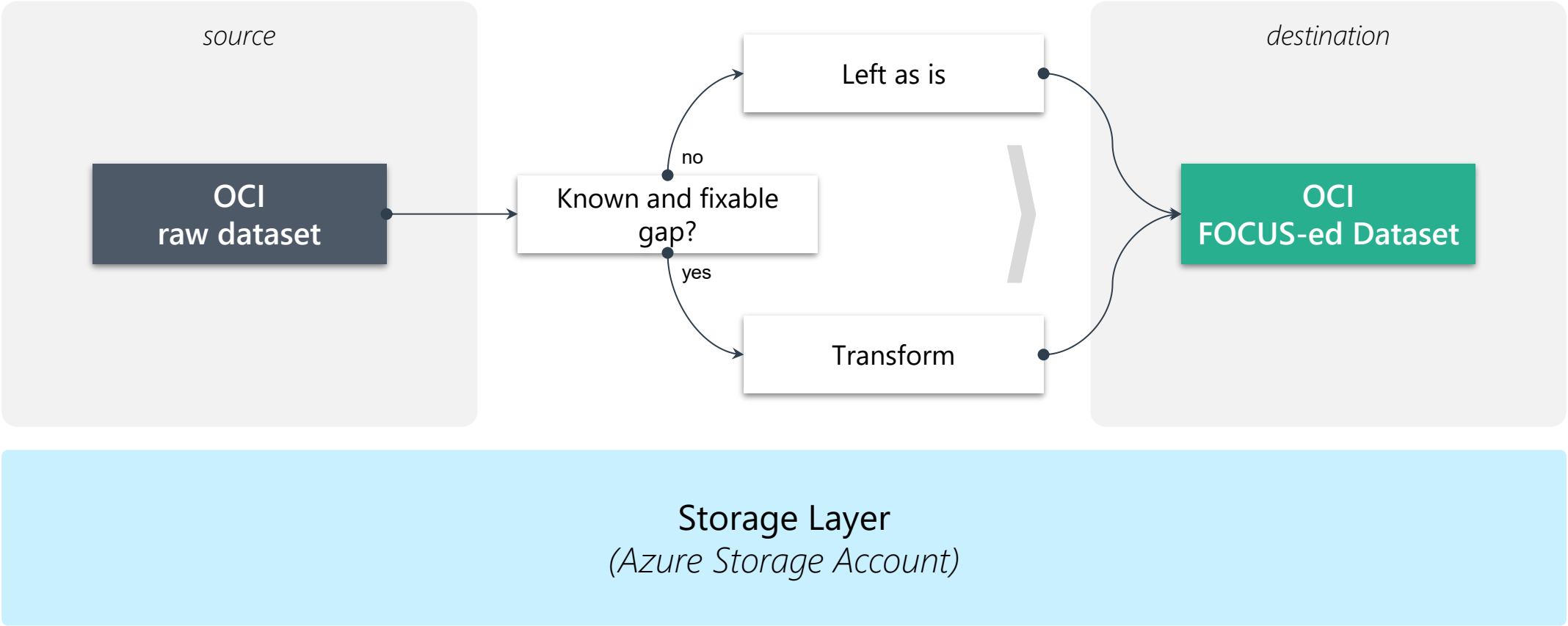
The exclusive end date and time of a billing period.

Source: FOCUS 1.0 Specification - GitHub

ELCA We make it work.

Step 2: (trust but) Verify your FOCUS datasets

Example with an OCI dataset

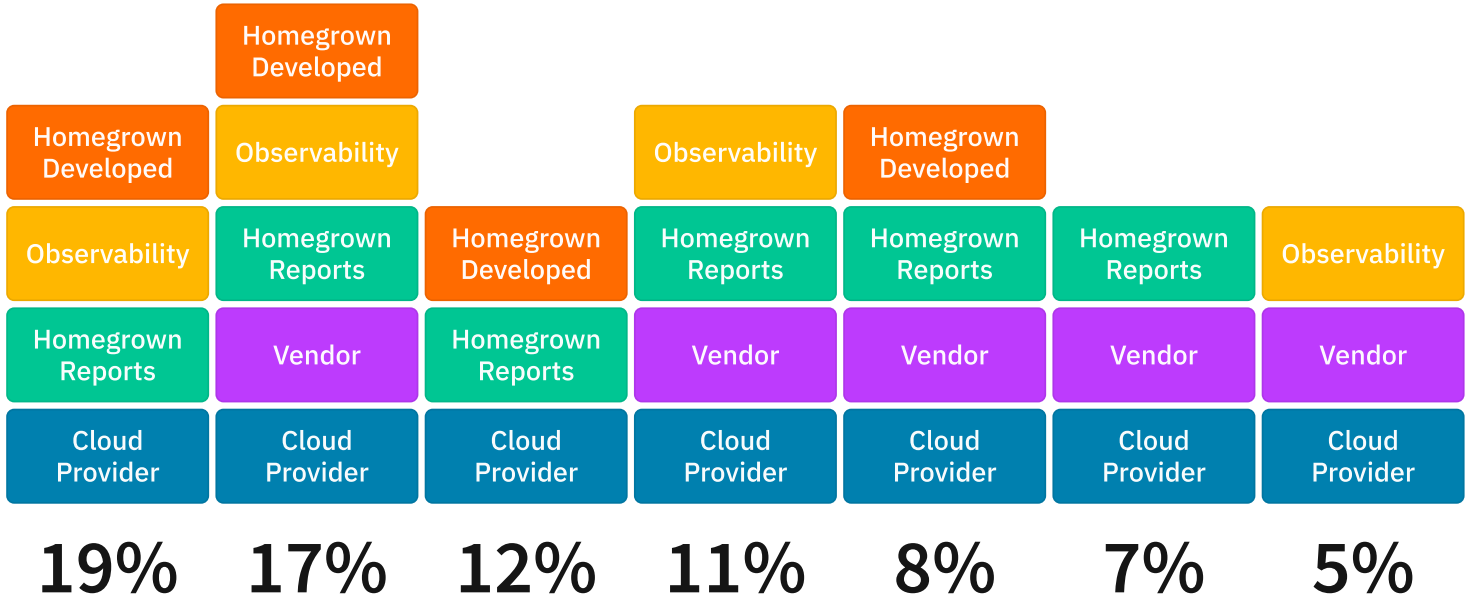


FOCUS with _____

ELCA We make it work.

Homegrown reports and tools are widely used

Practitioners use a wide range of combinations for tooling to fit around their organizational needs.



56% of Practitioners use Homegrown developed tools

74% of Practitioners use Homegrown Reports

Source: 2025 State of FinOps - Tooling: Wide range of combinations, CSP or Vendor tools often chosen as the Primary tool, homegrown reporting common in stack - The FinOps Foundation

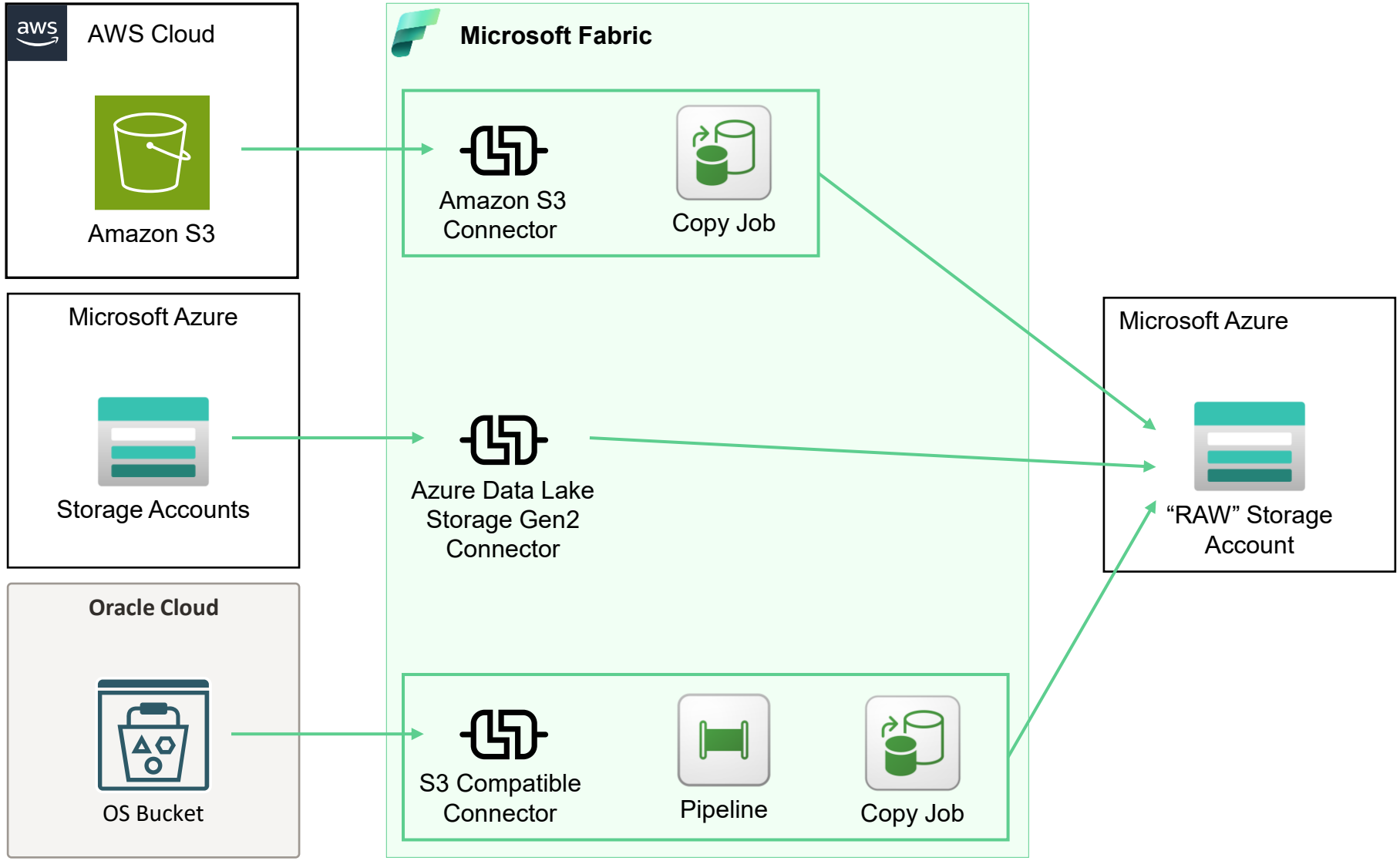
ELCA We make it work.

ELCA We make it work.

Let’s build our FinOps Hub in Microsoft Fabric using FOCUS datasets

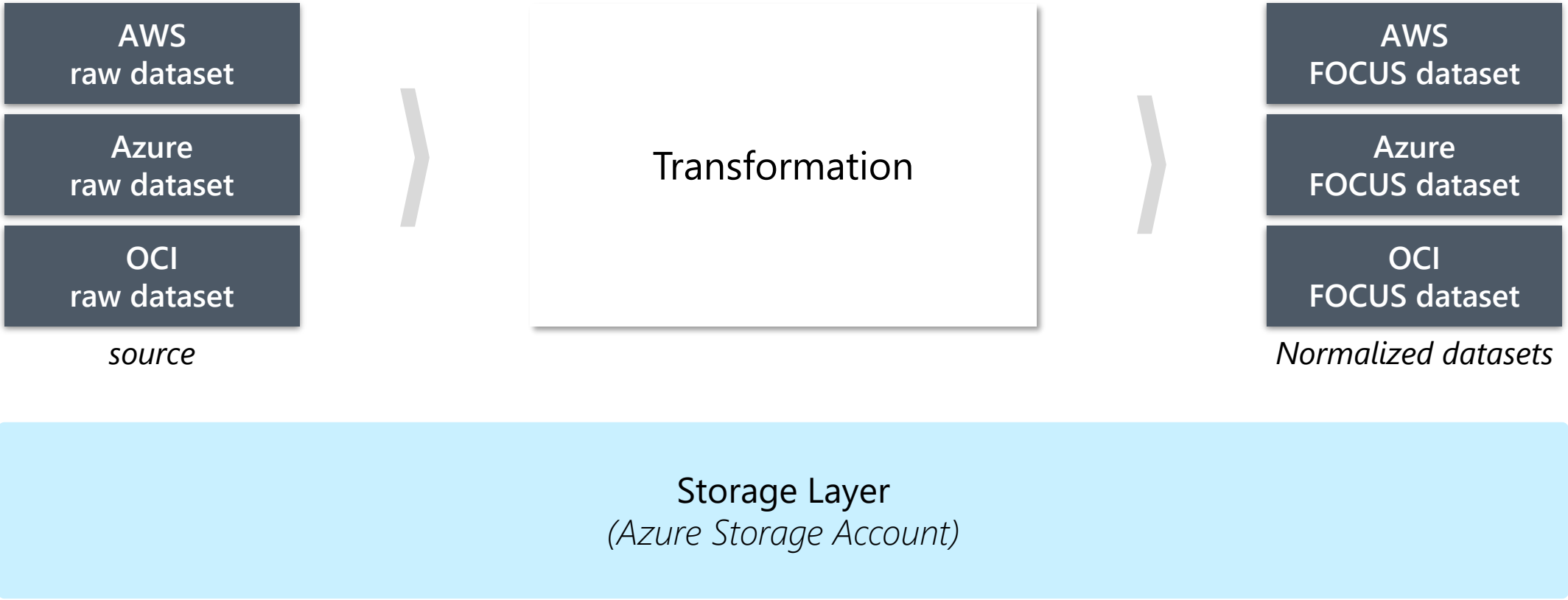
ELCA We make it work.

Ingestion into Microsoft Fabric



ELCA We make it work.

Transformation



ELCA We make it work.

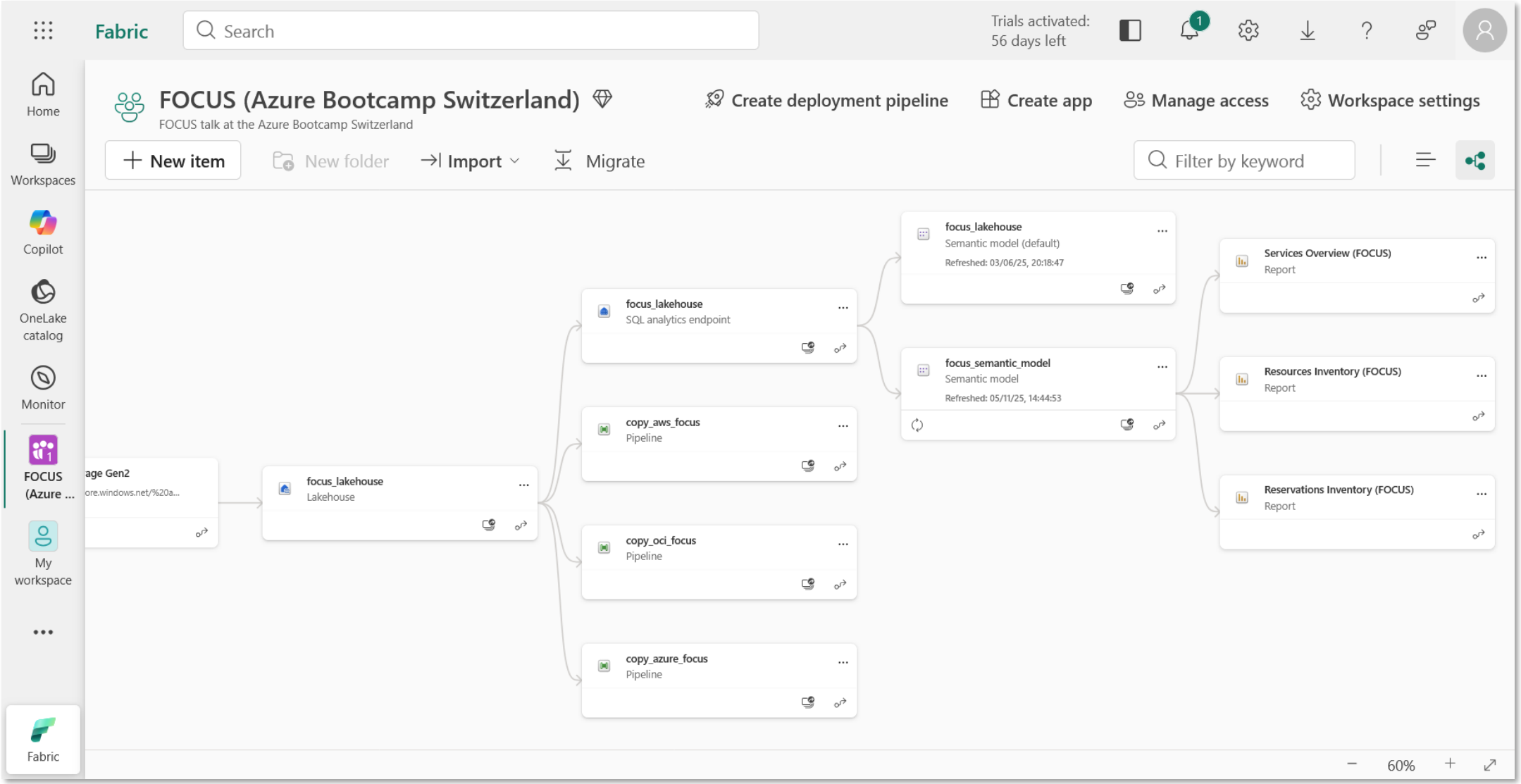
ELCA We make it work.

Main components of Microsoft Fabric

One Lake	Data Warehouse	Power BI
Storage Layer	*Processing & Query Layer*	*Visualization Layer*

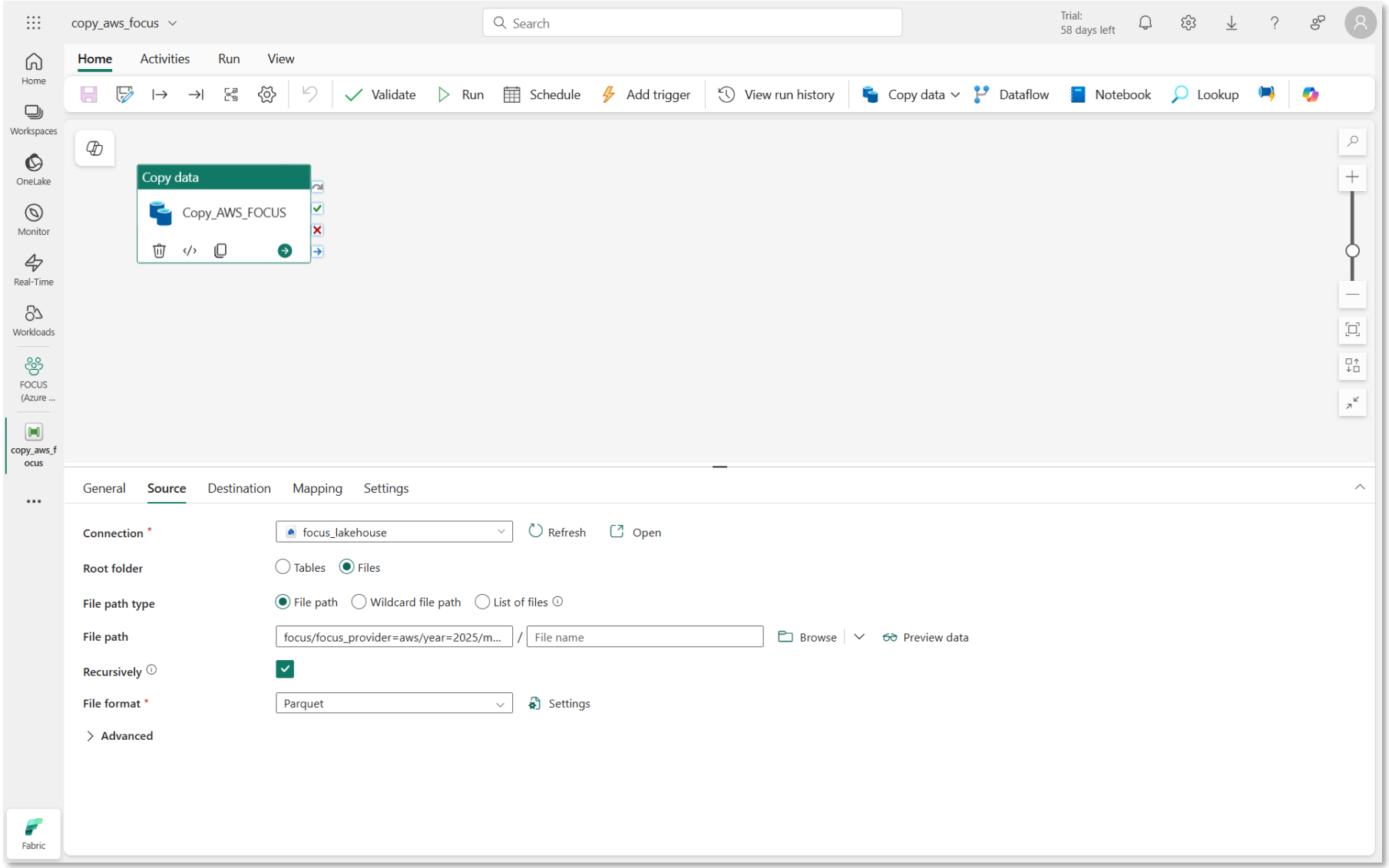
ELCA We make it work.

Getting started in minutes with MS Fabric



Microsoft Fabric Workspace for the Data Community Conference (Nov. 2025)
Same as the one presented at the Azure Bootcamp Switzerland (May 2025)

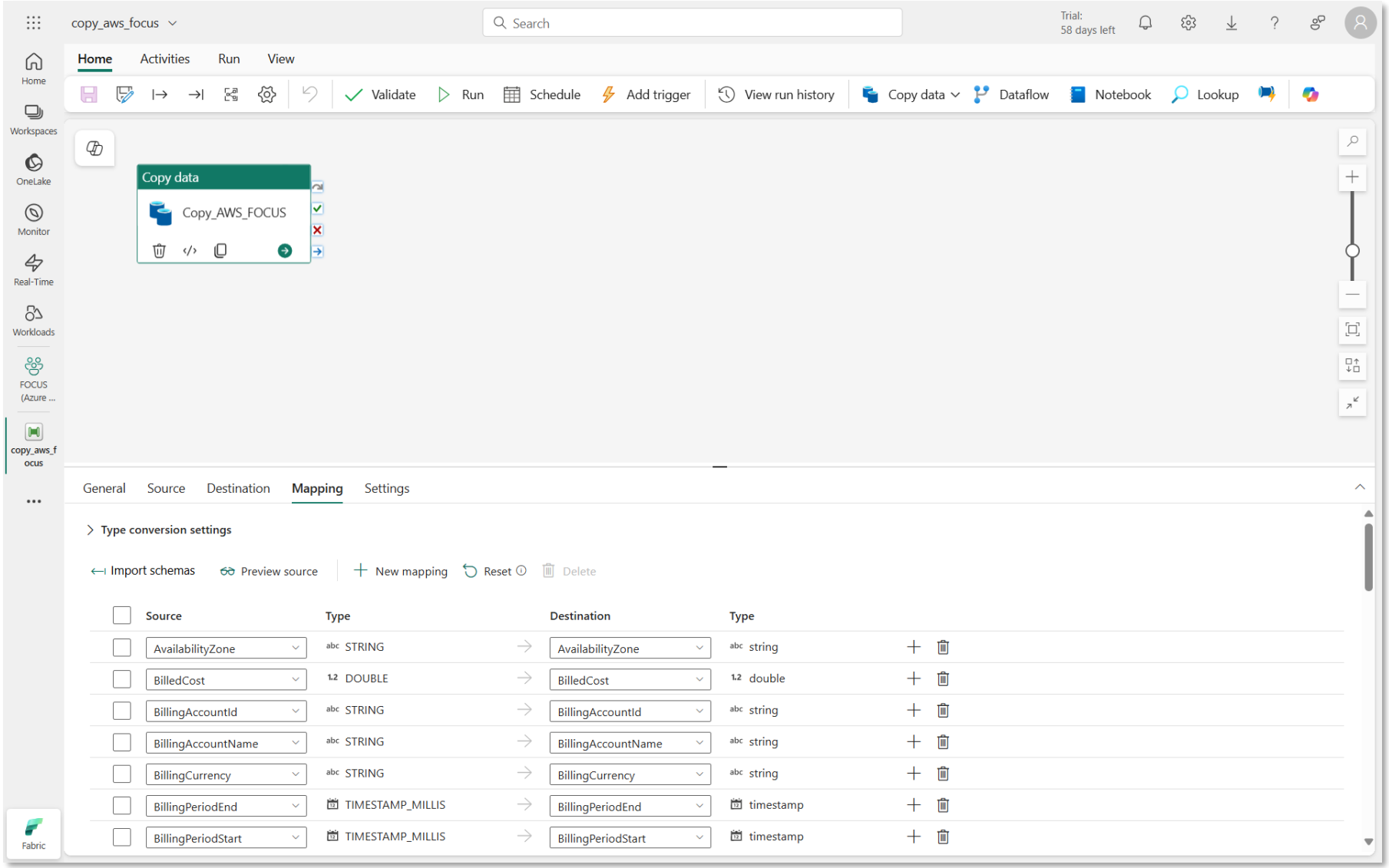
Getting started in minutes with MS Fabric



Data pipeline to import AWS FOCUS dataset

ELCA We make it work.

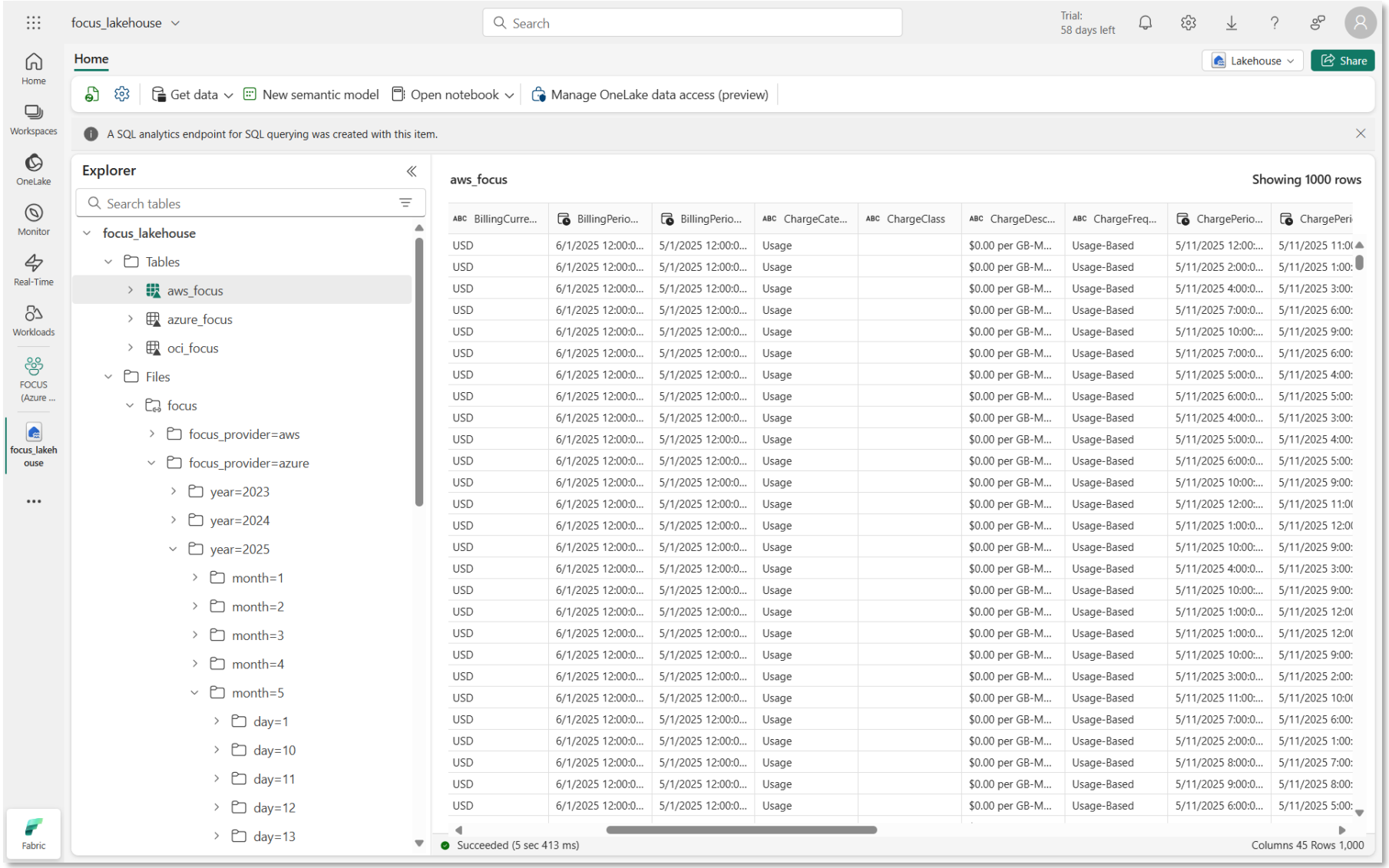
Getting started in minutes with MS Fabric



Schema mapping between FOCUS parquet and Lakehouse (auto import)

ELCA We make it work.

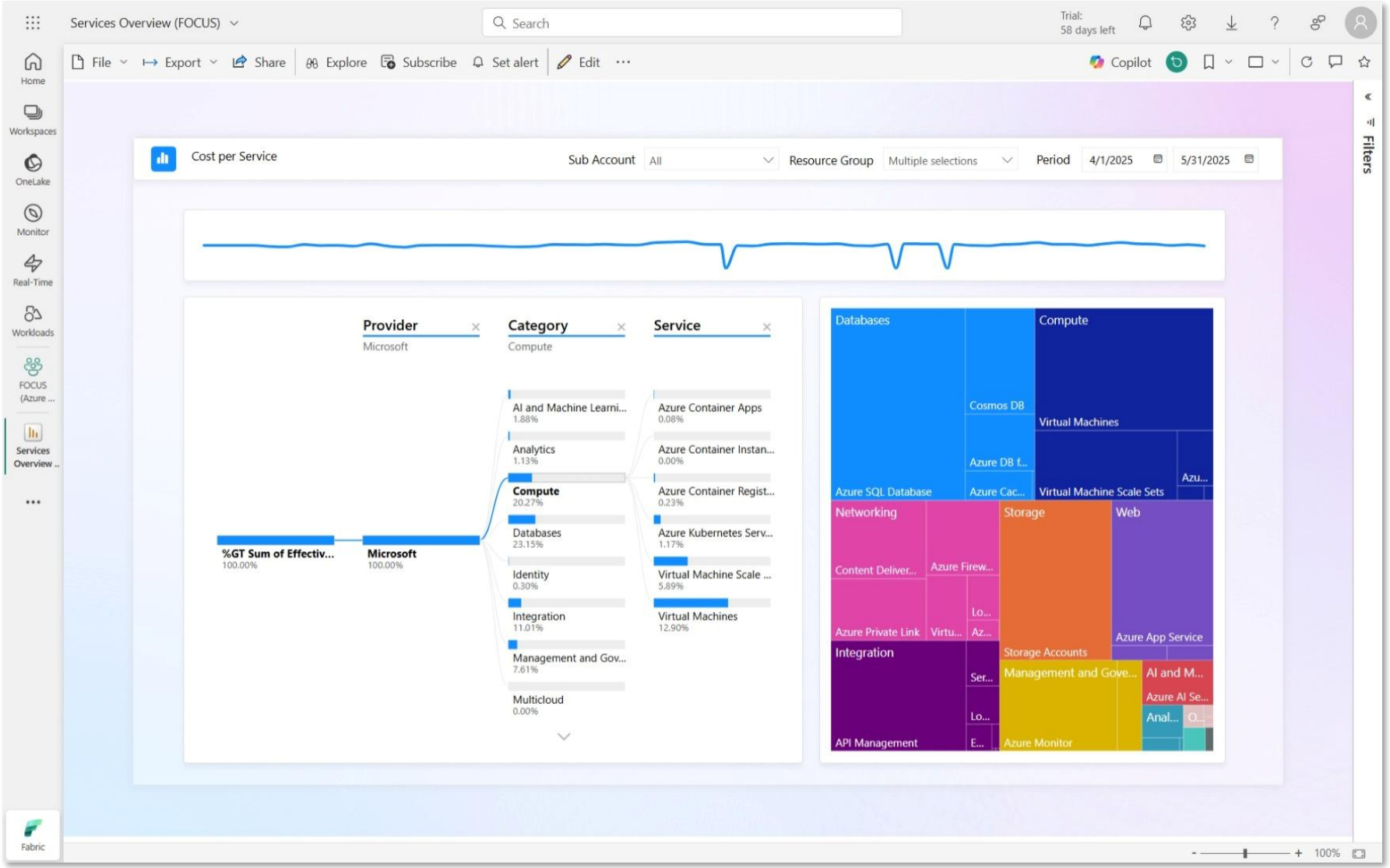
Getting started in minutes with MS Fabric



Imported FOCUS data in Lakehouse (Files as Shortcuts and Tables)

ELCA We make it work.

Getting started in minutes with MS Fabric

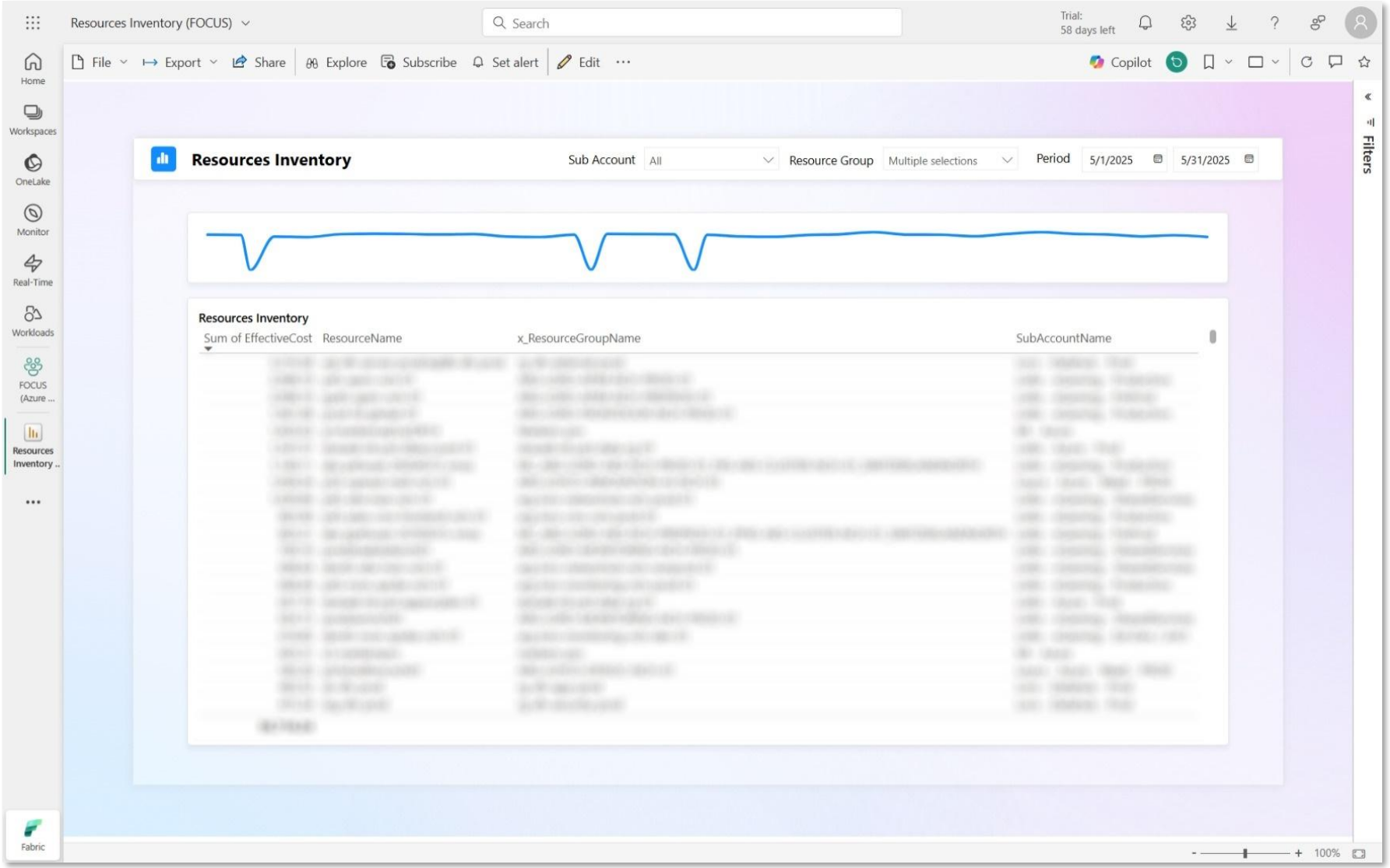


Example of a Power BI report with FOCUS dataset dataset (created partially with Copilot)

ELCA We make it work.

ELCA We make it work.

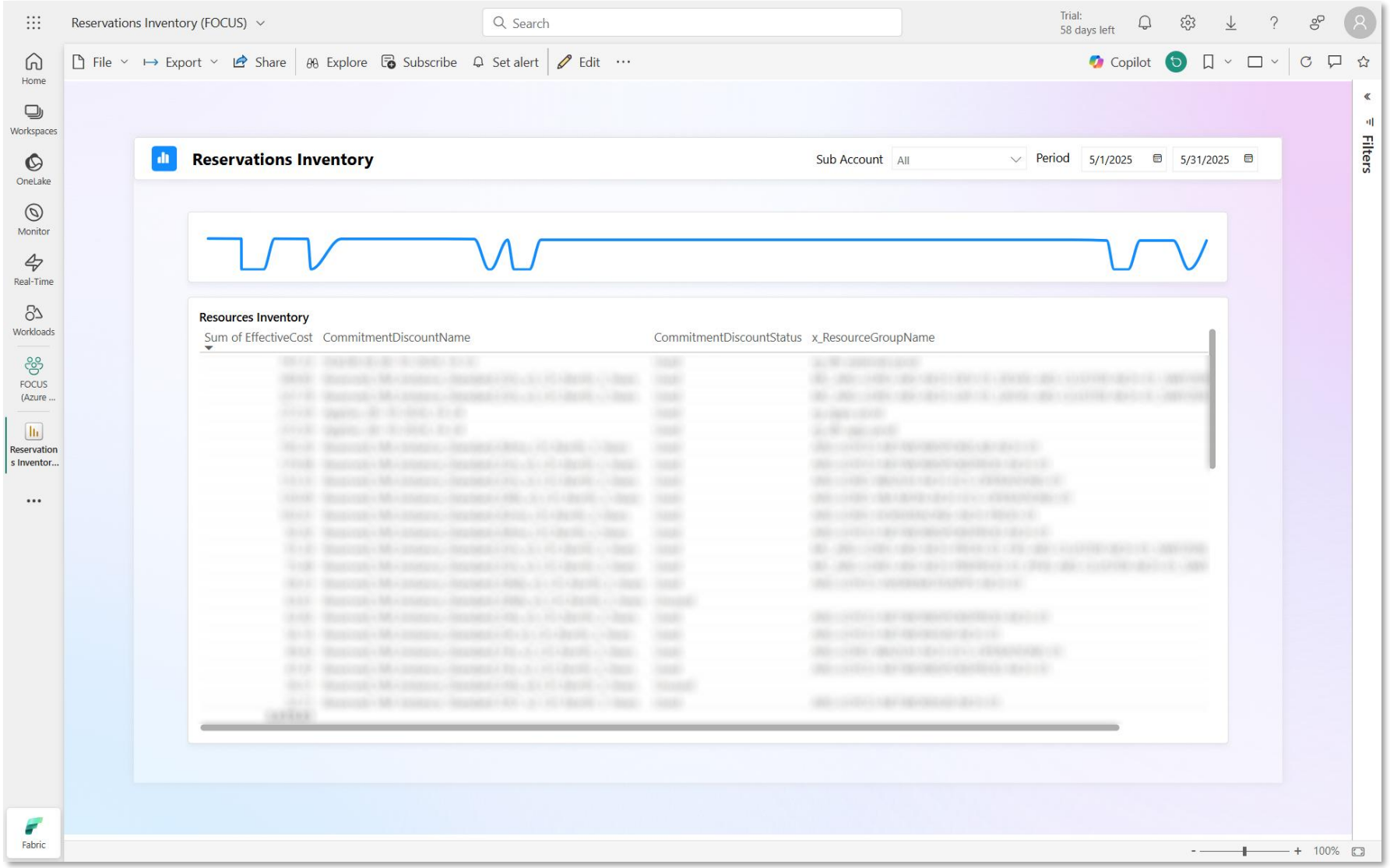
Getting started in minutes with MS Fabric



Example of a Power BI report with FOCUS dataset (created partially with Copilot)

ELCA We make it work.

Getting started in minutes with MS Fabric



Example of a Power BI report with FOCUS dataset (created partially with Copilot)

ELCA We make it work.

ELCA We make it work.

Benefits of using FOCUS™ with Microsoft Fabric

FOCUS™

Benefits

- Simple to use and understand
- Standard across multiple vendors
- Public example and use cases
- Open source and community driven

Limitations

- Adoption is in progress
- Quick release pace
- Compliance issues from Providers

Microsoft Fabric

Benefits

- Central source of truth
- Can leverage other source of data (CMDB, Finance, HR, etc.)
- Easy to use, central governance
- Can be assisted by Copilot to query the data or create reports

Limitations

- Pricing Model
- Might not be the company standard

ELCA We make it work.

Bonus: Deploy a FinOps Hub with Terraform on Microsoft Fabric!

```
workspace = {
  name  = "Data Community Conference Switzerland 2025"

  description = "FinOps Hub workspace"
}
connections = {
  oci_focus_reports_001 = {
    display_name = "OCI FOCUS Reports Connection"
    connector    = "Amazon S3 Compatible"
    inputs = {
      url = "https://bling.compact.objectstorage.eu-
zurich-1.oci.customer-oci.com"
      bucket_name = "ocid1.tenancy.oc1..aaaaaaa […]"
    }

    credentials   = {}
    privacy_level = "Private"
  }
  […]
}
pipelines = {
  copy_oci_to_adls_raw = {
    source_connection_key = "oci_focus_reports_001"
    destination_connection_key = "adls_raw"
    source_config = {
      file_path   = "FOCUS Reports"
      recursively = true
      file_format = “Binary”
    }
  […]
}
```

```
# FinOps Hub on Microsoft Fabric
module "finops_hub_with_fabric" {
  source  = "IAmFrench/finops-data-hub/fabric"
  version = "1.0.1"

  workspace = var.workspace

  fabric_capacity_id = var.fabric_capacity_id

  admins = var.admins

  connections = local.connections

  pipelines = var.pipelines

  [...]
}
```

TO BE RELEASED

Terraform Module to create a FinOps Hub on Microsoft Fabric – GitHub

ELCA We make it work.

Key takeaways

- ✓ Leverage FOCUS datasets from Cloud Services Providers
  - ✓ Automate data ingestion
- ✓ Most FOCUS datasets aren’t compliant with the specification (as of 2025-11-05)
  - ✓ Validate and fix raw datasets
  - ✓ Prepare FOCUS data pipeline for compliant FOCUS datasets
- ✓ Enrich FOCUS datasets with other data sources
- ✓ Build reports for specific FinOps Personas
  - ✓ The Central FinOps Team
  - ✓ Leadership
  - ✓ Engineering & Product (Application Teams)

ELCA We make it work.

What’s next?

ELCA We make it work.

Getting started with FOCUS & FinOps

Learn more about FinOps Certification and Training on learn.finops.org

ELCA We make it work.

ELCA *We make it work.*

Feedback Form

Thank **you**

Contact

Alexandre PARÈS

Sr. Cloud FinOps Engineer

FinOps Certified Professional
FinOps Certified FOCUS Analyst

ch.linkedin.com/in/alexandre-pares

ELCA *We make it work.*

Cite and attribute

doi.org/10.5281/zenodo.17533571

Level up your Cost Observability with the new FinOps standard FOCUS and Microsoft Fabric

By Alexandre PARÈS

On (November 2025) Data Community Conference, Switzerland

www.data-community.ch

ELCA We make it work.