

The background features a dark blue gradient with abstract, glowing shapes in shades of purple and magenta. Thin white lines intersect to form a geometric pattern. The text is positioned on the left side of the image.

AWS re:Invent

DECEMBER 2 - 6, 2024 | LAS VEGAS, NV

MAM236-S

Driving cost optimization at scale

Deeja Cruz

(she/her)
Senior Cloud FinOps Analyst
Datadog

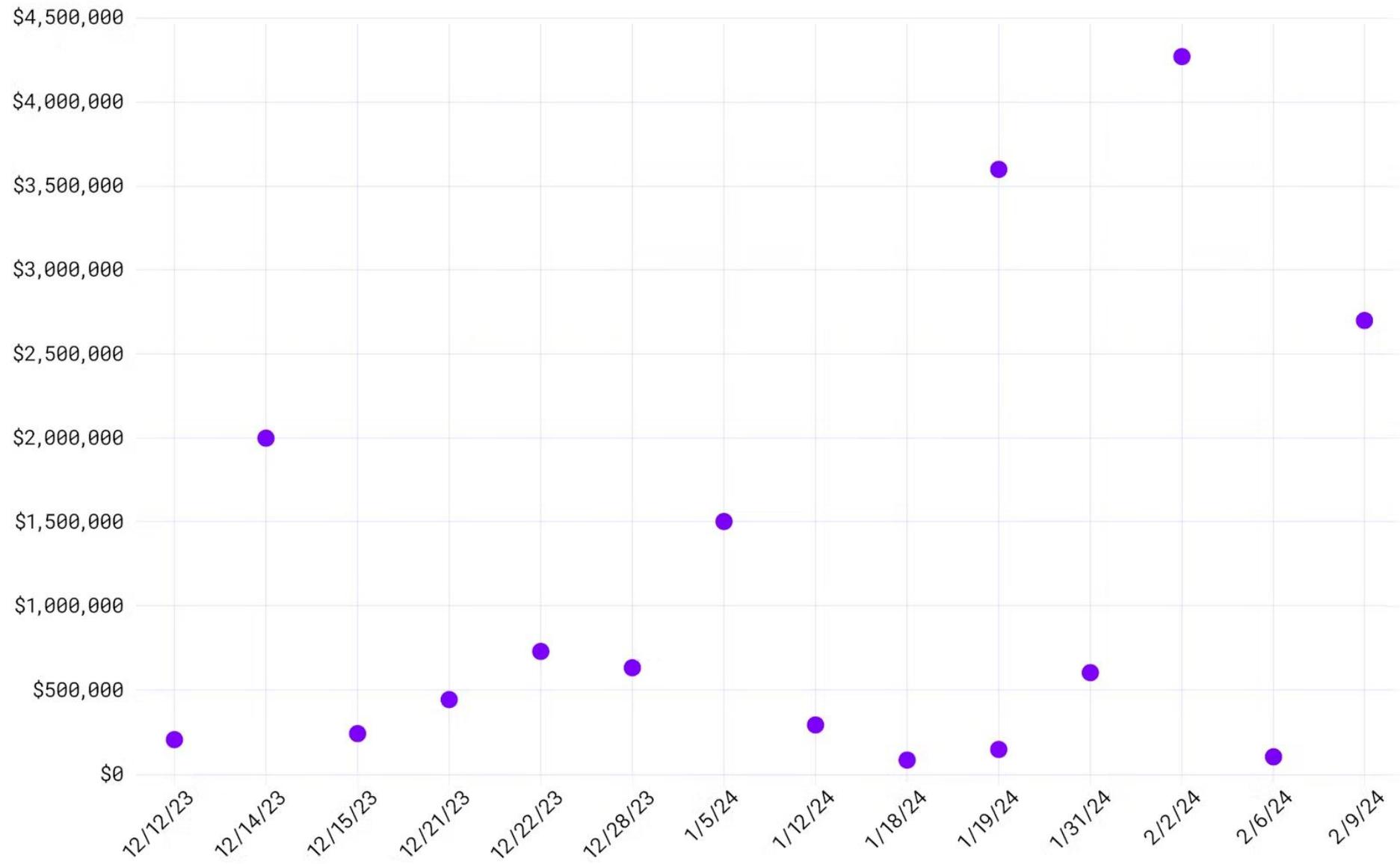
Ajuna Kyaruzi

(she/her)
Senior SRE & DevOps Advocate
Datadog



\$\$\$\$







Datadog Platform

Infrastructure

Infrastructure Monitoring
 Metrics
 Cloud Network Monitoring
 Network Device Monitoring
 Container Monitoring
 Serverless
 Cloud Cost Management
 Cloudcraft

Applications

Application Performance Monitoring
 Distributed Tracing
 Continuous Profiler
 Database Monitoring
 Universal Service Monitoring
 Data Streams Monitoring
 Data Jobs Monitoring
 LLM Observability
 Error Tracking

Digital Experience

Synthetics
 Mobile App Testing
 Browser Real User Monitoring
 Mobile Real User Monitoring
 Session Replay
 Error Tracking

Logs

Log Management
 Observability Pipelines
 Audit Trail
 Log Forwarding
 Sensitive Data Scanner
 Error Tracking

Cloud Security

Cloud Security Management
 Application Security Threat Management
 Software Composition Analysis
 Code Security
 Cloud SIEM

Software Delivery

CI Visibility
 Test Optimization
 Continuous Testing

Cloud Service Management

Incident Management
 Event Management
 Workflow Automation
 App Builder

AI

Natural Language Querying • Root Cause Analysis • Anomaly Detection • Impact Analysis • Proactive Alerts • Autonomous Investigations • Bits AI

Shared Platform Services

Dashboards • CoScreen • Teams • Agent • OpenTelemetry • Notebooks • Service Catalog • IDE Plugins • ChatOps • SLOs • Case Management



UNIFIED METRICS, LOGS, TRACES, SESSIONS

800+ INTEGRATIONS

60%

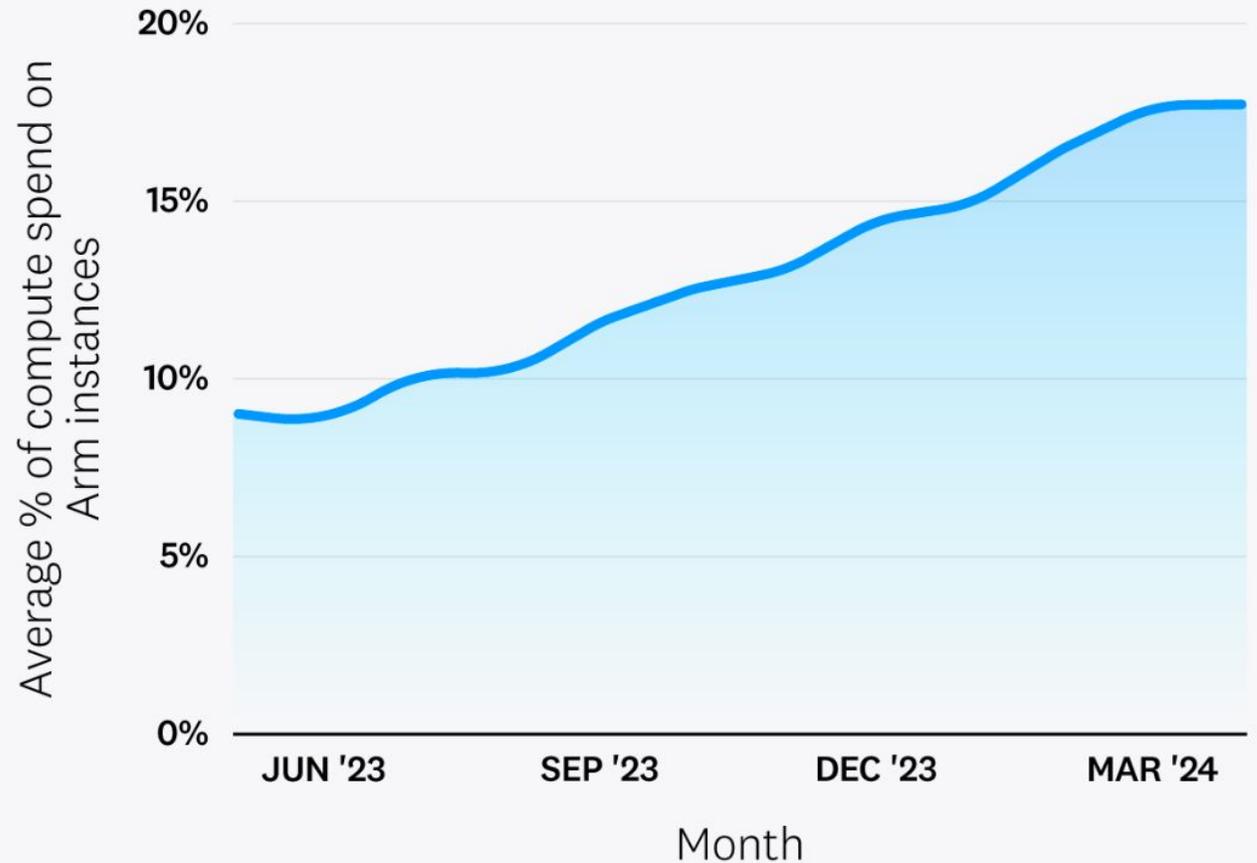
Instance types based on the Arm processor use up to 60 percent less energy than similar EC2s and often provide better performance at a lower cost.



**ORGANIZATIONS
SPEND TWICE AS
MUCH OF THEIR
COMPUTE
ALLOCATION
ON ARM
COMPARED TO
ONE YEAR AGO**

Source: Datadog

**Proportion of compute costs allocated
to Arm-based instances**



The video player shows a presentation slide with the following content:

- 2018
- NEW!
- Amazon EC2 A1 Instances
Powered by the AWS Graviton Processor
- Up to 45% lower cost for scale-out workloads
- AVAILABLE IN US EAST (N. VIRGINIA), US EAST (OHIO),
US WEST (OREGON), AND EUROPE (IRELAND) REGIONS

The slide features a diagram of an A1 processor chip connected to a lightbulb and a downward-pointing arrow with a dollar sign, symbolizing cost savings and innovation. The video player interface includes the AWS re:Invent logo, the AWS logo, and playback controls showing 0:58 / 43:59.

AWS re:Invent 2022 - What to know before adopting Arm: Lessons learned at Datadog (PRT265)

Datadog Used AWS Graviton to Deliver More Value While Keeping Costs Constant

Datadog, a cloud monitoring and analytics platform company, needed to offer more product features and serve more customers. They were looking for new innovation that they can adopt to deliver more value for our customers. Datadog adopted [AWS Graviton-based Amazon Elastic Compute Cloud \(Amazon EC2\) instances](#) to pass savings to their customers by delivering more features per compute cycles and serving more customers with an identical number of cores.



Feb 2023





Alexis Lê-Quôc

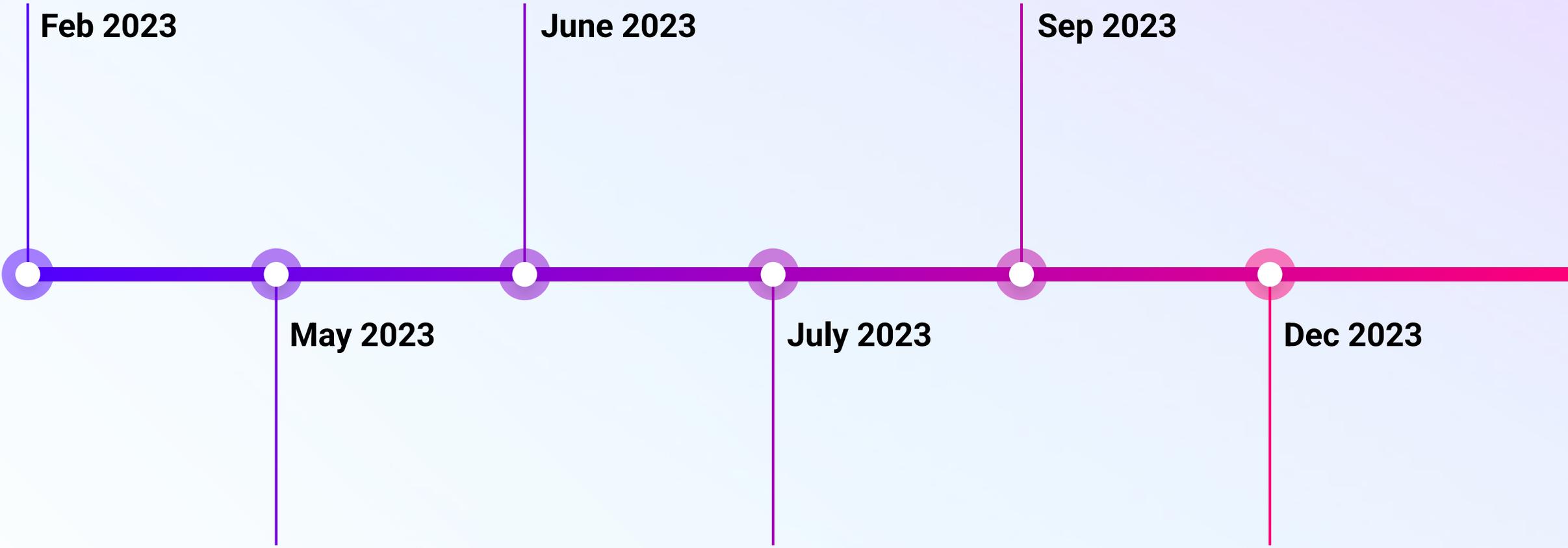
CTO & CO-FOUNDER

“

ARM - Our goal is to migrate 100% of services on AWS workloads to Arm by the **end of 2023**, with any exceptions documented and approved.

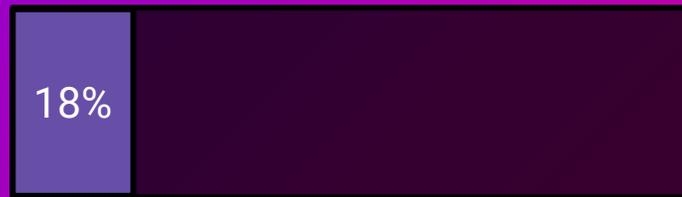
”

Timeline





May 2023





Filter tables and columns

Recent

| | |
|-------------------------|-----------|
| affected_customers | 2 fields |
| api_endpoint | 11 fields |
| api_endpoint_rum_action | 8 fields |
| api_endpoint_rum_view | 9 fields |
| arn_vip | 3 fields |

All Tables

| | |
|--------------------------------|------------|
| Compute | 15 tables |
| Containers | 36 tables |
| Data Processing | 7 tables |
| Databases | 58 tables |
| Hosts | 4 tables |
| aws_ec2_instance | 69 fields |
| azure_virtual_machine_instance | 27 fields |
| gcp_compute_instance | 53 fields |
| host | 49 fields |
| Identity | 56 tables |
| Monitoring | 36 tables |
| Networking | 96 tables |
| Queues | 11 tables |
| Real User Monitoring | 1 table |
| Reference Tables | 81 tables |
| Relationships | 218 tables |

Untitled, Oct 01 17:03:23



Untitled, Oct 01 1...

1h

Past 1 Hour



Save

Run

Modify your query with natural language



```

1 SELECT *
2 FROM aws_ec2_instance
3 LIMIT 10

```

Go to...

Recent

Dashboards

Monitors

Watchdog

Service Mgmt

Infrastructure

APM

Digital Experience

Software Delivery

Security

LLM Observability

Errors

Metrics

Logs

Integrations

Bits AI

Screen Share

ajuna.kyaruzi@...
Datadog Demo (11...

```

WITH cost_by_aws_instance_type AS (AGGR SUM(
AGGR SUM('aws.cost.net amortized')
FROM metrics
WHERE aws_product = 'ec2' AND aws_operation LIKE 'RunInstances*' AND aws_usage_type NOT LIKE
'*Bytes'
GROUP BY aws_instance_type, "runtimecosts-allocated_datacenter_env"
)
GROUP BY aws_instance_type, "runtimecosts-allocated_datacenter_env"
BUCKET BY ALL
),

savings_info AS (SELECT * FROM custom_arm_mappings_v0),

bucketed AS (
SELECT
    SUM(cost_by_aws_instance_type.timeseries) AS total_cost,
    savings_percent,
    aws_instance_type,
    "runtimecosts-allocated_datacenter_env" AS datacenter
FROM cost_by_aws_instance_type
JOIN savings_info ON cost_by_aws_instance_type.aws_instance_type = savings_info.start_type
GROUP BY savings_percent, aws_instance_type, "runtimecosts-allocated_datacenter_env"
)

SELECT
    cast(aws_instance_type AS string) AS instance_type,
    datacenter,
    ROUND (point_value(timeseries_last_point(total_cost)),2) AS total_cost,
    (ROUND(point_value(timeseries_last_point(total_cost)),2)) *
    (cast(savings_percent AS DECIMAL))
AS est_monthly_savings FROM bucketed
WHERE savings_percent > 0
ORDER BY 4 DESC

```

**How to calculate the
potential cost savings?**

Simple savings equation (cost reduction)

Example: migrate from **m5.2xlarge** to **m6g.2xlarge** (us-east-1)

Simple savings equation (cost reduction)

Example: migrate from **m5.2xlarge** to **m6g.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M5 Price} * 100$$

Simple savings equation (cost reduction)

Example: migrate from **m5.2xlarge** to **m6g.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M5 Price} * 100$$

$$(\$0.384 - \$0.308) / \$0.384 * 100$$

Simple savings equation (cost reduction)

Example: migrate from **m5.2xlarge** to **m6g.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M5 Price} * 100$$

$$(\$0.384 - \$0.308) / \$0.384 * 100$$

$$\$0.076 / \$0.384 * 100$$

Simple savings equation (cost reduction)

Example: migrate from **m5.2xlarge** to **m6g.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M5 Price} * 100$$

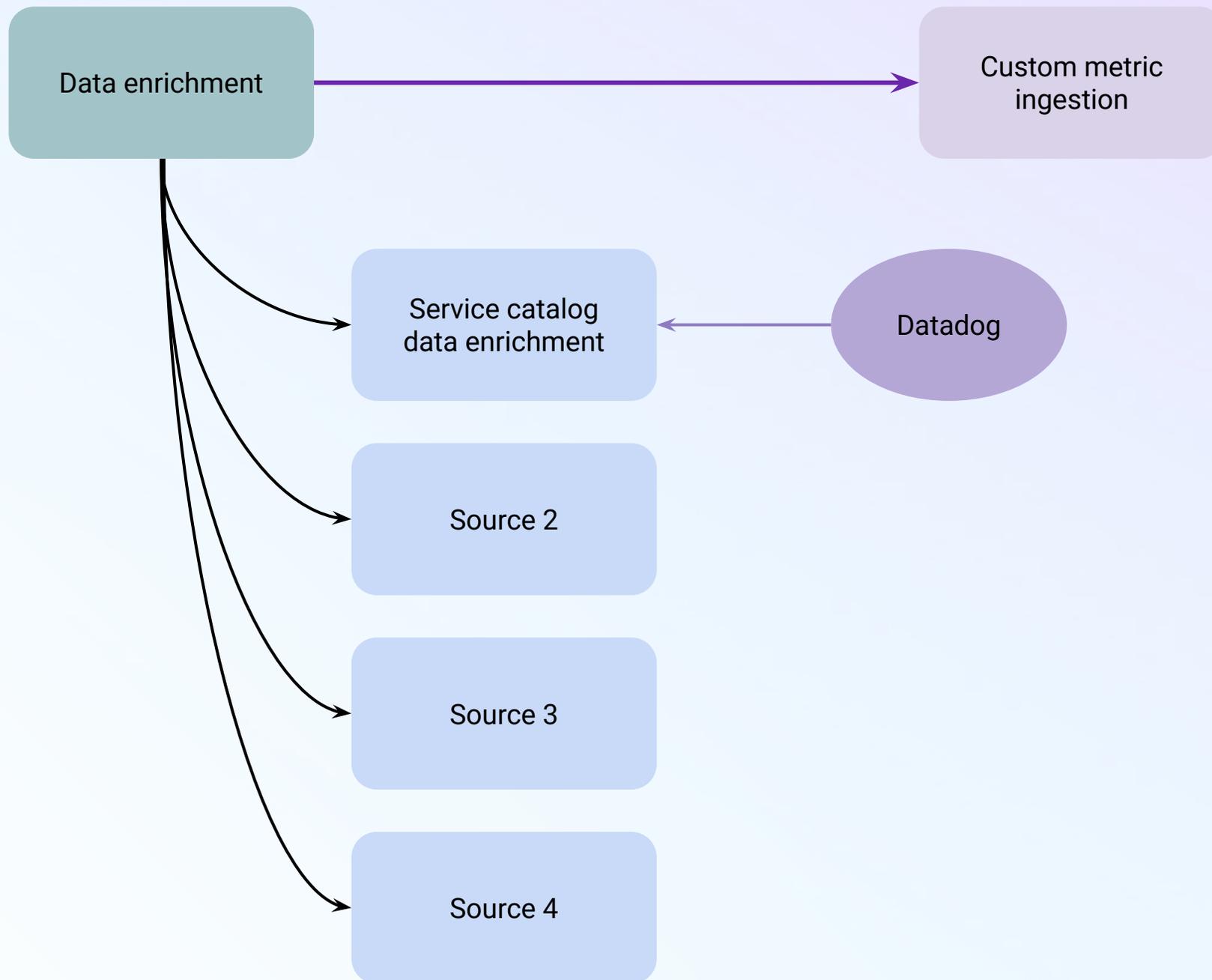
$$(\$0.384 - \$0.308) / \$0.384 * 100$$

$$\$0.076 / \$0.384 * 100$$

19.79% → **20% savings (reduction)**

Like-for-like changes

| start_type | end_type | savings_pct |
|-------------------|-----------------|--------------------|
| m5.large | m6g.large | 0.18 |
| m5.xlarge | m6g.xlarge | 0.18 |
| m5.2xlarge | m6g.2xlarge | 0.18 |
| m5.4xlarge | m6g.4xlarge | 0.18 |
| m5.8xlarge | m6g.8xlarge | 0.18 |
| m5.12xlarge | m6g.12xlarge | 0.18 |
| m5.16xlarge | m6g.16xlarge | 0.18 |



| ARM Family | \$/Core (/day) | x86 Family | \$/Core (/day) | \$ Delta | % Reduction | % Avoidance |
|------------|----------------|------------|----------------|----------|-------------|-------------|
| m6g | \$0.92 | m5 | \$1.15 | \$0.23 | 19.79% | -24.68% |
| c6g | \$0.82 | c5 | \$1.02 | \$0.20 | 20.00% | -25.00% |
| r6g | \$1.21 | r5 | \$1.51 | \$0.30 | 20.00% | -25.00% |
| m6gd | \$1.08 | m5d | \$1.36 | \$0.27 | 20.00% | -25.00% |
| c6gd | \$0.92 | c5d | \$1.15 | \$0.23 | 19.79% | -24.68% |
| r6gd | \$1.38 | r5d | \$1.73 | \$0.35 | 20.00% | -25.00% |
| c6gn | \$1.04 | c5n | \$1.30 | \$0.26 | 20.00% | -25.00% |
| m6gn | N/A | m5n | \$1.43 | N/A | N/A | N/A |
| m7g | \$0.98 | m5n | \$1.43 | \$0.45 | 31.43% | -45.83% |
| r6gn | N/A | r5n | \$1.79 | N/A | N/A | N/A |
| i4g | \$1.85 | i3 | \$1.87 | \$0.02 | 1.00% | -1.01% |
| im4gn | \$2.18 | i3en | \$2.71 | \$0.53 | 19.52% | -24.25% |

| ARM Family | \$/Core (/day) | x86 Family | \$/Core (/day) | \$ Delta | % Reduction | % Avoidance |
|------------|----------------|------------|----------------|----------|-------------|-------------|
| m6g | \$0.92 | m5 | \$1.15 | \$0.23 | 19.79% | -24.68% |
| c6g | \$0.82 | c5 | \$1.02 | \$0.20 | 20.00% | -25.00% |
| r6g | \$1.21 | r5 | \$1.51 | \$0.30 | 20.00% | -25.00% |
| m6gd | \$1.08 | m5d | \$1.36 | \$0.27 | 20.00% | -25.00% |
| c6gd | \$0.92 | c5d | \$1.15 | \$0.23 | 19.79% | -24.68% |
| r6gd | \$1.38 | r5d | \$1.73 | \$0.35 | 20.00% | -25.00% |
| c6gn | \$1.04 | c5n | \$1.30 | \$0.26 | 20.00% | -25.00% |
| m6gn | N/A | m5n | \$1.43 | N/A | N/A | N/A |
| m7g | \$0.98 | m5n | \$1.43 | \$0.45 | 31.43% | -45.83% |
| r6gn | N/A | r5n | \$1.79 | N/A | N/A | N/A |
| i4g | \$1.85 | i3 | \$1.87 | \$0.02 | 1.00% | -1.01% |
| im4gn | \$2.18 | i3en | \$2.71 | \$0.53 | 19.52% | -24.25% |

| ARM Family | \$/Core (/day) | x86 Family | \$/Core (/day) | \$ Delta | % Reduction | % Avoidance |
|------------|----------------|------------|----------------|----------|-------------|-------------|
| m6g | \$0.92 | m5 | \$1.15 | \$0.23 | 19.79% | -24.68% |
| c6g | \$0.82 | c5 | \$1.02 | \$0.20 | 20.00% | -25.00% |
| r6g | \$1.21 | r5 | \$1.51 | \$0.30 | 20.00% | -25.00% |
| m6gd | \$1.08 | m5d | \$1.36 | \$0.27 | 20.00% | -25.00% |
| c6gd | \$0.92 | c5d | \$1.15 | \$0.23 | 19.79% | -24.68% |
| r6gd | \$1.38 | r5d | \$1.73 | \$0.35 | 20.00% | -25.00% |
| c6gn | \$1.04 | c5n | \$1.30 | \$0.26 | 20.00% | -25.00% |
| m6gn | N/A | m5n | \$1.43 | N/A | N/A | N/A |
| m7g | \$0.98 | m5n | \$1.43 | \$0.45 | 31.43% | -45.83% |
| r6gn | N/A | r5n | \$1.79 | N/A | N/A | N/A |
| i4g | \$1.85 | i3 | \$1.87 | \$0.02 | 1.00% | -1.01% |
| im4gn | \$2.18 | i3en | \$2.71 | \$0.53 | 19.52% | -24.25% |

Simple savings equation (cost avoidance)

Example: spin up an **m6g.2xlarge** instead of **m5.2xlarge** (us-east-1)

Simple savings equation (cost avoidance)

Example: spin up an **m6g.2xlarge** instead of **m5.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M6g Price} * 100$$

Simple savings equation (cost avoidance)

Example: spin up an **m6g.2xlarge** instead of **m5.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M6g Price} * 100$$

$$(\$0.384 - \$0.308) / \$0.308 * 100$$

Simple savings equation (cost avoidance)

Example: spin up an **m6g.2xlarge** instead of **m5.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M6g Price} * 100$$

$$(\$0.384 - \$0.308) / \$0.308 * 100$$

$$\$0.076 / \$0.308 * 100$$

Simple savings equation (cost avoidance)

Example: spin up an **m6g.2xlarge** instead of **m5.2xlarge** (us-east-1)

$$(\text{M5 Price} - \text{M6g Price}) / \text{M6g Price} * 100$$

$$(\$0.384 - \$0.308) / \$0.308 * 100$$

$$\$0.076 / \$0.308 * 100$$

24.68% → **25% savings (avoidance)**

Lesson 1: Iterate Early & Often

Lesson 1: Iterate Early & Often

June 2023



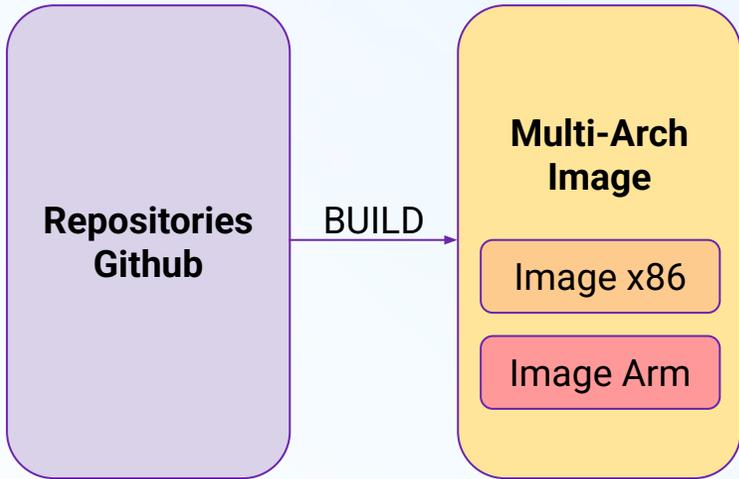


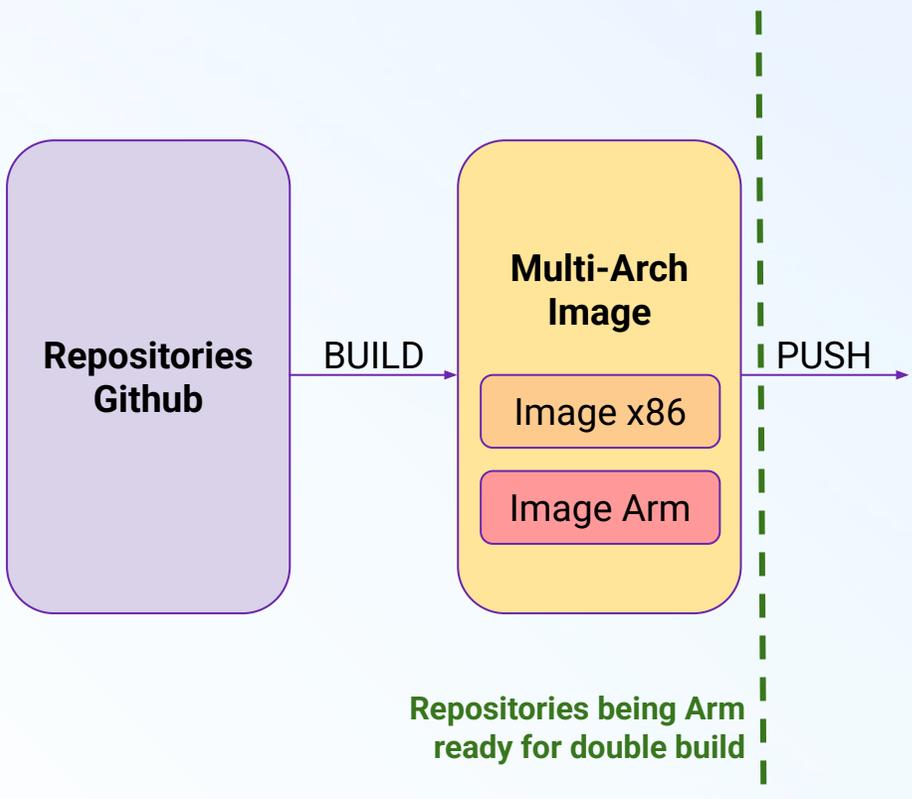
Meet our Org - Astro

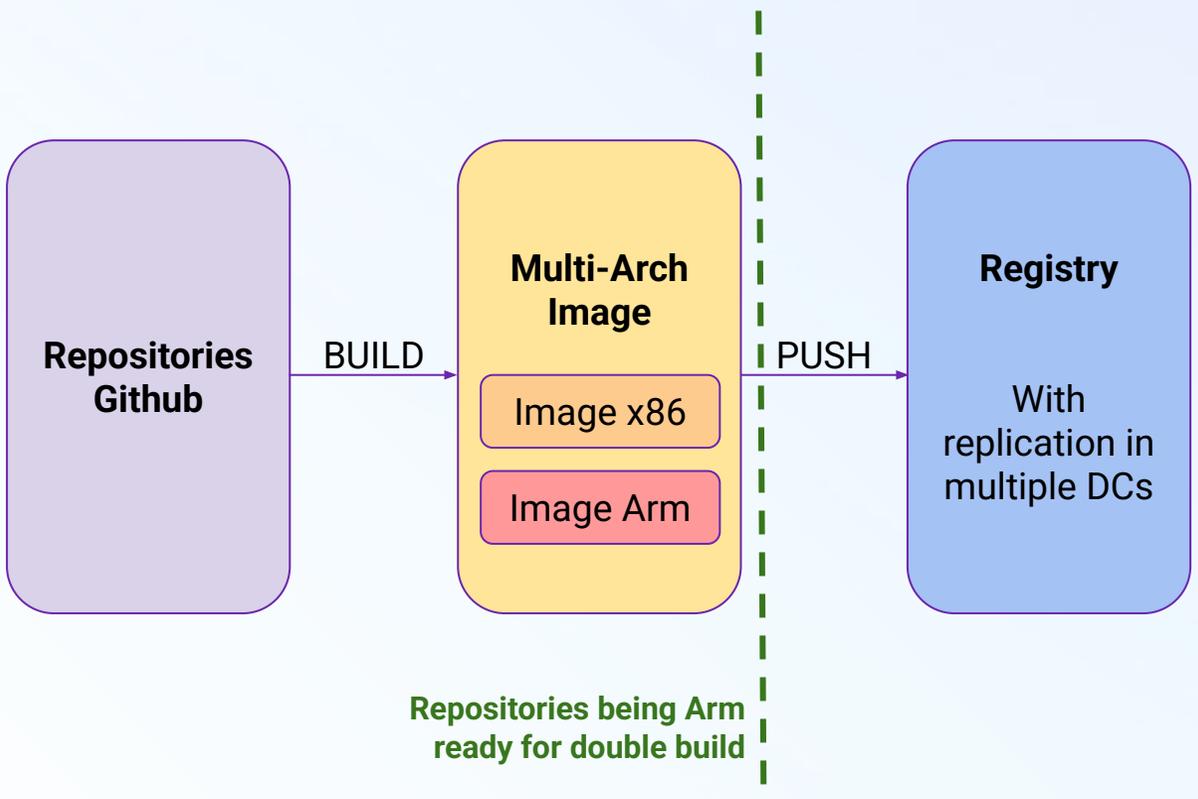
**How to track the
progress across all
services?**

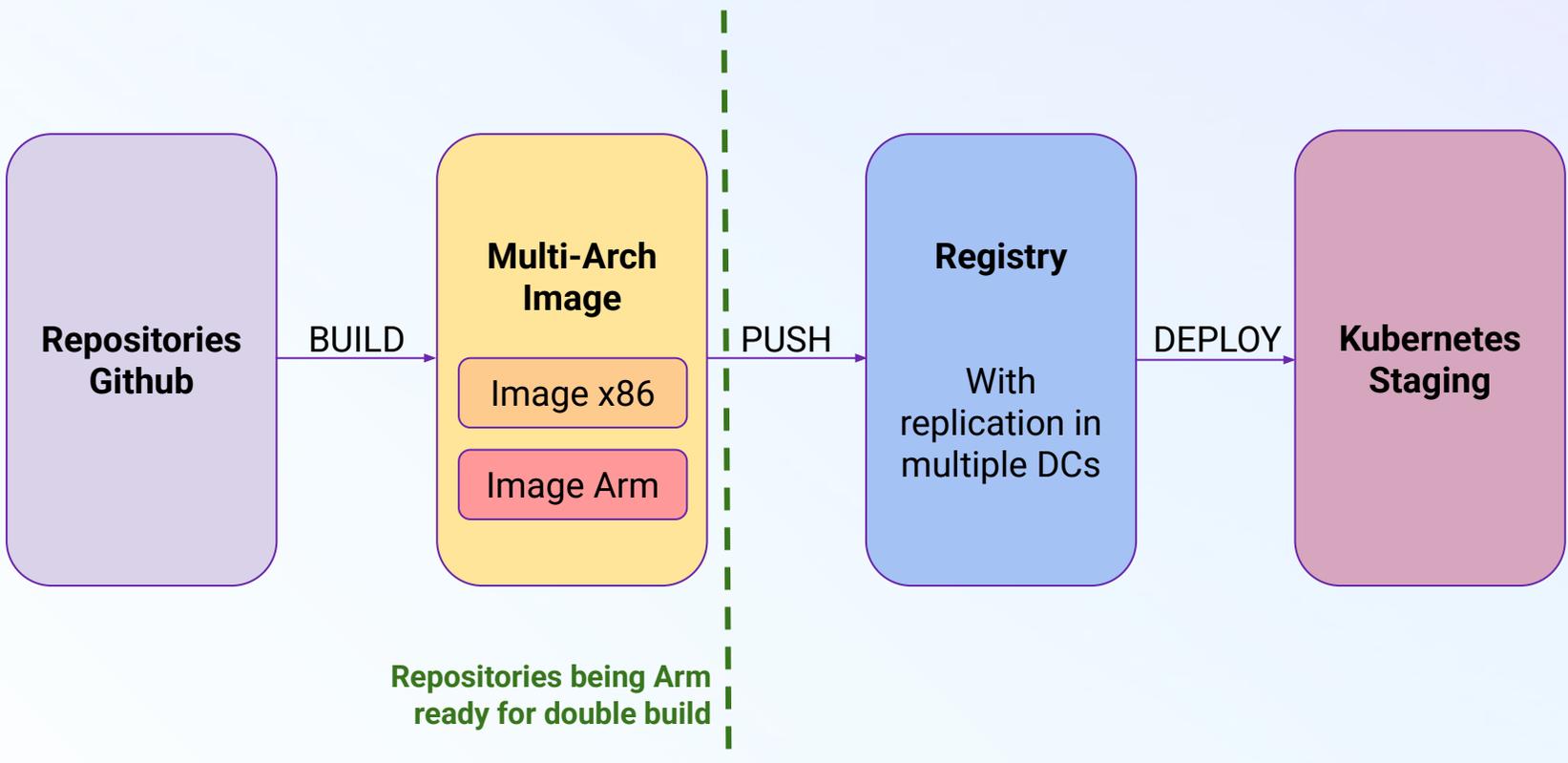


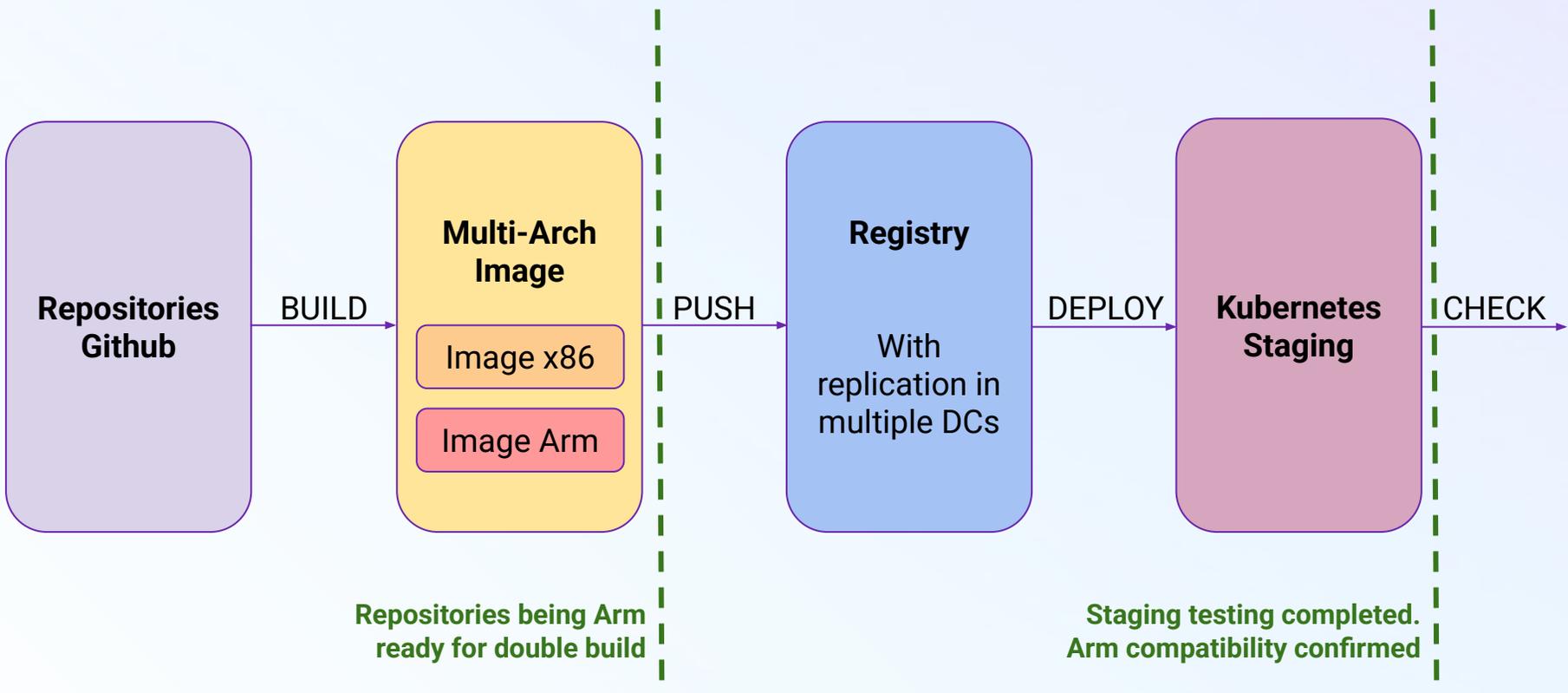
**Repositories
Github**

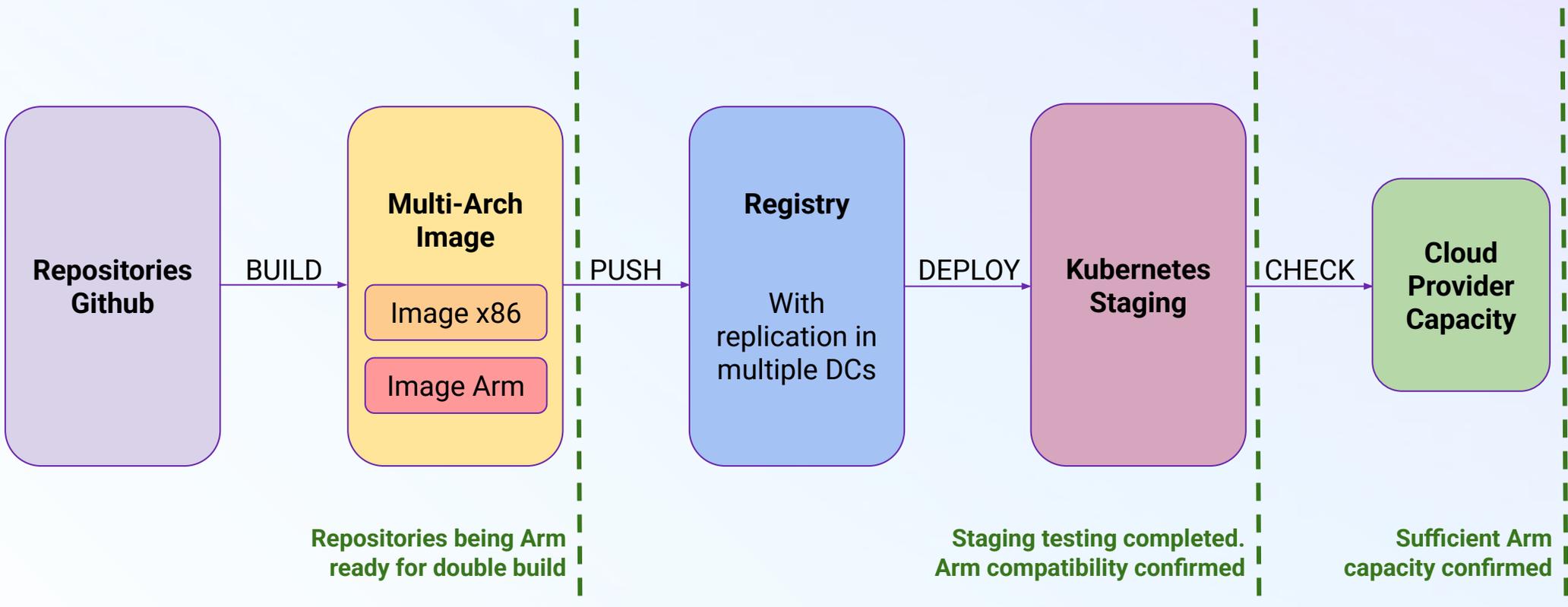


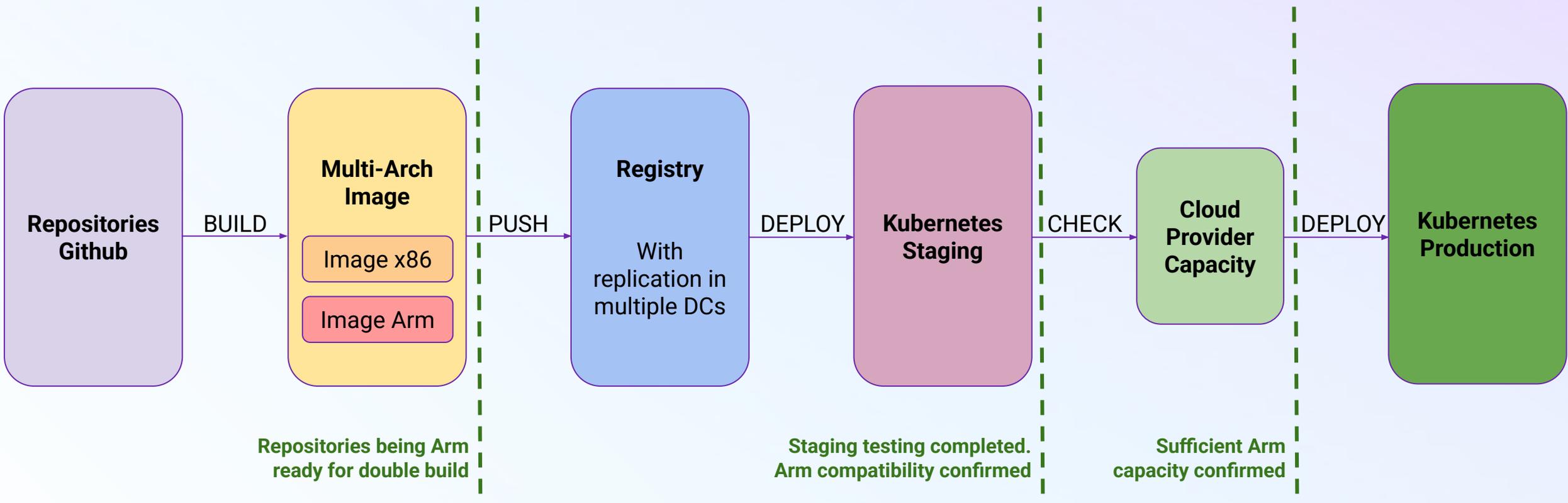












Create issue

Project

ARM Migration Program

Issue Type

 **Story**

Summary*

Components

None

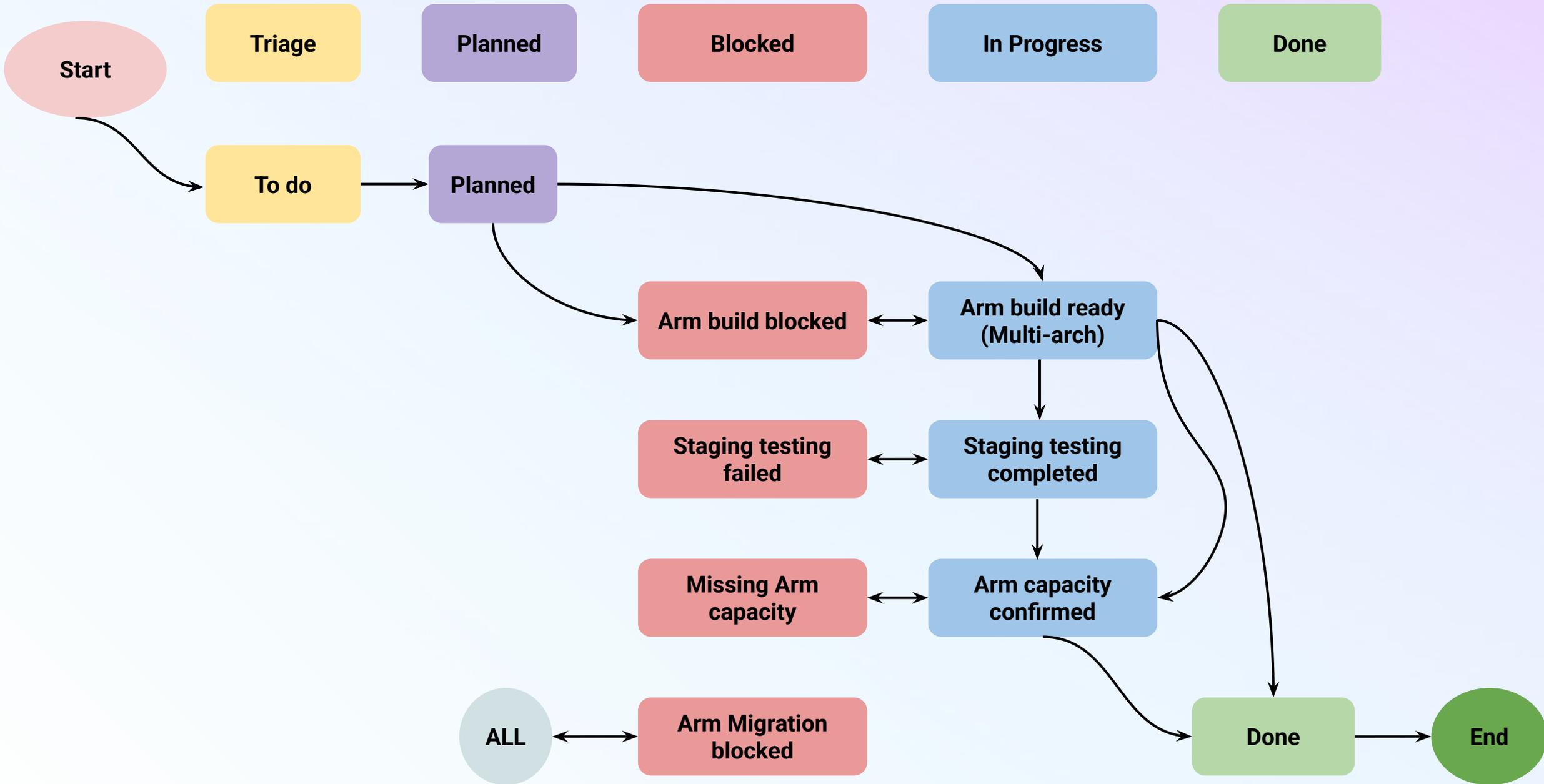
Attachment

 Drop files to attach, or browse.

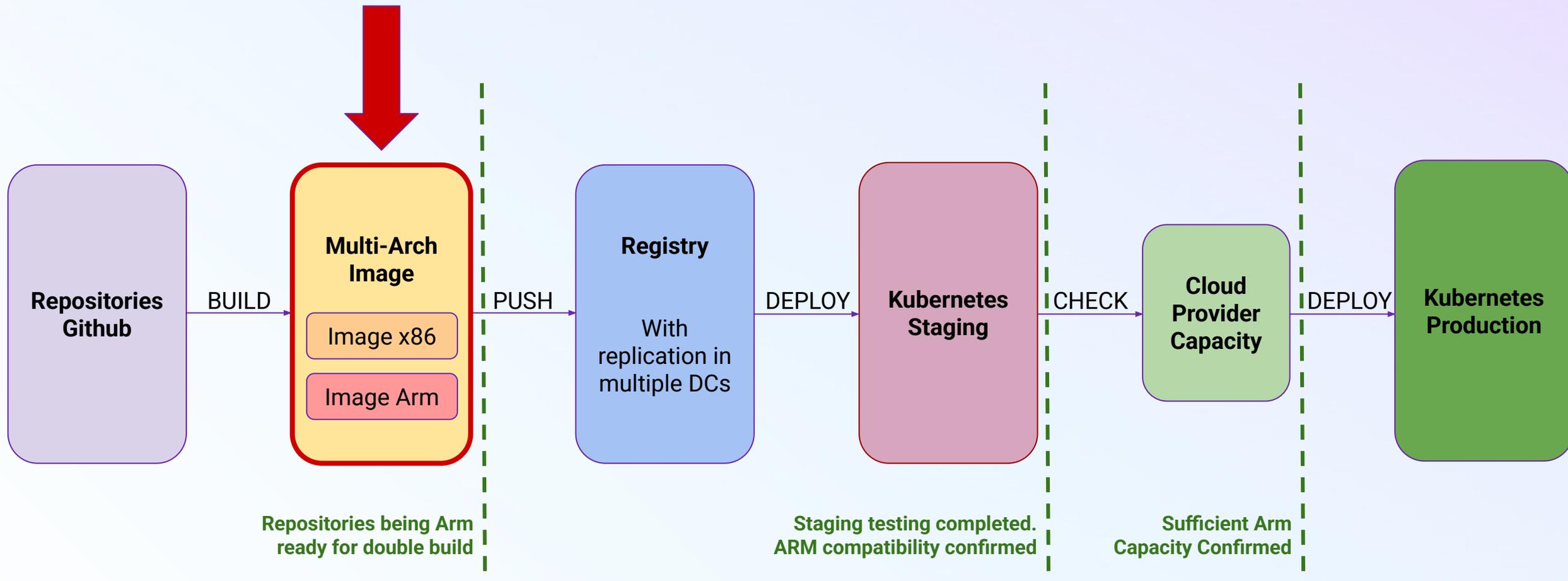
Description

Mandatory information

- Team Name: **<Complete here>**
- Service Name(s): **<Complete here - can be one or several values>**
 - List of services can be retrieved from the Service catalog <https://app.datadoghq.com/services>
- Slack Channel Name: **<Complete here>**
 - Capacity Management team can help in identify a suitable instance type
- Region and Availability Zone distribution: **<Complete here>**
- K8s cluster name(s): **<Complete here - can be one or several values>**
- Source repository: **<Complete here>**
- Migration ETA: **<Complete here>**
 - Consider a multi-step migration
 - ex: 2023-Q3 - *be realistic*



Step 1: Multi-Architecture Images



Getting started

In your `values.yaml`:

```
1 config:  
2   supportedArchs:  
3     - amd64  
4     - arm64
```

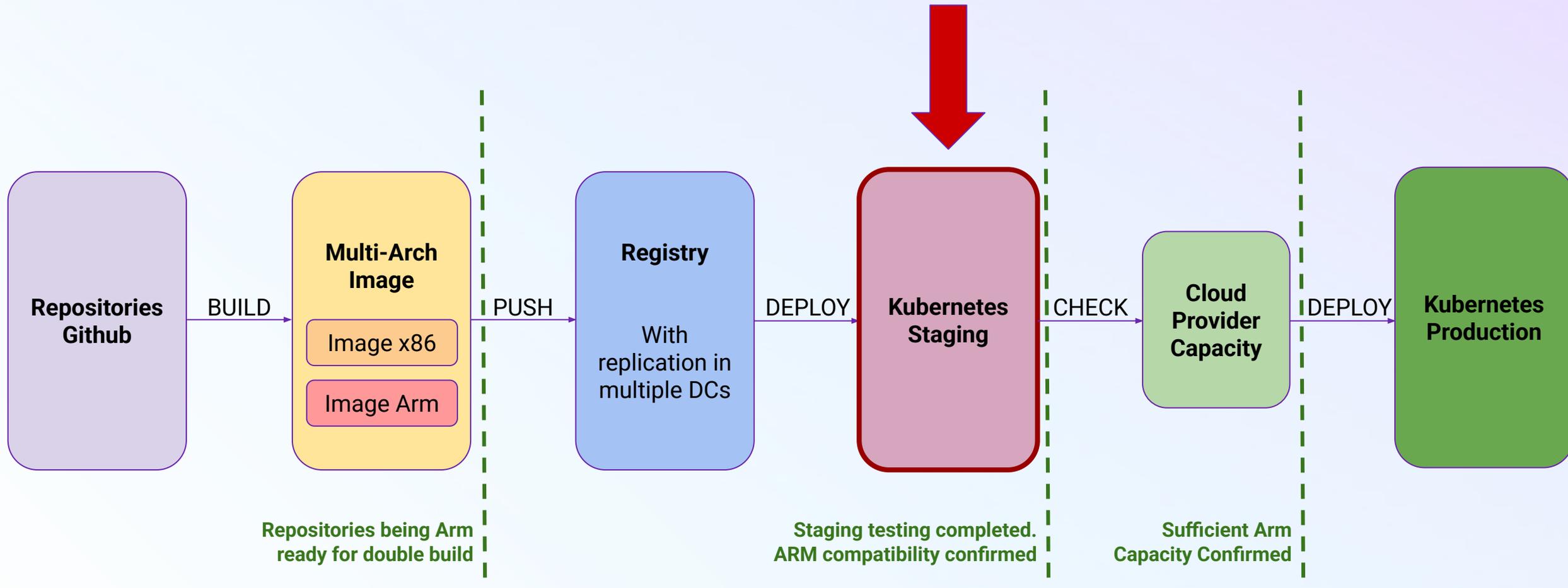
 `True` is the default value for `multi_arch` in the `logs_backend_service_image` function, skip this step in this case.

Step 1 / Building multi-arch images

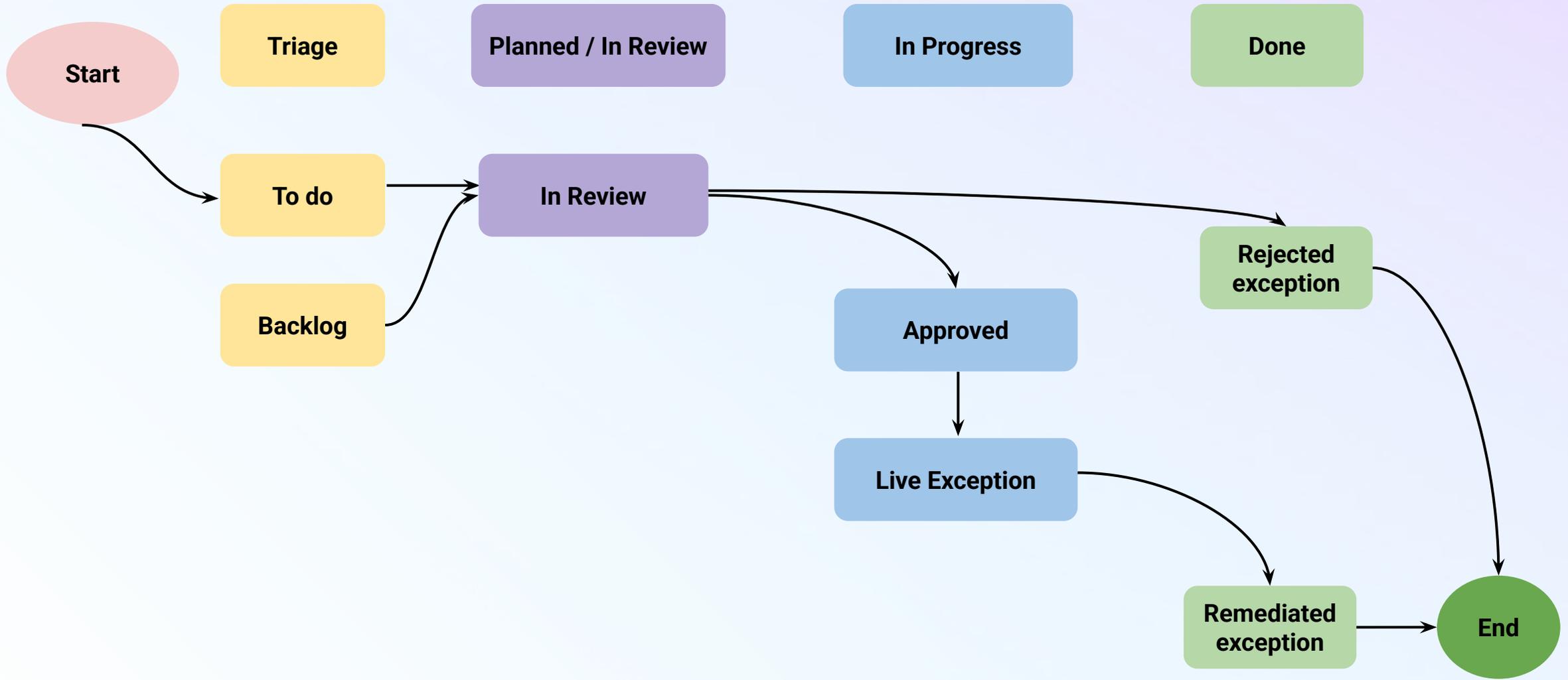
Your migration journey starts with building multi-arch images. Use the following table to identify the specific guide you should be using.

| Repositories | Status | Latest info / Next steps |
|--------------|---------------------------------|--|
| dd-go | ARM READY | <ul style="list-style-type: none">• Repo is ARM fully ARM ready.<ul style="list-style-type: none">◦ Exception: Only app relying on foundationDB are blocked• Getting started: Getting Started Build Multiarch images in dd go |
| logs backend | ARM READY ARM BY DEFAULT | <ul style="list-style-type: none">• Repo is ARM fully ARM ready.• All services using logs-backend-common chart and logs-shared or logs-backend pod p rofiles are enabled by default• Getting started: Logs-backend On ARM |
| Team managed | TEAM SPECIFIC | <ul style="list-style-type: none">• Changes to the toolchain to build multi-arch to be handle by the owning team. Reach out to (Language Tools team) for help. |

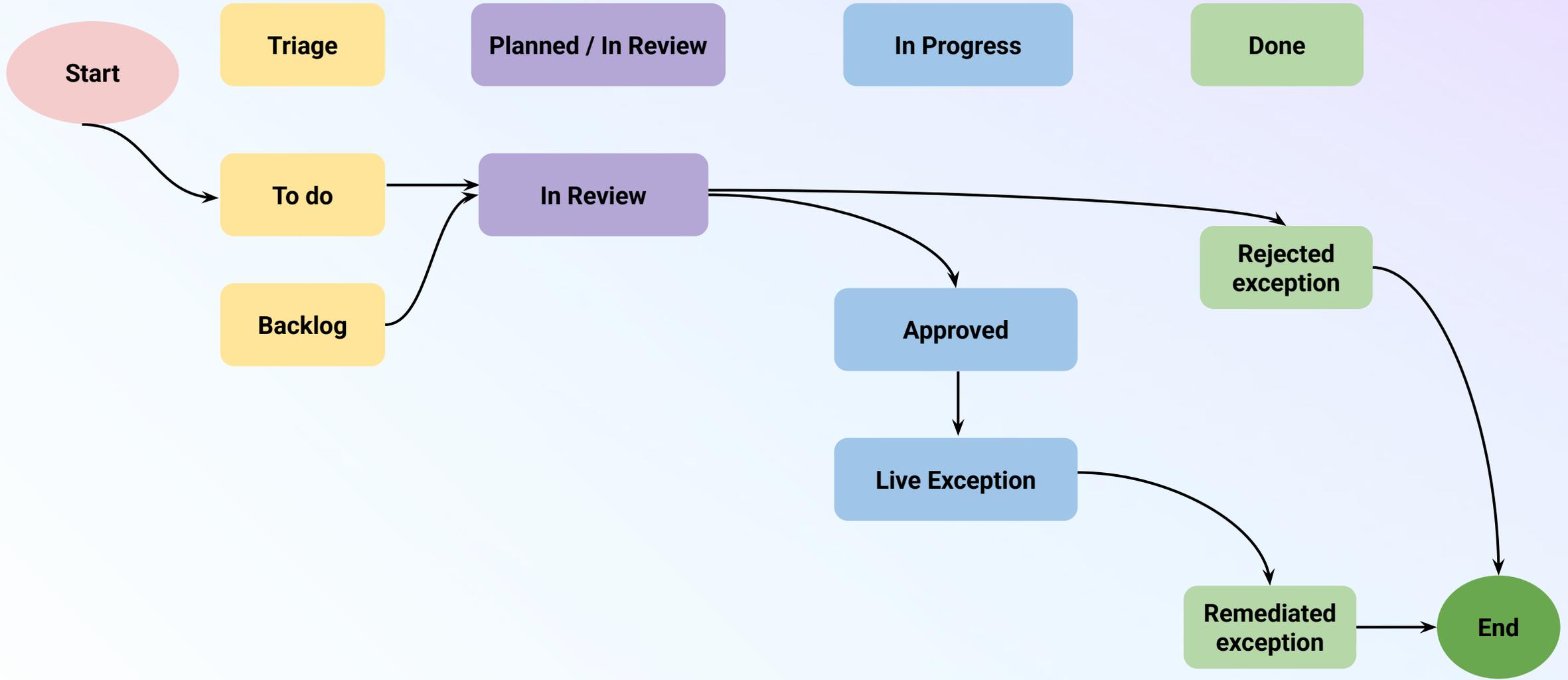
Next Step: Staging



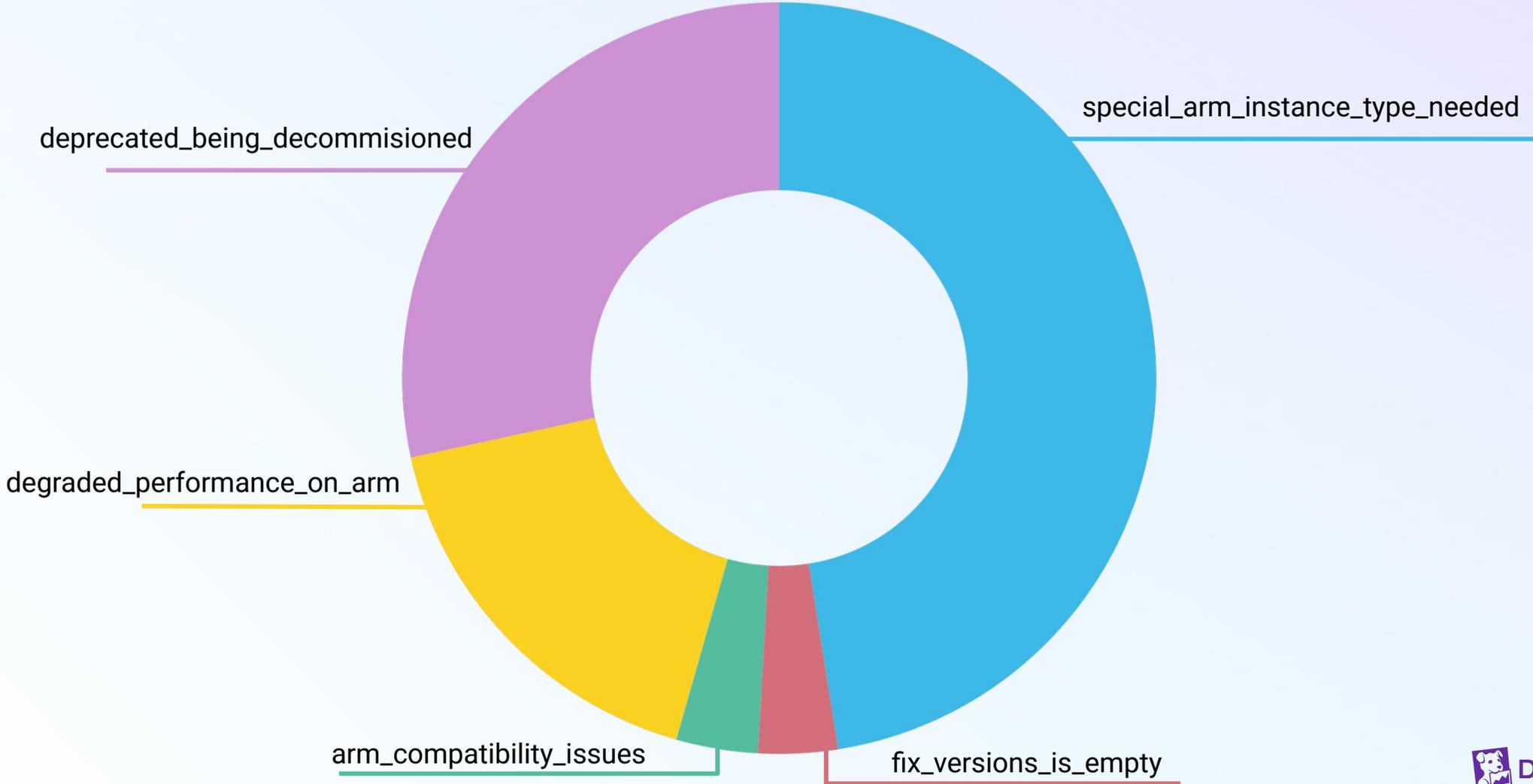
Exception Status Flow



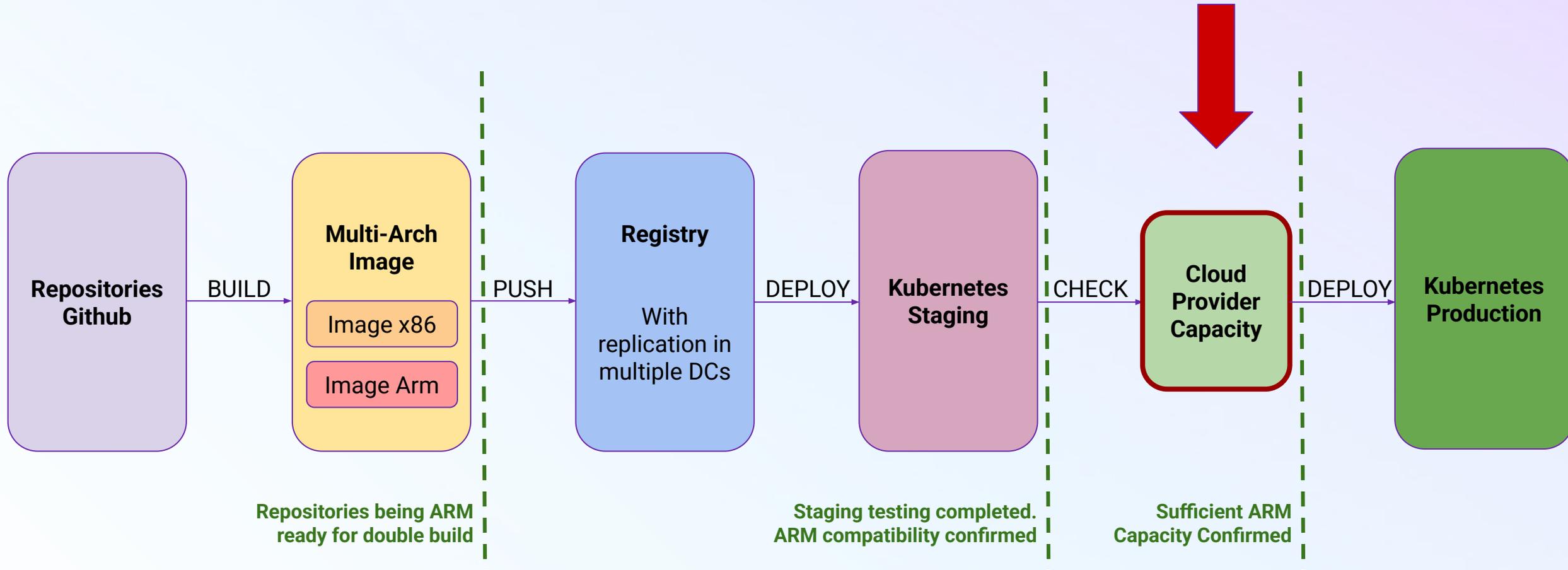
Exception Status Flow



Deferred Monthly Savings Estimates per Exception Category



Step 3: Capacity Check

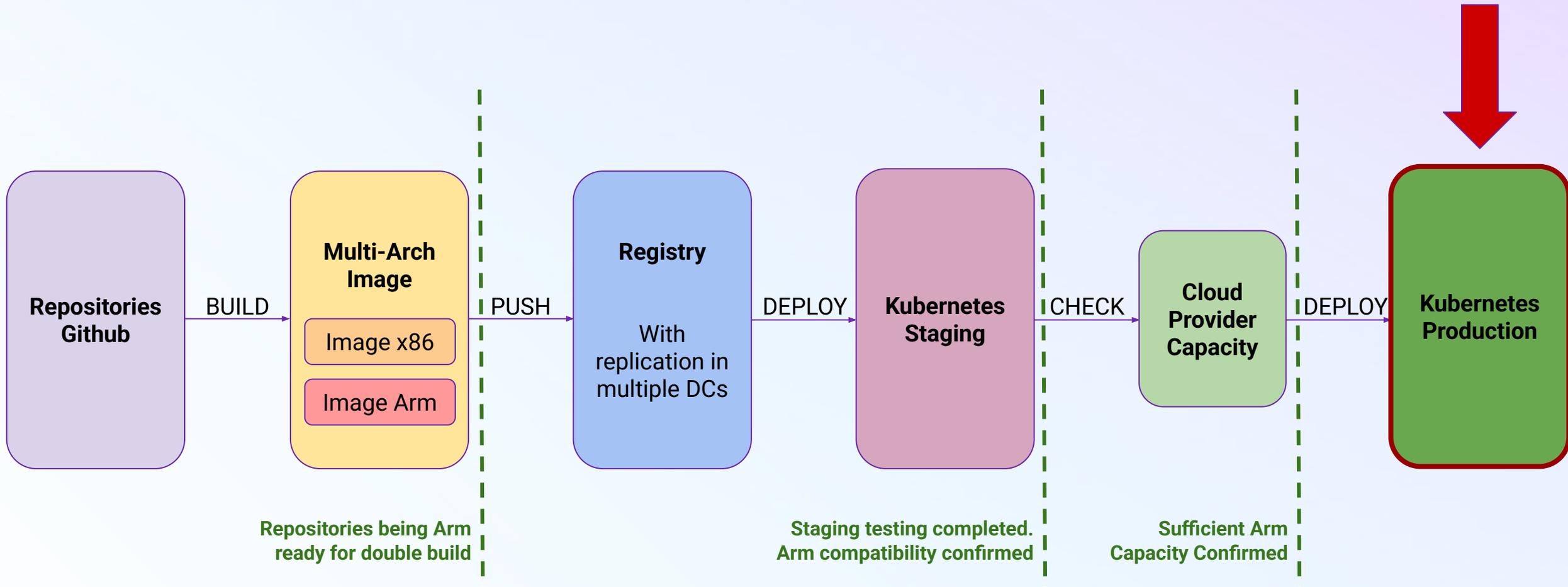


Reserve compute capacity with On-Demand Capacity Reservations

[PDF](#) | [RSS](#)

On-Demand Capacity Reservations allow you to reserve compute capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. If you have strict capacity requirements, and are running business-critical workloads that require a certain level of long or short-term capacity assurance, we recommend that you create a Capacity Reservation to ensure that you always have access to Amazon EC2 capacity when you need it, for as long as you need it.

Step 4: Production!

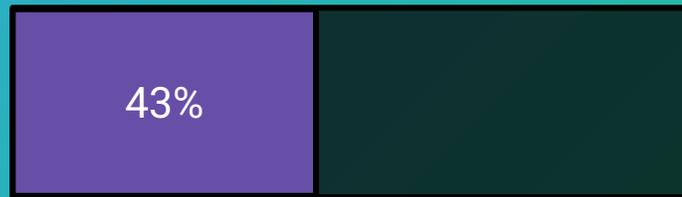


Step 4: Production!



Lesson 2: Reduce friction

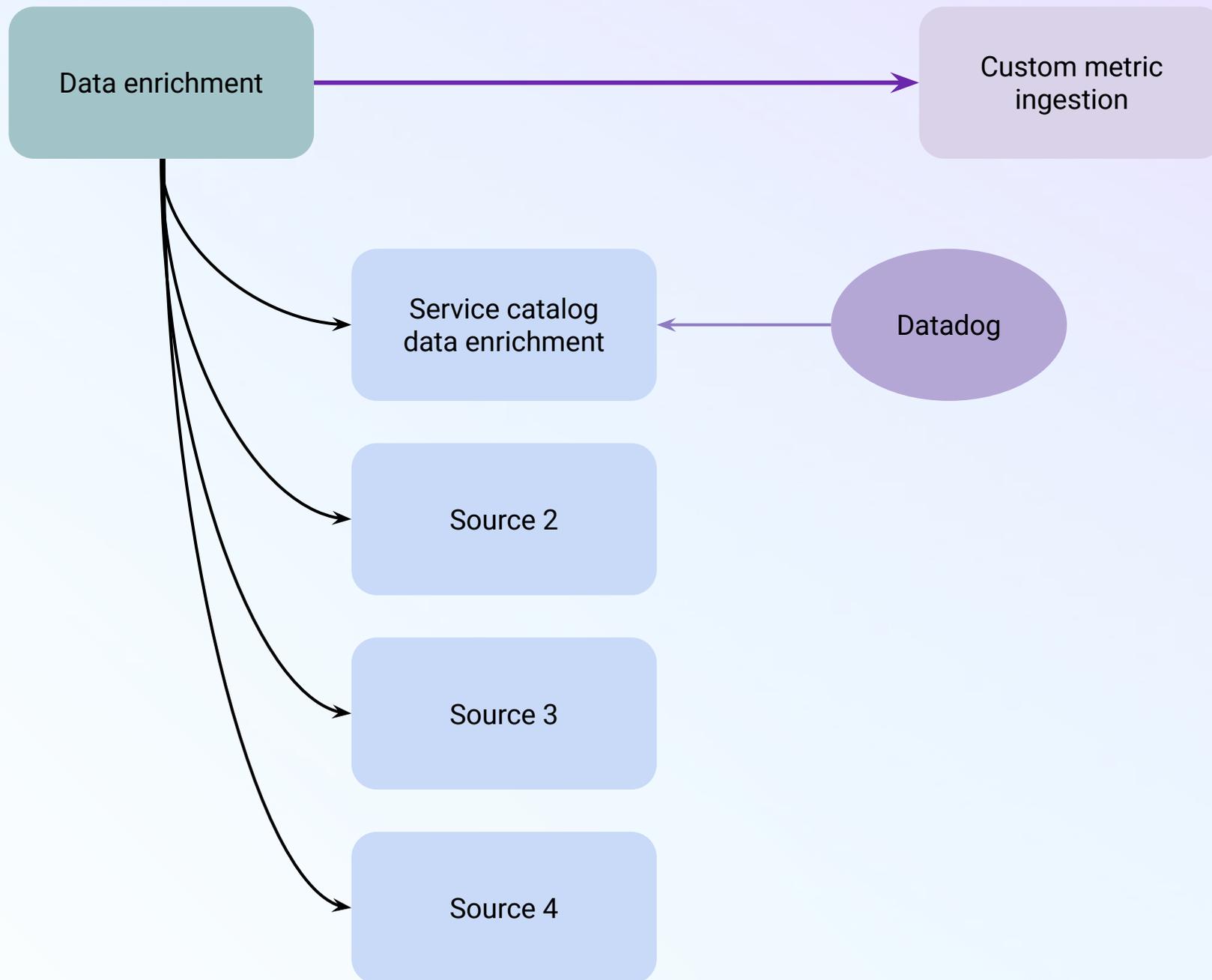
July 2023

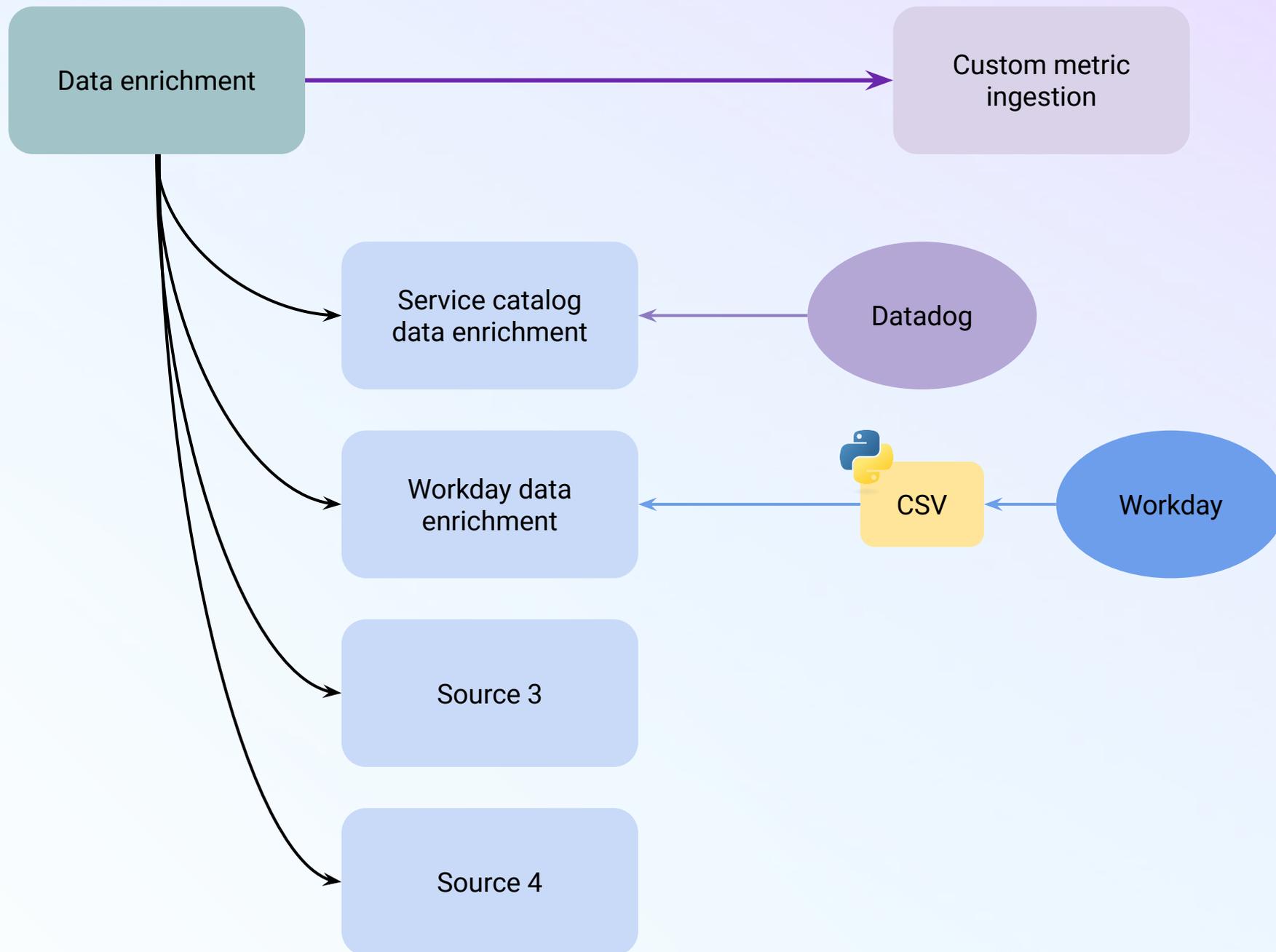


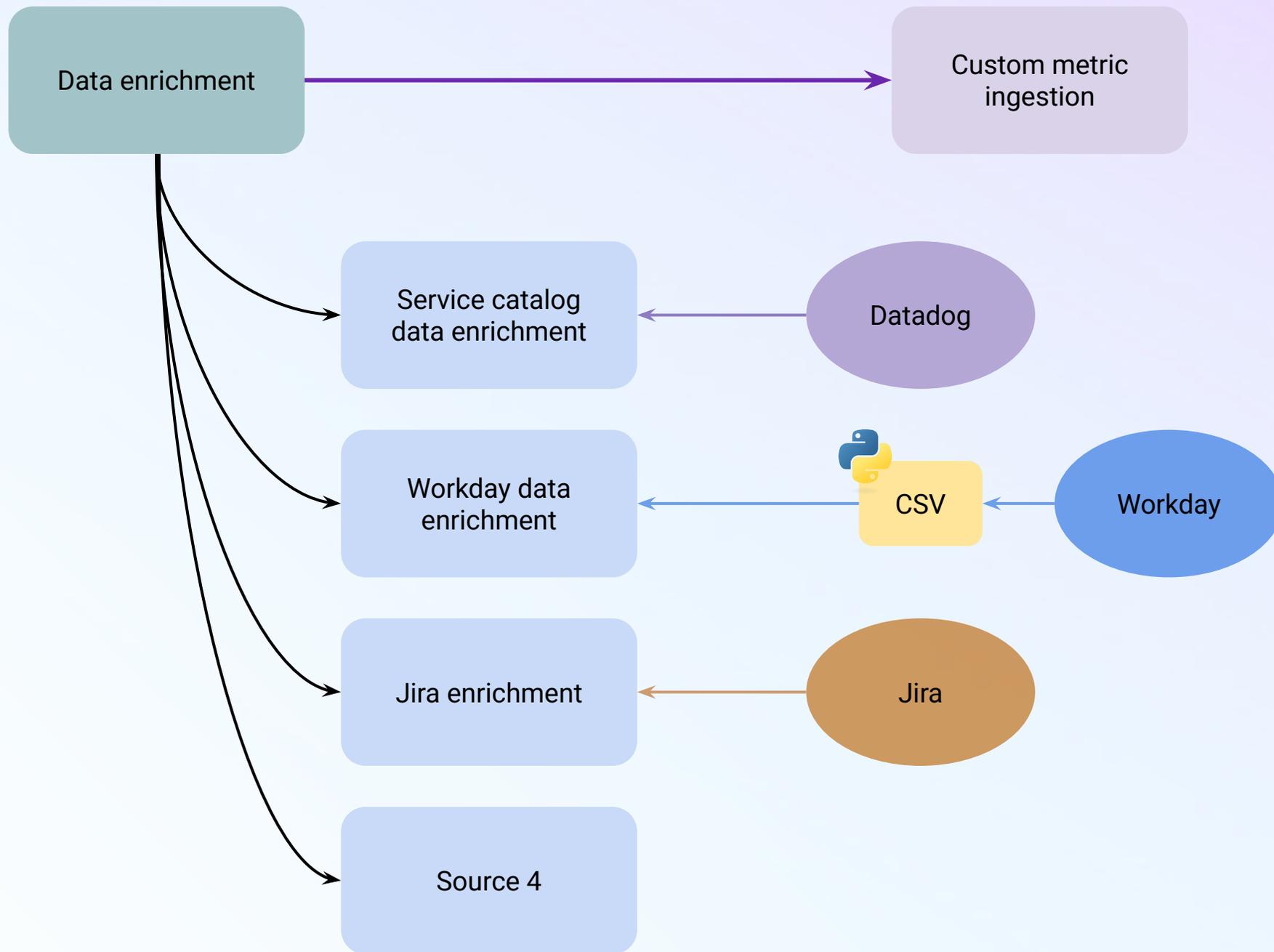


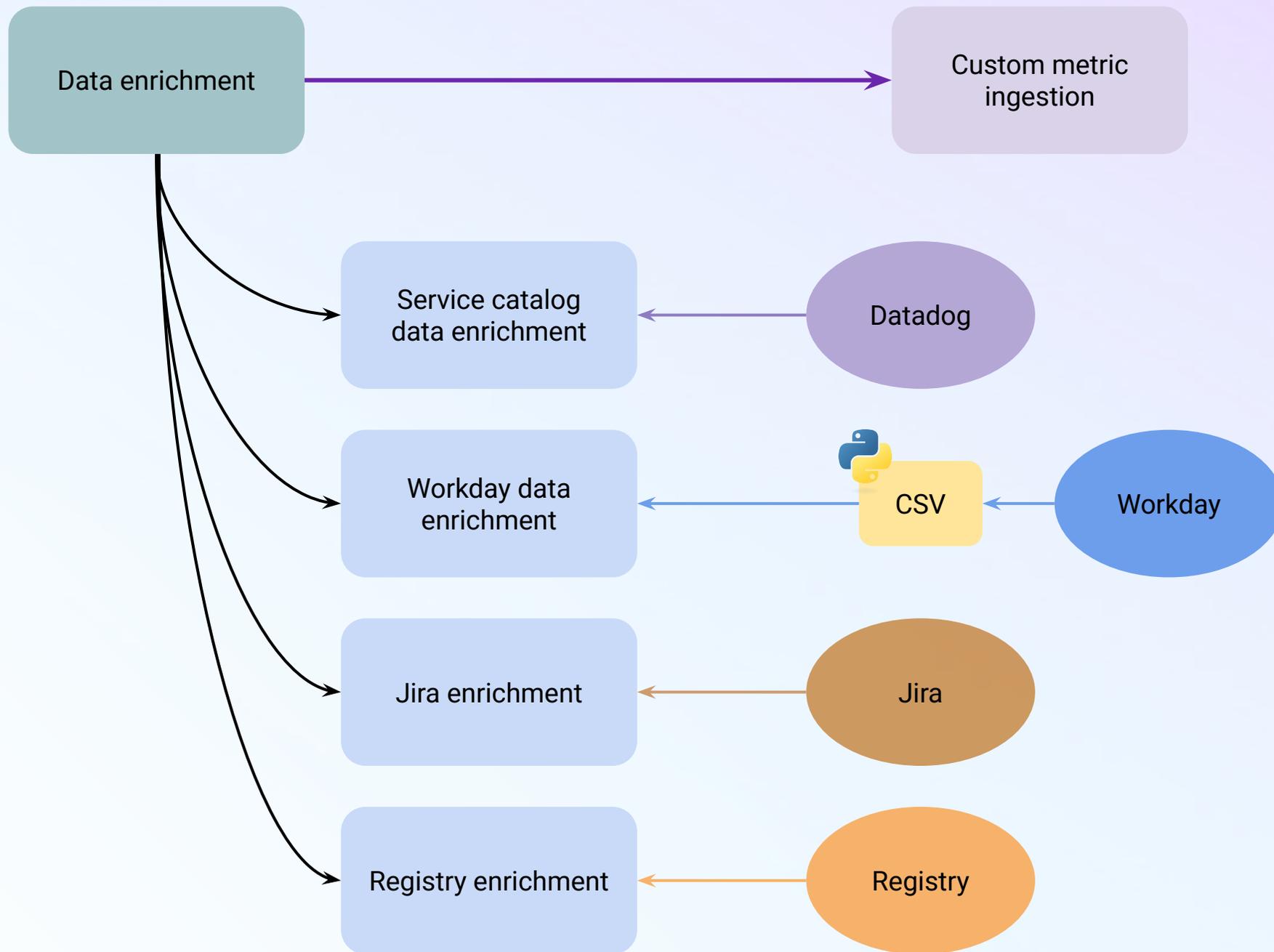
Meet our Astro VP - Rosie

How to get traction and velocity at scale?









Saved Views ▾ -service ▾ team_handle ▾ team ▾ cpu_arch ▾ cloud_provider ▾ datacenter ▾ kube_cluster_name ▾ namespace ▾ nodegroup ▾
 instance-type ▾ arm_eta ▾ cpu_arch_supported ▾ Edit

1mo Jul 1, 2023, 12:00 am - Jul 31, 2023, 1... ⌵ ⏪ ⏩ 🔍

ARM Adoption

45.71 %

ARM Registry Readiness (k8.cpu.reques...

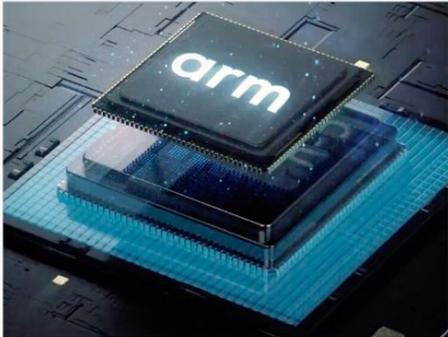
76.46 %

Exception total shares (cpu)

20.3 %

JIRA ARM Migration coverage (cpu)

60.64 %



Template variables should be helpful (service, team, team_handle)

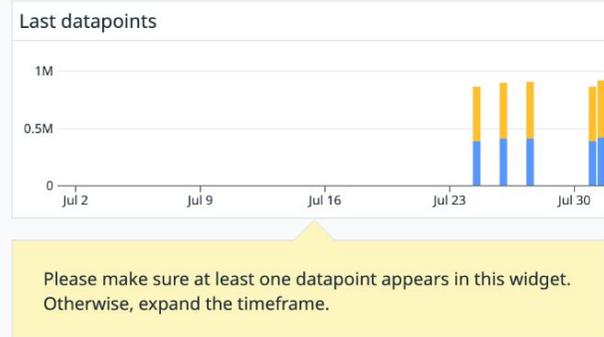
Some Key program information:

- ARM migration is intended to generate cost savings.
- Scope of the migration is AWS Production

Useful links:

- [How to migrate your workload on ARM - Confluence](#)
- [ARM FAQ - Confluence](#)

For support: use [#k8s-on-arm](#)



> What is the ARM migration status of my service(s)? What is the JIRA associated card?
 3 widgets

> Which Services should I migrate first?
 6 widgets

> Which services lack ARM Migration visibility?
 3 widgets

> How much \$ saving the ARM migration can generate?
 1 widget

> I am on ARM, I much \$ do I save per month?
 1 widget

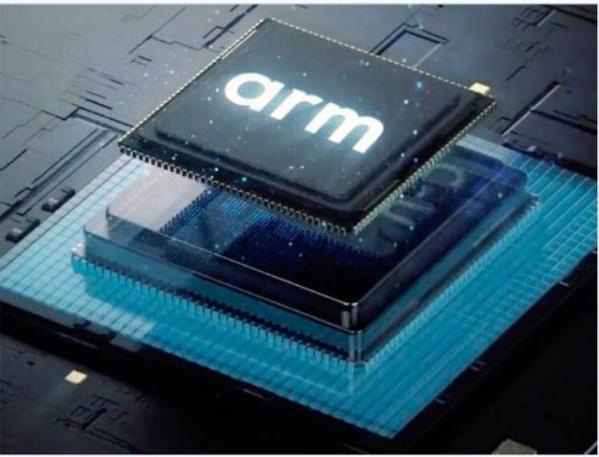
> What is the status of my organisation?
 1 widget

☆ [Program] Let's migrate to ARM !

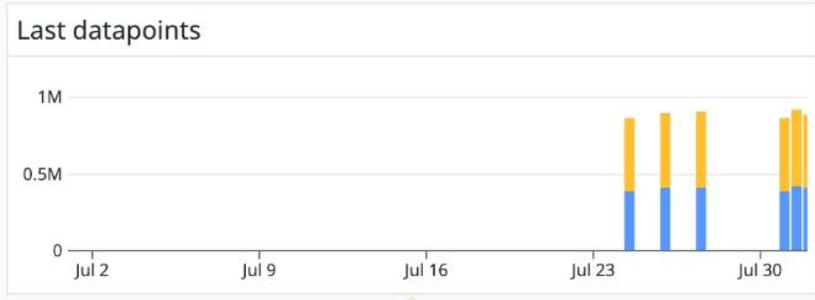
Share Show Overlays Configure Add Widgets

Filters: Saved Views, service, team_handle, team, cpu_arch, cloud_provider, datacenter, kube_cluster_name, namespace, nodegroup, instance-type, arm_eta, cpu_arch_supported, Edit

1mo Jul 1, 2023, 12:00 am - Jul 31, 2023, 1... [Navigation icons]



Template variables should be helpful (service, team, team_handle)
Some Key program information:
• ARM migration is intended to generate cost savings.
• Scope of the migration is AWS Production
Useful links:
• How to migrate your workload on ARM - Confluence
• ARM FAQ - Confluence
For support: use #k8s-on-arm



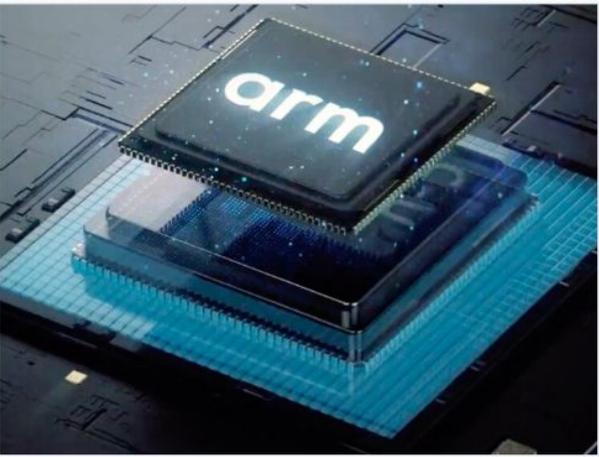
Please make sure at least one datapoint appears in this widget. Otherwise, expand the timeframe.

☆ [Program] Let's migrate to ARM !

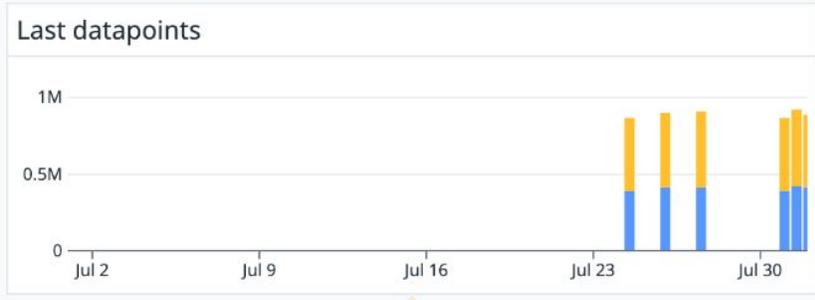
Share Show Overlays Configure Add Widgets

Filters: Saved Views, service, team_handle, team, cpu_arch, cloud_provider, datacenter, kube_cluster_name, namespace, nodegroup, instance-type, arm_eta, cpu_arch_supported, Edit

1mo Jul 1, 2023, 12:00 am - Jul 31, 2023, 1... [Navigation icons]



Template variables should be helpful (service, team, team_handle)
Some Key program information:
• ARM migration is intended to generate cost savings.
• Scope of the migration is AWS Production
Useful links:
• How to migrate your workload on ARM - Confluence
• ARM FAQ - Confluence
For support: use #k8s-on-arm



Please make sure at least one datapoint appears in this widget. Otherwise, expand the timeframe.



How much \$\$\$ savings has the ARM migration generated?

Cost Analysis is based on the following assumptions while moving away from AMD64 to the corresponding and recommended ARM64 instance family

Included instance types in savings calculation:

- c5. to c6g.
- m5. to m6g.
- r5. to r6g.
- c5d. to c6gd.
- m5d. to m6gd.
- r5d. to r6gd.
- c5n. to c6gn.
- c6i. to c7g.
- m6i. to m7g.
- r6i. to r7g.
- i3. to i4g.
- i4i. to i4g.
- i3en. to im4gn. (note: go up one size)
- m5n. to m7g. (note: using m7g for estimates; more likely move is m6g; m6gn not offered)

Instances that do not have an equivalent and therefore excluded from savings estimates:

- r5n.
- c6id.
- m6id.
- r6id.
- m6in.

Here is the amount of \$ saved based on the current overall ARM Adoption rate

Estimated monthly savings

20.74k\$

Remaining savings based on the AMD64 being used currently in Production

[Explanation schema here](#)

Remaining AMD64 Scope to migrate - Mo Savings Est.

12.56k\$

=

ARM addressable scope - Mo Savings Est. =

8.67k\$

Addressable scope

(sum of both blue numbers)

ARM ready - Mo Savings Est. (no exclusion & ARM ready)

5.33k\$

+

+ ARM Build not activated - Mo Savings Est. (running AMD64 in prod AN...)

3.24k\$

Exceptions - Mo Savings Est.

3.89k\$

Exception scope

This part is already done!

Running on Arm

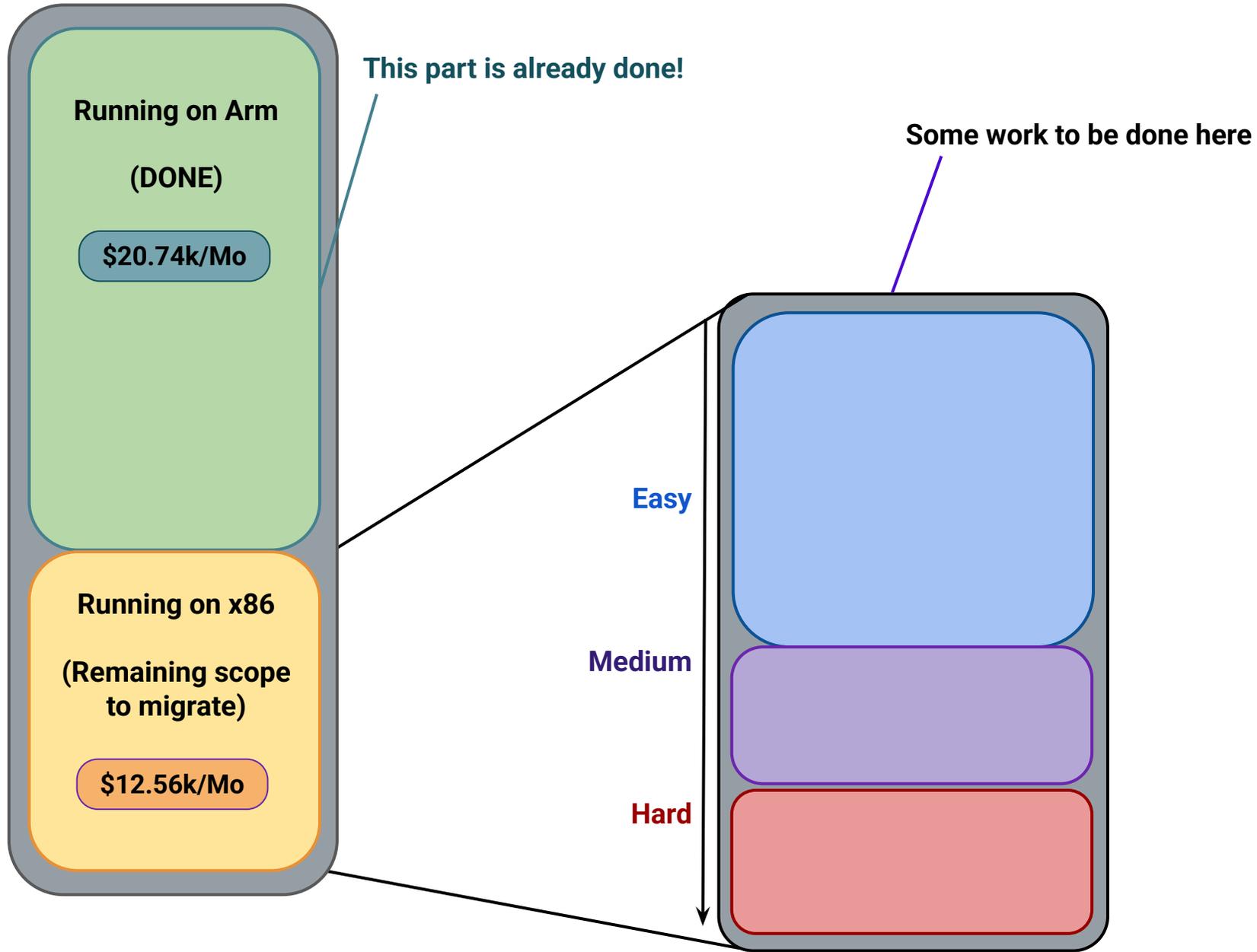
(DONE)

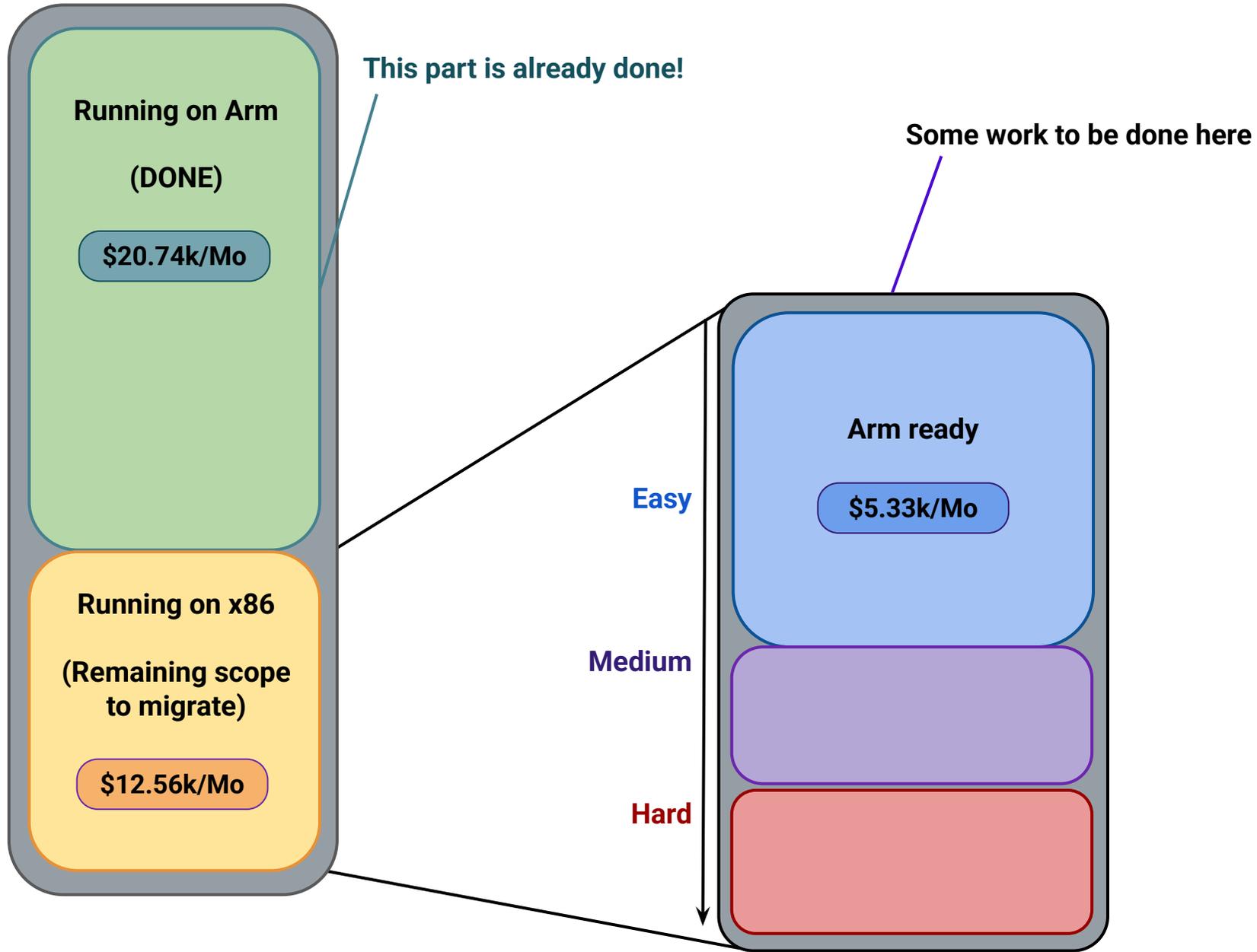
\$20.74k/Mo

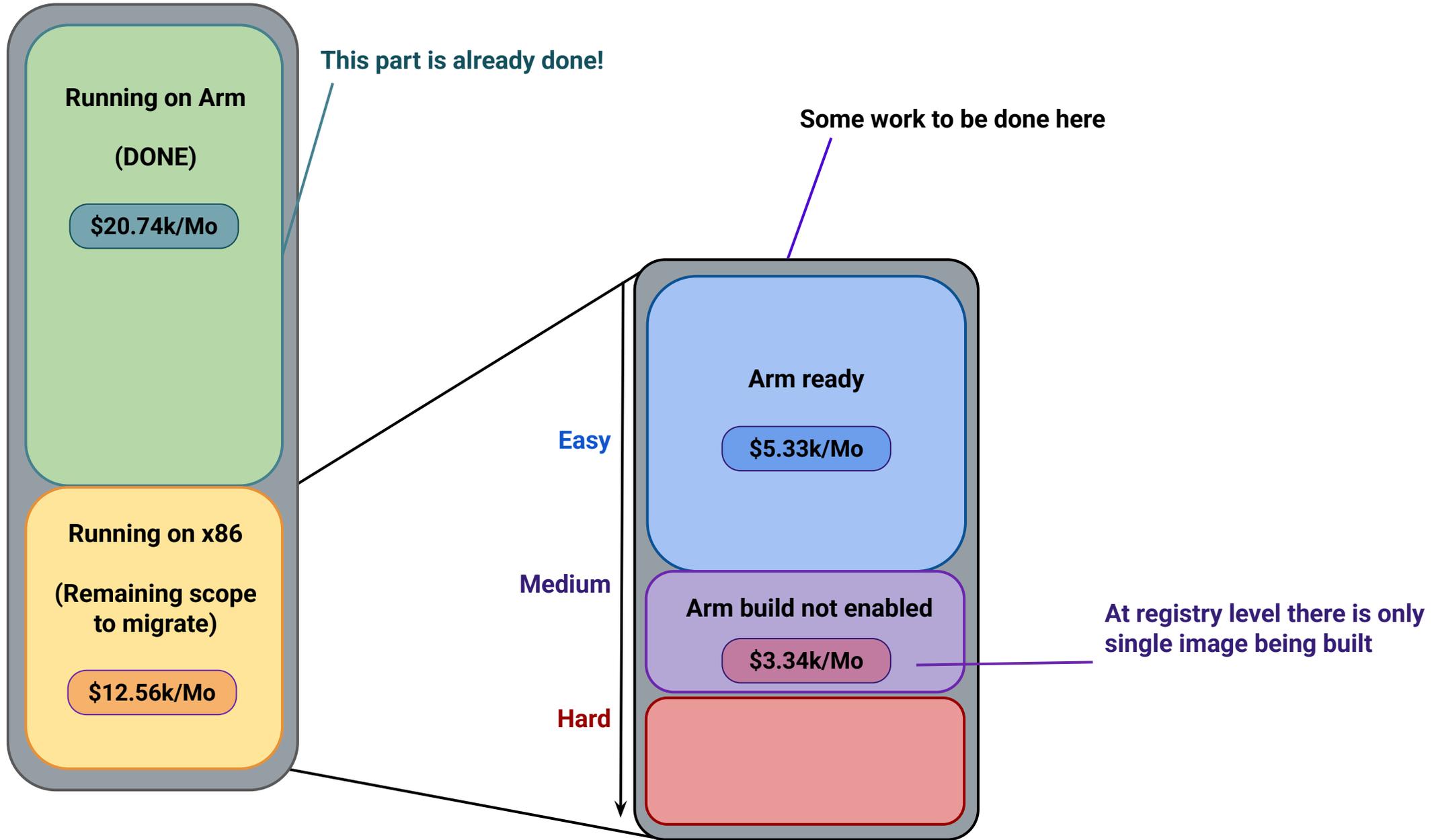
Running on x86

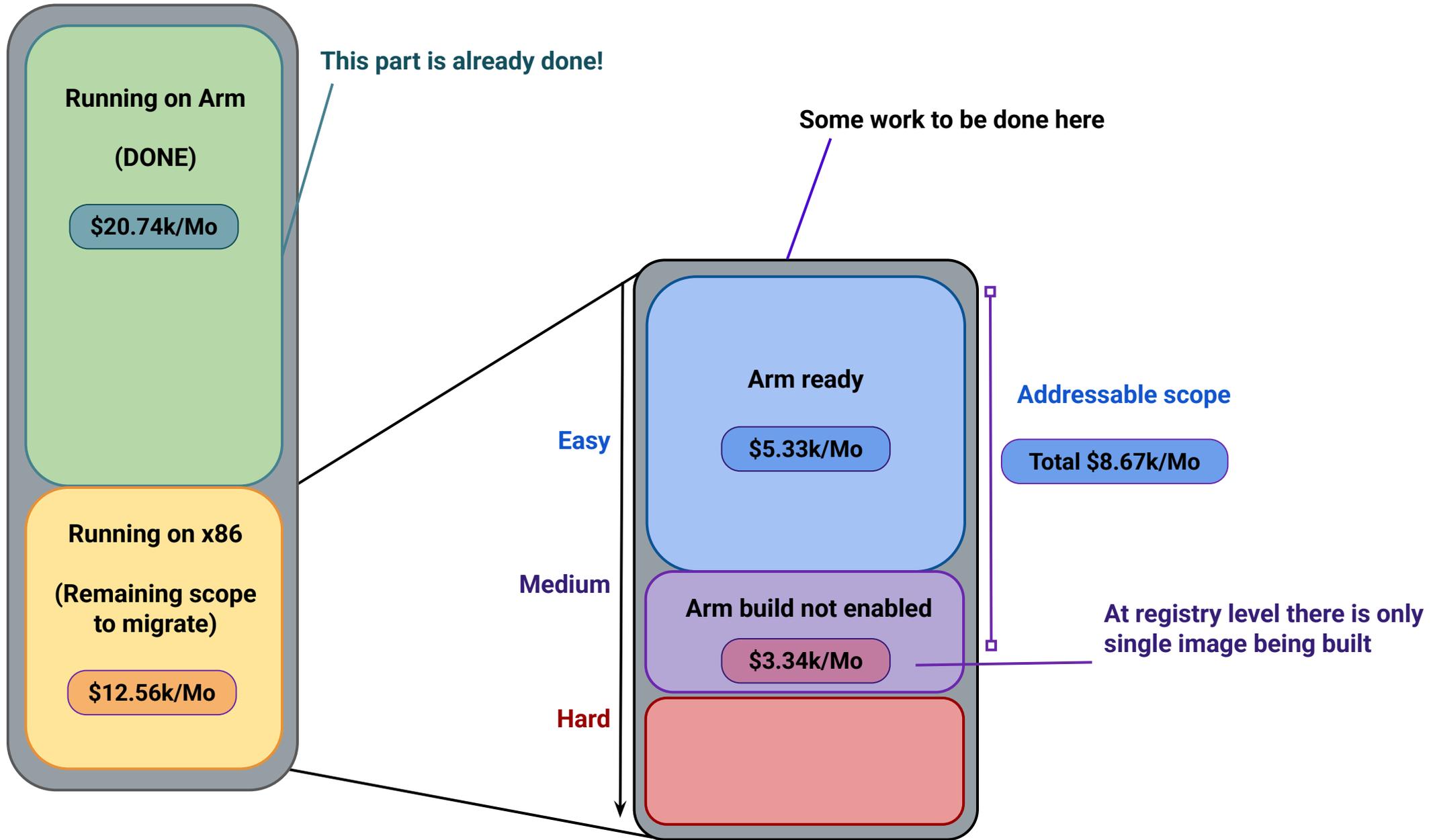
(Remaining scope
to migrate)

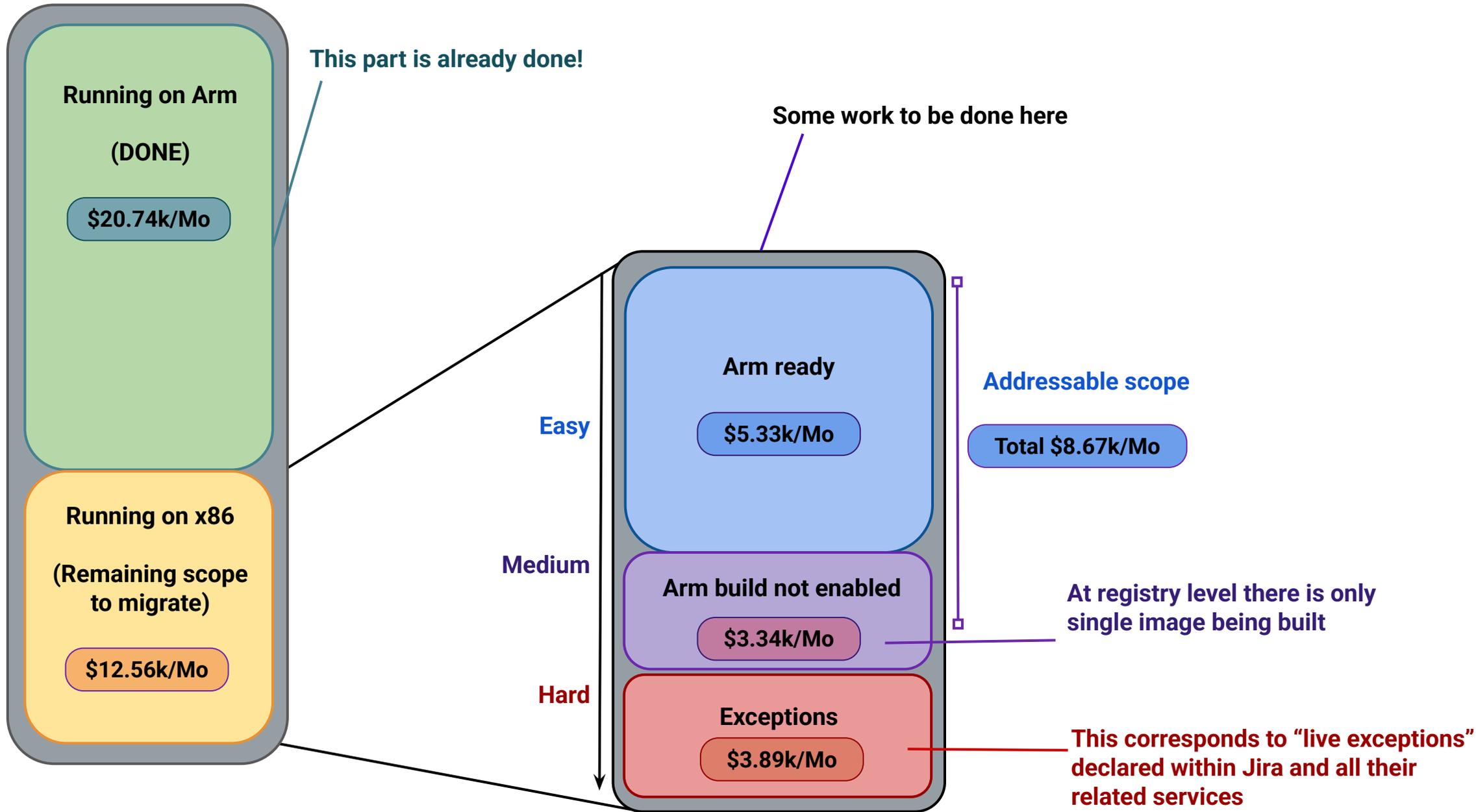
\$12.56k/Mo











Sep 2023



List of services

| SERVICE | ARM_ETA | JIRA_ARMA | ARM_EXCEPTION_CATEGORY | AMD64 CORES | ARM64 CORES | ALL CORES | AMD64 BUT ARM READY |
|---------------------------|----------------------|------------|------------------------|-------------|-------------|-----------|---------------------|
| kafka | blocked_by_exception | arma-36 | N/A | 43,500.0 | 0 | 43,500.0 | 43,500.0 |
| busly | blocked_by_exception | arma-25 | N/A | 32,276.0 | 0 | 32,276.0 | 32,276.0 |
| logs-processing | blocked_by_exception | arma-6 | N/A | 28,083.0 | 0 | 28,083.0 | 187.0 |
| event-processing | blocked_by_exception | arma-6 | N/A | 20,673.0 | 0 | 20,673.0 | 138.0 |
| intake-queue | 2023-q4 | arma-55 | N/A | 11,550.0 | 90.0 | 11,640.0 | 11,550.0 |
| sketch-db | blocked_by_exception | arma-25 | N/A | 11,004.9 | 481.6 | 11,004.9 | 43.9 |
| aws-generic-workers_ | _undefined | _undefined | N/A | 13,953.8 | 0 | 13,953.8 | 13,953.3 |
| alerting-metric-evaluator | blocked_by_exception | arma-23 | N/A | 11,497.7 | 0 | 11,497.7 | 11,343.3 |

Visualizing potential savings

| SERVICE | ARM_ETA | ↓ JIRA_ARMA | ARM_EXCEPTION_CATEGORY | AMD64 CORES | ARM64 CORES | ALL CORES | AMD64 BUT ARM READY | ESTIMATED MO. SAVINGS |
|--------------------------------|---------|-------------|------------------------|-------------|-------------|-----------|---------------------|-----------------------|
| ace | 2023-q4 | arma-92 | N/A | 12.7 | 0 | 12.69 | 12.69 | 30.3 \$ |
| api-key-usage-tracker | 2023-q4 | arma-92 | N/A | 88.5 | 0 | 88.47 | 14.47 | 244.2 \$ |
| api-key-usage-tracker-canary | 2023-q4 | arma-92 | N/A | 4.8 | 0 | 4.82 | 1.82 | 12.1 \$ |
| container-resource-writer | 2023-q3 | arma-91 | N/A | 240.0 | 0 | 240.00 | 0 | 1.16k \$ |
| event-platform-resource-writer | 2023-q3 | arma-91 | N/A | 158.4 | 0 | 158.40 | 14.40 | 325.7 \$ |
| event-platform-resource-writer | 2023-q4 | arma-91 | N/A | 150.7 | 0 | 150.70 | 13.70 | 316.5 \$ |
| container-resource-writer | 2023-q4 | arma-91 | N/A | 240.0 | 0 | 240.00 | 0 | 504.0 \$ |
| ci-app-demo | 2023-q3 | arma-90 | N/A | 0.6 | 0 | 0.60 | 0 | 1.4 \$ |
| resource-host-query | 2023-q3 | arma-9 | N/A | 0 | 54.00 | 54.00 | 0 | 0 \$ |
| resource-processor | 2023-q3 | arma-9 | N/A | 0 | 818.80 | 818.80 | 0 | 0 \$ |
| resource-synchronous-writer | 2023-q3 | arma-9 | N/A | 0 | 3.00 | 3.00 | 0 | 0 \$ |

Lesson 3:

Make work visible

Timeline

Feb 2023 (14%)

CTO issues a challenge:
Migrate 100% of prod
workloads on AWS to
Arm by the end of 2023,
with any exceptions
documented and
approved.



Timeline

Feb 2023 (14%)

CTO issues a challenge:
Migrate 100% of prod workloads on AWS to Arm by the end of 2023, with any exceptions documented and approved.

May 2023 (18%)

Phase 2 underway:
After much trial and error, the program materials and tracking are ready; teams are asked to declare their migration intentions.

Timeline

Feb 2023 (14%)

CTO issues a challenge:
Migrate 100% of prod workloads on AWS to Arm by the end of 2023, with any exceptions documented and approved.

June 2023 (23%)

First half of the year:
Asked teams to consider adding Arm migration as a top priority for their Q3 OKRs.

May 2023 (18%)

Phase 2 underway:
After much trial and error, the program materials and tracking are ready; teams are asked to declare their migration intentions.

Timeline

Feb 2023 (14%)

CTO issues a challenge: Migrate 100% of prod workloads on AWS to Arm by the end of 2023, with any exceptions documented and approved.

June 2023 (23%)

First half of the year: Asked teams to consider adding Arm migration as a top priority for their Q3 OKRs.

May 2023 (18%)

Phase 2 underway: After much trial and error, the program materials and tracking are ready; teams are asked to declare their migration intentions.

July 2023 (43%)

New tracking feature: Saw the biggest jump in adoption and rolled out new features for engineers and executives to track their progress.

Timeline

Feb 2023 (14%)

CTO issues a challenge: Migrate 100% of prod workloads on AWS to Arm by the end of 2023, with any exceptions documented and approved.

May 2023 (18%)

Phase 2 underway: After much trial and error, the program materials and tracking are ready; teams are asked to declare their migration intentions.

June 2023 (23%)

First half of the year: Asked teams to consider adding Arm migration as a top priority for their Q3 OKRs.

July 2023 (43%)

New tracking feature: Saw the biggest jump in adoption and rolled out new features for engineers and executives to track their progress.

Sep 2023 (49%)

Better than expected: Four months later, the program has added an additional savings beyond the previous trajectory.

Timeline

Feb 2023 (14%)

CTO issues a challenge: Migrate 100% of prod workloads on AWS to Arm by the end of 2023, with any exceptions documented and approved.

May 2023 (18%)

Phase 2 underway: After much trial and error, the program materials and tracking are ready; teams are asked to declare their migration intentions.

June 2023 (23%)

First half of the year: Asked teams to consider adding Arm migration as a top priority for their Q3 OKRs.

July 2023 (43%)

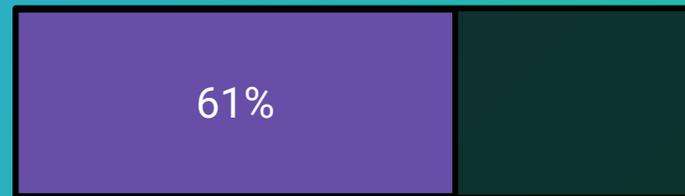
New tracking feature: Saw the biggest jump in adoption and rolled out new features for engineers and executives to track their progress.

Sep 2023 (49%)

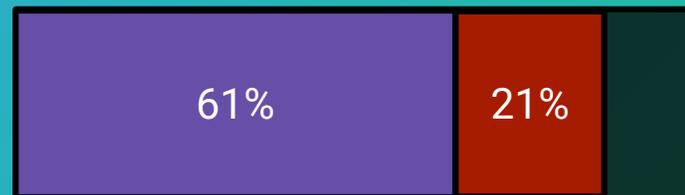
Better than expected: Four months later, the program has added an additional savings beyond the previous trajectory.

Dec 2023

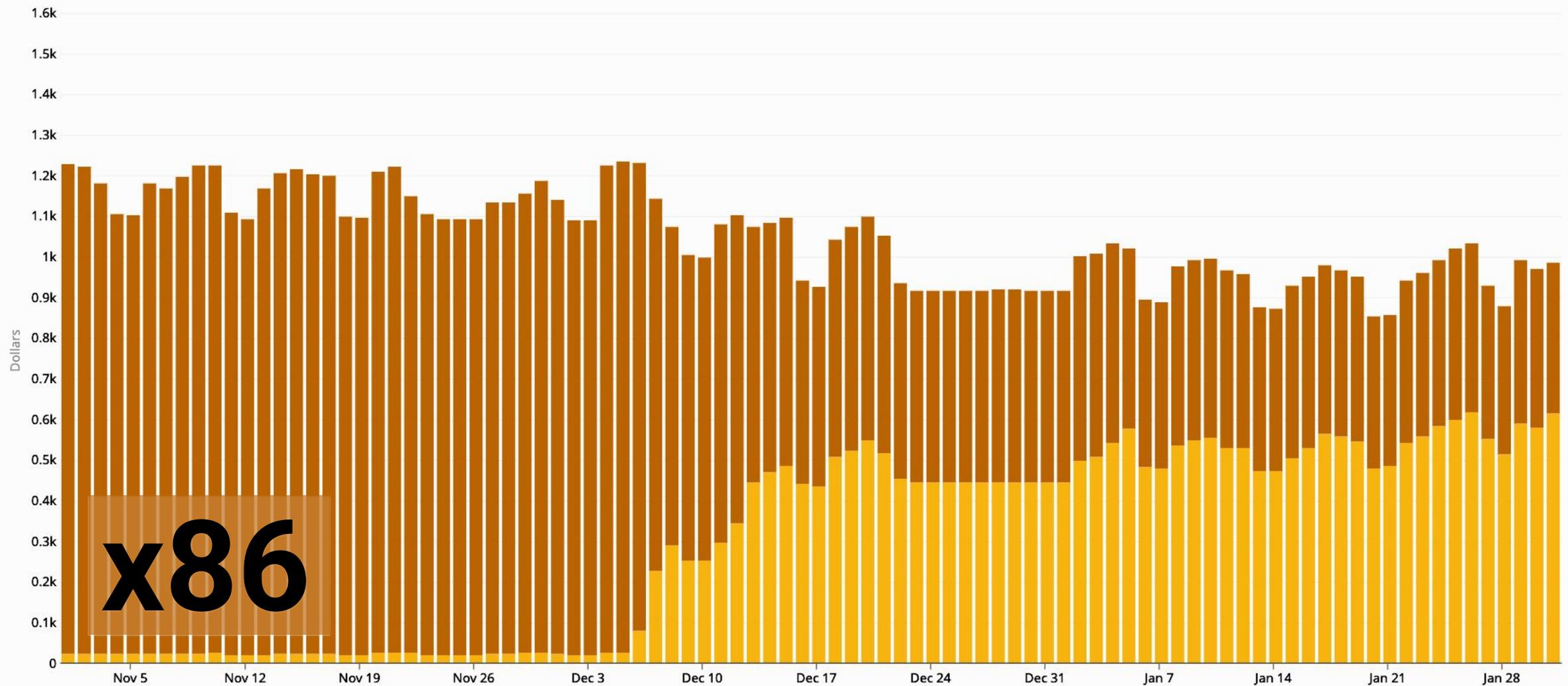
Dec 2023



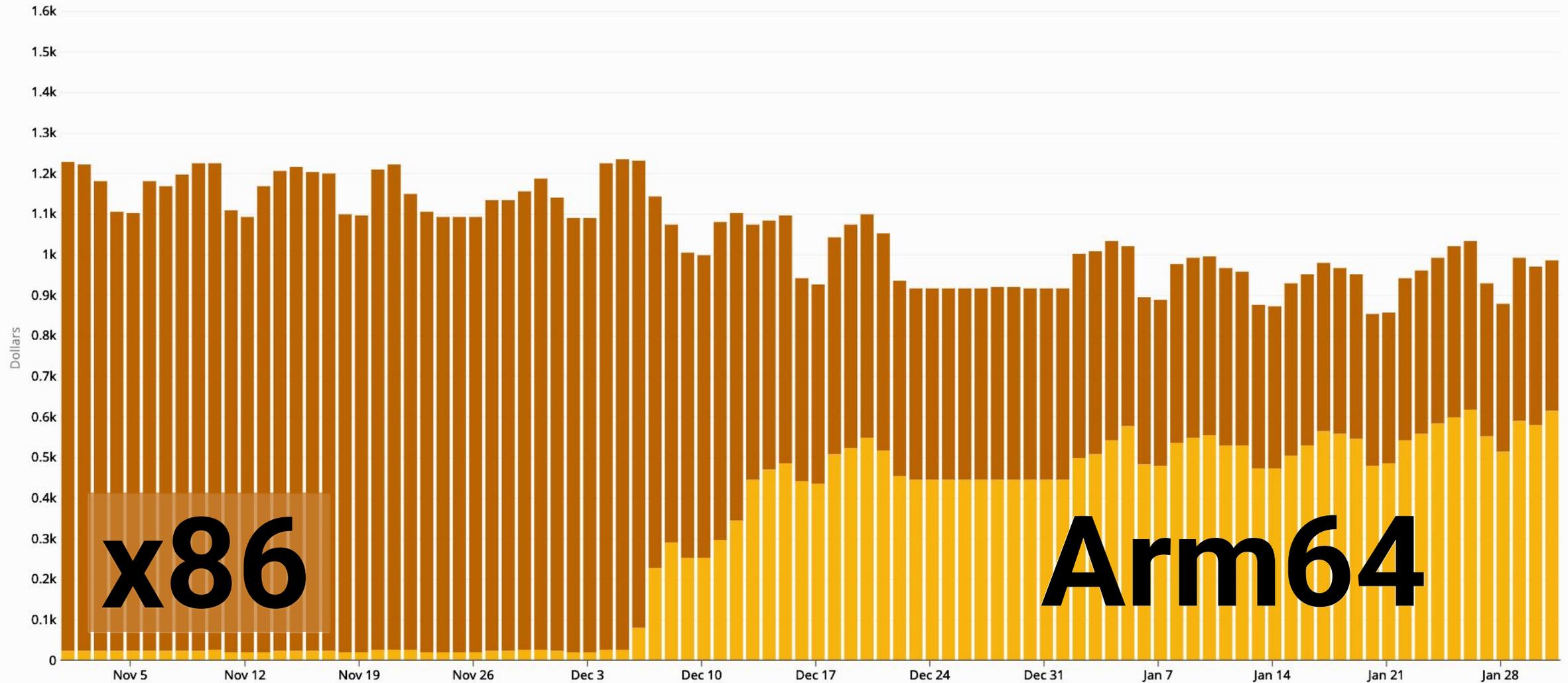
Dec 2023



Example of one service



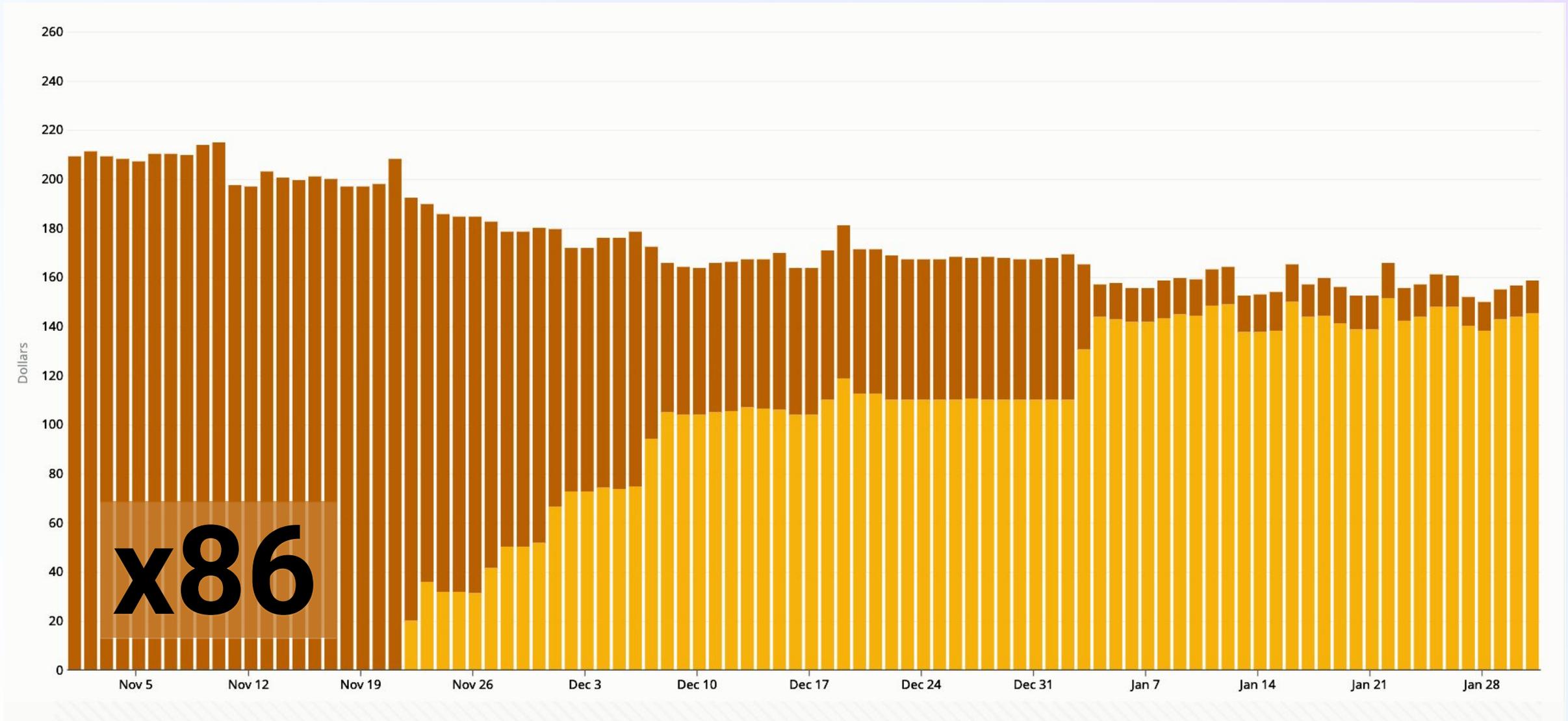
Example of one service



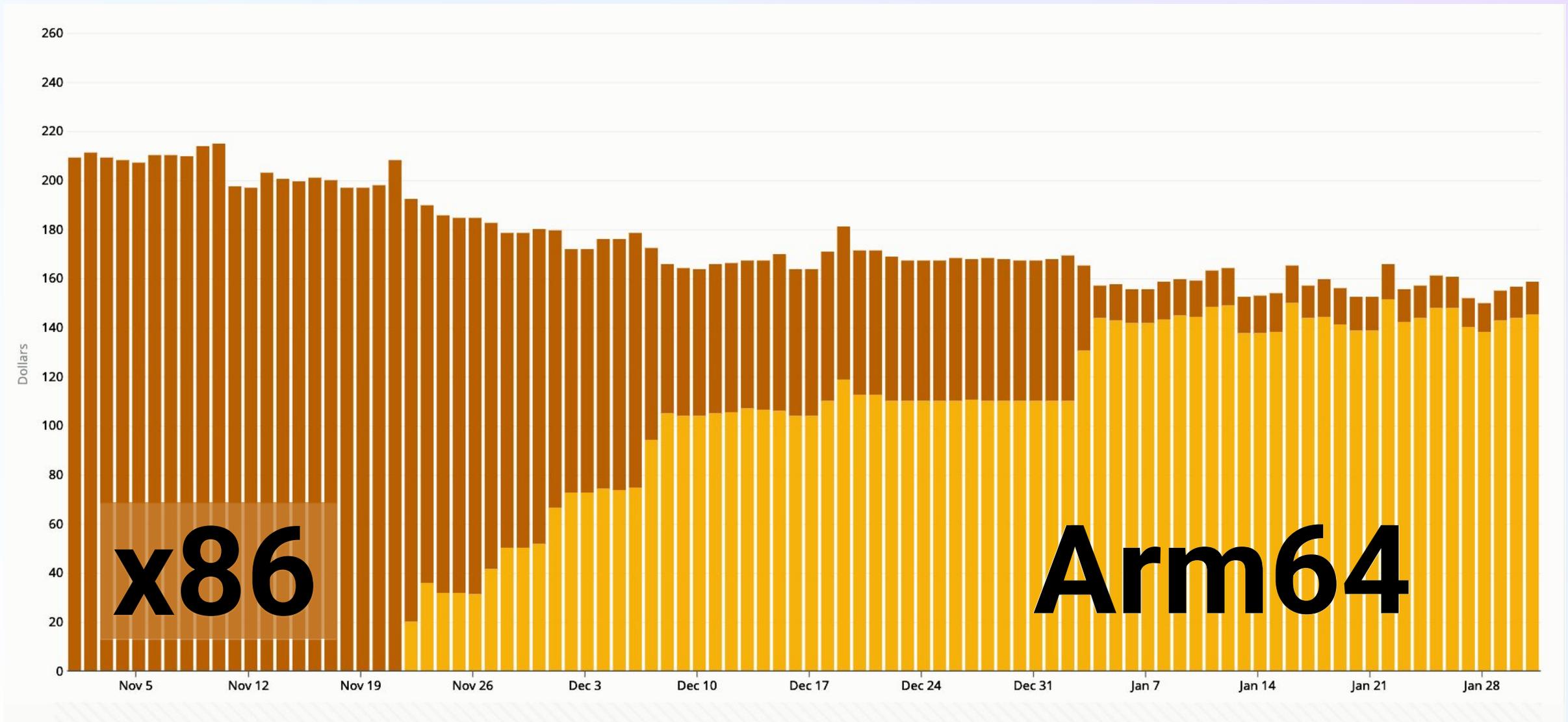
x86

Arm64

Another example of one service



Another example of one service

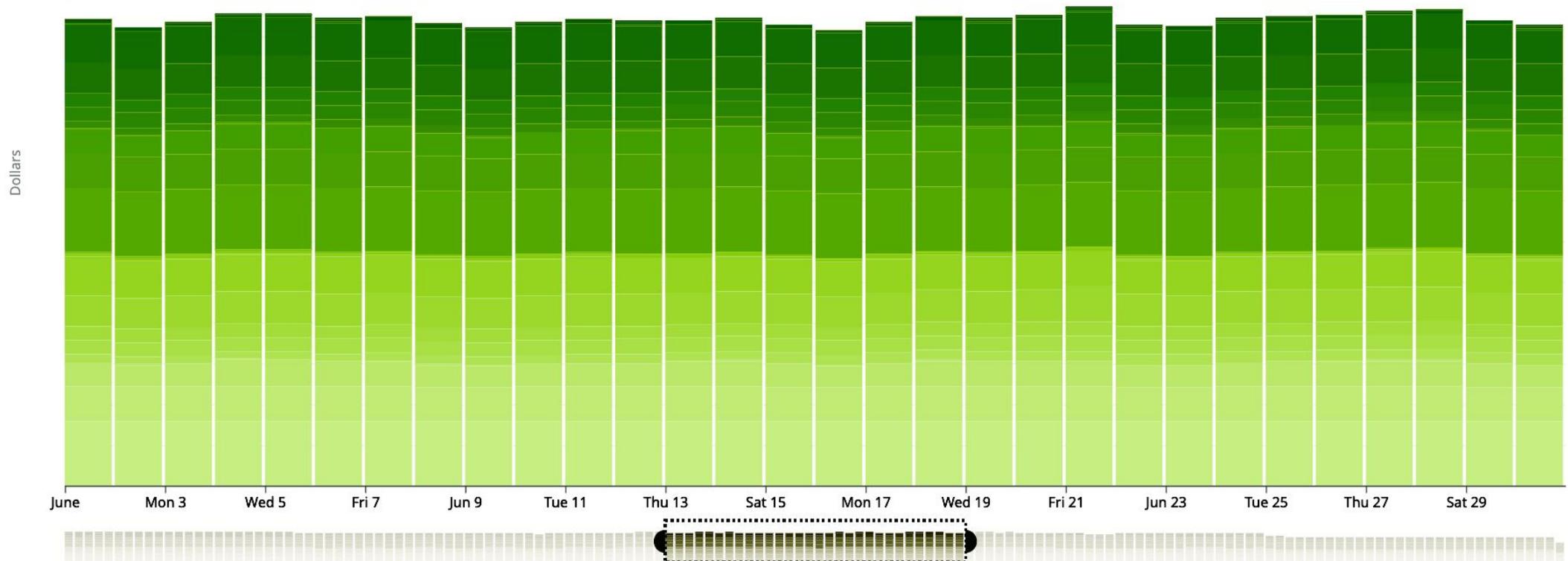


What's next?

True cost of exceptions due to depreciation

Services marked for depreciation/replacement

[Save to Dashboard](#) [More...](#)



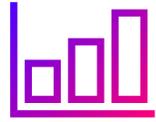
Previous-generation instance type utilization



**MOST
ORGANIZATIONS
STILL USE
PREVIOUS-
GENERATION
EC2 INSTANCE
TYPES**

Source: Datadog

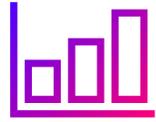
Recap



**Iterate early
and often**

- Share your ideas with others - build on it and get feedback to build the best custom solution

Recap



Iterate early and often

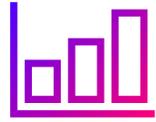
- Share your ideas with others - build on it and get feedback to build the best custom solution



Reduce friction

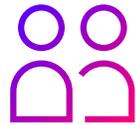
- Check as you go to make sure all of your tools work the way you anticipated

Recap



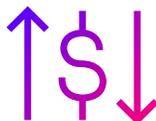
**Iterate early
and often**

- Share your ideas with others - build on it and get feedback to build the best custom solution



**Reduce
friction**

- Check as you go to make sure all of your tools work the way you anticipated



**Track
everything**

- Make the work as visible as possible so everyone has the same context and show the value of your work

Resources

State of Cloud Costs Report

datadoghq.com/state-of-cloud-costs

How Datadog migrated it's Kubernetes fleet on AWS to Arm at scale

datadoghq.com/blog/arm-migration-aws

Re:Invent 2022 Talk

dtdg.co/adopting-arm-lessons-reinvent-2022

Kubecon 2022 Talk

dtdg.co/building-containers-kubecon-2022

Kubecon 2024 Talk

dtdg.co/arm-wrestling-kubecon-2024



How Datadog migrated its Kubernetes fleet on AWS to Arm at scale



Matthieu Jaillais



Aaron Kaplan

Published: December 2, 2024



**Want to work on
projects like this? We're
hiring!**

Scan the QR code or visit
careers.datadoghq.com to see
our open roles.



Also come chat with us in our booth!

Thank you!



Please complete the session survey in the mobile app