Migration deep dive with Amazon OpenSearch Service

Kevin Fallis
AWS Principal Analytics Specialist
Solutions Architect

Agenda

- Amazon OpenSearch Service overview
  - Benefits
  - Open source
- Migration mechanisms
  - Planning
  - Patterns
  - Overall execution strategies
- Q&A

Amazon OpenSearch Service

Amazon OpenSearch Service makes it easy for you to perform interactive log analytics, real-time application monitoring, website search, and more. OpenSearch is an open source, distributed search and analytics suite derived from Elasticsearch.

Amazon OpenSearch Service – version support

Amazon OpenSearch Service offers the latest versions of OpenSearch, support for 19 versions of Elasticsearch (1.5 to 7.10 versions), and visualization capabilities powered by OpenSearch Dashboards and Kibana (1.5 to 7.10 versions)

Amazon OpenSearch Service – benefits

Easy to Deploy and Manage

Deploy your OpenSearch or Elasticsearch cluster in minutes. The service simplifies management tasks such as provisioning, patching, failure recovery, backups, and monitoring.

Highly Secure

Encrypt data at-rest and in-transit using keys you create, and manage authentication and access control with SAML and AWS IAM policies.

Cost Effective

You pay only for the resources you consume. You have a variety of pricing levers from on demand to reserved instance plans to dial in savings that meet your business needs.

Highly Scalable and Available

You can easily scale your cluster up or down via API or a few clicks in the AWS console. The service is designed to be highly available using multi-AZ deployments.

Community driven open source software

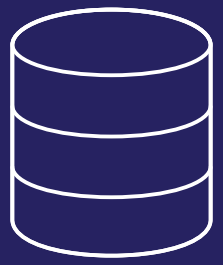
- Amazon OpenSearch Service is built using the OpenSearch project
- OpenSearch is an ALv2 distribution derived from Apache 2.0 licensed Elasticsearch 7.10.2
- The OpenSearch project consists of a distributed search engine powered by Apache Lucene, OpenSearch and a data visualization and user interface, OpenSearch Dashboards
- OpenSearch includes all of the advanced functionality ported over from Open Distro

Migration planning

Migration roadmap – qualify and plan

Qualify and Plan

Size first by storage then by CPU

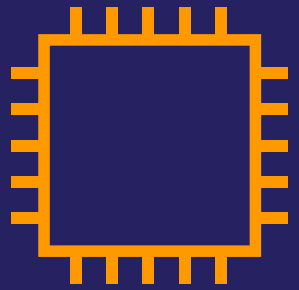


Storage

- On disk an index with 1 replica is ~2.2-2.5 times the source data size with default mappings. Deploy instances with that much storage plus 25% overhead

CPU

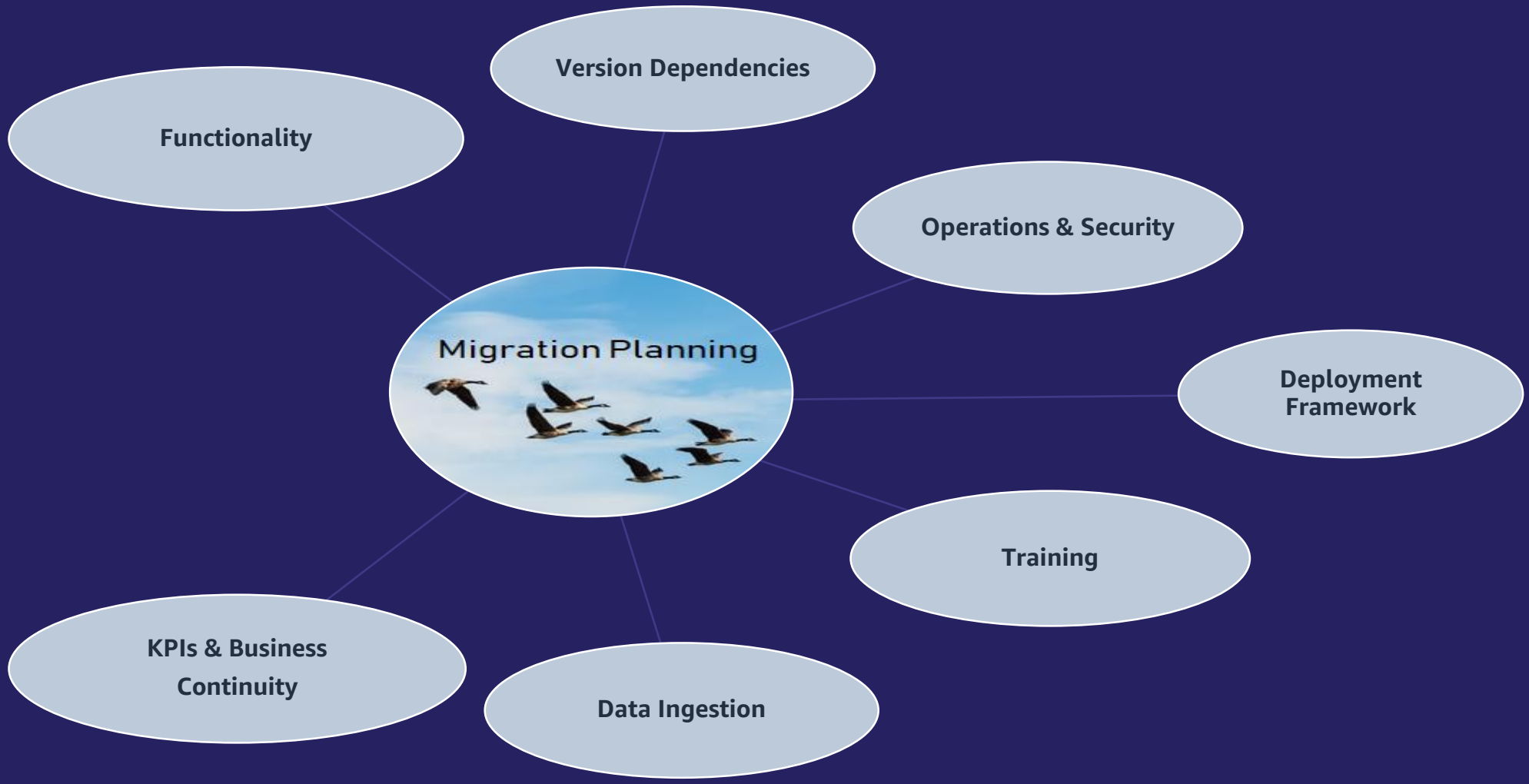
- Each shard uses CPU to process a request. Deploy instances with vCPUs equal to 1.5 times the number of shards that receive substantial traffic



Instance Types

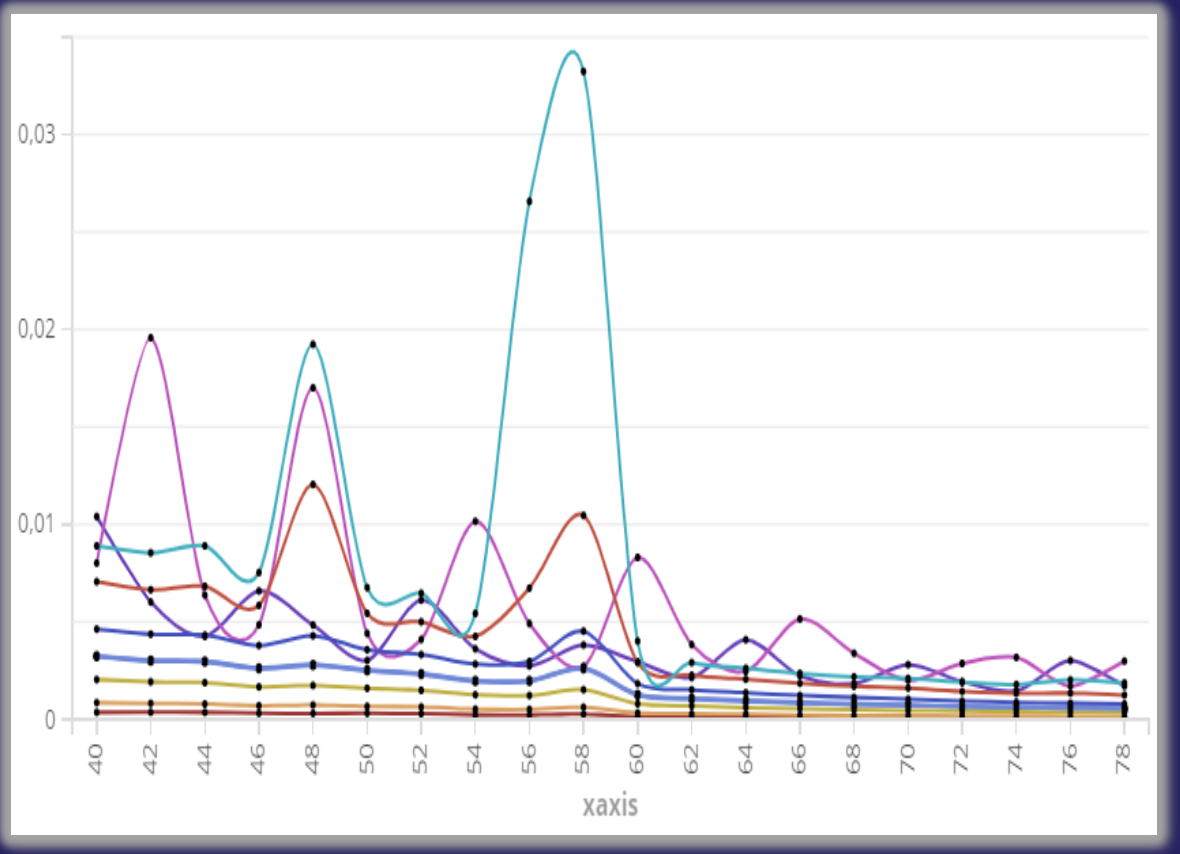
- Each instance family has unique attributes. Graviton 2 instances provide superior performance and price benefits over 5th generation instances

Focus areas for migration planning



Focus area – business continuity and KPIs

- **Operational performance**
  - Uptime & latency
  - Customer SLAs
  - Reads & writes per second
- **Process performance**
  - Tickets
  - MTTR
- **Seamless transition to new services**
- **Financials**
  - Solution cost
  - Staffing

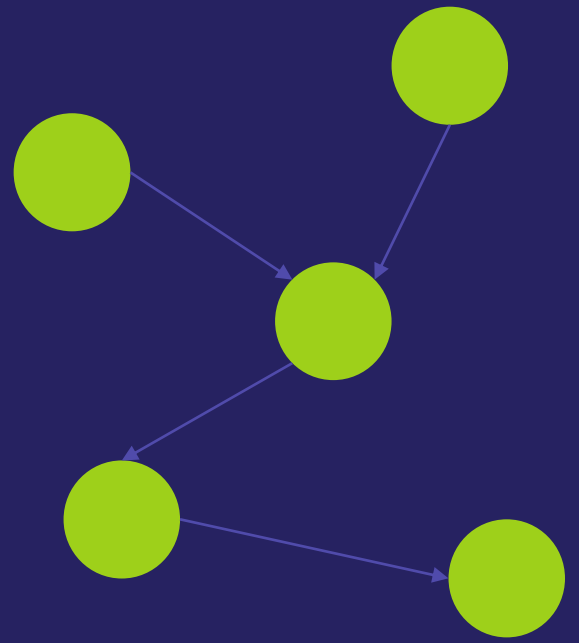


Focus area – business continuity and KPIs

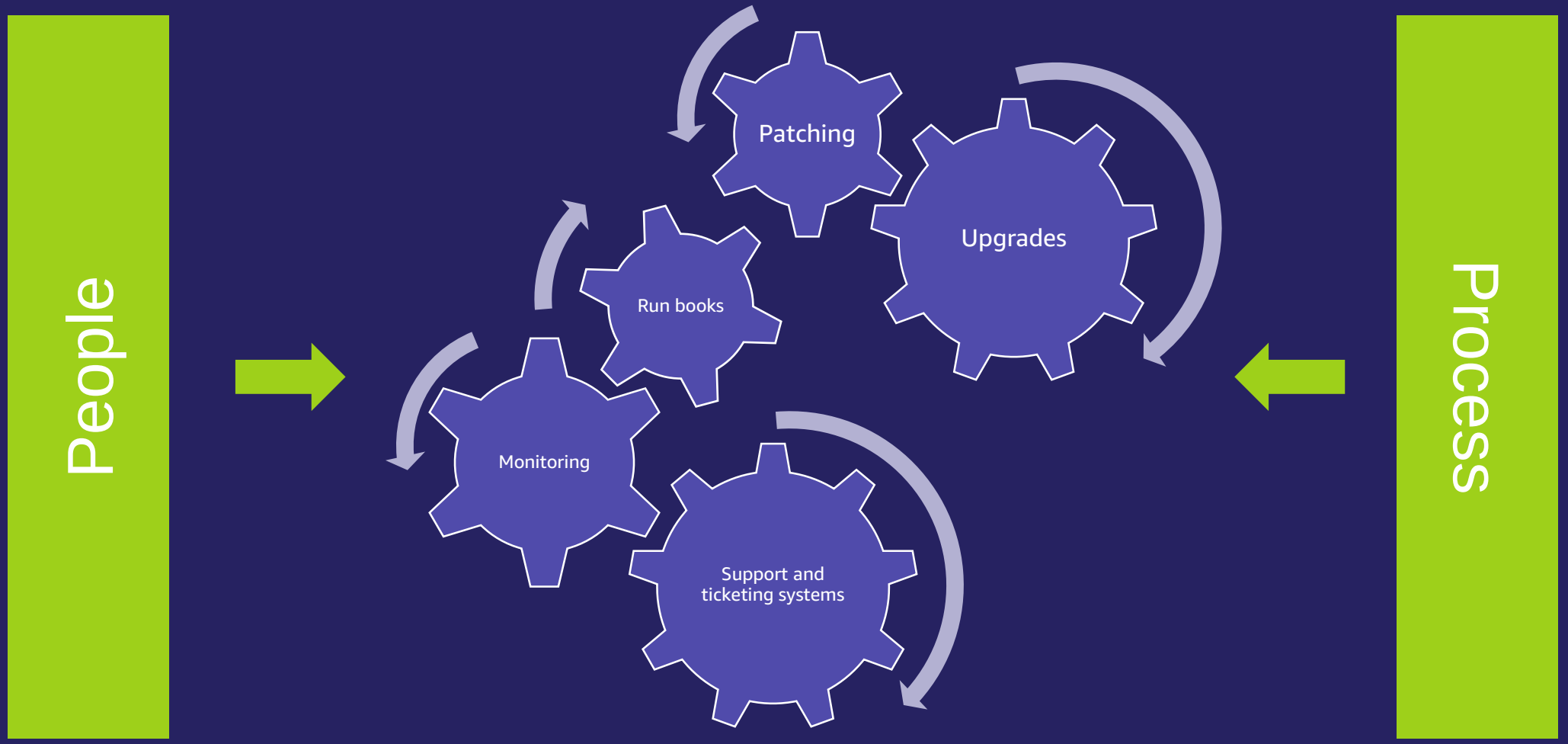
- Is critical business revenue dependent on customization of your current search engine?
- Does your target migration platform support features you need?
- If using plugins, are they Apache 2.0 open-sourced or licensed?
- Can you achieve similar business goals using open source Apache 2.0 licensing?

Focus area - functionality

- Does your interfacing software for your business require certain versions of a search engine?
- Is there a specific feature only supported in a certain version?
- Can you upgrade to newer versions for better security and stability?
- Do you have multiple mappings (if coming from Elasticsearch engine)



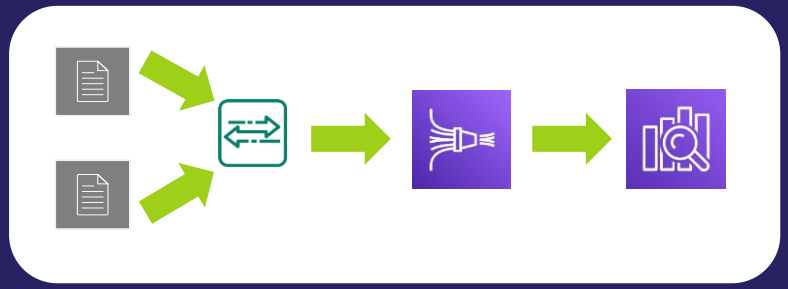
Focus area – operations and security



Focus area – data ingestion

- **Is your current ingestion framework extensible?**
  - Can you easily swap out search engines?
  - Do you have appropriate facades to abstract the search implementation?
- **Does your ingestion technology need modernization?**
- **Can you reuse the existing implementations?**

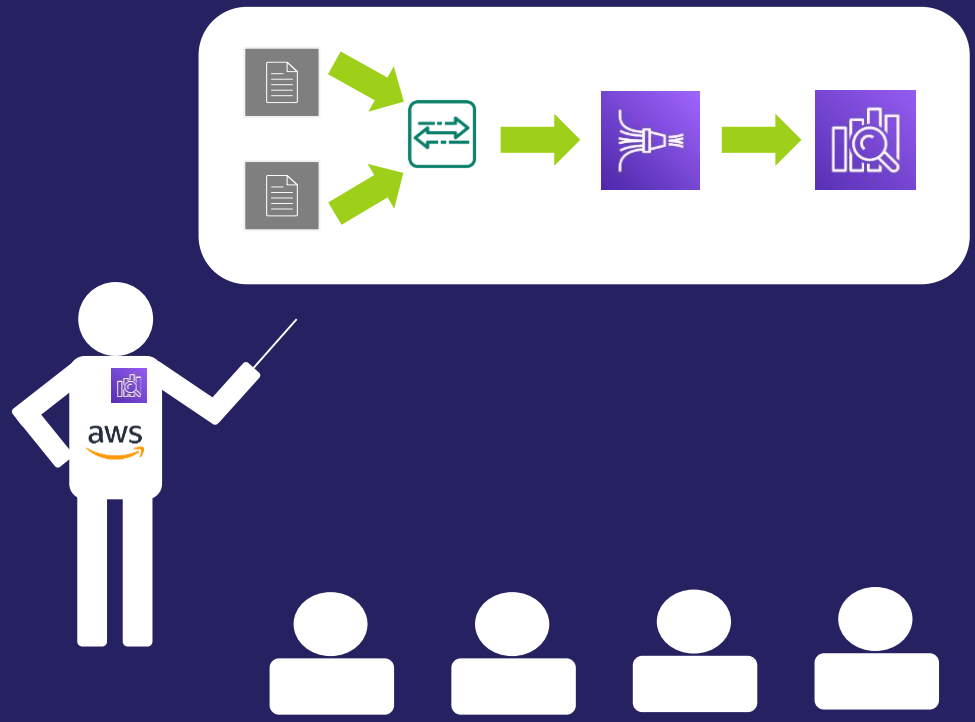
Managed Ingestion



Open Source Ingestion

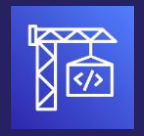
Focus area - training

- **Software development team**
  - APIs
  - Functionality
- **Operations team**
  - What to monitor
  - Event management
- **Support team**
  - Run books
- **Security team**
  - Granular control
  - IDP integration



Focus area – deployment frameworks

- **Is your development ecosystem based on CI/CD?**
- **Do you use open source tooling like Terraform, Chef or Puppet?**
- **Can you use cloud native tools to address deploying your environments to cloud providers?**


AWS CodeBuild


AWS CloudFormation

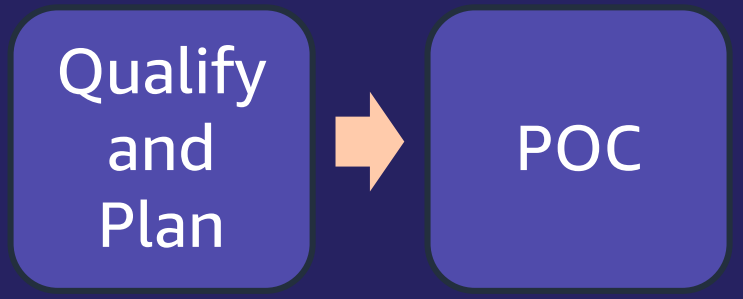

AWS CodeDeploy


AWS CodePipeline


AWS Cloud Development Kit

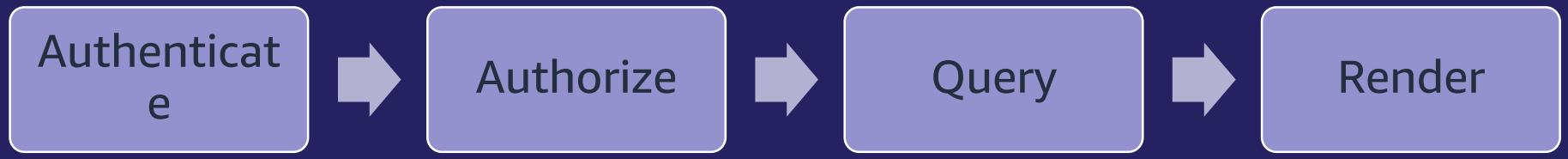

AWS CodeStar

Migration roadmap – proof of concept

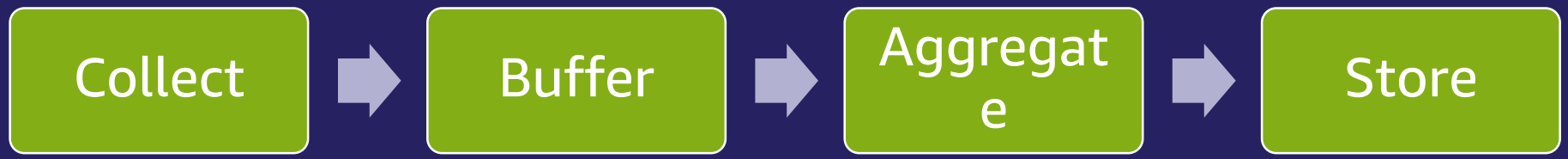


POC – proof of concept

- **Secure funding and ensure you have budgeted for all the components**
- **Automate everything and time box your testing**
- **Exercise all portions of the stack**
  - Presentation layer

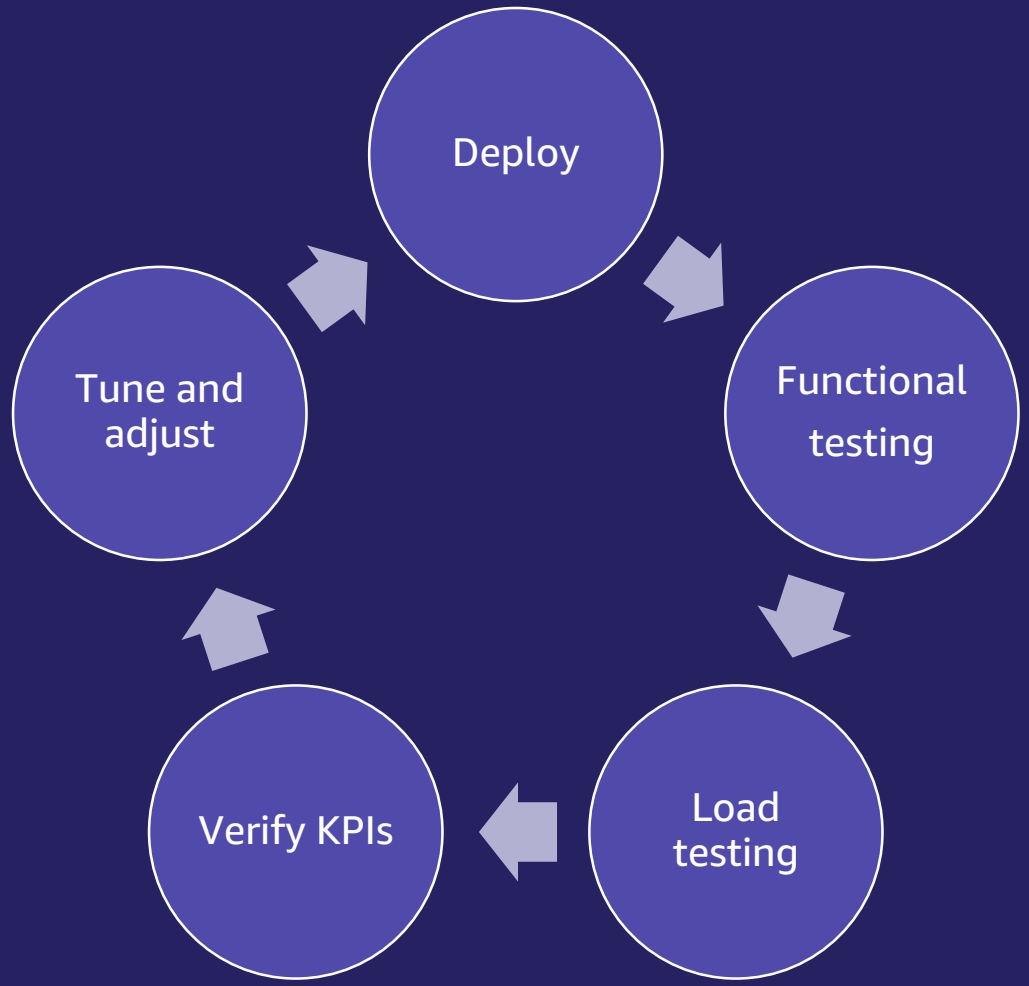


  - Ingestion layer



POC – proof of concept

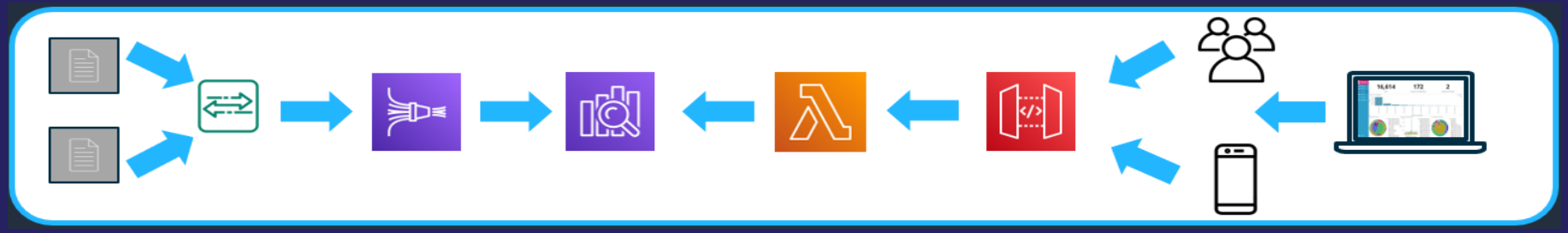
- Don’t be afraid to adjust your plan during this phase
- Thorough testing at all levels
- Simulate failures in all portions of the stack
- Raise issues early and often
- Ensure you are meeting KPIs
- Refine your automation and monitoring
- Build your run books in this phase



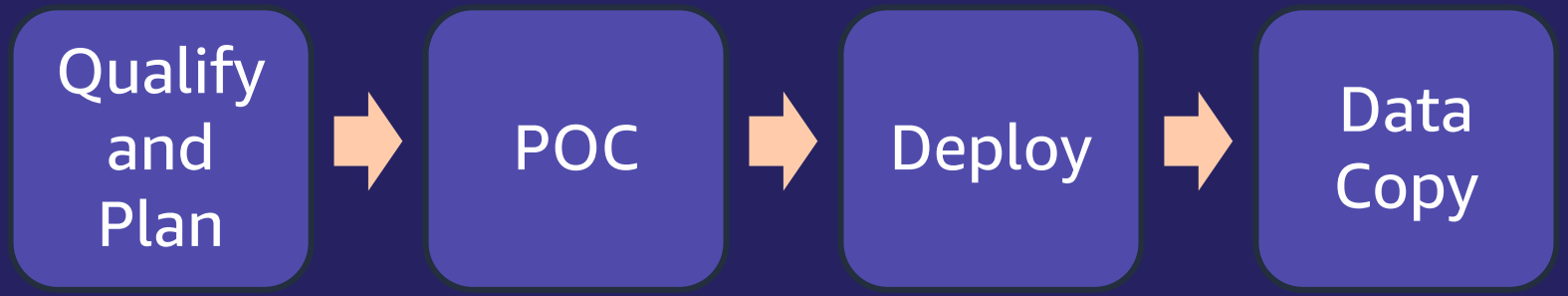
Migration roadmap – deploying to production

Deploying to production

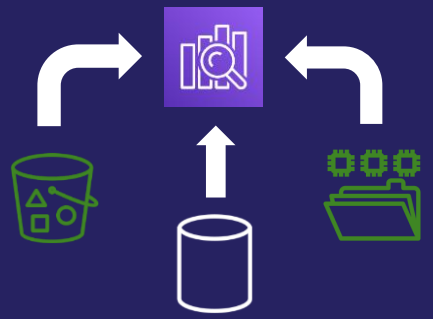
- Exercise the deployment automation and validate it works as expected
- Verify security mechanisms
- Ensure all monitoring tooling is in place
- Exercise alarms
- Finalize run books
- Pen test



Migration roadmap – data migration



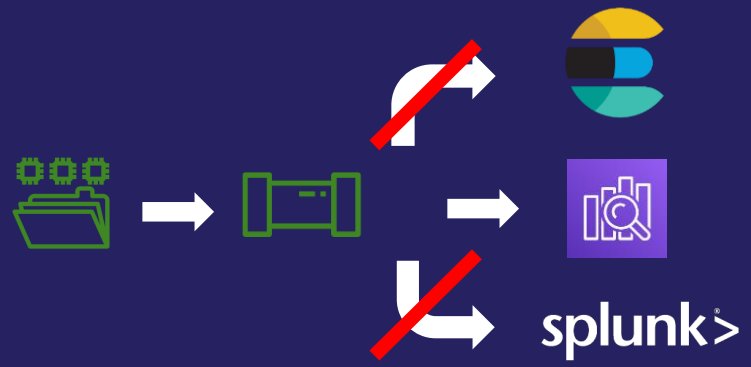
Data migration patterns



Build from document source

Build from snapshot

Build from existing search engine



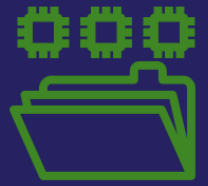
Repoint ingestion pipeline

Building from source

Building from source – popular data repositories

Relational databases

NoSQL Key Value stores



File systems

Object stores

Building from source – popular source formats

Building from source – tooling considerations

Level of effort

Custom code	Connector based tools	Purpose built tools
• You build • You deploy • You manage	• You configure • You deploy • You manage • Usually open source	• Managed service ***or*** AMI you deploy and manage • Data movement scripts

Build from source – custom code

Effort - High

- Choice of programming language
- Typically requires a development effort
- The effort is basically ***"throw away"*** once done
- Typically deploy on Amazon EC2, Containers, or AWS Lambda

```
private BulkRequest request;
private String esEndpoint;
public BulkIndexer(String esEndpoint, String region,
AWSCredentialsProvider credentialsProvider){
this.esEndpoint = esEndpoint;
        AWS4Signer signer = new AWS4Signer();
        signer.setServiceName("es");
        signer.setRegionName(region);
        HttpRequestInterceptor interceptor = new
AWSRequestSigningApacheInterceptor("es", signer,
credentialsProvider);
        rhlc = new
RestHighLevelClient(RestClient.builder(new
HttpHost(esEndpoint,443,"https")).setHttpClientConfigCal
lback(hacb -> hacb.addInterceptorLast(interceptor)));
        request = new BulkRequest();
}
```

Building from source – connector based tools

Effort - Medium

- Configuration driven
- Typically leverages open source
- Time is spent in creating the environment
- Typically deployed on Amazon EC2, Containers, AWS Lambda, or Hadoop ecosystem

```
input {
  s3 {
    "bucket" => "my_raw_data"
      "additional_settings" => {
        "force_path_style" => true }
  }
}
output {
  amazon_es {
    hosts => ["mycluster.eu-central-1.es.amazonaws.com"]
    region => ["eu-central-1"]
    index => "proxy-%{+YYYY.MM.dd}" }
}
```

Elasticdump

logstash

hadoop

ES Hadoop Connector

fluentd

Building from source – purpose built tools

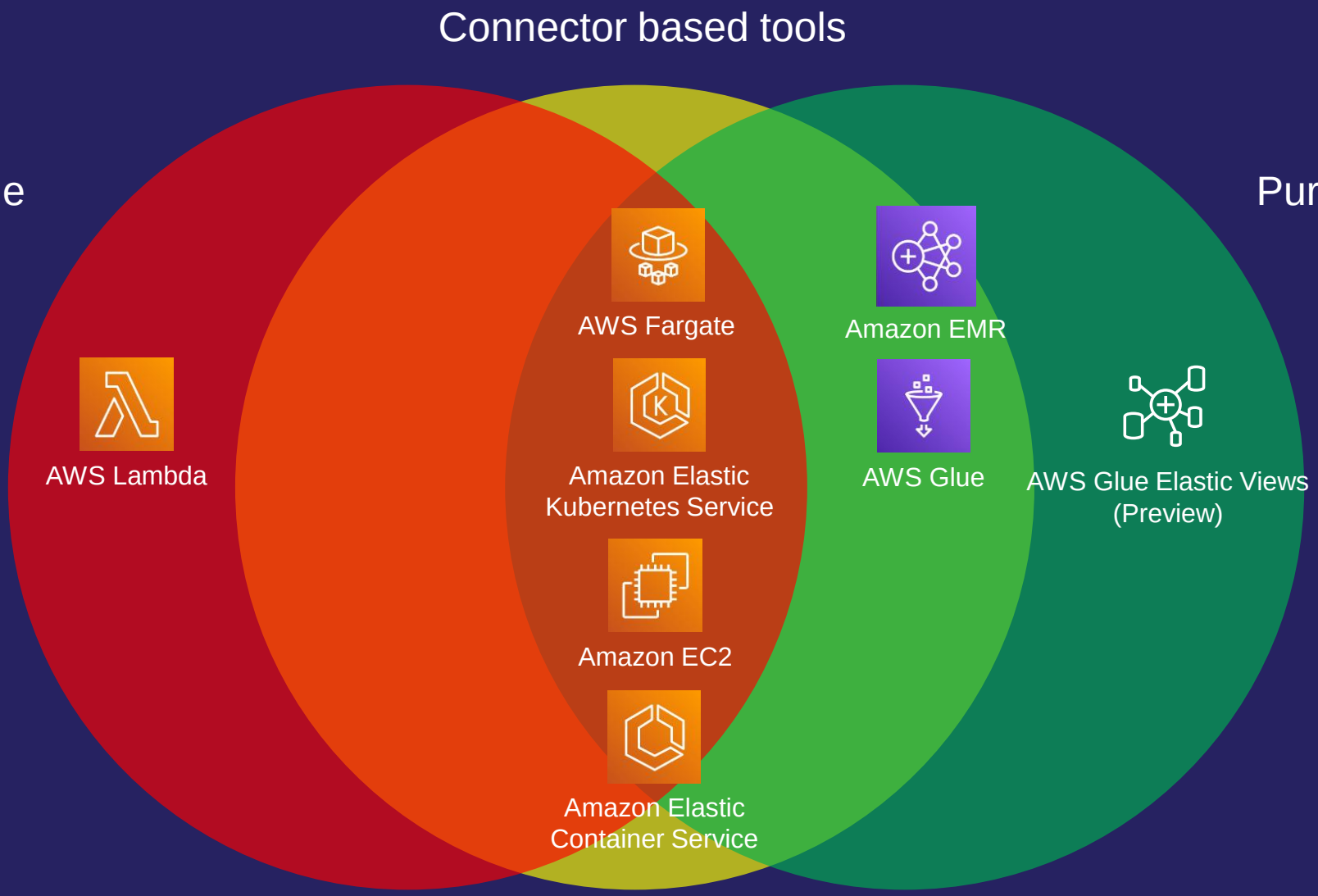
Effort - Low

- Configuration driven
- Provides out of the box integration to multiple sources
- Time is spent in creating high level data movement scripts
- Can be a managed service

```
CREATE VIEW hi_line_items AS
SELECT
  o.order_id, l.sku, l.qty, l.price,
  l.price * 0.045 AS tax
FROM
  orders AS o,
  o.line_items AS l
WHERE o.city = 'Honolulu'
```

AWS Glue Elastic Views (Preview)

Migration tooling – common AWS services used

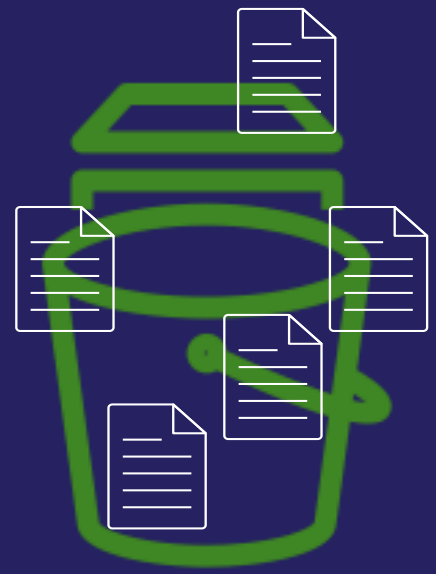


Building from snapshot

Building from snapshot

Effort - Low

- **Not supported for other search platforms**
  - Splunk
  - SOLR
- **Cannot put a newer version of the snapshot in an older version cluster**
  - Newer versions can typically read the last major version snapshot and those within the minor version less than or equal to current version
- **Typically should re-index the data once the snapshots are restored if earlier versions**



Building from existing engine

Building from existing engine

Effort - Medium

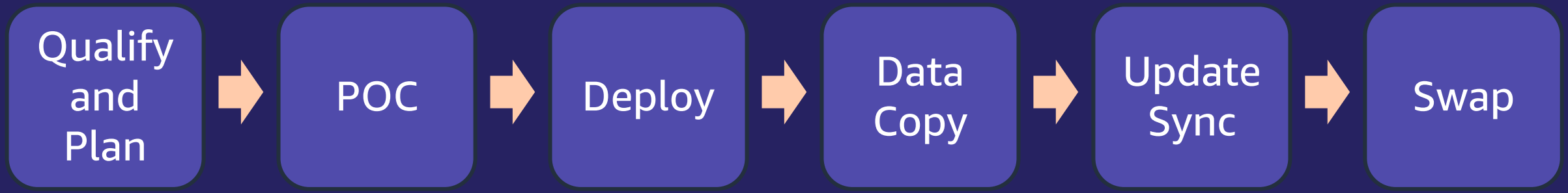
- **Reads directly from the source search engine and writes to the destination**
- **Typically is a tool you install on EC2 or containers**
    - Can be native like remote reindex
    - Data needs to be transformed
- **Typically plugin based such as the Logstash ecosystem**
- **Requires configuration and adjusting security settings**
- **Can involve complex networking and authorization frameworks**

Migration roadmap – update / sync

Update / sync

- **Prior to a cutover, you may choose to pause ingestion OR you may continue to use the older search engine until the first pass of data is migrated**
- **In either case you need to make the target current**
- **Log analytics**
    - Typically run side by side until retention period
    - Can repoint the ingestion pipeline let the new cluster be the primary
    - Sometimes backfills the prior data up to the cutover date
- **Search**
    - Ingestion is paused
    - CDC mechanisms backfill changes

Migration roadmap – swap to new engine



Swap to new engine

- **This is the final phase**
- **Ingestion is flowing normal and your clients now need to access the new data**
- **This can involve**
  - A new frontend application
  - DNS swaps from old to new via CNAME and Route 53 round robin weighting
- **The goal is that you have a fallback plan**
  - You will generally keep the old search solution around in case you need to roll back

Q&A

Thank you!