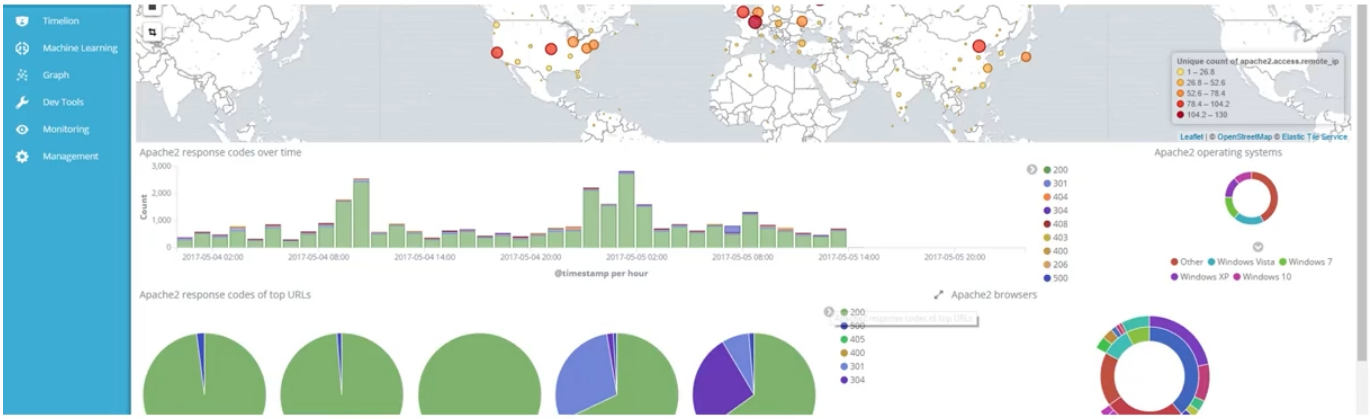
open Search (Elastic search)

basically most of the time we use log analysis

write source for query we use open search across a massive data sets

What are Dashboards?



Opensearch applications

- Full-text search
- Log analytics
- Application monitoring
- Security analytics
- Clickstream analytics

full text search is used for search engine the data

log analytics (hadoop is use to log analysis to detect error every minute 20trilo api log analysis analysis by elastic search)

application monitoring amazon use the price optim for all docker containers

security analytics

Opensearch concepts

documents

Documents are the things you're searching for. They can be more than text – any structured JSON data works. Every document has a unique ID, and a type.

types

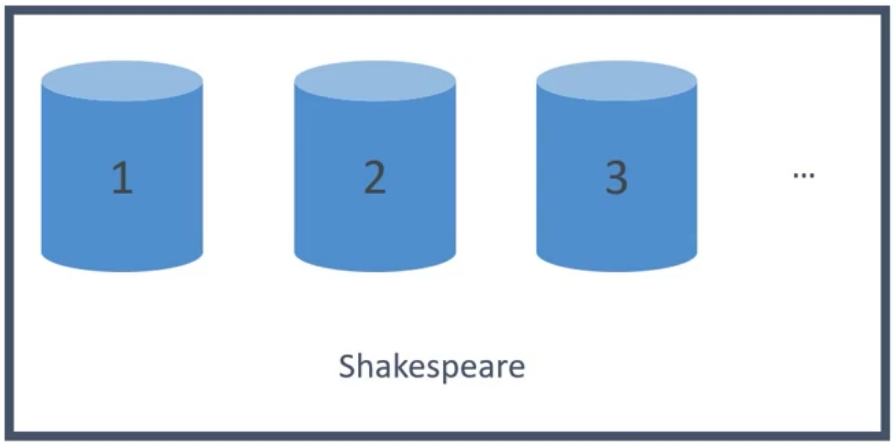
A type defines the schema and mapping shared by documents that represent the same sort of thing. (A log entry, an encyclopedia article, etc.)

indices

An index powers search into all documents within a collection of types. They contain inverted indices that let you search across everything within them at once.

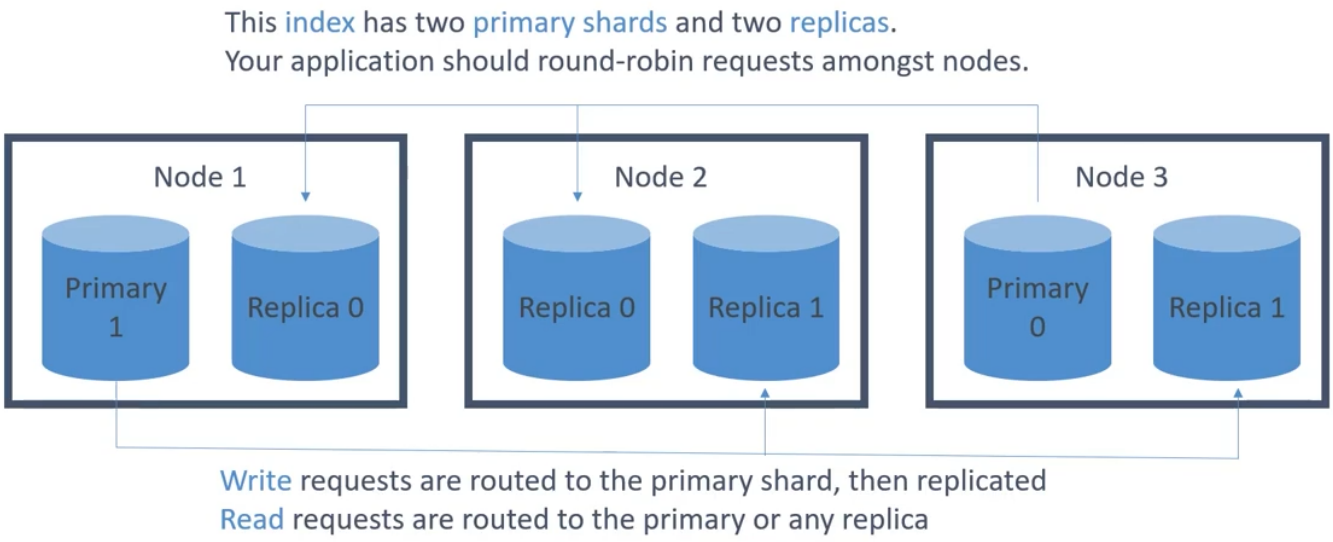
An index is split into shards

Documents are hashed to a particular shard.



Each shard may be on a different node in a cluster.
Every shard is a self-contained Lucene index of its own.

Redundancy



Amazon Opensearch Service

- Fully-managed (but not serverless)
- Scale up or down without downtime
  - But this isn't automatic
- Pay for what you use
  - Instance-hours, storage, data transfer
- Network isolation
- AWS integration
  - S3 buckets (via Lambda to Kinesis)
  - Kinesis Data Streams
  - DynamoDB Streams
  - CloudWatch / CloudTrail
  - Zone awareness

open search is not a server less you can run ec2 are emr aws give open search api and kibana dash board to view the data

Amazon Opensearch options

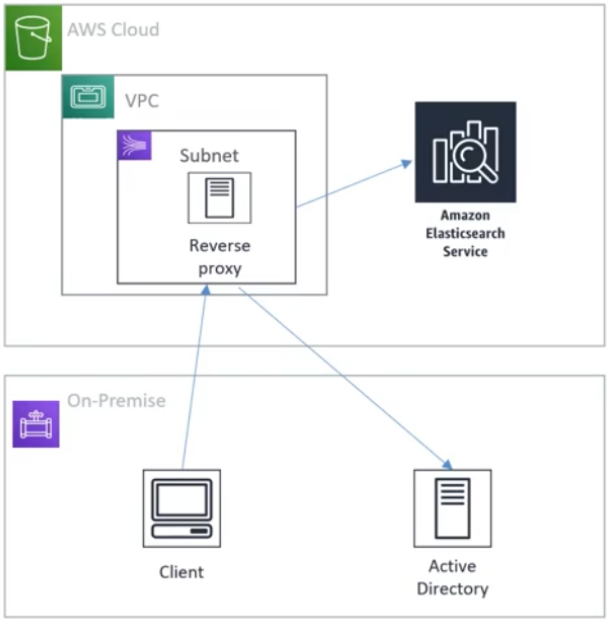
- Dedicated master node(s)
  - Choice of count and instance types
- “Domains”
- Snapshots to S3
- Zone Awareness

Amazon Opensearch Security

- Resource-based policies
- Identity-based policies
- IP-based policies
- Request signing
- VPC
- Cognito

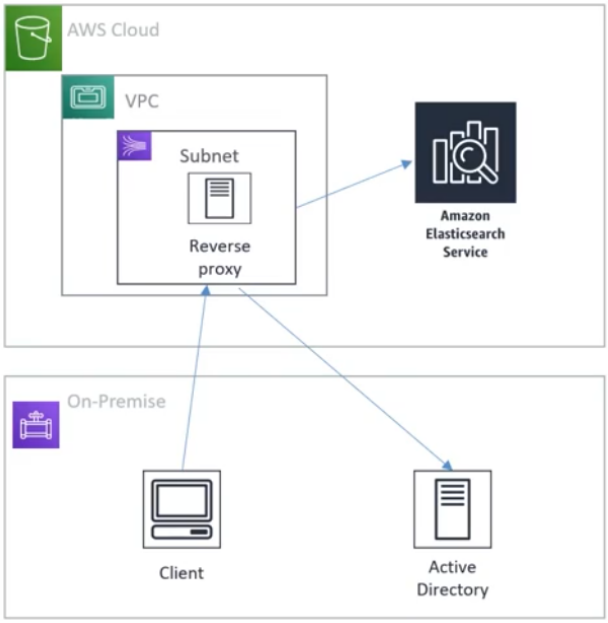
Securing Dashboards

- Cognito
- Getting inside a VPC from outside is hard…
  - Nginx reverse proxy on EC2 forwarding to ES domain
  - SSH tunnel for port 5601
  - VPC Direct Connect
  - VPN



Securing Dashboards

- Cognito
- Getting inside a VPC from outside is hard…
  - Nginx reverse proxy on EC2 forwarding to ES domain
  - SSH tunnel for port 5601
  - VPC Direct Connect
  - VPN



There three types of storage

Cold / warm / ultrawarm / hot storage

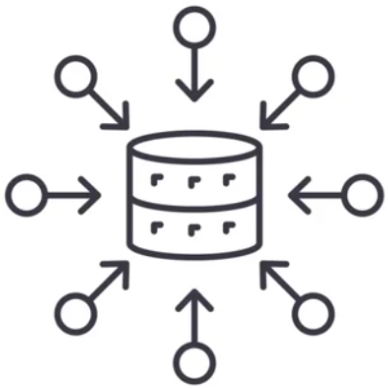
- Standard data nodes use “hot” storage
  - Instance stores or EBS volumes / fastest performance
- UltraWarm (warm) storage uses S3 + caching
  - Best for indices with few writes (like log data / immutable data)
  - Slower performance but much lower cost
  - Must have a dedicated master node
- Cold storage
  - Also uses S3
  - Even cheaper
  - For “periodic research or forensic analysis on older data”
  - Must have dedicated master and have UltraWarm enabled too.
  - Not compatible with T2 or T3 instance types on data nodes
  - If using fine-grained access control, must map users to cold_manager role in OpenSearch Dashboards

Index State Management

- Automates index management policies
- Examples
  - Delete old indices after a period of time
  - Move indices into read only state after a period of time
  - Move indices from hot -> UltraWarm -> cold storage over time
  - Reduce replica count over time
  - Automate index snapshots
- ISM policies are run every 30-48 minutes
  - Random jitter to ensure they don’t all run at once
- Can even send notifications when done

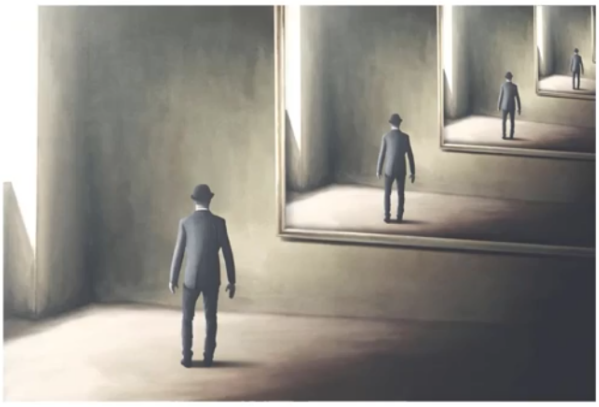
More Index Management

- Index rollups
  - Periodically roll up old data into summarized indices
  - Saves storage costs
  - New index may have fewer fields, coarser time buckets
- Index transforms
  - Like rollups, but purpose is to create a different view to analyze data differently.
  - Groupings and aggregations



Cross-cluster replication

- Replicate indices / mappings / metadata across domains
- Ensures high availability in an outage
- Replicate data geographically for better latency
- “Follower” index pulls data from “leader” index
- Requires fine-grained access control and node-to-node encryption
- “Remote Reindex” allows copying indices from one cluster to another on demand

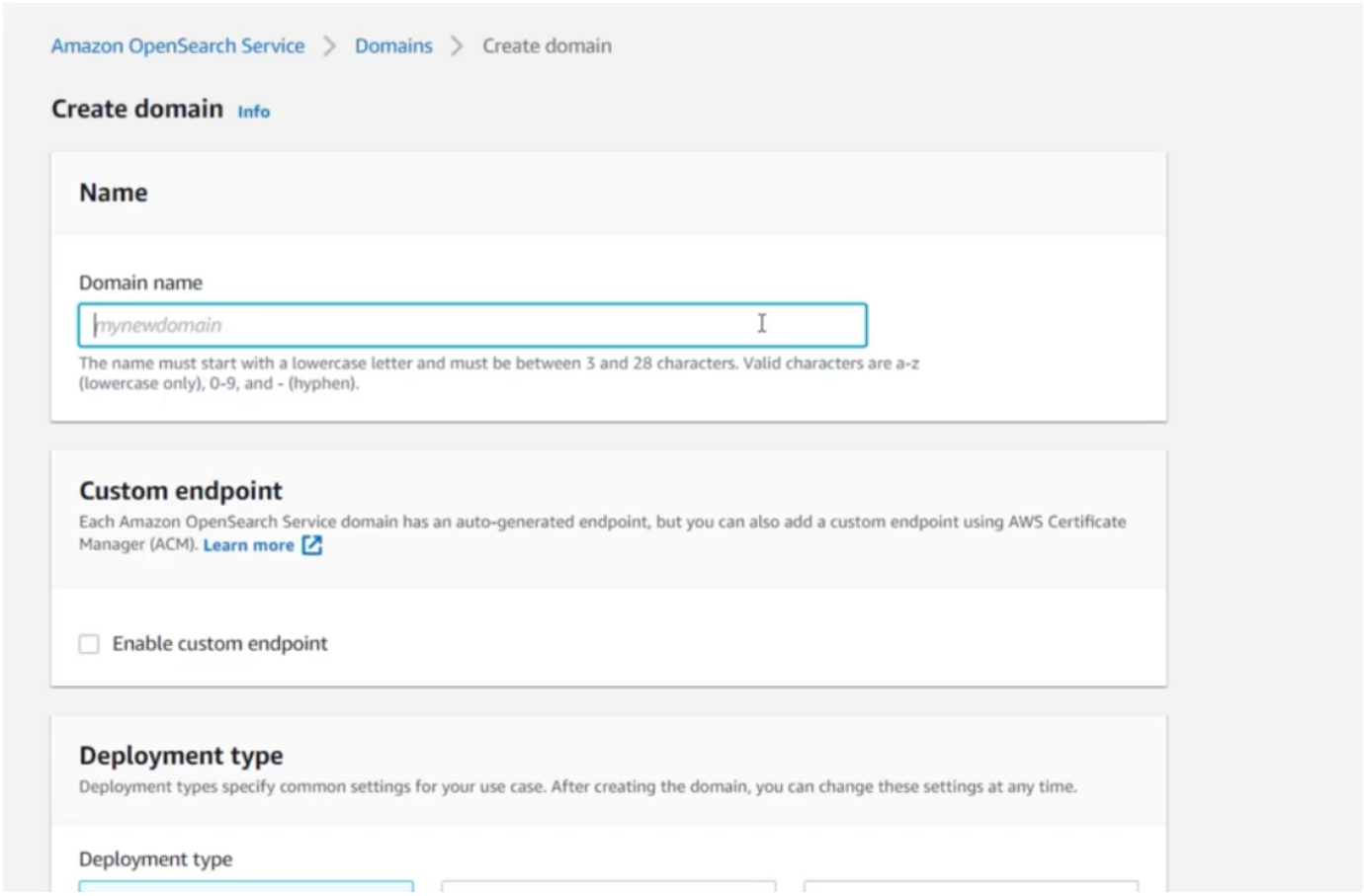


Opensearch Stability

- 3 dedicated master nodes is best
  - Avoids “split brain”
- Don’t run out of disk space
  - Minimum storage requirement is roughly: Source Data * (1 + Number of Replicas) * 1.45
- Choosing the number of shards
  - (source data + room to grow) * (1 + indexing overhead) / desired shard size
  - In rare cases you may need to limit the number of shards per node
    - You usually run out of disk space first.
- Choosing instance types
  - At least 3 nodes
  - Mostly about storage requirements
  - i.e., m6g.large.search, i3.4xlarge.search, i3.16xlarge.search

Amazon Opensearch performance

- Memory pressure in the JVM can result if:
  - You have unbalanced shard allocations across nodes
  - You have too many shards in a cluster
- Fewer shards can yield better performance if JVMMemoryPressure errors are encountered
  - Delete old or unused indices



Custom endpoint

Each Amazon OpenSearch Service domain has an auto-generated endpoint, but you can also add a custom endpoint using AWS Certificate Manager (ACM). Learn more

☐ Enable custom endpoint

Deployment type

Deployment types specify common settings for your use case. After creating the domain, you can change these settings at any time.

Deployment type

- ◉ **Production** — Multiple Availability Zones and dedicated master nodes for higher availability.
- ○ **Development and testing** — One Availability Zone for when you just need an OpenSearch endpoint.
- ○ **Custom** — Choose settings from all available options.

Version

1.0 (latest)

Certain features require specific OpenSearch/Elasticsearch versions. We recommend choosing the latest version.
Learn more

Auto-Tune analyzes cluster performance over time and suggests optimizations based on your workload. You can choose to deploy these changes or rollback to the default OpenSearch settings at any time.
Learn more

Auto-Tune

- ○ **Disable** — No automated changes to your cluster, Amazon OpenSearch Service will still send notifications on how to optimize cluster performance.
- ◉ **Enable** — Automatically makes node-level changes that require no downtime, such as tuning queues and cache sizes.

Maintenance window

☐ Add maintenance window
Some optimizations require blue/green deployments, which can impact cluster performance. Specify a low traffic time for Auto-Tune to start these deployments.

Data nodes

Select an instance type that corresponds to the compute, memory, and storage needs of your application. Consider the size of your indices, number of shards and replicas, type of queries, and volume of requests. Learn more

Instance type

r6g.large.search

r6g.large.search instance type needs EBS storage.

Number of nodes

Data nodes

Select an instance type that corresponds to the compute, memory, and storage needs of your application. Consider the size of your indices, number of shards and replicas, type of queries, and volume of requests. **Learn more**

Instance type

r6g.large.search

r6g.large.search instance type needs EBS storage.

Number of nodes

3

The number must be between 1 and 80.

Storage type

Choose a storage type for your data nodes.

EBS

EBS volume type

EBS volumes enable you to independently scale the storage resources of your domain from its compute resources. EBS volumes are most useful for domains with very large data sets, but without the need for large compute resources.

General Purpose (SSD)

EBS storage size per node

VPC access (recommended)

Public access

Fine-grained access control

Fine-grained access control provides numerous features to help you keep your data secure. Features include document-level security, field-level security, read-only users, and OpenSearch Dashboards/Kibana tenants. Fine-grained access control requires a master user. **Learn more**

Enable fine-grained access control

SAML authentication for OpenSearch Dashboards/Kibana

SAML authentication lets you use your existing identity provider for single sign-on for OpenSearch Dashboards/Kibana. **Learn more**

Prepare SAML authentication

To use SAML authentication, you must first enable fine-grained access control.

Amazon Cognito authentication

Enable to use Amazon Cognito authentication for OpenSearch Dashboards/Kibana. Amazon Cognito supports a variety of identity providers for username-password authentication. **Learn more**

No tags associated with this domain

Add new tag

You can add 50 more tags.

▼ Advanced cluster settings - *optional*

☑ Allow APIs that can span multiple indices and bypass index-specific access policies

Unchecking this option sets rest.action.multi.allow_explicit_index to false and breaks certain calls to the bulk, mget, and msearch APIs. If you need to manage access to those APIs, leave this option at its default value of true and enable fine-grained access control. Learn more

Fielddata cache allocation

Specifies the percentage of heap space that is allocated to fielddata. By default, this setting is 20.

20

The value must be either a number between 0 and 100.

Max clause count

Specifies the maximum number of allowed boolean clauses in a query. By default, this setting is 1024.

1024

The number must be between 1 and 2147483647.

Cancel Create

General information

Name
cadabra

Domain ARN
arn:aws:es:us-east-1:159107795666:domain/cadabra

Domain status
Active

Cluster health Info
Yellow

Version Info
OpenSearch 1.0 (latest)

Service software version Info
R20210816-P2 (latest)

OpenSearch Dashboards URL
https://search-cadabra-jkplghn4n5srwej7f37h3htuiu.us-east-1.es.amazonaws.com/_dashboards/

Domain endpoint
https://search-cadabra-jkplghn4n5srwej7f37h3htuiu.us-east-1.es.amazonaws.com

```
{
  "cloudwatch.emitMetrics": true,
  "kinesis.endpoint": "",
  "firehose.endpoint": "firehose.us-east-1.amazonaws.com",

  "flows": [
    {
        "filePattern": "/var/log/httpd/ssl_access*",
        "deliveryStream": "WebLogs",
        "initialPosition": "START_OF_FILE"
    }
  ]
}
```