Progress® MarkLogic®

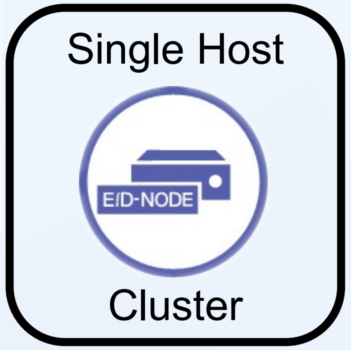
Architecture

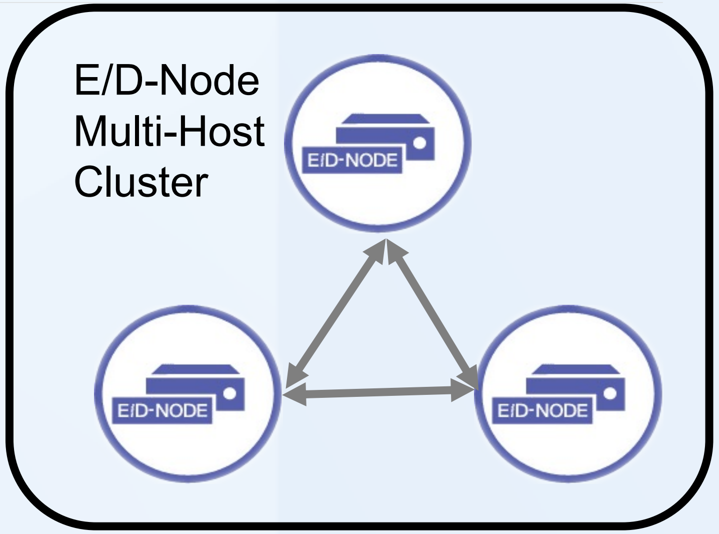
Key Concepts in this Unit

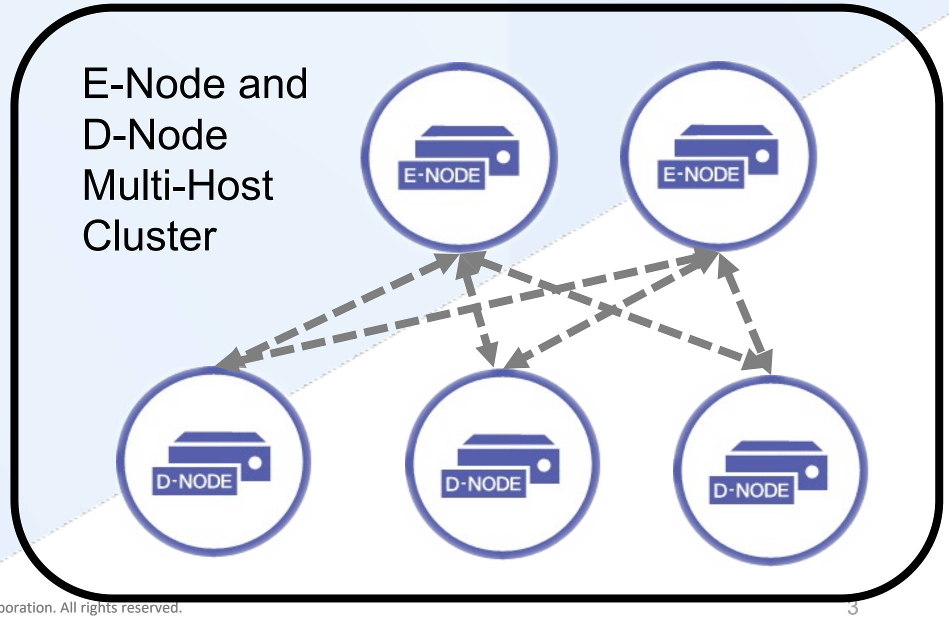
- What a MarkLogic Cluster is.
- How a database distributes documents on Hosts in a Cluster.
- How content is stored.
- Role of the Evaluator Node and Data Node.
- Communicating with a database.

MarkLogic Cluster

- Consists of 1 or more Hosts/Servers running a MarkLogic Server.
- Each Host can be a/an:
  - Evaluator Node (E-Node).
  - Data Node (D-Node).
  - Both E-Node and D-Node simultaneously.
- Mid-sized Cluster might contain:
  - 2 E-Nodes and 10 D-Nodes.
- Shared-nothing architecture - no single host in charge.

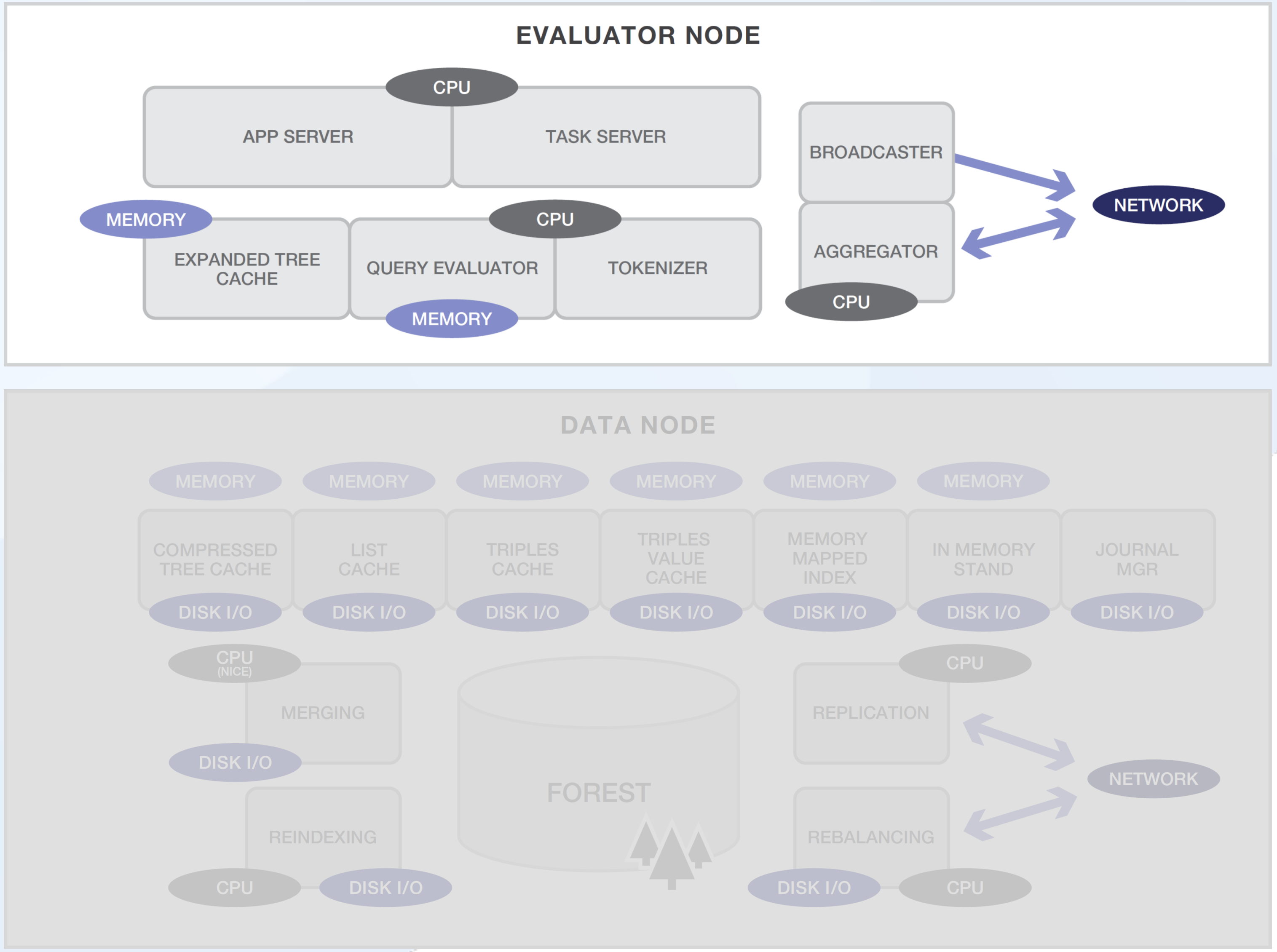






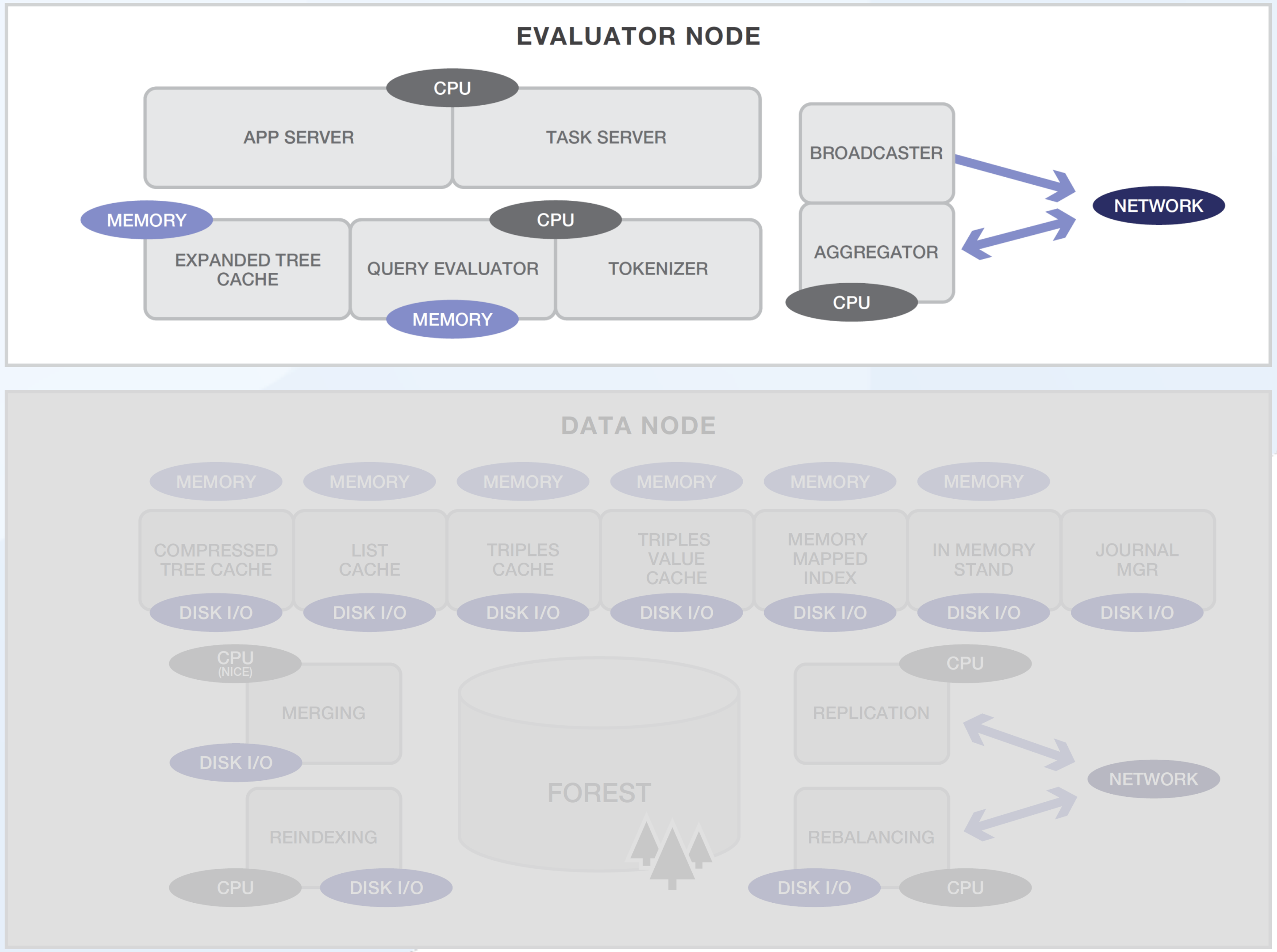
E-Node Tasks

- App Server listening on a specific port.
- Parses incoming requests/queries.
- Communicates with D-Nodes.
- Generates responses.
- Performs request/response transformations.
- Manages Host memory/cache.



E-Node Tasks

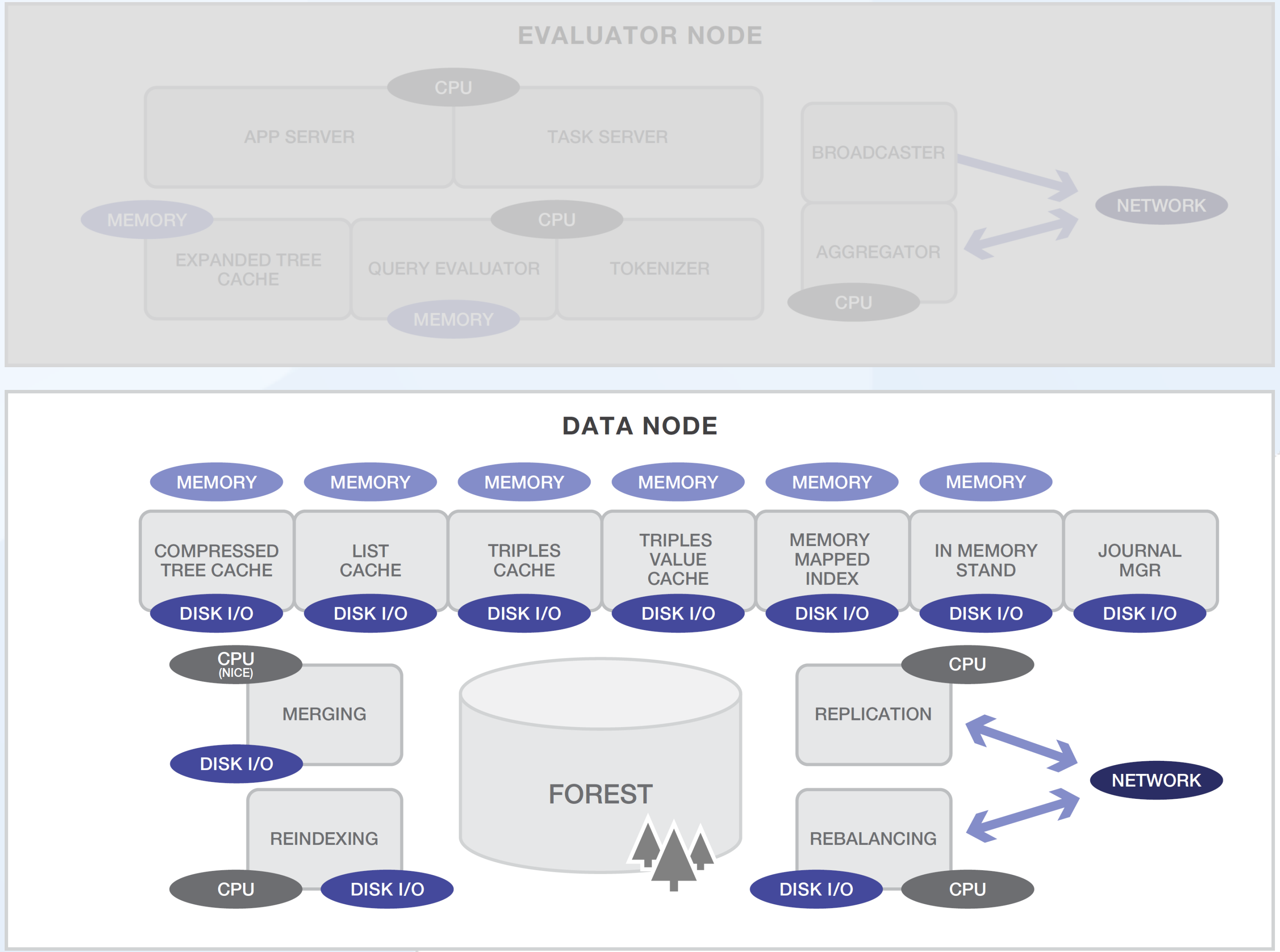
- Only one E-node is involved per transaction/request.
- The **ONLY** way to communicate with a MarkLogic database.
- Cache settings invest higher memory on “Expanded Tree Cache”.



Progress MarkLogic

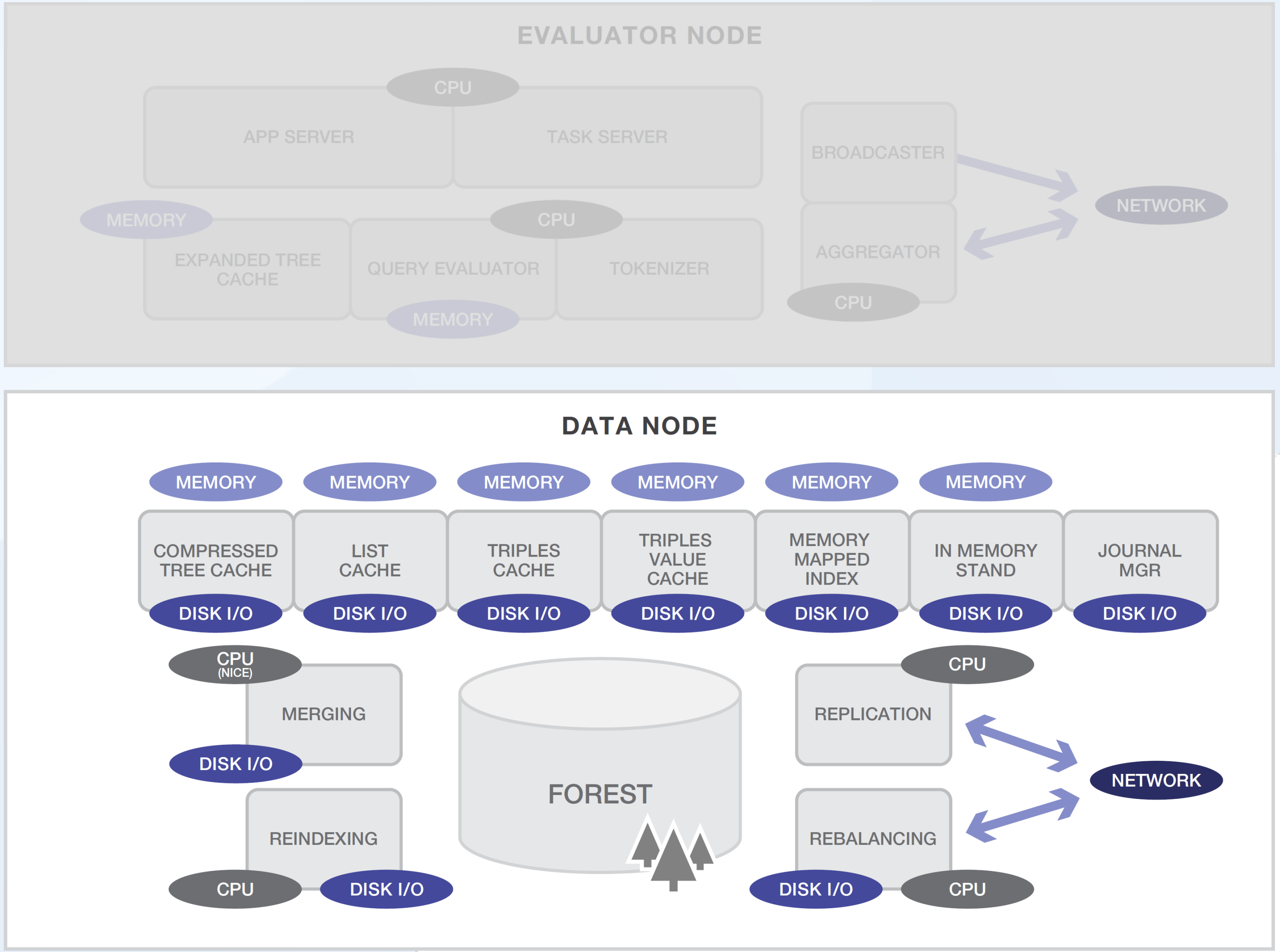
D-Node Tasks

- Securely stores data in Forests in a compressed, proprietary format.
- Creates and maintains Indexes and Word Lists.
- Provides data to E-Nodes.
- Manages Host memory/cache.
- Performs Merging, Reindexing, Replication and Rebalancing.
- Replies **ONLY** to E-Node requests.



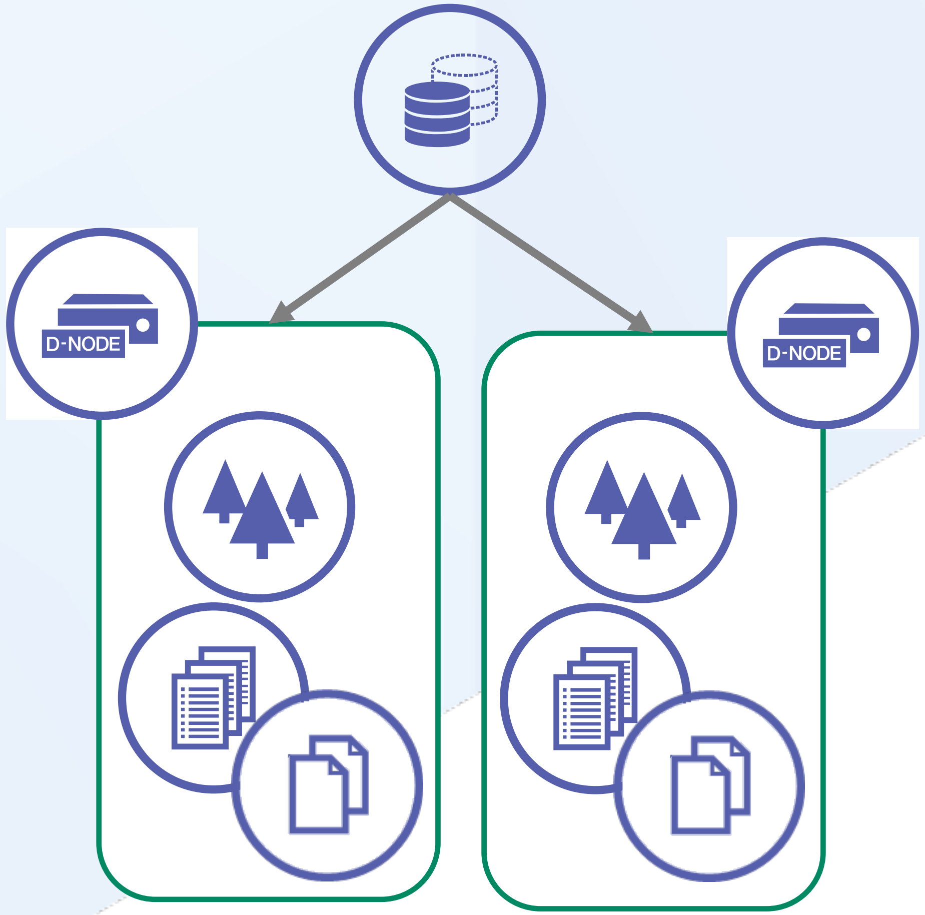
D-Node Tasks

- All D-Nodes are involved in processing a transaction.
- Cache settings invest higher memory on “List Cache” and “Compressed Tree Cache”.
- More information can be found in [MarkLogic Concepts Guide]().



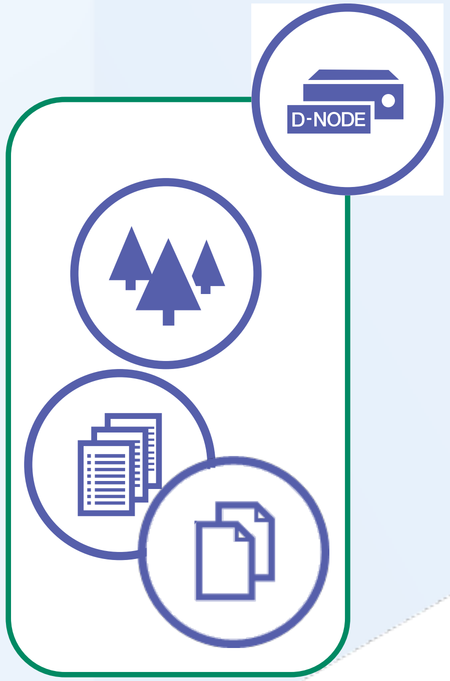
MarkLogic Database

- Exposed via an App Server.
- Is a set of configurations for:
  - What information should be tracked (Indexes/Word Lists).
  - How documents are to be distributed to forests (assignment policy).
  - Backup/Restore operations.
  - Connectivity to multiple forests (usually).



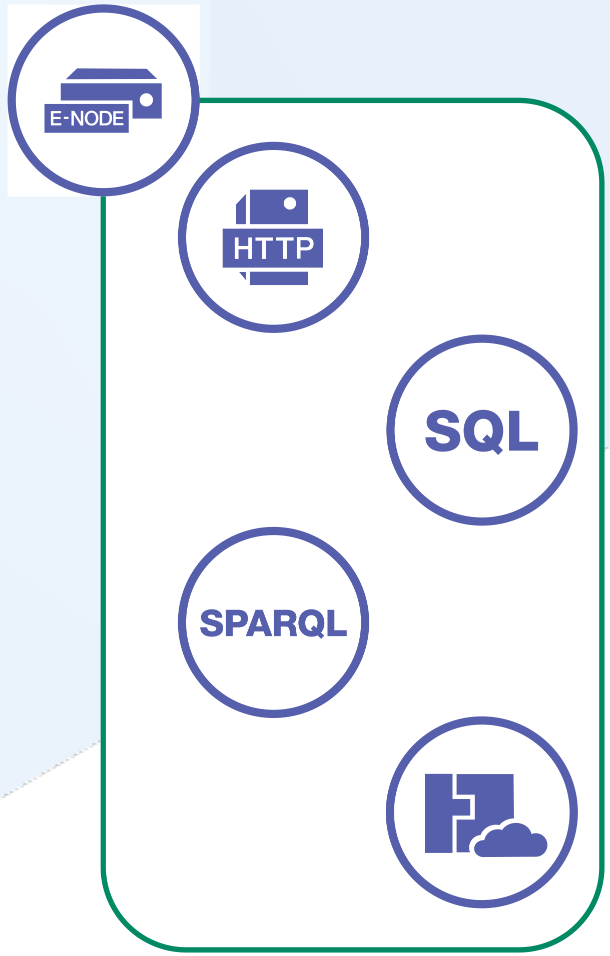
Forests

- Is the actual unit of storage for documents.
    - A host and location is specified.
    - A forest contains documents and their associated indexes/term lists.
    - A host can have multiple forests.
- A forest can be attached to **one and only one** database.
- A “partition” is a group of forests.
    - Forests in a partition can belong to different hosts.
- More information about forests is available in MarkLogic docs.



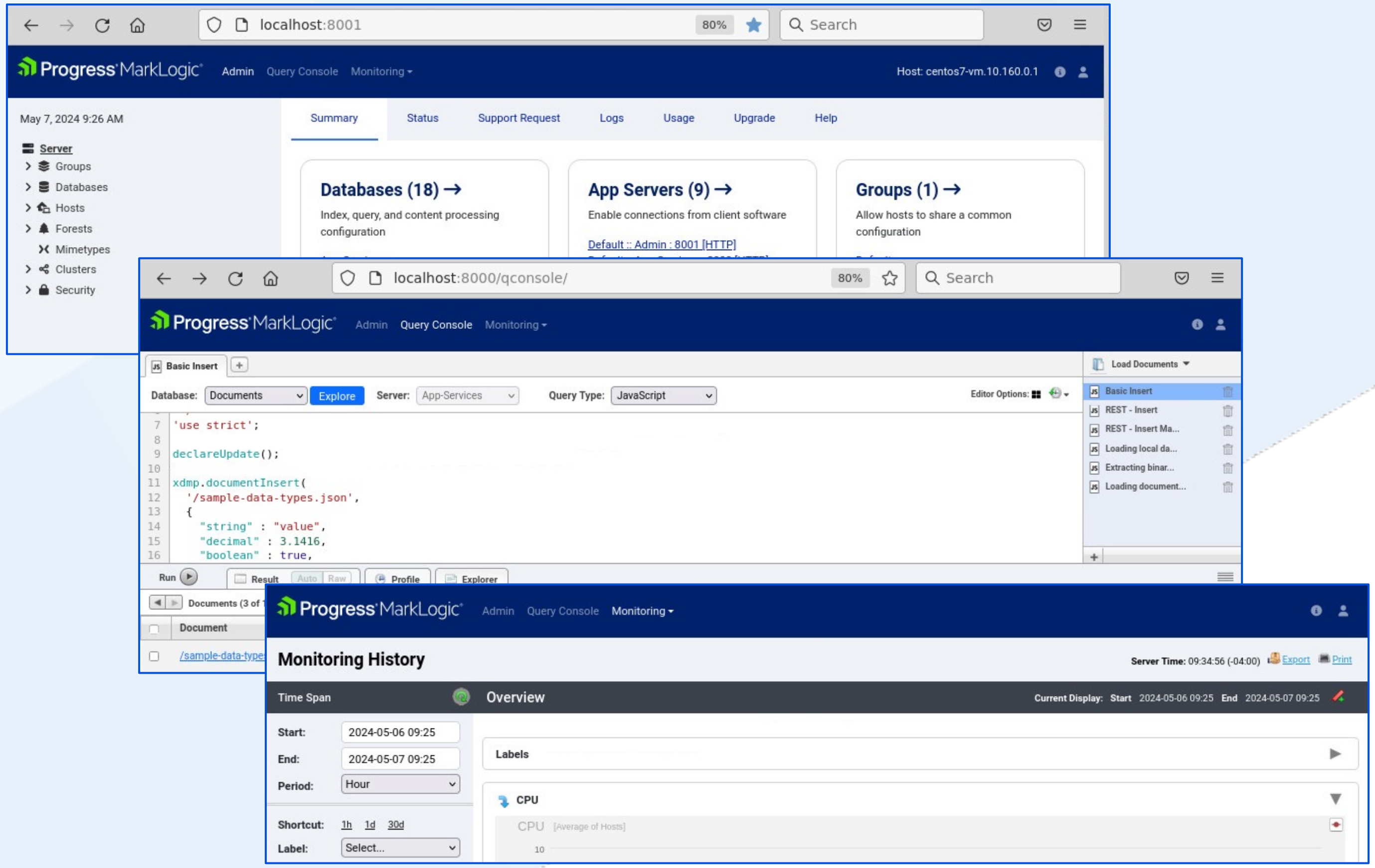
App Servers

- There are different types of App Servers for different purposes:
  - HTTP: Access modules via HTTP protocol, e.g. Admin UI (8001)
    - REST: A subtype of HTTP that makes use of special "url rewriters".
  - ODBC: Meant for your BI tools and ODBC connectors.
  - XDBC: Typically used by MLCP and CORB2.
  - WebDav: Use MarkLogic like a network drive.



HTTP App Server

- Accessed using HTTP.
- Runs modules from a specified location or a modules database.
- Built in App Servers:
  - Admin UI (8001)
  - Query Console (8000)
  - Monitoring (8002)

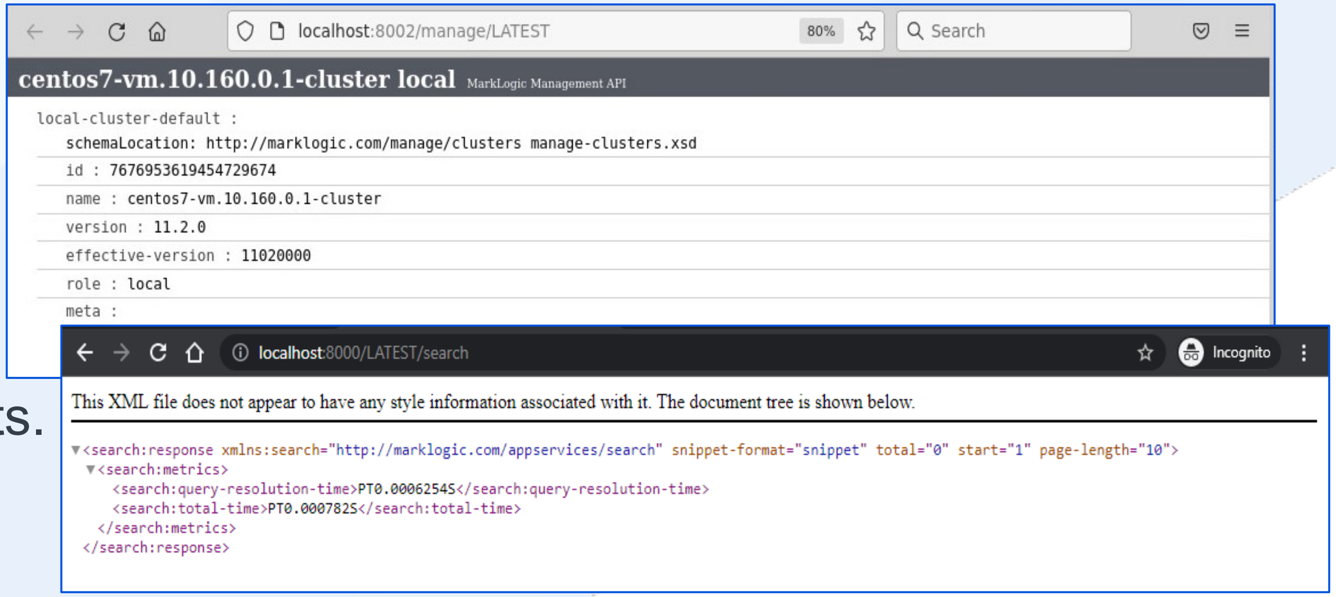


```
'use strict';

declareUpdate();

xdmp.documentInsert(
  '/sample-data-types.json',
  {
    "string" : "value",
    "decimal" : 3.1416,
    "boolean" : true,
```

Progress MarkLogic

REST API App Server

- Specialized HTTP App Server:
  - Uses special rewriters and error handlers:
    - `/MarkLogic/rest-api/rewriter.xml`
    - `/MarkLogic/rest-api/error-handler.xqy`
- Supports all Client API endpoints.
- Built in REST API App Servers:
  - App-Services (8000/LATEST)
  - Manage (8002/manage/LATEST)
    - Supports Management API endpoints.



centos7-vm.10.160.0.1-cluster local MarkLogic Management API

```
local-cluster-default :
    schemaLocation: http://marklogic.com/manage/clusters manage-clusters.xsd
    id : 7676953619454729674
    name : centos7-vm.10.160.0.1-cluster
    version : 11.2.0
    effective-version : 11020000
    role : local
    meta :
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<search:response xmlns:search="http://marklogic.com/appservices/search" snippet-format="snippet" total="0" start="1" page-length="10">
  ▼<search:metrics>
     <search:query-resolution-time>PT0.000625S</search:query-resolution-time>
     <search:total-time>PT0.000782S</search:total-time>
   </search:metrics>
 </search:response>
```

ODBC App Server

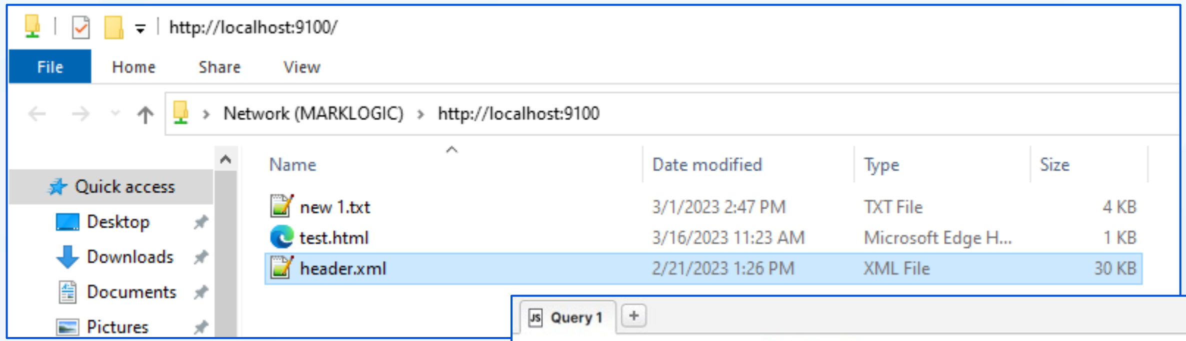
- Supports the connection of Business Intelligence (BI) tools to the MarkLogic Server.
    - Connectors for Power BI and Tableau.
- Can return relational-style data in response to SQL Queries.

XDBC App Server

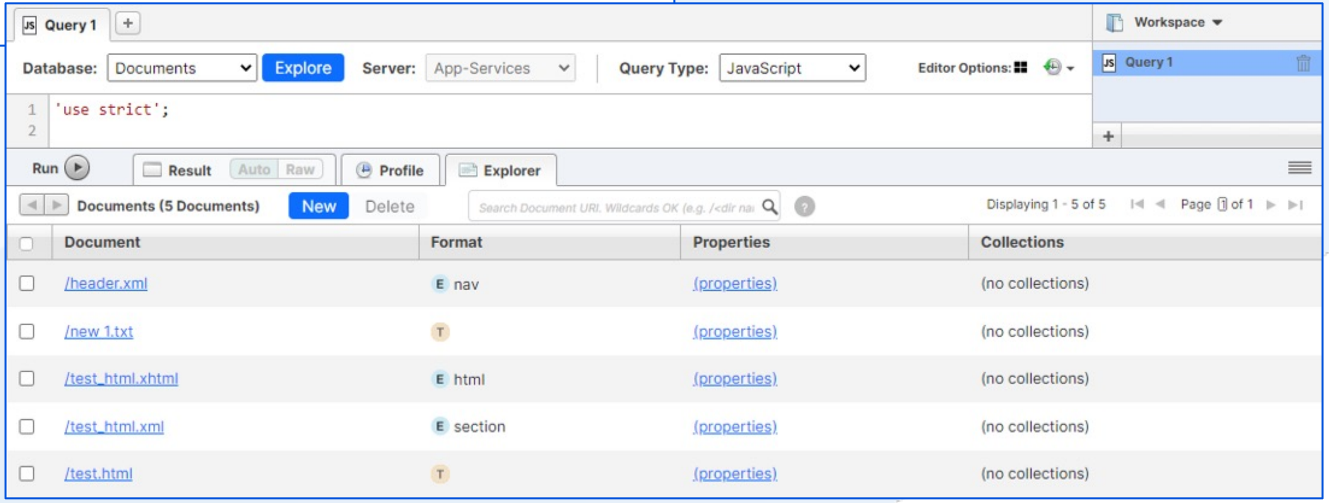
- Allows XML Contentbase Connector (XCC) applications to communicate with the MarkLogic Server.
- Used by CORB2, MLCP and similar apps.

WebDav App Server

- Read and write access a MarkLogic database from a folder.
- Load documents into a MarkLogic database by dragging and dropping into a local or network folder.



Name	Date modified	Type	Size
new 1.txt	3/1/2023 2:47 PM	TXT File	4 KB
test.html	3/16/2023 11:23 AM	Microsoft Edge H...	1 KB
header.xml	2/21/2023 1:26 PM	XML File	30 KB



Database: Documents   Server: App-Services   Query Type: JavaScript

```
'use strict';
```

Documents (5 Documents)

Document	Format	Properties	Collections
/header.xml	E nav	(properties)	(no collections)
/new 1.txt	T	(properties)	(no collections)
/test_html.xhtml	E html	(properties)	(no collections)
/test_html.xml	E section	(properties)	(no collections)
/test.html	T	(properties)	(no collections)

Progress MarkLogic

Labs:
Create a Forest and Database
Create an App Server

Progress® MarkLogic®

Configuration Deployment

Configuration Deployment Challenges

- In the previous labs the Admin UI was used for configuring:
  - a single app server, database and forest.
- Challenge:
  - Correct and Error free Deployment of:
    - This setup across 100 hosts in a single cluster.
    - Indexes, replication, sql views, across MarkLogic DEV, QA, UAT and Production clusters.
- Deployment Tool:
  - ml-gradle (recommended)

Deployment Tool: ml-gradle

- ml-gradle is:
  - MarkLogic’s Gradle plug-in that can quickly “stub out” a new project.
    - **`gradle mlNewProject`**
  - A tool to automate deployment across multiple environments.
  - A single command used to deploy all changes:
    - **`gradle mlDeploy`**

```
project
│    build.gradle
│    gradle-dev.properties
│    gradle-local.properties
│    gradle-prod.properties
│    gradle-qa.properties
│    gradle.properties
│
└───src
    └───main
        ├───ml-config
        │   │    rest-api.json
        │   │
        │   ├───databases
        │   │        content-database.json
   ...
        └───ml-modules
```

ml-gradle

- Supports the use of multiple environment configuration files:
  - argument to switch: `-**PenvironmentName=name**`
- `**gradle.properties**` is the base 'Properties' file.
- Use the `gradle-**<name>.**properties` to add or override properties for a specific target environment (Dev, QA, Prod).

```
project
    gradle-dev.properties
    gradle-local.properties
    gradle-prod.properties
    gradle-qa.properties
    gradle.properties
```

Progress® MarkLogic®

ml-gradle

- Encode project configuration (databases, indexes, app server, etc.) for version control.
    - This avoids as much “human error” as possible.
- Uses placeholders, `` `%%propertyName%%` ``, in these files to capture initialized values from `` `gradle.properties` `` during deployment.

```
project
└───src
    └───main
        └───ml-config
            │   rest-api.json
            │
            ├───databases
            │       content-database.json
            │       schema-database.json
            │
            └───servers
                    odbc.json
```

Progress® MarkLogic®

ml-gradle

- Used to develop your modules, configure your views with Template Driven Extraction (TDE) and deploy them across all environments.
- Less need to remember which Management REST API to use and how.
- More information about the project layout is available in the github wiki.

```
project
└───src
    └───main
        ├───ml-modules
        │   │   rest-properties.json
        │   │
        │   ├───options
        │   │       characters.json
        │   │       star-wars-options.xml
        │   │
        │   └───root
        │       └───mlcp
        │               extractMetadata.sjs
        │
        └───ml-schemas
            └───tde
                    characters.json
```

Progress® MarkLogic®

Progress MarkLogic

Lab: Use ml-gradle

Recap

- A host in a cluster with MarkLogic Server installed performs of E-node and/or D-node tasks.
- A “database”:
  - Is just a set of configuration parameters.
  - Can have many “forests”.
  - A “forest” can be attached to only one “database”.
- Documents are stored in “forests”.
- An “app server” communicates with a “database”.
- ml-gradle is the recommended deployment tool for MarkLogic implementations.