An overview of Sisense Linux microservices and what they do

This article provides an advanced description of Sisense's Microservices and how they work internally. This is mainly for Administrators responsible for supporting Sisense in their organization. When we deploy Sisense on Linux platform Sisense application deploys multiple pods this article helps understand how they are tied to each of the Sisense functionality and how they internally operate.

For a high-level overview, see Sisense Basic Concepts and Terminology and Sisense Architecture.

Sisense Deployments:

	Functionality	Details	Example
API-Gateway	Routes requests, manages ingress, and validates authentication and authorization.	API-Gateway provides an ingress controller to route requests to different services, where each Sisense service registers its paths on the API-Gateway. API-Gateway also validated that calls are authenticated and allows some authorization to be validated during the routing. The API-Gateway serves static browser client code which is composed from the multiple code components that are part of the Sisense Deployment. The API-Gateway Manages both HTTP and WS connections routes.	Directs a user's dashboard request to the appropriate service while ensuring the user is authenticated.
Oxygen	License management.	The service is responsible for license management, activation and deploymentID modules.Other services which require some license properties are able to request the active license through RPC.	Verifies if a user's environment has a valid license before allowing access to advanced features.
Exporting	Exports Sisense entities to various	Exporting-service responsible for	Creates a scheduled PDF export of

	formats and participates in reporting processes.	exporting of Sisense entities (dashboards and widgets) to different assets (PDF, PNG, PNG_BULK, XLSX). It is involved in the reporting process as well, when the .pdf or .png snapshots of the dashboard/widgets are exported internally and then sent over the email to the subscriber by the Galaxy service.	a sales dashboard and provides it to Galaxy to be sent via email.
External Plugins	Manages loading and execution of external plugins.	A service that is responsible for loading and executing all external plugins stored in the shared storage path `/opt/sisense/storage/external-plugins. All external plugins run and deploy as part of the external-plugins pod. Please note: external plugins and server-side plugins are not displayed in the Add-Ons section of the Admin tab, just the regular plugins are.	Loads a Report Manager plugin for enhanced reporting functionality.
Fluentd	Aggregates log data from Fluent-bit across nodes.	The main purpose of the fluentd service is to collect logging data from the fluent-bit pods on every node and write them as files to the logs folder (by default it is /var/log/sisense/{namespace} inside the fluentd pod).	Collects error logs for analysis to troubleshoot a dashboard loading issue.
Fluentbit	Collects logs from Kubernetes nodes.	Sisense uses Fluent-bit to collect logs from each Kubernetes node and pass them to a single fluentd process that aggregates / filters / augments and writes them to various destinations.	Gathers detailed logs from each node for performance monitoring.
Identity	Manages user authentication and roles	Identity Service is responsible for authenticating users directly and	Facilitates a user's login process and role assignment within the

		via SSO. Identity Service exposes API's for users, groups, authentication, roles (user management) and is heavily utilized by all Sisense services.	Sisense environment.
Configuration	Handles application configuration changes.	Configuration Service is responsible for Application configuration. Configurations can be changed via the Configuration page or through the SI cmd line. Configurations are stored in the Zookeeper. Configuration settings are exported as json files and backed up daily on Sisense Managed Cloud instances to the /opt/sisense/storage/configuration. You may also export Sisense configuration on demand following the instructions from this community post .	An admin changes a built timeout setting that affects build execution.
File Browser	Manages file interactions within Sisense.	It is used to manage files used by Sisense. Mainly responsible for the actions end users perform using the File Management tab.	A user uploads a csv file for use in a new elasticube.
Reporting	Manages report subscriptions and orchestrates reporting processes.	Reporting-service is responsible for managing dashboard reports subscriptions & orchestration of the reporting process. (Sisense includes an email reporting feature which consists of: • configuring default reporting schedule and settings by dashboard owner • configuring custom reporting schedule and settings by dashboard viewer/designer	Automates the distribution of weekly performance reports to a team.

		• execution of reporting process etc.)	
Query	Executes queries on data sources using JAQL. Translates JAQL to SQL in the New Analytical Engine.	Query service helps execute queries on data sources. The query input is JAQL. When using the New Analytical Engine JAQLs are translated to SQL in the query service which is targeted/optimized for specific data source type. Data sources can be either Live data sources or Elasticube.	Translates a complex widget query into a SQL query for a live data source.

Translation	Translates JAQL queries into SQL, particularly for the Old Analytical Engine.	The Translation Service is used in the Old Analytical Engine flow and is responsible for the following set of function:
		• Analytics:
		  • Translation of a JAQL query (a representation of an analytic query which underlies a widget in the dashboard) to a corresponding SQL query
		  • JAQL Formula validation
		  • Search for queryable dimensions (field-search)
		• Build support:
		  • Custom column/table validation
		  • Custom column/table conversion from
			Converts a custom user query into a SQL query for database retrieval.

		logical to physical SQL ● Custom column/table dependency detection ● SQL Support: ● Logical to physical SQL conversion ● Data Security: ● Application of the Data Security rules when running the queries for users.	
Knowledge Graph	Collects and stores query data for AI enhancements.	The main purpose of the Knowledge Graph service is to collect information about the queries run by the users and store this information to feed Sisense AI services. The knowledge-graph service collects information about queries, transforms the data, and writes the transformed data to a Dgraph.	Analyzes user query patterns to provide personalized dashboard recommendations.
Jobs	Schedules and manages long-term tasks.	Jobs service is responsible for scheduling repeating or long term tasks. Jobs service creates CRON job tasks inside and notify you when it is time to execute them. Under the hood there is a backup mechanism to restore/reinitialize all CRON jobs after system restart or fail via storing all needed information in MongoDb collection. So be sure that after you create a job item for Jobs service it will work until you remove it explicitly.	Sets up a recurring dashboard reporting job every night.
Build	Coordinates the process of	Build service is responsible to	Manages a data build process

	integrating data into the Data Warehouse called Elasticube.	bring data into the Data Warehouse - EC or External Data Warehouse such as Redshift and Snowflake — the process called 'Build'. Build Service is a strategic orchestrator of the Build process that coordinates several services to fetch data from the datasource, transform data, write it to the Data Warehouse and store the results externally (on s3 for instance) if needed.	aggregating information from various sources.
Ec-bld	A MonetDB instance created during the Elasticube build process.	Ec-bld service is an instance of the MonetDB database that is created during the Elasticube build process to bring data in the Sisense Data Warehouse called elasticube. Build service is managing the ec-bld service.	Temporarily holds data during the Elasticube build process for transformation.
Grafana	Visualizes cluster and Kubernetes metrics.	Grafana is an open source visualization and analytics software. It allows you to query, visualize, alert on, and explore aggregated metrics. it provides you with tools to turn your time-series database (TSDB) data into graphs and visualizations. In Sisense we use Grafana to visualize cluster/kubernetes metrics. The TSDB in Sisense deployment is maintained by Prometheus pod.A good overview article regarding the combination of Prometheus and Grafana can be found here. Every Sisense cluster (that includes one or more instances of Sisense application) have two levels of Grafana services.	Shows real-time analytics of system performance and resource usage by Sisense services.

Usage	Collect & save the usage data.	Responsible for collecting and saving off Sisense usage data. Tracking various activities.	contains various information users,groups, build failures, dashboards etc.
Plugins	Manages Sisense Add-ons.	Plugin service is responsible for managing Sisense customer-facing plugins (Add-ons). Plugins are used to augment or modify Sisense out-of-the-box functionality.	Installs and manages a new chart type plugin, allowing users to create more visually appealing dashboards
Pivot 2 be	Responsible for aggregating data from the Query service into Pivot structures.	Pivot2-be service responsible for aggregating data from Query service into a Pivot structure. This service is in fact taking care of the pivot widget.	Transforms raw data from a sales database into a structured format suitable for a pivot table in a dashboard.
Galaxy	Manages dashboard and widget rendering, and report distribution via emails.	Manages anything related to dashboards and widgets rendering in the UI as well as the reports distribution through emails.	Renders a complex dashboard and distributes it as an email report.
Intelligence	Part of Sisense AI Feature	Intelligence service is a part of the Sisense AI package. It is responsible for suggesting measures at the time of widget design and also used behind exploration paths features	Suggests relevant metrics to a user creating a new financial dashboard.
AI-Integration	Handles Natural Language Query related APIs.	Another part of the Sisense AI package responsible for all the Natural language query related API's. Prepares data for the NLQ API's, extract formulas etc. Query Cube and live models to collect members for the NLQ model. Save history of NLQ usage - The history of inputs of users are saved in the local MongoDB instance for future autocomplete.	Interprets a user's natural language query and translates it into a data query.

Storage	Provides shared storage for Sisense services	Storage service allows Sisense services to use shared storage. It is used for example during the email reporting flow when the exporting service exports a .PDF snapshot of the dashboard, stores its copy on the local storage, and then Galaxy service takes this file and sends it as an attachment to the report via email.	Stores exported PDF reports before they are emailed by the Galaxy service.
Custom code	Enables the use of custom Python code within Sisense.	The Custom Code functionality enables you to use an integrated Jupyter Notebook to run Python code to transform and cleanse data inside of an ElastiCube as part of the build process. The detailed documentation may be found here.	A data analyst writes a custom Python script to cleanse and transform incoming data before it's loaded into an Elasticube.
Compute-service	Provides computational resources for Python/R, supporting various Sisense features.	Compute service is component which is used by multiple Sisense features that require Python/R compute engine: • Notebooks - interactive python code execution • Spearfish/Swordfish - execution of python code during query/build flows respectively Compute service creates Pods/Services inside Kubernetes cluster, according to provided parameters, like compute type or image to run, resource allocation, additional configuration to apply on the compute instance, etc.	Powers the execution of a complex Python script in a Jupyter Notebook integrated within Sisense for data transformation.

Model-graphql	Manages operations on data models, primarily interacting with the MongoDB database for data-model-related requests.	Responsible for loading the Data page & for all the operations on data models. Sisense data models metadata is stored in the application database MongoDB the primary purpose of the model-graphql service is to query MongoDB for the data-model-related requests.	Retrieves and updates data model metadata for a newly created dashboard, ensuring the dashboard reflects the latest data structure.
Management	Oversees the management of elasticube pods, backups, restores, and task scheduling.	Management manages elasticubes pods, backups and restores. It creates ec-bld pods for the Build service during the build process for example. Scheduler logical service is hosted inside Management physical host. Scheduler service allows to schedule tasks which trigger rest requests to different services. Scheduler service uses Kubernetes Cron Jobs under the hood to schedule tasks.	Automates the backup of an Elasticube, ensuring data is securely stored and can be restored in case of a system failure. Creates ec-bld pods during the Build process.
Warehouse		This pod is used for the operations of SIsense Notebook Functionality, Notebooks	

Sisense Stateful Sets:

	Functionality	**Details**	**Example**
Dgraph	A graph database for storing and retrieving recommendation graphs.	Dgraph is a horizontally scale-able and distributed graph database, providing ACID transactions and consistent replication. Dgraph supports GraphQL-like query syntax, and responds in JSON and Protocol Buffers over GRPC and	Offers autocomplete suggestions for column names during dashboard creation.

		HTTP. This is where Sisense recommendations graphs are stored and retrieved from https://docs.dgraph.io/ it creates its own database, to allow recommendations. So when users are typing in a column name e.g. it completes it for us, hence recommending options. It also helps with search for dashboard names. With dgraph pod down, you might not see search results.	
Mongodb	Serves as the application database for metadata storage.	Sisense uses MongoDB as it's application database to store all Sisense metadata (dashboards, widgets, users, elasticubes, connections, data security etc.). Note: MongoDB doesn't contain actual analytics data, just metadata.	Stores user profiles, data models data security rules and dashboard configurations.
Zookeeper	Manages service discovery and configuration storage.	Zookeeper is used to store Sisense configuration and internal API endpoints of Sisense services (discovery).	Maintains a registry of Sisense configuration settings as well as microservice endpoints for efficient service communication.
RabbitMq	Handles inter-service communication and queuing mechanisms.	RabbitMQ is one of the components handling the inter-services communication system. It handles backend queuing mechanisms and is used mainly for internal requests involving queues (queries, builds etc). Direct inter-services communication not requiring any queues is handled directly via HTTP RPC requests using the Event-Driven Middleware (there is no separate pod for it, it is used as a separate node library).	Manages a queue of data processing requests for sequential execution.