Key Features In-Depth Breakdown

- Time-Series Logs Support
    - Ability to ingest ANY log into IVrixDB
    - No need to define a schema. Timestamp is extracted out of the logs, and the logs are stored as is.
    - Time Based retention (HOT, WARM, and COLD bucket tiers)
        - Recent data has a higher possibility of being read; the more recent the data, the higher data read possibility. Taking monitoring data as an example, we usually only care about what happened in the last few hours or the last few days, rather than what happened a month or a year ago.
    - Handling of late arrivals (old logs or delayed logs)
- Search-Time field Extraction
    - Ability to extract fields or remove fields post-indexing, effectively allowing the user to change the schema on the fly.
    - Makes the index schema-less, and drastically aids the user with the process of indexing and searching logs
        - Removes the necessity to pre-plan a schema before indexing (this saves a lot of time and effort for the user)
        - Removes the necessity to re-index data if schema needs to change (this also saves a lot of time and effort for the user)
        - Removes the headache of creating a common schema for an index with multiple log types
        - Allows the user to extract new information that a pre-defined schema simply cannot provide
- Easy Indexing
    - All nodes can accept any log format
    - Smooth, continuous, highly concurrent, and high throughput data writing (utilizing Solr's internal indexing mechanisms)
    - Built-in rollover of data based on capacity and timespans to gain predictability of index and search times
- Data Query and Analysis
    - HOT, WARM, and COLD tiers based on use, keeping the most recent, and most used data available for sub-second response.
    - Clear and easy API, best of breed choice (Elastic, Splunk, etc.)
        - Facets, Dynamic Timeline, and paged raw data is available and can be polled without any user interaction.
        - Incremental updates of results as the search runs for interactive response to the client.
    - Searching by time range and IVrixDB index, not Solr collection.
    - Using all available Solr streaming API functions to do analytics and data permutation.
    - Using the Lucene index to do full-text search and push-down filtering (and maybe analytics in the future).
- Highly scalable and fault tolerant

    - Automatic management of HW resources
    - Automatic Cold Tier for off-loading least recent use data
    - Handling of accidents, such as node death
- Simple operations
    - Easy creation and deletion of IVrixDB “Index”, a namespace that can be thought of as a collection of Solr collections.
    - No need to define collections and spanning nodes
    - Each node is independent for reliability, relying only on local storage
    - Spinning up a new node to add more index and/or search power w/o any configuration or dependency.