**Using NiFi to Create a Pipeline in Redis**

NiFi and Redis were started.

A screenshot of a computer

Description automatically generated

The NiFi UI was opened by navigating to <https://localhost:8080/nifi>.

A screenshot of a computer

Description automatically generated

In the NiFi Flow process group, the gear icon was selected to configure the NiFi Flow process group. In the CONTROLLER SERVICES tab, the plus sign was selected and RedisConnectionPoolService was chosen as the type. In the PROPERTIES tab, the Connection String property was set to localhost:6379. No other default values were changed.

A screenshot of a computer

Description automatically generated

Another controller named RedisDistributedMapCacheClientService was added inside the NiFi Flow process group. In the PROPERTIES tab, the previously created RedisConnectionPoolService controller was selected.

A screenshot of a computer

Description automatically generated

Both the RedisConnectionPoolService and RedisDistributedMapCacheClientService controllers were enabled.

A screenshot of a computer

Description automatically generated

A processor titled GenerateFlowFile was created. In the PROPERTIES tab, the Custom Text property was set to ${now()}.

A screenshot of a computer

Description automatically generated

In the SCHEDULING tab, the Run Schedule was changed to five seconds.

A screenshot of a computer

Description automatically generated

A processor titled PutDistributedMapCache was created. In the PROPERTIES tab, the Distributed Cache Service was set equal to Redis DMC Client Service, and the Cache Entry Identifier was set equal to date.

A screenshot of a computer

Description automatically generated

A processor titled FetchDistributedMapCache was created. In the PROPERTIES tab, the Distributed Cache Service was set equal to Redis DMC Client Service, the Cache Entry Identifier was set to date, and the Put Cache Value in Attribute was set to date.retrieved.

A screenshot of a computer

Description automatically generated

A processor titled LogAttribute was created. A screenshot was provided showing all the controllers, GenerateFlowFile, PutDistributedMapCache, FetchDistributedMapCache, and LogAttribute in NiFi.

A screenshot of a computer

Description automatically generated

All processors were connected in the order they were added using success as the relationship type. The following settings were ensured:

* In the PutDistributedMapCache processor settings, Relationships were set to Automatically Terminate on failure.
* In the FetchDistributedMapCache processor settings, Relationships were set to Automatically Terminate on failure and not-found.
* In the LogAttribute processor settings, Relationships were set to Automatically Terminate on success.

A screenshot of a computer

Description automatically generated

In the Terminal window, the nifi-app.log file was opened inside the libexec folder's logs folder using the nano text editor.

A screen shot of a computer screen

Description automatically generated