

git_comments:

1. `** Start function instance ** @param tenant * Tenant name * @param namespace * Namespace name * @param function * Function name ** @param instanceId * Function instanceId ** @throws PulsarAdminException * Unexpected error`
2. `** Start all function instances ** @param tenant * Tenant name * @param namespace * Namespace name * @param function * Function name ** @throws PulsarAdminException * Unexpected error`
3. `** Start sink instance ** @param tenant * Tenant name * @param namespace * Namespace name * @param sink * Sink name ** @param instanceId * Sink instanceId ** @throws PulsarAdminException * Unexpected error`
4. `** Start all sink instances ** @param tenant * Tenant name * @param namespace * Namespace name * @param sink * Sink name ** @throws PulsarAdminException * Unexpected error`
5. `** Start source instance ** @param tenant * Tenant name * @param namespace * Namespace name * @param source * Source name ** @param instanceId * Source instanceId ** @throws PulsarAdminException * Unexpected error`
6. `** Start all source instances ** @param tenant * Tenant name * @param namespace * Namespace name * @param source * Source name ** @throws PulsarAdminException * Unexpected error`
7. This means that all instances of the functions are running
8. `** Sends a start/stop function request to the FMT (Function Metadata Topic) for a function * @param tenant the tenant the function that needs to be deregistered belongs to * @param namespace the namespace the function that needs to be deregistered belongs to * @param functionName the name of the function * @param instanceId the instanceId of the function, -1 if for all instances * @param start do we need to start or stop * @return a completable future of when the start/stop has been applied`
9. want to change state for all instances
10. for externally managed functions, insert the start only if there is atleast one instance that needs to be started
11. validate parameters
12. add a stop
13. make sure terminate is not called since this is a update operation

git_commits:

1. **summary:** Fixed the behavior of Function start/stop (#3477)
message: Fixed the behavior of Function start/stop (#3477) * Added a state in the function metadata about what the state of the instances should be * Have start api for sources/sinks * Add missing pieces * more checks while handling request * Fixed bugs * Added unittests * Added unittest * Fix the all instances side logic

github_issues:**github_issues_comments:****github_pulls:**

1. **title:** Fixed the behavior of Function start/stop
body: `### Motivation` The current behavior of function/source/sink start/stop functionality is really confusing. We just change the state of the function temporarily, i.e. we don't record the change of the functionstate anywhere. Thus if function workers die and come back up, they will restart stopped function instances. This pr fixes this behaviour. We record the state of each function instance in the function metadata, thus if function worker dies and comes back up, the function will still remain in the state that it was in before the death. `### Modifications` Add a field in the FunctionMetaProto to reflect the current state of the function instances. `### Verifying this change - []` Make sure that the change passes the CI checks. `*(Please pick either of the following options)*` This change is a trivial rework / code cleanup without any test coverage. `*(or)*` This change is already covered by existing tests, such as `*(please describe tests)*`. `*(or)*` This change added tests and can be verified as follows: `*(example:)*` - `*Added integration tests for end-to-end deployment with large payloads (10MB)*` - `*Extended integration test for recovery after broker failure*` `#### Does this pull request potentially affect one of the following parts: *If `yes` was chosen, please highlight the changes*` - Dependencies (does it add or upgrade a

dependency): (yes / no) - The public API: (yes / no) - The schema: (yes / no / don't know) - The default values of configurations: (yes / no) - The wire protocol: (yes / no) - The rest endpoints: (yes / no) - The admin cli options: (yes / no) - Anything that affects deployment: (yes / no / don't know) ###
Documentation - Does this pull request introduce a new feature? (yes / no) - If yes, how is the feature documented? (not applicable / docs / JavaDocs / not documented) - If a feature is not applicable for documentation, explain why? - If a feature is not documented yet in this PR, please create a followup issue for adding the documentation

github_pulls_comments:

1. Mostly looks good. I think we can also modify the following logic to check the function instance state so that we don't have that race condition: <https://github.com/apache/pulsar/blob/master/pulsar-functions/worker/src/main/java/org/apache/pulsar/functions/worker/FunctionRuntimeManager.java#L647>
2. @jerrypeng I think that change can be done in a subsequent change since its kind of not related to this. I will merge this and work on that later.

github_pulls_reviews:

jira_issues:

jira_issues_comments: