CDNC-12163

Problem description

TL;DR: We have updated our Cassandra tables to store created_time and last_updated_time. But currently, we created those timestamps at a super low level, which is a bad pratice.

For example, we are storing a newly created workflow execution (in execution table): the requests go by this order: execution_manager -> nosql_exeuction_store -> db -> InsertWorkflowExecutionWithTasks -> insertOrUpsertWorkflowRequestRow.

- In insertOrUpsertWorkflowRequestRow, we talk to cassandra and finally store the execution.
- But we only created created_time at InsertWorkflowExecutionWithTasks, this is the best practice.

The high level idea of the refactor plan is to create the time stamps in the persistence managers. We can find data_stores and persistency_managers in the Appendix.

Proposed solution

Currently, timeSrc object is in the field of cbd as `timeSrc clock.TimeSource`

Approach 1

I propose to move this object into each persistence_managers (please refer to the persistence_manager col above).
For example:
Step 1:
Add fields to xxx_manager

```
C/C++
type (
	// executionManagerImpl implements ExecutionManager based on
ExecutionStore, statsComputer and PayloadSerializer
	executionManagerImpl struct {
		serializer    PayloadSerializer
		persistence   ExecutionStore
		statsComputer statsComputer
		logger        log.Logger
		timeSrc clock.TimeSource
```

```
        }
)

var _ ExecutionManager = (*executionManagerImpl)(nil)

// NewExecutionManagerImpl returns new ExecutionManager
func NewExecutionManagerImpl(
        persistence ExecutionStore,
        logger log.Logger,
        serializer PayloadSerializer,
) ExecutionManager {
        return &executionManagerImpl{
                serializer:    serializer,
                persistence:   persistence,
                statsComputer: statsComputer{},
                logger:        logger,
                timeSrc: clock.NewRealTimeSource(),
        }
}
```

Step 2:
And in order to not change the signature of method: `CreateWorkflowExecution`
We need to add fields created_time & last_updated_time in InternalCreateWorkflowExecutionRequest
Step 3:
Add created_time & last_updated_time in `nosqlplugin.WorkflowExecutionRequest`
And refactor the code in prepareCreateWorkflowExecutionRequestWithMaps to have those timestamps
Step 4:
In InsertWorkflowExecutionWithTasks, get timestamp by execution.CreatedTime etc.
Step 5:
Refactor the unit tests

Task break down:
I would suggest doing the execution table firstly and separately since it included so many changes. And we can also check to make sure we are in the right direction.

- Refactor execution manager and unit tests
- Refactor all other persistence managers and unit tests

Note:
All the time stamps that are needed to be refactors can be found in the Appendix.

Appendix

data_stores and persistence_managers

Table name	data_store	persistence_manager
executions	nosql_execution_store	execution_manager
history_node	nosql_history_store	history_manager
history_tree	nosql_history_store	history_manager
tasks	nosql_task_store	task_manager
domains	nosql_domain_store	domain_manager
domains_by_name_v2	nosql_domain_store	domain_manager
queue	nosql_queue_store	queue_manager
queue_metadata	nosql_queue_store	queue_manager
cluster_config	nosql_config_store	config_store_manager

Time stamps that need to be refactored into persistence layer:

Table_Name	**created_time**	**last_updated_time**
executions	YES	YES
history_node	YES	NO
history_tree	YES	NO
tasks	YES	YES
domains	YES	NO
domains_by_name_v2	YES	YES (But it is type bigint)
queue	YES	NO
queue_metadata	YES	YES
cluster_config	NO	NO