**Explain in brief with their uses.**

**1) Oozie Action and Decision Nodes**

**Action Nodes:**

All computation/processing tasks triggered by an action node are executed asynchronously by Oozie. For most types of computation/processing tasks triggered by workflow action, the workflow job has to wait until the computation/processing task completes before transitioning to the following node in the workflow.

The exception is the fs action that is handled as a synchronous action.

Oozie can detect completion of computation/processing tasks by two different means, callbacks and polling.

When a computation/processing tasks is started by Oozie, Oozie provides a unique callback URL to the task, the task should invoke the given URL to notify its completion.

For cases that the task failed to invoke the callback URL for any reason (i.e. a transient network failure) or when the type of task cannot invoke the callback URL upon completion, Oozie has a mechanism to poll computation/processing tasks for completion.

If a computation/processing task -triggered by a workflow- completes successfully, it transitions to ok .

If a computation/processing task -triggered by a workflow- fails to complete successfully, its transitions to error .

If a computation/processing task exits in error, there computation/processing task must provide error-code and error-message information to Oozie. This information can be used from decision nodes to implement a fine grain error handling at workflow application level.

Each action type must clearly define all the error codes it can produce

Oozie provides recovery capabilities when starting or ending actions.

**Decision Nodes:**

A decision node enables a workflow to make a selection on the execution path to follow.

The behavior of a decision node can be seen as a switch-case statement.

A decision node consists of a list of predicates-transition pairs plus a default transition.

Predicates are evaluated in order or appearance until one of them evaluates to true and the corresponding transition is taken.

If none of the predicates evaluates to true the default transition is taken.

Each case elements contains a predicate an a transition name.

The default element indicates the transition to take if none of the predicates evaluates to true .

All decision nodes must have a default element to avoid bringing the workflow into an error state if none of the predicates evaluates to true.

**2) Oozie Workflow Nodes**

**Control Flow nodes:**

Control flow nodes define the beginning and the end of a workflow (the start , end and kill nodes) and provide a mechanism to control the workflow execution path (the decision , fork and join nodes).

The start node is the entry point for a workflow job, it indicates the first workflow node the workflow job must transition to.

When a workflow is started, it automatically transitions to the node specified in the start .

A workflow definition must have one start node.

The end node is the end for a workflow job, it indicates that the workflow job has completed successfully.

When a workflow job reaches the end it finishes successfully (SUCCEEDED).

If one or more actions started by the workflow job are executing when the end node is reached, the actions will be killed. In this scenario the workflow job is still considered as successfully run.

A workflow definition must have one end node.

The kill node allows a workflow job to kill itself.

When a workflow job reaches the kill it finishes in error (KILLED).

If one or more actions started by the workflow job are executing when the kill node is reached, the actions will be killed.

A workflow definition may have zero or more kill nodes.

**Action Nodes:**

Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task.

All computation/processing tasks triggered by an action node are executed asynchronously by Oozie.

For most types of computation/processing tasks triggered by workflow action, the workflow job has to wait until the computation/processing task completes before transitioning to the following node in the workflow.

**3) Fork and Join**

A fork node splits one path of execution into multiple concurrent paths of execution.

A join node waits until every concurrent execution path of a previous fork node arrives to it.

The fork and join nodes must be used in pairs.

The join node assumes concurrent execution paths are children of the same fork node.

**4) Oozie Web Console**

Oozie web console is one which is used for monitoring the Status of the job, workflow applications status and workflow jobs status.

It is read only interface and we cannot create any workflow.

We can only read only.