**1. Moogsoft – CloudWatch Integration**

**Pre-requisites:**

* AWS account access with permissions to CloudWatch, Lambda, and IAM.
* Moogsoft AIOps or Moogsoft Cloud credentials with access to integrations panel.
* ITSM tool API credentials (e.g., ServiceNow).

**Steps:**

1. **Deploy Lambda Function:**
   * Lambda InH repository (Link share by Manju).
   * Modify configuration (e.g., Moogsoft endpoint, API token).
   * Set up Lambda triggers for:
     + CloudWatch Alarms
     + CloudWatch Logs (using log subscriptions)
2. **Create Sample CloudWatch Alarms:**
   * CPU Utilization > 80%
   * EC2 Instance Status Check Failed
   * EBS Burst Balance < 10%
3. **CloudWatch Logs Trigger Examples:**
   * Use synthetic logs:

bash

Copy

aws logs put-log-events --log-group-name /demo/errors --log-stream-name appStream --log-events timestamp=$(date +%s%3N),message="ERROR: DB connection failed"

1. **Verify Moogsoft Receives Events:**
   * Event ingestion time
   * Source, type, severity
   * Description and tags

**2. Event Aggregation**

**Verification Parameters:**

* **Ingestion latency**: Check how fast Moogsoft processes CloudWatch events.
* **Timestamp consistency**: Validate alignment between AWS and Moogsoft timestamps.
* **Correlation accuracy**:
  + Are similar errors from multiple nodes grouped?
  + Do repetitive alerts suppress correctly?

**Sample Aggregation Logic to Test:**

* Log pattern matching (regex)
* Group by resource ID or metric name
* Combine events from different services (e.g., EC2 + RDS alerts)

**3. Alerting**

**Alerting Types to Test:**

| **Type** | **Example** |
| --- | --- |
| Threshold | CPU > 90% for 5 mins |
| Anomaly | Sudden spike in Lambda invocation errors |
| Missing Data | No logs from an app for 10 minutes |
| Pattern Match | "OutOfMemory" in logs |
| Correlated Alerts | High CPU + Memory + Disk alerts grouped together |

**Moogsoft Alert Attributes:**

* Description, severity, class
* Tags (environment, region, component)
* Runbook URL link
* Owner or resolver group

**4. Ticket Generation**

**Validate:**

* Auto-ticket creation per alert
* Correct template fields (description, priority, assignment)
* State sync:
  + Close alert in Moogsoft → resolves ticket
  + Update status in ITSM reflects back in Moogsoft (optional bi-directional)

**Test Cases:**

1. Alert creates ticket in ServiceNow.
2. Moogsoft suppresses duplicate → only one ticket.
3. Ticket escalated in ITSM → Moogsoft reflects urgency.

**5. Licensing Confirmation**

**Checklist:**

* Is AWS CloudWatch integration supported under current license tier?
* How many integrations are allowed?
* Limit on daily events/alerts?
* Does it support ITSM connectors (ServiceNow)?
* SLA and support hours included