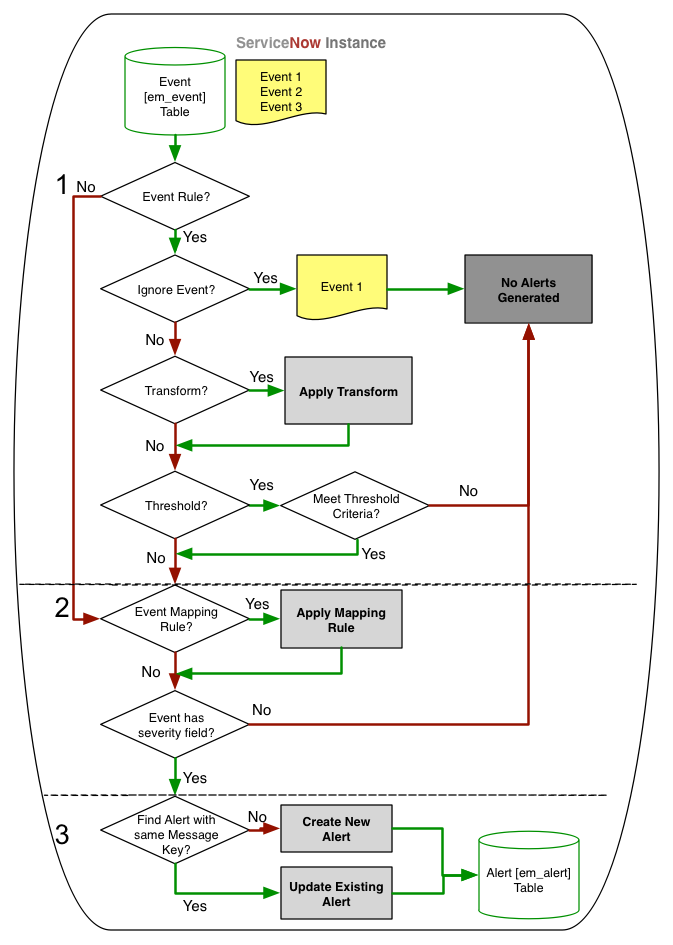
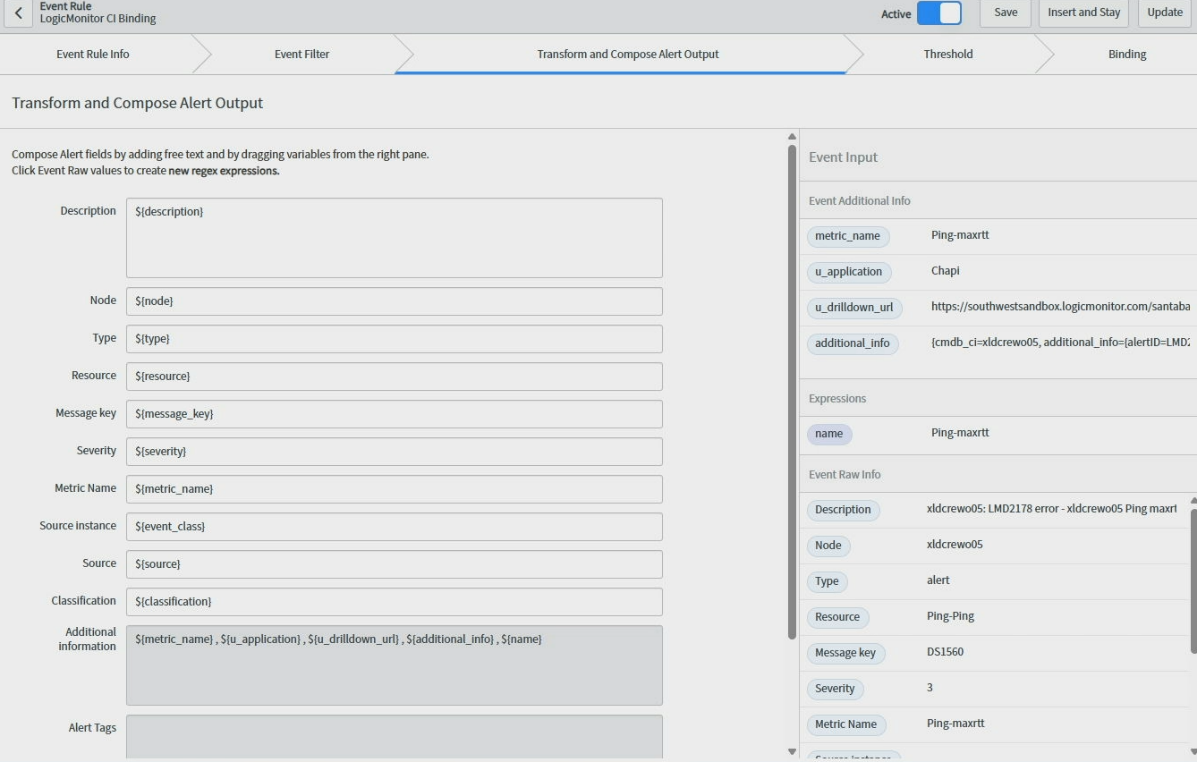
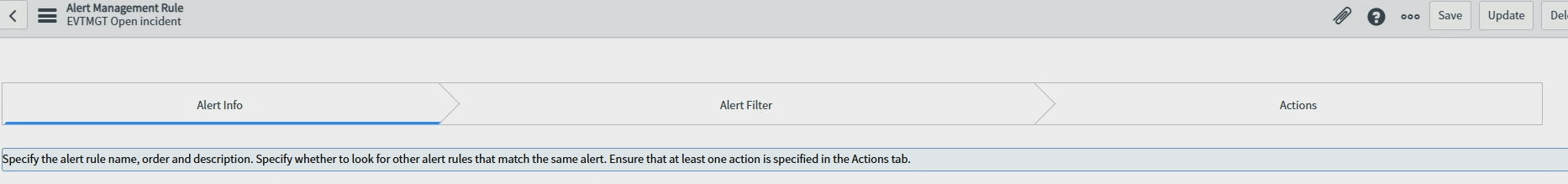
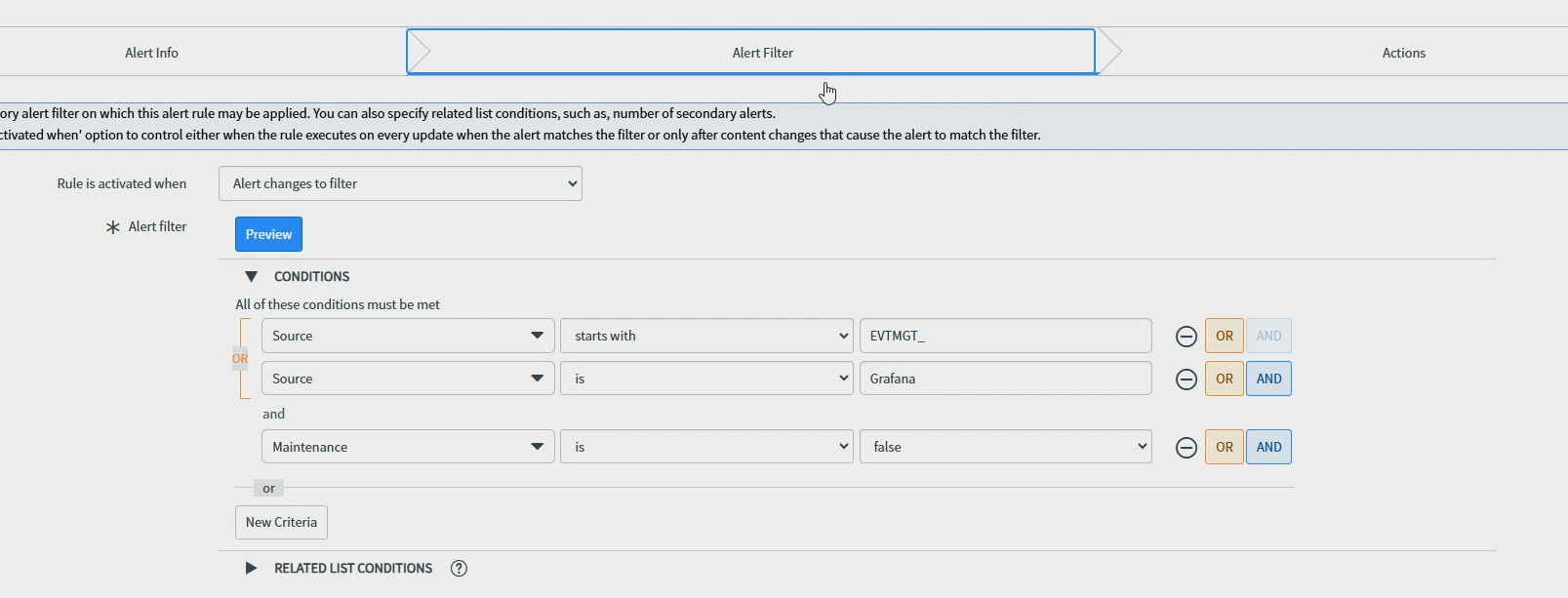
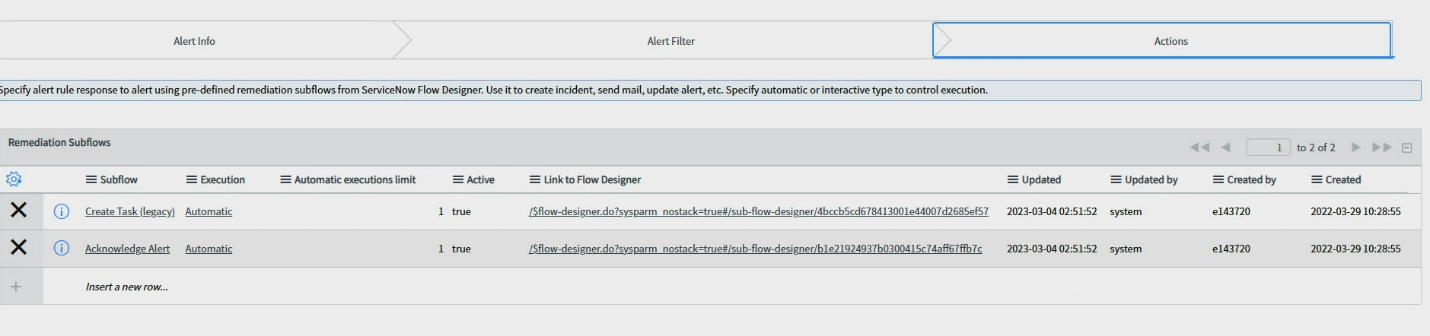
**Logic Monitor Event Management Setup Details**

* **Configuration For Sending Events from LM To ServiceNow**
* We have a user created with the ID: lm\_event and name: LogicMonitor Event Management, which we use to send alerts from LogicMonitor to ServiceNow.
* From ServiceNow, we provided a Scripted REST API, which the LM side uses to post to that endpoint with event details, and it sends the data to the **em\_event** table.
* Link to the Scripted REST API is this:
* **Event Processing Diagram**  
* **Event Processing Flow**

1. **Event Rule Check**
   * Incoming events are first evaluated against defined **Event Rules**.
   * If no rule matches, the system attempts default node binding using fields like FQDN, IP, Ci, or MAC.
   * In our case, we have a rule named ” LogicMonitor CI Binding”. Below is the URL for review:
2. **Ignore & Transform**
   * If the rule tells the system to **ignore**, no alert is created.
   * If not ignored, a **transform** may be applied to enrich or modify the event.
3. **Threshold Evaluation**
   * If threshold logic is configured, the system checks if the event meets criteria.
   * Events failing threshold checks are discarded (no alert generated).
   * Below is the screen short showing how we mapped things for alert generation in the Transform tab, which comes from events data in the event table.
4. **Field Mapping**
   * The system then applies **Field Mapping Rules** to map event fields into alert format.
   * If no valid mapping rule is found or severity is invalid, the event state is set to **error**.
5. **Alert Creation or Update**
   * If everything is valid and an alert rule exists, the system checks if an alert already exists (based on message key).
   * If yes, it **updates** the alert. If not, it **creates** a new one in the **[em\_alert]** table.

* **Alert Management Process Flow**
* Link of Alert Monument Rule:
* An Alert Rule has three parts: Alert Info, Alert Filter, and Actions.  
    
  
* In Alert Info, we put basic data like name and order. In our case, it's EVTMGT Open Incident.
* In Alert Filter, we can define for which alert it should run. For Logic Monitor, we have set it so that if the source name starts with EVTMGT\_, it should perform an action.  
    
  
* In the Action tab, we have set OOB sub flow, which triggers and performs actions like creating an incident.
* After this incident is created, it can be seen in the incident table.
* On same alert rule page, you can also review the execution details for how many alerts this rule has executed.

