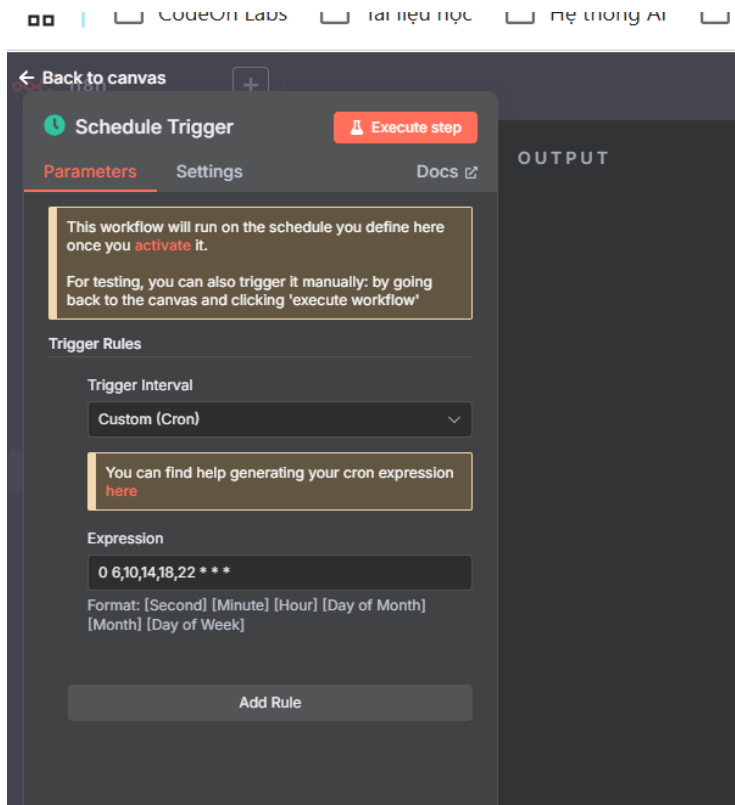


.3. Check SLA thời gian phản hồi lần đầu vV1.0

Node 1: Set schedule chạy 4 tiếng 1 lần, từ 6h sáng đến 22h hàng ngày để check

Cron: 0 6,10,14,18,22 * * *



Node 2: Dùng node HTTP request để kết nối với Jira

Tại sao không dùng Jira node để kết nối mà phải dùng HTTP request?

Trả lời: Khi check SLA team support trên Jira thường dựa vào thời gian comment, vì thế dùng HTTP request mới có thể lọc ra comment

Method: GET

URL: [https://jira.ezcloudhotel.com/rest/api/2/search?jql=project%3D%22ezCloud%20Support%20Team%22%20AND%20created%20%3E%3D%20-4h%20AND%20statusCategory%20!%3D%20Done%20AND%20status%20NOT%20IN%20\(%22Canceled%22%2C%20%22Cancelled%22%2C%20%22Duplicated%22%2C%20%22Resolved%22%2C%20%22Closed%22\)&fields=comment,created,priority,customfield_15718,assignee](https://jira.ezcloudhotel.com/rest/api/2/search?jql=project%3D%22ezCloud%20Support%20Team%22%20AND%20created%20%3E%3D%20-4h%20AND%20statusCategory%20!%3D%20Done%20AND%20status%20NOT%20IN%20(%22Canceled%22%2C%20%22Cancelled%22%2C%20%22Duplicated%22%2C%20%22Resolved%22%2C%20%22Closed%22)&fields=comment,created,priority,customfield_15718,assignee)

Node 3: Dùng code để trả ra các thông tin liên quan đến issue đang check

```
const issues = $input.all()[0].json.issues;
const now = new Date();

// Cấu hình múi giờ Việt Nam
const vnTimeOptions = {
  timeZone: 'Asia/Ho_Chi_Minh',
  year: 'numeric',
  month: '2-digit',
  day: '2-digit',
  hour: '2-digit',
  minute: '2-digit',
  second: '2-digit',
  hour12: false
};

// SLA cấu hình (giờ)
const slaTimes = {
  "Siêu VIP": { Block: 0.5, High: 0.5, Medium: 1, Low: 1 },
  "VIP": { Block: 1, High: 1, Medium: 2, Low: 2 },
  "Bình thường": { Block: 2, High: 2, Medium: 2, Low: 2 },
};

return issues.map(issue => {
  const created = new Date(issue.fields.created);

  const createdVN = created.toLocaleString('vi-VN', vnTimeOptions);

  // Chuyển về dạng YYYY-MM-DD theo múi giờ +7
  const recordDate = created.toLocaleDateString('en-CA', { timeZone: 'Asia/Ho_Chi_Minh' });

  // Nhóm khách hàng
  let customerType = issue.fields.customfield_15718;
  if (typeof customerType === 'object' && customerType?.value) {
    customerType = customerType.value;
  }
  customerType = customerType || 'Bình thường';

  // Ưu tiên
  const priority = issue.fields.priority?.name || 'Medium';
  const allowedHours = slaTimes[customerType]?.[priority] ?? 2;
  const allowedMinutes = allowedHours * 60;

  // Comment
  const comments = issue.fields.comment?.comments || [];
  const hasAnyComment = comments.length > 0;

  let firstCommentTime = null;
  let firstCommentTimeVN = null;
  let firstCommentBody = null;
  let firstCommentAuthor = null;

  if (hasAnyComment) {
    const sorted = [...comments].sort((a, b) => new Date(a.created) - new Date(b.created));
    const first = sorted[0];
    firstCommentTime = new Date(first.created);
    firstCommentTimeVN = firstCommentTime.toLocaleString('vi-VN', vnTimeOptions);
  }
});
```

```

firstCommentBody = first.body || '';
firstCommentAuthor = first.author?.displayName || first.author?.emailAddress || '';
}

const compareTime = firstCommentTime || now;
const diffMinutes = (compareTime - created) / 60000;

let slaViolated = false;
let reason = '';
if (diffMinutes > allowedMinutes) {
  slaViolated = true;
  reason = firstCommentTime
    ? `Tiếp nhận lần đầu trễ ${Math.round(diffMinutes)} phút > ${allowedMinutes} phút`
    : `Chưa có comment sau ${Math.round(diffMinutes)} phút`;
}

// Assignee
const assignee = issue.fields.assignee || {};
const assigneeName = assignee.displayName || '';
const assigneeEmail = assignee.emailAddress || '';

// accountId từ email (lấy phần trước dấu @)
const accountId = assigneeEmail.includes('@')
  ? assigneeEmail.split('@')[0]
  : '';

return {
  json: {
    issueKey: issue.key,
    customerType,
    priority,
    created: createdVN,
    recordDate,
    firstCommentTime: firstCommentTimeVN,
    diffMinutes: Math.round(diffMinutes),
    allowedMinutes,
    slaViolated,
    reason,
    firstCommentBody,
    firstCommentAuthor,
    assigneeName,
    assigneeEmail,
    accountId
  }
};
});

```

Node 4: Dùng code để trả ra những issue bị trễ hạn

```

const items = $input.all();

// Giữ lại toàn bộ thông tin nhưng chỉ trả ra những issue vi phạm SLA
return items.filter(item => item.json?.slaViolated === true).map(item => {
  return {
    json: {
      ...item.json
    }
  };
});

```

Node 5: Tạo NC trên Jira từ thông tin issue bị trễ hạn trả ra

Tại sao không dùng node Jira action để tạo issue mà phải dùng Node HTTP request?

Trả lời: Hiện tại n8n sẽ kết nối với Jira bản mới nhất. Đối với Jira bản mới nhất sẽ không chấp nhận những accountID có dấu (.) trong thành phần. Mà hiện tại ezCloud đang dùng Jira bản cũ, vì vậy phải dùng node HTTP để thỏa mãn tất cả điều kiện hiện tại.

Method: POST

URL: <https://jira.ezcloudhotel.com/rest/api/2/issue>

Body JSON:

```
{
  "fields": {
    "project": {
      "key": "NCT"
    },
    "summary": "NC tự động: Vi phạm SLA tiếp nhận issue lần đầu",
    "issuetype": {
      "name": "Nonconformity"
    },
    "assignee": {
      "name": "{{ $json.accountId }}"
    },
    "customfield_13428": "{{ $json.recordDate }}",
    "description": "Link issue: https://jira.ezcloudhotel.com/browse/{{ \$json.issueKey }}"
  }
}
```