



Data Sheet

# Integrating Snowflake with Microsoft Teams for Real-Time Alerts

5201 GREAT AMERICAN PARKWAY, SUITE 320

SANTA CLARA, CA 95054

Tel: (855) 695-8636

E-mail: [info@lumendata.com](mailto:info@lumendata.com)

Website: [www.lumendata.com](http://www.lumendata.com)

# Integrating Snowflake with Microsoft Teams for Real-Time Alerts

In today's fast-paced world, tracking data operations and alerts is essential. Integrating Snowflake with Microsoft Teams enables seamless communication by:

- Sending real-time alerts for task failures and critical events.
- Monitoring warehouse usage and notifying anomalies.
- Sharing SQL query results directly in Teams channels.

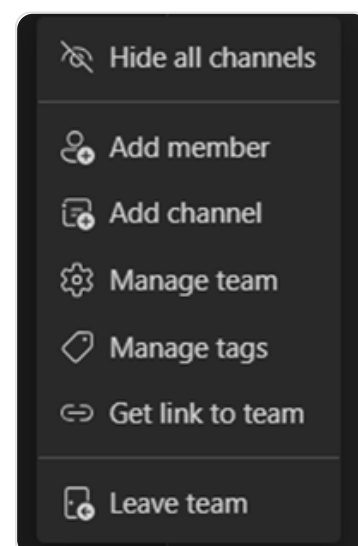
This integration ensures teams stay informed and act swiftly on key insights.

## Setting Up the Integration

### Step 1: Create a Channel in Microsoft Teams

To begin, you'll need a dedicated channel in Microsoft Teams where you can receive notifications from Snowflake.

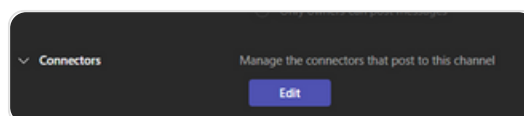
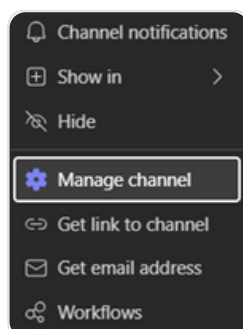
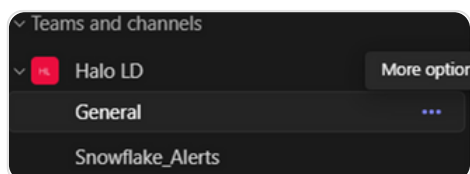
- Open Microsoft Teams and navigate to the Teams section.
- Create a Channel: If you don't already have a team, create one first.
- Then, within the team, create a channel where notifications will be sent.



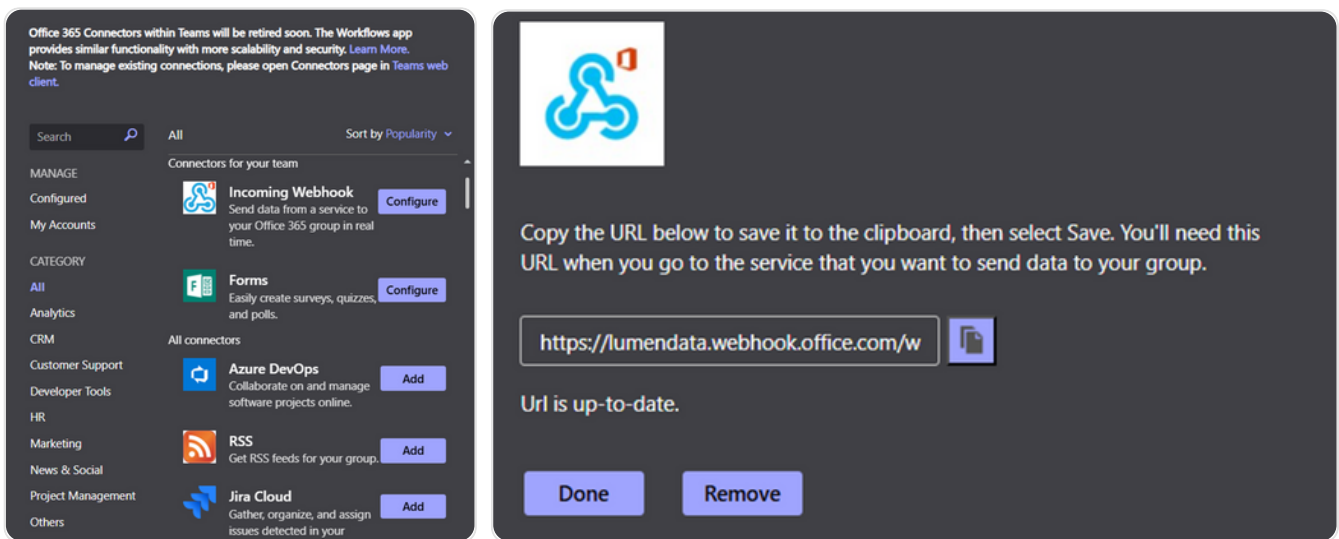
### Step 2: Configure an Incoming Webhook in Teams

Incoming Webhooks allow external applications like Snowflake to send content to your Teams channels.

- Navigate to the Manage Channel section of your newly created channel.
- Click on **Connectors** -> **Edit** -> **Add Incoming Webhook**.
- Name your webhook and copy the generated URL. You'll need this URL later to configure Snowflake.



# Integrating Snowflake with Microsoft Teams for Real-Time Alerts



## Step 3: Create a Network Rule in Snowflake

The next step is to configure Snowflake to allow outbound communication with the Teams webhook. This is done by creating a network rule that specifies the host and port for outgoing traffic.

### Creating a network rule:

```
create OR REPLACE network rule "EGRESS__TEAMS_WEBHOOKS"
```

```
type = HOST_PORT
```

```
mode = EGRESS
```

```
value_list = ('*.webhook.office.com') -- Populate with webhook host of the target Teams account, such as customer.webhook.office.com
```

```
comment = 'Allows traffic to be sent to webhooks in the target Teams environment'
```

```
;
```

This rule ensures that Snowflake can send data to the specific Teams webhook URL.

## Step 4: Create an External Access Integration in Snowflake

Next, you'll set up an external access integration to facilitate secure communication between Snowflake and Microsoft Teams.

### EXTERNAL ACCESS INTEGRATION:

```
CREATE OR REPLACE EXTERNAL ACCESS INTEGRATION "EAI__TEAMS_WEBHOOKS"  
ALLOWED_NETWORK_RULES = ("EGRESS__TEAMS_WEBHOOKS")  
  
ENABLED = TRUE  
  
COMMENT = 'External access integration to support traffic sent to webhooks in the  
target Teams environment';
```

## Step 5: Create a Function to Send Notifications to Teams

Now that the basic setup is complete, you can create a function in Snowflake that sends notifications to Microsoft Teams using the webhook URL.

```
CREATE OR REPLACE FUNCTION SEND_TEAMS_NOTIFICATION(  
    WEBHOOK_URL STRING,  
    MESSAGE STRING  
)  
  
RETURNS STRING  
  
LANGUAGE PYTHON  
  
RUNTIME_VERSION = 3.10  
  
PACKAGES = ('requests')  
  
EXTERNAL_ACCESS_INTEGRATIONS = ("EAI__TEAMS_WEBHOOKS")  
  
HANDLER = 'send_message'  
  
AS  
  
$$  
  
import requests  
  
def send_message(webhook_url, message):  
    payload = {"text": message}  
    response = requests.post(webhook_url, json=payload)  
  
    if response.status_code == 200:  
        return "Notification sent successfully"  
    else:  
        return f"Failed to send notification. Status code: {response.status_code}"  
  
$$;
```

This function takes a webhook URL and a message as input parameters and sends the message to the specified Teams channel.

## Step 6: Create a Procedure to Send Data Notifications

Finally, let's create a procedure that retrieves warehouse usage data and sends it as a formatted message to your Teams channel.

```
CREATE OR REPLACE PROCEDURE SEND_MAIL()
RETURNS STRING
LANGUAGE JAVASCRIPT
AS
$$
var webhook_url = 'https://lumendata.webhook.office.com/webhookb2/88fc7e5d-
a987-45f5-a317-cda365a6056b@6920e33a-c7f9-4e8f-b7bd-
9abdb2d207e8/IncomingWebhook/a87f942a8532467393f57753adb864d0/9b13ab88-
7920-4ffa-b687-4e5742794bc5/V2KUBPKEqwhnOcSZk-
f6UANmFMt_SuvtbDO2mnemW/mqC81'
var sql_query = `
    SELECT
    DATE(start_time) AS sdate,
    warehouse_name,
    SUM(credits_used) AS credits
    FROM
    snowflake.account_usage.warehouse_metering_history
    WHERE
        start_time >= DATEADD(day, -1, CURRENT_TIMESTAMP())
    GROUP BY
    DATE(start_time), warehouse_name
    ORDER BY
        1 DESC, 2 ASC`;
```

```
var sqlstmt = snowflake.createStatement({ sqlText: sql_query });
var rs = sqlstmt.execute();

var msg = `<html><head><style>
  table {
    border-collapse: collapse;
    width: 100%;
  }
  th, td {
    border: 1px solid black;
    padding: 8px;
    text-align: center;
  }
  tr:hover {
    background-color: coral;
  }
</style></head><body><table><tr><th>DATE</th><th>WAREHOUSE NAME</th>
<th>TOTAL CREDITS</th></tr>`;

while (rs.next()) {
  var DATE = rs.getColumnValue(1);
  var WAREHOUSE_NAME = rs.getColumnValue(2);
  var CREDITS = rs.getColumnValue(3);
  var rowColor = CREDITS > 1 ? '#fc353f' : '#0fffa3';

  msg += '<tr><td>' + DATE + '</td><td>' + WAREHOUSE_NAME + '</td><td>
style="background-color: ' + rowColor + '">' + CREDITS + '</td></tr>';
}
```

```
msg += '</table></body></html>';

var notificationResult = snowflake.execute({
  sqlText: `SELECT SEND_TEAMS_NOTIFICATION(?, ?)`,
  binds: [webhook_url, msg]
});

var result = "No result returned";
if (notificationResult.next()) {
  result = notificationResult.getColumnValue(1);
}

return result;
$$;
```

## Testing :



The screenshot shows the Snowflake SQL Editor interface. At the top, the database 'DINESH\_PRACTISE.PUBLIC' and a 'Settings' dropdown are visible. The SQL editor contains the following code:

```
100
101 CALL SEND_MAIL();
102
103
104
```

Below the editor, there are two tabs: 'Results' (selected) and 'Chart'. The 'Results' tab displays a table with one row and one column:

	SEND_MAIL
1	Notification sent successfully

# Integrating Snowflake with Microsoft Teams for Real-Time Alerts

## Result in Teams Channel:

The screenshot displays a Microsoft Teams channel interface for 'Snowflake\_Alerts'. The channel header includes tabs for 'Posts', 'Files', and 'Notes'. Two alert messages are visible, both originating from 'Snowflake\_Alerts\_POC'.

**Alert 1 (4:07 PM):**

DATE	WAREHOUSE NAME	TOTAL CREDITS
Mon Mar 03 2025 00:00:00 GMT-0800 (Pacific Standard Time)	COMPUTE_WH	0.166965833

**Alert 2 (4:19 PM):**

DATE	WAREHOUSE NAME	TOTAL CREDITS
Mon Mar 03 2025 00:00:00 GMT-0800 (Pacific Standard Time)	COMPUTE_WH	0.351688056

The second alert is marked as 'New' in a blue bubble. Both alerts include a 'Reply' button at the bottom.



## Authors



**Abhishek Gaur**  
Technical Lead - Level 1



**P Dinesh Reddy**  
Junior Data Analyst

## About LumenData

LumenData is a leading provider of **Enterprise Data Management, Cloud & Analytics** solutions. We help businesses navigate their data visualization and analytics anxieties and enable them to accelerate their innovation journeys.

**Founded in 2008**, with locations in multiple countries, LumenData is privileged to serve over 100 leading companies. LumenData is **SOC2 certified** and has instituted extensive controls to protect client data, including adherence to GDPR and CCPA regulations.



Get in touch with us:  
[info@lumendata.com](mailto:info@lumendata.com)

Let us know what you need:  
[lumendata.com/contact-us](https://lumendata.com/contact-us)

