**End-to-End Integration: Slack →API-GATEWAY- AWS Lambda → Jenkins**

**📖 Overview**

This document explains how to integrate:

* Slack (where a user sends a message)
* AWS Lambda (which parses and triggers builds)
* Jenkins (which performs the build and reports back)

It enables natural Slack commands like:  
build slack-pipeline main  
→ triggers the Jenkins job and posts success/failure back in Slack.

**🔧 2. Slack App & Bot Setup**

Configure Slack App (Step-by-Step)

🔹 Step 1: Create the App

1. Go to: https://api.slack.com/apps
2. Click "Create New App"
3. Choose From scratch
4. Name your app (e.g., JenkinsTriggerBot)
5. Choose your workspace

🔹 Step 2: Add Bot User

1. Go to Bot Users
2. Click Add a Bot User
3. Set Display Name and Default Username
4. Save

🔹 Step 3: Add OAuth Scopes

1. Navigate to OAuth & Permissions
2. Under Bot Token Scopes, add:
   * chat:write
   * users:read
   * channels:read
   * groups:read
3. Click Install to Workspace

⚠️ This will generate the SLACK\_BOT\_TOKEN. Save it securely.

**☁️ 3. AWS Lambda Setup**

**3.1 Create Lambda Function**

1. Go to AWS Lambda → Create Function
2. Name it (e.g., SlackToJenkinsTrigger)
3. Use Python 3.11 or 3.10 runtime
4. Create or attach an IAM role with:
   * AmazonSSMFullAccess
   * AWSLambdaBasicExecutionRole

**3.2 Set Environment Variables**

Add these to the Lambda function:

| **Key** | **Value** |
| --- | --- |
| SLACK\_BOT\_TOKEN | xoxb-... from Slack app |
| JENKINS\_URL | Jenkins base URL |
| JENKINS\_USER | Jenkins API username |
| JENKINS\_API\_TOKEN | Jenkins API token |
| SSM\_INSTANCE\_ID | EC2 instance ID (if needed) |
| JENKINS\_FOLDER | (Optional) Jenkins folder |

**3.3 Deploy Lambda Handler**

Add layers for python-request to handle slack requests

Use the script you shared — already updated to resolve user names using Slack API (users.info).

**4.Create HTTP API Gateway**

**Step 1: Open API Gateway Console**

Go to: <https://console.aws.amazon.com/apigateway>

**Step 2: Create New API**

* Choose: **HTTP API**
* Click **Build**
* Under “Integrations”, choose:
  + **Lambda Function**
  + Select your function (e.g., SlackToJenkinsTrigger)
* Click **Next**

**Step 3: Configure Routes**

* Add a route:
  + Method: POST
  + Resource path: /slack/events
* Click **Next**

**Step 4: Deploy the API**

* Create a new stage (e.g., prod)
* Click **Deploy**

**Step 5: Get the Public Endpoint URL**

After deployment, you’ll see something like:

**API Gateway URL** : https://abc123xyz.execute-api.us-east-1.amazonaws.com/slack/events

This is your **Request URL** for Slack.

**🔄 Step 6: Connect to Slack**

1. Go to your Slack App settings
2. Navigate to: **Event Subscriptions**
3. Toggle **ON**
4. Paste the **API Gateway URL** (e.g., https://.../slack/events) into the **Request URL** field
   * Slack will send a challenge request to verify(this should ne handled in lambda code)
5. If successful, it should show “Verified”

**5. Event Subscriptions**

**🔹 Step 1: Enable Event Subscriptions**

1. Go to Event Subscriptions
2. Turn ON Enable Events
3. Set Request URL to your Lambda’s API Gateway URL  
   e.g., https://xyz.execute-api.region.amazonaws.com/prod/slack
4. Under Subscribe to bot events, add:
   * message.channels (for public channels)
   * message.groups (for private channels)

⚠️ Slack will attempt a verification handshake — your Lambda must respond correctly.

**🔹 Step 2: Add the App to Your Slack Channel**

1. In Slack, go to the channel where you want to trigger builds
2. Type: /invite @JenkinsTriggerBot

**⚙️ 6. Jenkins Setup**

Launch server and install all dependencies , java 17 or later and then install Jenkins.

**6.1: Create Multibranch Pipeline Job**

1. Navigate to Jenkins Dashboard

* Click "New Item"

2. Create Multibranch Pipeline

* Enter a name (e.g., slack-pipeline)
* Choose Multibranch Pipeline
* Click OK

🗂️ 3. Configure Branch Source

1. In "Branch Sources" section:

* Click "Add Source" → Choose Git or GitHub
* Provide repository URL (e.g., https://github.com/your-org/your-repo.git)
* Add credentials if needed
* Set behaviors if necessary (e.g., filter by name or tag)

2. Define Jenkinsfile

* Ensure your repo has a valid Jenkinsfile in each branch you want Jenkins to build

🔁 4. Configure Scan Triggers (Optional but Recommended)

* Under "Scan Multibranch Pipeline Triggers"
  + Enable Periodically if not otherwise run

Set interval (e.g., every 5 minutes)

**6.2 Install Slack Notification Plugin**

1. Go to: Jenkins > Manage Jenkins > Plugin Manager
2. Under Available, search for Slack Notification Plugin
3. Install the plugin and restart Jenkins if needed

**🔹 Step 2: Add Slack Global Configuration**

1. Go to: Manage Jenkins > Configure System
2. Scroll to the Slack section
3. Fill in the details:
   * Workspace/Team Domain: your Slack domain (e.g., mycompany if your workspace is mycompany.slack.com)
   * Integration Token Credential ID:
     + Click Add → Jenkins
     + Choose Secret text
     + Enter your Slack Bot Token (xoxb-...)
     + Give an ID like slack-token
   * Channel: your channel name (e.g., #build-notifications)
   * Build Server URL: your Jenkins URL (e.g., <http://jenkins.mycompany.com/>) -optional
   * Test the connection with the "Test Connection" button

4.Save the configuration