Slack Integration for Bash Scripts (Minimal Permissions)

# 1. Create a Slack Account and Workspace

1. Go to https://slack.com.  
2. Click 'Get Started' and create a new workspace if you don’t already have one.  
3. Follow the on-screen steps to complete the setup.

# 2. Create a Slack App (Example Name: dbWatch Service Alert)

1. Visit https://api.slack.com/apps and click 'Create New App'.  
2. Choose 'From scratch'.  
3. Name your app 'dbWatch Service Alert' (or another descriptive name).  
4. Select your workspace and click 'Create App'.

# 3. Configure Minimal OAuth & Permissions

Go to 'OAuth & Permissions' in your app settings and add the following \*\*Bot Token Scopes\*\*:  
- chat:write — to send and delete the bot's own messages  
- channels:read — (optional) to resolve channel names or IDs  
  
Click 'Install App to Workspace' and approve the permissions.  
Copy the Bot User OAuth Token (starts with xoxb-...).

# 4. Invite the Bot to a Slack Channel

1. In your Slack workspace, navigate to the target channel.  
2. Type: /invite @dbWatch Service Alert  
(Replace with your actual app name if different.)

# 5. Modify Your Script to Use Slack Integration

Edit your CustomerOp.sh script to include your Slack token and channel. Example:

SLACK\_TOKEN="xoxb-xxxxxxxxxxxxxxxx"  
SLACK\_CHANNEL="C1234567890" # or "#alerts"

# 6. Script Functionality Summary

- Deletes the previous Slack message using the chat.delete API (allowed by chat:write).  
- Posts a new combined status message using chat.postMessage.  
- Stores the timestamp for the next run.

# 7. Script Installation Instructions

1. Place the CustomerOp.sh script on a Linux system with executable permissions.  
2. Ensure `ccc.sh` is available at /opt/dbwatch-controlcenter/ccc.sh and is executable.  
3. Create a file called `customers.ini` in the same directory as CustomerOp.sh, formatted like:

<ACCESSPOINT>,<TARGET>,<DOMAINNAME>,<TOKEN>

- <ACCESSPOINT>: IP address and port of dbWatch Control Center domaincontroller (e.g., 192.168.7.30:7100).  
- <TARGET>: Node ID of the domaincontroller, found under menu 'Server' → 'Domain Configuration' → 'Nodes'. Example: -6526196164457313900/-749536326377100173.  
- <DOMAINNAME>: As shown in the Domain Configuration and license file.  
- <TOKEN>: The Cloud Router access token for the domain.  
  
You can list multiple lines in the ini file to monitor multiple domains/servers(instance hubs).

4. Upload `slack.xml` to dbWatch Control Center:  
- Go to main menu → 'Server' → 'Upload resource'  
- Select and upload the `slack.xml` file

5. On first execution, the script will fail if access to the domaincontroller hasn't been approved.  
Follow the approval instructions at: <https://wiki.dbwatch.com/controlcenter/advanced-topics/ccc/setup>

The script SetupStatus.sh can be used to ease the setup of access privileges. The ccc node would require at least read on “All Instances” and read on “All Servers” needed for the monitoring.