Prepare to Send Slack Notifications from Autonomous Database

Introduction

Slack, a leading collaboration tool, provides teams to have seamless communication and efficient data management. Also, Slack has cool integrations with external app and sources. This is precisely what unites Oracle Autonomous Database and Slack in this lab.

This lab walks you through the steps to configure your Slack application to receive messages,alerts and output of a query from Autonomous Database. Also, you will learn the required procedures to send Slack notifications.

Before we started, this lab requires the following:

- Slack account and workspace
- Slack app
- Slack channel

> **Note:** A Slack workspace is made up of channels, where team member can communicate and work together. To join a workspace, you will create an account using your email address.

Estimated Time: 20 minutes

Objectives

In this lab, you will:

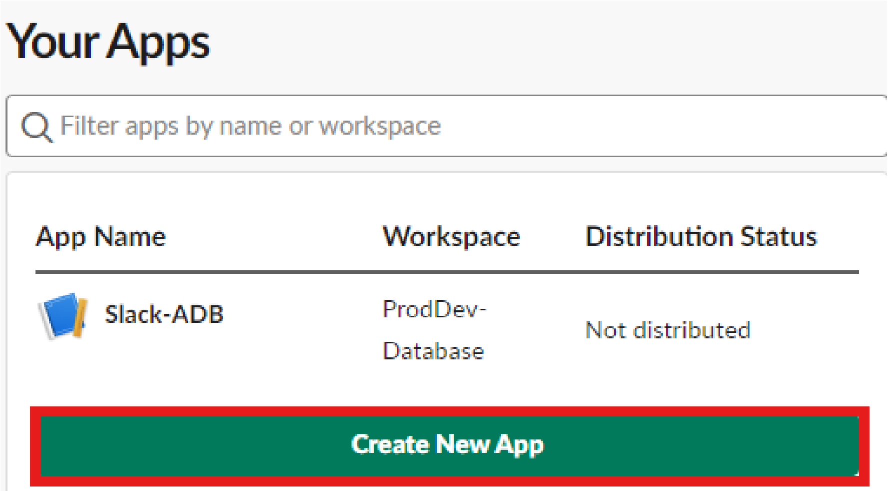
- Create your Slack application
- Configure your Slack application
- Create your Slack channel
- Create a credential object to access the Slack application from Autonomous Database
- Configure access control to allow user access to external network services
- Send messages to a Slack channel
- Send query results to a Slack Channel

Prerequisites

- This lab requires completion of the lab **Set up the environment** from the **Contents** menu on the left.

Task 1: Create your Slack application

1. To create a Slack application, click this link to access **Your Apps** page, and select **Create New App**.

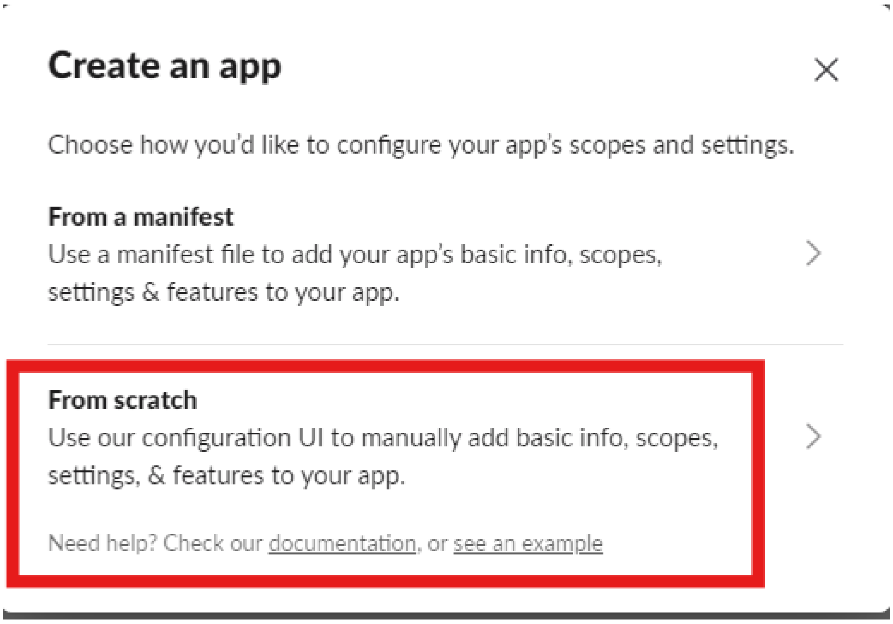


Your Apps

Filter apps by name or workspace

App Name	Workspace	Distribution Status
Slack-ADB	ProdDev-Database	Not distributed

2. Select **From scratch**.



Create an app

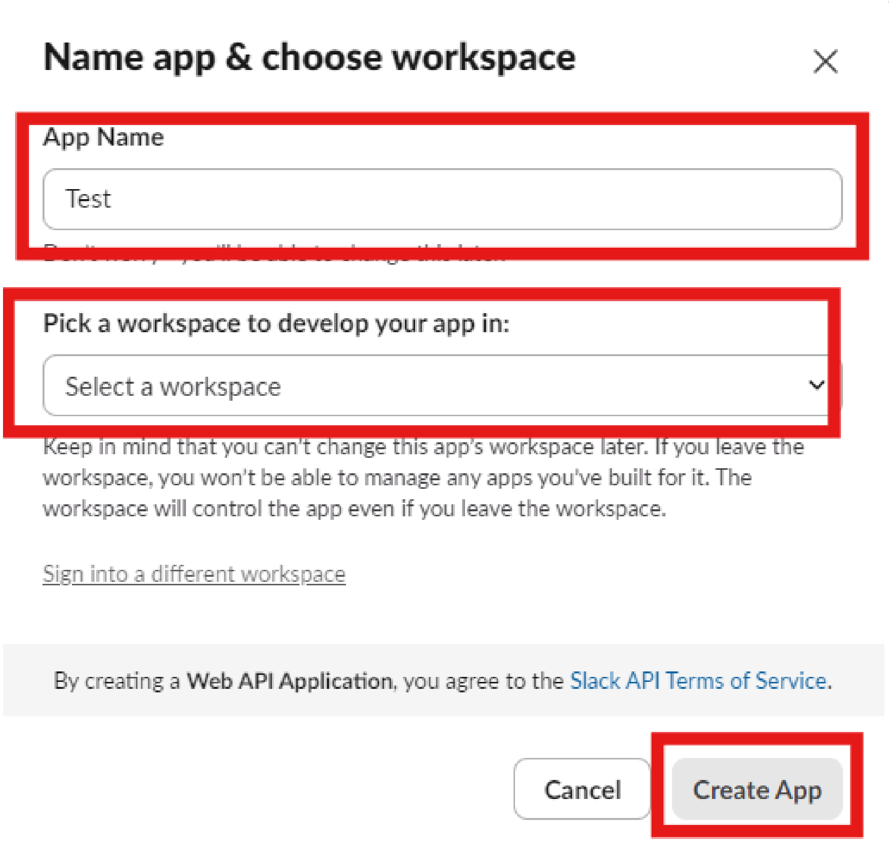
Choose how you'd like to configure your app's scopes and settings.

From a manifest
Use a manifest file to add your app's basic info, scopes, settings & features to your app.

From scratch
Use our configuration UI to manually add basic info, scopes, settings, & features to your app.

Need help? Check our documentation, or see an example

3. Enter your **App Name**, then select your **Workspace**, and click **Create App**.



Name app & choose workspace

App Name

Test

Don't worry; you'll be able to change this later.

Pick a workspace to develop your app in:

Select a workspace

Keep in mind that you can't change this app's workspace later. If you leave the workspace, you won't be able to manage any apps you've built for it. The workspace will control the app even if you leave the workspace.

Sign into a different workspace

By creating a **Web API Application**, you agree to the Slack API Terms of Service.

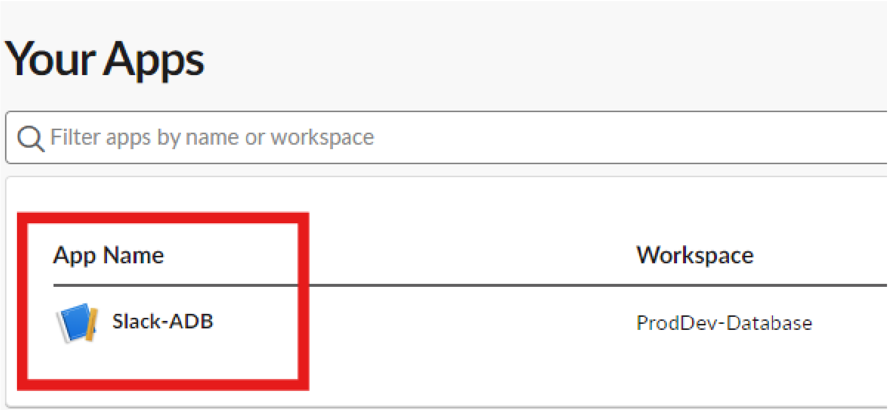
Task 2: Configure your Slack application

After creating your application, you must request scopes which will give your application permission to perform actions such as viewing basic information, posting messages, and uploading files in your workspace.

These following scopes are required:

- channels:read
- chat:write
- files:write

1. Go to main application page, then click your **App Name**.

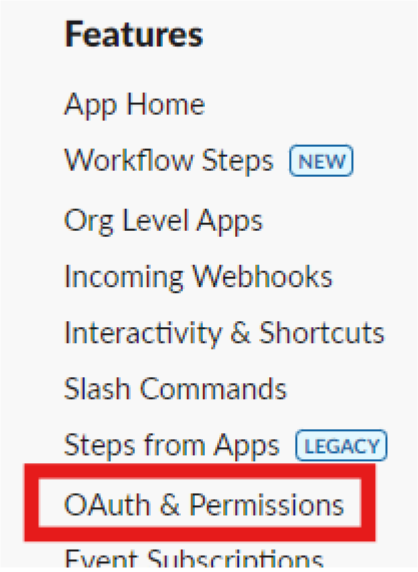


Your Apps

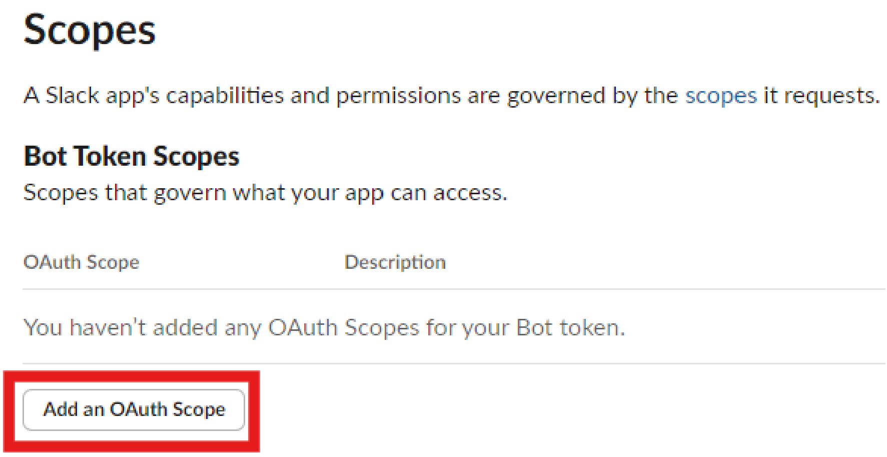
Filter apps by name or workspace

App Name	Workspace
Slack-ADB	ProdDev-Database

2. Scroll down to **Features**, and click **OAuth & Permissions**.



3. Scroll down to **Scopes**, and click **Add an OAuth Scope**.



Scopes

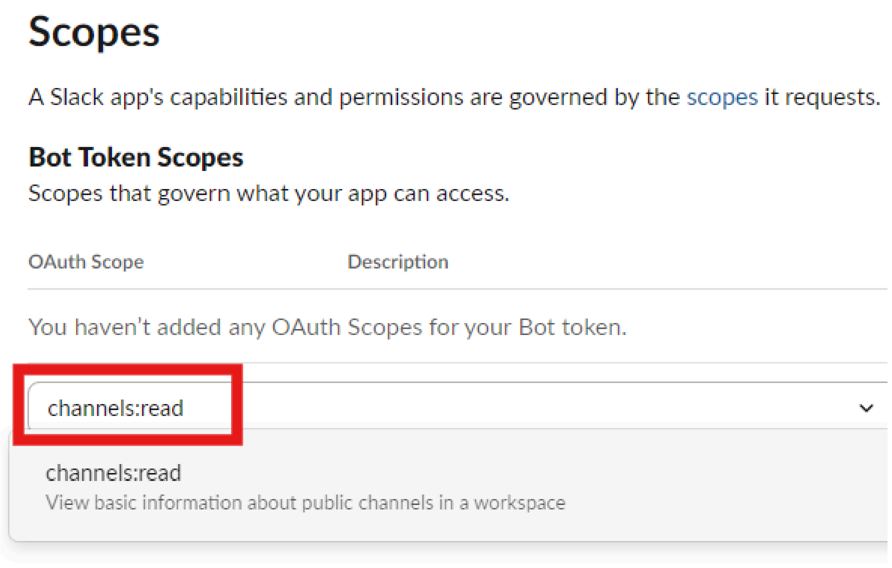
A Slack app's capabilities and permissions are governed by the scopes it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description
You haven't added any OAuth Scopes for your Bot token.	

4. For OAuth Scope, enter **channels:read**, and select the scope from drop-down list. This scope allows your app to access public Slack channels.



Scopes

A Slack app's capabilities and permissions are governed by the scopes it requests.

Bot Token Scopes

Scopes that govern what your app can access.

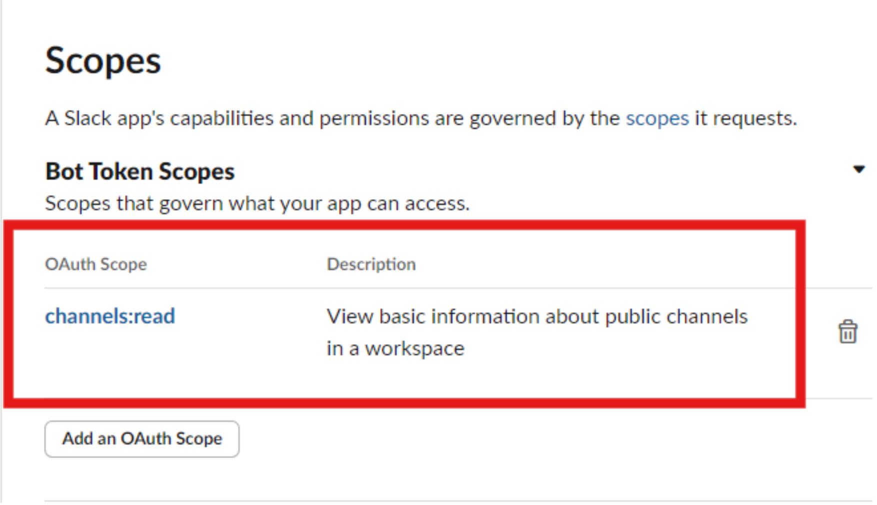
OAuth Scope	Description

You haven't added any OAuth Scopes for your Bot token.

channels:read

channels:read
View basic information about public channels in a workspace

5. If successful, you will see recently added **OAuth Scope** with **Description**.



Scopes

A Slack app's capabilities and permissions are governed by the scopes it requests.

Bot Token Scopes

Scopes that govern what your app can access.

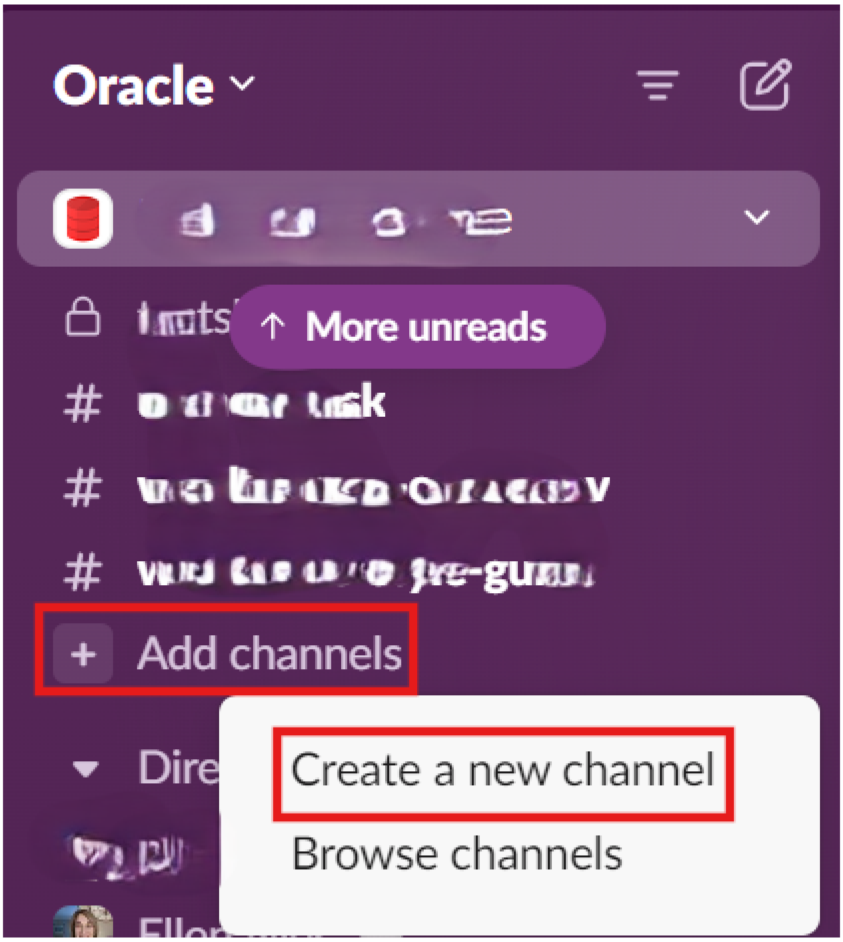
OAuth Scope	Description
channels:read	View basic information about public channels in a workspace

6. Repeat Task 5, to add these scopes **chat:write** and **files:write** to your application as well.

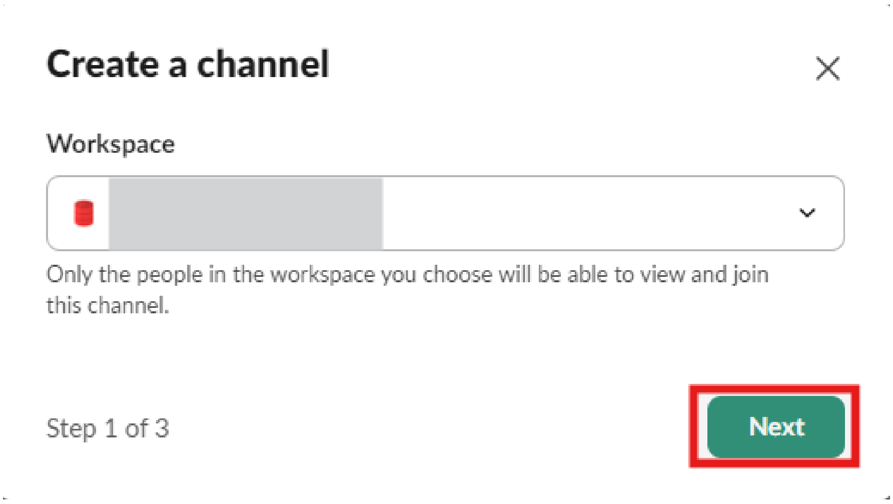
Task 4: Create a Slack channel

To use Slack with DBMS_CLOUD_NOTIFICATION procedure, your Slack admin must add the Slack app to the channels so that you can send message through the **Integrations** in the channel.

1. To create a Slack channel, open the Slack app and click **Add channels**.

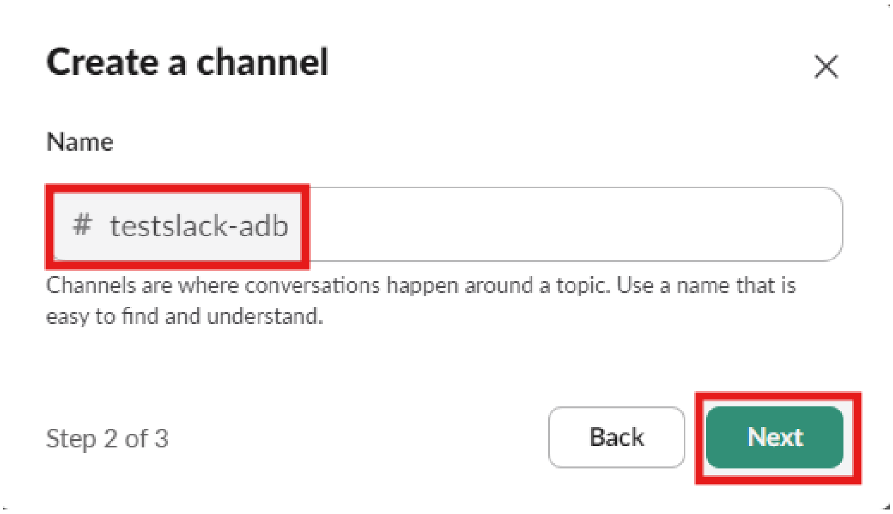


2. Select your workspace, and click **Next**.



Only the people in the workspace you choose will be able to view and join this channel.

3. Enter a **Name** for your channel, and click **Next**.



Create a channel

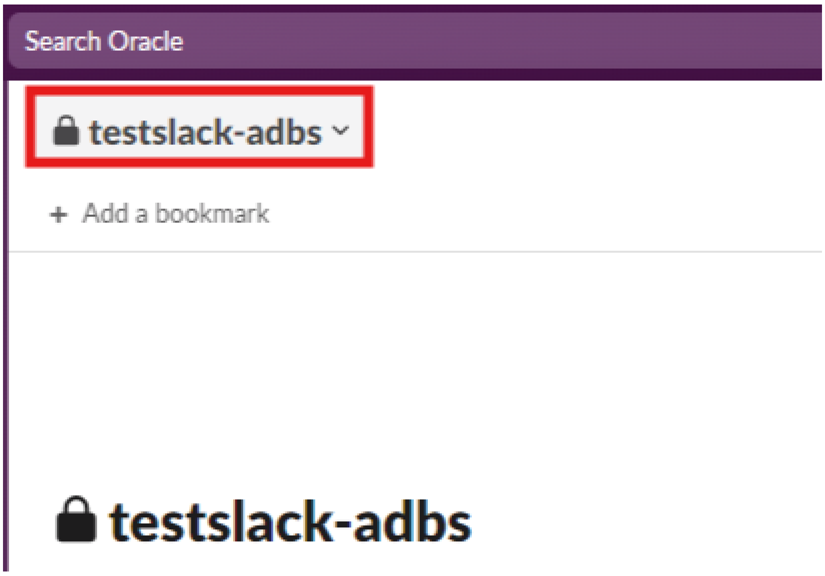
Name

testslack-adb

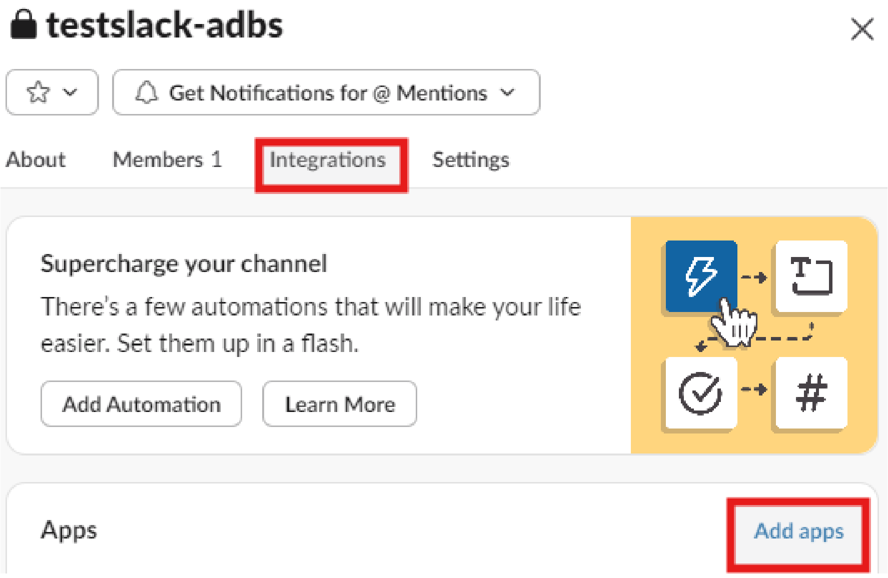
Channels are where conversations happen around a topic. Use a name that is easy to find and understand.

Step 2 of 3

4. If successful, you will see your channel in your workspace. Click your **Channel name** (arrow icon), and then select **Integrations**.



5. Click **Add apps** to integrate the application into your channel.



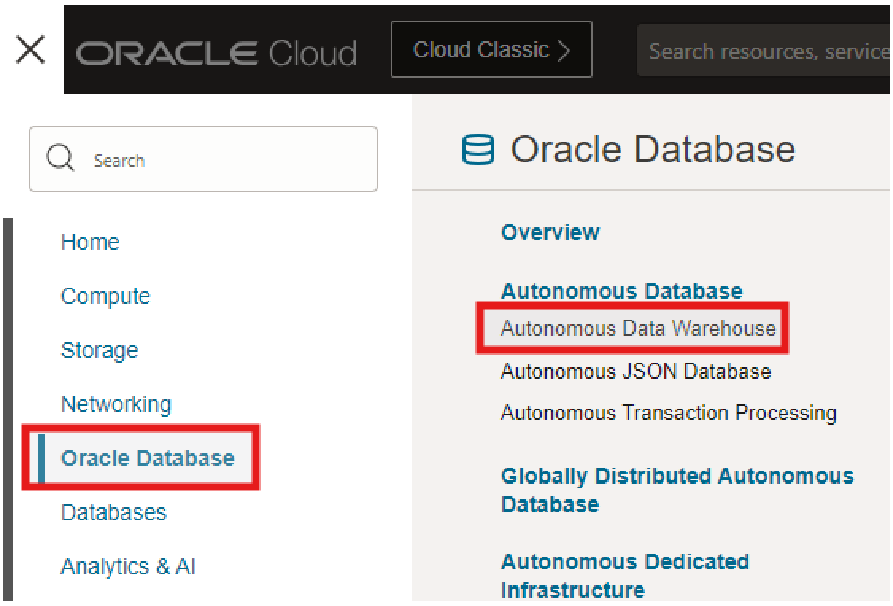
testslack-adbs

Supercharge your channel

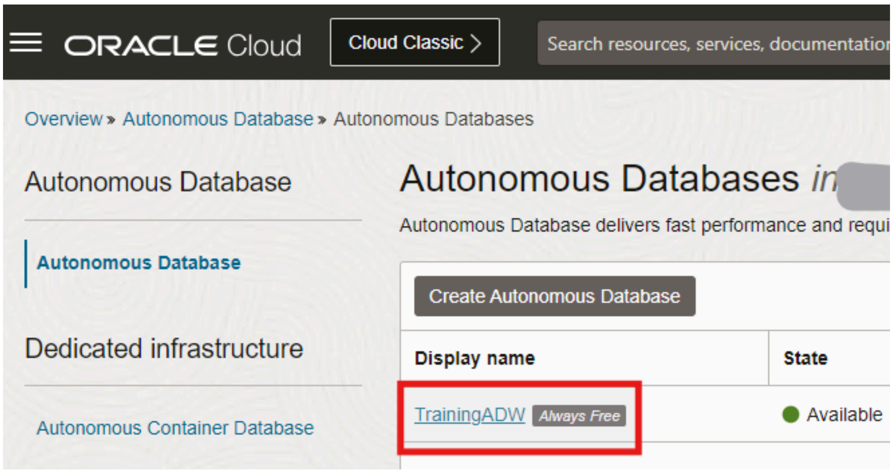
There's a few automations that will make your life easier. Set them up in a flash.

Task 5: Create a credential object to access the Slack application from Autonomous Database

1. In the Oracle Cloud console, open the **Navigation menu**, navigate to **Oracle Database**, and then select **Autonomous Data Warehouse**.

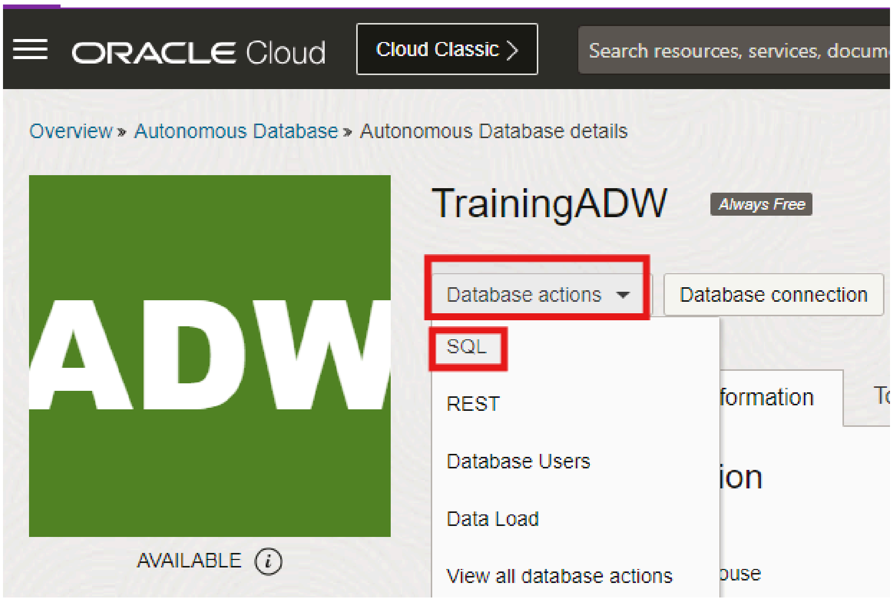


2. Select your **Autonomous Database instance**.


ORACLE Cloud
Cloud Classic >
Search resources, services, documentatio
Overview » Autonomous Database » Autonomous Databases
Autonomous Database
Autonomous Database
Dedicated infrastructure
Autonomous Container Database
Autonomous Databases in
Autonomous Database delivers fast performance and requi
Create Autonomous Database

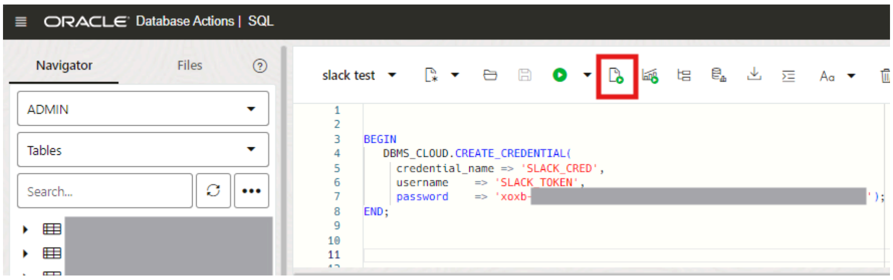
Display name	State
TrainingADW Always Free	● Available

3. Click **Database actions** drop-down list, and select **SQL** .



4. To create credential object to access the Slack app from Autonomous Database, obtain the **password** from your Slack admin. Copy and paste the following code into your SQL Worksheet, and then click the **Run Script (F5)** icon in the Worksheet toolbar.

```
<copy>BEGIN DBMS_CLOUD.CREATE_CREDENTIAL(   credential_na
```



```


BEGIN
   DBMS_CLOUD.CREATE_CREDENTIAL(
     credential_name => 'SLACK_CRED',
     username     => 'SLACK_TOKEN',
     password     => 'xoxb-[redacted]');
END;
```

For more information, see CREATE_CREDENTIAL Procedure.

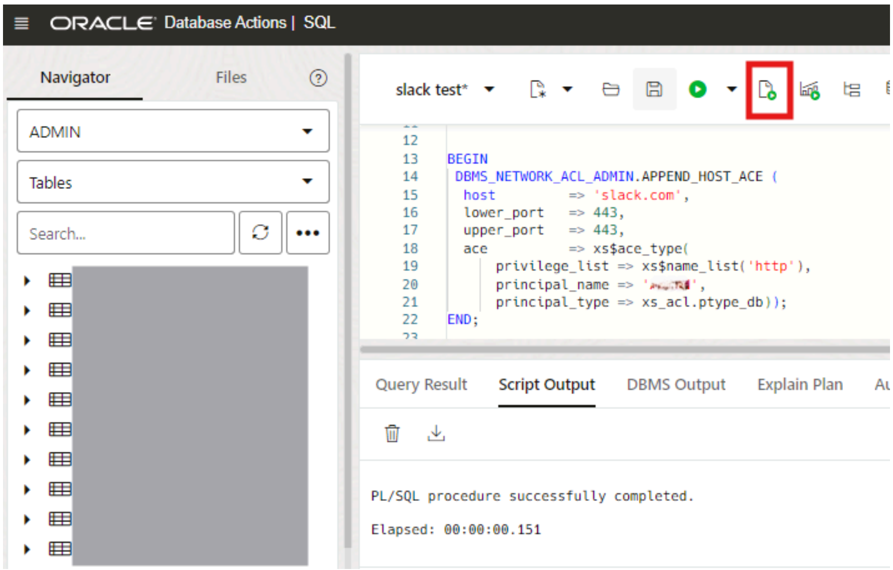
> **Note:** If you can not run the CREATE_CREDENTIAL procedure successfully, you can consult the ADMIN user to grant execute access on DBMS_CLOUD packages.

Task 6: Configure access control to allow user access to external network services

In this task, you will use DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE procedure to grant the access control privileges to a user. For more information, see Configuring Access Control for External Network Services.

1. In the source database SQL window, enter the following script, edit your **principal_name** according to your user name or user role, and then click **Run Statement**.

```
<copy> BEGIN    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE
```



```

BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE (
  host          => 'slack.com',
  lower_port    => 443,
  upper_port    => 443,
  ace           => xs$ace_type(
      privilege_list => xs$name_list('http'),
      principal_name => '[redacted]',
      principal_type => xs_acl.ptype_db));
END;
```

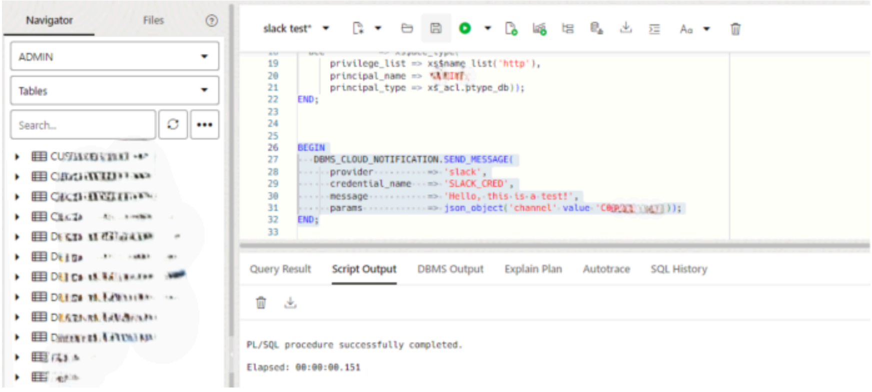
```
PL/SQL procedure successfully completed.

Elapsed: 00:00:00.151
```

Task 6: Send messages to a Slack channel

After creating required credential and configuration, you can use the DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE procedure to send a message to your Slack channel. For more information, see SEND_MESSAGE Procedure.

1. Copy and paste the following code into your SQL Worksheet, and enter your **Channel ID**, then click the **Run Script (F5)**.

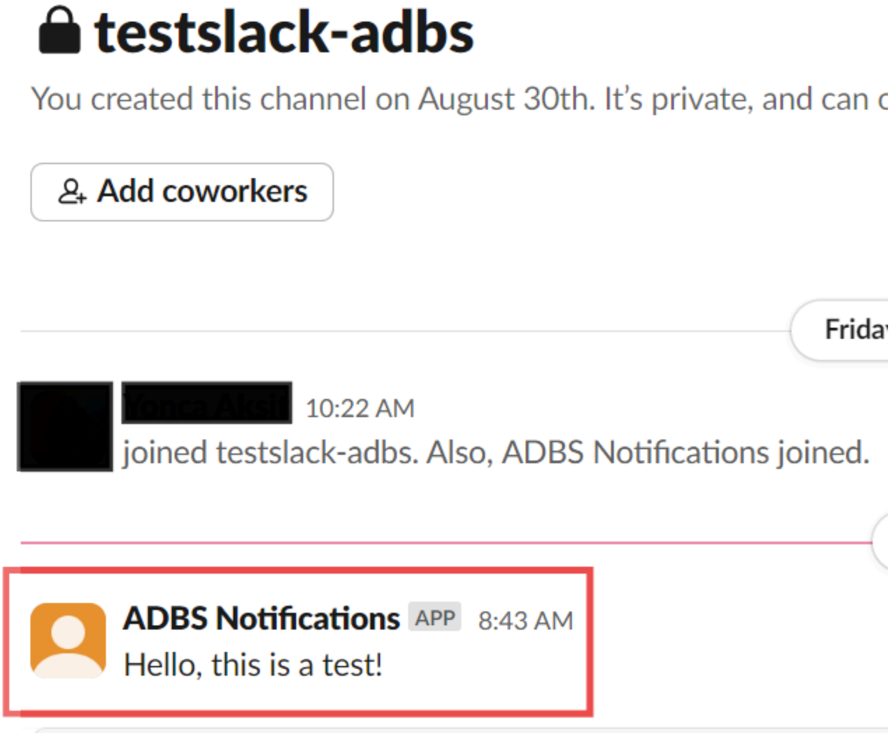
```
<copy> BEGIN    DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE(
```



```
        privilege_list => xs$name_list('http'),
        principal_name => [redacted]
        principal_type => xs_acl.ptype_db));
END;



BEGIN
  DBMS_CLOUD_NOTIFICATION.SEND_MESSAGE(
    provider        => 'slack',
    credential_name => 'SLACK_CRED',
    message         => 'Hello, this is a test!',
    params          => json_object('channel' value '[redacted]'));
END;
```

```
PL/SQL procedure successfully completed.

Elapsed: 00:00:00.151
```

2. Open your Slack channel, and confirm the receipt of the message.



testslack-adbs

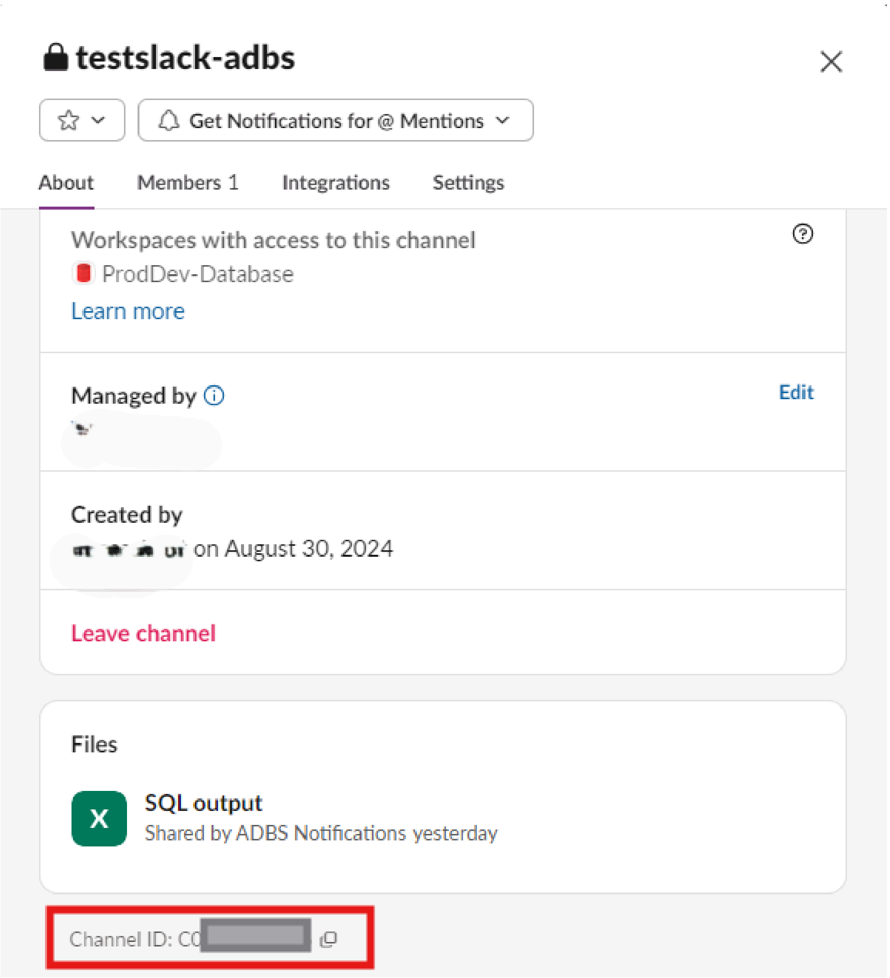
You created this channel on August 30th. It's private, and can c

[redacted] [redacted] 10:22 AM
joined testslack-adbs. Also, ADBS Notifications joined.

ADBS Notifications APP 8:43 AM
Hello, this is a test!

> **Note:** Use the **params** parameter to specify the Slack channel. The Channel ID is unique ID for a channel and is different from the channel name. You can find your channel ID at the bottom of the channel page.

3. Select your channel, click **Get Channel Details**, scroll down to the bottom, and copy the **Channel ID**.



testslack-adbs

Get Notifications for @ Mentions

About Members 1 Integrations Settings

Workspaces with access to this channel

ProdDev-Database

Learn more

Managed by

[redacted]

Edit

Created by

[redacted] on August 30, 2024

Leave channel

Files

SQL output

Shared by ADBS Notifications yesterday

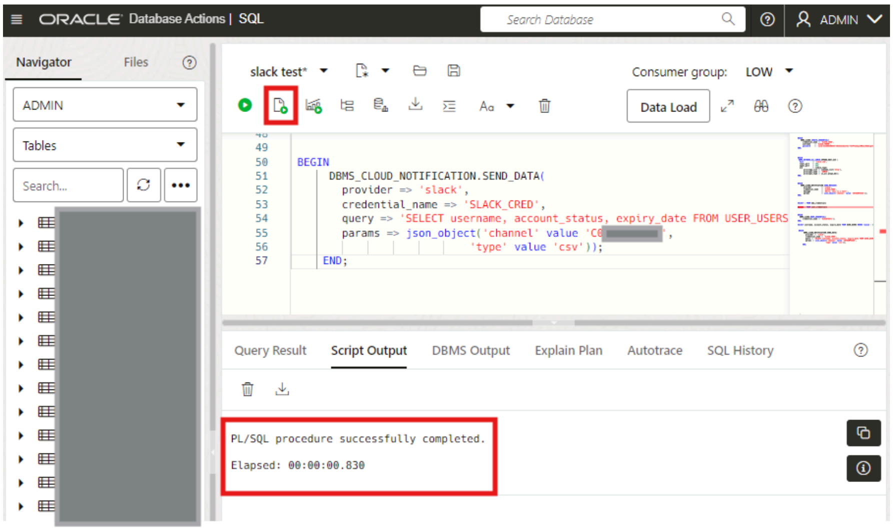
Channel ID: C0[redacted]

Task 7: Send query results to a Slack Channel

After creating required credential and configuration, you can use the DBMS_CLOUD_NOTIFICATION.SEND_DATA procedure to send the output of a query to your Slack channel. For more information, see SEND_DATA Procedure.

1. Copy your **Channel ID** from Task 6 (step 3).
2. Enter the following script, enter your **Channel ID** into **params** and then click **Run Statement**.

```
<copy> BEGIN    DBMS_CLOUD_NOTIFICATION.SEND_DATA(      p
```

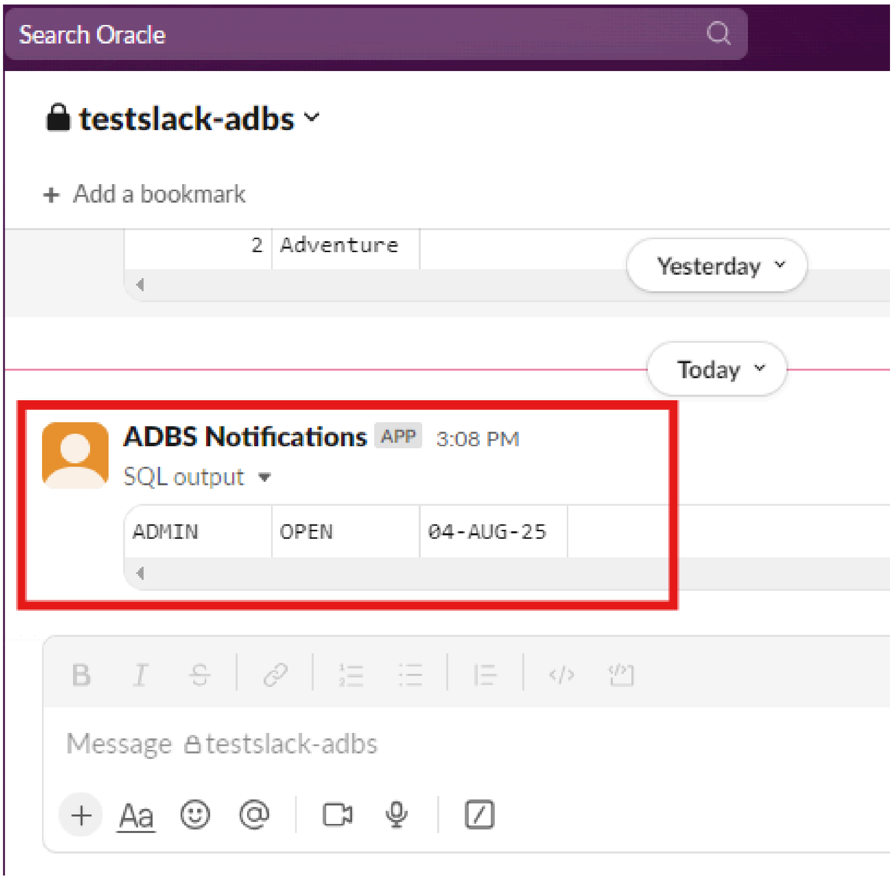


```
BEGIN
    DBMS_CLOUD_NOTIFICATION.SEND_DATA(
      provider => 'slack',
      credential_name => 'SLACK_CRED',
      query => 'SELECT username, account_status, expiry_date FROM USER_USERS
      params => json_object('channel' value 'C0[redacted]',
                            'type' value 'csv'));
   END;
```

```
PL/SQL procedure successfully completed.

Elapsed: 00:00:00.830
```

> **Note:** To specify the output type, use **params** section. Valid output types include `csv` and `json`.

3. Open your Slack channel, and confirm the receipt of query result.



2	Adventure

ADBS Notifications APP 3:08 PM

SQL output

ADMIN	OPEN	04-AUG-25

Summary

You learned how to create and configure a Slack app to receive messages and query results form Autonomous Database. Autonomous Database supports sending alerts,messages and query results directly to your channels to enhance productivity. Next, let's see how to use Microsoft Teams with Autonomous Database.

You may now **proceed to the next lab**.

Acknowledgements

- **Authors:**
    - Yonca Aksit, User Assistance Developer Intern
    - Nilay Panchal, ADB Product Management
    - Marty Gubar, Autonomous Database Product Management
- **Contributor:**
- **Last Updated By/Date:** Yonca Aksit, September 2024

Data about movies in this workshop were sourced from Wikipedia.