

CP4WatsonAIOps CP4WAIOPS v.3.4.0

Demo Environment Installation - Short Track 🚀



©2022 Niklaus Hirt / IBM

! THIS IS WORK IN PROGRESS

Please drop me a note on Slack or by mail nikh@ch.ibm.com if you find glitches or problems.

Installation

Demo Installation

Those are the steps that you have to execute to install a complete demo environment:

1. [AI Manager Installation](#)
2. [AI Manager Configuration](#)
3. [Slack integration](#)
4. [Demo the Solution](#)

! You can find a PDF version of this guide here: [PDF](#).

  **Here is a video that walks you through the complete [installation process](#).**

TLDR - Fast Track

These are the high level steps that you need to execute to install the demo environment

1. Install AI Manager
 1. Install directly from the OCP Web UI
 1. In the the OCP Web UI click on the + sign in the right upper corner
 2. Copy and paste the content from [this file](#)
 3. Replace `<REGISTRY_TOKEN>` at the end of the file with your pull token from step 1.3.1
 4. Click **Save**
 2. Install from your PC

```
ansible-playbook ./ansible/01_cp4waiops-aimanager-all.yaml -e  
CP_ENTITLEMENT_KEY=<REGISTRY_TOKEN>
```

1. [AI Manager Configuration](#)
2. [Slack integration](#)

Old documentation for reference

- Info
 - [Changelog](#)
 - [Demo Architecture](#)
 - [Detailed Prerequisites](#)
 - [Troubleshooting](#)
- Installation
 - [Event Manager Install](#)
 - [Event Manager Configuration](#)
 - [Uninstall CP4WAIOPS](#)
- Configuration
 - [Manual Runbook Configuration](#)
 - [Additional Configuration](#)
 - [Service Now integration](#)
- Install additional components
 - [Installing Turbonomic](#)
 - [Installing ELK](#)

1 Introduction

This document is a short version of the full [README](#) 🐛 that contains only the essential steps.

This is provided **as-is**:

- I'm sure there are errors
- I'm sure it's not complete
- It clearly can be improved

! This has been tested for the new CP4WAIOPS v.3.4.0 release on OpenShift 4.8 (4.10 not being available on Techzone yet) on ROKS

So please if you have any feedback contact me

- on Slack: @niklaushirt or
- by Mail: nikh@ch.ibm.com

1.1 Get the code

Clone the GitHub Repository

From IBM internal:

```
git clone https://<YOUR GIT TOKEN>@github.ibm.com/NIKH/cp4waiops-deployer.git
```

Or my external repo:

```
git clone https://github.com/niklaushirt/cp4waiops-deployer.git
```

1.2 Prerequisites

1.2.1 OpenShift requirements

I installed the demo in a ROKS environment.

You'll need:

- ROKS 4.8
- 5x worker nodes Flavor **b3c.16x64** (so 16 CPU / 64 GB)

You **might** get away with less if you don't install some components (Event Manager, ELK, Turbonomic,...) but no guarantee:

- Typically 4x worker nodes Flavor **b3c.16x64** *for only AI Manager*

1.2.2 Tooling

You need the following tools installed in order to follow through this guide (if you decide to install from your PC):

- ansible
- oc (4.7 or greater)
- jq
- kafkacat (only for training and debugging)
- elasticsearchdump (only for training and debugging)
- IBM cloudctl (only for LDAP)

1.2.1 On Mac - Automated (preferred)

Only needed if you decide to install from your PC

Just run:

```
./10_install_prerequisites_mac.sh
```

1.2.2 On Ubuntu - Automated (preferred)

Only needed if you decide to install from your PC

Just run:

```
./11_install_prerequisites_ubuntu.sh
```

1.3 Pull Secrets

1.3.1 Get the CP4WAIOPS installation token

You can get the installation (pull) token from <https://myibm.ibm.com/products-services/containerlibrary>.

This allows the CP4WAIOPS images to be pulled from the IBM Container Registry.

2 AI Manager Installation

You have different options:

1. **Install directly from the OCP Web UI** *(no need to install anything on your PC)*

1. In the the OCP Web UI click on the + sign in the right upper corner
2. Copy and paste the content from [this file](#)
3. Replace `<REGISTRY_TOKEN>` at the end of the file with your pull token from step 1.3.1
4. Click **Save**

2. **Install from your PC** *with the token from 1.3.1*

```
ansible-playbook ./ansible/01_cp4waiops-aimanager-all.yaml -e CP_ENTITLEMENT_KEY=
<REGISTRY_TOKEN>
```

3. **Install with the Easy Installer** *with the token from 1.3.1*

1. Just run:

```
./01_easy-install.sh -t <REGISTRY_TOKEN>
```

2. Select option  **01** to install the complete **AI Manager** environment with Demo Content.

This takes about 1.5 to 2 hours.

After completion Easy Installer will exit, open the documentation and the AI Manager webpage (on Mac) and you'll have to to perform the last manual steps.

You now have a full, basic installtion of AI Manager with:

- AI Manager
- Open LDAP & Register with AI Manager
- RobotShop demo application
- Trained Models based on precanned data (Log- and Metric Anomalies, Similar Incidents, Change Risk)
- Topologies for demo scenarios
- AWX (OpenSource Ansible Tower) with runbooks for the demo scenarios
- Demo UI
- Demo Service Account
- Creates valid certificate for Ingress (Slack)
- External Routes (Flink, Topology, ...)
- Create Policy Creation for Stories and Runbooks

3. AI Manager Configuration

Those are the manual configurations you'll need to demo the system and that are covered by the flow above.

Configure Topology

1. Re-Run Kubernetes Observer

Configure Slack

1. Setup Slack

3.1 First Login

After successful installation, the Playbook creates a file `./LOGINS.txt` in your installation directory (only if you installed from your PC).

i You can also run `./tools/20_get_logins.sh` at any moment. This will print out all the relevant passwords and credentials.

3.1.1 Login as admin

- Open the `LOGINS.txt` file that has been created by the Installer in your root directory

```
*****
*****
CloudPak for Watson AI0ps
*****
*****
AI Manager
*****
*****
AI Manager
URL: https://cpd-cp4waiops.itzroks-270003bu3k-q580lw-6ccd7f378ae819553d37d5f2ee142bd6-0000.eu-gb.containers.appdomain.cloud
User: demo
Password: P4ssw0rd!
User: admin
Password: XoFT1bfIu5Ng4EUJWEM7Mq8rIAi1QIHN
```

- Open the URL from the `LOGINS.txt` file
- Click on `IBM provided credentials (admin only)`

Log in to IBM Cloud Pak

Select your authentication type:

Enterprise LDAP

OpenShift authentication

IBM provided credentials (admin only)



- Login as **admin** with the password from the **LOGINS.txt** file

Log in to IBM Cloud Pak

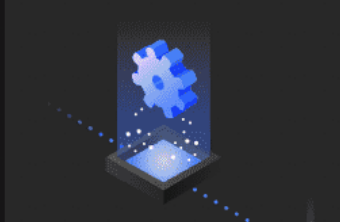
Enter your username and password.

Username

Password

Log in

[Change your authentication method](#)



3.1.2 Login as demo User

- Open the **LOGINS.txt** file that has been created by the Installer in your root directory

```
*****
CloudPak for Watson AI0ps
*****

-----
AI Manager
-----

AI Manager
URL: https://cpd-cp4waiops.itzroks-270003bu3k-q580lw-6ccd7f378ae819553d37d5f2ee142bd6-0000.eu-gb.containers.appdomain.cloud
User: demo
Password: P4ssw0rd!

User: admin
Password: XoFT1bfIu5Ng4EUJWEM7Mq8rIAi1QIHN
```

- Open the URL from the `LOGINS.txt` file
- Click on **Enterprise LDAP**
- Login as `demo` with the password `P4ssw0rd!`

3.2 Re-Run Kubernetes Integration

In the AI Manager (CP4WAIOPS)

1. In the **AI Manager** "Hamburger" Menu select **Define / Data and tool integrations**
2. Click **Kubernetes**
3. Under **robot-shop**, click on **Run** (with the small play button)

4 Event Manager Installation

You have different options:

1. **Install directly from the OCP Web UI** *(no need to install anything on your PC)*

1. In the the OCP Web UI click on the + sign in the right upper corner
2. Copy and paste the content from [this file](#)
3. Replace `<REGISTRY_TOKEN>` at the end of the file with your pull token from step 1.3.1
4. Click **Save**

2. **Install from your PC** *with the token from 1.3.1*

```
ansible-playbook ./ansible/04_cp4waiops-eventmanager-all.yaml -e  
CP_ENTITLEMENT_KEY=<REGISTRY_TOKEN>
```

3. **Install with the Easy Installer** *with the token from 1.3.1*

1. Just run:

```
./01_easy-install.sh -t <REGISTRY_TOKEN>
```

2. Select option  **02** to install the complete **Event Manager** environment with Demo Content.

This takes about 1 hour.

5. Event Manager Configuration

5.1 First Login

After successful installation, the Playbook creates a file `./LOGINS.txt` in your installation directory (only if you installed from your PC).

i You can also run `./tools/20_get_logins.sh` at any moment. This will print out all the relevant passwords and credentials.

5.1.1 Login as smadmin

- Open the `LOGINS.txt` file that has been created by the Installer in your root directory
- Open the URL from the `LOGINS.txt` file
- Login as `smadmin` with the password from the `LOGINS.txt` file

5.2 Topology

5.2.1 Create Kubernetes Observer for the Demo Applications

This is basically the same as for AI Manager as we need two separate instances of the Topology Manager.

- In the **Event Manager** "Hamburger" Menu select **Administration / Topology Management**
- Under **Observer jobs** click **Configure**
- Click **Add new job**
- Under **Kubernetes**, click on **Configure**
- Choose **local** for **Connection Type**
- Set **Unique ID** to **robot-shop**
- Set **data_center** to **robot-shop**
- Under **Additional Parameters**
- Set **Terminated pods** to **true**
- Set **Correlate** to **true**
- Set Namespace to **robot-shop**
- Under **Job Schedule**
- Set **Time Interval** to 5 Minutes
- Click **Save**

5.3 EventManager Webhook

Create Webhooks in EventManager for Event injection and incident simulation for the Demo.

The demo scripts (in the `demo` folder) give you the possibility to simulate an outage without relying on the integrations with other systems.

At this time it simulates:

- Git push event
- Log Events (ELK)
- Security Events (Falco)
- Instana Events
- Metric Manager Events (Predictive)
- Turbonomic Events
- CP4MCM Synthetic Selenium Test Events

You have to define the following Webhook in EventManager (NOI):

- **Administration / Integration with other Systems**
- **Incoming / New Integration**
- **Webhook**
- Name it **Demo Generic**
- Jot down the WebHook URL and copy it to the **NETCOOL_WEBHOOK_GENERIC** in the **./tools/01_demo/incident_robotshop-noi.sh** file
- Click on **Optional event attributes**
- Scroll down and click on the + sign for **URL**
- Click **Confirm Selections**

Use this json:

```
{
  "timestamp": "1619706828000",
  "severity": "Critical",
  "summary": "Test Event",
  "nodename": "productpage-v1",
  "alertgroup": "robotshop",
  "url": "https://pirsoscom.github.io/grafana-robotshop.html"
}
```

Fill out the following fields and save:

- Severity: **severity**
- Summary: **summary**
- Resource name: **nodename**
- Event type: **alertgroup**
- Url: **url**
- Description: **"URL"**

Optionnally you can also add **Expiry Time** from **Optional event attributes** and set it to a convenient number of seconds (just make sure that you have time to run the demo before they expire).

5.4 Create custom Filters and Views

5.4.1 Filter

- In the **Event Manager** "Hamburger" Menu select **Netcool WebGui**
- Click **Administration**
- Click **Filters**
- Select **Global Filters** from the DropDown menu
- Select **Default**
- Click **Copy Filter** (the two papers on the top left)
- Set to **global**
- Click **Ok**
- Name: ALOPS
- Logic: **Any !** (the right hand option)
- Filter:
 - AlertGroup = 'CEACorrelationKeyParent'
 - AlertGroup = 'robot-shop'

Edit Filter: New Filter

Filter Attributes

* Name:

Default view:

Collection:

Description:

Data Source: [Click to show](#)

Filter Conditions

Basic Advanced Dependent Metric

☐ All ☒ Any

Field	Comparator	Value	
<input type="text" value="AlertGroup"/>	<input type="text" value="="/>	<input type="text" value="'CEACorrelationKeyParent'"/>	<input type="button" value="-"/> <input type="button" value="+"/>
<input type="text" value="AlertGroup"/>	<input type="text"/>	<input type="text" value="'robot-shop'"/>	<input type="button" value="-"/> <input type="button" value="+"/>

Clear All

5.4.2 View

- In the **Event Manager** "Hamburger" Menu select **Netcool WebGui**
- Click **Administration**
- Click **Views**
- Select **System Views** from the DropDown menu
- Select **Example_IBM_CloudAnalytics**
- Click **Copy View** (the two papers on the top left)
- Set to **global**
- Click **Ok**
- Name: AIOPS
- Configure to your likings.

5.5 Create grouping Policy

- In the **Event Manager** "Hamburger" Menu select **Netcool WebGui**
- Click **Insights**
- Click **Scope Based Grouping**
- Click **Create Policy**
- **Action** select field **Alert Group**
- Toggle **Enabled** to **On**
- Save

5.6 Create Menu item

In the Netcool WebGUI

- Go to **Administration** / **Tool Configuration**
- Click on **LaunchRunbook**
- Copy it (the middle button with the two sheets)
- Name it **Launch URL**
- Replace the Script Command with the following code

```
var urlId = '{$selected_rows.URL}';

if (urlId == '') {
    alert('This event is not linked to an URL');
} else {
    var wnd = window.open(urlId, '_blank');
}
```

- Save

Then

- Go to **Administration** / **Menu Configuration**
- Select **alerts**
- Click on **Modify**
- Move Launch URL to the right column
- Save

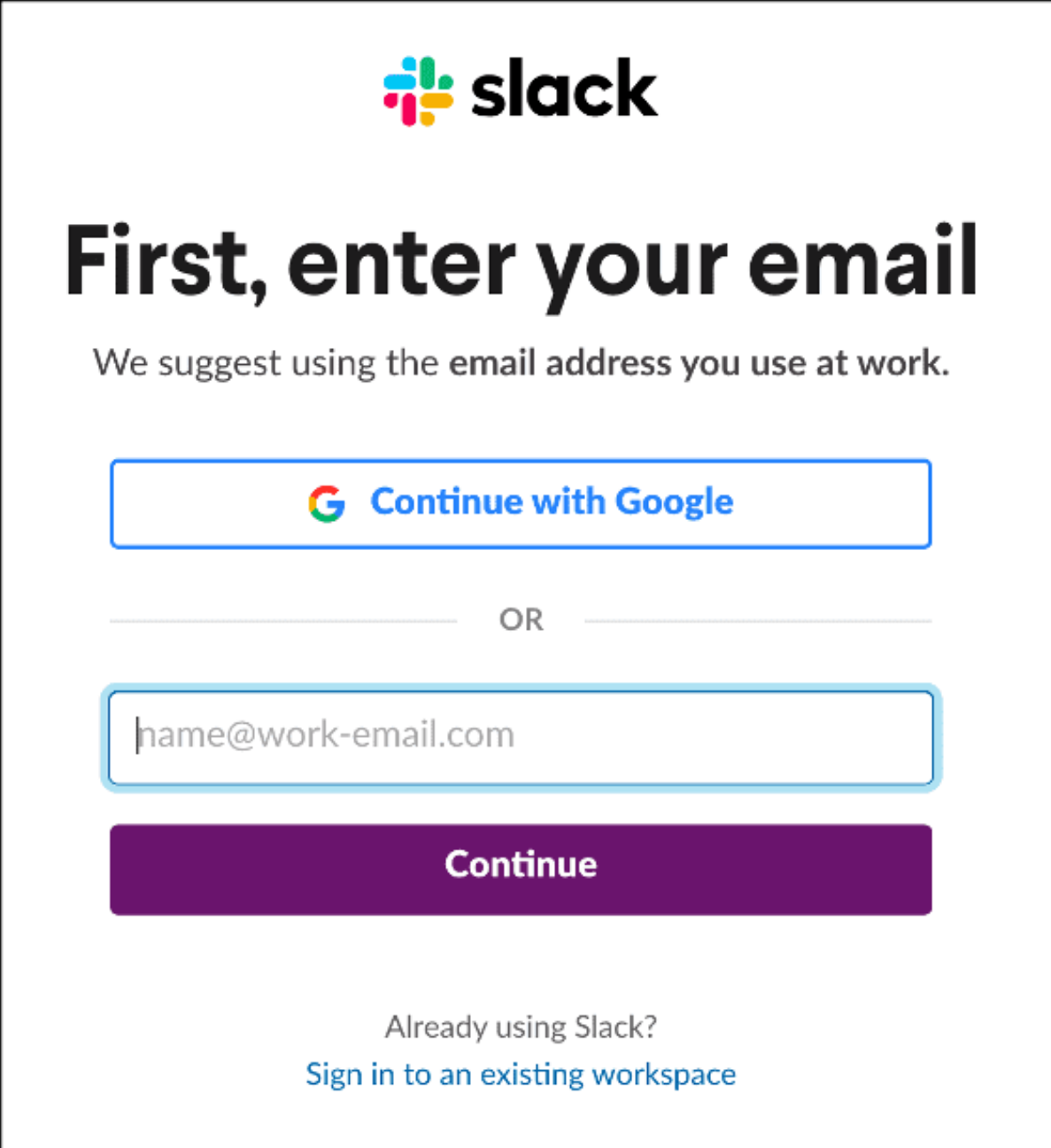
6. Slack integration

For the system to work you need to follow those steps:

1. Create Slack Workspace
2. Create Slack App
3. Create Slack Channels
4. Create Slack Integration
5. Get the Integration URL
6. Create Slack App Communications
7. Slack Reset

6.1 Create your Slack Workspace

1. Create a Slack workspace by going to <https://slack.com/get-started#/createnew> and logging in with an email **which is not your IBM email**. Your IBM email is part of the IBM Slack enterprise account and you will not be able to create an independent Slack workspace outside of the IBM slack service.



The image shows the Slack 'First, enter your email' login screen. At the top is the Slack logo. Below it is the heading 'First, enter your email' in a large, bold font. Underneath the heading is a suggestion: 'We suggest using the email address you use at work.' There are two main options for login. The first is a button with the Google 'G' logo and the text 'Continue with Google'. Below this is a horizontal line with the word 'OR' in the center. The second option is a text input field containing the placeholder text 'name@work-email.com'. Below the input field is a large purple button with the text 'Continue'. At the bottom of the screen, there is a link that says 'Already using Slack? Sign in to an existing workspace'.

2. After authentication, you will see the following screen:



Create a new Slack workspace

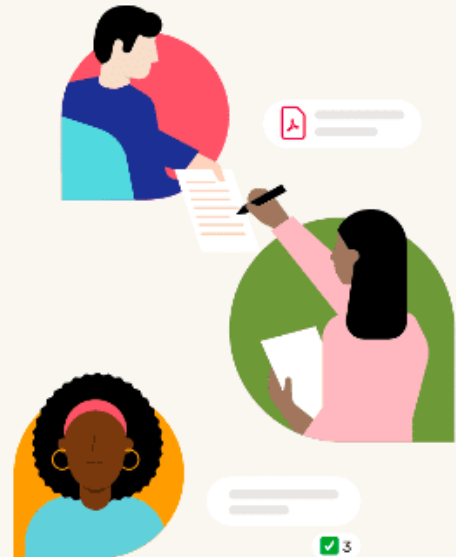
Slack gives your team a home — a place where they can talk and work together. To create a new workspace, click the button below.

Tip: Use the email you use for work. That makes it easy to get the rest of your team on Slack. [Change email](#)

Create a Workspace →

☒ It's okay to send me emails about Slack.

By continuing, you're agreeing to our [Customer Terms of Service](#), [Privacy Policy](#), and [Cookie Policy](#).



3. Click **Create a Workspace** ->
4. Name your Slack workspace

Step 1 of 3

What's the name of your company or team?

This will be the name of your Slack workspace — choose something that your team will recognize.

|Ex: Acme Marketing or Acme Co

255

Next

Give your workspace a unique name such as aiops-<yourname>.

5. Describe the workspace current purpose

Step 2 of 3

What's your team working on right now?

This could be anything: a project, campaign, event, or the deal you're trying to close.

Ex: Q4 budget, autumn campaign

80

Next

This is free text, you may simply write "demo for Watson AIOps" or whatever you like.

6.

Step 3 of 3

Who do you email most about demo-environment?

To give Slack a spin, add a few coworkers you talk with regularly.

Ex. ellis@gmail.com

+ Add another

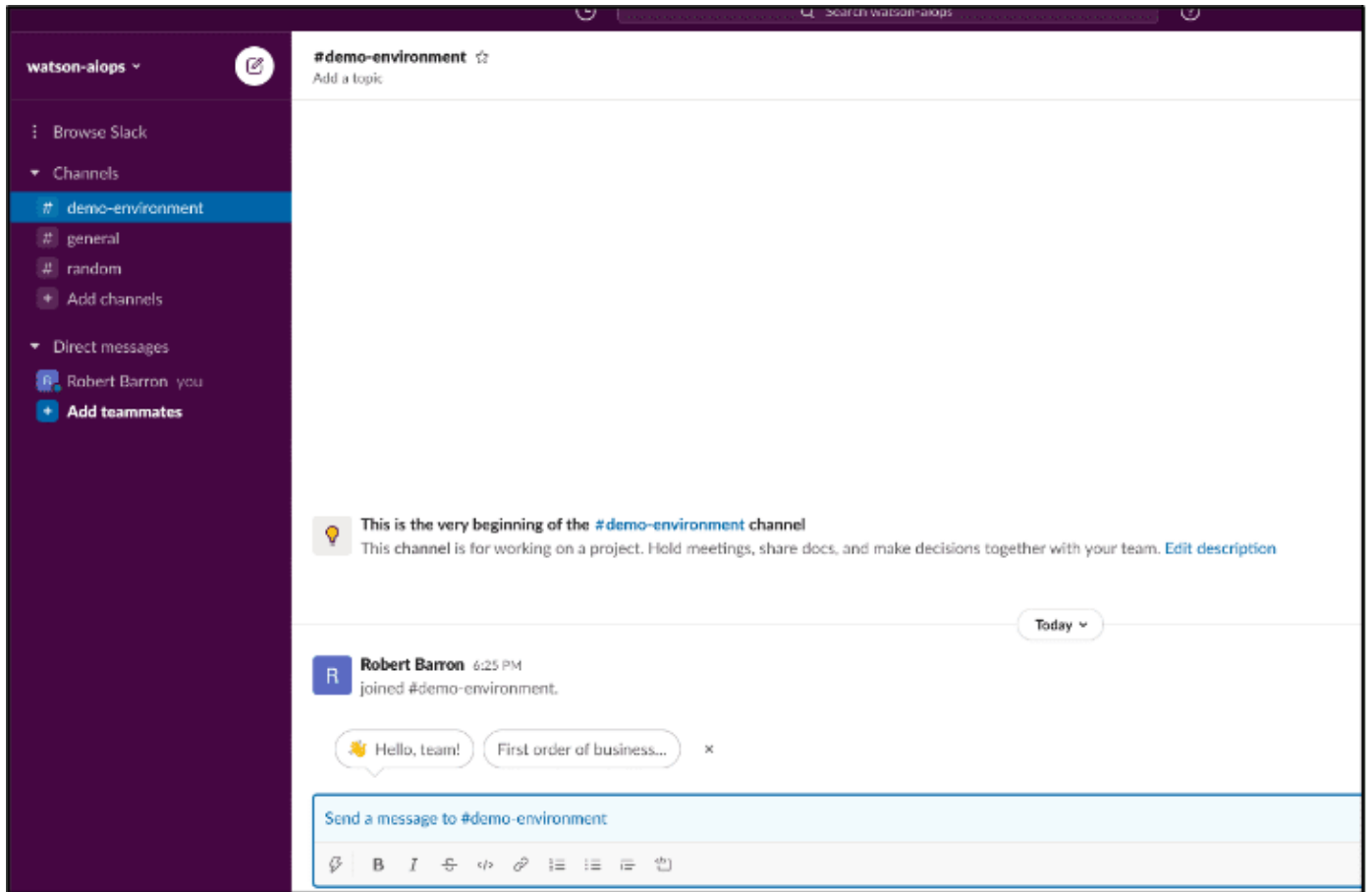
Get a shareable invite link instead

Add Teammates

Skip this step

You may add team members to your new Slack workspace or skip this step.

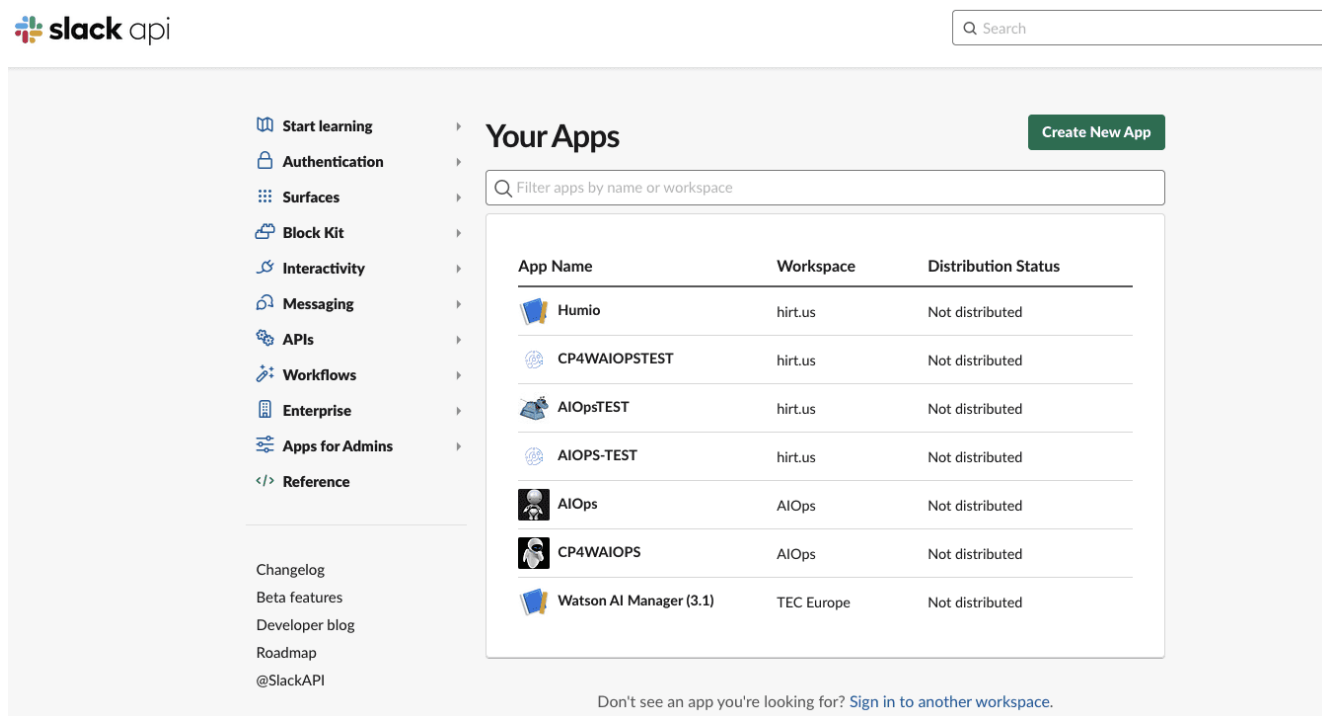
At this point you have created your own Slack workspace where you are the administrator and can perform all the necessary steps to integrate with CP4WAOps.



Note : This Slack workspace is outside the control of IBM and must be treated as a completely public environment. Do not place any confidential material in this Slack workspace.

6.2 Create Your Slack App

1. Create a Slack app, by going to <https://api.slack.com/apps> and clicking **Create New App**.



2. Select **From an app manifest**

Create an app

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

From scratch

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

From an app manifest **BETA**

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

Need help? Check our [documentation](#), or [see an example](#)

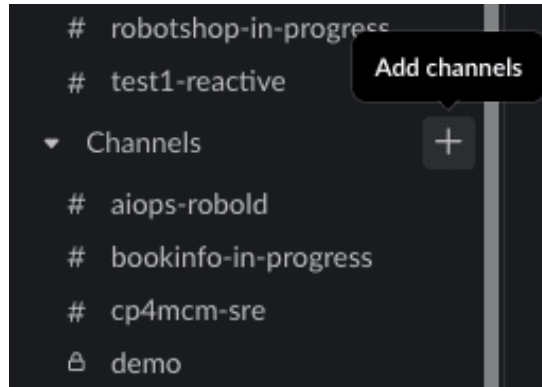
3. Select the appropriate workspace that you have created before and click **Next**
4. Copy and paste the content of this file [./doc/slack/slack-app-manifest.yaml](#).
Don't bother with the URLs just yet, we will adapt them as needed.
5. Click **Next**

6. Click **Create**
7. Scroll down to Display Information and name your CP4WAIOPS app.
8. You can add an icon to the app (there are some sample icons in the `./tools/4_integrations/slack/icons` folder).
9. Click save changes
10. In the **Basic Information** menu click on **Install to Workspace** then click **Allow**

6.3 Create Your Slack Channels

1. In Slack add a two new channels:

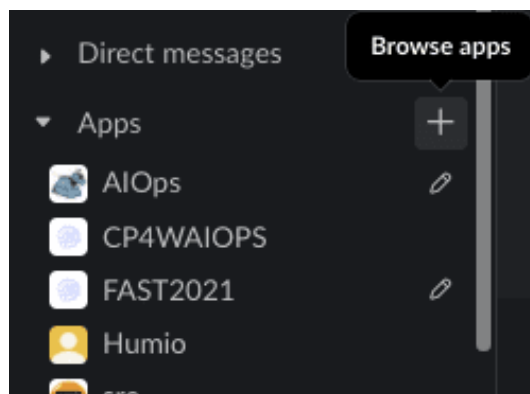
- aiops-demo-reactive
- aiops-demo-proactive



2. Right click on each channel and select **Copy Link**

This should get you something like this <https://xxxx.slack.com/archives/C021QOY16BW>
The last part of the URL is the channel ID (i.e. C021QOY16BW)
Jot them down for both channels

3. Under Apps click Browse Apps



4. Select the App you just have created

5. Invite the Application to each of the two channels by typing

@<MyAppname>

6. Select **Add to channel**

You should get a message from saying **was added to #<your-channel> by ...**

6.4 Integrate Your Slack App

In the Slack App:

1. In the **Basic Information** menu get the **Signing Secret** (not the Client Secret!) and jot it down

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

App ID	Date of App Creation
A02MJTPE7M2	November 16, 2021

Client ID

1624757694871.2732941483716

Client Secret

..... **Show** **Regenerate**

You'll need to send this secret along with your client ID when making your [oauth.v2.access](#) request.

Signing Secret

a117e0170bbef1e017c11bc07dc0512 **Show** **Regenerate**

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token

woBhrC5m0IZg2X0CgfShrTLV **Regenerate**

This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.

2. In the **OAuth & Permissions** get the **Bot User OAuth Token** (not the User OAuth Token!) and jot it down

CP4WAIOPSTE... ▾

Settings

Basic Information

Collaborators

Socket Mode

Install App

Manage Distribution

Features

App Home

Org Level Apps

Incoming Webhooks

Interactivity & Shortcuts

Slash Commands

Workflow Steps

OAuth & Permissions

Event Subscriptions

User ID Translation

App Manifest NEW

Beta Features

Submit to App Directory

Review & Submit


Give feedback

Slack ❤️

OAuth & Permissions

Advanced token security via token rotation

Recommended for developers building on or for security-minded organizations – opting into token rotation allows app tokens to automatically expire after they're issued within your app code. [View documentation.](#)

 At least one redirect URL needs to be set below before this app can be opted into token rotation

Opt in

OAuth Tokens for Your Workspace

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. [Learn more.](#)

User OAuth Token

xoxp-1624757694871-1639736885955-2723982398998-593c61defc81d2a8 Copy

Access Level: Workspace

Bot User OAuth Token

Copy

Access Level: Workspace

Reinstall to Workspace

In the AI Manager (CP4WAIOPS)

1. In the **AI Manager** "Hamburger" Menu select **Define / Data and tool integrations**
2. Click **Add connection**

IBM Cloud Pak | Automation

Data and tool connections

Connect to your tools to provide data that will help gather insights for your environment.

Add connection

Learn more

Show details ▾

Manage connections

Schedule connections

Q Filter table

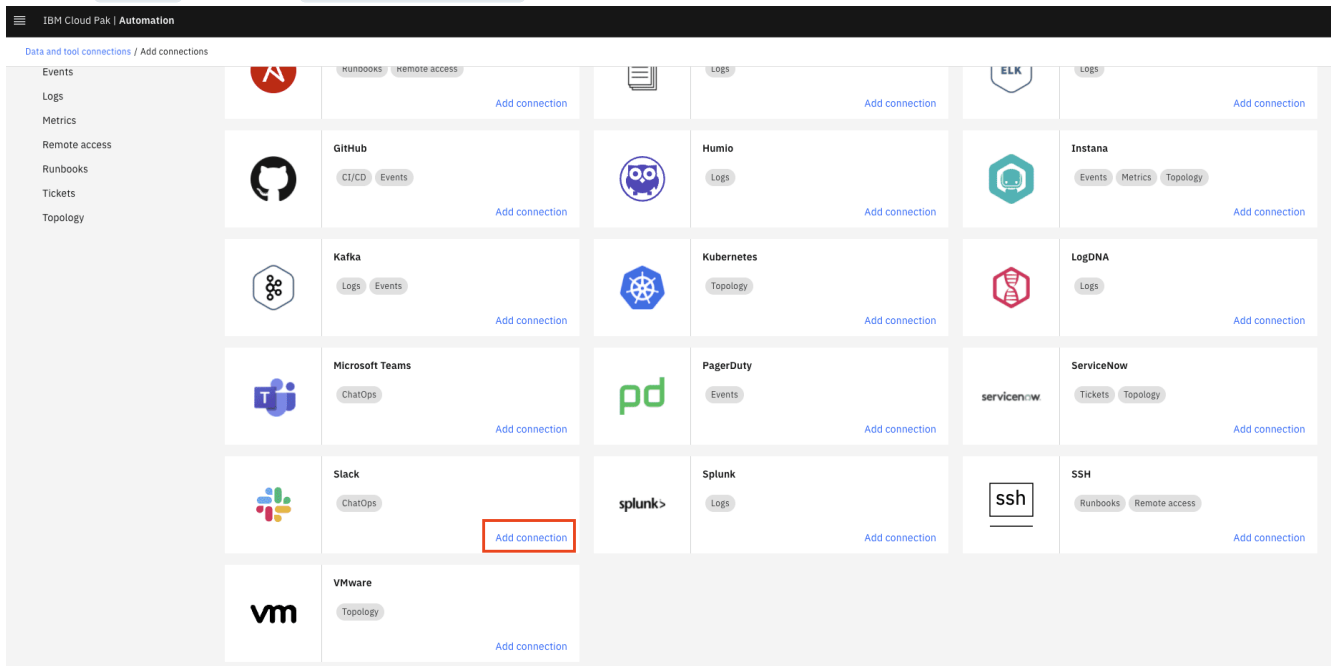
Connection type	Total connections	Connection status	Categories
<div><div></div>Start by adding a connection Click the Add connection button to get started</div>			

Items per page: 10 ▾

0-0 of 0 items

1 ▾ of 1 page

3. Under **Slack**, click on **Add Connection**



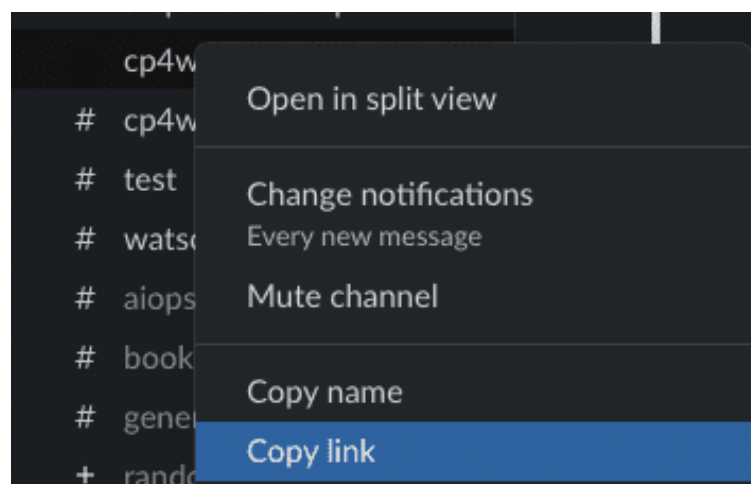
4. Name it "Slack"

5. Paste the **Signing Secret** from above

6. Paste the **Bot User OAuth Token** from above

Slack

7. Paste the channel IDs from the channel creation step in the respective fields



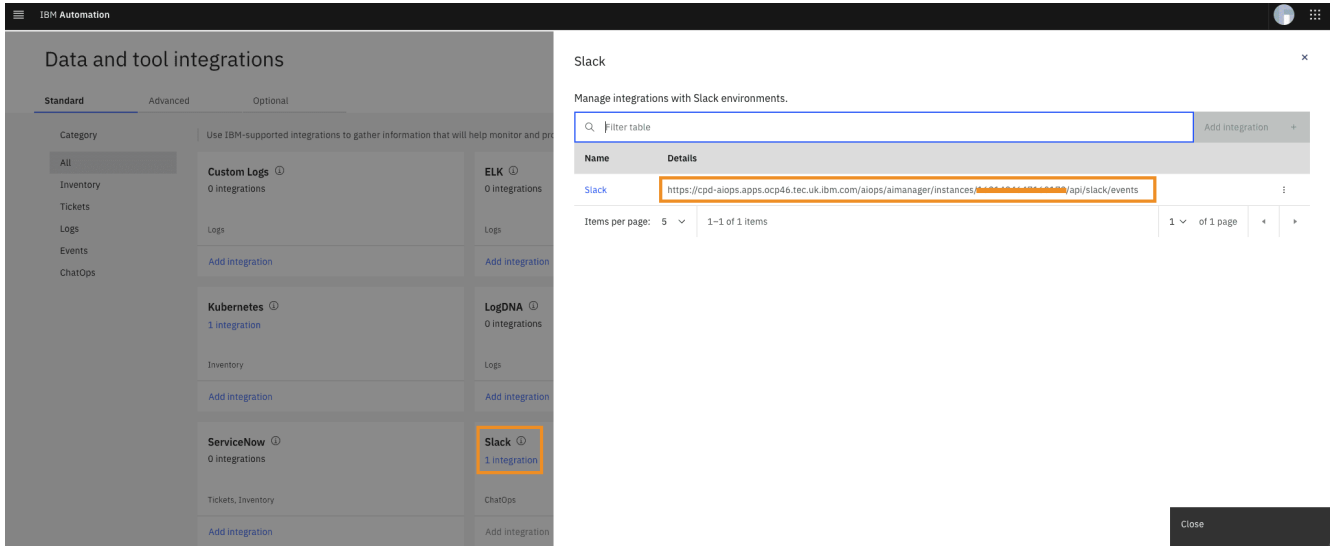
Proactive channel	
C035U4CONCV	
Enter a proactive channel for the connection.	
Reactive channel	
C035U4CAAA	
Enter a reactive channel for the connection.	
Test connection to slack.com	☑ Test succeeded

8. Test the connection and click save

6.5 Create the Integration URL

In the AI Manager (CP4WAIOPS)

- 1. Go to **Data and tool integrations**
- 2. Under **Slack** click on **1 integration**
- 3. Copy out the URL



This is the URL you will be using for step 6.

6.6 Create Slack App Communications

Return to the browser tab for the Slack app.

6.6.1 Event Subscriptions

1. Select **Event Subscriptions**.
2. In the **Enable Events** section, click the slider to enable events.
3. For the Request URL field use the **Request URL** from step 5.

e.g: `https://<my-url>/aiops/aimanager/instances/xxxxx/api/slack/events`

4. After pasting the value in the field, a *Verified* message should display.

Enable Events On

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more.](#)

Request URL Verified ✓

Change



We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a **challenge** parameter, and your endpoint must respond with the challenge value. [Learn more.](#)

If you get an error please check 5.7

5. Verify that on the **Subscribe to bot events** section you got:
 - **app_mention** and
 - **member_joined_channel** events.

Subscribe to bot events

Apps can subscribe to receive events the bot user has access to (like new messages in a channel). If you add an event here, we'll add the necessary [OAuth scope](#) for you.

Event Name	Description	Required Scope
app_mention	Subscribe to only the message events that mention your app or bot	app_mentions:read 
member_joined_channel	A user joined a public or private channel	channels:read or groups:read 

Add Bot User Event

- Click [Save Changes](#) button.

6.6.2 Interactivity & Shortcuts

- Select [Interactivity & Shortcuts](#).
- In the Interactivity section, click the slider to enable interactivity. For the [Request URL](#) field, use the URL from above.

There is no automatic verification for this form

Interactivity & Shortcuts

Interactivity

On 

Any interactions with shortcuts, modals, or interactive components (such as buttons, select menus, and datepickers) will be sent to a URL you specify. [Learn more.](#)

Request URL

[https://](#)  0000.t

Slack will send an HTTP POST request with information to this URL when users interact with a shortcut or interactive component.

- Click [Save Changes](#) button.

6.6.3 Slash Commands

Now, configure the **welcome** slash command. With this command, you can trigger the welcome message again if you closed it.

1. Select **Slash Commands**
2. Click **Create New Command** to create a new slash command.

Use the following values:

Field	Value
Command	/welcome
Request URL	the URL from above
Short Description	Welcome to Watson AIOps

3. Click **Save**.

6.6.4 Reinstall App

The Slack app must be reinstalled, as several permissions have changed.

1. Select **Install App**
2. Click **Reinstall to Workspace**

Once the workspace request is approved, the Slack integration is complete.

If you run into problems validating the **Event Subscription** in the Slack Application, see 5.2

6.7 Slack Reset

6.7.1 Get the User OAuth Token

This is needed for the reset scripts in order to empty/reset the Slack channels.

This is based on [Slack Cleaner2](#).

You might have to install this:

```
pip3 install slack-cleaner2
```

Reset reactive channel

In your Slack app

1. In the **OAuth & Permissions** get the **User OAuth Token** (not the Bot User OAuth Token this time!) and jot it down

In file `./tools/98_reset/13_reset-slack.sh`

2. Replace **not_configured** for the **SLACK_TOKEN** parameter with the token
3. Adapt the channel name for the **SLACK_REACTIVE** parameter

Reset proactive channel

In your Slack app

1. In the **OAuth & Permissions** get the **User OAuth Token** (not the Bot User OAuth Token this time!) and jot it down (same token as above)

In file `./tools/98_reset/14_reset-slack-changerisk.sh`

2. Replace **not_configured** for the **SLACK_TOKEN** parameter with the token
3. Adapt the channel name for the **SLACK_PROACTIVE** parameter

6.7.2 Perform Slack Reset

Call either of the scripts above to reset the channel:

```
./tools/98_reset/13_reset-slack.sh  
  
or  
  
./tools/98_reset/14_reset-slack-changerisk.sh
```

7. Demo the Solution



7.1 Simulate incident - Command Line

Make sure you are logged-in to the Kubernetes Cluster first

In the terminal type

```
./22_simulate_incident_robotshop.sh
```

This will delete all existing Alerts/Stories and inject pre-canned event, metrics and logs to create a story.

-  Give it a minute or two for all events and anomalies to arrive in Slack.
-  You might have to run the script 3-4 times for the log anomalies to start appearing.