

# CP4WatsonAIOps CP4WAIOPS v3.5.0

Demo Environment Installation 🚀



©2022 Niklaus Hirt / IBM

# **! THIS IS WORK IN PROGRESS**

Please drop me a note on Slack or by mail [nikh@ch.ibm.com](mailto:nikh@ch.ibm.com) if you find glitches or problems.

# Installation

## 🚀 Demo Installation

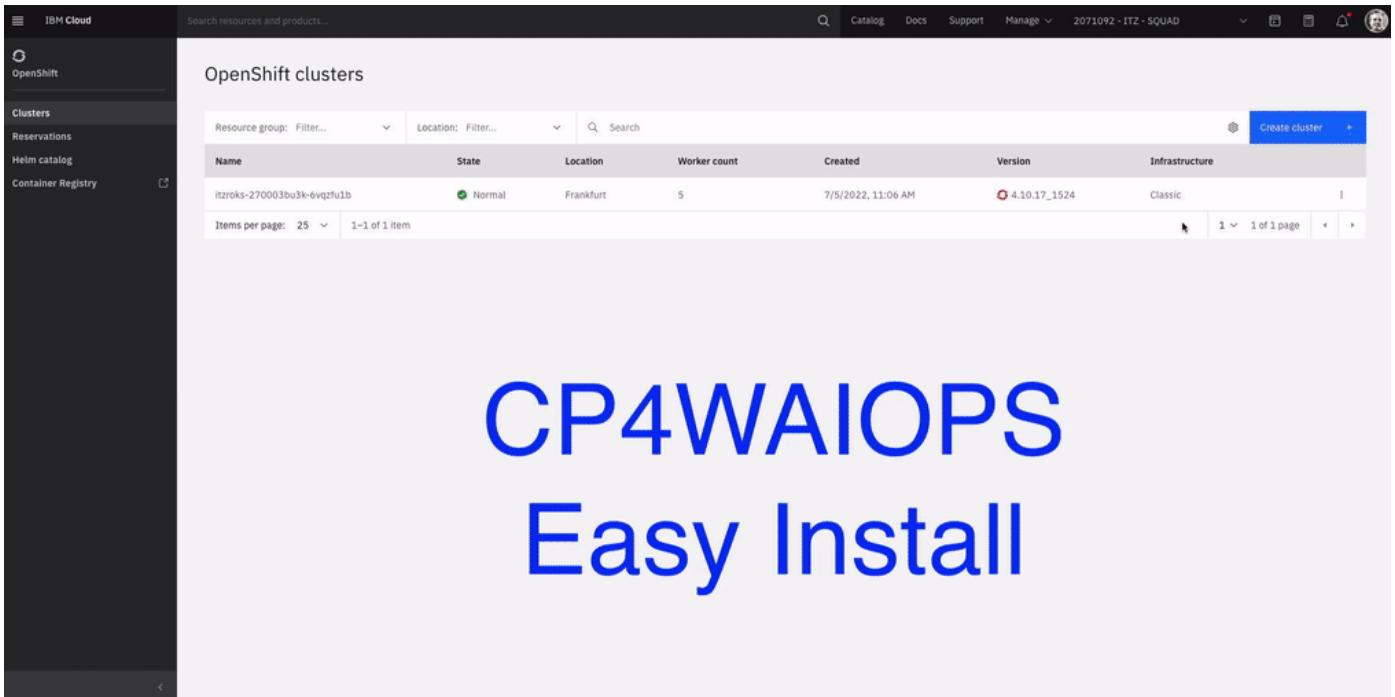
Those are the steps that you have to execute to install a complete demo environment:

1. [TLDR - Fast Track](#)
2. [AI Manager Installation](#)
3. [AI Manager Configuration](#)
4. [Event Manager Installation](#)
5. [Event Manager Configuration](#)
6. [Slack integration](#)
7. [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



The screenshot shows the IBM Cloud OpenShift console interface. On the left, there's a sidebar with 'OpenShift' selected. The main area is titled 'OpenShift clusters' and displays a table with one row. The table columns are: Name, State, Location, Worker count, Created, Version, and Infrastructure. The single cluster listed is 'itzroks-270003bu3k-6vqzf1b', which is 'Normal' and located in 'Frankfurt' with a worker count of 5. It was created on '7/5/2022, 11:06 AM' and has version '4.10.17\_1524'. The infrastructure type is 'Classic'. At the bottom of the table, it says '1 of 1 page'.

# CP4WAIOPS Easy Install

These are the options you have to install the demo environment.

## 1. Install AI Manager with demo content

1. In 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 (the Entitlement key from <https://myibm.ibm.com>)

#### 4. Click **Save**

The screenshot shows the Red Hat OpenShift Container Platform web interface. The left sidebar shows various project options like Home, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, User Management, and Administration. The main area is titled 'Import YAML' and contains a text editor with a large amount of YAML code. The code includes sections for cloning a repository, installing Ansible collections, and setting environment variables. A red box labeled '1' points to the '+' icon in the top right corner. A red box labeled '2' points to the 'Import YAML' section. A red box labeled '3' points to the 'env' section of the YAML code. A red box labeled '4' points to the 'Create' button at the bottom left.

## 2. Install Event Manager with demo content

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 (the Entitlement key from <https://myibm.ibm.com>)
4. Click **Save**

## 3. Install AI Manager, Event Manager and Turbonomic with demo content

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 (the Entitlement key from <https://myibm.ibm.com>)
4. Click **Save**

## 4. Install other components

1. In the OCP Web UI click on the + sign in the right upper corner
2. Select the content file from [this directory](#)
3. Replace **<REGISTRY\_TOKEN>** at the end of the file with your pull token from step 1.3.1 (the Entitlement key from <https://myibm.ibm.com>)
4. Click **Save**

 If you get a ClusterRoleBinding already exists, just delete it at the beginning of the YAML

---

# 1 Introduction

---

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 v3.5.0 release on OpenShift 4.8 and 4.10 on ROKS**

So please if you have any feedback contact me

- on Slack: @niklaushirt or
- by Mail: [nikh@ch.ibm.com](mailto:nikh@ch.ibm.com)

## 1.1 Get the code

Clone the GitHub Repository

```
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 or 4.10
- 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 Get a ROKS Cluster (IBMer only)

IBMer can get a temporary one from [Techzone](#) (usually valid for 2 weeks)

1. Create a cluster for **Practice/Self Education** if you don't have an Opportunity Number

The screenshot shows the 'Create a reservation' interface for 'Custom ROKS requests'. At the top, there's a navigation bar with tabs: 'Select a environment/infrastructure', 'Select a reservation type', 'Fill out your reservation', and 'Complete'. Below the tabs, there's a 'Name' field containing 'IBM RedHat Openshift Kubernetes Service (ROKS)'. A note says 'Name this reservation. This will help identify it in your reservation list.' Under 'Purpose', 'Practice / Self-Education' is selected. In the 'Customer name(s)' section, there's a field for 'Enter a customer name' and another for 'Enter an opportunity number(s)'. A note says 'Providing an IBM Sales Cloud opportunity.number or a GainSite Relationship ID will allow you to extend your reservation date.' In the 'Purpose description' section, there's a text area with 'Test3.4'. At the bottom, a note asks 'What are you doing? Why do you need this? What are you trying to accomplish?' On the right side, there's a sidebar titled 'Collection: Custom ROKS requests' with a note about requesting custom Managed OpenShift clusters. It also lists 'Environment: IBM RedHat Openshift Kubernetes Service (ROKS)' and 'Reservation policy: test\_self-education'.

2. Select the maximum end date that fits your needs (2 weeks maximum)

The screenshot shows a date and time selection interface. On the left is a calendar for July 2022, with the 19th highlighted. On the right is a time picker showing '9:50 AM' in 'Europe/Paris'.

3. Fill-in the remaining fields

1. Geography: whatever is closest to you
2. Worker node count: 5
3. Flavour: b3c.16x64
4. OpenShift Version: 4.10

Preferred Geography

London 5

End date and time

Select a date Select a time

07/19/2022 9:50 AM Europe/Paris

Available for up to 2 weeks (336 hours)

Worker Node Count (required)

5

Worker Node Flavor (required)

b3c.16x64 (16 vCPU x 64GB - 100GB Secondary Storage)

NFS Size (required)

None

OpenShift Version (required)

4.10

Notes

Enter any notes you would like to attach to this reservation

Cancel Reset Submit

4. Click **Submit**

## 1.2.3 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)
- elasticsearch (only for training and debugging)
- IBM cloudctl (only for LDAP)

### 1.2.3.1 On Mac - Automated (preferred)

*Only needed if you decide to install from your PC*

Just run:

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

### **1.2.3.2 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 (registry pull token)

You can get the installation (registry pull token) 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 (the Entitlement key from <https://myibm.ibm.com>)
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 open the documentation and the AI Manager webpage (on Mac) and you'll have to perform the last manual steps.

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

- AI Manager
- Open LDAP & Register with AI Manager
- RobotShop demo application
- Trained Models based on pre-canned 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
- Valid certificate for Ingress (Slack)
- External Routes (Flink, Topology, ...)
- Policies 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).

 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 Get the URL

- Run `./tools/20_get_logins.sh` to get all the logins and URLs

or

- Run:

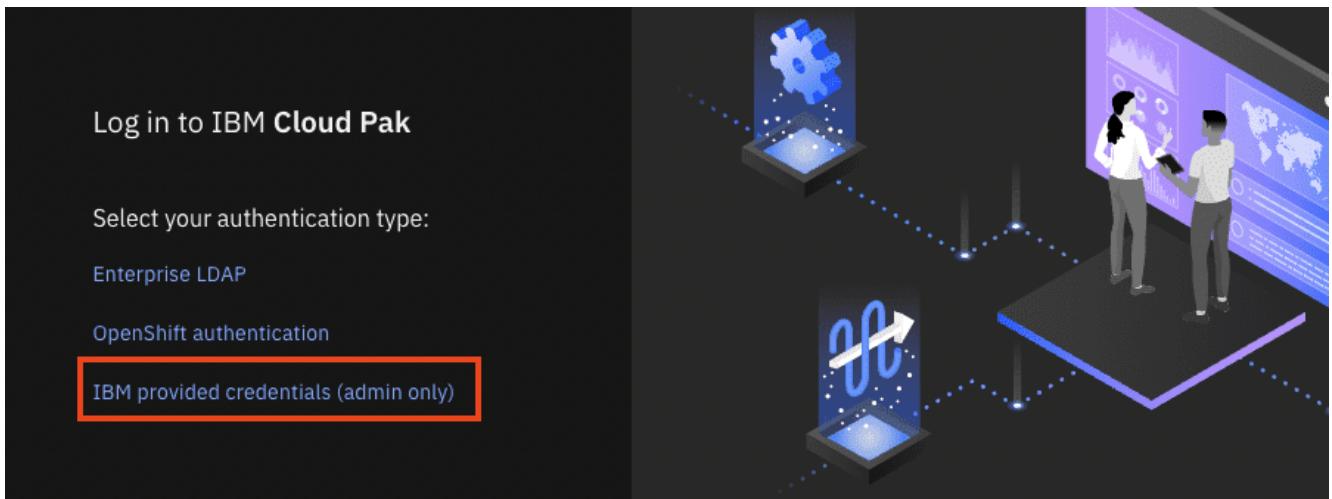
```
export WAIOPS_NAMESPACE=$(oc get po -A|grep aiops-orchestrator-controller |awk  
'{print$1}')  
  
echo "🌐 AI Manager: https://$(oc get route -n $WAIOPS_NAMESPACE cpd -o  
jsonpath={.spec.host})"  
echo "🌐 Demo UI: https://$(oc get route -n $WAIOPS_NAMESPACE waiops-demo-  
ui-python -o jsonpath={.spec.host})"
```

### 3.1.2 Login as demo User (preferred)

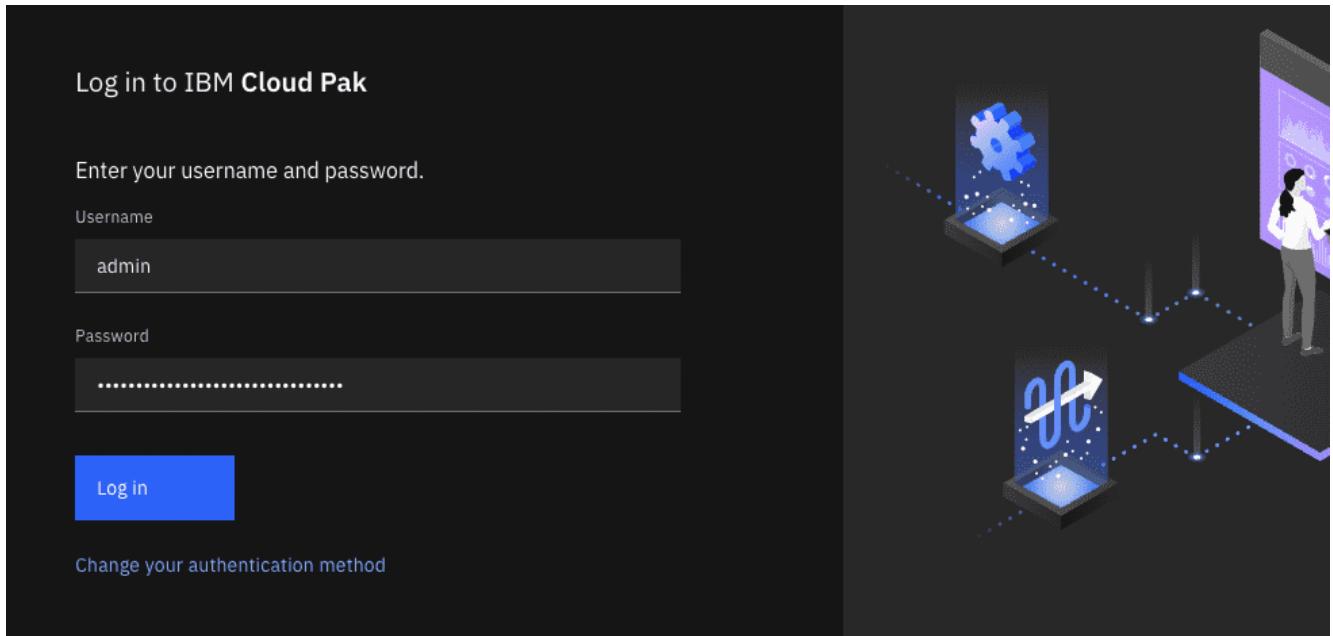
- Open the URL from the above
- Click on `Enterprise LDAP`
- Login as `demo` with the password `P4ssw0rd!`

### 3.1.2.1 Login as admin

- Open the URL from the above
- Click on **IBM provided credentials (admin only)**



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



The image shows the IBM Cloud Pak login interface on the left and a conceptual illustration on the right. The login interface has a dark background with white text. It says "Log in to IBM Cloud Pak" at the top, followed by "Enter your username and password." Below that are two input fields: "Username" containing "admin" and "Password" with several dots. A blue "Log in" button is below the password field. At the bottom, there's a link "Change your authentication method". To the right is a 3D-style illustration of a person standing next to a large screen displaying a dashboard. A gear icon is above the screen, and a blue arrow icon is on the screen itself.

## 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)

## 3.3 Configure Slack

Continue [here](#) for [Slack integration](#)



# 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).

 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 Integration with AI Manager

- To get the connection parameters, run:

```
./tools/97_addons/prepareNetcoolIntegration.sh
```

Execute the listed commands at the Objectserver prompt.

This gives you all the parameters needed for creating the connection.

- In the `AI Manager` "Hamburger" Menu select `Operate / Data and tool integrations`
- Click `Add connection`
- Under `Netcool`, click on `Add connection`
- Click `Connect`
- Name it `Netcool`
- Fill-in the information from the script above

## Netcool

Name  
Netcool

Description (optional)  
Enter description.

Deployment options  
 Local: Deploy in the same cluster where Cloud Pak for Watson AIOps is installed.  
 Remote: Deploy anywhere you choose, for example, a different network region, on SaaS, or remote on premises (VM, cluster, container, etc.).

ObjectServer pair

Admin username  
root

Admin password  
.....

Primary ObjectServer

Hostname  
evtmanager-objserv-agg-primary.cp4waiops-evtmgr.svc.cluster.local

Port  
4100

Test connection

Backup ObjectServer (optional)

Hostname  
evtmanager-objserv-agg-backup.cp4waiops-evtmgr.svc.cluster.local

Port  
4100

Cancel

Next

- Click **Test Connection**
- Click **Next**
- Toggle **Enable Data Collection** to the **ON** position
- 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: AIOPS
- 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

All  Any

Field	Comparator	Value	-	+
AlertGroup	=	'CEACorrelationKeyParent'		
AlertGroup	=	'robot-shop'		

## 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 screenshot shows the initial sign-in screen for creating a Slack workspace. At the top is the Slack logo. Below it, the text "First, enter your email" is displayed in large, bold, black font. A sub-instruction "We suggest using the email address you use at work." follows. There are two main sign-in options: a blue button with the Google "G" icon labeled "Continue with Google" and a text input field containing "name@work-email.com". Below these is a purple "Continue" button. At the bottom, there is a link for users who already have an account: "Already using Slack? Sign in to an existing workspace".

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



# Create a new Slack workspace

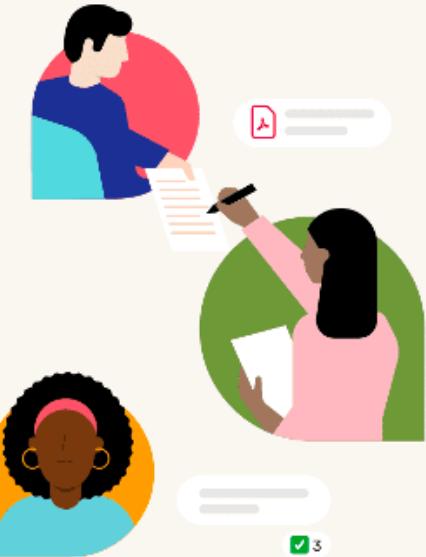
Slack gives your team a home — a place where they can talk and work together. To create a new workspace, click the button below.

**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.

The screenshot shows the Slack interface for the '#demo-environment' channel. The left sidebar has a dark purple background with the workspace name 'watson-lops' and a user icon. It includes sections for 'Browse Slack', 'Channels' (with '# demo-environment selected), 'Direct messages', and 'Add teammates'. The main area has a light gray header with the channel name '#demo-environment' and a search bar. Below the header, there's a message from 'Robert Barron' at 6:25 PM stating 'joined #demo-environment.' A message from 'Hello, team!' follows, saying 'First order of business...'. At the bottom, there's a message input field with placeholder text 'Send a message to #demo-environment' and a set of rich text editing icons.

#demo-environment

Add a topic

Browse Slack

Channels

# demo-environment

# general

# random

+ Add channels

Direct messages

R Robert Barron you

+ Add teammates

This is the very beginning of the #demo-environment channel

This channel is for working on a project. Hold meetings, share docs, and make decisions together with your team. [Edit description](#)

Today

R Robert Barron 6:25 PM joined #demo-environment.

Hello, team! First order of business... x

Send a message to #demo-environment

Rich text editor icons: bold, italic, underline, etc.

**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**.

The screenshot shows the 'Your Apps' section of the Slack API documentation. On the left, there's a sidebar with various API categories like Start learning, Authentication, Surfaces, etc. The main area displays a table of installed apps:

App Name	Workspace	Distribution Status
Humio	hirt.us	Not distributed
CP4WAIOPSTEST	hirt.us	Not distributed
AIOpsTEST	hirt.us	Not distributed
AIOPS-TEST	hirt.us	Not distributed
AIOps	AIOps	Not distributed
CP4WAIOPS	AIOps	Not distributed
Watson AI Manager (3.1)	TEC Europe	Not distributed

At the bottom, a note says "Don't see an app you're looking for? Sign in to another workspace."

2. Select **From an app manifest**

### Create an app X

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](#)

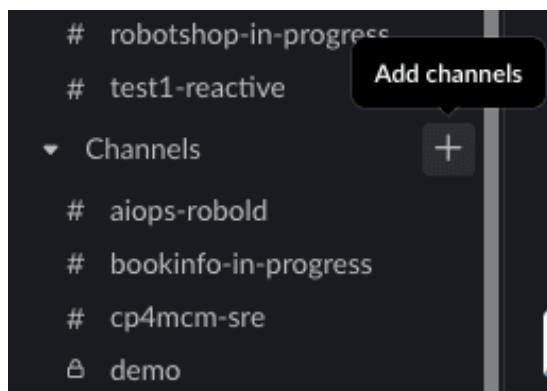
3. Select the appropriate workspace that you have created before and click **Next**
4. Copy and paste the content of this file [./doc/slack/slack-app-manifest.yaml](#).  
Don't bother with the URLs just yet, we will adapt them as needed.
5. Click **Next**

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:

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



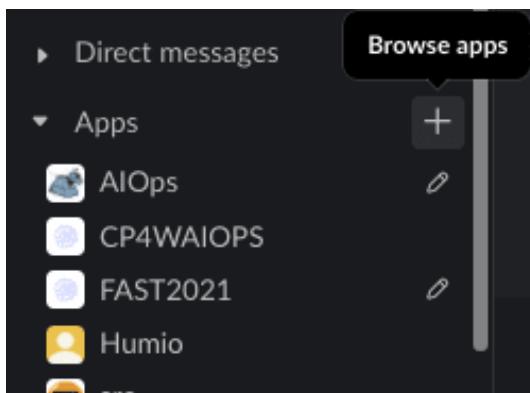
2. Right click on each channel and select **Copy Link**

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

A02MJTPE7M2

#### Date of App Creation

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

a...cor...o...c...r...c...n...c...a...c...s...z...-

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... ▾

# OAuth & Permissions

**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 ❤️

**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**

[REDACTED] [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.

[Learn more](#) [Show details](#) [Add connection +](#)

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

Items per page: 10 ▾ 0-0 of 0 items 1 ▾ of 1 page ▶

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

The screenshot shows the 'Data and tool connections / Add connections' section of the IBM Cloud Pak | Automation interface. A grid of icons and names for various tools like Runbooks, GitHub, Humio, ELK, Instana, Kafka, Kubernetes, LogDNA, Microsoft Teams, PagerDuty, ServiceNow, Slack, Splunk, SSH, and VMware. Each entry has an 'Add connection' button. The 'Slack' entry is specifically highlighted with a red box around its 'Add connection' button.

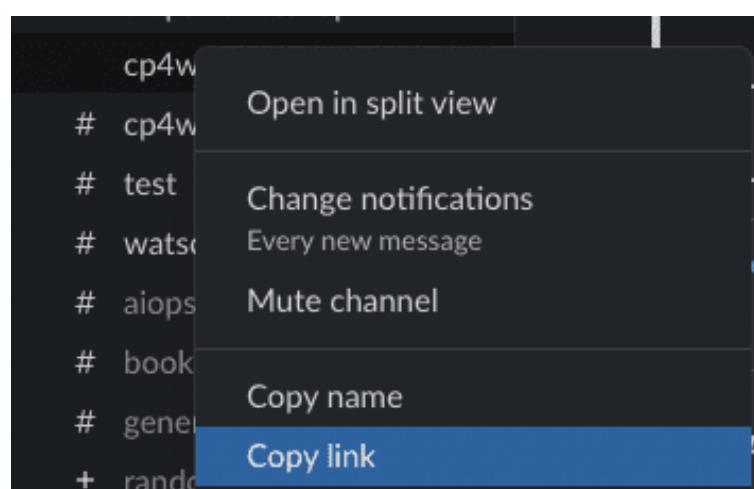
4. Name it "Slack"

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

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

The screenshot shows the 'Add connection' configuration for Slack. It includes fields for 'Name' (set to 'Slack'), 'Description (optional)' (containing a URL), 'Signing secret' (redacted), 'Language ID' (set to 'en'), and 'Bot token' (redacted).

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

The screenshot shows the IBM Automation AI Manager interface. On the left, a sidebar lists categories: Standard, Advanced, and Optional. Under Standard, there are sections for Category (All, Inventory, Tickets, Logs, Events, ChatOps), Custom Logs (0 integrations), ELK (0 integrations), Kubernetes (1 integration), ServiceNow (0 integrations), and Slack (1 integration). On the right, a detailed view of the Slack integration is shown. The title is "Slack" and the sub-title is "Manage integrations with Slack environments." A table lists one integration named "Slack" with the URL <https://cpd-aiops.apps.ocp46.tec.uk.ibm.com/aimanager/instances/0000000000000000/api/slack/events>. The table has columns for Name and Details. At the bottom, there are pagination controls: "Items per page: 5" and "1 of 1 items". A "Close" button is at the bottom right.

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 ✓**

`https://[REDACTED]` 465 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](#)

6. Click **Save Changes** button.

## 6.6.2 Interactivity & Shortcuts

7. Select **Interactivity & Shortcuts**.
8. 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://[REDACTED] 0000.t

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

9. 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 - Web Demo UI

### 7.1.1 Get the URL

- Run:

```
export WAIOPS_NAMESPACE=$(oc get po -A|grep aiops-orchestrator-controller |awk  
'{print$1}')  
  
echo "🌐 AI Manager: https://$(oc get route -n $WAIOPS_NAMESPACE cpd -o  
jsonpath={.spec.host})"  
echo "🌐 Demo UI: https://$(oc get route -n $WAIOPS_NAMESPACE waiops-  
demo-ui-python -o jsonpath={.spec.host})"
```

```
~/Mo/com~apple~C/1_0/4/1/2/2/cp4waiops-deployer ➜ main !1 ?5 ➤ export WAIOPS_NAMESPACE=$(oc get po -A|grep aiops-orchestrator-controller |awk '{print$1}')  
echo "🌐 AI Manager: https://$(oc get route -n $WAIOPS_NAMESPACE cpd -o jsonpath={.spec.host})"  
echo "🌐 Demo UI: https://$(oc get route -n $WAIOPS_NAMESPACE waiops-demo-ui-python -o jsonpath={.spec.host})"  
🌐 AI Manager: https://cpd-cp4waiops.itzroks-270003bu3k-6vqzfu-4b4a324f027aea19c5cbc0c3275c4656-0000.eu-de.containers.appdomain.cloud  
🌐 Demo UI: https://waiops-demo-ui-python-cp4waiops.itzroks-270003bu3k-6vqzfu-4b4a324f027aea19c5cbc0c3275c4656-0000.eu-de.containers.appdomain.cloud  
~/Mo/com~apple~C/1_0/4/1/2/2/cp4waiops-deployer ➜ main !1 ?5 ➤ 4s base ◆ < system ◆
```

### 7.1.2 Open the Web Demo UI

- Open the Demo UI URL from the above
- Login with the password **P4ssw0rd!**

### 7.1.3 Simulate the incident

Click on the red **Create Incident Memory Leak** button

This will create alerts and a story in AI Manager.

## 7.1.2 Login to AI Manager as demo User

- Open the AI Manager URL from the above
- Click on **Enterprise LDAP**
- Login as **demo** with the password **P4ssw0rd!**

**i** Give it a minute or two for all events and anomalies to arrive in AI Manager and Slack.

## 7.2 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.

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

## 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](#)