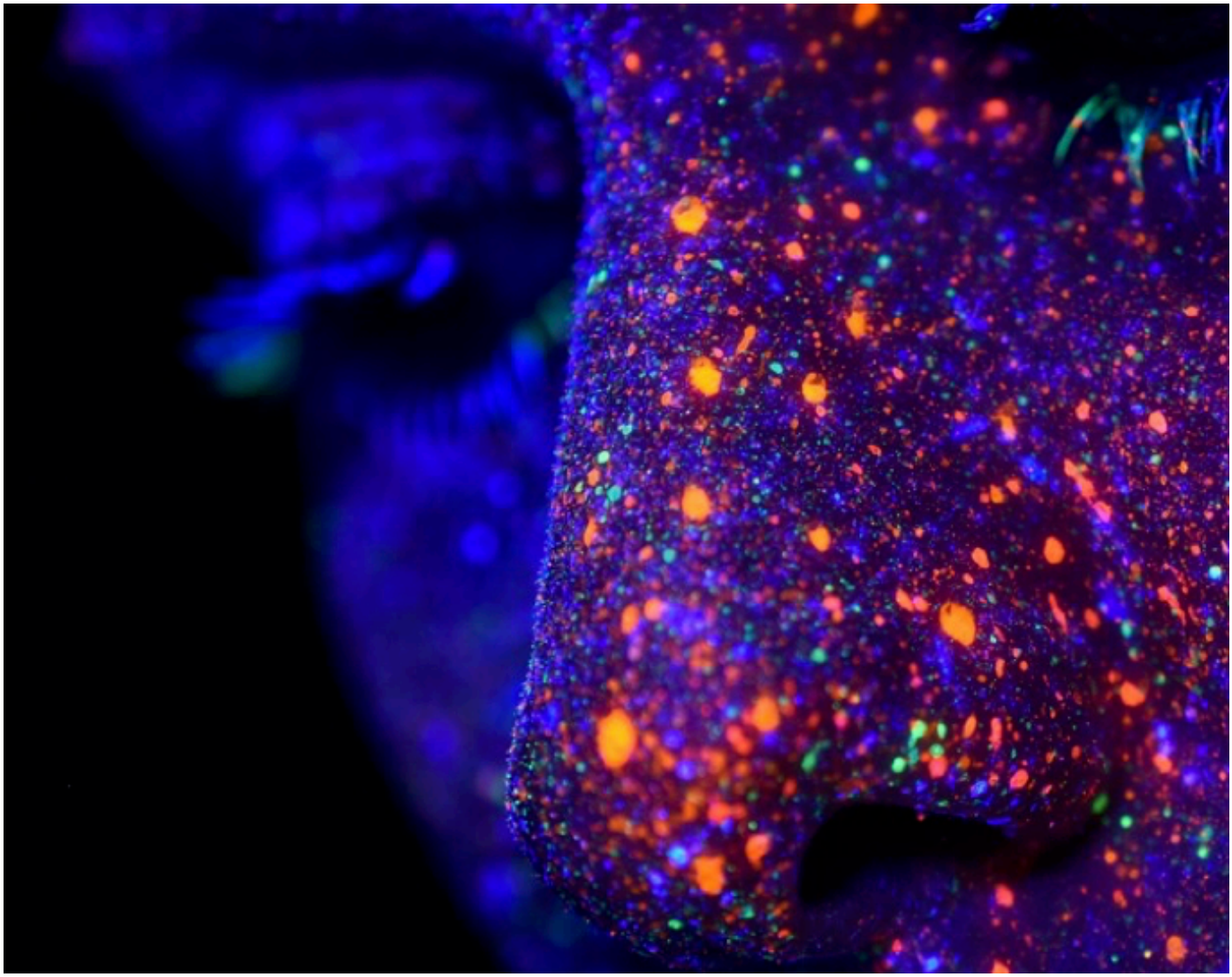
CP4WatsonAIOps V3.2

Demo Environment Installation with Ansible



Changes

Date	Description	Files
17 September 2021	First Draft	
20 September 2021	Turbonomic, Humio and Tooling	
20 September 2021	Roles	
21 September 2021	Improved robustness and checks	
22 September 2021	Corrected some bugs	Thanks Henning Sternkicker
24 September 2021	Corrected some bugs in the debug script	Thanks Philippe Thomas
24 September 2021	First beta release	
06 October 2021	Resiliency and Usability	
16 October 2021	Added EventManager (NOI) Standalone Option	
20 October 2021	Added AWX Option	Open Source Ansible Tower
21 October 2021	Added ManageIQ Option	Open Source Cloudforms
26 October 2021	10_debug_install.sh script updated	Still work in progress
27 October 2021	New template structure	
10 November 2021	First version for GA 3.2	

Installation

> ❗ You can find a handy install checklist here: INSTALLATION CHECKLIST.

Introduction

This repository contains the scrips for installing a Watson AIOps demo environment with an Ansible based installer.

They have been ported over from the shell scripts here https://github.ibm.com/NIKH/aiops-3.1.

As of 3.2 and going forward I will only update the Ansible scripts in this repository.

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 3.2 release on OpenShift 4.7 and 4.8.**

I have tested on ROKS 4.7 and 4.8 and Fyre 4.6 and the scripts run to completion.

❗ **Then NOI-->AI Manager Gateway is not working yet on ROKS**

So please if you have any feedback contact me

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

1. Prerequisites

1.1 OpenShift requirements

I installed the demo in a ROKS environment.

You'll need:

- ROKS 4.8 (4.7 should work also)
- 5x worker nodes Flavor `b3c.16x64` (so 16 CPU / 64 GB)

You might get away with less if you don't install some components (Humio, Turbonomic,...)

1.2 Tooling

You need the following tools installed in order to follow through this guide:

- ansible
- oc (4.7 or greater)
- jq
- kubectl (Not needed anymore - replaced by `oc`)
- kafkacat (only for training and debugging)
- elasticdump (only for training and debugging)
- IBM cloudctl (only for LDAP)

1.2.1 On Mac - Automated (preferred)

You can either run:

```
sudo ./13_install_prerequisites_mac.sh
```

1.2.1.1 On Mac - Manual

Or install them manually:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
brew install ansible
brew install kafkacat
brew install node
npm install elasticdump -g
brew install jq

curl -L https://github.com/IBM/cloud-pak-cli/releases/latest/download/cloudctl-darwin-
amd64.tar.gz -o cloudctl-darwin-amd64.tar.gz
tar xfvz cloudctl-darwin-amd64.tar.gz
sudo mv cloudctl-darwin-amd64 /usr/local/bin/cloudctl
rm cloudctl-darwin-amd64.tar.gz
```

Get oc and kubectl (optional) from [here]

or use :

```
wget https://github.com/openshift/okd/releases/download/4.7.0-0.okd-2021-07-03-
190901/openshift-client-mac-4.7.0-0.okd-2021-07-03-190901.tar.gz -O oc.tar.gz
tar xfzv oc.tar.gz
sudo mv oc /usr/local/bin
sudo mv kubectl /usr/local/bin.  (this is optional)
rm oc.tar.gz
rm README.md
```

I highly recomment installing the `k9s` tool :

```
wget
https://github.com/derailed/k9s/releases/download/v0.24.15/k9s_Darwin_x86_64.tar.gz
tar xfzv k9s_Darwin_x86_64.tar.gz
sudo mv k9s /usr/local/bin
rm LICENSE
rm README.md
```

1.2.2 On Ubuntu Linux - Automated (preferred)

For Ubuntu you can either run (for other distros you're on your own, sorry):

```
sudo ./14_install_prerequisites_ubuntu.sh
```

1.2.2.1 On Ubuntu Linux - Manual

Or install them manually:

`sed` comes preinstalled

```
sudo apt-get install -y ansible
sudo apt-get install -y kafkacat
sudo apt-get install -y npm
sudo apt-get install -y jq
sudo npm install elasticdump -g

curl -L https://github.com/IBM/cloud-pak-cli/releases/latest/download/cloudctl-linux-amd64.tar.gz -o cloudctl-linux-amd64.tar.gz
tar xfvz cloudctl-linux-amd64.tar.gz
sudo mv cloudctl-linux-amd64 /usr/local/bin/cloudctl
rm cloudctl-linux-amd64.tar.gz
```

Get oc and oc from here

or use :

```
wget https://github.com/openshift/okd/releases/download/4.7.0-0.okd-2021-07-03-190901/openshift-client-linux-4.7.0-0.okd-2021-07-03-190901.tar.gz -O oc.tar.gz
tar xfzv oc.tar.gz
sudo mv oc /usr/local/bin
sudo mv kubectl /usr/local/bin
rm oc.tar.gz
rm README.md
```

I highly recomment installing the `k9s` tool :

```
wget https://github.com/derailed/k9s/releases/download/v0.24.15/k9s_Linux_x86_64.tar.gz
tar xfzv k9s_Linux_x86_64.tar.gz
sudo mv k9s /usr/local/bin
rm LICENSE
rm README.md
```

1.4 Get the scripts and code from GitHub

1.4.1 Clone the GitHub Repository (preferred)

And obviosuly you'll need to download this repository to use the scripts.

```
git clone https://<YOUR GIT TOKEN>@github.ibm.com/NIKH/aiops-install-ansible.git
```

You can create your GIT token here.

1.4.1.1 Refresh the code from GitHub

Make sure you have the latest version:

```
git checkout origin/master -f | git checkout master -f | git pull origin master
```

Or create an alias for reuse:

```
alias gitrefresh='git checkout origin/master -f | git checkout master -f | git pull
origin master'
```

1.4.2 Download the GitHub Repository in a ZIP (not preferred)

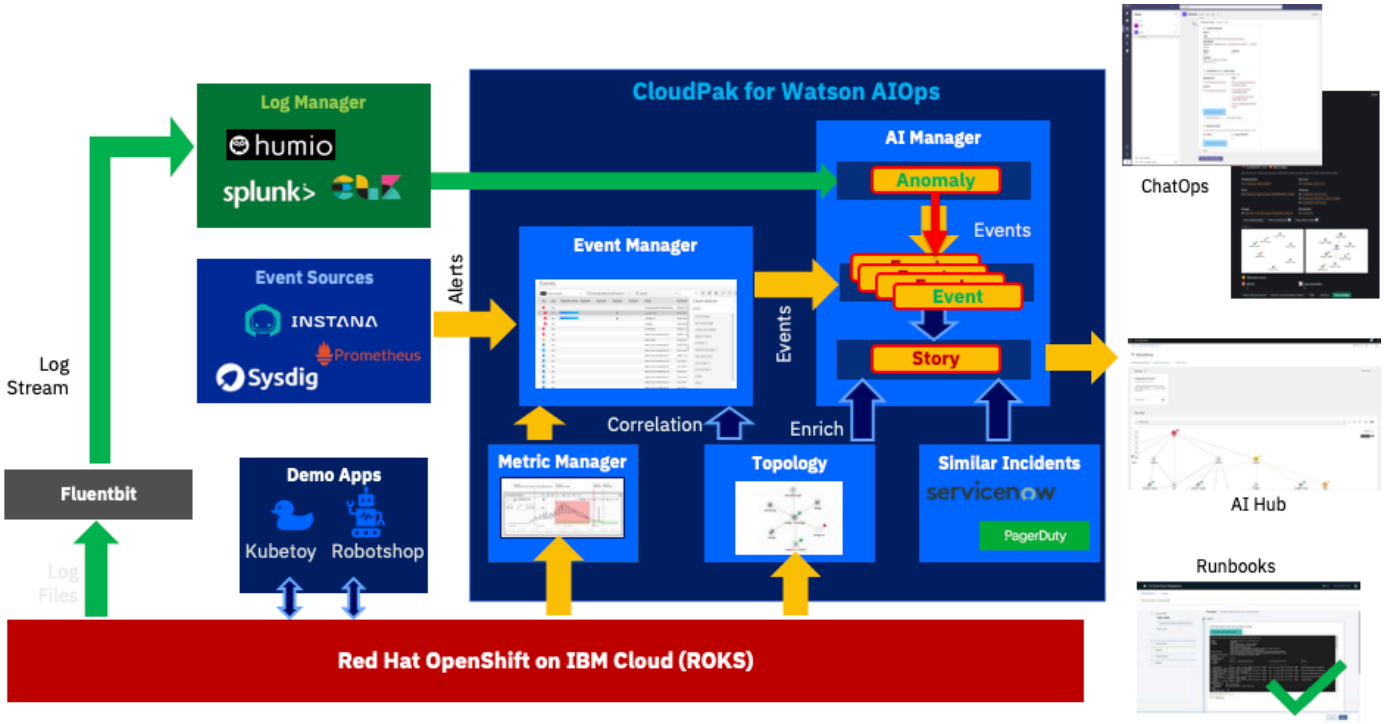
Simply click on the green `CODE` button and select `Download Zip` to download the scripts and code.

❗ If there are updates you have to re-download the ZIP.

2. Architecture

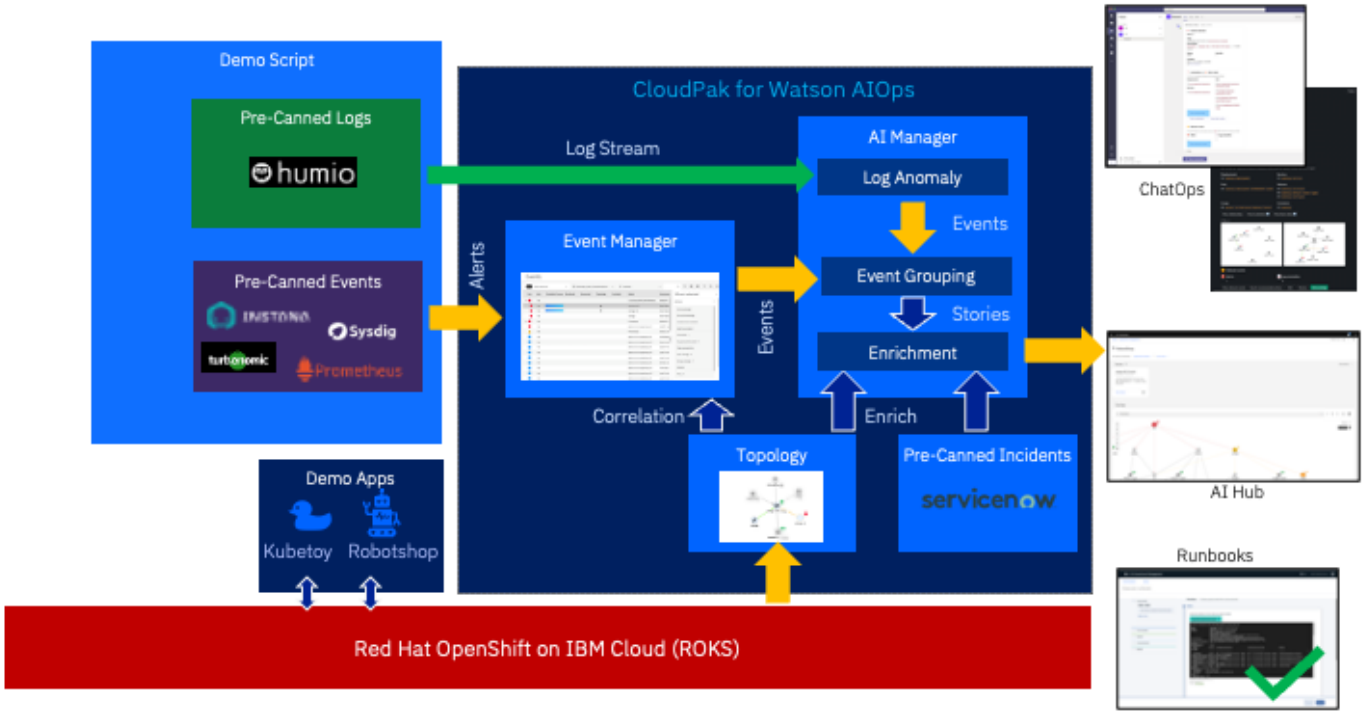
2.1 Basic Architecture

The environement (Kubernetes, Applications, ...) create logs that are being fed into a Log Management Tool (Humio in this case).



1. External Systems generate Alerts and send them into the Event Manager (Netcool Operations Insight), which in turn sends them to the AI Manager for Event Grouping.
2. At the same time AI Manager ingests the raw logs coming from the Log Management Tool (Humio) and looks for anomalies in the stream based on the trained model.
3. If it finds an anomaly it forwards it to the Event Grouping as well.
4. Out of this, AI Manager creates a Story that is being enriched with Topology (Localization and Blast Radius) and with Similar Incidents that might help correct the problem.
5. The Story is then sent to Slack.
6. A Runbook is available to correct the problem but not launched automatically.

2.2 Optimized Demo Architecture



For the this specific Demo environment:

- Humio is not needed as I am using pre-canned logs for training and for the anomaly detection (inception)
- The Events are also created from pre-canned content that is injected into AI Manager
- There are also pre-canned ServiceNow Incidents if you don't want to do the live integration with SNOW
- The Webpages that are reachable from the Events are static and hosted on my GitHub
- The same goes for ServiceNow Incident pages if you don't integrate with live SNOW

This allows us to:

- Install the whole Demo Environment in a self-contained OCP Cluster
- Trigger the Anomalies reliably
- Get Events from sources that would normally not be available (Instana, Turbonomic, Metrics Manager, ...)
- Show some examples of SNOW integration without a live system

3. CP4WAIOPS Base Install

3.1 Install AI Manager

3.1.1 Adapt configuration

Adapt the `00_config_cp4waiops.yaml` file with the desired parameters:

3.1.1.1 Automatic Login

The Playbook provides the means to automatically login to the cluster by filling out the following section of the config file:

```
#
*****************************************************************************************
*****************************************************
# ---------------------------------------------------------------------------------------
-----------------------------------------------
# OCP LOGIN PARAMETERS
# ---------------------------------------------------------------------------------------
-----------------------------------------------
#
*****************************************************************************************
*****************************************************
OCP_LOGIN: true
OCP_URL: https://c100-e.eu-gb.containers.cloud.ibm.com:31513
OCP_TOKEN: sha256~T6-cxxxxxxxxxxxx-dtuj3ELQfpioUhHms

#Version of your OCP Cluster (override by setting manually - 4.6, 4.7,...)
OCP_MAJOR_VERSION: automatic
```

3.1.1.2 Adapt AI Manager Config

```
#
*****************************************************************************************
*******************************************************
# ---------------------------------------------------------------------------------------
-----------------------------------------------
# CP4WAIOPS INSTALL PARAMETERS
# ---------------------------------------------------------------------------------------
-----------------------------------------------
#
*****************************************************************************************
*******************************************************

# CP4WAIOPS Namespace for installation
WAIOPS_NAMESPACE: cp4waiops

# CP4WAIOPS Size of the install (small: PoC/Demo, tall: Production)
WAIOPS_SIZE: small # Leave at small unless you know what you're doing

# Override the Storage Class auto detection
STORAGECLASS_FILE_OVERRIDE: not_configured
STORAGECLASS_BLOCK_OVERRIDE: not_configured
```

> There is no need to manually define the Storage Class anymore.
> The Playbook sets the storage class to `ibmc-file-gold-gid` for ROKS and `rook-cephfs` for Fyre.
> Otherwise it uses the default Storage Class.

> It is possible to override the Storage Class detection and force a custom Storage Class by setting `STORAGECLASS_XXX_OVERRIDE` in the config file.

3.1.1.3 Adapt Optional Components

```
#
*****************************************************************************************
*******************************************************
# ---------------------------------------------------------------------------------------
-----------------------------------------------
# DEMO INSTALL PARAMETERS
# ---------------------------------------------------------------------------------------
-----------------------------------------------
#
*****************************************************************************************
*******************************************************

# Create a demo user in the OCP cluster
CREATE_DEMO_USER: true

# Install Demo Applications
INSTALL_DEMO_APPS: true

# Print all credentials at the end of the installation
PRINT_LOGINS: true

# Install Bastion Server for Runbook Automation
INSTALL_RUNBOOK_BASTION: true

# Should Rook-Ceph be installed (automatic: install when on IBM Fyre) (enable,
automatic, disable)
ROOK_CEPH_INSTALL_MODE: automatic


#
*****************************************************************************************
*******************************************************
# ---------------------------------------------------------------------------------------
-----------------------------------------------
# MODULE INSTALL PARAMETERS
# ---------------------------------------------------------------------------------------
-----------------------------------------------
#
*****************************************************************************************
*******************************************************

# Install LDAP Server
INSTALL_LDAP: true

# Install Turbonomic (experimental - needs separate license)
INSTALL_TURBONOMIC: false
```

```yaml
# Turbonomic Storage Class (ibmc-block-gold, rook-cephfs, nfs-client, ...)
STORAGE_CLASS_TURBO: ibmc-block-gold
# Install Turbonomic Metrics simulation (highly experimental!)
INSTALL_TURBONOMIC_METRICS: false
# Install Turbonomic --> NOI Gateway (highly experimental!)
INSTALL_TURBONOMIC_GATEWAY: false


# Install Humio (needs separate license)
INSTALL_HUMIO: false

# Install ELK Stack
INSTALL_ELK: false

# Install AWX (Open Source Ansible Tower)
INSTALL_AWX: false

# Install ManageIQ (Open Source CloudForms)
INSTALL_MANAGEIQ: false
```

3.1.2 Get the 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.

This token is being referred to as <PULL_SECRET_TOKEN> below and should look something like this (this is NOT a valid token):


```
eyJhbGciOiJIUzI1NiJ9.eyJpc3adsgJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE1Nzg0NzQzMjgsImp0aSI6Ij
RjYTM3gsdgdMzExNjQxZDdiMDJhMjRmMGMxMWgdsmZhIn0.Z-rqfSLJA-R-
ow__tI3RmLx4mssdggdabvdcgdgYEkbYY
```

3.1.3 🚀 Start installation

Just run:

```
./10_install_ai_manager.sh -t <PULL_SECRET_TOKEN> [-v true]


Example:
./10_install_ai_manager.sh -t
eyJhbGciOiJIUzI1NiJ9.eyJpc3adsgJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE1Nzg0NzQzMjgsImp0aSI6Ij
RjYTM3gsdgdMzExNjQxZDdiMDJhMjRmMGMxMWgdsmZhIn0.Z-rqfSLJA-R-
ow__tI3RmLx4mssdggdabvdcgdgYEkbYY
```

This will install:

- CP4WAIOPS AI Manager
- OpenLDAP (if enabled)
- Demo Apps (if enabled)
- Register LDAP Users (if enabled)
- Housekeeping
  - Additional Routes (Topology, Flink)
  - Create OCP User (serviceaccount demo-admin)
  - Patch Ingress
  - Adapt NGINX Certificates
  - Adapt Slack Welcome message to /welcome
- Turbonomic (if enabled)
- Humio (if enabled)
- OCP ELK Stack (if enabled)
- AWX (Open Source Ansible Tower - if enabled)
- ManageIQ (Open Source CloudForms - if enabled)

3.2 Install Event Manager

To get the token, see here

3.1.3 🚀 Start installation

Just run:

```
./11_install_event_manager.sh -t <PULL_SECRET_TOKEN> [-v true]


Example:
./11_install_event_manager.sh -t
eyJhbGciOiJIUzI1NiJ9.eyJpc3adsgJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE1Nzg0NzQzMjgsImp0aSI6Ij
RjYTM3gsdgdMzExNjQxZDdiMDJhMjRmMGMxMWgdsmZhIn0.Z-rqfSLJA-R-
ow__tI3RmLx4mssdggdabvdcgdgYEkbYY
```

This will install:

- CP4WAIOPS Event Manager
- Gateway

3.3 Get Passwords and Credentials

At any moment you can run `./tools/20_get_logins.sh` that will print out all the relevant passwords and credentials.

Usually it's a good idea to store this in a file for later use:

```
./tools/20_get_logins.sh > my_credentials.txt
```

3.4 Check status of installation

At any moment you can run `./tools/10_debug_install.sh` and select `Option 1` to check your installation.

4. Configure Applications and Topology

4.1 Create Kubernetes Observer for the Demo Applications

Do this for your applications (RobotShop by default)

- In the `AI Manager` "Hamburger" Menu select `Operate` / `Data and tool integrations`
- Click `Add connection`
- Under `Kubernetes`, click on `Add Integration`
- Click `Connect`
- Name it `RobotShop`
- Data Center `demo`
- Click `Next`
- Choose `local` for Connection Type
- Set `Hide pods that have been terminated` to `On`
- Set `Correlate analytics events on the namespace groups created by this job` to `On`
- Set Namespace to `robot-shop`
- Click `Next`
- Click `Done`

4.2 Create REST Observer to Load Topologies

- In the `AI Manager` "Hamburger" Menu select `Operate` / `Data and tool integrations`
- Click `Add connection`
- On the left click on `Topology`
- On the top right click on `You can also configure, schedule, and manage other observer jobs`
- Click on `Add a new Job`
- Select `REST` / `Configure`
- Choose “bulk_replace”
- Set Unique ID to “listenJob” (important!)
- Set Provider to whatever you like (usually I set it to “listenJob” as well)
- `Save`

4.3 Create Merge Rules for Kubernetes Observer

Launch the following:

```
./60_load_robotshop_topology.sh
```

This will create:

- Merge Rules
- Merge Topologies for RobotShop.

❗ Please manually re-run the Kubernetes Observer to make sure that the merge has been done.

4.5 Create AIOps Application

Robotshop

- In the `AI Manager` go into `Operate` / `Application Management`
- Click `Create Application`
- Select `robot-shop` namespace
- Click `Add to Application`
- Name your Application (RobotShop)
- If you like check `Mark as favorite`
- Click `Save`

5. Configure Event Manager

❗ You only have to do this if you have installed Event Manager (NOI). For basic demoing with AI MAnager this is not needed.

5.1 Event Manager Webhooks

Create Webhooks in Event Manager 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 (Humio)
- Security Events (Falco)
- Instana Events
- Metric Manager Events (Predictive)
- Turbonomic Events
- CP4MCM Synthetic Selenium Test Events

5.1.1 Generic Demo Webhook

You have to define the following Webhook in Event Manager (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 `00_config-secrets.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.

5.2 Create custom Filter and View in NOI (optional)

5.2.1 Filter

Duplicate the `Default` filter and set to global.

- Name: AIOPS
- Logic: **Any** (!)
- Filter:
  - AlertGroup = 'CEACorrelationKeyParent'
  - AlertGroup = 'robot-shop'

5.2.2 View

Duplicate the `Example_IBM_CloudAnalytics` View and set to global.

- Name: AIOPS

Configure to your likings.

5.3 Create Templates for Topology Grouping (optional)

This gives you probale cause and is not strictly needed if you don't show Event Manager!

- In the Event Manager "Hamburger" Menu select `Operate` / `Topology Viewer`
- Then, in the top right corner, click on the icon with the three squares (just right of the cog)
- Select `Create a new Template`
- Select `Dynamic Template`

Create a template for RobotShop:

- Search for `web-deployment` (deployment)
- Create Topology 3 Levels
- Name the template (robotshop)
- Select `Namespace` in `Group type`
- Enter `robotshop_` for `Name prefix`
- Select `Application`
- Add tag `namespace:robot-shop`
- Save

5.4 Create grouping Policy

- NetCool Web Gui --> `Insights` / `Scope Based Grouping`
- Click `Create Policy`
- `Action` select fielt `Alert Group`
- Toggle `Enabled` to `On`
- Save

5.5 Create NOI Menu item - Open URL

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. Training

6.1 Prepare Training

6.1.1 Create Kafka Humio Log Training Integration

- In the `AI Manager` "Hamburger" Menu select `Operate` / `Data and tool integrations`
- Click `Add connection`
- Under `Kafka`, click on `Add Integration`
- Click `Connect`
- Name it `HumioInject`
- Click `Next`
- Select `Data Source` / `Logs`
- Select `Mapping Type` / `Humio`
- Paste the following in `Mapping` (the default is **incorrect**!:

```
{
"codec": "humio",
"message_field": "@rawstring",
"log_entity_types":
"kubernetes.namespace_name,kubernetes.container_hash,kubernetes.host,kubernetes.con
tainer_name,kubernetes.pod_name",
"instance_id_field": "kubernetes.container_name",
"rolling_time": 10,
"timestamp_field": "@timestamp"
}
```

- Click `Next`
- Toggle `Data Flow` to the `ON` position
- Select `Live data for continuous AI training and anomaly detection`
- Click `Save`

6.1.2 Create Kafka Netcool Training Integration

- In the `AI Manager` "Hamburger" Menu select `Operate` / `Data and tool integrations`
- Click `Add connection`
- Under `Kafka`, click on `Add Integration`
- Click `Connect`
- Name it `NOI`
- Click `Next`
- Select `Data Source` / `Events`
- Select `Mapping Type` / `NOI`
- Click `Next`
- Toggle `Data Flow` to the `ON` position
- Click `Save`

6.1.3 Create ElasticSearch Port Forward

Please start port forward in **separate** terminal.

Run the following:

```
while true; do oc port-forward statefulset/iaf-system-elasticsearch-es-aiops 9200; done
```

or use the script that does it automatically

```
./tools/28_access_elastic.sh
```

6.2 Load Training Data

Run the following scripts to inject training data:

```
./50_load_robotshop_data.sh
```

This takes some time (20-60 minutes depending on your Internet speed).

6.3 Train Log Anomaly

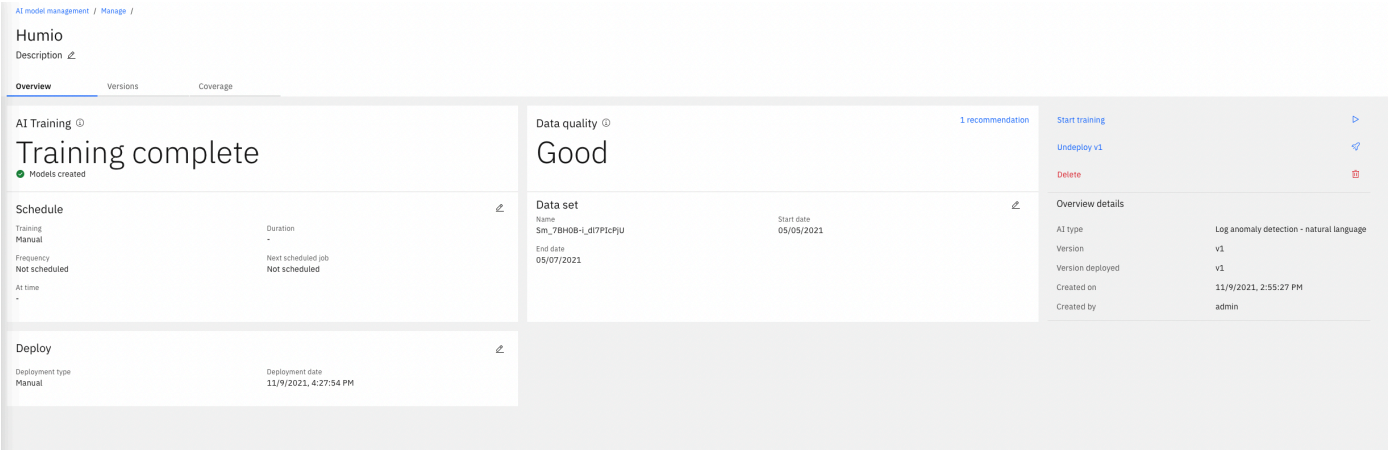
6.3.1 Create Training Definition for Log Anomaly

- In the `AI Manager` "Hamburger" Menu select `Operate` / `AI model management`
- Under `Log anomaly detection - natural language` click on `Configure`
- Click `Next`
- Name it `LogAnomaly`
- Click `Next`
- Select `Custom`
- Select `05/05/21` (May 5th 2021 - dd/mm/yy) to `07/05/21` (May 7th 2021) as date range (this is when the logs we're going to inject have been created)
- Click `Next`
- Click `Next`
- Click `Create`

6.3.2 Train the Log Anomaly model

- Click on the `Manager` Tab
- Click on the `LogAnomaly` entry
- Click `Start Training`
- This will start a precheck that should tell you after a while that you are ready for training ant then start the training

After successful training you should get:



- Click on `Deploy vXYZ`

⚠️ If the training shows errors, please make sure that the date range of the training data is set to May 5th 2021 through May 7th 2021 (this is when the logs we're going to inject have been created)

6.4 Train Event Grouping

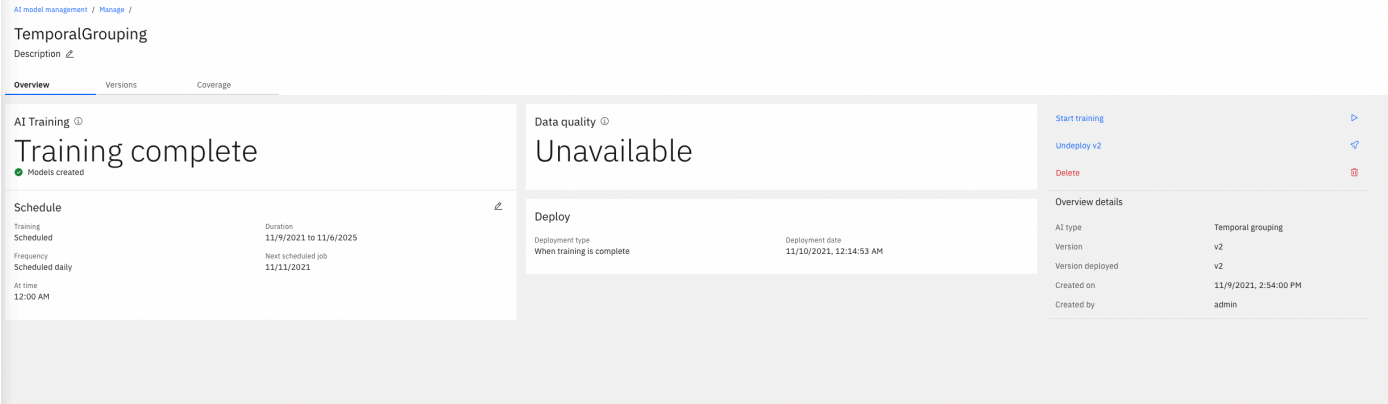
6.4.1 Create Training Definition for Event Grouping

- In the `AI Manager` "Hamburger" Menu select `Operate` / `AI model management`
- Under `Temporal grouping` click on `Configure`
- Click `Next`
- Name it `EventGrouping`
- Click `Next`
- Click `Done`

6.4.2 Train the Event Grouping Model

- Click on the `Manager` Tab
- Click on the `EventGrouping` entry
- Click `Start Training`
- This will start the training

After successful training you should get:



- The model is deployed automatically

6.5 Train Incident Similarity

❗ Only needed if you don't plan on doing the Service Now Integration

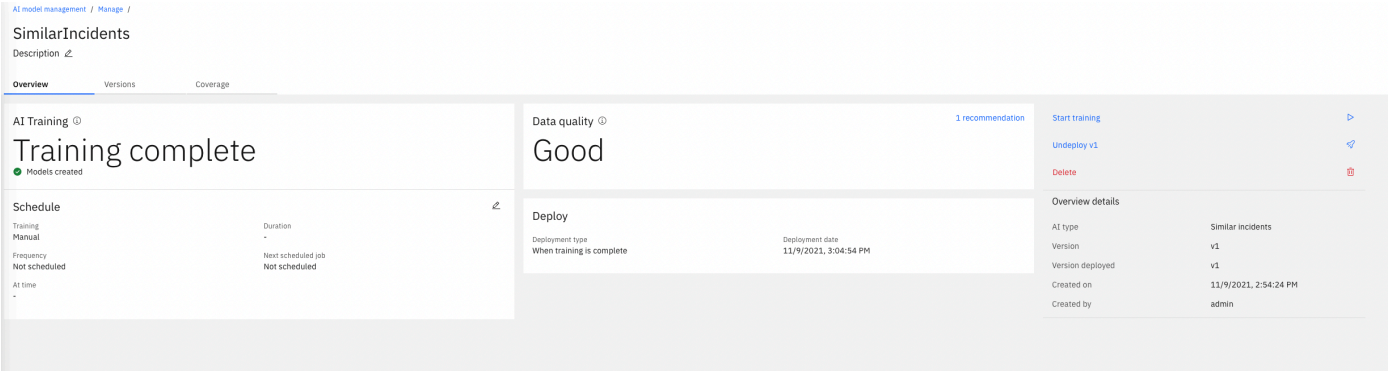
6.5.1 Create Training Definition

- In the `AI Manager` "Hamburger" Menu select `Operate` / `AI model management`
- Under `Similar incidents` click on `Configure`
- Click `Next`
- Name it `SimilarIncidents`
- Click `Next`
- Click `Next`
- Click `Done`

6.5.2 Train the Incident Similarity Model

- Click on the `Manager` Tab
- Click on the `SimilarIncidents` entry
- Click `Start Training`
- This will start the training

After successful training you should get:



- The model is deployed automatically

6.6 Train Change Risk

❗ Only needed if you don't plan on doing the Service Now Integration

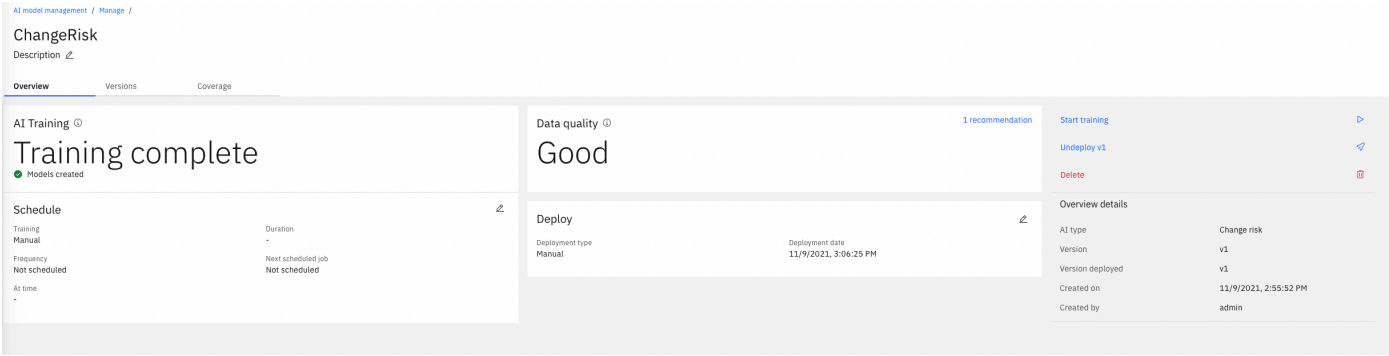
6.6.1 Create Training Definition

- In the `AI Manager` "Hamburger" Menu select `Operate` / `AI model management`
- Under `Change risk` click on `Configure`
- Click `Next`
- Name it `ChangeRisk`
- Click `Next`
- Click `Next`
- Click `Done`

6.6.2 Train the Change Risk Model

- Click on the `Manager` Tab
- Click on the `ChangeRisk` entry
- Click `Start Training`
- This will start the training

After successful training you should get:



- Click on `Deploy vXYZ`

7. Configure Runbooks

7.1 Create Bastion Server

A simple Pod with the needed tools (oc, kubectl) being used as a bastion host for Runbook Automation should already have been created by the install script.

7.2 Create the NOI Integration

7.2.1 In NOI

- Go to `Administration` / `Integration with other Systems` / `Automation Type` / `Script`
- Copy the SSH KEY

7.2.2 Adapt SSL Certificate in Bastion Host Deployment.

- Select the `bastion-host` Deployment in Namespace `default`
- Adapt Environment Variable SSH_KEY with the key you have copied above.

7.3 Create Automation

7.3.1 Connect to Cluster

`Automation` / `Runbooks` / `Automations` / `New Automation`

```
oc login --token=$token --server=$ocp_url
```

Use these default values

```
target: bastion-host-service.default.svc
user:   root
$token   : Token from your login (from ./tools/20_get_logins.sh)
$ocp_url : URL from your login (from ./tools/20_get_logins.sh, something like
https://c102-e.eu-de.containers.cloud.ibm.com:32236)
```

7.3.2 RobotShop Mitigate MySql

`Automation` / `Runbooks` / `Automations` / `New Automation`

```
oc scale deployment --replicas=1 -n robot-shop ratings
oc delete pod -n robot-shop $(oc get po -n robot-shop|grep ratings |awk '{print$1}') --
force --grace-period=0
```

Use these default values

```
target: bastion-host-service.default.svc
user:   root
```

7.4 Create Runbooks

- `Library` / `New Runbook`
- Name it `Mitigate RobotShop Problem`
- `Add Automated Step`
- Add `Connect to Cluster`
- Select `Use default value` for all parameters
- Then `RobotShop Mitigate Ratings`
- Select `Use default value` for all parameters
- Click `Publish`

7.5 Add Runbook Triggers

- `Triggers` / `New Trigger`
- Name and Description: `Mitigate RobotShop Problem`
- Conditions
  - Name: RobotShop
  - Attribute: Node
  - Operator: Equals
  - Value: mysql-deployment or web
- Click `Run Test`
- You should get an Event `[Instana] Robotshop available replicas is less than desired replicas - Check conditions and error events - ratings`
- Select `Mitigate RobotShop Problem`
- Click `Select This Runbook`
- Toggle `Execution` / `Automatic` to `off`
- Click `Save`

8. Slack integration

8.1 Initial Slack Setup

For the system to work you need to setup your own secure gateway and slack workspace. It is suggested that you do this within the public slack so that you can invite the customer to the experience as well. It also makes it easier for is to release this image to Business partners

You will need to create your own workspace to connect to your instance of CP4WAOps.

Here are the steps to follow:

1. Create Slack Workspace
2. Create Slack App
3. Create Slack Channels
4. Create Slack Integration
5. Get the Integration URL - Public Cloud - ROKS OR
6. Get the Integration URL - Private Cloud - Fyre/TEC
7. Create Slack App Communications
8. Prepare Slack Reset

8.2 NGNIX Certificate for V3.1.1 - If the integration is not working

In order for Slack integration to work, there must be a signed certicate on the NGNIX pods. The default certificate is self-signed and Slack will not accept that. The method for updating the certificate has changed between AIOps v2.1 and V3.1.1. The NGNIX pods in V3.1.1 mount the certificate through a secret called `external-tls-secret` and that takes precedent over the certificates staged under `/user-home/_global_/customer-certs/`.

For customer deployments, it is required for the customer to provide their own signed certificates. An easy workaround for this is to use the Openshift certificate when deploying on ROKS. **Caveat**: The CA signed certificate used by Openshift is automatically cycled by ROKS (I think every 90 days), so you will need to repeat the below once the existing certificate is expired and possibly reconfigure Slack.

This method replaces the existing secret/certificate with the one that OpenShift ingress uses, not altering the NGINX deployment. An important note, these instructions are for configuring the certificate post-install. Best practice is to follow the installation instructions for configuring certificates during that time.

The custom resource `AutomationUIConfig/iaf-system` controls the certificates and the NGINX pods that use those certificates. Any direct update to the certificates or pods will eventually get overwritten, unless you first reconfigure `iaf-system`. It's a bit tricky post-install as you will have to recreate the `iaf-system` resource quickly after deleting it, or else the installation operator will recreate it. For this reason it's important to run all the commands one after the other. **Ensure that you are in the project for AIOps**, then paste all the code on your command line to replace the `iaf-system` resource.

```
NAMESPACE=$(oc project -q)
IAF_STORAGE=$(oc get AutomationUIConfig -n $NAMESPACE -o jsonpath='{
.items[*].spec.storage.class }')
oc get -n $NAMESPACE AutomationUIConfig iaf-system -oyaml > iaf-system-backup.yaml
oc delete -n $NAMESPACE AutomationUIConfig iaf-system
cat <<EOF | oc apply -f -
apiVersion: core.automation.ibm.com/v1beta1
kind: AutomationUIConfig
metadata:
  name: iaf-system
  namespace: $NAMESPACE
spec:
  description: AutomationUIConfig for cp4waiops
  license:
    accept: true
  version: v1.0
  storage:
    class: $IAF_STORAGE
  tls:
    caSecret:
      key: ca.crt
      secretName: external-tls-secret
    certificateSecret:
      secretName: external-tls-secret
EOF
```

Again, **ensure that you are in the project for AIOps** and run the following to replace the existing secret with a secret containing the OpenShift ingress certificate.

```
NAMESPACE=$(oc project -q)
# collect certificate from OpenShift ingress
ingress_pod=$(oc get secrets -n openshift-ingress | grep tls | grep -v router-metrics-
certs-default | awk '{print $1}')
oc get secret -n openshift-ingress -o 'go-template={{index .data "tls.crt"}}'
${ingress_pod} | base64 -d > cert.crt
oc get secret -n openshift-ingress -o 'go-template={{index .data "tls.key"}}'
${ingress_pod} | base64 -d > cert.key
oc get secret -n $WAIOPS_NAMESPACE external-tls-secret -o 'go-template={{index .data
"ca.crt"}}'| base64 -d > ca.crt
# backup existing secret
oc get secret -n $WAIOPS_NAMESPACE external-tls-secret -o yaml > external-tls-
secret$(date +%Y-%m-%dT%H:%M:%S).yaml
# delete existing secret
oc delete secret -n $WAIOPS_NAMESPACE external-tls-secret
# create new secret
oc create secret generic -n $WAIOPS_NAMESPACE external-tls-secret --from-
file=ca.crt=ca.crt --from-file=cert.crt=cert.crt --from-file=cert.key=cert.key --dry-
run=client -o yaml | oc apply -f -
# scale down nginx
REPLICAS=2
oc scale Deployment/ibm-nginx --replicas=0
# scale up nginx
sleep 3
oc scale Deployment/ibm-nginx --replicas=${REPLICAS}
rm cert.crt
rm cert.key
rm ca.crt
rm external-tls-secret
```

Wait for the nginx pods to come back up

```
oc get pods -l component=ibm-nginx
```

When the integration is running, remove the backup file

```
rm ./iaf-system-backup.yaml
```

The last few lines scales down the NGINX pods and scales them back up. It takes about 3 minutes for the pods to fully come back up.

Once those pods have come back up, you can verify the certificate is secure by logging in to AIOps. Note that the login page is not part of AIOps, but rather part of Foundational Services. So you will have to login first and then check that the certificate is valid once logged in. If you want to update the certicate for Foundational Services you can find instructions here.

8.3 Change the Slack Slash Welcome Message (optional)

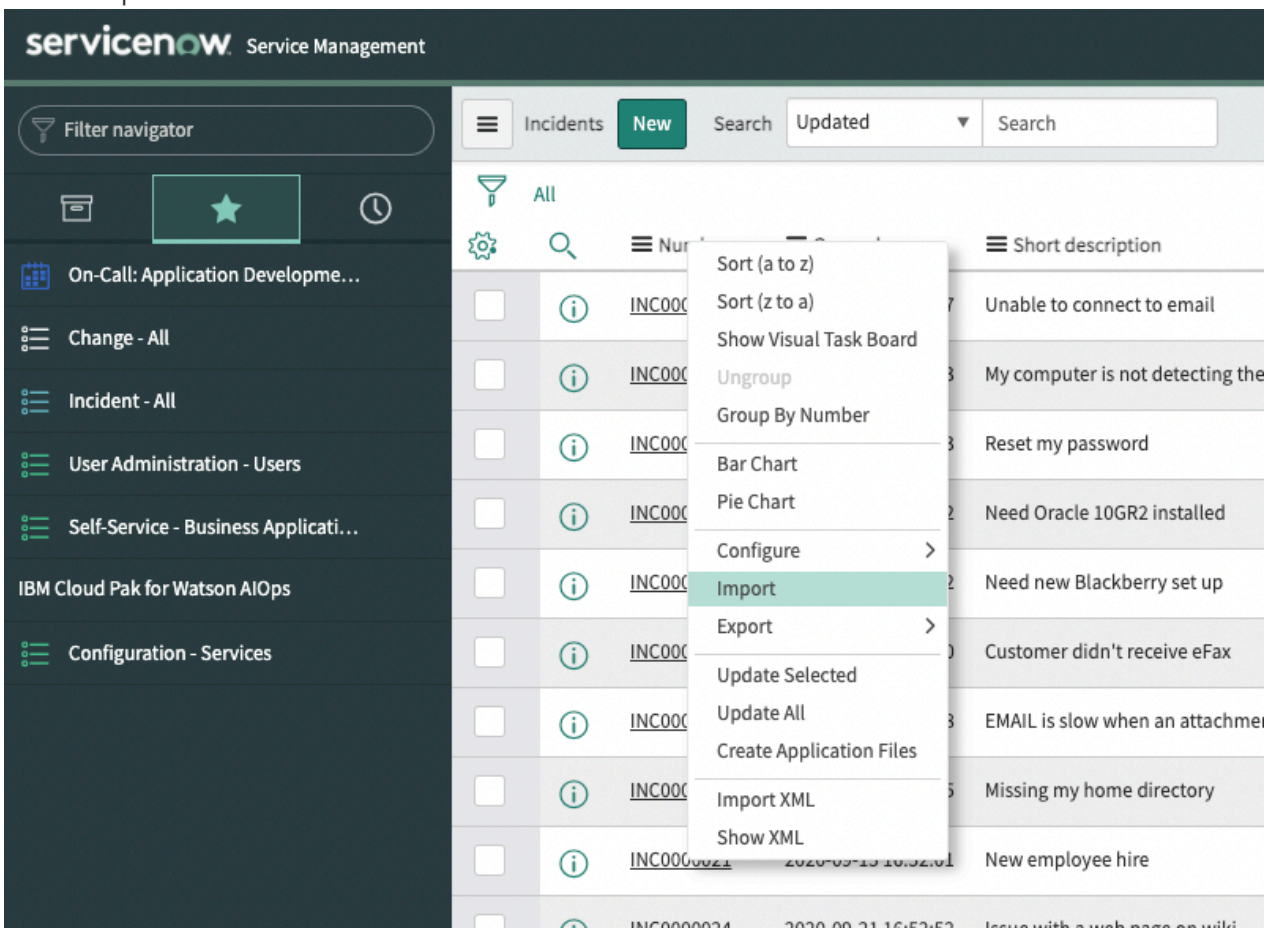
If you want to change the welcome message

```
oc set env deployment/$(oc get deploy -l app.kubernetes.io/component=chatops-slack-integrator -o jsonpath='{.items[*].metadata.name }') SLACK_WELCOME_COMMAND_NAME=/aiops-help
```

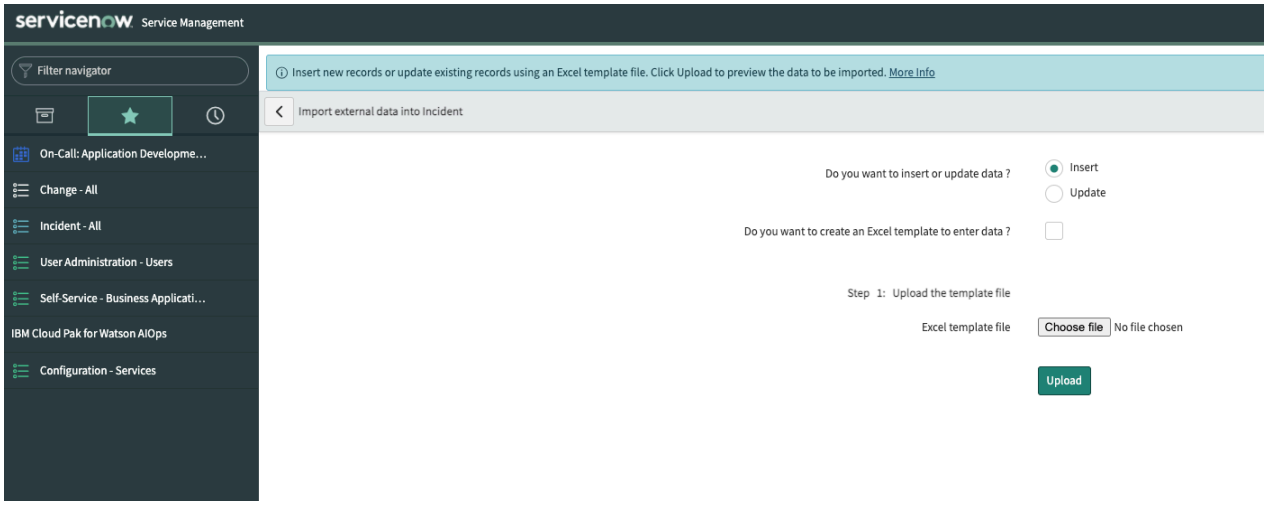
9. Service Now integration

9.1 Integration

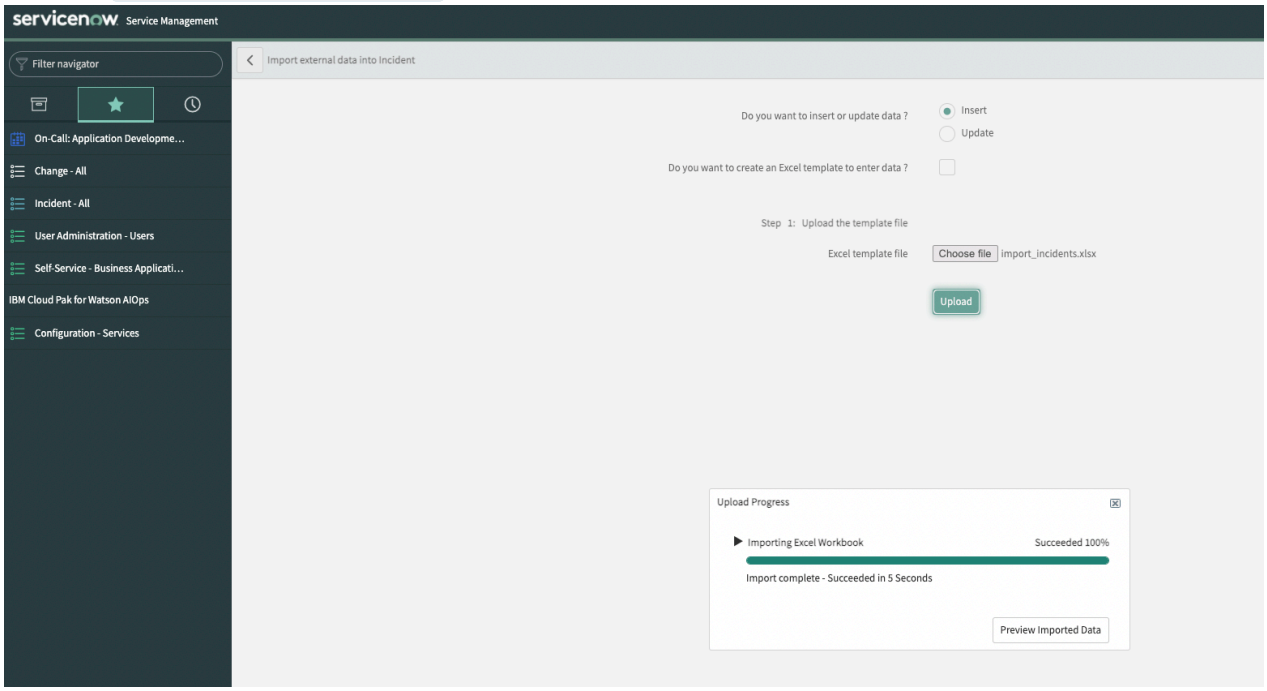
1. Follow [this](#) document to get and configure your Service Now Dev instance with CP4WAIOPS.
   Stop at `Testing the ServiceNow Integration`.
   ❗ ❗ Don't do the training as of yet.
2. Import the Changes from ./doc/servicenow/import_change.xlsx
   1. Select `Change - All` from the right-hand menu
   2. Right Click on `Number` in the header column
   3. Select Import



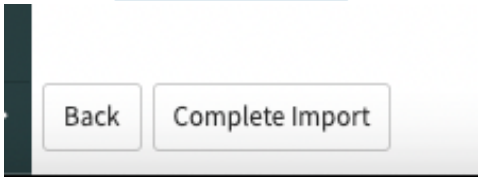
   4. Chose the ./doc/servicenow/import_change.xlsx file and click `Upload`



5. Click on `Preview Imported Data`



6. Click on `Complete Import` (if there are errors or warnings just ignore them and import anyway)



3. Import the Incidents from ./doc/servicenow/import_incidents.xlsx
    1. Select `Incidents - All` from the right-hand menu
    2. Proceed as for the Changes but for Incidents
4. Now you can finish configuring your Service Now Dev instance with CP4WAIOPS by going back and continue whre you left off at `Testing the ServiceNow Integration`.

10. Some Polishing

10.1 Add LDAP Logins to CP4WAIOPS

- Go to `AI Manager` Dashboard
- Click on the top left "Hamburger" menu
- Select `User Management`
- Select `User Groups` Tab
- Click `New User Group`
- Enter demo (or whatever you like)
- Click Next
- Select `LDAP Groups`
- Search for `demo`
- Select `cn=demo,ou=Groups,dc=ibm,dc=com`
- Click Next
- Select Roles (I use Administrator for the demo environment)
- Click Next
- Click Create

10.2 Monitor Kafka Topics

At any moment you can run `./tools/22_monitor_kafka.sh` this allows you to:

- List all Kafka Topics
- Monitor Derived Stories
- Monitor any specific Topic

10.3 Monitor ElasticSearch Indexes

At any moment you can run `./tools/28_access_elastic.sh` in a separate terminal window.

This allows you to access ElasticSearch and gives you:

- ES User
- ES Password

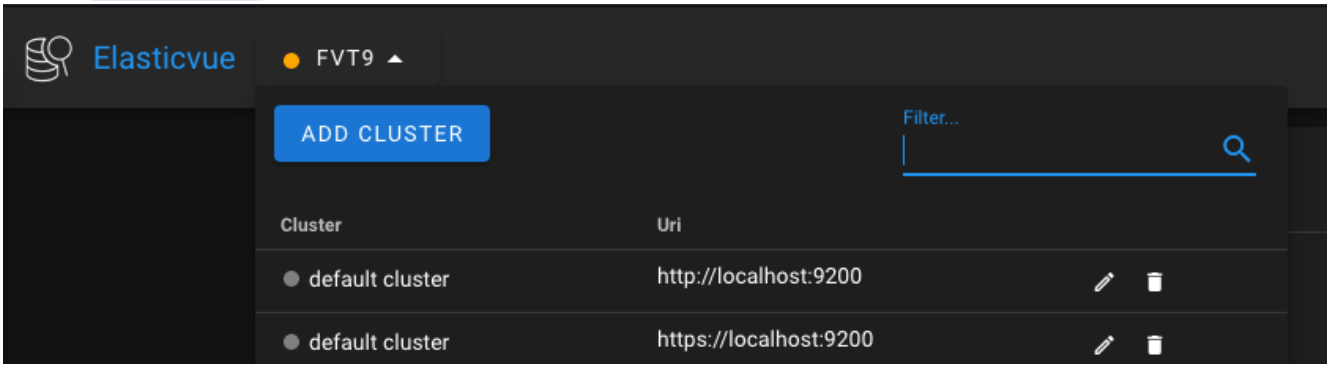
```
AI OPS DEBUG - Enable ElastcSearch remote access
Initializing.....
Getting credentials
Already on project "cp4waiops" on server "https://c100-e.eu-de.containers.cloud.ibm.com:30783".
OK

Checking credentials
OK - Elasticsearch Username
OK - Elasticsearch Password

ElasticSearch Access

URL                    : https://localhost:9200
User                   : cp4waiops-cartridge
Password               : s29tRmiTWa
You can use any ElasticSearch Browser. I usually use https://elasticvue.com/

Starting Port Forwarding

Forwarding from 127.0.0.1:9200 -> 9200
Forwarding from [::1]:9200 -> 9200
```

10.3.1 Monitor ElasticSearch Indexes from Firefox

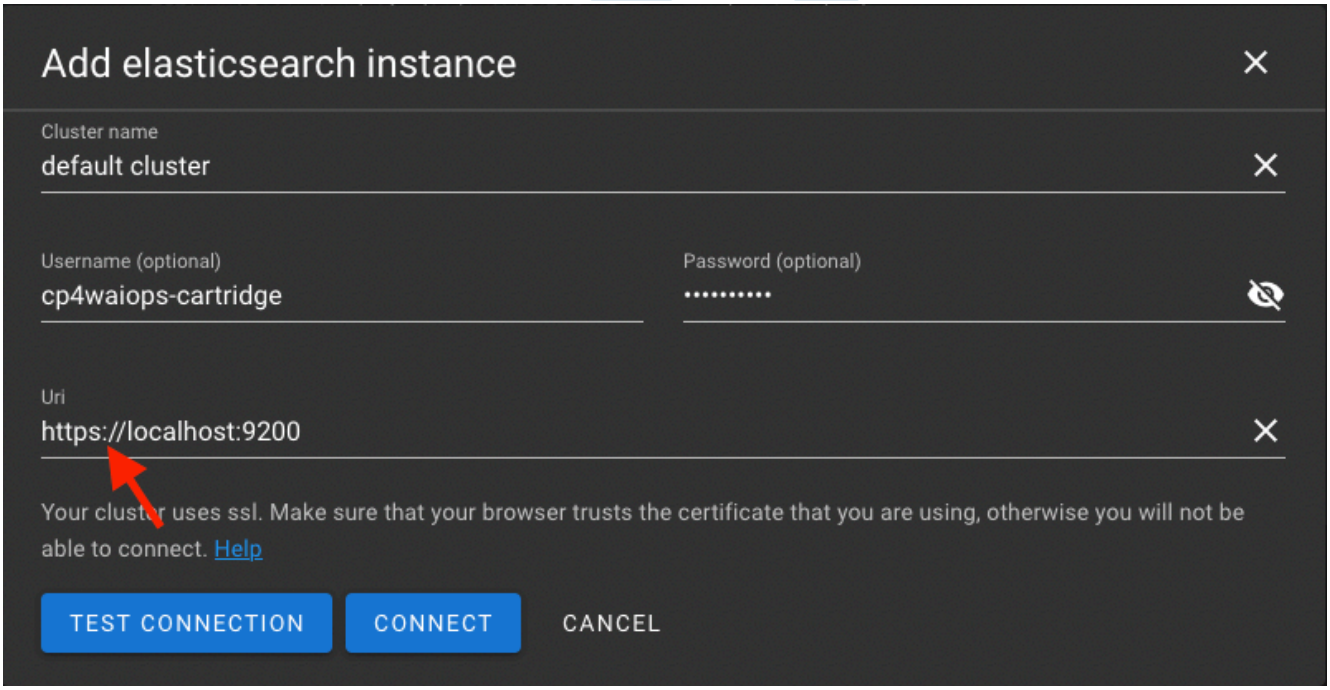
I use the Elasticvue Firefox plugin.

Follow these steps to connects from Elasticvue:

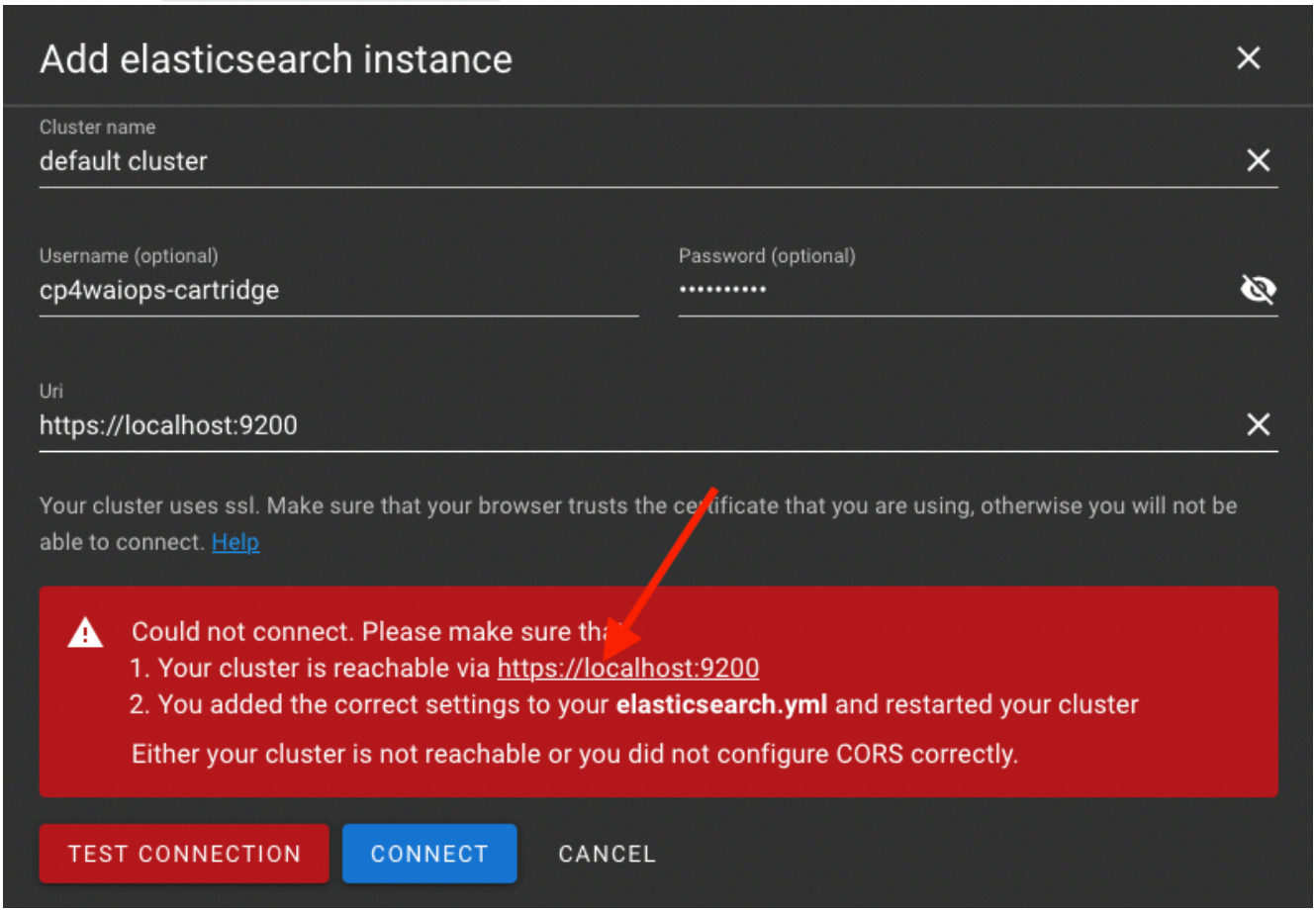
- Select `Add Cluster`



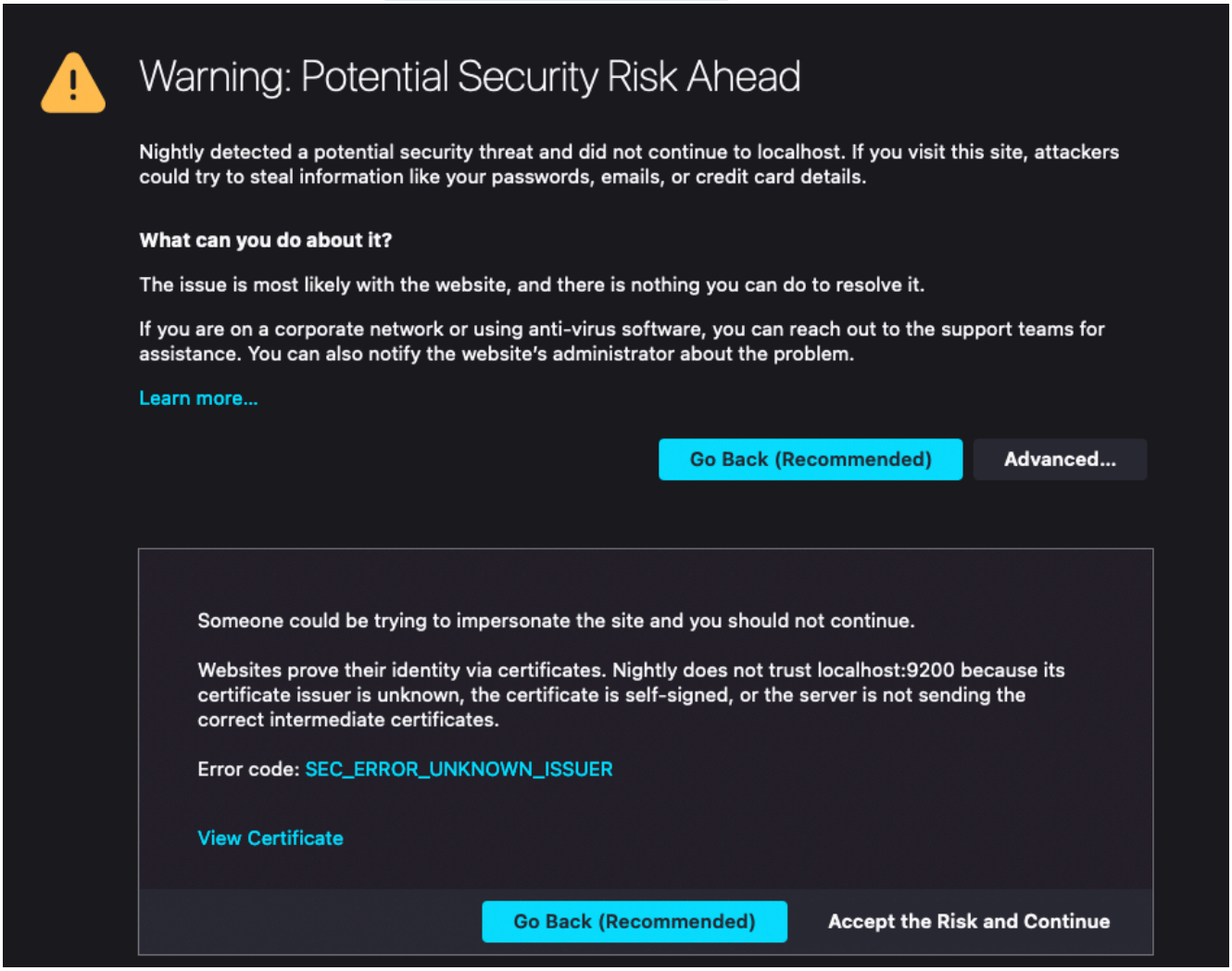
- Put in the credentials and make sure you put `https` and not `http` in the URL



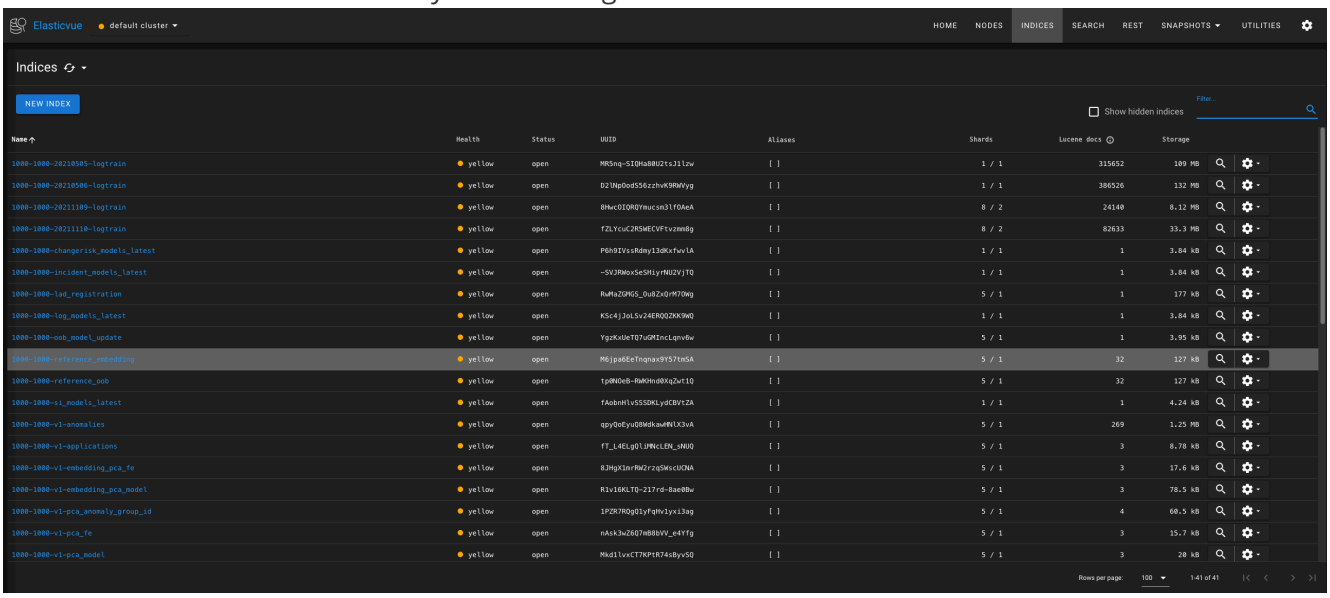
- Click `Test Connection` - you will get an error
- Click on the `https://localhost:9200` URL



- This will open a new Tab, select `Accept Risk and Continue`



- Cancel the login screen and go back to the previous tab
- Click `Connect`
- You should now be connected to your AI Manager ElasticSearch instance



11. Demo the Solution

11.1 Simulate incident

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

In the terminal type

```
./tools/01_demo/incident_robotshop.sh
```

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

ℹ️ Give it a minute or two for all events and anomalies to arrive in Slack.

12. TROUBLESHOOTING

12.1 Check with script

❗ There is a new script that can help you automate some common problems in your CP4WAIOPS installation.

Just run:

```
./tools/10_debug_install.sh
```

and select `Option 1`

12.2 Pods in Crashloop

If the evtmanager-topology-merge and/or evtmanager-ibm-hdm-analytics-dev-inferenceservice are crashlooping, apply the following patches. I have only seen this happen on ROKS.

```
oc patch deployment evtmanager-topology-merge -n <YOUR WAIOPS NAMESPACE> --patch-file
./yaml/waiops/pazch/topology-merge-patch.yaml


oc patch deployment evtmanager-ibm-hdm-analytics-dev-inferenceservice -n <YOUR WAIOPS
NAMESPACE> --patch-file ./yaml/waiops/patch/evtmanager-inferenceservice-patch.yaml
```

12.3 Slack integration not working

See here

12.4 Check if data is flowing

12.4.1 Check Log injection

To check if logs are being injected through the demo script:

1. Launch

   ```
   ./tools/22_monitor_kafka.sh
   ```

2. Select option 4

You should see data coming in.

12.4.2 Check Events injection

To check if events are being injected through the demo script:

1. Launch

   ```
   ./tools/22_monitor_kafka.sh
   ```

2. Select option 3

You should see data coming in.

12.4.3 Check Stories being generated

To check if stories are being generated:

1. Launch

   ```
   ./tools/22_monitor_kafka.sh
   ```

2. Select option 2

You should see data being generated.

12.5 Docker Pull secret

❗ ⚠️ Make a copy of the secret before modifying

❗ ⚠️ On ROKS (any version) and before 4.7 you have to restart the worker nodes after the modification

We learnt this the hard way...

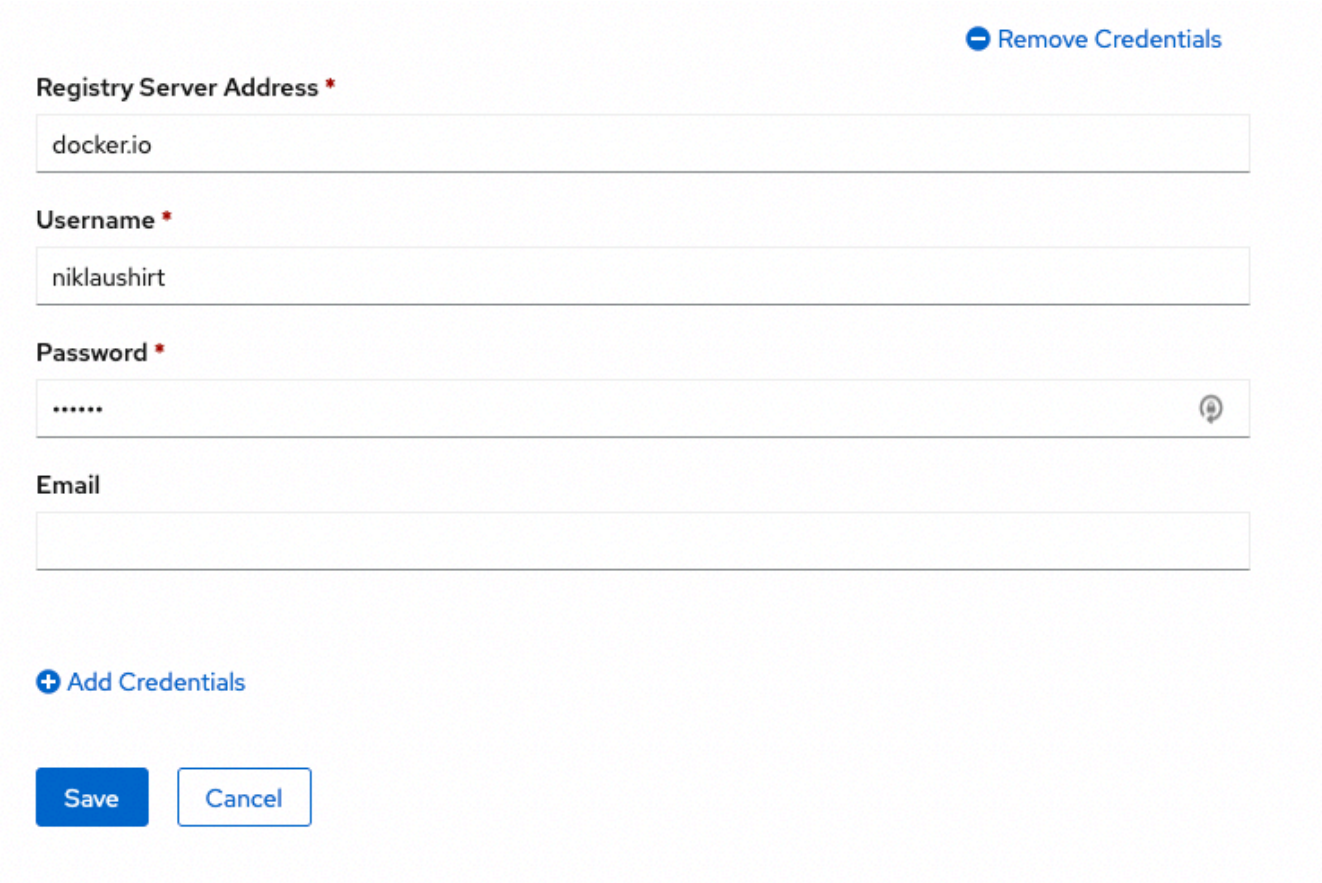
```
oc get secret -n openshift-config pull-secret -oyaml > pull-secret_backup.yaml
```

or more elegant

```
oc get Secret -n openshift-config pull-secret -ojson | jq 'del(.metadata.annotations,
.metadata.creationTimestamp, .metadata.generation, .metadata.managedFields,
.metadata.resourceVersion , .metadata.selfLink , .metadata.uid, .status)' > pull-
secret_backup.json
```

In order to avoid errors with Docker Registry pull rate limits, you should add your Docker credentials to the Cluster.
This can occur especially with Rook/Ceph installation.

- Go to Secrets in Namespace `openshift-config`
- Open the `pull-secret` Secret
- Select `Actions` / `Edit Secret`
- Scroll down and click `Add Credentials`
- Enter your Docker credentials



- Click Save

If you already have Pods in ImagePullBackoff state then just delete them. They will recreate and should pull the image correctly.

13. Uninstall

❗ The scritps are coming from here https://github.com/IBM/cp4waiops-samples.git

If you run into problems check back if there have been some updates.

I have tested those on 3.1.1 as well and it seemed to work (was able to do a complete reinstall afterwards).

Just run:

```
./tools/99_uninstall/3.2/uninstall-cp4waiops.props
```

14. Installing Turbonomic

14.1 Installing Turbonomic

You can install Turbonomic into the same cluster as CP4WAIOPS.

❗ **You need a license in order to use Turbonomic.**

1. Launch

   ```
   ansible-playbook ./ansible/20_install-turbonomic.yaml
   ```

2. Wait for the pods to come up
3. Open Turbonomic
4. Enter the license
5. Add the default target (local Kubernetes cluster is already instrumented with `kubeturbo`)

It can take several hours for the Supply Chain to populate, so be patient.

14.2 Installing kubeturbo

In order to get other Kubernetes clusters to show up in Turbonomic, you have to install `kubeturbo` (your main cluster is already registered).

1. Adapt `./ansible/templates/kubeturbo/my_kubeturbo_instance_cr.yaml` with the Turbonomic URL and the login
2. Launch

   ```
   ansible-playbook ./ansible/20_1_aiops-addons-kubeturbo.yaml
   ```

14.3 Turbo to WAIOPS Gateway

❗ **This is not an officialy supported tool by any means and is still under heavy development!**

In order to push Turbonomic Actions into Event Manager you can use my tool.
This tool needs existing `Business Applications`, you can either integrate with Instana (or other APMs) or create one under Settings/Topology.

1. Adapt the `./ansible/templates/turbo-gateway/create-turbo-gateway.yaml` file

Variable	Default Value	Description
POLLING_INTERVAL	'300'	Poll every X seconds
NOI_SUMMARY_PREFIX	'[Turbonomic] '	Prefix in the event summary
NOI_WEBHOOK_URL	netcool-evtmanager.apps.clustername.domain	Event Manager hostname
NOI_WEBHOOK_PATH	/norml/xxxx	Webhook URL from Event Manager (does not inclue the hostname, only `/norml/xxxx`)
TURBO_API_URL	api-turbonomic.apps.clustername.domain	Turbonomic API URL
TURBO_BA_NAME	'RobotShop:robot-shop'	Turbonomic application name in the format APPNAME:ALERTGROUP. This links an event manager alertgroup with an application
ACTION_STATES	'SUCCEEDED,FAILED,READY,IN_PROGRESS'	The list of ACTION_STATES to filter on
ACTION_TYPES	'MOVE,RESIZE_FOR_PERFORMANCE,RESIZE_FOR_EFFICIENCY,RESIZE'	The list of ACTION_TYPES to filter on
DEBUG_ENABLED	'false'	Enable additional log output
ENTITY_TYPES	'VirtualMachine,Application,PhysicalMachine,ContainerSpec,WorkloadController,Container'	The list of ENTITY_TYPES to filter on
ACTION_START_TIME	'-30m'	Period of time in which actions are retrieved. E.g. -5m, -30m, -1h, -1d, -3d, -7d

2. Create Turbonomic Credentials Secret

   You can either:

   - create the secret from the command line (which will throw a warning for the already existing Secret when installing)

     ```
     oc -n default create secret generic turbo-creds --from-literal=TURBO_USER=<youruser> --from-literal=TURBO_PWD=<yourpw>
     ```

   - replace the secret in the yaml file with

```
```
oc -n default create secret generic turbo-creds --from-literal=TURBO_USER=apiuser --from-literal=TURBO_PWD=turboadmin -o yaml --dry-run=client
```
```

3. Create Generic Webhook in NOI with:

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

4. Launch

```
ansible-playbook ./ansible/20_3_aiops-addons-turbonomic-gateway.yaml
```

14.4 Generate Metrics

❗ **This is not an officialy supported tool by any means and is still under heavy development!**

If you have manually created a `Business Applications` you won't get any ResponseTime and Transactions metrics.
With this tool you can you can add randomized ResponseTime and Transactions metrics to the `Business Application` through the `Data Integration Framework (DIF)`.

> Note: The metrics pod can also serve metrics for other `Entity` types (businessApplication, businessTransaction, service, databaseServer, application)
>
> Note: There is also a Route being created by the installer, so that you can test the URLs.

1. Launch

   ```
   ansible-playbook ./ansible/20_2_aiops-addons-turbonomic-metrics.yaml
   ```

2. Wait for the Pod to become available
3. Add the DIF Target
   2. Go to `Settings/Target Configurations`
   3. Click `New Target`
   4. Select `Custom/DataIngestionFramework`
   5. Put in the URL for the metrics (see below) and a name
   6. Click `Add`
   7. Make sure that Target is green and reads `Validated`

It takes some time for the metrics to start showing up. Polling is every 10 minutes

14.4.1 Test URL

You can use the following URL to test if everything is working:

`http://turbo-dif-service.default:3000/helloworld`

This will create a standalone `Business Application` called `Hello World` without any other `Entities` attached to it.
But with metrics being ingested.

14.4.2 Construct the URL

The URL has the format of:

```
http://turbo-dif-service.default:3000/<TYPE>/<NAME>/<UUID>
```

where:

- TYPE: Type of the `Entity` (businessApplication/businessTransaction/service/databaseServer/application)

- NAME: The name of the `Entity`
- UUID: The UUID that you can find under `Entity Information / Show All / Vendor ID`

So an example might be:
`http://turbo-dif-service.default:3000/service/Service-robot-shop%2Fcatalogue/b2d6fd52-c895-469e-bb98-2a791faefce7`
`http://turbo-dif-service.default:3000/businessApplication/RobotShop/285215220007744`

15. Installing OCP ELK

You can easily install ELK into the same cluster as CP4WAIOPS.

1. Launch

   ```
   ansible-playbook ./ansible/22_install-elk-ocp.yaml
   ```

2. Wait for the pods to come up
3. Open Kibana

16. HUMIO

This is optional

> ❗ This demo supports pre-canned events and logs, so you don't need to install and configure Humio unless you want to do a live integration (only partially covered in this document).

16.1 Install Humio and Fluentbit

16.1.1 Automatic installation

```
ansible-playbook ./ansible/21_install-humio.yaml
```

16.1 Configure Humio

- Create Repository `aiops`
- Get Ingest token (<TOKEN_FOR_HUMIO_AIOPS_REPOSITORY>) (`Settings` / `API tokens`)

16.1.1 Limit retention

This is important as your PVCs will fill up otherwise and Humio can become unavailable.

16.1.1.1 Change retention size for aiops

You have to change the retention options for the aiops repository

16.1.1.2 Change retention size for humio

You have to change the retention options for the humio repository

16.2 Live Humio integration with AIManager (disabled by default)

16.2.1 Humio URL

- Get the Humio Base URL from your browser
- Add at the end `/api/v1/repositories/aiops/query`

16.2.2 Accounts Token

Get it from Humio --> `Owl` in the top right corner / `Your Account` / `API Token`

16.2.3 Create Humio Integration

- In the `AI Manager` "Hamburger" Menu select `Operate` / `Data and tool integrations`
- Under `Humio`, click on `Add Integration`
- Name it `Humio`
- Paste the URL from above (`Humio service URL`)
- Paste the Token from above (`API key`)
- In `Filters (optional)` put the following:

```
"kubernetes.namespace_name" = /robot-shop/
| "kubernetes.container_name" != load
```

- Click `Test Connection`
- Switch `Data Flow` to the `ON` position ❗
- Select `Live data for continuous AI training and anomaly detection`
- Click `Save`