CP4WatsonAIOps V3.2

Demo Environment Installation with OpenShift GitOps/ARGOCD

©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.

Changes

Date	Description	Files
17.01.2022	First Draft	

Installation

1. Easy Install
2. Prerequisites
3. OpenShift GitOps Install
4. AI Manager Base Install
    - Install AI Manager
    - Install Event Manager
5. Configure Applications and Topology
6. Training
7. Slack integration
8. Service Now integration
9. Some Polishing
10. Demo the Solution
11. Troubleshooting
12. Uninstall CP4WAIOPS
13. EventManager Install
    - Configure EventManager
    - Configure Runbooks
14. Installing Turbonomic
15. Installing ELK (optional)
16. Installing Humio (optional)
17. Installing ServiceMest/Istio (optional)
18. Installing and configuring AWX/AnsibleTower (optional)
19. Tips
    1. Setup remote Kubernetes Observer
    2. Generic Webhook to AIManager Event Gateway

> ❗ You can find a PDF version of this guide here: PDF.

Introduction

This repository contains the scrips for installing a Watson AIOps demo environment with an OpenShift GitOps/ARGOCD based installer.

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 4.8.**

❗ **Then EventManager/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

How it works

Helm Charts

The installations are packaged as Helm Charts.
They reside in the ./charts directory.

> For example the `./charts/1_cp4waiops/3.2/aimanager/` directory contains the helm chart for CP4WAIOPS `AI Manager`.

Openshift GitOps

Openshift GitOps is based on ArgoCD.
You can define Applications within Openshift GitOps that are being synched with a GitRepository.

Applications

Openshift GitOps Applications reside in the `./argocd/applications` directory.

Example for CP4WAIOPS AI Manager:

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: cp4waiops-aimanager
  namespace: openshift-gitops
spec:
  destination:
    name: ''
    namespace: cp4waiops
    server: 'https://kubernetes.default.svc'
  source:
    path: charts/cp4waiops/3.2/aimanager
    repoURL: 'https://github.com/niklaushirt/cp4waiops-gitops'
    targetRevision: HEAD
    helm:
      valueFiles:
        - values.yaml
      parameters:
        - name: spec.dockerPassword
          value: >-
            <PULL_TOKEN>
  project: default
  syncPolicy:
    automated:
      prune: true
      selfHeal: true
```

You can either use the provided scripts to install them or create the application directly in ArgoCD.

Make it your own

If you want to modify and/or play around with the values you just have to:

- Clone my repository
- In `./argocd/applications` replace all occurences in of `https://github.com/niklaushirt/cp4waiops-gitops` with your cloned repository

0. Easy Install

I have provided a tool to very easily install the different components.

Get the code

Clone the GitHub Repository

```
git clone https://github.com/niklaushirt/cp4waiops-gitops.git --branch 3.2-stable
```

First launch

Just launch:

```
./00_install.sh
```

For a vanilla install you will see this:

Select

- Option 1 to prepare the OpenShift GitOps Installation
- Option 2 or 3 depending on your environment

The tool quits after OpenShift GitOps has been installed.

Installing

Just re-launch:

```
./00_install.sh
```

If OpenShift GitOps has been correctly installed you will see this:

```
🚀 CloudPak for Watson AIOPs - INSTALL

ℹ️ This script provides different options to install CP4WAIOPS demo environments through OpenShift GitOps(ArgoCD)

🚀 Connect to OpenShift GitOps to check your deployments

🌏 URL:      https://openshift-gitops-server-openshift-gitops.itzroks-270003b[redacted]9553d37d5f2ee142bd6-0000.eu-gb.containers.appdomain.cloud

👷 User:       admin
🔐 Password:   JxQ3[redacted]uLp9


📩 Prerequisites Install
      ✅  - Install Openshift GitOps
      2  - Install Prerequisites Mac                          - Install Prerequisites for Mac
      3  - Install Prerequisites Ubuntu                       - Install Prerequisites for Ubuntu

🚀 CP4WAIOPS - Base Install
      ✅  - Install AI Manager
      12  - Install Event Manager                             - Install CP4WAIOPS Event Manager Component

🌏 Solutions
      ✅  - Install Turbonomic
      ✅  - Install Humio
      23  - Install AWX                                       - Install AWX (open source Ansible Tower)

📛 CP4WAIOPS Addons
      ✅  - Install CP4WAIOPS Demo Application
      ✅  - Install OpenLdap
      ✅  - Install RobotShop


      0  -  Exit

Enter selection:
```

Select the options you want to install.
The ones marked with ✅ have already been detected as being present in the cluster.

1. Prerequisites

1.1 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 (Humio, Turbonomic,...)

1.2 Tooling

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

> ℹ️ This can be done with the Easy Install Tool

- 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)

> ℹ️ This can be done with the [Easy Install Tool]() - **Option 2**

Or just run:

```
sudo ./argocd/02_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
ansible-galaxy collection install community.kubernetes:1.2.1
brew install kafkacat
brew install node
brew install wget
npm install elasticdump -g
brew install jq
brew install argocd

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)

> ℹ️ This can be done with the [Easy Install Tool]() - **Option 3**

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

```
sudo ./argocd/03_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
ansible-galaxy collection install community.kubernetes:1.2.1
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/argoproj/argo-cd/releases/download/v2.2.2/argocd-linux-amd64
-o argocd
sudo mv argocd /usr/local/bin/argocd

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
```

1.3 Get the scripts and code from GitHub

1.3.1 Clone the GitHub Repository (preferred)

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

```
git clone https://github.com/niklaushirt/cp4waiops-gitops.git --branch 3.2-stable
```

You can create your GIT token here.

1.3.2 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. OpenShift GitOps Install

2.1 Install OpenShift GitOps

> ℹ️ This can be done with the [Easy Install Tool](#) - **Option 1**

Or just run the following:

```
./argocd/01_install_gitops.sh
```

2.2 Accessing OpenShift GitOps

Once the installation finished, you can access OpenShift GitOps either via the link in the Terminal

```
🚀 CloudPak for Watson AIOps 3.2 - AI MANAGER Install

🚀 Starting installation

📩 Create Namespace
Error from server (AlreadyExists): namespaces "argocd" already exists

📩 Create Openshift GitOps Operator
subscription.operators.coreos.com/openshift-gitops-operator unchanged
✅ OK

📩 Create Openshift GitOps Instance
error: failed to create clusterrolebinding: clusterrolebindings.rbac.authorization.k8s.io "argocd-admin" already exists
error: failed to create clusterrolebinding: clusterrolebindings.rbac.authorization.k8s.io "default-admin" already exists
argocd.argoproj.io/openshift-gitops unchanged
✅ OK

🔐 Login Credentials

🌏 URL:      https://openshift-gitops-server-openshift-gitops.itzroks-270[redacted].eu-de.containers.appdomain.cloud

👨 User:      admin
🔐 Password:  wes2Z[redacted]Wth3OPE
```

or the menu in the Openshift Web Interface:

In both cases use the login credentials from the install script.

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

3. CP4WAIOPS Base Install

3.1 Install AI Manager

3.1.1 Adapt configuration

If needed, adapt the `./charts/1_cp4waiops/3.2/aimanager/values.yaml` file with the desired parameters:

```
spec:

  ## AI Manager catalog source image
  ##
  imageCatalog: icr.io/cpopen/ibm-operator-catalog:latest

  ## dockerUsername is the usrname of IBM® Entitled Registry.
  ## It is used to create a docker-registry secret to enable your deployment to pull
the AI Manager images
  ## from the IBM® Entitled Registry.
  ## Default is cp
  dockerUsername: cp

  ## Obtain the entitlement key that is assigned to your IBMid.
  ## Log in to MyIBM Container Software Library: https://myibm.ibm.com/products-
services/containerlibrary
  ## Opens in a new tab with the IBMid and password details
  ## that are associated with the entitled software.
  ## DO NOT Commit your docker password here, but always specify it in UI or CLI when
creating the ArgoCD app.
  ##
  dockerPassword: <will be set by install script>

  ## storageClass is the storage class that you want to use.
  ## If the storage provider for your deployment is Red Hat OpenShift Data Foundation,
  ## previously called Red Hat OpenShift Container Storage, then set this to ocs-
storagecluster-cephfs
  ##
  storageClass: ibmc-file-gold-gid

  ## If the storage provider for your deployment is Red Hat OpenShift Data Foundation,
  ## previously called Red Hat OpenShift Container Storage, then set this to ocs-
storagecluster-ceph-rbd
  storageClassLargeBlock: ibmc-file-gold-gid
```

```yaml
  aiManager:

    installationName: ibm-aiops
    ## Enable AI Manager
    ##
    enabled: true

    ## A channel defines a stream of updates for an Operator and is used to roll out
updates for subscribers.
    ## For example, if you want to install AI Manager 3.2, the channel should be v3.2
    ##
    channel: v3.2

    ## size is the size that you require for your AI Manager installation. It can be
small or large.
    ## More information: https://www.ibm.com/docs/en/cloud-paks/cloud-pak-watson-
aiops/3.2.0?topic=requirements-ai-manager
    size: small

    ## namespace is the project (namespace) that you want to create the AI Manager
instance in.
    ## You must create a custom project (namespace) and not use the default, kube-
system,
    ## kube-public, openshift-node, openshift-infra, or openshift projects
(namespaces).
    ## This is because AI Manager uses Security Context Constraints (SCC),
    ## and SCCs cannot be assigned to pods created in one of the default OpenShift
projects (namespaces).
    ##
    namespace: cp4waiops


    ## Install demo content
    democontent:

        ## Install RobotShop Application
        robotshop:
          install: true

        ## Install and register OpenLdap
        ldap:
          install: true
          ldapDomain: ibm.com
          ldapBase: dc=ibm,dc=com
          ldapPassword: P4ssw0rd!
```

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

> ℹ️ This can be done with the Easy Install Tool - **Option 11**

Or just run:

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


Example:
./argocd/11_install_ai_manager.sh -t
eyJhbGciOiJIUzI1vvvvNzQzMjgsImp0aSI6IjRjYTM3gsdgdMzExNjQxZDdiMDJhMjRmMGMxMWgdsmZhIn0.Z-
rqfSLJA-R-ow__tI3RmLx4mssdggdabvdcgdgYEkbYY
```

This will install:

- CP4WAIOPS AI Manager
- RobotShop Application (if enabled)
- OpenLDAP (if enabled)
- Register OpenLDAP with AI Manager (if enabled)

3.1.4 Verify installation

Click on the `cp4waiops-aimanager` Application Tile

and check that `Sync Status` and `Sync Result` are OK

3.2 Install Event Manager

To get the token, see here

3.2.1 Adapt configuration

If needed, adapt the `./charts/1_cp4waiops/3.2/eventmanager/values.yaml` file with the desired parameters:

```yaml
spec:

  ## AI Manager catalog source image
  ##
  imageCatalog: icr.io/cpopen/ibm-operator-catalog:latest

  ## dockerUsername is the usrname of IBM® Entitled Registry.
  ## It is used to create a docker-registry secret to enable your deployment to pull
the AI Manager images
  ## from the IBM® Entitled Registry.
  ## Default is cp
  dockerUsername: cp

  ## Obtain the entitlement key that is assigned to your IBMid.
  ## Log in to MyIBM Container Software Library: https://myibm.ibm.com/products-
services/containerlibrary
  ## Opens in a new tab with the IBMid and password details
  ## that are associated with the entitled software.
  ## DO NOT Commit your docker password here, but always specify it in UI or CLI when
creating the ArgoCD app.
  ##
  dockerPassword: <will be set by install script>

  ## storageClass is the storage class that you want to use.
  ## If the storage provider for your deployment is Red Hat OpenShift Data Foundation,
  ## previously called Red Hat OpenShift Container Storage, then set this to ocs-
storagecluster-cephfs
  ##
  storageClass: ibmc-file-gold-gid

  ## If the storage provider for your deployment is Red Hat OpenShift Data Foundation,
  ## previously called Red Hat OpenShift Container Storage, then set this to ocs-
storagecluster-ceph-rbd
  storageClassLargeBlock: ibmc-file-gold-gid


  eventManager:
    # eventManager version
```

```
    version: 1.6.3.2


    ## A channel defines a stream of updates for an Operator and is used to roll out
updates for subscribers.
    ## For example, if you want to install Evemt Manager 1.5, the channel should be
v1.5
    ##
    channel: v1.5

    ## Deployment type (trial or production)
    ##
    deploymentType: trial

    ## namespace is the project (namespace) that you want to create the Event Manager
instance in.
    ## You must create a custom project (namespace) and not use the default, kube-
system,
    ## kube-public, openshift-node, openshift-infra, or openshift projects
(namespaces).
    ##
    namespace: cp4waiops-evtmgr
```

3.2.2 🚀 Start installation

> ℹ️ This can be done with the [Easy Install Tool]() - **Option 12**

Or just run:

```
./argocd/12_install_event_manager.sh -t <PULL_SECRET_TOKEN> [-v true]


Example:
./argocd/12_install_event_manager.sh -t
eyJhbGciOiJIUzI1NiJ9.eyJpc3adsgJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE1Nzg0NzQzMjgsImp0aSI6Ij
RjYTM3gsdgdMzExNjQxZDdiMDJhMjRmMGMxMWgdsmZhIn0.Z-rqfSLJA-R-
ow__tI3RmLx4mssdggdabvdcgdgYEkbYY
```

This will install:

- CP4WAIOPS EventManager
- Gateway

3.2.3 Verify installation

Click on the `cp4waiops-eventmanager` Application Tile

and check that `Sync Status` and `Sync Result` are OK

3.3 Install OpenLDAP

> ❗ Only needed if disabled in AI Manager Base Install

3.3.1 Adapt configuration

If needed, adapt the `./charts/2_addons/ldap/values.yaml` file with the desired parameters:

```
ldapDomain: ibm.com
ldapBase: dc=ibm,dc=com

ldapPassword: P4ssw0rd!

aiManagerNamespace: cp4waiops
```

3.3.2 🚀 Start installation

> ℹ️ This can be done with the Easy Install Tool - **Option 32**

Or just run:

```
./argocd/32_install_ldap.sh
```

This will install:

- OpenLDAP
- Register OpenLDAP with AI Manager

3.3.3 Verify installation

Click on the `ldap` Application Tile

and check that `Sync Status` and `Sync Result` are OK

3.3.4 Configure LDAP Users

1. Log in to AI Manager as admin
2. Select `Administration/Access` control from the "Hamburger manu"
3. Click on the `Identity provider configuration` (upper right) you should get the LDAP being registered
4. Go back
5. Select `User Groups Tab`
6. Click `New User Group`
7. Call it `demo`
8. Click `Next`
9. Click on `Identity provider groups`
10. Search for `demo`
11. Select `cn=demo,ou=Groups,dc=ibm,dc=com`
12. Click `Next`
13. Select `Administrator` rights
14. Click `Next`
15. Click `Create`

Now you will be able to login with all LDAP users that are part of the demo group (for example demo/P4ssw0rd!).

You can check/modify those in the OpenLDAPAdmin interface that you can access with the credentials described in 3.3.

3.4 Install RobotShop

> ❗ Only needed if disabled in AI Manager Base Install

3.4.1 🚀 Start installation

> ℹ️ This can be done with the [Easy Install Tool]() - **Option 33**

Or just run:

```
./argocd/33_addons_robotshop.sh
```

This will install:

- OpenLDAP
- Register OpenLDAP with AI Manager

3.4.2 Verify installation

Click on the `robot-shop` Application Tile

and check that `Sync Status` and `Sync Result` are OK

3.5 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
```

3.6 Check status of installation

At any moment you can run `./tools/11_check_install.sh` or for a more in-depth examination and troubleshooting `./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:

```
./tools/60_load_robotshop_topology_aimanager.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 `Define Application`
- Select `robot-shop` namespace
- Click `Next`
- Click `Next`
- Name your Application (RobotShop)
- If you like check `Mark as favorite`
- Click `Define Application`

5. Training

5.1 Prepare Training

5.1.1 Create Kafka Humio Log Training Integration

- In the `AI Manager` "Hamburger" Menu select `Define` / `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`

5.1.2 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 `EvetnManager`
- Click `Next`
- Select `Data Source` / `Events`
- Select `Mapping Type` / `NOI`
- Click `Next`
- Toggle `Data Flow` to the `ON` position
- Click `Save`

5.1.3 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
```

5.2 Load Training Data

Run the following scripts to inject training data:

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

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

5.3 Train Log Anomaly

5.3.1 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`

5.3.2 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)

5.4 Train Event Grouping

5.4.1 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`

5.4.2 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

5.5 Train Incident Similarity

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

5.5.1 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`

5.5.2 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

5.6 Train Change Risk

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

5.6.1 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`

5.6.2 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`

6. Slack integration

6.1 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

6.2 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.

```
WAIOPS_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 iaf-system-automationui-aui-zen-ca -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 -
#oc create secret generic -n $WAIOPS_NAMESPACE external-tls-secret --from-
file=cert.crt=cert.crt --from-file=cert.key=cert.key --dry-run=client -o yaml | oc
apply -f -
# scale down nginx
REPLICAS=2
oc scale Deployment/ibm-nginx --replicas=0
# scale up nginx
sleep 3
oc scale Deployment/ibm-nginx --replicas=${REPLICAS}
rm 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
```

And then restart the Slack integration Pod

```
oc delete pod $(oc get po -n $WAIOPS_NAMESPACE|grep slack|awk '{print$1}') -n
$WAIOPS_NAMESPACE --grace-period 0 --force
```

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.

6.3 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
```

7. Service Now integration

7.1 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`.

8. Some Polishing

8.1 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

8.2 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

8.3 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
```

8.3.1 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

9. Demo the Solution

9.1 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.

10. TROUBLESHOOTING

10.1 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`

10.2 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.

```
export WAIOPS_NAMESPACE=cp4waiops

oc patch deployment evtmanager-topology-merge -n $WAIOPS_NAMESPACE --patch-file
./yaml/waiops/pazch/topology-merge-patch.yaml


oc patch deployment evtmanager-ibm-hdm-analytics-dev-inferenceservice -n
$WAIOPS_NAMESPACE --patch-file ./yaml/waiops/patch/evtmanager-inferenceservice-
patch.yaml
```

10.3 Pods with Pull Error

If the ir-analytics or cassandra job pods are having pull errors, apply the following patches.

```
export WAIOPS_NAMESPACE=cp4waiops

kubectl patch -n $WAIOPS_NAMESPACE serviceaccount aiops-topology-service-account -p
'{"imagePullSecrets": [{"name": "ibm-entitlement-key"}]}'
kubectl patch -n $WAIOPS_NAMESPACE serviceaccount aiops-ir-analytics-spark-worker -p
'{"imagePullSecrets": [{"name": "ibm-entitlement-key"}]}'
kubectl patch -n $WAIOPS_NAMESPACE serviceaccount aiops-ir-analytics-spark-pipeline-
composer -p '{"imagePullSecrets": [{"name": "ibm-entitlement-key"}]}'
kubectl patch -n $WAIOPS_NAMESPACE serviceaccount aiops-ir-analytics-spark-master -p
'{"imagePullSecrets": [{"name": "ibm-entitlement-key"}]}'
kubectl patch -n $WAIOPS_NAMESPACE serviceaccount aiops-ir-analytics-probablecause -p
'{"imagePullSecrets": [{"name": "ibm-entitlement-key"}]}'
kubectl patch -n $WAIOPS_NAMESPACE serviceaccount aiops-ir-analytics-classifier -p
'{"imagePullSecrets": [{"name": "ibm-entitlement-key"}]}'
kubectl patch -n $WAIOPS_NAMESPACE serviceaccount aiops-ir-lifecycle-eventprocessor-ep
-p '{"imagePullSecrets": [{"name": "ibm-entitlement-key"}]}'
oc delete pod $(oc get po -n $WAIOPS_NAMESPACE|grep ImagePull|awk '{print$1}') -n
$WAIOPS_NAMESPACE
```

10.4 Camel-K Handlers Error

If the scm-handler or snow-handler pods are not coming up, apply the following patches.

```
export WAIOPS_NAMESPACE=cp4waiops

oc patch vaultaccess/ibm-vault-access -p '{"spec":{"token_period":"760h"}}' --
type=merge -n $WAIOPS_NAMESPACE
oc delete pod $(oc get po -n $WAIOPS_NAMESPACE|grep 0/| grep -v "Completed"|awk
'{print$1}') -n $WAIOPS_NAMESPACE
```

10.5 Slack integration not working

See here

10.6 Check if data is flowing

10.6.1 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.

10.6.2 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.

10.6.3 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.

10.7 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.

11. 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
```

12. EventManager Installation

12.1. Configure EventManager

❗ You only have to do this if you have installed EventManager/NOI (As described in Chapter 3.2). For basic demoing with AI MAnager this is not needed.

12.1.1 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`

12.1.2 Create REST Observer to Load Topologies

- In the `Event Manager` "Hamburger" Menu select `Administration` / `Topology Management`
- Under `Observer jobs` click `Configure`
- Click `Add new job`
- Under `REST`, click on `Configure`
- Choose `bulk_replace` for `Job Type`
- Set `Unique ID` to `listenJob` (important!)
- Set `Provider` to `listenJob`
- Click `Save`

12.1.3 Create Merge Rules for Kubernetes Observer

Launch the following:

```
./tools/61_load_robotshop_topology_noi.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.

12.1.4 EventManager Webhooks

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

12.1.4.1 Generic Demo Webhook

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.

12.1.5 Create custom Filter and View in EventManager/ (optional)

12.1.5.1 Filter

Duplicate the `Default` filter and set to global.

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

12.1.5.2 View

Duplicate the `Example_IBM_CloudAnalytics` View and set to global.

- Name: AIOPS

Configure to your likings.

12.1.6 Create Templates for Topology Grouping (optional)

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

- In the EventManager "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

12.1.7 Create grouping Policy

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

12.1.8 Create EventManager/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

12.2 Configure Runbooks

12.2.1 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.

12.2.2 Create the EventManager/NOI Integration

12.2.2.1 In EventManager/NOI

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

12.2.2.2 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.

12.2.3 Create Automation

12.2.3.1 Connect to Cluster

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

```
oc login --token=$token --server=$ocp_url --insecure-skip-tls-verify
```

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)
```

12.2.3.2 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
```

12.2.4 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`

12.2.5 Add Runbook Triggers

- `Triggers` / `New Trigger`
- Name and Description: `Mitigate RobotShop Problem`
- Conditions
  - Name: RobotShop
  - Attribute: Node
  - Operator: Equals
  - Value: mysql-instana or mysql-predictive
- 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`

13. Installing Turbonomic

13.1 Installing Turbonomic

You can install Turbonomic into the same cluster as CP4WAIOPS.

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

> ℹ️ This can be done with the [Easy Install Tool]() - **Option 21**

Or just:

1. Launch

   ```
   ./argocd/21_addons_turbonomic.sh
   ```

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.

13.2 Verify installation

Click on the `turbonomic` Application Tile

and check that `Sync Status` and `Sync Result` are OK

14. Installing OCP ELK

❗ NOT WORKING YET

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

1. Launch

   ```
   xxxx
   ```

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

15. HUMIO

> ❗ 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).

15.1 Install Humio and Fluentbit

> ℹ️ This can be done with the Easy Install Tool - **Option 22**

Or just launch the following and this should automatically install:

- Kafka
- Zookeeper
- Humio Core
- Humio Repository
- Humio Ingest Token
- Fluentbit

```
./argocd/22_addons_humio.sh -l <HUMIO_LICENSE>
```

Example:

```
./argocd/22_addons_humio.sh -l
eyJhbGciOiJFUzI1NiJyyyyyyyyyyyyyyyyyyyyQCtxzXF5wLjWCkcyOcbQ5mqU9yow_UoqtnWBOS_Z9DgLgIhA
LCMDC00HunDMk62S6GzDHIm9rYtZ0aWmdRTrr_kesMa
```

15.2 Verify installation

Click on the `humio` Application Tile

and check that `Sync Status` and `Sync Result` are OK

15.3 Live Humio integration with AIManager

15.3.1 Humio URL

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

15.3.2 Accounts Token

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

15.3.3 Create Humio Log Integration

- In the `AI Manager` "Hamburger" Menu select `Operate` / `Data and tool integrations`
- Under `Humio`, click on `Add Connection`
- Click `Connect`
- 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" = web or ratings or catalogue
```

- Click `Next`
- Put in the following mapping:

```
{
  "codec": "humio",
  "message_field": "@rawstring",
  "log_entity_types": "clusterName, kubernetes.container_image_id, kubernetes.host,
kubernetes.container_name, kubernetes.pod_name",
  "instance_id_field": "kubernetes.container_name",
  "rolling_time": 10,
  "timestamp_field": "@timestamp"
}
```

- Click `Test Connection`
- Switch `Data Flow` to the `ON` position ❗
- Select the option for your use case:
  - `Live data for continuous AI training and anomaly detection` if you want to enable log anomaly detection
  - `Live data for initial AI training` if you want to start ingesting live data for later training

- `Historical data for initial AI training` if you want to ingest historical data to start training rapidly
- Click `Done`

15.3.4 Create Humio Events Integration

Events integration is done via EventManager/NOI.

For the time being this only takes the first alert being pushed over (no way to handle arrays).
The native Humio integration seems to have a bug that gives "mergeAdvanced is not a function".

15.3.4.1 On the EventManager/NOI side

Create a Webhook integration:

Field	Value
Severity	"Critical"
Summary	alert.name
Resource name	events[0]."kubernetes.container_name"
Event type	events[0]."kubernetes.namespace_name"

With this sample payload:


```
{
  "repository": "aiops",
  "timestamp": "2021-11-19T15:50:04.958Z",
  "alert": {
    "name": "test1",
    "description": "",
    "query": {
      "queryString": "\"kubernetes.container_name\" = ratings\n| @rawstring = /error/i
",
      "end": "now",
      "start": "2s"
    },
    "notifierID": "Rq4a9KUbomSIBvEcdC7kzzmdBtPI3yPb",
    "id": "rCA2w5zaIE6Xr3RKlFfhAxqqbGqGxGLC"
  },
  "warnings": "",
  "events": [
    {
      "kubernetes.annotations.openshift_io/scc": "anyuid",
      "kubernetes.annotations.k8s_v1_cni_cncf_io/network-status": "[{\n    \"name\":
\"k8s-pod-network\",\n    \"ips\": [\n        \"172.30.30.153\"\n    ],\n
\"default\": true,\n    \"dns\": {}\n}]",
```

```
      "kubernetes.annotations.cni_projectcalico_org/podIPs": "172.30.30.153/32",
      "@timestamp.nanos": "0",
      "kubernetes.annotations.k8s_v1_cni_cncf_io/networks-status": "[{\n    \"name\": \"k8s-pod-network\",\n    \"ips\": [\n        \"172.30.30.153\"\n    ],\n    \"default\": true,\n    \"dns\": {}\n}]",
      "kubernetes.pod_name": "ratings-5d9dff56bd-864kq",
      "kubernetes.labels.service": "ratings",
      "kubernetes.annotations.cni_projectcalico_org/podIP": "172.30.30.153/32",
      "kubernetes.host": "10.112.243.226",
      "kubernetes.container_name": "ratings",
      "kubernetes.labels.pod-template-hash": "5d9dff56bd",
      "kubernetes.docker_id": "87a98617a14684c02d9d52a6245af377f8b1a246d196f232cad494a7a2d125b7",
      "@ingesttimestamp": "1637337004272",
      "kubernetes.container_hash": "docker.io/robotshop/rs-ratings@sha256:4899c686c2494647836633426204 25dc8c75a5d59ca55c247cf6aec62a5fff1a",
      "kubernetes.container_image": "docker.io/robotshop/rs-ratings:latest",
      "#repo": "aiops",
      "@timestamp": 1637337003872,
      "kubernetes.namespace_name": "robot-shop",
      "@timezone": "Z",
      "@rawstring": "2021-11-19T09:50:03.872288692-06:00 stdout F [2021-11-19 15:50:03] php.INFO: User Deprecated: Since symfony/http-kernel 5.3: \"Symfony\\Component\\HttpKernel\\Event\\KernelEvent::isMasterRequest()\" is deprecated, use \"isMainRequest()\" instead. {\"exception\":\"[object] (ErrorException(code: 0): User Deprecated: Since symfony/http-kernel 5.3: \\\"Symfony\\\\Component\\\\HttpKernel\\\\Event\\\\KernelEvent::isMasterRequest()\\\" is deprecated, use \\\"isMainRequest()\\\" instead. at /var/www/html/vendor/symfony/http-kernel/Event/KernelEvent.php:88)\"} []",
      "@id": "tiMU0F8kdNf6x0qMduS9T31q_269_400_1637337003",
      "kubernetes.pod_id": "09d64ec8-c09f-4650-871f-adde27ca863e",
      "#type": "unparsed",
      "kubernetes.annotations.cni_projectcalico_org/containerID": "337bf300371c84500756a6e94e58b2d8ee54a1b9d1bc7e38eb410f1c1bbd6991"
    }
  ],
  "numberOfEvents": 1
}
```


15.3.4.2 On Humio:

- Create Action:
  - Use the Webbhook from EventManager/NOI
  - Select `Skip Certificate Validation`
  - Click `Test Action` and check that you get it in EventManager/NOI Events
- Create Alert:
  - With Query (for example):

```
"kubernetes.container_name" = ratings
| @rawstring = /error/i
```

- Time Window 2 seconds
- 1 second throttle window
- Add action from above

15.3.5 Easily simulate erros

Simulate MySQL error by cutting the communication with the Pod:

```
oc patch -n robot-shop service mysql -p '{"spec": {"selector": {"service": "mysql-deactivate"}}}'
```

Restore the communication:

```
oc patch -n robot-shop service mysql -p '{"spec": {"selector": {"service": "mysql"}}}'
```

16. ServiceMesh

❗ NOT WORKING YET

You can easily install ServiceMesh/Istio into the same cluster as CP4WAIOPS.

This will instrument the RobotShop Application at the same time.

1. Launch

```
xxx
```

2. Wait for the pods to come up
3. You can get the different URLs (RobotShop, Kibana, Grafana, Jaeger) by launching:

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

17. AWX

You can easily install AWX (OpenSource Ansible Tower) into the same cluster as CP4WAIOPS.

17.1. Install AWX

> ℹ️ This can be done with the Easy Install Tool - **Option 23**

Or just do:

1. Launch

   ```
   ./argocd/23_addons_awx.sh
   ```

2. Wait for the pods to come up
3. You can get the URLs and access credentials by launching:

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

17.2 Verify installation

Click on the `awx` Application Tile

and check that `Sync Status` and `Sync Result` are OK

17.3. Configure AWX

There is some demo content available to RobotShop.

1. Log in to AWX
2. Add a new Project
    1. Name it `DemoCP4WAIOPS`
    2. Source Control Credential Type to `Git`
    3. Set source control URL to `https://github.com/niklaushirt/ansible-demo`
    4. Save
3. Add new Job Template
    1. Name it `Mitigate Robotshop Ratings Outage`
    2. Select Inventory `Demo Inventory`
    3. Select Project `DemoCP4WAIOPS`
    4. Select Playbook `cp4waiops/robotshop-restart/start-ratings.yaml`
    5. Select `Prompt on launch` for `Variables` ❗
    6. Save

17.4. Configure AWX Integration

In EventManager:

1. Select `Administration` / `Integration with other Systems`
2. Select `Automation type` tab
3. For `Ansible Tower` click `Configure`
4. Enter the URL and credentials for your AWX instance (you can use the defautl `admin` user)
5. Click Save

17.5. Configure Runbook

In EventManager:

1. Select `Automations` / `Runbooks`
2. Select `Library` tab
3. Click `New Runbook`
4. Name it `Mitigate Robotshop Ratings Outage`
5. Click `Add automated Step`
6. Select the `Mitigate Robotshop Ratings Outage` Job
7. Click `Select this automation`
8. Select `New Runbook Parameter`
9. Name it `ClusterCredentials`
10. Input the login credentials in JSON Format (get the URL and token from the 20_get_logins.sh script)

```
{
  "my_k8s_apiurl": "https://c117-e.xyz.containers.cloud.ibm.com:12345",
  "my_k8s_apikey": "PASTE YOUR API KEY"
}
```

11. Click Save
12. Click Publish

Now you can test the Runbook by clicking on `Run`.

18. Tips

18.1 Setup remote Kubernetes Observer

18.1.1. Get Kubernetes Cluster Access Details

As part of the kubernetes observer, it is required to communicate with the target cluster. So it is required to have the URL and Access token details of the target cluster.

Do the following.

18.1.1.1. Login

Login into the remote Kubernetes cluster on the Command Line.

18.1.1.2. Access user/token

Run the following:

```
./tools/97_addons/k8s-remote/remote_user.sh
```

This will create the remote user if it does not exist and print the access token (also if you have already created).

Please jot this down.

18.1.1. Create Kubernetes Observer Connection

- 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 `Load` for Connection Type
- Input the URL you have gotten from the step above in `Kubernetes master IP address` (without the https://)
- Input the port for the URL you have gotten from the step above in `Kubernetes API port`
- Input the `Token` you have gotten from the step above
- Set `Trust all certificates by bypassing certificate verification` to `On`

- 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`

18.2 AiManager Event Gateway

A Simple Webhook to Kafka Gateway for CP4WAIOPS.
This allows you to push generic JSON to AIManager Events throught a Webhook into Kafka.

> Source code is included if you want to mess around a bit.

18.2.1 Message mapping Parameters

Those Strings define how the message is being decoded.

To adapt the mapping parameters to your needs, you have to modify in the `cp4waiops-event-gateway-config` ConfigMap in file `./tools/97_addons/webhook/create-cp4mcm-event-gateway.yaml`.

The following paramters have to be mapped:

```
ITERATE_ELEMENT: 'events'
NODE_ELEMENT: 'kubernetes.container_name'
ALERT_ELEMENT: 'kubernetes.namespace_name'
SUMMARY_ELEMENT: '@rawstring'
TIMESTAMP_ELEMENT: '@timestamp'
URL_ELEMENT: 'none'
SEVERITY_ELEMENT: '5'
MANAGER_ELEMENT: 'KafkaWebhook'
```

1. The **`ITERATE_ELEMENT`** is the element of the Message that we iterate over.
   This means that the Gateway will get the **`ITERATE_ELEMENT`** element and iterate, map and push all messages in the array.
2. The sub-elements that will be mapped for each element in the array are:
   - Node
   - AlertGroup
   - Summary
   - URL
   - Severity
   - Manager
   - Timestamp

> Any element that cannot be found will be defaulted by the indicated value.
> Example for Severity: If we put the mapping value "5" in the config, this probably won't correspond to a JSON key and the severity for all messages is forced to 5.

> Exception is **`Timestamp`** which, when not found will default to the current EPOCH date.

18.2.2 Getting the Kafka Conncetion Parameters

This gives you the Parameters for the Kafka Connection that you have to modify in the **`cp4waiops-event-gateway-config`** ConfigMap in file **`./tools/97_addons/webhook/create-cp4mcm-event-gateway.yaml`**.

```
export WAIOPS_NAMESPACE=cp4waiops
export KAFKA_TOPIC=$(oc get kafkatopics -n $WAIOPS_NAMESPACE | grep -v
cp4waiopscp4waiops| grep cp4waiops-cartridge-alerts-$EVENTS_TYPE| awk '{print $1;}')
export KAFKA_USER=$(oc get secret ibm-aiops-kafka-secret -n $WAIOPS_NAMESPACE --
template={{.data.username}} | base64 --decode)
export KAFKA_PWD=$(oc get secret ibm-aiops-kafka-secret -n $WAIOPS_NAMESPACE --
template={{.data.password}} | base64 --decode)
export KAFKA_BROKER=$(oc get routes iaf-system-kafka-0 -n $WAIOPS_NAMESPACE -
o=jsonpath='{.status.ingress[0].host}{"\n"}'):443
export CERT_ELEMENT=$(oc get secret -n $WAIOPS_NAMESPACE kafka-secrets  -o 'go-
template={{index .data "ca.crt"}}'| base64 -d)
```

```
echo "KAFKA_BROKER: '"$KAFKA_BROKER"'"
echo "KAFKA_USER: '"$KAFKA_USER"'"
echo "KAFKA_PWD: '"$KAFKA_PWD"'"
echo "KAFKA_TOPIC: '"$KAFKA_TOPIC"'"
echo "CERT_ELEMENT:  |- "
echo $CERT_ELEMENT
```

> You will have to indent the Certificate!

18.2.2 Deploying

```
oc apply -n default -f ./tools/97_addons/k8s-remote/create-cp4mcm-event-gateway.yaml

oc get route -n cp4waiops cp4waiops-event-gateway  -o jsonpath={.spec.host}
```

18.2.3 Using the Webhook

For the following example we will iterate over the `events` array and epush them to mapped version to Kafka:

```
curl -X "POST" "http://cp4waiops-event-gateway-cp4waiops.itzroks-270003bu3k-azsa8n-6ccd7f378ae819553d37d5f2ee142bd6-0000.us-south.containers.appdomain.cloud/webhook" \
     -H 'Content-Type: application/json' \
     -H 'Cookie: 36c13f7095ac25e696d30d7857fd2483=e345512191b5598e33b76be85dd7d3b6' \
     -d $'{
  "numberOfEvents": 3,
  "repository": "aiops",
  "timestamp": "2021-11-19T15:50:04.958Z",
  "alert": {
    "id": "rCA2w5zaIE6Xr3RKlFfhAxqqbGqGxGLC",
    "query": {
      "end": "now",
      "queryString": "\\"kubernetes.container_name\\" = ratings| @rawstring = /error/i",
      "start": "2s"
    },
    "name": "MyAlert",
    "description": "",
    "notifierID": "Rq4a9KUbomSIBvEcdC7kzzmdBtPI3yPb"
  },
  "events": [
    {
```

```
      "@rawstring": "Message 1",
      "@timestamp": 1639143464971,
      "kubernetes.container_name": "ratings",
      "kubernetes.namespace_name": "robot-shop",
    },
    {
      "@rawstring": "Message 2",
      "@timestamp": 1639143464982,
      "kubernetes.container_name": "catalogue",
      "kubernetes.namespace_name": "robot-shop",
    },
    {
      "@rawstring": "Message 3",
      "@timestamp": 1639143464992,
      "kubernetes.container_name": "web",
      "kubernetes.namespace_name": "robot-shop",
    }
  ],
  "warnings": ""
}'
```