

# Installing WebSphere Automation v1.2

## Table of Contents

Preparation .....	2
Create and adjust the infrastucture:.....	2
Adjust the OpenShift cluster:.....	4
Prepare WebSphere Automation 1.2 installation.....	5
Installation of WebSphere Automation .....	8
Setup WebSphere Automation with the OpenShift CLI.....	8
Access the WebSphere Automation GUI .....	13
Install and configure runtimes for WSA.....	15
WebSphere Automation tracking of security vulnerabilities.....	15
WebSphere Automation tracking for Out of Memory .....	24
Install Instana Backend.....	25
Register RHEL7WAS1 to Instana .....	28
Install components for additional demos .....	43
Install DayTrader onto WAS ND: .....	43
Create WAS ND cell for DayTrader:.....	43
Install DayTrader app and DB2 .....	46
Install Transformation Advisor into OCP.....	49
Install Mono2Micro UI into OCP .....	57
IBM Container Licensing Service .....	58
Overview about the endpoints .....	60
Overview about the Demo scenarios .....	62
Demo WebSphere Automation .....	63
Demo WSA Security with Liberty.....	63
Demo WSA Security with traditional WAS .....	67
Demo WSA Health with Liberty .....	69
Demo Transformation Advisor .....	75
Demonstrate the basic concepts of TA via the collection AppSrv1_collection.zip:.....	75
Demonstrate the Common Code Feature of TA via the collection ACME.zip .....	77
Demo Mono2Micro .....	79
Demo License Service .....	80
Demo DayTrader with Instana.....	82

## Preparation

### Create and adjust the infrastructure:

#### Create an OpenShift cluster in TEC based on template PH385250077921 (Instana/Turbonomic)

I used that template as it contained next to an OpenShift cluster already an additional VM with Instana.

10.99.96.102 10.99.96.103 10.99.96.104 10.99.96.107 10.99.96.108 10.99.96.109 10.99.96.110

10.99.96.111 10.99.96.112 10.99.96.113 10.99.96.114

Pick up the second address, here **10.99.96.103** and add it to /etc/hosts

10.99.96.103 bastion console-openshift-console.apps.ocp46.tec.uk.ibm.com oauth-  
openshift.apps.ocp46.tec.uk.ibm.com api-turbonomic.apps.ocp46.tec.uk.ibm.com web-robot-  
shop.apps.ocp46.tec.uk.ibm.com

10.99.96.114 instana.ocp46.tec.uk.ibm.com

Enhance the template with two additional RHEL VMs to run WebSphere Application Server.

Ask the TEC support team to enhance the environment with two additional nodes based on template PH128070503692 (Liberty Bootcamp VM based on RHEL 7)

RHEL VMs:

RHEL1 - 10.99.98.86 192.168.1.110

RHEL2 - 10.99.98.88 192.168.1.111

```
ssh root@bastion
```

Login root/passw0rd

```
[root@bastion ~]# mkdir -p /root/wsa/
```

```
[root@bastion ~]# cd /root/wsa/
```

Change hostname for the two RHEL machines

RHEL1:

```
ssh 192.168.1.110
```

Adjust /etc/hostname if needed

```
cat /etc/hostname
```

*RHEL7WAS1*

Reboot system

Reboot

RHEL2:

```
ssh 192.168.1.111
```

Adjust /etc/hostname if needed

```
cat /etc/hostname
```

*RHEL7WAS2*

Reboot system

Reboot

#### Add RHEL VMs to all hostfiles

Edit /etc/hosts on bastion node:

```
192.168.1.110 RHEL7WAS1.tec.uk.ibm RHEL7WAS1
```

```
192.168.1.111 RHEL7WAS2.tec.uk.ibm RHEL7WAS2
```

## **Cleanup OCP**

As we need the OCP cluster mainly for WebSphere Automation, we want to get rid of the installed Turbonomic.

- Uninstall Turbo operators
- Delete namespaces turbonomics and turbo

## **Stop agent on Instana host**

```
ssh root@instanabackend  
systemctl stop instana-agent.service
```

## Adjust the OpenShift cluster:

### Update OCP from OCP 4.6.16 to OCP 4.6.53+

Access the console at URL

<https://console-openshift-console.apps.ocp46.tec.uk.ibm.com/>

User: admin, password: passw0rd

### Check if the pull secrets contain already the user

<https://console-openshift-console.apps.ocp46.tec.uk.ibm.com/k8s/ns/openshift-config/secrets>

Search for “pull-secret”, then click on the dots behind “pull-secret”, select edit secret and review if for the different registries the user id and password is set.

If the secrets are not set, access OpenShift via CLI:

```
$ oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u oadmin -p passw0rd --insecure-skip-tls-verify=true
```

Add your Red Hat pull secrets:

- Retrieve you RH pull secrets: <https://console.redhat.com/openshift/install/vsphere/user-provisioned>
- Apply secrets as described here: [https://docs.openshift.com/container-platform/4.6/openshift\\_images/managing\\_images/using-image-pull-secrets.html](https://docs.openshift.com/container-platform/4.6/openshift_images/managing_images/using-image-pull-secrets.html)

Update cluster to add the security

[https://docs.openshift.com/container-platform/4.6/openshift\\_images/managing\\_images/using-image-pull-secrets.html](https://docs.openshift.com/container-platform/4.6/openshift_images/managing_images/using-image-pull-secrets.html)

### Verify that the persistent volume claim "image-registry-pvc" exists.

In my environment, only “pvc-image-registry” was defined. If missing, run the following command:  
`/ocp_install/scripts/create_registry_storage.sh`

Update OCP via console

<https://console-openshift-console.apps.ocp46.tec.uk.ibm.com/settings/cluster>

From 4.6.16 via 4.6.54

## Adjust OCP configuration

```
$ oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u admin -p passw0rd --insecure-skip-tls-verify=true
```

### Make nfs-client the default storage class

```
$ kubectl get storageclass
NAME          PROVISIONER          RECLAIMPOLICY    VOLUMEBINDINGMODE    ALLOWVOLUMEEXPANSION
AGE
nfs-client    icpd-nfs.io/nfs      Delete           Immediate             false
20h
```

```
$ kubectl patch storageclass nfs-client -p '{"metadata":{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/nfs-client patched
```

```
$ kubectl get storageclass | grep '(default)'
nfs-client (default)    icpd-nfs.io/nfs      Delete           Immediate             false
8h
```

## Prepare WebSphere Automation 1.2 installation

<https://www.ibm.com/docs/en/ws-automation?topic=installing-preparing-installation>

If not already done, update the OCP cluster to OCP 4.6.45+

OCP Console > Administration > Cluster Settings > Edit the Channel

### Step 1: Adding the IBM operator catalog

<https://www.ibm.com/docs/en/cloud-paks/1.0?topic=clusters-adding-operator-catalog>

Create the IBM® Operator catalog source.

```
$ sudo su
$ cat <<EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: docker.io/ibmcom/ibm-operator-catalog
  updateStrategy:
    registryPoll:
      interval: 45m
EOF
```

```
catalogsource.operators.coreos.com/ibm-operator-catalog created
```

```
$ oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

NAME	DISPLAY	TYPE	PUBLISHER	AGE
ibm-operator-catalog	IBM Operator Catalog	grpc	IBM	26s

### Step 2: Create the IBM Cloud Pak® foundational services catalog source.

<https://www.ibm.com/docs/en/cpfs?topic=online-installing-foundational-services-by-using-console>

BUT follow only until 2.1

**Create file opencloudio-source.yaml with content:**

```
$ cat <<EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencloud-operators
  namespace: openshift-marketplace
spec:
  displayName: IBMCS Operators
  publisher: IBM
  sourceType: grpc
  image: docker.io/ibmcom/ibm-common-service-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
EOF
```

```
catalogsource.operators.coreos.com/opencloud-operators created
```

Verify that the source container is running.

```
$ oc -n openshift-marketplace get pod | grep opencloud-operators
```

opencloud-operators-hh5w4	1/1	Running	0	26s
---------------------------	-----	---------	---	-----

### Step 3: Verify the CatalogSources installation

Verify the CatalogSources installation.

```
$ oc get catalogsource -n openshift-marketplace
```

NAME	DISPLAY	TYPE	PUBLISHER	AGE
certified-operators	Certified Operators	grpc	Red Hat	151m
community-operators	Community Operators	grpc	Red Hat	151m
ibm-operator-catalog	IBM Operator Catalog	grpc	IBM	4m37s
opencloud-operators	IBMCS Operators	grpc	IBM	83s
redhat-marketplace	Red Hat Marketplace	grpc	Red Hat	151m
redhat-operators	Red Hat Operators	grpc	Red Hat	151m

### Check the CatalogSources operators.

```
$ oc get pods -n openshift-marketplace
```

NAME	READY	STATUS	RESTARTS	AGE
certified-operators-bqwtx	1/1	Running	0	151m
community-operators-pt7jw	1/1	Running	0	151m
ibm-operator-catalog-6g4ql	1/1	Running	0	5m19s
marketplace-operator-67cb5b6d45-n2ht5	1/1	Running	0	154m
opencloud-operators-hh5w4	1/1	Running	0	2m5s
redhat-marketplace-wrz7d	1/1	Running	0	151m
redhat-operators-wwct8	1/1	Running	0	151m

Create a pull secret

## Use the OCP console to configure the global pull secret

(see <https://www.ibm.com/docs/en/ws-automation?topic=installing-preparing-installation> )

Access OCP console via URL

<https://console-openshift-console.apps.ocp46.tec.uk.ibm.com/dashboards>

and login as htadmin

Workloads > secrets, select namespace openshift-config, search for pull-secret

Project: openshift-config ▼


## Secrets

▼ Filter ▼

Name ▼ pull

Name pull ✕

Clear all filters

Name ↑	Type ↓
 pull-secret	kubernetes.io/dockerconfigjson

Select Edit Secret and click Add Credentials to add a new entry for the entitled registry.

Registry Server Address: cp.icr.io

Username: cp

Password: entitled key (from <https://myibm.ibm.com/products-services/containerlibrary> )

**Registry Server Address \***

**Username \***

**Password \***

**Email**

+ Add Credentials

Save

Cancel

# Installation of WebSphere Automation

## Setup WebSphere Automation with the OpenShift CLI

[https://www.ibm.com/docs/en/ws-automation?topic=installing-openshift-cli#in-t-cli\\_install-op-cli](https://www.ibm.com/docs/en/ws-automation?topic=installing-openshift-cli#in-t-cli_install-op-cli)

### Create a workspace for WebSphere-automation components

We will use the OwnNamespace mode, so set WSA\_OPERATOR\_NAMESPACE to the same value as WSA\_INSTANCE\_NAMESPACE.

```
WSA_OPERATOR_NAMESPACE=websphere-automation
WSA_INSTANCE_NAMESPACE=websphere-automation
oc create namespace $WSA_INSTANCE_NAMESPACE
```

### Create an operator group

```
cat <<EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha2
kind: OperatorGroup
metadata:
  name: ibm-websphere-automation
  namespace: $WSA_OPERATOR_NAMESPACE
spec:
  targetNamespaces:
    - $WSA_INSTANCE_NAMESPACE
EOF
operatorgroup.operators.coreos.com/ibm-websphere-automation created
```

### Create a WebSphere Automation Subscription in namespace ws-automation.

```
cat <<EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-websphere-automation
  namespace: $WSA_OPERATOR_NAMESPACE
spec:
  channel: v1.2
  installPlanApproval: Automatic
  name: ibm-websphere-automation
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
subscription.operators.coreos.com/ibm-websphere-automation created
```

### Create an instance of WebSphere Automation health and secure

```
cat <<EOF | oc apply -f -
apiVersion: automation.websphere.ibm.com/v1
kind: WebSphereHealth
metadata:
  name: wsa-health
  namespace: $WSA_INSTANCE_NAMESPACE
spec:
  license:
    accept: true
EOF
webspherehealth.automation.websphere.ibm.com/wsa-health created
```

```
oc project $WSA_INSTANCE_NAMESPACE
Now using project "websphere-automation" on server "https://10.16.99.81:6443".
```

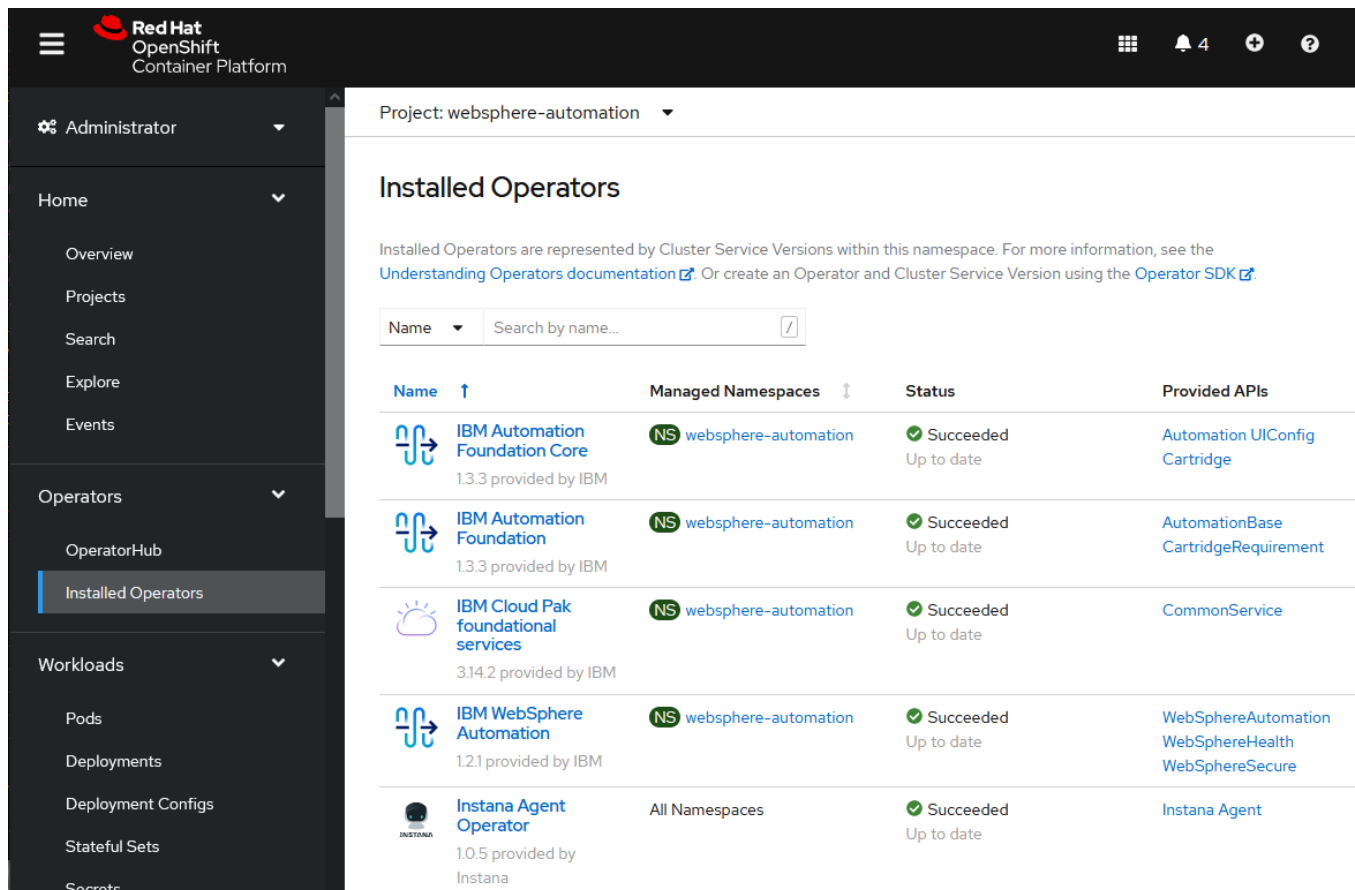
Check status of the operators (it can take up to 20 minutes and several restarts before all are running):



```
oc get pods | grep operator
```

```
[root@bastion ~]# oc get pods | grep operator
iaf-core-operator-controller-manager-79856987d7-dbcrb      1/1      Running    0          38m
iaf-eventprocessing-operator-controller-manager-694989445cnqvhc 1/1      Running    0          35m
iaf-flink-operator-controller-manager-8dfb5fc95-g8jzg      1/1      Running    0          38m
iaf-operator-controller-manager-745b67787-p4wx5           1/1      Running    0          33m
iaf-system-entity-operator-cf864fcb7-jhzmd                3/3      Running    0          6m21s
ibm-common-service-operator-695fd789cc-sl2dc              1/1      Running    0          35m
ibm-elastic-operator-controller-manager-b7f9c8b6c-tnmrr    1/1      Running    0          38m
websphere-automation-operator-controller-manager-68fb44f5drxs9j 1/1      Running    0          35m
```

Finally, the Operators are installed

















Project: websphere-automation

### Installed Operators

Installed Operators are represented by Cluster Service Versions within this namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and Cluster Service Version using the [Operator SDK](#).

Name Search by name...

Name	Managed Namespaces	Status	Provided APIs
 <b>IBM Automation Foundation Core</b> 1.3.3 provided by IBM	 websphere-automation	 Succeeded Up to date	<a href="#">Automation UIConfig</a> <a href="#">Cartridge</a>
 <b>IBM Automation Foundation</b> 1.3.3 provided by IBM	 websphere-automation	 Succeeded Up to date	<a href="#">AutomationBase</a> <a href="#">CartridgeRequirement</a>
 <b>IBM Cloud Pak foundational services</b> 3.14.2 provided by IBM	 websphere-automation	 Succeeded Up to date	<a href="#">CommonService</a>
 <b>IBM WebSphere Automation</b> 1.2.1 provided by IBM	 websphere-automation	 Succeeded Up to date	<a href="#">WebSphereAutomation</a> <a href="#">WebSphereHealth</a> <a href="#">WebSphereSecure</a>
 <b>Instana Agent Operator</b> 1.0.5 provided by Instana	All Namespaces	 Succeeded Up to date	<a href="#">Instana Agent</a>

Wait until WebSphere Health is ready (takes 80+ minutes):

```
oc get WebSphereHealth
```

```
NAME      READY   AGE
wsa-health True     3h
```

## Validate installation

<https://www.ibm.com/docs/en/ws-automation?topic=installing-validating-installation>

```
oc login
oc project websphere-automation
```

Verify whether the operator subscription exists.

```
oc get subscription | grep websphere-automation
```

```
[root@wsa-larsb-inf ibmdemo]# oc get subscription | grep websphere-automation
ibm-websphere-automation          ibm-websphere-automation          ibm-operator-catalog    v1.2
```

Verify the installation status of the operator.

```
oc get csv | grep websphere-automation
```

```
[root@wsa-larsb-inf ibmdemo]# oc get csv | grep websphere-automation
ibm-websphere-automation.v1.2.1    IBM WebSphere Automation          1.2.1    ibm-websphere-automation.v1.2.0    Succeeded
[root@wsa-larsb-inf ibmdemo]#
```

## Check the status of the WebSphereAutomation custom resource

Verify that all status conditions are True.

```
oc describe websphereautomation
```

```
Status:
Conditions:
  Message: All prerequisites and installed components are ready
  Status: True
  Type: Ready
  Status: True
  Type: CartridgeReady
  Status: True
  Type: AutomationBaseReady
  Status: True
  Type: CartridgeRequirementsReady
  Message: Kafka cluster is ready
  Status: True
  Type: KafkaReady
  Message: Kafka resources are ready
  Status: True
  Type: KafkaResourcesReady
  Message: Data store is ready
  Status: True
  Type: DataStoreReady
  Message: All prerequisites and WebSphere Secure components are ready
  Status: True
  Type: WebSphereSecureReady
  Message: All prerequisites and WebSphere Health components are ready
  Status: True
  Type: WebSphereHealthReady
  Status: True
  Type: RunbookManagerReady
  Message: All updates to WebSphereAutomation instance have been processed
  Status: True
  Type: Reconciled
Endpoints:
  Automation UI: https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com
References:
  Truststore Secret Version: 122924868
Versions:
  Reconciled: 1.2.1
Events: <none>
```

## Check the status of the WebSphereSecure custom resource.

```
oc describe webspheresecure
```

Status:

Conditions:

Message: All prerequisites and WebSphere Secure components are ready  
Status: True  
Type: Ready  
Message: Kafka cluster is ready  
Status: True  
Type: KafkaReady  
Message: Data store is ready  
Status: True  
Type: DataStoreReady  
Message: Kafka resources are ready  
Status: True  
Type: KafkaResourcesReady  
Message: Dependencies are ready  
Status: True  
Type: DependenciesReady  
Status: True  
Type: CVEMonitorReady  
Status: True  
Type: ServerRegistrationProcessorReady  
Status: True  
Type: VulnerabilityNotifierReady  
Status: True  
Type: VulnerabilityManagerReady  
Status: True  
Type: WebSphereSecureAPIsReady  
Status: True  
Type: MeteringAPIsReady  
Status: True  
Type: WebSphereSecureUIReady  
Message: All updates to WebSphereSecure instance have been processed  
Status: True  
Type: Reconciled

**Check the status of the WebSphereHealth custom resource.**

oc describe webspherehealth

Status:

Conditions:

Message: All prerequisites and WebSphere Health components are ready  
Status: True  
Type: Ready  
Message: File store is ready  
Status: True  
Type: FileStoreReady  
Message: Kafka cluster is ready  
Status: True  
Type: KafkaReady  
Message: Dependencies are ready  
Status: True  
Type: DependenciesReady  
Message: Data store is ready  
Status: True  
Type: DataStoreReady  
Message: Kafka resources are ready  
Status: True  
Type: KafkaResourcesReady  
Status: True  
Type: InvestigationManagerReady  
Status: True  
Type: AnalysisManagerReady  
Status: True  
Type: WebSphereHealthAPIsReady  
Status: True  
Type: WebhooksAPIsReady  
Message: All updates to WebSphereHealth instance have been processed  
Status: True  
Type: Reconciled

Endpoints:

Automation UI: <https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com>

References:

Versions:

Reconciled: 1.2.1

Events: <none>

## Access the WebSphere Automation GUI

### The URL to the WebSphere Automation UI is displayed in the WebSphereAutomation CR

You can retrieve the URL via the command:

```
oc describe websphereautomation | grep "Automation UI"
[root@bastion ~]# oc describe websphereautomation | grep "Automation UI"
Automation UI: https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com
https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com
```

Add to /etc/hosts the additional hostnames 10.99.96.103 bastion console-openshift-console.apps.ocp46.tec.uk.ibm.com oauth-openshift.apps.ocp46.tec.uk.ibm.com web-robot-shop.apps.ocp46.tec.uk.ibm.com cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com cp-console.apps.ocp46.tec.uk.ibm.com

You can retrieve the username for login via the command:

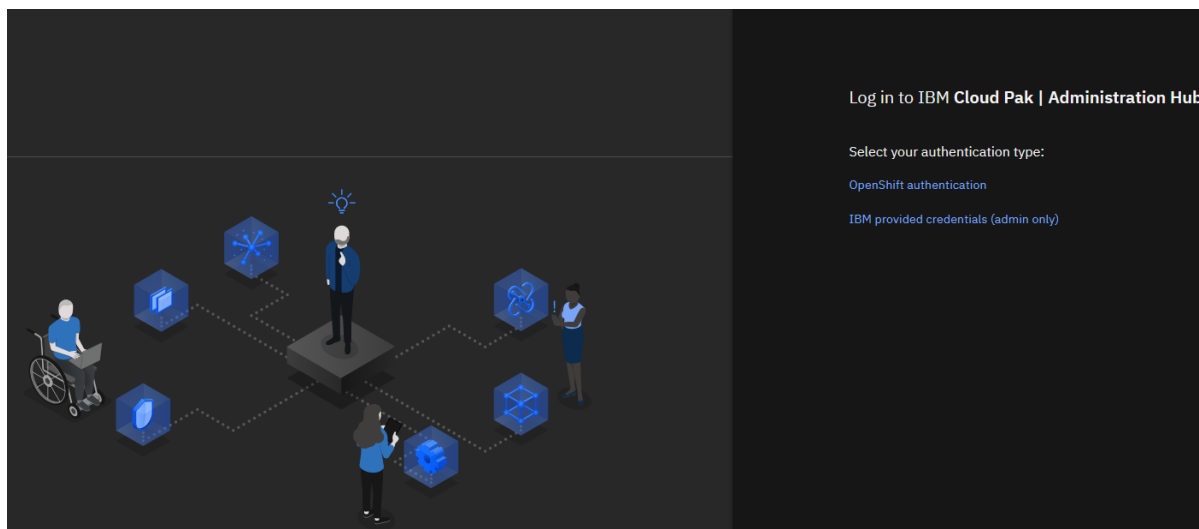
```
oc -n ibm-common-services get secret platform-auth-idp-credentials -o
jsonpath='{.data.admin_username}' | base64 --decode
admin
```

You can retrieve the password for login via the command:

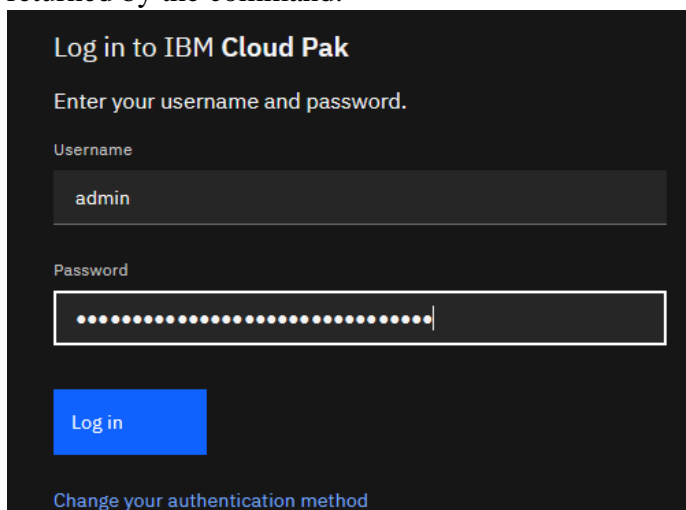
```
oc -n ibm-common-services get secret platform-auth-idp-credentials -o
jsonpath='{.data.admin_password}' | base64 --decode
JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW
```

### Open a via browser at URL

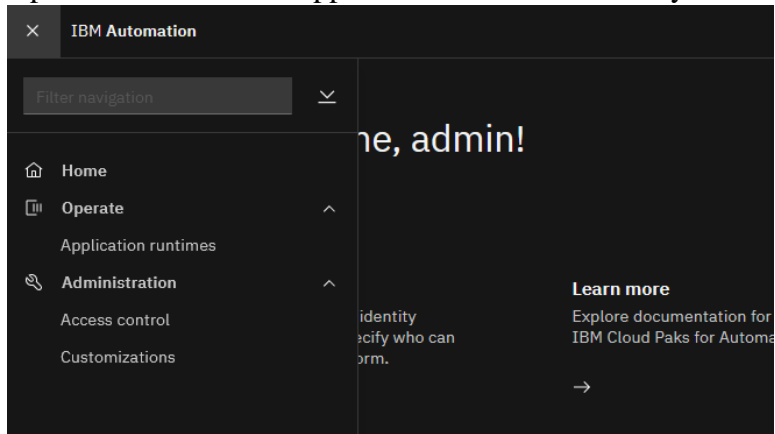
<https://cpd-websphere-automation.apps.wsa-larsb.os.fyre.ibm.com>



Select IBM provided credentials (admin only) and use as credentials the user admin and the password as returned by the command.

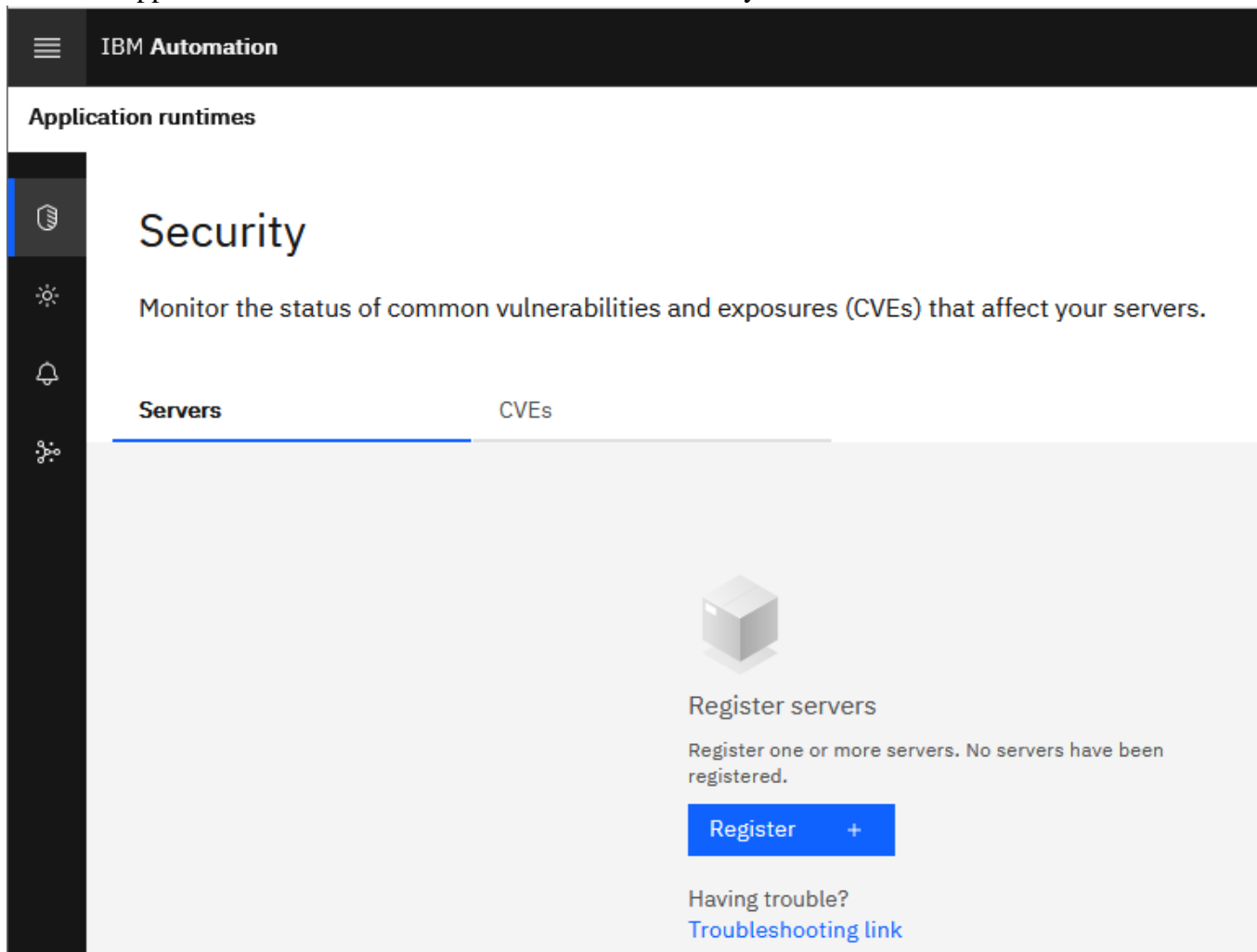


Open the menu on the upper left and select the task you want to perform.



## Application runtimes

Click on Application Runtimes to see that there is no runtime yet defined



## Install and configure runtimes for WSA

### WebSphere Automation tracking of security vulnerabilities

#### **RHEL7WAS1**

#### **Switch to RHEL7WAS1 (user ibmdemo, password passw0rd)**

```
ssh ibmdemo@rhel7was1
```

#### **Create as user ibmdemo target directories**

```
sudo mkdir -p /var/IBM/software
sudo chown -R ibmdemo:ibmdemo /var/IBM/software/
sudo mkdir -p /usr/IBM
sudo chown -R ibmdemo:ibmdemo /usr/IBM
mkdir -p /usr/IBM/Liberty
mkdir -p /usr/IBM/scripts
```

#### **Install java**

Java will be required to use the keytool (to generate the Liberty keystore)

```
cd /usr/IBM/
tar -zxvf /var/IBM/software/Java/ibm-semeru-open-jdk_x64_linux_8u312b07_openj9-0.29.0.tar.gz
export PATH=$PATH:/usr/IBM/jdk8u312-b07/bin/
```

#### **Adjust the hostfile**

```
vi /etc/hosts
192.168.1.100 cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com cp-
console.apps.ocp46.tec.uk.ibm.com api.apps.ocp46.tec.uk.ibm.com oauth-
openshift.apps.ocp46.tec.uk.ibm.com
192.168.1.110 RHEL7WAS1.tec.uk.ibm.com RHEL7WAS1
192.168.1.111 RHEL7WAS2.tec.uk.ibm.com RHEL7WAS2
```

## Retrieving WSA details:

<https://www.ibm.com/docs/en/ws-automation?topic=automation-setting-up-security-monitoring>

To configure the usage metering feature in each of your application servers, you must obtain the following usage metering items:

- URL - The URL of the usage metering service in WebSphere Automation.
- API key - token to authenticate the WAS servers and Liberty servers during registration
- Usage metering certificate - The certificate that contains the public key.

## Create the directory to store the WSA assets

```
mkdir -p /usr/IBM/WSA
cd /usr/IBM/WSA
```

## Log into OpenShift and switch to the project ws-automation-tec

```
oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u admin -p passw0rd --insecure-skip-tls-verify=true
oc project websphere-automation
```

## Retrieve the URL of the usage metering service in WebSphere Automation

Get the URL of the usage metering service in WebSphere Automation and save it to a file

/usr/IBM/WSA/WSA\_metering\_URL.txt.

```
oc get route cpd -o jsonpath=https://{.spec.host}/websphereauto/meteringapi >
WSA_metering_URL.txt && cat WSA_metering_URL.txt
```

## Retrieve the API Key from WebSphere Automation

Get the api-key that will be used to authenticate the WebSphere Application Server and Liberty servers during the registration process. Save it to a file named /usr/IBM/WSA\WSA\_metering\_api-key.txt.

```
oc get secret wsa-secure-metering-apis-encrypted-tokens -o jsonpath='{.data.wsa-secure-metering-apis-sa}' | base64 -d > WSA_metering_api-key.txt && cat WSA_metering_api-key.txt
```

## Retrieve the server certificate from WebSphere Automation

Get the Server certificate that is used for SSL handshake between the servers and IBM Automation, and save it to a file named /usr/IBM/WSA\WSA\_metering\_certificate\_file.pem.

```
oc get secret external-tls-secret -o jsonpath='{.data.cert\.crt}' | base64 -d >
WSA_metering_certificate_file.pem && cat WSA_metering_certificate_file.pem
```

## Log out of OpenShift

You have retrieved all necessary information, so log out

```
oc logout
```

## Create a Keystore for metering

```
keytool -import -trustcacerts -file WSA_metering_certificate_file.pem -keystore
WSA_metering_Key.p12 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -noprompt
```

## List all generated assets

```
ls -lrt WSA_metering*
```

```
[ibmdemo@wsa-larsb-inf Liberty]$ ls -lrt WSA_metering*
-rw-rw-r-- 1 ibmdemo ibmdemo 89 Jan 6 03:13 WSA_metering_URL.txt
-rw-rw-r-- 1 ibmdemo ibmdemo 1368 Jan 6 03:15 WSA_metering_api-key.txt
-rw-rw-r-- 1 ibmdemo ibmdemo 1310 Jan 6 03:17 WSA_metering_certificate_file.pem
-rw-rw-r-- 1 ibmdemo ibmdemo 1218 Jan 6 03:18 WSA_metering_Key.p12
```



## Create a Liberty instance

### Install Liberty:

```
cd /usr/IBM/Liberty
java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.12.jar -acceptLicense
```

### Register the Liberty instance to WSA

Create a new Liberty server, using the command below:

```
/usr/IBM/Liberty/wlp/bin/server create libertyServer1
```

### Create a re-usable server.xml for the WSA settings

```
cat /var/IBM/software/WAS/WSA_server.xml
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
    <!-- Enable features -->
    <featureManager>
        <feature>usageMetering-1.0</feature>
        <feature>transportSecurity-1.0</feature>
    </featureManager>

    <keyStore id="WSA_metering_keyStore"
        password="meterPwd"
        location="{WSA_metering_keystore}"
        type="PKCS12" />

    <ssl id="WSA_metering_SSL" keyStoreRef="defaultKeyStore"
        trustStoreRef="WSA_metering_keyStore" sslProtocol="TLSv1.2" />
    <usageMetering
        url="{WSA_metering_URL}"
        sslRef="WSA_metering_SSL"
        apiKey="{WSA_metering_api-key}" />
</server>
```

### Create a config dropins directory for the new Liberty instance and copy the WSA server configuration:

```
mkdir -p /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/configDropins/defaults
cp /var/IBM/software/WAS/WSA_server.xml
/usr/IBM/Liberty/wlp/usr/servers/libertyServer1/configDropins/defaults/
```

### Define the variables WSA\_Metering\_Keystore, WSA\_Metering\_URL and WSA\_Metering\_api-key.

```
mkdir -p /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/variables
cp /usr/IBM/WSA/WSA_metering_URL.txt
/usr/IBM/Liberty/wlp/usr/servers/libertyServer1/variables/WSA_metering_URL
echo "/usr/IBM/WSA/WSA_metering_Key.pl2" >
/usr/IBM/Liberty/wlp/usr/servers/libertyServer1/variables/WSA_metering_keystore
cp /usr/IBM/WSA/WSA_metering_api-key.txt
/usr/IBM/Liberty/wlp/usr/servers/libertyServer1/variables/WSA_metering_api-key
ls /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/variables
```

```
[ibmdemo@RHEL7WAS1 libertyServer1]$ ls /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/variables
WSA_metering_api-key  WSA_metering_keystore  WSA_metering_URL
```

### Start the Liberty instance and it should get registered

```
/usr/IBM/Liberty/wlp/bin/server start libertyServer1
cat /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/logs/messages.log
```

```
[ibmdemo@wsa-larsb-inf Liberty]$ /usr/IBM/Liberty/wlp/bin/server start libertyServer1

Starting server libertyServer1.
Server libertyServer1 started with process ID 18731.
[ibmdemo@wsa-larsb-inf Liberty]$ cat /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/logs/messages.log
*****
product = WebSphere Application Server 21.0.0.6 (wlp-1.0.53.c1210620210527-1900)
wlp.install.dir = /usr/IBM/Liberty/wlp/
java.home = /usr/IBM/jdk8u312-b07/jre
java.version = 1.8_0_312
java.runtime = IBM Semeru Runtime Open Edition (1.8_0_312-b07)
os = Linux (3.10.0-1160.45.1.el7.x86_64; amd64) (en_US)
process = 18731@9.30.122.155
*****
[1/6/22 4:41:58:205 PST] 00000001 com.ibm.ws.kernel.launch.internal.FrameworkManager A CWMKE0001I: The server libertyServer1 has been launched.
[1/6/22 4:41:58:563 PST] 0000001d com.ibm.ws.config.xml.internal.ServerXMLConfiguration A CWMKG0093A: Processing configuration drop-ins resource: /usr/IBM/Liberty/wlp/us
r/servers/libertyServer1/configDropins/defaults/WSA_server.xml
[1/6/22 4:41:58:718 PST] 00000001 com.ibm.ws.kernel.launch.internal.FrameworkManager I CWMKE0002I: The kernel started after 0.645 seconds
[1/6/22 4:41:58:750 PST] 00000027 com.ibm.ws.kernel.feature.internal.FeatureManager I CWMKF0007I: Feature update started.
[1/6/22 4:41:58:944 PST] 0000001d com.ibm.ws.app.manager.internal.monitor.DropinMonitor A CWMKZ0058I: Monitoring dropins for applications.
[1/6/22 4:41:59:364 PST] 00000027 com.ibm.ws.tcpchannel.internal.TCPPort I CWMK00219I: TCP Channel defaultHttpEndpoint has been started and is now listeni
ng for requests on host localhost (IPv4: 127.0.0.1) port 9080.
[1/6/22 4:41:59:366 PST] 00000027 com.ibm.ws.kernel.feature.internal.FeatureManager A CWMKF0012I: The server installed the following features: [el-3.0, jsp-2.3, serv
let-3.1, ssl-1.0, transportSecurity-1.0, usageMetering-1.0].
[1/6/22 4:41:59:367 PST] 00000027 com.ibm.ws.kernel.feature.internal.FeatureManager I CWMKF0008I: Feature update completed in 0.647 seconds.
[1/6/22 4:41:59:367 PST] 00000027 com.ibm.ws.kernel.feature.internal.FeatureManager A CWMKF0011I: The libertyServer1 server is ready to run a smarter planet. The lib
ertyServer1 server started in 1.295 seconds.
[1/6/22 4:41:59:418 PST] 00000022 com.ibm.ws.ssl.config.WSKeyStore A CWPKI0820A: The default keystore has been created using the 'keystore_password'
environment variable.
[1/6/22 4:41:59:526 PST] 00000022 com.ibm.ws.ssl.config.WSKeyStore I Successfully loaded default keystore: /usr/IBM/Liberty/wlp/usr/servers/libertyS
erver1/resources/security/key.p12 of type: PKCS12
[1/6/22 4:41:59:553 PST] 0000002e com.ibm.ws.tcpchannel.internal.TCPPort I CWMK00219I: TCP Channel defaultHttpEndpoint-ssl has been started and is now lis
tening for requests on host localhost (IPv4: 127.0.0.1) port 9443.
[1/6/22 4:41:59:599 PST] 0000002c com.ibm.ws.session.WASessionCore I SESN8501I: The session manager did not find a persistent storage location; Http
Session objects will be stored in the local application server's memory.
[1/6/22 4:42:00:303 PST] 00000027 com.ibm.ws.usage.metering.common.RegisterTask I CWMKR0400I: The server was registered with the IBM Cloud Private Metering servi
ce on the specified URL https://cpd-websphere-automation.apps.wsa-larsb.os.fyre.ibm.com/websphereauto/meteringapi.
```

Verify in the WSA UI that the server has been registered

IBM Automation

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers

CVEs

Filter by

Cell

CVE

WebSphere version

Export CSV

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
None (0.0)	libertyServer1	—	RHEL7WAS1	21.0.0.12	1.8.0_312	—	—

Stop the Liberty instance to save resources

```
/usr/IBM/Liberty/wlp/bin/server stop libertyServer1
```

# Create a WAS Traditional instance

## Install the WAS Traditional software

Prepare environment:

```
export INSTALL_SOURCE_PATH=/var/IBM/software/WAS
```

## Install IBM Installation Manager

### Create temporary directory and extract Installation Manager image

```
mkdir -p /var/IBM/temp/IM
unzip $INSTALL_SOURCE_PATH/agent.installer.linux.gtk.x86_64_1.9.1004.20201109_1718.zip
-d /var/IBM/temp/IM
```

### Install Installation Manager

```
export IM_VAR_HOME=/var/IBM/IM
export IM_USR_HOME=/usr/IBM/IM
/var/IBM/temp/IM/userinstc -installationDirectory $IM_USR_HOME -dataLocation
$IM_VAR_HOME/data -log $IM_VAR_HOME/log/IM_Install.log -acceptLicense
```

```
[ibmdemo@wsa-larsb-inf IM]$ /var/IBM/temp/IM/userinstc -installationDirectory $IM_USR_HOME -dataLocation $IM_VAR_HOME/data
-log $IM_VAR_HOME/log/IM_Install.log -acceptLicense
Installed com.ibm.cic.agent_1.9.1004.20201109_1718 to the /usr/IBM/IM/eclipse directory.
```

### Cleanup temporary directories:

```
rm -rf /var/IBM/temp/IM
```

To uninstall IM if necessary:

```
List installed packages via /usr/IBM/IM/eclipse/tools/imcl listInstalledPackages
com.ibm.cic.agent_1.9.1004.20201109_1718
com.ibm.websphere.ND.v85_8.5.5018.20200910_1821
Then remove them via /usr/IBM/IM/eclipse/tools/imcl uninstall
com.ibm.websphere.ND.v85_8.5.5018.20200910_1821
```

## Install WAS ND 8.5.5

### Create temporary directory and extract WAS ND 8.5 image

```
mkdir -p /var/IBM/temp/WAS/WAS85ND
unzip $INSTALL_SOURCE_PATH/WASND_v8.5.5_1*.zip -d /var/IBM/temp/WAS/WAS85ND
unzip $INSTALL_SOURCE_PATH/WASND_v8.5.5_2*.zip -d /var/IBM/temp/WAS/WAS85ND
unzip $INSTALL_SOURCE_PATH/WASND_v8.5.5_3*.zip -d /var/IBM/temp/WAS/WAS85ND
mkdir /var/IBM/temp/WAS/WAS85FP
unzip $INSTALL_SOURCE_PATH/8.5.5-WS-WAS-FP018-part1* -d /var/IBM/temp/WAS/WAS85FP
unzip $INSTALL_SOURCE_PATH/8.5.5-WS-WAS-FP018-part2* -d /var/IBM/temp/WAS/WAS85FP
unzip $INSTALL_SOURCE_PATH/8.5.5-WS-WAS-FP018-part3* -d /var/IBM/temp/WAS/WAS85FP
```

### Install WAS ND 8.5 in silent mode via Installation Manager

```
/usr/IBM/IM/eclipse/tools/imcl input
$INSTALL_SOURCE_PATH/WAS855_18_ND_install_silent_ibmdemo.xml -acceptLicense
```

```
[ibmdemo@wsa-larsb-inf temp]$ /usr/IBM/IM/eclipse/tools/imcl input $INSTALL_SOURCE_PATH/WAS855_18_ND_install_silen
t_ibmdemo.xml -acceptLicense
Installed com.ibm.websphere.ND.v85_8.5.5018.20200910_1821 to the /usr/IBM/WAS855ND directory.
```

```
[ibmdemo@wsa-larsb-inf IM]$ cat
$INSTALL_SOURCE_PATH/WAS855_18_ND_install_silent_ibmdemo.xml
<?xml version='1.0' encoding='UTF-8'?>
<agent-input>
  <variables>
    <variable name='sharedLocation' value='/var/IBM/IMShared'/>
  </variables>
  <server>
    <repository location='/var/IBM/temp/WAS/WAS85ND'/>
  </server>
</agent-input>
```

```

    <repository location='/var/IBM/temp/WAS/WAS85FP' />
  </server>
  <profile id='IBM WebSphere Application Server V8.5'
installLocation='/usr/IBM/WAS855ND'>
    <data key='cic.selector.arch' value='x86' />
    <data key='user.wasjava' value='java8' />
    <data key='user.internal.use.only.prev.wasjava' value='java8' />
  </profile>
  <install>
    <!-- IBM WebSphere Application Server Network Deployment 8.5.5.18 -->
    <offering profile='IBM WebSphere Application Server V8.5'
id='com.ibm.websphere.ND.v85' version='8.5.5018.20200910_1821'
features='core.feature,ejbdeploy,thinclient,embeddablecontainer,com.ibm.sdk.6_64bit' />
  </install>
  <preference name='com.ibm.cic.common.core.preferences.eclipseCache'
value='${sharedLocation}' />
  <preference name='offering.service.repositories.areUsed' value='false' />
</agent-input>

```

## Verify installation:

```

/usr/IBM/WAS855ND/bin/versionInfo.sh
[ibmdemo@wsa-larsb-inf temp]$ /usr/IBM/WAS855ND/bin/versionInfo.sh
WVER0010I: Copyright (c) IBM Corporation 2002, 2012; All rights reserved.
WVER0012I: VersionInfo reporter version 1.15.1.50, dated 12/20/18

```

IBM WebSphere Product Installation Status Report	
Report at date and time January 6, 2022 5:29:40 AM PST	
Installation	
Product Directory	/usr/IBM/WAS855ND
Version Directory	/usr/IBM/WAS855ND/properties/version
DTD Directory	/usr/IBM/WAS855ND/properties/version/dtd
Log Directory	/var/IBM/IM/data/logs
Product List	
ND	installed
Installed Product	
Name	IBM WebSphere Application Server Network Deployment
Version	8.5.5.18
ID	ND
Build Level	cf182036.01
Build Date	9/10/20
Package	com.ibm.websphere.ND.v85_8.5.5018.20200910_1821
Java SE Version	8
Architecture	x86-64 (64 bit)
Installed Features	IBM 64-bit WebSphere SDK for Java WebSphere Application Server Full Profile EJBDeploy tool for pre-EJB 3.0 modules Embeddable EJB container Stand-alone thin clients and resource adapters
End Installation Status Report	

## Cleanup temporary directories:

```
rm -rf /var/IBM/temp/WAS
```

## Create WAS Profile:

```
export WAS_HOME=/usr/IBM/WAS855ND
$WAS_HOME/bin/manageprofiles.sh -create \
  -profileName WSA \
  -templatePath $WAS_HOME/profileTemplates/default \
  -serverName twasServer1 \
  -nodeName AppSrv01 \
  -cellName twasCell01 \
  -enableAdminSecurity true \
  -adminUserName wasadmin \
  -adminPassword passw0rd
```

```
[ibmdemo@RHEL7WAS1 ~]$ $WAS_HOME/bin/manageprofiles.sh -create \
> -profileName WSA \
> -templatePath $WAS_HOME/profileTemplates/default \
> -serverName twasServer1\
> -nodeName AppSrv01\
> -cellName twasCell01\
> -enableAdminSecurity true\
> -adminUserName wasadmin\
> -adminPassword passw0rd
INSTCONFSUCCESS: Success: Profile WSA now exists. Please consult /usr/IBM/WAS855ND/profiles/WSA/logs/AboutThisProfile.txt
for more information about this profile.
```

## Review WAS profile settings

```
cat /usr/IBM/WAS855ND/profiles/WSA/logs/AboutThisProfile.txt
[ibmdemo@RHEL7WAS1 ~]$ cat /usr/IBM/WAS855ND/profiles/WSA/logs/AboutThisProfile.txt
Application server environment to create: Application server
Location: /usr/IBM/WAS855ND/profiles/WSA
Disk space required: 200 MB
Profile name: WSA
Make this profile the default: True
Node name: AppSrv01
Host name: RHEL7WAS1.tec.uk.ibm.com
Enable administrative security (recommended): True
Administrative console port: 9060
Administrative console secure port: 9043
HTTP transport port: 9080
HTTPS transport port: 9443
Bootstrap port: 2809
SOAP connector port: 8880
Run application server as a service: False
Create a Web server definition: False
Performance tuning setting: Standard
```

## Adjust soap.client.props to add credentials

```
vi /usr/IBM/WAS855ND/profiles/WSA/properties/soap.client.props
com.ibm.SOAP.securityEnabled=true
com.ibm.SOAP.loginUserId=wasadmin
com.ibm.SOAP.loginPassword=passw0rd
```

## Start the server

```
/usr/IBM/WAS855ND/profiles/WSA/bin/startServer.sh twasServer1
```

## Adjust ports to avoid conflict with Liberty

```
export INSTALL_SOURCE_PATH=/var/IBM/software/WAS
cp $INSTALL_SOURCE_PATH/adjustPorts.py /usr/IBM/WAS855ND/profiles/WSA/bin
cat /usr/IBM/WAS855ND/profiles/WSA/bin/adjustPorts.py
# Move ports from 9080 to 19080 and 9443 to 19443
# Application servers > server1 > Ports > WC_defaulthost
AdminTask.modifyServerPort('twasServer1', '[-nodeName AppSrv01 -endPointName
WC_defaulthost -host * -port 19080 -modifyShared true]')

# Application servers > server1 > Ports > WC_defaulthost_secure
AdminTask.modifyServerPort('twasServer1', '[-nodeName AppSrv01 -endPointName
WC_defaulthost_secure -host * -port 19443 -modifyShared true]')

# Virtual Hosts > default_host > Host Aliases > New...
AdminConfig.create('HostAlias',
AdminConfig.getid('/Cell:twasCell01/VirtualHost:default_host/'), '[[hostname "*"] [port
"19080"]]]')
AdminConfig.create('HostAlias',
AdminConfig.getid('/Cell:twasCell01/VirtualHost:default_host/'), '[[hostname "*"] [port
"19443"]]]')

# Save
AdminConfig.save()
```

## Adjust the file to fit to your nodename, then apply it.

```
cd /usr/IBM/WAS855ND/profiles/WSA/bin
./wsadmin.sh -lang jython -conntype SOAP -port 8880 -f adjustPorts.py
```

## Restart the server

```
/usr/IBM/WAS855ND/profiles/WSA/bin/stopServer.sh twasServer1
/usr/IBM/WAS855ND/profiles/WSA/bin/startServer.sh twasServer1
```

## Apply fix to remove Risk Level 10 issue (Log4J)

```
/var/IBM/software/WAS/was_CVE.sh
[ibmdemo@RHEL7WAS1 bin]$ cat /var/IBM/software/WAS/was_CVE.sh
export fixID="8.5.5.11-WS-WASProd-IFPH42762"
export fixRepo="8.5.5.11-ws-wasprod-ifph42762.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSA"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
echo "Stop Server"
$WAS855ND_PROFILE/bin/stopServer.sh twasServer1
echo "Apply Fix $fixID"
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
export fixID="8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271"
export fixRepo="8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271.zip"
echo "Apply Fix $fixID"
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
echo "Start Server"
$WAS855ND_PROFILE/bin/startServer.sh twasServer1
```

Register the tWAS instance to WSA

https://www.ibm.com/docs/en/ws-automation?topic=vulnerabilities-adding-websphere-application-server-server

The approach to configure WAS Traditional is a bit different than the one for Liberty:

- To retrieve the metering URL and the API key, we will re-use the content of the two WAS files.
- The WSA certificate will be retrieved from the WSA instance directly.
- To configure WAS, IBM provides a ready to use wsadmin script, you can find details here:

https://www.ibm.com/docs/en/ws-automation?topic=monitoring-registering-websphere-application-server-traditional-servers

The content of the script has been copied into the file configuretWasUsageMetering.py.

Copy the file into the WAS bin directory of the server.

```
export INSTALL_SOURCE_PATH=/var/IBM/software/WAS
cp $INSTALL_SOURCE_PATH/configuretWasUsageMetering.py
/usr/IBM/WAS855ND/profiles/WSA/bin
```

Register the server by running the script

```
cd /usr/IBM/WAS855ND/profiles/WSA/bin
./wsadmin.sh -lang jython -conntype SOAP -port 8880 -f configuretWasUsageMetering.py
url=$(cat /usr/IBM/WSA/WSA_metering_URL.txt) apiKey=$(cat
/usr/IBM/WSA/WSA_metering_api-key.txt) trustStorePassword=meterPwd
```

```
[ibmdemo@RHEL7WAS1 bin]$ ./wsadmin.sh -lang jython -conntype SOAP -port 8880 -f configuretWasUsageMetering.py url=$(cat /usr/IBM/WSA/WSA_metering_URL.txt)
apiKey=$(cat /usr/IBM/WSA/WSA_metering_api-key.txt) trustStorePassword=meterPwd
WASX7209I: Connected to process "twasServer1" on node AppSrv01 using SOAP connector; The type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable: "[url=https://
/cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi, apiKey=cu9xY11564jKmhHkfaMHbxD0CzsRPOA8xntR7fL1n1Xi4Gcx2K6KD1kH8J3FGODGAYKKR
ZBCg0f+7QE2btWNJhekHnfFdHZTCuN+hrRjJ11g2BwTnTptyXEa+Fg2AjXHPN1InnCSKo9LZht0B1ZYC18ccdnbrN1C+AyrtnLZnuF3MyYf0raGfmfEFuKcgvEb0gS1ULFCv0v1ZikTFjwD8C14Gw+NTPhxJ
4oxNBuFca2Tdc41eife7H/frYaHvW+102WaLOZDBQE4fEaQYutVoy+5N9utxjGsPaDuWI+/POv0nwULQ4z/3XAFQia1+B9dkkBJyDdJJORpHKRatCBMQ91JTCYFFiAerUf+be1+3Td5cjcA4ekV0pmc8rw
d/cayPZDS4cPD0zNXD0cURkVI4Mv6QLYfTxa+Pt+6w395wXRQkcF3tJ4iyELwzb0XSEWjE5X21a5bdsHug7txGmB3eg+BV+kIMM200FNyRVadWiXgiXSKnhX7cgBbWpUnaP813chl1jN1V6jgYfItoFM2nK
d2VEXY0WICCIgznWDH22jkindYN6pskeesfBR5fxpEe71pdYnpa5xtrV6SeKTY6BgCZ3SPUwfe2pqfJ45B0VdxNNSYKzjcX3r0908H4F/A3hGwlvGos6qd59w0rhf32IKECa0Z0YUynapBfOGfD/GCICKM3aR
ZkgIwbfTPUvBR2EGRzBz4zJh+NEWnDiMImUvKszK1zbQ+OZLk9HqSDmXKIM10vQdLu7Hxsj1NxxjKRYjHmBxks+v0tBn08vqR5eN9CEj9wnOwP6E9FQPACAO7A7XGIA6IC0QFHTA+3XFj36bs/CMmkDGPd
YQ0wRit1L6FKWRVeqbRzZ/dPOki3v/yNynS1qWvTLV8oYeQukQkVzPtkHeI3hegnfk0Kd+pLkZMKHQ1kD15jMppqcu+VDNj4pmvPQfLo1ON7qKKG9nfsNFni/2MqJkzQB4r02YwVuDs/iDXT8aGcQ+B7R
waCcZCcxSivLKMgQWKAU2z5YAhoC9IilWxZPUFR0PUCFuhTeweyMVxILayJMFHG55Y3rU4pG0Q++IKjjgT29Kc8/ZDPEB64qZQDyxYXQE5d3yU+Hj6hqxskBS8V2jVqhTHHU7r2kLIugpSXiX+kGuJsdnpua
F5Y0khL3Jpules7TSxm9Qjigc/+FokMSPka055LMgWgNBX2NZW7yVCuor9BJ/VQ+n5eDxPuwjP5RycjDuQ9nREJ41aUvRE0p06245N/V5VK+z0cwsTL9R1VEHUOYcAT+sTKjN9/8JBjY/bDHwBWMiB+br9B
3N/TInY3Bwvs7LXpeAcYNh4XKbUWQR7uEkCC+xSmaMXjJq+tpMTHSf7FQ=, trustStorePassword=meterPwd]"
Input arguments:
url: https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi
apiKey: cu9xY11564jKmhHkfaMHbxD0CzsRPOA8xntR7fL1n1Xi4Gcx2K6KD1kH8J3FGODGAYKKRZBCg0f+7QE2btWNJhekHnfFdHZTCuN+hrRjJ11g2BwTnTptyXEa+Fg2AjXHPN1InnCSKo9LZht0B
1ZYC18ccdnbrN1C+AyrtnLZnuF3MyYf0raGfmfEFuKcgvEb0gS1ULFCv0v1ZikTFjwD8C14Gw+NTPhxJ4oxNBuFca2Tdc41eife7H/frYaHvW+102WaLOZDBQE4fEaQYutVoy+5N9utxjGsPaDuWI+/POv0
nwULQ4z/3XAFQia1+B9dkkBJyDdJJORpHKRatCBMQ91JTCYFFiAerUf+be1+3Td5cjcA4ekV0pmc8rwd/cayPZDS4cPD0zNXD0cURkVI4Mv6QLYfTxa+Pt+6w395wXRQkcF3tJ4iyELwzb0XSEWjE5X21a5
bdsHug7txGmB3eg+BV+kIMM200FNyRVadWiXgiXSKnhX7cgBbWpUnaP813chl1jN1V6jgYfItoFM2nKd2VEXY0WICCIgznWDH22jkindYN6pskeesfBR5fxpEe71pdYnpa5xtrV6SeKTY6BgCZ3SPUwfe2p
qfJ45B0VdxNNSYKzjcX3r0908H4F/A3hGwlvGos6qd59w0rhf32IKECa0Z0YUynapBfOGfD/GCICKM3aRZkgIwbfTPUvBR2EGRzBz4zJh+NEWnDiMImUvKszK1zbQ+OZLk9HqSDmXKIM10vQdLu7Hxsj1Nxxj
KRYjHmBxks+v0tBn08vqR5eN9CEj9wnOwP6E9FQPACAO7A7XGIA6IC0QFHTA+3XFj36bs/CMmkDGPdYQ0wRit1L6FKWRVeqbRzZ/dPOki3v/yNynS1qWvTLV8oYeQukQkVzPtkHeI3hegnfk0Kd+pLkZJ
MKHQ1kD15jMppqcu+VDNj4pmvPQfLo1ON7qKKG9nfsNFni/2MqJkzQB4r02YwVuDs/iDXT8aGcQ+B7RwaCcZCcxSivLKMgQWKAU2z5YAhoC9IilWxZPUFR0PUCFuhTeweyMVxILayJMFHG55Y3rU4pG0Q++
IKjjgT29Kc8/ZDPEB64qZQDyxYXQE5d3yU+Hj6hqxskBS8V2jVqhTHHU7r2kLIugpSXiX+kGuJsdnpuaF5Y0khL3Jpules7TSxm9Qjigc/+FokMSPka055LMgWgNBX2NZW7yVCuor9BJ/VQ+n5eDxPuwjP5
RycjDuQ9nREJ41aUvRE0p06245N/V5VK+z0cwsTL9R1VEHUOYcAT+sTKjN9/8JBjY/bDHwBWMiB+br9B3N/TInY3Bwvs7LXpeAcYNh4XKbUWQR7uEkCC+xSmaMXjJq+tpMTHSf7FQ
trustStorePassword: *****
Creating keystore meteringTrustStore ...
Keystore was created: meteringTrustStore(cells/twasCell01|security.xml#KeyStore_1643038541489)
Retrieving signer from port ...
Signer was retrieved from host: cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com, port: 443 and store to keystore: meteringTrustStore
Creating was-usage-metering.properties file with all specified properties ...
Copying keystore meteringTrustStore.p12 and was-usage-metering.properties to all servers ...
keystoreFile meteringTrustStore.p12 was created on all servers.
was-usage-metering.properties was created on all servers.
No sync on WebSphere Base Server!
```

Go to the WSA UI to verify the registration

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.8)	twasServer1	CVE-2021-26296 (+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034 (+3 more)	twasCell01
None (0.0)	libertyServer1	—	RHEL7WAS1	21.0.0.12	1.8.0_312	—	—

Stop the server to save resources

```
/usr/IBM/WAS855ND/profiles/WSA/bin/stopServer.sh twasServer1
```



### Create a Liberty instance

#### Switch to RHEL7WAS1 (user ibmdemo, password passw0rd)

```
ssh ibmdemo@rhel7was1
```

#### Install a new set of Liberty binaries

```
cd /usr/IBM
java -jar /var/IBM/software/WAS/wlp-base-all-21.0.0.12.jar -acceptLicense
/usr/IBM/Liberty2
```

#### Create a Liberty server instance

```
/usr/IBM/Liberty2/wlp/bin/server create oomServer1
cp /var/IBM/software/WAS/MLApp.war /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/dropins
```

#### Adjust Liberty configuration to add monitor feature and change port

```
vi /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/server.xml
    <feature>monitor-1.0</feature>
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="29080"
        httpsPort="29443" />
```

### Register the Liberty instance to WSA

#### Create a config dropins directory for the new Liberty instance and copy the WSA server configuration:

```
mkdir -p /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/configDropins/defaults
cp /var/IBM/software/WAS/WSA_server.xml
/usr/IBM/Liberty2/wlp/usr/servers/oomServer1/configDropins/defaults/
```

#### Define the variables WSA\_Metering\_Keystore, WSA\_Metering\_URL and WSA\_Metering\_api-key.

```
mkdir -p /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/variables
cp /usr/IBM/WSA/WSA_metering_URL.txt
/usr/IBM/Liberty2/wlp/usr/servers/oomServer1/variables/WSA_metering_URL
echo "/usr/IBM/WSA/WSA_metering_Key.p12" >
/usr/IBM/Liberty2/wlp/usr/servers/oomServer1/variables/WSA_metering_keystore
cp /usr/IBM/WSA/WSA_metering_api-key.txt
/usr/IBM/Liberty2/wlp/usr/servers/oomServer1/variables/WSA_metering_api-key
ls /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/variables
```

#### Start the Liberty server oomServer1

```
/usr/IBM/Liberty2/wlp/bin/server start oomServer1
cat /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/logs/messages.log
```



## Install Instana Backend

### Prepare the Instana installation:

The installed Instana backend was quite old and there was a file system error which prevented updates. (/mnt/metrics/d9a554ec-6c04-44ab-a783-23925222b731.json.gz: Input/output error). Therefore, we install Instana from scratch and adjust settings like hostname etc.

### Change hostname from redhat.base to instanabackend.tec.uk.ibm.com

```
ssh root@instana.ocp46.tec.uk.ibm.com
[root@instana.ocp46.tec.uk.ibm.com ~]
vi /etc/hostname
adjust /etc/hosts
Reboot system
```

Adjust on the other systems the /etc/hosts file to replace

*instana.ocp46.tec.uk.ibm.com* with *instanabackend.tec.uk.ibm.com* instanabackend

### Change on instanabackend the mapping for the file system

```
ssh root@instanabackend
[root@instanabackend ~]
umount /mnt/install
umount /mnt/metrics
cat /etc/fstab
# Filesystem problem on /dev/instana/metrics, so switch install and metrics
# /dev/instana/install      /mnt/install  ext4    defaults    0          0
# /dev/instana/metrics     /mnt/metrics  ext4    defaults    0          0
/dev/instana/metrics       /mnt/install  ext4    defaults    0          0
/dev/instana/traces        /mnt/traces   ext4    defaults    0          0
/dev/instana/install       /mnt/metrics  ext4    defaults    0          0
/dev/instana/data          /mnt/data     ext4    defaults    0          0
mount /mnt/install
mount /mnt/metrics
```

## Install the Instana backend:

<https://www.ibm.com/docs/en/obi/current?topic=references-package-based-installation>

<https://github.ibm.com/up-and-running/watson-aiops/tree/master/docs/Instana/On-Prem%20installation%20-%20RHEL>

### Register the Instana repository

```
cat >/etc/yum.repos.d/Instana-Product.repo <<EOF
[instana-product]
name=Instana-Product
baseurl=https://self-hosted.instana.io/rpm/release/product/rpm/generic/x86_64/Packages
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://self-hosted.instana.io/signing_key.gpg
priority=5
sslverify=1
EOF
```

### Install the Instana console package 213 and lock the version

```
yum makecache -y
yum install -y instana-console-213
yum versionlock add instana-console
```

### Initialize Instana (use --force to overwrite existing data)

```
[root@instanabackend ~]# instana init --force
Setup host environment ✓
? Resuming will delete your data! Are you sure? Yes
? [Please choose Instana installation type] single
? [What is your tenant name?] ibminternalxl
? [What is your unit name?] educational
? [Insert your agent key (optional). If none is specified, one is generated which does not allow downloads.] qUMhYJxjSv6uZh2SyqTEnw
? [Insert your download key or official agent key (optional).] qUMhYJxjSv6uZh2SyqTEnw
? [Insert your sales key] pgABSBp_SnqIr5oMD68HoQ
? [Insert the FQDN of the host] instanabackend.tec.uk.ibm.com
? [Where should your data be stored?] /mnt/data
? [Where should your trace data be stored?] /mnt/traces
? [Where should your metric data be stored?] /mnt/metrics
? [Where should your logs be stored?] /var/log/instana
? [Path to your signed certificate file?] /root/cert/tls.crt
? [Path to your private key file?] /root/cert/tls.key
Handle certificates ✓
Ensure images ✓
Clean docker containers ✓
Check data directories ✓
Create configurations ✓
Run data stores ✓
Migrate data stores ✓
Run components ✓
Check components ✓
Setup environment urls ✓
Run nginx ✓
Initialize tenant unit ✓
Welcome to the World of Automatic Infrastructure and Application Monitoring
https://instanabackend.tec.uk.ibm.com
E-Mail: admin@instana.local
Password: CvmWEY7Fy0
```

## Add the Instana license

```
instana license download
```

```
Download license ✓
```

```
instana license import
```

```
Import license ✓
```

```
License is successfully imported
```

```
instana license verify
```

```
Verifying local license status ✓
```

```
Nr: 1
```

```
Valid from: 2021-02-22 00:00:00 +0000 UTC
```

```
Valid till: 2121-12-01 23:59:59.999 +0000 UTC
```

## Verify the installed Instana version

```
instana version
```

```
Instana self-hosted: 213-1 3.213.298-0 (6197a68574a46edccff1b5f63be7df5d7792fa7c)
```

## Access Instana dashboard

### Log into the Instana dashboard

```
URL: https://instanabackend.tec.uk.ibm.com
```

```
User: admin@instana.local/ Password: CvmWEY7Fy0
```

## Install Instana Agent

```
#!/bin/bash
```

```
curl -o setup_agent.sh https://setup.instana.io/agent && chmod 700 ./setup_agent.sh &&  
sudo ./setup_agent.sh -a qUMhYJxjSv6uZh2SyqTEnw -t dynamic -e  
instanabackend.tec.uk.ibm.com:1444 -j
```

If the install fails due to an existing package, use

```
yum erase instana-agent-dynamic-<package version>
```

## How to update Instana:

<https://www.ibm.com/docs/en/obi/current?topic=requirements-installing-host-agent-linux#rpm-based-distribution>

```
# Verify that the repository has been defined:
```

```
cat /etc/yum.repos.d/Instana-Agent.repo
```

```
# List available packages
```

```
yum list available | grep instana-console
```

```
instana-console.x86_64                215-6                instana-product
```

You cannot upgrade directly from an old to the latest version, you can only skip one version. So you might have to go step by step

```
yum install instana-console-xxx
```

```
instana update
```

## Register RHEL7WAS1 to Instana

### Log into the Instana dashboard

URL: <https://instanabackend.tec.uk.ibm.com>

User: admin@instana.local/ Password: DQgJvQ6x2Q

### Copy the command to install an agent on Linux

```
ssh ibmdemo@rhel7was1
#!/bin/bash
curl -o setup_agent.sh https://setup.instana.io/agent && chmod 700 ./setup_agent.sh &&
sudo ./setup_agent.sh -a qUMhYJxjSv6uZh2SyqTEnw -t dynamic -e
instanabackend.tec.uk.ibm.com:1444 -j
```

### Set the zone in Instana to 'WAS'

```
sudo vi /opt/instana/agent/etc/instana/configuration.yaml
```

### Verify that the Instana endpoint is defined in the hosts file /etc/hosts

```
192.168.1.109 instanabackend.tec.uk.ibm.com instanabackend
```

### Run the agent

```
systemctl start instana-agent.service
tail -f /opt/instana/agent/data/log/agent.log
```

### Adjust tWAS to allow Instana Monitoring

Add JVM options to make tWAS 8.5 monitored by Instana

```
cat /var/IBM/software/WAS/setJVMprops.py
```

```
AdminTask.setJVMProperties(['-nodeName AppSrv01 -serverName twasServer1 -
verboseModeClass false -verboseModeGarbageCollection false -verboseModeJNI false -
runHProf false -hprofArguments -debugMode false -debugArgs "-
agentlib:jdpw=transport=dt_socket,server=y,suspend=n,address=7777" -
executableJarFileName -genericJvmArguments "-XX:ShareClassesEnableBCI" -disableJIT
false'])
AdminConfig.save()
```

```
/usr/IBM/WAS855ND/profiles/WSA/bin/startServer.sh twasServer1
cd /usr/IBM/WAS855ND/profiles/WSA/bin
./wsadmin.sh -lang jython -connType SOAP -port 8880 -f
/var/IBM/software/WAS/setJVMprops.py
```

```
/usr/IBM/WAS855ND/profiles/WSA/bin/stopServer.sh twasServer1
```

## Register WSA to Instana (Webhook for alerts)

```
ssh root@instanabackend
```

```
vi /etc/hosts
```

```
192.168.1.109 instanabackend.tec.uk.ibm.com instanabackend
```

```
192.168.1.100 bastion.ocp46.tec.uk.ibm.com bastion cpd-websphere-
```

```
automation.apps.ocp46.tec.uk.ibm.com cp-console.apps.ocp46.tec.uk.ibm.com
```

```
api.apps.ocp46.tec.uk.ibm.com oauth-openshift.apps.ocp46.tec.uk.ibm.com
```

```
192.168.1.110 RHEL7WAS1.tec.uk.ibm.com RHEL7WAS1
```

```
192.168.1.111 RHEL7WAS2.tec.uk.ibm.com RHEL7WAS2
```

## [Setting up health monitoring - IBM Documentation](https://www.ibm.com/docs/en/ws-automation?topic=automation-setting-up-health-monitoring)

<https://www.ibm.com/docs/en/ws-automation?topic=automation-setting-up-health-monitoring>

Create a custom payload, an alert channel and an alert via Instana admin console as described in

<https://www.ibm.com/docs/en/ws-automation?topic=monitoring-setting-up-instana-send-alerts-websphere-automation>

## Create a custom payload

In the Instana dashboard, go to Settings > Custom Payload

Create a row with key: **PID**, select as Value Type **Dynamic** and select as Value **Internal > Process ID**

Save the created custom payload.

The screenshot shows the Instana Settings interface. The left sidebar contains icons for various settings categories. The main content area is titled 'Configure Custom Payload'. It includes a warning message: 'Each key/value pair will be included as additional payload to each Issue or Incident alert notification. Please refer to the Custom Payload documentation for more details.' Below this is a table with columns 'Key', 'Value type', and 'Value'. A single row is configured with 'custom: PID' as the key, 'Dynamic' as the value type, and 'Internal > Process ID' as the value. The value field also shows a 'string[]' type indicator and a trash icon. An 'Add Row' button is visible in the top right of the table area.

Key	Value type	Value
custom: PID	Dynamic	Internal > Process ID <span>string[]</span>

```
# Retrieve Instana Webhook URL
ssh root@bastion
oc login
oc project websphere-automation
Now using project "websphere-automation" on server
"https://api.apps.ocp46.tec.uk.ibm.com:6443".
oc get route cpd -o
jsonpath=https://{.spec.host}/websphereauto/health/webhooks/instana/memleak
https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/health/webhooks/instana/memleak
```

You will need both settings in the next step.

For details about the setting, look at the WSA documentation at <https://www.ibm.com/docs/en/ws-automation?topic=monitoring-setting-up-instana-send-alerts-websphere-automation>

[illegible]

Test the created channel:

To test the created channel, click on the created channel to open it, then use the pencil to edit it.

## WebSphere Automation Healing Webhook

Properties



Click on “**Test Channel**”.

If Test Channel fails with “cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com: Name or service not known”, verify that the WSA endpoint has been defined in the Instana’s /etc/hosts

```
ssh root@instanabackend
cat /etc/hosts
192.168.1.100 bastion.ocp46.tec.uk.ibm.com bastion cpd-websphere-
automation.apps.ocp46.tec.uk.ibm.com
```

You might have to restart Instana to activate the changes

```
sudo instana stop
sudo instana start
```

Go back to the Instana dashboard and click again to edit the alert channel, then click on “**Test Channel**”.

The test will likely fail if you use a self-signed certificate in WSA which has not been imported yet. Error message:

*PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target*

## Import the WSA certificate into Instana

# More details at

[https://www.instana.com/docs/self\\_hosted\\_instana/configuration/#custom-certificates-for-webhook-or-mail-integrations](https://www.instana.com/docs/self_hosted_instana/configuration/#custom-certificates-for-webhook-or-mail-integrations)

<https://www.ibm.com/docs/en/obi/current?topic=installer-configuring-instana#custom-certificates-for-webhook-or-mail-integrations>

# Create a cacerts keystore file and import the WSA certificate

```
ssh root@instanabackend
mkdir /var/instana
```

# Copy a current cacerts, for example:

```
cp /opt/instana/agent/jvm/lib/security/cacerts /var/instana/
```

# The WSA certificate can be retrieved via

```
echo | openssl s_client -showcerts -servername cpd-websphere-
automation.apps.ocp46.tec.uk.ibm.com -connect cpd-websphere-
automation.apps.ocp46.tec.uk.ibm.com:443 2>/dev/null | openssl x509 -inform pem >
cpd.pem
```

# Import WSA certificate

```
/opt/instana/agent/jvm/bin/keytool -importcert -file /var/instana/cpd.pem -alias
ibm.com -keystore /var/instana/cacerts -storepass changeit -noprompt
```

Verify that the certificate has been imported

```
/opt/instana/agent/jvm/bin/keytool -list -alias ibm.com -keystore /var/instana/cacerts
-storepass changeit
instana.com, 26 Jan 2022, trustedCertEntry,
Certificate fingerprint (SHA-256):
80:A5:69:7A:11:22:BD:07:98:70:3F:74:BC:9F:01:69:16:02:92:BC:62:2B:91:CF:7E:79:CA:2C:12:
05:B2:F0
```

Change keystore owner to instana

```
chown instana:instana /var/instana/cacerts
```

# Adjust Instana settings to make keystore known

```
cp /var/instana/settings.hcl /var/instana/settings.hcl.sav
vi /var/instana/settings.hcl
# Add at the beginning the line custom_keystore="/var/instana/cacerts"

instana update -f /root/settings.hcl
```

Verify that the custom keystore has been enabled:

```
cat /var/log/instana/console.log | grep Keystore
```

```
2022-01-26T13:08:50.129 INFO [ ] Custom Keystore is enabled: /var/instana/cacerts
```

Click again on **“Test Channel”**.

The test should work now.

## Modify WebSphere Automation Healing Webhook Alert Channel

Name

WebSphere Automation Healing Webhook

Webhook URLs

<https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com/websphereauto/health/webhooks/instana/memleak>

Remove

[Add Webhook URL](#)

### Custom HTTP Request Headers

Custom HTTP headers are useful to support authentication mechanisms other than HTTP basic authentication or to circumvent security mechanisms commonly available in content-delivery networks, e.g. bot detection.

Key

Content-Type

Value

application/json



Key

Authorization

Value

eyJhbGciOiJIUzI1NiIsImtpZCI6IjE6k5wOVNaRGtMWjdULXQ0Z2F1enpTa1EtYzM4C



[Add Header](#)

Test Channel

Alerting Channel was successfully triggered, please check the channel!

Cancel

Save



Create an alert via Instana admin console as described in <https://www.ibm.com/docs/en/ws-automation?topic=monitoring-setting-up-instana-send-alerts-websphere-automation>  
Add alert with event Memory leak (which has been introduced with Instana 213).

ACCESS CONTROL

Users

Pending Invitations

Groups

API Tokens

EVENTS & ALERTS

Events

Alerts

Alert Channels

Maintenance Windows

Custom Payload

LOG MANAGEMENT

Coralogix

ELK

Humio

LogDNA

Splunk

AUDIT

EVENTS & ALERTS

Events

Alerts

Alert Channels

Maintenance Windows

Custom Payload

LOG MANAGEMENT

Coralogix

ELK

Humio

LogDNA

Splunk

Create New Alert

1. Name

Name

WebSphere Automation Healing Memory Leak

Shows up in the list of alerts. Should be unique and meaningful.

2. Events




Only send alerts for event types or on selected events.

Alert on Event(s)

Events (1)

Add Events

Filter Events...

Name ↑	Description	Entity type
 Memory leak Built-in	This detector monitors garbage collection patterns to detect potential memory leaks.	 JVM 

3. Scope

Apply on (required)

All available entities

Caution! All events that match the event types will enter the notification stream.


Your selection matches no events in the past 2 weeks

4. Alerting

Alert Channels (1)

Add Alert Channels


Filter...

Name ↑	Properties
WebSphere Automation Healing Webhook Generic Webhook	

5. Custom Payloads (Optional)

Each key/value pair will be included as additional payload to each Issue or Incident alert notification.

Add Row

Key	Value
 No Custom Payloads Configured	

Click Create

For demos, add two additional events as the Memory Leak event is hard to trigger during a demo:

## Create a custom “Memory Usage High” event as shown in the screenshot:

Name: Memory Usage High

Description: Memory usage over 80 %

ACCESS CONTROL

Users

Pending Invitations

Groups

API Tokens

EVENTS & ALERTS

Events

Alerts

Alert Channels

Maintenance Windows

Custom Payload

LOG MANAGEMENT

Coralogix

ELK

Humio

LogDNA

Splunk

AUDIT

### 1. Event Details

Name

Memory Usage High

Shows up in the list of events. This is also the name of issues. Should be unique and meaningful.

Description

Memory usage over 80 %

Shows up in the issue description. Should be as descriptive as possible. Supports markdown.

Issue Severity

Warning

Incident

☒

Grace Period

90 s

### 2. Condition

Source

Built-in metrics

Entity type

JVM

Metric

Memory > Used percentage

Time Window

10 s

Aggregation

max

Operator

≥

Percentage

80


### 3. Scope

Apply on (required)

All available entities

Issue Preview

Started:  
2022-01-27, 07:19:54

 **MEMORY USAGE HIGH**

Memory usage over 80 %

Add the custom event as well the built-in event “**Garbage collection activity high**” to the alert.

Go to Alerts, select the WSA Health alert and click on Add Events

Pending Invitations

Groups

API Tokens

EVENTS & ALERTS

Events

Alerts

Alert Channels

Maintenance Windows

Custom Payload

LOG MANAGEMENT

Coralogix

### 1. Name

Name

WebSphere Automation Healing Memory Leak


Shows up in the list of alerts. Should be unique and meaningful.


### 2. Events




Only send alerts for event types or on selected events.

Alert on Event(s)

Events (1)

 Add Events

Filter Events... 

Name ↑	Description	Entity type
 Memory leak Built-in	This detector monitors garbage collection patterns to detect potential memory leaks.	 JVM 

Select the two events as shown in the screenshot.

Events Select All (7)

Type

Incidents & Severity...

JVM X

State...

Filter Events...

<input type="checkbox"/>	Name ↑	Description	Entity type
<input checked="" type="checkbox"/>	Garbage collection activity high Built-in	This detector establishes a processing pipeline to monitor the Garbage Collection time spent by the JVM Runtime Platform and validates it against the threshold value.	JVM
<input type="checkbox"/>	JVM Code Cache is full Built-in	This detector establishes a processing pipeline to monitor the maximum Code Cache usage of JVM Runtime Platform.	JVM
<input checked="" type="checkbox"/>	Memory Usage High	Memory usage over 80 %	JVM

Then click on Add 2 Events.

Events (3) + Add Events Filter Events...

Name ↑	Description	Entity type	
Garbage collection activity high Built-in	This detector establishes a processing pipeline to monitor the Garbage Collection time spent by the JVM Runtime Platform and validates it against the threshold value.	JVM	
Memory leak Built-in	This detector monitors garbage collection patterns to detect potential memory leaks.	JVM	
Memory Usage High	Memory usage over 80 %	JVM	

Finally click on Save to save the changes.

## Now configure the remote access

URL: <https://www.ibm.com/docs/en/ws-automation?topic=monitoring-setting-up-secure-remote-access>

Verify the requirements for the managed service

<https://www.ibm.com/docs/en/ws-automation?topic=requirements-managed-server#in-r-sysreqs-managed>

Verify on the servers to be monitored that python is installed via command

`python --version` or `python3 --version`

```
[ibmdemo@RHEL7WAS1 ~]$ python --version
```

```
Python 2.7.5
```

```
[ibmdemo@RHEL7WAS2 ~]$ python --version
```

```
Python 2.7.5
```

## If you want to use Memory Leak Detection with tWAS:

Verify WAS credentials being set in `soap.client.props`

Adjust tWAS to allow Instana Monitoring

Add JVM options to make tWAS 8.5 monitored by Instana

```
cat /var/IBM/software/WAS/setJVMprops.py
```

```
AdminTask.setJVMProperties('[-nodeName AppSrv01 -serverName twasServer1 -  
verboseModeClass false -verboseModeGarbageCollection false -verboseModeJNI false -  
runHProf false -hprofArguments -debugMode false -debugArgs "-  
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777" -  
executableJarFileName -genericJvmArguments "-XX:ShareClassesEnableBCI" -disableJIT  
false]')
```

```
AdminConfig.save()
```

```
/usr/IBM/WAS855ND/profiles/WSA/bin/startServer.sh twasServer1
```

```
cd /usr/IBM/WAS855ND/profiles/WSA/bin
```

```
./wsadmin.sh -lang jython -connType SOAP -port 8880 -f
```

```
/var/IBM/software/WAS/setJVMprops.py
```

```
/usr/IBM/WAS855ND/profiles/WSA/bin/stopServer.sh twasServer1
```

## Create an ssh key on the bastion node and copy it over to all WAS instances

Generate on WSA a new SSH key with passphrase passwOrd

```
ssh-keygen -f ~/.ssh/wsa
```

```
[root@bastion ~]# ssh-keygen -f ~/.ssh/wsa
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/wsa.
Your public key has been saved in /root/.ssh/wsa.pub.
The key fingerprint is:
SHA256:JFF8lPjZCIIlDJ0U/8TUPTYs88LgH+6fPhiWnwVSPwc root@bastion.ocp46.tec.uk.ibm.com
The key's randomart image is:
+---[RSA 3072]-----+
|  .+=000+.+         |
|    =0.+ = * B E    |
|   oo+* X + .       |
|    =. 0 + o .      |
|   So = . o         |
|    *               |
|   o + o            |
|    o +.            |
|    o+.             |
+---[SHA256]-----+
```

Copy the key over to the server to be monitored:

```
ssh-copy-id -i ~/.ssh/wsa ibmdemo@rhel7was1.tec.uk.ibm.com
```

```
[root@bastion ~]# [root@bastion ~]# ssh-copy-id -i ~/.ssh/wsa ibmdemo@rhel7was1.tec.uk.ibm.com
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/wsa.pub"
The authenticity of host 'rhel7was1.tec.uk.ibm.com (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:wm0919Wf8weQUglgSJ84VfUftfinzbqZWdUhQV1T1ZQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ibmdemo@rhel7was1.tec.uk.ibm.com's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ibmdemo@rhel7was1.tec.uk.ibm.com'"
and check to make sure that only the key(s) you wanted were added.
```

Test login via ssh key:

```
ssh -i ~/.ssh/wsa ibmdemo@rhel7was1.tec.uk.ibm.com
```

```
[root@bastion ~]# ssh -i ~/.ssh/wsa ibmdemo@rhel7was1.tec.uk.ibm.com
Enter passphrase for key '/root/.ssh/wsa':
Last login: Thu Jan 27 08:38:30 2022 from 10.139.253.68
```

Do the same for RHEL7WAS2

```
ssh-copy-id -i ~/.ssh/wsa ibmdemo@rhel7was2.tec.uk.ibm.com
```

```
ssh -i ~/.ssh/wsa ibmdemo@rhel7was2.tec.uk.ibm.com
```

## Setting up WebSphere Automation for SSH to Linux or UNIX servers

```
[root@bastion ~]
oc login -u admin -p passw0rd
oc project websphere-automation
oc create secret generic wsa-ansible \
  --from-literal=ansible_user=ibmdemo \
  --from-literal=ansible_port=22 \
  --from-file=ssh_private_key_file=/root/.ssh/wsa \
  --from-literal=ssh_private_key_password=passw0rd
[root@bastion ~]# oc create secret generic wsa-ansible \
> --from-literal=ansible_user=ibmdemo \
> --from-literal=ansible_port=22 \
> --from-file=ssh_private_key_file=/root/.ssh/wsa \
> --from-literal=ssh_private_key_password=passw0rd
secret/wsa-ansible created

ssh-keyscan rhel7was1.tec.uk.ibm.com >> /root/wsa_known_hosts
ssh-keyscan rhel7was2.tec.uk.ibm.com >> /root/wsa_known_hosts
[root@bastion ~]# ssh-keyscan rhel7was1.tec.uk.ibm.com >> /root/wsa_known_hosts
# rhel7was1.tec.uk.ibm.com:22 SSH-2.0-OpenSSH_7.4
# rhel7was1.tec.uk.ibm.com:22 SSH-2.0-OpenSSH_7.4
# rhel7was1.tec.uk.ibm.com:22 SSH-2.0-OpenSSH_7.4
[root@bastion ~]# ssh-keyscan rhel7was2.tec.uk.ibm.com >> /root/wsa_known_hosts
# rhel7was2.tec.uk.ibm.com:22 SSH-2.0-OpenSSH_7.4
# rhel7was2.tec.uk.ibm.com:22 SSH-2.0-OpenSSH_7.4
# rhel7was2.tec.uk.ibm.com:22 SSH-2.0-OpenSSH_7.4
```

```
oc create configmap wsa-ansible-known-hosts --from-
file=known_hosts=/root/wsa_known_hosts
[root@bastion ~]# oc create configmap wsa-ansible-known-hosts --from-file=known_hosts=/root/wsa_known_hosts
configmap/wsa-ansible-known-hosts created
```

```
# Test connection
MANAGER_POD=$(oc get pod -l app.kubernetes.io/component=runbook-manager -o name | head
-n 1)
oc rsh $MANAGER_POD runcli testConnection rhel7was1 linux
```

```
[root@bastion ~]# MANAGER_POD=$(oc get pod -l app.kubernetes.io/component=runbook-manager -o name | head -n 1)
[root@bastion ~]# oc rsh $MANAGER_POD runcli testConnection rhel7was1 linux
Jan 27, 2022 9:06:28 AM com.ibm.ws.automation.core.runbook.manager.RunbookManagerCLI clientMode
INFO: starting
Created job: test-connection-1643274389933
To see the job logs run: oc logs --tail=100 -l job-name=test-connection-1643274389933
[root@bastion ~]# oc rsh $MANAGER_POD runcli testConnection rhel7was2 linux
Jan 27, 2022 9:06:40 AM com.ibm.ws.automation.core.runbook.manager.RunbookManagerCLI clientMode
INFO: starting
Created job: test-connection-1643274401721
To see the job logs run: oc logs --tail=100 -l job-name=test-connection-1643274401721
```

## Prepare Liberty for OOM

Set the Java heap low enough to trigger an OOM event in Instana

```
echo "-Xmx98m" >> /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/jvm.options
cat /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/jvm.options
[ibmdemo@RHEL7WAS1 ~]$ echo "-Xmx98m" >> /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/jvm.options
[ibmdemo@RHEL7WAS1 ~]$ cat /usr/IBM/Liberty2/wlp/usr/servers/oomServer1/jvm.options
-Xmx98m
```

## Restart the Liberty instance

```
/usr/IBM/Liberty2/wlp/bin/server stop oomServer1
/usr/IBM/Liberty2/wlp/bin/server start oomServer1
```

## Access the MLApp

```
curl http://localhost:29080/MLApp
```

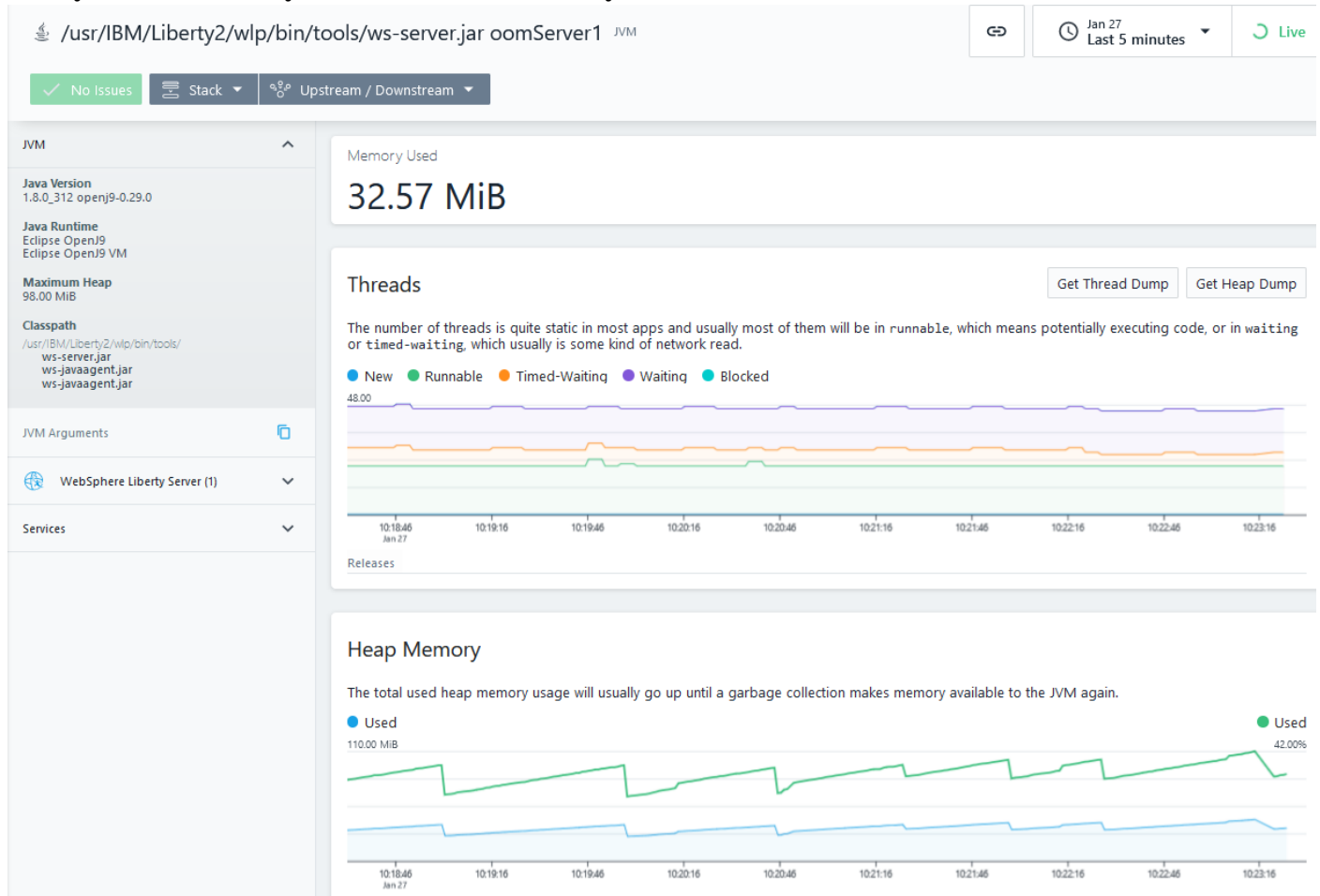
```
curl http://localhost:29080/MLApp/MLVectorParam?myaction=add
```

```
[ibmdemo@RHEL7WAS1 ~]$ curl http://localhost:29080/MLApp
```

```
[ibmdemo@RHEL7WAS1 ~]$ curl http://localhost:29080/MLApp/MLVectorParam?myaction=add
```

```
<html><body><center><h1>MLVector</h1></center><hr><hr><center>RHEL7WAS1 -uHM7j8Rx_pogZtTtV4uNi_ </center><hr></body></html>
```

## Verify that the Liberty instance is monitored by Instana



## Create stress test scripts on the Liberty host

```
cat /usr/IBM/scripts/wsaHealth_loadHeap.sh
#!/bin/sh
echo "Cleanup Java Heap"
curl --silent --output /dev/null
http://localhost:29080/MlApp/MlVectorParam?myaction=removeAll
number=0
#echo "sleep"
#sleep 5
while [ "$number" -lt 25 ]
do
    curl --silent --output /dev/null
http://localhost:29080/MlApp/MlVectorParam?myaction=add
    number=`expr $number + 1 `
    sleep 1
    echo "Grow heap, call $number"
done
```

```
cat /usr/IBM/scripts/wsaHealth_overLoad.sh
#!/bin/sh
number=0
while [ "$number" -lt 25 ]
do
    curl --silent --output /dev/null
http://localhost:29080/MlApp/MlVectorParam?myaction=add
    number=`expr $number + 1 `
    sleep 1
    echo "Cause OOM, call $number"
done
echo "wait 3 minute"
sleep 180
echo "Cleanup Java Heap"
curl --silent --output /dev/null
http://localhost:29080/MlApp/MlVectorParam?myaction=removeAll
```

```
cat /usr/IBM/scripts/wsaHealth_offLoad.sh
#!/bin/sh
echo "Cleanup Java Heap"
curl --silent --output /dev/null
http://localhost:29080/MlApp/MlVectorParam?myaction=removeAll

chmod +x /usr/IBM/scripts/wsaHealth_*
```

## Run stress test

### Test WSA Health

#### Start the warmup script on the system running the oomServer1

```
/usr/IBM/Liberty2/wlp/bin/server stop oomServer1
/usr/IBM/Liberty2/wlp/bin/server start oomServer1
/usr/IBM/scripts/wsaHealth_loadHeap.sh
/usr/IBM/scripts/wsaHealth_overLoad.sh
```

!!! Test, how many requests are required to trigger OOM condition !!!



## Troubleshooting

Investigation failed in WSA

Detected 1/27/2022 12:17:00 PM

Investigation ID cbc17e7c-2992-424f-81b4-1c17711e6b58

Problem type Memory leak

Status Investigation failed

Server Not identified yet

Host RHEL7WAS1.tec.uk.ibm.com

### Why investigation started

An alert was received that indicated a potential memory leak.

[View in Instana](#)

### Analysis failed

The analysis of the investigation failed to complete. Expand View history to see if any steps completed.

- Failed to identify the server on host 'RHEL7WAS1.tec.uk.ibm.com'.

[Download all files](#)

When downloading the files, the runbook.log shows:

```
"Failed to connect to the host via ssh: ssh: Could not resolve hostname
rhel7was1.tec.uk.ibm.com: Name or service not known"
```

<https://www.ibm.com/docs/en/ws-automation?topic=troubleshooting-setup-issues>

```
cat <<EOF | oc apply -f -
apiVersion: automation.websphere.ibm.com/v1
kind: WebSphereAutomation
metadata:
  name: wsa
spec:
  runbookManager:
    runbookRunner:
      hostAliases:
      - hostnames:
        - rhel7was1.tec.uk.ibm.com
        ip: 192.168.1.110
EOF
```

This did not resolve the issue, so set hostname on all OCP nodes

```
[root@bastion ~]# ssh core@master-1
sudo -i
vi /etc/hosts
192.168.1.110 RHEL7WAS1.tec.uk.ibm.com RHEL7WAS1
192.168.1.111 RHEL7WAS2.tec.uk.ibm.com RHEL7WAS2
```

Repeat the same steps for master-2, master-3, worker-1 to worker-5



## Install components for additional demos

### Install DayTrader onto WAS ND:

Log as ibmdemo into RHEL7WAS2

[ibmdemo@RHEL7WAS2 WAS855ND]

### Create WAS ND cell for DayTrader:

Create a Deployment Manager profile via command:

```
[ibmdemo@RHEL7WAS2 WAS855ND]
export WAS_HOME=/usr/IBM/WAS855ND
$WAS_HOME/bin/manageprofiles.sh -create \
-templatePath $WAS_HOME/profileTemplates/management \
-serverType DEPLOYMENT_MANAGER \
-profileName Dmgr01 \
-cellName Default01Cell \
-nodeName Dmgr01Node \
-enableAdminSecurity true \
-adminUserName wasadmin \
-adminPassword passw0rd
```

Get SOAP port via command:

```
cat /usr/IBM/WAS855ND/profiles/Dmgr01/logs/AboutThisProfile.txt | grep SOAP
```

Add WAS credentials:

```
vi /usr/IBM/WAS855ND/profiles/Dmgr01/properties/soap.client.props
```

Start Dmgr:

```
$WAS_HOME/profiles/Dmgr01/bin/startManager.sh
```

Set JVM property for Instana discovery:

```
vi $WAS_HOME/profiles/Dmgr01/bin/configureInstana.py
AdminTask.setJVMProperties('[-nodeName Dmgr01Node -serverName dmgr -verboseModeClass
false -verboseModeGarbageCollection false -verboseModeJNI false -runHProf false -
hprofArguments -debugMode false -debugArgs "-Djava.compiler=NONE -Xdebug -Xnoagent -
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=7792" -executableJarFileName -
genericJvmArguments "-XX:ShareClassesEnableBCI" -disableJIT false]')
AdminConfig.save()
```

```
$WAS_HOME/profiles/Dmgr01/bin/wsadmin.sh -lang jython -connType SOAP -port 8879 -f
$WAS_HOME/profiles/Dmgr01/bin/configureInstana.py
```

Restart Dmgr:

```
$WAS_HOME/profiles/Dmgr01/bin/stopManager.sh
$WAS_HOME/profiles/Dmgr01/bin/startManager.sh
```

Create profile for standalone WAS

```
export WAS_HOME=/usr/IBM/WAS855ND
$WAS_HOME/bin/manageprofiles.sh -create \
-profileName AppSrv1 \
-templatePath $WAS_HOME/profileTemplates/default \
-nodeName AppSrv1Node \
-cellName AppSrv1Cell \
-enableAdminSecurity true \
-adminUserName wasadmin \
-adminPassword passw0rd
```

Add WAS credentials:

```
vi $WAS_HOME/profiles/AppSrv1/properties/soap.client.props
```

Federate node into Dmgr cell:

```
$WAS_HOME/profiles/AppSrv1/bin/addNode.sh RHEL7WAS2 -includeapps -user wasadmin -
password passw0rd
```

**Check status:**

```
$WAS_HOME/profiles/AppSrv1/bin/serverStatus.sh -all
```

**Start Server1:**

```
$WAS_HOME/profiles/AppSrv1/bin/startServer.sh server1
```

**Test Default app:**

```
curl http://localhost:9080/snoop
```

**Set JVM property for Instana discovery:**

```
vi $WAS_HOME/profiles/AppSrv1/bin/configureInstana.py
AdminTask.setJVMProperties('[-nodeName AppSrv1Node -serverName nodeagent -
verboseModeClass false -verboseModeGarbageCollection false -verboseModeJNI false -
runHProf false -hprofArguments -debugMode false -debugArgs "-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777" -
executableJarFileName -genericJvmArguments "-XX:ShareClassesEnableBCI" -disableJIT
false]')

AdminTask.setJVMProperties('[-nodeName AppSrv1Node -serverName server1 -
verboseModeClass false -verboseModeGarbageCollection false -verboseModeJNI false -
runHProf false -hprofArguments -debugMode false -debugArgs "-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777" -
executableJarFileName -genericJvmArguments "-XX:ShareClassesEnableBCI" -disableJIT
false]')

AdminConfig.save()
```

```
$WAS_HOME/profiles/Dmgr01/bin/wsadmin.sh -lang jython -connntype SOAP -port 8879 -f
$WAS_HOME/profiles/AppSrv1/bin/configureInstana.py
```

**Restart Node:**

```
$WAS_HOME/profiles/AppSrv1/bin/stopServer.sh server1
$WAS_HOME/profiles/AppSrv1/bin/stopNode.sh
$WAS_HOME/profiles/AppSrv1/bin/startNode.sh
$WAS_HOME/profiles/AppSrv1/bin/startServer.sh server1
```

**Check status:**

```
$WAS_HOME/profiles/AppSrv1/bin/serverStatus.sh -all
```

Now you can access the Deployment Manager console via URL

<https://rhel7was2:9043/ibm/console>

## Apply fix to remove Risk Level 10 issue (Log4J)

```
$WAS_HOME/profiles/AppSrv1/bin/stopServer.sh TradeServer1
$WAS_HOME/profiles/AppSrv1/bin/stopServer.sh TradeServer2
$WAS_HOME/profiles/AppSrv1/bin/stopServer.sh server1
$WAS_HOME/profiles/AppSrv1/bin/stopNode.sh
$WAS_HOME/profiles/Dmgr01/bin/stopManager.sh
```

```
/var/IBM/software/WAS/was_CVE.sh
[ibmdemo@RHEL7WAS1 bin]$ cat /var/IBM/software/WAS/was_CVE.sh
export fixID="8.5.5.11-WS-WASProd-IFPH42762"
export fixRepo="8.5.5.11-ws-wasprod-ifph42762.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
echo "Apply Fix $fixID"
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
export fixID="8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271"
export fixRepo="8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271.zip"
echo "Apply Fix $fixID"
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
```

## Install DayTrader app and DB2

### Install DayTrader application

#### Copy files

```
scp D:\Software\WAS\sampleApps\DayTrader\DayTrader3Install_v85.zip
ibmdemo@rhel7was2://var/IBM
scp D:\Software\DB2\db2drivers.zip ibmdemo@rhel7was2://var/IBM
```

#### Create db2drivers directory

```
unzip /var/IBM/db2drivers.zip -d /usr/IBM/
Files are now in /usr/IBM/db2drivers
```

#### Create a Daytrader DB2 database instance

```
docker pull ibmcom/db2
sudo mkdir -p /var/IBM/db2/trade3
sudo chown -R ibmdemo:ibmdemo /var/IBM
docker run -itd --name mytrade3db --privileged=true -p 50003:50000 -e LICENSE=accept -e
DB2INST1_PASSWORD=passw0rd -e DBNAME=tradedb -v /var/IBM/db2/trade3 ibmcom/db2
```

#### Install DayTrader3

```
mkdir -p /var/IBM/temp
unzip /var/IBM/DayTrader3Install_v85.zip -d /var/IBM/temp
cd /var/IBM/temp/DayTrader3Install/DayTrader3-EE6
```

```
$WAS_HOME/profiles/Dmgr01/bin/wsadmin.sh -lang jython -conntype SOAP -port 8879 -f
daytrader_cluster.py
True
```

```
/usr/IBM/db2drivers/db2jcc.jar: /usr/IBM/db2drivers/db2jcc_license_cu.jar
```

```
[ibmdemo@RHEL7WAS2 DayTrader3-EE6]$ $WAS_HOME/profiles/Dmgr01/bin/wsadmin.sh -lang
jython -conntype SOAP -port 8879 -f daytrader_cluster.py
```

```
Global security is (or will be) enabled (true|false) [false]:true
Have all nodes been federated and network connectivity verified? (yes|no) [yes]:yes
Please enter the cluster name [TradeCluster]:
Select the desired node [AppSrv1Node]:
Please enter the cluster member name [TradeServer1]:
Add more cluster members (yes|no) [yes]:
Select the desired node [AppSrv1Node]:
Please enter the cluster member name [TradeServer2]:
Current Cluster Nodes and Members:
AppSrv1Node - TradeServer1
AppSrv1Node - TradeServer2
Add more cluster members (yes|no) [yes]:no
Select the JDBC provider type [DB2 Universal]:
Select the EJB deployment target [DB2UDB_V82]:
Please enter the location of JDBC driver (jar) files:
/usr/IBM/db2drivers/db2jcc.jar: /usr/IBM/db2drivers/db2jcc_license_cu.jar
Please enter the database name (location) [tradedb]:
Please enter the database hostname [localhost]:rhel7was2
Please enter the database port number [50000]:50003
Please enter the database username [userid]:db2inst1
Please enter the database password [password]:passw0rd
Please enter a valid administrative username [AdminUserID]:wasadmin
Please enter a valid administrative password [password]:passw0rd
```

...

```
-----
DayTrader Installation Completed!!!
-----
```

## Restart the cell

```
$WAS_HOME/profiles/AppSrv1/bin/stopServer.sh server1
$WAS_HOME/profiles/AppSrv1/bin/stopNode.sh
$WAS_HOME/profiles/Dmgr01/bin/stopManager.sh
$WAS_HOME/profiles/Dmgr01/bin/startManager.sh
$WAS_HOME/profiles/AppSrv1/bin/startNode.sh
```

## Set JVM property for Instana discovery:

```
vi $WAS_HOME/profiles/AppSrv1/bin/configureInstana4Trade.py
AdminTask.setJVMProperties('[-nodeName AppSrv1Node -serverName TradeServer1 -
verboseModeClass false -verboseModeGarbageCollection false -verboseModeJNI false -
runHProf false -hprofArguments -debugMode false -debugArgs "-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777" -
executableJarFileName -genericJvmArguments "-XX:ShareClassesEnableBCI" -disableJIT
false]')

AdminTask.setJVMProperties('[-nodeName AppSrv1Node -serverName TradeServer2 -
verboseModeClass false -verboseModeGarbageCollection false -verboseModeJNI false -
runHProf false -hprofArguments -debugMode false -debugArgs "-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777" -
executableJarFileName -genericJvmArguments "-XX:ShareClassesEnableBCI" -disableJIT
false]')

AdminConfig.save()

$WAS_HOME/profiles/Dmgr01/bin/wsadmin.sh -lang jython -connType SOAP -port 8879 -f
$WAS_HOME/profiles/AppSrv1/bin/configureInstana4Trade.py
```

## Start the TradeServers

```
$WAS_HOME/profiles/AppSrv1/bin/startServer.sh TradeServer1
$WAS_HOME/profiles/AppSrv1/bin/startServer.sh TradeServer2
```

Access the application via <http://rhel7was2:9081/daytrader>  
and recreate and repopulate the database

```
Install jmeter
scp D:\PoTs\Liberty\Liberty_vPoT\apache-jmeter-4.0.zip
ibmdemo@rhel7was2:/var/IBM/software
unzip /var/IBM/software/apache-jmeter-4.0.zip -d /usr/IBM
cp -R /var/IBM/temp/DayTrader3Install/DayTrader3-EE6/Apache_JMeter_script
/usr/IBM/scripts/
```

## Run a loadtest:

```
/usr/IBM/apache-jmeter-4.0/bin/jmeter -n -t
/usr/IBM/scripts/Apache_JMeter_script/daytrader3.jmx -JHOST=localhost -JPORT=9081 -
JDURATION=300
/usr/IBM/apache-jmeter-4.0/bin/jmeter -n -t
/usr/IBM/scripts/Apache_JMeter_script/daytrader3.jmx -JHOST=localhost -JPORT=9082 -
JDURATION=300
```

Create an Instana application perspective for DayTrader

New Application Perspective

Switch to Advanced Mode

Step 1: Select Model

Step 2: Specify Application

Step 3: Provide Details

Specify your Application Perspective

ServicesEndpoints

→ Agent Zone = WAS AND

→ Endpoint Name contains trade Add filter

× Clear

Which downstream services would you like to include?

☐ No downstream services

☐ Immediate downstream database and messaging services

☒ All downstream services

Matched services in the last hour

TradeServer1HTTPRPC

TradeServer2RPC

tradedbDATABASE

New Application Perspective

Switch to Advanced Mode

Step 1: Select Model

Step 2: Specify Application

Step 3: Provide Details

What is the name of this Application Perspective?

Application Perspective Name

DayTrader

Are you interested just in the calls to this application, or also the internal calls?

Inbound Calls

☐ The dashboard is displaying information based only on the calls that are performed by the consumers of this application.

All Calls

☒ The dashboard is displaying information based on all calls that are performed within this application, by both consumers as well as internally.

Matched services in the last hour

TradeServer1HTTPRPC

TradeServer2RPC

tradedbDATABASE

Back

Create



## Install Transformation Advisor into OCP

<https://www.ibm.com/docs/en/cta?topic=started-operator-install-ocp>

```
ssh root@bastion
oc login
```

### Perform a "Single Namespace" Operator Installation

<https://www.ibm.com/docs/en/cta?topic=started-operator-install-ocp>

```
oc new-project wshe
```

```
cat <<EOF | oc apply -f -
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ta-operator-sa
  namespace: wshe
  labels:
    release: "operator"
    app.kubernetes.io/name: ta-operator
    app.kubernetes.io/instance: ta-instance
    app.kubernetes.io/managed-by: helm
EOF
```

```
cat <<EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ta-ownnamespace-operator-group
  namespace: wshe
spec:
  serviceAccountName: ta-operator-sa
  targetNamespaces:
    - wshe
EOF
```

```
cat <<EOF | oc apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  creationTimestamp: null
  name: ta-operator-sa
  namespace: wshe
  labels:
    release: "operator"
    app.kubernetes.io/name: ta-operator
    app.kubernetes.io/instance: ta-instance
    app.kubernetes.io/managed-by: helm
rules:
- apiGroups: ["networking.k8s.io"]
  resources: ["networkpolicies"]
  verbs: ["get", "list", "watch", "create", "delete", "patch"]
- apiGroups: ["console.openshift.io"]
  resources: ["consoleyamlsamples"]
  verbs: ["create", "patch"]
- apiGroups: ["config.openshift.io"]
  resources: ["ingresses", "infrastructures", "dnses"]
  verbs: ["get"]
- apiGroups: ["oauth.openshift.io"]
  resources: ["oauthclients", "oauthclients/finalizers"]
  verbs: ["get", "list", "create", "delete"]
- apiGroups: ["operators.coreos.com"]
  resources: ["clusterserviceversions"]
  verbs: ["get", "list", "watch", "create", "delete", "patch"]
- apiGroups: ["apiextensions.k8s.io"]
  resources: ["customresourcedefinitions", "customresourcedefinitions/finalizers"]
```

```

  verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: [""]
  resources: ["namespaces"]
  verbs: ["get"]
- apiGroups: ["batch"]
  resources: ["jobs"]
  verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: [""]
  resources: ["configmaps", "persistentvolumeclaims", "pods", "services", "secrets",
"serviceaccounts"]
  verbs: ["*"]
- apiGroups: ["apps"]
  resources: ["deployments", "statefulsets"]
  verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: ["monitoring.coreos.com"]
  resources: ["servicemonitors"]
  verbs: ["get", "create"]
- apiGroups: ["apps"]
  resourceNames: ["ta-operator"]
  resources: ["deployments/finalizers", "statefulsets/finalizers"]
  verbs: ["update"]
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get"]
- apiGroups: ["apps"]
  resources: ["replicasets"]
  verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: ["ta.ibm.com"]
  resources: ["*"]
  verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: ["route.openshift.io"]
  resources: ["routes", "routes/custom-host"]
  verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: ["rbac.authorization.k8s.io"]
  resources: ["clusterrolebindings", "clusterroles", "clusterroles/finalizers",
"roles", "rolebindings", "roles/finalizers", "rolebindings/finalizers"]
  verbs: ["*"]
EOF

```

```

cat <<EOF | oc apply -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ta-operator-sa
  namespace: wshe
  labels:
    release: "operator"
    app.kubernetes.io/name: ta-operator
    app.kubernetes.io/instance: ta-instance
    app.kubernetes.io/managed-by: helm
subjects:
- kind: ServiceAccount
  name: ta-operator-sa
roleRef:
  kind: Role
  name: ta-operator-sa
  apiGroup: rbac.authorization.k8s.io
EOF

```

```

cat <<EOF | oc apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: ta-operator-sa
  labels:
    release: "operator"
    app.kubernetes.io/name: ta-operator

```

```

    app.kubernetes.io/instance: ta-instance
    app.kubernetes.io/managed-by: helm
rules:
- apiGroups: ["oauth.openshift.io"]
  resources: ["oauthclients", "oauthclients/finalizers"]
  verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: ["config.openshift.io"]
  resources: ["ingresses", "infrastructures", "dnses"]
  verbs: ["get"]
EOF

cat <<EOF | oc apply -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ta-operator-sa
  labels:
    release: "operator"
    app.kubernetes.io/name: ta-operator
    app.kubernetes.io/instance: ta-instance
    app.kubernetes.io/managed-by: helm
subjects:
- kind: ServiceAccount
  name: ta-operator-sa
  namespace: wshe
roleRef:
  kind: ClusterRole
  name: ta-operator-sa
  apiGroup: rbac.authorization.k8s.io
EOF

```

## Install the operator using the Red Hat OpenShift UI

### 1. Find TA 3.0 in the Operator Hub

### 2. Create the required objects as described in the OCP UI TA Operator:

```

Create custom SecurityContextConstraint
cat <<EOF | oc apply -f -
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities: null
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: MustRunAs
  ranges:
  - max: 1002
    min: 1001
groups: []
kind: SecurityContextConstraints
metadata:
  annotations:
    name: transadv-scc
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities:
- KILL
- MKNOD
- SETUID
- SETGID
runAsUser:
  type: MustRunAs

```

```
uid: 1001
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users: []
volumes:
- configMap
- downwardAPI
- emptyDir
- persistentVolumeClaim
- projected
- secret
EOF
```

```
oc adm policy add-scc-to-user transadv-scc system:serviceaccount:wshe:ta-operator
```

### 3. Click on Install


- specify wshe as target namespace
- Click on Install




The installation of the TA operator can take up to 20 minutes before succeeding.

Project: wshe ▾

#### Installed Operators

Installed Operators are represented by Cluster Service Versions within this namespace. For more information, see the [Understanding Operators documentation](#) or create an Operator and Cluster Service Version using the [Operator SDK](#).


Name ▾ Search by name... 

Name ↑	Managed Namespaces ⓘ	Status	Last Updated	Provided APIs
 <b>Instana Agent Operator</b> 1.0.5 provided by Instana	All Namespaces	✓ Succeeded Up to date	🕒 Jan 21, 12:23 pm	<a href="#">Instana Agent</a> ⋮
 <b>IBM Transformation Advisor</b> 3.0.0 provided by IBM	 wshe	✓ Succeeded Up to date	🕒 Jan 25, 9:20 am	<a href="#">Transformation Advisor</a> ⋮

### Create a TA instance

Project: wshe ▾

[Installed Operators](#) > [Operator Details](#)

 **IBM Transformation Advisor**  
3.0.0 provided by IBM

Actions ▾

[Details](#) [YAML](#) [Subscription](#) [Events](#) [Transformation Advisor](#)

#### TransAdv

Create TransAdv

No Operands Found

Operands are declarative components used to define the behavior of the application.

Specify the license and accept it. Specify if persistence is needed (in my case not)

## Create TransAdv

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: ☒ Form View ☐ YAML View

**Note:** Some fields may not be represented in this form view. Please select "YAML view" for full control.

### Name \*

ta

### Labels

release=operator × app.kubernetes.io/instance=ta-instance × app.kubernetes.io/name=ta-operator ×

### License

Transformation Advisor License Configuration

#### License Type \*

IBM WebSphere Hybrid Edition 5.0.1 - L-DCHS-C3TPUM ▾

License information is available here: <https://ibm.biz/transadv-license>

#### License Accept \*



true

License information is available here: <https://ibm.biz/transadv-license>

### Database Persistence Configuration

Protect your Transformation Advisor data by configuring persistence (recommended)

#### Persistence Enable \*



false

Enable or disable persistence. Disabling is not recommended.

## Related YAML

```
apiVersion: ta.ibm.com/v2
kind: TransAdv
metadata:
  name: ta
  labels:
    release: operator
    app.kubernetes.io/instance: ta-instance
    app.kubernetes.io/name: ta-operator
    app.kubernetes.io/managed-by: helm
  namespace: wshe
spec:
  networkPolicy:
    enabled: true
  license:
    aLicenseType: IBM WebSphere Hybrid Edition 5.0.1 - L-DCHS-C3TPUM
    accept: true
  nodePortConfig:
    libertyPort: ''
    routerNodeIp: ''
    uiPort: ''
  fips:
    enabled: false
  route:
    enabled: true
    hostname: ta.apps.ocp46.tec.uk.ibm.com
    rateLimit:
      enabled: true
      rateHttp: '200'
      rateTcp: '200'
  commonServices:
    cloudpakId: __TA_PLACEHOLDER_CLOUDPAK_ID__
    cloudpakName: __TA_PLACEHOLDER_CLOUDPAK_NAME__
    cloudpakVersion: __TA_PLACEHOLDER_CLOUDPAK_VERSION__
  transadv:
    livenessProbe:
      failureThreshold: 15
      initialDelaySeconds: 60
      periodSeconds: 5
      timeoutSeconds: 30
    logLevel: info
    publicUrl: 'https://ta.apps.ocp46.tec.uk.ibm.com'
    readinessProbe:
      failureThreshold: 15
      initialDelaySeconds: 60
      periodSeconds: 5
      timeoutSeconds: 30
  resources:
    limits:
      cpu: 16000m
      ephemeral-storage: 20Gi
      memory: 8Gi
    requests:
```

```
    cpu: 500m
    ephemeral-storage: 3Gi
    memory: 1Gi
couchdb:
  livenessProbe:
    failureThreshold: 6
    initialDelaySeconds: 60
    periodSeconds: 5
    timeoutSeconds: 3
  readinessProbe:
    failureThreshold: 6
    initialDelaySeconds: 5
    periodSeconds: 5
    timeoutSeconds: 3
  resources:
    limits:
      cpu: 16000m
      ephemeral-storage: 20Gi
      memory: 20Gi
    requests:
      cpu: 500m
      ephemeral-storage: 1Gi
      memory: 1Gi
neo4j:
  livenessProbe:
    failureThreshold: 6
    initialDelaySeconds: 60
    periodSeconds: 5
    timeoutSeconds: 3
  readinessProbe:
    failureThreshold: 6
    initialDelaySeconds: 5
    periodSeconds: 5
    timeoutSeconds: 3
  resources:
    limits:
      cpu: 16000m
      ephemeral-storage: 5Gi
      memory: 8Gi
    requests:
      cpu: 500m
      ephemeral-storage: 1Gi
      memory: 1Gi
custom:
  landingIcon: ''
  landingIcon2: ''
  landingIcon3: ''
  landingIcon4: ''
  titleText: ''
authentication:
  disabled:
    liberty: false
    ui: false
ocp:
```

```
apiEndpoint: 'https://api.ocp46.tec.uk.ibm.com:6443'
authIssuerEndpoint: 'https://oauth-openshift.apps.ocp46.tec.uk.ibm.com'
secretName: transformation-advisor-secret
oidc:
  clientId: 310687cf3341734f8931ad1562e8f153d48c00ad
  clientSecret: 8a907494e19510af2d67a4df6756704f1c193e9b
thirdParty: {}
tls:
  caCert: ''
  enabled: true
transadvui:
  livenessProbe:
    failureThreshold: 15
    initialDelaySeconds: 60
    periodSeconds: 30
    timeoutSeconds: 5
  logLevel: info
  readinessProbe:
    failureThreshold: 15
    initialDelaySeconds: 5
    periodSeconds: 30
    timeoutSeconds: 5
  resources:
    limits:
      cpu: 16000m
      ephemeral-storage: 5Gi
      memory: 4Gi
    requests:
      cpu: 500m
      ephemeral-storage: 1Gi
      memory: 1Gi
  useSecureCookie: 'true'
persistence:
  couchdb:
    accessMode: ReadWriteOnce
    existingClaim: ''
    size: 20Gi
    storageClassName: ''
    supplementalGroups: []
    useDynamicProvisioning: true
  enabled: false
  neo4j:
    accessMode: ReadWriteOnce
    existingClaim: ''
    size: 5Gi
    storageClassName: ''
    supplementalGroups: []
    useDynamicProvisioning: true
```



Wait until the pods are ready:

```
[root@bastion ~]# oc get pod
```

NAME	READY	STATUS	RESTARTS	AGE
ta-couchdb-0	1/1	Running	0	11m
ta-neo4j-0	1/1	Running	0	11m
ta-operator-6b4578c8f9-j7jvc	1/1	Running	0	46m
ta-operator-instance-preinstall-2gl7d	0/1	Completed	0	11m
ta-server-7c676d6f45-kbh5v	1/1	Running	0	11m
ta-ui-8548695cb8-qmfj8	1/1	Running	0	11m

Get the URL via command

```
oc get route
```

```
[root@bastion ~]# oc get route
```

NAME	HOST/PORT	PATH	SERVICES	PORT	TERMINATION	WILDCARD
ta-openapi-route	openapi.ta.apps.ocp46.tec.uk.ibm.com	/	ta-server	ta-server	reencrypt	None
ta-server-route	ta.apps.ocp46.tec.uk.ibm.com	/lands_advisor	ta-server	ta-server	reencrypt	None
ta-ui-route	ta.apps.ocp46.tec.uk.ibm.com		ta-ui	ta-ui	reencrypt/Redirect	None

Add the hostname ta.apps.ocp46.tec.uk.ibm.com to your local etc/hosts

Access TA via URL <https://ta.apps.ocp46.tec.uk.ibm.com>

## Install Mono2Micro UI into OCP

The M2M UI will be installed into the existing namespace wshe which was created for TA.

### Install M2M UI:

```
oc project wshe
oc new-app ibmcom/mono2micro-ui --name m2m-ui -e LICENSE=accept
oc expose service/m2m-ui
oc get route
```

```
[root@bastion ~]# oc get route
```

NAME	HOST/PORT	PATH	SERVICES	PORT
m2m-ui	m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com		m2m-ui	3000-tcp

### Access M2M UI:

To access the Mono2Micro UI, add m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com to your local /etc/hosts

<http://m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com/>

## IBM Container Licensing Service

It is already installed as part of WSA and Common Services.

Licensing service is installed in the namespace: ibm-common-services

### Get details for Licensing Service

```
$ oc get pods -n ibm-common-services | grep licensing
ibm-licensing-operator-6cd489cc78-gngl4      1/1      Running      21      12d
ibm-licensing-service-instance-75574f5544-v8wvg  1/1      Running      0       8d
```

### Check logs:

```
podName=`kubectl get pod -n ibm-common-services -o jsonpath="{range
.items[*]}{.metadata.name}{'\n'}" | grep ibm-licensing-service-instance`
kubectl logs $podName -n ibm-common-services
```

### Get route and password for Licensing Service

```
oc get route -n ibm-common-services | grep licensing
[root@bastion ~]# oc get route -n ibm-common-services | grep licensing
ibm-licensing-service-instance  ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com
```

Route:

<https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com>

Retrieve the API key: (<https://www.ibm.com/docs/en/cpfs?topic=service-obtaining-updating-api-token>)

```
$ kubectl get secret ibm-licensing-token -o jsonpath={.data.token} -n ibm-common-
services | base64 -d
tc4yVHd1MXyDGT48qVnX60gt
```

### Deploy annotated containers with Liberty

```
oc new-project was
# Deploy Open Liberty container with 4 replicas and CPU limited to 1500m
cat <<EOF | oc apply -f -
apiVersion: apps/v1
kind: Deployment
metadata:
  name: olp-wshe
spec:
  selector:
    matchLabels:
      app: olp-wshe
  replicas: 4
  template:
    metadata:
      labels:
        app: olp-wshe
    annotations:
      cloudpakName: "IBM WebSphere Hybrid Edition"
      cloudpakId: "6358611af04743f99f42dadcd6e39d52"
      productCloudpakRatio: "4:1"
      productId: "e7daacc46bbe4e2dacd2af49145a4723"
      productName: "IBM WebSphere Application Server"
      productMetric: "VIRTUAL_PROCESSOR_CORE"
      productChargedContainers: "All"
  spec:
    containers:
      - name: olp
        image: open-liberty
        ports:
          - containerPort: 9080
        resources:
          limits:
            cpu: 1500m
            memory: 512Mi
          requests:
            cpu: 100m
```

memory: 256Mi

EOF

oc get pods

NAME	READY	STATUS	RESTARTS	AGE
olp-wshe-7d6bc45459-7c665	1/1	Running	0	12h
olp-wshe-7d6bc45459-pvk6z	1/1	Running	0	12h
olp-wshe-7d6bc45459-tzmvj	1/1	Running	0	12h
olp-wshe-7d6bc45459-vpc4r	1/1	Running	0	12h

This results into  $4 \times 1500\text{m} = 6000\text{m} = 6 \text{ Cores}$

## Access Licensing Service

Add `ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com` to `/etc/hosts`

URL: <https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com>

## License Service - Products:

<https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/products?token=tc4yVHdlMXyDGT48qVnX60gt>

▼ 0:

```
name: "IBM WebSphere Hybrid Edition"
id: "6358611af04743f99f42dadcd6e39d52"
metricPeakDate: "2022-01-27"
metricName: "VIRTUAL_PROCESSOR_CORE"
metricQuantity: 2
```

## License Service – Bundled Products:

[https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/bundled\\_products?token=tc4yVHdlMXyDGT48qVnX60gt](https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/bundled_products?token=tc4yVHdlMXyDGT48qVnX60gt)

▼ 0:

```
productName: "IBM WebSphere Application Server"
productId: "e7daacc46bbe4e2dadcd2af49145a4723"
cloudpakId: "6358611af04743f99f42dadcd6e39d52"
cloudpakMetricName: "VIRTUAL_PROCESSOR_CORE"
metricName: "VIRTUAL_PROCESSOR_CORE"
metricPeakDate: "2022-01-27"
metricMeasuredQuantity: 6
metricConversion: "4:1"
metricConvertedQuantity: 2
```

## Overview about the endpoints

WSA

<https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com>  
admin  
JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW

Instana

<https://instanabackend.tec.uk.ibm.com>  
E-Mail: [admin@instana.local](mailto:admin@instana.local)  
Password: CvmWEY7Fy0

Transformation Advisor on OpenShift

<https://ta.apps.ocp46.tec.uk.ibm.com>

Mono2Micro UI on OpenShift

<http://m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com/>

License Service on OpenShift

<https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/tc4yVHdlMXyDGT48qVnX60gt>

WAS on RHEL7WAS2

<http://rhel7was2:9060/ibm/console>

DayTrader

<http://rhel7was2:9081/daytrader>  
<http://rhel7was2:9082/daytrader>  
[ibmdemo@RHEL7WAS1 assets]\$ sudo vi /etc/hosts  
[sudo] password for ibmdemo:  
[ibmdemo@RHEL7WAS1 assets]\$ cat endpoints.txt

WSA

<https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com>  
admin  
JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW

Instana

<https://instanabackend.tec.uk.ibm.com>  
E-Mail: [admin@instana.local](mailto:admin@instana.local)  
Password: CvmWEY7Fy0

Transformation Advisor on OpenShift

<https://ta.apps.ocp46.tec.uk.ibm.com>

Mono2Micro UI on OpenShift

<http://m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com/>

License Service on OpenShift

<https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/tc4yVHdlMXyDGT48qVnX60gt>

WAS on RHEL7WAS2

<http://rhel7was2:9060/ibm/console>

DayTrader

<http://rhel7was2:9081/daytrader>  
<http://rhel7was2:9082/daytrader>

```
[ibmdemo@RHEL7WAS1 assets]$ [ibmdemo@RHEL7WAS1 assets]$ cat endpoints.html
<html>

<head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Endpoints</title>

</head>

<body bgcolor="#FFFFFF">

<H1>Demo-Endpoints</H1>
<H3>WebSphere Automation</H3>
<A HREF="https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com"> WebSphere
Automation Console </A>
-
Credentials: admin, JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW
<p>
<A HREF="https://ibm-licensing-service-instance-ibm-common-
services.apps.ocp46.tec.uk.ibm.com"> License Service </A>
-
Token: tc4yVHdlMXYDGT48qVnX60gt
<p>

<H3>Instana</H3>
<A HREF="https://instanabackend.tec.uk.ibm.com"> Instana Dashboard </A>
-
User: admin@instana.local
Password: CvmWEY7Fy0
<p>

<H3>WebSphere Hybrid Edition</H3>
<A HREF="https://ta.apps.ocp46.tec.uk.ibm.com">Transformation Advisor on OpenShift </A>
<p>
<A HREF="http://m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com">Mono2Micro on OpenShift </A>
<p>

<H3>WebSphere Application Server Cluster on RHEL7WAS2</H3>
<A HREF="http://rhel7was2:9060/ibm/console">WAS ND Admin Console</A>
-
User: wasadmin, Password: passw0rd
<p>
<A HREF="http://rhel7was2:9081/daytrader">WAS ND DayTrader - Instance 1</A>
<p>
<A HREF="http://rhel7was2:9082/daytrader">WAS ND DayTrader - Instance 2</A>
<p>

<H3>OpenShift Cluster</H3>
<A HREF="https://console-openshift-
console.apps.ocp46.tec.uk.ibm.com/dashboards">OpenShift Dashboard</A>
-
User: admin, Password: passw0rd
<p>
</body>

</html>
```

## Overview about the Demo scenarios

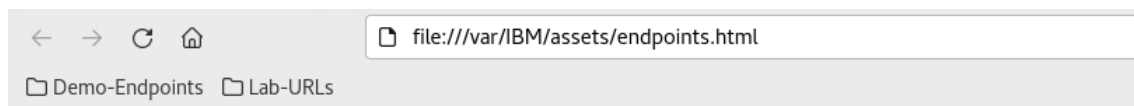
### The environment can be used to demonstrate the following tools and scenarios

- WebSphere Automation Security - Vulnerability tracking for tWAS and Liberty
- WebSphere Automation Health – Out of Memory Detection and Analysis
- IBM Cloud Transformation Advisor
- IBM Mono2Micro
- Instana Monitoring of WAS Traditional cell

### Installed software:

- OpenShift cluster version 4.6.54:
  - WebSphere Automation 1.2
  - Transformation Advisor 3.0
  - Mono2Micro UI 21.0.12.0
  - Open Liberty pods with 22.0.0.1
- Instanabackend
  - Instana self-hosted: 213-1 3.213.298-0 (6197a68574a46edccff1b5f63be7df5d7792fa7c)
- RHEL7WAS1:
  - WAS ND 8.5.5.18 plus iFixes IFPH42762 for Log4J
  - WLP 21.0.0.12
- RHEL7WAS2:
  - WAS ND 8.5.5.18 plus iFixes IFPH42762 for Log4J
  - WLP 21.0.0.12
  - DB2 container on docker (ibmcom/db2)

**Hints:** Open in the browser the homepage to get to the page with the Demo-Endpoints



## Demo-Endpoints

### WebSphere Automation

[WebSphere Automation Console](#) - Credentials: admin, JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW

[License Service](#) - Token: tc4yVHdlMXyDGT48qVnX60gt

### Instana

[Instana Dashboard](#) - User: admin@instana.local Password: CvmWEY7Fy0

### WebSphere Hybrid Edition

[Transformation Advisor on OpenShift](#)

[Mono2Micro on OpenShift](#)

### WebSphere Application Server Cluster on RHEL7WAS2

[WAS ND Admin Console](#) - User: wasadmin, Password: passw0rd

[WAS ND DayTrader - Instance 1](#)

[WAS ND DayTrader - Instance 2](#)

## Demo WebSphere Automation

### Demo WSA Security with Liberty

#### What to show:

- Demonstrate WSA vulnerability tracking based on the configured Liberty features.

#### Preparation:

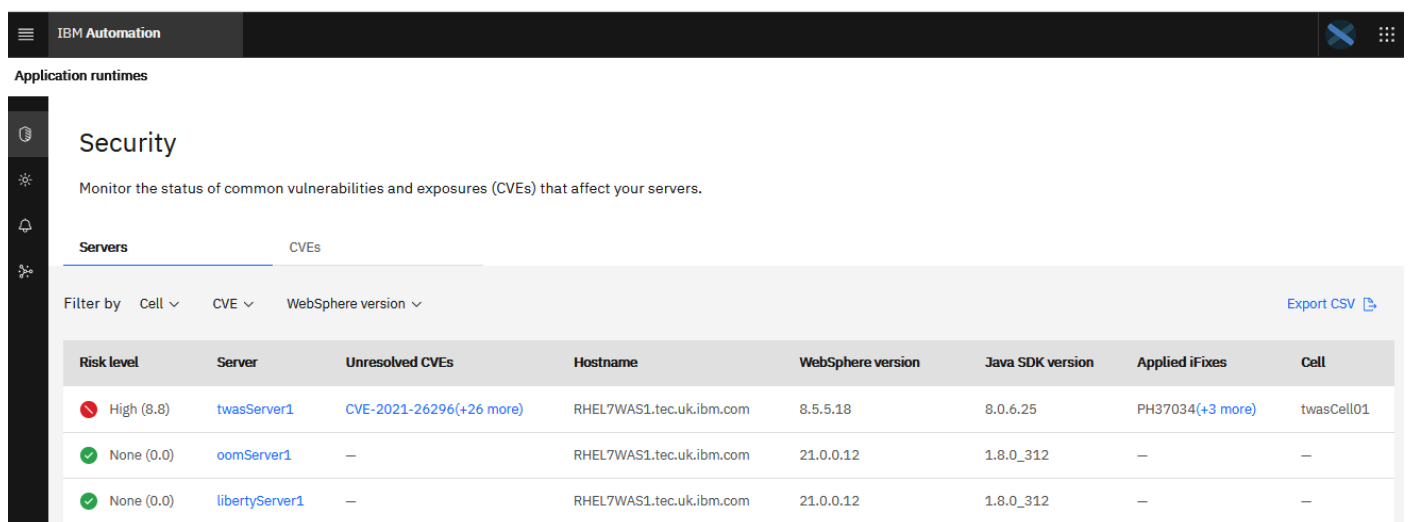
- Access the host RHEL7WAS1 as user: ibmdemo (password: passw0rd)
- Open a browser window and access the WSA dashboard
- Open a command Terminal on rhel7was1
- Start the Liberty instance via `/usr/IBM/Liberty/wlp/bin/server start libertyServer1`

#### Comments:

- As new vulnerabilities might come up, there is the chance that the initial Liberty configuration is not on risk level 0. In that case, you might have to adjust your story a bit.

#### Demo-Flow:

##### Step 1: Show in the WSA dashboard, that the Liberty instance libertyServer1 has no risk



The screenshot shows the IBM Automation WSA Security dashboard. The 'Security' section is active, displaying a table of servers and their vulnerability status. The table has columns for Risk level, Server, Unresolved CVEs, Hostname, WebSphere version, Java SDK version, Applied iFixes, and Cell. Three servers are listed: twasServer1 (High risk, 8.8), oomServer1 (None risk, 0.0), and libertyServer1 (None risk, 0.0).

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.8)	twasServer1	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	twasCell01
None (0.0)	oomServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

##### Step 2: Demonstrate WSA vulnerability tracking based on the configured Liberty features.

To show this, add the jaxws-2.2 feature

```
vi /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/server.xml
```

```
<feature>jaxws-2.2</feature>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

  <!-- Enable features -->
  <featureManager>
    <feature>jsp-2.3</feature>
    <feature>jaxws-2.2</feature>

  <!-- Enable features - Demo
  <feature>jaxws-2.2</feature>
  Enable features -->
```

##### Step 3: Switch to the WSA security panel to see that the security risk has changed to 4.8

IBM Automation

Application runtimes

Security

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Servers

CVEs

Filter by

Cell

CVE

WebSphere version

Export CSV

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
<div></div> <div>Medium (4.8)</div>	libertyServer1	CVE-2022-22310	RHEL7WAS1	21.0.0.12	1.8.0_312	—	—

## Step 4: Click on the CVE link to show the related vulnerability

<https://www.ibm.com/support/pages/node/6541530>

Security Bulletin: IBM WebSphere Application Server Liberty is vulnerable to an Information Disclosure (CVE-2022-22310)

### Security Bulletin

#### Summary

IBM WebSphere Application Server Liberty is vulnerable to an Information Disclosure. This has been addressed.

#### Vulnerability Details

CVEID: [CVE-2022-22310](#)

**DESCRIPTION:** IBM WebSphere Application Server Liberty could provide weaker than expected security. A remote attacker could exploit this weakness to obtain sensitive information and gain unauthorized access to JAX-WS applications.

CVSS Base score: 4.8

CVSS Temporal Score: See: <https://exchange.xforce.ibmcloud.com/vulnerabilities/217224> for the current score.

CVSS Vector: (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N)

#### Affected Products and Versions

Affected Product(s)	Version(s)
WebSphere Application Server Liberty	21.0.0.10 - 21.0.0.12

#### Remediation/Fixes

The recommended solution is to apply the interim fix or Fix Pack containing APAR for each named product as soon as practical.

For WebSphere Application Server Liberty 21.0.0.10 - 21.0.0.12 using the jaxws-2.2 feature:

## Step 5: Apply the fix by executing the script /usr/IBM/scripts/wlp\_applyFix.sh

```

Select ibmdemo@RHEL7WAS1:~
[ibmdemo@RHEL7WAS1 ~]$ /usr/IBM/scripts/wlp_applyFix.sh
Stopping server libertyServer1.
Server libertyServer1 is not running.
Apply iFix
Successfully extracted all product files.
Starting server libertyServer1.
Server libertyServer1 started with process ID 10488.
```

## Step 6: Switch to the WSA security panel to see that the security risk has changed to 0

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
<div></div> <div>None (0.0)</div>	libertyServer1	—	RHEL7WAS1	21.0.0.12	1.8.0_312	PH42074	—

## Step 7: Remove the fix by executing the script /usr/IBM/scripts/wlp\_removeFix.sh




```
[ibmdemo@RHEL7WAS1 ~]$ /usr/IBM/scripts/wlp_removeFix.sh
Stopping all servers

Stopping server libertyServer1.
Server libertyServer1 stopped.
Remove iFix
Starting all servers

Starting server libertyServer1.
Server libertyServer1 started with process ID 7057.
```

## Step 8: Switch to the WSA security panel to see that the security risk has changed back to 4.8


Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
 Medium (4.8)	libertyServer1	<a href="#">CVE-2022-22310</a>	RHEL7WAS1	21.0.0.12	1.8.0_312	—	—

## Step 9: Remove jaxws-feature from server.xml

```
vi /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/server.xml
```

```
<!-- Enable features -->
<featureManager>
  <feature>jsp-2.3</feature>
</featureManager>
```

## Step 10: Switch to the WSA security panel to see that the security risk has changed to 0

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
 None (0.0)	libertyServer1	—	RHEL7WAS1	21.0.0.12	1.8.0_312	—	—

## Step 11: Stop the Liberty instance.

```
/usr/IBM/Liberty/wlp/bin/server stop libertyServer1
```

## Related scripts:

### Script: /usr/IBM/scripts/wlp\_applyFix.sh

```
export WLP_DIR=/usr/IBM/Liberty
cd /usr/IBM/scripts
echo "Stop all Liberty Instances"
./wlp_stop.sh all
echo "Apply iFix 200012-wlp-archive-ifph36923.jar to resolve CVE-2021-26296"
java -jar /var/IBM/software/WAS/210012-extended-archive-ifph42074.jar --installLocation
$WLP_DIR/wlp --suppressInfo
echo "Start Liberty Instances"
./wlp_start.sh all
```

### Script: /usr/IBM/scripts/wlp\_removeFix.sh

```
cd /usr/IBM/scripts
export WLP_DIR=/usr/IBM/Liberty
./wlp_stop.sh all
echo "Remove iFix"
rm $WLP_DIR/wlp/lib/com.ibm.ws.jaxws.common_1.0.59.cl211220211208-1644.jar
rm $WLP_DIR/wlp/lib/fixes/210012-extended-archive-IFPH42074_21.0.0012.20220115_0043.xml
```

```
rm $WLP_DIR/wlp/lib/fixes/210012-extended-archive-  
IFPH42074_21.0.0012.20220115_0043.lpmf  
./wlp_start.sh all
```

## Demo WSA Security with traditional WAS

### What to show:

- Demonstrate WSA vulnerability tracking for traditional WAS.

### Preparation:

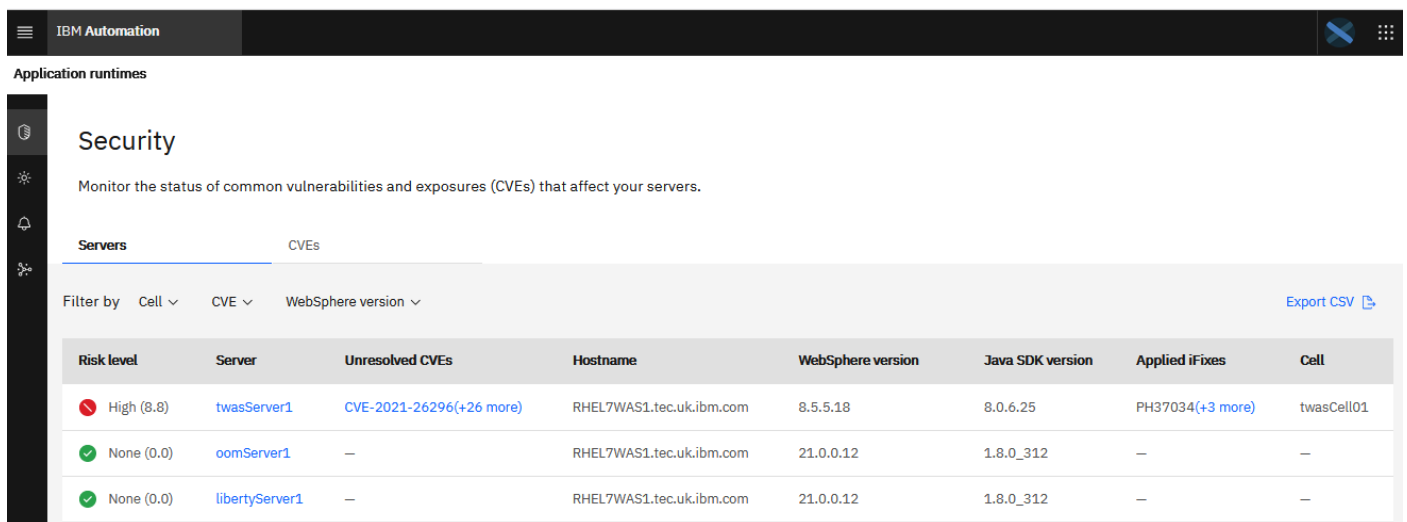
- Access the host RHEL7WAS1 as user: ibmdemo (password: passw0rd)
- Open a browser window and access the WSA dashboard
- Open a command Terminal on rhel7was1
- Start the tWAS instance via `/usr/IBM/scripts/twas_Start.sh`

### Comments:

- As new vulnerabilities might come up, there is the chance that the tWAS instance is not on the shown risk level but maybe higher. In that case, you might have to adjust your story a bit.

### Demo-Flow:

#### Step 1: Show in the WSA dashboard, that the tWAS instance has risk level 8.8



Application runtimes

**Security**

Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

**Servers** | CVEs

Filter by: Cell | CVE | WebSphere version | [Export CSV](#)

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.8)	twasServer1	CVE-2021-26296(+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034(+3 more)	twasCell01
None (0.0)	oomServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—
None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

#### Step 2: Navigate through the WSA dashboard and how unresolved CVEs, open a CVE, etc.

#### Step 3: Apply fixes to tWAS

Apply a fix to traditional WAS to reduce the risk level

`/usr/IBM/scripts/was_applyFixes.sh`


#### Step 4: Show in the WSA dashboard, that the tWAS instance has now risk level 8.2

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
High (8.2)	twasServer1	CVE-2020-4949 (+25 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034 (+5 more)	twasCell01
None (0.0)	libertyServer1	—	RHEL7WAS1	21.0.0.12	1.8.0_312	—	—

#### Step 5: Remove the fixes from tWAS

`/usr/IBM/scripts/was_removeFixes.sh`

#### Step 6: Show in the WSA dashboard, that the tWAS instance is back on risk level 8.8

Risk level	Server	Unresolved CVEs	Hostname	WebSphere version	Java SDK version	Applied iFixes	Cell
 High (8.8)	twasServer1	CVE-2021-26296 (+26 more)	RHEL7WAS1.tec.uk.ibm.com	8.5.5.18	8.0.6.25	PH37034 (+3 more)	twasCell01
 None (0.0)	libertyServer1	—	RHEL7WAS1.tec.uk.ibm.com	21.0.0.12	1.8.0_312	—	—

## Step 7: Stop the tWAS instance via /usr/IBM/scripts/twas\_Stop.sh

### Related scripts:

#### cat /usr/IBM/scripts/was\_Start.sh

```
export WAS855ND_HOME="/usr/IBM/WAS855ND"
$WAS855ND_HOME/bin/startServer.sh server1
```

#### cat /usr/IBM/scripts/was\_Stop.sh

```
export WAS855ND_HOME="/usr/IBM/WAS855ND"
$WAS855ND_HOME/bin/stopServer.sh server1
```

#### cat /usr/IBM/scripts/was\_applyFixes.sh

```
export fixID="8.5.5.5-WS-WAS-IFPH36923"
export fixRepo="8.5.5.5-ws-was-ifph36923.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSA"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
echo "Stop Server"
$WAS855ND_PROFILE/bin/stopServer.sh twasServer1
echo "Apply Fix $fixID"
$IMCL_HOME/imcl install $fixID -repositories /var/IBM/software/WAS/$fixRepo -
installationDirectory $WAS855ND_HOME -log /var/IBM/temp/$fixID.log
echo "Start Server"
$WAS855ND_PROFILE/bin/startServer.sh twasServer1
```

#### cat /usr/IBM/scripts/was\_removeFixes.sh

```
export fixID="8.5.5.5-WS-WAS-IFPH36923"
export fixRepo="8.5.5.5-ws-was-ifph36923.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSA"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
echo "Stop server"
$WAS855ND_PROFILE/bin/stopServer.sh twasServer1
echo "Remove Fix $fixID"
$IMCL_HOME/imcl uninstall $fixID -installationDirectory $WAS855ND_HOME -log
/var/IBM/temp/$fixID.log
echo "Start server"
$WAS855ND_PROFILE/bin/startServer.sh twasServer1
```

## Demo WSA Health with Liberty

### What to show:

- Demonstrate WSA vulnerability tracking based on the configured Liberty features.

### Preparation:

- Access the host RHEL7WAS1 as user: ibmdemo (password: passw0rd)
- Open a browser window and access the WSA dashboard
- Open a command Terminal on rhel7was1
- Make sure that the Instana agent is started via command  
`systemctl start instana-agent.service`
- Restart the Liberty instance via  
`/usr/IBM/Liberty2/wlp/bin/server stop oomServer1`  
`/usr/IBM/Liberty2/wlp/bin/server start oomServer1`

### Demo-Flow:

#### Step 1: Start the warmup script on the system running the oomServer1

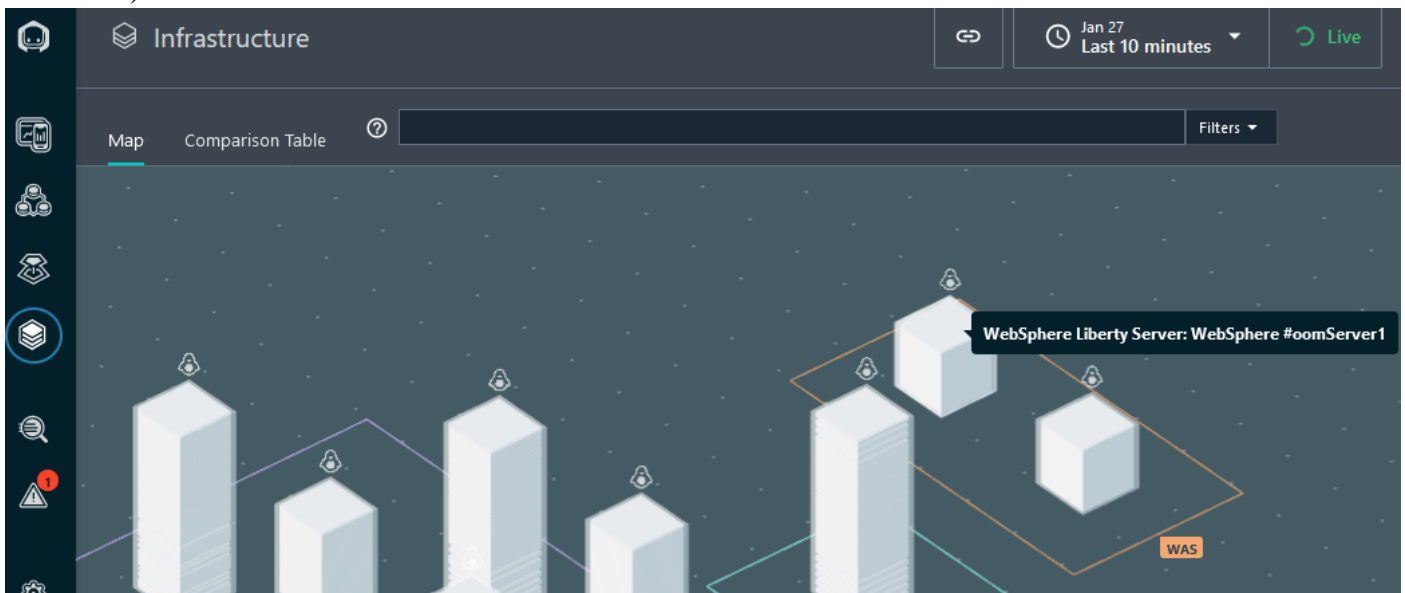
`/usr/IBM/scripts/wsaHealth_loadHeap.sh`

#### Step 2: Switch to the Instana dashboard and show Liberty instance


Click on Infrastructure, identify the Liberty instance on

Identify oomServer1 in the Instana dashboard and show heap consumption.

(If WebSphere Liberty is not displayed but only JVM, you have to wait until Instana has finished the detection.)



#### Step 3: Click on the server to see metrics

 WebSphere #oomServer1  
WebSphere Liberty Server

Open Dashboard

✓ No Issues

WebSphere Liberty Server Info

Name

oomServer1

Version

21.0.0.12

Install Dir

/usr/IBM/Liberty2/wlp/

Ports

Application States

Services

**Step 4:** Click on **Open Dashboard** to see the Liberty metrics, then select **Stack**

WebSphere #oomServer1 WebSphere Liberty Server

✓ No Issues

Stack

Upstream / Downstream

WebSphere Liberty Server

Name

oomServer1

Version

21.0.0.12

Install Dir

/usr/IBM/Liberty2/wlp/

Ports

Application States

Services

Application

Kubernetes

Infrastructure

Runs within 1 Custom Zone

WAS

Runs on 1 Host

RHEL7WAS1.tec.uk.ibm.com

CPU Used  
1%

Memory Used  
23%

Runs 1 Process

/usr/IBM/.../java ...

CPU Usage (User)  
1%

Virtual Memory  
2.81 GiB

Runs 1 JVM

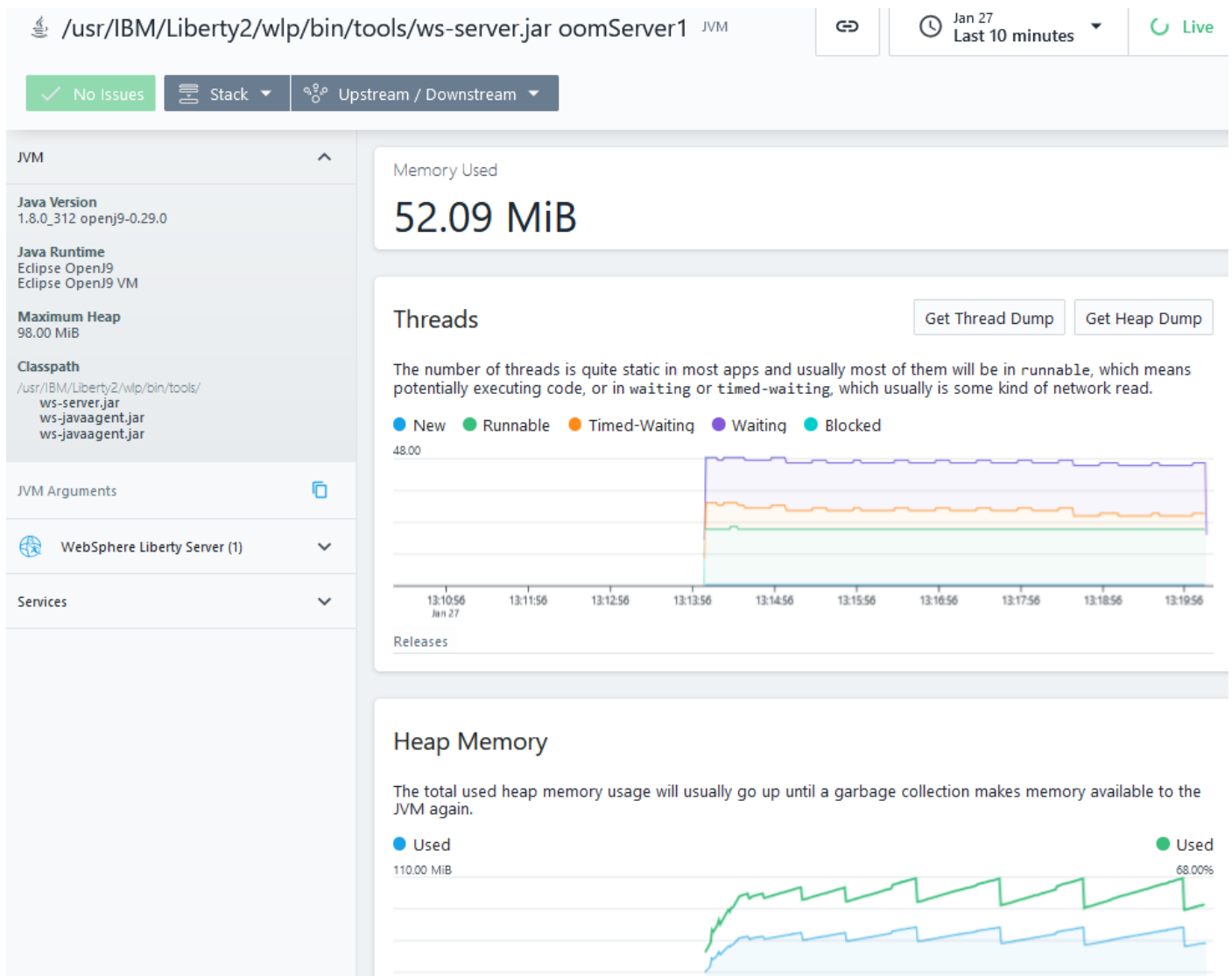
/usr/IBM/Liberty2/wlp/bin/tools/ws-server.jar oomServe...

Memory Used  
51.53 MiB

Blocked Threads  
0.00

**Step 5:** Explain the stack, then click on the **JVM**

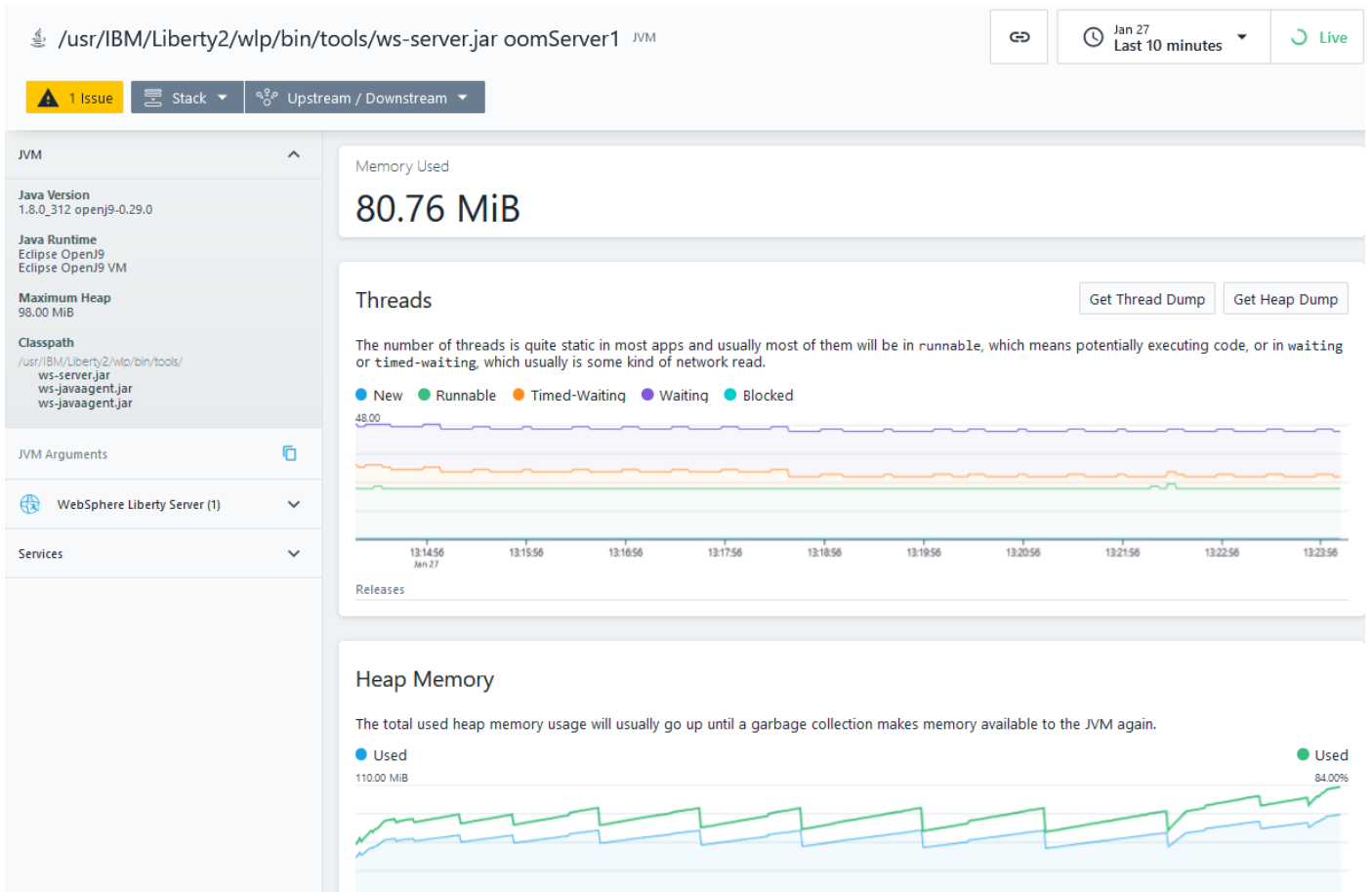
**Step 6:** Show the memory usage and the **Heap Memory** diagram



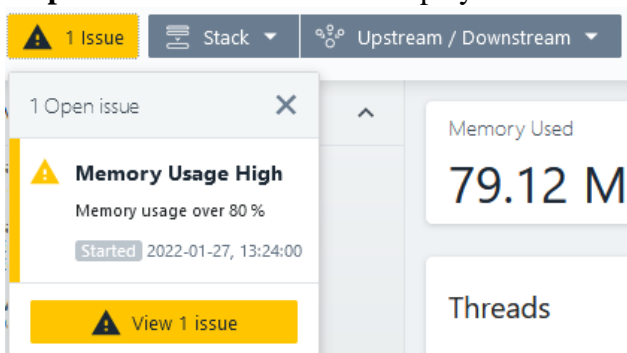
### Step 7: Overload the heap to trigger an event

Run the script `/usr/IBM/scripts/wsaHealth_overLoad.sh`

**Step 8:** Switch back to the Instana dashboard and wait for the alert.

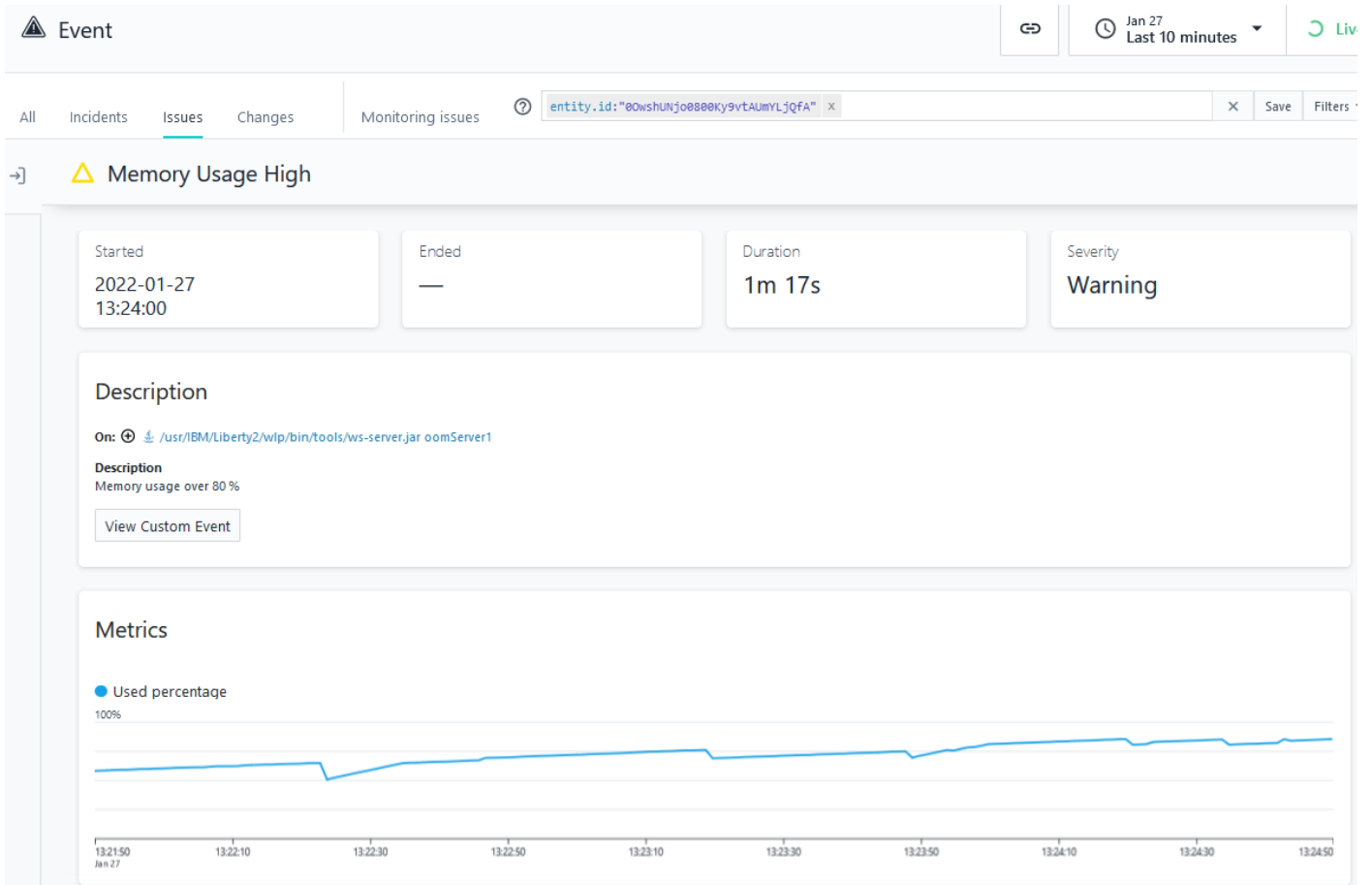


**Step 9:** Click on the **Issue** to display details



**Step 10:** Click on **View 1 issue** to see more details





**Step 12:** Switch in the **WebSphere Automation** dashboard to the **Health** panel  
Show that an investigation has been started.

IBM Automation

Application runtimes

### Health

View and analyze server health problems.

Investigation number	Status	Server	Hostname	Cell	Investigation type	Time detected
6	Investigating	Not identified yet	RHEL7WAS1.tec.uk.ibm.com	—	Memory leak	01/27 13:43
5	Investigation complete	oomServer1	RHEL7WAS1.tec.uk.ibm.com	—	Memory leak	01/27 13:24

**Step 13:** Show the **Investigation details**

The investigation could take several minutes, so you can either wait or use an already completed investigation. Click on the Investigation to display the Analysis summary.

IBM Automation

Application runtimes

Health

View and analyze server health problems.

Investigation number	Status	Server	Hostname	Cell
6	Investigation complete	oomServer1	RHEL7WAS1.tec.uk.ibm.com	—
5	Investigation complete	oomServer1	RHEL7WAS1.tec.uk.ibm.com	—
4	Investigation complete	oomServer1	RHEL7WAS1.tec.uk.ibm.com	—

Investigation number 6

Detected 1/27/2022 1:43:15 PM

Investigation ID

e40a9931-dc9a-4f6f-9458-16ec255e16aa

Problem type

Memory leak

Status

Investigation complete

Server

oomServer1

Host

RHEL7WAS1.tec.uk.ibm.com

Why investigation started

An alert was received that indicated a potential memory leak.

[View in Instana](#)

Analysis summary

- 'com.ibm.ws.webcontainer.osgi.servlet.ServletWrapper @ 0xfac3bcd8' (in '1' instances) retains '73.85' percent ( '50.00 MB' ) of live heap. This accumulates into 'java.lang.Object[80] @ 0xfa325b58' which is referenced directly, or through a chain of references, by 'java.util.Vector @ 0xfac39888'. 'java.util.Vector @ 0xfac39888' is referenced directly by 'servlets.MLVectorParam @ 0xfac39638'.

Recommended next step

Download all files and provide them to the application owner to resolve the issue.

[Download all files](#)

View history

Explain the findings of the analysis.

**Step 14:** Open the **View history** section to see the steps that have been performed.

IBM Automation

Application runtimes

Health

View and analyze server health problems.

Investigation number	Status	Server	Hostname	Cell
6	Investigation complete	oomServer1	RHEL7WAS1.tec.uk.ibm.com	—
5	Investigation complete	oomServer1	RHEL7WAS1.tec.uk.ibm.com	—
4	Investigation complete	oomServer1	RHEL7WAS1.tec.uk.ibm.com	—

View history

1/27/2022 1:45:18 PM

Investigation complete

1/27/2022 1:44:24 PM

Duration: 54 sec

Analyzing heap dump

'com.ibm.ws.webcontainer.osgi.servlet.ServletWrapper @ 0xfac3bcd8' (in '1' instances) retains '73.85' percent ( '50.00 MB' ) of live heap. This accumulates into 'java.lang.Object[80] @ 0xfa325b58' which is referenced directly, or through a chain of references, by 'java.util.Vector @ 0xfac39888'. 'java.util.Vector @ 0xfac39888' is referenced directly by 'servlets.MLVectorParam @ 0xfac39638'.

heapdump.20220127.124407.9418....\_Query.zip

Created 1/27/2022 1:45:17 PM

10 KB

1/27/2022 1:43:49 PM

Duration: 35 sec

Collecting heap dump

heapdump.20220127.124407.9418.0001.phd

Created 1/27/2022 1:44:14 PM

6 MB

1/27/2022 1:43:15 PM

Duration: 34 sec

Identifying server

1/27/2022 1:43:15 PM

Memory leak detected

An alert was received that indicated a potential memory leak.

## Demo Transformation Advisor

Demo-Collections are available in /var/IBM/assets/TA

- AppSrv1\_collection.zip is a good collection to explain the basic concepts of TA
- ACME.zip is a good collection to explain the common code feature of TA

Demonstrate the basic concepts of TA via the collection AppSrv1\_collection.zip:

### What to show:

- Demonstrate the basic capabilities of Transformation Advisor

### Preparation:

- Access the host RHEL7WAS1 as user: ibmdemo (password: passw0rd)
- Open a browser window and access the Transformation Advisor User Interface

### Demo-Flow:

**Step 1:** Create a new collection, show the option to download collectors and the commands behind it

**Step 2:** Upload the collection /var/IBM/assets/TA/AppSrv1\_collection.zip

**Step 3:** Navigate through the different applications and explain which is a good fit for Open Liberty

- Explain why WebSphereBank is complex
- Explain that modresorts can be deployed right-away
- Ignore CustomerOrderServicesApp at the moment

### Java

Collections	Total Applications	Workspace estimated total costs ①
Collections	4	Common Code 0 days
Migration target ①	Avg. cost per application 7.3	Unique app code 29.5 days
1 x Migration targets		Total cost 29.5 days

Java applications (4)					
Migration assessment					
Java applications					
Q Search					
Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ①
CustomerOrderServicesApp.ear.ear Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Moderate	▲ 4   ■ 1   ● 8	0	8
DayTrader3-EE6.ear Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Moderate	▲ 1   ● 4	0	0.5
modresorts-1_0_war.ear Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Simple	● 2	0	0
WebSphereBank.ear Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Complex	▲ 2   ● 7	0	21.5

**Step 4:** Switch the migration target to WLP and OLP and compare the results for WebSphere Liberty and Open Liberty

Collections

Collections

Total Applications

4

Workspace estimated total costs ⓘ

Open Liberty 29.5 days

WebSphere Liberty 22 days

Migration target ⓘ

2 × Migration targets

#### Java applications (4)

#### Migration assessment Java applications

Q Search

Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ⓘ
<a href="#">CustomerOrderServicesApp.ear.ear</a> Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Moderate	▲ 4 ■ 1 ● 8	0	8
<a href="#">CustomerOrderServicesApp.ear.ear</a> WebSphere Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Moderate	▲ 1 ● 4	0	0.5
<a href="#">DayTrader3-EE6.ear</a> Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Moderate	▲ 1 ● 4	0	0.5
<a href="#">DayTrader3-EE6.ear</a> WebSphere Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Moderate	▲ 1 ● 4	0	0.5
<a href="#">modresorts-1_0_war.ear</a> Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Simple	● 2	0	0
<a href="#">modresorts-1_0_war.ear</a> WebSphere Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Simple	● 2	0	0
<a href="#">WebSphereBank.ear</a> Open Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Complex	▲ 2 ● 7	0	21.5
<a href="#">WebSphereBank.ear</a> WebSphere Liberty	ibmdemo-virtual-machine / AppSrv1_collection	Complex	▲ 2 ● 7	0	21.5

**Step 5:** Investigate into CustomerOrderServicesApp to see why WebSphere Liberty is a better fit than Open Liberty

explain the migration challenges with Open Liberty via reports

- Click on the CustomerOrderServicesApp with target Open Liberty, look at the challenges and explain the reports
- Click on the CustomerOrderServicesApp with target WebSphere Liberty, look at the challenges and explain that there are no changes required for JPA and JAXRS which results into a significantly reduced migration effort.
- Show the migration bundle for CustomerOrderServicesApp, especially the server.xml

## Demonstrate the Common Code Feature of TA via the collection ACME.zip

### What to show:

- Demonstrate the new Common Code capability of Transformation Advisor

### Preparation:

- Access the host RHEL7WAS1 as user: ibmdemo (password: passw0rd)
- Open a browser window and access the Transformation Advisor User Interface

### Demo-Flow:

**Step 1:** Create a new workspace and upload the collection /var/IBM/assets/TA/ACME.zip  
(You can also use instead the existing workspace named ACME)

### Step 2:

 Explain the common code feature

Explain that each application EARs on its own requires more than 10 days of migration effort.  
But fortunately, the main effort is in common code which reduces the effort from 55.5 to 13 days.

Collections	Total Applications	Workspace estimated total costs ⓘ
Collections	5	Common Code 10.5 days
Migration target ⓘ	Avg. cost per application	Unique app code 2.5 days
1 × Migration targets	2.6	Total cost 13 days

Java applications (5)	Common code files (6)
-----------------------	-----------------------

Migration assessment					
Java applications					
Q Search					
Java application	Collection/ Profile	Complexity	Issues	Common code files	Application cost in days ⓘ
ACMEAnnuityEJBMDB.ear Open Liberty	acme.webserver.com / ACME	Moderate	▲ 7 ■ 1 ● 13	2	10
ACMEAnnuityEJBWSes.ear Open Liberty	acme.webserver.com / ACME	Moderate	▲ 9 ■ 1 ● 14	5	12
ACMEAnnuityJAXRSWSes.ear Open Liberty	acme.webserver.com / ACME	Moderate	▲ 8 ■ 1 ● 13	6	11.5
ACMEAnnuityPojoWSes.ear Open Liberty	acme.webserver.com / ACME	Moderate	▲ 7 ■ 1 ● 12	4	10
ACMEAnnuityWeb.ear Open Liberty	acme.webserver.com / ACME	Moderate	▲ 7 ■ 1 ● 17	5	10

### Step 3:

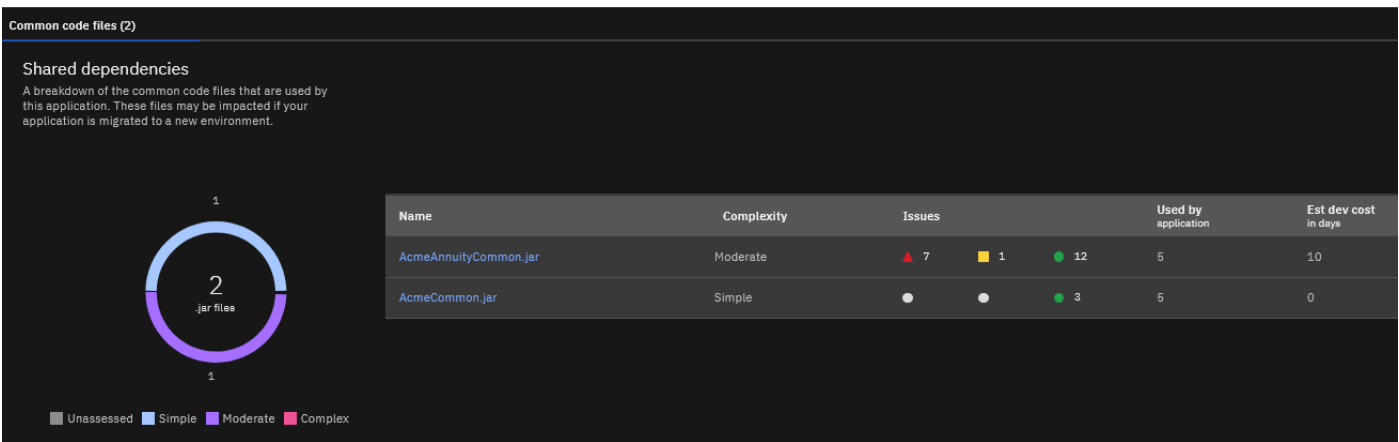
 Click on the application ACMEAnnuityEJBMDB.ear

Explain that the app requires 10 days of development and contains two jars.

As you can see, the ACMEAnnuityCommon.jar causes the main efforts.  
ACMEAnnuityEJBMDB.ear

[View migration plan](#)

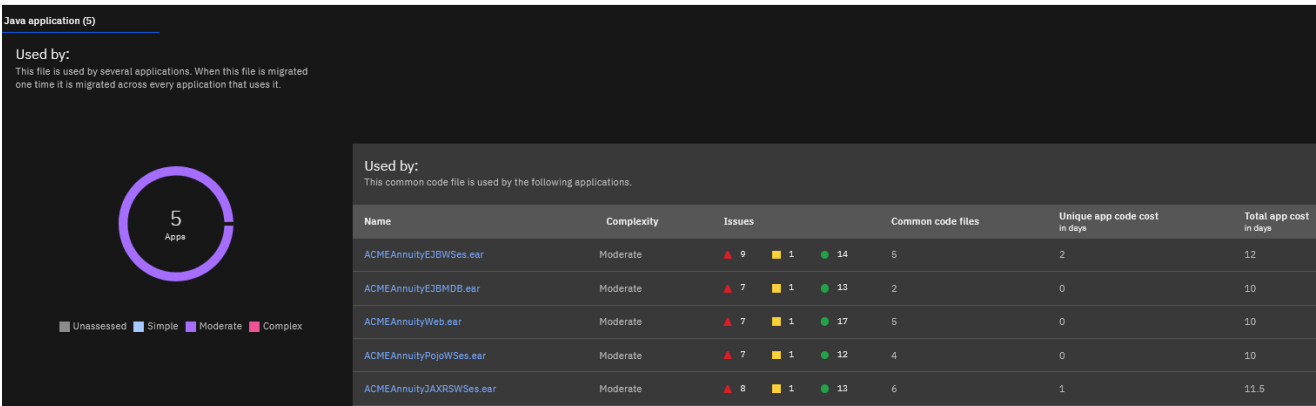
Source environment	Migration Target	Complexity	Issues ⓘ	Common code files	Unique code cost	Total application cost
IBM WebSphere Application Server Network Deployment ACME Version: 8.5.5.13	Open Liberty	Moderate	▲ 7   ■ 1   ● 13	2	0	10 days



**Step 4:** Look into ACMEAnnuityCommon.jar  
Click on the ACMEAnnuityCommon.jar to see where it is used.  
You can also see, how much additional effort is needed to migrate the apps using the common code.  
As you could see, if you migrate the common code, 3 of the 5 applications can be migrated without additional effort.

AcmeAnnuityCommon.jar

Migration Target	Shared library	Used By	Complexity	Issues ⓘ	Total application cost
Open Liberty	none ⓘ	5 applications	Moderate	▲ 7   ■ 1   ● 12	10 days



**Summary:**  
Transformation Advisor provides you with insight, how much development effort is estimated to migrate your applications. You gain this insight by just running the collector. By using the new common code feature, you can see easily how much effort is unique to one application and what has only to be migrated once. This gives you a realistic view on the real efforts that you have to expect.

## Demo Mono2Micro

### What to show:

- Demonstrate Mono2Micro UI

### Preparation:

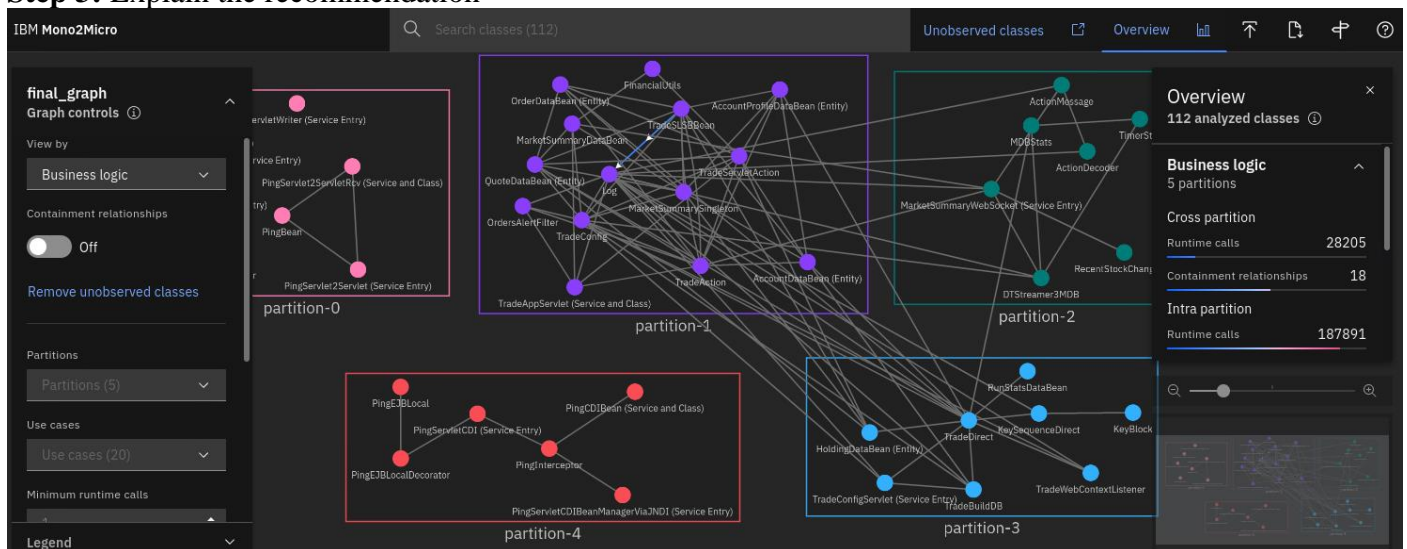
- Access the host RHEL7WAS1 as user: ibmdemo (password: passw0rd)
- Open a browser window
- Open a File Browser and navigate to /var/IBM/assets/M2M.

### Demo-Flow:

**Step 1:** Access the Mono2Micro UI via URL <http://m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com/>.

**Step 2:** Load the sample graph from /var/IBM/assets/M2M.

**Step 3:** Explain the recommendation



**Step 4:** Next to the graph, the directory also contains the related reports.

- Cardinal-Report.html
- Oriole-Report.html

The image also contains the CardinalFileSummary.txt with the list of generated classes. Finally, it also contains the source project Mono2Micro-Example\_20211216.zip

## Demo License Service

### What to show:

- Demonstrate the license service and how to use for Liberty containers

### Preparation:

- Access the host RHEL7WAS1 as user: ibmdemo (password: passw0rd)
- Open a browser window
- Have the License Service token by hand tc4yVHdIMXYDGT48qVnX60gt

### Demo-Flow:

#### Step 1: Explain the scenario

An Open Liberty pod with 4 replica and annotated for WebSphere Hybrid Edition has already been deployed. Each pod has a CPU limit of 1500m which results into  $4 \times 1500m = 6000m = 6$  Cores.

If you want, you can show the running pods via

```
oc get pods -n was
```

NAME	READY	STATUS	RESTARTS	AGE
olp-wshe-7d6bc45459-7c665	1/1	Running	0	12h
olp-wshe-7d6bc45459-pvk6z	1/1	Running	0	12h
olp-wshe-7d6bc45459-tzmvj	1/1	Running	0	12h
olp-wshe-7d6bc45459-vpc4r	1/1	Running	0	12h

#### Step 2: Access the **License Service** menu via URL

<https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com>

Explain the options

#### Step 3: Navigate to the **License Service - Products:**

<https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/products?token=tc4yVHdIMXYDGT48qVnX60gt>

▼ 0:

```
name:          "IBM WebSphere Hybrid Edition"
id:            "6358611af04743f99f42dadcd6e39d52"
metricPeakDate: "2022-01-27"
metricName:     "VIRTUAL_PROCESSOR_CORE"
metricQuantity: 2
```

To explain why the metricQuantity is 2, goto the next step:

#### Step 3: Navigate to the **License Service – Bundled Products:**

[https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/bundled\\_products?token=tc4yVHdIMXYDGT48qVnX60gt](https://ibm-licensing-service-instance-ibm-common-services.apps.ocp46.tec.uk.ibm.com/bundled_products?token=tc4yVHdIMXYDGT48qVnX60gt)



```
▼ 0:
  productName:      "IBM WebSphere Application Server"
  productId:        "e7daacc46bbe4e2dacd2af49145a4723"
  cloudpakId:       "6358611af04743f99f42dadcd6e39d52"
  cloudpakMetricName: "VIRTUAL_PROCESSOR_CORE"
  metricName:       "VIRTUAL_PROCESSOR_CORE"
  metricPeakDate:   "2022-01-27"
  metricMeasuredQuantity: 6
  metricConversion:  "4:1"
  metricConvertedQuantity: 2
```

**Step 4:** If the customer is interested in the yaml or you want to adjust the scenario:

/var/IBM/assets/LicenseService/ olp\_wshe.yaml

You can also deploy it to your own namespace or adjust it

## Demo DayTrader with Instana

### What to show:

- Demonstrate Instana to monitor WebSphere Traditional

### Preparation:

- Access the host RHEL7WAS2 as user: ibmdemo (password: passw0rd)
- Open a command Terminal on rhel7was2
- Start the environment (see below)
- Access the application via <http://rhel7was2:9081/daytrader>, switch to the tab Trading & Portfolio and log in, then go trading to verify the application works
- Run load scripts to produce some basic load (see below)
- Open a browser window and access the Instana dashboard and verify that you can see the DayTrader servers and successful requests come in (see steps below)

### Start the environment

Open a Terminal on rhel7was2 and do the following steps:

Make sure that the Instana agent is started via command

```
systemctl start instana-agent.service
```

Start the database via

```
docker start mytrade3db
```

Start the WAS ND cell

```
/usr/IBM/WAS855ND/profiles/Dmgr01/bin/startManager.sh  
/usr/IBM/WAS855ND/profiles/AppSrv1/bin/startNode.sh  
/usr/IBM/WAS855ND/profiles/AppSrv1/bin/startServer.sh TradeServer1  
/usr/IBM/WAS855ND/profiles/AppSrv1/bin/startServer.sh TradeServer2
```

### Run load scripts to produce some basic load

Open a Terminal on rhel7was2 and run load against both cluster members

```
/usr/IBM/apache-jmeter-4.0/bin/jmeter -n -t /usr/IBM/scripts/Apache_JMeter_script/daytrader3.jmx -  
JHOST=localhost -JPORT=9081 -JDURATION=3000 -JTHREADS=2&  
/usr/IBM/apache-jmeter-4.0/bin/jmeter -n -t /usr/IBM/scripts/Apache_JMeter_script/daytrader3.jmx -  
JHOST=localhost -JPORT=9082 -JDURATION=3000 -JTHREADS=2&
```

## Demo-Flow:

### Step 1: Access the Instana dashboard and show the WAS part of the environment

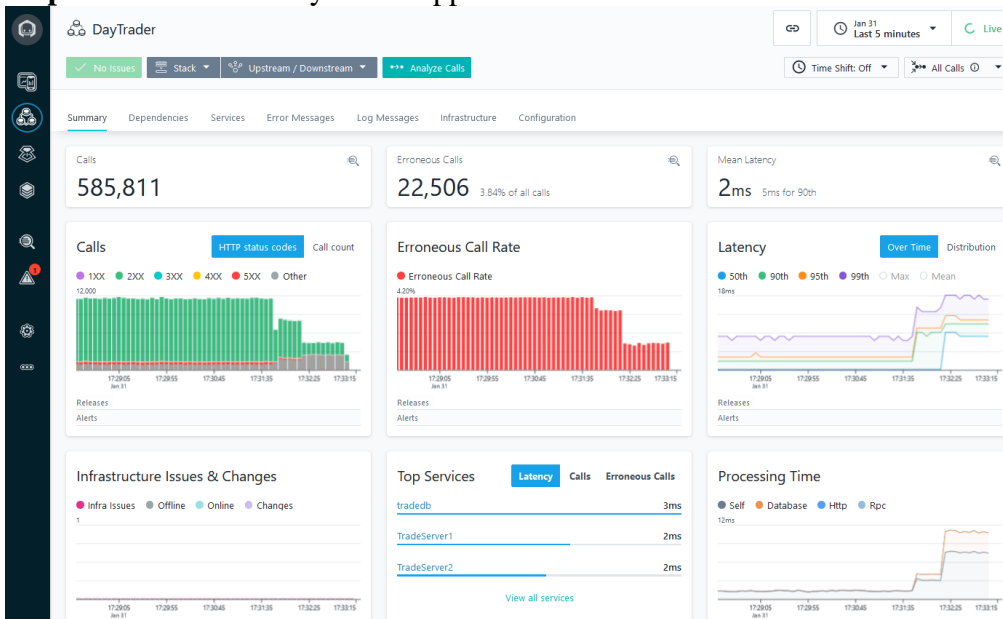
- 2 WebSpheres are the application servers DayTraderServer 1 and 2
- 2 JVMs are the Deployment Manager and the Node Agent
- 1 DB2 is the DayTrader database



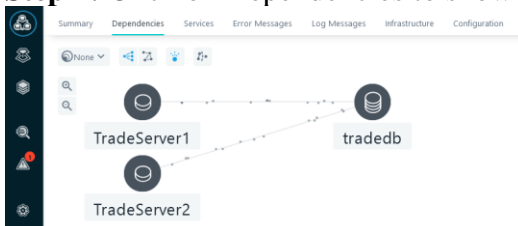
### Step 2: Switch to the applications view and explain the concept of an application perspective.

Name	Scope	Services	Calls ↓	Latency	Erroneous Call Rate	Health
DayTrader		3	2,197,472	9ms	3.13%	✓
All Services		39	1,773,654	29ms	3.91%	✓

### Step 3: Click on the DayTrader application and show the dashboard.



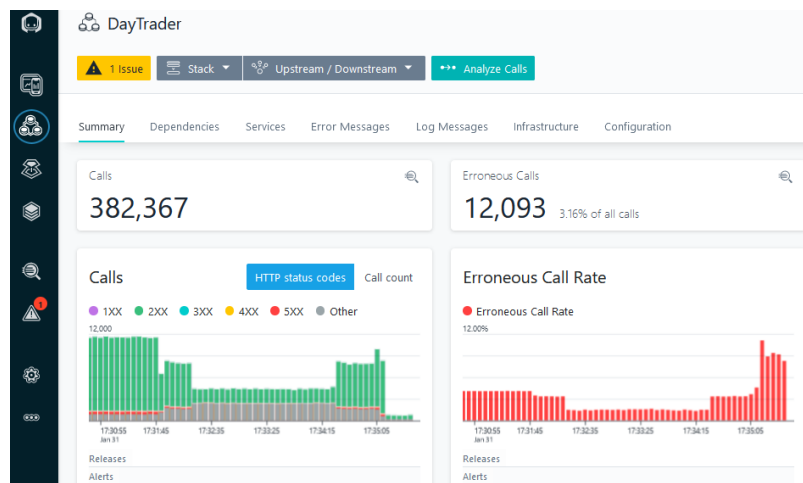
### Step 4: Click on Dependencies to show the application flow



### Step 5: Switch to a Terminal window and stop the database

```
docker stop mytrade3db
```

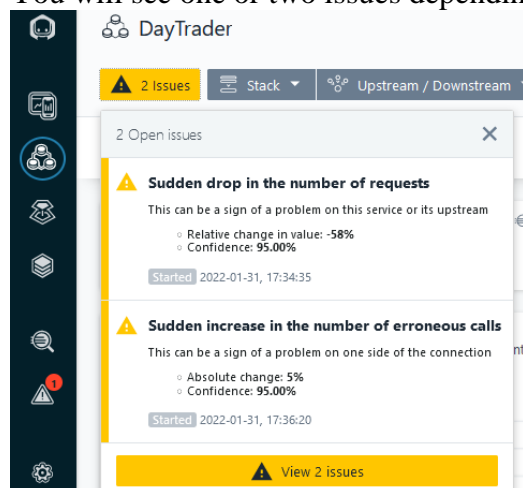
## Step 6: Switch to the Instana Dashboard



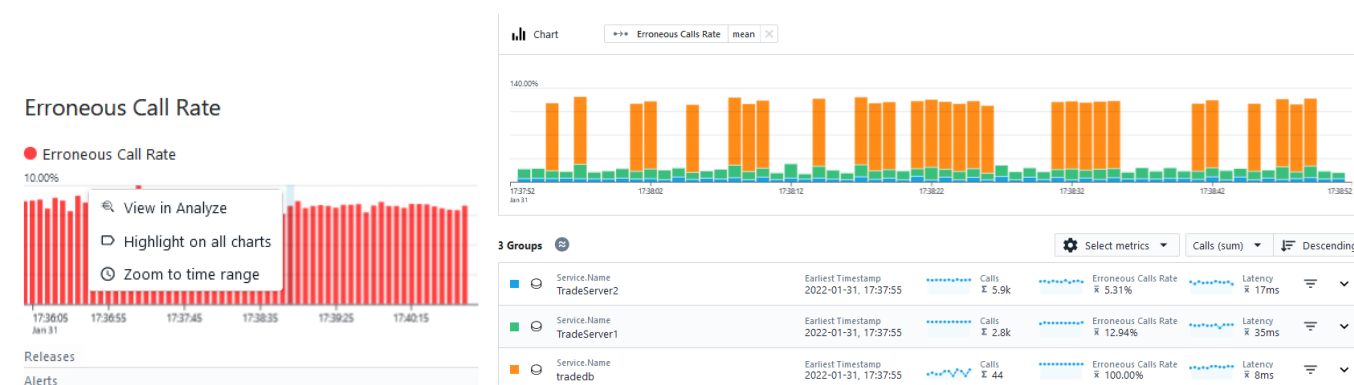
Explain that the number of Erroneous Calls has increased and that an issue has been detected.

## Step 7: Click on the issue

You will see one or two issues depending how long you wait:



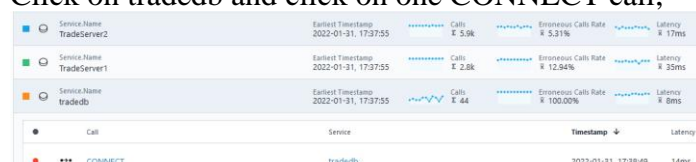
## Step 8: Click on the Erroneous Calls, then View in Analyze



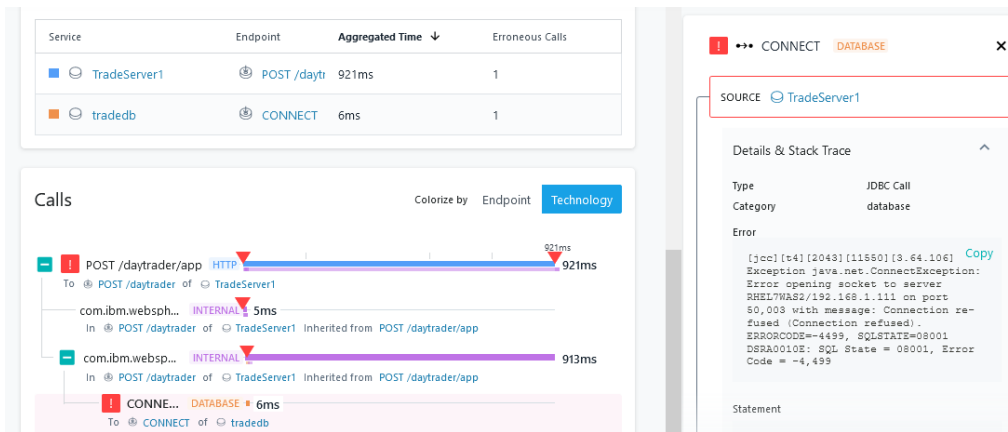
As you can see, there is an issue with the database.

## Step 9: Investigate into the error

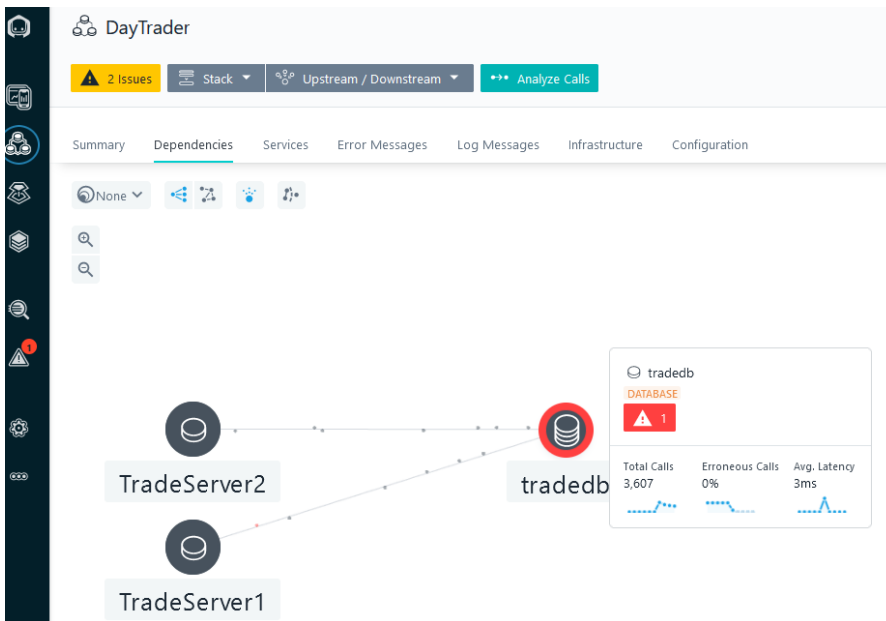
Click on tradedb and click on one CONNECT call,



You can see the flow and the exception thrown by the database



## Step 10: Show the error in the Dependencies view



## Step 11: Switch to a Terminal window and start the database

`docker start mytrade3db`

## Step 12: Go back to the application dashboard

The application should recover. You have to click on the Live view to see it.

