Strengthen Operational Security with IBM WebSphere Automation

Author(s): Tien-Thanh Le, Ajay Reddy, Brian Hanczaryk, Jagadeeswar R Gangaraju, Yee-Kang Chang, Rafael Osorio, Lars Besselmann

Last updated: January 2022

Duration: 45 mins

Contents

Note: To ease the copy and paste, the commands used in the lab have been stored into the file. Use the command

```
cat /usr/IBM/scripts/WSAcommands.txt
```

or an editor to have them available for copy and paste

Introduction to WebSphere Automation

IBM WebSphere Automation is focused on delivering value into existing WebSphere Application Server (WAS) environments, helping administrators reduce the cost, effort, and risk of addressing common vulnerabilities, automating tasks, and remediating capacity incidents.

It removes manual toil so that your team can spend more time innovating while minimizing the cost of extending the life and maximizing the ROI of your WebSphere investments.

In conversations with customers, the same three concerns come up repeatedly. Organizations need to keep their IT estate secure and compliant, resilient to disruption and running optimally while reducing costs and maximizing ROI.

WebSphere Automation helps organizations gain visibility, operational efficiencies, and cost savings quickly by extending the life of WebSphere investments and giving teams time back to focus on unlocking new value and fixing the imbalance of pure maintenance versus innovation tasks.

- WebSphere operators and administrators save time and embrace DevSecOps by implementing patches more efficiently on virtual and container environments to keep operations compliant and secure.
- Enhance remediation capabilities with insights and recommendations to improve the speed and depth of understanding of outages and anomalies as they occur.
- Augment the operational experience with access to simplified and consolidated information that enables teams to act.

With WebSphere Automation, security, business efficiency and resiliency become standard. IBM can meet you wherever you are in your optimization and automation journeys to help you quickly deliver value and increase ROI, all while laying a solid automation foundation to support future growth.

IBM WebSphere Automation is available as a stand-alone offering or as an addition to IBM Cloud Pak® for Watson AIOps. As part of IBM Automation platform, IBM WebSphere Automation includes containerized components and common software services on top of a common automation layer, to manage WebSphere's incidents, hybrid applications, and cost with complete observability, governance, and compliance.

Deploy virtually anywhere through containers supported by Red Hat® OpenShift® software, on IBM Cloud®, on essentially any existing infrastructure on-premises, or through private and public clouds. Use only the capabilities you need with a fully modular approach that's designed to be easy to consume.

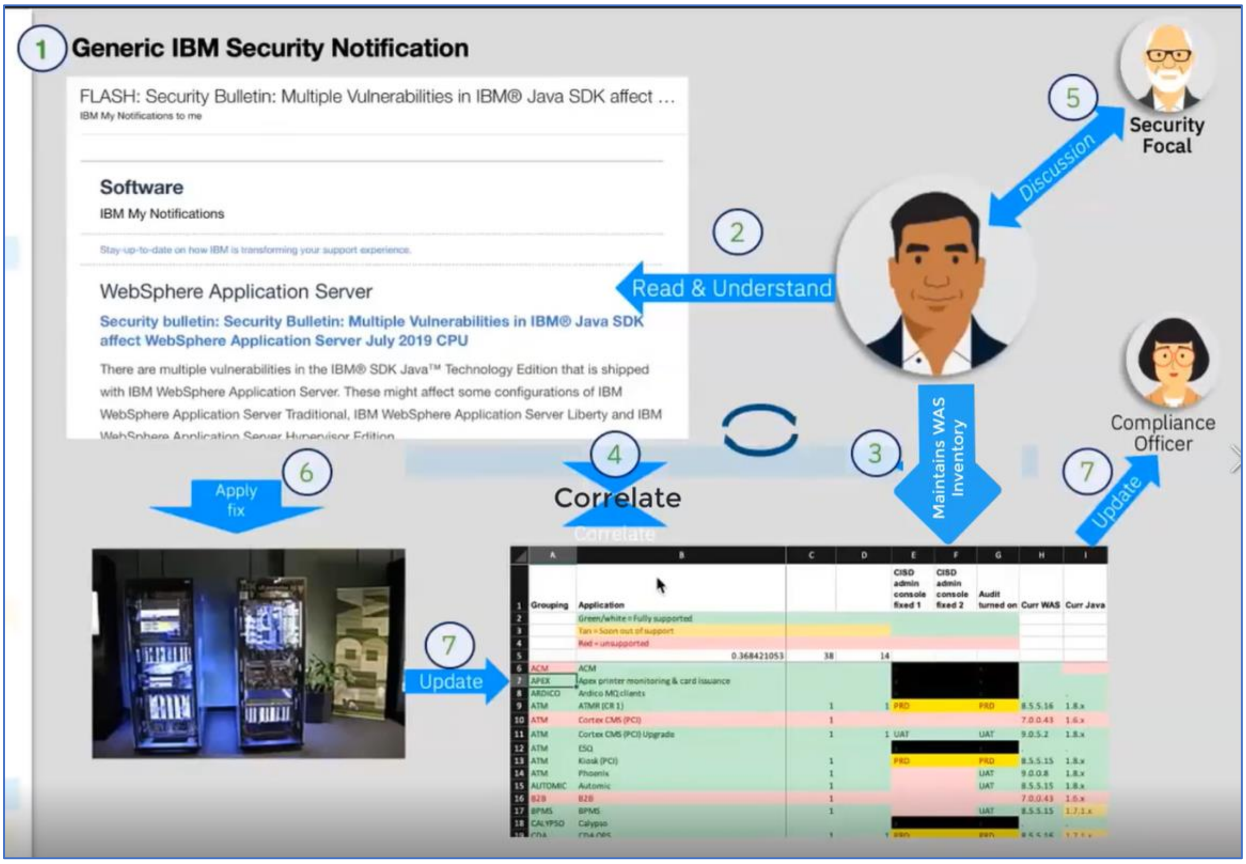
Business Context

You are a WebSphere Administrator, part of a WebSphere Operations Team responsible for maintaining security compliance of the WebSphere estate in the enterprise. A typical “as-is” process for maintaining security compliance for WebSphere environments is depicted below.

Today (as-is):

1. IBM sends generic “FLASH” to indicate a new WAS security bulletin.
2. You subscribe and receive IBM Security Bulletins to be aware about vulnerabilities, its potential impact, severity, and recommended solutions.
3. Generally, WAS inventory is maintained in spreadsheets.
4. Based on that, you check if this CVE applies to your managed inventory (Spreadsheet)
5. You determine if an APAR / Fix Pack upgrade should be applied to existing environment
6. You deploy the fix to the impacted environments
7. You update the WAS inventory (Spreadsheet) and provide up-to-date reports to audit and compliance teams

As is, your inventory is a spreadsheet, containing all information about your servers, such as the versions of the installed servers, which operating system they're installed on, and iFixes which have been applied, etc.



FLASH: Security Bulletin: Multiple Vulnerabilities in IBM® Java SDK affect ...
IBM My Notifications to me

Software
IBM My Notifications

Stay-up-to-date on how IBM is transforming your support experience.

WebSphere Application Server

Security bulletin: Security Bulletin: Multiple Vulnerabilities in IBM® Java SDK affect WebSphere Application Server July 2019 CPU

There are multiple vulnerabilities in the IBM® SDK Java™ Technology Edition that is shipped with IBM WebSphere Application Server. These might affect some configurations of IBM WebSphere Application Server Traditional, IBM WebSphere Application Server Liberty and IBM WebSphere Application Server Hypervisor Edition

	A	B	C	D	E	F	G	H	I
1	Grouping	Application			CISD admin console fixed 1	CISD admin console fixed 2	Audit turned on	Curr WAS	Curr Java
2		Green/white = Fully supported							
3		Tan = Soon out of support							
4		Red = unsupported							
5		0.368421053	38	14					
6	ACM	ACM			[redacted]	[redacted]	[redacted]		
7	APEX	Apex printer monitoring & card issuance			[redacted]	[redacted]	[redacted]		
8	ARDICO	Ardico MQ clients			[redacted]	[redacted]	[redacted]		
9	ATM	ATMR (CR 1)	1	1	PRD		PRD	8.5.5.16	1.8.x
10	ATM	Cortex CMS (PCI)	1					7.0.0.43	1.6.x
11	ATM	Cortex CMS (PCI) Upgrade	1	1	UAT		UAT	9.0.5.2	1.8.x
12	ATM	ESQ			[redacted]	[redacted]	[redacted]		
13	ATM	Kiosk (PCI)	1		PRD		PRD	8.5.5.15	1.8.x
14	ATM	Phoenix	1				UAT	9.0.0.8	1.8.x
15	AUTOMIC	Automic	1				UAT	8.5.5.15	1.8.x
16	B2B	B2B	1					7.0.0.43	1.6.x
17	BPMS	BPMS	1				UAT	8.5.5.15	1.7.1.x
18	CALYPSO	Calypso			[redacted]	[redacted]	[redacted]		
19	CDA	CDA OPS	1	1	PRD		PRD	8.5.5.16	1.7.1.x

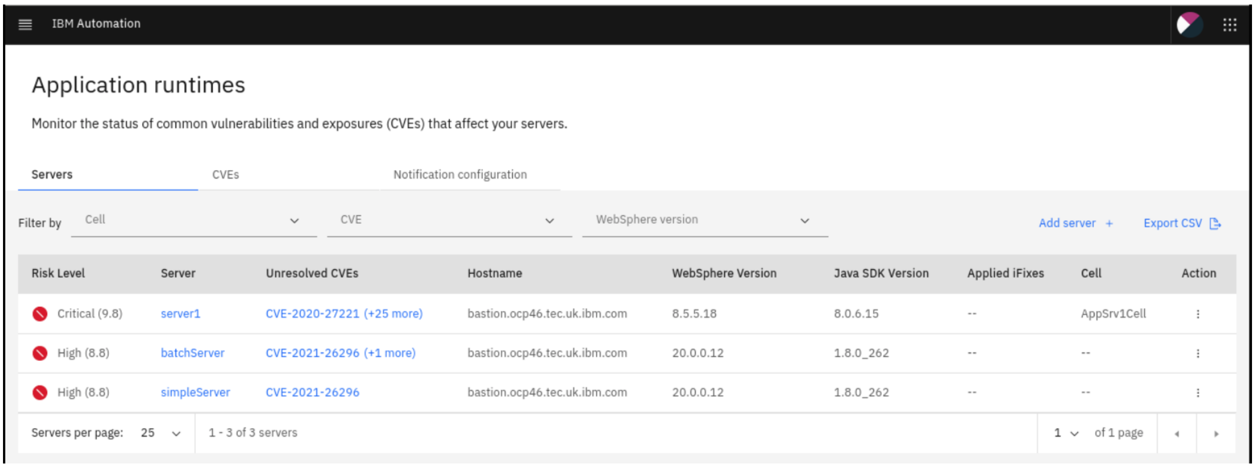
Currently, this is a very manual, time-consuming process, and you'd like to automate this process to direct valuable time and resource elsewhere. This is where **IBM WebSphere Automation will help!**

You would like to have:

- **Management dashboard:** Consolidated dashboard increases awareness and response time to common vulnerabilities and exposures (CVEs).
- **Automated vulnerability tracking:** Let WebSphere Automation track new security bulletins across your existing traditional WebSphere and Liberty environments, on virtual machines or containers.
- **Contextual notifications:** Receive security bulletin notifications only when new vulnerabilities affect the environment you manage, reducing noise and interruptions to the WebSphere operations team.
- **Shared, live visibility to key stakeholders:** WebSphere operators and security compliance teams can see the real-time security posture of the WebSphere estate, accelerating action and minimizing the risk of miscommunication.

In this lab, you use the IBM WebSphere Automation to secure operations to reduce risk and meet compliance.

At the end of this lab, you will be able to connect teams with the most relevant information through a single dashboard. This enables you to discover, analyze and remediate common vulnerabilities and exposures across instances. Furthermore, this information can be exported to a CSV file to be shared amongst the broader team.



Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes	Cell	Action
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	bastion.ocp46.tec.uk.ibm.com	8.5.5.18	8.0.6.15	--	AppSrv1Cell	⋮
High (8.8)	batchServer	CVE-2021-26296 (+1 more)	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
High (8.8)	simpleServer	CVE-2021-26296	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮

Servers per page: 25 · 1 - 3 of 3 servers · 1 of 1 page

Accessing and starting the environment

The environment consists of two instances:

- A workstation which is a RHEL VM dedicated to one user. It is called ***Student VM*** in the lab instructions and contains WebSphere Application Server Traditional and Liberty installations.
- A WebSphere Automation instance which a shared environment. It is called **WSA environment** in the lab instructions.

You get access to the WSA environment via the Student VM.

1. Access the Student VM

a. Use the connection details That have been provided to you.

b. If you are connected via VNC, use the URL https://iccve.uk.ibm.com/cloudhur2.

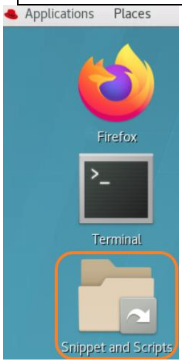
2. Login to the Student VM.

a. If you are connected via VNC, you should be automatically logged in as ibmdemo.

Otherwise log in as user “ibmdemo” and enter “passw0rd” as the password:
Password: `passw0rd` (lowercase with a zero instead of the `o`)

For your convenience, there are several scripts that ease the administration.
In addition, the main commands have been stored in the file
/usr/IBM/scripts/lab_WSAcommands.txt

ℹ	**Note:** To ease the copy and paste, the commands used in the lab have been stored into the file. Use the command `cat /usr/IBM/scripts/lab_WSAcommands.txt` or an editor to have them available for copy and paste

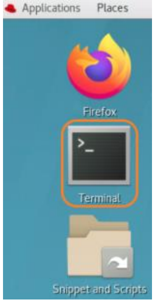


Complete the setup

Set the address for the WSA target

As this is a shared environment, the WebSphere Automation instance is rebuilt from time to time with a new IP address. Therefore you might have to adjust the hostname file.
If you are provided a newIP address and are asked to change the address, these are the steps to do so.

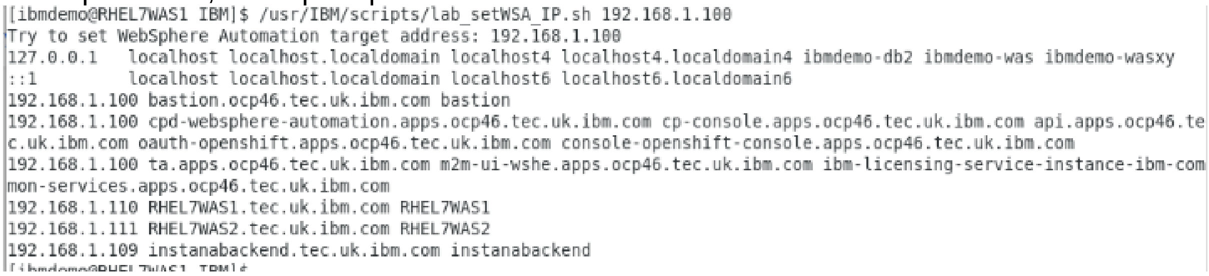
1. Open a terminal window by clicking its icon from the Desktop toolbar.



2. Run the following command with the IP address provided by the instructor.
(This is only needed if a remote WSA instance is used.)

`/usr/IBM/scripts/lab_setWSA_IP.sh` ***`<ip-address>`***

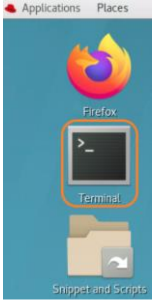
Enter passw0rd, when prompted.



```
[ibmdemo@RHEL7WAS1 IBM]$ /usr/IBM/scripts/lab_setWSA_IP.sh 192.168.1.100
Try to set WebSphere Automation target address: 192.168.1.100
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4 ibmdemo-db2 ibmdemo-was ibmdemo-wasxy
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.100 bastion.ocp46.tec.uk.ibm.com bastion
192.168.1.100 cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com cp-console.apps.ocp46.tec.uk.ibm.com api.apps.ocp46.te
c.uk.ibm.com oauth-openshift.apps.ocp46.tec.uk.ibm.com console-openshift-console.apps.ocp46.tec.uk.ibm.com
192.168.1.100 ta.apps.ocp46.tec.uk.ibm.com m2m-ui-wshe.apps.ocp46.tec.uk.ibm.com ibm-licensing-service-instance-ibm-com
mon-services.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
192.168.1.109 instanabackend.tec.uk.ibm.com instanabackend
[ibmdemo@RHEL7WAS1 IBM]$
```

Create your working directory

1. Open a terminal window by clicking its icon from the Desktop toolbar.



2. Run the following command and replace XX with the student number provided by the instructor.

```
export myUser=userXX
```

3. Create your working environment

```
export myWorkingDir=/var/IBM/$myUser
mkdir $myWorkingDir
cd $myWorkingDir
```

Receiving vulnerability notifications

Accessing the WebSphere Automation UI

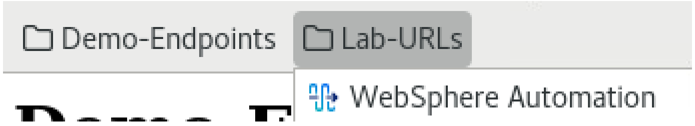
A WebSphere administrator sets up WebSphere Automation by registering and configuring WebSphere Application Servers and WebSphere Liberty servers for vulnerability tracking and by configuring email notifications.

WebSphere administrators can also view the results of vulnerability assessment in WebSphere Automation to plan their response for the WebSphere Application Server and WebSphere Liberty servers that they manage.

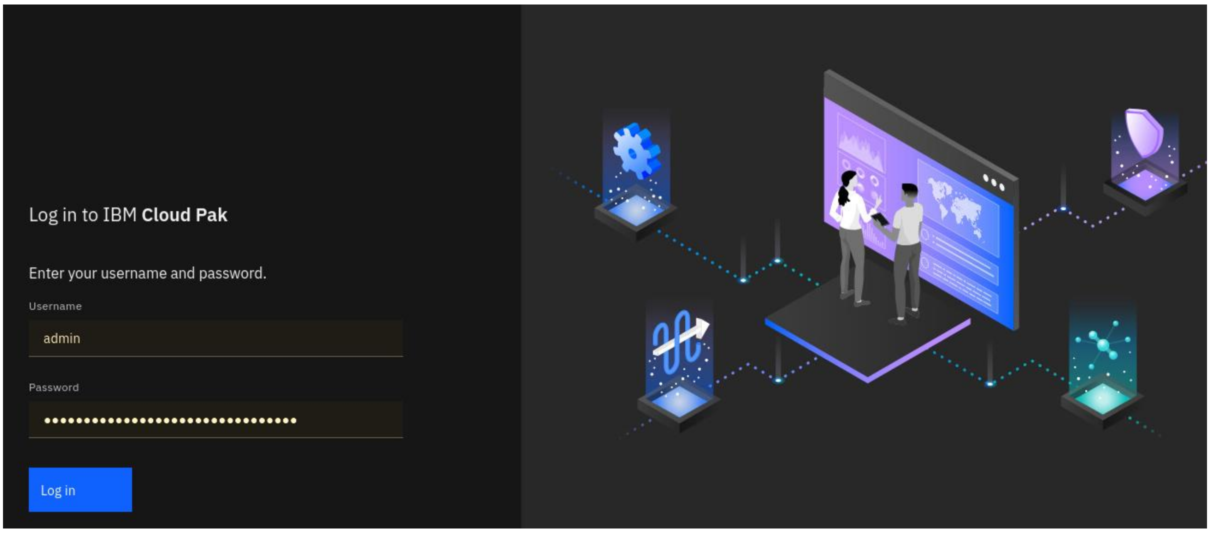
For this lab, WebSphere Automation is pre-installed on an OCP cluster. You have your individual WebSphere Automation installation. Let’s access your environment.

1. On the *Student VM*, open a browser and enter the following URL (there is a WebSphere Automation link on bookmark toolbar):

   https://cpd-websphere-automation.apps.ocp46.tec.uk.ibm.com

2. On the login panel, click on IBM provided credentials, then use the prefilled login credentials and click on **Log in**.
   (user: admin, password: JnarVX84CKz3bAWWqrtjXHF4N3M3UwiW)

   Log in to IBM **Cloud Pak**

   Enter your username and password.

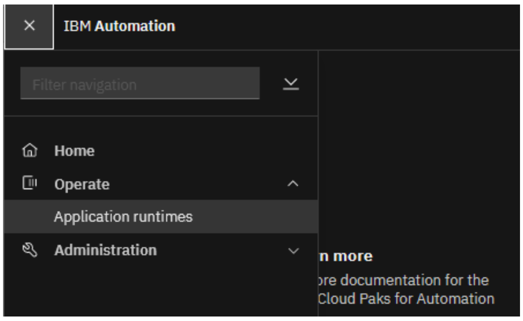
   Username: admin

   Password: ••••••••••••••••••••••••••••••••

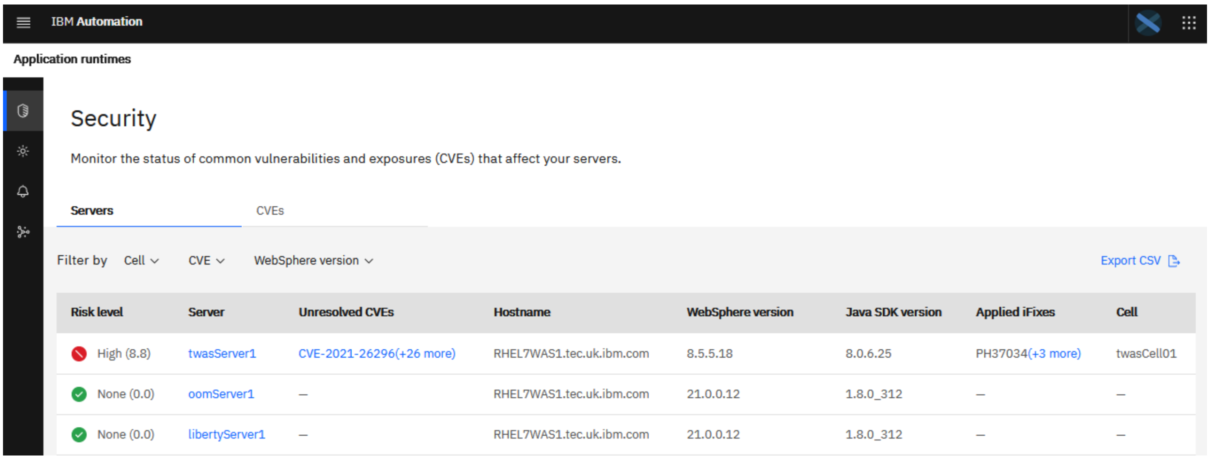
   Note: If necessary, accept all the warnings and certificates. Depending on your browser, you might have to scroll down to permit access.

3. You should automatically be routed to the **Application Runtimes** page.
   If not, open the **Menu**, click **Operate**, and then click **Application runtimes**.

   Application Runtimes represent the Traditional WebSphere and WebSphere Liberty servers that have been registered with IBM Automation.



4. The Application runtimes looks like the screenshot below. As you can see, there are three servers already registered. For all registered servers, you can see the version unresolved CVEs as well as applied fixes. The sorting is initially based on Risk Level.



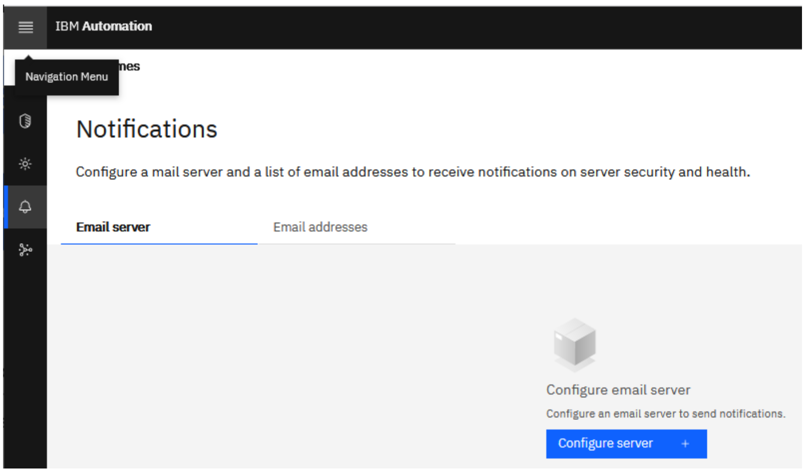
Security

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

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	–	–

5. Before you start to register servers to the Dashboard, you can configure an email to received notifications about CVEs. We do not use the notification in the lab, as we have a shared environment. But below you can find the steps how to configure it.

   a. Open the **Notification** menu.



Notifications

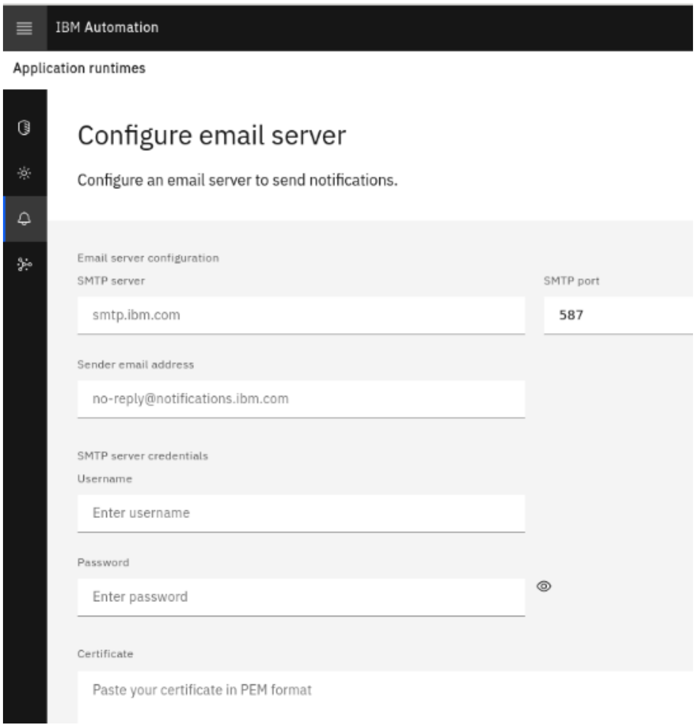
Configure a mail server and a list of email addresses to receive notifications on server security and health.

Configure email server

Configure an email server to send notifications.

b. On the panel, you could enter
   i. Your corporate email server
   ii. The email address of the security administrators to be notified

Please do not enter any details here as this is a shared environment.



Configure email server

Configure an email server to send notifications.

Email server configuration

SMTP server: smtp.ibm.com

SMTP port: 587

Sender email address: no-reply@notifications.ibm.com

SMTP server credentials

Username: Enter username

Password: Enter password

Certificate: Paste your certificate in PEM format

Getting configuration parameters

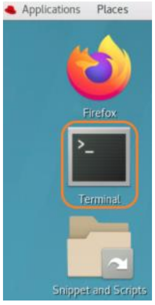
Add each of your WebSphere Application Server and WebSphere Liberty servers to WebSphere Automation by registering them with the **usage metering** service.

To register your application servers with the usage metering service in WebSphere Automation, you must configure the usage metering feature in each application server. To configure the usage metering feature in each of your application servers, you must obtain the following usage metering details:

- **URL**: The URL of the usage metering service in WebSphere Automation. This service registers WebSphere Application Server and Liberty servers with WebSphere Automation so that you can track security vulnerabilities.
- **API Key**: The token used to authenticate the WebSphere Application Server and Liberty servers during the registration process.
- **Usage metering certificate**: The certificate that contains the public key. This key allows an application server that is registering with WebSphere Automation to do an SSL handshake with the metering service.

Usually, you would get them directly from the WebSphere Automation administrator as you would not have an OpenShift CLI on your WebSphere machines. But in the lab environment, we have the client installed and access to the cluster. Let’s get these configuration parameters.

1. Return to the desktop and open a new **terminal** window.



2. The script **/usr/IBM/scripts/lab_retrieveWSADetails.sh** contains all necessary steps and stores the results under /usr/IBM/WSA. You can either run the script in one step or follow the steps in this section. We recommend to follow the step by step approach.

```
#  /usr/IBM/scripts/lab_retrieveWSADetails.sh
echo "******** Retrieve WSA Details ********"
mkdir /usr/IBM/WSA
cd /usr/IBM/WSA
oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u ocadmin -p passw0rd
--insecure-skip-tls-verify=true
oc project ws-automation-tec
# Meetering API
echo "******** Retrieve WSA metering URL ********"
oc get route cpd -o
jsonpath=https://{.spec.host}/websphereauto/meteringapi >
WSA_metering_URL.txt && cat WSA_metering_URL.txt
# API Key:
echo "******** Retrieve WSA API Key ********"
```

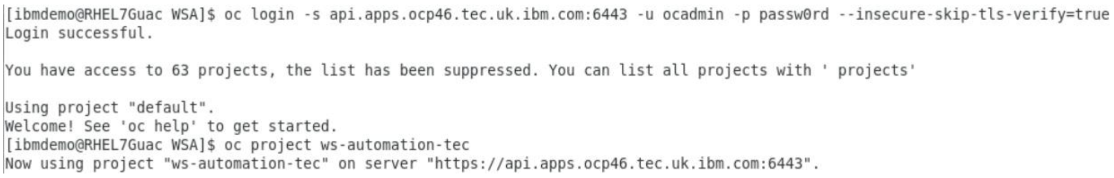
```
oc get secret wsa-secure-metering-apis-encrypted-tokens -o
jsonpath='{.data.wsa-secure-metering-apis-sa}' | base64 -d >
WSA_metering_api-key.txt && cat WSA_metering_api-key.txt
#Usage Metering Certificate
echo "******** Retrieve WSA Metering Certificate ********"
oc get secret external-tls-secret -o jsonpath='{.data.cert\.crt}' |
base64 -d >  WSA_metering_certificate_file.pem && cat
WSA_metering_certificate_file.pem
# Log out
oc logout

# Create a Keystore for metering
echo "******** Create WSA truststore ********"
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
echo "******** List Files with WSA Details ********"
ls -lrt WSA_metering*
```

a. Create the directory to store the WSA assets

```
mkdir /usr/IBM/WSA
cd /usr/IBM/WSA
```

b. Log into OpenShift and switch to the project **ws-automation-tec**:

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

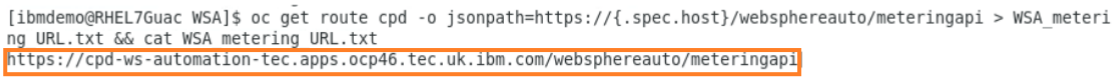


```
[ibmdemo@RHEL7Guac WSA]$ oc login -s api.apps.ocp46.tec.uk.ibm.com:6443 -u ocadmin -p passw0rd --insecure-skip-tls-verify=true
Login successful.

You have access to 63 projects, the list has been suppressed. You can list all projects with ' projects'

Using project "default".
Welcome! See 'oc help' to get started.
[ibmdemo@RHEL7Guac WSA]$ oc project ws-automation-tec
Now using project "ws-automation-tec" on server "https://api.apps.ocp46.tec.uk.ibm.com:6443".
```

c. Use the **oc** command to 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}/webspherauto/meteringapi >
WSA_metering_URL.txt && cat WSA_metering_URL.txt
```

The command will also display the result:

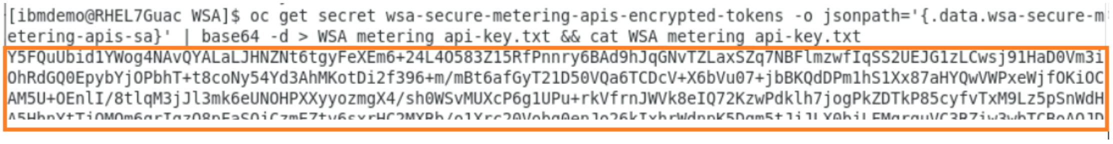


```
[ibmdemo@RHEL7Guac WSA]$ oc get route cpd -o jsonpath=https://{.spec.host}/webspherauto/meteringapi > WSA_meteri
ng URL.txt && cat WSA metering URL.txt
https://cpd-ws-automation-tec.apps.ocp46.tec.uk.ibm.com/webspherauto/meteringapi
```

d. 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
```

The command will also display the result:

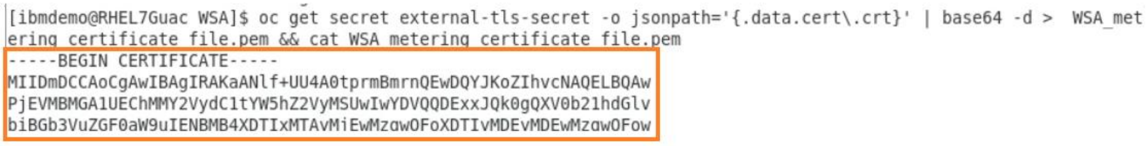


```
[ibmdemo@RHEL7Guac WSA]$ oc get secret wsa-secure-metering-apis-encrypted-tokens -o jsonpath='{.data.wsa-secure-m
etering-apis-sa}' | base64 -d > WSA_metering_api-key.txt && cat WSA_metering_api-key.txt
Y5FQuUbid1YWog4NAvQYALaLJHNZNt6tgyFeXEm6+24L40583Z15RfPnnry6BAd9hJqGNvTZLaxSZq7NBFlmzwfIqSS2UEJG1zLCwsj91HaD0Vm3i
OhRdGQ0EpybYjOPbhT+t8coNy54Yd3AhMKotDi2f396+m/mBt6afGyT21D50VQa6TCDcV+X6bVu07+jbBKQdDPm1hS1Xx87aHYQwVWPxeWjfOKiOC
AM5U+OEnlI/8tlqM3jJl3mk6eUNOHPXXyyozmgX4/sh0WSvMUXcP6g1UPu+rkVfrnJWVk8eIQ72KzwPdklh7jogPkZDTkP85cyfvTxM9Lz5pSnWdH
```

e. Finally, 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
```

The command will also display the result:



```
[ibmdemo@RHEL7Guac WSA]$ oc get secret external-tls-secret -o jsonpath='{.data.cert\.crt}' | base64 -d >  WSA_met
ering_certificate_file.pem && cat WSA_metering_certificate_file.pem
-----BEGIN CERTIFICATE-----
MIIDmDCCAoCgAwIBAgIRAKaANlf+UU4A0tprmBmrnQEwDQYJKoZIhvcNAQELBQAw
PjEVMBMGA1UEChMMY2VydC1tYW5hZ2VyMSUwIwYDVQQDExxJQk0gQXV0b21hdGlv
biBGb3VuZGF0aW9uIENBMB4XDTIxMTAvMiEwMzawOFoXDTIvMDEvMDEwMzawOFow
```

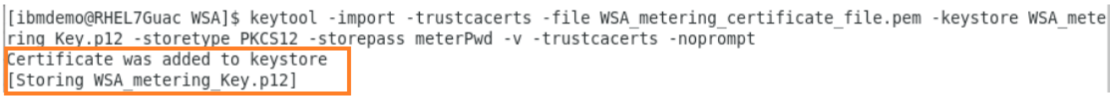
f. As we retrieved all required information from WSA, log out of OpenShift

```
oc logout
```

g. To ease the reuse, we store the certificate in a separate keystore that can be reused for any outbound connectivity to WebSphere Automation. We use the keytool that is part of the JDK to create the keystore.

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

Output of the command:

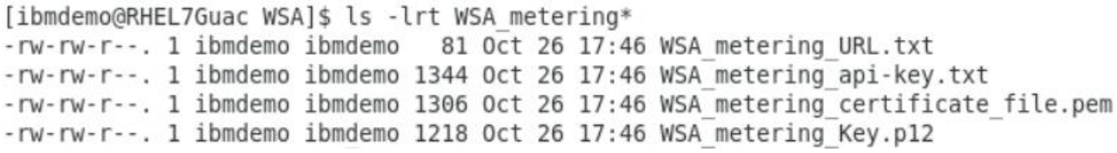


```
[ibmdemo@RHEL7Guac WSA]$ keytool -import -trustcacerts -file WSA_metering_certificate_file.pem -keystore WSA_mete
ring_Key.p12 -storetype PKCS12 -storepass meterPwd -v -trustcacerts -noprompt
Certificate was added to keystore
[Storing WSA_metering_Key.p12]
```

h. Now let's list all generated assets:

```
ls -lrt WSA_metering*
```

As you can see the following files have been generated:



```
[ibmdemo@RHEL7Guac WSA]$ ls -lrt WSA_metering*
-rw-rw-r--. 1 ibmdemo ibmdemo   81 Oct 26 17:46 WSA_metering_URL.txt
-rw-rw-r--. 1 ibmdemo ibmdemo 1344 Oct 26 17:46 WSA_metering_api-key.txt
-rw-rw-r--. 1 ibmdemo ibmdemo 1306 Oct 26 17:46 WSA_metering_certificate_file.pem
-rw-rw-r--. 1 ibmdemo ibmdemo 1218 Oct 26 17:46 WSA_metering_Key.p12
```

Great! Now you have all the configuration parameters necessary to register the application servers with the usage metering service in WebSphere Automation.

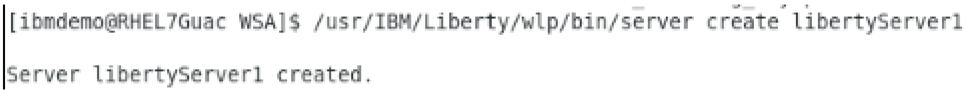
In the next section, you register your first server in WebSphere Automation.

Configuring Liberty server

In this section, you configure a Liberty Server instance to register to WebSphere Automation.
The Liberty binaries have been installed to /usr/IBM/Liberty/wlp.
Since Liberty servers are easily created, you will first create a new Liberty server and start it.

1. Create a new Liberty server, using the command below:

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



```
[ibmdemo@RHEL7Guac WSA]$ /usr/IBM/Liberty/wlp/bin/server create libertyServer1

Server libertyServer1 created.
```

2. Configure the usage metering in the new server. This is configured in the Liberty **server.xml** file. To allow reuse, we configure a separate server.xml with parameters.

```
<?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>
```

   a. Take a look at the server.xml file above

      i. The usageMetering feature has been enabled and defined

      ii. SSL has been configured to use the Truststore containing the WSA certificate.

      iii. The WSA details have been specified via variables WSA_Metering_Keystore, WSA_Metering_URL and WSA_Metering_api-key, which will be defined later.

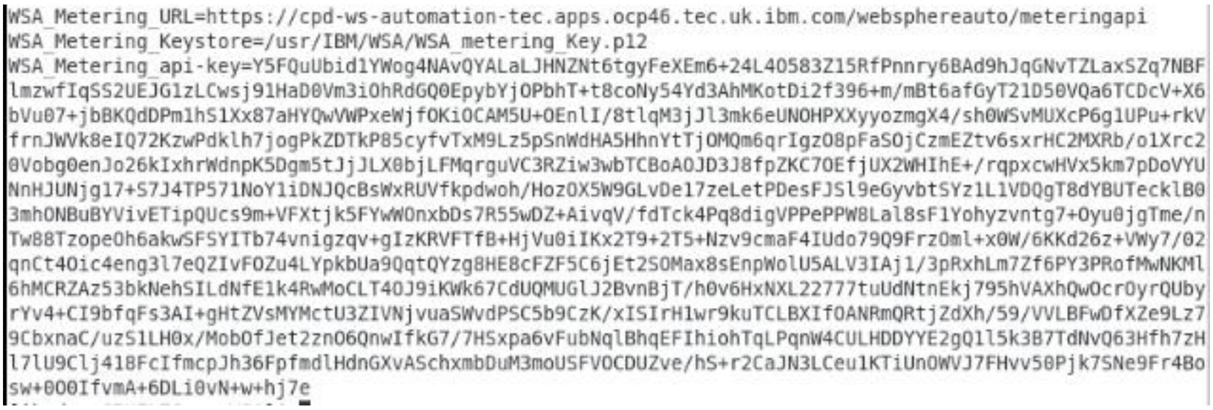
   b. The above shown server.xml file has already been created. You could copy the content into the existing server.xml file (which has been created via server create),

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

you could also use an include statement in the existing server.xml file. A third option, that you will use here, is the concept of config dropins, where you just copy the configuration into the appropriate directory and if will be picked up automatically. Here, you will use the configDropins/defaults directory.

3. Next you have to define the variables WSA_Metering_Keystore, WSA_Metering_URL and WSA_Metering_api-key. This can be done in the bootstrap.properties file. Instead of doing copy and paste, use the commands below.

```
echo "WSA_Metering_URL=$(cat /usr/IBM/WSA/WSA_metering_URL.txt)" >
/usr/IBM/WSA/bootstrap.properties
echo "WSA_Metering_Keystore=/usr/IBM/WSA/WSA_metering_Key.p12" >>
/usr/IBM/WSA/bootstrap.properties
echo "WSA_Metering_api-key=$(cat /usr/IBM/WSA/WSA_metering_api-key.txt)"
>> /usr/IBM/WSA/bootstrap.properties
cat /usr/IBM/WSA/bootstrap.properties
```



```
WSA_Metering_URL=https://cpd-ws-automation-tec.apps.ocp46.tec.uk.ibm.com/webspherauto/meteringapi
WSA_Metering_Keystore=/usr/IBM/WSA/WSA_metering_Key.p12
WSA_Metering_api-key=Y5FQuUbid1YWog4NAvQYALaLJHNZNt6tgyFeXEm6+24L40583Z15RfPnnry6BAd9hJqGNvTZLaxSZq7NBF
lmzwfIqSS2UEJG1zLCwsj91HaD0Vm3iOhRdGQ0EpybYjOPbhT+t8coNy54Yd3AhMKotDi2f396+m/mBt6afGyT21D50VQa6TCDcV+X6
bVu07+jbBKQdDPm1hS1Xx87aHYQwVWPxeWjfOKiOCAM5U+OEnlI/8tlqM3jJl3mk6eUNOHPXXyyozmgX4/sh0WSvMUXcP6g1UPu+rkV
frnJWVk8eIQ72KzwPdklh7jogPkZDTkP85cyfvTxM9Lz5pSnWdHA5HhnYtTjOMQm6qrIgzO8pFaSOjCzmEZtv6sxrHC2MXRb/o1Xrc2
0Vobg0enJo26kIxhrWdnpK5Dgm5tJjJLX8bjLFMqrguVC3RZiw3wbTCBoAOJD3J8fpZKC7OEfjUX2WHIhE+/rqpxcwHVx5km7pDoVYU
NnHJUNjg17+S7J4TP571NoY1iDNJQcBsWxRUVfkpdwoh/HozOX5W9GLvDe17zeLetPDesFJSl9eGyvbtSYz1L1VDQgT8dYBUTecklB0
3mhONBuBYVivETipQUcs9m+VFXtjk5FYwWOnxbDs7R55wDZ+AivqV/fdTck4Pq8digVPPePPW8Lal8sF1Yohyzvntg7+Oyu0jgTme/n
Tw88TzopeOh6akwSFSYITb74vnigzqv+gIzKRVFTfB+HjVu0iIKx2T9+2T5+Nzv9cmaF4IUdo79Q9FrzOml+x0W/6KKd26z+VWy7/02
qnCt4Oic4eng3l7eQZIvFOZu4LYpkbUa9QqtQYzg8HE8cFZF5C6jEt2SOMax8sEnpWolU5ALV3IAj1/3pRxhLm7Zf6PY3PRofMwNKMl
6hMCRZAz53bkNehSILdNfE1k4RwMoCLT4OJ9iKWk67CdUQMUGlJ2BvnBjT/h0v6HxNXL22777tuUdNtnEkj795hVAXhQwOcrOyrQUby
rYv4+CI9bfqFs3AI+gHtZVsMYMctU3ZIVNjvuaSWvdPSC5b9CzK/xISIrH1wr9kuTCLBXIfOANRmQRtjZdXh/59/VVLBFwDfXZe9Lz7
9CbxnaC/uzS1LH0x/MobOfJet2znO6QnwIfkG7/7HSxpa6vFubNqlBhqEFIhiohTqLPqnW4CULHDDYYE2gQ1l5k3B7TdNvQ63Hfh7zH
l7lU9Clj418FcIfmcpJh36FpfmdlHdnGXvASchxmbDuM3moUSFVOCDUZve/hS+r2CaJN3LCeu1KTiUnOWVJ7FHvv50Pjk7SNe9Fr4Bo
sw+0O0IfvmA+6DLi0vN+w+hj7e
```

4. Finally you have to add the bootstrap.properties file to the Liberty configuration. This could be done by merging it with the Liberty bootstrap file via copy and paste, another approach is to use an include to add it to the bootstrap file.

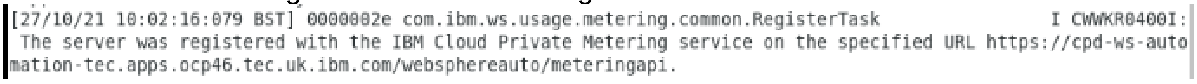
```
echo "bootstrap.include=/usr/IBM/WSA/bootstrap.properties" >>
/usr/IBM/Liberty/wlp/usr/servers/libertyServer1/bootstrap.properties
```

5. Now everything is configured, so let's start the Liberty server and it should register to the WebSphere Automation instance automatically.

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

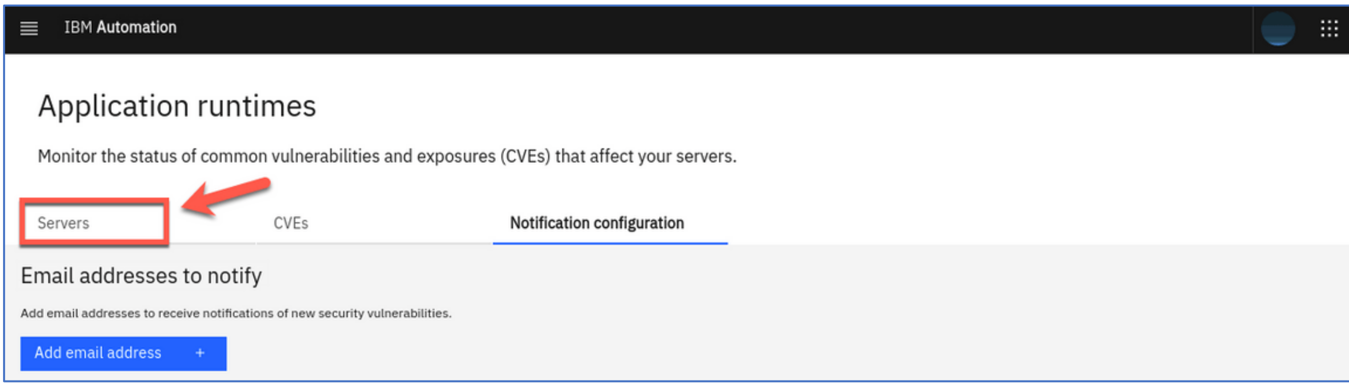
```
cat /usr/IBM/Liberty/wlp/usr/servers/libertyServer1/logs/messages.log
```

6. View the Liberty server messages.log file with cat and find the message indicating that the server was registered to the metering service.



```
[27/10/21 10:02:16:079 BST] 0000002e com.ibm.ws.usage.metering.common.RegisterTask           I CWWKR0400I:
 The server was registered with the IBM Cloud Private Metering service on the specified URL https://cpd-ws-auto
mation-tec.apps.ocp46.tec.uk.ibm.com/webspherauto/meteringapi.
```

7. In the WebSphere Automation UI, navigate to the **Servers** tab, to view the list of registered servers. The new Liberty server should be registered.



Application runtimes

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

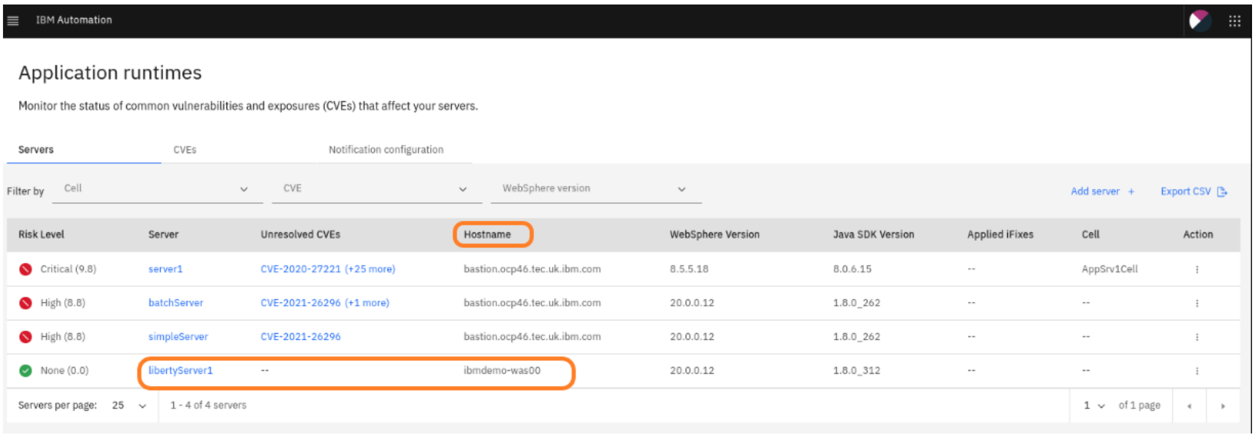
Email addresses to notify

Add email addresses to receive notifications of new security vulnerabilities.

a. Confirm that the Liberty server is registered in the WebSphere Automation Application runtimes page. If the Liberty server was successfully registered, it is displayed in the Application Runtimes in IBM automation UI.

    i. The name of the server is the same for all attendees, the Hostname is different, so you might have to scroll to find your server.

    ii. You can click on **Hostname** to sort by hostname



Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes	Cell	Action
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	bastion.ocp46.tec.uk.ibm.com	8.5.5.18	8.0.6.15	--	AppSrv1Cell	⋮
High (8.8)	batchServer	CVE-2021-26296 (+1 more)	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
High (8.8)	simpleServer	CVE-2021-26296	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
None (0.0)	libertyServer1	--	ibmdemo-was00	20.0.0.12	1.8.0_312	--	--	⋮

Servers per page: 25 | 1 - 4 of 4 servers | 1 of 1 page

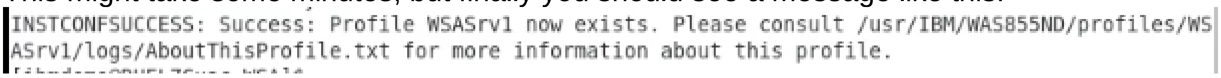
You have now registered your first Liberty server and can see if there are any vulnerabilities. As the server does not use a lot of features so far, there should be a Risk Level of 0, but this could change if new vulnerabilities are identified. It will also change during the lab.

Configuring traditional WebSphere (tWAS) v8.5.5

In this section, you configure a traditional WebSphere Application Server to your WebSphere Automation dashboard. With traditional WebSphere, you use the wsadmin script to configure the usage metering service.

```
# Create profile for standalone WAS
export WAS_HOME=/usr/IBM/WAS855ND
$WAS_HOME/bin/manageprofiles.sh -create \
   -profileName WSASrv1 \
   -templatePath $WAS_HOME/profileTemplates/default \
   -enableAdminSecurity false
```

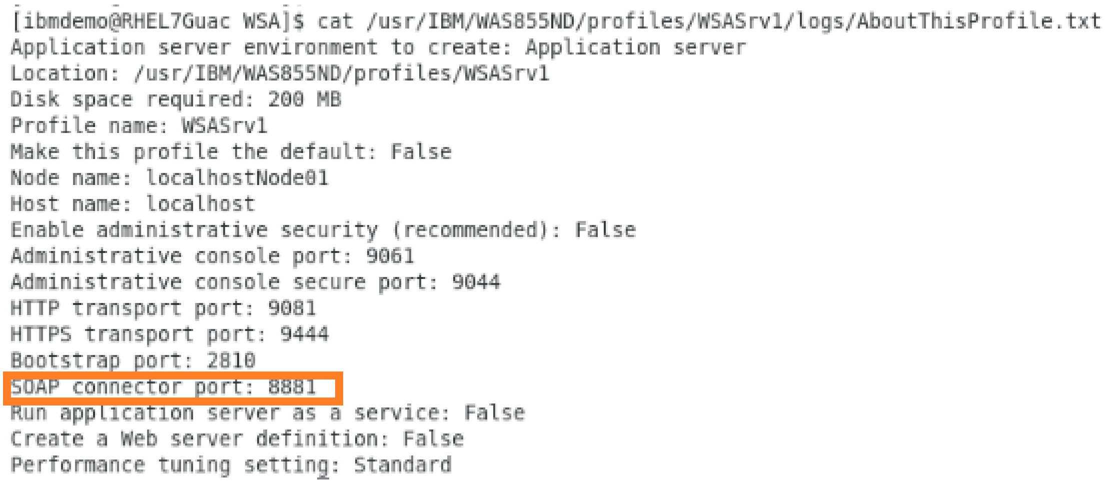
This might take some minutes, but finally you should see a message like this:



```
INSTCONFSUCCESS: Success: Profile WSASrv1 now exists. Please consult /usr/IBM/WAS855ND/profiles/WS
ASrv1/logs/AboutThisProfile.txt for more information about this profile.
```

Find out the SOAP port – this is the port we use to configure tWAS via script.

cat /usr/IBM/WAS855ND/profiles/WSASrv1/logs/AboutThisProfile.txt

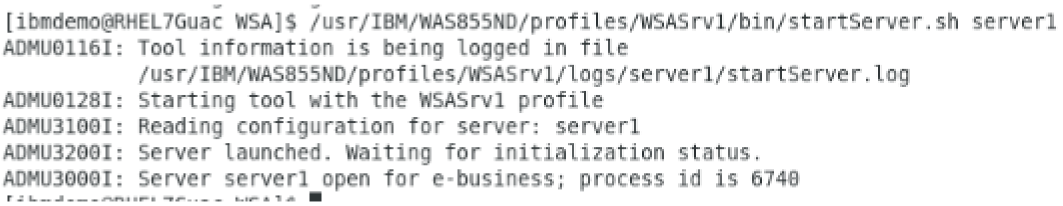


```
[ibmdemo@RHEL7Guac WSA]$ cat /usr/IBM/WAS855ND/profiles/WSASrv1/logs/AboutThisProfile.txt
Application server environment to create: Application server
Location: /usr/IBM/WAS855ND/profiles/WSASrv1
Disk space required: 200 MB
Profile name: WSASrv1
Make this profile the default: False
Node name: localhostNode01
Host name: localhost
Enable administrative security (recommended): False
Administrative console port: 9061
Administrative console secure port: 9044
HTTP transport port: 9081
HTTPS transport port: 9444
Bootstrap port: 2810
SOAP connector port: 8881
Run application server as a service: False
Create a Web server definition: False
Performance tuning setting: Standard
```

Now let’s start the newly created server instance:

/usr/IBM/WAS855ND/profiles/WSASrv1/bin/startServer.sh server1

After a minute or so, you should see a message that it has been started.



```
[ibmdemo@RHEL7Guac WSA]$ /usr/IBM/WAS855ND/profiles/WSASrv1/bin/startServer.sh server1
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrv1/logs/server1/startServer.log
ADMU0128I: Starting tool with the WSASrv1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 6740
```

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

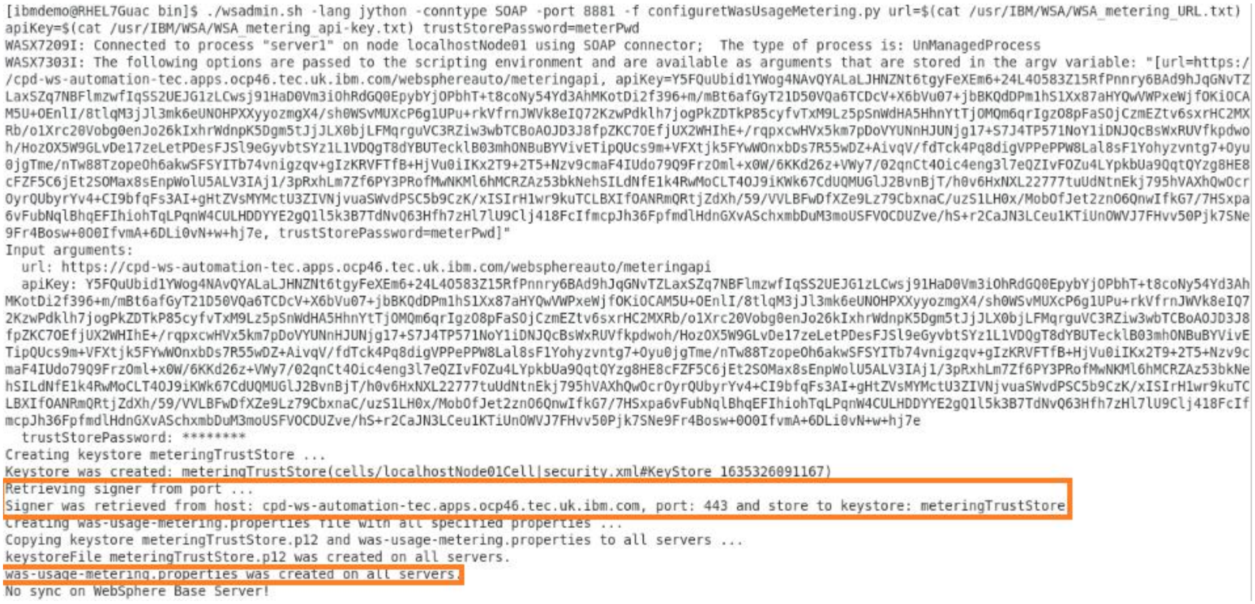
- As with Liberty, you first have to retrieve the metering URL as well as the API key. We will re-use the content of the two files that we created for Liberty.
- The WSA certificate will be retrieved from the WSA instance directly.
- To configure WAS, IBM provides a ready to use wsadmin script, you can find details here: https://www.ibm.com/docs/en/ws-automation?topic=vulnerabilities-adding-websphere-application-server-traditional-server
  The content of the script has been copied into the file configuretWasUsageMetering.py.

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

```
cp /var/IBM/software/WAS/configuretWasUsageMetering.py
/usr/IBM/WAS855ND/profiles/WSASrv1/bin
```

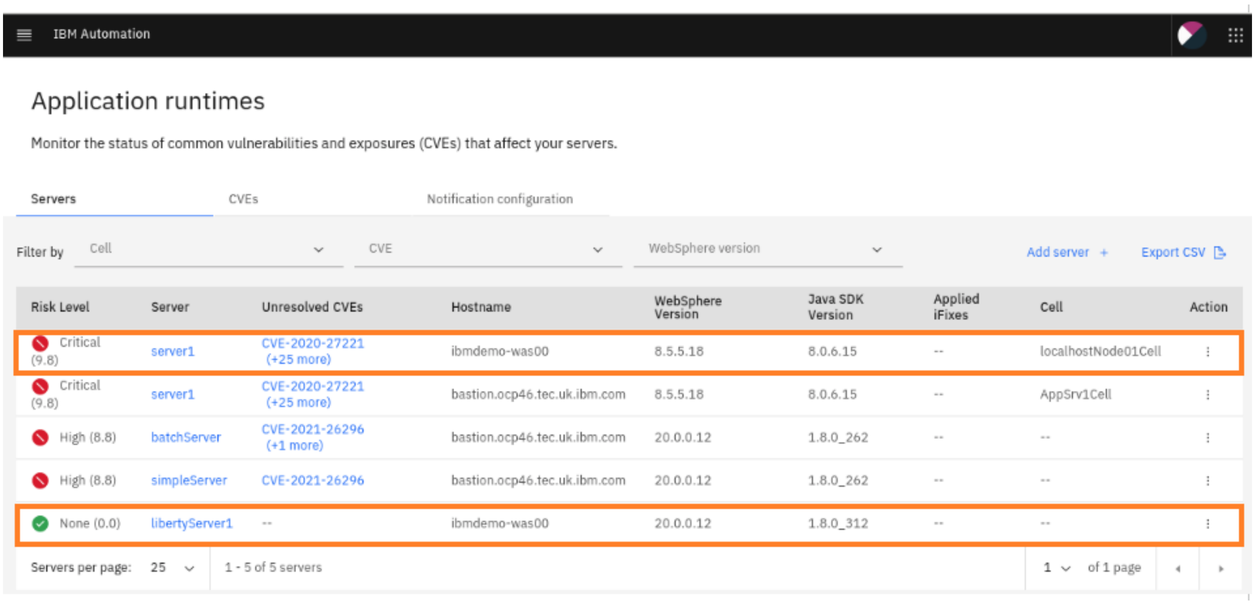
Switch to the WAS bin directory and run the wsadmin script

```
cd /usr/IBM/WAS855ND/profiles/WSASrv1/bin
./wsadmin.sh -lang jython -conntype SOAP -port 8881 -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@RHEL7Guac bin]$ ./wsadmin.sh -lang jython -conntype SOAP -port 8881 -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 "server1" on node localhostNode01 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-ws-automation-tec.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi, apiKey=Y5FQuUbid1YWog4NAvQYALaLJHNZNt6tgyFeXEm6+24L40583Z15RfPnnry6BAd9hJqGNvTZ
LaxSZq7NBFlmzwfIqSS2UEJG1zLCwsj91HaD0Vm3iOhRdGQ0EpybYjOPbhT+t8coNy54Yd3AhMKotDi2f396+m/mBt6afGyT21D50VQa6TCDcV+X6bVu07+jbBKQdDPm1hS1Xx87aHYQwVWPxeWjfOKiOCA
M5U+OEnlI/8tlqM3jJl3mk6eUNOHPXXyyozmgX4/sh0WSvMUXcP6g1UPu+rkVfrnJWVk8eIQ72KzwPdklh7jogPkZDTkP85cyfvTxM9Lz5pSnWdHA5HhnYtTjOMQm6qrIgzO8pFaSOjCzmEZtv6sxrHC2MX
Rb/o1Xrc20Vobg0enJo26kIxhrWdnpK5Dgm5tJjJLX0bjLFMqrguVC3RZiw3wbTCBoAOJD3J8fpZKC7OEfjUX2WHIhE+/rqpxcwHVx5km7pDoVYUNnHJUNjg17+S7J4TP571NoY1iDNJQcBsWxRUVfkpdwo
h/HozOX5W9GLvDe17zeLetPDesFJSl9eGyvbtSYz1L1VDQgT8dYBUTecklB03mhONBuBYVivETipQUcs9m+VFXtjk5FYwWOnxbDs7R55wDZ+AivqV/fdTck4Pq8digVPPePPW8Lal8sF1Yohyzvntg7+Oyu
0jgTme/nTw88TzopeOh6akwSFSYITb74vnigzqv+gIzKRVFTfB+HjVu0iIKx2T9+2T5+Nzv9cmaF4IUdo79Q9FrzOml+x0W/6KKd26z+VWy7/02qnCt4Oic4eng3l7eQZIvFOZu4LYpkbUa9QqtQYzg8HE8
cFZF5C6jEt2SOMax8sEnpWolU5ALV3IAj1/3pRxhLm7Zf6PY3PRofMwNKMl6hMCRZAz53bkNehSILdNfE1k4RwMoCLT4OJ9iKWk67CdUQMUGlJ2BvnBjT/h0v6HxNXL22777tuUdNtnEkj795hVAXhQwOcr
OyrQUbyrYv4+CI9bfqFs3AI+gHtZVsMYMctU3ZIVNjvuaSWvdPSC5b9CzK/xISIrH1wr9kuTCLBXIfOANRmQRtjZdXh/59/VVLBFwDfXZe9Lz79CbxnaC/uzS1LH0x/MobOfJet2znO6QnwIfkG7/7HSxpa
6vFubNqlBhqEFIhiohTqLPqnW4CULHDDYYE2gQ1l5k3B7TdNvQ63Hfh7zHl7lU9Clj418FcIfmcpJh36FpfmdlHdnGXvASchxmbDuM3moUSFVOCDUZve/hS+r2CaJN3LCeu1KTiUnOWVJ7FHvv50Pjk7SNe
9Fr4Bosw+0O0IfvmA+6DLi0vN+w+hj7e, trustStorePassword=meterPwd]"
Input arguments:
  url: https://cpd-ws-automation-tec.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi
  apiKey: Y5FQuUbid1YWog4NAvQYALaLJHNZNt6tgyFeXEm6+24L40583Z15RfPnnry6BAd9hJqGNvTZLaxSZq7NBFlmzwfIqSS2UEJG1zLCwsj91HaD0Vm3iOhRdGQ0EpybYjOPbhT+t8coNy54Yd3Ah
MKotDi2f396+m/mBt6afGyT21D50VQa6TCDcV+X6bVu07+jbBKQdDPm1hS1Xx87aHYQwVWPxeWjfOKiOCAM5U+OEnlI/8tlqM3jJl3mk6eUNOHPXXyyozmgX4/sh0WSvMUXcP6g1UPu+rkVfrnJWVk8eIQ7
2KzwPdklh7jogPkZDTkP85cyfvTxM9Lz5pSnWdHA5HhnYtTjOMQm6qrIgzO8pFaSOjCzmEZtv6sxrHC2MXRb/o1Xrc20Vobg0enJo26kIxhrWdnpK5Dgm5tJjJLX0bjLFMqrguVC3RZiw3wbTCBoAOJD3J8
fpZKC7OEfjUX2WHIhE+/rqpxcwHVx5km7pDoVYUNnHJUNjg17+S7J4TP571NoY1iDNJQcBsWxRUVfkpdwoh/HozOX5W9GLvDe17zeLetPDesFJSl9eGyvbtSYz1L1VDQgT8dYBUTecklB03mhONBuBYVivE
TipQUcs9m+VFXtjk5FYwWOnxbDs7R55wDZ+AivqV/fdTck4Pq8digVPPePPW8Lal8sF1Yohyzvntg7+Oyu0jgTme/nTw88TzopeOh6akwSFSYITb74vnigzqv+gIzKRVFTfB+HjVu0iIKx2T9+2T5+Nzv9c
maF4IUdo79Q9FrzOml+x0W/6KKd26z+VWy7/02qnCt4Oic4eng3l7eQZIvFOZu4LYpkbUa9QqtQYzg8HE8cFZF5C6jEt2SOMax8sEnpWolU5ALV3IAj1/3pRxhLm7Zf6PY3PRofMwNKMl6hMCRZAz53bkNe
hSILdNfE1k4RwMoCLT4OJ9iKWk67CdUQMUGlJ2BvnBjT/h0v6HxNXL22777tuUdNtnEkj795hVAXhQwOcrOyrQUbyrYv4+CI9bfqFs3AI+gHtZVsMYMctU3ZIVNjvuaSWvdPSC5b9CzK/xISIrH1wr9kuTC
LBXIfOANRmQRtjZdXh/59/VVLBFwDfXZe9Lz79CbxnaC/uzS1LH0x/MobOfJet2znO6QnwIfkG7/7HSxpa6vFubNqlBhqEFIhiohTqLPqnW4CULHDDYYE2gQ1l5k3B7TdNvQ63Hfh7zHl7lU9Clj418FcIf
mcpJh36FpfmdlHdnGXvASchxmbDuM3moUSFVOCDUZve/hS+r2CaJN3LCeu1KTiUnOWVJ7FHvv50Pjk7SNe9Fr4Bosw+0O0IfvmA+6DLi0vN+w+hj7e
  trustStorePassword: ********
Creating keystore meteringTrustStore ...
Keystore was created: meteringTrustStore(cells/localhostNode01Cell|security.xml#KeyStore_1635326091167)
Retrieving signer from port ...
Signer was retrieved from host: cpd-ws-automation-tec.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!
```

Switch back to the browser and you should see that the server has been registered.



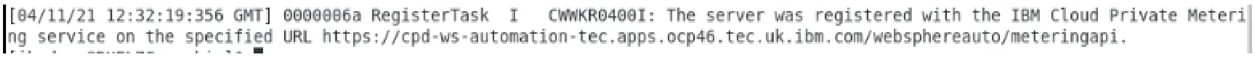
Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes	Cell	Action
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15	--	localhostNode01Cell	⋮
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	bastion.ocp46.tec.uk.ibm.com	8.5.5.18	8.0.6.15	--	AppSrv1Cell	⋮
High (8.8)	batchServer	CVE-2021-26296 (+1 more)	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
High (8.8)	simpleServer	CVE-2021-26296	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
None (0.0)	libertyServer1	--	ibmdemo-was00	20.0.0.12	1.8.0_312	--	--	⋮

You can also find a message in the log file SystemOut.log.

(for ex. via cat /usr/IBM/WAS855ND/profiles/WSASrv1/logs/server1/SystemOut.log)



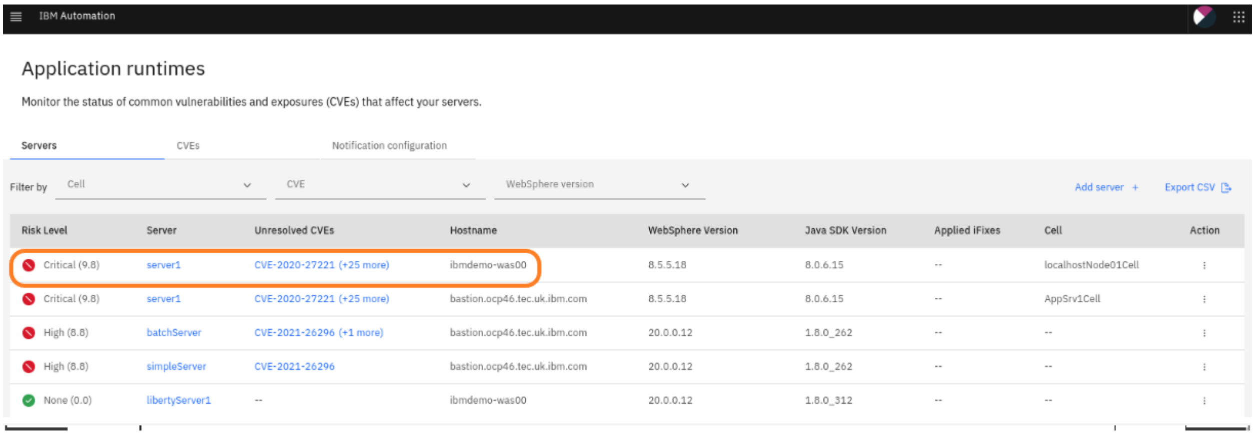
```
[04/11/21 12:32:19:356 GMT] 0000006a RegisterTask  I   CWWKR0400I: The server was registered with the IBM Cloud Private Meteri
ng service on the specified URL https://cpd-ws-automation-tec.apps.ocp46.tec.uk.ibm.com/websphereauto/meteringapi.
```

If there is an issue, then restart the server.

```
/usr/IBM/WAS85ND/profiles/WSASrv1/bin/stopServer.sh server1
/usr/IBM/WAS855ND/profiles/WSASrv1/bin/startServer.sh server1
```

Get insight from WebSphere Automation

Switch to the browser tab for WebSphere Automation and you can see that the WebSphere Traditional instance server1 has several unresolved CVEs.

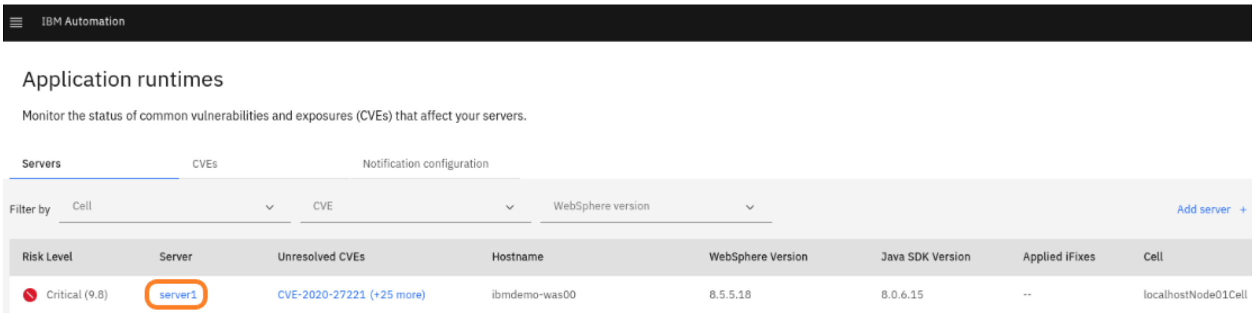


Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes	Cell	Action
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15	--	localhostNode01Cell	⋮
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	bastion.ocp46.tec.uk.ibm.com	8.5.5.18	8.0.6.15	--	AppSrv1Cell	⋮
High (8.8)	batchServer	CVE-2021-26296 (+1 more)	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
High (8.8)	simpleServer	CVE-2021-26296	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
None (0.0)	libertyServer1	--	ibmdemo-was00	20.0.0.12	1.8.0_312	--	--	⋮

Click on **server1** to get more details on the issues related to server1.

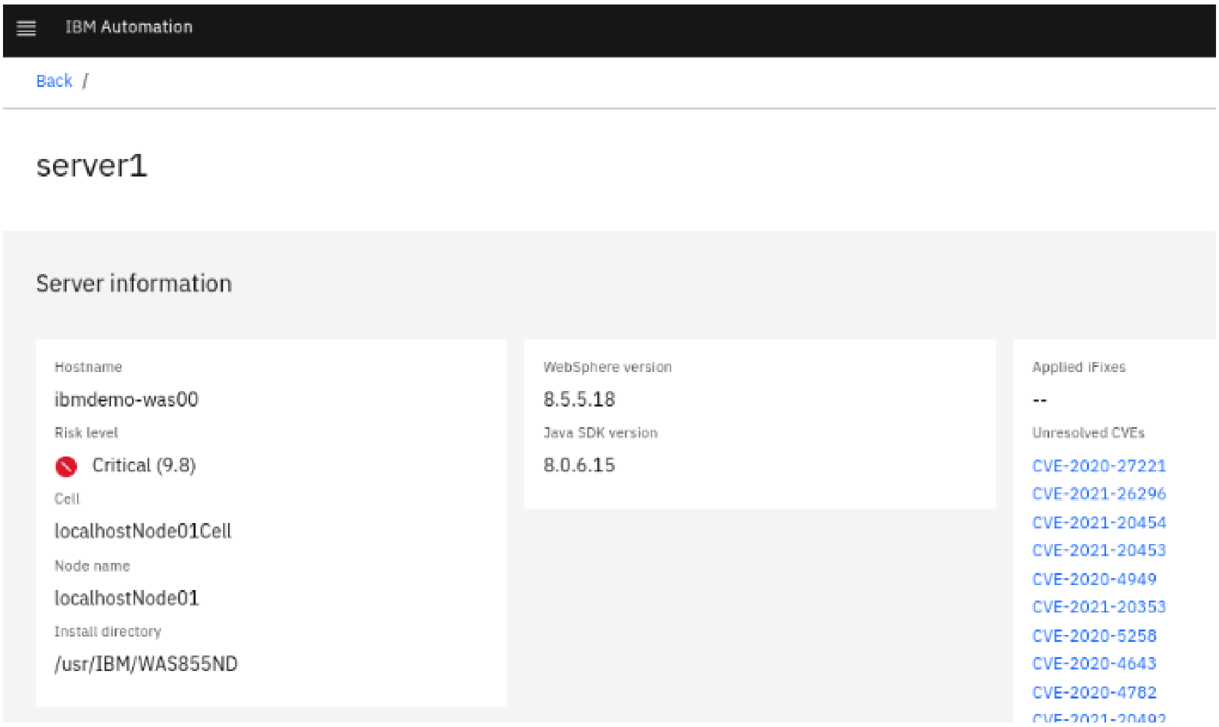


Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes	Cell
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15	--	localhostNode01Cell

This will show all the vulnerabilities that have been found for server1.



server1

Server information

Hostname
ibmdemo-was00

Risk level
Critical (9.8)

Cell
localhostNode01Cell

Node name
localhostNode01

Install directory
/usr/IBM/WAS855ND

WebSphere version
8.5.5.18

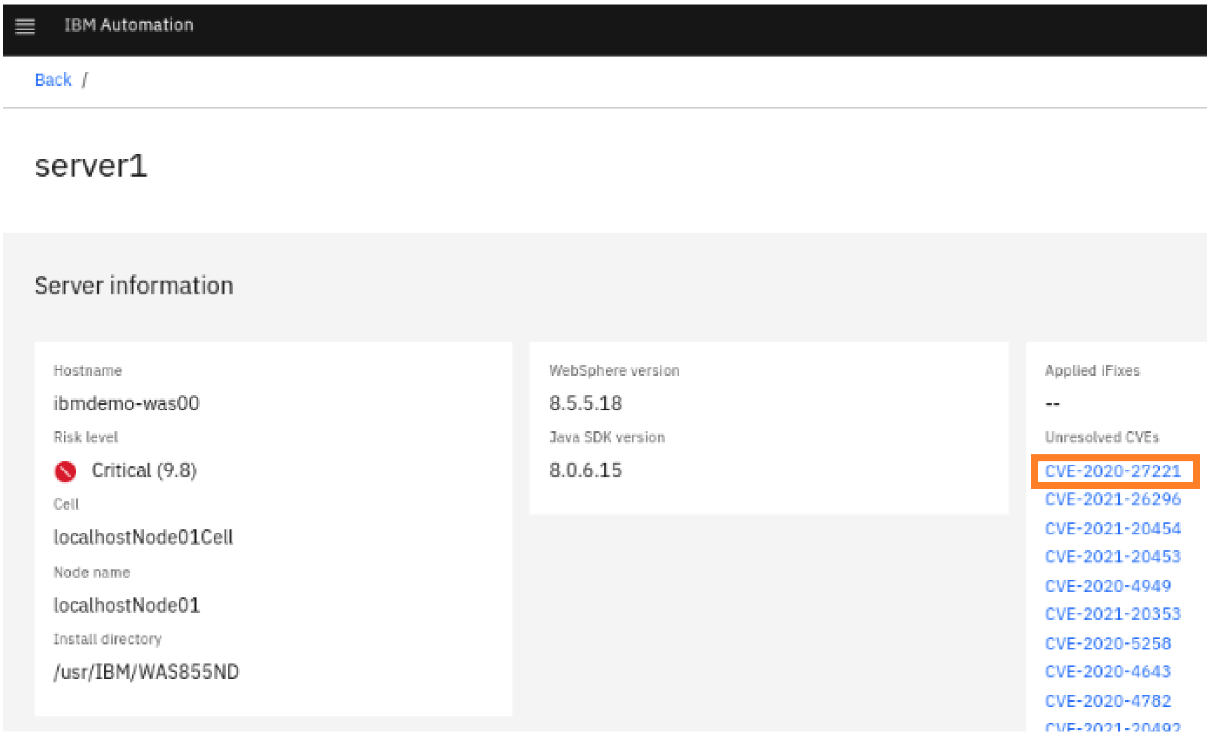
Java SDK version
8.0.6.15

Applied iFixes
--

Unresolved CVEs
CVE-2020-27221
CVE-2021-26296
CVE-2021-20454
CVE-2021-20453
CVE-2020-4949
CVE-2021-20353
CVE-2020-5258
CVE-2020-4643
CVE-2020-4782
CVE-2021-20492

The list will help you to consolidate maintenance efforts.

Click on one of the **Unresolved CVEs** to get details about the CVE.



IBM Automation

Back /

server1

Server information

Hostname
ibmdemo-was00

Risk level
Critical (9.8)

Cell
localhostNode01Cell

Node name
localhostNode01

Install directory
/usr/IBM/WAS855ND

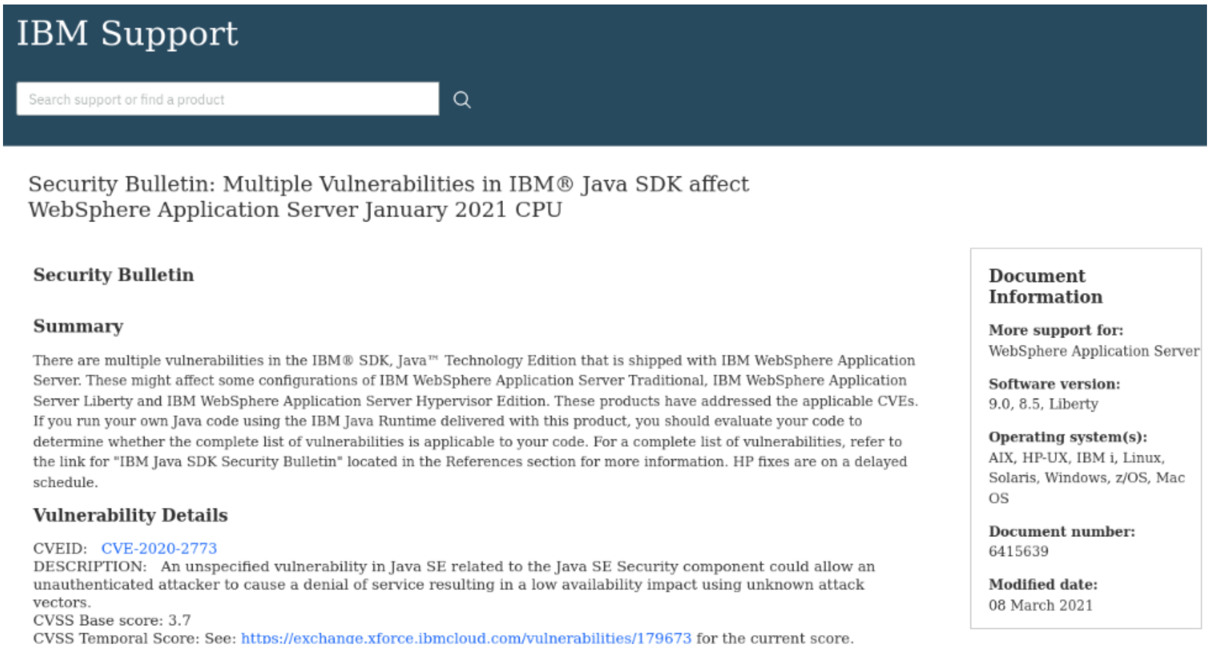
WebSphere version
8.5.5.18

Java SDK version
8.0.6.15

Applied iFixes
--

Unresolved CVEs
CVE-2020-27221
CVE-2021-26296
CVE-2021-20454
CVE-2021-20453
CVE-2020-4949
CVE-2021-20353
CVE-2020-5258
CVE-2020-4643
CVE-2020-4782
CVE-2021-20492

A new tab will open, and you will be routed to the support page which provide the details about the CVE and how to resolve it. Close the tab or switch back to the WebSphere Automation tab.



IBM Support

Search support or find a product

Security Bulletin: Multiple Vulnerabilities in IBM® Java SDK affect WebSphere Application Server January 2021 CPU

Security Bulletin

Summary

There are multiple vulnerabilities in the IBM® SDK, Java™ Technology Edition that is shipped with IBM WebSphere Application Server. These might affect some configurations of IBM WebSphere Application Server Traditional, IBM WebSphere Application Server Liberty and IBM WebSphere Application Server Hypervisor Edition. These products have addressed the applicable CVEs. If you run your own Java code using the IBM Java Runtime delivered with this product, you should evaluate your code to determine whether the complete list of vulnerabilities is applicable to your code. For a complete list of vulnerabilities, refer to the link for "IBM Java SDK Security Bulletin" located in the References section for more information. HP fixes are on a delayed schedule.

Vulnerability Details

CVEID: CVE-2020-2773
DESCRIPTION: An unspecified vulnerability in Java SE related to the Java SE Security component could allow an unauthenticated attacker to cause a denial of service resulting in a low availability impact using unknown attack vectors.
CVSS Base score: 3.7
CVSS Temporal Score: See: https://exchange.xforce.ibmcloud.com/vulnerabilities/179673 for the current score.

Document Information

More support for:
WebSphere Application Server

Software version:
9.0, 8.5, Liberty

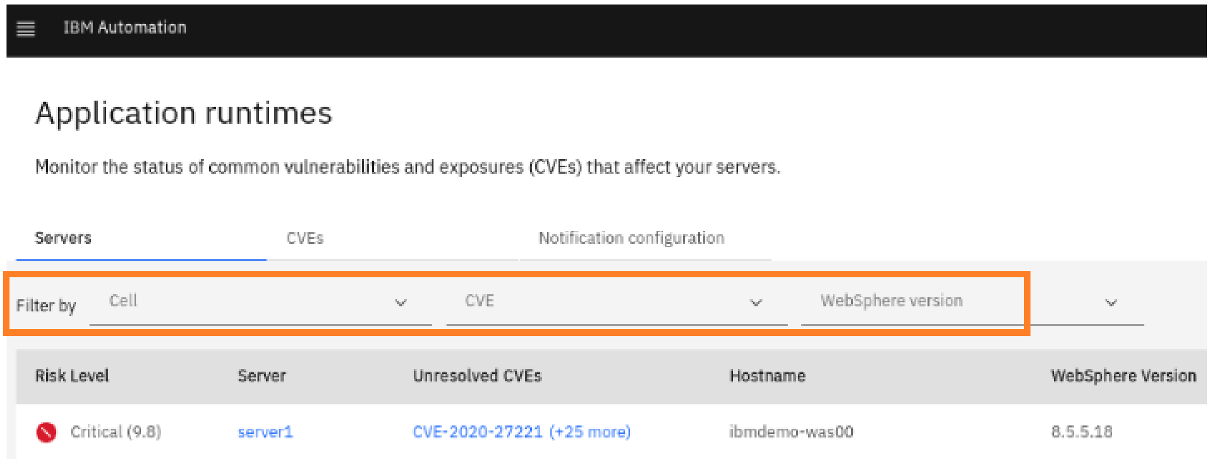
Operating system(s):
AIX, HP-UX, IBM i, Linux, Solaris, Windows, z/OS, Mac OS

Document number:
6415639

Modified date:
08 March 2021

Switch back to the WebSphere Automation Runtime panel.

You can also filter by Cell, CVE or WebSphere version for example.



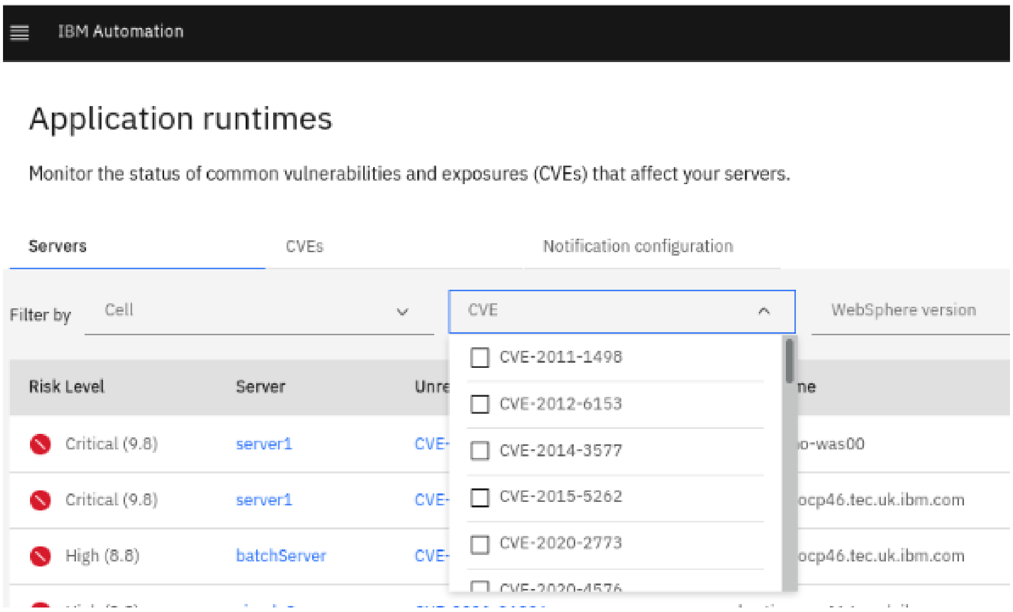
IBM Automation

Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18

Open the CVE twisty to filter by CVE.



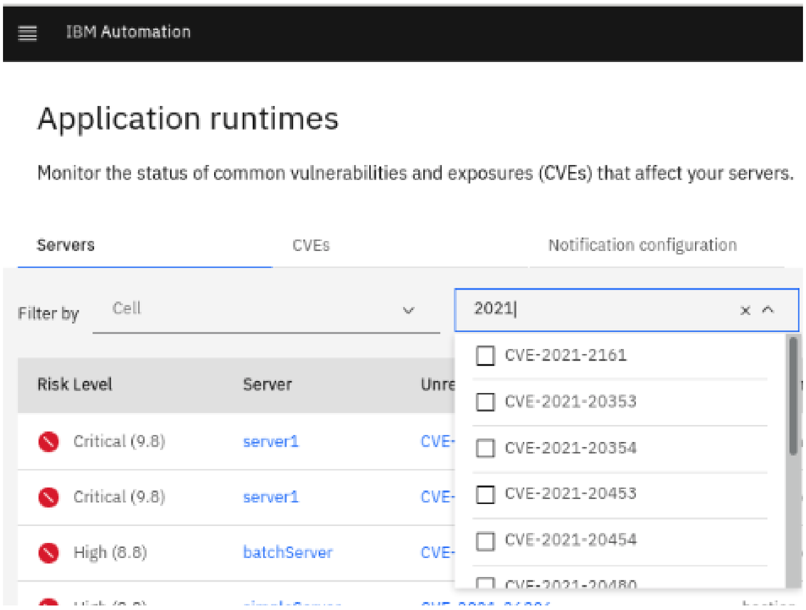
IBM Automation

Application runtimes

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

□ CVE-2011-1498
□ CVE-2012-6153
□ CVE-2014-3577
□ CVE-2015-5262
□ CVE-2020-2773
□ CVE-2020-4576

Risk Level	Server	Unre	ne
Critical (9.8)	server1	CVE-	o-was00
Critical (9.8)	server1	CVE-	ocp46.tec.uk.ibm.com
High (8.8)	batchServer	CVE-	ocp46.tec.uk.ibm.com



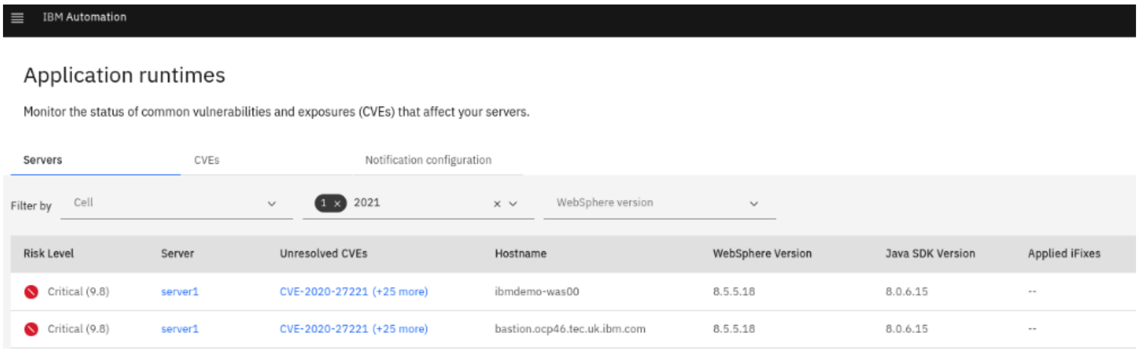
Application runtimes

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

2021|

- □ CVE-2021-2161
- □ CVE-2021-20353
- □ CVE-2021-20354
- □ CVE-2021-20453
- □ CVE-2021-20454
- □ CVE-2021-20480

Risk Level	Server	Unre
Critical (9.8)	server1	CVE-
Critical (9.8)	server1	CVE-
High (8.8)	batchServer	CVE-



Application runtimes

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

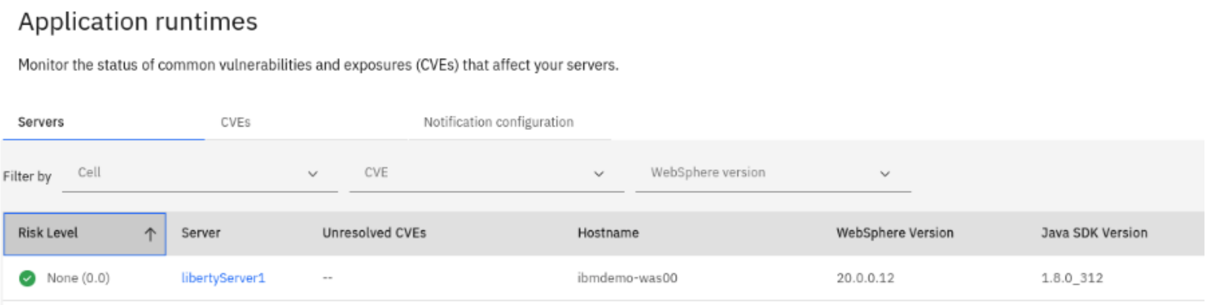
2021

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15	--
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	bastion.ocp46.tec.uk.ibm.com	8.5.5.18	8.0.6.15	--

This helps to see which servers overall are in risk and if work can be consolidated, it also helps to answer questions like “are we impacted by CVE xxx?”.

Update the Liberty server configuration

As you have seen, the Liberty instance right now does not have any issue.



Application runtimes

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

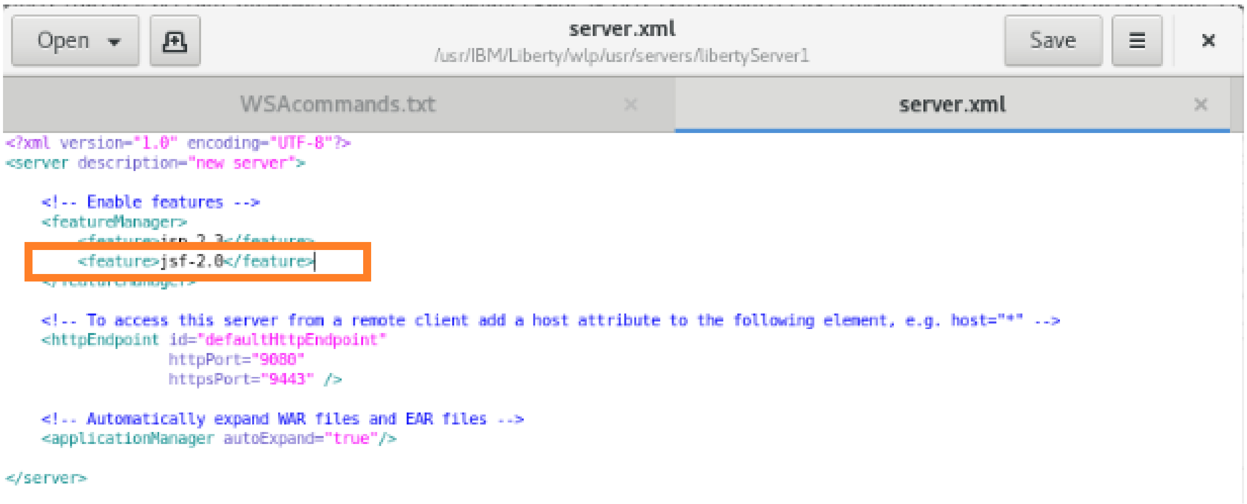
Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version
✓ None (0.0)	libertyServer1	--	ibmdemo-was00	20.0.0.12	1.8.0_312

In this section, you will simulate the situation of an application deployment, where an updated application introduces new Liberty features.

Open the Liberty server configuration to add jsf-2.0 as new features.

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

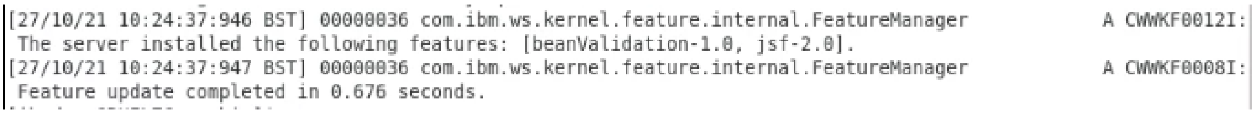
Add the row <feature>jsf-2.0</feature> as shown in the screenshot below.



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

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.3</feature>
        <feature>jsf-2.0</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9080"
                  httpsPort="9443" />

    <!-- Automatically expand WAR files and EAR files -->
    <applicationManager autoExpand="true"/>

</server>
```

As Liberty is configured for dynamic updates, the configuration change is applied on the fly. You can see this in the server log messages.log

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

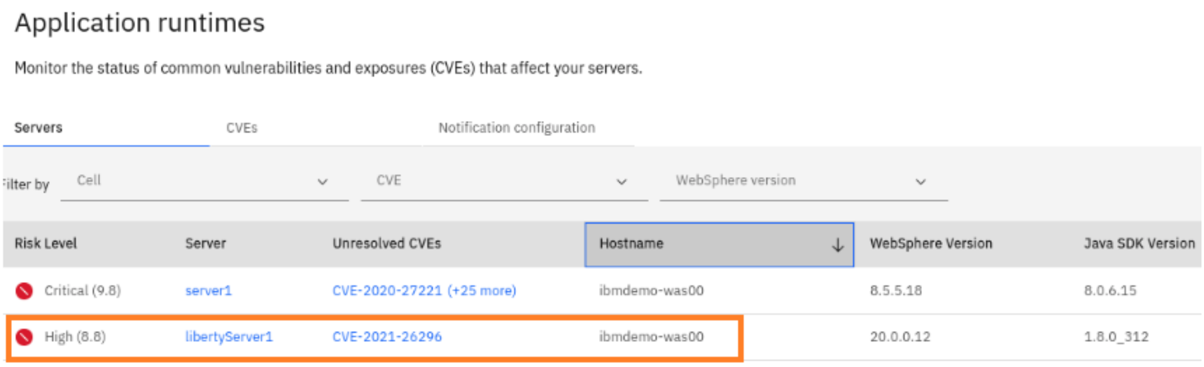


```
[27/10/21 10:24:37:946 BST] 00000036 com.ibm.ws.kernel.feature.internal.FeatureManager           A CWWKF0012I:
 The server installed the following features: [beanValidation-1.0, jsf-2.0].
[27/10/21 10:24:37:947 BST] 00000036 com.ibm.ws.kernel.feature.internal.FeatureManager           A CWWKF0008I:
 Feature update completed in 0.676 seconds.
```

As you can see, the jsf-2.0 feature activated the beanValidation feature under the cover.

Wasn’t there a CVE alert around beanValication some weeks ago?

Let’s switch to WebSphere Automation and sort by Hostname, then look for your instance. You should see, that WebSphere Automation identified an unresolved vulnerability for your liberty server instance.

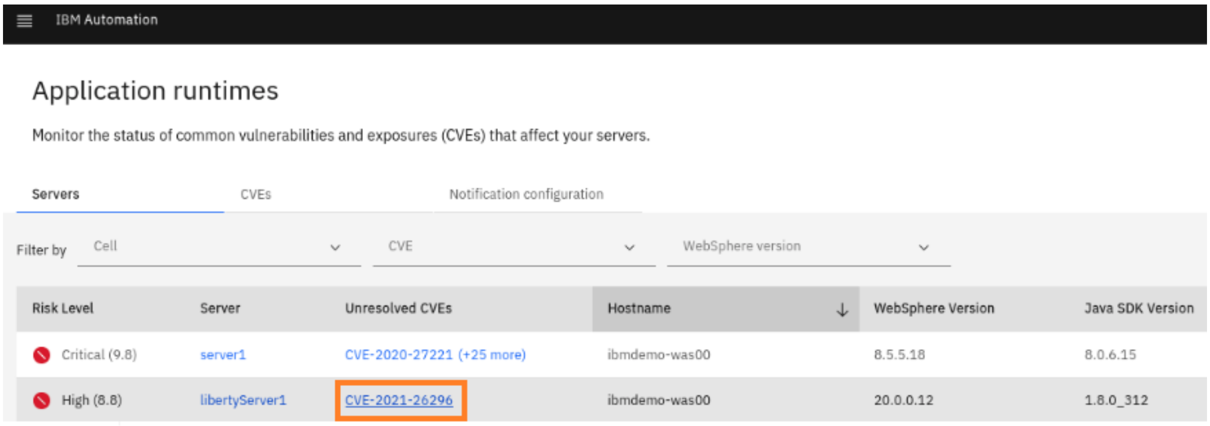


Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname ↓	WebSphere Version	Java SDK Version
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15
High (8.8)	libertyServer1	CVE-2021-26296	ibmdemo-was00	20.0.0.12	1.8.0_312

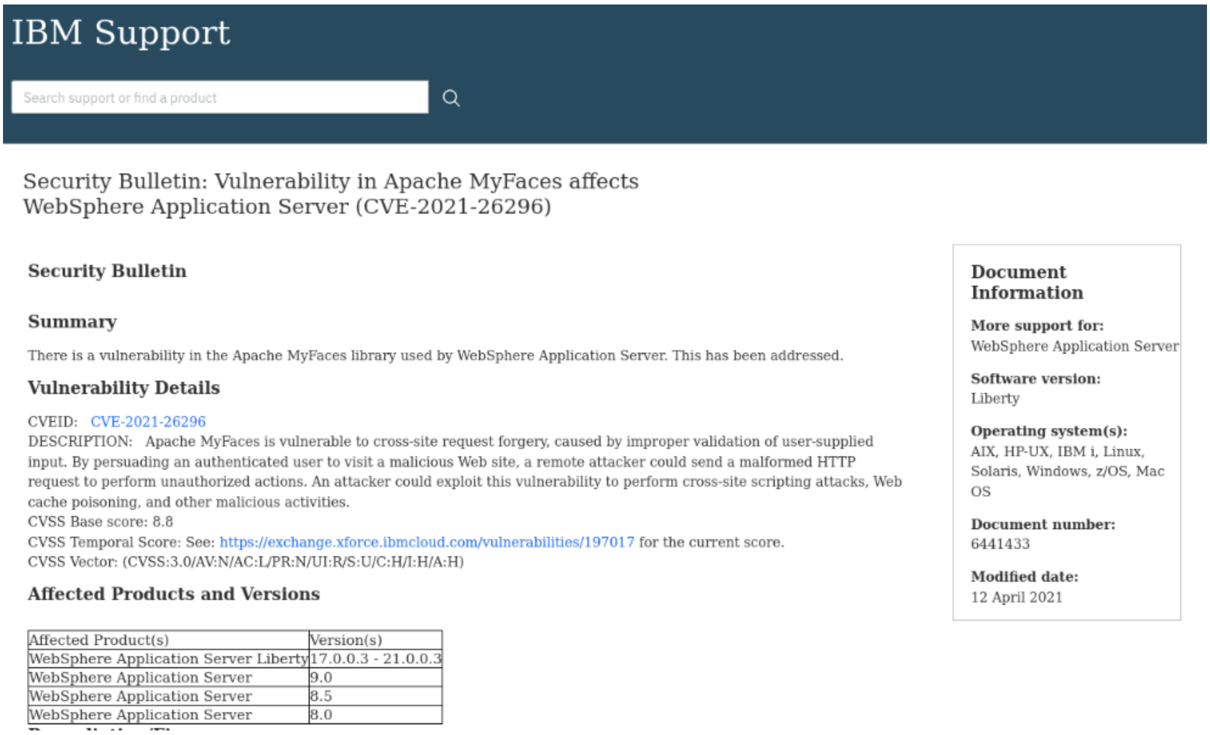
Click on the CVE to open the related CVE alert.



Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname ↓	WebSphere Version	Java SDK Version
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15
High (8.8)	libertyServer1	CVE-2021-26296	ibmdemo-was00	20.0.0.12	1.8.0_312



Security Bulletin: Vulnerability in Apache MyFaces affects WebSphere Application Server (CVE-2021-26296)

Security Bulletin

Summary

There is a vulnerability in the Apache MyFaces library used by WebSphere Application Server. This has been addressed.

Vulnerability Details

CVEID: CVE-2021-26296
DESCRIPTION: Apache MyFaces is vulnerable to cross-site request forgery, caused by improper validation of user-supplied input. By persuading an authenticated user to visit a malicious Web site, a remote attacker could send a malformed HTTP request to perform unauthorized actions. An attacker could exploit this vulnerability to perform cross-site scripting attacks, Web cache poisoning, and other malicious activities.
CVSS Base score: 8.8
CVSS Temporal Score: See: https://exchange.xforce.ibmcloud.com/vulnerabilities/197017 for the current score.
CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

Affected Products and Versions

Affected Product(s)	Version(s)
WebSphere Application Server Liberty	17.0.0.3 - 21.0.0.3
WebSphere Application Server	9.0
WebSphere Application Server	8.5
WebSphere Application Server	8.0

Document Information

More support for:
WebSphere Application Server

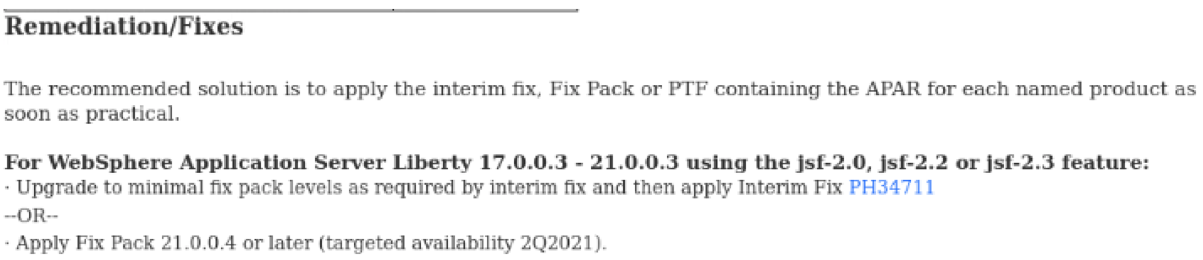
Software version:
Liberty

Operating system(s):
AIX, HP-UX, IBM i, Linux, Solaris, Windows, z/OS, Mac OS

Document number:
6441433

Modified date:
12 April 2021

If you scroll down, you can see how to resolve this.



Remediation/Fixes

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

For WebSphere Application Server Liberty 17.0.0.3 - 21.0.0.3 using the jsf-2.0, jsf-2.2 or jsf-2.3 feature:
· Upgrade to minimal fix pack levels as required by interim fix and then apply Interim Fix PH34711
--OR--
· Apply Fix Pack 21.0.0.4 or later (targeted availability 2Q2021).

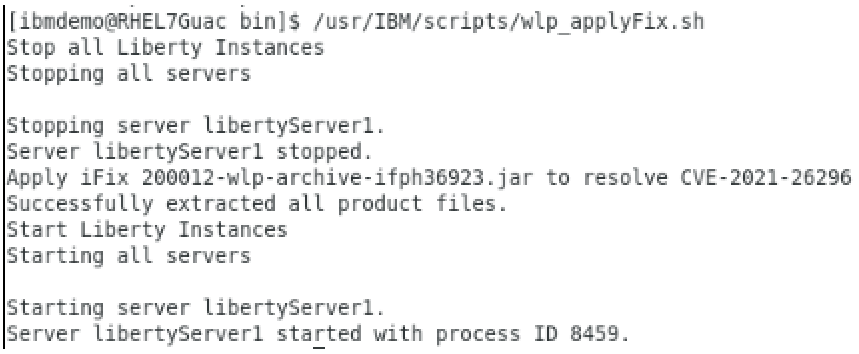
So, let’s apply the related fix.

The related fix has already been downloaded and a small script has been created. The script mainly stops the server, applies the fix and starts the server. Content:

```
echo "Stop server"
/usr/IBM/Liberty/wlp/bin/server stop libertyServer1

echo "Apply iFix 200012-wlp-archive-ifph36923.jar to resolve CVE-2021-
26296"
java -jar /var/IBM/software/WAS/200012-wlp-archive-ifph36923.jar --
installLocation $WLP_DIR/wlp –suppressInfo

echo "Start server"
/usr/IBM/Liberty/wlp/bin/server start libertyServer1
```

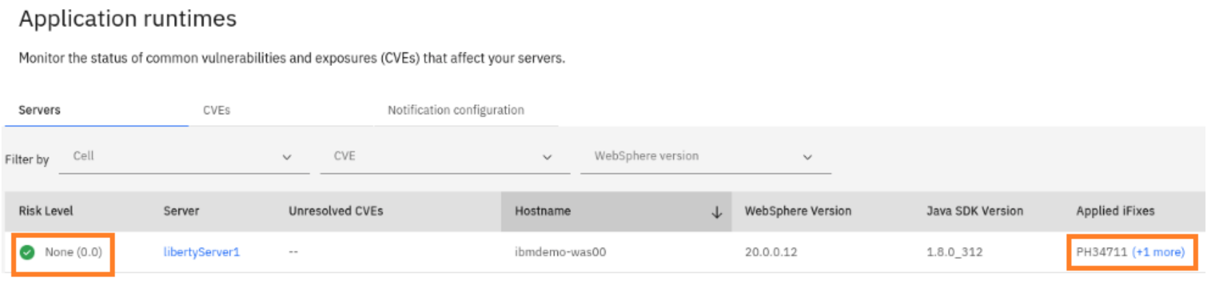
To apply the Liberty fix, go to a command shell and execute the script

```
/usr/IBM/scripts/wlp_applyFix.sh
```



```
[ibmdemo@RHEL7Guac bin]$ /usr/IBM/scripts/wlp_applyFix.sh
Stop all Liberty Instances
Stopping all servers

Stopping server libertyServer1.
Server libertyServer1 stopped.
Apply iFix 200012-wlp-archive-ifph36923.jar to resolve CVE-2021-26296
Successfully extracted all product files.
Start Liberty Instances
Starting all servers

Starting server libertyServer1.
Server libertyServer1 started with process ID 8459.
```

Switch to the WebSphere Automation panel and you should see that the unresolved CVE in your Liberty instance should vanish, in addition you should see the applied fix.



Application runtimes

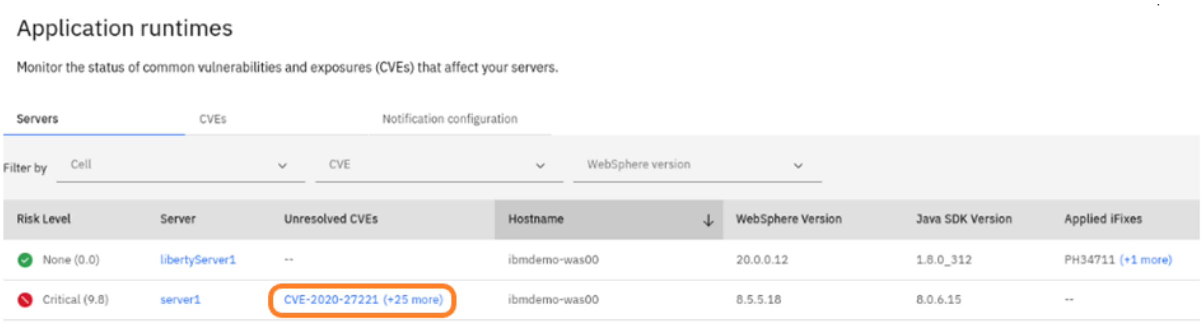
Monitor the status of common vulnerabilities and exposures (CVEs) that affect your servers.

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes
✓ None (0.0)	libertyServer1	--	ibmdemo-was00	20.0.0.12	1.8.0_312	PH34711 (+1 more)

Update tWAS server to fix a vulnerability

In this section, you will apply an iFix to the traditional WebSphere server to remove some vulnerability.

As you can see there are several unresolved vulnerabilities in server1. Click on the highest CVE

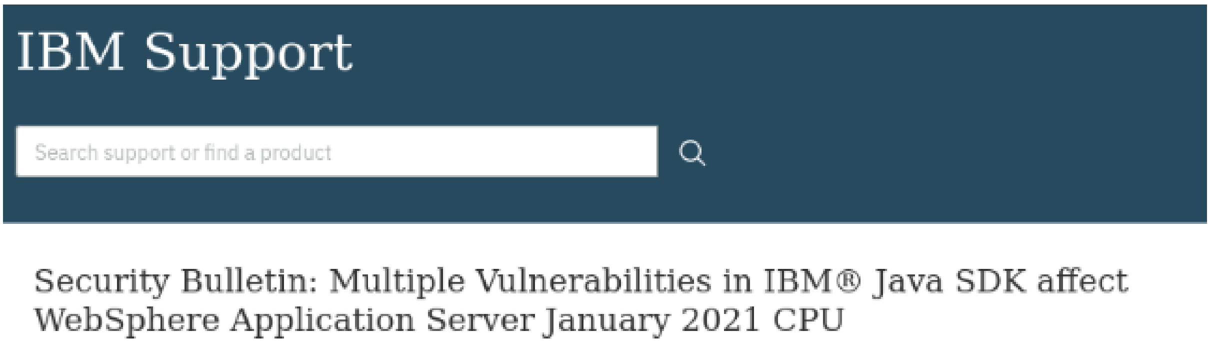


Application runtimes

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

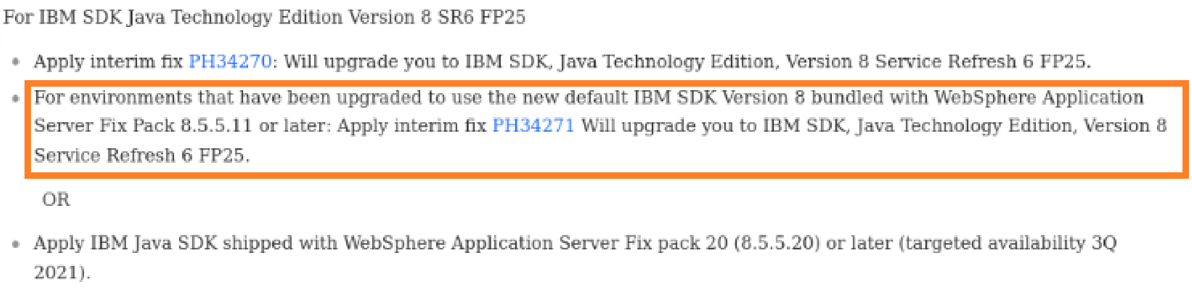
Risk Level	Server	Unresolved CVEs	Hostname ↓	WebSphere Version	Java SDK Version	Applied iFixes
✓ None (0.0)	libertyServer1	--	ibmdemo-was00	20.0.0.12	1.8.0_312	PH34711 (+1 more)
● Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15	--

The security bulletin opens and you can see that the vulnerability is in the IBM Java SDK.



Security Bulletin: Multiple Vulnerabilities in IBM® Java SDK affect WebSphere Application Server January 2021 CPU

By scrolling down, you can see that one option for remediation is to apply iFix PH34271.



For IBM SDK Java Technology Edition Version 8 SR6 FP25

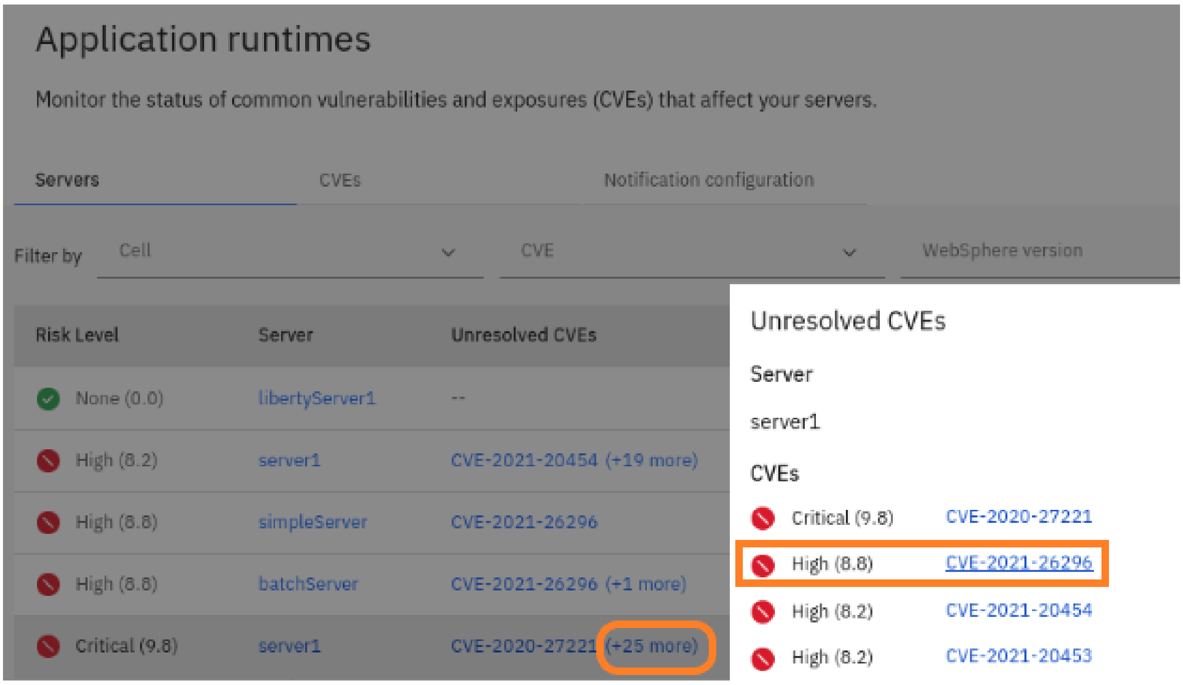
- Apply interim fix PH34270: Will upgrade you to IBM SDK, Java Technology Edition, Version 8 Service Refresh 6 FP25.
- For environments that have been upgraded to use the new default IBM SDK Version 8 bundled with WebSphere Application Server Fix Pack 8.5.5.11 or later: Apply interim fix PH34271 Will upgrade you to IBM SDK, Java Technology Edition, Version 8 Service Refresh 6 FP25.

OR

- Apply IBM Java SDK shipped with WebSphere Application Server Fix pack 20 (8.5.5.20) or later (targeted availability 3Q 2021).

Back on the runtime panel, click on the (+25 more) to see which other CVEs are unresolved.

Click on second highest and you see from the related support page, that it is around Apache MyFaces and that the iFix PH34711 (which has been superseded by PH36923) should resolve it. So let’s apply those fixes to decrease the risk level..



Application runtimes

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

Risk Level	Server	Unresolved CVEs
None (0.0)	libertyServer1	--
High (8.2)	server1	CVE-2021-20454 (+19 more)
High (8.8)	simpleServer	CVE-2021-26296
High (8.8)	batchServer	CVE-2021-26296 (+1 more)
Critical (9.8)	server1	CVE-2020-27221 (+25 more)

Unresolved CVEs

Server

server1

CVEs

Critical (9.8)	CVE-2020-27221
High (8.8)	CVE-2021-26296
High (8.2)	CVE-2021-20454
High (8.2)	CVE-2021-20453

The fixes have already been downloaded and a script has been created.
Content of the script:

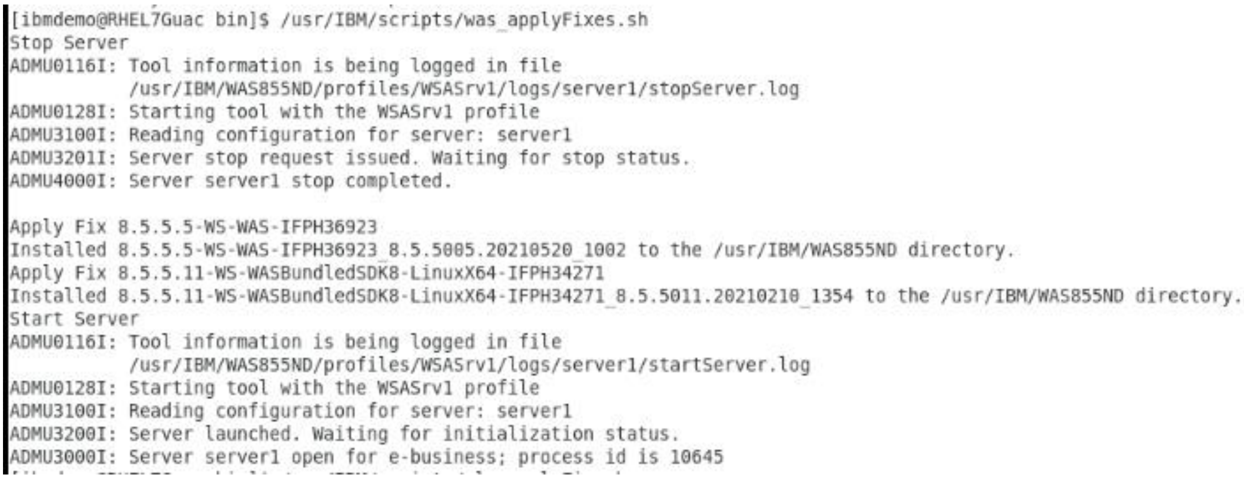
```
/usr/IBM/WAS855ND/profiles/WSASrv1/bin/stopServer.sh server1
export fixID="8.5.5.5-WS-WAS-IFPH36923"
export fixRepo="8.5.5.5-ws-was-ifph36923.zip"
export WAS855ND_HOME="/usr/IBM/WAS855ND"
export WAS855ND_PROFILE="$WAS855ND_HOME/profiles/WSASrv1"
export IMCL_HOME="/usr/IBM/IM/eclipse/tools"
echo "Stop Server"
$WAS855ND_PROFILE/bin/stopServer.sh server1
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 server1
```

The script basically stops the tWAS instance, uses IBM Installation Manager to apply a fix for WAS and a fix for the IBM Java SDK and then starts the tWAS instance again.

To apply the tWAS fix, go to a command shell and execute the script

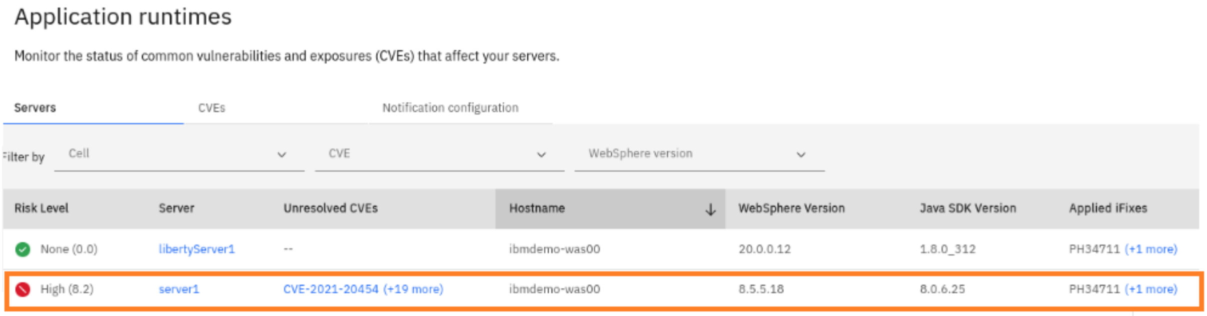
```
/usr/IBM/scripts/was_applyFixes.sh
```

This might take some minutes but finally you should see something like



```
[ibmdemo@RHEL7Guac bin]$ /usr/IBM/scripts/was_applyFixes.sh
Stop Server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrv1/logs/server1/stopServer.log
ADMU0128I: Starting tool with the WSASrv1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.

Apply Fix 8.5.5.5-WS-WAS-IFPH36923
Installed 8.5.5.5-WS-WAS-IFPH36923_8.5.5005.20210520_1002 to the /usr/IBM/WAS855ND directory.
Apply Fix 8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271
Installed 8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271_8.5.5011.20210210_1354 to the /usr/IBM/WAS855ND directory.
Start Server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrv1/logs/server1/startServer.log
ADMU0128I: Starting tool with the WSASrv1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 10645
```

Switch to the browser tab with the WebSphere Automation Runtime panel and you should see that the fixes have been applied and that the risk level dropped as expected from 9.8 (Critical) to 8.2 (High).



Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname ↓	WebSphere Version	Java SDK Version	Applied iFixes
✓ None (0.0)	libertyServer1	--	ibmdemo-was00	20.0.0.12	1.8.0_312	PH34711 (+1 more)
⛔ High (8.2)	server1	CVE-2021-20454 (+19 more)	ibmdemo-was00	8.5.5.18	8.0.6.25	PH34711 (+1 more)

Update the Liberty server to re-introduce a vulnerability

The same as you can apply fixes to resolve a vulnerability, the same an uninstall can re-introduce an issue.

If there is time, you can use the following script to uninstall an iFix from Liberty and see how the security vulnerability gets back into WebSphere Automation.

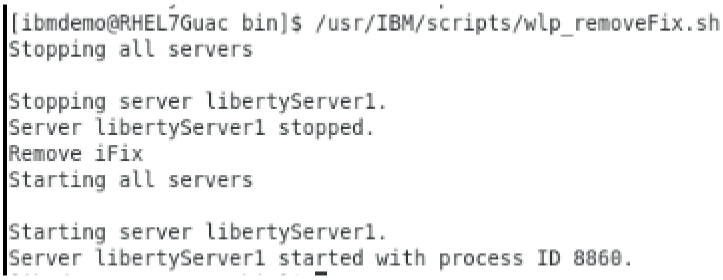
Content of the script:

```
echo "Stop server"
/usr/IBM/Liberty/wlp/bin/server stop libertyServer1
echo "Remove iFix"
rm /usr/IBM/Liberty/wlp/lib/com.ibm.ws.jsf.2.2_1.0.47.cl201220210331-
1851.jar
rm /usr/IBM/Liberty/wlp/lib/com.ibm.ws.jsf_1.0.47.cl201220210514-
1702.jar
rm
/usr/IBM/Liberty/wlp/lib/com.ibm.ws.org.apache.myfaces.2.3_1.0.47.cl2012
20210331-1851.jar
rm /usr/IBM/Liberty/wlp/lib/fixes/200012-wlp-archive-
IFPH36923_20.0.0012.20210514_2128.xml
rm /usr/IBM/Liberty/wlp/lib/fixes/200012-wlp-archive-
IFPH36923_20.0.0012.20210514_2128.lpmf
echo "Start server"
/usr/IBM/Liberty/wlp/bin/server start libertyServer1
```

The script basically stops the Liberty instance, uninstalls the fixes and then starts the Liberty instance again.

To remove the Liberty fix, go to a command shell and execute the script

```
/usr/IBM/scripts/wlp_removeFix.sh
```

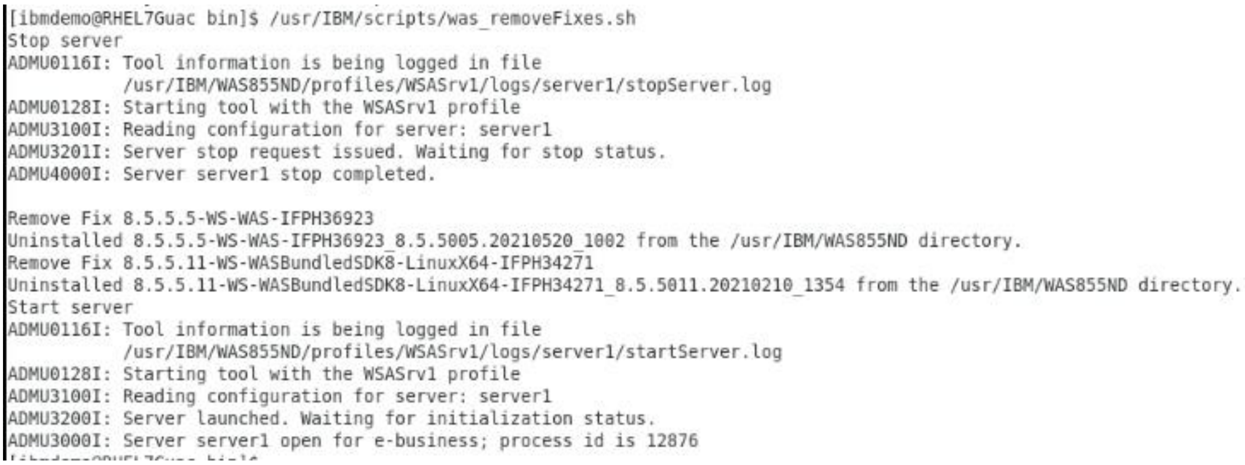


```
[ibmdemo@RHEL7Guac bin]$ /usr/IBM/scripts/wlp_removeFix.sh
Stopping all servers

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

Starting server libertyServer1.
Server libertyServer1 started with process ID 8860.
```

To remove the tWAS fix, go to a command shell and execute the script

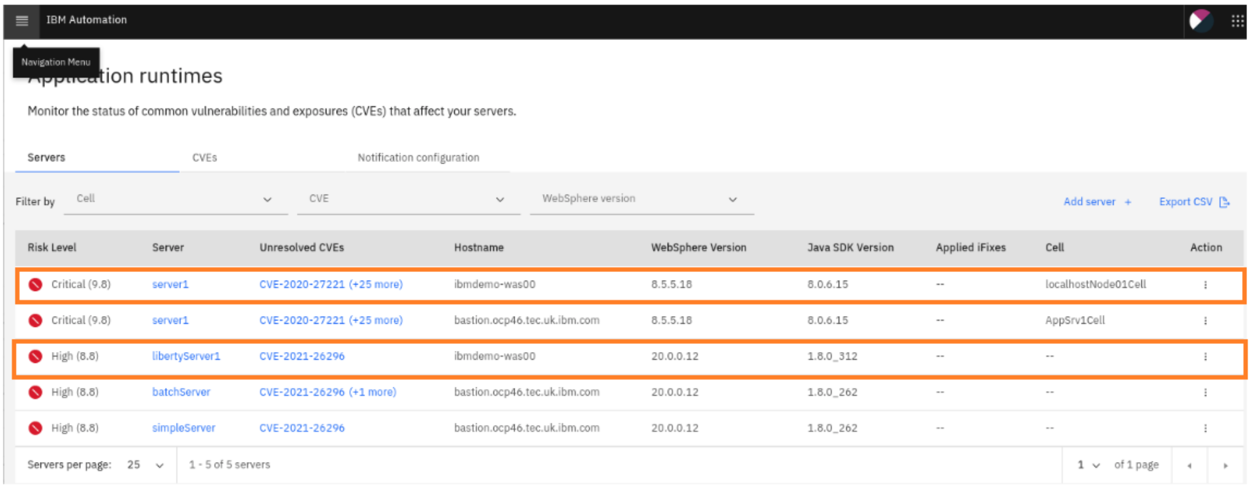
```
/usr/IBM/scripts/was_removeFixes.sh
```

Remove fix from Liberty



```
[ibmdemo@RHEL7Guac bin]$ /usr/IBM/scripts/was_removeFixes.sh
Stop server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrv1/logs/server1/stopServer.log
ADMU0128I: Starting tool with the WSASrv1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.

Remove Fix 8.5.5.5-WS-WAS-IFPH36923
Uninstalled 8.5.5.5-WS-WAS-IFPH36923_8.5.5005.20210520_1002 from the /usr/IBM/WAS855ND directory.
Remove Fix 8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271
Uninstalled 8.5.5.11-WS-WASBundledSDK8-LinuxX64-IFPH34271_8.5.5011.20210210_1354 from the /usr/IBM/WAS855ND directory.
Start server
ADMU0116I: Tool information is being logged in file
           /usr/IBM/WAS855ND/profiles/WSASrv1/logs/server1/startServer.log
ADMU0128I: Starting tool with the WSASrv1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 12876
```

Finally, your WebSphere Automation panel should look as the initial one after registering the servers.



Application runtimes

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

Risk Level	Server	Unresolved CVEs	Hostname	WebSphere Version	Java SDK Version	Applied iFixes	Cell	Action
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	ibmdemo-was00	8.5.5.18	8.0.6.15	--	localhostNode01Cell	⋮
Critical (9.8)	server1	CVE-2020-27221 (+25 more)	bastion.ocp46.tec.uk.ibm.com	8.5.5.18	8.0.6.15	--	AppSrv1Cell	⋮
High (8.8)	libertyServer1	CVE-2021-26296	ibmdemo-was00	20.0.0.12	1.8.0_312	--	--	⋮
High (8.8)	batchServer	CVE-2021-26296 (+1 more)	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮
High (8.8)	simpleServer	CVE-2021-26296	bastion.ocp46.tec.uk.ibm.com	20.0.0.12	1.8.0_262	--	--	⋮

Summary

Congratulations! You have completed the WebSphere Automation lab.

With automated tooling and insights, IBM WebSphere Automation enables teams to modernize and secure IT estates, adapt and respond to incidents efficiently, and optimize WebSphere operations. WebSphere system operators and administrators can reduce the cost, effort, and risk of addressing vulnerabilities, automate critical activities, and preserve uptime with early detection, notification, and remediation of incidents.

IBM WebSphere Automation helps teams remove manual toil to work less on maintenance tasks and more on strategic activities, while unlocking new value, extending the life, and increasing ROI of WebSphere investments.

IBM WebSphere Automation is part of IBM Automation, a set of shared automation services that help you get insight into how your processes run, visualize hotspots and bottlenecks, and use financial impact information to prioritize which issues to address first.

To learn more about IBM WebSphere Automation, visit ibm.com/cloud/websphere-automation.