Quick Start

Scalable Deployers in SDL Web 8.5

Feb 2017 – SDL Web

Document owner: Richard Hamlyn (rhamlyn@sdl.com)

Contents

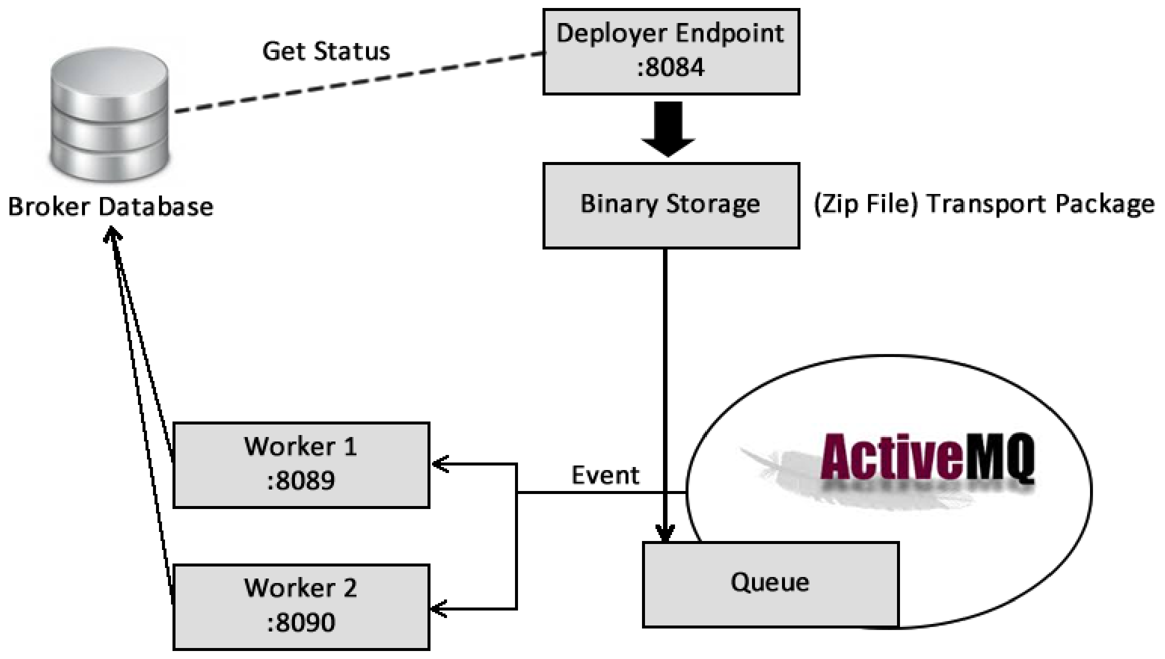
Scalable Deployment

Information

SDL Web 8.5 provides the option to optimize content deployment by scaling multiple Deployers. This guide will show you how to apply this method to publish your content, using multiple Deployers, to the same Broker Database.

Overview

The following diagram is an overview of the architecture that this guide will implement. This model can be extended for up to four Deployers; however, SDL does not advise the implementation of more than four deployers.



Advisory note:
You should not deploy more than 4 workers!

Steps:

1. Content is passed to the Deployer after a user Publishes and item in the CME.
2. The Deployer Endpoint passes the Transport Package (zip file) to the defined Binary Storage (File System or Redis Database).
3. The Deployer Endpoint also passes the item to the Queue in ActiveMQ (JMS).
4. ActiveMQ triggers an event that informs the Worker Deployers that a new package has been received.

5. The first available Worker Deployer picks up the job from the Queue and contacts the Binary Storage to get the respective Transport Package.
6. After rendering the Transport Package the Worker Deployer passes the item to the Broker Database.
7. The Deployer then gets the status of the job from the Broker Database, which is updated by the Worker Deployer responsible for that job.

Pre-requisites

The following pre-requisites need to be installed/implemented to provide the optimum solution for scaled out Deployers.

SDL Web 8.5 Content Manager and Content Delivery
A suitable Topology and at least a Discovery Service
Redis Database and Redis Desktop Manager installed and running (Annex A)
ActiveMQ installed and running (Annex B)

Note! It is possible to setup scaled out Deployers without using these tool (by using the File System). However, this guide assumes both will be employed.

Installation

Microservices. In order to employ scalable deployment technology with SDL Web 8 you need to have the following services installed:

- Endpoint (Deployer)
- Worker 1
- Worker 2
- Worker 3 (optional)
- Worker 4 (optional)

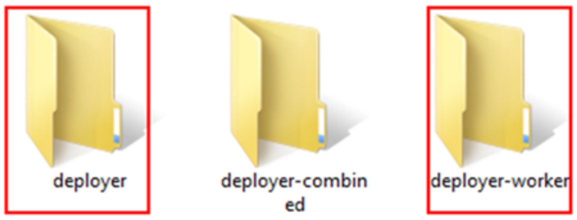
This guide will demonstrate how to install a single Endpoint and two Workers. The procedure for this is in three phases:

Prepare installation directories
Configure each service
Install the Microservices

Step 1: Prepare installation directories

1. From the Installation Media for SDL Web 8.5, install the following services:

   Installation Media/Content Delivery/roles/deployer

   Note! The deployer-combination service is used in a non-scalable environment where the Endpoint and Worker are combined in a single Microservice.

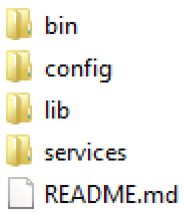
   Deployer – Endpoint
   Deployer-worker - Worker

2. Copy the contents of the **deployer\standalone** folder to a suitable location in your server i.e. **C:\SDL\Web\deployers\endpoint**

3. Copy the contents of the **deployer-worker\standalone** to your service's location on your server i.e. **C:\SDL\Web\deployers\worker1**

4. Repeat step 4 for your second worker to **C:\SDL\Web\deployers\worker2**

5. You now have each service's files in place ready for you to setup with your service directories looking like the below



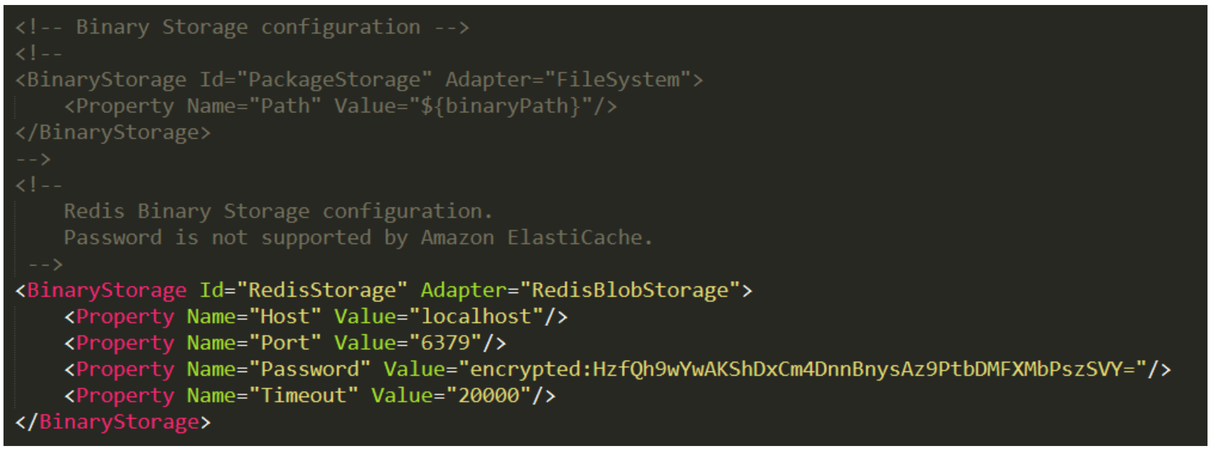
```
bin
config
lib
services
README.md
```

Step 2: Configure each of the three services

1. Each service requires you update the **deployer-conf.xml** and the worker services require you to also update the **cd_storage.xml**, both of which are located **\config** directory of the service.

2. For the deployer-endpoint service open the **deployer-conf.xml** found in:
C:\SDL\Web\deployers \deployer-endpoint\conf\

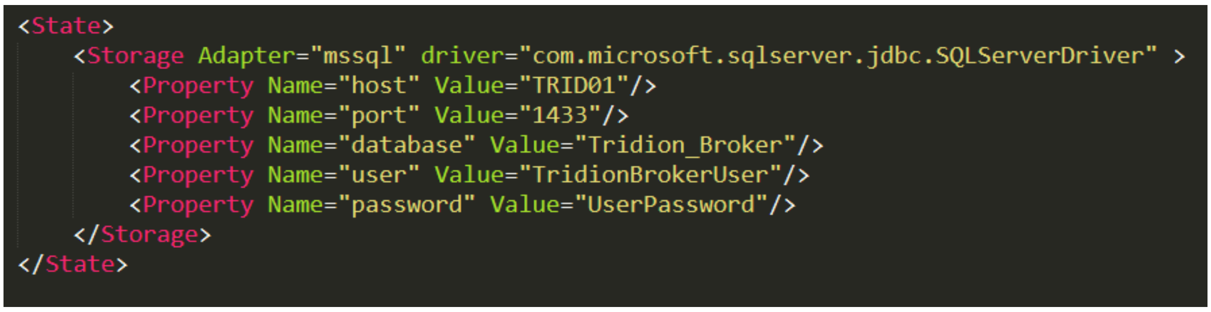
3. Update the **<BinaryStorage>** node in the configuration file to point to your **Redis** data store as shown below:



```
<!-- Binary Storage configuration -->
<!--
<BinaryStorage Id="PackageStorage" Adapter="FileSystem">
    <Property Name="Path" Value="${binaryPath}"/>
</BinaryStorage>
-->
<!--
    Redis Binary Storage configuration.
    Password is not supported by Amazon ElastiCache.
 -->
<BinaryStorage Id="RedisStorage" Adapter="RedisBlobStorage">
    <Property Name="Host" Value="localhost"/>
    <Property Name="Port" Value="6379"/>
    <Property Name="Password" Value="encrypted:HzfQh9wYwAKShDxCm4DnnBnysAz9PtbDMFXMbPszSVY="/>
    <Property Name="Timeout" Value="20000"/>
</BinaryStorage>
```

Note! You could also choose to you use the file system to store the Binary Transport Package by uncommenting the <BinaryStorage ID="PackageStorage" Adapter="FileSystem" entry. However, this is not recommended in a scaled out deployment setup.

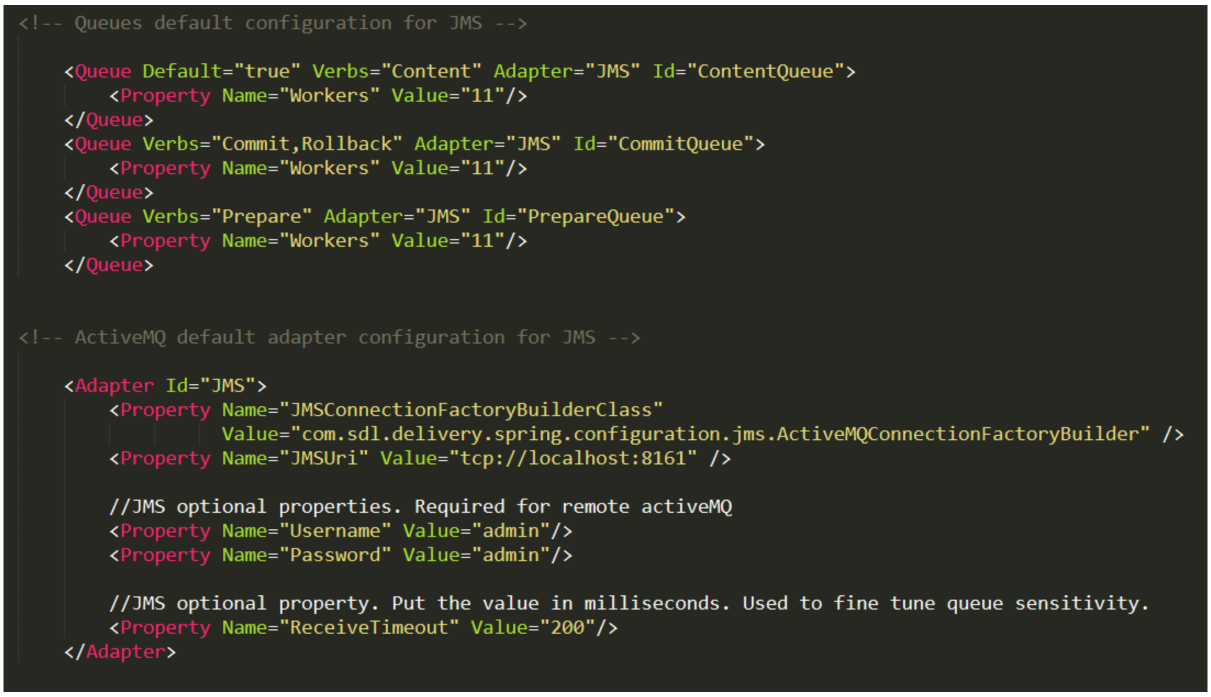
4. Make sure you have only **one** **<State>** defined as shown:
This is so the workers know where to update the status of a job and the Endpoint knows where to get the status of each job.



```
<State>
    <Storage Adapter="mssql" driver="com.microsoft.sqlserver.jdbc.SQLServerDriver" >
        <Property Name="host" Value="TRID01"/>
        <Property Name="port" Value="1433"/>
        <Property Name="database" Value="Tridion_Broker"/>
        <Property Name="user" Value="TridionBrokerUser"/>
        <Property Name="password" Value="UserPassword"/>
    </Storage>
</State>
```

Note! This assumes you are using an SQL Database for your Broker. You will need your own Broker details for host, port, database, user and password.

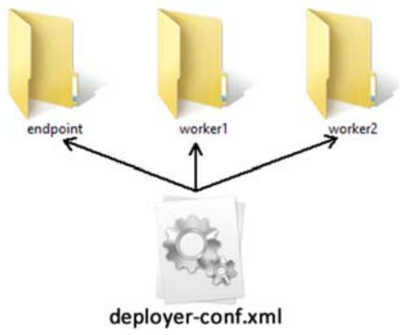
5. Ensure all filesystem **<Queue>** entries (uncommented by default) are either deleted or commented out. The following shows how you should configure the deployer to use Active MQ (JMS):


```
<!-- Queues default configuration for JMS -->

    <Queue Default="true" Verbs="Content" Adapter="JMS" Id="ContentQueue">
        <Property Name="Workers" Value="11"/>
    </Queue>
    <Queue Verbs="Commit,Rollback" Adapter="JMS" Id="CommitQueue">
        <Property Name="Workers" Value="11"/>
    </Queue>
    <Queue Verbs="Prepare" Adapter="JMS" Id="PrepareQueue">
        <Property Name="Workers" Value="11"/>
    </Queue>


<!-- ActiveMQ default adapter configuration for JMS -->

    <Adapter Id="JMS">
        <Property Name="JMSConnectionFactoryBuilderClass"
                  Value="com.sdl.delivery.spring.configuration.jms.ActiveMQConnectionFactoryBuilder" />
        <Property Name="JMSUri" Value="tcp://localhost:8161" />

        //JMS optional properties. Required for remote activeMQ
        <Property Name="Username" Value="admin"/>
        <Property Name="Password" Value="admin"/>

        //JMS optional property. Put the value in milliseconds. Used to fine tune queue sensitivity.
        <Property Name="ReceiveTimeout" Value="200"/>
    </Adapter>
```

Note! The Queue entries shown are the only Queue entries required! You can remove or comment out all other <Queue> entries.

6. Save and close the **deployer-conf.xml**.

7. You will use the exact same configuration in the two worker services. Copy the **deployer-conf.xml** and paste it in to the **config** directory of Worker1 and Worker2.

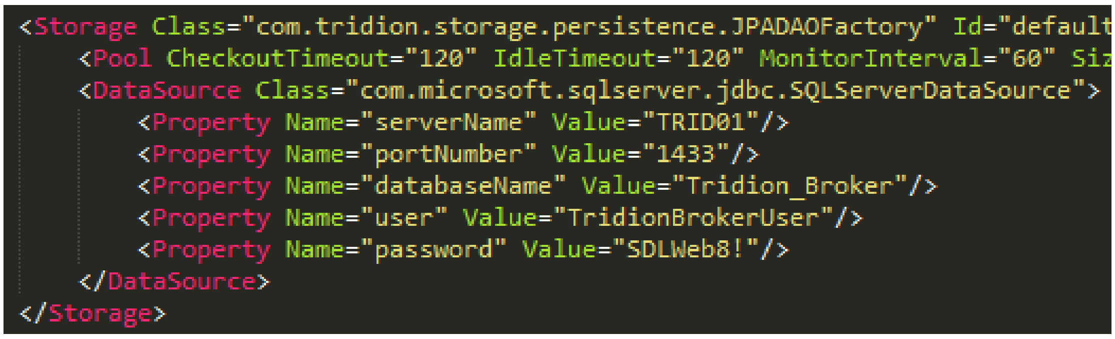


Note!

The deployer-conf.xml lives in the config directory of each service

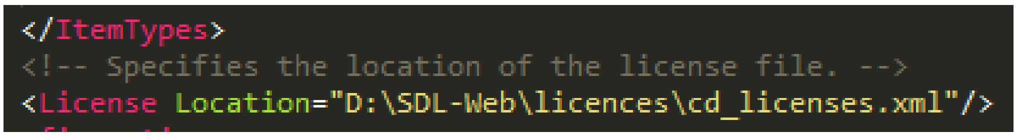
8. Now update the **cd_storage_conf.xml** for both Workers (not the endpoint). Navigate to **C:\SDL\Web\deployers\worker1\conf** and open the **cd_storage_conf.xml**.

9. Update the **<storages>** node so that it contains only one entry that points to your target. In this case both workers will deploy to the same Broker database; however, it is possible to change this should you need to:



```
<Storage Class="com.tridion.storage.persistence.JPADAOFactory" Id="default
    <Pool CheckoutTimeout="120" IdleTimeout="120" MonitorInterval="60" Siz
    <DataSource Class="com.microsoft.sqlserver.jdbc.SQLServerDataSource">
        <Property Name="serverName" Value="TRID01"/>
        <Property Name="portNumber" Value="1433"/>
        <Property Name="databaseName" Value="Tridion_Broker"/>
        <Property Name="user" Value="TridionBrokerUser"/>
        <Property Name="password" Value="SDLWeb8!"/>
    </DataSource>
</Storage>
```

10. If you do not have your **cd_licenses.xml** (Licence file for Content Delivery) in your service's **config** directory then you also need to have the following entry in the **cd_storage_conf.xml**.



```
</ItemTypes>
<!-- Specifies the location of the license file. -->
<License Location="D:\SDL-Web\licences\cd_licenses.xml"/>
```

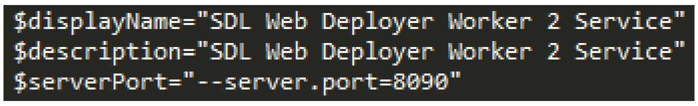
Note! Where D:\SDL-Web\licenses\ is the central location for your Content Delivery licences.

11. Save the **cd_storage_conf.xml** and copy/paste the exact same file in to config directory of the Worker2 service.

    Note! If you are not using the default host:port for the Discovery Service then you would need to update this in **cd_storage_conf.xml** of the Endpoint service. In this guide we are using the default and the Discovery service is at http://localhost:8082/discovery.svc. So you do not need to do any changes to the **cd_storage_conf.xml for** the Endpoint service. We also have the cd_licenses.xml file in the Endpoint's config directory, so the service will check this location by default.

12. Preparing for installation: For the **Endpoint** and **Worker1** services we can simply use the default installation. However, **Worker2** must be updated as we are installing it on the same server as **Worker1** so it needs a different service name and port.

13. Navigate to **Worker2/bin** and open the **installService.ps1** in an editor, such as Notepad++. Update the following as shown and then save and close the file.



```
$displayName="SDL Web Deployer Worker 2 Service"
$description="SDL Web Deployer Worker 2 Service"
$serverPort="--server.port=8090"
```

Note! If you encounter any conflicts with default ports (shown below) then you can also update the installService.ps1 for the Endpoint and Worker1 to ensure they use a free port. In the following steps you will see that the Worker 1 service name was also updated, but this is not necessary.

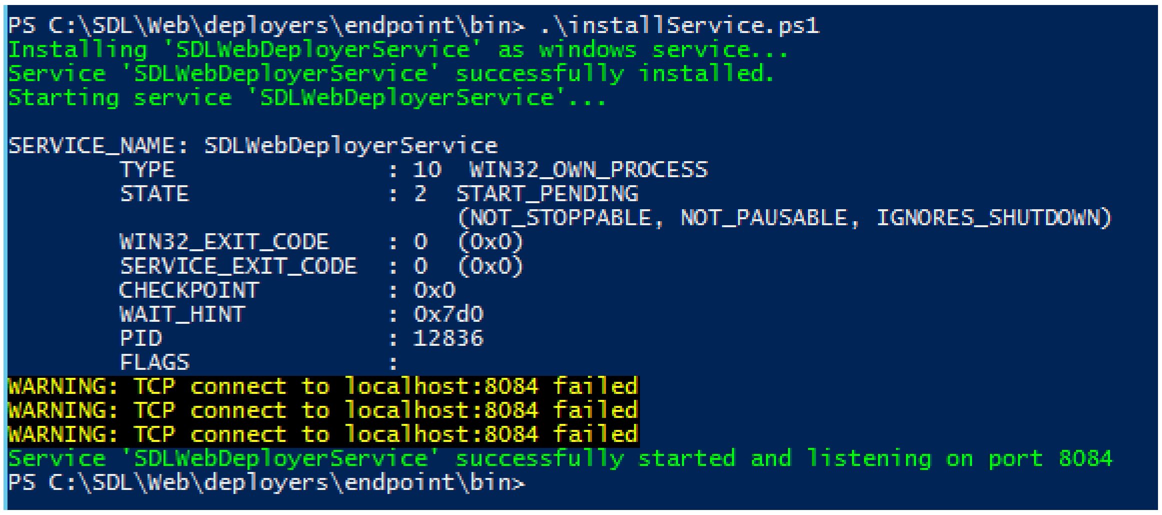
Endpoint: **http//localhost:8084** (default)
Worker 1: **http://localhost:8089** (default)
Worker 2: **http://localhost:8090** (updated)

Step 3: Installing the Microservices

1. Navigate to the **Endpoint/bin** directory where the **installService.ps1** exists.

2. In the address bar of the folder, type **PowerShell** and press enter. PowerShell will load with the command line set to the correct directory.

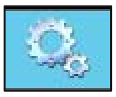
3. Enter the following command to run the installation script **.\installService.ps1**.
   a. After a few attempts to connect to the Endpoint service you should a success message as shown below. If you do not get a success message then you need to check your log files to find the reason (check the logback.xml file in the config directory).

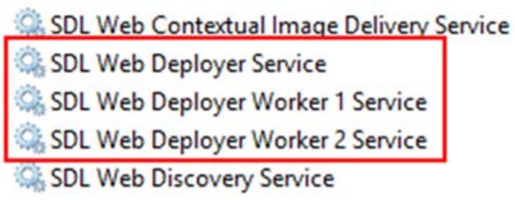


```
PS C:\SDL\Web\deployers\endpoint\bin> .\installService.ps1
Installing 'SDLWebDeployerService' as windows service...
Service 'SDLWebDeployerService' successfully installed.
Starting service 'SDLWebDeployerService'...

SERVICE_NAME: SDLWebDeployerService
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 12836
        FLAGS              :
WARNING: TCP connect to localhost:8084 failed
WARNING: TCP connect to localhost:8084 failed
WARNING: TCP connect to localhost:8084 failed
Service 'SDLWebDeployerService' successfully started and listening on port 8084
PS C:\SDL\Web\deployers\endpoint\bin>
```

4. Now perform the exact same steps for **Worker1** and **Worker2**.

5. Once you have successfully installed all three services then open the Services MMC to see the following services:


Services


SDL Web Contextual Image Delivery Service
SDL Web Deployer Service
SDL Web Deployer Worker 1 Service
SDL Web Deployer Worker 2 Service
SDL Web Discovery Service

6. You now have all your services configured and running.

7. The final step is to register your Deployer (Endpoint) with your Discovery Service.

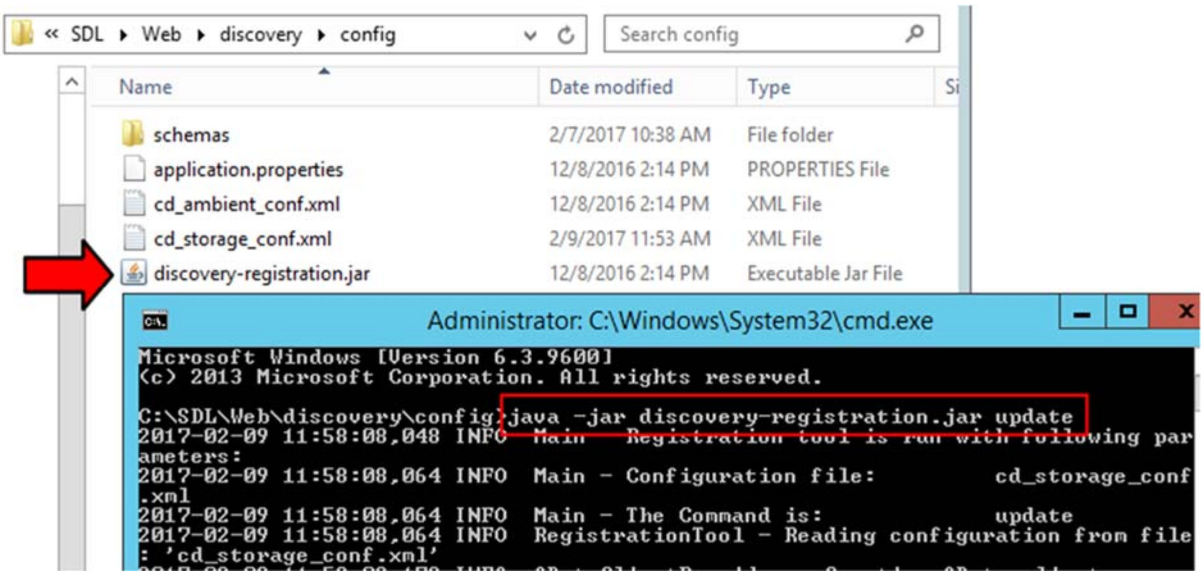
8. Navigate to your **Discovery** service's **config** directory. Open the **cd_storage_conf.xml** and ensure the following entry is added to the <roles> node to reflect your Endpoint service:

```
<Role Name="DeployerCapability" Strategy="DEFAULT" Url="http://localhost:8084/httpupload">
    <Property Name="undo.enabled" Value="false"/>
    <Property Name="encoding" Value="UTF-8"/>
</Role>
```

Note! The above Capability entry assumes you used the default host and port.

9. Save and close the **cd_storage_conf.xml.**

10. From the Discovery service's config directory open the command line and run the registration tool as shown.



Name	Date modified	Type
schemas	2/7/2017 10:38 AM	File folder
application.properties	12/8/2016 2:14 PM	PROPERTIES File
cd_ambient_conf.xml	12/8/2016 2:14 PM	XML File
cd_storage_conf.xml	2/9/2017 11:53 AM	XML File
discovery-registration.jar	12/8/2016 2:14 PM	Executable Jar File

Administrator: C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\SDL\Web\discovery\config>java -jar discovery-registration.jar update
2017-02-09 11:58:08,048 INFO  Main - Registration tool is run with following par
ameters:
2017-02-09 11:58:08,064 INFO  Main - Configuration file:        cd_storage_conf
.xml
2017-02-09 11:58:08,064 INFO  Main - The Command is:            update
2017-02-09 11:58:08,064 INFO  RegistrationTool - Reading configuration from file
: 'cd_storage_conf.xml'
```

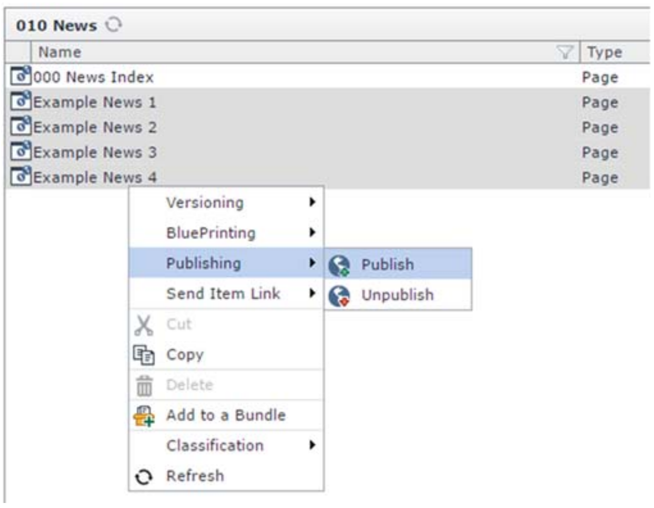
Note! Run the command: **java –jar discovery-registration.jar update**

Note! The registration tool can be found in the SDL Web 8.5 installation media:
Content Delivery/roles/discovery/registration

Testing

You can now test your scaled out Deployers are working.

1. Open your Content Manager Explorer and navigate to a Structure Group that has several Pages that are Publishable.



010 News

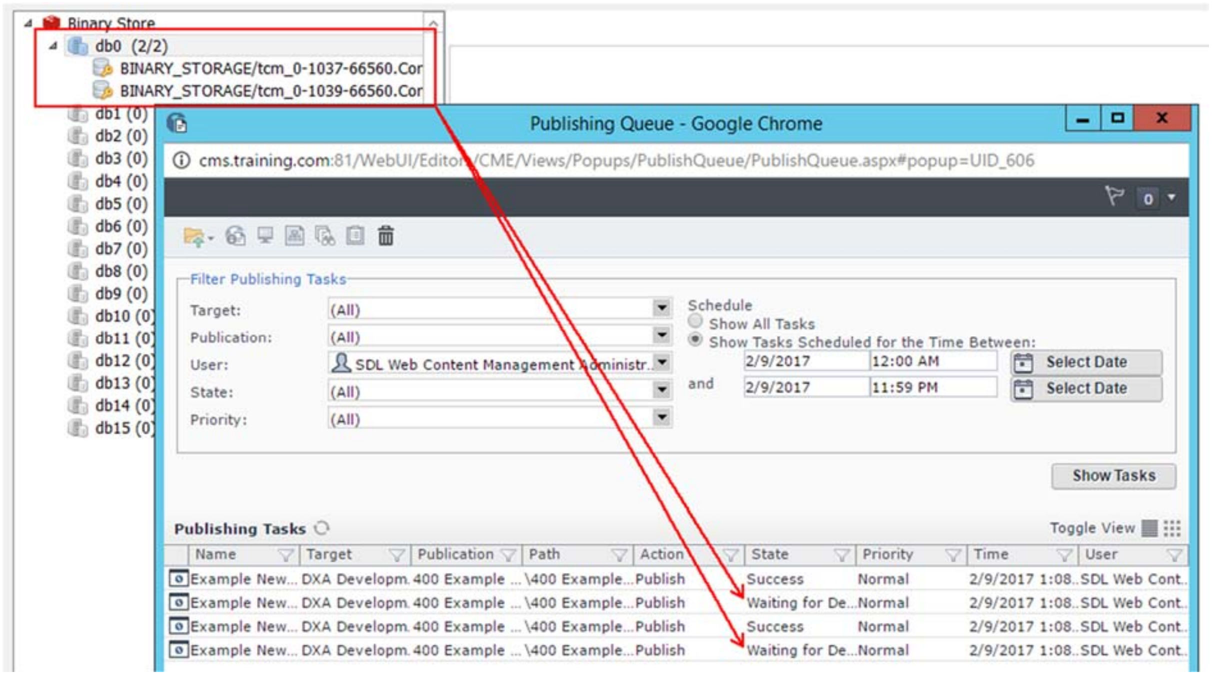
Name	Type
000 News Index	Page
Example News 1	Page
Example News 2	Page
Example News 3	Page
Example News 4	Page

Note!
Instruction on installing Redis Database and Redis Desktop Manager can be found at Annex B of this Guide.

2. Now open Redis Desktop Manager and Connect to your Redis Server.

3. Right-click on the Binary Store and select Reload.

4. Expand the database node that shows it has items and you will see the Binary items (Transport Packages (zip files) that were sent to Redis awaiting collection from one of the Workers.



Binary Store
db0 (2/2)
BINARY_STORAGE/tcm_0-1037-66560.Cor
BINARY_STORAGE/tcm_0-1039-66560.Cor
db1 (0)
db2 (0)
db3 (0)
db4 (0)
db5 (0)
db6 (0)
db7 (0)
db8 (0)
db9 (0)
db10 (0)
db11 (0)
db12 (0)
db13 (0)
db14 (0)
db15 (0)

Publishing Queue - Google Chrome

cms.training.com:81/WebUI/Editors/CME/Views/Popups/PublishQueue/PublishQueue.aspx#popup=UID_606

Filter Publishing Tasks

Target: (All)
Publication: (All)
User: SDL Web Content Management Administr...
State: (All)
Priority: (All)

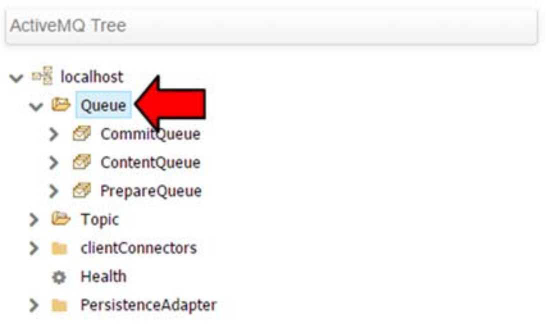
Schedule
Show All Tasks
Show Tasks Scheduled for the Time Between:
2/9/2017 12:00 AM
and 2/9/2017 11:59 PM

Publishing Tasks

Name	Target	Publication	Path	Action	State	Priority	Time	User
Example New...	DXA Developm.	400 Example ...	\400 Example...	Publish	Success	Normal	2/9/2017 1:08..	SDL Web Cont.
Example New...	DXA Developm.	400 Example ...	\400 Example...	Publish	Waiting for De...	Normal	2/9/2017 1:08..	SDL Web Cont.
Example New...	DXA Developm.	400 Example ...	\400 Example...	Publish	Success	Normal	2/9/2017 1:08..	SDL Web Cont.
Example New...	DXA Developm.	400 Example ...	\400 Example...	Publish	Waiting for De...	Normal	2/9/2017 1:08..	SDL Web Cont.

Note! In the example above, two of the items have been collected by the Workers (Success) and two are in the database awaiting processing.

5. Now open ActiveMQ (http://localhost:8161)
   a. Open Queue and change the view to Diagram
   b. You should see your items in the Queue



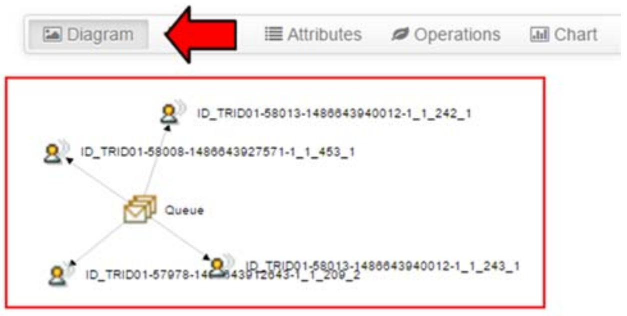


Diagram
Attributes
Operations
Chart
ID_TRID01-58013-1486643940012-1_1_242_1
ID_TRID01-58008-1486643927571-1_1_453_1
Queue
ID_TRID01-57978-1486643912643-1_1_209_2
ID_TRID01-58013-1486643940012-1_1_243_1

Note! If you do not see items in the Queue it could be that you were too late in opening things up and everything has been processed. If this is the case then simply repeat the Publishing of the four Pages.

Note! ActiveMQ will inform the Workers that it has Queue items to be collected. The Workers will collect a Queue item and then collect the Binary Package for deployment from the Redis database. It will then deploy the Package to the Broker database.

Note!
Instruction on installing ActiveMQ can be found at Annex A of this Guide.

Annex A: Installing ActiveMQ

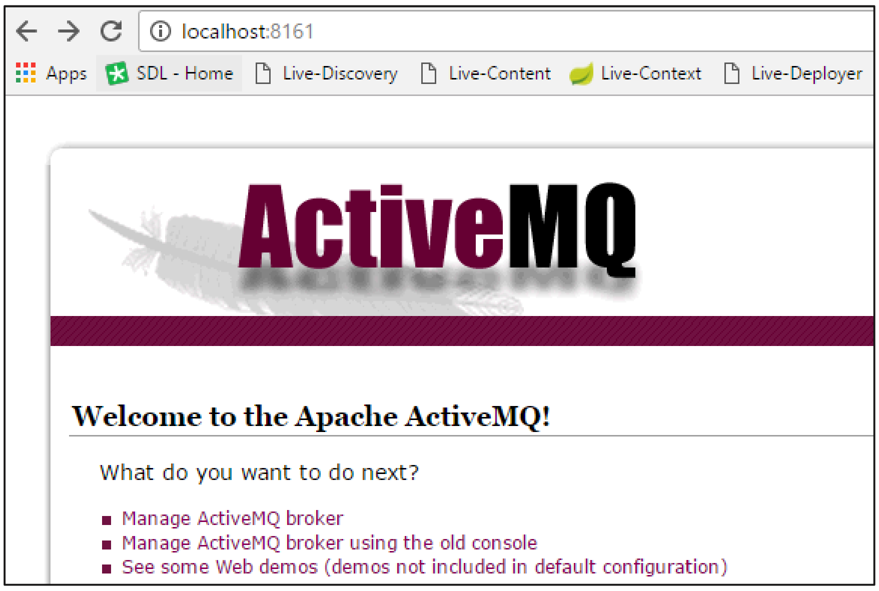
ActiveMQ is a message broker that implements the JMS API. It allows you to create listeners and event actions. In the case of implementing this for Scalable Deployers in SDL Web, ActiveMQ receives the message that a Binary item has been Published to the Binary Storage (Redis database or File System). This in turn triggers an event to inform the Worker Deployers that there is a package to be collected and deployed.

Installation

Version 5.9.0

Download: http://activemq.apache.org/activemq-590-release.html

1. Download ActiveMQ from the link above.
   (you need the windows zip apache-activemq-5.9.0-bin.zip).
2. Unzip the package and place it in a suitable location on your server, such as **d:\activemq**.
3. From within the installation, navigate to d:\apachemq\bin\win64 (win 32 if applicable to your server).
4. In the directory address bar enter **cmd** and press enter.
5. The Command Line Interface will open.
6. Enter the following command: **activemq start**.
7. Open your browser and navigate to the Admin Console at http://localhost:8161

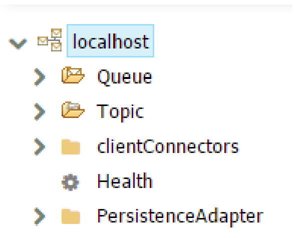


Welcome to the Apache ActiveMQ!

What do you want to do next?

- Manage ActiveMQ broker
- Manage ActiveMQ broker using the old console
- See some Web demos (demos not included in default configuration)

8. Now click on Manage ActiveMQ broker and you will see the following:



Annex B: Installing Redis Database

Redis is an in-memory data structure store that is used as a database, cache and message broker. SDL Web 8.5 uses the Redis store for caching (Client-side and Server-side) but also as the message broker that receives the Binary item when a user in the Content Manager Explorer Publishes an item. This allows you to employ 2 or more (maximum of 4) Worker Deployers to collect and Publish the Binary Package to the destination. This allows you to spread the work for deployment across several workers.

Note!
Using Redis for Caching is covered in a separate Quick Start Guid

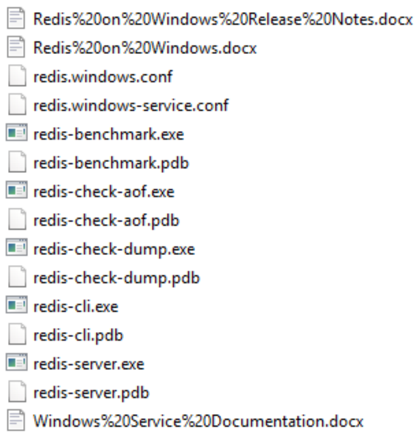
Installation

Redis Version 3.0
Redis Desktop Manager Version 0.8.8

Redis Download**:** https://github.com/ServiceStack/redis-windows/blob/master/downloads/redis64-3.0.501.zip

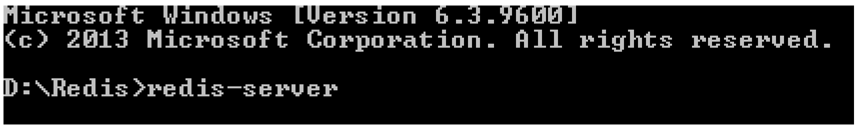
Redis Desktop Manager Download: https://redisdesktop.com/

1. Download the Redis installation package from the link above. The unzipped package will look like this:



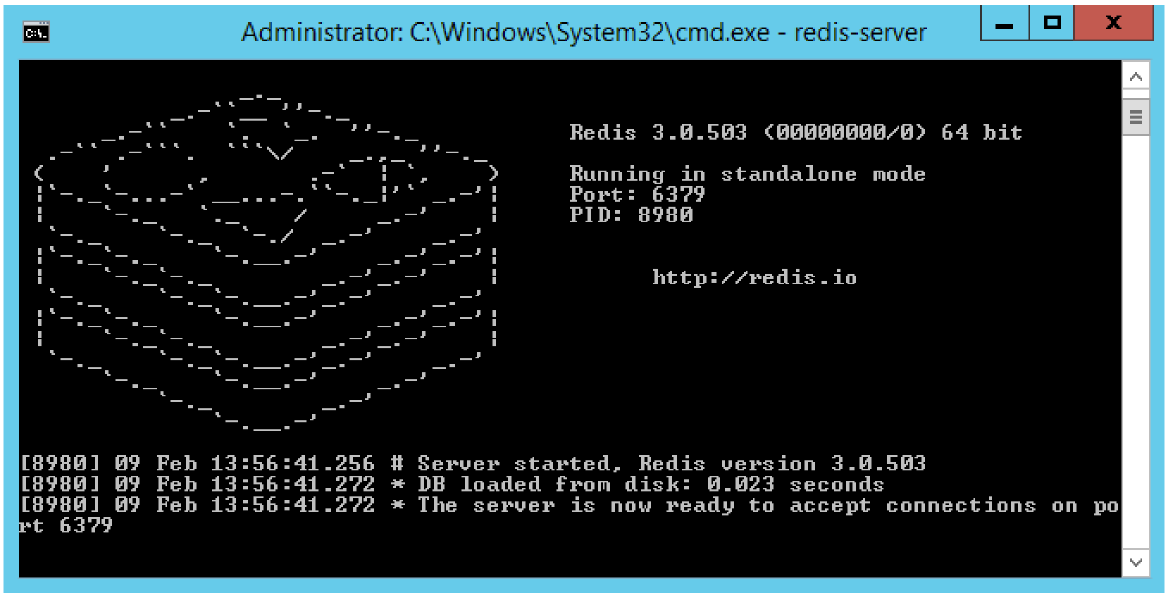
```
Redis%20on%20Windows%20Release%20Notes.docx
Redis%20on%20Windows.docx
redis.windows.conf
redis.windows-service.conf
redis-benchmark.exe
redis-benchmark.pdb
redis-check-aof.exe
redis-check-aof.pdb
redis-check-dump.exe
redis-check-dump.pdb
redis-cli.exe
redis-cli.pdb
redis-server.exe
redis-server.pdb
Windows%20Service%20Documentation.docx
```

2. Create a directory on your server to host the Redis files, such as d:\redis.
3. Navigate to **d:\redis** and open the command line interface form the address bar.
4. In the Command Line Interface enter **redis-server**.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:\Redis>redis-server
```

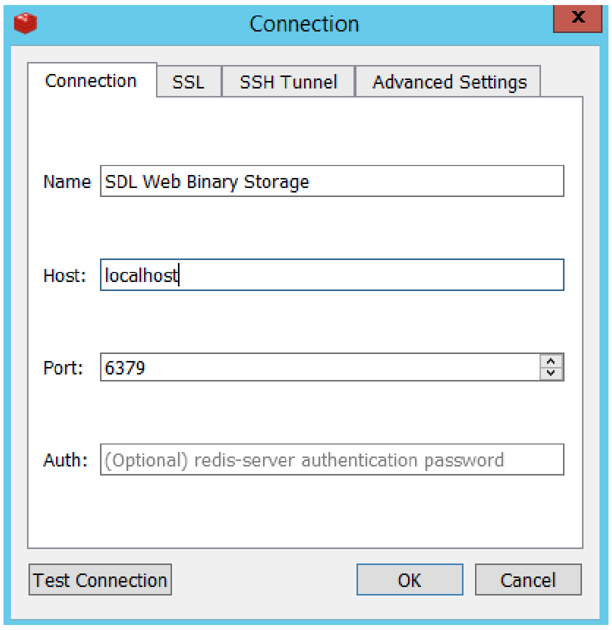
Press enter and the Redis Server will very quickly start up.



```
Administrator: C:\Windows\System32\cmd.exe - redis-server

                                   Redis 3.0.503 (00000000/0) 64 bit

                                   Running in standalone mode
                                   Port: 6379
                                   PID: 8980

                                         http://redis.io

[8980] 09 Feb 13:56:41.256 # Server started, Redis version 3.0.503
[8980] 09 Feb 13:56:41.272 * DB loaded from disk: 0.023 seconds
[8980] 09 Feb 13:56:41.272 * The server is now ready to accept connections on po
rt 6379
```

5. You should see a response similar to the above screenshot. Note, Redis is now running at **localhost:6379**.
6. Now, so you can visualize what is happening in the data store, install Redis Desktop Manager from the link at the beginning of this Annex.
7. Once the installation is complete, open Redis Desktop Manager.

8. In the bottom left of the Redis Desktop Manager interface, there is a Connection to Redis Server button. Click this.
9. Complete as shown:



Name: SDL Web Binary Storage

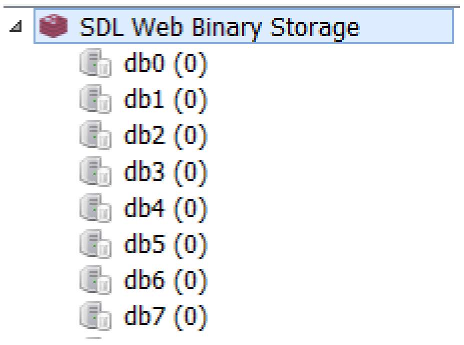
Host: localhost

Port: 6379

Auth: (Optional) redis-server authentication password

Note! This example is not using any authentication, so you can leave the 'Auth' box empty.

10. Once the Redis Desktop Manager is open, click on the 'SDL Web Binary Storage' node and you will see all the data stores that are available. The Binary packages will be Published to a Redis db ready for collection from the Worker Deployers.



About SDL

SDL (LSE: SDL) is the global innovator in language translation technology, services and content management. For more than 20 years, SDL has transformed business results by enabling nuanced digital experiences with customers across the globe so they can create personalized connections anywhere and on any device. Are you in the know? Find out more at SDL.com.