

Installing a deployment environment on RedHat, CentOS or Amazon Linux

The packages will work on CentOS and RedHat Enterprise 5.x or 6.x, or Amazon Linux, with Apache httpd 2.2 or 2.4.

Installing a deployment environment on platforms that uses RPM packages is now even easier than ever.

1. Install a JDK. You can install openjdk with Yum (`yum install java-1.6.0-openjdk`) or get the JDK from [Oracle](#).
2. Install the Yum repository of the WOCommunity like this:

```
sudo curl -o /etc/yum.repos.d/wocommunity.repo https://raw.githubusercontent.com/wocommunity/wonder/master/Utilities/Linux/wocommunity.repo
```

3. Once it's done, you can use Yum to install wotaskd and JavaMonitor.

```
sudo yum install wotaskd; sudo yum install womonitor;
```

4. To install the module for Apache httpd 2.2, run:

```
sudo yum install woadaptor
```

for Apache httpd 2.4:

```
sudo yum install woadaptor-httpd24
```

5. If SELinux is enabled on your system, wotaskd won't be reachable due to its preventing Apache from opening TCP connections. If you're comfortable with allowing Apache to connect to any TCP ports (including external hosts), you can run the following:

```
sudo setsebool -P httpd_can_network_connect=1
```

Alternatively, [Steven Klassen](#) has written a blog post ([Custom SELinux Port Access](#)) that outlines how to create and install a wotaskd SELinux module that specifically allows connections to port 1085.

And you're done! When the packages are installed, the services are started automatically. JavaMonitor will be running on port 56789. Your Project Wonder applications goes into `/opt/Local/Library/WebObjects/Applications`.

If you wish to restart wotaskd or Monitor, startup scripts are located in `/etc/init.d`. You can use the `service` tool to stop or restart them:

```
sudo /sbin/service wotaskd stop
sudo /sbin/service womonitor stop
```

(replace `stop` with `restart` to restart them in one step).

When Monitor is started, it will report to the console that is available at address like "ip-10-212-110-28" but that is the **internal** address of the Amazon server. The **external** address, which you must use, is available in the AWS console, and of the form "ec2-xx-vv-zz-ww.compute-1.amazonaws.com".

Don't forget to change the password in JavaMonitor after you made the installation!

Notes on CentOS 7 64/ Apache 2.4

The yum repositories for deployment artifacts is a little out of date, so these instructions are based on wonder source and compiling the adaptor on the target system.

Compiling the Apache Adaptor

Compiling Apache 2.4 WebObjects Adaptor

```
$ sudo yum install httpd httpd-devel
$ #I had to use: sudo yum install httpd24 httpd24-devel
$ # get the latest wonder source
$ git clone https://github.com/wocommunity/wonder.git
$ cd wonder/Utilities/Adaptors
$ # thanks to alextu for these commands to fix the make.config file
$ # (from his docker recipe)
$ sed -ri 's/ADAPTOR_OS = MACOS/ADAPTOR_OS = LINUX/g' make.config
$ sed -ri 's/ADAPTORS = CGI Apache2.2/ADAPTORS = Apache2.4/g' make.config
$ make
$ # If you get error about make not found, install dev tools
$ sudo yum install gcc make
$ # If you get error about redhat-hardened-ccl: No such file or directory
$ sudo yum install redhat-rpm-config
# #
$ # but the default on CentOS7 is /usr/lib64/httpd/modules
$ sudo cp Apache2.4/mod_WebObjects.so /usr/lib64/httpd/modules/
```

Configuring Apache for WebObjects

The standard apache.conf (found in the wonder source at wonder/Utilities/Adaptors/Apache2.4/apache.conf) should work fine. Here are a couple of modifications that worked for me:

The authorization for Apache changed between 2.2 and 2.4. If you're used to "Allow from all" or "Deny from all" you should know that Apache 2.4 syntax has changed so this is how those allows/denies work now. This is a mostly permissive structure restricting some of the verbs that are not used (We have some configs that allow PUT, DELETE, PROPFIND, etc... but they don't call me "Little Paranoid Larry" for nothing.

Apache Permissions Config

```
WebObjectsAlias /cgi-bin/WebObjects
<Location /cgi-bin/WebObjects/>
  <Limit GET POST OPTIONS >
    Require all granted
  </Limit>
  Require all denied
</Location>
```

Firewalld

CentOS7 switched to using Firewalld from IP Tables. I've tried to understand firewalld and I'm actually digging it even though it's a big change from IP Tables. Useful info on Firewalld can be found at [Linux firewalld information at Oracle](#) or [Firewalld at Fedora](#).

The cheat sheet commands that help get started are:

Firewalld Cheat Sheet

```
getting help on the command line: # firewall-cmd --help
checking the state of the firewall: # firewall-cmd --state
panic: # firewall-cmd --panic-on
stop panic: # firewall-cmd --panic-off
set a service to a zone for runtime (not persistent): # firewall-cmd --zone=public --add-service=https
set a service to a zone permanently: # firewall-cmd --permanent --zone=public --add-service=https
setup specific ports (like for WO instances) # firewall-cmd --zone=public --permanent --add-port=2000-2020/tcp
reload the configuration: # firewall-cmd --reload
find out what services the zone thinks should be handled at runtime (or permanently): # firewall-cmd (-permanent) --zone=public --list-services
go nuts and specify rules to within an inch of their lives: # firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4" source address="192.168.0.4/24" service name="http" accept"
```

SELinux

As is mentioned in <https://wiki.wocommunity.org/display/documentation/Installing+a+deployment+environment+on+RedHat,+CentOS+or+Amazon+Linux>, you can open up apache to network connections using the command:

SELinux

```
$ # make sure you have the policy utils (yum install policycoreutils-python)
$ sudo setsebool -P httpd_can_network_connect=1
```

If you have additional problems with the SELinux configuration, check the context and ownership of files. More useful information on SELinux can be found at <http://wiki.centos.org/HowTos/SELinux>.