CentOS 7 – Jetty 9

Version:	1.0.0
Created by:	cloudimg

Table of Contents

1.) Overview

This document is provided as a user guide for the CentOS 7 – Jetty 9 product offering on the Azure Marketplace. Please reach out to support@cloudimg.co.uk if any issues are encountered following this user guide for the chosen product offering.

2.) Access & Security

Please update the security group of the target instance to allow the below ports and protocols for access and connectivity.

Protocol	Type	Port	Description
SSH	TCP	22	SSH connectivity
HTTP	TCP	8080	Jetty Front End

3.) System Requirements

The minimum system requirements for the chosen product offering can be found below

Minimum CPU	Minimum RAM	Required Disk Space
1	1 GB	20 GB

4.) Connecting to the Instance

Once launched in the Azure Virtual Machines Service, please connect to the instance via an SSH client using the **azureuser** with the key pair associated at launch. Once connected as the **azureuser**, you will be able to **sudo** to the **root** user by issuing the below command.

Switch to the root user

```
sudo su -
```

5.) On Startup

An OS package update script has been configured to run on boot to ensure the image is fully up to date at first use. You can disable this feature by removing the script from /stage/scripts/ and deleting the entry in crontab for the root user.

Disable the OS update script from running on reboot

```
rm -f /stage/scripts/initial_boot_update.sh

crontab -e

#DELETE THE BELOW LINE. SAVE AND EXIT THE FILE.
@reboot /stage/scripts/initial_boot_update.sh
```

6.) Filesystem Configuration

Please see below for a screenshot of the server disk configuration and specific mount point mappings for software locations.

```
Filesystem       Size  Used Avail Use% Mounted on

devtmpfs         1.9G     0  1.9G   0% /dev

tmpfs            2.0G     0  2.0G   0% /dev/shm

tmpfs            2.0G  8.5M  1.9G   1% /run

tmpfs            2.0G     0  2.0G   0% /sys/fs/cgroup

/dev/nvme0n1p2    38G  2.5G   33G   7% /

/dev/nvme0n1p1   2.0G  121M  1.7G   7% /boot

tmpfs            391M     0  391M   0% /run/user/1002

/dev/nvme1n1     9.8G  308M  9.0G   4% /apps
```

Mount Point	Description
/boot	Operating System Kernel files
/apps	Java & Jetty installation directory

7.) Server Components

Please see below for a list of installed server components and their respective installation paths. The below versions are subject to change on initial boot based on the initial_boot_update.sh script finding new versions of the software in the systems package repositories.

Component	Version	Software Home
Cloud-Init	19.4	/etc/cloud
Java	1.8	/apps/java
Jetty	9.4.44	/apps/jetty
Azure CLI	2.53.1	/lib64/az

8.) Scripts and Log Files

The below table provides a breakdown of any scripts & log files created to enhance the useability of the chosen offering.

Script/Log	Path	Description
Initial_boot_update.sh	/stage/scripts	Update the Operating System with the latest updates available.
Initial_boot_update.log	/stage/scripts	Provides output for initial_boot_update.sh

9.) Using System Components

Instructions can be found below for using each component of the server build mentioned in section 7 of this user guide document.

Azure CLI

Using Azure CLI – as any OS user.

```
az
```

Cloud-Init

Edit the /etc/cloud/cloud.cfg file to reflect your desired configuration. A link to the cloud-init official documentation can be found below for referencing best practise for your use case.

https://cloudinit.readthedocs.io/en/latest/

```
vi /etc/cloud/cloud.cfg
```

Java

Java has been preinstalled on the instance and the below command can be used to verify the version currently installed.

```
java -version
```

Jetty

Start and Stop scripts have been created under the home directory of the jetty user. Issue the below commands to start and or stop the Jetty service as the jetty user.

```
#Switch to the jetty OS user

sudo su - jetty

#Start the Jetty service

$HOME/jetty-start.sh

#Stop the Jetty service

$HOME/jetty-stop.sh

#Check the status of the Jetty service

$HOME/jetty-status.sh
```

Once the Jetty service has started, you will be able to access the Jetty front end via the below URL exchanging the values between <> to match that of your own instance.

<PRIVATE/PUBLICIP>:8080/async-rest

Alibaba Cloud Partner Network | Registered Technology Partner

cloudimg
(+44) 02045382725
3rd Floor 86-90 Paul Street London EC2A 4NE
support@cloudimg.co.uk
https://cloudimg.co.uk

Blocking vs Asynchronous REST

This demo calls the EBay WS API both synchronously and asynchronously, to obtain items matching each of the keywords passed on the query string. The time the request thread is held by the servlet is displayed in red for both.
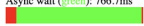
Blocking: kayak
Total Time: 1246.4ms
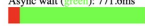Thread held (red): 1246.4ms

Blocking: mouse,beer,gnome
Total Time: 1554.6ms
Thread held (red): 1554.6ms

Asynchronous: kayak
Total Time: 835.3ms
Thread held (red): 68.6ms (68.2 initial + 0.4 generate)
Async wait (green): 766.7ms

Asynchronous: mouse,beer,gnome
Total Time: 841.0ms
Thread held (red): 69.5ms (69.3 initial + 0.2 generate)
Async wait (green): 771.6ms

By the use of Asynchronous Servlets and the Jetty Asynchronous client, the server is able to release the thread (green) while waiting for the response from Ebay. This thread goes back into the thread pool and can service many other requests during the wait. This greatly reduces the number of threads needed, which in turn greatly reduces the memory requirements of the server.

Press reload to see even better results after JIT and TCP/IP warmup!