

Create EC2 for DSpace Docker



These instructions describe how to try out DSpace Docker on an Amazon EC2 Instance. These instructions are in preliminary form and are intended for testing purposes only.

Select Amazon Linux

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the

Quick Start

My AMIs

AWS Marketplace

Community AMIs

☐ Free tier only ?

Amazon Linux

Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-061392db613a6357b (64-bit x86) / ami-062ce7f8c1e7ffd3c (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and more.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Amazon Linux

Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-01e24be29428c15b2

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include the latest versions of the software.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select "t2.large" (8G RAM)

A minimum of 5-6 G is required to run DSpace Docker.

By lowering the default JAVA_OPTS memory allocations, it might be possible to run within a "t2.medium" (4G RAM).

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying cc
[Learn more](#) about instance types and how they can meet your computing needs.

Filter by:

All instance types

Current generation

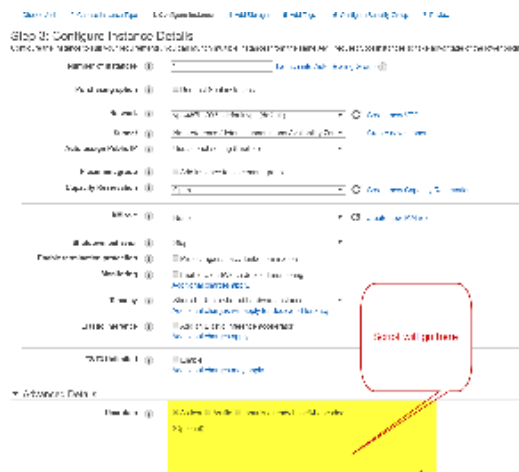
Show/Hide Columns

Currently selected: t2.large (Variable ECUs, 2 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 8 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)
☐	General purpose	t2.nano	1	0.5
☐	General purpose	t2.micro Free tier eligible	1	1
☐	General purpose	t2.small	1	2
☐	General purpose	t2.medium	2	4
☒	General purpose	t2.large	2	8
☐	General purpose	t2.xlarge	4	16
☐	General purpose	t2.2xlarge	8	32

Copy the start up script detailed below

This will ensure the docker-compose and git are available.



Create enough storage for docker images and volumes

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⁽ⁱ⁾	Device ⁽ⁱ⁾	Snapshot ⁽ⁱ⁾	Size (GiB) ⁽ⁱ⁾	Volume Type ⁽ⁱ⁾	IOPS ⁽ⁱ⁾	Throughput (MB/s) ⁽ⁱ⁾	Delete on Termination ⁽ⁱ⁾
Root	/dev/xvda	snap-0af9c0b1d247238d6	50	General Purpose SSD (gp2)	150 / 3000	N/A	☒

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Make port 8080 and 3000 available

These settings are recommended for **testing** purposes.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, you can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group

☐ Select an existing security group

Security group name:

Description:

Type ⁽ⁱ⁾	Protocol ⁽ⁱ⁾	Port Range ⁽ⁱ⁾	Source ⁽ⁱ⁾
SSH	TCP	22	Anywhere ⁽ⁱ⁾ 0.0.0.0/0, ::/0
Custom TCP	TCP	8080	Anywhere ⁽ⁱ⁾ 0.0.0.0/0, ::/0
Custom TCP	TCP	3000	Anywhere ⁽ⁱ⁾ 0.0.0.0/0, ::/0

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Start up Script

```
# Update OS software
sudo -n yum -y update

# Install Java 8
sudo -n yum -y install java-1.8.0-openjdk-devel
sudo -n yum -y remove java-1.7.0-openjdk

# Install Git
sudo -n yum -y install git

# install docker
sudo yum install docker -y

# Start the Docker Service
sudo service docker start

# Add the ec2-user to the docker group so you can execute Docker commands without using sudo.
sudo usermod -a -G docker ec2-user

# Install Docker compose
# https://docs.docker.com/compose/install/

sudo curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

# Clone DSpace-Docker-Images
cd
git clone https://github.com/DSpace-Labs/DSpace-Docker-Images.git
cd DSpace-Docker-Images/docker-compose-files/dspace-compose
```

Reboot the Machine

This ensures that the permission changes granted to the docker-compose executable will have taken effect.

Follow the documented DSpace Docker instructions

See <https://github.com/DSpace-Labs/DSpace-Docker-Images/blob/webinar/documentation/run.DSpace7.md>