Update the Drupal application in the same instance 2
Migrate your application to a new BitNami Instance 2

How can I create a backup of my instance? 2

How do I migrate the data to a new BitNami Drupal instance? 3

Backing up the files 3
Copying the files to the new instance 4
Restoring files in the new instance 4

How to associate the previous IP address to a new instance? 6

There are different ways to upgrade and maintain your Drupal installation up-to-date easily.

Update the Drupal application in the same instance

Drupal has a built-in mechanism for upgrading it from within the application itself. To do so, complete the steps below. Before starting the update process, we recommend that you make a backup. See the '[how to backup an instance]' section for instructions for doing so.

1. Go to Administration > Configuration > Development > Maintenance mode. Enable the "Put site into maintenance mode" checkbox and save the configuration.
2. Download the new Drupal version from Drupal.org. In this example, we will use the 7.14 version:
   $ wget http://ftp.drupal.org/files/projects...al-7.14.tar.gz
   $ tar -xzvf drupal-7.14.tar.gz
3. Create a backup of the .htaccess and settings.php files:
   $ cp /opt/bitnami/apps/drupal/htdocs/.htaccess /home/bitnami/.htaccess.backup
   $ cp /opt/bitnami/apps/drupal/htdocs/sites/default/settings.php /home/bitnami/settings.php.backup
4. Overwrite the Drupal installation with the new files:
   $ cp -R drupal-7.14/* /opt/bitnami/apps/drupal/htdocs
5. If you have customized the .htaccess or settings.php file you can restore them.
6. Check the Report -> Status and run the upgrade script if necessary.
7. Disable the Maintenance mode.

Starting with Drupal version 7.12-3, BitNami Drupal Stack ships Drush, a command line shell and scripting interface for Drupal. This allows you to upgrade the Drupal site using the following command:

```
$ cd /opt/bitnami/apps/drupal/htdocs
$ drush update
```

More information is available on Drush at http://www.drush.org

Migrate your application to a new BitNami Instance

In addition to upgrading your application, it is also important to use the latest versions of PHP, Apache and other dependencies in your production site. We regularly refresh the images available in the BitNami library to make sure that you always have access to the most up-to-date software.

This guide describes how to migrate your BitNami Drupal instance to a new AMI (Amazon Machine Image) version. In this guide, we will refer to the servers that you have running Drupal as "instances", which is the terminology that Amazon uses.

How can I create a backup of my instance?

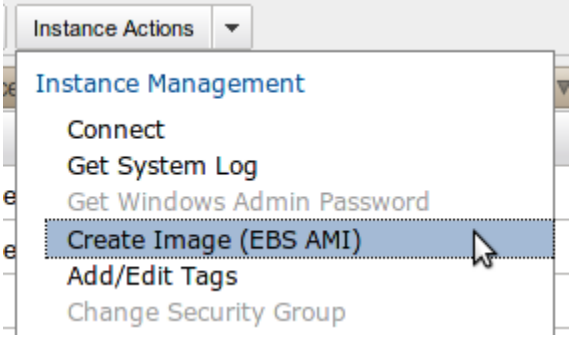
We strongly recommend that you create a backup of your current instance if you are going to do major changes or upgrades. It is not absolutely necessary if you are migrating the data to a new instance because the process does not modify your current instance, but in general it is a best-practice to make a backup before starting on any process where you will be accessing or making changes to your server, particularly if you are new to the process.

There are different ways to create a backup. For this tutorial, we will use the snapshot feature of the Amazon Cloud.

To create a snapshot with a consistent file system, you should stop the server(s) before taking the snapshot. If you have never logged into your machine, see this guide for instructions on connecting to your instance. Once you are logged in, run this command to stop the server(s):

```
$ sudo /opt/bitnami/ctlscript.sh stop
```

You can now create a new EBS image (snapshot) based on this instance.



If you run into issues during this process, you can simply launch a new instance from this snapshot and restart the process.

How do I migrate the data to a new BitNami Drupal instance?

Backing up the files

This section describes how to sync the Drupal data from your previous instance to a new instance running a later release of the BitNami Drupal AMI. First, log in to your current instance (or your backed up version) of Drupal.

- **Create a database backup.** Run the following command line command to create a database backup:

  ```
  $ mysqldump -u root -p bitnami_drupal7 > /home/bitnami/
  drupal_backup.sql
  ```

  The default database root password is "bitnami". If you are using BitNami Cloud Hosting, the password is the one that you specified during the server launch process. If you did not specify a password at that time, one was automatically generated for you. In that case, you can retrieve it by going to Servers -> Manage and then click on "show" in the Application Password row to display your password.

You can also create a backup through the phpMyAdmin interface.

- **Create a backup of the files**. The uploaded images, themes or plugins are saved in the "sites" folder. If you want to preserve this data, create a backup by issuing the following commands:

```
$ cd /opt/bitnami/apps/drupal/htdocs
$ sudo tar -czvf /home/bitnami/sites_backup.tar.gz sites
```

- **Create a backup of the configuration files**. If you have configured Drupal to run in the root URL, you may have modified the Drupal configuration file /opt/bitnami/apps/drupal/conf/drupal.conf file. If this is the case, create a copy of this file:

```
$ cp /opt/bitnami/apps/drupal/htdocs/.htaccess /home/bitnami/
htaccess.backup
$ cp /opt/bitnami/apps/drupal/conf/drupal.conf /home/bitnami/
drupal_backup.conf
```

You now have all of the required data to migrate your site to a new instance. If you have customized the php.ini, my.cnf, httpd.conf or any other configuration files, you may also want to update the configuration files in the new instance.

Copying the files to the new instance

To copy your files from the old instance to the new one, complete the following steps:

- **Download the backup files to your machine.** To download the backed up files to your local machine (so you can upload them to a new one), you can either use the "SCP" command to download them or use your preferred sFTP client. For this tutorial, we will use the SCP command line tool:

```
$ scp -i /path/to/your/private_key.pem bitnami@old_server:*backup* .
```

Tip: You can also send the files from the old one to the new one directly. Assuming both servers were started with the same AWS cloud credentials, you can use the same SSH key to connect to both servers. On your local host, add your ssh key to the SSH agent:

```
$ ssh-add /path/to/your/private_key.pem
```

- **Start the new AMI**. Now, start the new version of the BitNami Drupal AMI. You can view and access the latest version from the BitNami Drupal page. You can also start a BitNami AMI with a different operating system: Ubuntu, RedHat, etc, if you would like to switch operating systems at this time.
- **Upload the files to the new instance**. To upload the database backup and files that you downloaded in the first step in this section to the new instance, use the sFTP client:

```
$ scp -i /path/to/your/private_key.pem drupal_backup.sql
sites_backup.tar.gz htaccess.backup drupal_backup.conf bitnami@new_server:
```

Restoring files in the new instance

You can now restore the database and the Drupal files in the new installation by following the steps below. Before starting the process, it is important to **save a copy of the new settings.php file,** which contains the database settings:

```
$ cp /opt/bitnami/apps/drupal/htdocs/sites/default/settings.php /home/
bitnami
```

- **Restore the database**. Populate the new application with your data from the previous installation:

  ```
  $ mysql -u root -p bitnami_drupal7 < /home/bitnami/drupal_backup.sql
  ```

- **Restore the Drupal files**. If you want to restore the previously installed plugins, themes or the uploaded files, you can uncompress the backed up tarball that you created previously in the old server:

  ```
  $ sudo tar -xzvf /home/bitnami/sites_backup.tar.gz /opt/bitnami/apps/
  drupal/htdocs
  ```

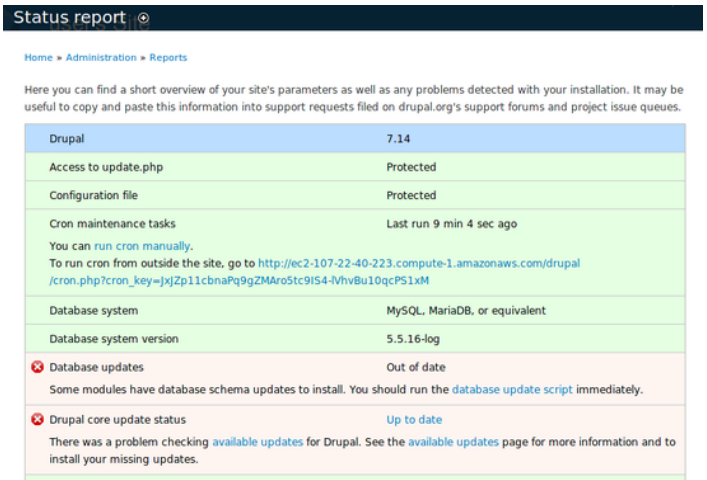
- **Restore the configuration files**. If you have customized the Drupal configuration, restore the Apache configuration file:

  ```
  $ cp /home/bitnami/htaccess.backup /opt/bitnami/apps/drupal/htdocs/
  .htaccess
  $ cp /home/bitnami/drupal_backup.conf /opt/bitnami/apps/drupal/conf
  $ sudo cp /home/bitnami/settings.php /opt/bitnami/apps/drupal/htdocs/
  sites/default
  ```

Restart the servers with the command below. You should then be able to access your Drupal installation in the new instance.

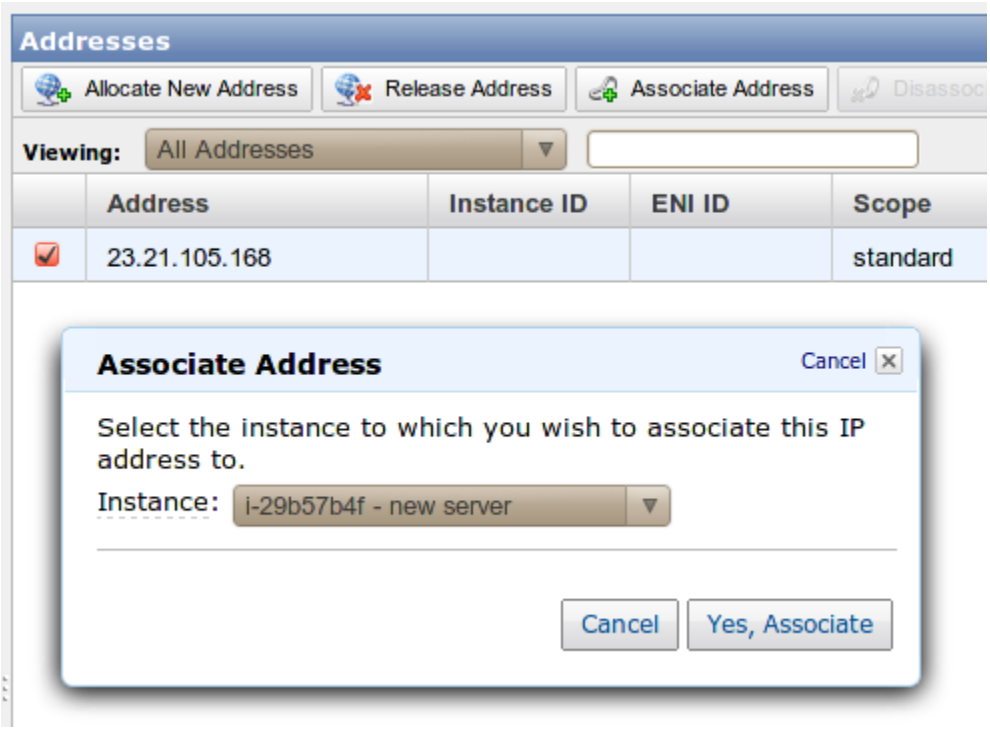
```
$ sudo /opt/bitnami/ctlscript.sh restart
```

Drupal may require that you upgrade the database schema. Check the Report -> Status and run the database migration script if this is the case.



How to associate the previous IP address to a new instance?

Once you have checked that the application is running properly and all of the data was successfully migrated, you can associate the old server IP address with the new instance. To do so, go to "Elastic IP's" in the Amazon AWS Console and change the instance associated with it. If you do not have any IP address, you can create a new one:



If you are using BitNami Cloud Hosting, you can either delete the old server (if you made a backup, you will still be able to restore it in the future) or use the "assign IP' button in the Manage section for your server to change the IP of the old server. You will then be able to go to the new server, click on the "assign IP" button and select the old IP address for your new server.

Once it has been updated (note that this sometimes takes a few minutes), you can access the new instance using the same domain name that you were using for the previous instance.