Hosting WordPress Using Amazon S3

**SPL-39 Version 4.1.14**

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Overview**

This lab demonstrates how to use Amazon S3 for static website hosting. In this lab, you convert a WordPress website that is running on an Amazon EC2 instance into a static website hosted on Amazon S3. WordPress is an open-source content management system that can be used to host web pages, blog posts, and even shopping websites. It runs more than 60 million websites on the Internet and is highly extensible via a plugin and templating system.

WordPress by its nature is a dynamic website. A dynamic website relies on server-side processing, including server-side scripts such as PHP, JSP, or ASP.NET. In order to host a dynamic website, some form of a web server, such as an Amazon EC2 instance, is required to host the web application and database. This requires you to maintain not only the web application, but also the host operating system where it is running. You may also need to scale your application as the load increases, requiring more resources and adding to your maintenance and costs.

An alternative is to convert your WordPress site to a static website, which can dramatically reduce the hosting cost for a website while supporting large-scale user base. With a static website, individual webpages include static content and might also contain client-side scripts, but server-side scripts are not necessary. Note that Amazon S3 does not support server-side scripting.

The same concepts covered in this lab to enable static website hosting in Amazon S3 can be used for hosting any static website.



OBJECTIVES

By the end of this lab, you will be able to do the following:

* Configure WordPress on an Amazon EC2 instance.
* Export a WordPress site to static files.
* Copy static files to an Amazon S3 static website.
* Create a script to send Wordpress changes to Amazon S3.

TECHNICAL KNOWLEDGE PREREQUISITES

* In order to successfully complete this exercise, you should be familiar with basic website concepts.

DURATION

This lab requires approximately **60 minutes** to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Note:** A note, tip, or important guidance.
* **Additional information:** Where to find more information.
* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Configure WordPress on Amazon EC2**

An Amazon EC2 instance containing WordPress has been automatically provisioned as part of this lab.

In this task, you perform the initial configuration of WordPress and create a blog post.

1. To the left of these instructions, copy the **WordPressURL**.

This URL is a link to your WordPress website.

1. Paste the URL into a new web browser tab and press Enter.

The WordPress configuration page displays.

1. Configure the following values:

* **Site Title:** Enter any title you wish
* **Username:**

student

* **Password:** Copy the **Administrator Password** value to the left of these instructions and paste it in this field
* **Your Email:**

student@example.com

* **Search Engine Visibility:** Leave deselected

1. Choose **Install WordPress**.

You are presented with a *Success* screen.

1. Choose **Log In** and then log in with the following credentials:

* **Username:**

student

* **Password:** Copy the **Administrator Password** value to the left of these instructions and paste it in this field

1. Choose **Log In**.

The WordPress dashboard displays.

You can now create a blog post to add information to your website.

1. Choose the  **New** link at the top of the page to create a new blog post.
2. If you see *Welcome to the Block Editor*, click **Close**.
3. Enter a **Title** and write some text. Be creative!
4. After you have finished, at the top-right corner of the page choose **Publish…**
5. Choose **Publish** again to save your blog post.

You can now view your website.

1. At the right side of the screen, click **View Post**.

Your website is displayed, with your most recent blog post at the top.

**Congratulations!** You have successfully configured wordpress website on an Amazon EC2 instance and created new blog post.

Leave this browser tab open. You use it again later.

**Task 2: Create an Amazon S3 bucket with static website hosting**

In this task, you create an Amazon S3 bucket and configure it for static website hosting. This makes the bucket accessible on the Internet via a URL.

In a later step, you use this bucket to host a static (unchanging) version of your WordPress website.

1. Return to the browser tab containing the AWS Management Console that you opened at the start of this lab.

**Note:** If you cannot find the correct browser tab, choose the **Open Console** button on the top of these instructions to open a new tab with the AWS Management Console.

1. On the **AWS Management Console**, use the **AWS search bar** to search for

S3

 and then choose the service from the list of results.

1. Choose **Create bucket** and then configure:

* For **Bucket name**, enter

wordpress-NUMBER

* Replace **NUMBER** with a random number

**Note:** For the solution to work properly, the bucket name must start with **wordpress-** .

1. In the **Object Ownership** section, configure the following:

* Choose **ACLs enabled** .
* Choose  **Object writer**.

1. In the **Block Public Access settings for this bucket** section, de-Select  **Block *all* public access** option, and then leave all other options **deselected**.

**Additional Information:** Notice all of the individual options remain deselected. When deselecting all public access, you must then select the individual options that apply to your situation and security objectives. In a production environment, it is recommended to use the least permissive settings possible.

1. Select  **I acknowledge that …**
2. Choose **Create bucket** .

**Note:** Amazon S3 bucket names must be globally unique. If you receive an error stating *Bucket with the same name already exists*, choose a different random number in the bucket name and try again until it works.

1. Choose the link for the name of the bucket you created. The bucket overview window opens.
2. Choose the **Properties** tab.
3. Scroll down to **Static website hosting**.
4. Choose **Edit** , then configure the following:

* For **Static website hosting**, select  **Enable**
* For **Hosting type**, select  **Host a static website**
* For **Index document**, enter

index.html

* For **Error document**, enter

error.html

* Choose **Save changes** .

1. Scroll down to the **Static website hosting** section and copy the **Bucket website endpoint** to your text editor for later use.

It should look similar to: *http://wordpress-123456789.s3-website-us-west-2.amazonaws.com*

Your Amazon S3 bucket is now ready to receive content from your WordPress website.

**Congratulations!** You have successfully created an S3 bucket and configured it to host static website.

**Task 3: Generate a static version of WordPress**

In this task, you use the **wp-static** utility to generate a static copy of your WordPress website as HTML pages. These pages will contain a copy of the entire website and can be used without the WordPress server.

**Additional Information:** For more information about the **wp-static** utility, refer to the **Additional resources** section at the end of this guide.

1. To connect to the **WordPress-Instance** terminal, copy **SSMSessionURL** value from the list to the left of these instructions and then open this URL in a new browser tab.

The terminal of the instance is now open with the **WordPress-Instance$** prompt. You use this terminal to run  **Commands** in the

code blocks

 below, as per the instructions in the following steps.

1. **Command:** To configure the Apache web server to allow permalinks override, run the following command:

**WordPress-Instance$**

sudo sed -i.bak -e 's/AllowOverride None/AllowOverride All/g' /etc/httpd/conf/httpd.conf;

**Expected output:**

None, unless there is an error.

1. **Command:** To restart the Apache web server to apply the permalinks override change, run the following command:

**WordPress-Instance$**

sudo service httpd restart

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Redirecting to /bin/systemctl restart httpd.service

1. **Command:** To download the *wpstatic* tool and generate a static HTML version of the WordPress web site, run the following commands:

**WordPress-Instance$**

cd /var/www/html/wordpress;

sudo wget https://us-west-2-aws-training.s3.amazonaws.com/courses/spl-39/v4.1.14.prod-c48d9fd0/scripts/wpstatic.sh;

sudo /bin/sh wpstatic.sh -a;

**Expected output:**

The tool is downloaded and installed. Then it runs to convert the HTML files into a static version.

**Note:** Many lines should be displayed in your terminal session window. If this did not happen, you might need to press ENTER to run the last line of the commands.

1. **Command:** To make the Apache webserver user account the owner of the files created by the wpstatic.sh script in /var/www/html/wordpress/, run the following command:

**WordPress-Instance$**

sudo chown -R apache:apache /var/www/html/wordpress

**Expected output:**

None, unless there is an error.

You now have a static version of your website. It can be viewed in your web browser.

1. Copy the **StaticURL** address shown to the left of these instructions.
2. Paste the URL into a new web browser tab, and then press Enter.

Your WordPress website displays. Examine the page URLs and notice that it is now being served as static HTML pages. WordPress is not involved in serving these static pages.

**Congratulations!** You have successfully generated static version of your wordpress website.

**Task 4: Uploading static WordPress pages to Amazon S3**

In this task, you use the AWS Command Line Interface (AWS CLI) to copy the static pages you generated in the previous task to Amazon S3. You are then able to access the website even when the WordPress web server is turned off.

This has several advantages:

* Amazon S3 is highly scalable and can service more users than a single web server.
* Serving content from Amazon S3 does not require any Amazon EC2 instances to be running, thereby lowering costs.

1. **Command:** To retrieve the name of AWS region and the name of your Amazon S3 bucket and store the values in variables, run the following commands:

# Determine Region

AZ=`curl --silent http://169.254.169.254/latest/meta-data/placement/availability-zone/`

REGION=${AZ::-1}

# Retrieve Amazon S3 bucket name starting with wordpress-\*

BUCKET=`aws s3api list-buckets --query "Buckets[?starts\_with(Name, 'wordpress-')].Name | [0]" --output text`

**Expected output:**

None, unless there is an error.

**Note:** By running these commands, you have stored the AWS Region of the EC2 instance in a variable named *AZ*. You have also used an AWS CLI command to retrieve any Amazon S3 bucket names in your account with names that start with *wordpress-* and stored the result in a variable named *Bucket*.

1. **Command:** To copy the WordPress static files to your Amazon S3 hosted website, run the following command:

aws s3 sync --acl public-read /var/www/html/wordpress/wordpress-static s3://$BUCKET

**Expected output:**

The files are uploaded to the S3 bucket.

**Note:** In the previous task, you used the wp-static command to generate static HTML pages of your WordPress site. By running this command you have copied the static files from the *wordpress-static* directory to the Amazon S3 bucket you created. Notice the command uses the *Bucket* variable from the previous command.

Your website is now available in Amazon S3!

1. Retrieve the **Endpoint** that you previously saved into a text editor. It should look something like: *http://wordpress-123456789.s3-website-us-west-2.amazonaws.com*

Copy and paste the endpoint URL into a new browser tab to view your static WordPress site hosted from Amazon S3.

**Note:** If you cannot find the endpoint, return to the web browser tab with the S3 console, choose **Static website hosting** and then choose the **Endpoint** that is displayed.

**Congratulations!** You have successfully hosted your worodpress website in Amazon S3.

**Task 5: Using scripts to upload changes to Amazon S3**

Now that you have created a static version of your website and have uploaded changes, you may want to simplify this operation in the future. In this task, you create a script to update the static files with any changes from WordPress and upload the changes Amazon S3 to replace running each command manually.

1. **Command:** To create a new shell script that extracts the pages from WordPress and copies them to your Amazon S3 bucket, run the following commands:

echo "cd /var/www/html/wordpress; sudo rm -rf wordpress-static; sudo /bin/sh wpstatic.sh -a; aws s3 sync --acl public-read --delete /var/www/html/wordpress/wordpress-static s3://$BUCKET" > $HOME/wordpress-to-s3.sh;

chmod 0755 $HOME/wordpress-to-s3.sh;

**Expected output:**

None, unless there is an error.

**Note:** Below is a breakdown of each command in the script, with comments to clarify what each command does:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DO NOT COPY THESE COMMANDS \*

\* THIS IS JUST AN EXPLANATION \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Change to the wordpress directory

cd /var/www/html/wordpress

# Delete the wordpress-static directory

sudo rm -rf wordpress-static

# Run the wp-static utility to create static HTML pages of the current WordPress site

sudo /bin/sh wpstatic.sh -a

# Copy the WordPress static files to the S3 bucket

aws s3 sync --acl public-read --delete /var/www/html/wordpress/wordpress-static s3://$BUCKET

You can now add a new page to WordPress and then run the script to confirm that the content is being copied to Amazon S3.

1. Return to your web browser tab that is running WordPress. If you cannot find the tab, copy the **WordPressURL** shown to the left of these instructions and paste it into a new browser tab.
2. Choose the  **New** link at the top of the page to create a new blog post.
3. Enter a title and some text, choose **Publish…**, and then choose **Publish** again.
4. **Command:** Return to the terminal session and enter the following command to run the script you created previously:

/home/ssm-user/wordpress-to-s3.sh

**Expected output:**

The script generates a static HTML version of the WordPress website and uploads the files to the S3 bucket.

1. Return to your browser tab that is connected to your Amazon S3 bucket and refresh the page. Your new blog post is displayed on the page. If it does not appear, wait one minute and refresh the browser page.

If you cannot find the tab, retrieve the **Endpoint** that you earlier saved into a Text Editor. It should look something like: *http://wordpress-123456789.s3-website-us-west-2.amazonaws.com*

**Note:** If you cannot find the Endpoint, return to the web browser tab with the S3 Management Console, choose **Static website hosting** and choose the **Endpoint** that is displayed.

**Congratulations!** You have successfully created a script to copy pages from your EC2 instance to S3 bucket which is hosting wordpress website. Now whenever you make changes to your wordpress website running on your EC2 instance, you just have to run this script instead of copying files over manually!.

**Some things to consider**

**Turning off the Amazon EC2 instance:** Once your static pages have been copied to Amazon S3, you could *stop* your Amazon EC2 instance to save money. Simply *start* it again when you wish to write another blog post, then turn it off once the updates have been copied to Amazon S3.

**A friendly URL:** The URL to your Amazon S3 hosted website is not easy to remember. You could create a more friendly custom domain name using Amazon Route 53.

**Adding dynamic content to pages:** While WordPress is an interactive application, pages stored in Amazon S3 are static and cannot respond to user input. If you want dynamic content on pages, consider using Javascript to add logic within the web browser, or use plugin services such as Disqus to host content from a different website.

**Summary**

**Congratulations!** You have now successfully:

* Configured WordPress on Amazon EC2.
* Exported WordPress to static files.
* Copied static files to an Amazon S3 static website.
* Created a script to send Wordpress changes to Amazon S3.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources:**

* [Run a hosted WordPress website](http://wordpress.com/)
* [wp-static plugin for WordPress](https://wordpress.org/plugins/static-html-output-plugin/)
* [wp-static script on GitHub](https://github.com/chnm/WP-Static)
* [EC2 Instance Connect](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Connect-using-EC2-Instance-Connect.html)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).