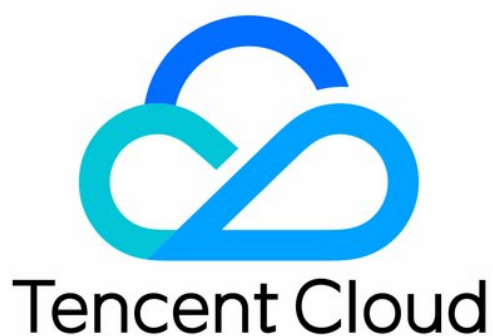


# **Tencent Cloud Lighthouse**

## **Best Practices**

### **Product Documentation**



## Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## Best Practices

### Websites

- Building up a Website Using WordPress Application Template

### Building Development Environment

- Building a Platform with Theia IDE Template

- Building LAMP Development Environment

- Building Node.js Development Environment

- Building ASP.NET Development Environment

- Building Docker Container Environment with Template

- Managing K3s Container Cluster with Image

### Cloud Storage System

- Building up a Cloud Storage System with Cloudfire Application template

### Building an E-commerce Platform

- Building an Individual E-commerce Site Using WooCommerce Application Template

### Setting up Live Streaming Service with SRS Application Template

### Setting Up an FTP Server

- Setting Up an FTP Server on Linux Lighthouse Instance

- Setting Up an FTP Server on Windows Lighthouse Instance

### Installing SSL Certificate

- Installing SSL Certificate

- Installing Certificate on NGINX Server

- Installing Certificate on Apache Server (Linux)

- Installing Certificate on Apache Server (Windows)

# Best Practices

## Websites

### Building up a Website Using WordPress Application Template

Last updated : 2023-02-16 15:38:35

## Overview

WordPress is the world's most popular open source blog and content management website building platform. It is easy to use, flexible and extensible. It has powerful features and provides rich theme plugins. Tencent Cloud Lighthouse provides WordPress application image, allowing you to quickly build various websites such as blogs, corporate official websites, e-commerce websites, and forums.

### Note:

In the following example, we use the WordPress application image, which is based on CentOS 7.6 64-bit. Note that application images are subject to updates from time to time, and the actual image information on the purchase page shall prevail.

Lighthouse also provides the WordPress plugin image, which has pre-installed Nginx, MariaDB, and PHP software, and integrates plugins such as Tencent Cloud Captcha, CDN, COS, IMS, TMS, and SMS. You can configure a WordPress plugin image when purchasing an instance. For how to use the plugin, see [Tencent Cloud Open Source Application Plugin](#).

## Directions

1. Log in to the [Lighthouse console](#).
2. Click **Create** to enter the Lighthouse purchase page.

Region ⓘ

Hong Kong
Singapore
Tokyo
Silicon Valley
Toronto
Frankfurt
Mumbai

Lighthouse instances in different regions cannot communicate with each another over a private network. Selecting the region closest to your end users can minimize download speed. You cannot change the region after creating a Lighthouse instance. [Region and connectivity](#)

Availability zone ⓘ

☒ Randomly assigned ⓘ

Image

Official image
Individual image

Application image
System image

WordPress 5.7.1

WooCommerce 6.5.1

SRS Streaming Server 4.5

Cloud

Matomo 4.9.1

LAMP 7.4.16

Node.js 14.16.1

Theia

Docker 19.03.9

K3s 1.23.6

ASP.NET 4.8

WordPress 5.7.1

WordPress is the world's most popular open source blog and content management site building platform, with simple to use, powerful, flexible and scalable features. With a wealth of thematic plug-ins, you can use it to build blogs, corporate websites, e-commerce, forums and other types of websites. The image is based on CentOS system, has pre-installed Nginx, MariaDB, PHP software.

Instance bundle ⓘ

General
Enterprise

2SD/month

CPU 2 cores (dedicated)

Memory 2GB

System disk 30GB SSD

Bandwidth 30Mbps

Transfer 1024 GB/month

50SD/month

CPU 2 cores (dedicated)

Memory 2GB

System disk 50GB SSD

Bandwidth 30Mbps

Transfer 2048 GB/month

60SD/month

CPU 2 cores (dedicated)

Memory 4GB

System disk 60GB SSD

Bandwidth 30Mbps

Transfer 2560 GB/month

80SD/month

CPU 2 cores (dedicated)

Memory 4GB

System disk 80GB SSD

Bandwidth 30Mbps

Transfer 3072 GB/month

90SD/month

CPU 2 cores (dedicated)

Memory 8GB

System disk 90GB SSD

Bandwidth 30Mbps

Transfer 3584 GB/month

100SD/month

CPU 2 cores (dedicated)

Memory 8GB

System disk 100GB SSD

Bandwidth 30Mbps

Transfer 4096 GB/month

An independent fixed public IP is assigned for free. The public network outbound traffic beyond the transfer quota will incur additional fees. [View pricing](#)

Instance name

Optional. Defaults to "image name-four random characters" if it's left empty

The multiple instances created in batch will be suffixed with sequential numbers. You can enter 60 characters

Purchase period

1 month
2
3
6 months
1 year
2 years
3 years
4 years
5 years
More

☐ Auto-renew the device every month when my account has sufficient balance

Quantity

- 1 +

**Region:** We recommend you select a region near your target users to reduce the network latency and improve their access speed.

**Availability zone:** **Randomly assigned** is selected by default. You can select one as well.

**Image:** Select the "WordPress 5.7.1 Community" application image.

**Instance bundle:** Select an instance bundle according to the required instance configuration (including CPU, memory, system disk, bandwidth, and monthly traffic).

**Instance name:** Enter a custom instance name. If it is left empty, an "image name + 4-digit random string" will be used as the name by default. When multiple instances are created in a batch, their names will be consecutive with auto-incrementing suffixes. For example, if you enter "LH" as the name and purchase three instances, the three instances are named "LH1", "LH2", and "LH3".

**Purchase period:** Default to **1 month**.

**Quantity:** One instance by default.

3. Click **Buy now** to submit your order and make the payment as prompted.

4. Go back to the Lighthouse console.

5. After the instance is created, select the instance from the list to enter its details page.

You can view the configurations of the WordPress application on this details page.

6. Select the **Pre-installed application** tab, and enter the application details page.

7.

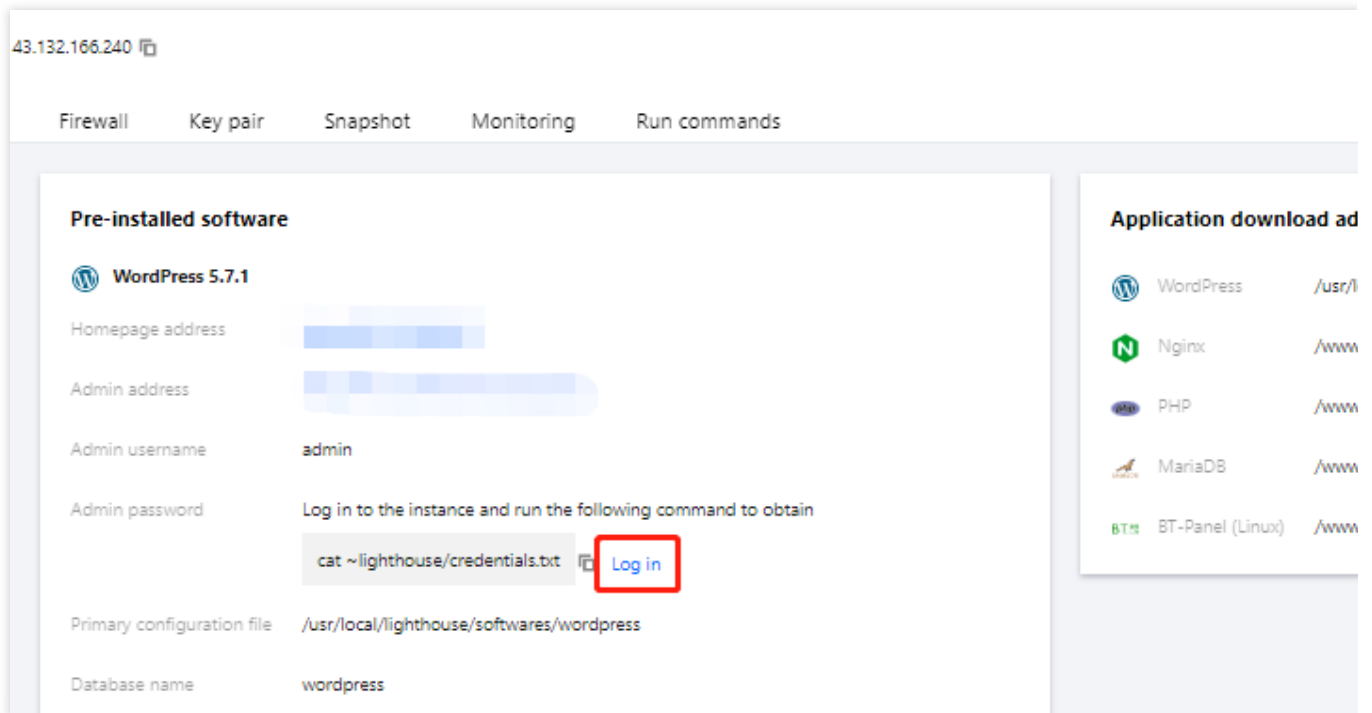
In the **Pre-installed software** section

, click



to copy the command for obtaining WordPress admin account and password.

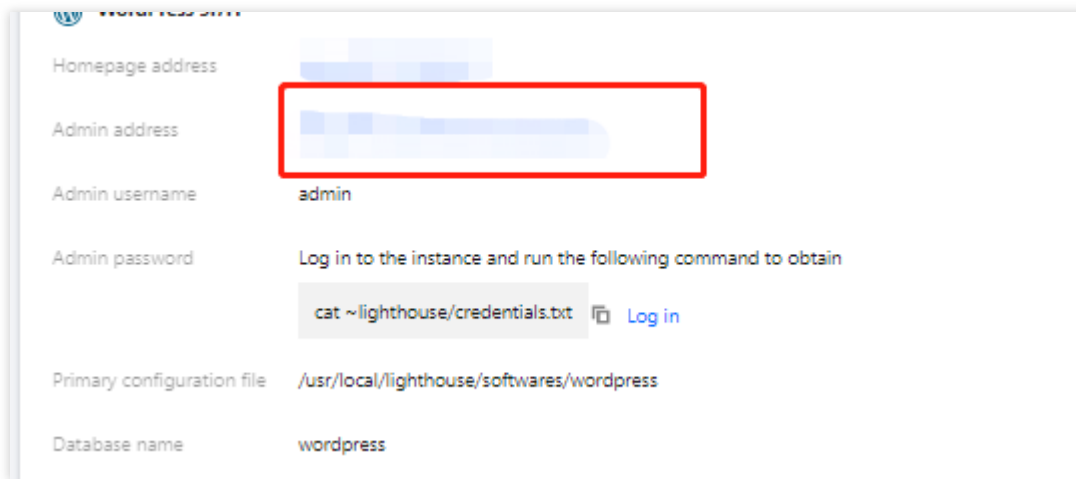
8. In the **Pre-installed software** section, click **Log in** beside the command or at the upper right corner.



9. In the pop-up login window, paste the command obtained in [step 7](#) and press **Enter**. Then, you can get the WordPress admin account ( `admin` ) and the corresponding password.

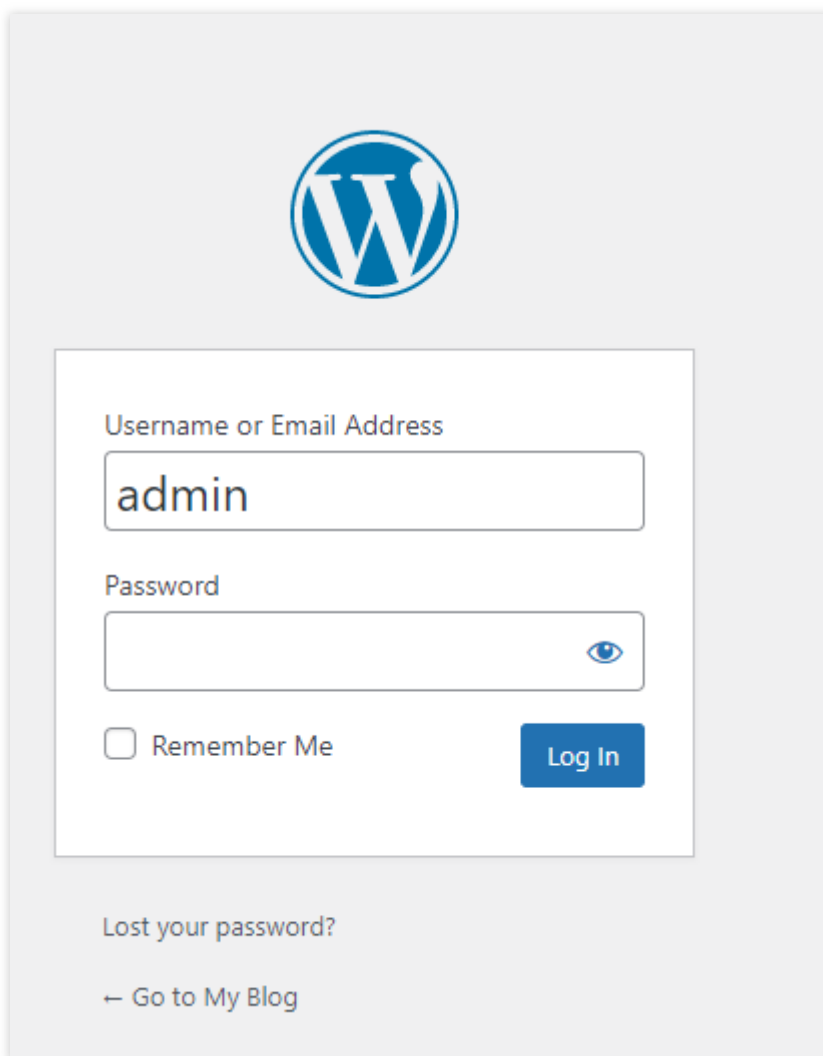
```
[lighthouse@VM_8_16_centos ~]$ cat ~lighthouse/credentials.txt
wordpress_username = admin
wordpress_password = [REDACTED]
mariadb_password = [REDACTED]
[lighthouse@VM_8_16_centos ~]$
```


10. Copy and record the WordPress admin account and password.
11. Close the login window and go back to the application details page of the instance.
12. In the **Pre-installed software** section, click **Admin address** of WordPress.



Homepage address	
Admin address	
Admin username	admin
Admin password	Log in to the instance and run the following command to obtain <pre>cat ~/lighthouse/credentials.txt</pre> <a href="#">Log in</a>
Primary configuration file	/usr/local/lighthouse/softwares/wordpress
Database name	wordpress

13. In the newly opened browser window, enter the username and password recorded in [Step 10](#) and click **Log in**.





Username or Email Address

admin

Password

☐ Remember Me

Log In

[Lost your password?](#)

[← Go to My Blog](#)

**Note:**

For WordPress 5.7.1 application image, you need to check the email and database updates, and confirm the information by clicking **This address is correct** and **Upgrade Wordpress database**.



After successful login, you can manage, customize, and configure WordPress based on your actual needs.

## Related Operations

### Updating WordPress Admin Profile

1. In the left sidebar of the WordPress admin interface, select **Users > All users**.
2. Find the `admin` user and click **Edit**.
3. Set personal information according to actual needs.

For example:






In the "Contact information" field, enter your email address.

In the "Account management" column, click **Generate password** and enter a new admin password.

4. Click **Update profile**.

### Viewing Other Configuration Information

You can view the configuration information of WordPress and other items such as homepage address, Nginx master configuration file saving path, MariaDB admin password, and software installation path on the application details page of the WordPress instance.

Application download address		
	WordPress	/usr/local/lighthouse/softwares/wordpress
	Nginx	/www/server/nginx/
	PHP	/www/server/php
	MariaDB	/www/server/mysql/
	BT-Panel (Linux)	/www

### Enabling HTTPS Access

You can install an SSL certificate and enable HTTPS access for your WordPress instance as instructed in [Installing Certificate on NGINX Server](#).

# Building Development Environment

## Building a Platform with Theia IDE Template

Last updated : 2023-02-16 15:40:30

### Overview

Theia IDE is an open-source extensible framework for building web-based cloud IDEs, with proper multi-language support and VS Code extensions. Tencent Cloud Lighthouse provides the Theia IDE image with Go, Python, Node.js, Clang, and OpenJDK development environments installed, allowing you to easily and quickly develop projects and businesses across platforms.

### Directions

1. Log in to the [Lighthouse console](#).
2. Click **Create** to enter the Lighthouse purchase page.

Region ⓘ

Hong Kong
Singapore
Tokyo
Silicon Valley
Frankfurt
Mumbai

Lighthouse instances in different regions cannot communicate with each another over a private network. Selecting the region closest to your end users can mini improve download speed. You cannot change the region after creating a Lighthouse instance. [Region and connectivity](#)

Availability zone ⓘ

☒ Randomly assigned ⓘ

Image

Official image
Individual image

Application image
System image

**SRS Streaming Server 4.4**

**WordPress 5.7.1**

**Typecho 1.1.0**

**Cloud**

**Matomo 4.9.1**

**LAMP 7.4.16**

**Node.js 14.16.1**

**Thick**

**Docker 19.03.9**

**K3s 1.23.6**

**ASP.NET 4.8**

**SRS Streaming Server 4.4**

SRS is the popular open source audio and video server, mainly used in live streaming and WebRTC, supporting RTMP, WebRTC, HLS, HTTP-FLV and SRT protocols, the Star and the most active in the streaming server industry, with users distributed worldwide. SRS entered the Mulan community incubation in 2021 driven open source project and gradually in build an open source solution for audio and video to make audio and video development easy. The image is base on operating system.

Instance bundle ⓘ

General
Enterprise

**5 USD/month**

CPU 2 cores (dedicated)  
Memory 2GB  
System disk 30GB SSD  
Bandwidth 30Mbps  
Transfer 1024 GB/month

**7 USD/month**

CPU 2 cores (dedicated)  
Memory 2GB  
System disk 50GB SSD  
Bandwidth 30Mbps  
Transfer 2048 GB/month

**9 USD/month**

CPU 2 cores (dedicated)  
Memory 4GB  
System disk 60GB SSD  
Bandwidth 30Mbps  
Transfer 2560 GB/month

**11 USD/month**

CPU 2 cores (dedicated)  
Memory 4GB  
System disk 80GB SSD  
Bandwidth 30Mbps  
Transfer 3072 GB/month

**16 USD/month**

CPU 2 cores (dedicated)  
Memory 8GB  
System disk 90GB SSD  
Bandwidth 30Mbps  
Transfer 3584 GB/month

**22 USD/month**

CPU 2 cores (dedicated)  
Memory 8GB  
System disk 100GB SSD  
Bandwidth 30Mbps  
Transfer 4096 GB/month

An independent fixed public IP is assigned for free. The public network outbound traffic beyond the transfer quota will incur additional fees. [View pricing](#)

Instance name

Optional. Defaults to "image name-four random characters" if it's left empty

The multiple instances created in batch will be suffixed with s default. You can enter 60 characters

Purchase period

1 month
2
3
6 months
1 year
2 years
3 years
4 years
5 years
More

☐ Auto-renew the device every month when my account has sufficient balance

Quantity

- 1 +

**Region:** Select a region near your target users to reduce the network latency and improve their access speed.

**Image:** Select the **Theia IDE 1.21.1** application image.

**Availability zone:** **Randomly assigned** is selected by default. You can select one as well.

**Instance bundle:** Select an instance bundle according to the required instance configuration (including CPU, memory, system disk, bandwidth, and monthly traffic).

**Instance name:** Enter a custom instance name. If it is left empty, an "image name + 4-digit random string" will be used as the name by default. When instances are created in batches, their names will be consecutive with auto-incrementing suffixes. For example, if you enter "LH" as the name and purchase three instances, the three instances are named "LH1", "LH2", and "LH3".

**Purchase period:** Default to **1 month**.

**Quantity:** Default to **1**.

3. Click **Buy now** to submit your order and make the payment as prompted. Then, return to the Lighthouse console.

4. After the instance is created, select the instance from the list to enter its details page.

You can view the configuration items of the Theia IDE application.

5. Select the **Pre-installed application** tab to enter the application details page.

6.

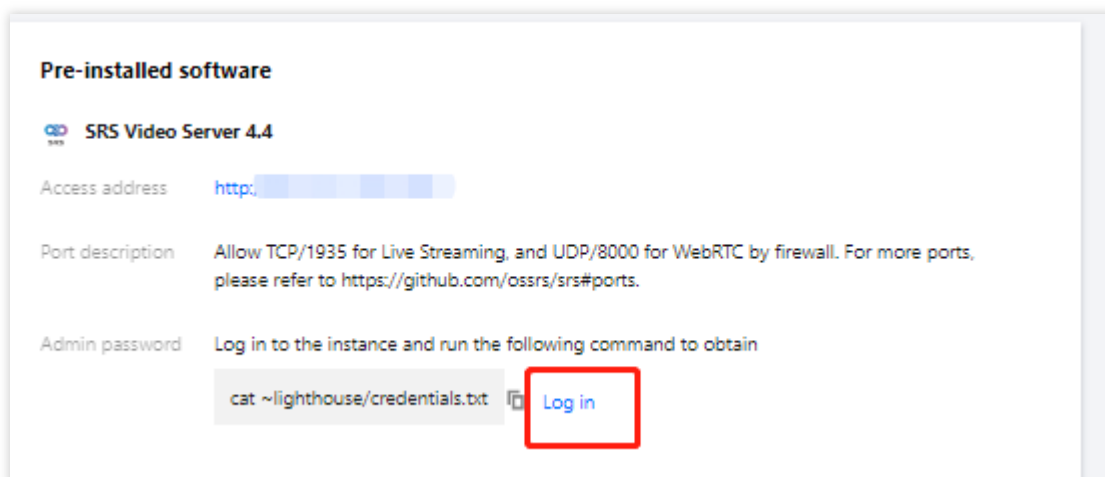
In the **Pre-installed software** section

, click



to copy the command for getting the admin password of Theia 1.21.1.

7. In the **Pre-installed software** section, click **Log in**.



8.

In the pop-up login window

, paste the command copied in [step 6](#) and press **Enter**.

Then, you can get the Theia IDE admin account (admin) and password. Store and record them properly.

9. Close the login window and go back to the application details page of the instance.

10. In the **Pre-installed software** section, click the **Access address** of Theia 1.21.1.

**Note:**

We recommend you use Chrome or Firefox for this operation, as other browsers (such as Safari) may have compatibility issues.

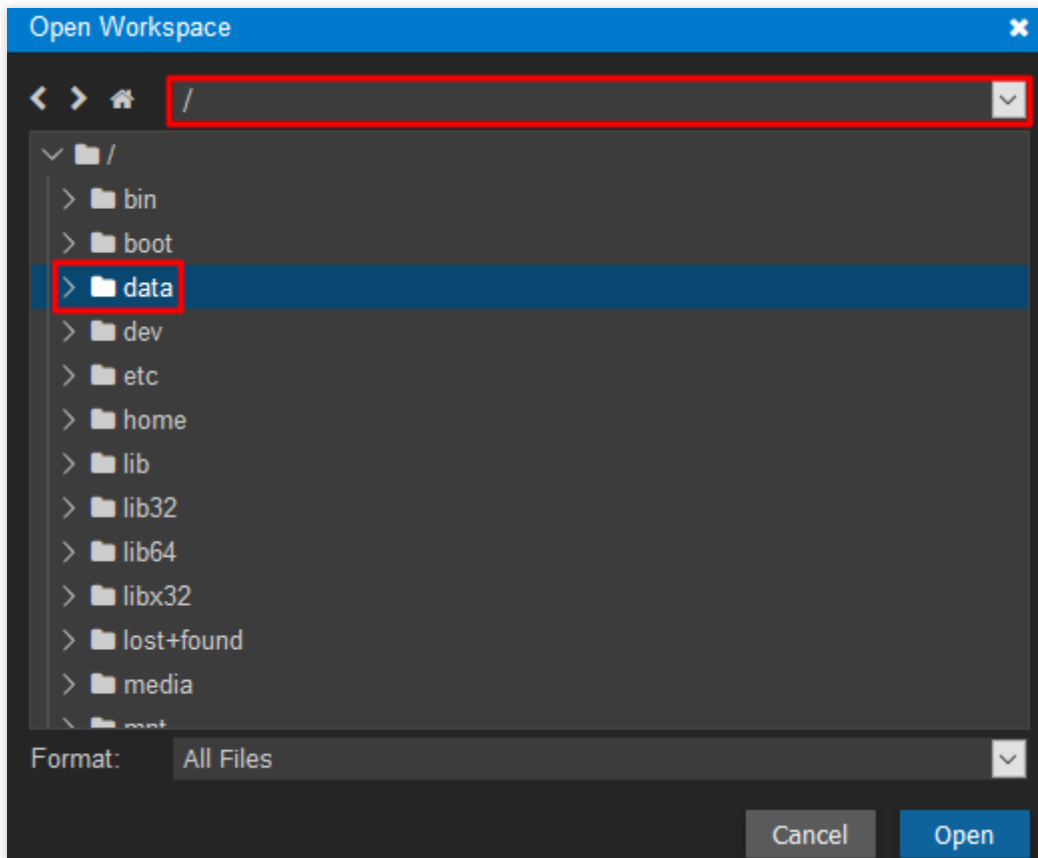
11. In the pop-up window, enter the admin account and password obtained in [step 8](#) and click **OK**.

After successful verification, you can enter the Theia IDE GUI.

## Subsequent Operations

### Selecting the workspace

1. Select **Open Workspace** on the Theia IDE Getting Started page.
2. In the **Open Workspace** pop-up window, select `/` from the drop-down list to open the directory. In Theia IDE, a directory is a workspace. `/data` is used as an example in this document.



3. Click **Open** to enter the `/data` workspace.

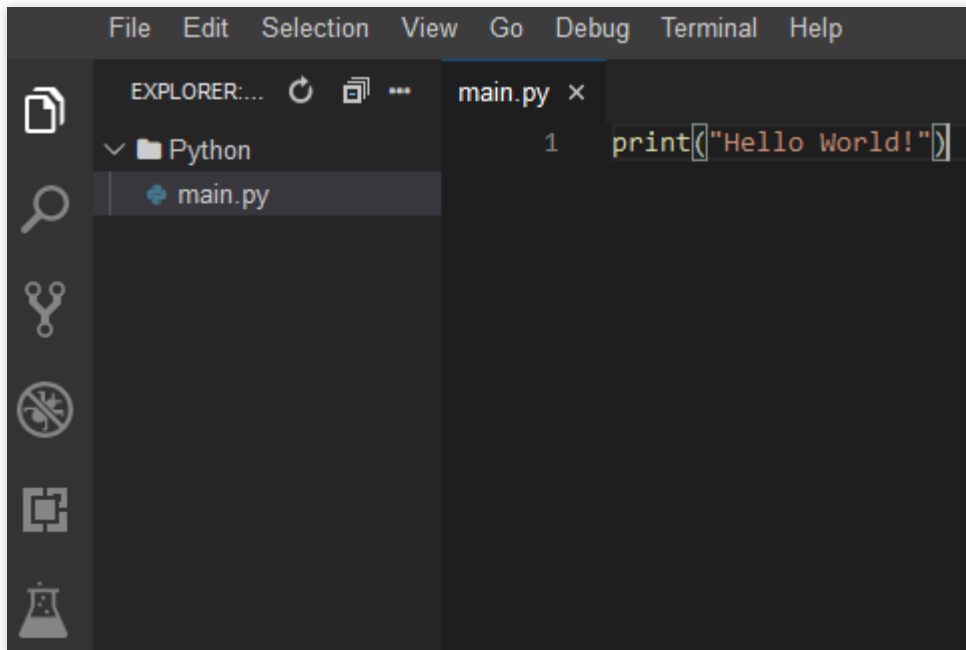
### Examples

**Note:**

Theia IDE supports Python, Java, Go, C/C++, and Node.js languages. Sample programs in Python, Go, and C++ are run on the command line and in the GUI here.

## Python

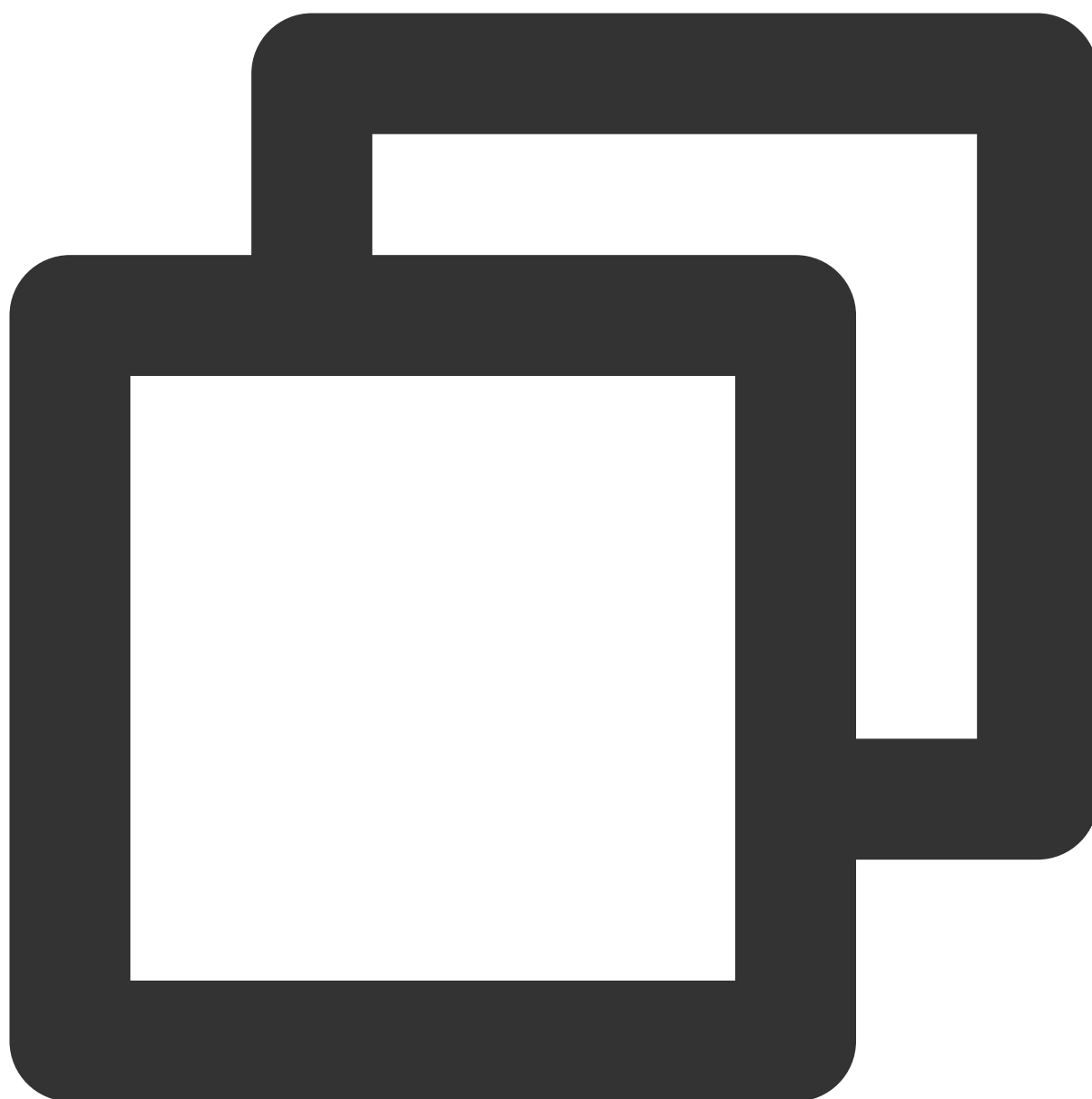
1. In the workspace, select **File > New Folder** at the top of the window.
2. In the pop-up window, create a folder named `Python` and a simple sample file `main.py` under it.



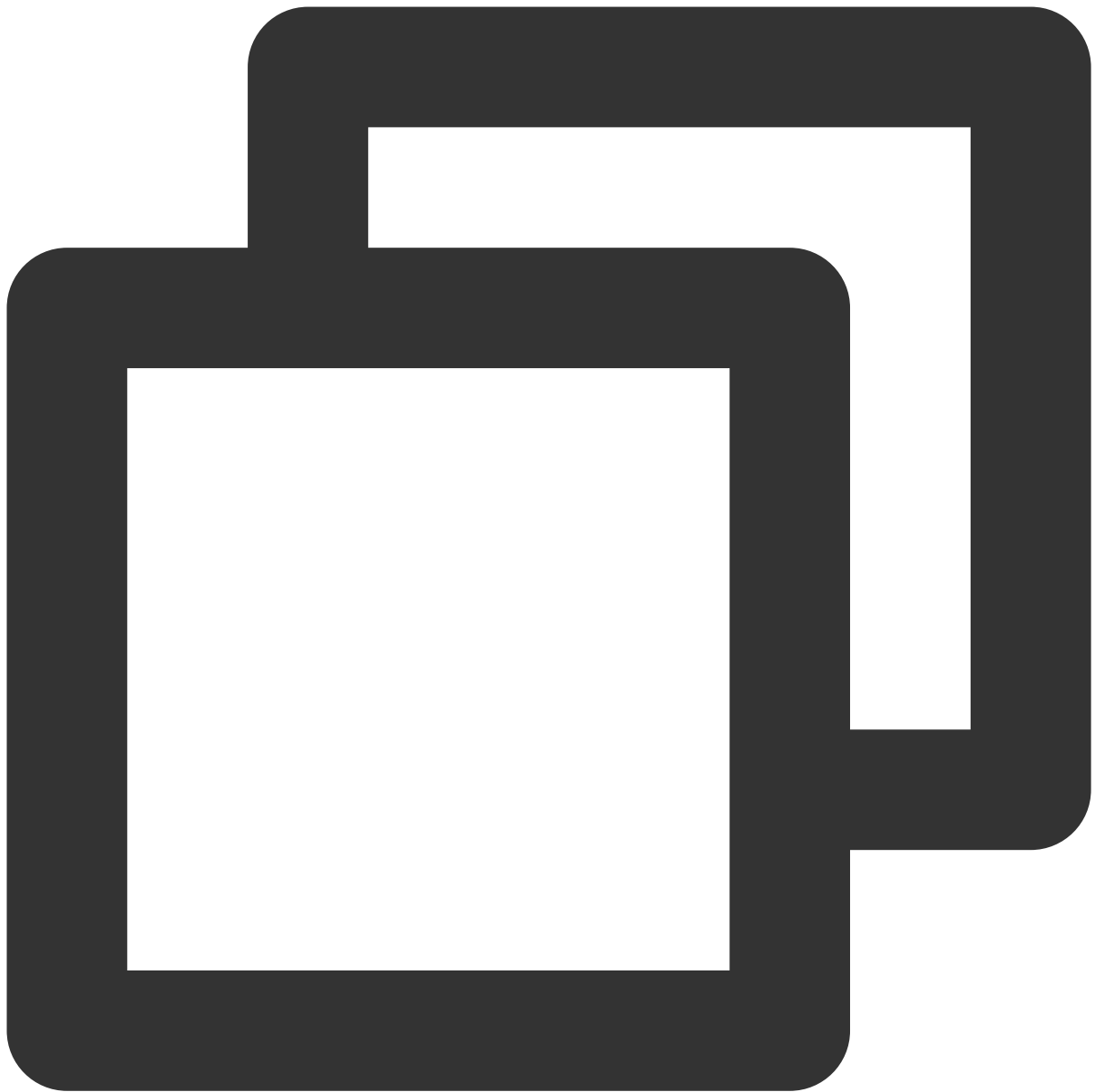
3. You can run the program in either of the following ways:

Command line:

- 3.1.1 Select Terminal > New Terminal at the top of the window to open a terminal.
- 3.1.2 Run the following commands in sequence in the terminal to run the program.



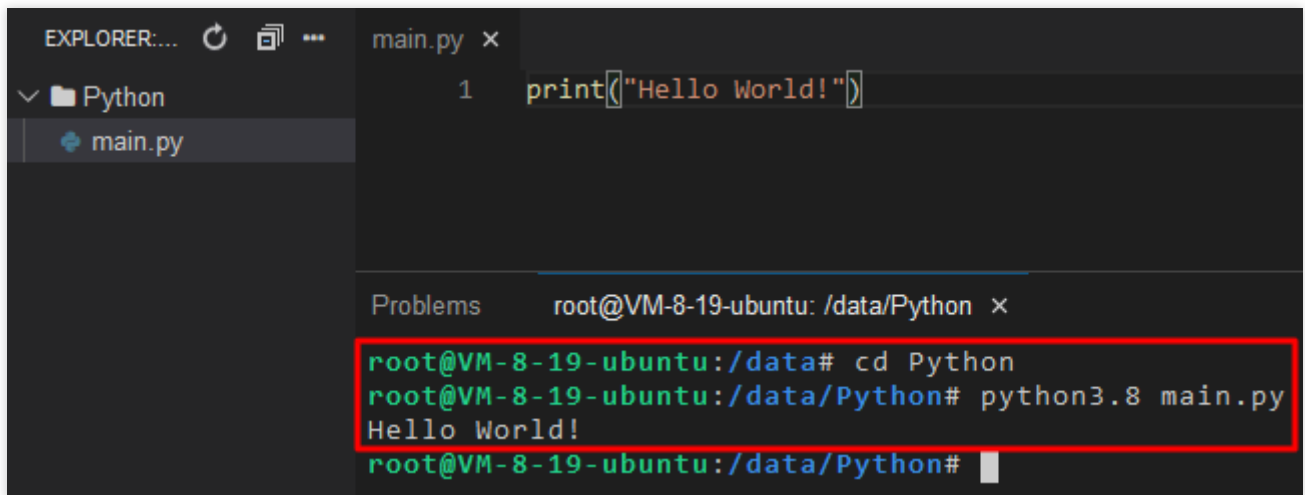
```
cd Python
```



```
python3.8 main.py
```

The execution result is as shown below:





```
EXPLORER:... Python main.py x
1 print("Hello World!")

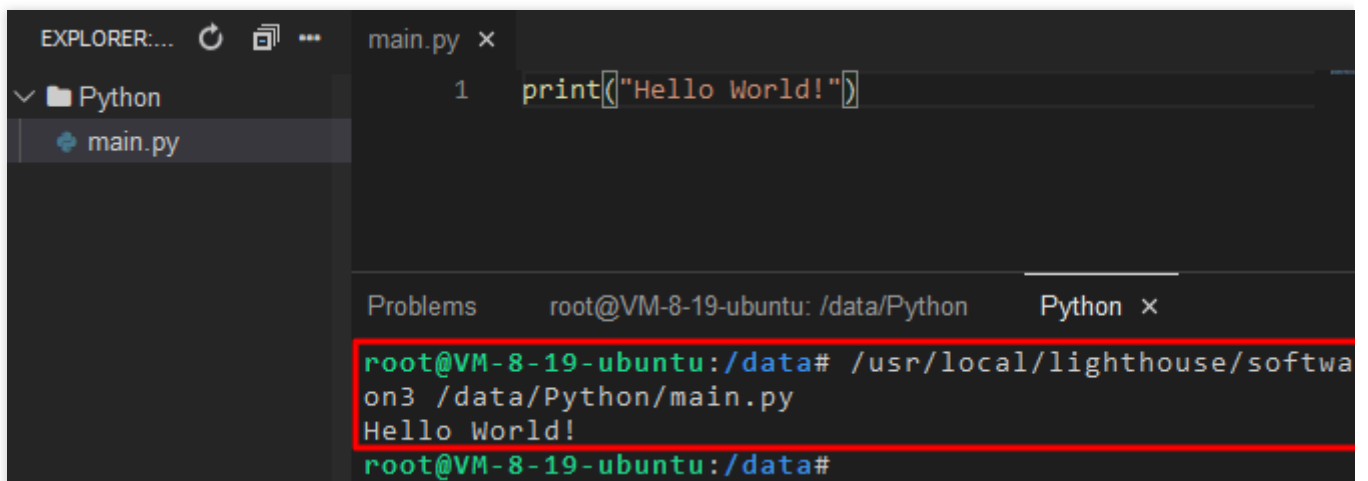
Problems root@VM-8-19-ubuntu: /data/Python x
root@VM-8-19-ubuntu:/data# cd Python
root@VM-8-19-ubuntu:/data/Python# python3.8 main.py
Hello World!
root@VM-8-19-ubuntu:/data/Python#
```

GUI:

Click



in the top-right corner of the window to run the program. The execution result is as shown below:

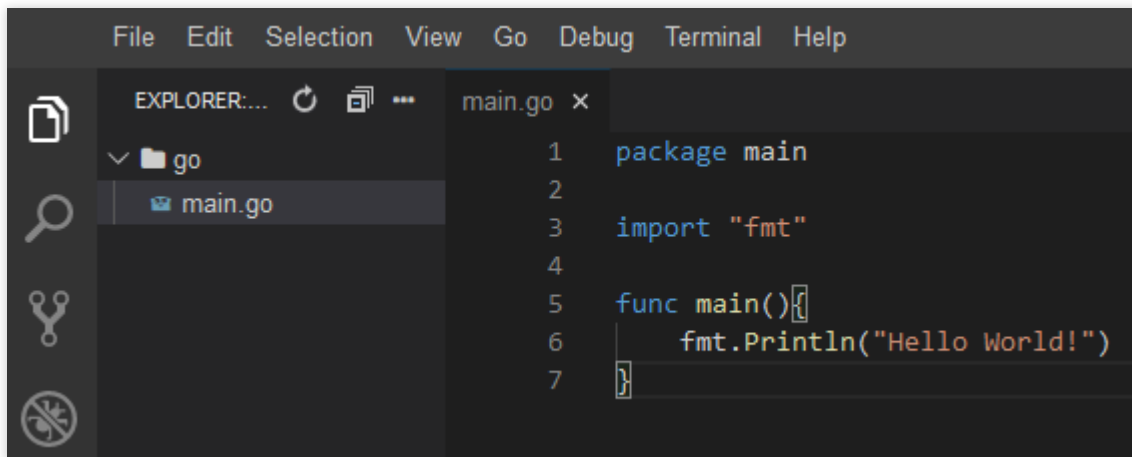


```
EXPLORER:... Python main.py x
1 print("Hello World!")

Problems root@VM-8-19-ubuntu: /data/Python Python x
root@VM-8-19-ubuntu:/data# /usr/local/lighthouse/software3 /data/Python/main.py
Hello World!
root@VM-8-19-ubuntu:/data#
```

## Go

1. In the workspace, select **File > New Folder** at the top of the window.
2. In the pop-up window, create a folder named `go` and a simple sample file `main.go` under it.

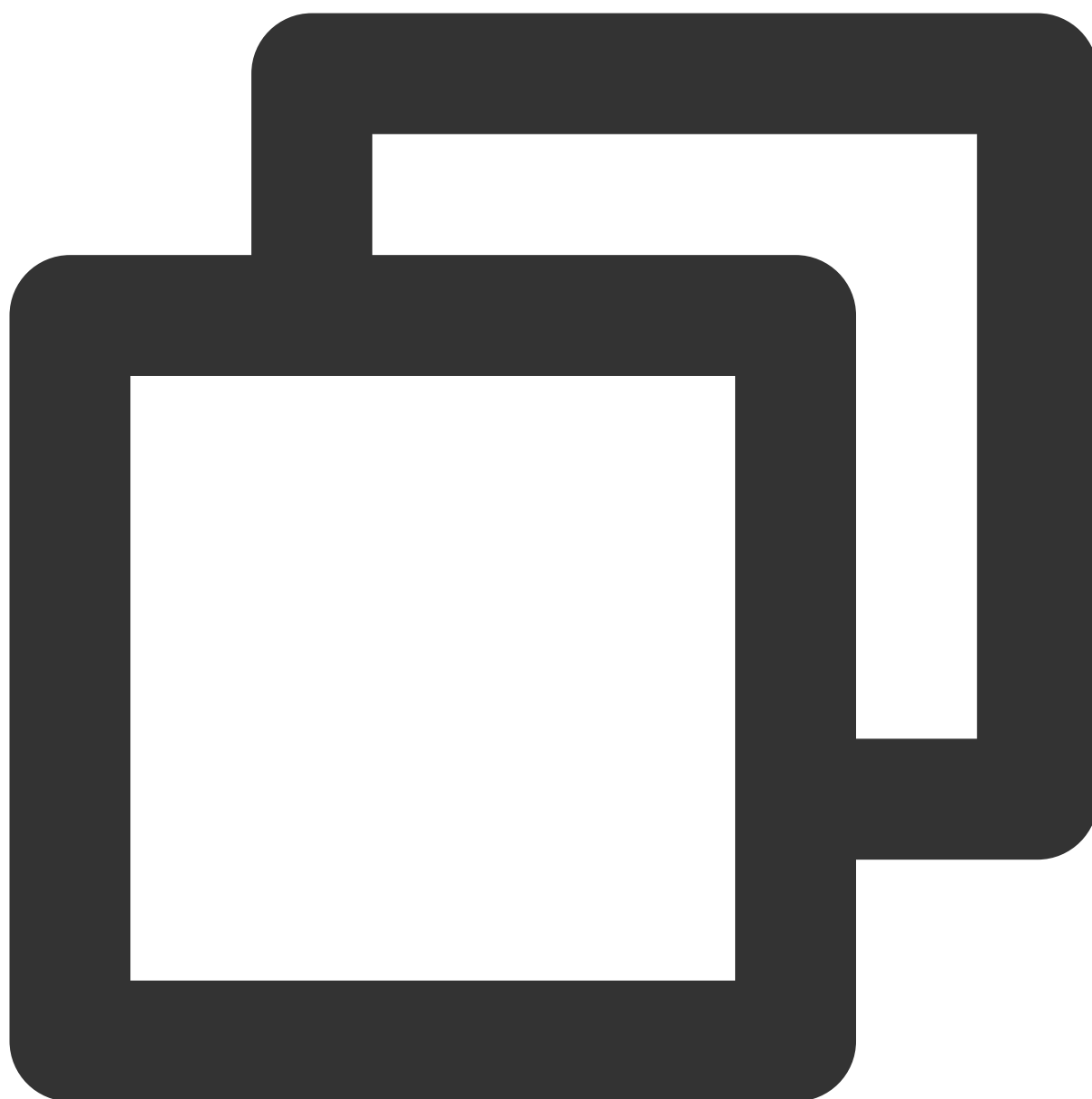


3. You can run the program in either of the following ways:

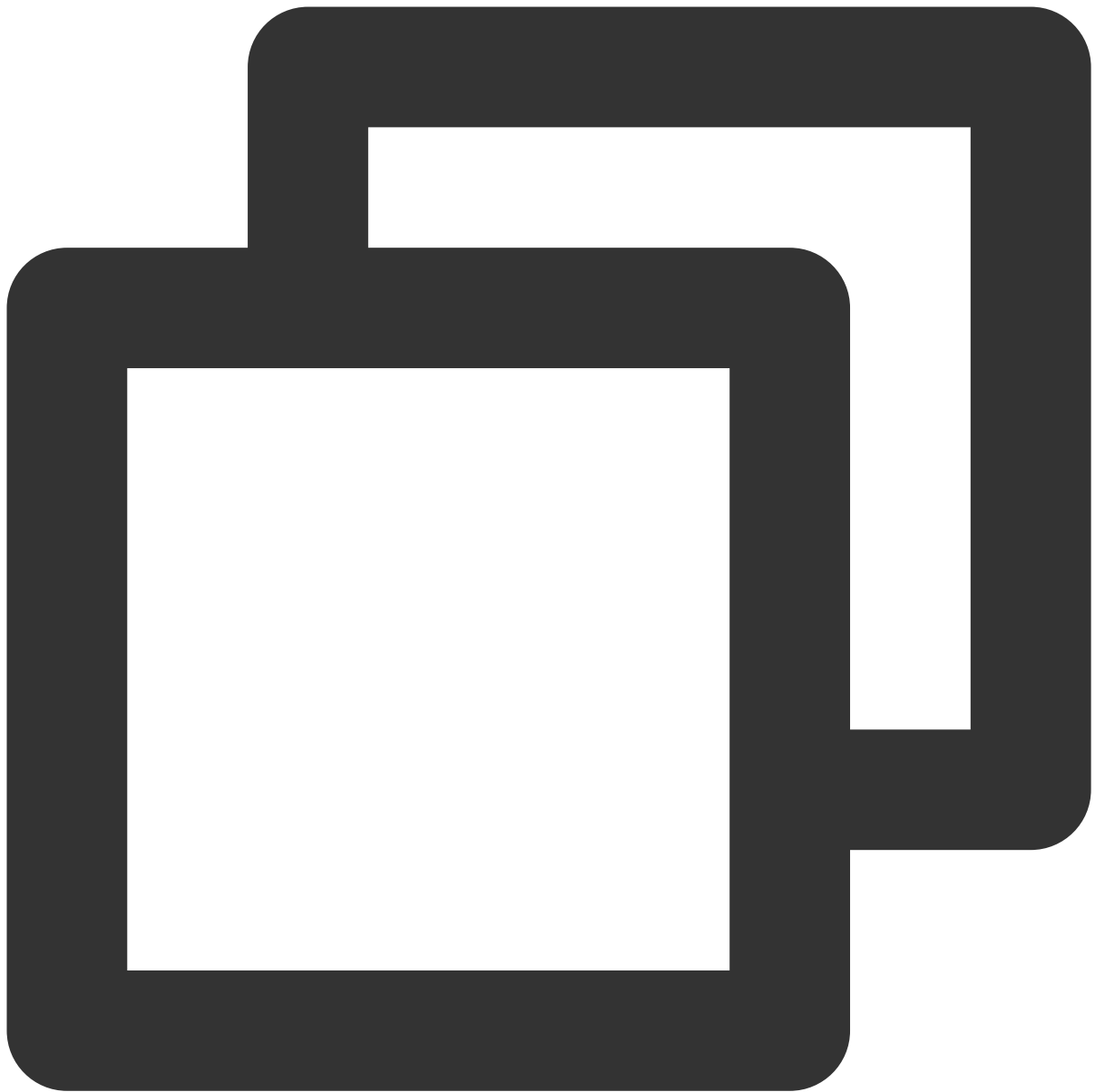
Command line:

3.1.1 Select Terminal > New Terminal at the top of the window to open a terminal.

3.1.2 Run the following commands in sequence in the terminal to run the program.

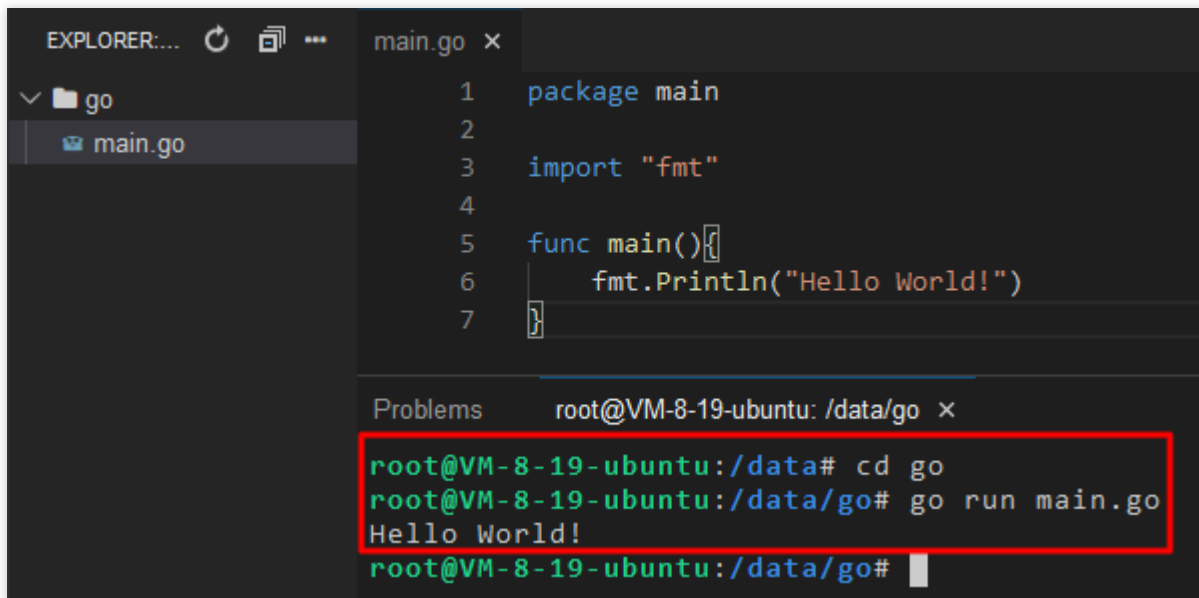


cd go



```
go run main.go
```

The execution result is as shown below:



The screenshot shows an IDE with a file explorer on the left displaying a folder 'go' containing 'main.go'. The main editor shows the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main(){
6     fmt.Println("Hello World!")
7 }
```

Below the code editor is a terminal window titled 'root@VM-8-19-ubuntu: /data/go x'. The terminal output is as follows:

```
root@VM-8-19-ubuntu:/data# cd go
root@VM-8-19-ubuntu:/data/go# go run main.go
Hello World!
root@VM-8-19-ubuntu:/data/go#
```

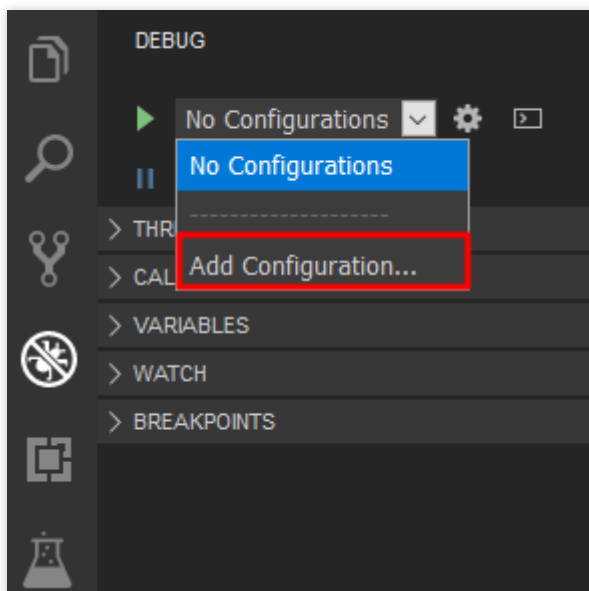
GUI:

### 3.2.1 Click



on the left to open the **DEBUG** section.

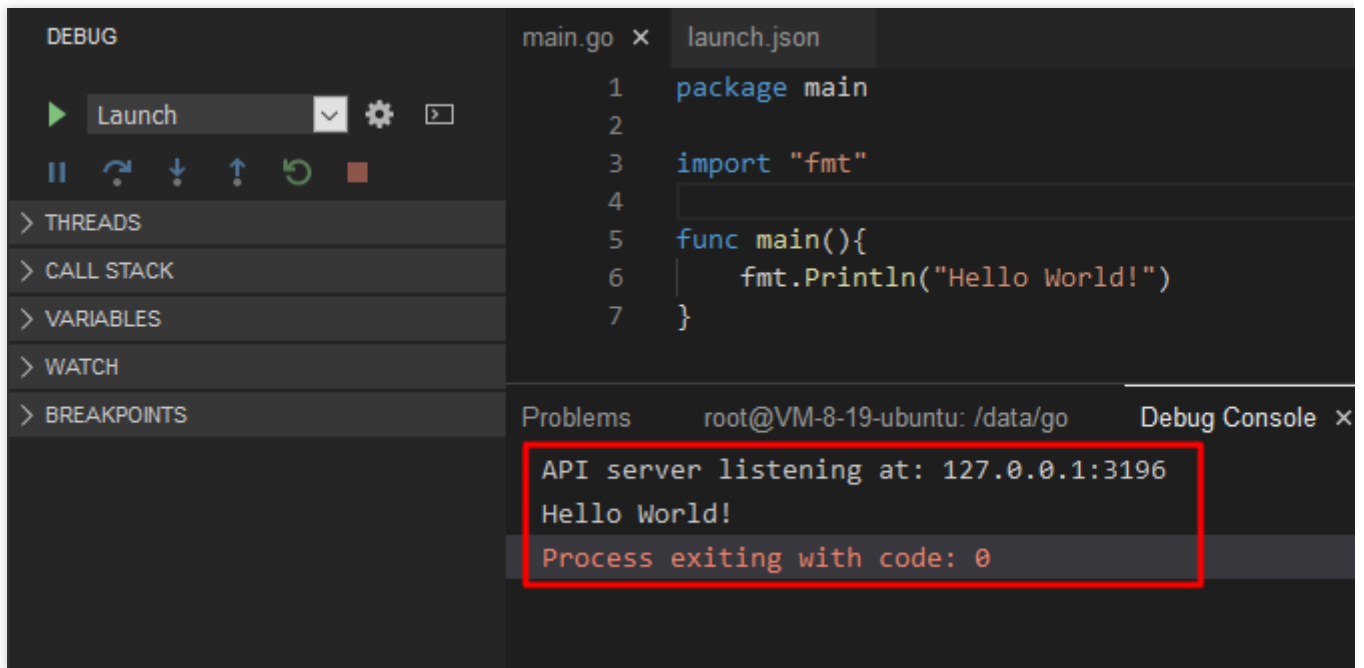
3.2.2 In **DEBUG**, select Add Configuration from the drop-down list to generate the configuration file.



3.2.3 Open the main.go file and select

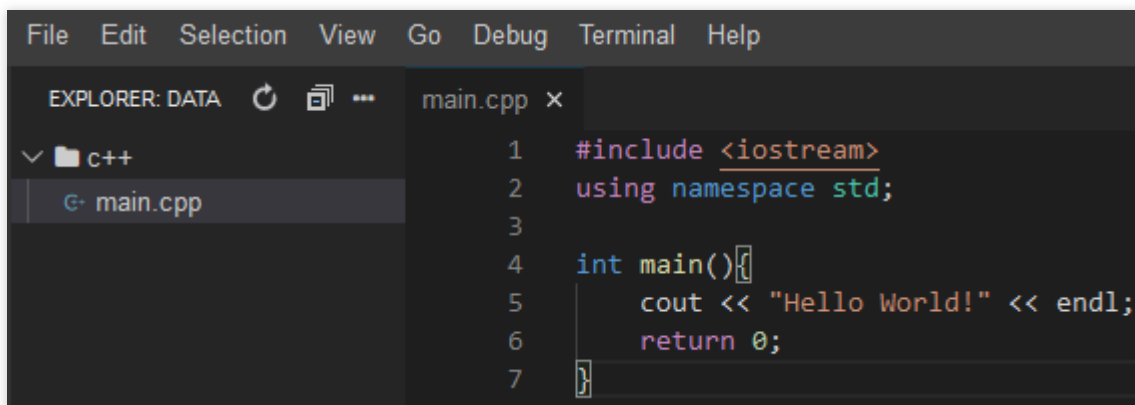


in the **\*\*DEBUG\*\*** section to run the program. The execution result is as shown below:



## C++

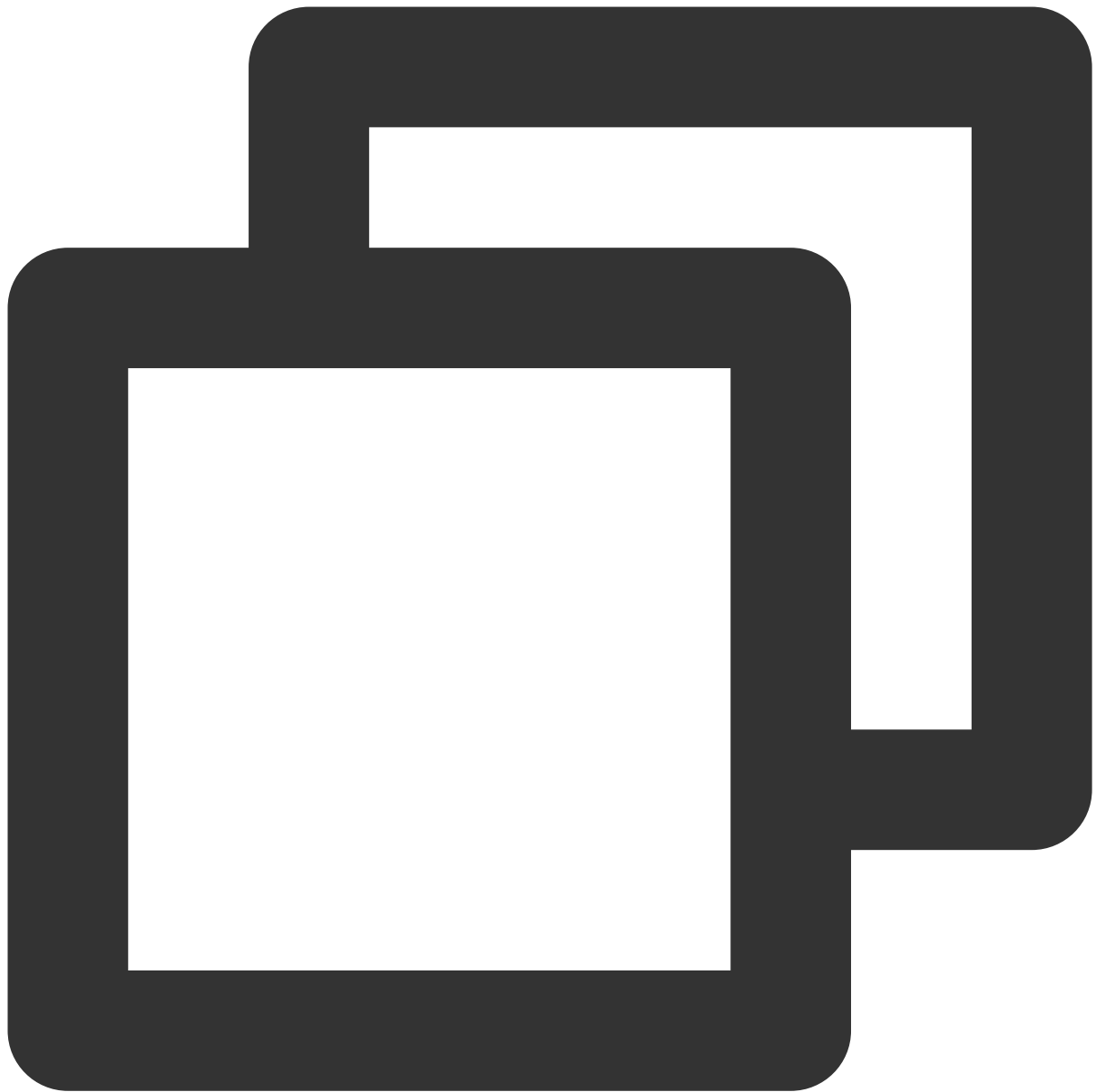
1. In the workspace, select **File > New Folder** at the top of the window.
2. In the pop-up window, create a folder named `c++` and a simple sample file `main.cpp` under it.



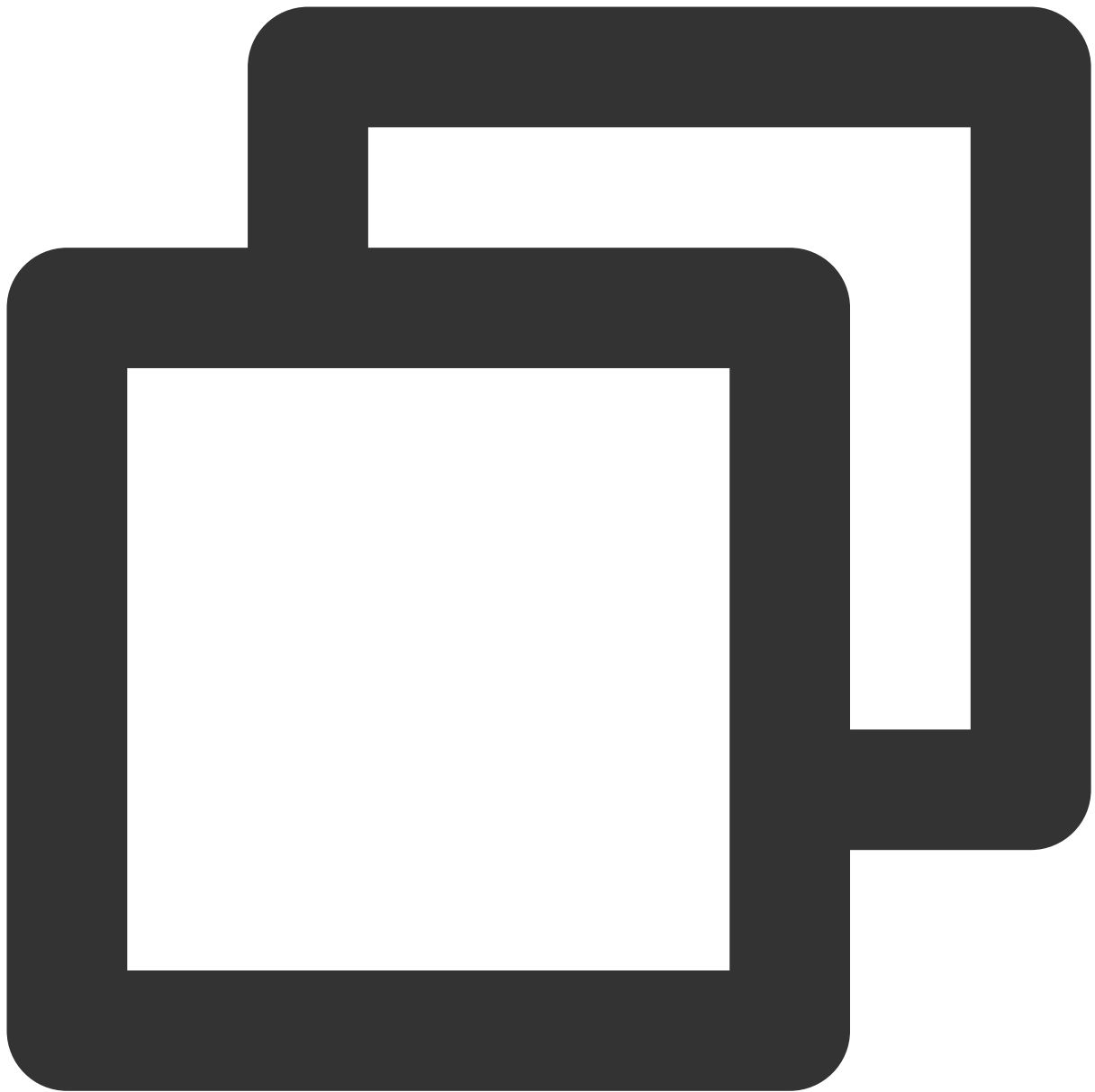
3. You can run the program in either of the following ways:

Command line:

- 3.1.1 Select **Terminal > New Terminal** at the top of the window to open a terminal.
- 3.1.2 Run the following commands in sequence in the terminal to run the program.

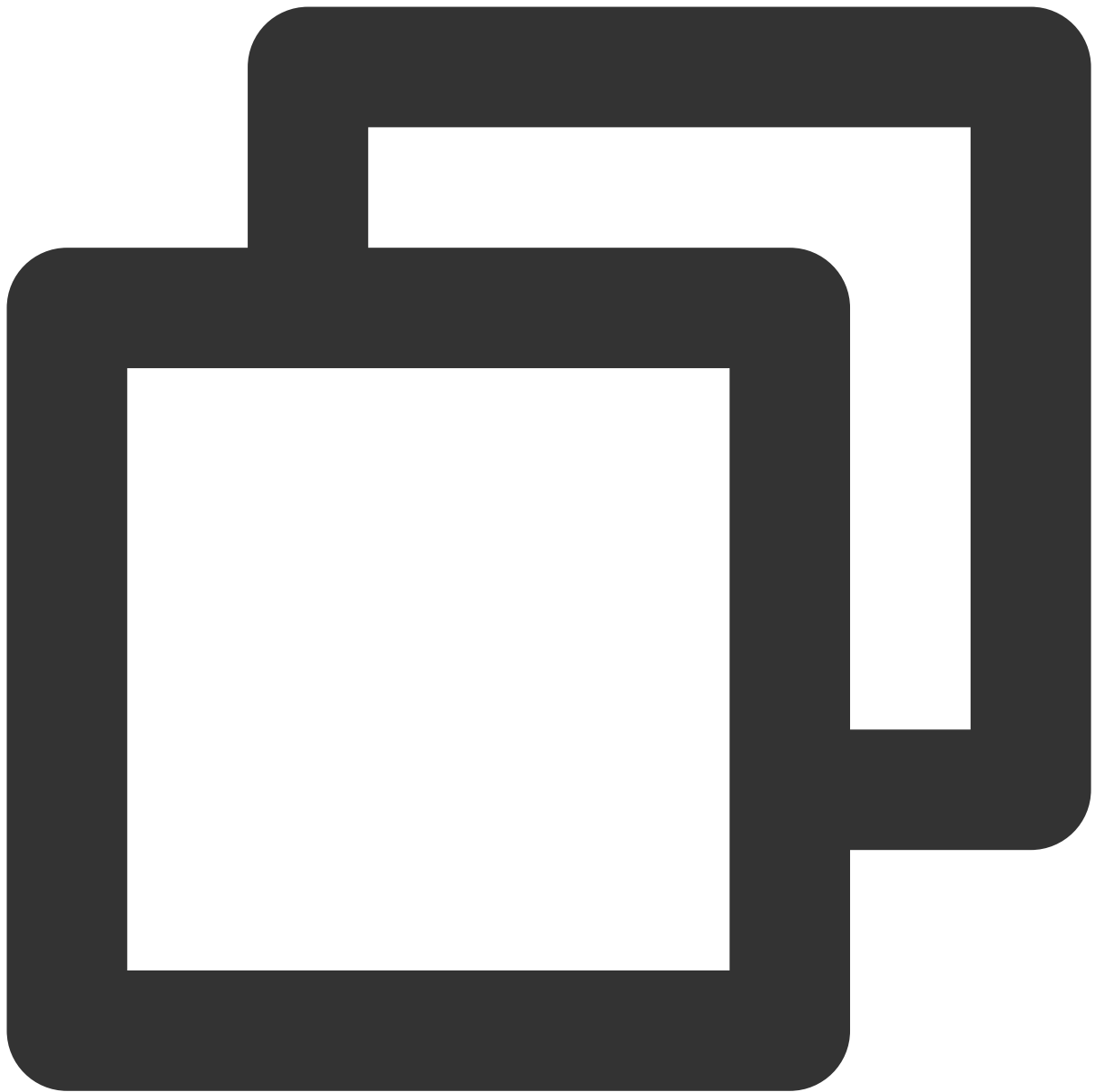


```
cd c++
```



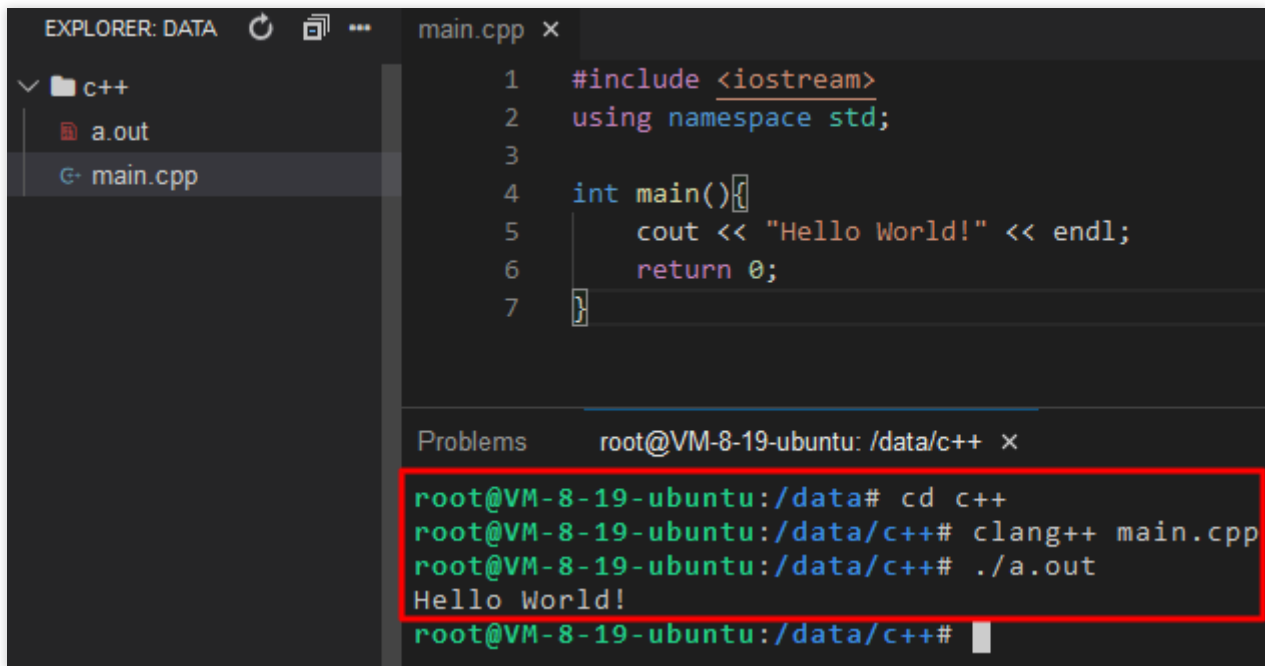
```
clang++ main.c
```





```
./a.out
```

The execution result is as shown below:



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a folder named 'c++' containing 'a.out' and 'main.cpp'. The code editor displays the contents of 'main.cpp', which is a simple C++ program that prints 'Hello World!'. The terminal shows the execution of the program, with the output 'Hello World!'.

```
EXPLORER: DATA  main.cpp x
c++
a.out
main.cpp

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Hello World!" << endl;
7      return 0;
8  }

Problems  root@VM-8-19-ubuntu: /data/c++ x
root@VM-8-19-ubuntu:/data# cd c++
root@VM-8-19-ubuntu:/data/c++# clang++ main.cpp
root@VM-8-19-ubuntu:/data/c++# ./a.out
Hello World!
root@VM-8-19-ubuntu:/data/c++#
```

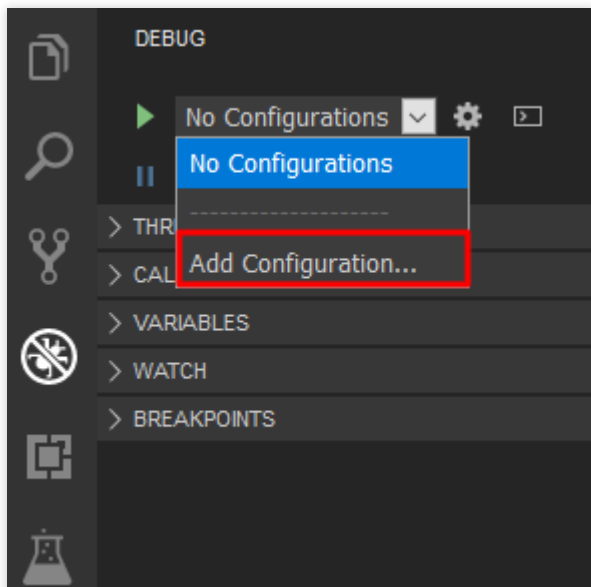
GUI:

### 3.2.1 Click



on the left to open the **DEBUG** section.

3.2.2 In **DEBUG**, select Add Configuration from the drop-down list to generate the configuration file as.



3.2.3 In the drop-down list of the configuration file, select { } GDB CDT Local debugging.

3.2.4 Replace /\${command:askProgramPath} in the configuration file with /c++/a.out and save the change.

3.2.5 Select

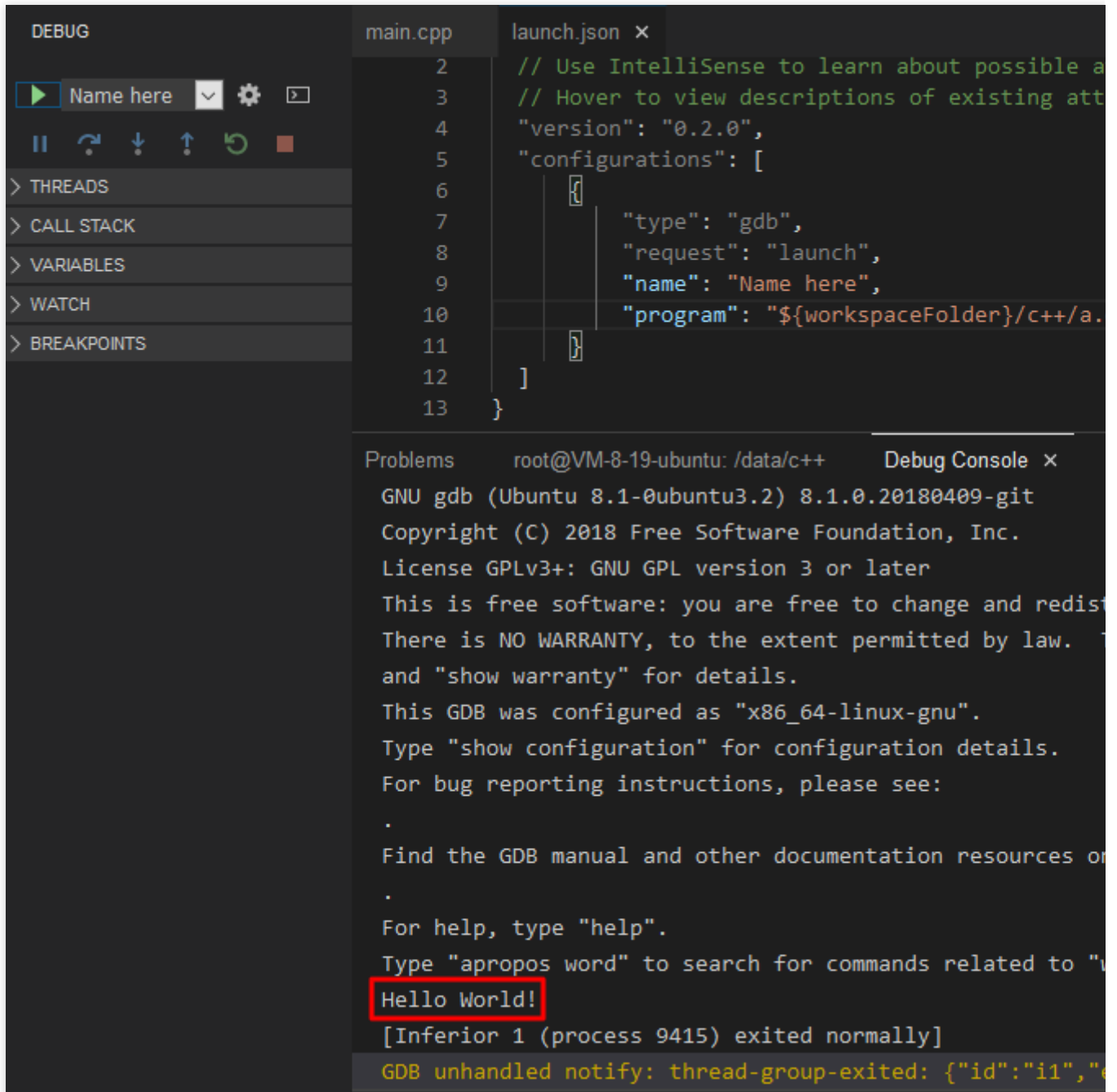


in the **DEBUG** section to open the Debug console.

### 3.2.6 Select



in the **DEBUG** section to run the program. The execution result is as shown below:



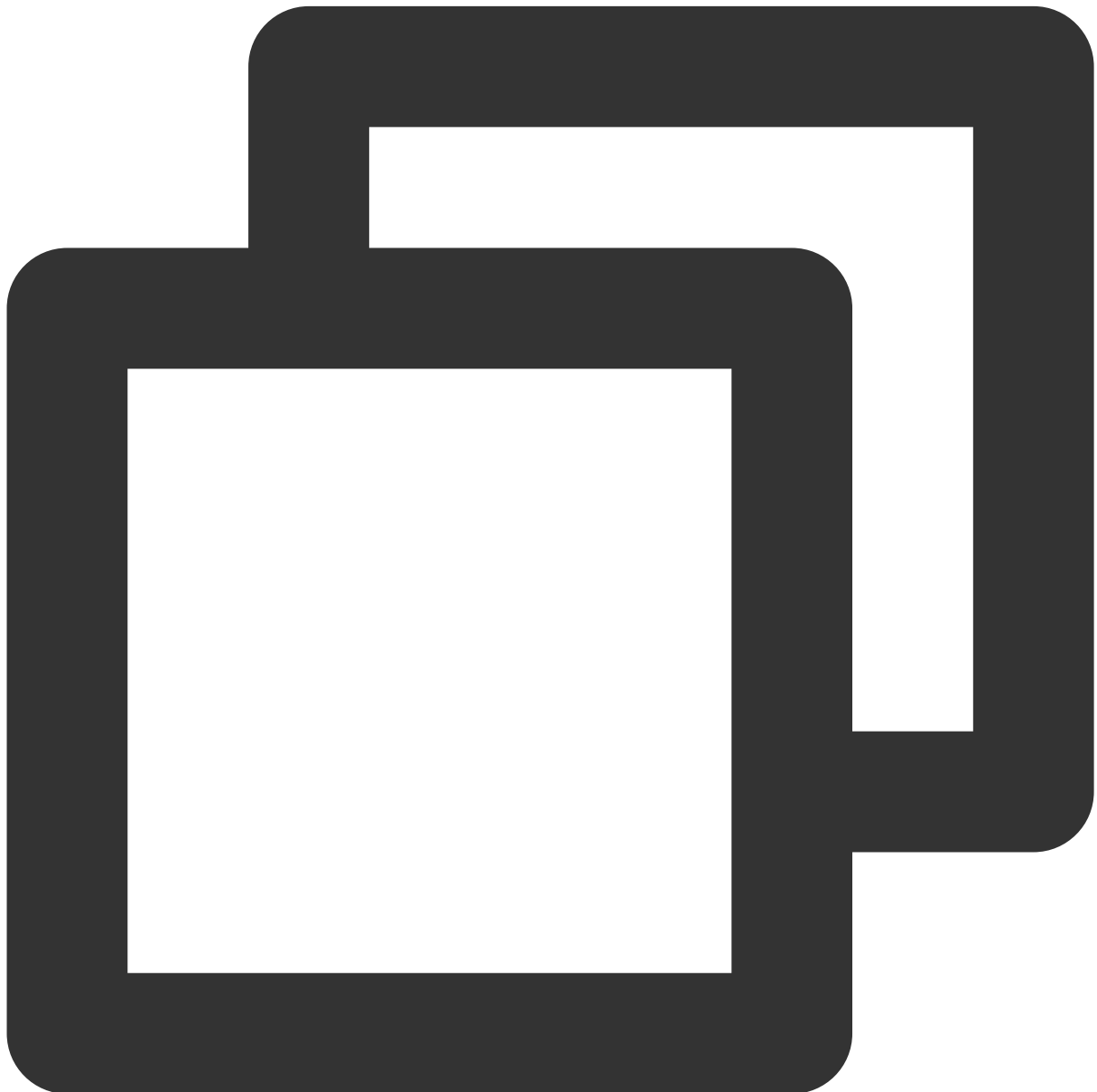
## Enabling HTTPS access

You can install an SSL certificate and enable HTTPS access for your Theia IDE instance as instructed in [Installing Certificate on NGINX Server](#).

**Note:**

You only need to modify the `/usr/local/lighthouse/softwares/nginx/conf/include/theia.conf` configuration file but not `/usr/local/lighthouse/softwares/nginx/conf/nginx.conf` for your Theia IDE instance.

See the following configuration to modify the file:



```
server {  
    listen 443 ssl;
```

```

server_tokens off;
keepalive_timeout 5;
root /usr/local/lighthouse/softwares/nginx/html;
index index.php index.html;
access_log logs/theia.log combinedio;
error_log logs/theia.error.log;
server_name cloud.tencent.com;    # Enter the domain name bound to your certific
ssl_certificate 1_cloud.tencent.com_bundle.crt;    # Enter the name of your cert
ssl_certificate_key 2_cloud.tencent.com.key;    # Enter the name of your privat
ssl_session_timeout 5m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;    # You can see this SSL protocol for confi
ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;    # You can
ssl_prefer_server_ciphers on;

auth_digest_user_file /home/lighthouse/passwd.digest;
auth_digest_shm_size 8m;    # the storage space allocated for tracking active s

location / {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";

    auth_digest 'lighthouse';
    auth_digest_timeout 120s;    # allow users to wait 2 minute between receivin
                                # challenge and hitting send in the browser dia
    auth_digest_expires 600s;    # after a successful challenge/response, let th
                                # continue to use the same nonce for additional
                                # for 600 seconds before generating a new chall
    auth_digest_replays 60;    # also generate a new challenge if the client u
                                # same nonce more than 60 times before the expi

    proxy_pass http://127.0.0.1:3000;
}
}

server {
    listen 80;
    server_name cloud.tencent.com;    # Enter the domain name bound to your certifi
    return 301 https://$host$request_uri;    # Redirect HTTP requests to HTTPS
}

```

&lt;style&gt;

.roman{

list-style-type:lower-roman

}

&lt;/style&gt;

# Building LAMP Development Environment

Last updated : 2022-06-16 16:39:53

## Overview

Linux, Apache, MySQL, PHP (LAMP) is an internationally popular web application framework. It encompasses the Linux OS, Apache web server, MySQL/MariaDB database, PHP environment, and related components.

**Note:**

The LAMP application image is based on CentOS 7.6 64-bit.

## Directions

1. Log in to the [Lighthouse console](#).

Region ⓘ

Hong Kong
Singapore
Tokyo
Silicon Valley
Frankfurt
Mumbai

Lighthouse instances in different regions cannot communicate with each another over a private network. Selecting the region closest to your end users can mini improve download speed. You cannot change the region after creating a Lighthouse instance. [Region and connectivity](#)

Availability zone ⓘ

☒ Randomly assigned ⓘ

Image

Official image
Individual image

Application image
System image

**SRS Streaming Server 4.4**

**WordPress 5.7.1**

**Typecho 1.1.0**

**Cloud**

**Matomo 4.9.1**

**LAMP 7.4.16**

**Node.js 14.16.1**

**Th**

**Docker 19.03.9**

**K3s 1.23.6**

**ASP.NET 4.8**

**SRS Streaming Server 4.4**

SRS is the popular open source audio and video server, mainly used in live streaming and WebRTC, supporting RTMP, WebRTC, HLS, HTTP-FLV and SRT protocols, the Star and the most active in the streaming server industry, with users distributed worldwide. SRS entered the Mulan community incubation in 2021 driven open source project and gradually in build an open source solution for audio and video to make audio and video development easy. The image is base on operating system.

Instance bundle ⓘ

General
Enterprise

**5 USD/month**

CPU 2 cores (dedicated)  
Memory 2GB  
System disk 30GB SSD  
Bandwidth 30Mbps  
Transfer 1024 GB/month

**7 USD/month**

CPU 2 cores (dedicated)  
Memory 2GB  
System disk 50GB SSD  
Bandwidth 30Mbps  
Transfer 2048 GB/month

**9 USD/month**

CPU 2 cores (dedicated)  
Memory 4GB  
System disk 60GB SSD  
Bandwidth 30Mbps  
Transfer 2560 GB/month

**11 USD/month**

CPU 2 cores (dedicated)  
Memory 4GB  
System disk 80GB SSD  
Bandwidth 30Mbps  
Transfer 3072 GB/month

**16 USD/month**

CPU 2 cores (dedicated)  
Memory 8GB  
System disk 90GB SSD  
Bandwidth 30Mbps  
Transfer 3584 GB/month

**22 USD/month**

CPU 2 cores (dedicated)  
Memory 8GB  
System disk 100GB SSD  
Bandwidth 30Mbps  
Transfer 4096 GB/month

An independent fixed public IP is assigned for free. The public network outbound traffic beyond the transfer quota will incur additional fees. [View pricing](#)

Instance name

Optional. Defaults to "image name-four random characters" if it's left empty

The multiple instances created in batch will be suffixed with s default. You can enter 60 characters

Purchase period

1 month
2
3
6 months
1 year
2 years
3 years
4 years
5 years
More

☐ Auto-renew the device every month when my account has sufficient balance

Quantity

- 1 +

**Region:** Select a region near your target users to reduce the network latency and improve their access speed.

**Availability zone:** **Randomly assigned** is selected by default. You can select one as well.

**Image:** Select the **LAMP 7.4.16** application image.

**Instance bundle:** Select an instance bundle according to the required instance configuration (including CPU, memory, system disk, bandwidth, and monthly traffic).

**Instance name:** Enter a custom instance name. If it is left empty, the selected image name will be used as the name by default. When multiple instances are created in a batch, their names will be consecutive with auto-incrementing suffixes. For example, if you enter "LH" as the name and create three instances, the three instances are named "LH1", "LH2", and "LH3".

**Purchase period:** Default to **1 month**.

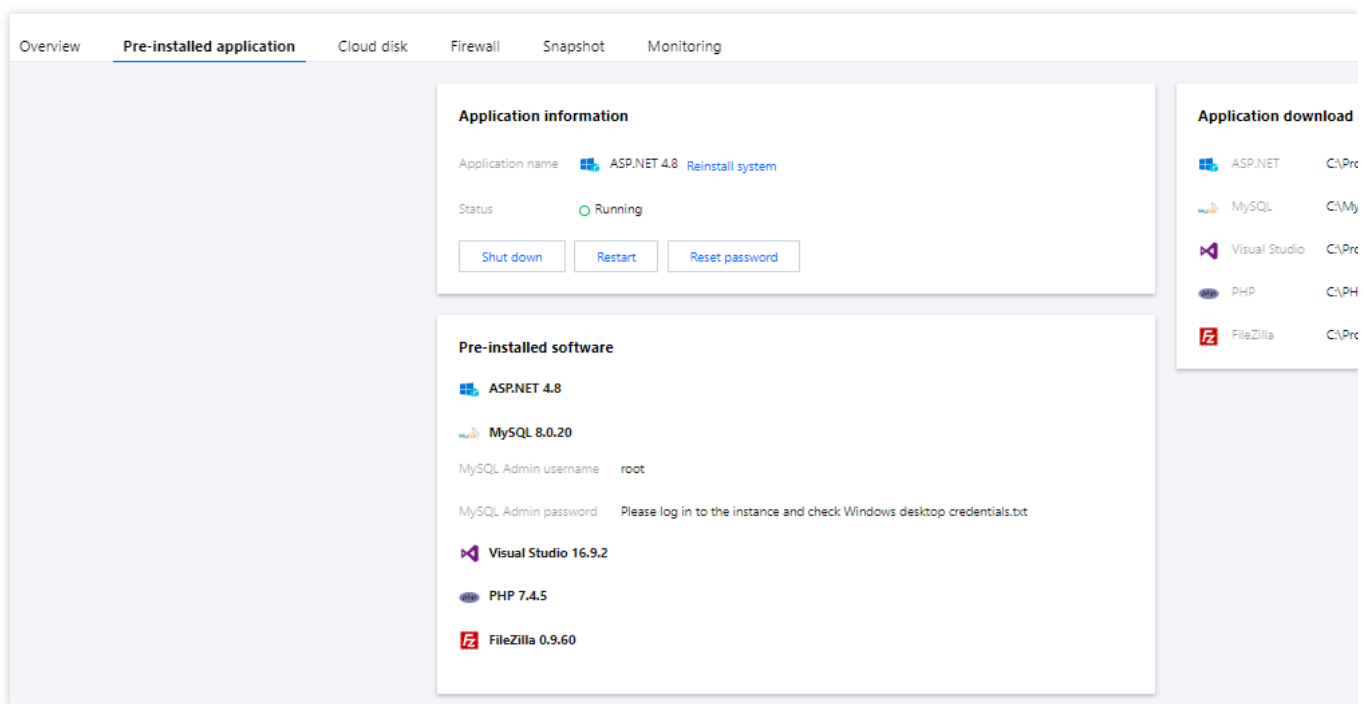
**Quantity:** Default to **1**.

2. Click **Buy now** to submit your order and make the payment as prompted.

## Relevant Operations

### Viewing LAMP configuration items

1. Log in to the [Lighthouse console](#).
2. In the instance list, select the target instance created with the LAMP application image to enter its details page.
3. Select the **Pre-installed application** tab to enter the application details page.



You can view the configuration items of the LAMP application.

Apache's homepage address and website root directory.

MariaDB's admin account (root) and password, database address, and database name.



**Note:**

You can get the admin password by logging in to the instance via webshell and running the `cat ~lighthouse/credentials.txt` command.

Apache, MariaDB, and PHP software installation paths on CentOS.

**Note:**

Access `http://LAMP instance public IP/phpinfo.php` to view the PHP configuration information.

**Enabling HTTPS access**

Install the SSL certificate and enable HTTPS access for your LAMP instance as instructed in [Apache Server Certificate Installation](#).

# Building Node.js Development Environment

Last updated : 2022-06-16 16:42:25

## Overview

Node.js is an event-driven I/O server-side JavaScript environment based on the Chrome V8 engine. Fast and powerful, it can be used to build all kinds of web applications and work as the backend service environment for mini programs.

### Note:

In the following example, we use the Node.js application image, which is based on CentOS 8.2 64-bit. Note that application images are subject to updates from time to time, and the actual image information on the purchase page shall prevail.

## Directions

1. Log in to the [Lighthouse console](#) and configure the following parameters:

**Region and Availability zone:** Select the region and AZ near your target users to reduce the network latency and improve their access speed.

**Image:** Select the **Node.js** application image.

**Instance bundle:** Select an instance bundle according to the required instance configuration (including CPU, memory, system disk, bandwidth, and monthly traffic).

**Instance name:** Enter a custom instance name. If it is left empty, the selected image name will be used as the name by default. When multiple instances are created in a batch, their names will be consecutive with auto-incrementing suffixes. For example, if you enter "LH" as the name and select 3 as the quantity, 3 instances "LH1", "LH2", and "LH3" will be created.

**Purchase period:** One month by default.

**Quantity:** One instance by default.

2. Click **Buy now** to submit your order and make the payment as prompted.

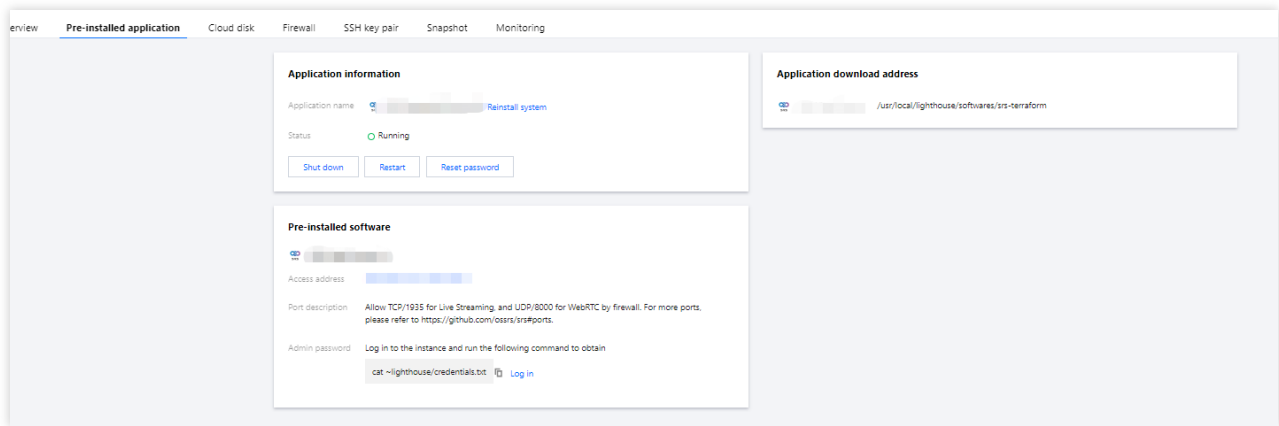
## Relevant Operations

### Viewing application information

1. Log in to the [Lighthouse console](#).

2. In the instance list, select the target instance created with the Node.js application image to enter its details page.

3. Select the **Pre-installed application** tab to enter the application details page.



You can view the configuration items of pre-installed applications.

Node.js and NGINX software installation paths.

The npm and npx paths of Node.js, and the primary configuration file path of NGINX.

## Using FTP tool to upload code and debug

1. Log in to the instance created with the Node.js application image. Set up the FTP service as instructed in [Building FTP Service using Linux Instance](#).
2. Use an FTP tool (such as WinSCP) on your local computer to upload your website code to the instance and test and debug the service.

## Enabling HTTPS access

You can install an SSL certificate and enable HTTPS access for your website. See [Installing Certificate on NGINX Server](#).

# Building ASP.NET Development Environment

Last updated : 2022-06-16 16:43:56

## Overview

The ASP.NET application image provides an open-source server-side web application framework for building dynamic webpages, applications, and services.

### Note:

The underlying layer of the sample ASP.NET application image in this document is based on Windows Server 2019. Application images are subject to updates from time to time, and the actual image information on the purchase page shall prevail.

This image requires an SSD system disk of at least 50 GB.

## Directions

1. Log in to the [Lighthouse console](#) and configure the following parameters:

**Region and Availability zone:** Select the region and AZ near your target users to reduce the network latency and improve their access speed.

**Image:** Select the **ASP.NET** application image.

**Instance bundle:** Select an instance bundle according to the required instance configuration (including CPU, memory, system disk, bandwidth, and monthly traffic).

**Instance name:** Enter a custom instance name. If it is left empty, the selected image name will be used as the name by default. When multiple instances are created in a batch, their names will be consecutive with auto-incrementing suffixes. For example, if you enter "LH" as the name and create three instances, the three instances are named "LH1", "LH2", and "LH3".

**Purchase period:** Default to **1 month**.

**Quantity:** Default to **1**.

2. Click **Buy now** to submit your order and make the payment as prompted.

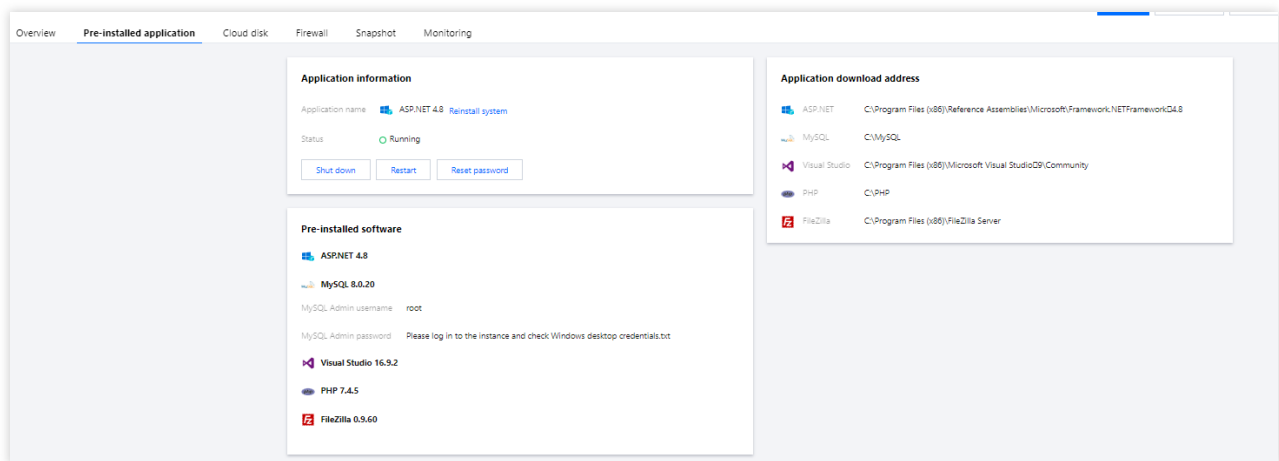
## Relevant Operations

### Viewing instance configuration items

1. Log in to the [Lighthouse console](#).

2. In the instance list, select the target instance created with the ASP.NET application image to enter its details page.

3. Select the **Pre-installed application** tab to enter the application details page.



You can view the configuration items of pre-installed applications.

ASP.NET, MySQL, and Visual Studio software installation paths.

MySQL admin account and login password.

## Using FTP tool to upload code and debug

The ASP.NET application image contains FileZilla, which allows you to connect to your local computer, upload your website code, and perform testing and debugging.

## Enabling HTTPS access

Install the SSL certificate and enable HTTPS access for your website as instructed in [Selecting an Installation Type for an SSL Certificate](#).

# Building Docker Container Environment with Template

Last updated : 2023-02-16 15:46:36

## Overview

As one of the most popular open-source container engine, Docker allows developers to package applications and dependencies into lightweight, portable containers conveniently and efficiently for faster application delivery, deployment, migration, and scaling. This document describes how to use the Docker CE application image to build a Docker container environment. The Docker image source has been set to Tencent Cloud Docker image source by default, which can accelerate Docker image downloads.

### Note:

The underlying layer of the sample Docker CE image in this document is based on CentOS 7.6 64-bit. Application images are subject to updates from time to time, and the actual image information on the purchase page shall prevail.

## Directions

1. Log in to the [Lighthouse console](#) and configure the following parameters:

**Region and Availability zone:** Select the region and AZ near your target users to reduce the network latency and improve their access speed.

**Image:** Select **Official image > Docker CE application image**.

**Instance bundle:** Select an instance bundle according to the required instance configuration (including CPU, memory, system disk, bandwidth, and monthly traffic).

**Instance name:** Enter a custom instance name. If it is left empty, the selected image name will be used as the name by default. When multiple instances are created in a batch, their names will be consecutive with auto-incrementing suffixes. For example, if you enter "LH" as the name and create three instances, the three instances are named "LH1", "LH2", and "LH3".

**Purchase period:** Default to **1 month**.

**Quantity:** Default to **1**.

2. Click **Buy now** to submit your order and make the payment as prompted.

## Relevant Operations

### Enabling HTTPS access

You can install an SSL certificate and enable HTTPS access for your website as instructed in [Installing Certificate on NGINX Server](#).

# Managing K3s Container Cluster with Image

Last updated : 2022-06-16 16:48:15

## Overview

This document describes how to build a Kubernetes cluster management environment by using the K3s application image. K3s is an open-source, extremely lightweight Kubernetes distribution. Currently, it is a Cloud Native Computing Foundation (CNCF) sandbox project. It has low requirements for server computing resources and can run in standalone mode. The K3s application image is preconfigured with the Kubernetes Dashboard visualization tool for easy Kubernetes cluster management in a browser.

### Note:

The underlying layer of the sample K3s image in this document is based on CentOS 8.2 64-bit. Application images are subject to updates from time to time, and the actual image information on the purchase page shall prevail.

## Directions

### Using K3s image to create instance

1. Log in to the [Lighthouse console](#) and configure the following parameters:

**Region** and **Availability zone**: Select the region and AZ near your target users to reduce the network latency and improve their access speed.

**Image**: Select the **K3s** application image.

**Instance bundle**: Select an instance bundle according to the required instance configuration (including CPU, memory, system disk, bandwidth, and monthly traffic).

**Instance name**: Enter a custom instance name. If it is left empty, the selected image name will be used as the name by default. When multiple instances are created in a batch, their names will be consecutive with auto-incrementing suffixes. For example, if you enter "LH" as the name and create three instances, the three instances are named "LH1", "LH2", and "LH3".

**Purchase period**: Default to **1 month**.

**Quantity**: Default to **1**.

2. Click **Buy now** to submit your order and make the payment as prompted.

### Configuring Lighthouse firewall

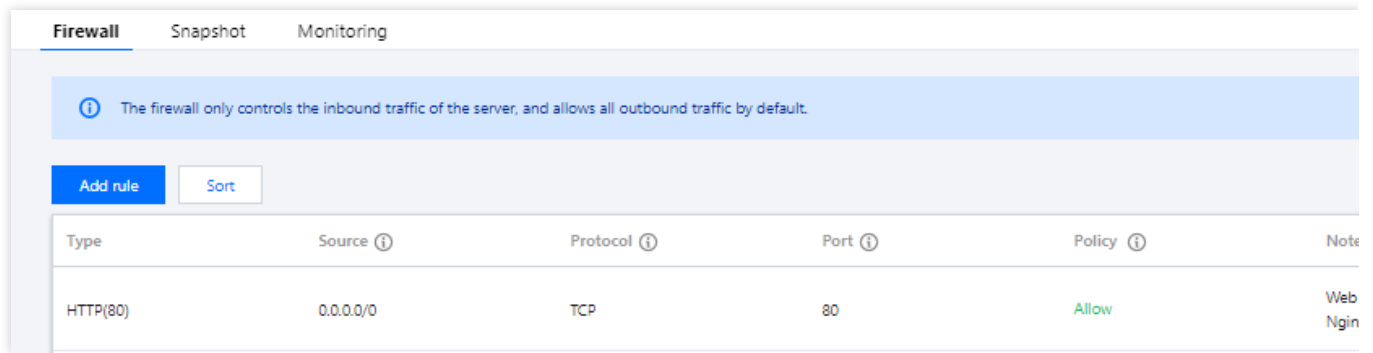
1. On the **Instances** page, select the target instance to enter its details page.

2. Select the **Firewall** tab, click **Add rule**, and open port 9090.

### Note:



The default port for Kubernetes Dashboard is 9090.



## Logging in to Kubernetes Dashboard

1. On the **Instances** page, select the target instance to enter its details page.
2. Select the **Pre-installed application** tab to enter the application details page, where you can view the configuration items of the application.

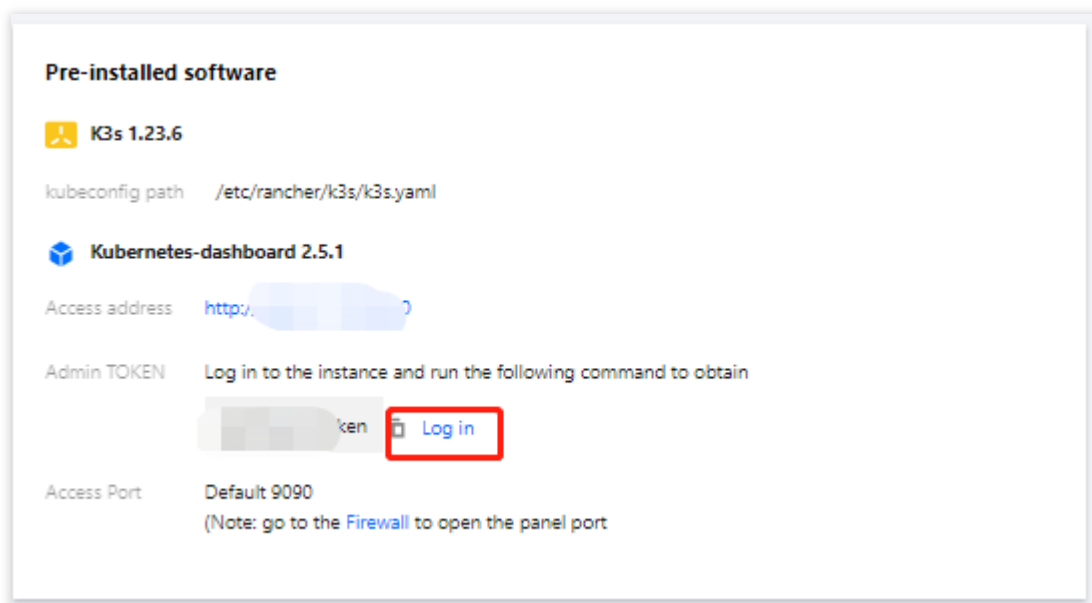
3.

In the **Pre-installed software** section

, click



to copy the admin token of Kubernetes Dashboard.



4. In the **Pre-installed software** section, click **Log in**.

6.

```
[lighthouse@VM-8-31-centos ~]$ dashboard-token
eyJhbGciOiJIUzU1NiIsImtpZCI6IjRBVj1MNGp6dThVLTFyND1Uby1HdHRYOEJhYUdTRjg3SWFkY3hRVjhaRmMifQ.eyJpc
iJrdWJlcm5ldGVzL3N1cnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWw1YWNjb3VudC9uYW1lc3BhY2UiOiJrd
Q5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWVjY291bnQvc2VjcmV0Lm5hbWUiOiJrdWJlcm5ldGVzLWRhc2hib2FyZ
2AkB-EL4fxmOZ1RLlQ4QigUvio2sEsvFSi2SR7rbBJqv44YzLgI-0oQxvDMT1A
[lighthouse@VM-8-31-centos ~]$
```

8. On the login page, enter the token obtained in [step 6](#) and click **Sign in**.

## Kubernetes Dashboard

☒ Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the [Authentication](#) section.

☐ Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the [Configure Access to Multiple Clusters](#) section.

Enter token \*

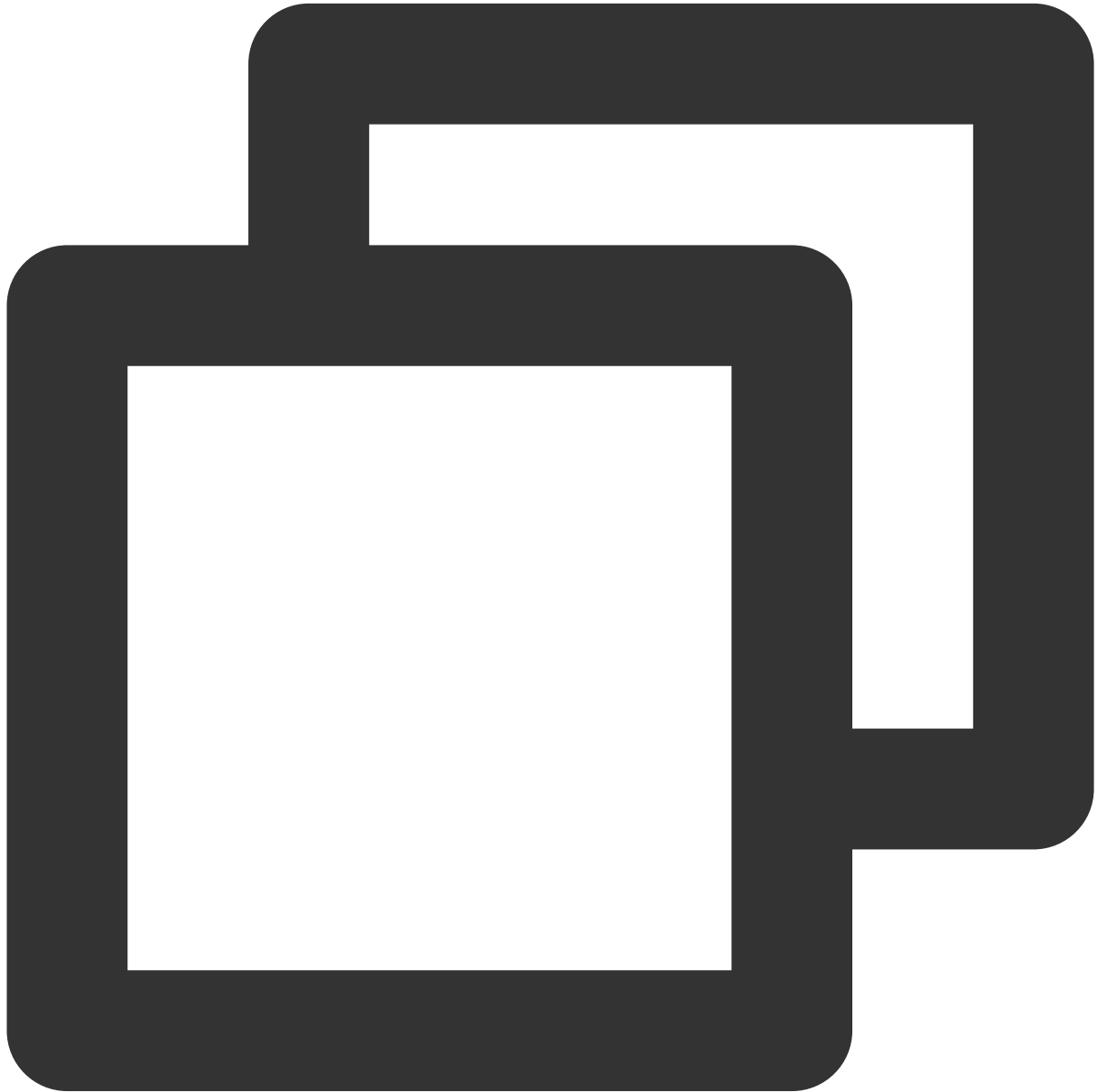
Sign in

## Relevant Operations

## Adding nodes

The instance created with the K3s application image works as the Master in the cluster. You can add nodes to the cluster as instructed below:

1. Enter the instance details page, select the **Firewall** tab, and open ports `TCP:6443` , `UDP:8472` , and `TCP:10250` (for node monitoring) as instructed in [Configuring Lighthouse firewall](#).
2. In the **Pre-installed software** section, click **Log in**.
3. In the pop-up window, run the following command to add the node IPs.



```
k3s-add-node {node-ip}
```

**Note:**

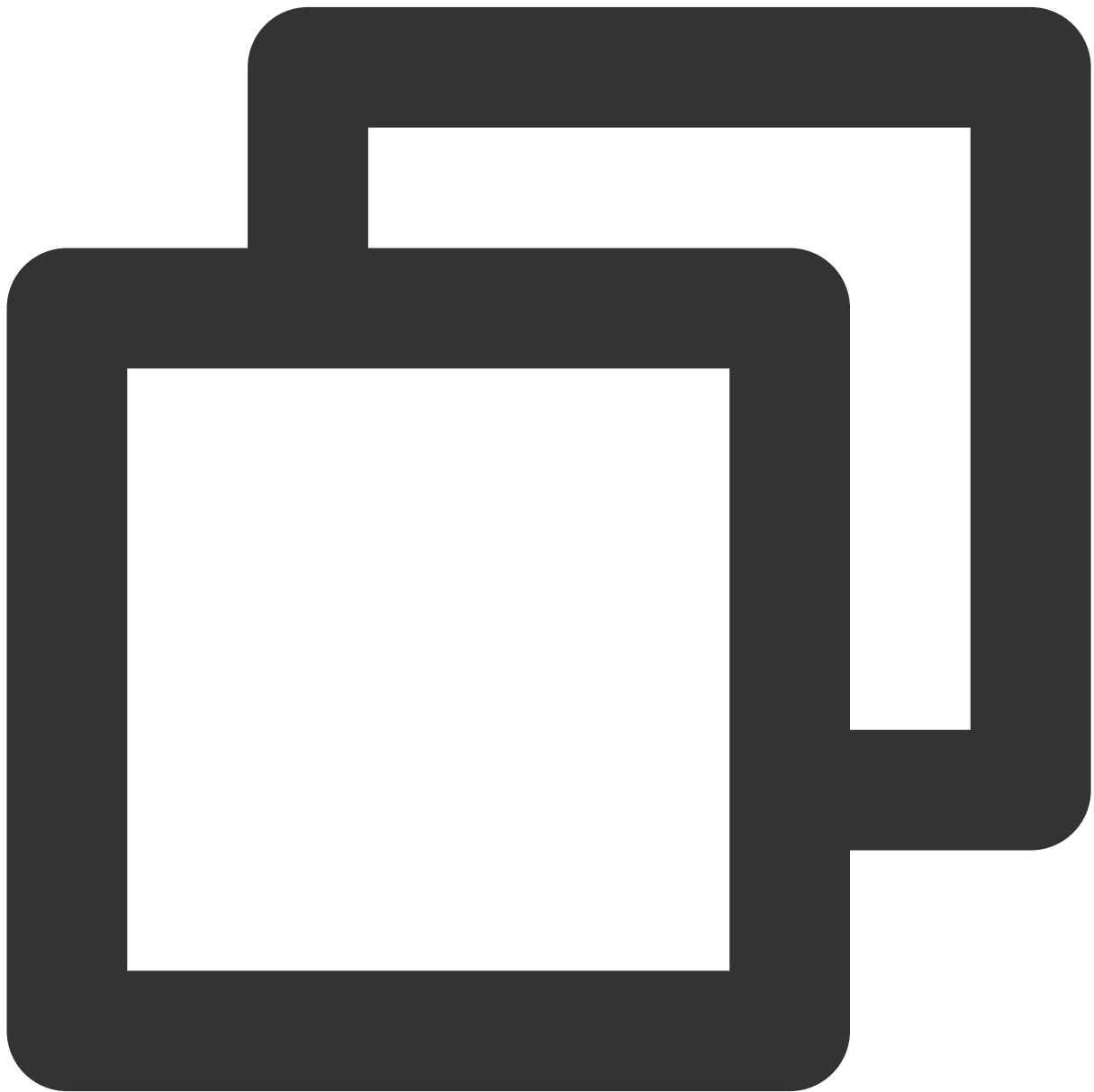
The Master is on CentOS 8.2. We recommend you [create Lighthouse instances](#) with the same OS in the same AZ as nodes in the cluster.

Nodes need to be connected to the Master over the private network.

Lighthouse instances in the same region under the same account are interconnected over the private network by default. For more information, see [Region and Interconnection](#).

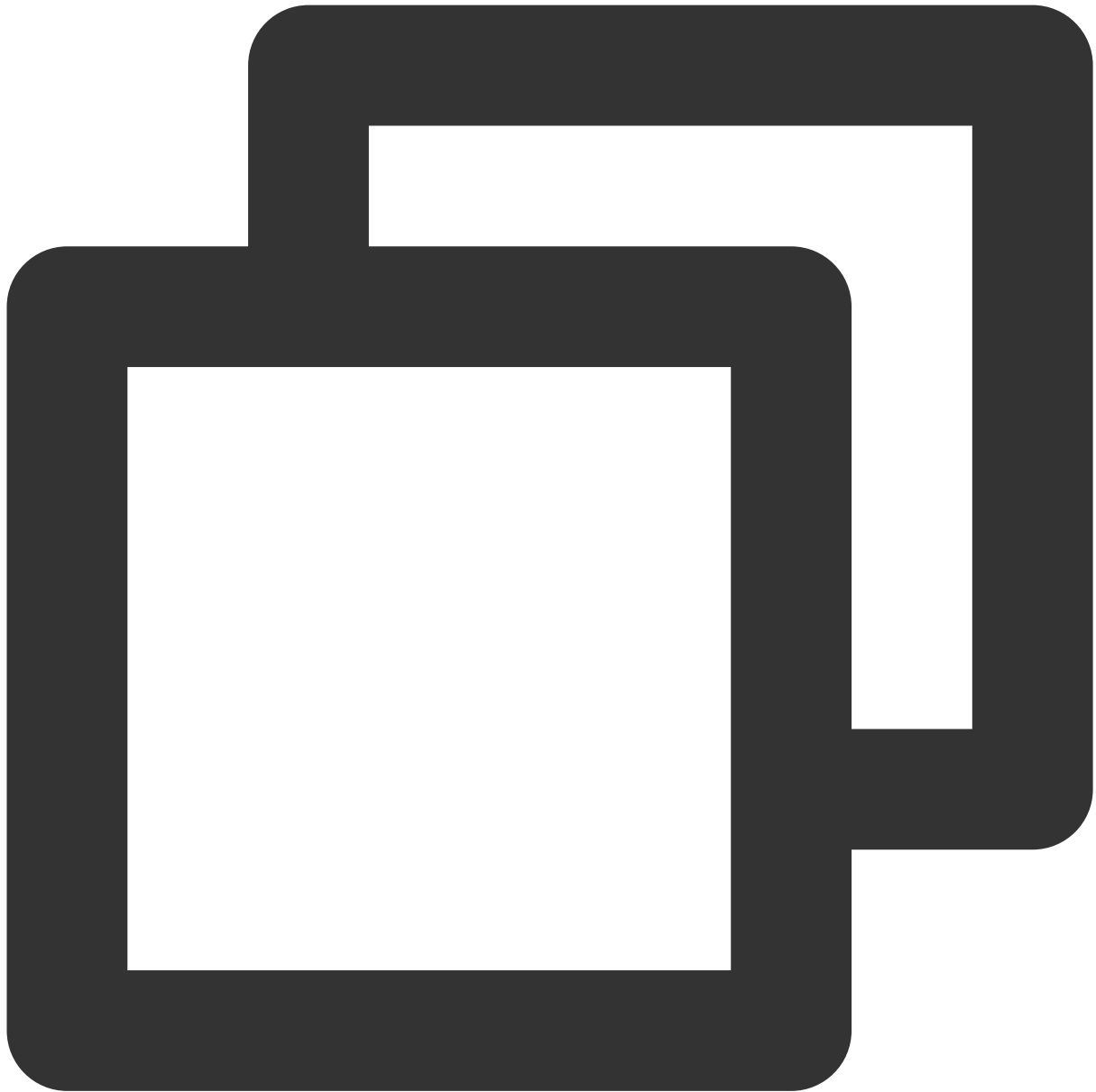
To allow the communication among nodes in the cluster, open port `TCP:6443` , `UDP:8472` , and `TCP:10250` in the firewall rule as instructed in [step1](#).

The sample command is as follows:



```
[lighthouse@VM-5-100-centos ~]$ k3s-add-node 10.0.5.158
```

The response is as follows:



```
Please ensure firewall rule(TCP:6443) of master node has been allowed!  
root@10.0.5.158's password:
```

4. Enter the `root` user password of the node to be added and press **Enter**. After the node is initiated, it will be added to the cluster.

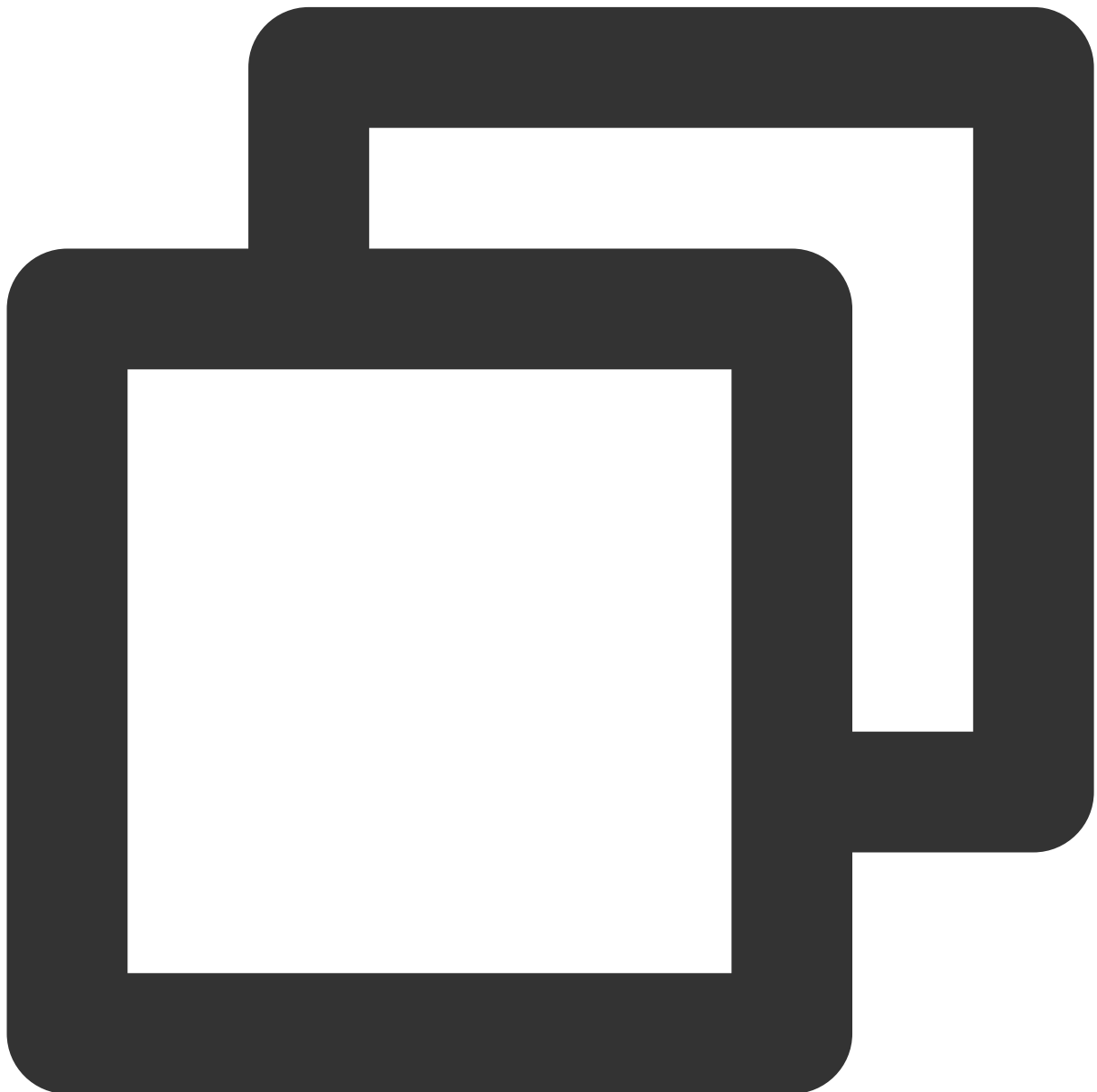
**Note:**

If you haven't set or forgotten the `root` user password, see [Resetting Password](#).

## Modifying NodePort

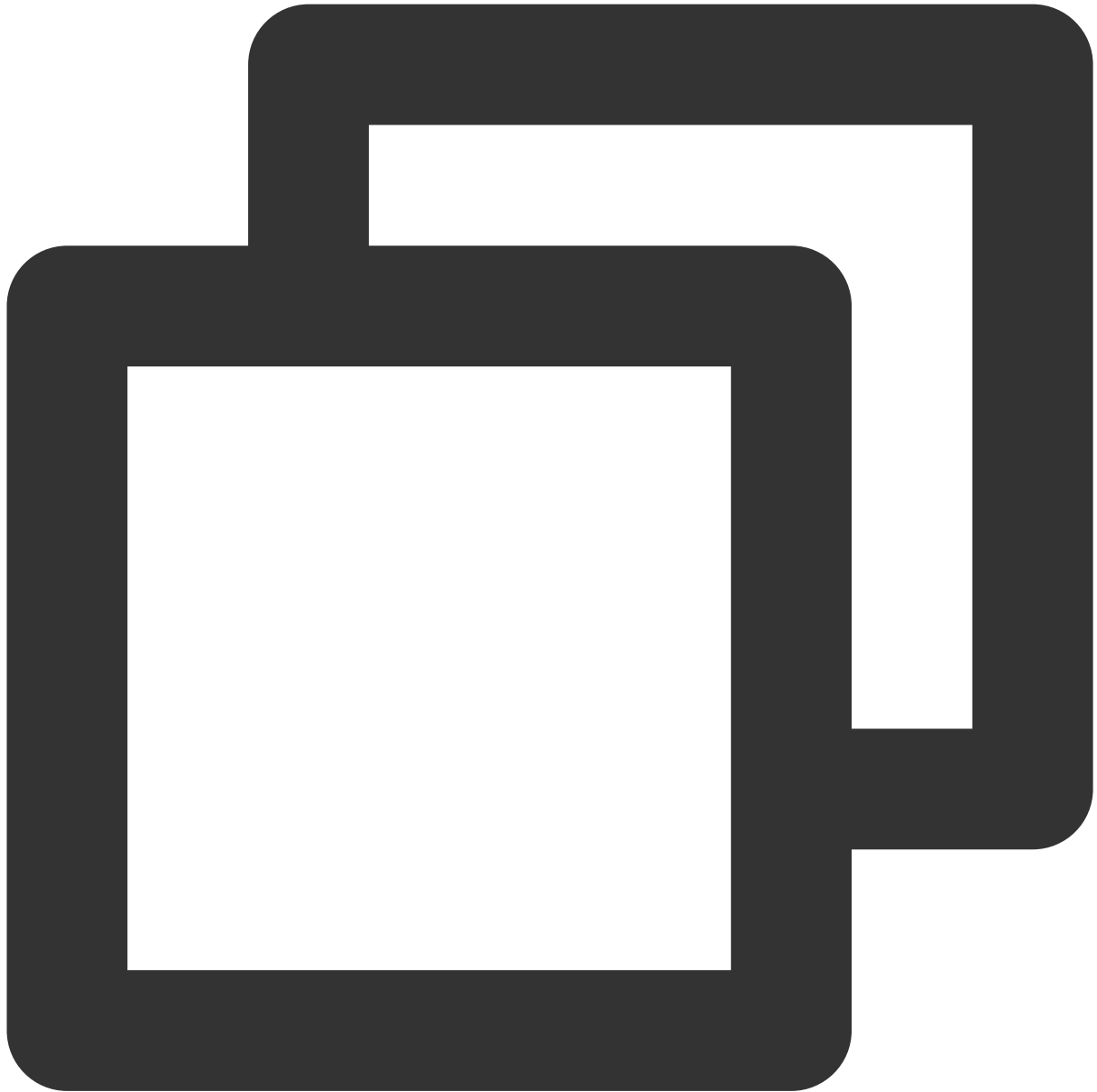
The default NodePort range is 30000-32767. In some cases, due to network policy restrictions, you may need to modify this range in the following steps:

1. Enter the instance details page, select the **Firewall** tab, and open the modified NodePort (e.g., `30000-42767`) as instructed in [Configuring Lighthouse firewall](#).
2. In **Remote login** on the instance details page, click **Log in**.
3. Run the following command to edit the `k3s.service` configuration file.



```
sudo vi /etc/systemd/system/k3s.service
```

4. Press **i** to enter the edit mode, find `ExecStart` , and add the `--service-node-port-range` parameter to specify the NodePort; for example:



```
ExecStart=/usr/local/bin/k3s server --write-kubeconfig-mode=644 --service-node-port-range=2443
```

The result should be as follows:

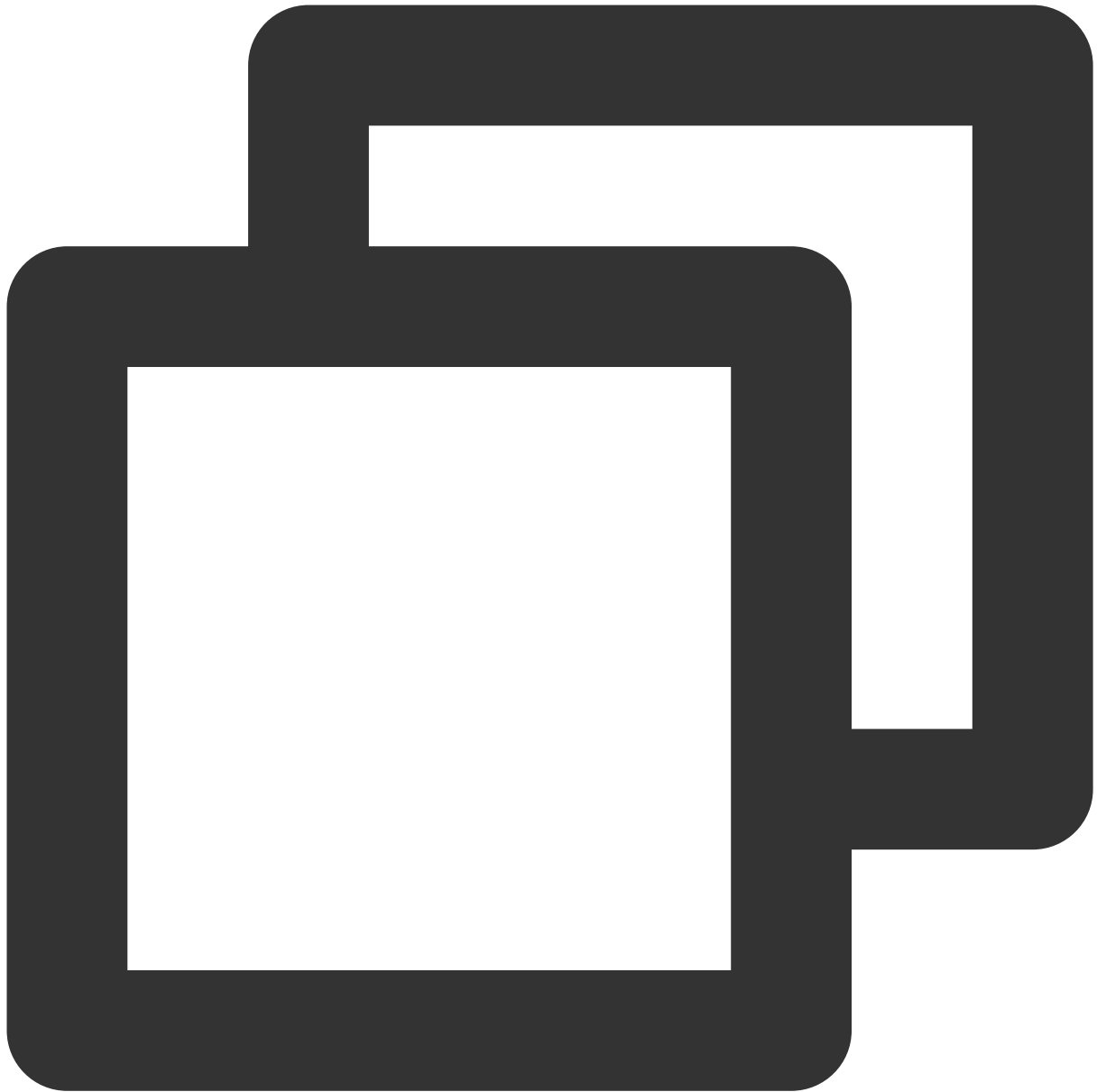
```
[Unit]
Description=Lightweight Kubernetes
Documentation=https://k3s.io
Wants=network-online.target
After=network-online.target

[Install]
WantedBy=multi-user.target

[Service]
Type=notify
EnvironmentFile=/etc/systemd/system/k3s.service.env
KillMode=process
Delegate=yes
# Having non-zero Limit*s causes performance problems due to accounting overhead
# in the kernel. We recommend using cgroups to do container-local accounting.
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity
TasksMax=infinity
TimeoutStartSec=0
Restart=always
RestartSec=5s
ExecStartPre=/sbin/modprobe br_netfilter
ExecStartPre=/sbin/modprobe overlay
ExecStart=/usr/local/bin/k3s server --write-kubeconfig-mode=644 --service-node-port-range=30000-
7
```

5. Press **Esc** and enter **:wq** to save the change and exit the edit mode.
6. Run the following command and enter the `root` user password to restart the K3s service and make the configuration take effect.





```
systemctl daemon-reload && systemctl restart k3s
```

## Enabling HTTPS access

You can install an SSL certificate and enable HTTPS access for your website as instructed in [Installing Certificate on NGINX Server](#).

# Cloud Storage System

## Building up a Cloud Storage System with Cloudfire Application template

Last updated : 2023-04-04 10:28:23

### Overview

Cloudfire is an open source network disk software that supports various storage methods such as server native storage and Tencent Cloud COS. Its features, such as offline download, drag and drop file upload, and online preview, allow you to quickly build a private or shared cloud storage system. The Cloudfire application image, with Nginx, Aria2, and MariaDB software pre-installed, is based on the CentOS 8.2 64-bit operating system.

This document introduces how to use the Cloudfire application image to build a Cloudfire cloud disk to implement file upload, share and offline download features.

### Directions

#### Creating a Lighthouse Instance Using Cloudfire Image

1. Log in to the [Lighthouse console](#). On the **Instances** page, click **Create**.
2. On the Lighthouse purchase page, purchase a Lighthouse instance with needed configurations selected. For image configuration, select **Application image > Cloudfire 3.3.1**. Configure other parameters as instructed in [Purchase Methods](#).

#### Note:

If the instance is in the Chinese mainland region, it is recommended that you choose a storage-optimized bundle for creating cloud disks. For details, see [Basic Bundle](#).

In this example, we use the application image Cloudfire 3.3.1. Note that the image may undergo version upgrades and updates. The actual version on the purchase page shall prevail.

#### Using Cloudfire

##### Logging in to Cloudfire

1. On the instance details page, select the **Pre-installed application** tab, and enter the application details page. You can view various configuration information of the Cloudfire application on this page.

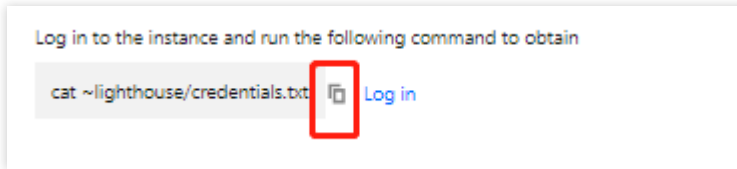
2.

In the **Pre-installed software** section

, click



to copy the command for getting the Cloudfire admin account and password.



3. In the **Pre-installed software** section, click **Log in**.

4. In the pop-up window, paste the command obtained in [step 2](#) and press **Enter**.

5.

Record the Cloudfire admin name and password

(that is, the "cloudfire\_username" and "cloudfire\_password" values) in the returned result.

```
[lighthouse@VM-12-13-centos ~]$ cat ~/lighthouse/credentials.txt
cloudfire_username = admin@cloudfire.org
cloudfire_password = e^s06+2U7KoE
mariadb_password = 1x0jZ5@3BB53
[lighthouse@VM-12-13-centos ~]$
```

6. Use a browser to access the **Homepage address** in **Pre-installed software**, enter the username and password obtained in [Step 5](#), and click **Log in**.

## Uploading files to Cloudfire

On the Cloudfire page, you can directly drag and drop local files to the specified area, or right-click to select files/directories for uploading.

## Sharing files

Cloudfire supports sharing the download link of a file or folder. It also enables you to set a password or expiration time for the download link as instructed below:

1. On the Cloudfire page, right-click the file to be shared, and select **Create share link** in the pop-up menu.

2. In the pop-up window, make settings as required, and click **Create share link**.

3. After obtaining the link, you can download the file simply with `Homepage address + share link`.

For example, if the homepage address is `http://xxx.xxx.xxx`, and the share link is `/s/jRfM`, you can download the file via `http://xxx.xxx.xxx/s/jRfM`.

## Offline download

Aria2 is pre-configured in the Cloudfire application image. Cloudfire supports offline download of Aria2 drivers.

Before using this feature, you need to understand Aria2 configuration and Cloudfire access settings via the following

steps.

1. On the Cloudfire page, click the user profile photo in the upper right corner, and click **Admin panel** in the pop-up menu.

2. Go to the **Cloudfire dashboard** page, and select **Parameter settings > Offline download** in the left sidebar.

You can refer to [Offline download](#) to modify relevant parameter settings as needed.

The steps to create an offline download are as follows:

1. On the Cloudfire page, select **Offline download** in the left sidebar.
2. Go to the **Offline download** page and select the **+** in the lower right corner of the page.
3. In the **Create offline download task** pop-up window, follow the instructions to create a download task.

### Admin panel

1. On the Cloudfire page, click the user profile photo in the upper right corner, and click **Admin panel** in the pop-up menu.

2. Enter the **Cloudfire dashboard** page, and configure parameters such as user group permissions and storage policies.

You can configure the permissions on capacity limit, download speed, creating share, sharing download link and webDAV according to the type of user group the user belongs to.

You can change the default storage policy. For a comparison of various storage policies, see [Comparison - Cloudfire](#).

# Building an E-commerce Platform

## Building an Individual E-commerce Site Using WooCommerce Application Template

Last updated : 2023-02-16 15:49:46

### Overview

WooCommerce is a popular tool for building independent e-commerce websites. It is open source, free of charge and easy to use. With powerful features, it allows you to quickly build an independent WordPress-based e-commerce website. This image comes pre-installed with WordPress (including WooCommerce plugin), Nginx, MariaDB, PHP software.

### Directions

#### Creating a Lighthouse Instance Using WooCommerce Application Image

1. Log in to the [Lighthouse console](#). On the **Instances** page, click **Create**.
2. On the Lighthouse purchase page, purchase a Lighthouse instance with needed configurations selected. For image configuration, select **Application image > WooCommerce 6.5.1**. Configure other parameters as instructed in [Purchase Methods](#).

##### Note:

To set up live streaming service using a created instance, you can use the WooCommerce application image to [reinstall system](#).

In this example, we use the application image WooCommerce 6.5.1. Note that the image may undergo version upgrades and updates. The actual version on the purchase page shall prevail.

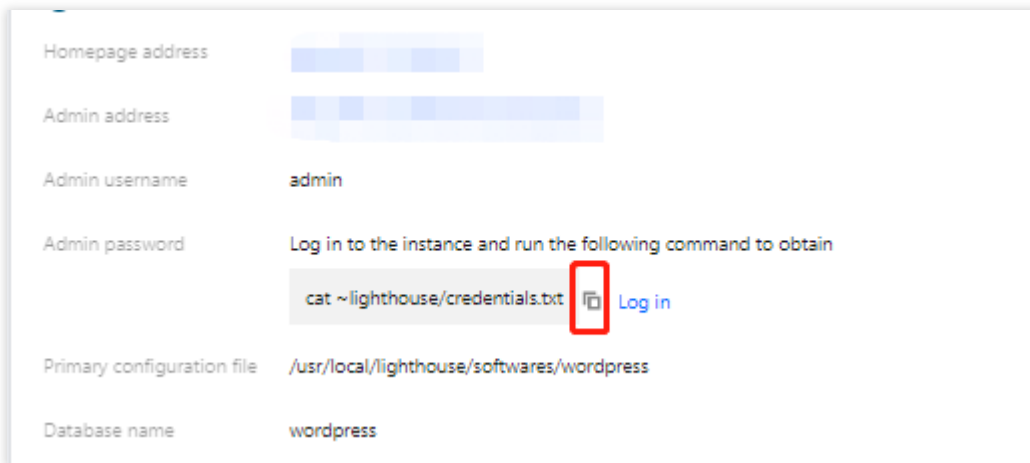
#### Logging in to the Website Admin Page


1. On the instance details page, select **Pre-installed application** tab, and enter the application details page.
- 2.

In the **Pre-installed software** section  
, click



to copy the command for getting WordPress admin account and password.



Homepage address	
Admin address	
Admin username	admin
Admin password	Log in to the instance and run the following command to obtain <code>cat ~lighthouse/credentials.txt</code>  <a href="#">Log in</a>
Primary configuration file	/usr/local/lighthouse/softwares/wordpress
Database name	wordpress

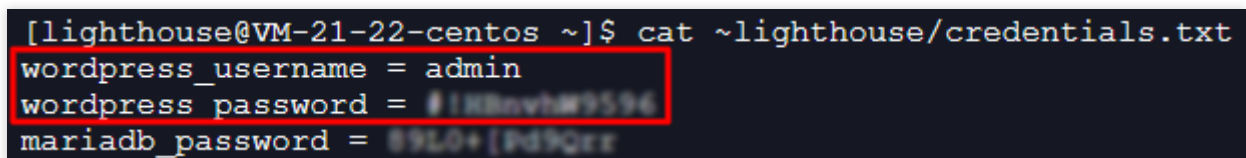
3. In the **Pre-installed software** section, click **Log in** beside the command or at the upper right corner.

4.

In the pop-up login window

, paste the command obtained in [Step 2](#) and press **Enter**.

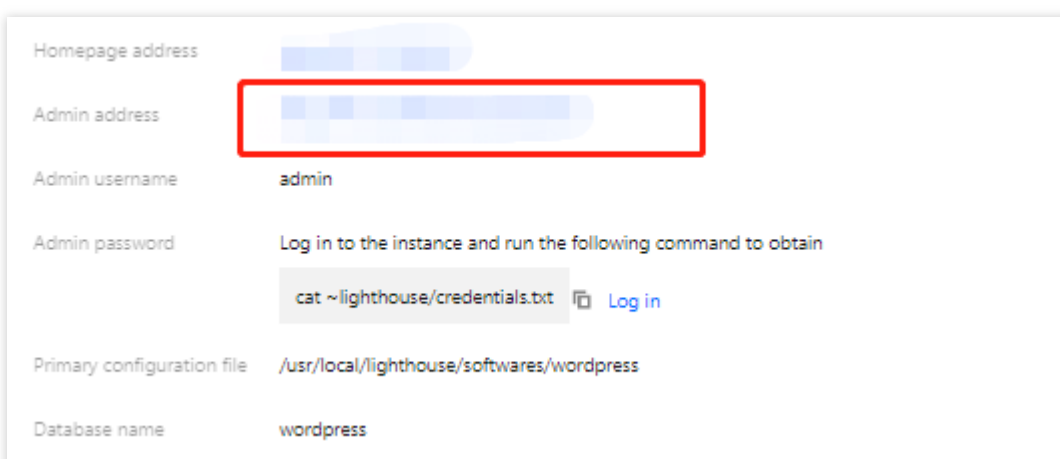
Then, you can obtain admin account (admin) and password (wordpress\_password).




```
[lighthouse@VM-21-22-centos ~]$ cat ~lighthouse/credentials.txt
wordpress_username = admin
wordpress_password = #1K8nvhW9596
mariadb_password = 89L0+ [Pd9Qrr
```

5. Record admin account and password, close the login window, and go back to the application details page.

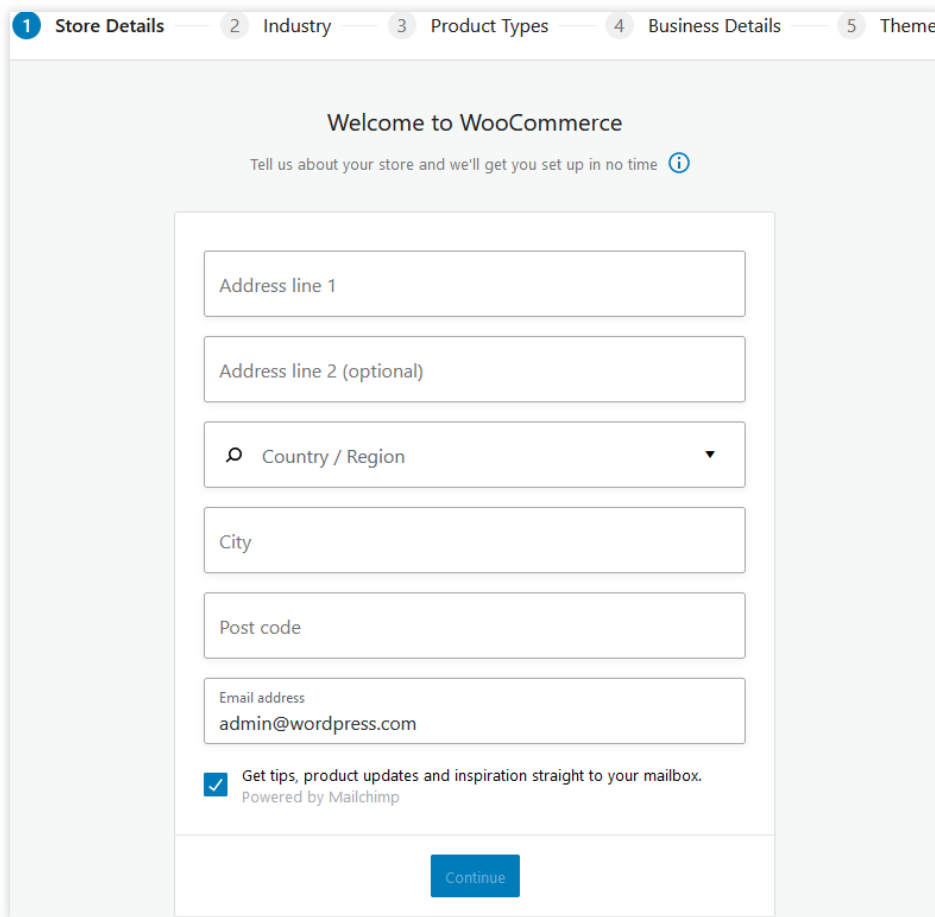
6. In the **Pre-installed software** section, click **Admin address**.



Homepage address	
Admin address	
Admin username	admin
Admin password	Log in to the instance and run the following command to obtain <code>cat ~lighthouse/credentials.txt</code>  <a href="#">Log in</a>
Primary configuration file	/usr/local/lighthouse/softwares/wordpress
Database name	wordpress

7. In the opened browser window, enter the account and password recorded in [Step 4](#), and click **Log in**.

8. Select **WooCommerce** > **Home** in the left sidebar. After entering the page as shown below, you can start configuring your own independent e-commerce website.



To get started with WooCommerce, see [WooCommerce](#).

## Related Operations

### Switching WordPress Admin Page Language

1. Log in to the admin page as instructed in [Logging in to the Website Admin Page](#) Step 1 - Step 7.
2. Select **Settings** in the left sidebar to enter the "General options" page.
3. Find **Site language** and select the target language.
4. Scroll to the bottom of the page and click **Save changes**.

### Using WordPress Theme

The free Kadence and Astra themes are installed by default, and other WordPress themes are also available. This section introduces you how to switch, add and update WordPress themes.

1. Log in to the admin page as instructed in [Logging in to the Website Admin Page](#) Step 1 - Step 7.
2. Select **Appearance > Theme** in the left sidebar.
3. On the **Theme** page, you can do the following:

Add a theme

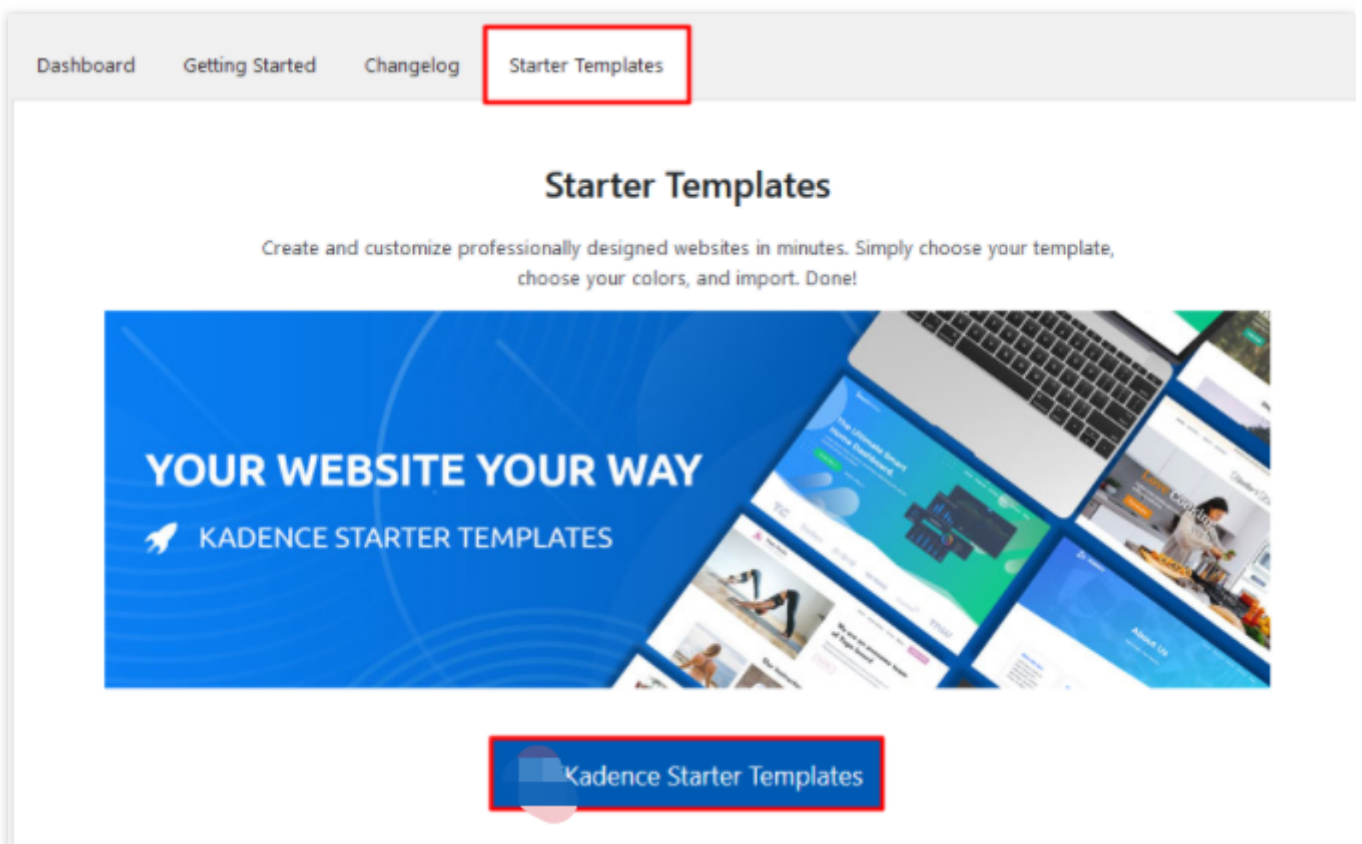
Switch a theme

On the page of adding a theme, click **Upload theme** to install a new theme.

On the **Theme** page, select the target theme, and click **Enable** to switch the theme.

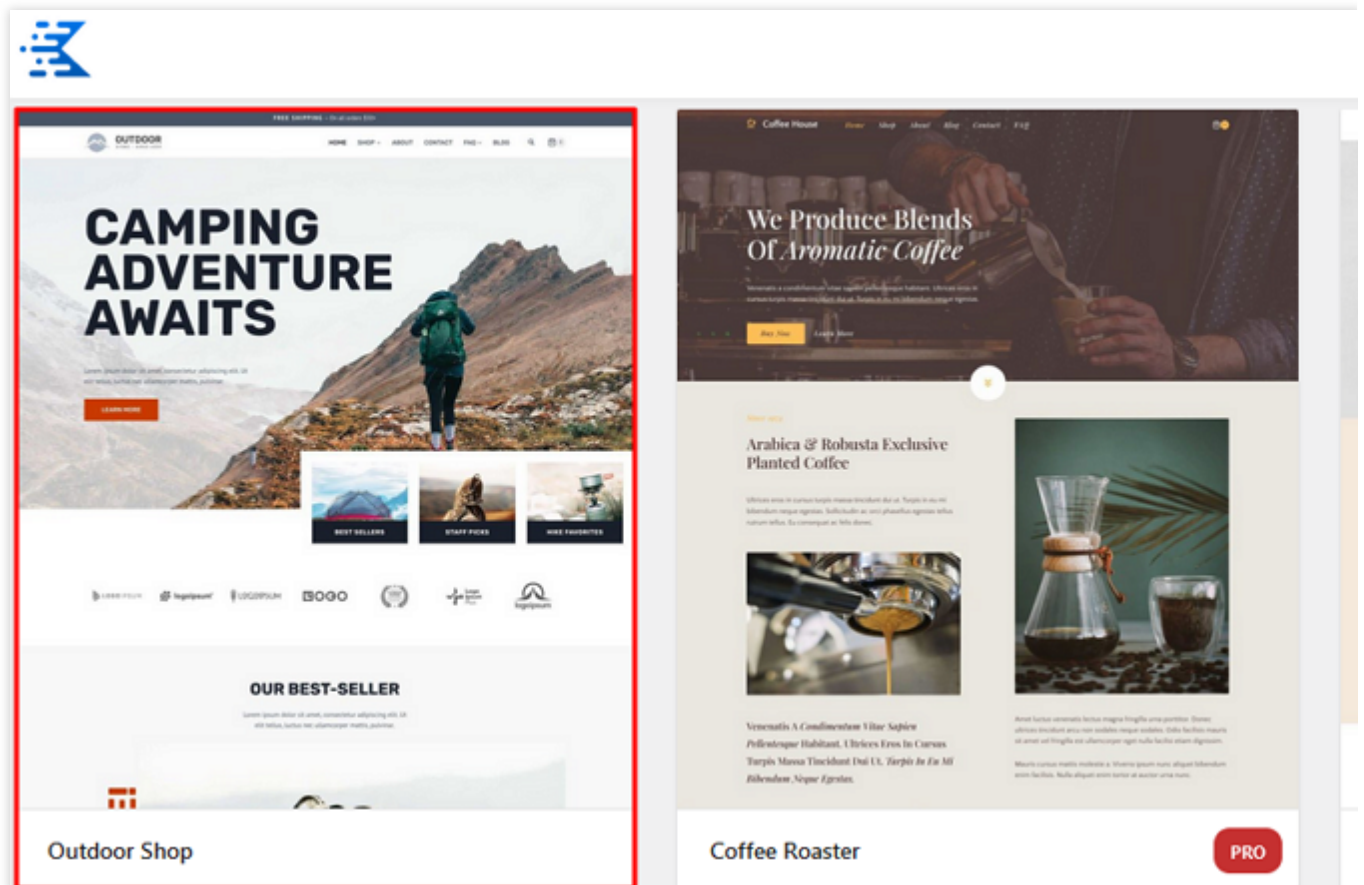
This document takes the Kadence theme installed by default as an example to introduce how to use the independent website template in the Kadence theme to make the online store more beautiful. The operation steps are as follows:

1. Log in to the admin page as instructed in [Logging in to the Website Admin Page](#) Step 1 - Step 7.
2. Select **Appearance > Theme** in the left sidebar, go to the **Theme** page, and click the Kadence theme.
3. On the details page of the Kadence theme, click **Kadence**.
4. Select the **Starter Templates** tab and click **Install Kadence Starter Templates**.



5. Select a template on the page. In this example, the **Outdoor Shop** template is selected. Click on the template.





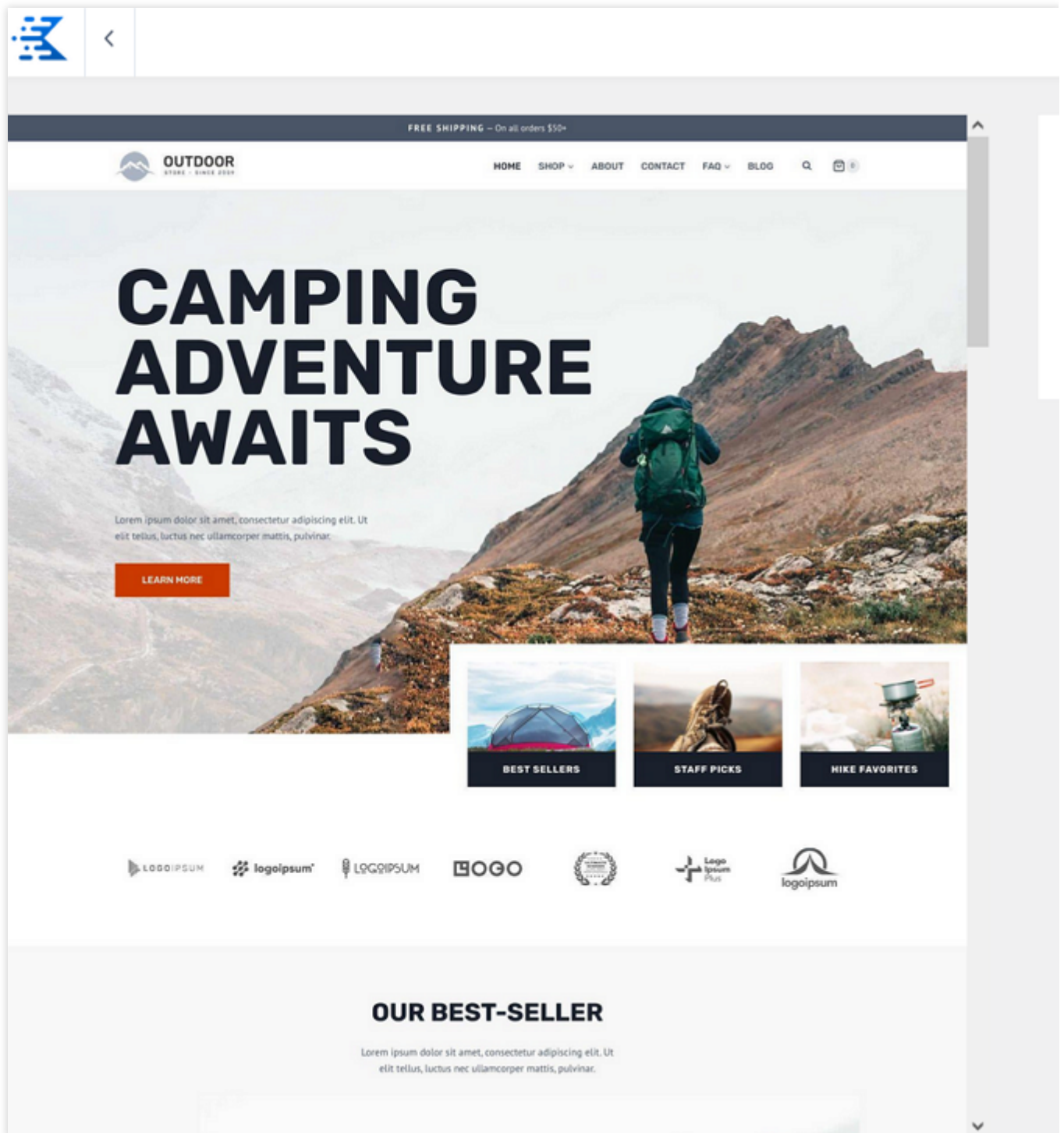
6. After editing the template as needed, select **Single Page** or **Full Site** in the **IMPORT OPTIONS** in the lower left corner of the page. In this example, **Full Site** is selected.

7. Check the **Must-knows** in the **Import Starter Template** pop-up window, and import.

**Note:**

This method will overwrite your site customizer settings, widgets, and menus. If you are testing different starter templates, it is recommended that you enable "Delete Previously Imported Posts and Images".

8. The following page indicates that you have successfully applied the template to your online store. You can click **Finished! View your site** to go to the store homepage.



## Enabling HTTPS access

You can install an SSL certificate and enable HTTPS access for your WooCommerce instance as instructed in [Installing Certificate on NGINX Server](#).

# Setting up Live Streaming Service with SRS Application Template

Last updated : 2024-03-20 14:40:12

## Overview

SRS is a simple and efficient real-time video server that supports RTMP, WebRTC, HLS, HTTP-FLV, and SRT/GB28181.

Lighthouse provides SRS application images, enabling you to easily set up live streaming service without complicated deployment operations.

## Related protocols

### HTTP-FLV

HTTP-FLV is another video format (a container format used to deliver streaming media data over the network) launched by Adobe. It is simple and lightweight, and does not require a lot of media header information. The entire FLV is composed of a header, a body and tags, implementing extremely fast loading speed.

FLV (also known as Flash Video) is a network video format known for its small size and extremely fast loading speed. The suffix of file formatted with FLV is `.flv`. HTTP-FLV encapsulates the streaming media data into FLV format, and then transmits it to the client through the HTTP protocol.

### HLS

HLS (also known as HTTP Live Streaming) is an HTTP-based adaptive bitrate streaming communications protocol developed by Apple Inc. The protocol is mainly widespread in audio and video services on PC and Apple's terminals. HLS breaks the overall stream into continuous small `ts` files on the server, and accesses the `ts` files in sequence via M3U8 index. The client only needs to continuously play the files obtained from the server in sequence, so as to realize the playback of audio and video.

HLS outperforms HTTP-FLV in the following aspects:

Supported natively on Apple's full range of products and also on Android system or PCs.

Supports HTTP/HTTPS transmission, effectively avoiding firewall blocks.

Higher performance.

However, HLS has the following disadvantages due to the transmission protocol.

Poor sync performance : The latency is often greater than 10s.

Requires high storage and cache performance due to sliced file transfer.

HTTP-FLV plays the best role in the scenario of interactive live streaming (such as live streaming e-commerce), while HLS is more applicable in some latency-insensitive scenarios such as general live streaming.

## Directions

### Creating a Lighthouse Instance Using SRS Application Image

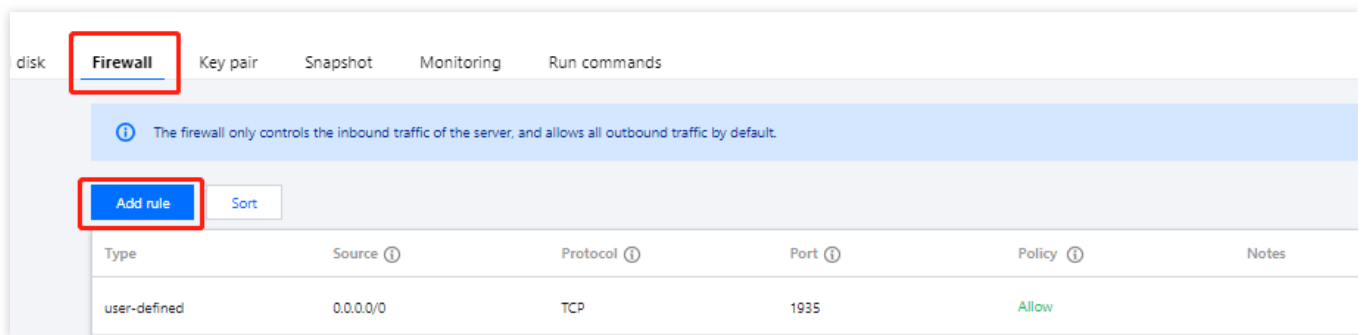
1. Log in to the [Lighthouse console](#). On the **Instances** page, click **Create**.
2. On the Lighthouse purchase page, purchase a Lighthouse instance with needed configurations selected. For image configuration, select **Application image** > **SRS 4.2**. Configure other parameters as instructed in [Purchase Methods](#).

#### Note:

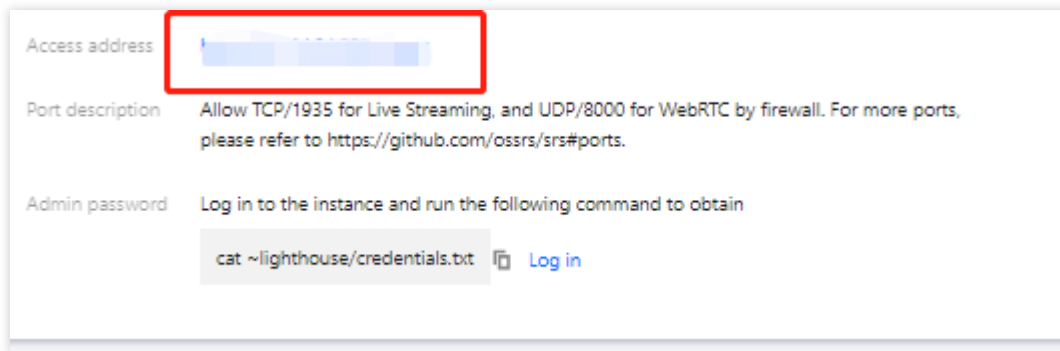
To set up live streaming service using a created instance, you can use the SRS application image to [reinstall system](#). In this example, we use SRS 4.2. Note that the image may undergo version upgrades and updates. The actual version on the purchase page shall prevail.

### Configuring an Instance

1. On the **Instances** page, select the target instance, and enter its details page.
2. Select the **Firewall** tab, click **Add rule**, and open port 1935.



3. Select the **Pre-installed application** tab, click **Access address** in **Pre-installed software** to enter the SRS admin page.

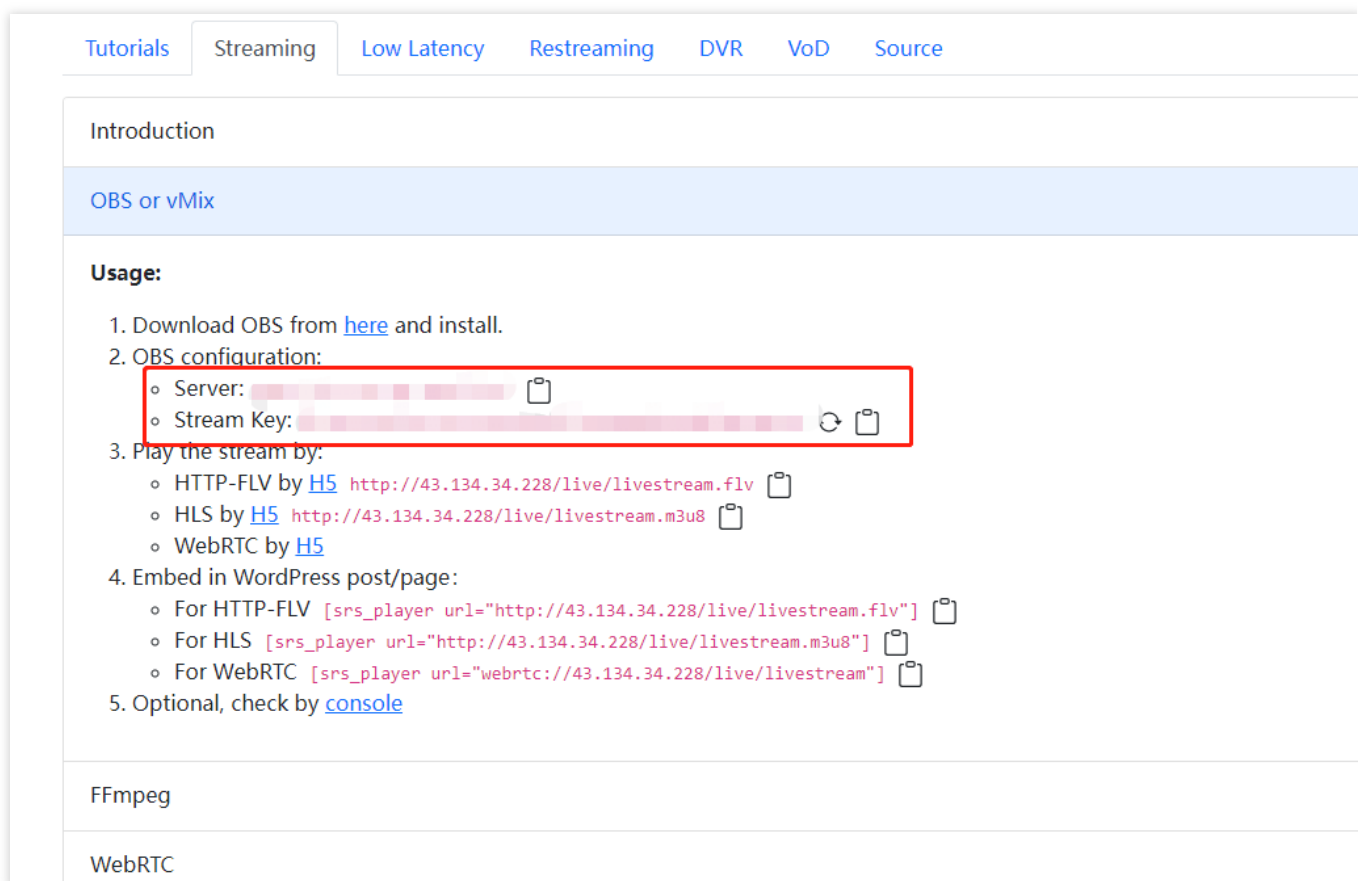


4. Set the admin password as instructed by prompts and keep it private when entering the SRS admin page for the first time.

5.

Log in to the SRS admin page

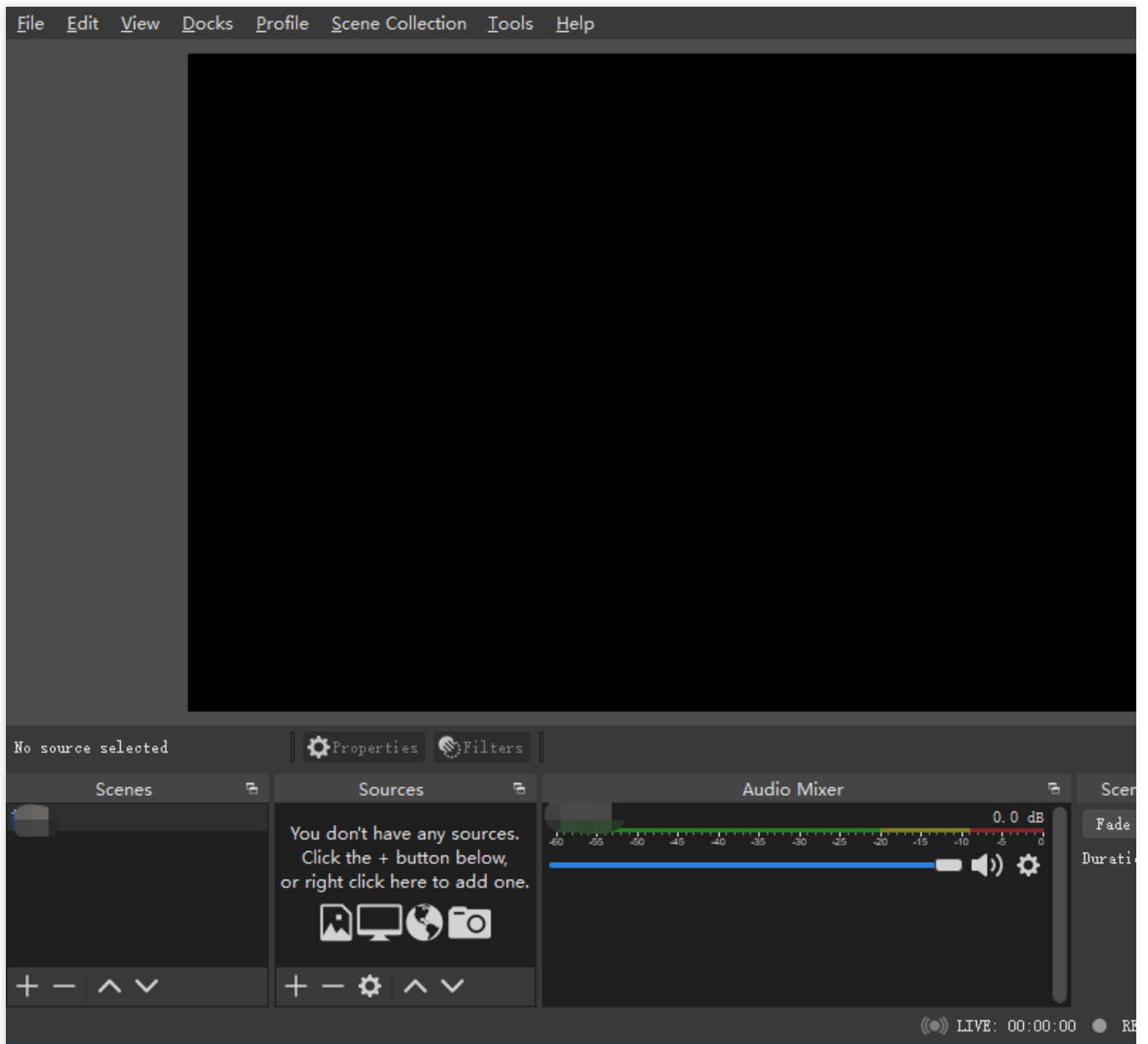
, and record the OBS push address and key.



## Installing and Configuring OBS Push Software

1. In this example, the push via OBS is adopted. Please download and install [OBS Studio](#).

2. Run OBS. The OBS interface displays the following items:



## 2.1 Scenes

## 2.2 Scene Collection

## 2.3 Sources

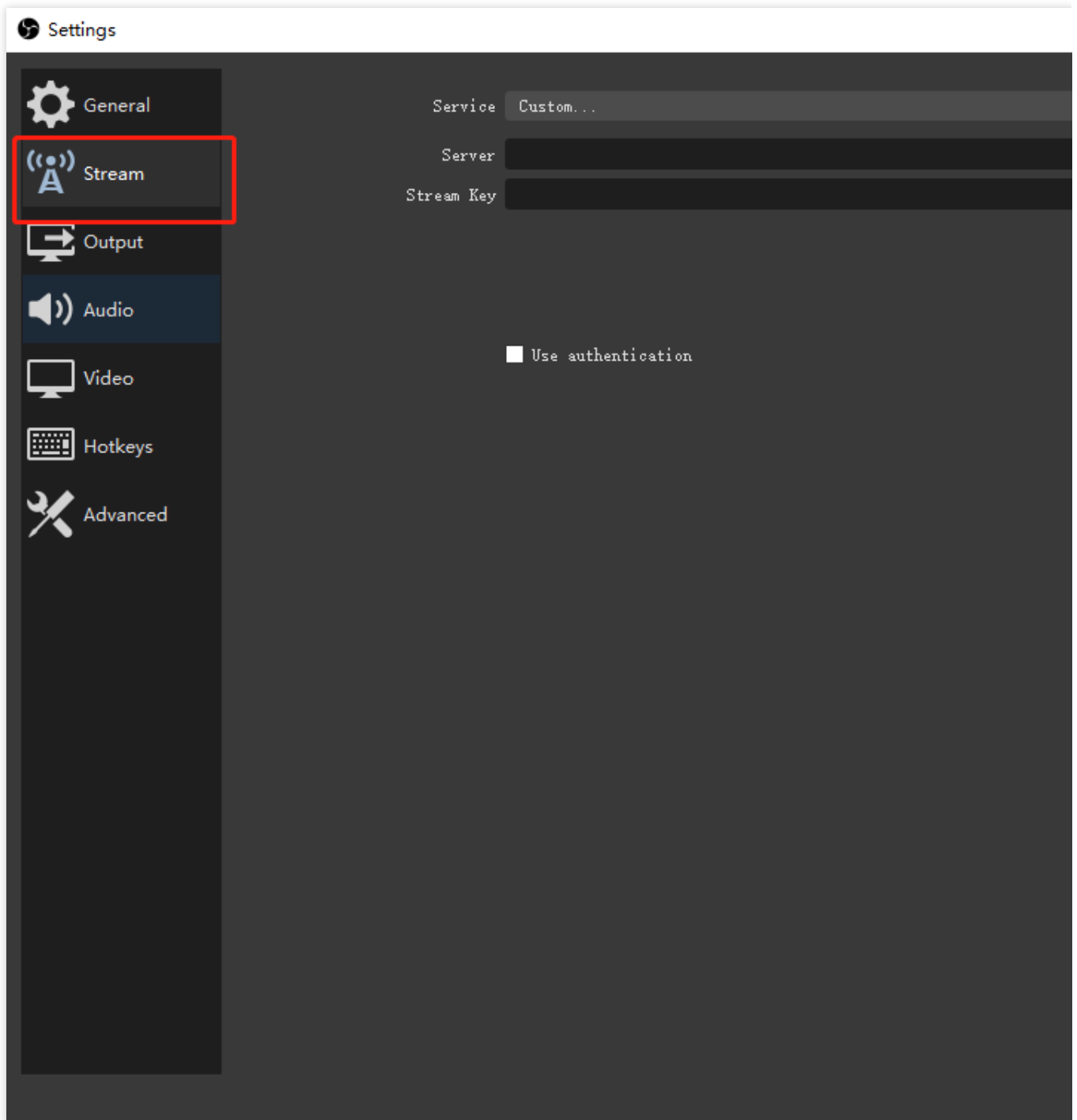
## 2.4 Audio Mixer

## 2.5 Controls

To learn more about OBS, see [OBS official website](#).

3. Select **File > Settings** on the top left corner of the interface.

4. Select **Push** on the left sidebar, and configure the following:



**Service:** Select **Custom** from the drop-down list.

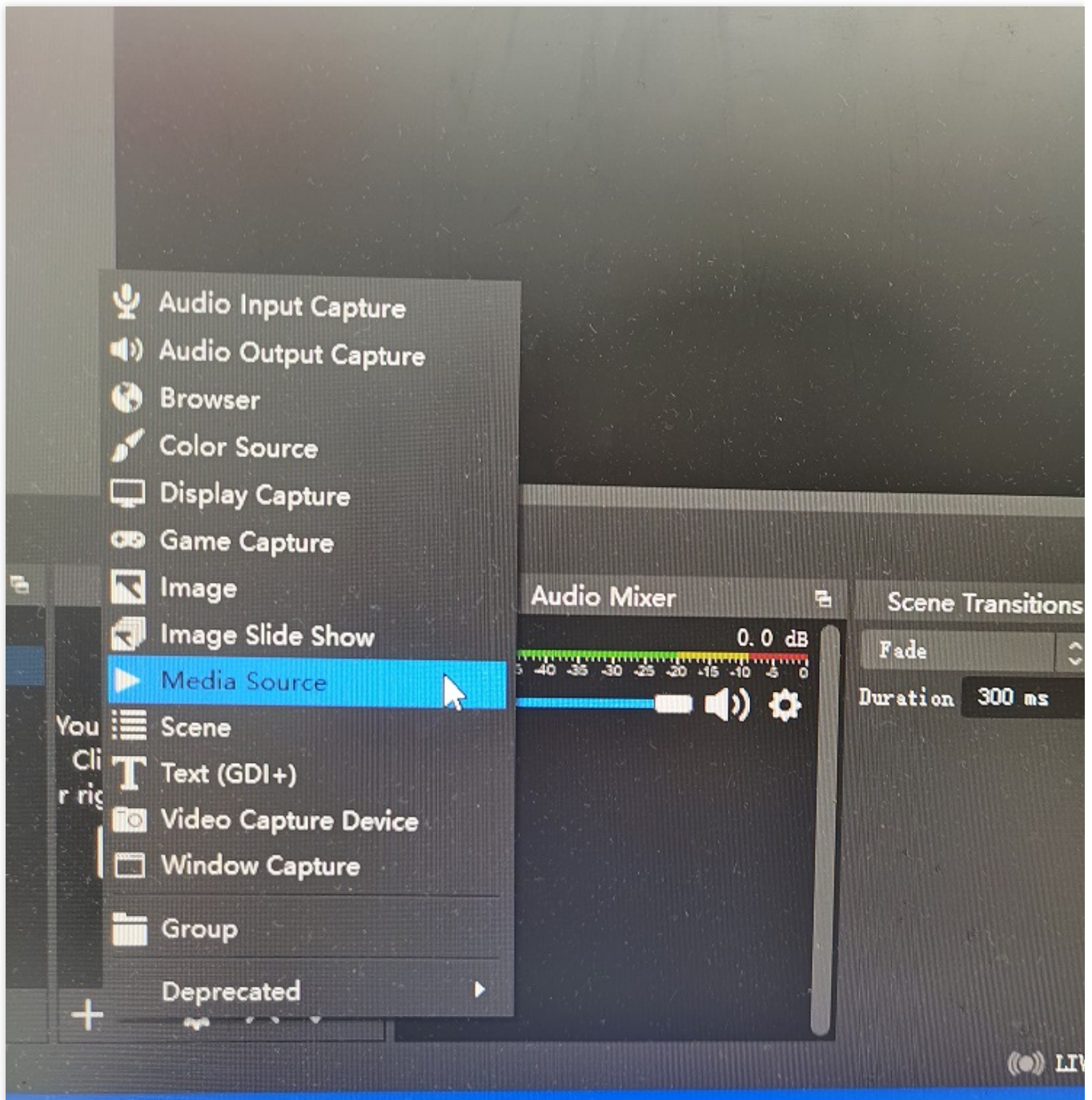
**Service:** Enter the URL obtained in [Step 5](#) for push via OBS.

**Stream Key:** Enter the stream key obtained in [Step 5](#).

5. Click **OK**.

6. Select **Media Source** in **Sources** settings on the main interface, and click **Start Streaming** under the **Controls** column.

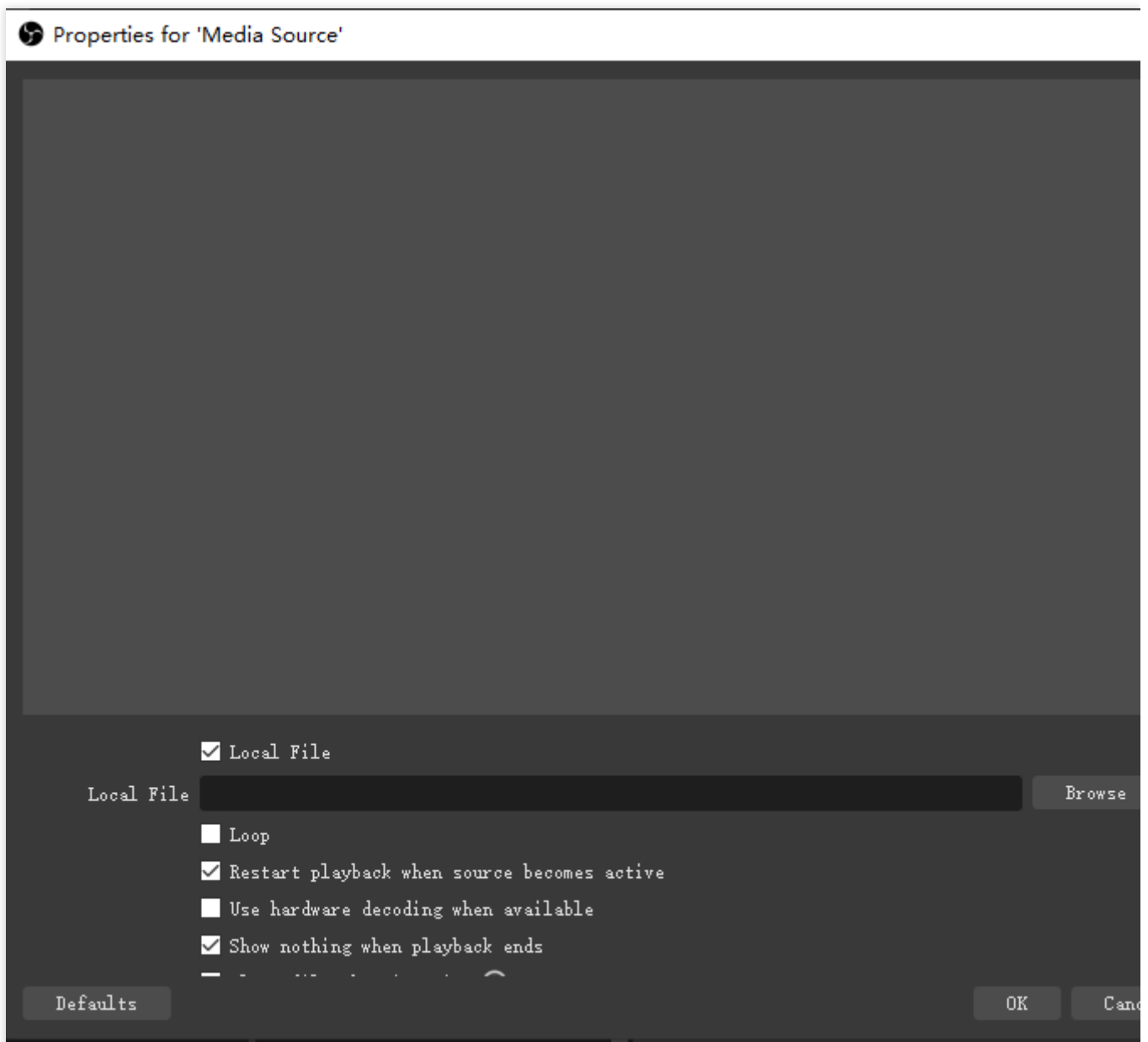




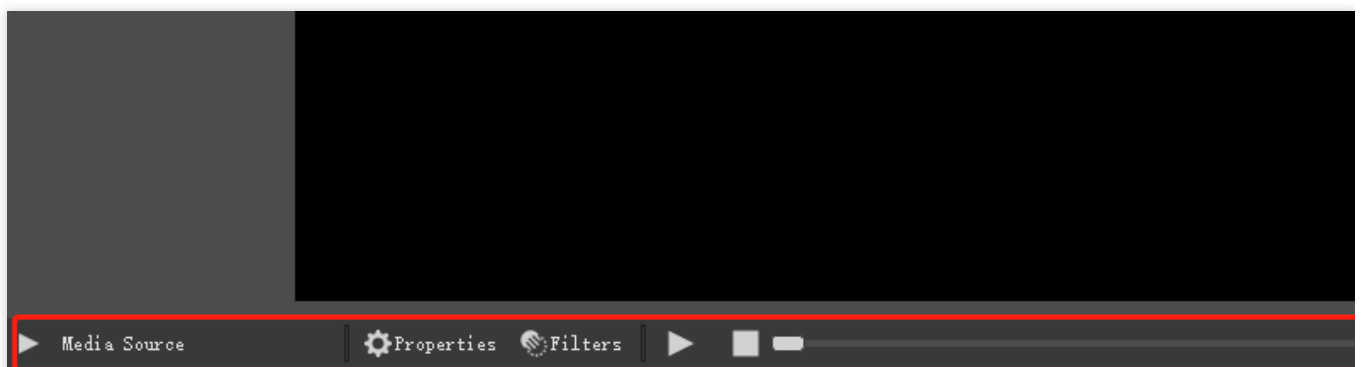
7. In the **Create or Select Source** pop-up window, create or select a source as required, and click **OK**.

8. In the **Properties for 'Media Source'** pop-up window, select the content to be pushed or (for live streaming). In this example, a local video resource is selected.





9. Click **OK** to upload and start streaming. You can right-click the screen in OBS to adjust the scene size and view orientation of the live streaming in real time.



## Viewing Live Streaming








1. Go to the **Instances** page in the console, and enter the details page of the target SRS instance.
2. Select the **Pre-installed application** tab, click **Access address** in **Pre-installed software** to enter the SRS admin page.
3. Click **HTTP-FLV by H5** or **HLS by H5** as shown in this example to view the live streaming scene.

[Tutorials](#) [Streaming](#) [Low Latency](#) [Restreaming](#) [DVR](#) [VoD](#) [Source](#)

Introduction

OBS or vMix

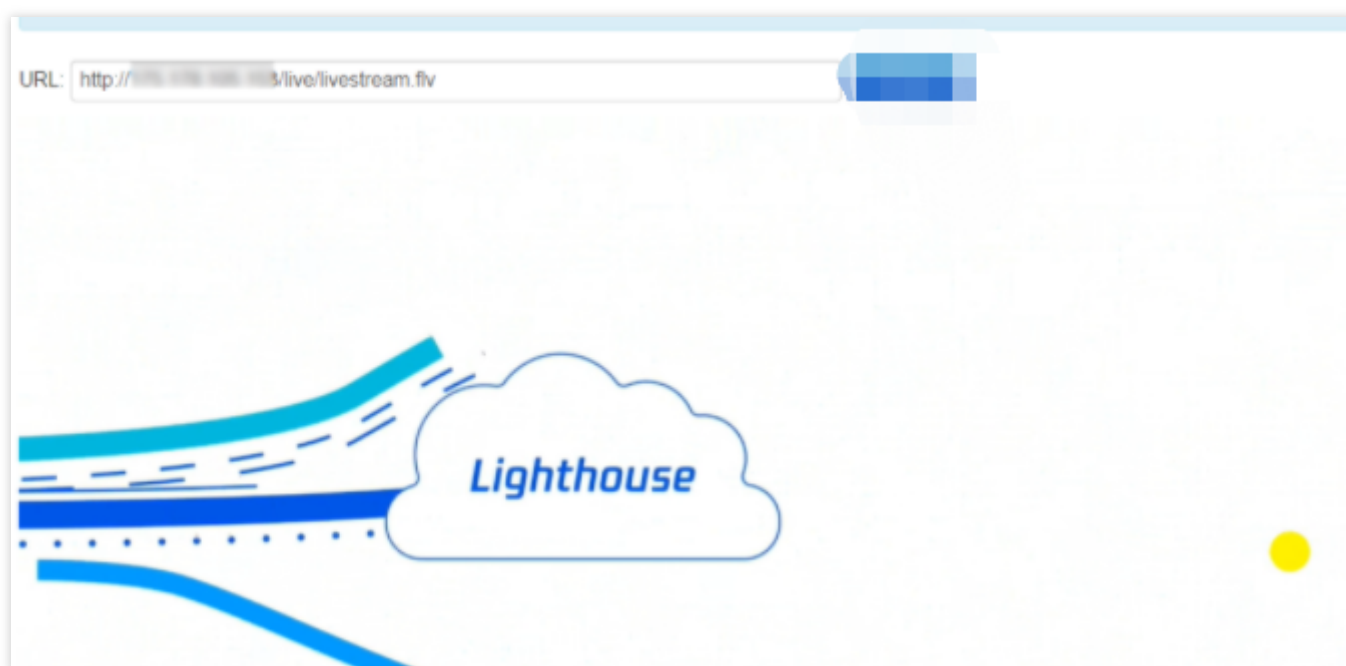
**Usage:**

1. Download OBS from [here](#) and install.
2. OBS configuration:
  - Server: 
  - Stream Key: 
3. Play the stream by:
  - HTTP-FLV by [H5](#) 
  - HLS by [H5](#) 
  - WebRTC by [H5](#)
4. Embed in WordPress post/page:
  - For HTTP-FLV 
  - For HLS 
  - For WebRTC 
5. Optional, check by [console](#)

FFmpeg

WebRTC

The live streaming scene is shown below:



# Setting Up an FTP Server

## Setting Up an FTP Server on Linux Lighthouse Instance

Last updated : 2024-03-20 14:39:38

### Overview

Very Secure FTP Daemon (vsftpd) is the default FTP server for most Linux distributions. This document describes how to use vsftpd to set up the FTP service on a Linux Lighthouse instance on CentOS 7.6 64-bit.

### Software

The following software programs are used to build the FTP service:

Linux: CentOS 7.6 system image.

vsftpd: vsftpd 3.0.2.

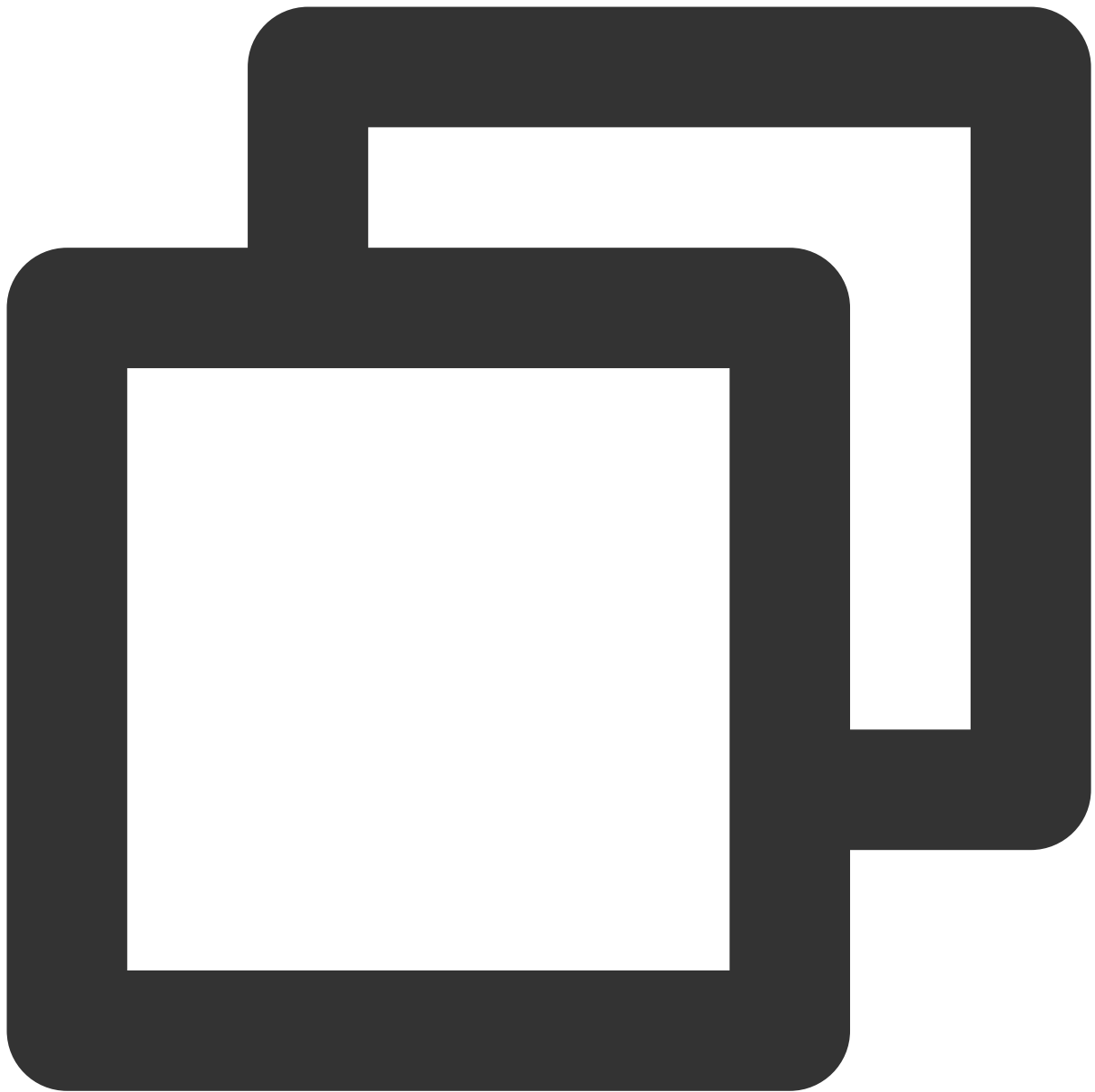
### Directions

#### Step 1. Log in to the Lighthouse instance

You can log in to the Linux instance via [WebShell](#) or [other methods](#).

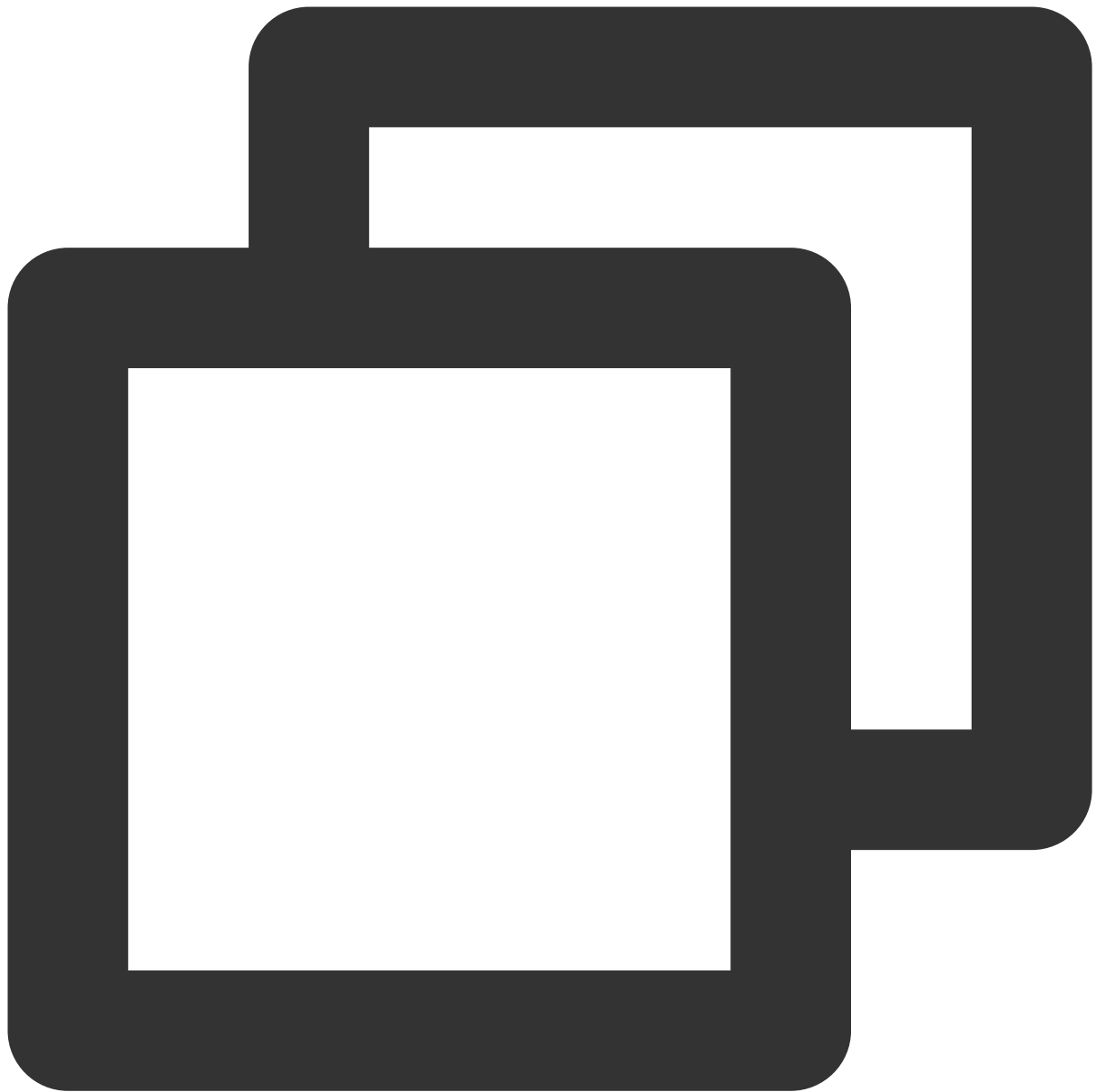
#### Step 2. Install vsftpd

1. Run the following command to install vsftpd.



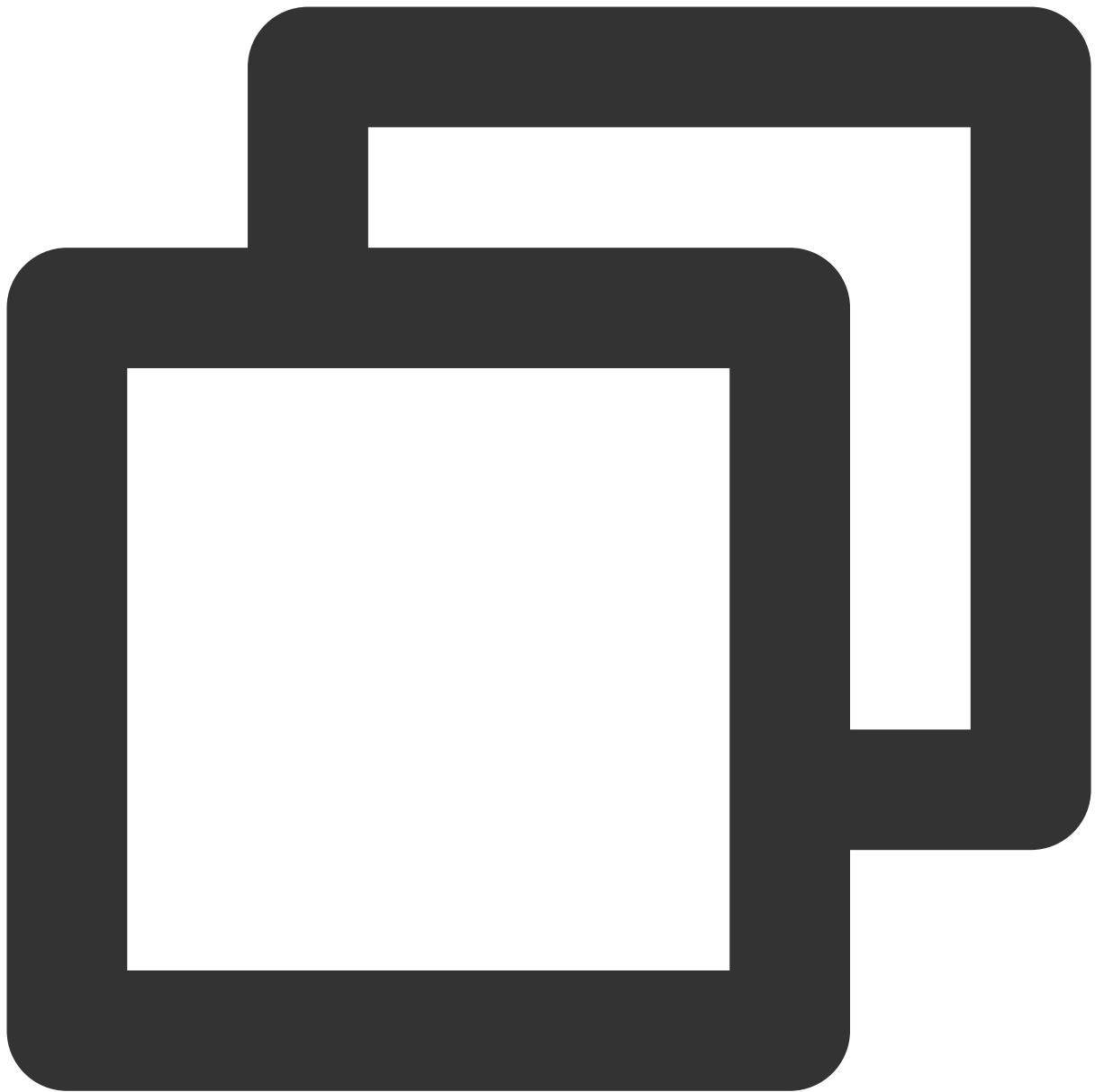
```
sudo yum install -y vsftpd
```

2. Run the following command to automatically start vsftpd upon system startup.



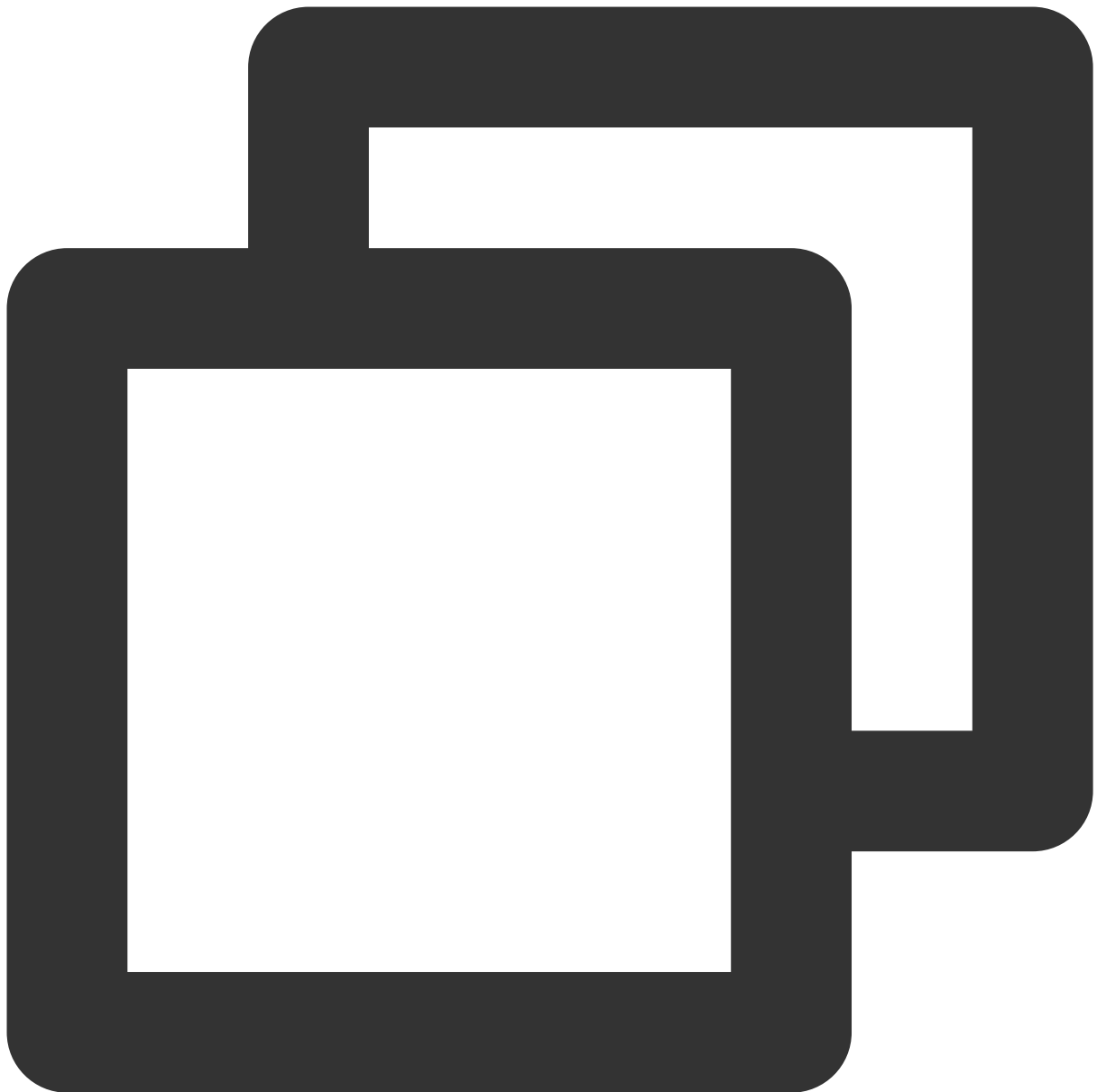
```
sudo systemctl enable vsftpd
```

3. Run the following command to start the FTP service.



```
sudo systemctl start vsftpd
```

4. Run the following command to check that the service has been started.



```
sudo netstat -antup | grep ftp
```

If the following information appears, the FTP service has been started.

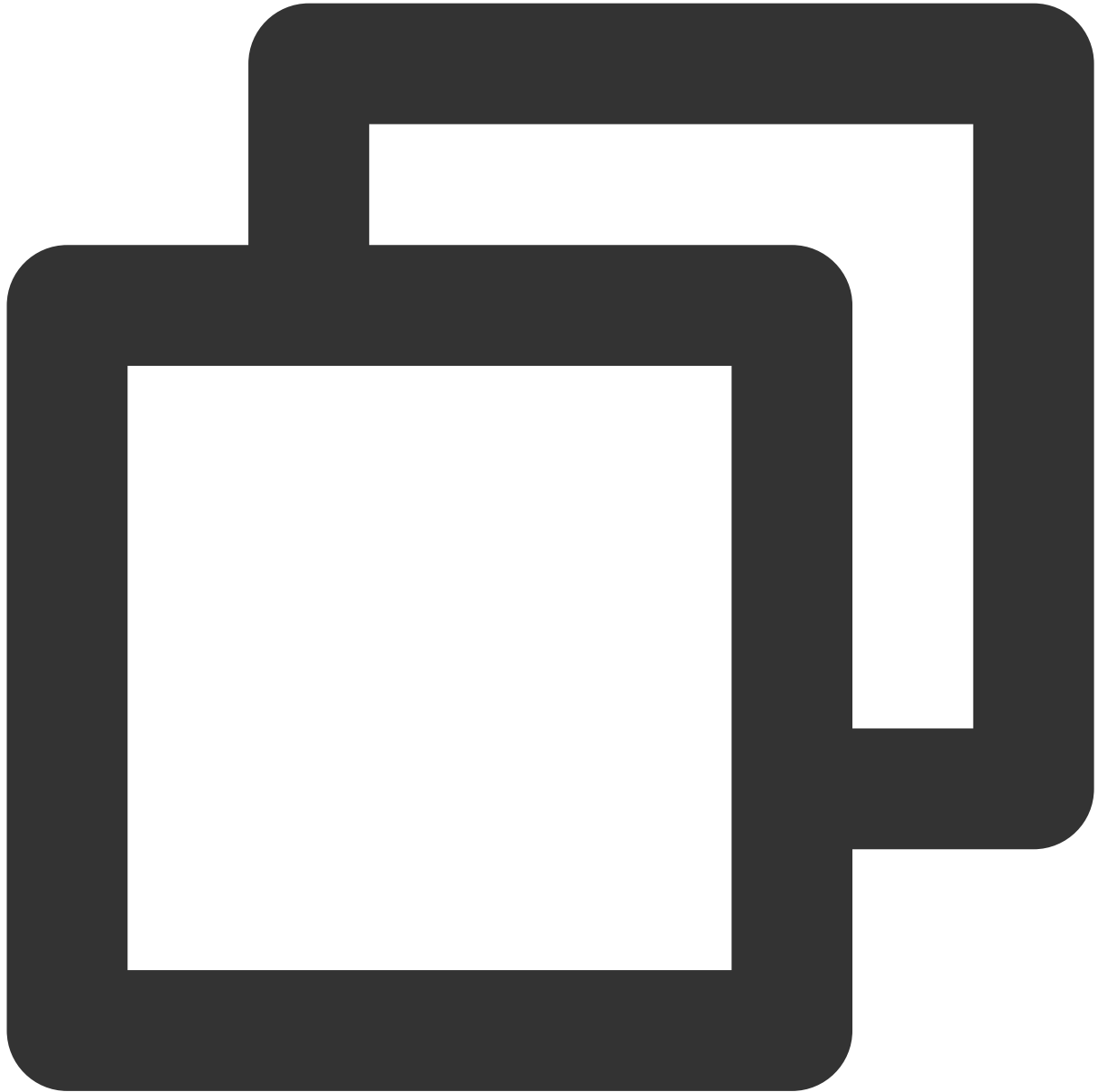
```
[lighthouse@VM-8-48-centos ~]$ sudo netstat -antup | grep ftp
tcp6      0      0 :::21          :::*            LISTEN          3960/vsftpd
```

By default, the anonymous access mode is enabled in vsftpd. You can log in to the FTP server without entering a username or password. However, you do not have permissions to modify or upload files in this login mode.



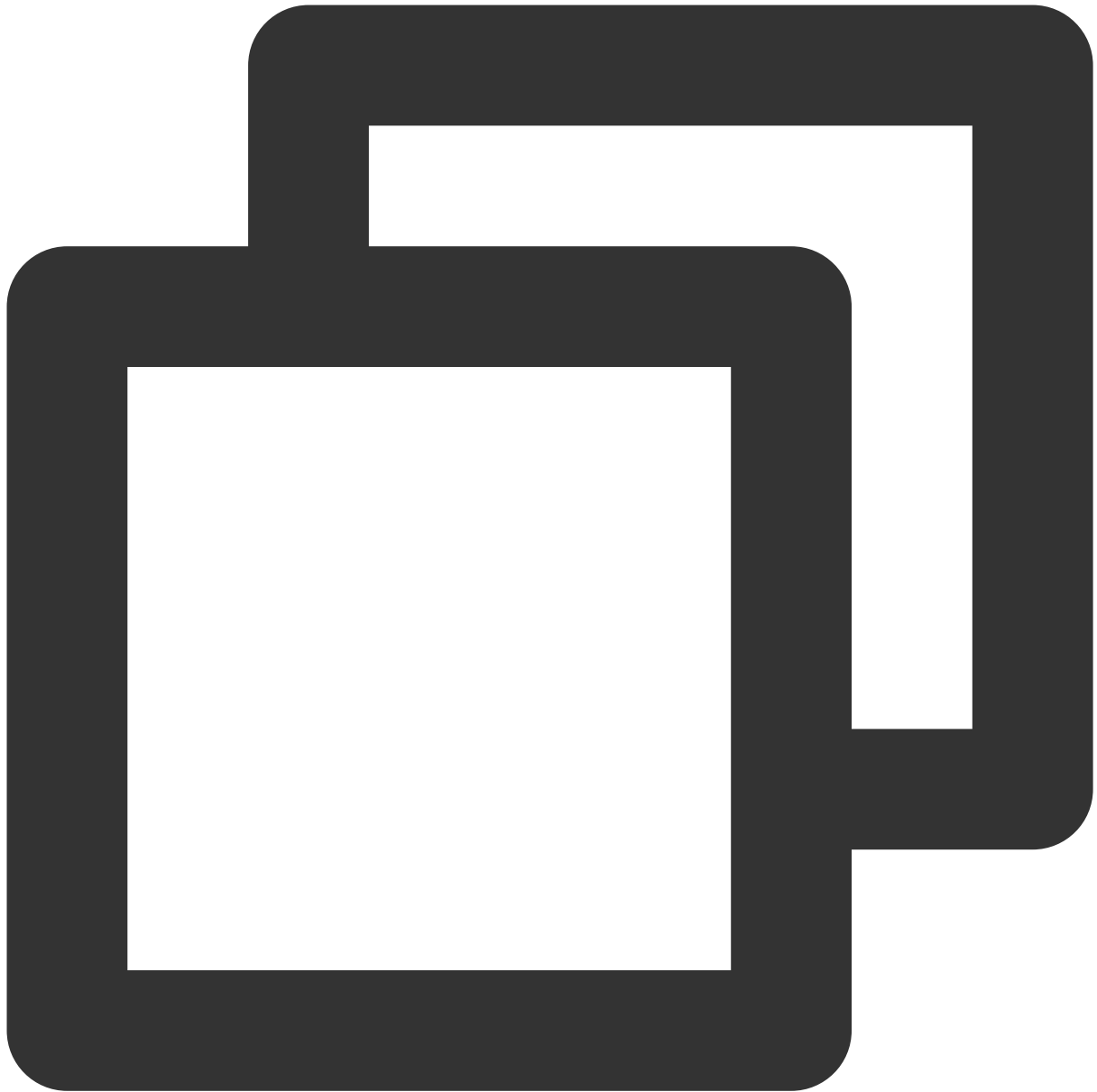
### Step 3. Configure vsftpd

1. Run the following command to create a user (such as `ftpuser`) for the FTP service:



```
sudo useradd ftpuser
```

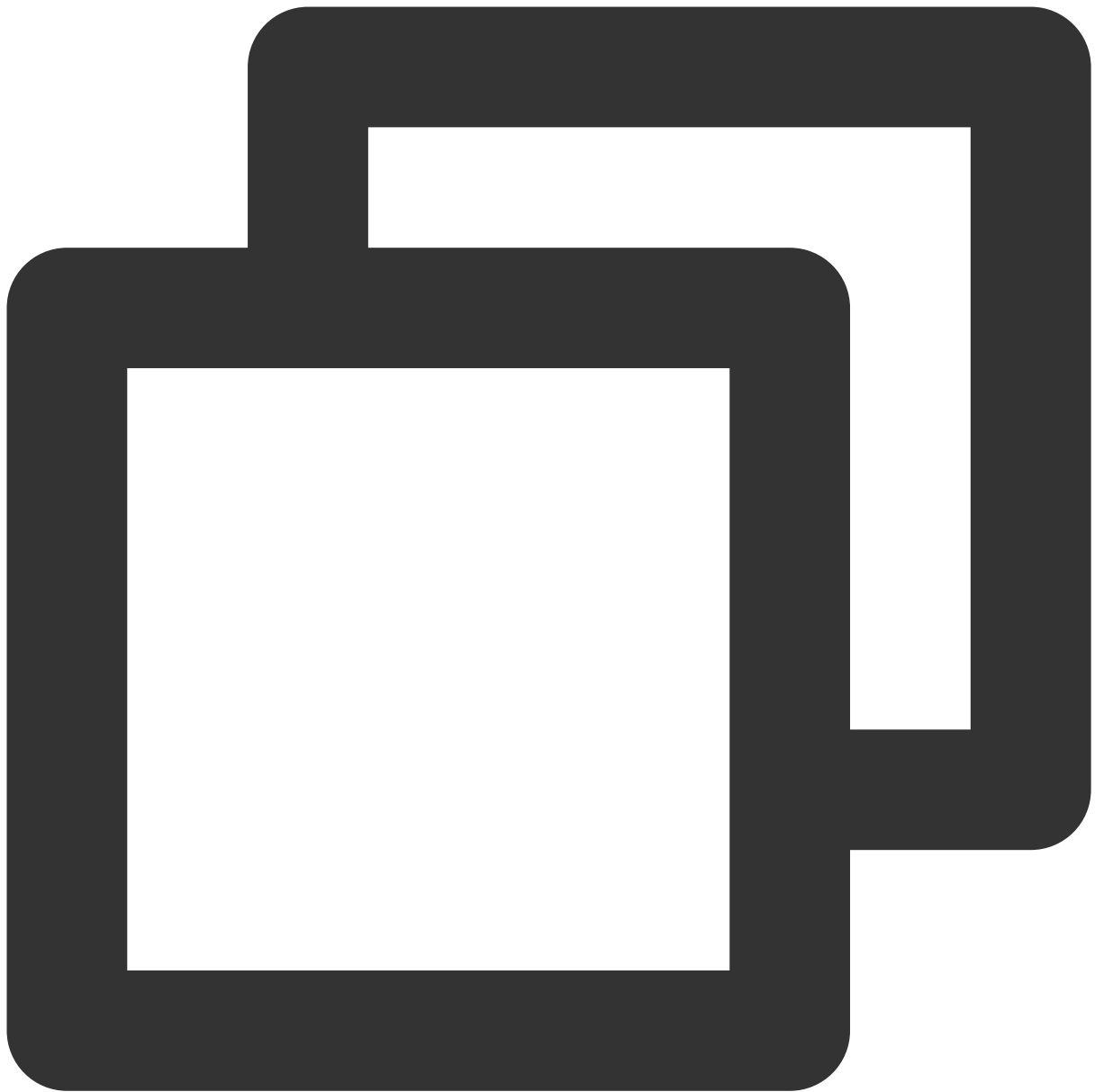
2. Run the following command to set the password for `ftpuser`.



```
sudo passwd ftpuser
```

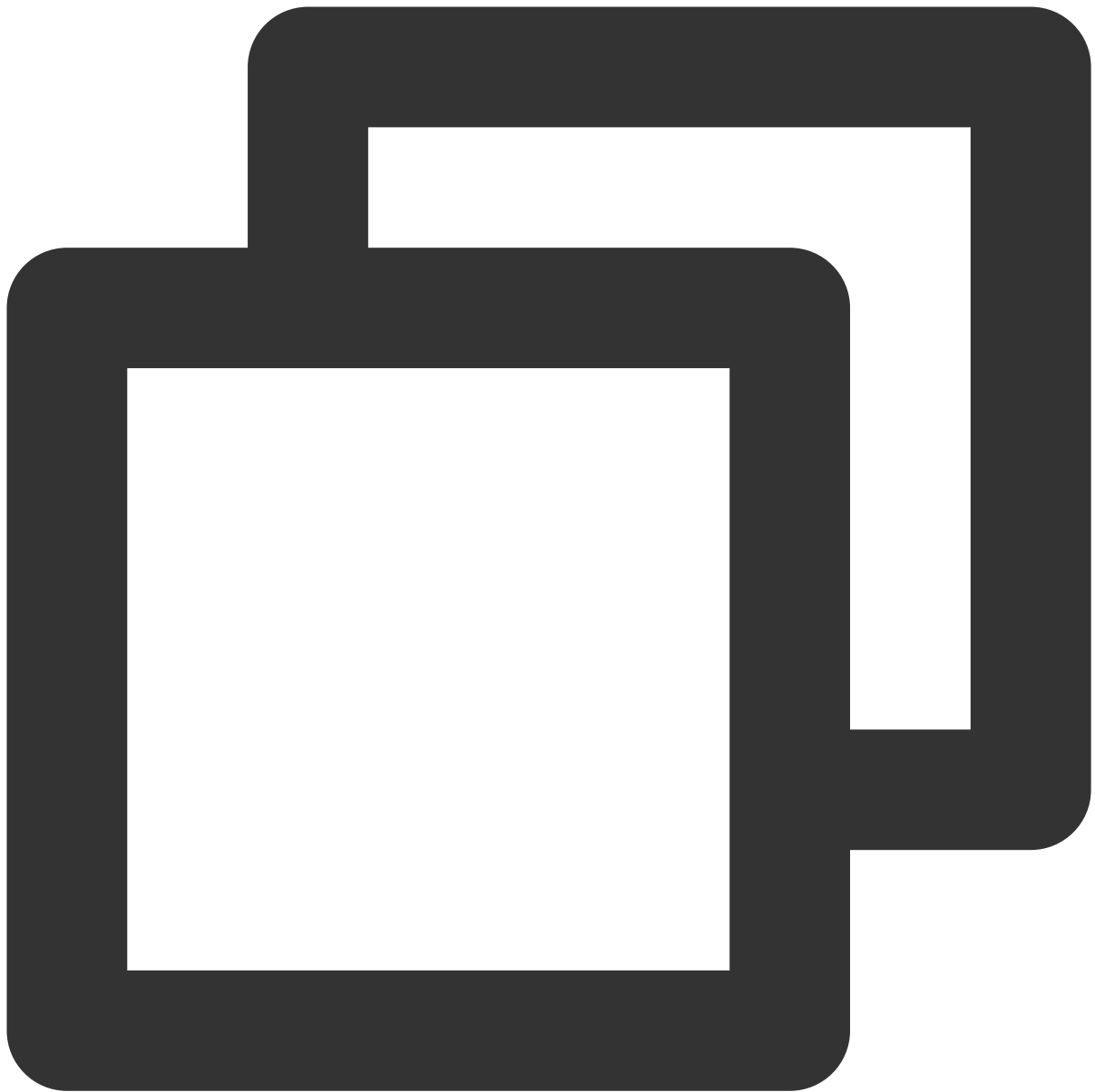
After entering the password, press **Enter** to confirm it. The password is hidden by default.

3. Run the following command to create a file directory (such as `/var/ftp/test` ) for the FTP service.



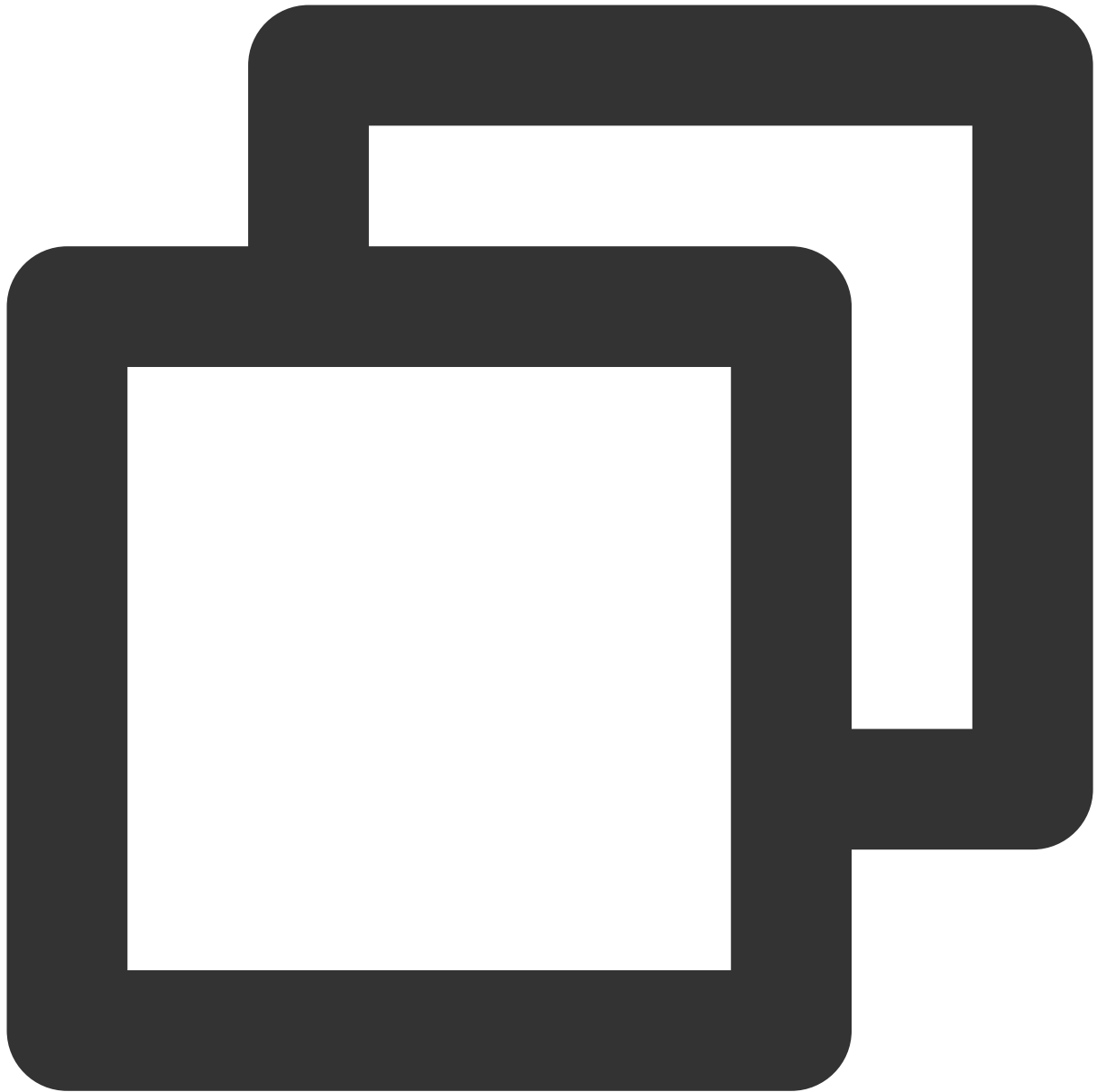
```
sudo mkdir /var/ftp/test
```

4. Run the following command to modify the directory permission.



```
sudo chown -R ftpuser:ftpuser /var/ftp/test
```

5. Run the following command to open the `vsftpd.conf` file.



```
sudo vim /etc/vsftpd/vsftpd.conf
```

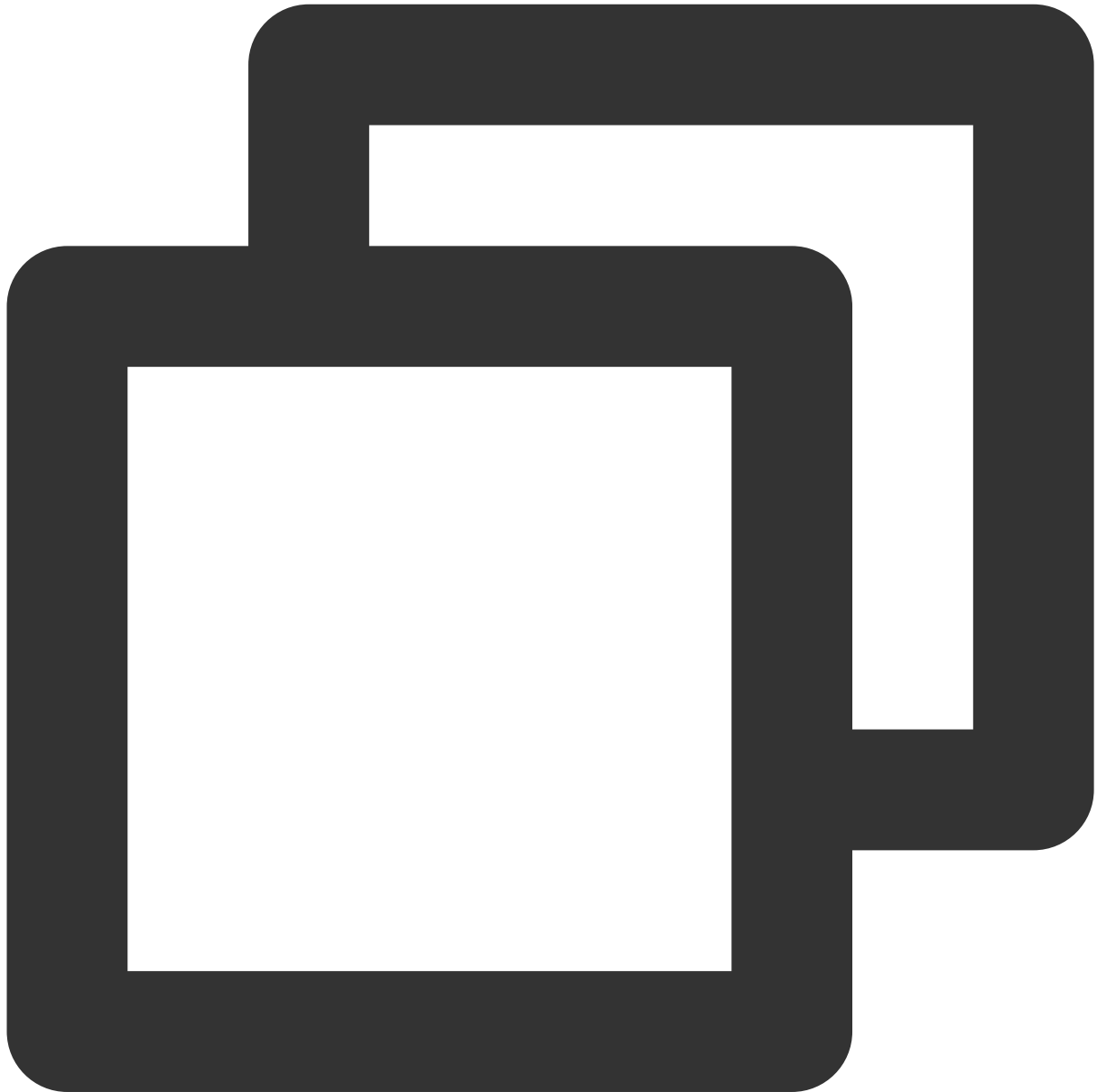
6.  
Press **i** to switch to the edit mode  
. Select an FTP mode as needed and modify the `vsftpd.conf` configuration file.

**Note:**

The FTP server can connect to the client in either active or passive mode for data transmission. Due to the firewall settings of most clients and the fact that the actual IP address cannot be obtained, we recommend that you use the

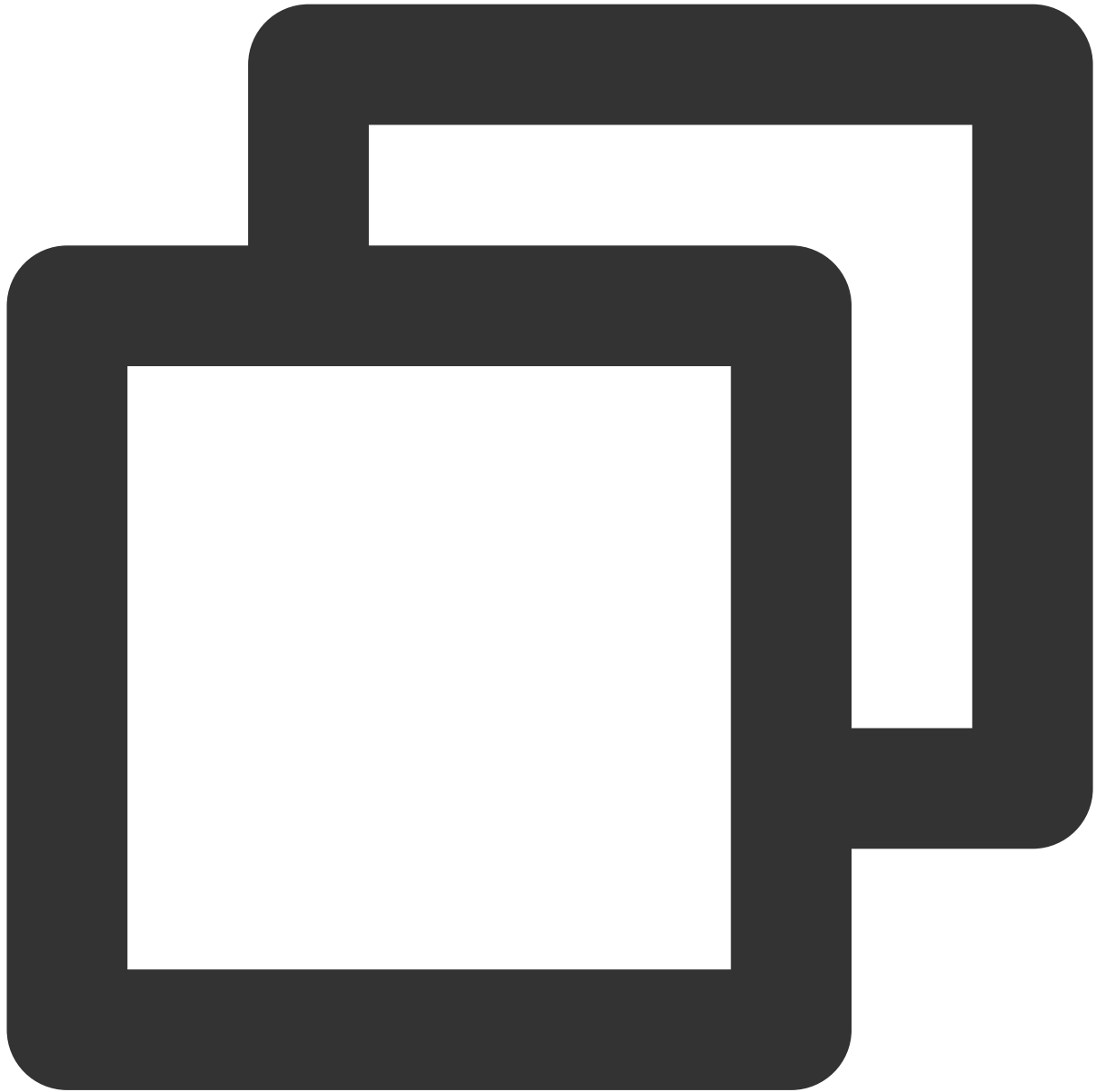
**passive mode** to set up the FTP service. The following modification uses the passive mode as an example. To use the active mode, see [Setting the FTP active mode](#).

6.1 Modify the following configuration parameters to set login permissions for anonymous and local users, set the path for storing the exceptional user list, and enable listening on IPv4 sockets.



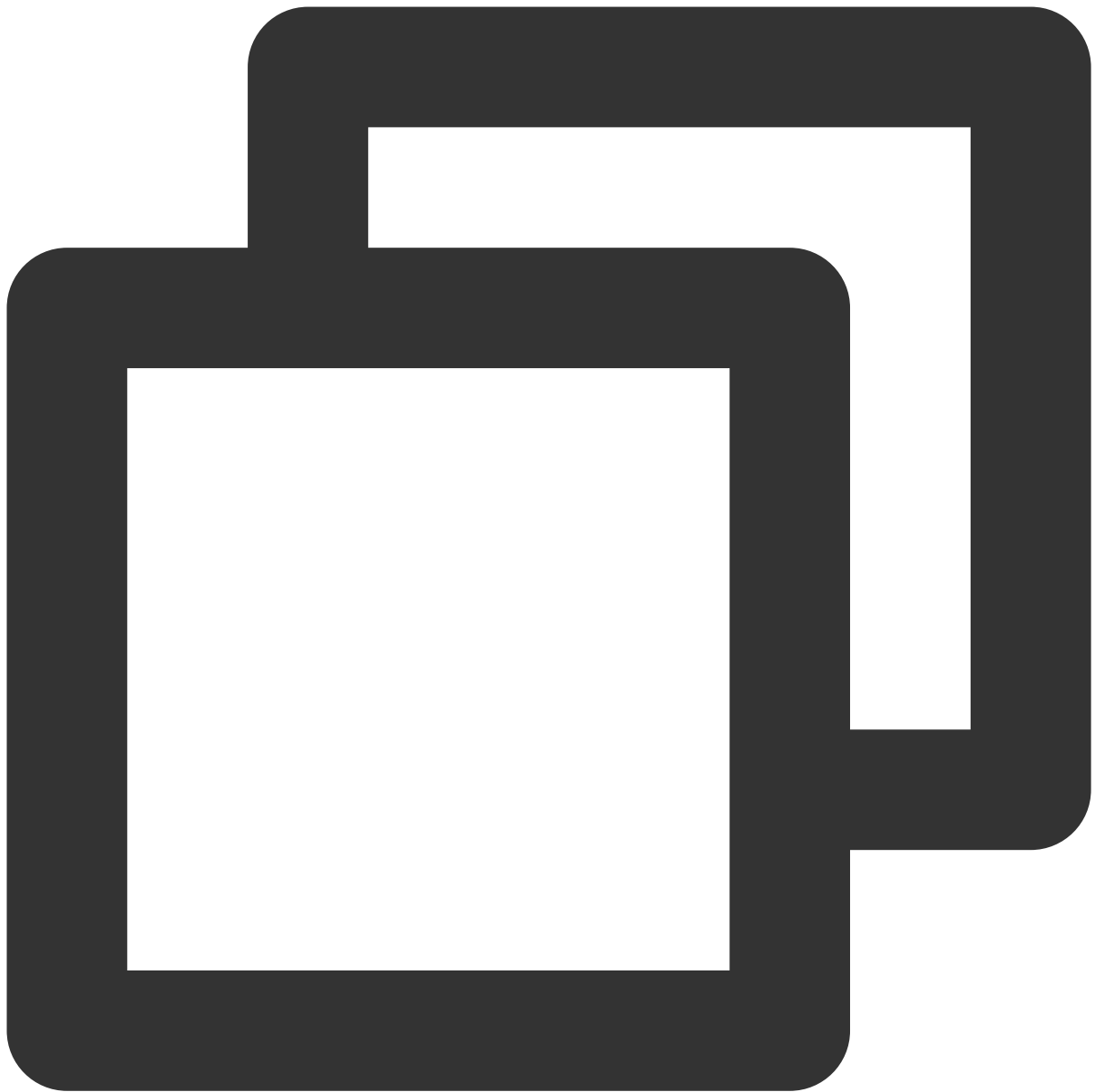
```
anonymous_enable=NO
local_enable=YES
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
listen=YES
```

6.2 Add the pound sign ( # ) at the beginning of the following line to comment out `listen_ipv6=YES` and disable listening on IPv6 sockets.



```
#listen_ipv6=YES
```

6.3 Add the following configuration parameters to enable the passive mode, set the directory where local users reside after login, and set the port range for transmitting data by the CVM.



```
local_root=/var/ftp/test
allow_writeable_chroot=YES
pasv_enable=YES
pasv_address=xxx.xx.xxx.xx # Replace xxx.xx.xxx.xx with the public IP address of yo
pasv_min_port=40000
pasv_max_port=45000
```

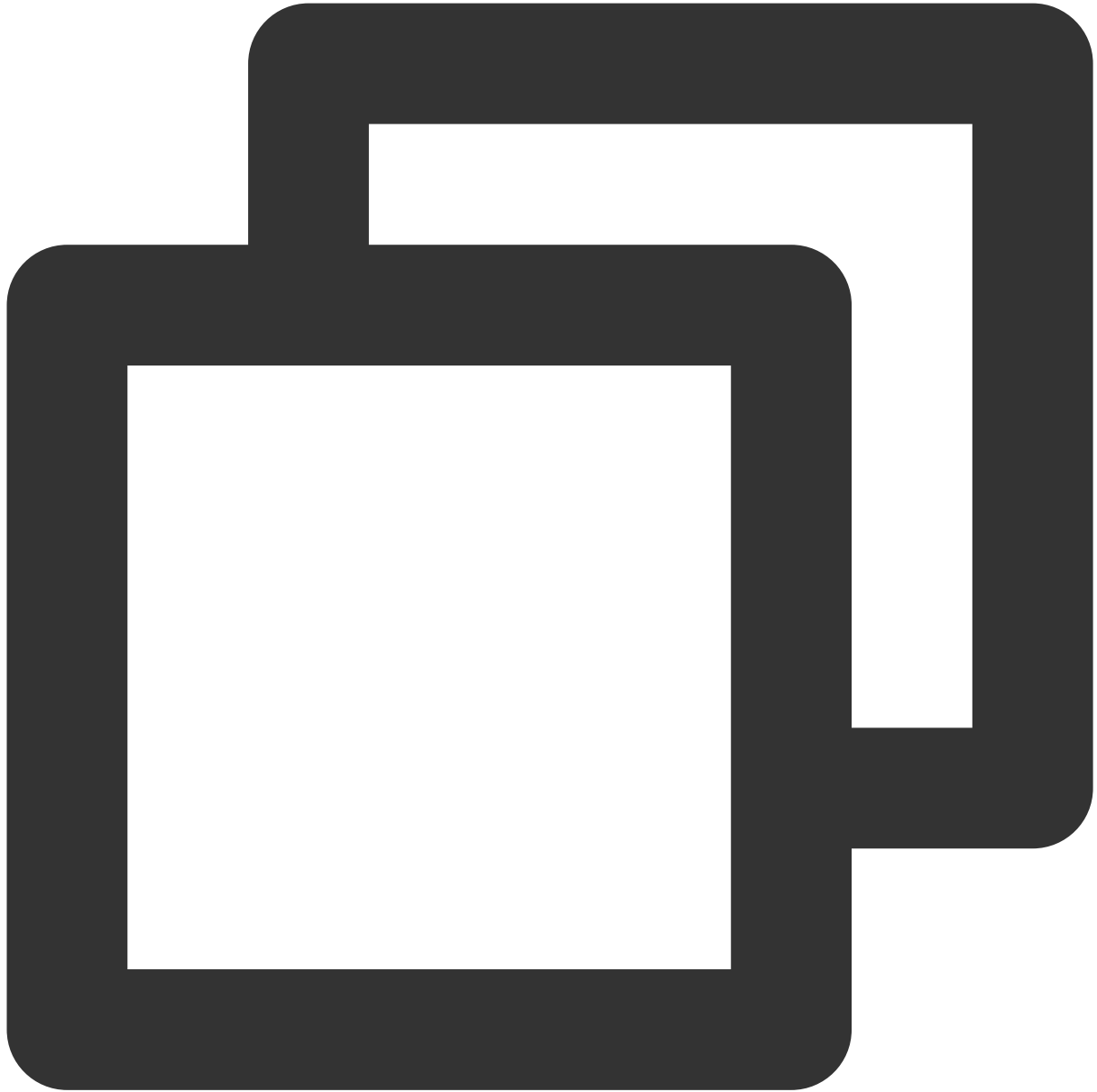
7. Press **Esc** and enter **:wq** to save and close the file.

8.

Run the following command to create



and edit the `chroot_list` file.

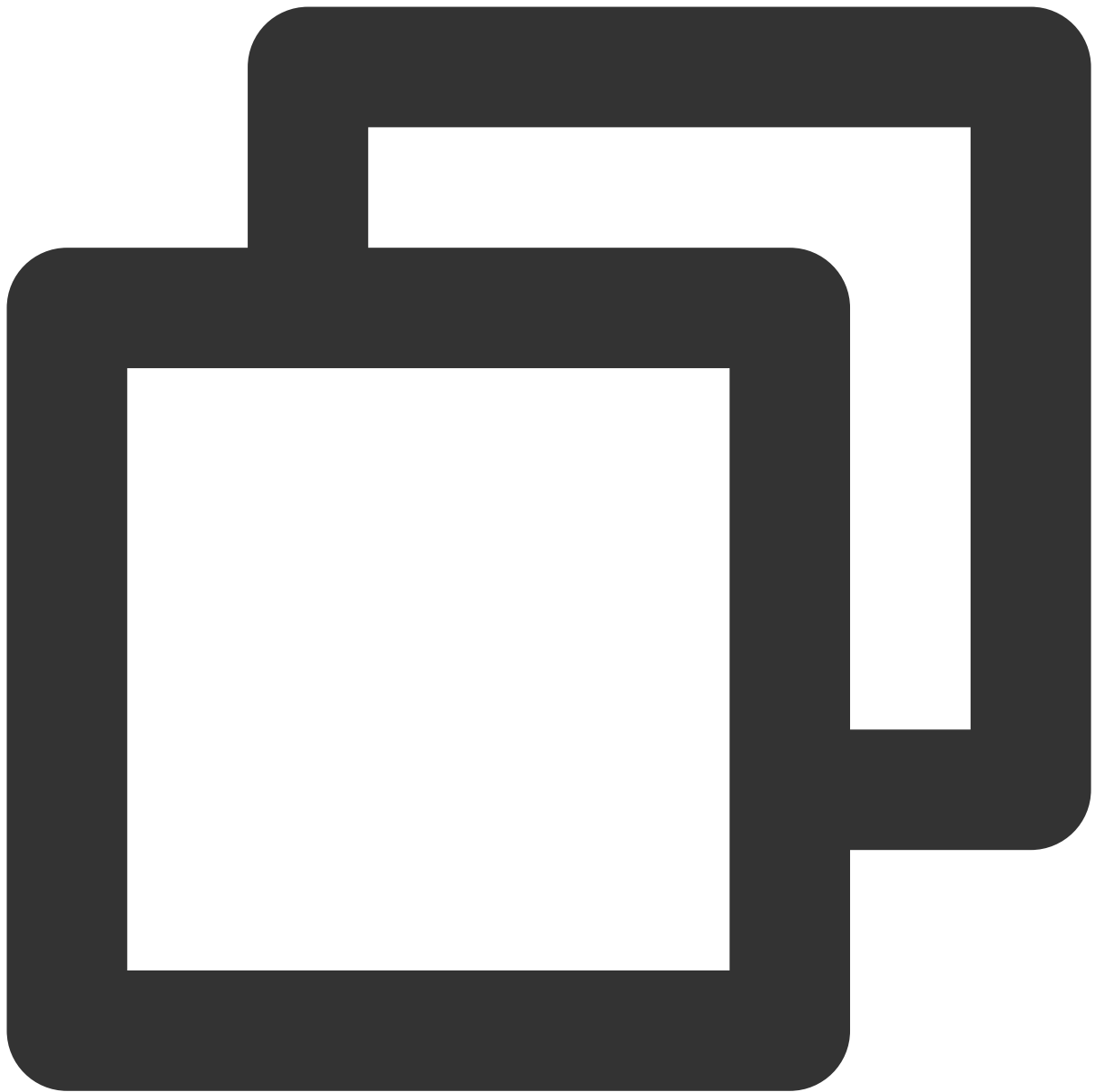


```
sudo vim /etc/vsftpd/chroot_list
```

9. Press **i** to enter the edit mode and enter usernames. Note that each username occupies one line. After finishing the configuration, press **Esc** and enter **:wq** to save and close the file.

If you do not need to set exceptional users, skip this step by entering **:wq** to close the file.

10. Run the following command to restart the FTP service.



```
sudo systemctl restart vsftpd
```

#### Step 4. Configure the security group

After setting up the FTP service, you need to open the corresponding ports of the Linux Lighthouse instance according to the FTP mode actually used. For more information, see [Adding firewall rules](#).

Most clients convert IP addresses in LANs. If you are using the FTP active mode, ensure that the client has obtained the actual IP address. Otherwise, the client may fail to log in to the FTP server.

Active mode: Open port 21.

Passive mode: Open port 21 and all ports ranging from `pasv_min_port` to `pasv_max_port` set in the [configuration file](#), such as ports 40000 to 45000 in this document.

## Step 5. Verify the FTP service

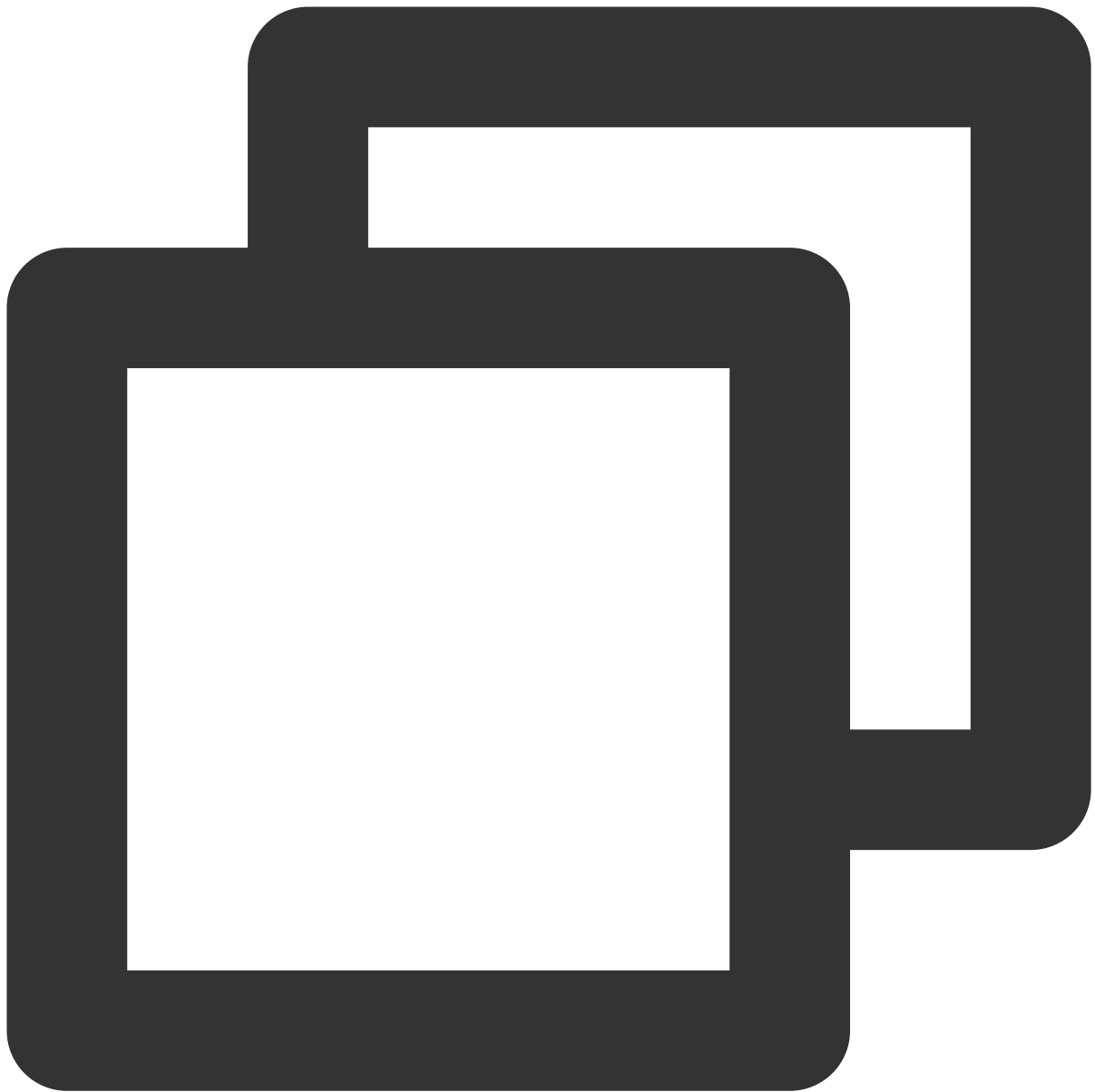
You can use tools such as the FTP client software, browser, or file manager to verify the FTP server. This document uses the file manager of the client as an example.

1. Open Internet Explorer on the client, choose **Tools > Internet Options**, and click the **Advanced** tab. Make the following changes based on the selected FTP mode.

Active mode: Deselect **Passive FTP**.

Passive mode: Select **Passive FTP**.

2. Open the computer where the client is installed and enter the following path in the search box:



```
ftp://Lighthouse instance public IP:21
```

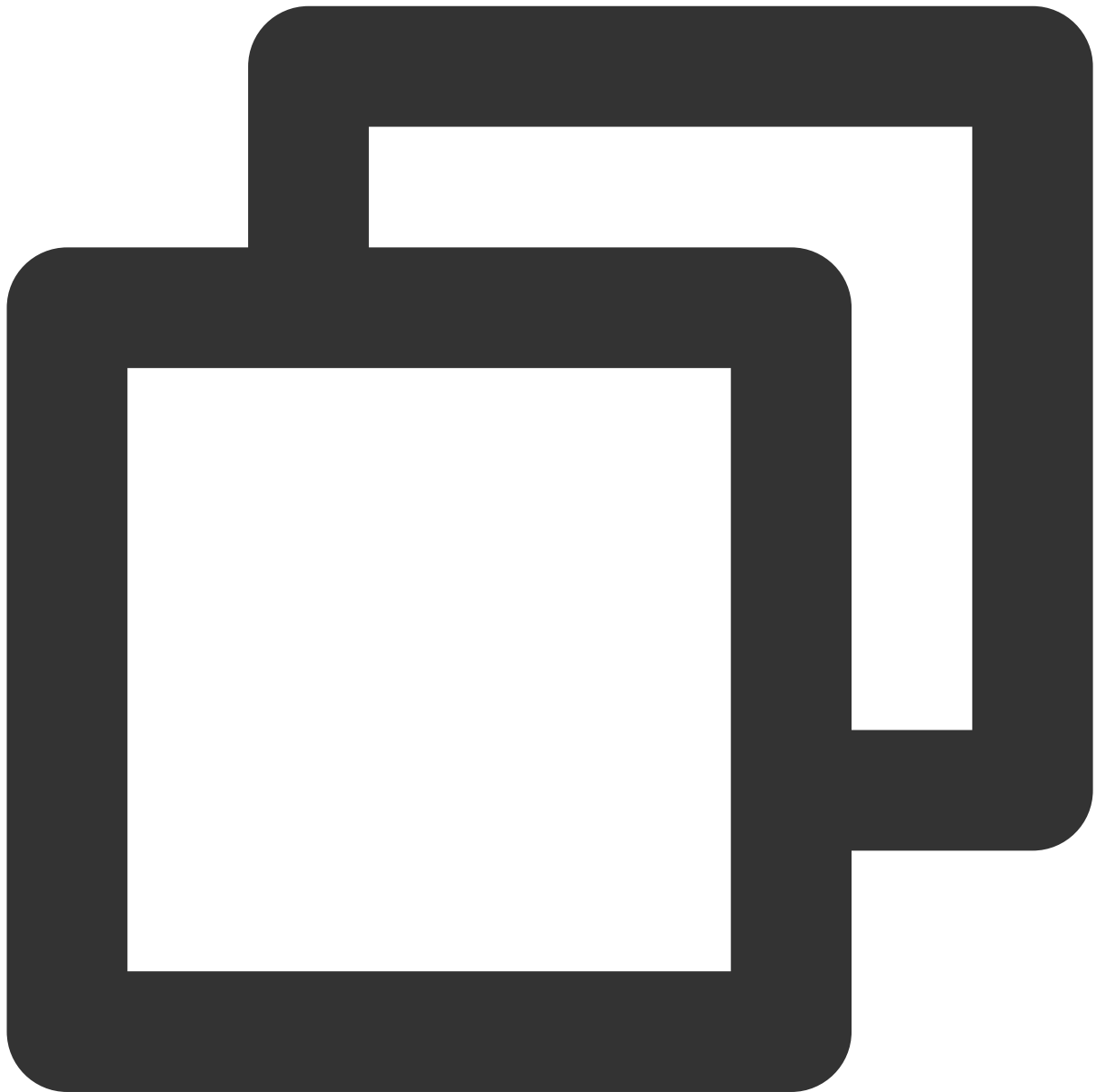


3. On the login page that appears, enter the username and password set in [Configure vsftpd](#).
4. You can upload and download files after a successful login.

## Appendix

### Setting FTP active mode

To use the active mode, modify the following configuration parameters and leave others as their defaults:



```
anonymous_enable=NO      # Forbid anonymous users to log in
local_enable=YES         # Allow local users to log in
chroot_local_user=YES     # Restrict all users to access only the root directory
chroot_list_enable=YES    # Enable the exceptional user list
chroot_list_file=/etc/vsftpd/chroot_list # Specify the user list, in which the lis
```

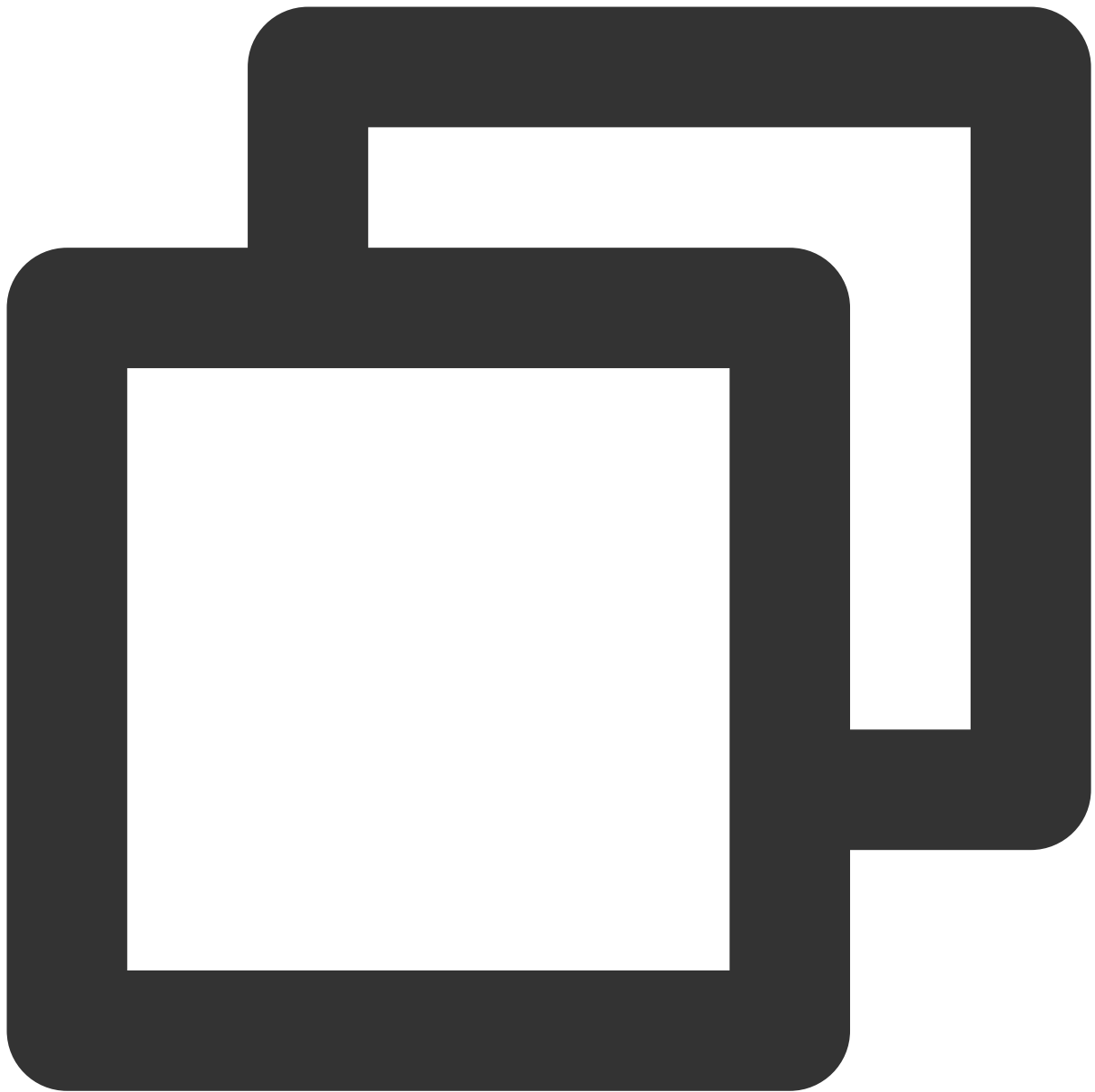
```
listen=YES                # Enable listening on IPv4 sockets
# Add the pound sign (#) at the beginning of the following line to comment it out
#listen_ipv6=YES          # Disable listening on IPv6 sockets
# Add the following parameters
allow_writeable_chroot=YES
local_root=/var/ftp/test # Set the directory where local users reside after login
```

Press **Esc** and enter **:wq** to save and close the file. After that, proceed to [Step 11](#) of “Step 3. Configure vsftpd”.

## FTP client failure to upload file

### Problem

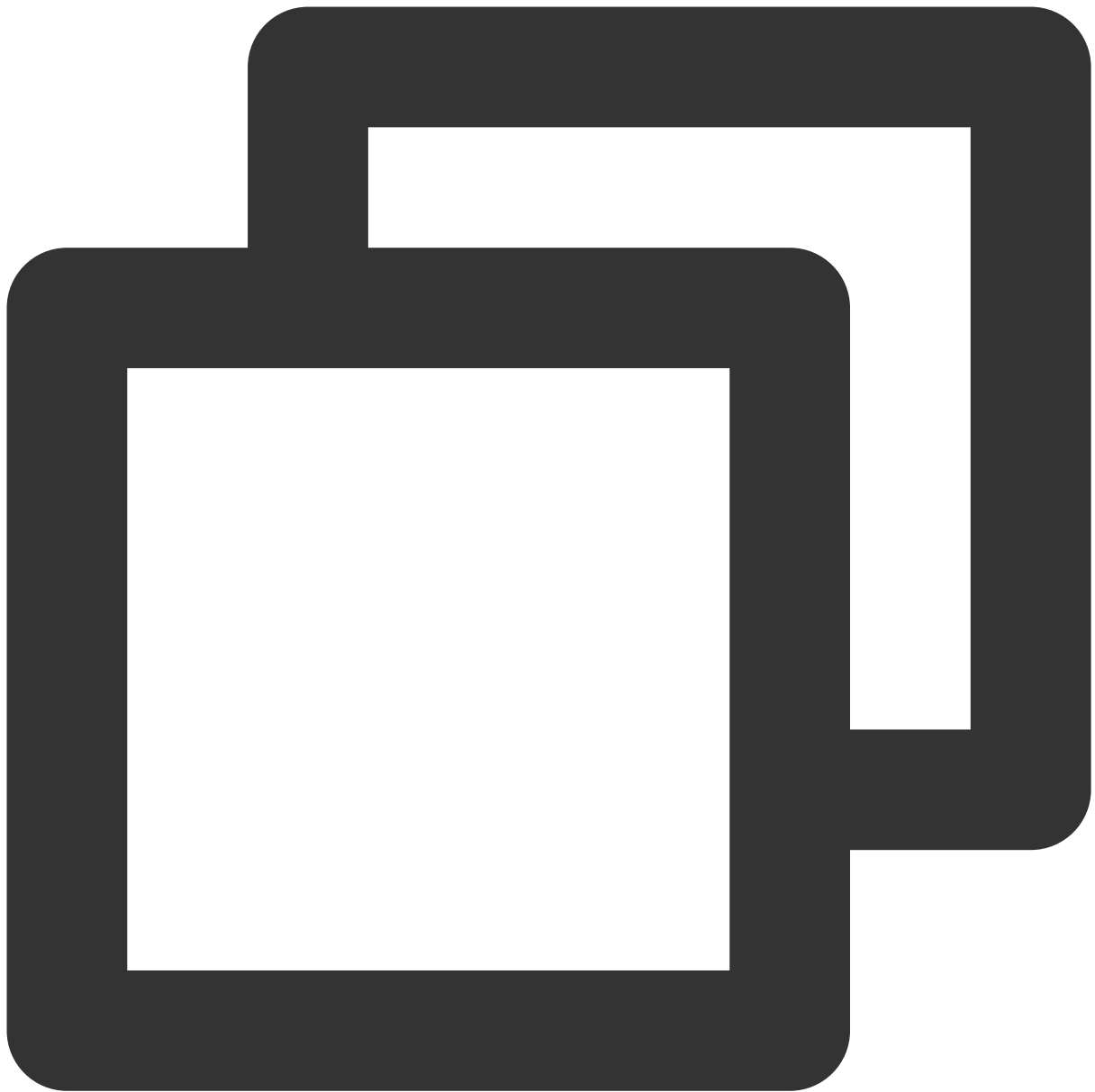
In the Linux environment, users encounter the following error message when uploading files with vsftpd.



```
553 Could not create file
```

### Solution

1. Run the following command to check the disk space utilization of the server.



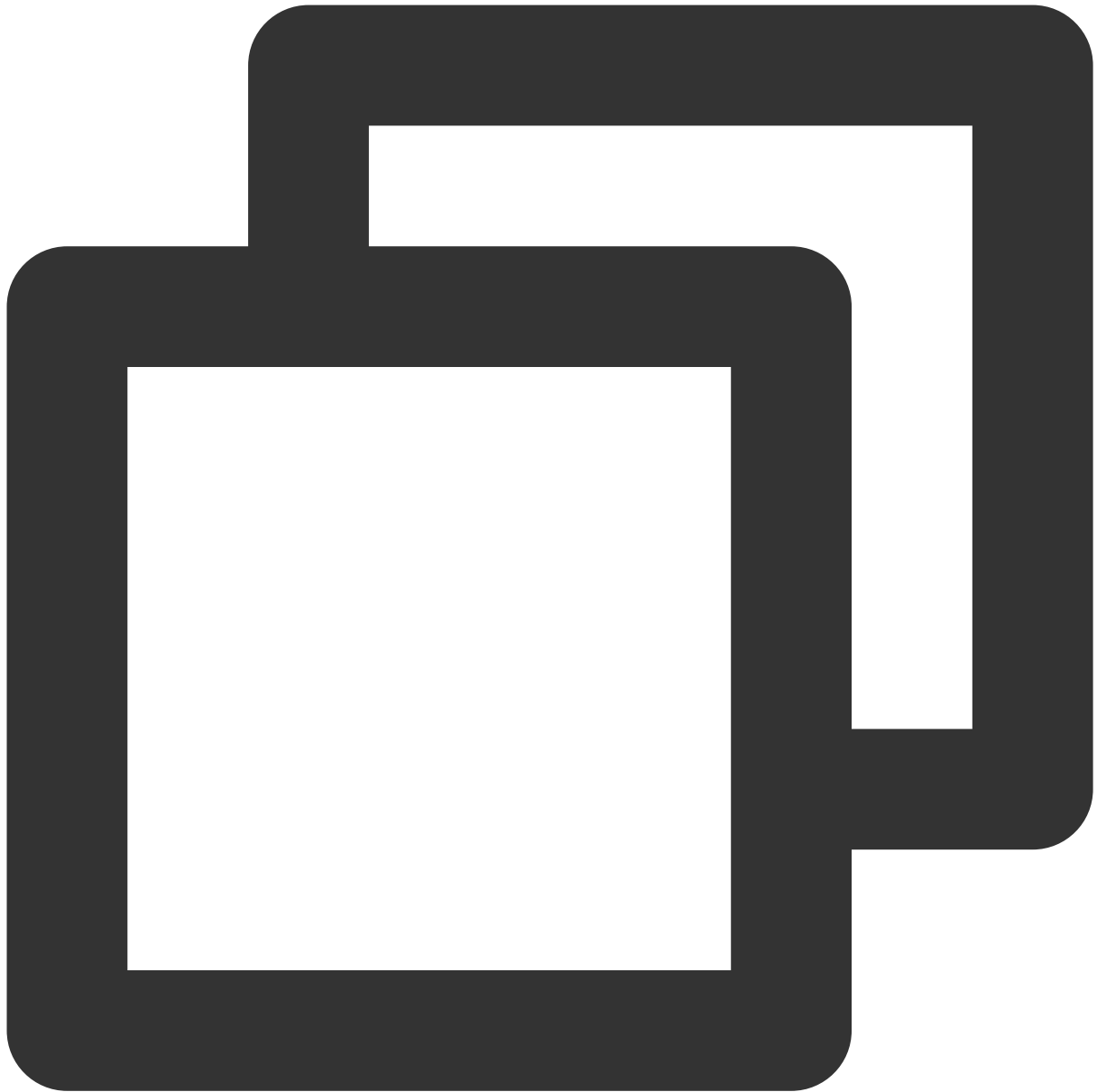
```
df -h
```

If the disk space is insufficient, you cannot upload files. In this case, we recommend you delete some unnecessary large files from the disk.

If the disk space is sufficient, proceed to the next step.

2. Run the following command to check whether you have the write permission to the FTP directory.





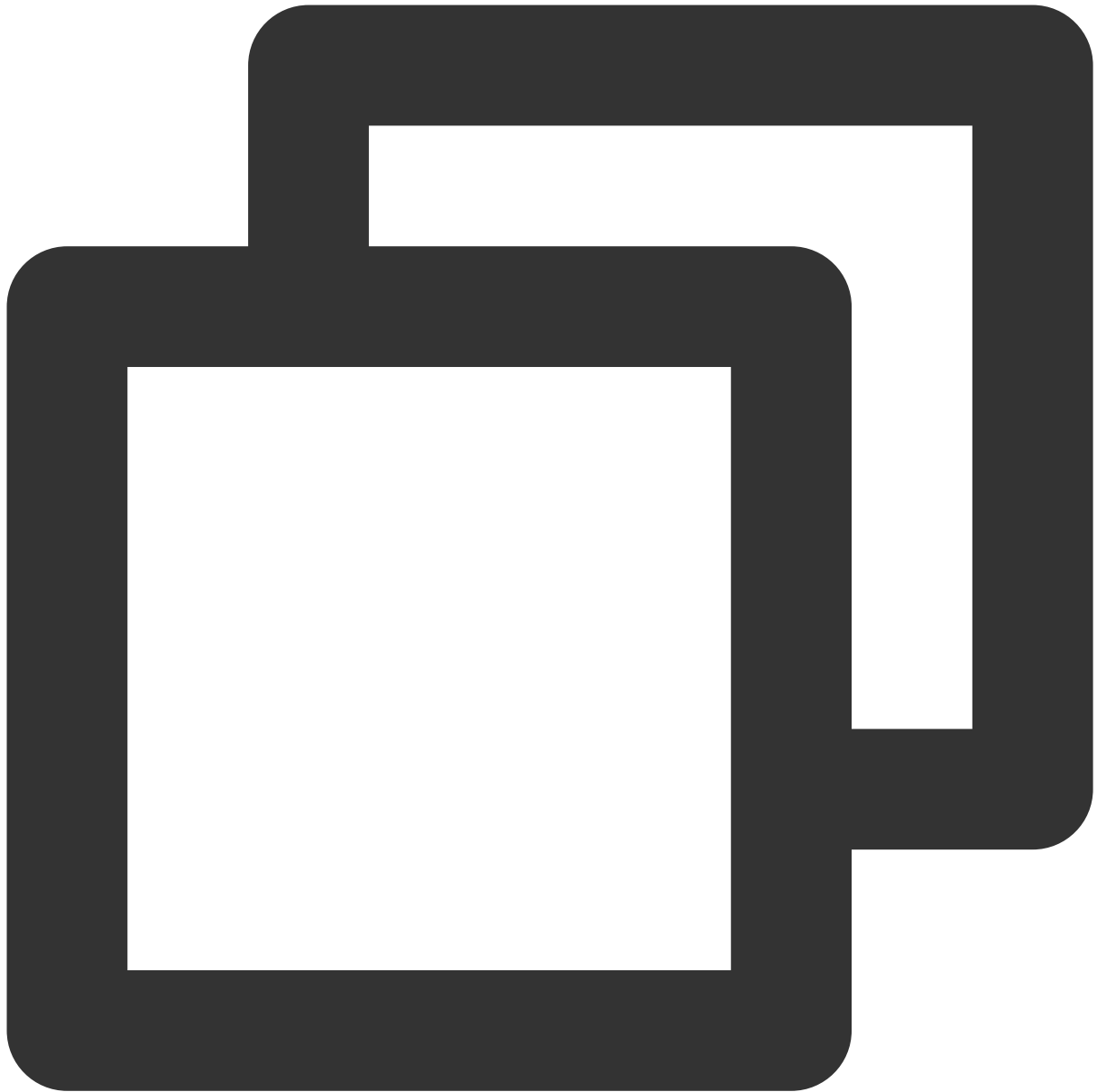
```
ls -l /home/test
```

```
# Here, `/home/test` indicates the FTP directory. Replace it with your actual FTP d
```

If `w` is not returned in the result, you don't have the write permission to the directory. In this case, proceed to the next step.

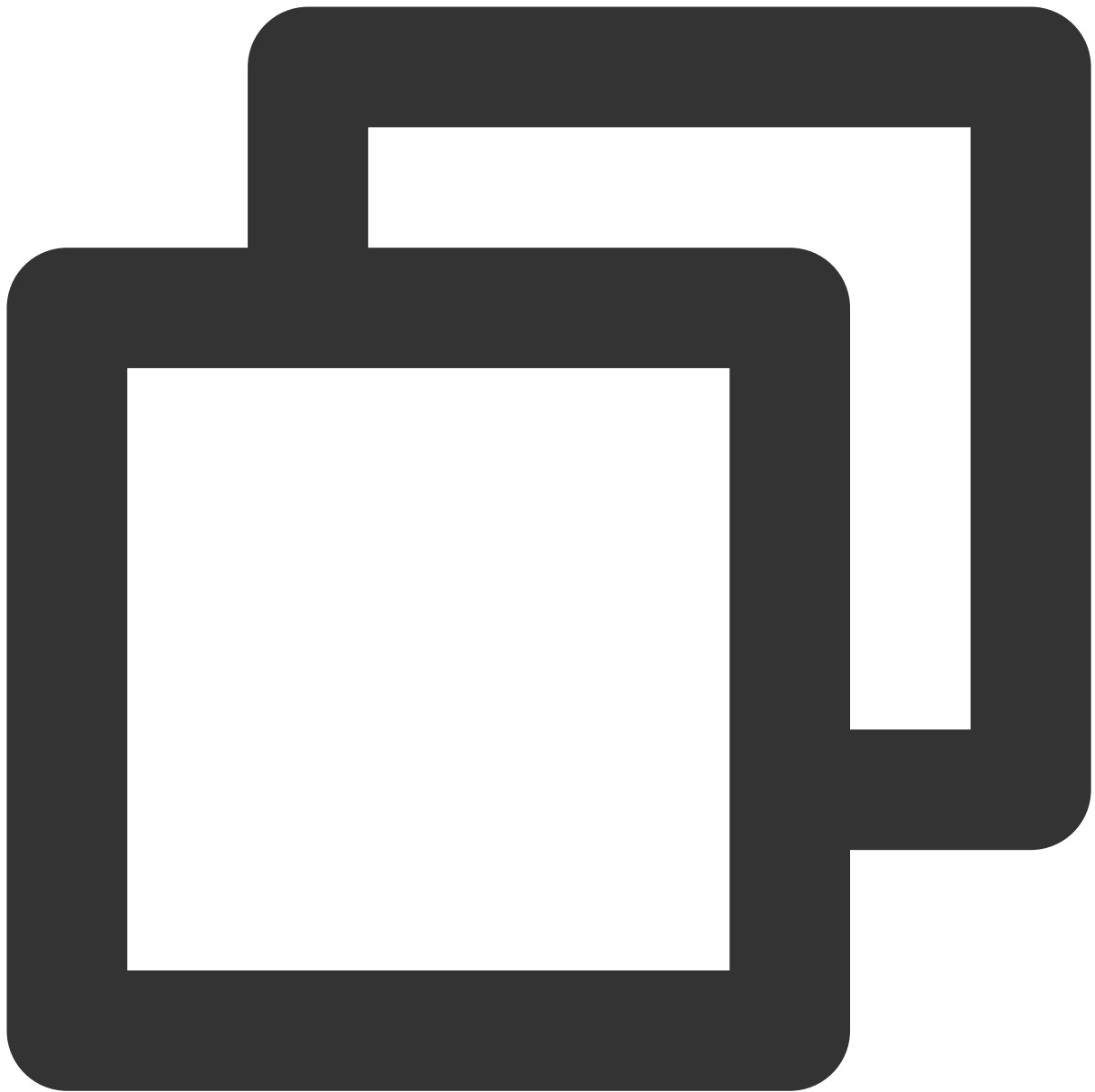
If `w` is returned in the result, [submit a ticket](#) for assistance.

3. Run the following command to grant the write permission to the FTP directory.



```
sudo chmod +w /home/test  
# Here, `/home/test` indicates the FTP directory. Replace it with your actual FTP d
```

4. Run the following command to check whether the write permission is successfully granted:



```
ls -l /home/test
```

```
# Here, `/home/test` indicates the FTP directory. Replace it with your actual FTP d
```

# Setting Up an FTP Server on Windows Lighthouse Instance

Last updated : 2022-06-15 18:21:41

## Overview

This document describes how to use IIS to build an FTP site on a Windows Lighthouse instance.

## Software

The following software programs are used to build the FTP service:

Windows OS: This document uses the Windows Server 2012 system image as an example.

IIS: Web server. This document uses IIS 8.5 as an example.

## Directions

### Step 1. Log in to the Lighthouse instance

You can log in to the Windows instance via [VNC](#) or [remote desktop connection](#).

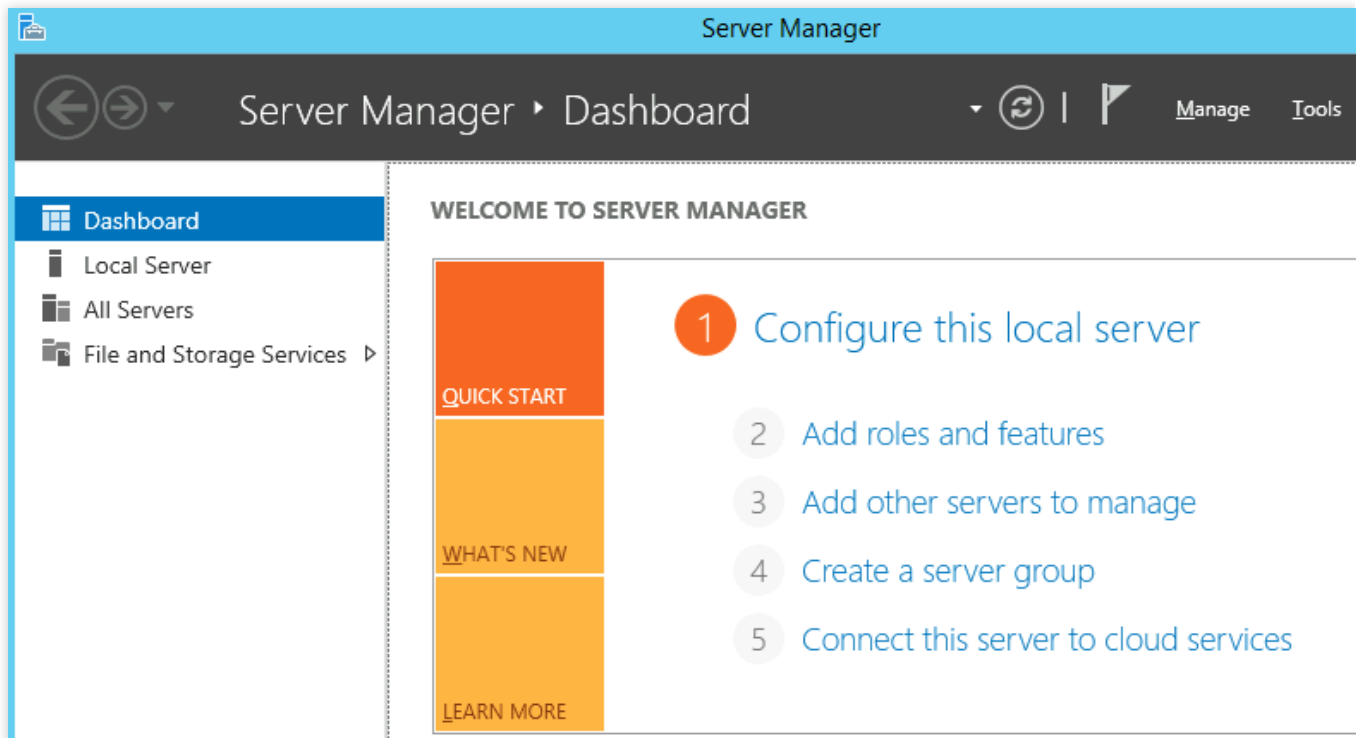
### Step 2. Install the FTP service on IIS

1. On the desktop, click



to open the server manager.


2. In the **Server Manager** window, click **Add roles and features**.



3. In the **Add Roles and Features Wizard** pop-up window, click **Next** to enter the **Installation Type** page.

4. Select **Role-based or feature-based installation** and click **Next**.

5. On the **Select destination server** page, keep the default configurations and click **Next**.

 Add Roles and Features Wizard

Select destination server

Before You Begin

Installation Type

**Server Selection**

Server Roles

Features

Confirmation

Results

Select a server or a virtual hard disk on which to install roles and features.

☒ Select a server from the server pool

☐ Select a virtual hard disk

Server Pool

Filter:

Name	IP Address	Operating System
10_0_21_22	10.0.21.22	Microsoft Windows Server 2012 R2 Da

1 Computer(s) found

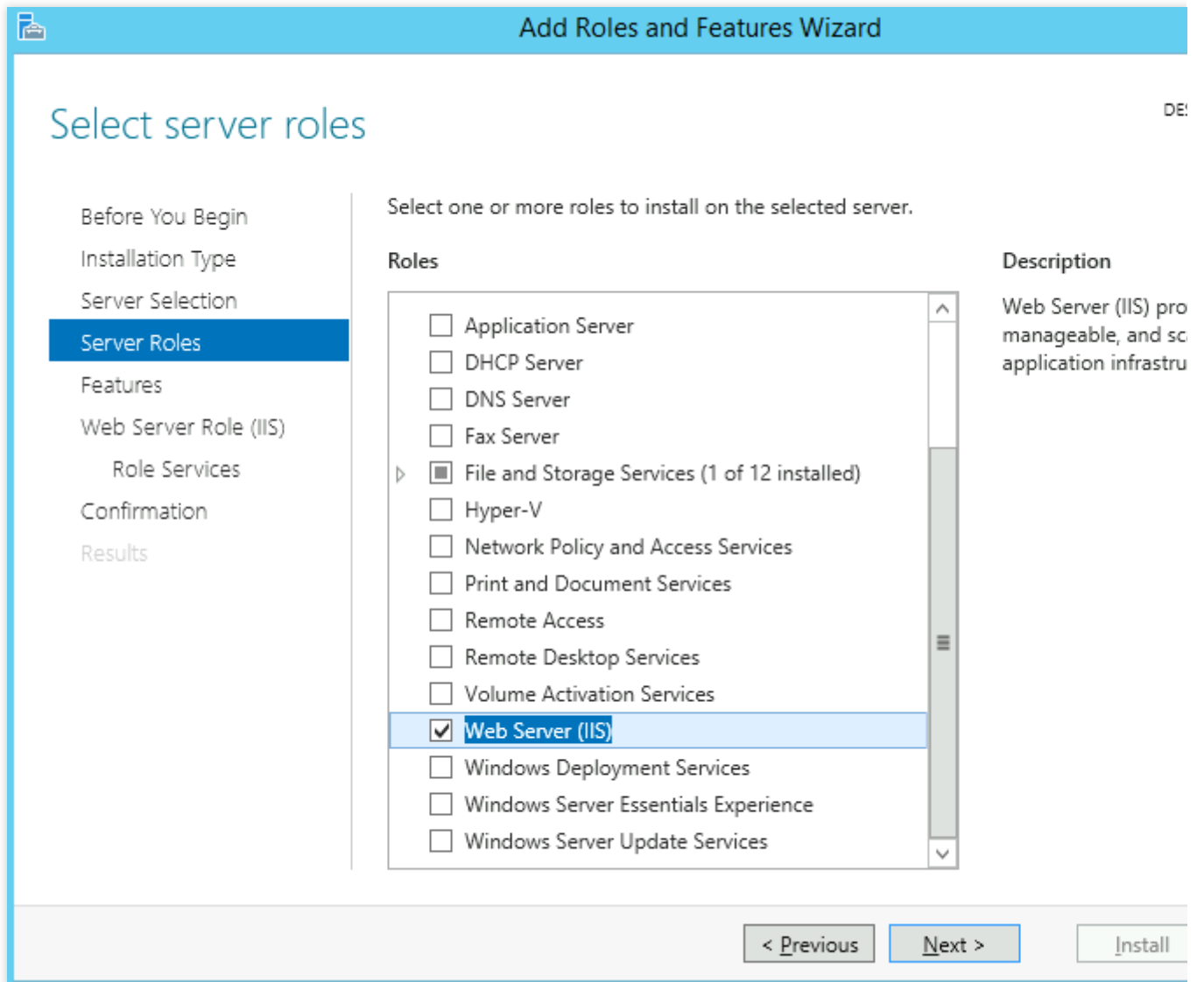
This page shows servers that are running Windows Server 2012, and that have been added by the Add Servers command in Server Manager. Offline servers and newly-added servers from the collection is still incomplete are not shown.

< Previous

Next >

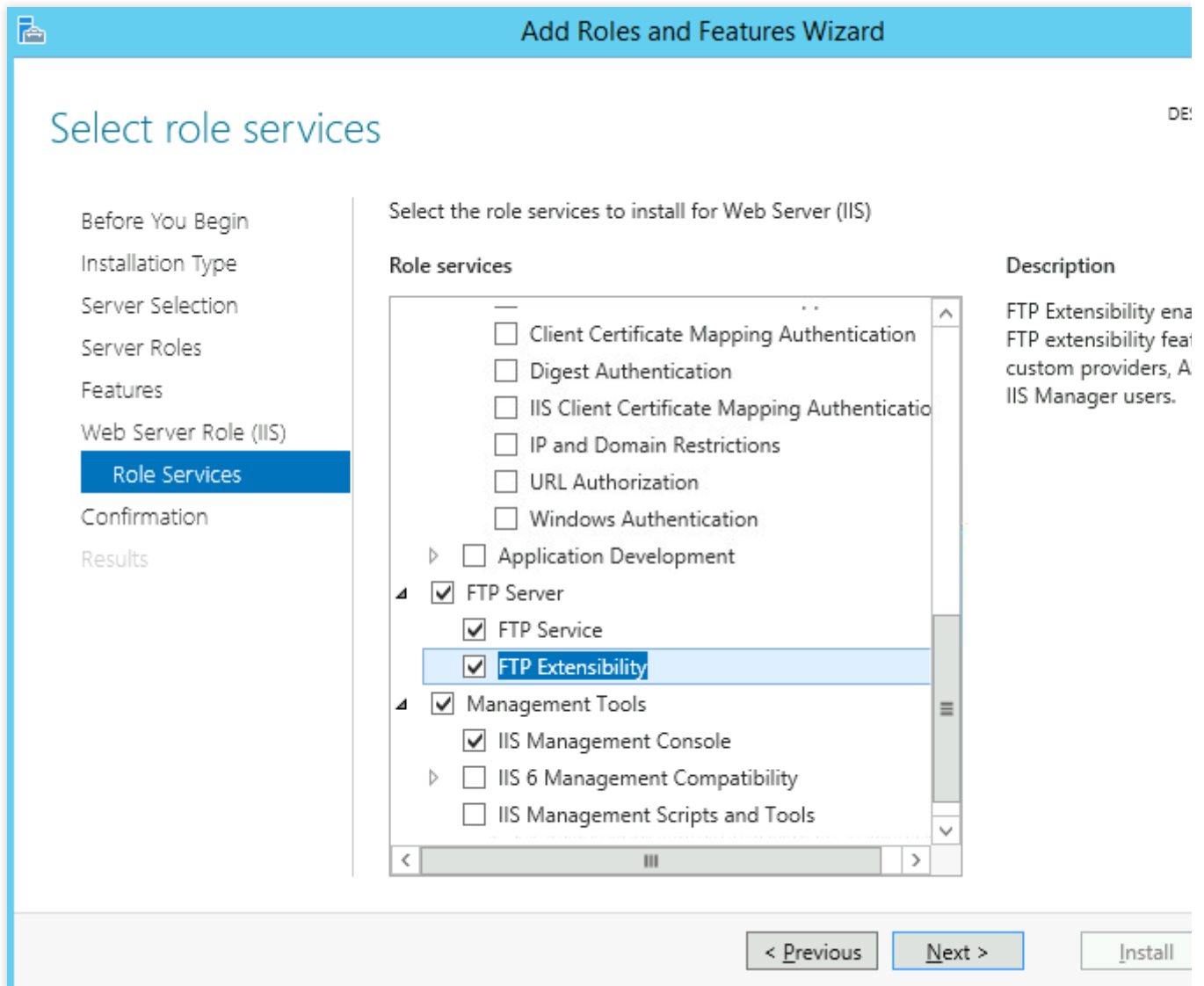
Install

6. On the **Select server roles** page, select **Web Server (IIS)** and click **Add Feature** in the pop-up window.



7. Click **Next** for the next three pages to enter the **Select role services** page.

8. On the **Select role services** page, select **FTP Service** and **FTP Extensibility** and click **Next**.



9. Click **Install**.

10. After the installation is completed, click **Close**.

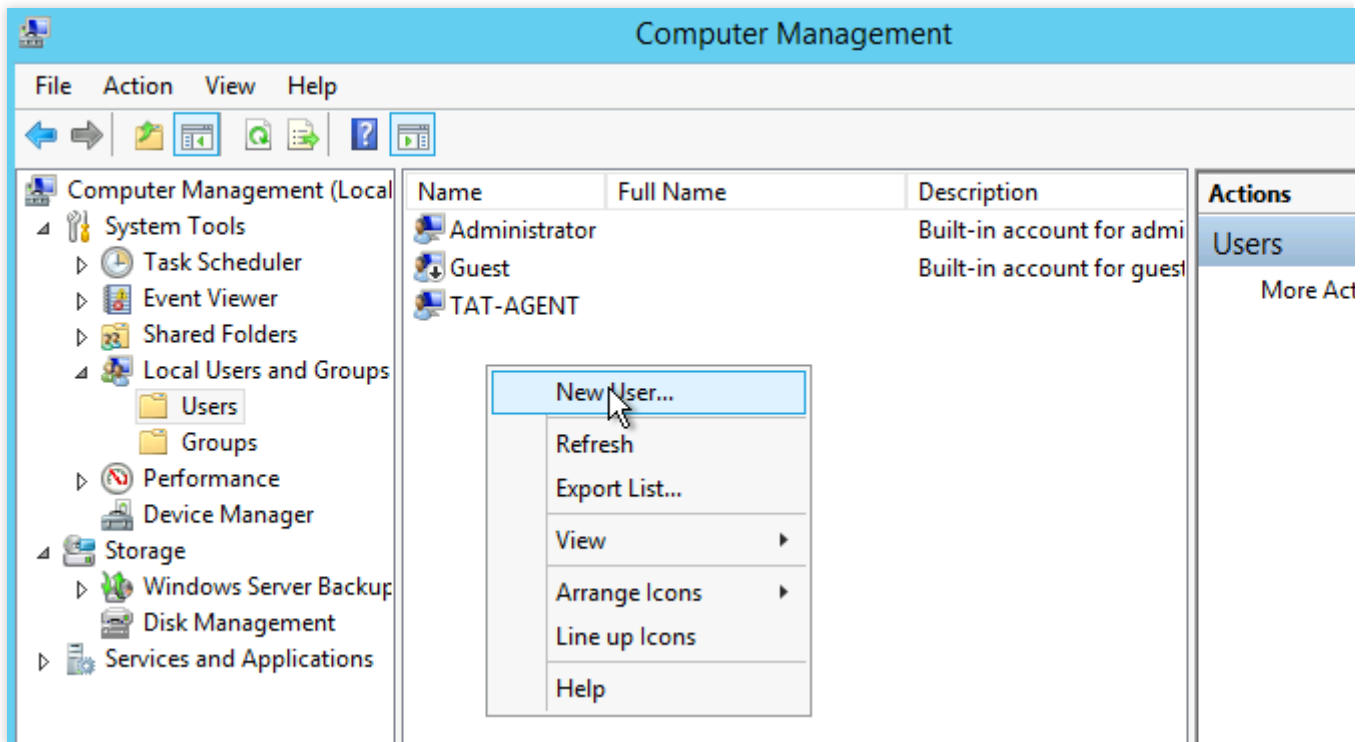
### Step 3. Set up the FTP username and password

#### Note:

Follow the steps below to configure your FTP username and password. If you plan to use anonymous access only, skip this section.

1. In the **Server Manager** window, select **Tools > Computer Management** on the top-right navigation bar to open the **Computer Management** window.
2. On the **Computer Management** page, select **System Tools > Local Users and Groups > Users** on the left sidebar.
3. In the right of the **Users** page, right-click the blank space and select **New User**.





4. On the **New User** page, configure the username and password as prompted and click **Create**.

The 'New User' dialog box is shown. It has a title bar with a question mark and a close button. The form contains the following fields and options:

- User name:** ftpuser
- Full name:** (empty)
- Description:** (empty)
- Password:** (masked with dots)
- Confirm password:** (masked with dots)
- ☐ User must change password at next logon
- ☐ User cannot change password
- ☒ Password never expires
- ☐ Account is disabled
- Buttons:** Help, Create, Close

Set the main parameters as follows:

**Username:** Custom. This document uses `ftpuser` as an example.

**Password and Confirm Password:** Set a custom password that meets the following requirements:

Cannot contain the username or more than two consecutive characters of the username.

Must contain at least six characters.

Must contain characters in at least three of the four character types: `[A - Z]` , `[a - z]` , `[0 - 9]` , and special symbols (such as `!$#%` ).

Deselect **User must change password at next logon** and select **Password never expires**.

Select options based on your actual needs. This document uses **Password never expires** as an example.

5. Click **Close**. You can see the created `ftpuser` in the list.

## Step 4. Set the shared folder permission

### Note:

This document uses the `C:\\test` folder as the shared folder of the FTP site. The folder contains the `test.txt` file you want to share with others. Create the `C:\\test` folder and the `test.txt` file under it as instructed. You can also use any other folder as needed.

1. On the desktop, click



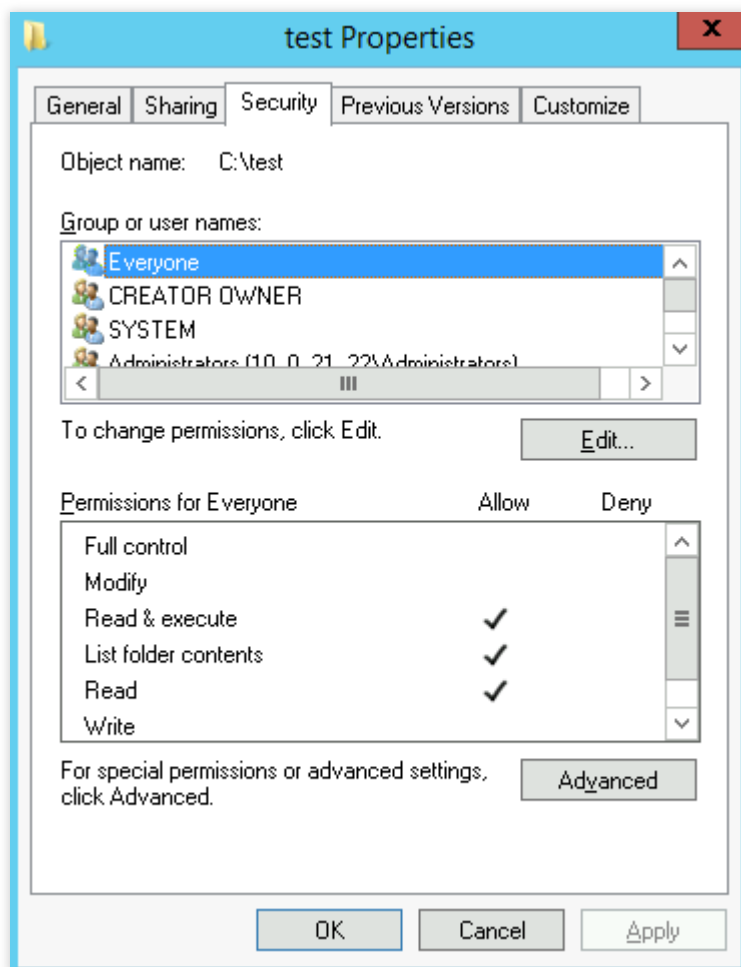
to open the **This PC** window.

2. Select and right-click the `test` folder under the C drive. Select **Properties**.

3. In the **test Properties** window, select the **Security** tab.

4. Select `Everyone` and click **Edit**.

If **Group or user names** does not contain `Everyone` , see [Adding Everyone user](#) to add the user.

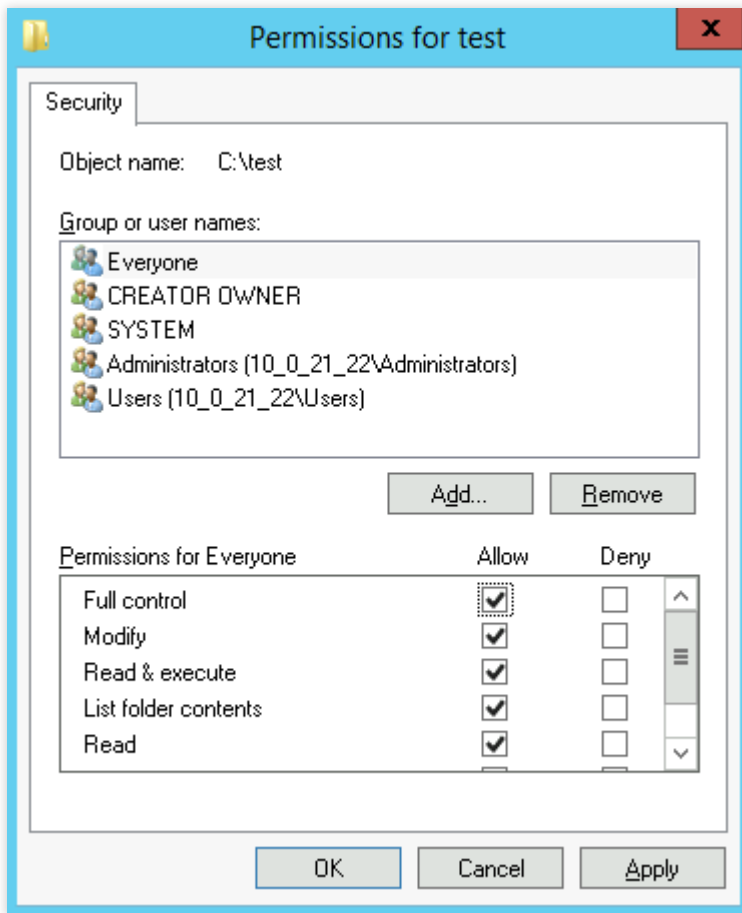


5.

On the **Permissions for test** page

, set the permission for `Everyone` and click **OK**.

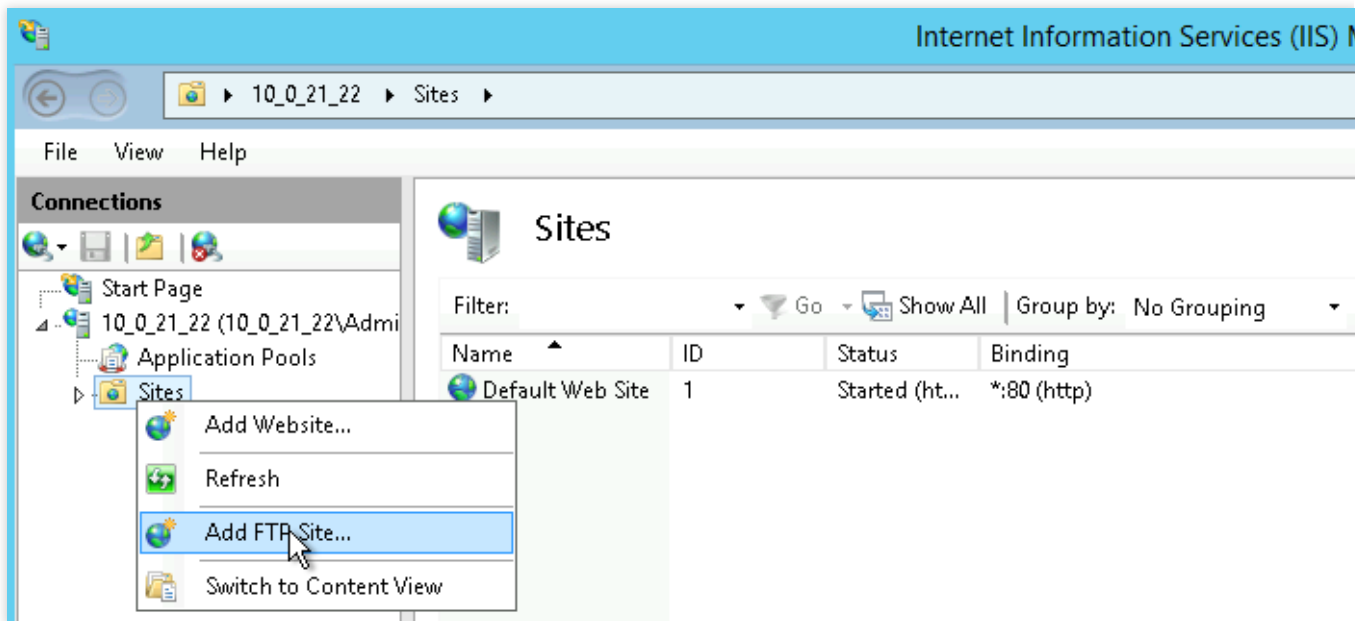
This document uses granting `Everyone` all permissions as an example.



6. Click **OK** to complete the configuration.

## Step 5. Add an FTP site

1. In the **Server Manager** window, select **Tools > Internet Information Services (IIS) Manager** on the top-right navigation bar.
2. In the **Internet Information Services (IIS) Manager** pop-up window, expand your server in the left sidebar, right-click **Sites**, and select **Add FTP Site**.



3. On the **Site Information** page, enter the following information and click **Next**.

The screenshot shows the 'Add FTP Site' wizard, specifically the 'Site Information' page. It has a title bar 'Add FTP Site' with a question mark and close button. Below the title bar is a globe icon and the text 'Site Information'. The form contains two main sections: 'FTP site name:' with a text box containing 'ftp', and 'Content Directory' which includes 'Physical path:' with a text box containing 'C:\test' and a browse button (...).

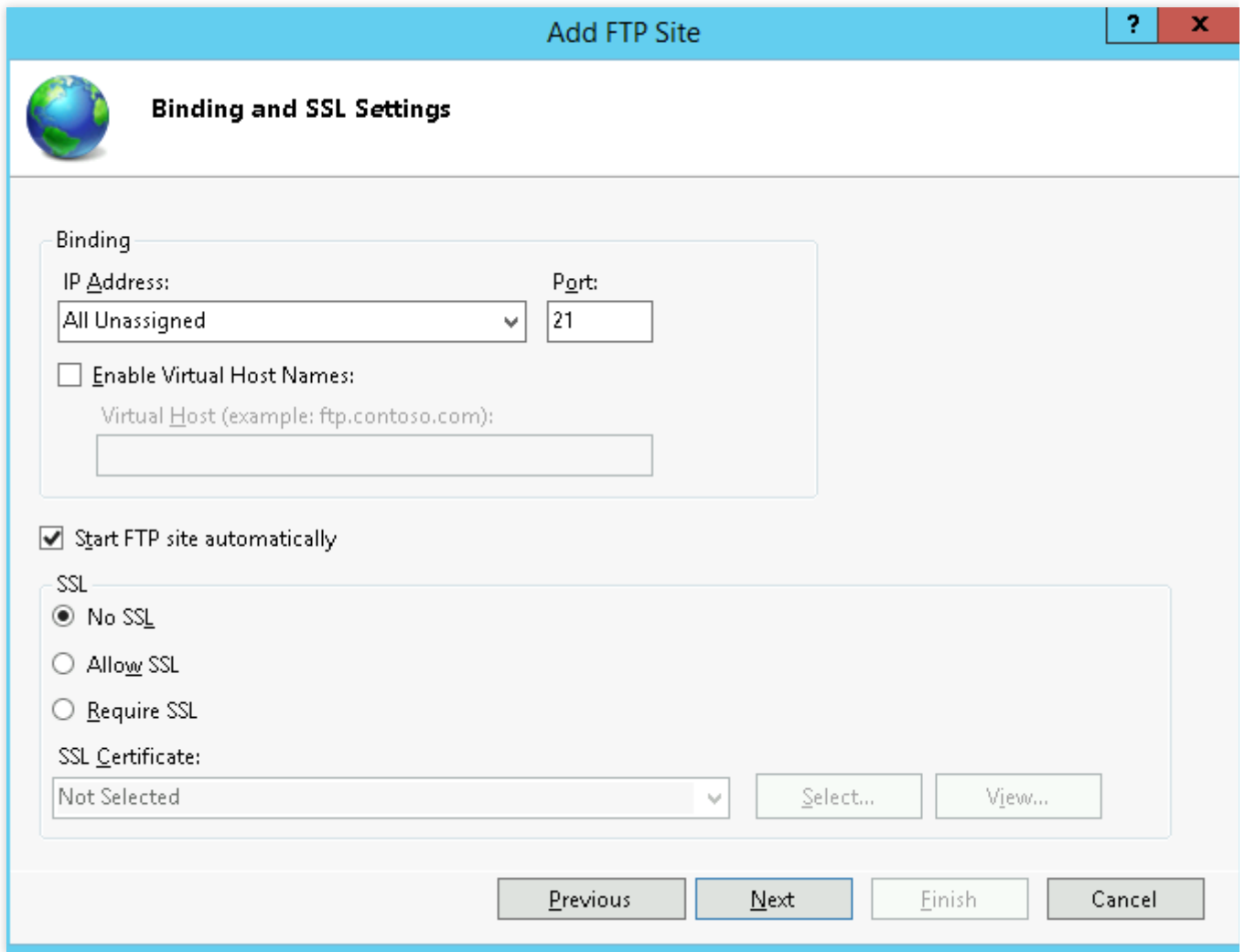
FTP site name:  
ftp

Content Directory  
Physical path:  
C:\test ...

**FTP site name:** Name of your FTP site. This document uses `ftp` as an example.

**Physical path:** Path of the shared folder with permissions configured. This document uses `C:\\test` as an example.

4. On the **Binding and SSL Settings** page, enter the following information and click **Next**.



**Add FTP Site**

**Binding and SSL Settings**

**Binding**

IP Address: All Unassigned Port: 21

☐ Enable Virtual Host Names:

Virtual Host (example: ftp.contoso.com):

☒ Start FTP site automatically

**SSL**

☒ No SSL

☐ Allow SSL

☐ Require SSL

SSL Certificate: Not Selected

Select... View...

Previous Next Finish Cancel

Configure the main parameters as follows:

**Binding:** The **IP Address** defaults to **All Unassigned**. The default FTP port number is 21. You can set a custom port number.

**SSL:** Select an option. In this document, **No SSL** is selected.

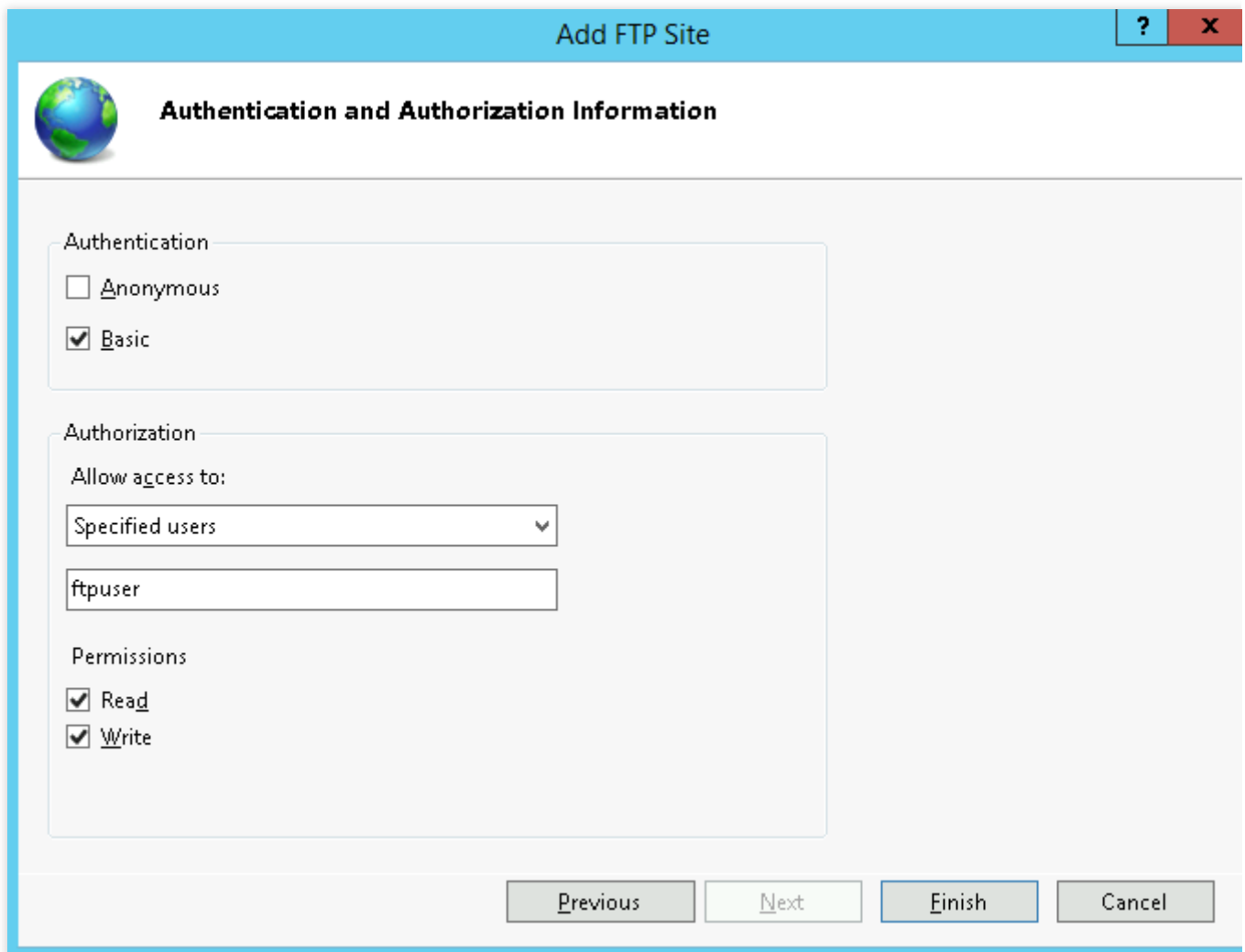
**No SSL:** No SSL is used.

**Allow SSL:** Allow the FTP server to connect with clients with or without SSL.

**Require SSL:** SSL encryption is required for communication between the FTP server and clients.

If you select **Allow SSL** or **Require SSL**, you can select an existing SSL certificate in **SSL Certificates** or [create an SSL certificate](#).

5. On the **Authentication and Authorization Information** page, enter the following information and click **Next** as shown below:



The screenshot shows a window titled "Add FTP Site" with a standard Windows-style title bar (blue with a question mark and close button). The main content area is titled "Authentication and Authorization Information" and features a globe icon. It contains three sections: "Authentication" with checkboxes for "Anonymous" (unchecked) and "Basic" (checked); "Authorization" with a dropdown menu set to "Specified users" and a text box containing "ftpuser"; and "Permissions" with checkboxes for "Read" (checked) and "Write" (checked). At the bottom, there are four buttons: "Previous", "Next", "Finish" (highlighted with a blue border), and "Cancel".

**Authentication:** Select an identity verification method. This document uses **Basic** as an example.

**Anonymous:** Allow users that provide the anonymous or FTP username to access the content.

**Basic:** Require users to provide valid user names and passwords to access the content. Under this mode, passwords are transmitted without encryption. Therefore, select this authentication mode only when you know that the connection between the clients and the FTP server is secure (for example, by using SSL).

**Authorization:** Select one of the following options from the **Allow access to** drop-down list. This document uses the specified `ftpuser` user as an example.

**All users:** All users, anonymous or identified, can access the content.

**Anonymous users:** Anonymous users can access the content.

**Specified role or user group:** Only the specified roles or members of the specified groups can access the content. If you choose this option, you need to specify the roles or user groups.

**Specified users:** Only the specified user can access the content. If you choose this option, you need to specify the username.

**Permissions:** Set permissions as needed. This document selects both **Read** and **Write** permissions.

**Read:** Allow the authorized user to read the shared content.

**Write:** Allow the authorized user to write into the directory.

6. Click **Finish** to successfully create the FTP site.

## Step 6. Configure the security group and firewall

1. After the FTP site is created, add an inbound rule that allows traffic to the FTP port based on the FTP access mode:

**Active mode:** Open ports 20 and 21.

**Passive mode:** Open the ports 21 and 1024–65535.

For more information on how to open ports, see [Adding firewall rule](#).

2. (Optional) See Microsoft documentation to configure firewall for the FTP site, so that the FTP server can accept passive connections from the firewall.

## Step 7. Test the FTP site

You can use tools such as the FTP client software, browser, or file manager to verify the FTP server. This document uses the file manager of the client as an example.

1. Configure Internet Explorer as needed:

Firewall has been configured (active mode):

Open Internet Explorer on the **Client**, select **Tools > Internet Options > Advanced**, deselect **Use Passive FTP (for firewall and DSL model compatibility)**, and click **OK**.

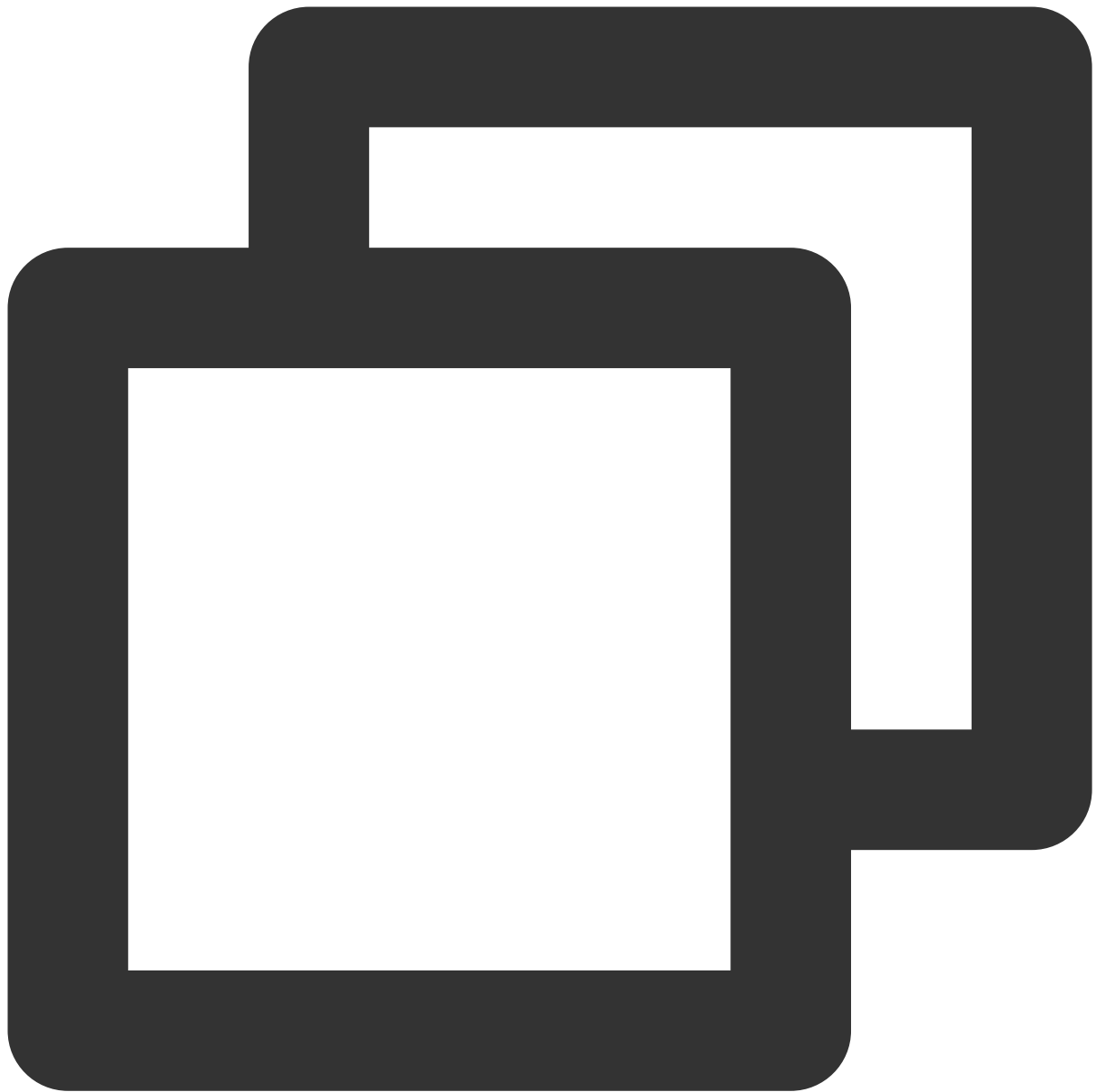
Firewall has not been configured (passive mode):

1.1.1 Open Internet Explorer on the **FTP Server**, select **Tools > Internet Options > Advanced**, deselect **Use Passive FTP (for firewall and DSL model compatibility)**, and click **OK**.

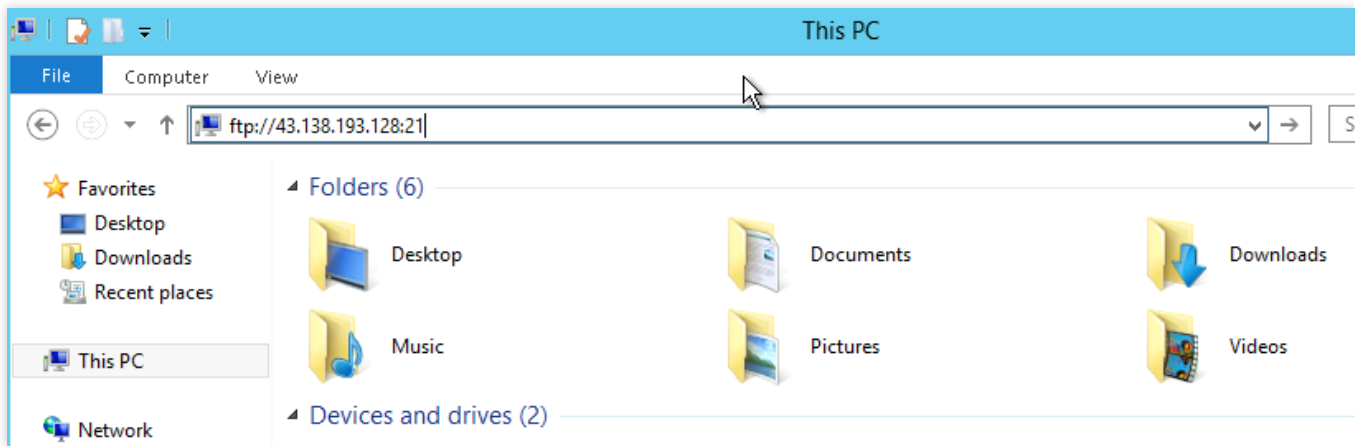
1.1.2 Open Internet Explorer on the **Client**, select **Tools > Internet Options > Advanced**, select **Use Passive FTP (for firewall and DSL model compatibility)**, and click **OK**.

2. Open the computer where the client is installed and enter the following path in the search box:





```
ftp://Lighthouse instance public IP:21
```

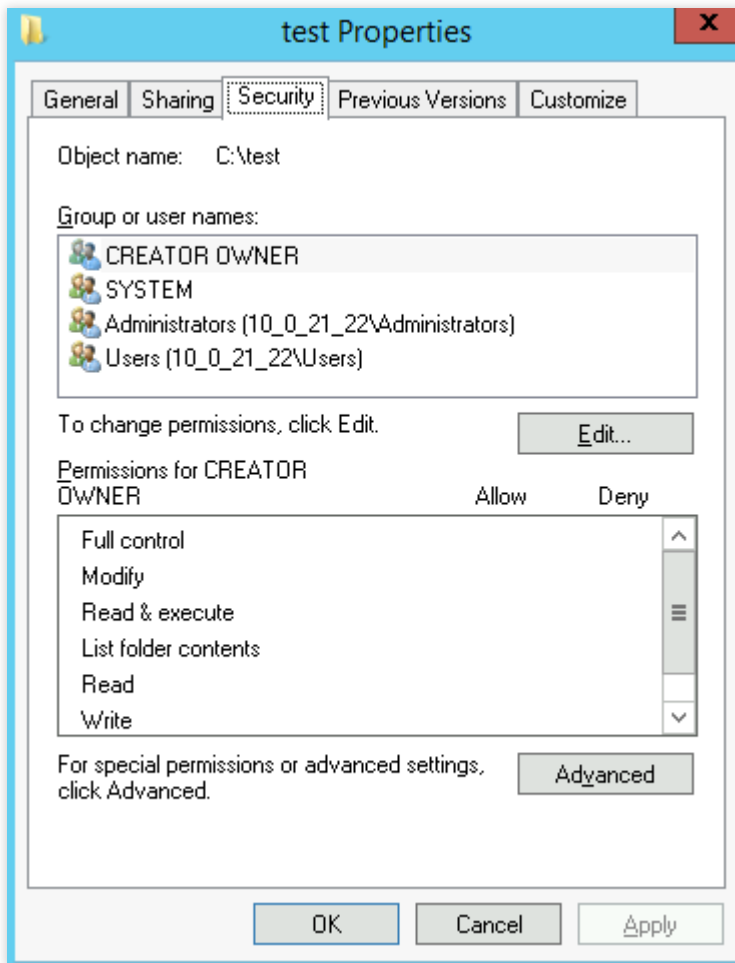


3. In the pop-up window, enter the username and password configured in [creating the FTP username and password](#).
4. You can upload and download files after a successful login.

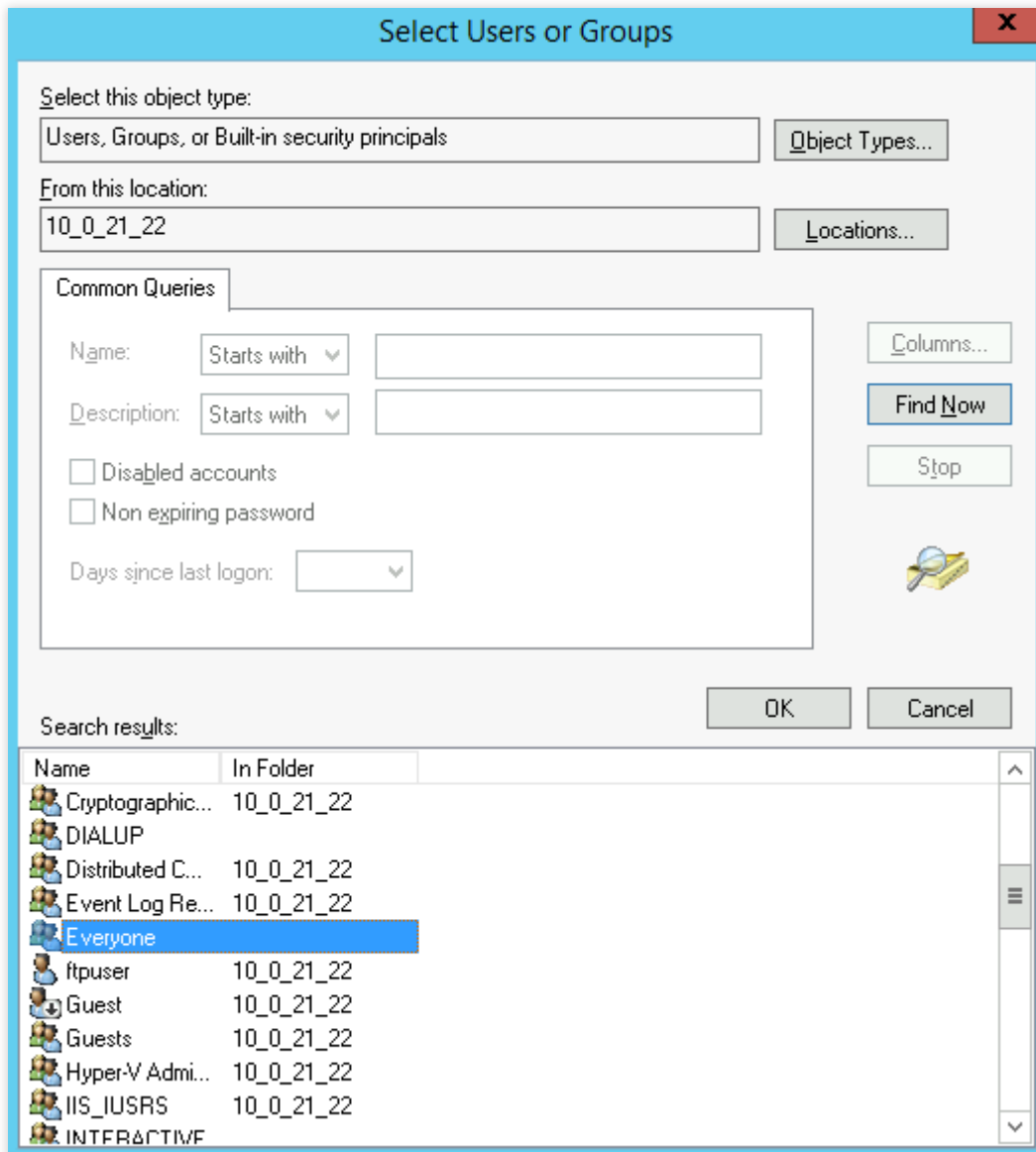
## Appendix

### Adding **Everyone** user

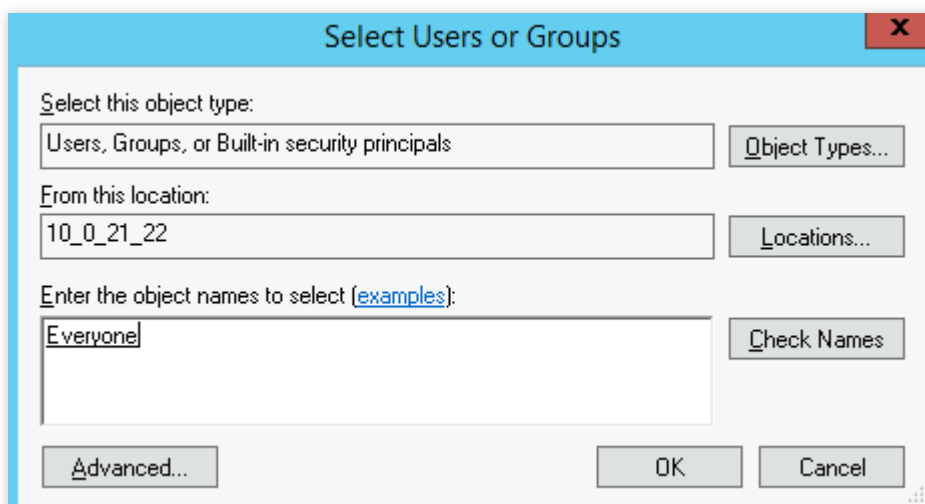
1. In the **test Properties** window, select the **Security** tab and click **Edit**.



2. On the **Permissions for test** page, click **Add**.
3. On the **Select Users or Groups** page, click **Advanced**.
4. In the pop-up window, click **Find Now**.
5. Select **Everyone** under **Search results** and click **OK**.



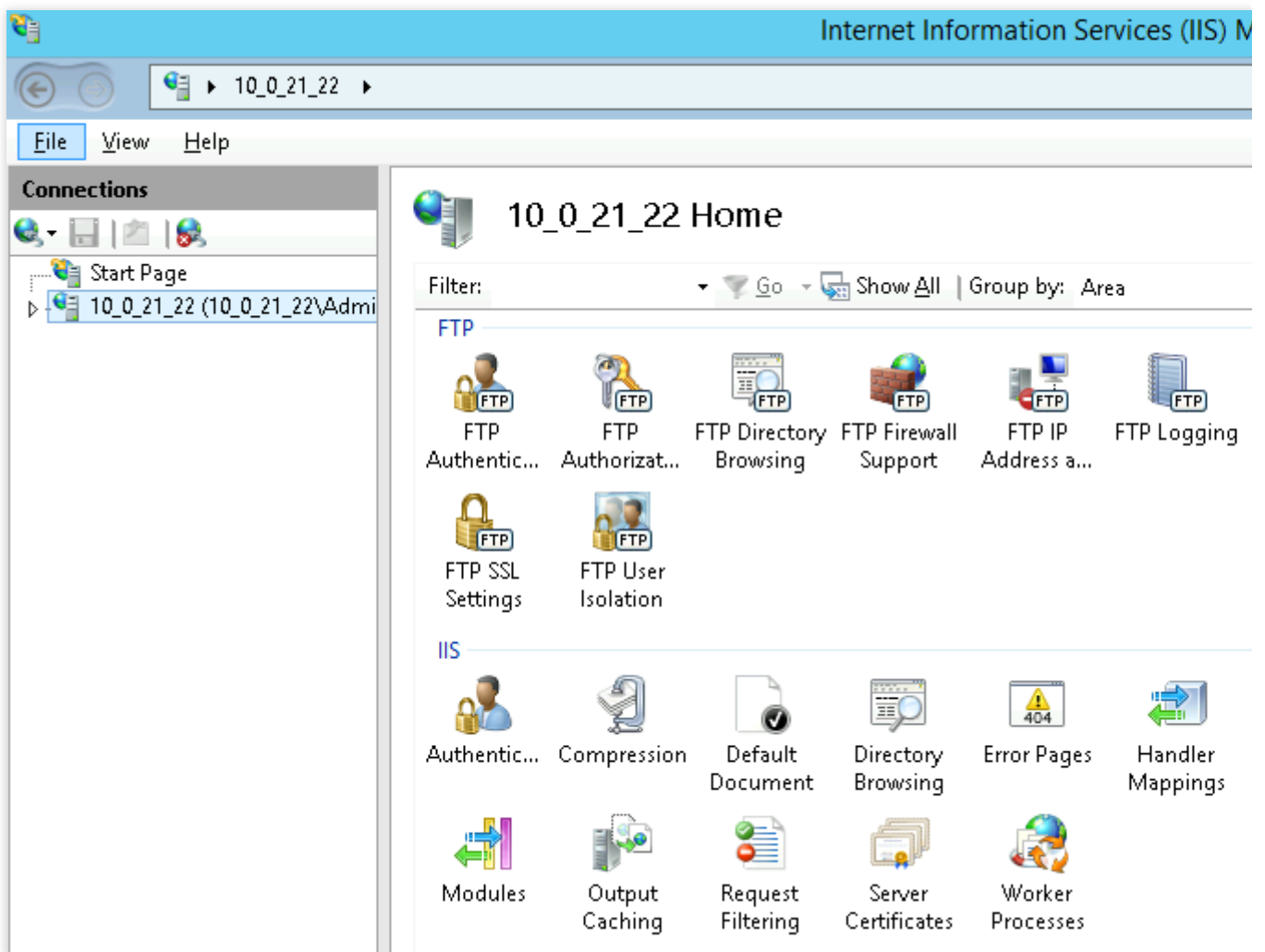
6. On the **Select Users or Groups** page, click **OK**.



Proceed to [step 5](#) to configure the permission for `Everyone` .


## Creating instance certificate

1. In the **Server Manager** window, select **Tools > Internet Information Services (IIS) Manager** on the top-right navigation bar.
2. In the **Internet Information Services (IIS) Manager** pop-up window, select the server on the left sidebar and double-click **Server Certificates** on the right panel.



3. Select **Create Self-Signed Certificate** in the right operation column.
4. In the **Create Self-Signed Certificate** pop-up window, enter the certificate name and storage type.  
This document uses creating an SSL certificate for personal storage as an example.

Create Self-Signed Certificate



Specify Friendly Name

Specify a file name for the certificate request. This information can be sent to a certificate authority for signing:

Specify a friendly name for the certificate:

Select a certificate store for the new certificate:

Personal

OK

Cancel

5. Click **OK**.

# Installing SSL Certificate

## Installing SSL Certificate

Last updated : 2022-06-15 16:05:02

An SSL certificate provides a complete HTTPS solution for authentication and encrypted data transfer for your website, mobile application, web API, and other applications. This document describes how to install an SSL certificate for your Lighthouse instance.

## Certificate Installation Methods

Select an appropriate SSL certificate installation method according to the type of your server.

Certificate Type	Lighthouse OS	Server Type
International standard certificate	Linux	<a href="#">Installing a Certificate - NGINX Server</a>
		<a href="#">Installing a Certificate - Apache Server</a>

For more information on other certificate installation methods, see the following documents:

[Installing a Certificate on IIS Servers](#)

[Installing an SSL Certificate \(JKS Format\) on a Tomcat Server](#)

## See Also

[Overview of Tencent Cloud SSL Certificate Service](#)

[Pricing of Tencent Cloud SSL Certificate Service](#)

# Installing Certificate on NGINX Server

Last updated : 2022-06-15 16:05:02

## Overview

This document describes how to install an SSL certificate in a Lighthouse instance and enable HTTPS access, with a WordPress 5.7.1-based instance as an example. NGINX software programs have been preinstalled in the instance by default.

### Note:

The SSL certificate used in the document is provided by Tencent Cloud. For more information on this service, see [Overview](#) and [Purchase Guide](#).

## Prerequisites

Install the remote file copy tool such as WinSCP.

Install the remote login tool such as PuTTY or Xshell.

Open port 443 in your firewall policy. For more information, see [Managing Firewall](#).

The data required to install the SSL certificate includes the following:

Name	Description
Lighthouse instance's public IP address	Instance IP address used to connect a local computer to the instance.
Username	The username used to log in to the Lighthouse instance, such as `root`.
Password or SSH key	The password matching the username used to log in to the Lighthouse instance, or the bound SSH key.

### Note:

To get the public IP of the instance, you can log in to the [Lighthouse console](#), find the target instance, and enter its details page to view its public IP address. After the instance is created, first reset the password and remember it, or bind an SSH key and save the private key file. For more information, see [Resetting Password](#) and [Managing Keys](#).

## Directions

### Installing the certificate



1. Log in to the [SSL Certificates Service console](#), download and decompress the SSL certificate file (with the name `cloud.tencent.com` as an example here) to a local directory.

After decompression, you can get the certificate files in the relevant types, including Nginx folders and CSR files:

**Folder name:** Nginx

**Files in the folder:**

`cloud.tencent.com_bundle.crt` : Certificate file

`cloud.tencent.com.key` : Private key file

**CSR file:** `cloud.tencent.com.csr` file

**Note:**

You can upload the CSR file when applying for a certificate or have it generated online by the system. It is provided to the CA and irrelevant to the installation.

2. Use a remote login tool (such as WinSCP) on the local computer to log in to the Lighthouse instance with the username and password or SSH key pair. For more information, see [Logging in to Linux Instance via Remote Login Software](#).

3. Copy the obtained `cloud.tencent.com_bundle.crt` and `cloud.tencent.com.key` files from the local directory to NGINX's default configuration file directory of the Lighthouse instance. For more information, see [Uploading Local Files to Lighthouse](#).

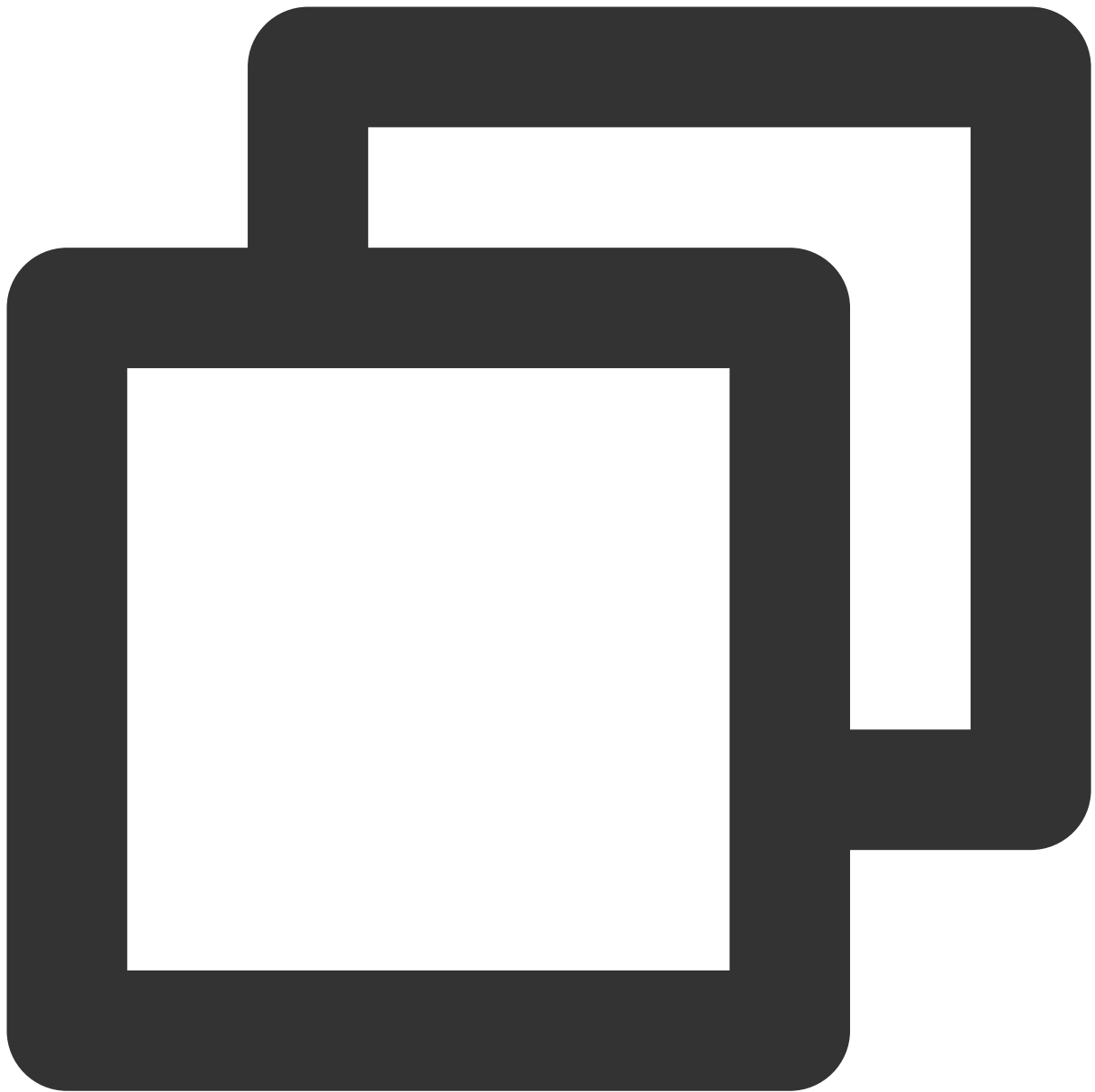
**Note:**

The default configuration file directory of the WordPress image is `/www/server/nginx/conf`.

- 4.

For instances created with the WordPress image

, run the following command to edit the `nginx.conf` file in NGINX's default configuration file directory.

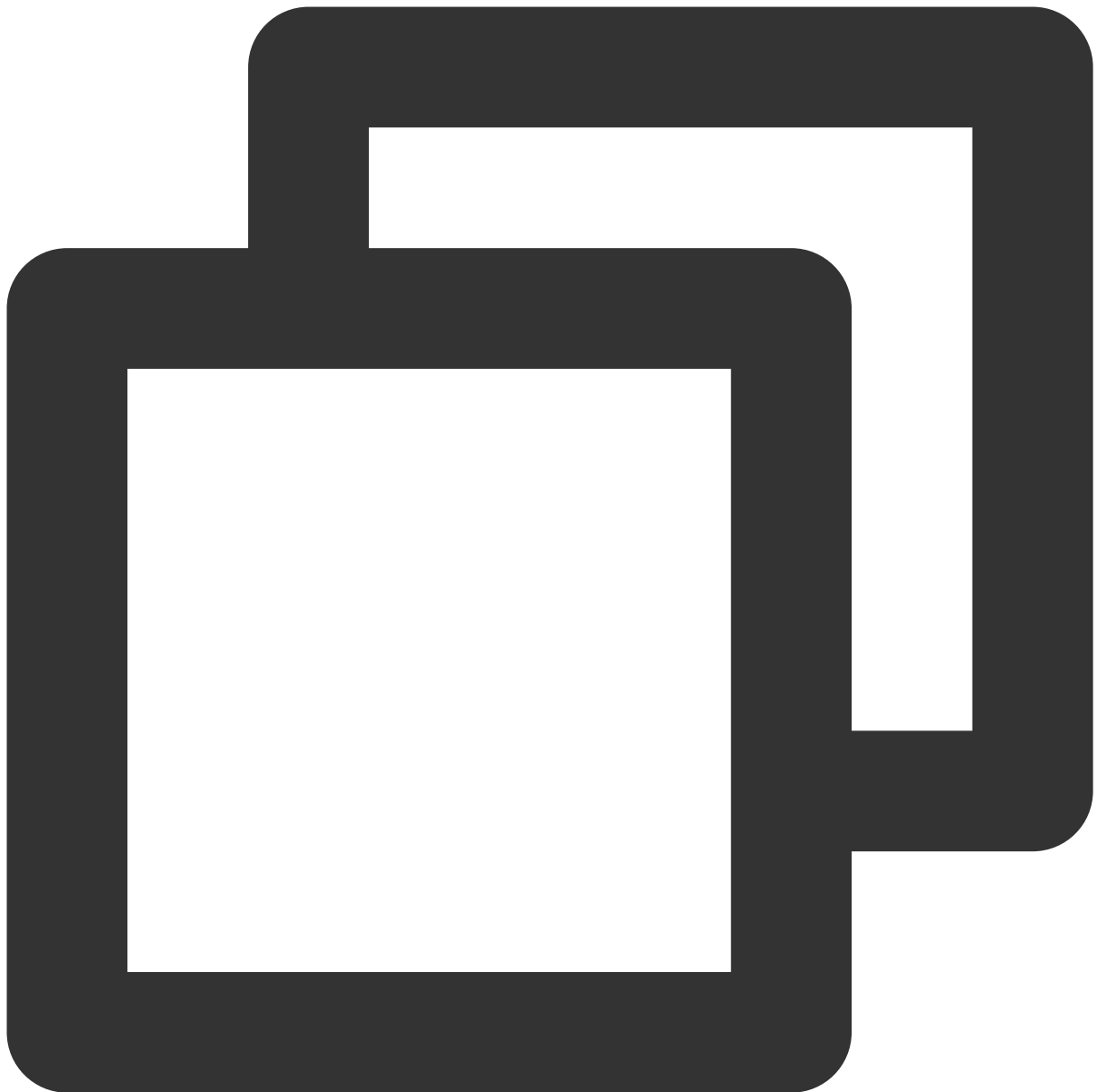


```
sudo vim /www/server/nginx/conf/nginx.conf
```

Find `server {...}` and replace the configuration information inside the braces ({} ) with the following content.

**Note:**

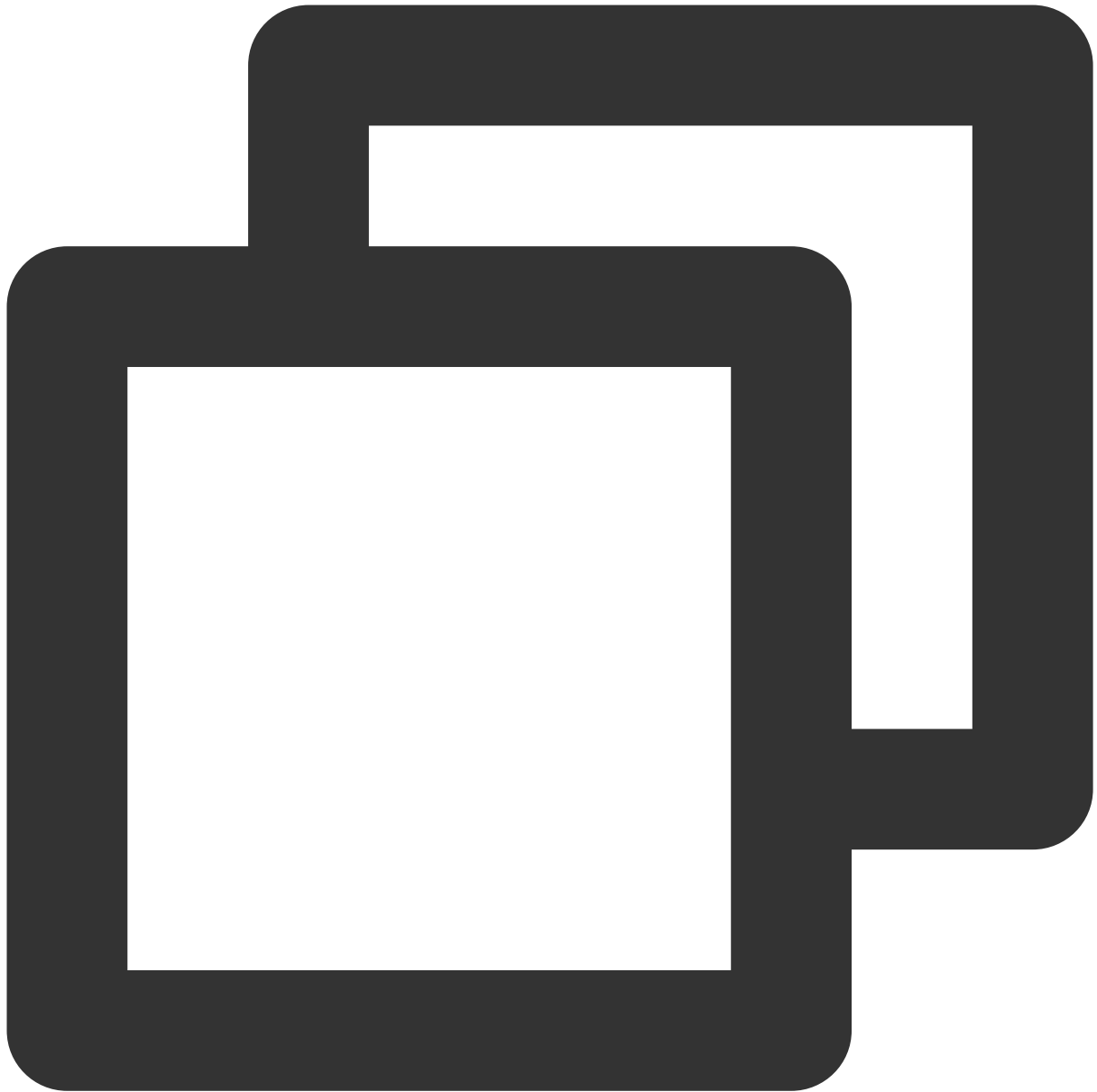
This configuration is for reference only. You can modify it as needed according to the comments or NGINX documentation based on your actual environment.



```
server {  
    listen 443 ssl;  
    server_tokens off;  
    keepalive_timeout 5;  
    root /usr/local/lighthouse/softwares/wordpress; # Enter the root directory of y  
    index index.php index.html;  
    access_log logs/wordpress.log;  
    error_log logs/wordpress.error.log;  
    server_name cloud.tencent.com; # Enter the domain name bound to your certificat  
    ssl_certificate cloud.tencent.com_bundle.crt; # Enter the name of your certific  
    ssl_certificate_key cloud.tencent.com.key; # Enter the name of your private key
```

```
ssl_session_timeout 5m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # You can see this SSL protocol for confi
ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE; # You can
ssl_prefer_server_ciphers on;
location ~* /\.php$ {
    fastcgi_pass 127.0.0.1:9000;
    include fastcgi.conf;
    client_max_body_size 20m;
    fastcgi_connect_timeout 30s;
    fastcgi_send_timeout 30s;
    fastcgi_read_timeout 30s;
    fastcgi_intercept_errors on;
}
}
```

5. Find `http{...}` and enter the following configuration information.

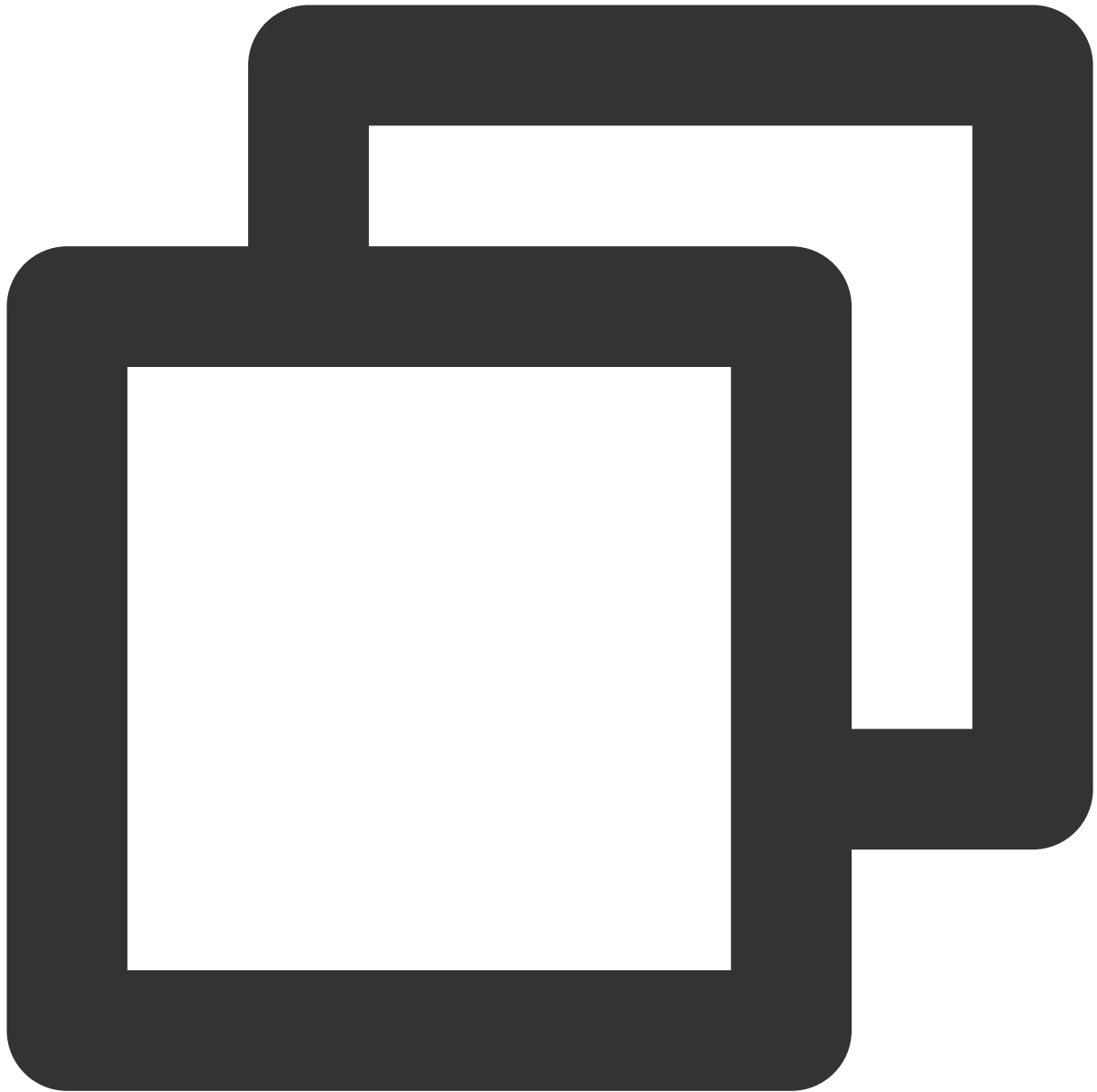


```
ssl_certificate cloud.tencent.com_bundle.crt;    # Enter the name of your certificat
ssl_certificate_key cloud.tencent.com.key;       # Enter the name of your private key
```

6. Save the modified `nginx.conf` file and exit.

7.

Run the following command to verify that there is no problem with the configuration file.



```
sudo nginx -t
```

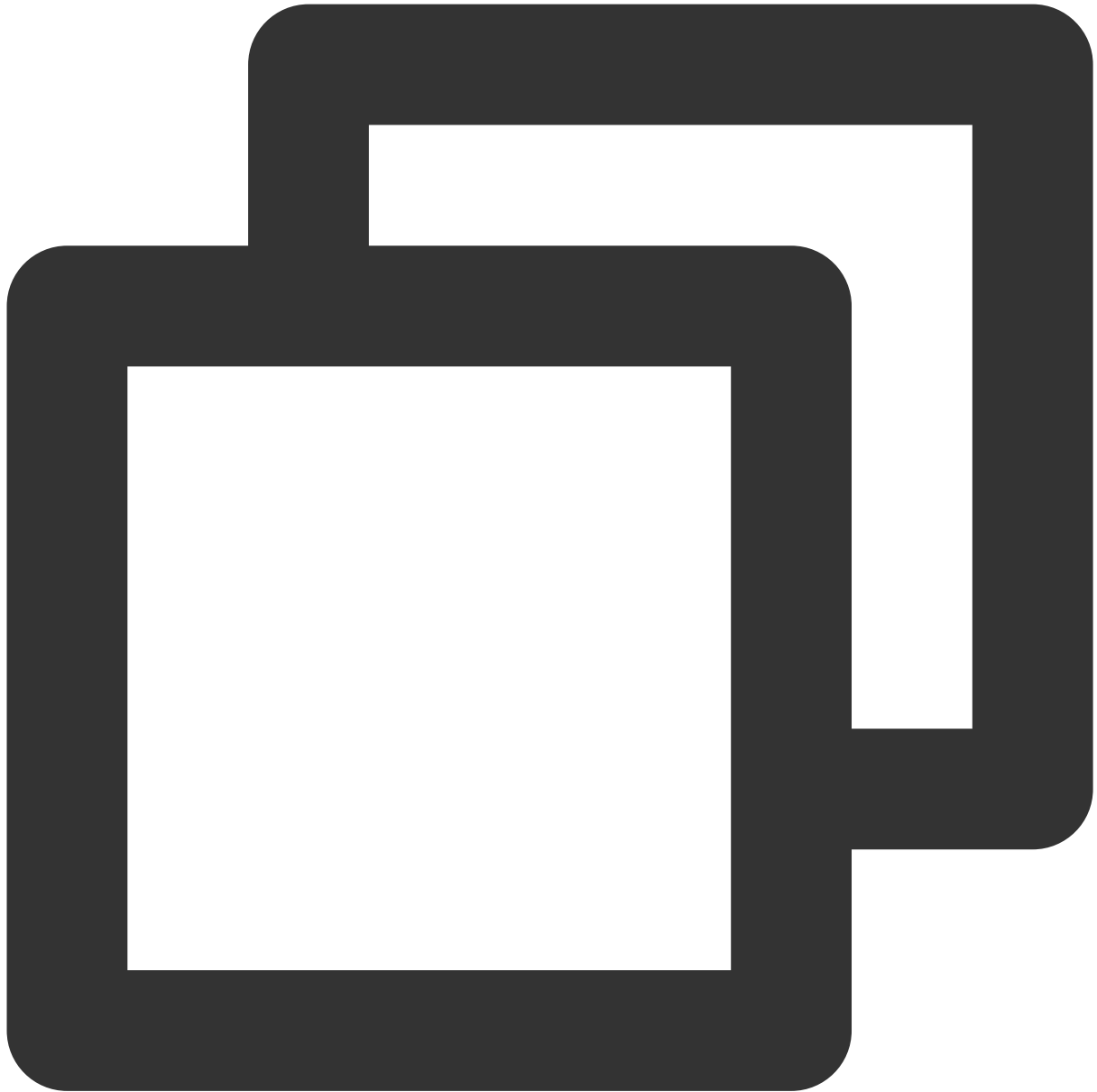
If the following output information is displayed, the configuration is successful. Proceed to [step 8](#).

```
[lighthouse@VM-8-12-centos ~]$ sudo nginx -t
nginx: the configuration file /www/server/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /www/server/nginx/conf/nginx.conf test is successful
```

If there is an error message, reconfigure or fix the problem as prompted.

8.

Run the following command to restart NGINX.



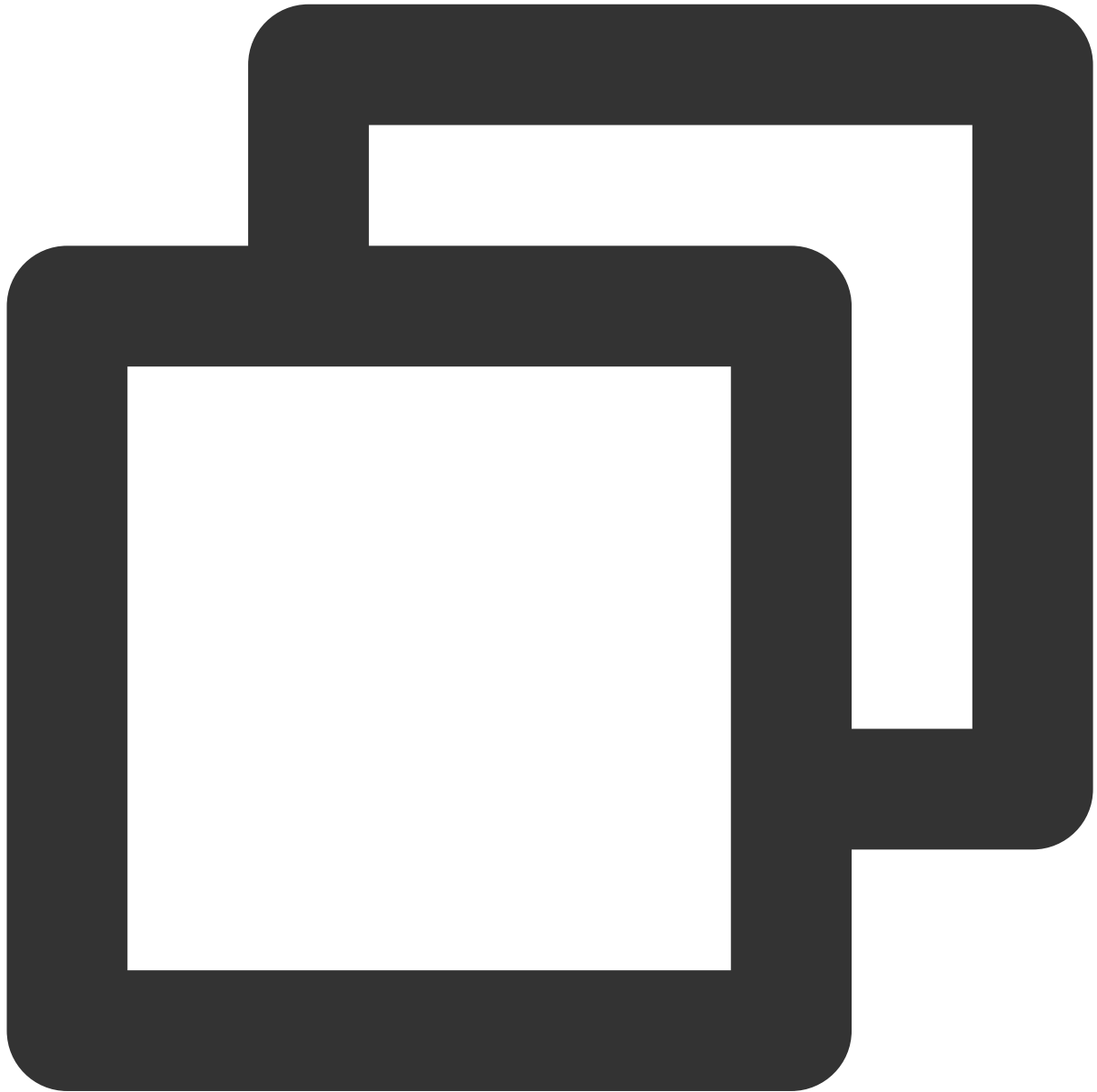
```
sudo systemctl reload nginx
```

At this point, the installation is successful. You can use `https://cloud.tencent.com` (sample) for access.

### **(Optional) Setting automatic redirect of HTTP request to HTTPS**

You can configure the instance to automatically redirect HTTP requests to HTTPS in the following steps:

1. NGINX supports rewrite. If you did not remove `pcre` during compilation, you can add `return 301 https://$host$request_uri;` to the HTTP server to redirect requests made to the default port 80 to HTTPS. You need to modify the `nginx.conf` file by adding the following configuration after [Step 4](#) in the **Installing the certificate** section.



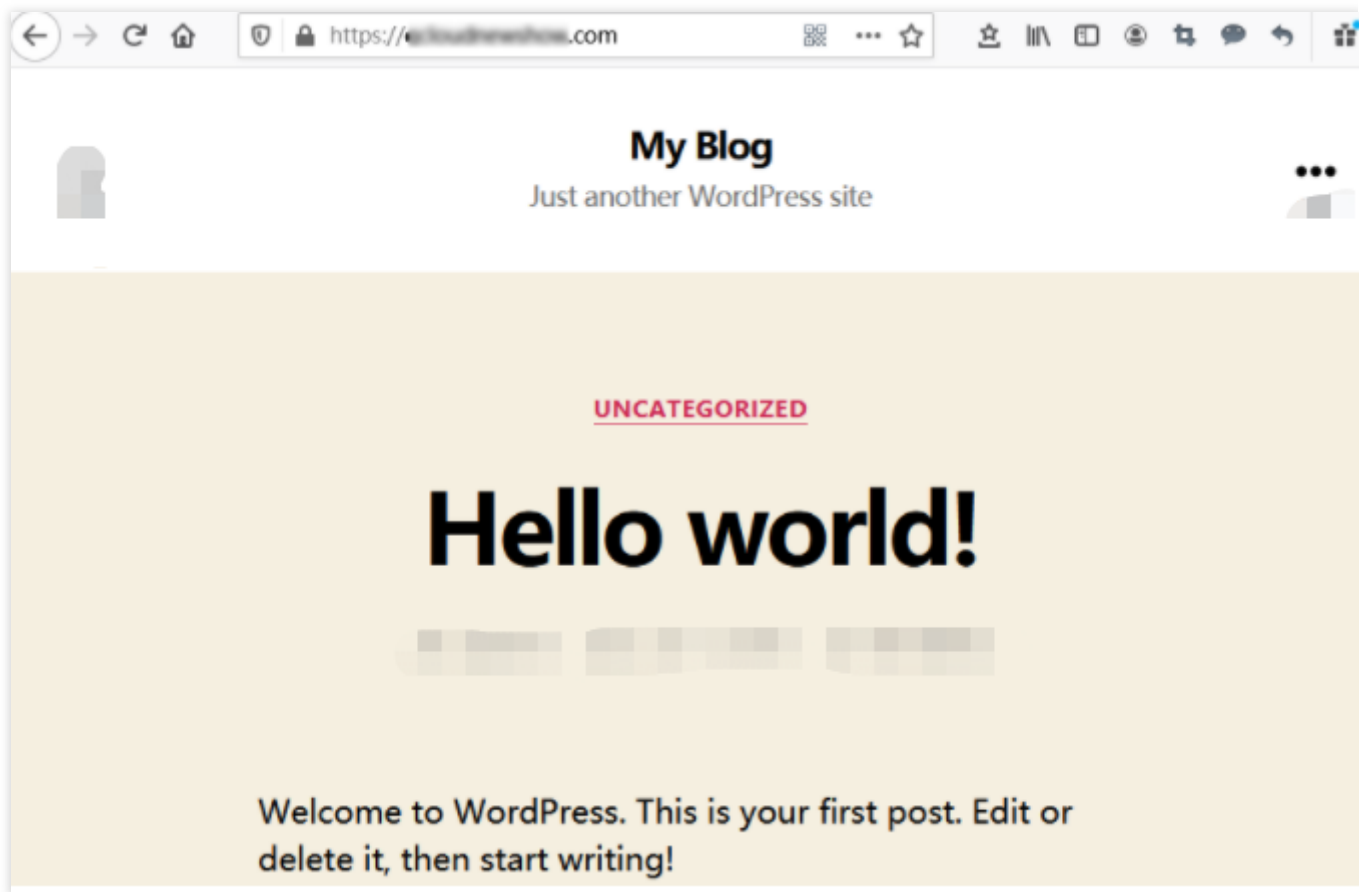
```
server {  
    listen 80;  
    server_name cloud.tencent.com;    # Enter the domain name bound to your certificat  
    return 301 https://$host$request_uri;    # Redirect HTTP requests to HTTPS  
}
```



2. Save the modified `nginx.conf` file and exit. Verify and restart NGINX according to [Step 7](#) and [Step 8](#) in the **Installing the certificate** section.

At this point, you have successfully set the automatic redirect to HTTPS. You can use

`http://cloud.tencent.com` (sample) to redirect to the HTTPS page as shown below:



# Installing Certificate on Apache Server (Linux)

Last updated : 2024-03-20 14:38:45

## Overview

This document describes how to install an SSL certificate in a Lighthouse instance and enable HTTPS access. The example instance uses an LAMP application image with Apache software pre-installed.

### Note:

The SSL certificate used in the document is provided by Tencent Cloud. For more information on this service, see [Overview](#) and [Purchase Guide](#).

## Preparation

Install the remote file copy tool such as WinSCP. The latest official version is recommended.

Install the remote login tool such as PuTTY or Xshell. The latest official version is recommended.

Open port 443 in your firewall policy. For more information, see [Managing Firewall](#).

The data required to install the SSL certificate includes the following:

Name	Description
Lighthouse instance's public IP address	Instance IP address used to connect a local computer to the instance.
Username	The username used to log in to the Lighthouse instance, such as `root`.
Password or SSH key	The password matching the username used to log in to the Lighthouse instance, or the bound SSH key.

### Note:

You can log in to the [Lighthouse console](#), find the target instance, and enter its details page to view its public IP address. After the instance is created, first reset the password and remember it, or bind an SSH key and save the private key file. For more information, see [Resetting Password](#) and [Managing Keys](#).

## Directions

### Installing certificate

1. Log in to the [SSL Certificates Service console](#), download and decompress the SSL certificate file (with the name `cloud.tencent.com` as an example here) to a local directory.

After decompression, you can get the relevant certificate files, including the Apache folder and CSR file:

**Folder name:** Apache

**Files in the folder:**

`1_root_bundle.crt` : Certificate file

`2_cloud.tencent.com.crt` : Certificate file

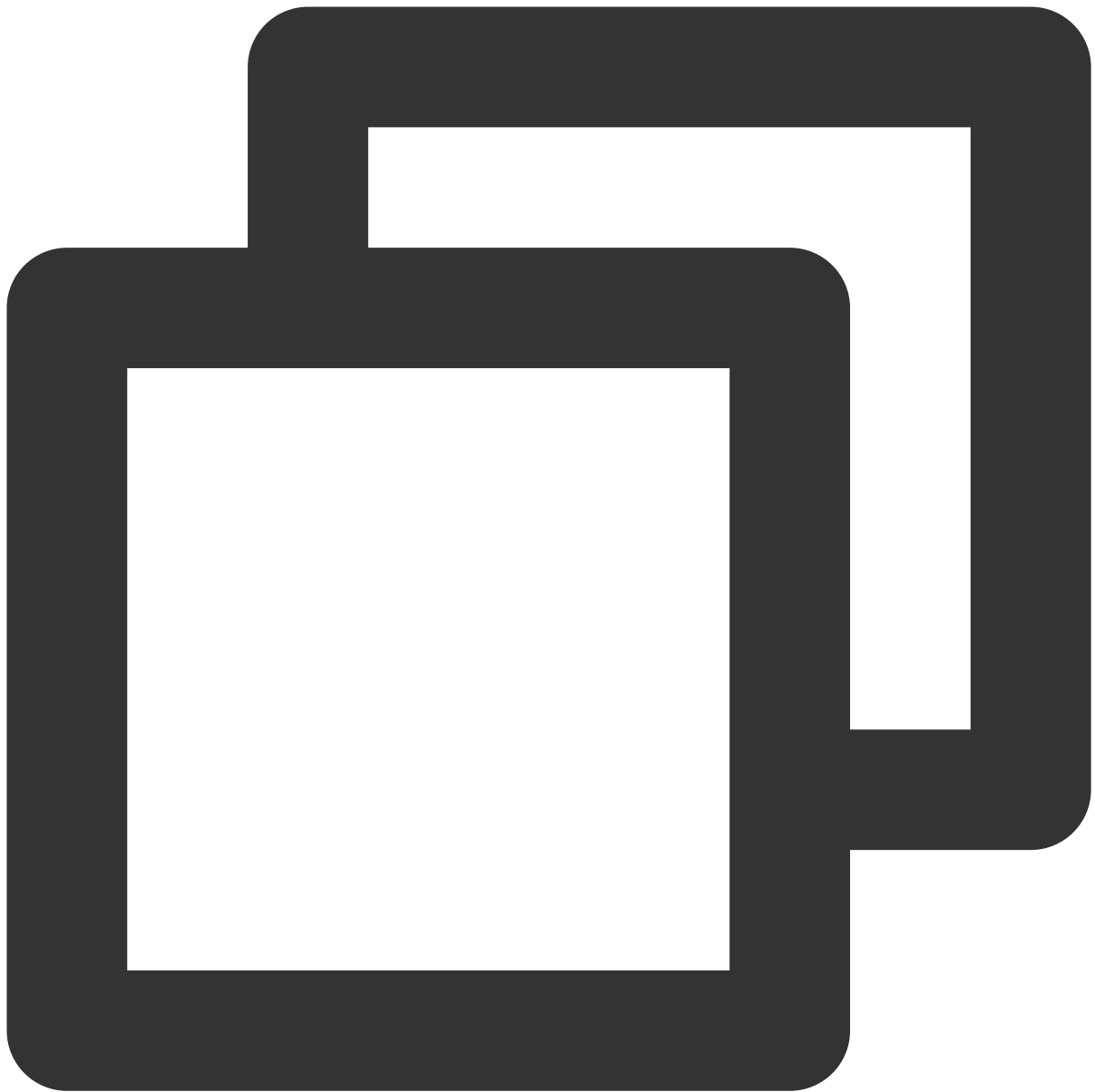
`3_cloud.tencent.com.key` : Private key file

**CSR file:** `cloud.tencent.com.csr` file

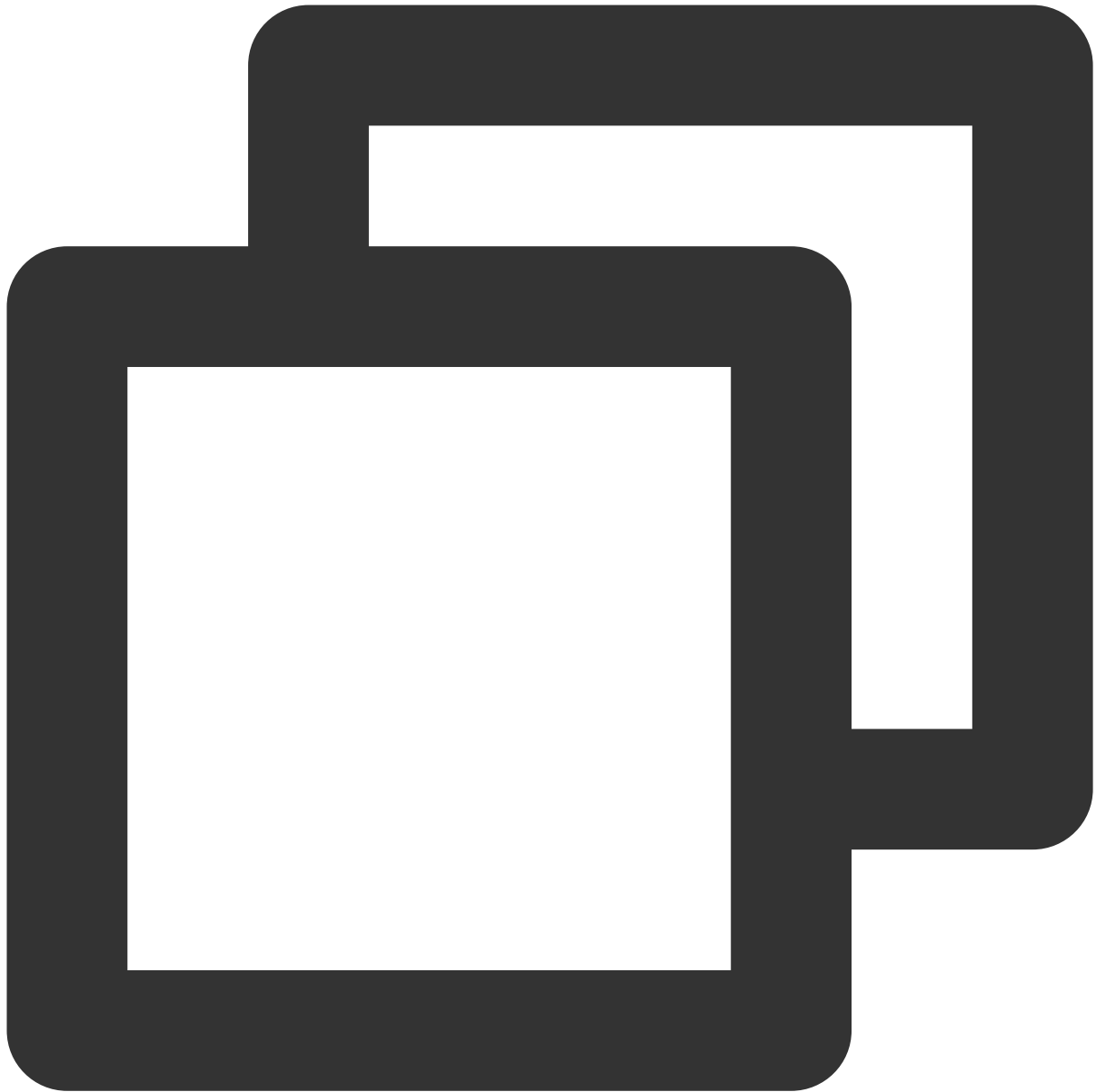
**Note:**

You can upload the CSR file when applying for a certificate or have it generated online by the system. It is provided to the CA and irrelevant to the installation.

2. Log in to the Lighthouse instance. See [Logging In to Linux Instance via WebShell](#).
3. Run the following commands in sequence to enter the Apache installation directory and create the `ssl` folder.

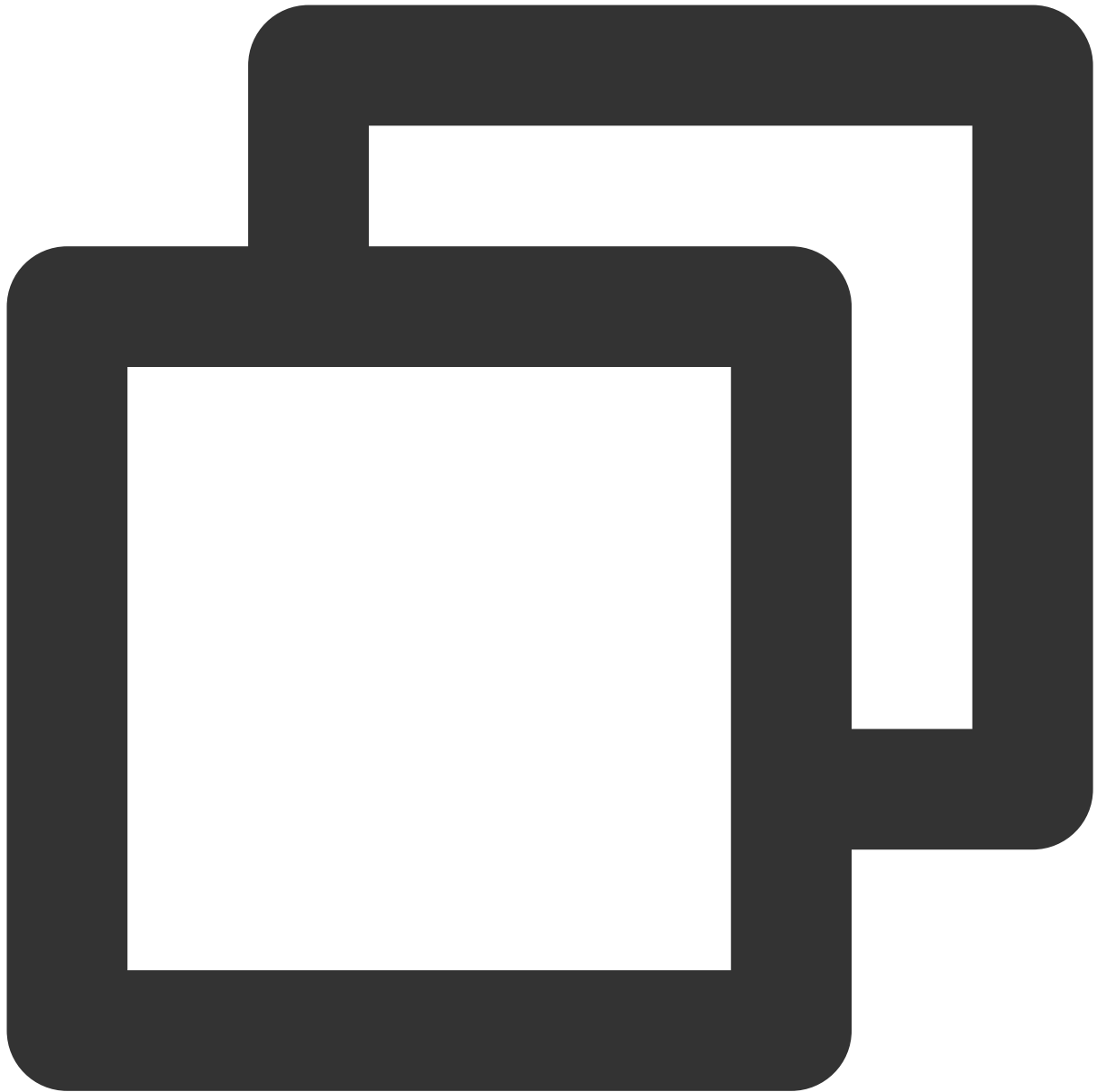


```
cd /usr/local/lighthouse/softwares/apache
```



```
sudo mkdir ssl
```

4. Copy the obtained `1_root_bundle.crt` , `2_cloud.tencent.com.crt` , and `3_cloud.tencent.com.key` files from the local directory to the created `/usr/local/lighthouse/softwares/apache/ssl` directory. For more information, see [Uploading Local Files to Lighthouse](#).
5. Run the following command to edit the `httpd.conf` configuration file.



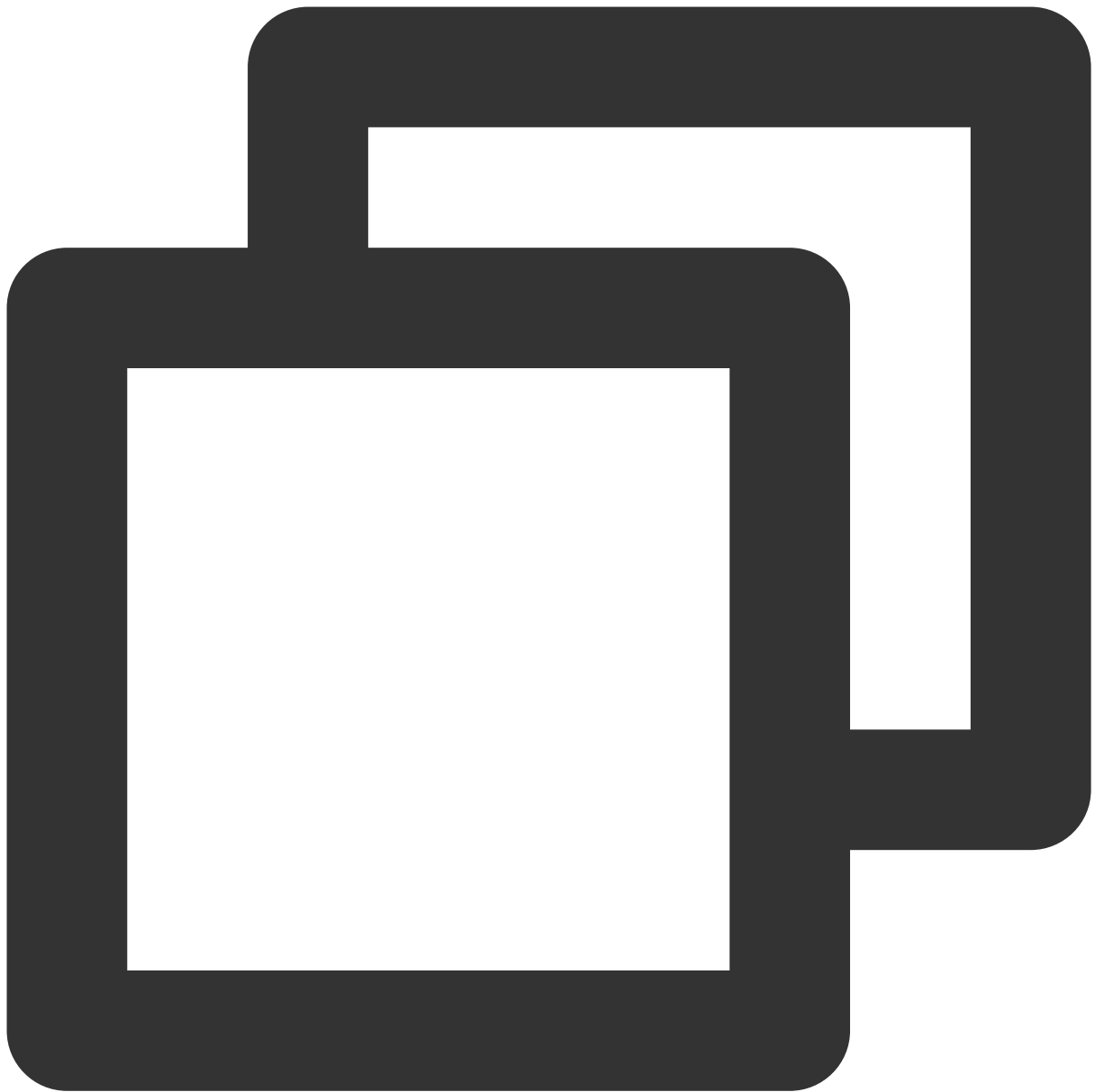
```
sudo vim /usr/local/lighthouse/software/apache/conf/httpd.conf
```

6. Press **i** to enter the edit mode and make the following changes:

6.1 Delete the `#` in `#LoadModule ssl_module modules/mod_ssl.so` .

6.2 Delete the `#` in `#LoadModule socache_shmcb_module modules/mod_socache_shmcb.so` .

6.3 Replace `localhost` in `ServerName localhost` with the certificate name. A modified sample is as shown below:

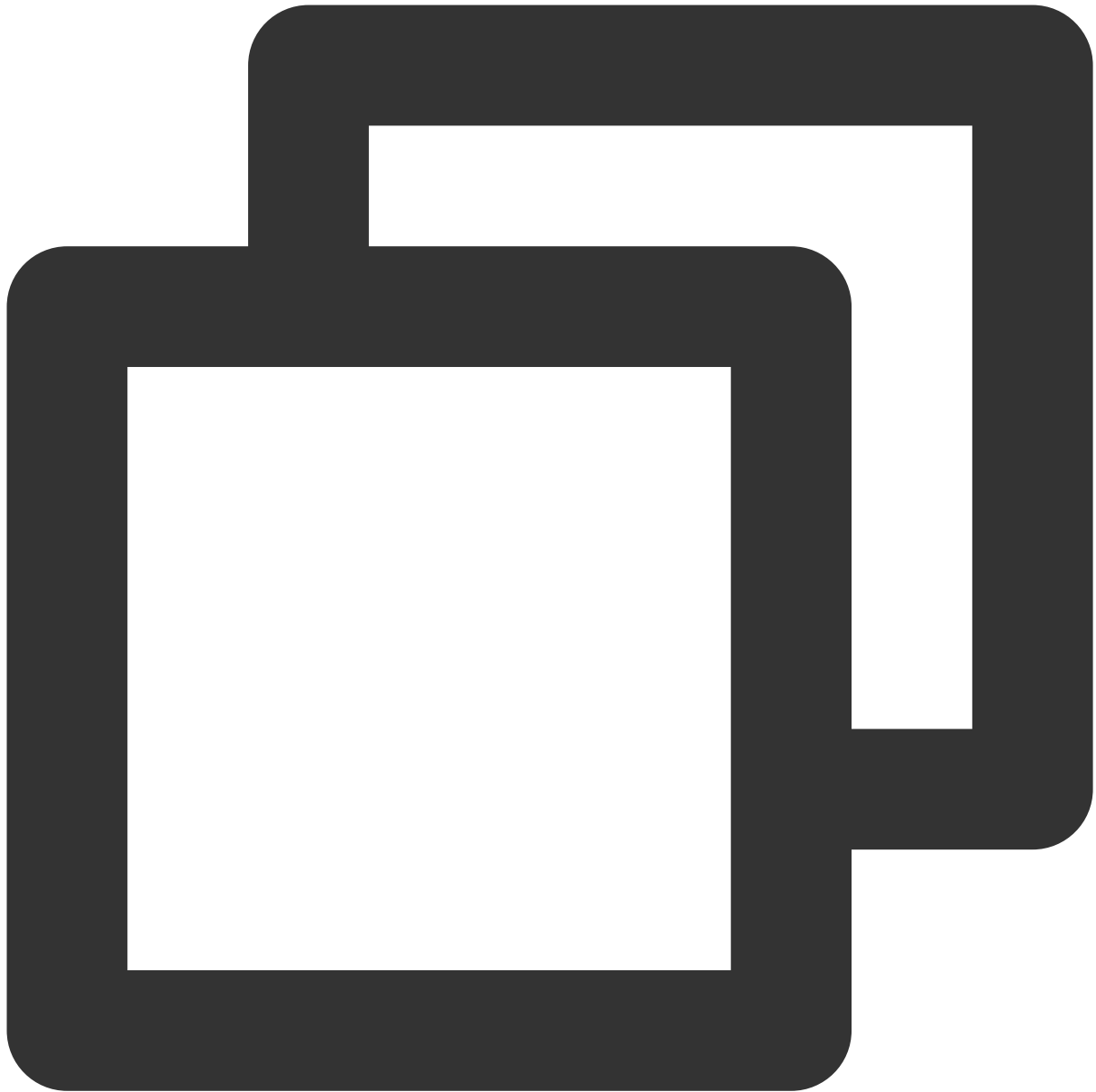


```
ServerName cloud.tencent.com
```

6.4 Delete the `#` in `#Include conf/extra/httpd-ssl.conf` .

7. Press **Esc** and enter `:wq` to save the changes.

8. Run the following command to modify the `httpd-ssl.conf` configuration file.

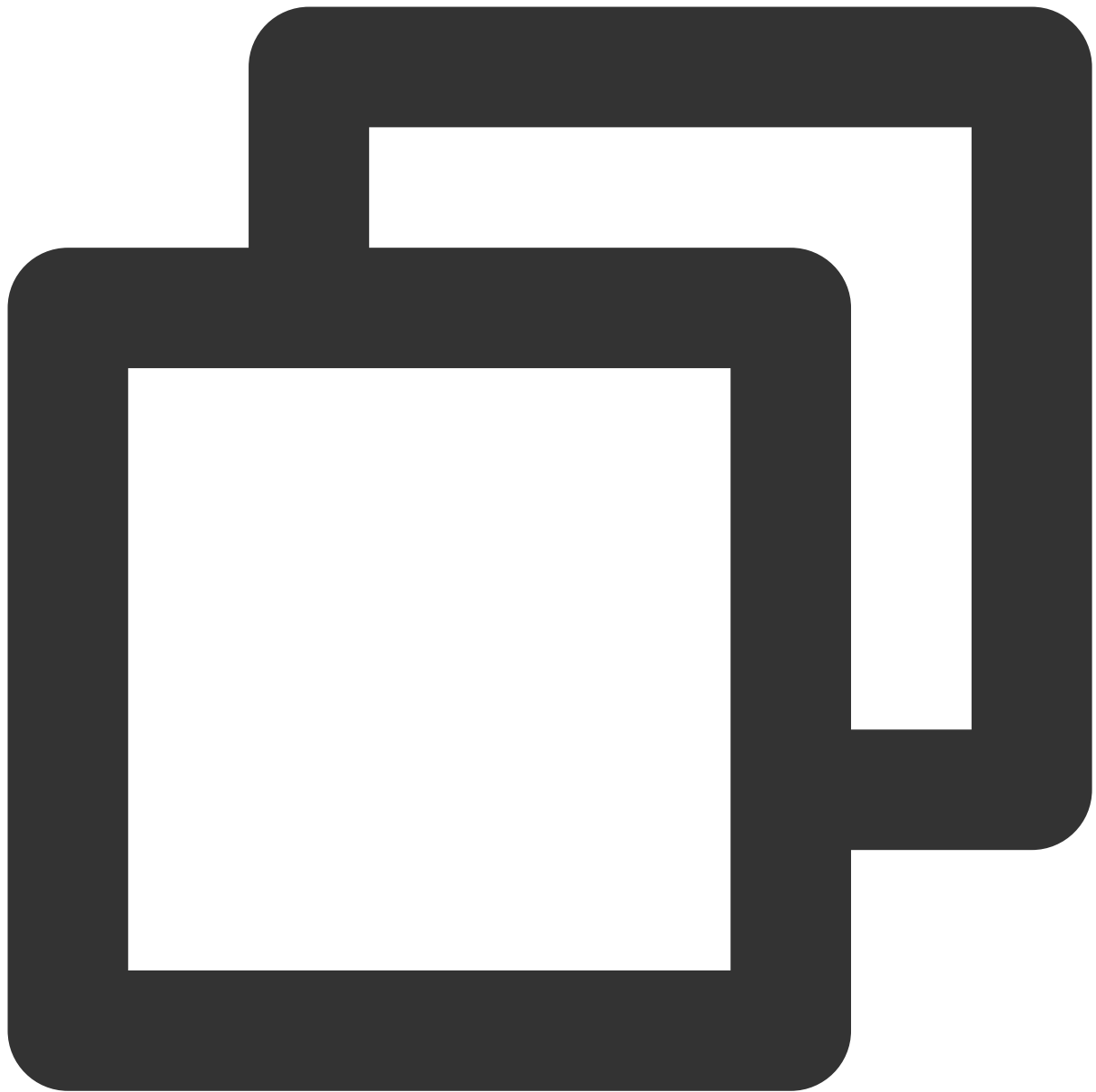


```
sudo vim /usr/local/lighthouse/softwares/apache/conf/extra/httpd-ssl.conf
```

9. Press **i** to enter the edit mode and make the following changes in `<VirtualHost _default_:443>` :

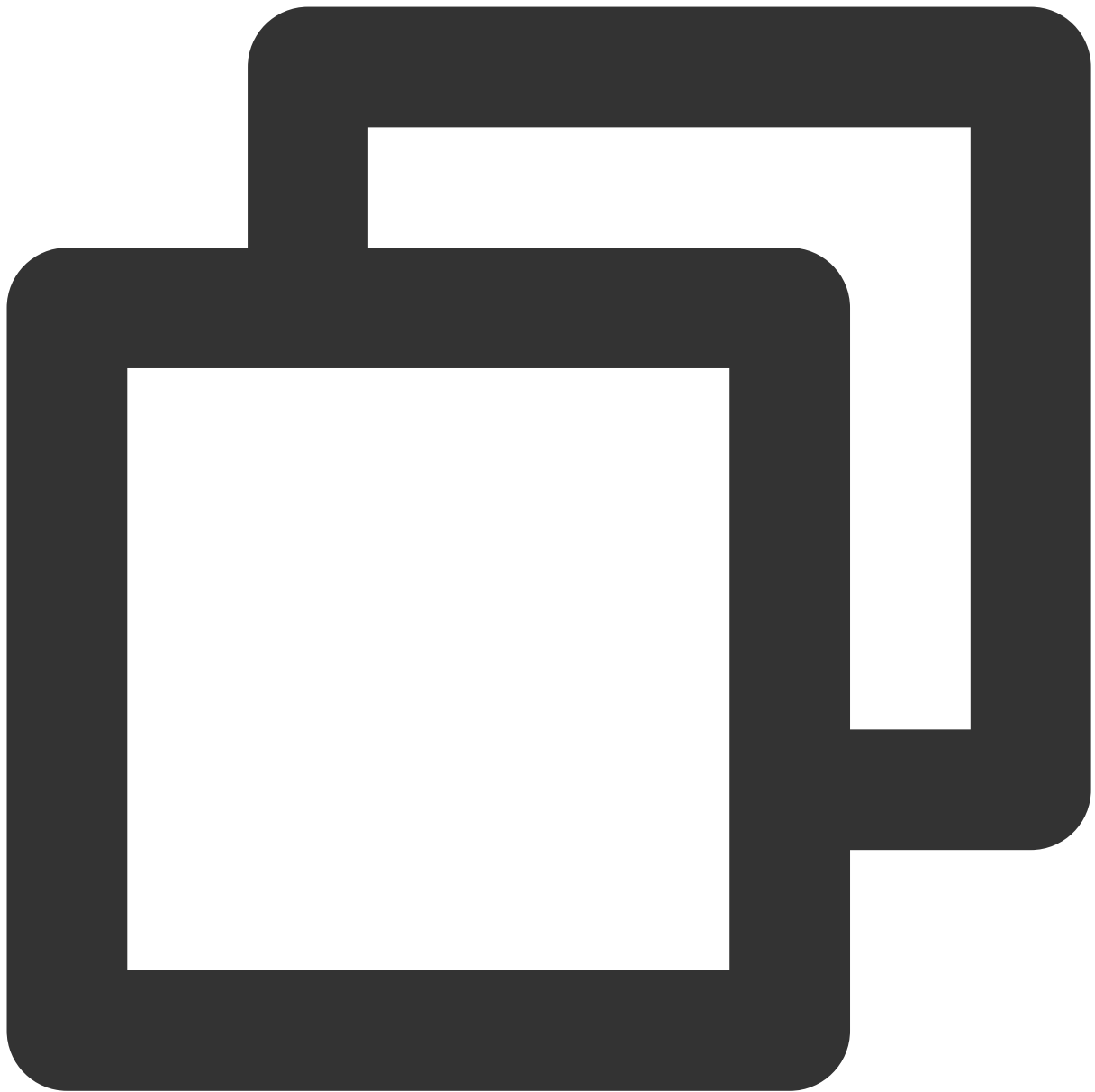
9.1 Replace `www.example.com:443` in `ServerName www.example.com:443` with the certificate name. A modified sample is as shown below:





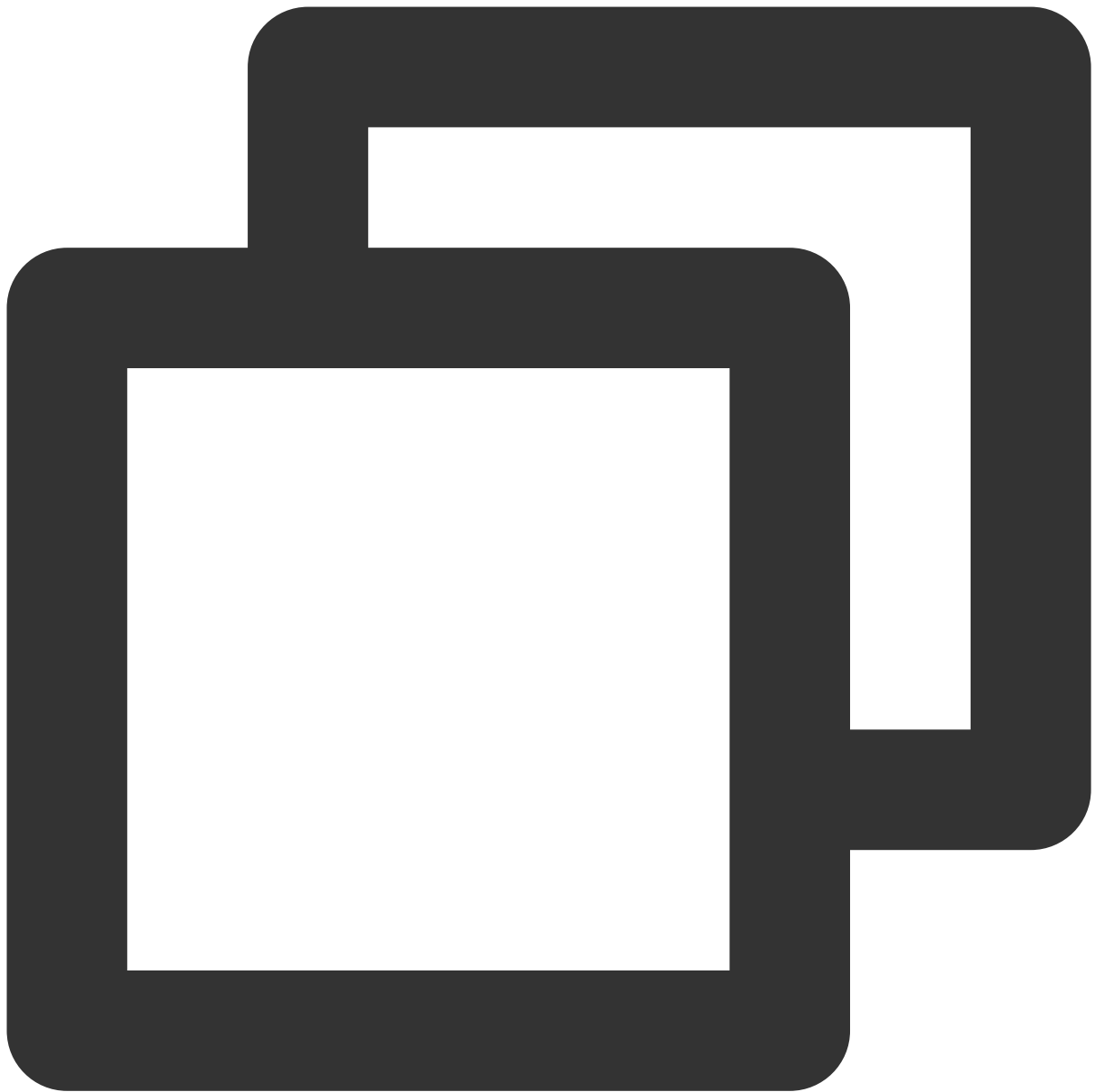
```
ServerName cloud.tencent.com
```

9.2 Modify the paths of the certificate files:



```
SSLCertificateFile "/usr/local/lighthouse/softwares/apache/ssl/2_cloud.tencent.com.  
SSLCertificateKeyFile "/usr/local/lighthouse/softwares/apache/ssl/3_cloud.tencent.c  
SSLCertificateChainFile "/usr/local/lighthouse/softwares/apache/ssl/1_root_bundle.c
```

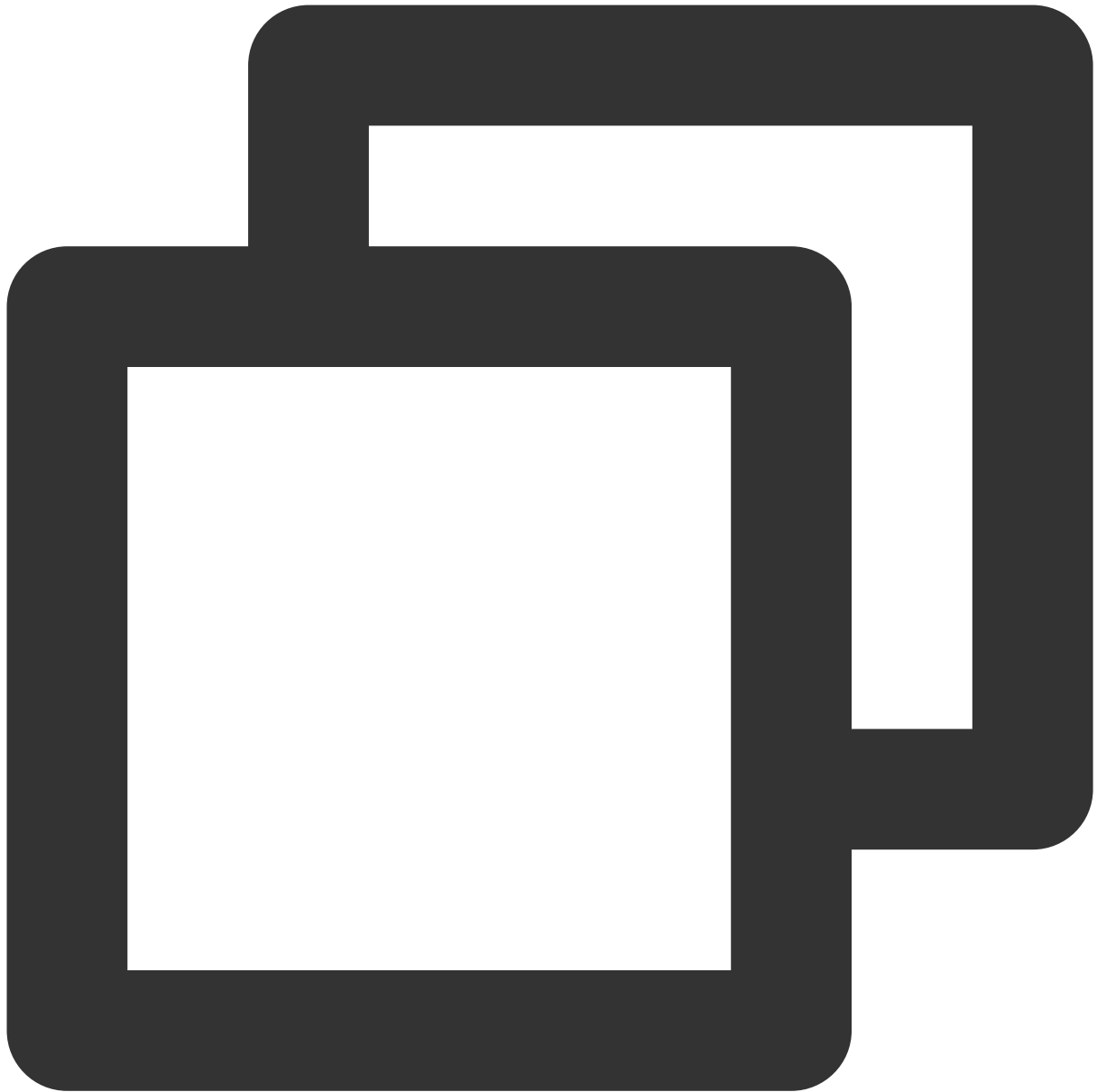
### 9.3 Add the following content:



```
<Directory "/usr/local/lighthouse/softwares/apache/htdocs">  
    Options Indexes FollowSymLinks  
    AllowOverride all  
    Require all granted  
</Directory>
```

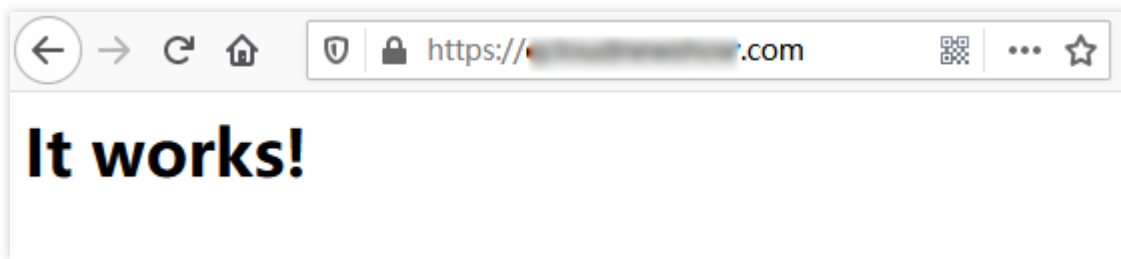
10. Press **Esc** and enter **:wq** to save the changes.

11. Run the following command to restart the Apache service.



```
sudo /usr/local/lighthouse/softwares/apache/bin/httpd -k restart
```

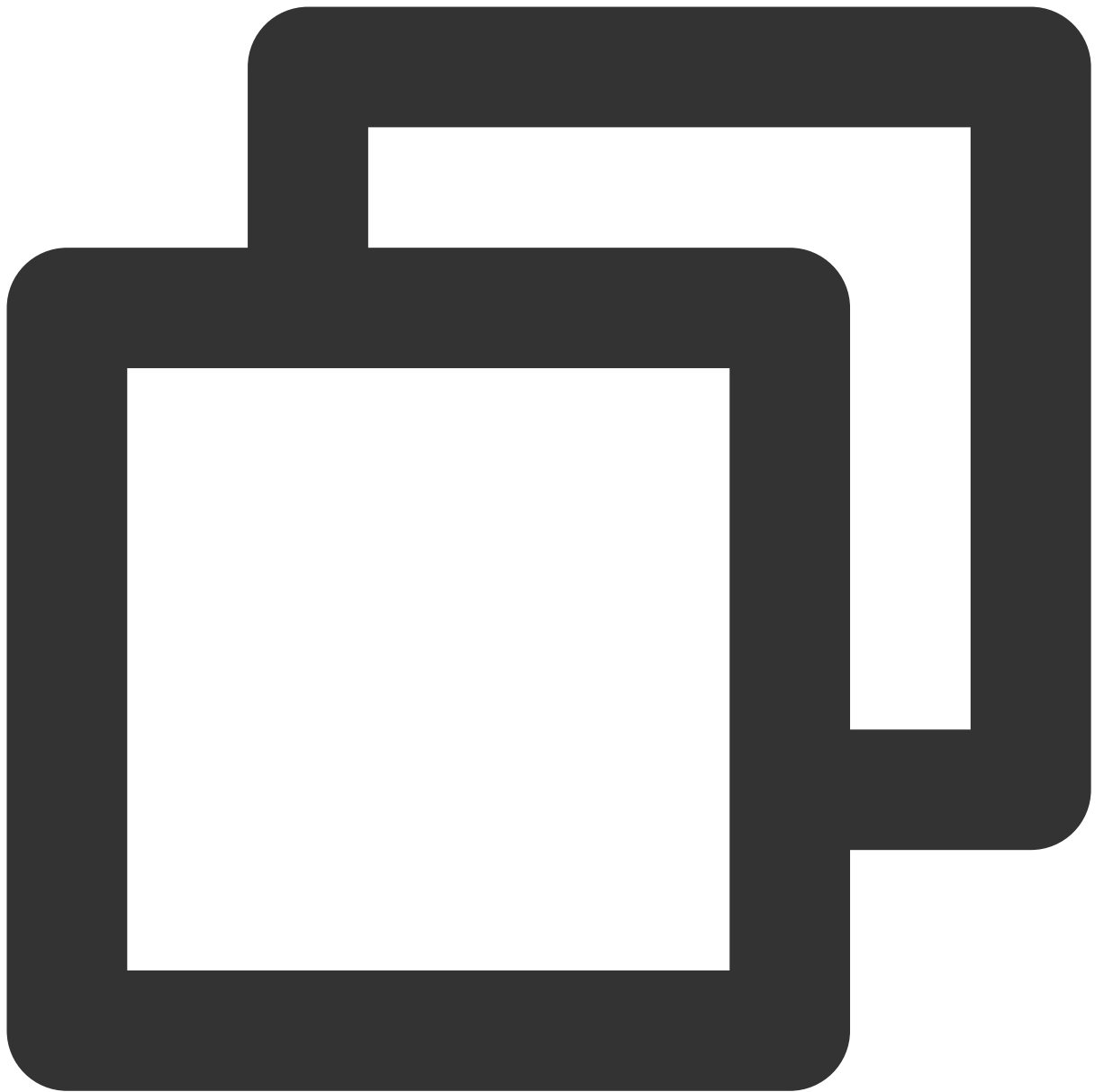
After the successful restart, you can use <https://cloud.tencent.com> for access as shown below:



### (Optional) Setting automatic redirect of HTTP request to HTTPS

You can configure the instance to automatically redirect HTTP requests to HTTPS in the following steps:

1. Run the following command to edit the `httpd.conf` configuration file .

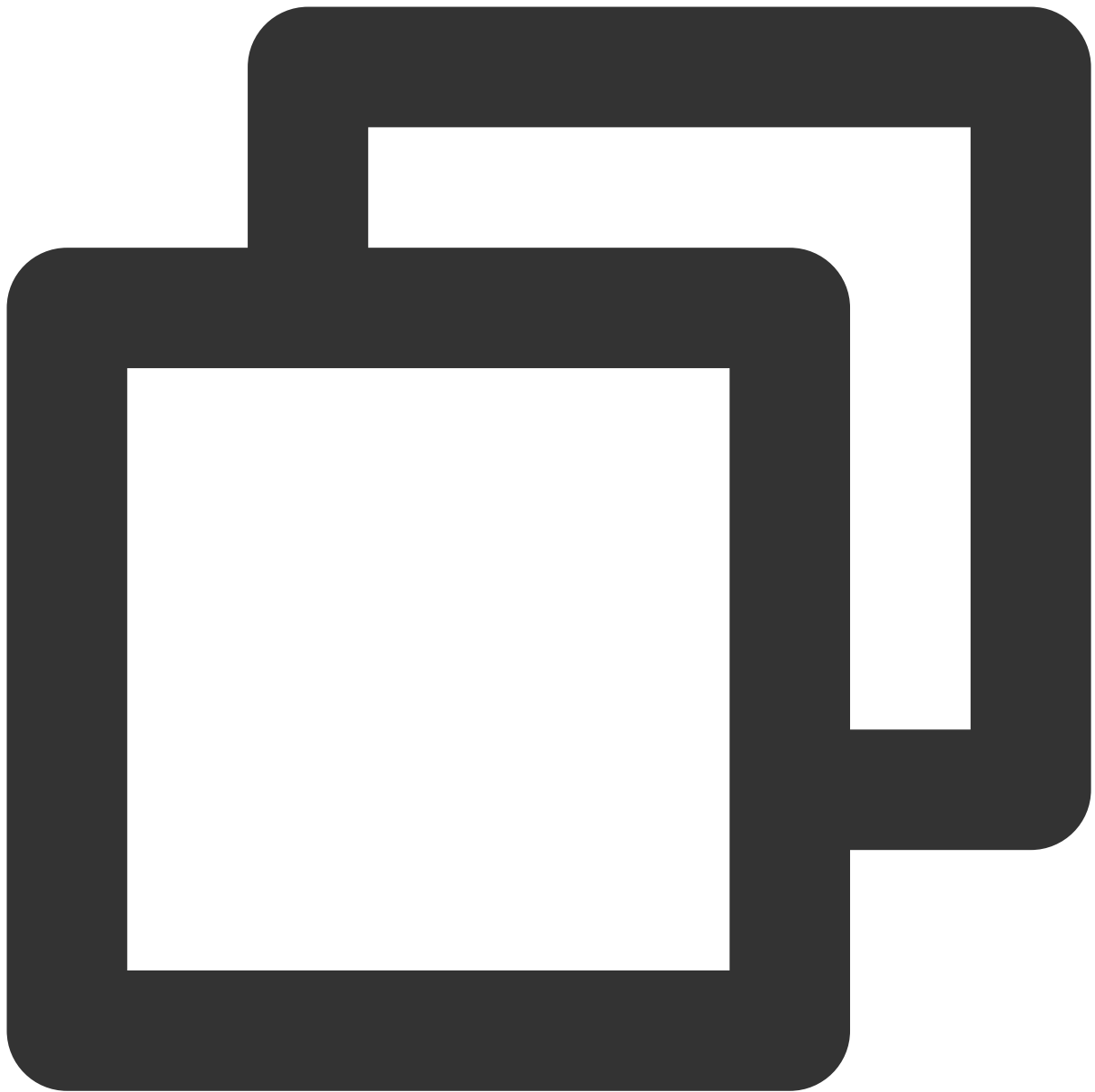


```
sudo vim /usr/local/lighthouse/softwares/apache/conf/httpd.conf
```

2. Press **i** to enter the edit mode and make the following changes:

2.1 Delete the `#` in `#LoadModule rewrite_module modules/mod_rewrite.so` .

2.2 Find `<Directory "/home/www/htdocs/">` and add the following content:

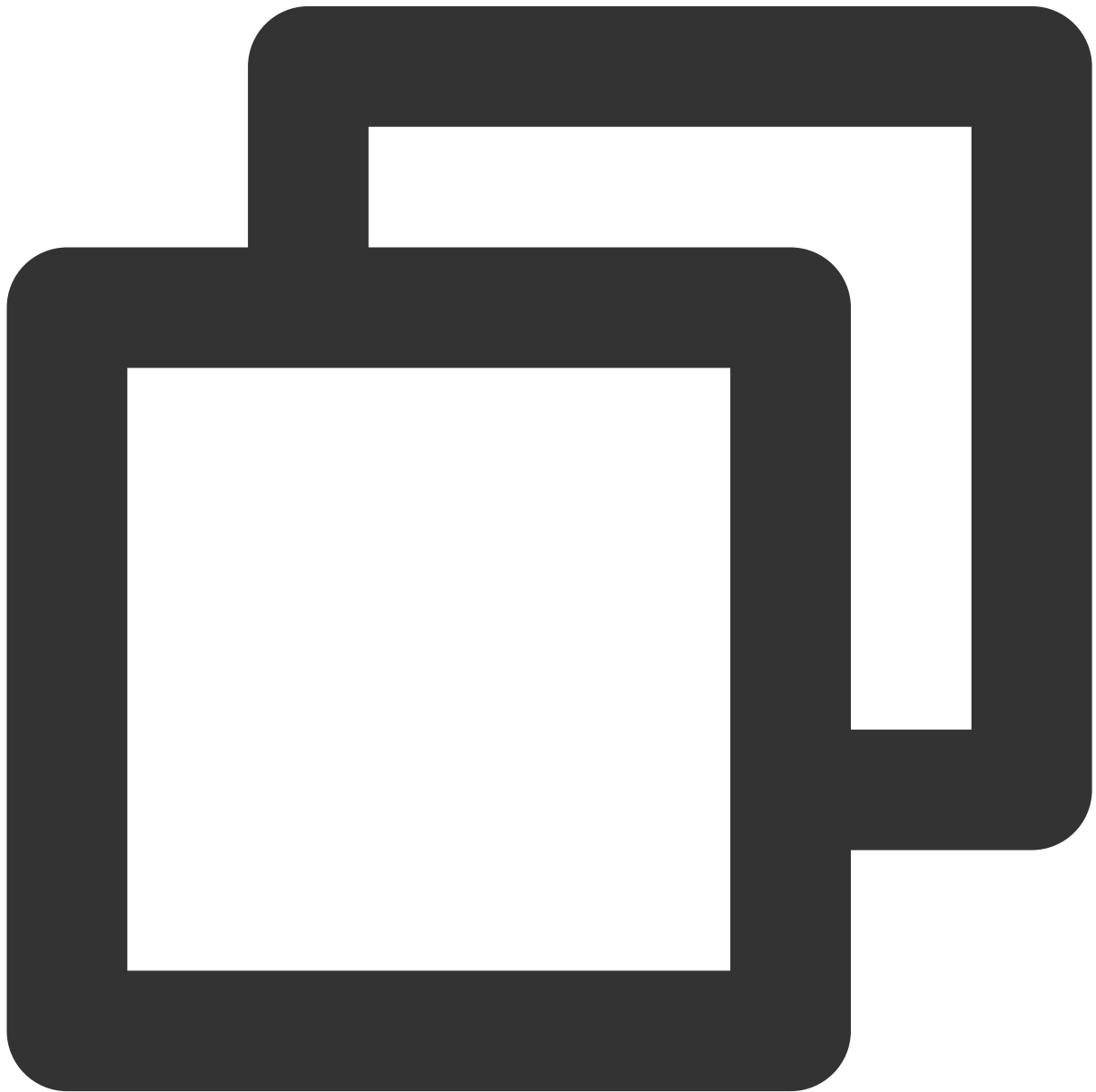


```
RewriteEngine on
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^(.*)?$ https://%{SERVER_NAME}%{REQUEST_URI} [L,R]
```

The result should be as follows:

```
<Directory "/home/www/htdocs/">
  Options FollowSymLinks
  AllowOverride All
  Require all granted
  RewriteEngine on
  RewriteCond %{SERVER_PORT} !^443$
  RewriteRule ^(.*)?$ https://%{SERVER_NAME}%{REQUEST_URI} [L,R]
</Directory>
```

3. Press **Esc** and enter **:wq** to save the changes.
4. Run the following command to restart the Apache service.





```
sudo /usr/local/lighthouse/softwares/apache/bin/httpd -k restart
```

At this point, you have successfully set the automatic redirect to HTTPS. You can use

`http://cloud.tencent.com` to redirect to the HTTPS page.

# Installing Certificate on Apache Server (Windows)

Last updated : 2022-06-15 16:05:02

## Overview

This document describes how to install an SSL certificate in a Lighthouse instance and enable HTTPS access, with a Windows Server 2012 R2 system image-based instance as an example.

### Note:

The SSL certificate used in the document is provided by Tencent Cloud. For more information on this service, see [Overview](#) and [Purchase Guide](#).

## Sample information

**Certificate name:** cloud.tencent.com

**Apache version:** Apache/2.4.53. You can download it [here](#). If you need another version, [contact us](#).

**OS:** Windows Server 2012 R2. The detailed steps may differ by version.

## Prerequisites

Install the Apache service on the current server.

Open port 443 and 80 in your firewall policy. For more information, see [Managing Firewall](#).

The data required to install the SSL certificate includes the following:

Name	Description
Lighthouse instance's public IP address	Instance IP address used to connect a local computer to the instance.
Username	The username used to log in to the Lighthouse instance, such as `Administrator`.
Password	The password matching the username used to log in to the Lighthouse instance.

### Note:

You can log into the [Lighthouse console](#), find the target instance, and enter its details page to view its public IP address. After the instance is created, first reset the password and remember it. For more information, see [Resetting Password](#).

## Directions

### Uploading certificate file

1. Log in to the [SSL Certificate Service console](#) and click **Download** for the certificate you need to install.
2. In the pop-up window, select **Apache** for the server type, click **Download**, and decompress the `cloud.tencent.com` certificate file package to a local directory.

After decompression, you can get the certificate file of the corresponding type, which includes the `cloud.tencent.com_apache` folder.

**Folder:** `cloud.tencent.com_apache`

#### Files in the folder:

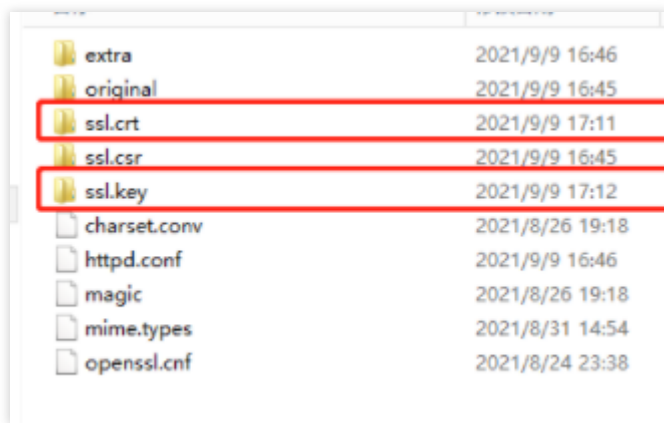
`root_bundle.crt` : Certificate file  
`cloud.tencent.com.crt` : Certificate file  
`cloud.tencent.com.key` : Private key file  
`cloud.tencent.com.csr` : CSR file

#### Note:

You can upload the CSR file when applying for a certificate or have it generated online by the system. It is provided to the CA and irrelevant to the installation.

3. Log in to the Lighthouse instance as instructed in [Logging In to Windows Instance via Remote Desktop Connection](#).
4. Copy the obtained `root_bundle.crt` , `cloud.tencent.com.crt` , and `cloud.tencent.com.key` files from the local directory to the Apache server. For more information on how to upload a certificate file, see [How to Upload Local File to Lighthouse Instance](#).

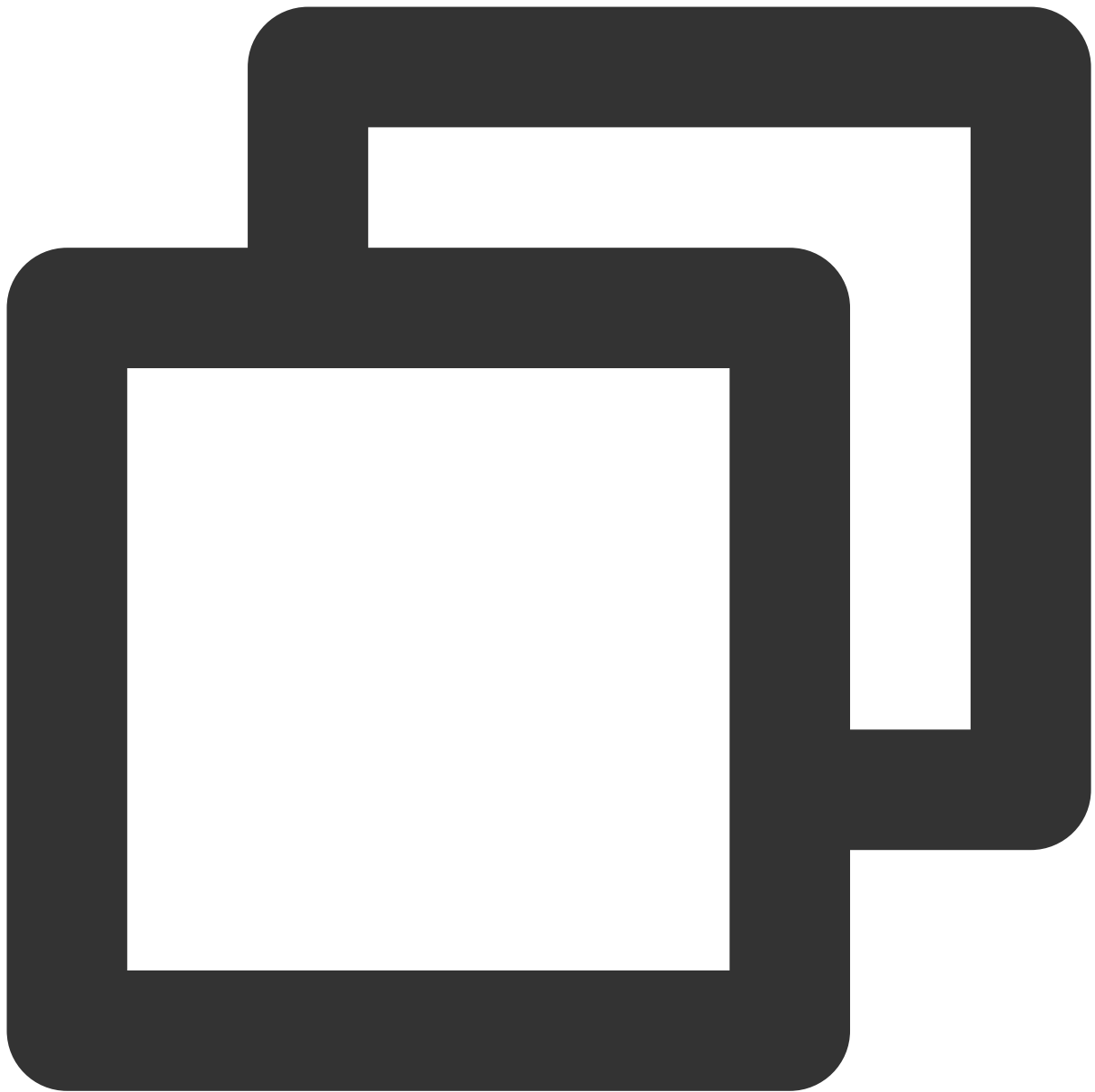
Here, the files are copied to the `ssl.crt` and `ssl.key` folders under the `\\conf` directory. You can specify the file location. The sample directory in this document is as shown below:



SSL Certificate File	Folder
root_bundle.crt	ssl.crt
cloud.tencent.com.crt	
cloud.tencent.com.key	ssl.key

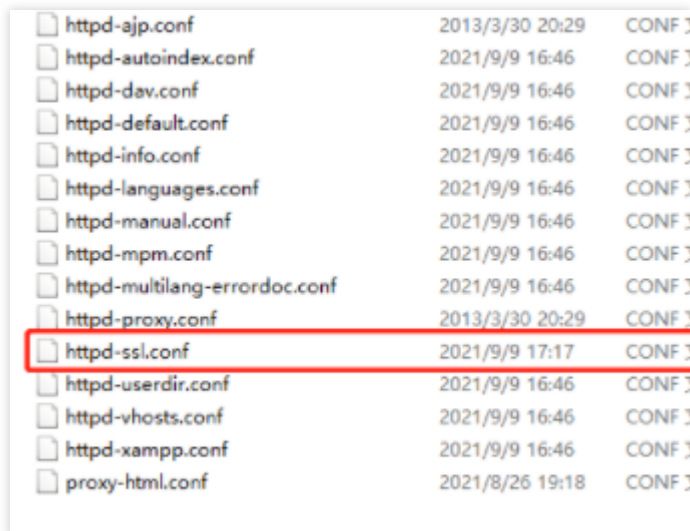
## Configuration file

1. Open the `httpd.conf` file in the `conf` directory of the Apache server with a text editor and delete the `#` before the following fields.



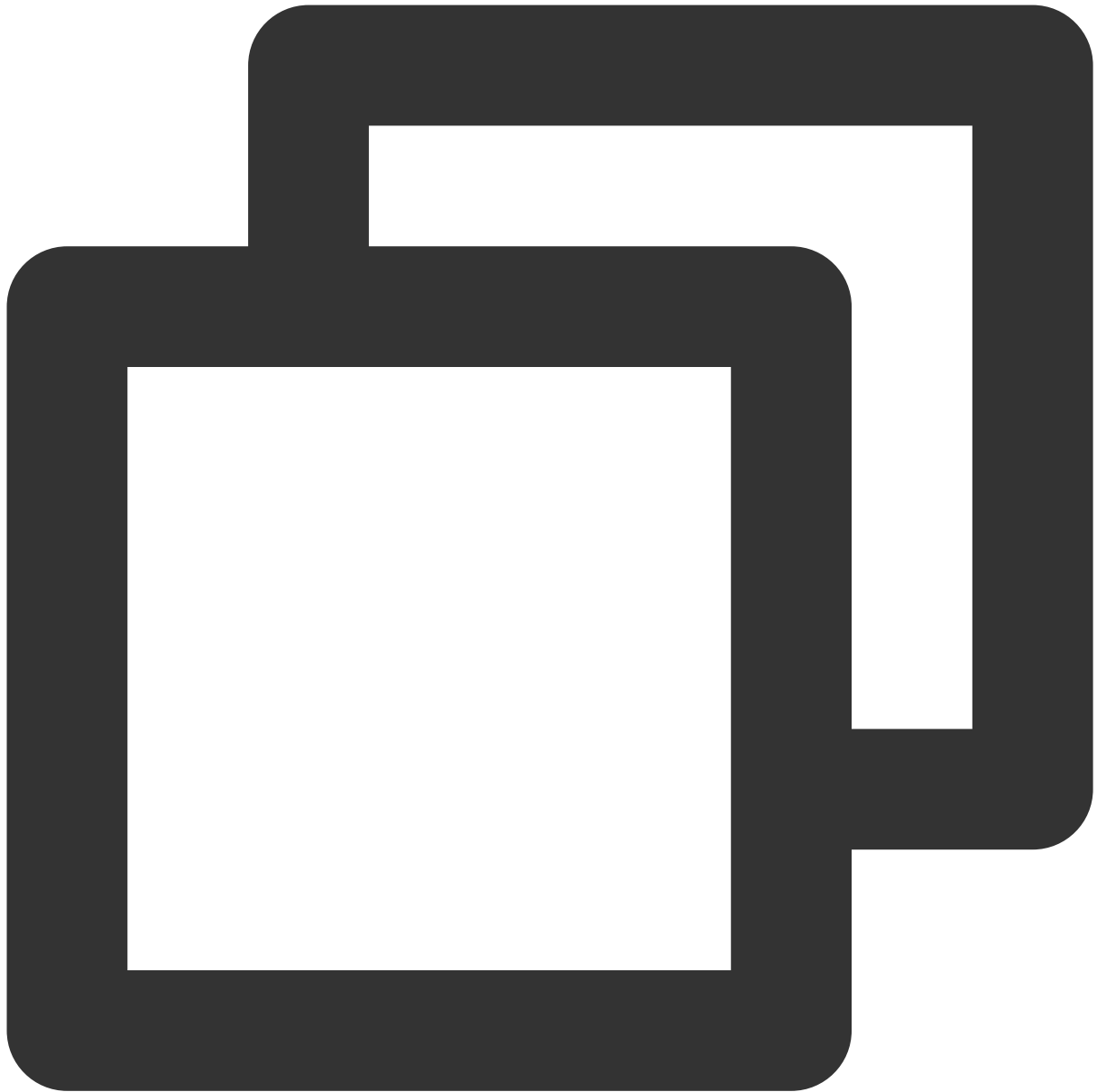
```
#LoadModule ssl_module modules/mod_ssl.so
#Include conf/extra/httpd-ssl.conf
```

2. Open the `httpd-ssl.conf` file in the `conf\extra` directory of the Apache server with a text editor as shown below:



httpd-ajp.conf	2013/3/30 20:29	CONF 3
httpd-autoindex.conf	2021/9/9 16:46	CONF 3
httpd-dav.conf	2021/9/9 16:46	CONF 3
httpd-default.conf	2021/9/9 16:46	CONF 3
httpd-info.conf	2021/9/9 16:46	CONF 3
httpd-languages.conf	2021/9/9 16:46	CONF 3
httpd-manual.conf	2021/9/9 16:46	CONF 3
httpd-mpm.conf	2021/9/9 16:46	CONF 3
httpd-multilang-errordoc.conf	2021/9/9 16:46	CONF 3
httpd-proxy.conf	2013/3/30 20:29	CONF 3
httpd-ssl.conf	2021/9/9 17:17	CONF 3
httpd-userdir.conf	2021/9/9 16:46	CONF 3
httpd-vhosts.conf	2021/9/9 16:46	CONF 3
httpd-xampp.conf	2021/9/9 16:46	CONF 3
proxy-html.conf	2021/8/26 19:18	CONF 3

3. Modify the `httpd-ssl.conf` file and set the following field parameters to the paths of the uploaded certificate files as shown below:



```
SSLCertificateFile "C:/apache/conf/ssl.crt/cloud.tencent.com.crt"  
SSLCertificateKeyFile "C:/apache/conf/ssl.key/cloud.tencent.com.key"  
SSLCertificateChainFile "C:/apache/conf/ssl.crt/root_bundle.crt"
```

**Note:**

If there is no `SSLCertificateChainFile` entry in the `httpd-ssl.conf` configuration file, add it to the corresponding location as shown below:

```
# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCACertificatePath you need hash symlinks
#       to point to the certificate files. Use the provided
#       Makefile to update the hash symlinks after changes.
#SSLCACertificatePath "${SRVROOT}/conf/ssl.crt"
#SSLCACertificateFile "${SRVROOT}/conf/ssl.crt/ca-bundle.crt"
SSLCertificateChainFile "C:/apache/conf/ssl.crt/root_bundle.crt"
```

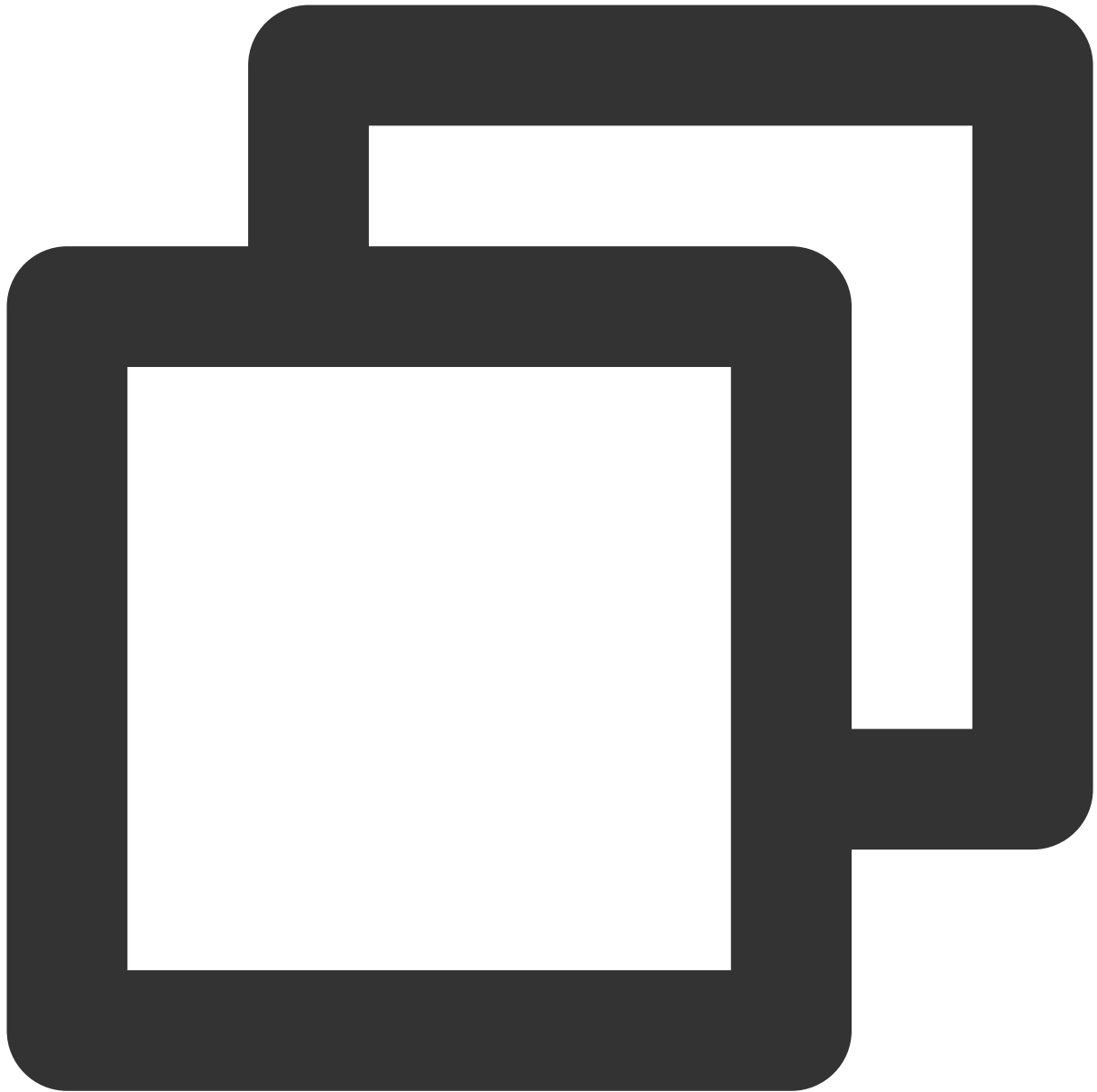
4. Restart the Apache server and then you access it through `https://cloud.tencent.com` .

If the "AH00526: Syntax error on line 18 of C:/apache/conf/extra/httpd-ahssl.conf:Cannot define multiple Listeners on the same IP:port" error is reported during the restart, there is a listening port conflict. Replace the listening port `443` in `conf\extra\httpd-ahssl.conf` with another port.

### (Optional) Security configuration for automatic redirect from HTTP to HTTPS

1. Open the `httpd.conf` file in the `conf` directory of the Apache server with a text editor and delete the `#` before the following fields.

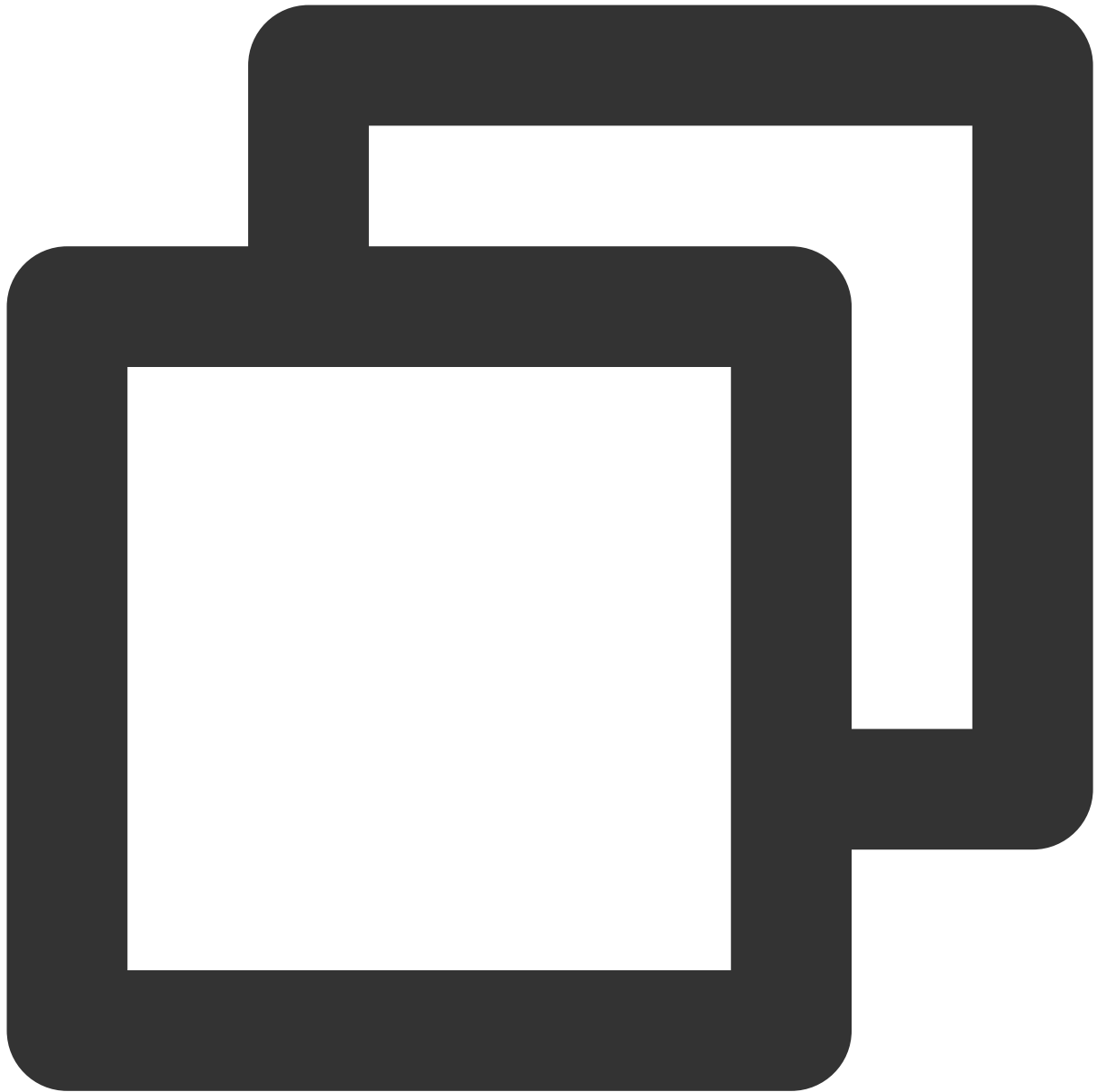




```
#LoadModule rewrite_module modules/mod_rewrite.so
```

2. Configure the fields in the website running directory. For example, add the following content to the `<Directory`

```
"C:/xampp/htdocs"> field:
```



```
<Directory "C:/xampp/htdocs">
RewriteEngine on
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^(.*)?$ https://%{SERVER_NAME}%{REQUEST_URI} [L,R]
</Directory>
```

3. Restart the Apache server and then you can access it through both `http://cloud.tencent.com` (which will be automatically redirected to `https://cloud.tencent.com` ) and `https://cloud.tencent.com` .