



---

# Computer Networks-Lab 06

---



---

**Instructor: HURMAT HIDAYAT**

**CL30001 – Computer Networks-Lab**

**SEMESTER Fall 2022**

---

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES, FAST- PESHAWAR CAMPUS  
Department of Computer Science & Software Engineering

# Computer Networks- Lab 06

---

## Objectives

After these Lab students shall be able to perform

- **Introduction with SMTP, IMAP, POP3, FTP**
- **Configure the of SMTP and POP3 on generic server.**
  - Send email from client1 to client2 and analyze them.
- **Configure Ftp on generic server**
  - Send files from client to sever and analyze them  
(Delete, Update and rename these files)
  - Send .html files to http directory of server through ftp.
  - Open these html file from client PC

## Table of Contents

Objectives .....	1
Email protocols .....	3
What is SMTP? .....	3
What is POP3? .....	4
What is IMAP? .....	4
What is the difference between SMTP, POP3, and IMAP? .....	5
Incoming vs. outgoing protocols.....	5
IMAP vs. POP3.....	5
How to configure an email server in Packet Tracer .....	6
File Transfer Protocol (FTP).....	13
How to configure an FTP server in Packet Tracer .....	13
Create and Upload html file to HTTP server directory Using FTP .....	19
Tasks for students: .....	28

## Email protocols

Email is an essential part of business and personal communication online. The email protocols define the mechanism of the email exchange between servers and clients. This way, they allow us to send and receive messages over the network correctly.

Email is an essential part of business and personal communication online. The email protocols define the mechanism of the email exchange between servers and clients. This way, they allow us to send and receive messages over the network correctly.

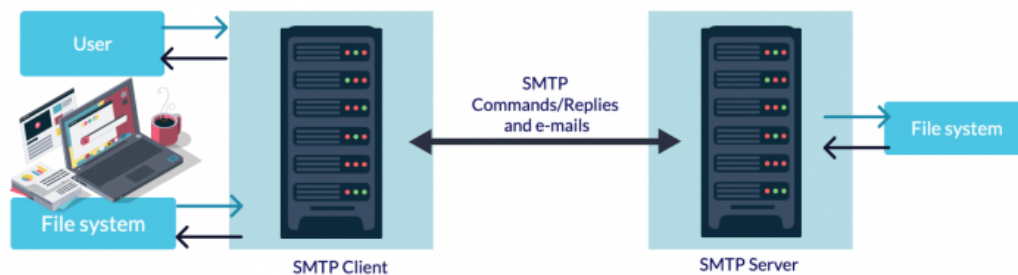
Email protocols list

Email is an essential part of business and personal communication online. The email protocols define the mechanism of the email exchange between servers and clients. This way, they allow us to send and receive messages over the network correctly.

## What is SMTP?

SMTP stands for Simple Mail Transfer Protocol, and it is responsible for sending email messages. This protocol is used by email clients and mail servers to exchange emails between computers.

# How SMTP works



A mail client and the SMTP server communicate with each other over a connection established through a particular email port. Both entities are using SMTP commands and replies to process your outgoing emails. Thanks to the Simple Mail Transfer Protocol, messages can be sent from the same account on different email applications.

## What is POP3?

### How POP3 works?

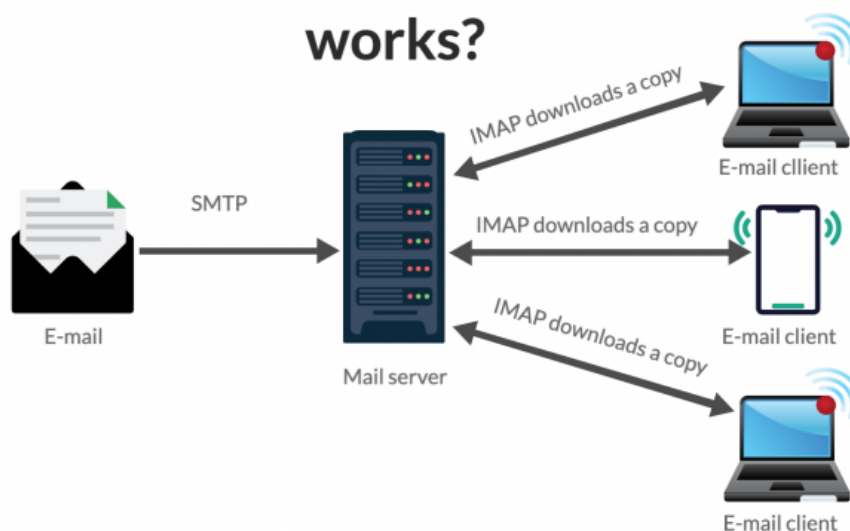


The POP3 abbreviation stands for Post Office Protocol version 3, which provides access to an inbox stored in an email server. It executes the download and delete operations for messages. Thus, when a POP3 client connects to the mail server, it retrieves all messages from the mailbox. Then it stores them on your local computer and deletes them from the remote server.

Thanks to this protocol, you are able to access the messages locally in offline mode as well. Modern POP3 clients allow you to keep a copy of your messages on the server if you explicitly select this option.

## What is IMAP?

### How IMAP works?



The Internet Message Access Protocol (IMAP) allows you to access and manage your email messages on the email server. This protocol permits you to manipulate folders, permanently delete and efficiently search through messages. It also gives you the option to set or remove email flags, or fetch email attributes selectively. By default, all messages remain on the server until the user specifically deletes them.

IMAP supports the connection of multiple users to a single mail server.

## **What is the difference between SMTP, POP3, and IMAP?**

### **Incoming vs. outgoing protocols**

POP3 and IMAP are handling the incoming emails and they operate in different ways to retrieve or access your email messages. Thus, they are considered mail access protocols.

On the other hand, the Simple Mail Transfer Protocol is behind the message transfer from server to server, or mail client to server. As this is the protocol handling the email sending from an email account, it is labeled as the outgoing protocol.

In short, thanks to IMAP and POP3, you are able to receive emails, and SMTP allows you to send messages.

### **IMAP vs. POP3**

As we already mentioned, both of these protocols relate to email retrieval. All modern servers support both protocols, although they function in different manners.

While the POP3 protocol assumes that your email is being accessed only from one application, IMAP allows simultaneous access by multiple clients. This is why IMAP is more suitable for you, if you're going to access your email from different locations or if your messages are managed by multiple users.

On the other hand, POP3 downloads your emails to your local computer, deleting them from the server. Thus, it reduces the space your email account uses on your web server.

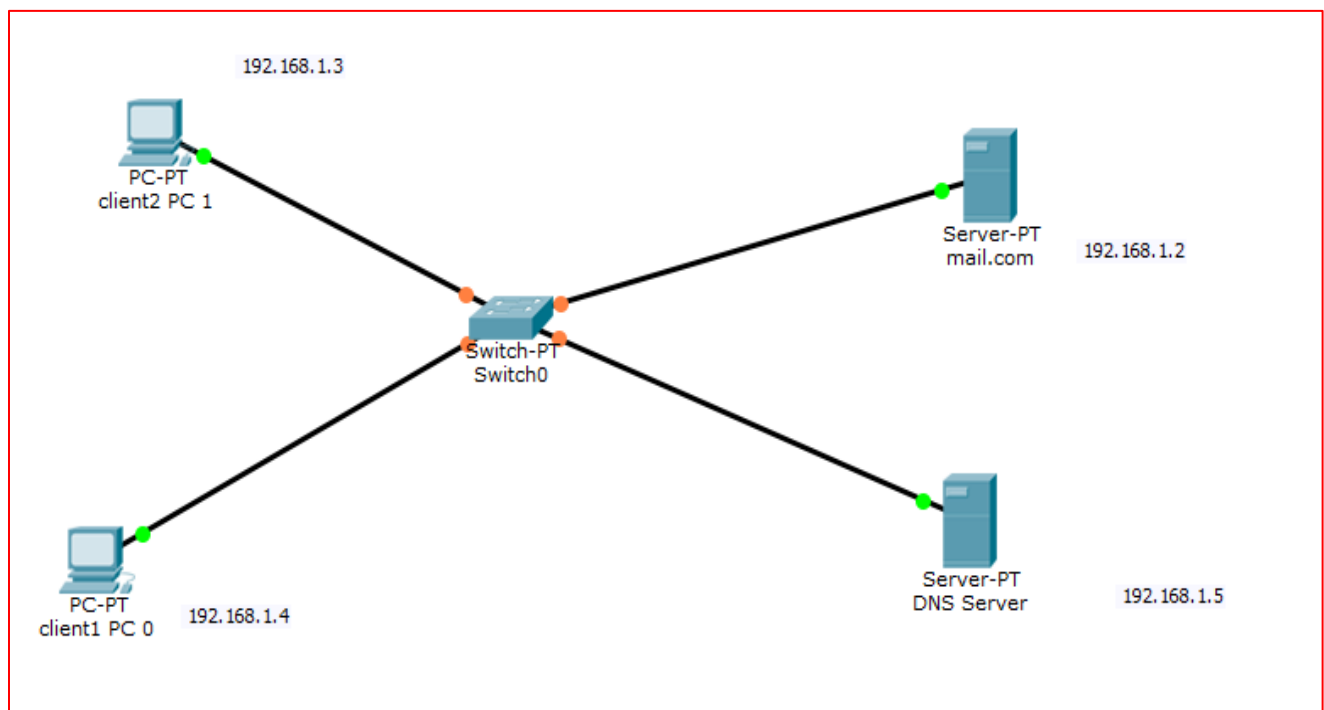
To sum it up, there are 3 email protocols – SMTP, POP3, and IMAP. Each of them works on specific port numbers and operates differently. If you are having trouble connecting to an incoming or outgoing server, try using an alternative port number.

## How to configure an email server in Packet Tracer

We'll do a simple configuration of an email server in Packet Tracer. An email server, such as Gmail stores and sends email messages to email clients on request. We often send and receive emails on our mobile devices or computers. Have you ever imagined how this happens? Well, whenever you compose and send an email to another person, the message you send first goes to a mail server. It's the mail server which then sends the email when it is requested from the email client(e.g. Gmail App) of the recipient's device.

So now, let's configure a mail server in Packet Tracer. And have in mind that although our main focus is configuring an email server, we'll still need services of a DNS server at one point. Let's dive right in.

### 1. Build the network topology



### 2. Configure IP addresses on the PCs, DNS Server and the Mail Server

- Mail Server IP address: 192.168.1.2
- PC0 IP address: 192.168.1.4
- PC1 IP address: 192.168.1.3
- DNS server IP address: 192.168.1.5

### 3. Configure mail clients on the PCs and mail service on the generic server.

#### Mail Clients:

Click on **PC0**. Go to its **Desktop** tab, and click on **Email**. Configure the email client by filling in the user, server and login information. Be sure to **Save**.

**PC0:**

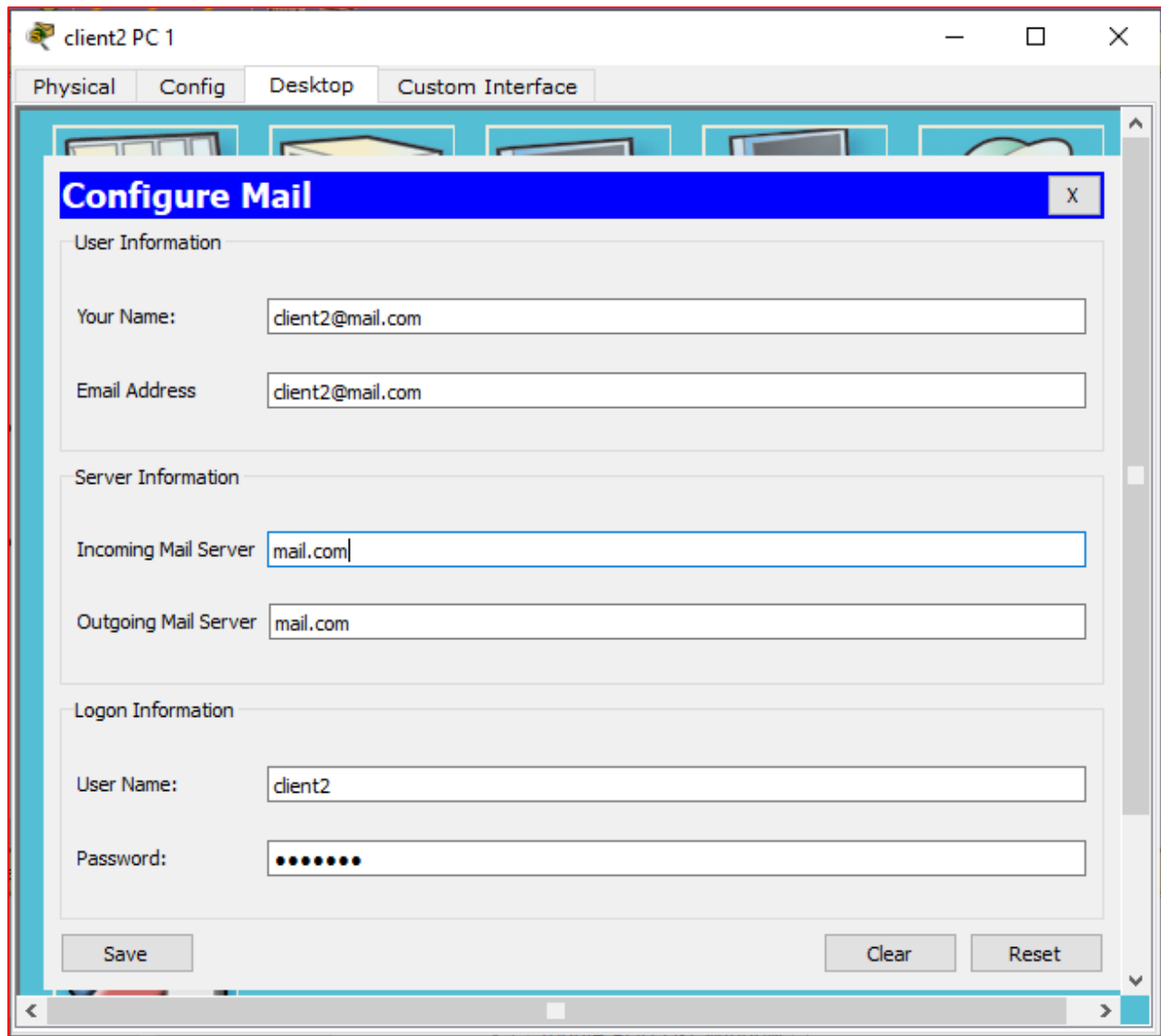
The screenshot shows a window titled 'client1 PC 0' with tabs for 'Physical', 'Config', 'Desktop', and 'Custom Interface'. The 'Config' tab is active, displaying a 'Configure Mail' dialog box. The dialog box has a blue title bar with a close button (X). It is divided into three sections: 'User Information', 'Server Information', and 'Logon Information'. In the 'User Information' section, 'Your Name' is 'client1' and 'Email Address' is 'client1@mail.com'. In the 'Server Information' section, both 'Incoming Mail Server' and 'Outgoing Mail Server' are 'mail.com'. In the 'Logon Information' section, 'User Name' is 'client1' and 'Password' is masked with dots. At the bottom are 'Save', 'Clear', and 'Reset' buttons.

Section	Field	Value
User Information	Your Name:	client1
	Email Address	client1@mail.com
Server Information	Incoming Mail Server	mail.com
	Outgoing Mail Server	mail.com
Logon Information	User Name:	client1
	Password:	.....

**PC1:**

Configure mail client on PC1 in a similar way we did for PC0.





The screenshot shows a window titled 'client2 PC 1' with tabs for 'Physical', 'Config', 'Desktop', and 'Custom Interface'. The 'Config' tab is active, displaying a 'Configure Mail' dialog box. The dialog has three sections: 'User Information', 'Server Information', and 'Logon Information'. In the 'User Information' section, 'Your Name' and 'Email Address' are both set to 'client2@mail.com'. In the 'Server Information' section, 'Incoming Mail Server' and 'Outgoing Mail Server' are both set to 'mail.com'. In the 'Logon Information' section, 'User Name' is set to 'client2' and 'Password' is masked with dots. At the bottom of the dialog are 'Save', 'Clear', and 'Reset' buttons.

### Mail Server:

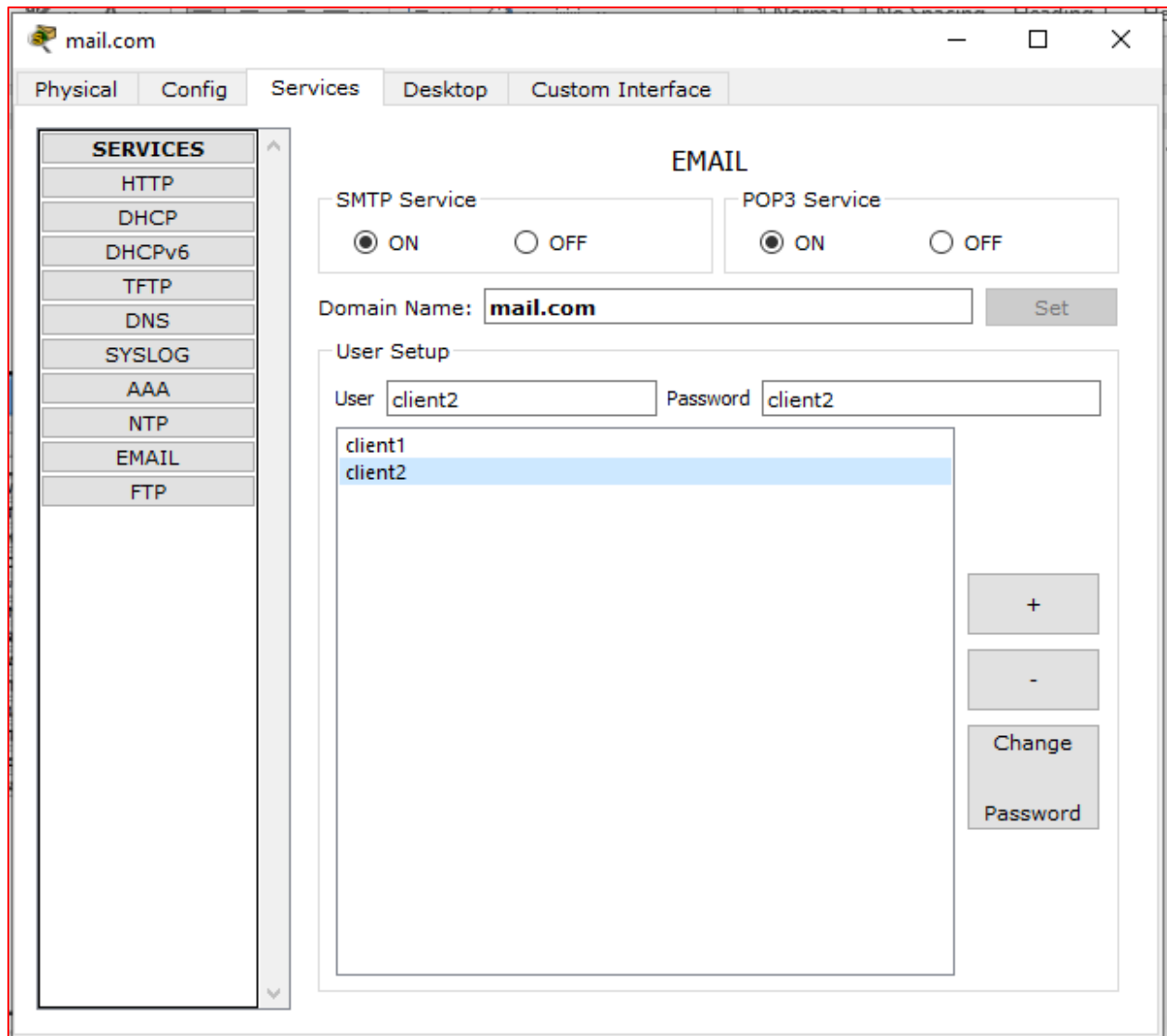
Next, we'll configure the **email server**.

To do this, click on the server, then click **Services** tab, pick **email** server from the menu.

Provide the **Domain name** of the server then click on **Set** to set it. In this example I've used the name 'mail.com'.

Proceed and add **users** and provide their **passwords**. I have two email clients(users) with usernames 'client1' and 'client2' with a common password 'admin'.

After entering a username and password, click on **Add(+)** to add the user to the server. You can optionally remove a user by clicking on **Remove (-)**. You can change a user's password by clicking on **change password**.



Try to relate this process to what happens when you register an email account with a mail service provider(mail server) like Gmail. The processes appear to agree, isn't it?

Now, notice that we set a **domain name** for the email server. For that reason, we should have a **DNS server** that will resolve this domain name (plus other domain names if there were) to an IP address.

So let's configure a DNS server.

Click DNS server, click **Services** tab, then pick **DNS**. Turn the service **ON**. Set name-address pairs and add them to the server. You can view the DNS entry below:

**DNS Server**

Physical Config **Services** Desktop Custom Interface

**SERVICES**

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS**
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP

**DNS**

DNS Service ☒ On ☐ Off

Resource Records

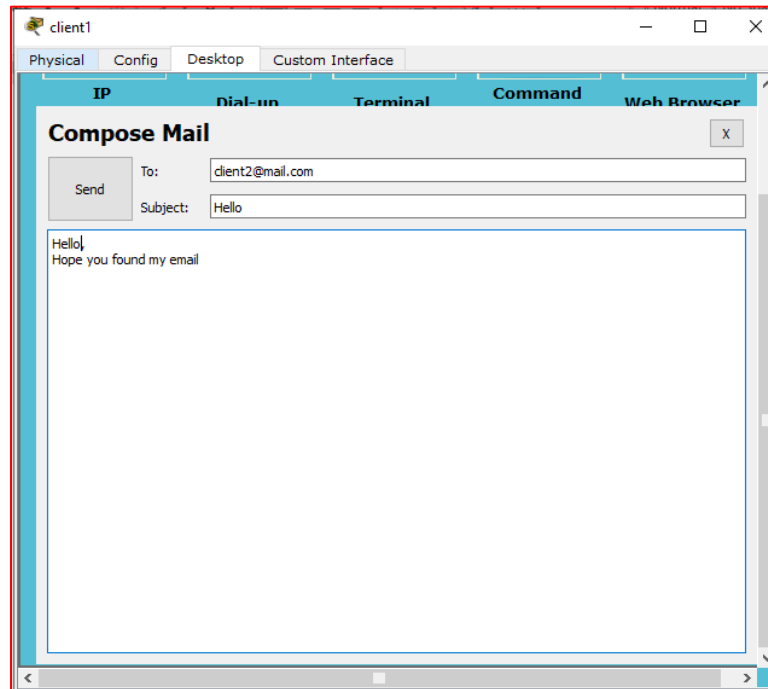
Name  Type **A Record** ▼

Address

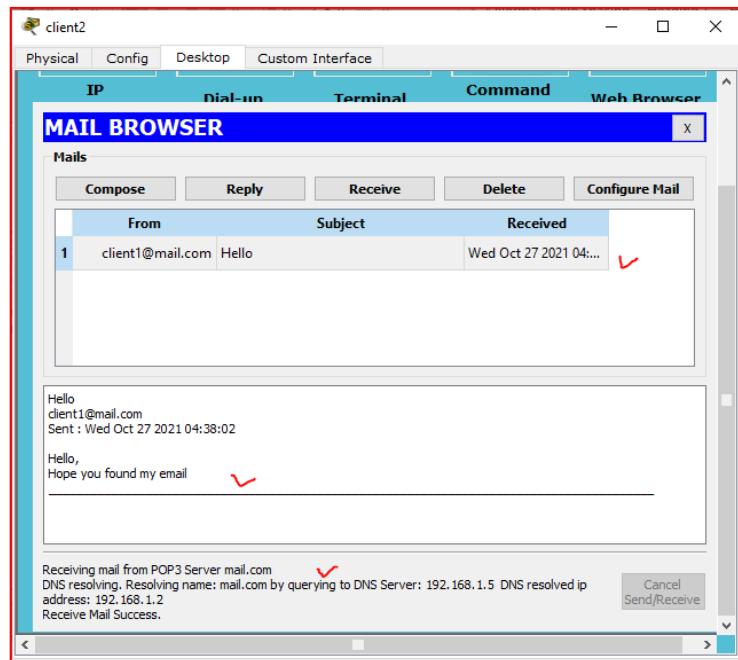
No.	Name	Type	Detail
0	mail.com	A Record	192.168.1.2

4. Lastly test the email service.

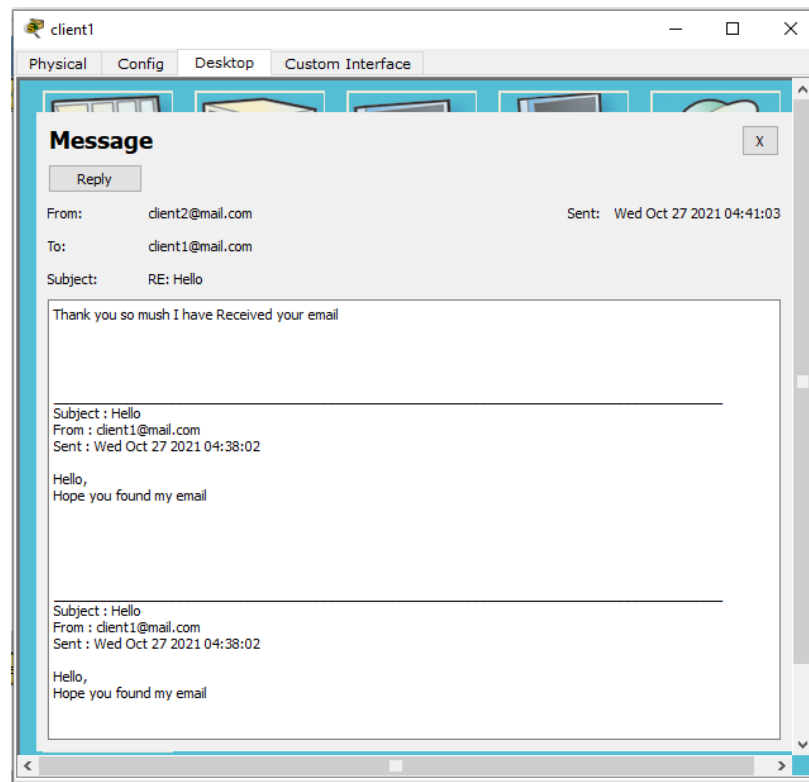
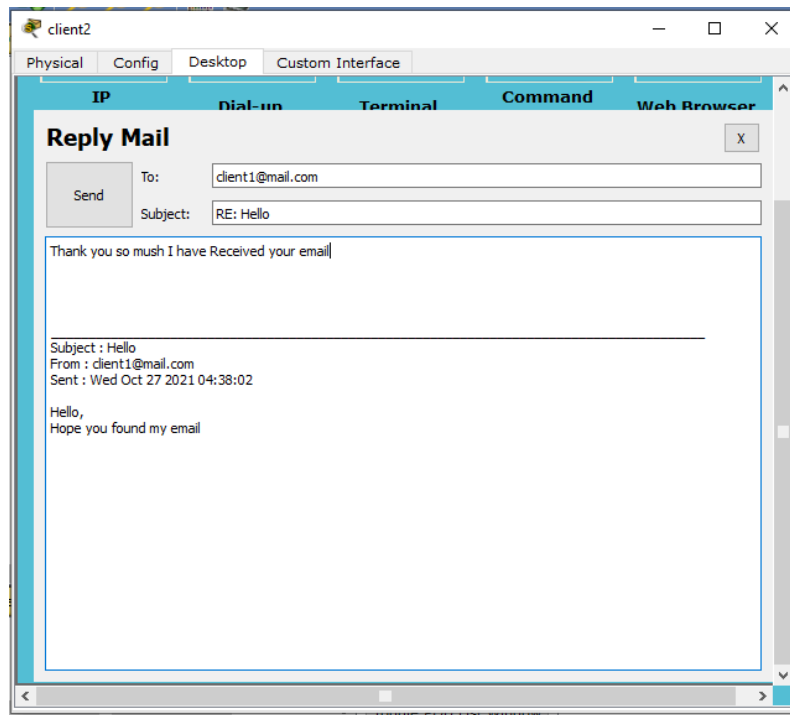
Go to **PC0** **email** client, **compose** an email and **send** it to **PC1** email address (**client2@mail.com**).



Try to see whether the email from PC0 is received on PC1. On the **email** client of PC1, click on **Receive**.



**Note:** If everything is well set up, the email from PC0 will be received on PC1.



## File Transfer Protocol (FTP)

The **File Transfer Protocol (FTP)** is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

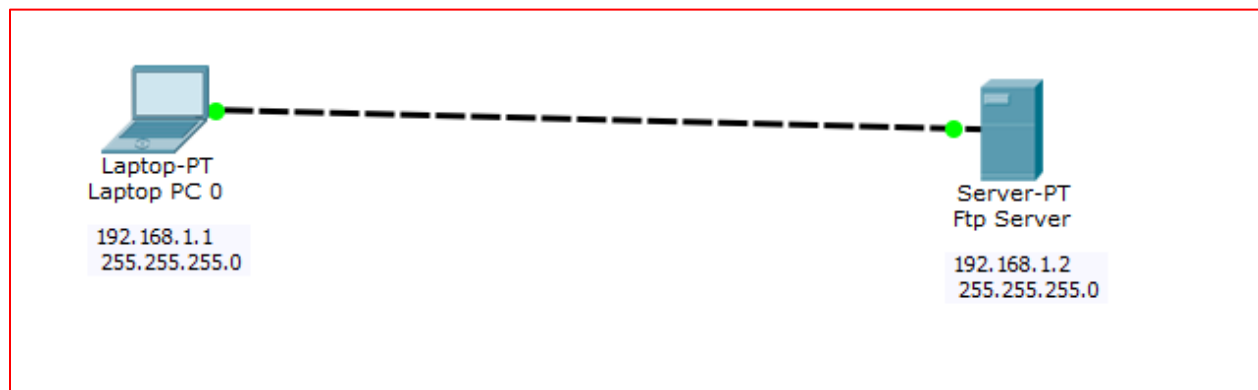
FTP employs a **client-server** architecture whereby the client machine has an **FTP client** installed and establishes a connection to an **FTP server** running on a remote machine. After the connection has been established and the user is successfully authenticated, the data transfer phase can begin.

**Worth noting:** Although FTP does support **user authentication**, all data is sent in clear text, including usernames and passwords. For **secure** transmission that protects the username and password, and encrypts the content, FTP is often secured with **SSL/TLS (FTPS)** or replaced with **SSH File Transfer Protocol (SFTP)**

## How to configure an FTP server in Packet Tracer

Let's now do FTP configuration in Packet Tracer:

### 1. Build the network topology.



### 2. Configure static IP addresses on the Laptop and the server

**Laptop:** IP address: 192.168.1.1 Subnet Mask: 255.255.255.0

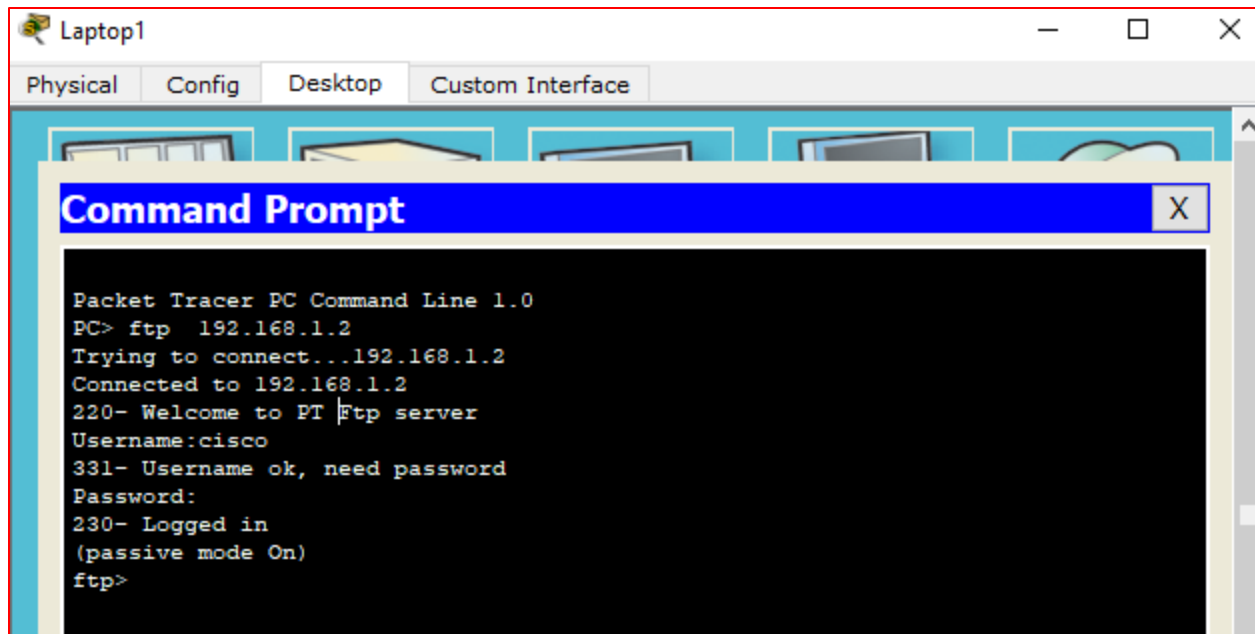
**Server:** IP address: 192.168.1.2 Subnet Mask: 255.255.255.0

### 3. Now try using an FTP client built in the Laptop to send files to an FTP server configured in the Server.

From the Laptop's command prompt, FTP the server using the server IP address by typing:

```
ftp 192.168.1.2
```

Provide the **username(cisco)** and **password(cisco)** [which are the defaults] for ftp login.



You are now in the **FTP prompt** .

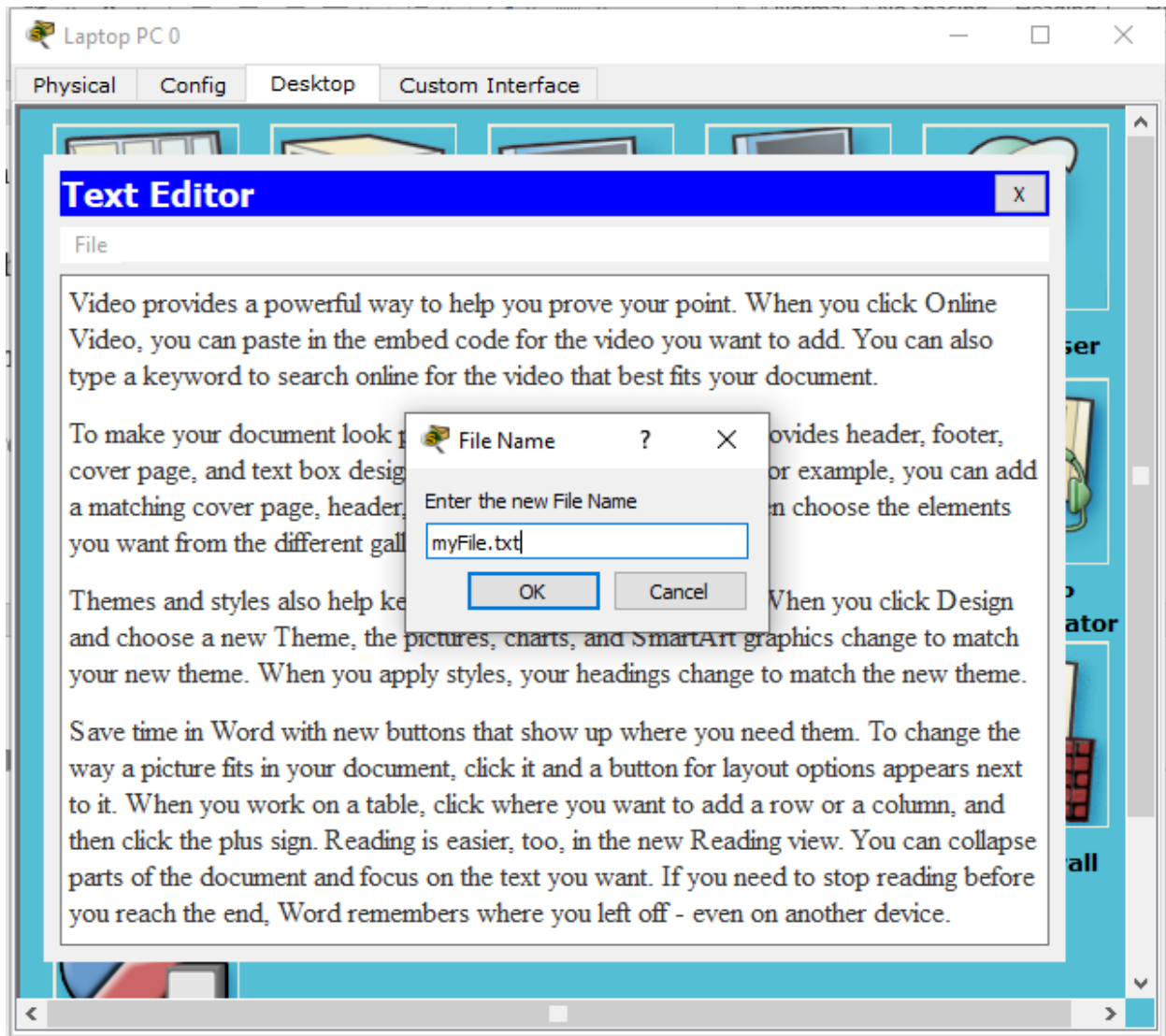
**PC0** has an **FTP client** which can be used to **read**, **write**, **delete** and **rename** files present in the FTP server.

The **FTP server** can be used to **read** and **write** configuration files as well as IOS images. Additionally, the FTP server also supports file operations such **rename**, **delete** and **listing** directory.

With that in mind, we can do something extra. So let's do this:

**4. Create a file in the Laptop then upload it to the server using FTP.**

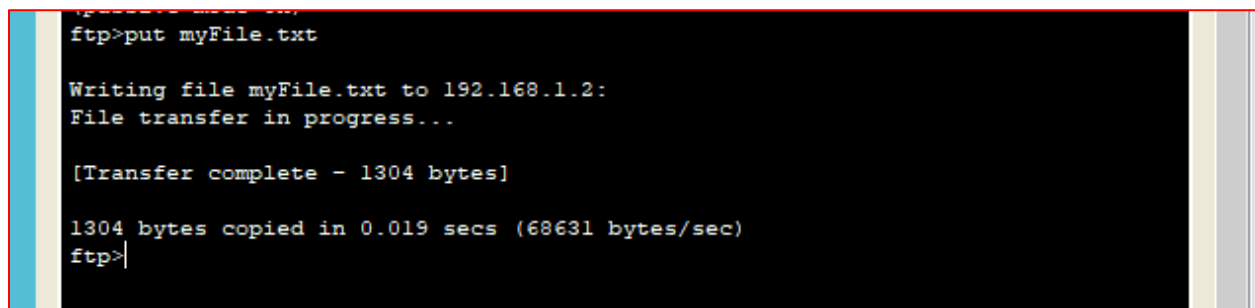
To do this, open the **Text Editor** in the Laptop, create a file and give it your name of choice. Type any text in the editor then **save** your file. e.g. **myFile.txt**.



Now upload the file from the Laptop to the server using FTP. (An FTP connection has to be started first. But this is what we've done in step 3)

So to do an FTP upload, we'll type:

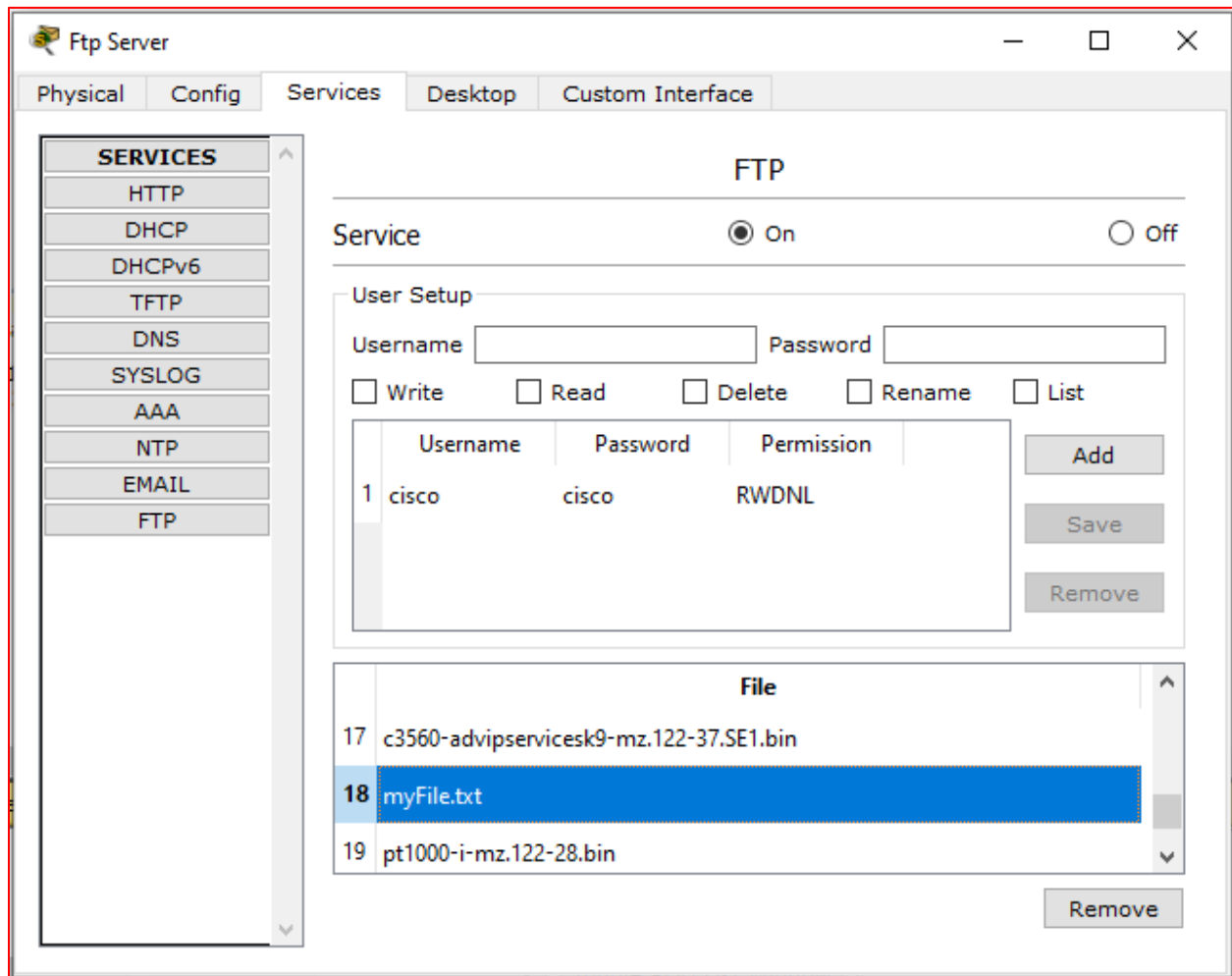
```
put myFile.txt
```





**5. Go to the Server FTP directory to verify if the file sent has been received**

Once file upload is successful, go to the Server **FTP directory** to verify if the file sent has been received. To do this, go to **Server-> Services->FTP**. Here look for **myFile.txt** sent from the laptop.



**Something extra:** To check other FTP commands supported by the FTP client running on the Laptop(or PC), you can use a question mark (?) on the Laptop's command prompt as shown below:

```
ftp>?
?
cd
delete
dir
get
help
passive
put
pwd
quit
rename
ftp>|
```

You can see the put command that we used to upload our file to the FTP server. Other commands listed include:

*get*—used to get(download) a file from the server.

For example: **get myFile.txt**

*delete*— to delete a file in the FTP directory with the server

For example: **delete myFile.txt**

*Rename*— used to Rename a file

*cd* – used to change directory.

## 6. Open HTTP directory and upload file there.

For example, we can open an **HTTP directory** in the server by typing: **cd /http**. This will change the current directory from FTP directory to HTTP directory

Once the http directory is open, you can upload a file to the HTTP server. You're now uploading a file to an HTTP folder(directory) using FTP.

For example: **put myFile.txt**

To see this working, let's **open an HTTP directory** and upload(**put**) a file to it using FTP:

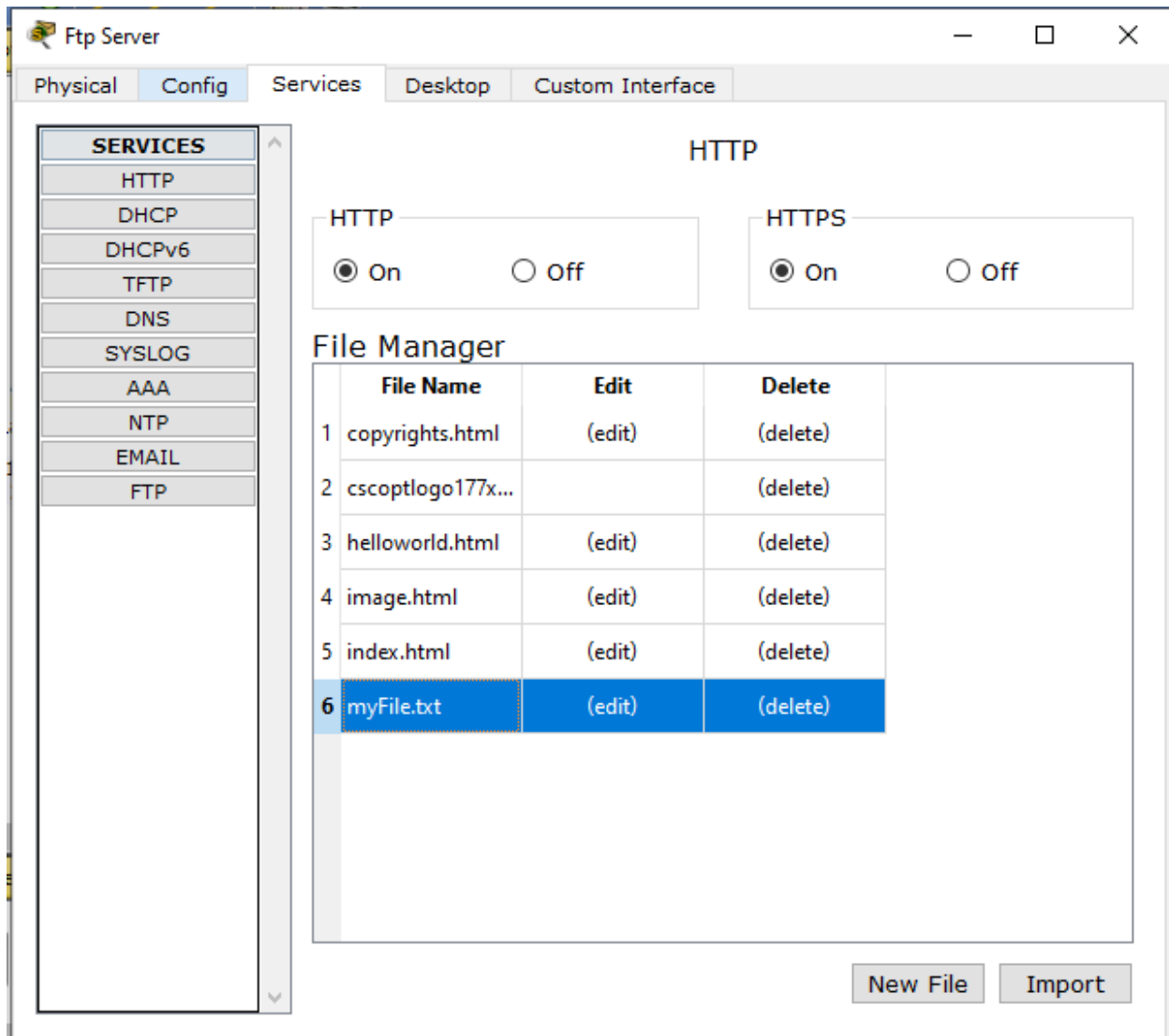
```
ftp>cd /http
ftp>
Working directory changed to /http successfully
ftp>put myFile.txt

Writing file myFile.txt to 192.168.1.2:
File transfer in progress...

[Transfer complete - 1304 bytes]

1304 bytes copied in 0.028 secs (46571 bytes/sec)
ftp>
```

You can now check up in the **HTTP directory** in the server and verify that the file uploaded from the Laptop(**myFile.txt**) is well received:



Notice that we are uploading files to an HTTP Server directory using File Transfer Protocol.(FTP). This is what actually happens when you use an **FTP client** such as [FileZilla client](#) to upload files to a website. In our case here, we are using an FTP client **built-in** the Laptop.

***This may interest you:*** The first FTP client applications were [command-line programs](#) developed before operating systems had graphical user interfaces, and are still shipped with most Windows and Linux operating systems. (Actually this is what we have been using this far). Many FTP clients(e.g. FileZilla) and automation utilities have since been developed for desktops, servers, mobile devices, and hardware. FTP has also been incorporated into productivity applications, such as HTML *editors*.

## Create and Upload html file to HTTP server directory Using FTP

We'll **create** an html file in our Laptop, **upload** it to HTTP server directory using FTP, then try to **access** the file from the Laptop's browser.

On the Laptop, open the **text editor**, then type some markup(html) and save the file with the extension **.html**. See all this below:

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {font-family: Arial, Helvetica, sans-serif;}

form {border: 3px solid #f1f1f1;}


input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;
}


button {
background-color: #0471F3;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
}
```

```
button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.avatar {
  width: 30%;
  height: 30%;
}

}

.container {
  padding: 16px;
}

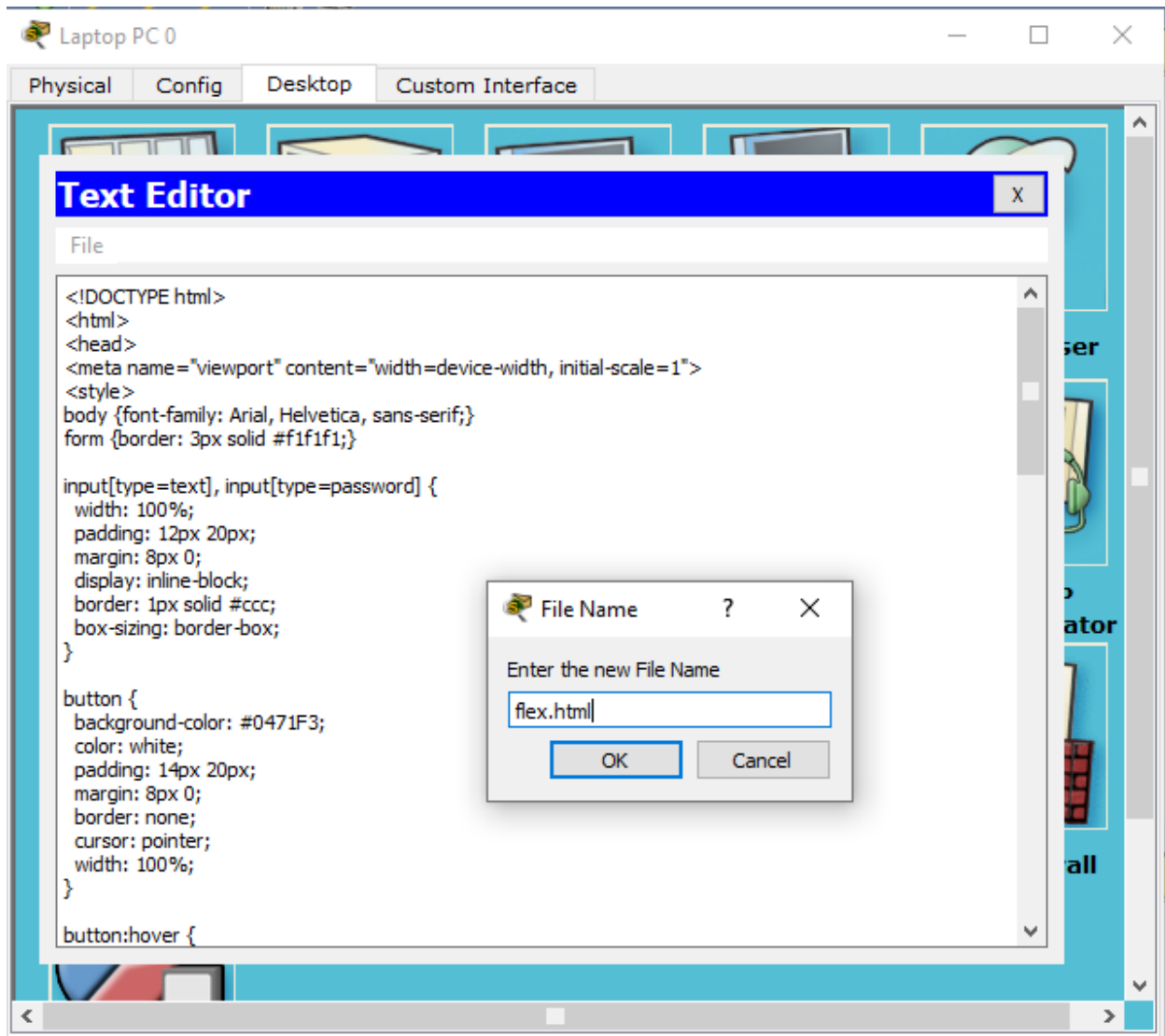
span.psw {
  float: right;
  padding-top: 16px;
}

}

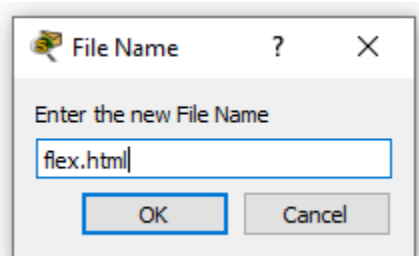
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
    display: block;
    float: none;
  }

  .cancelbtn {
    width: 100%;
  }
}
```

```
}  
}  
  
</style>  
  
</head>  
  
<body>  
  
<form action="/action_page.php" method="post">  
  
<div class="container">  
  
<h2>Flex Academic Suite</h2>  
  
</div>  
  
<div class="imgcontainer">  
  
  
  
</div>  
  
<div class="container">  
  
<label for="uname"><b>Username</b></label>  
  
<input type="text" placeholder="Enter Username" name="uname" required>  
  
  
  
<label for="psw"><b>Password</b></label>  
  
<input type="password" placeholder="Enter Password" name="psw" required>  
  
<button type="submit">Login</button>  
  
<label>  
  
<input type="checkbox" checked="checked" name="remember"> Remember me  
  
</label>  
  
</div>  
  
  
  
<div class="container" style="background-color:#f1f1f1">  
  
<button type="button" class="cancelbtn">Cancel</button>  
  
<span class="psw">Forgot <a href="#">password?</a></span>  
  
</div>  
  
</form>  
  
  
  
</body>  
  
</html>
```

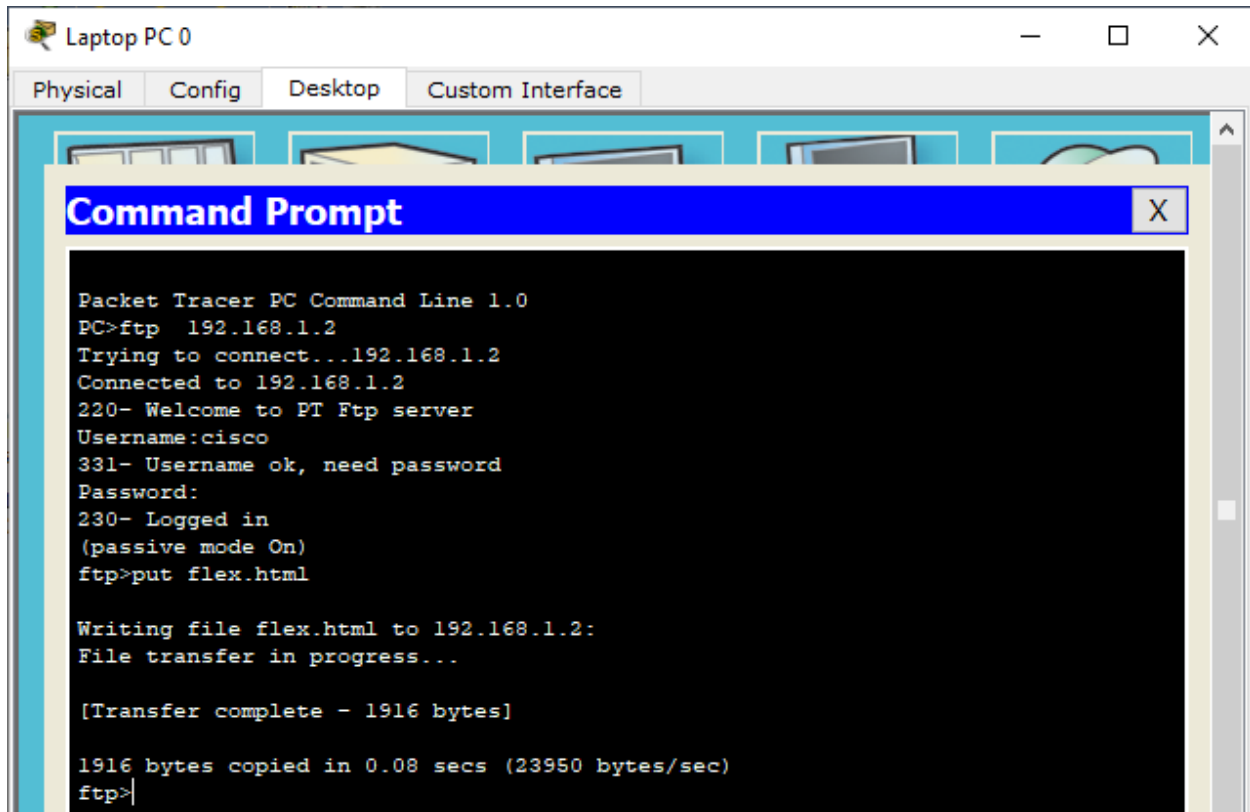


Save your file as an html file like this:



Now upload the file( **flex.html**) to the HTTP server using FTP. This is easy. We've already done it previously!

If you're already in the **HTTP directory**, you just need to type: **put flex.html**. If no, first ftp the server(**ftp 192.168.1.2**), provide the login username(**cisco**) and password(**cisco**); change the current directory to HTTP(**cd /http**) , and finally upload the html file onto the HTTP directory(**put flex.html**)



The screenshot shows a Packet Tracer PC Command Line window for 'Laptop PC 0'. The window has tabs for 'Physical', 'Config', 'Desktop', and 'Custom Interface'. The 'Desktop' tab is active, showing a virtual desktop with a 'Command Prompt' window. The Command Prompt displays the following text:

```
Packet Tracer PC Command Line 1.0
PC>ftp 192.168.1.2
Trying to connect...192.168.1.2
Connected to 192.168.1.2
220- Welcome to PT Ftp server
Username:cisco
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put flex.html

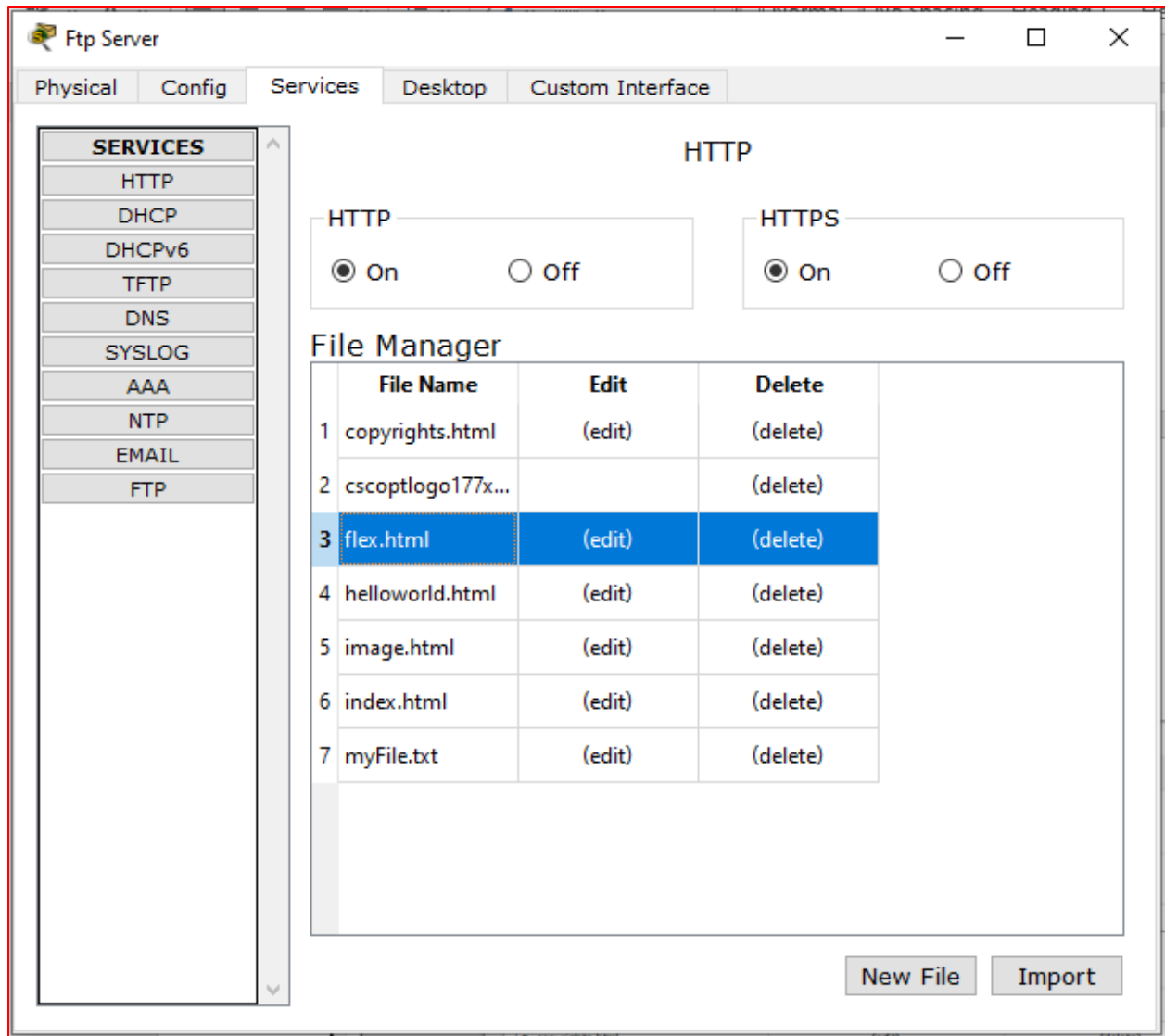
Writing file flex.html to 192.168.1.2:
File transfer in progress...

[Transfer complete - 1916 bytes]

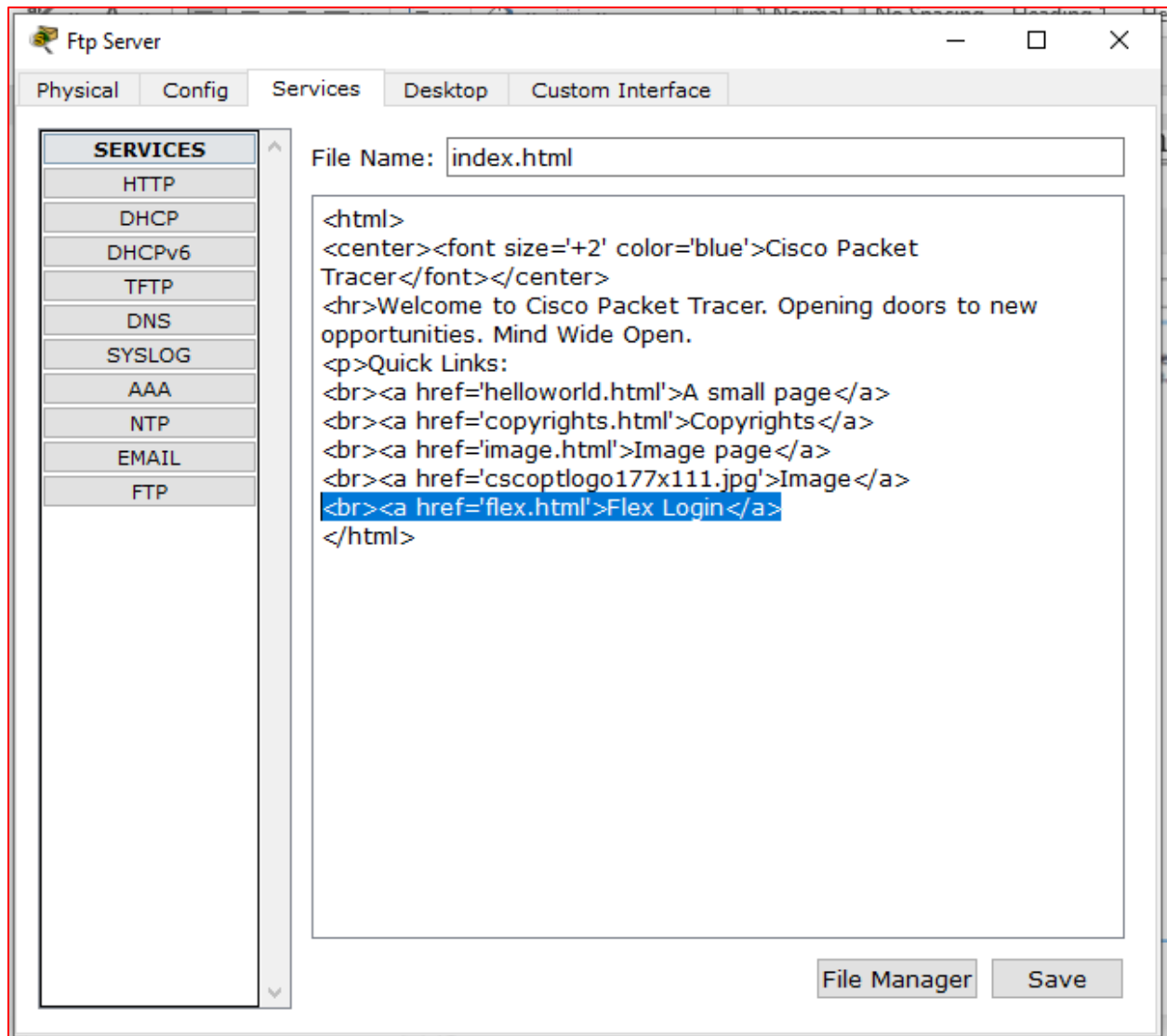
1916 bytes copied in 0.08 secs (23950 bytes/sec)
ftp>
```

Check whether the html file uploaded has been received in the **HTTP directory**:  
Go to **Server->Services-> HTTP**. Then look up for the file in the File Manager.



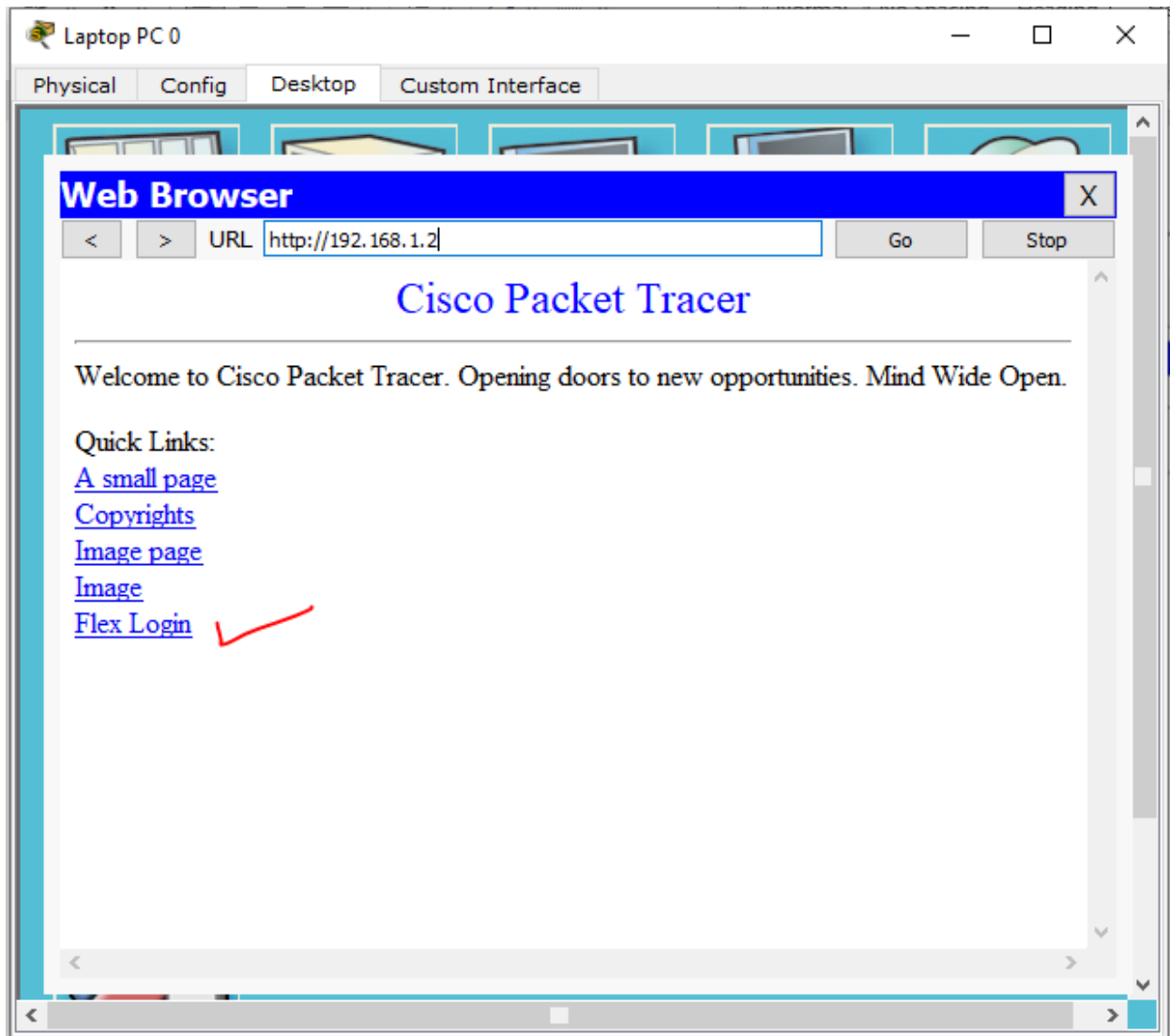


Now edit **index.html** file in the **HTTP directory** so as to include a link to **flex** that we've just uploaded. This will make **flex** accessible from the Laptop's browser. To do this, locate **index.html** then click **edit**. Proceed to edit it as shown below. Then **save** and accept overwrite.

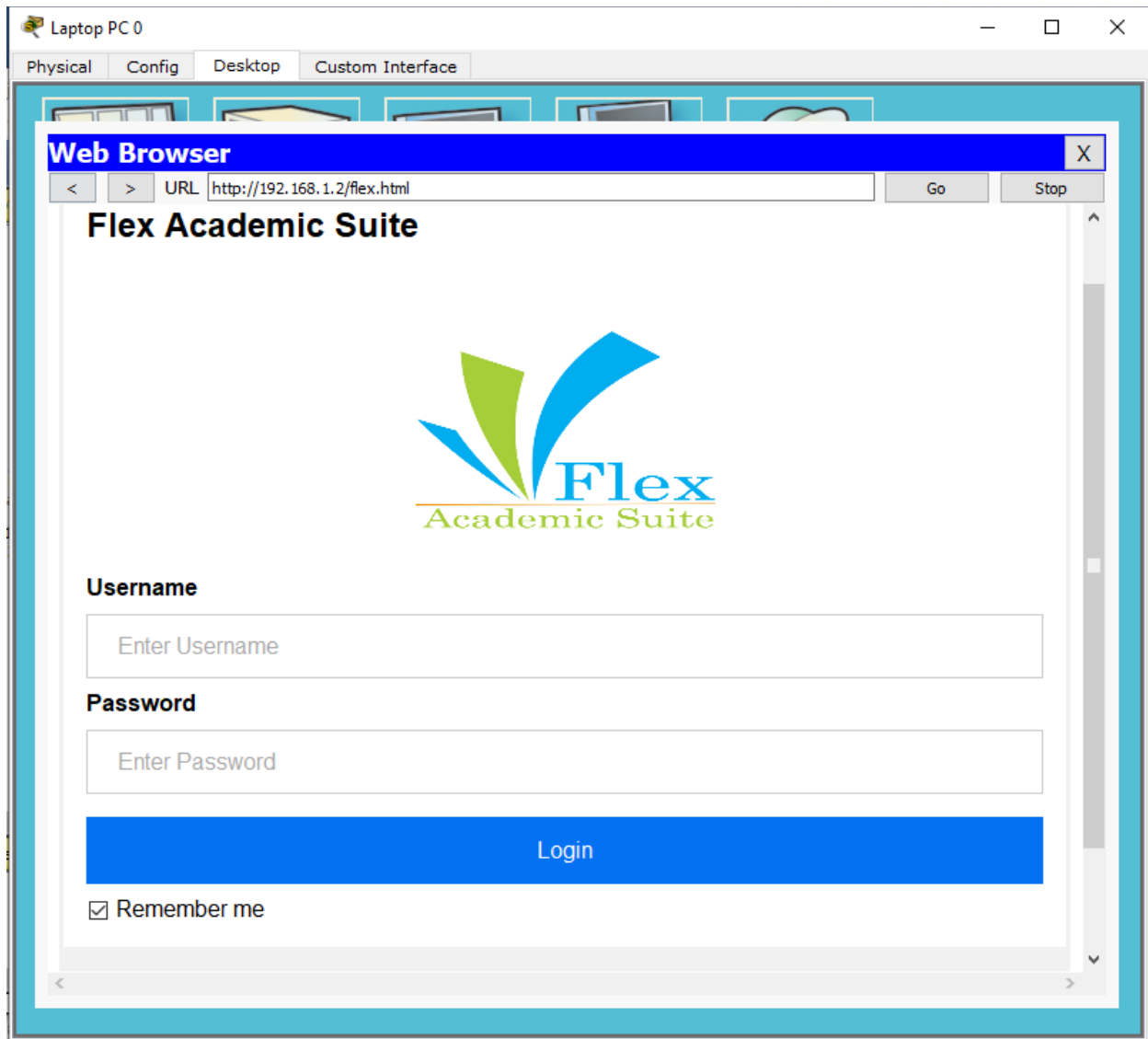


Finally, try to access the newly uploaded file from the Laptop's browser.

So go to the Laptop's browser and access the server using the server's IP address. By doing this, the browser is making an http request to the server. The server will respond to the Laptop with the **index.html** file containing a **link** to **flex** which we've uploaded from the Laptop using FTP.



Click **flex** link to view the contents of the file in the browser.



### **Tasks for students:**

1. Configure an email server in Packet Tracer
2. Configure an FTP server in Packet Tracer
3. Create and Upload html file to HTTP server directory Using FTP
4. Configure Mail server, Ftp Server, DHCP Server, DNS Server and web Server in a single topology, use router and switch.

**Note:** Submit the following: PDF and Packet tracer files