<epam>

Scripts

- What is script
- Why we need scripts
- How to create scripts

TRAINING

CENTER

<epam>

What is script

Script – a text file with the list of consecutively executed commands.

Each shell script is saved with **.sh** file extension

By default, your OS considers files with this extension to be plain text files, so in order to make it executable, you need to add an executable flag (x) to file permissions. For that purposes use the command:

chmod +x ./filename.sh

What is script

There are two ways for running scripts in Linux:

- Running in the current shell process:
  Command: **./filename.sh** or **<fullpath>/filename.sh**

- Running in a new shell process:
  Command: bash **filename.sh** or **bash <fullpath>/filename.sh**

What is script

For changing execution binary for your script you can put a **shebang expression** to the top of the file

```
[root@server vagrant]# cat script_sample.sh
#!/bin/bash

Recipient=admin@example.com
Subject=Greeting
Message=Welcome to our site
`mail -s $Subject $Recipient <<< $Message`
[root@server vagrant]#
```

```
[root@server vagrant]# cat script_python_sample.sh
#! /usr/bin/env python

def hello_command(name, print_counter=False, repeat=10):
    """Print nice greetings."""
    for i in range(repeat):
        if print_counter:
            print i+1,
        print 'Hello, %s!' % name

if __name__ == '__main__':
    import scriptine
    scriptine.run()
[root@server vagrant]#
```

Why we need scripts

Scripts help up with:

- Avoiding repetitive work
- Reducing input errors
- Adding automation features

Script usecases

- Installing and updating packages
- Adding new users and groups
- Database backup and restore
- Mount new volumes
- System monitoring

How to create script

To sum up, there are steps for creating shell script:

- Create file with **.sh** extension
- Add a shebang in top of the file
- Write some code
- Save the file
- Set the executable flag in file permissions

Create sample script

Script for creating directory with input variable

```
# !/bin/bash

echo "Enter dir name"
read dirname
mkdir $dirname
```