

Writing simple scripts to automate tasks in centos

Topics : [Centos Server](#)

Written on [March 05, 2024](#)

To automate tasks on CentOS using shell scripts, you can write simple scripts to perform repetitive tasks efficiently. Here's an example of writing a simple script to automate the backup of a directory:

```
#!/bin/bash

# Define variables
backup_dir="/path/to/backup"
source_dir="/path/to/source"

# Create backup directory if it doesn't exist
mkdir -p "$backup_dir"

# Timestamp for backup file
timestamp=$(date +"%Y-%m-%d_%H-%M-%S")

# Backup source directory
tar -czf "$backup_dir/backup_${timestamp}.tar.gz" "$source_dir"

# Check if backup was successful
if [ $? -eq 0 ]; then
    echo "Backup completed successfully."
else
    echo "Backup failed. Check logs for details."
fi
```

Save this script to a file, e.g., `backup_script.sh`, and make it executable using `chmod +x backup_script.sh`. Then, you can run it with `./backup_script.sh` to execute the backup process.

Explanation of the script:

1. **Shebang Line:** `#!/bin/bash` specifies the shell interpreter to use (Bash).
2. **Variables:** `backup_dir` and `source_dir` define the backup directory and source directory paths.
3. **Create Backup Directory:** `mkdir -p "$backup_dir"` creates the backup directory if it doesn't exist.

4. **Timestamp:** `timestamp=$(date +%Y-%m-%d_%H-%M-%S)` generates a timestamp for the backup file.
5. **Backup Process:** `tar -czf "$backup_dir/backup_${timestamp}.tar.gz" "$source_dir"` creates a compressed tar archive of the source directory and saves it with a timestamped filename in the backup directory.
6. **Check Backup Status:** `if [$? -eq 0]; then ...` checks the exit status of the tar command. If it's 0 (success), it prints a success message; otherwise, it prints a failure message.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)

ARYATECHNO