# Code for Chapter 2

## Introduction

This chapter provides essential scripts and configurations relevant to managing files, directories, and cloud-based storage in a Rocky Linux environment. The provided scripts automate common administrative tasks, ensuring efficiency and accuracy.

## 1. Creating and Managing Files and Directories

### Script: create\_files.sh

***#!/bin/bash***

***# Define variables***

***DIR\_NAME="project\_data"***

***FILE\_NAME="report.txt"***

***LINK\_TARGET="/var/log/syslog"***

***LINK\_NAME="log\_link"***

***echo "Starting script execution..."***

***# Create directory if it doesn't exist***

***if [ ! -d "$DIR\_NAME" ]; then***

***mkdir "$DIR\_NAME"***

***chmod 755 "$DIR\_NAME"***

***echo " Directory '$DIR\_NAME' created with permissions: drwxr-xr-x"***

***else***

***echo " Directory '$DIR\_NAME' already exists."***

***fi***

***# Create and configure file inside the directory***

***FULL\_PATH="$DIR\_NAME/$FILE\_NAME"***

***touch "$FULL\_PATH"***

***chmod 644 "$FULL\_PATH"***

***truncate -s 1024 "$FULL\_PATH"***

***echo " File '$FULL\_PATH' created with permissions: -rw-r--r-- and size: 1024 bytes"***

***# Create symbolic link if target exists***

***if [ -e "$LINK\_TARGET" ]; then***

***ln -s "$LINK\_TARGET" "$LINK\_NAME"***

***echo "Symbolic link '$LINK\_NAME' created pointing to '$LINK\_TARGET'"***

***else***

***echo "Warning: Target '$LINK\_TARGET' does not exist. Symbolic link '$LINK\_NAME' will be broken."***

***fi***

***# Display summary***

***echo -e "\n File and Directory Details:"***

***ls -lh "$DIR\_NAME"***

***[ -e "$LINK\_NAME" ] && ls -l "$LINK\_NAME"***

***echo -e "\n Script execution completed."***

ls -l directory\_name file\_name.txt link\_name

### Explanation:

- Checks if a directory exists and creates it with `drwxr-xr-x` permissions if necessary.  
- Creates a text file with appropriate read/write permissions (`-rw-r--r--`).  
- Ensures a specific file size using the `truncate` command.  
- Creates a symbolic link only if the target file exists.  
- Lists created files and their permissions for verification.

## 2. Uploading Files to an S3 Bucket

### Script: upload\_to\_s3.sh

#!/bin/bash  
# Define bucket name (fixed as "bobby-fisher")  
BUCKET\_NAME="bobby-fisher"  
  
# Check if the file name is provided  
if [ "$#" -ne 1 ]; then  
 echo "Usage: $0 <file-name>"  
 exit 1  
fi  
  
# Assign the file name from the argument  
FILE\_NAME=$1  
  
# Check if the file exists  
if [ ! -f "$FILE\_NAME" ]; then  
 echo "Error: File '$FILE\_NAME' does not exist."  
 exit 1  
fi  
  
# Create the S3 bucket (fixed bucket name)  
echo "Creating S3 bucket: $BUCKET\_NAME..."  
aws s3 mb s3://$BUCKET\_NAME  
if [ $? -ne 0 ]; then  
 echo "Error: Failed to create S3 bucket."  
 exit 1  
fi  
  
# Upload the file to the S3 bucket  
echo "Uploading file: $FILE\_NAME to bucket: $BUCKET\_NAME..."  
aws s3 cp "$FILE\_NAME" s3://$BUCKET\_NAME  
if [ $? -ne 0 ]; then  
 echo "Error: Failed to upload the file."  
 exit 1  
fi

### Explanation:

- Defines a fixed S3 bucket name (`bobby-fisher`).  
- Verifies that an argument (file name) is provided.  
- Ensures the file exists before attempting to upload it.  
- Creates the specified S3 bucket.  
- Uploads the file to the bucket and checks for errors.

## Summary

The scripts presented in this chapter provide automation for file handling and cloud storage operations. These scripts help streamline Linux system administration, ensuring consistency in managing directories, files, and cloud-based storage solutions.