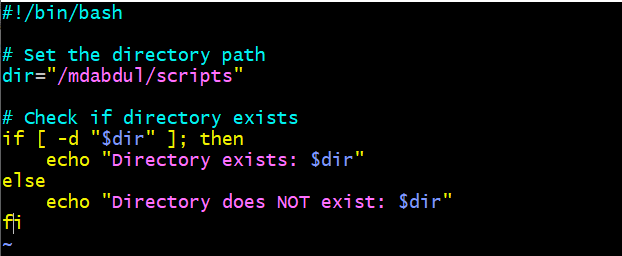
1. Create on Bash script to check if a directory is available or not.

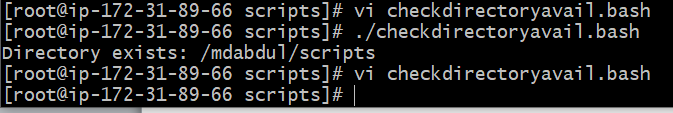
Creat a scriptile

Give permissions to script file

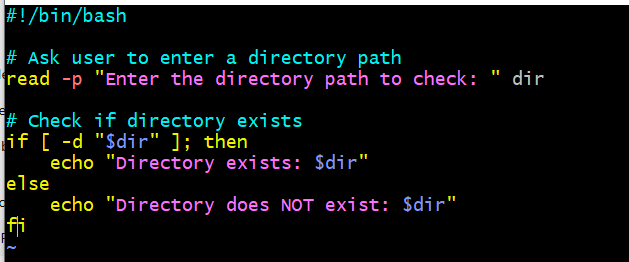
Write a script according to a suitable condition

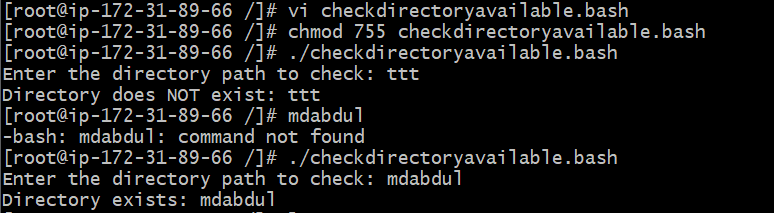


Execute the script



Or you can go the below script





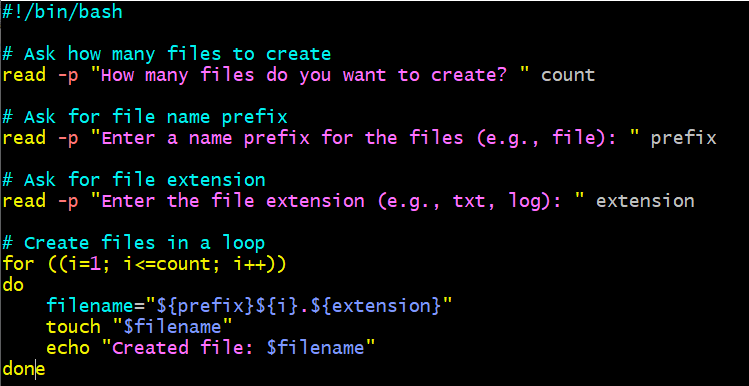
Here one thing need to research which command use to reach or to find a particular file or directory path

1. Create a bash script which will create multiple files.

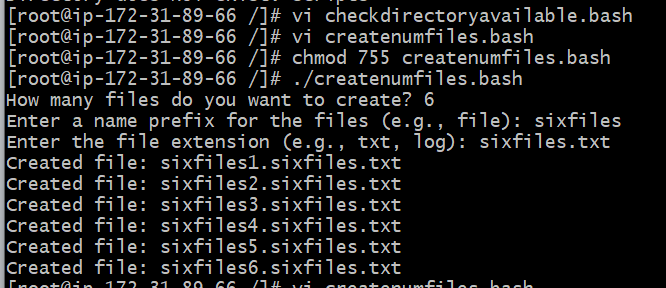
Creat a scriptile

Give permissions to script file

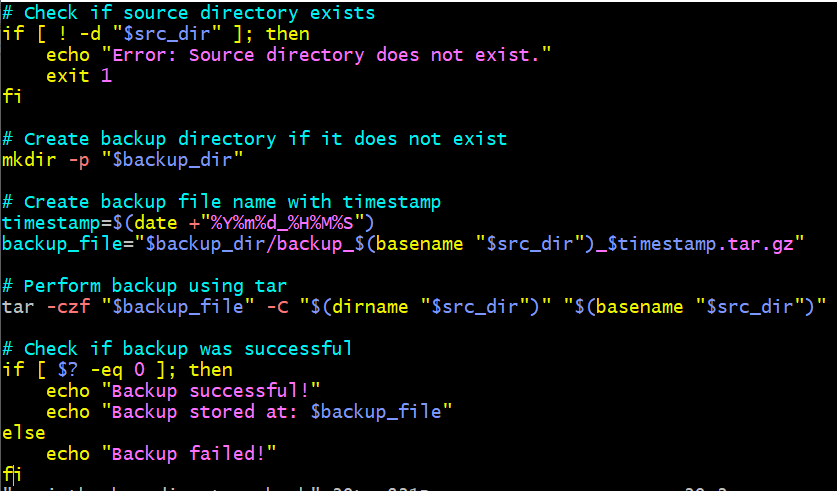
Write a script according to a suitable condition

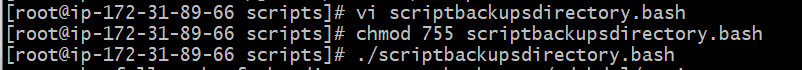


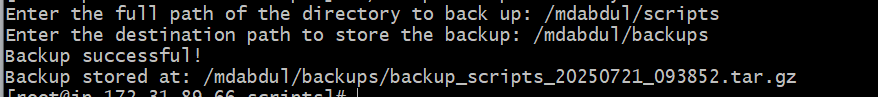
Execute the script ile



1. Create a bash script to take backup of a directory.

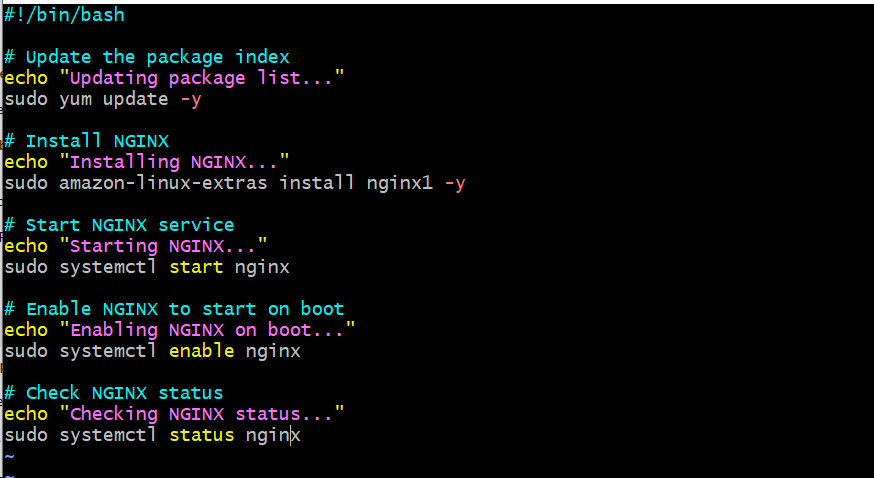




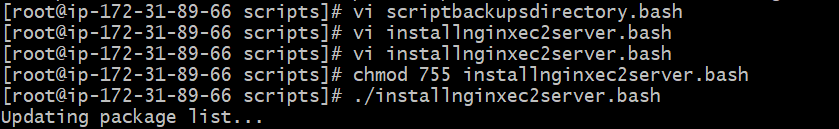


It will create automatically one directory naming backups according to a script.

1. Create a bash script to install nginx in ec2 server.



1. Create a bash script to install ApacheTomcat in ec2 server.



#!/bin/bash

# Update the package index

echo "Updating package list..."

sudo yum update -y

# Install NGINX

echo "Installing NGINX..."

sudo amazon-linux-extras install nginx1 -y

# Start NGINX service

echo "Starting NGINX..."

sudo systemctl start nginx

# Enable NGINX to start on boot

echo "Enabling NGINX on boot..."

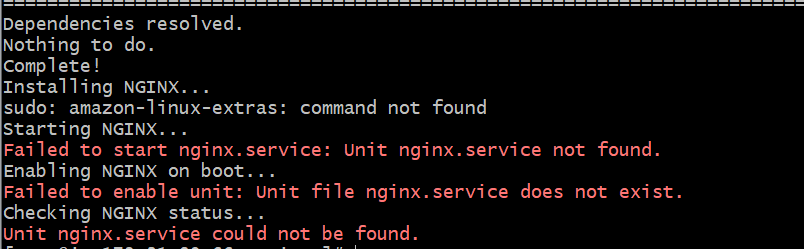
sudo systemctl enable nginx

# Check NGINX status

echo "Checking NGINX status..."

sudo systemctl status nginx

I wrote this script it gave error



Because

You're using **Amazon Linux 2023**, which **does not use amazon-linux-extras** (it was available in Amazon Linux 2). Instead, you should use dnf to install packages.

So again re write the script as below

#!/bin/bash

# Update the system

echo "Updating system..."

sudo dnf update -y

# Install NGINX using dnf

echo "Installing NGINX..."

sudo dnf install nginx -y

# Start NGINX service

echo "Starting NGINX..."

sudo systemctl start nginx

# Enable NGINX to start on boot

echo "Enabling NGINX to start on boot..."

sudo systemctl enable nginx

# Check status

echo "Checking NGINX status..."

sudo systemctl status nginx

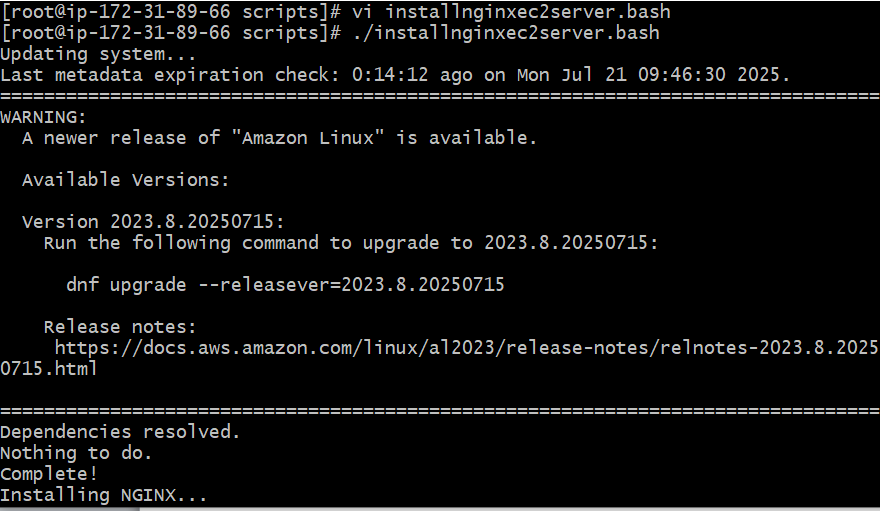
**what is dnf**

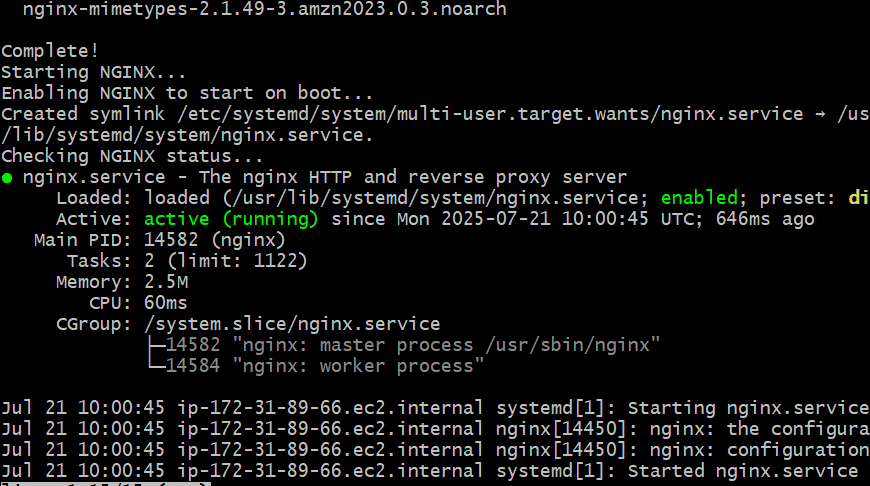
dnf stands for **Dandified YUM** — it is a **modern package manager** for **RPM-based Linux distributions**, such as:

* Amazon Linux 2023
* Fedora
* Red Hat Enterprise Linux (RHEL) 8+
* CentOS 8+

It replaces the older yum package manager and offers better performance, dependency resolution, and features.

Execute





1. Create a bash script to check list if nginx service is running or not,if not running then script should start the service.

#!/bin/bash

# Check the status of nginx service

if systemctl is-active --quiet nginx; then

echo "Nginx service is already running."

else

echo "Nginx service is not running. Starting the service..."

sudo systemctl start nginx

# Re-check if it started successfully

if systemctl is-active --quiet nginx; then

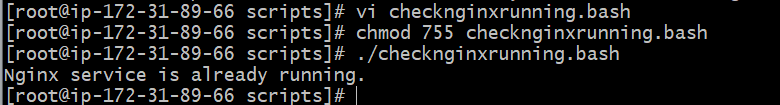
echo "Nginx service started successfully."

else

echo "Failed to start Nginx service."

fi

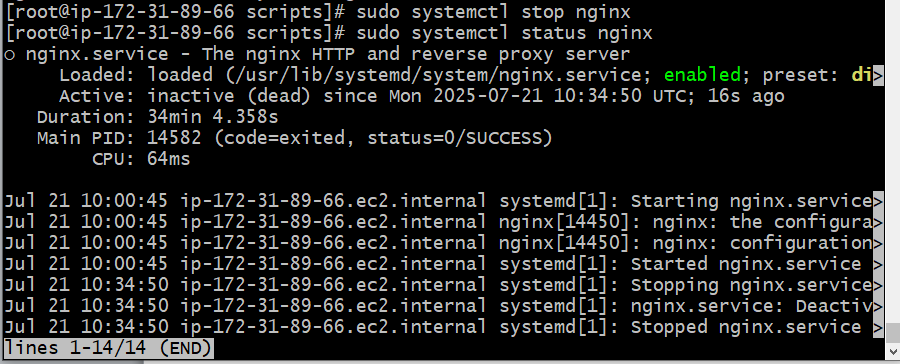
fi



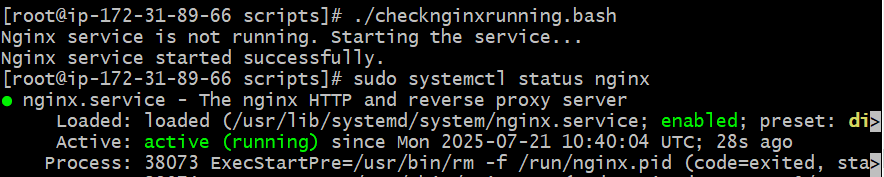
Let stop nginx and execute the same script again

**sudo systemctl stop nginx**

**sudo systemctl status nginx**



**Now execute the above script again**



1. Create a bash script for calculator.

#!/bin/bash

# Function to perform calculation

calculate() {

case $2 in

+)

result=$(echo "$1 + $3" | bc)

;;

-)

result=$(echo "$1 - $3" | bc)

;;

\\*)

result=$(echo "$1 \* $3" | bc)

;;

/)

if [ "$3" == "0" ]; then

echo "Error: Division by zero is not allowed."

exit 1

fi

result=$(echo "scale=2; $1 / $3" | bc)

;;

\*)

echo "Invalid operator. Use +, -, \*, or /"

exit 1

;;

esac

echo "Result: $result"

}

# User input

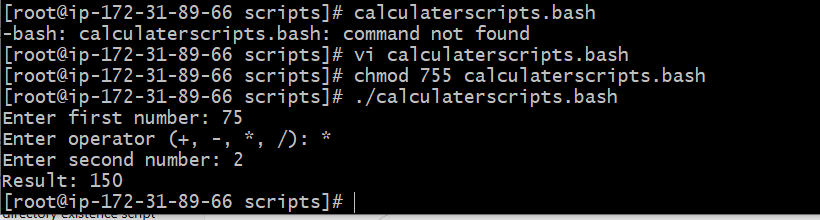
read -p "Enter first number: " num1

read -p "Enter operator (+, -, \*, /): " op

read -p "Enter second number: " num2

# Call the function

calculate "$num1" "$op" "$num2"



1. Create a bash script to check if directory is avaialble or not,if not then create a directory.

#!/bin/bash

# Ask user to enter a directory path

read -p "Enter the directory path: " dir

# Check if directory exists

if [ -d "$dir" ]; then

echo "Directory already exists: $dir"

else

echo "Directory does not exist. Creating now..."

mkdir -p "$dir"

# Confirm if directory was created

if [ -d "$dir" ]; then

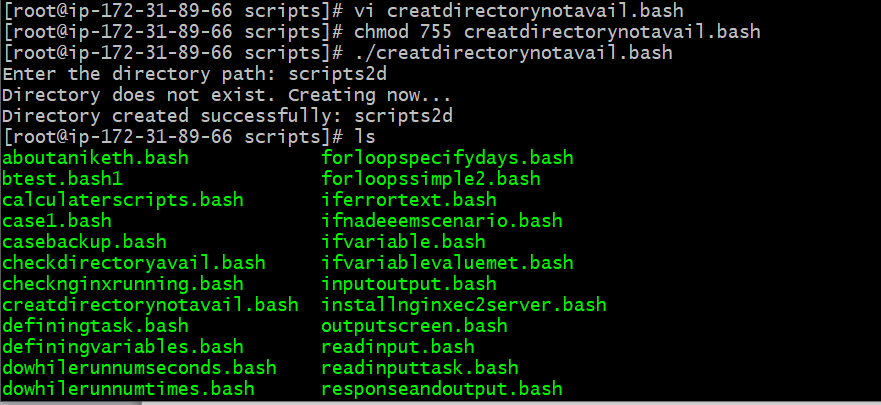
echo "Directory created successfully: $dir"

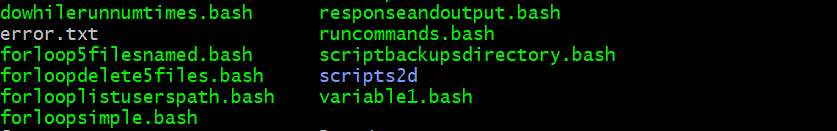
else

echo "Failed to create directory: $dir"

fi

fi





1. Create bash script to delete last 3 lines for a file.

#!/bin/bash

# Ask user for the filename

read -p "Enter the file path: " file

# Check if file exists

if [ ! -f "$file" ]; then

echo "File does not exist!"

exit 1

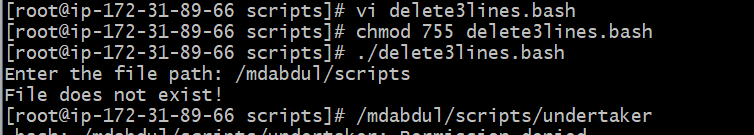
fi

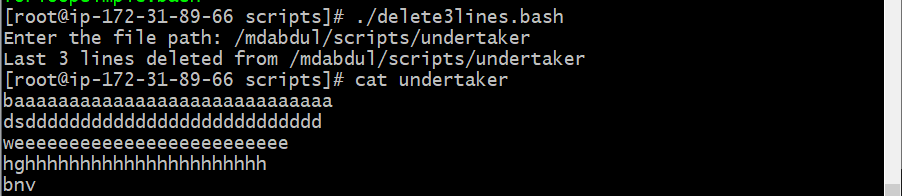
# Delete the last 3 lines and save the result

# This will overwrite the original file

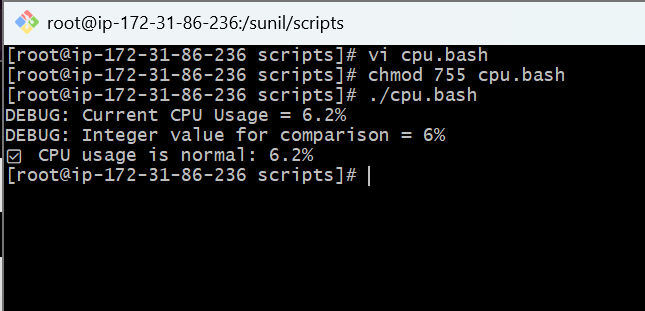
head -n -3 "$file" > temp\_file && mv temp\_file "$file"

echo "Last 3 lines deleted from $file"

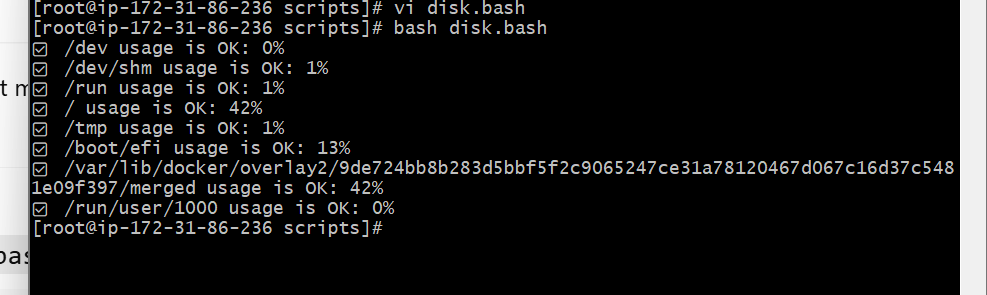




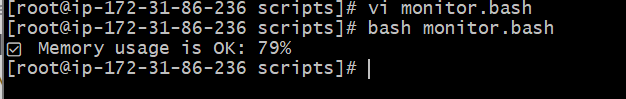
1. Bash script to monitor cpu and if it is more than 80% then send email notification.



1. Bash script to monitor disk space and if it is more than 80% then send email notification.



1. Bash script to monitor memory and if it is more than 80% then send email notification.



1. Exercise Crontab Entries :  
   1) April 5th Midnight  
    2) 5 th of Every November,Jan,June if it is a Thursday.  
   3) At 05 and 27th minutes of 9,10,11 hours everyday.  
   4) 34 min. of 9th hour on 15th Aug.  
   5) Every midnight  
   6) Every Weekend ( Saturday night 11.59 )  
   7) After every reboot  
   **Note:**  
   **1) All the Scripts needs to be executed on Amazon Linux Ec2.**  
   **2) Document all the task and also push the task to github reposiroty.**  
   **3) Share the document and github Repo url once done for evaluation in**  
   **dvps-cloud-status channel.**

