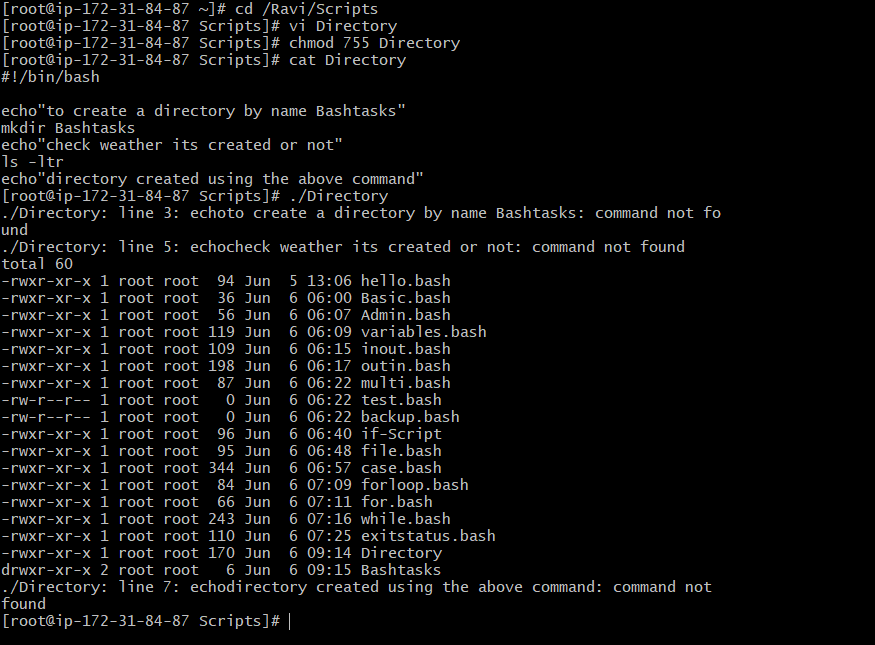
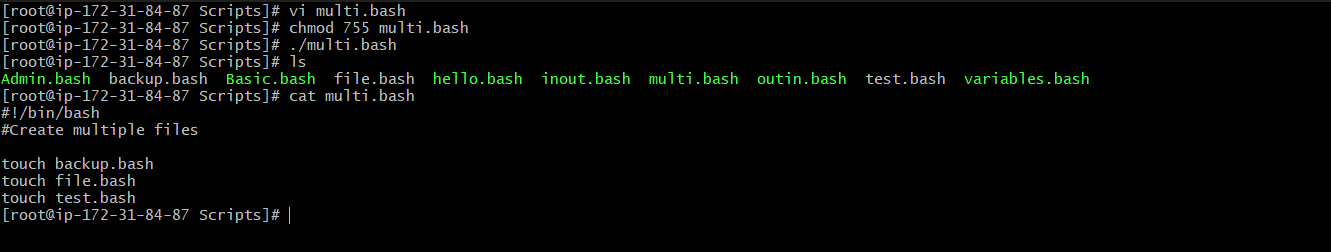
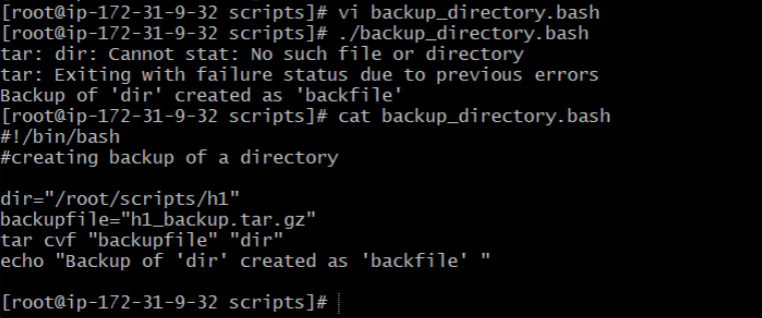
1. Execute all scripts from the [devops-cloud-documents-13](https://techiehorizon.slack.com/archives/C08TMFAKPNU) channel.
2. Create a bash script to check if a directory is available or not.



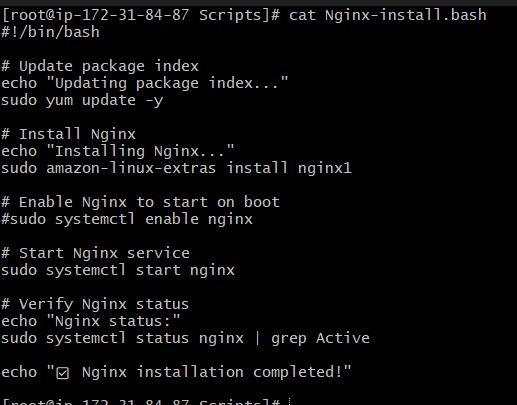
1. Create a bash script to create multiple files.

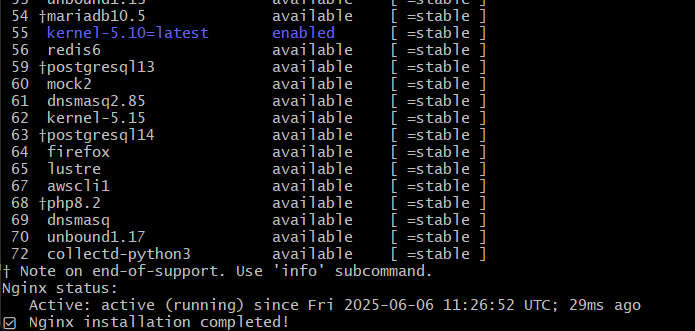


1. Create a bash script to take a backup of a directory.

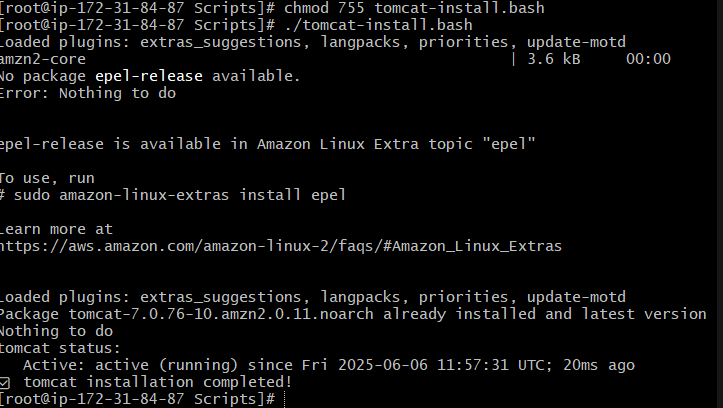


1. Create a bash script to install Nginx on an EC2 server.

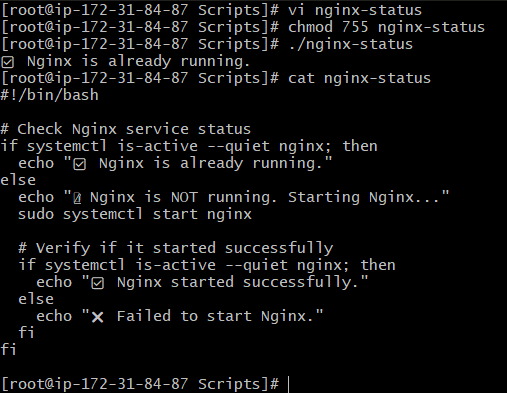




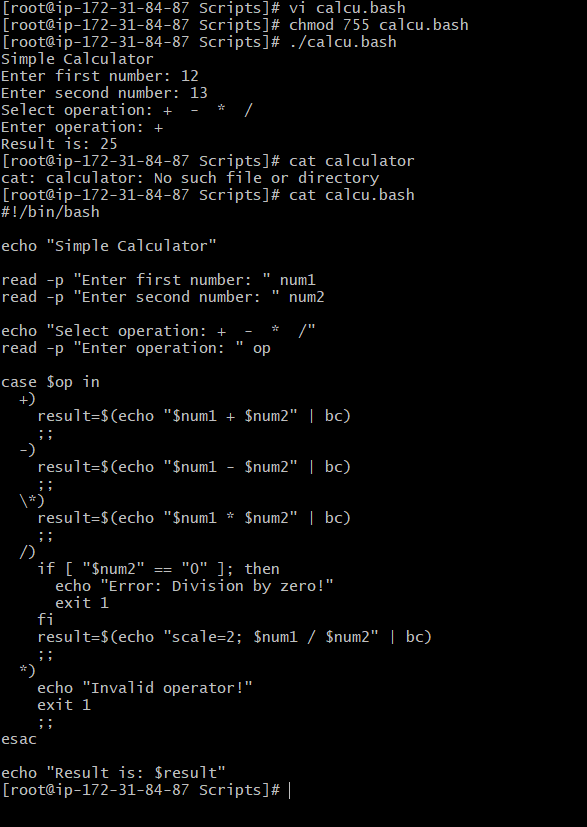
1. Create a bash script to install Apache Tomcat on an EC2 server.



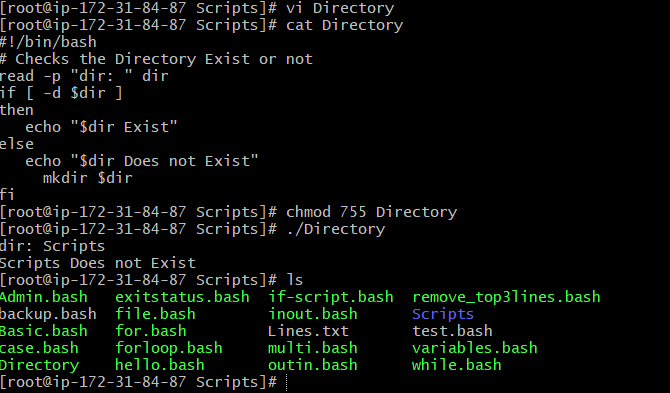
1. Create a bash script to check if the Nginx service is running, and start it if not.



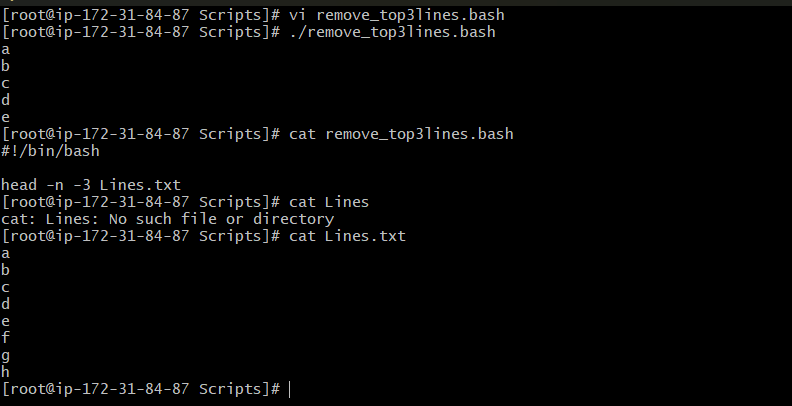
1. Create a bash script for a calculator.



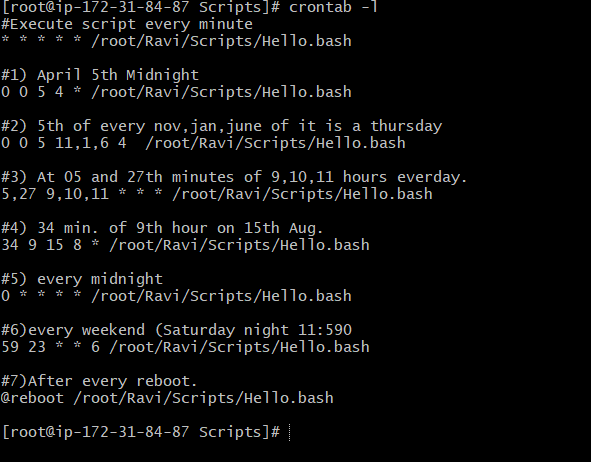
1. Create a bash script to check if a directory exists, and create it if not.



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



Crontab List



Here are the crontab entries for each of your scheduling requirements:

A crontab entry follows this format:

minute hour day\_of\_month month day\_of\_week command\_to\_execute

* **Minute (0-59)**
* **Hour (0-23)**
* **Day of Month (1-31)**
* **Month (1-12 or Jan-Dec)**
* **Day of Week (0-7, where 0 and 7 are Sunday)**

1. **April 5th Midnight**
   * Midnight is 00:00.
   * Crontab: 0 0 5 4 \*
2. **5th of Every November, Jan, June if it is a Thursday.**
   * **Important Note on day\_of\_month and day\_of\_week:** In crontab, if *both* day\_of\_month and day\_of\_week fields are specified (not using \*), the command runs when *either* condition is true. To achieve a strict "AND" (i.e., "only if the 5th is *also* a Thursday"), you typically need to add a conditional check within the command itself.
   * Here's the crontab entry that runs on the 5th of Jan, June, and Nov *and also* on every Thursday.
     + Crontab: \* \* 5 1,6,11 4
   * **To enforce the "AND" condition more strictly (recommended for precise logic):** You would put a check inside the script or command executed by cron. For example, if you wanted it to run *only* if the day is the 5th *and* it's a Thursday:

Bash

# This part goes into the cron job:

# \* \* 5 1,6,11 \* /path/to/your\_script.sh

# And inside your\_script.sh, you'd add:

#!/bin/bash

# Check if today is Thursday (day of week 4, where Monday is 1)

if [ "$(date +\%u)" -eq 4 ]; then

# Your actual command goes here

/usr/bin/your\_actual\_command

fi

1. **At 05 and 27th minutes of 9, 10, 11 hours everyday.**
   * Crontab: 5,27 9,10,11 \* \* \*
2. **15th second of 34 min. of 9th hour on 15th Aug.**
   * **Important Note on Seconds:** Standard crontab does **not** support scheduling by seconds. The smallest time unit is the minute.
   * If you meant at the 34th minute of the 9th hour on August 15th:
     + Crontab: 34 9 15 8 \*
   * To execute something at a specific second, you would need a different tool (like systemd timers with OnCalendar which supports seconds, or a script that loops for the first minute and then executes at the desired second).
3. **Every midnight**
   * Midnight is 00:00.
   * Crontab: 0 0 \* \* \*
4. **Every Weekend (Saturday night 11.59)**
   * Saturday night 11:59 PM is 23:59.
   * Saturday is day of week 6.
   * Crontab: 59 23 \* \* 6
5. **After every reboot**
   * Crontab: @reboot