Application Health Checker

1. Purpose

A small tool that checks whether an application and its host system are healthy and notifies you when something needs attention.

2. Quick facts

What it does: Runs simple health checks (service up, HTTP response).

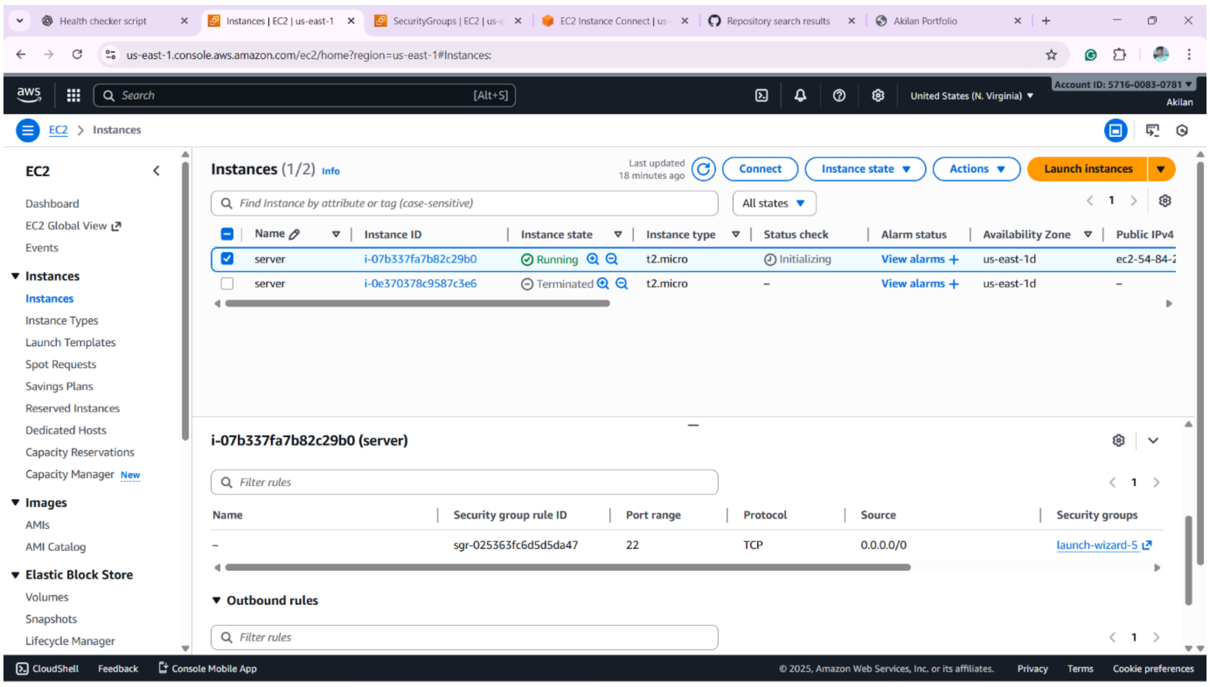
How it runs: As a script scheduled by cron or run manually.

Output: Console logs, optional email/slack alert, and a brief log file.

Prerequisites

- A Linux server
- Bash installed
- Access to the application (URL or port)

Step1: Create an EC2 instance and copy the public IP.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
server	i-07b337fa7b82c29b0	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-54-84-2
server	i-0e370378c9587c3e6	Terminated	t2.micro	–	View alarms +	us-east-1d	–

i-07b337fa7b82c29b0 (server)

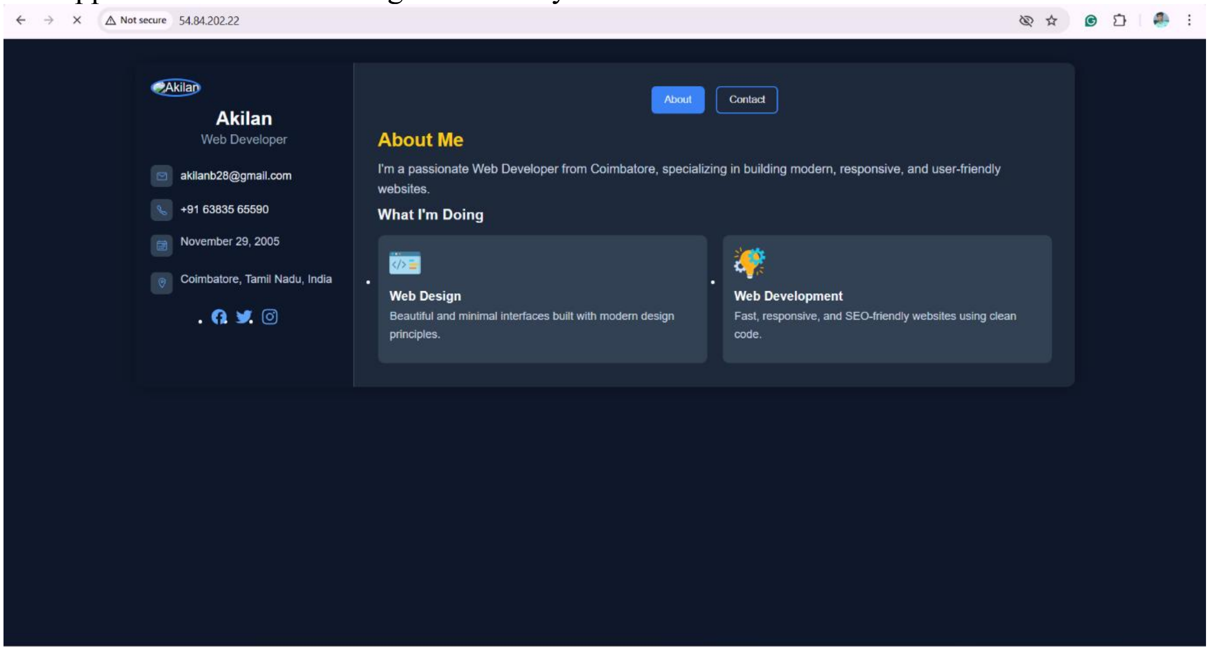
Filter rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
–	sgr-025363fc6d5d5da47	22	TCP	0.0.0.0/0	launch-wizard-5

Outbound rules

Filter rules

Our application is now running successfully.



Akilan
Web Developer

akilanb28@gmail.com
+91 63835 65590
November 29, 2005
Coimbatore, Tamil Nadu, India

About Me

I'm a passionate Web Developer from Coimbatore, specializing in building modern, responsive, and user-friendly websites.

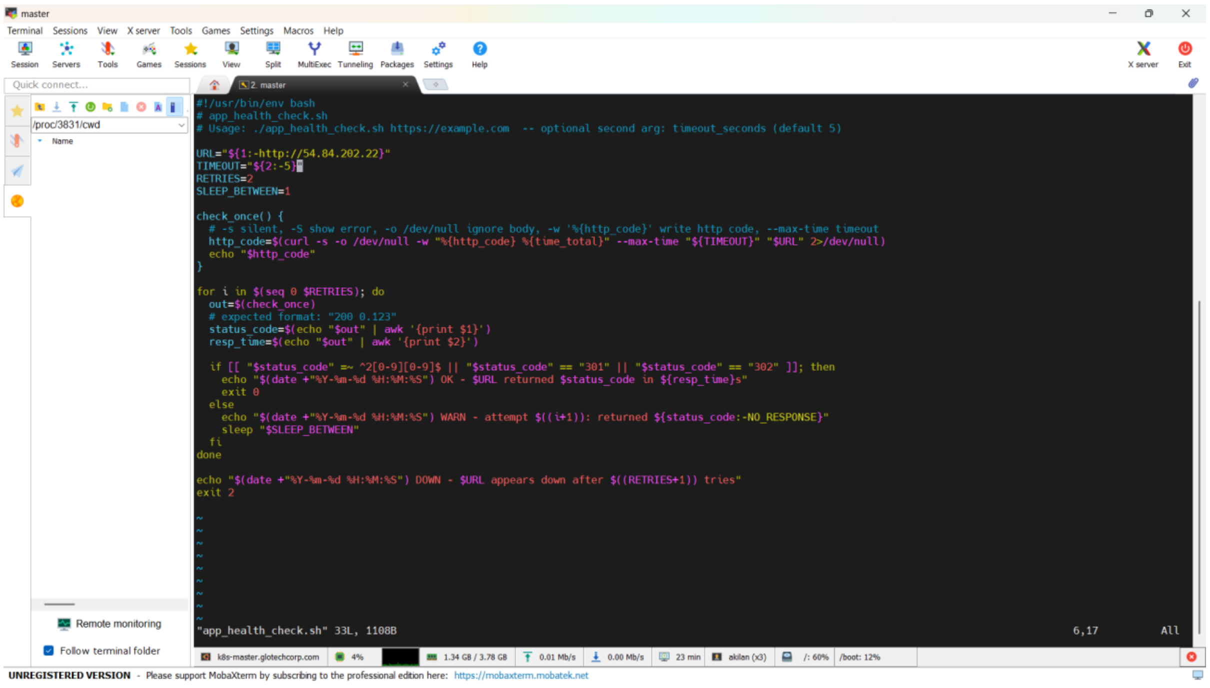
What I'm Doing

Web Design
Beautiful and minimal interfaces built with modern design principles.

Web Development
Fast, responsive, and SEO-friendly websites using clean code.

Step 2: Create the script file

vi app.sh


master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
/proc/3831/cwd
Name
2. master
Remote monitoring
Follow terminal folder

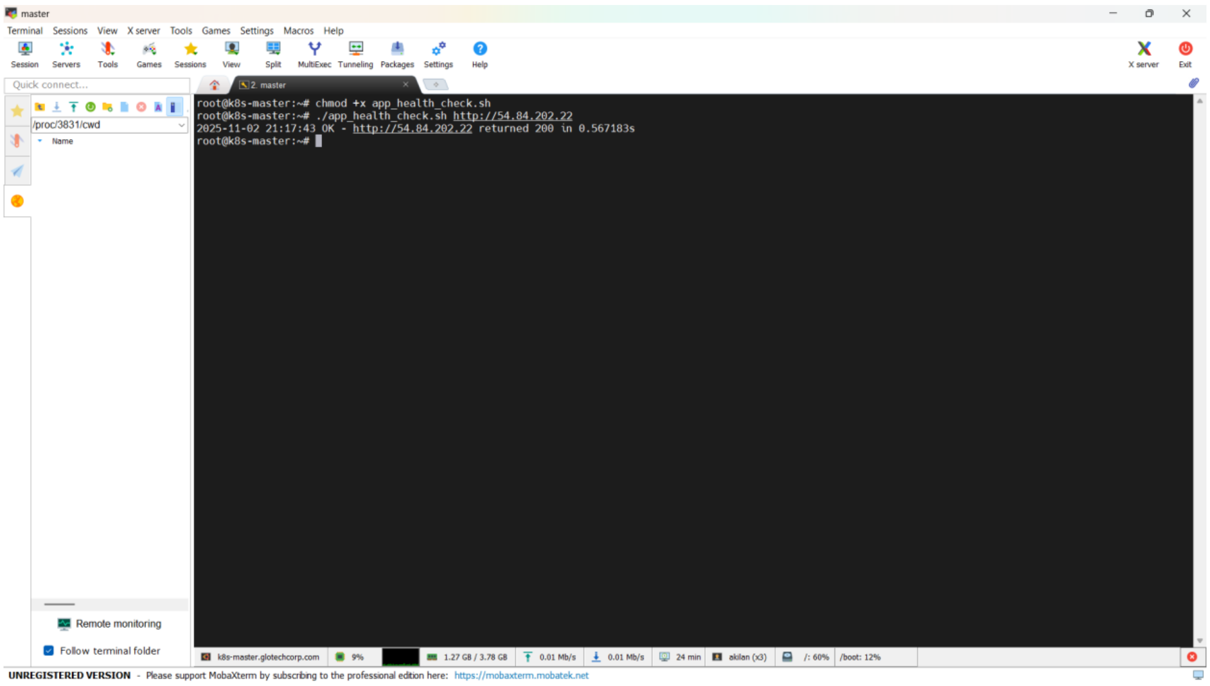
```
#!/usr/bin/env bash
# app_health_check.sh
# Usage: ./app_health_check.sh https://example.com  -- optional second arg: timeout_seconds (default 5)

URL="${1:-http://54.84.202.22}"
TIMEOUT="${2:-5}"
RETRIES=2
SLEEP_BETWEEN=1

check_once() {
  # -s silent, -S show error, -o /dev/null ignore body, -w '%{http_code}' write http code, --max-time timeout
  http_code=$(curl -s -o /dev/null -w "%{http_code} %{time_total}" --max-time "${TIMEOUT}" "$URL" 2>/dev/null)
  echo "$http_code"
}

for i in $(seq 0 $RETRIES); do
  out=$(check_once)
  # expected format: "200 0.123"
  status_code=$(echo "$out" | awk '{print $1}')
  resp_time=$(echo "$out" | awk '{print $2}')

  if [[ "$status_code" =~ ^2[0-9][0-9]$ || "$status_code" == "301" || "$status_code" == "302" ]]; then
    echo "$(date +"%Y-%m-%d %H:%M:%S") OK - $URL returned $status_code in ${resp_time}s"
    exit 0
  else
    echo "$(date +"%Y-%m-%d %H:%M:%S") WARN - attempt $((i+1)): returned ${status_code:-NO_RESPONSE}"
    sleep "$SLEEP_BETWEEN"
  fi
done

echo "$(date +"%Y-%m-%d %H:%M:%S") DOWN - $URL appears down after $((RETRIES+1)) tries"
exit 2

~
~
~
~
~
~
~
~
~
"app_health_check.sh" 33L, 1108B                                    6,17          All
```

k8s-master.glotechcorp.com   4%   1.34 GB / 3.78 GB   0.01 Mb/s   0.00 Mb/s   23 min   akilan (x3)   /: 60%   /boot: 12%

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Step 3: Run the script

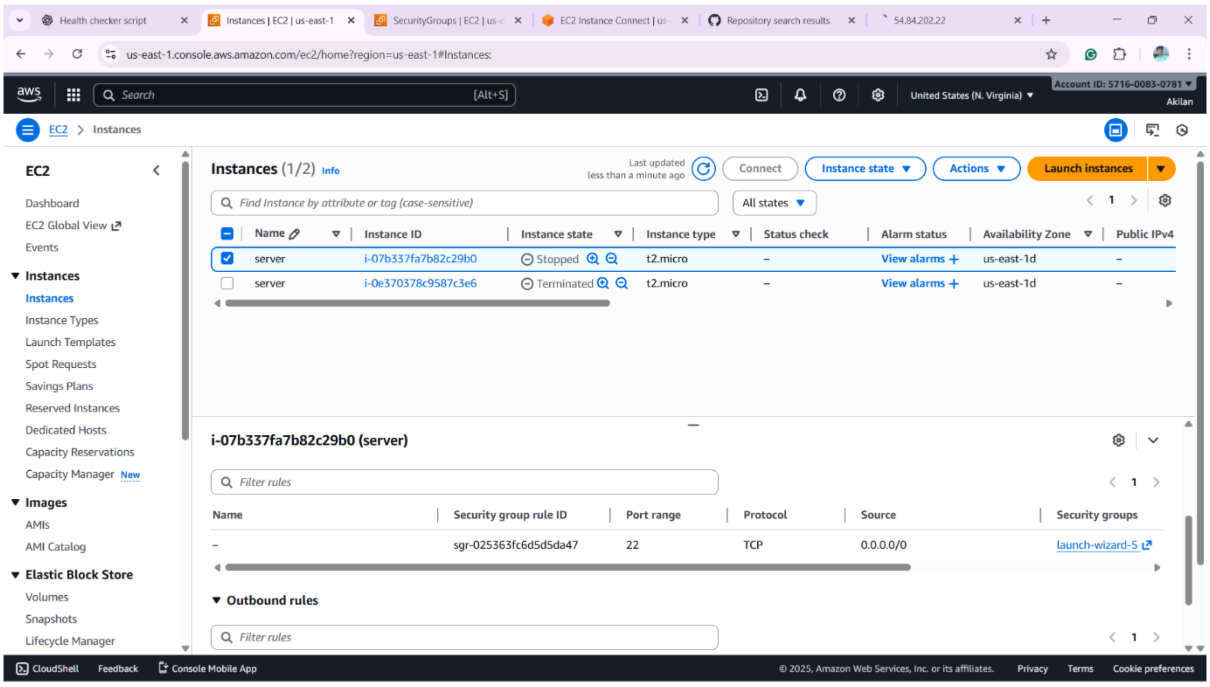
You will see the result as **OK**, indicating that the application is up and running.



```
root@k8s-master:~# chmod +x app_health_check.sh
root@k8s-master:~# ./app_health_check.sh http://54.84.202.22
2025-11-02 21:17:43 OK - http://54.84.202.22 returned 200 in 0.567183s
root@k8s-master:~#
```

master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. master
/proc/3831/cwd
Name
Remote monitoring
Follow terminal folder
k8s-master.glotechcorp.com
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Step 4: Stop the EC2 instance.



Instances (1/2)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
✓	server	i-07b337fa7b82c29b0	Stopped	t2.micro	–	View alarms +	us-east-1d	–
	server	i-0e370378c9587c3e6	Terminated	t2.micro	–	View alarms +	us-east-1d	–

i-07b337fa7b82c29b0 (server)

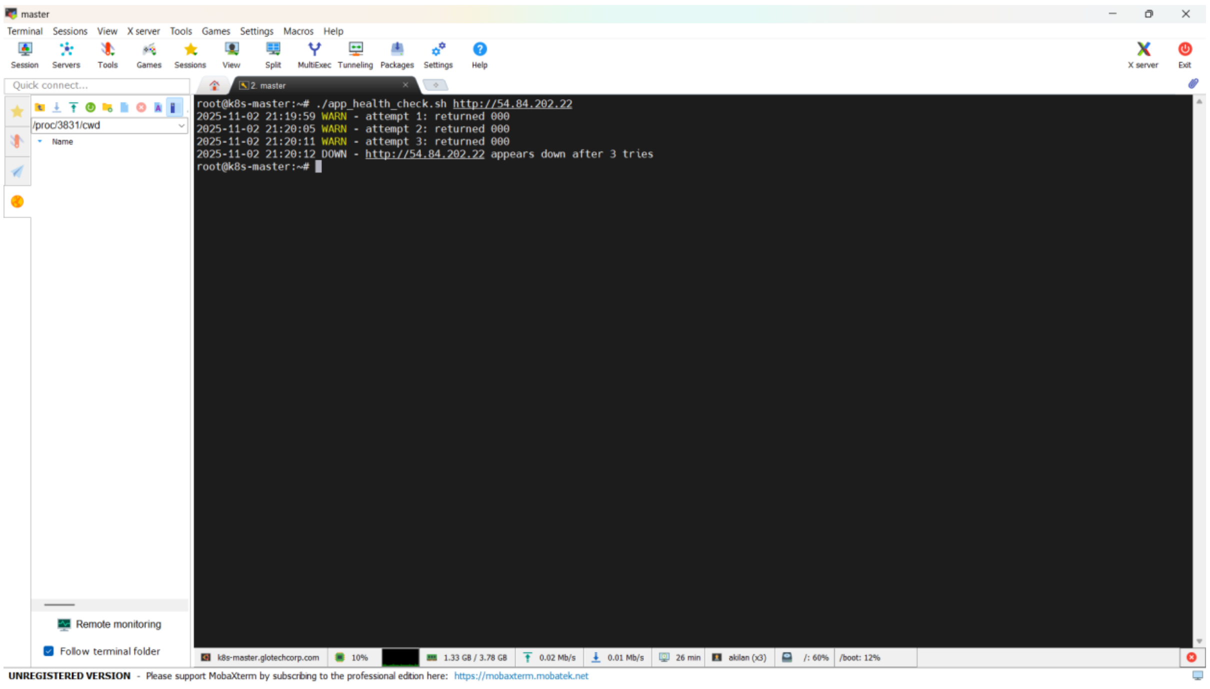
Name	Security group rule ID	Port range	Protocol	Source	Security groups
–	sgr-025363fc6d5d5da47	22	TCP	0.0.0.0/0	launch-wizard-5

Outbound rules

Health checker script
Instances | EC2 | us-east-1
SecurityGroups | EC2 | us-e
EC2 Instance Connect | us-
Repository search results
54.84.202.22
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
Search [Alt+S]
United States (N. Virginia)
Account ID: 5716-0083-0781
Akilan
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Filter rules
CloudShell Feedback Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 5: Check the application again.

It will now show **DOWN**, indicating the application is not reachable.



```
root@k8s-master:~# ./app_health_check.sh http://54.84.202.22
2025-11-02 21:19:59 WARN - attempt 1: returned 000
2025-11-02 21:20:05 WARN - attempt 2: returned 000
2025-11-02 21:20:11 WARN - attempt 3: returned 000
2025-11-02 21:20:12 DOWN - http://54.84.202.22 appears down after 3 tries
root@k8s-master:~#
```

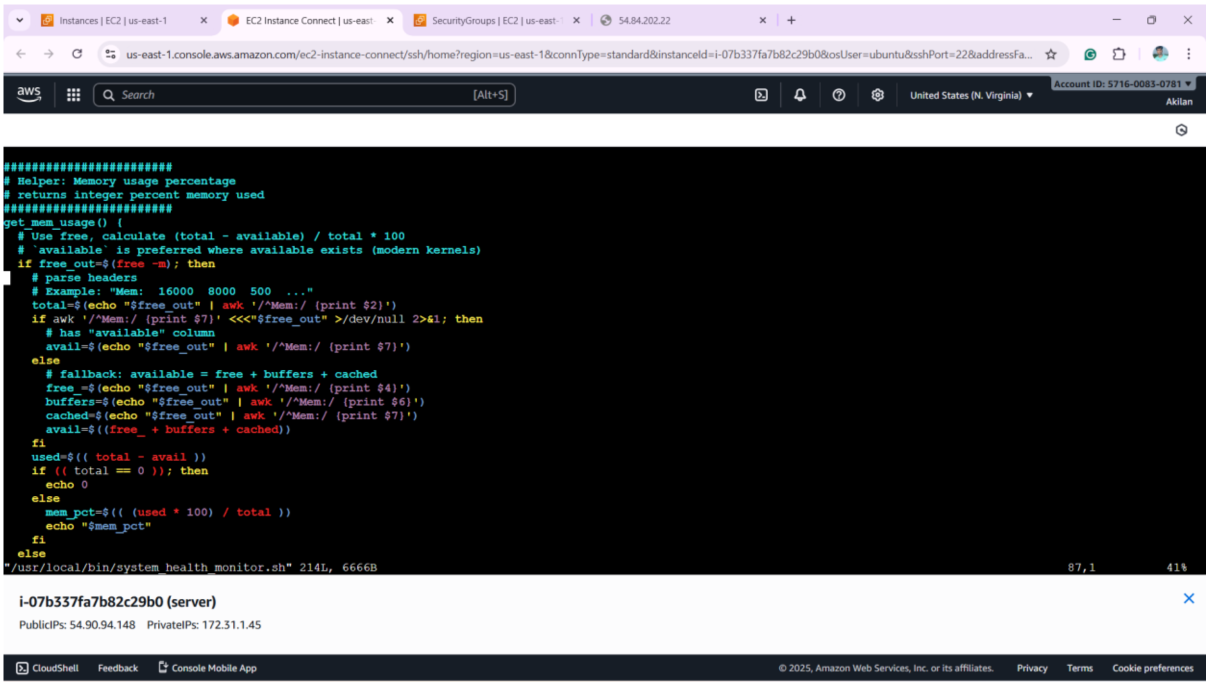
master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. master
/proc/3831/cwd
Name
Remote monitoring
Follow terminal folder
k8s-master.glotechcorp.com 10% 1.33 GB / 3.78 GB 0.02 Mb/s 0.01 Mb/s 26 min akilan (x3) /: 60% /boot: 12%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

System Health Monitoring Script:

To monitor system performance (CPU, memory, disk usage, and specific processes), create the following script:

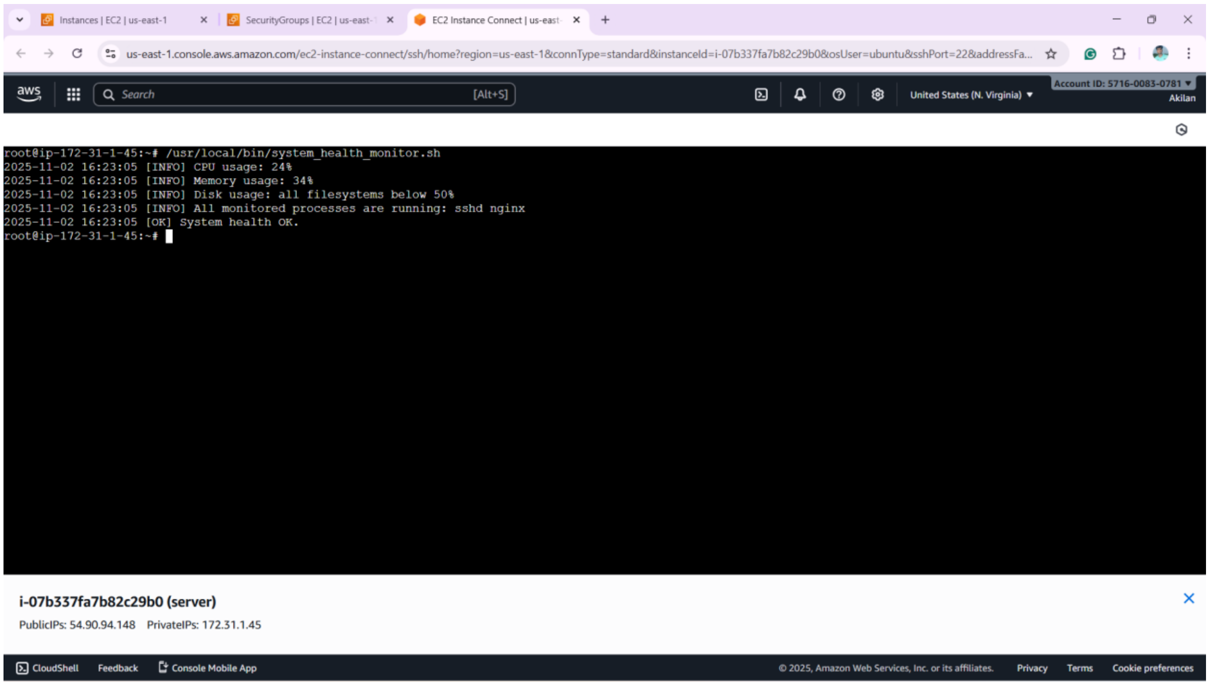
vi /usr/local/bin/system_health_monitor.sh

Paste the bash code inside the file.



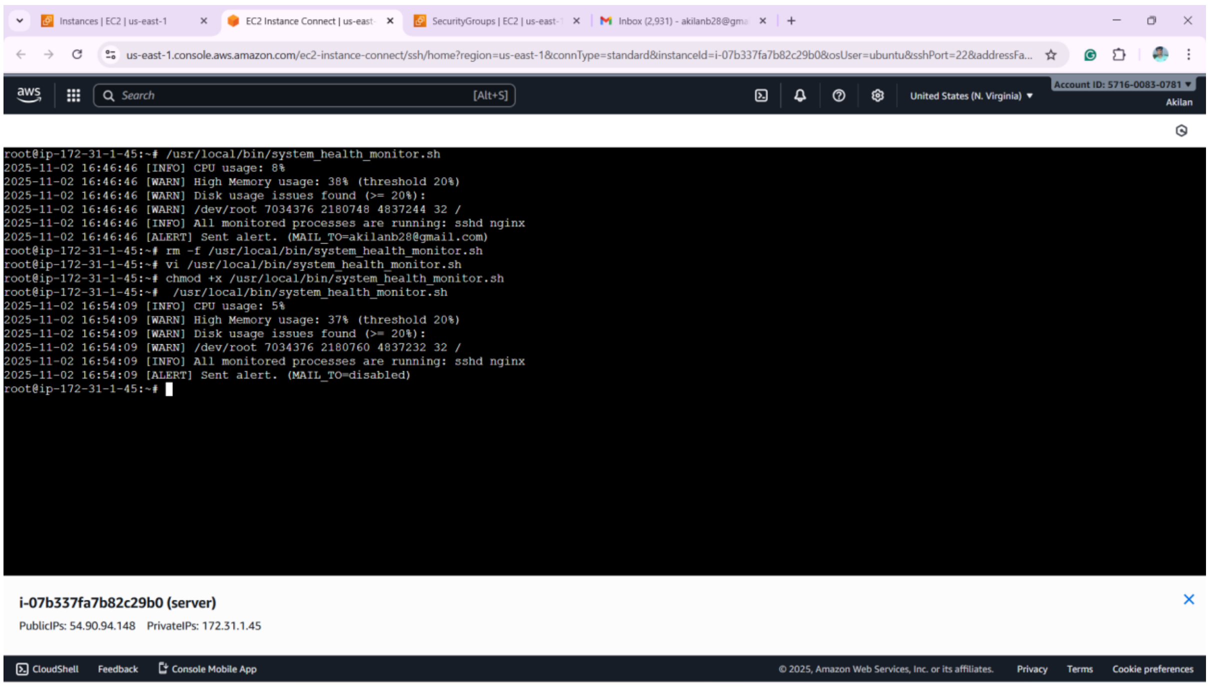
```
########################
# Helper: Memory usage percentage
# returns integer percent memory used
########################
get_mem_usage() {
  # Use free, calculate (total - available) / total * 100
  # `available` is preferred where available exists (modern kernels)
  if free_out=$(free -m); then
    # parse headers
    # Example: "Mem:  16000  8000  500  ..."
    total=$(echo "$free_out" | awk '/^Mem:/ {print $2}')
    if awk '/^Mem:/ {print $7}' <<<"$free_out" >/dev/null 2>&1; then
      # has "available" column
      avail=$(echo "$free_out" | awk '/^Mem:/ {print $7}')
    else
      # fallback: available = free + buffers + cached
      free_=$(echo "$free_out" | awk '/^Mem:/ {print $4}')
      buffers=$(echo "$free_out" | awk '/^Mem:/ {print $6}')
      cached=$(echo "$free_out" | awk '/^Mem:/ {print $7}')
      avail=$((free_ + buffers + cached))
    fi
    used=$(( total - avail ))
    if (( total == 0 )); then
      echo 0
    else
      mem_pct=$(( (used * 100) / total ))
      echo "$mem_pct"
    fi
  else
"/usr/local/bin/system_health_monitor.sh" 214L, 6666B                 87,1          41%
```

Once executed, the system health monitor script will check the configured parameters.



```
root@ip-172-31-1-45:~# /usr/local/bin/system_health_monitor.sh
2025-11-02 16:23:05 [INFO] CPU usage: 24%
2025-11-02 16:23:05 [INFO] Memory usage: 34%
2025-11-02 16:23:05 [INFO] Disk usage: all filesystems below 50%
2025-11-02 16:23:05 [INFO] All monitored processes are running: sshd nginx
2025-11-02 16:23:05 [OK] System health OK.
root@ip-172-31-1-45:~#
```

If any metric exceeds the defined threshold, it sends a warning message and an email notification to the user.



```
root@ip-172-31-1-45:~# /usr/local/bin/system_health_monitor.sh
2025-11-02 16:46:46 [INFO] CPU usage: 8%
2025-11-02 16:46:46 [WARN] High Memory usage: 38% (threshold 20%)
2025-11-02 16:46:46 [WARN] Disk usage issues found (>= 20%):
2025-11-02 16:46:46 [WARN] /dev/root 7034376 2180748 4837244 32 /
2025-11-02 16:46:46 [INFO] All monitored processes are running: sshd nginx
2025-11-02 16:46:46 [ALERT] Sent alert. (MAIL_TO=akilanb28@gmail.com)
root@ip-172-31-1-45:~# rm -f /usr/local/bin/system_health_monitor.sh
root@ip-172-31-1-45:~# vi /usr/local/bin/system_health_monitor.sh
root@ip-172-31-1-45:~# chmod +x /usr/local/bin/system_health_monitor.sh
root@ip-172-31-1-45:~#  /usr/local/bin/system_health_monitor.sh
2025-11-02 16:54:09 [INFO] CPU usage: 5%
2025-11-02 16:54:09 [WARN] High Memory usage: 37% (threshold 20%)
2025-11-02 16:54:09 [WARN] Disk usage issues found (>= 20%):
2025-11-02 16:54:09 [WARN] /dev/root 7034376 2180760 4837232 32 /
2025-11-02 16:54:09 [INFO] All monitored processes are running: sshd nginx
2025-11-02 16:54:09 [ALERT] Sent alert. (MAIL_TO=disabled)
root@ip-172-31-1-45:~#
```

Instances | EC2 | us-east-1
EC2 Instance Connect | us-east-
SecurityGroups | EC2 | us-east-
Inbox (2,931) - akilanb28@gm
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-07b337fa7b82c29b0&osUser=ubuntu&sshPort=22&addressFa...
Search
[Alt+S]
United States (N. Virginia)
Account ID: 5716-0083-0781
Akilan
i-07b337fa7b82c29b0 (server)
PublicIPs: 54.90.94.148 PrivateIPs: 172.31.1.45
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences