

Q1. Python Scripting

Step 1: Creating the Log File.

```
GNU nano 4.8 /tmp/timestamp.log Modified
2025-04-04 22:01:15,123 - INFO - Application started successfully.
2025-04-04 22:01:15,567 - DEBUG - Initializing database connection.
2025-04-04 22:01:16,234 - INFO - User 'Alice' logged in.
2025-04-04 22:01:17,890 - ERROR - Failed to connect to external service. Retrying in 5 seconds. Error: Connection refused.
2025-04-04 22:01:22,956 - WARNING - High CPU usage detected: 95%.
2025-04-04 22:01:25,345 - ERROR - Database query failed. Error: Table 'users' not found.
2025-04-04 22:01:28,789 - INFO - User 'Bob' logged in.
2025-04-04 22:01:31,456 - DEBUG - Processing request for resource '/data'.
2025-04-04 22:01:33,012 - ERROR - File not found. Path: /tmp/important_file.txt.
2025-04-04 22:01:36,543 - INFO - Application shutting down.
2025-04-04 22:01:37,098 - WARNING - Unsaved data might be lost.
2025-04-04 22:01:38,675 - ERROR - Exception during shutdown. Error: 'NoneType' object has no attribute 'close'.

␣

Activate Windows
Go to Settings to activate Windows.

␣ Get Help ␣ Write Out ␣ Where Is ␣ Cut Text ␣ Justify ␣ Cur Pos ␣ M-U Undo ␣ M-A Mark Text ␣ M-] To Bracket
␣ Exit ␣ Read File ␣ Replace ␣ Paste Text ␣ To Spell ␣ Go To Line ␣ M-E Redo ␣ M-C Copy Text ␣ M-? Where Was
```

Step 2: Creating the Python Script file to see the timestamp and log errors.

```
GNU nano 4.8 parse_logs.py
import re
import json

log_file_path = "/tmp/timestamp.log"
output_file_path = "error_logs.json"

error_logs = []

# Regular expression to extract timestamp and error messages
log_pattern = re.compile(r"(?P<timestamp>\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{3}) - ERROR - (?P<message>.+)")

with open(log_file_path, "r") as file:
    for line in file:
        match = log_pattern.search(line)
        if match:
            error_logs.append({
                "timestamp": match.group("timestamp"),
                "message": match.group("message").strip()
            })

# Write to JSON file
with open(output_file_path, "w") as json_file:
    json.dump(error_logs, json_file, indent=4)

print(f"Extracted {len(error_logs)} error logs and saved to '{output_file_path}'")

Read 25 Lines

Activate Windows
Go to Settings to activate Windows.

␣ Get Help ␣ Write Out ␣ Where Is ␣ Cut Text ␣ Justify ␣ Cur Pos ␣ M-U Undo ␣ M-A Mark Text ␣ M-] To Bracket
␣ Exit ␣ Read File ␣ Replace ␣ Paste Text ␣ To Spell ␣ Go To Line ␣ M-E Redo ␣ M-C Copy Text ␣ M-? Where Was
```

Step 3: After Creating the file providing the necessary permissions with chmod a+x and Running the Script.

- It then generates a log file- **parse_logs.py** in Json format.

```
master@master-vm:~/python$ nano parse_logs.py
master@master-vm:~/python$ chmod a+x parse_logs.py
master@master-vm:~/python$ python3 parse_logs.py
Extracted 4 error logs and saved to 'error_logs.json'
master@master-vm:~/python$ ls
error_logs.json  parse_logs.py
```

Step 4: Viewing the Log files.

```
master@master-vm:~/python$ cat error_logs.json
[
  {
    "timestamp": "2025-04-04 22:01:17,890",
    "message": "Failed to connect to external service. Retrying in 5 seconds. Error: Connection refused."
  },
  {
    "timestamp": "2025-04-04 22:01:25,345",
    "message": "Database query failed. Error: Table 'users' not found."
  },
  {
    "timestamp": "2025-04-04 22:01:33,012",
    "message": "File not found. Path: /tmp/important_file.txt."
  },
  {
    "timestamp": "2025-04-04 22:01:38,675",
    "message": "Exception during shutdown. Error: 'NoneType' object has no attribute 'close'."
  }
]
```