**Вправа - 1**

Ми працюватимемо з файлом журналу під назвою **syslog.log** , який містить журнали, пов’язані з *ticky* .

Ви можете переглянути цей файл за допомогою:

**cat syslog.log**

Скопійовано!

content\_copy

Лінії колод слідують візерунку, подібному до тих, які ми бачили раніше. Щось на зразок цього:

May 27 11:45:40 ubuntu.local ticky: INFO: Created ticket [#1234] (username)

Jun 1 11:06:48 ubuntu.local ticky: ERROR: Connection to DB failed (username)

Коли служба працює належним чином, вона записує INFO-повідомлення до syslog. Потім у ньому вказується, що він зробив, і вказується ім’я користувача та номер квитка, пов’язані з подією. Якщо служба стикається з проблемою, вона записує повідомлення про ПОМИЛКУ в syslog. У цьому повідомленні про помилку вказується, що сталося не так, і вказується ім’я користувача, яке ініціювало дію, яка спричинила проблему.

У цьому розділі ми шукатимемо та переглядатимемо різні типи повідомлень про помилки. Повідомлення про помилки для *ticky* докладно описують проблеми з файлом із міткою часу, коли виникла кожна проблема.

Нижче наведено кілька типів перелічених помилок.

* Timeout while retrieving information
* The ticket was modified while updating
* Connection to DB failed
* Tried to add information to a closed ticket
* Permission denied while closing ticket
* Ticket doesn't exist

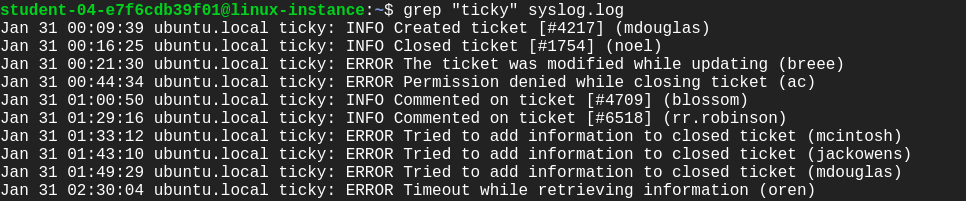
Щоб отримати всі журнали з *ticky* , скористайтеся такою командою:

**grep ticky syslog.log**

Скопійовано!

content\_copy

Вихід:



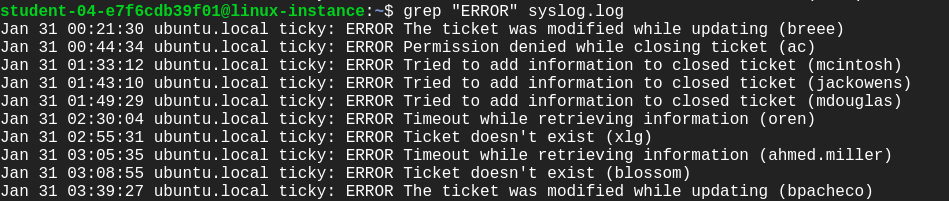
Щоб виконати пошук у всіх журналах **ERROR** , скористайтеся такою командою:

**grep "ERROR" syslog.log**

Скопійовано!

content\_copy

Вихід:



Щоб залучити всі повідомлення про ПОМИЛКУ певного типу , використовуйте наведений нижче синтаксис.

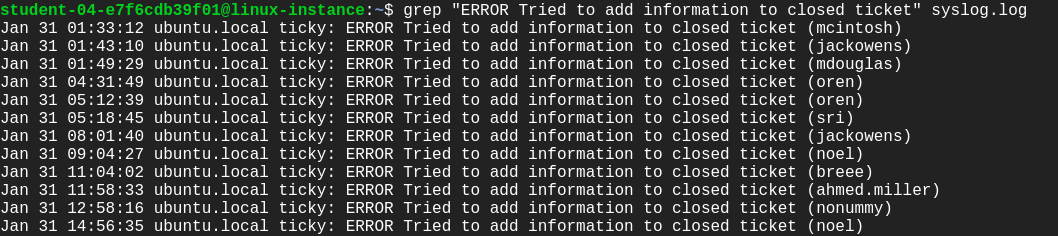
**Syntax:** grep ERROR [message] [file-name]

**grep "ERROR Tried to add information to closed ticket" syslog.log**

Скопійовано!

content\_copy

Вихід:



Давайте тепер напишемо кілька регулярних виразів за допомогою інтерпретатора python3.

Ми також можемо grep повідомлення ERROR/INFO у пітонічний спосіб за допомогою регулярного виразу. Давайте тепер напишемо кілька регулярних виразів за допомогою інтерпретатора **python3** .

Відкрийте **оболонку Python** за допомогою команди нижче:

**python3**

Скопійовано!

content\_copy

Це відкриває **оболонку** . Python надає **Python Shell** (також відому як **Python** Interactive **Shell** ), яка використовується для виконання однієї команди **Python** і отримання результату.

Імпортуйте модуль регулярного виразу (re).

**import re**

**line = "May 27 11:45:40 ubuntu.local ticky: INFO: Created ticket [#1234] (username)"**

Скопійовано!

content\_copy

Щоб зіставити рядок, що зберігається в **рядковій** змінній, ми використовуємо метод **search( ),** визначаючи шаблон.

**re.search(r"ticky: INFO: ([\w ]\*) ", line)**

Скопійовано!

content\_copy

Вихід:

<re.Match object; span=(29, 57), match='ticky: INFO: Created ticket '>

Скопійовано!

content\_copy

Ви також можете отримати повідомлення **ERROR** , як ми робили для журналу **INFO** вище, з рядка журналу ERROR.

line = "May 27 11:45:40 ubuntu.local ticky: ERROR: Error creating ticket [#1234] (username)"

Скопійовано!

content\_copy

Щоб зіставити рядок, що зберігається в **рядковій** змінній, ми використовуємо метод search( ), визначаючи шаблон.

**re.search(r"ticky: ERROR: ([\w ]\*) ", line)**

Скопійовано!

content\_copy

Вихід:

<re.Match object; span=(29, 65), match='ticky: ERROR: Error creating ticket '>

Скопійовано!

content\_copy

Тепер, коли ви знаєте, як використовувати регулярні вирази з Python, почніть отримувати журнали *ticky* для певного імені користувача. Вони нам знадобляться в наступних розділах.