Problem with deploying web almond

Question

Hi,

My name is Anggrio and I'm a student from the Australian National University. I'm currently working on a project that wants to use Almond as a base for developing a text based virtual assistant that can filter news. My team has so far followed the documentation that is written here for the **web Almond only** option but we are stuck on step 3 where we are required to provide some keys. Specifically, we are unsure on how to provide keys for the `SECRET_KEY` , `JWT_SIGNING_KEY` and `AES_SECRET_KEY` variables. We are all very new to Almond, open source, and web deployment, so any help would be greatly appreciated.

Thank you

```
大家好，我叫安格里奧，是澳大利亞國立大學的學生。 我目前正在從事一個項目，該項目希望使用
Almond 作為基礎來開發可以過濾新聞的基於文本的虛擬助手。 到目前為止，我的團隊一直遵循此處為僅
網絡杏仁選項編寫的文檔，但我們仍停留在需要提供一些密鑰的第 3 步。 具體來說，我們不確定如何為
SECRET_KEY 、 JWT_SIGNING_KEY 和 AES_SECRET_KEY 變量提供密鑰。 我們對 Almond、開源和
web 部署都很陌生，因此我們將不勝感激。 謝謝
```

Answer

Hi @Anggrio, welcome to the community.

Those keys should be generated randomly and then kept secret.
AES_SECRET_KEY needs to be exactly 128 bits (16 bytes), formatted as 32 hex characters.
SECRET_KEY and JWT_SIGNING_KEY can be anything as long as they are formatted in hex.
Choosing a 128 or 256 bit random string is recommended.

On Linux, this is a simple command to generate a strong random key of 32:

```
dd if=/dev/random of=/dev/stdout bs=32 count=1 | od -t x8 -w
```

You can then copy the hex digits and remove the spaces.

```
嗨@Anggrio，歡迎來到社區。

這些密鑰應該隨機生成，然後保密。
AES_SECRET_KEY 需要正好是 128 位（16 字節），格式為 32 個十六進製字符。 SECRET_KEY 和
JWT_SIGNING_KEY 可以是任何東西，只要它們是十六進制格式即可。 建議選擇 128 或 256 位隨機字
符串。

在 Linux 上，這是一個生成 32 強隨機密鑰的簡單命令：

dd if=/dev/random of=/dev/stdout bs=32 count=1 | od -t x8 -w
然後，您可以復制十六進制數字並刪除空格。
```

Hi Giovanni,

I tried running the command you gave and it gave me 4 sets of 16 hex characters like as follows:
`4d626d8cfe4a7120 54e68c7d7dcaf6cc 37053ad1642b07e5 25774060484f1c9a`
Is it correct that 32 hex characters would be something like
`4d626d8cfe4a712054e68c7d7dcaf6cc`? As in only needing half of it (I will generate a new one for the actual key) or am I misunderstanding the requirement.

To assign the characters to the key, is it safe to just paste them in the `config.js` file? From previous web projects I assumed that the standard is using environment variables, but the documentation for Almond mentions that the change is done in the file itself.

Thanks again for your help

```
嗨，喬瓦尼，

我嘗試運行您提供的命令，它給了我 4 組 16 個十六進製字符，如下所示：
4d626d8cfe4a7120 54e68c7d7dcaf6cc 37053ad1642b07e5 25774060484f1c9a
32 個十六進製字符類似於 4d626d8cfe4a712054e68c7d7dcaf6cc 是否正確？ 因為只需要一半（我將為實際密鑰生成一個新密鑰）或者我誤解了要求。

要將字符分配給鍵，將它們粘貼到 config.js 文件中是否安全？ 從以前的 web 項目中，我假設標準使用的是環境變量，但 Almond 的文檔提到更改是在文件本身中完成的。

再次感謝你的幫助
```

---

1. You should use half of it for AES_SECRET_KEY (128 bits), and the whole 64 characters for the other two (use three different strings, to be clear)
2. Save them in /etc/almond-cloud/config.js (alternatively: /etc/almond-cloud/config.yaml or /etc/almond-cloud/config.json). This is in fact safer than environment variables because the variables won't accidentally leak to spawned processes, and you can use file system permissions for additional isolation.

```
1. 您應該將其中的一半用於 AES_SECRET_KEY（128 位），將整個 64 個字符用於另外兩個（使用三個不同的字符串，要清楚）

2. 將它們保存在 /etc/almond-cloud/config.js（或者：/etc/almond-cloud/config.yaml 或 /etc/almond-cloud/config.json）。 這實際上比環境變量更安全，因為變量不會意外洩漏到衍生進程，並且您可以使用文件系統權限進行額外隔離。
```

We use a public github repository to store the file. Would the key be visible to visitors then?

Our project has to be put in a public repository (from the course requirement), but I suppose the normal case is to put it into private/organization repository for security right? or are there ways to isolate config.js in the repository?

```
我們使用公共 github 存儲庫來存儲文件。 那麼訪問者會看到鑰匙嗎？

我們的項目必須放在公共存儲庫中（根據課程要求），但我認為正常情況是將其放入私人/組織存儲庫以確保安全，對嗎？ 或者有沒有辦法在存儲庫中隔離 config.js？
```

You should not store the configuration file in a public repository. The normal case is to store the configuration in some private repository (separate from the actual code repository), or some configuration management system. Barring that, having the configuration file only stored in the machine where the deployment occurs is the safest option.

> 您不應將配置文件存儲在公共存儲庫中。 正常情況是將配置存儲在某個私有存儲庫（與實際代碼存儲庫分開）或某個配置管理系統中。 除此之外，將配置文件僅存儲在部署發生的機器中是最安全的選擇。

Ok, I think I will have to continue following the documentation to see how we will need to do it. Would it be better to post any following questions in this discussion or should I make a different one?

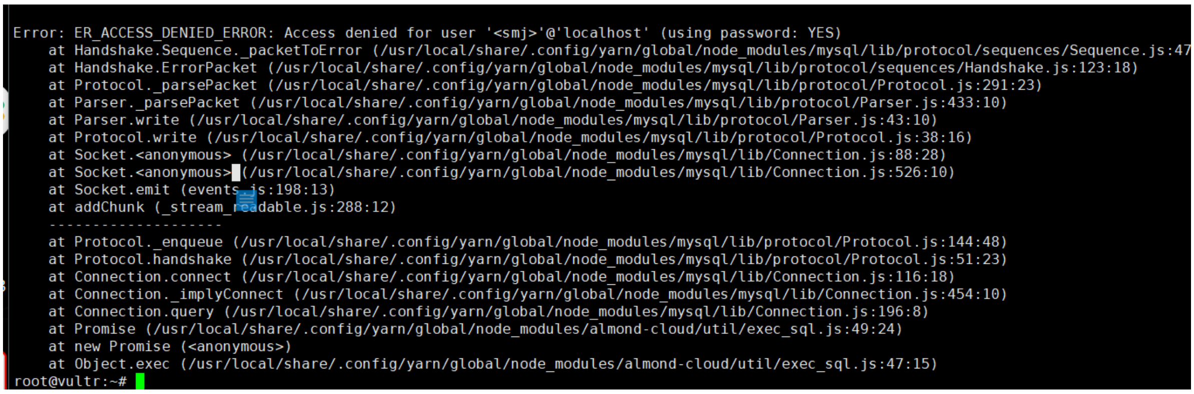
> 好的，我想我將不得不繼續遵循文檔以了解我們需要如何做。 在此討論中發布以下任何問題會更好還是我應該提出不同的問題？

Feel free to ask here or create new topics, as you prefer.

> 您可以隨意在這裡提問或創建新主題。

Hi Giovanni,

My team has completed step 3 by adding the secret key variables as well as the mysql server link. However, when we tried to run the command in step 4 (almond-cloud bootstrap), the following error happens:

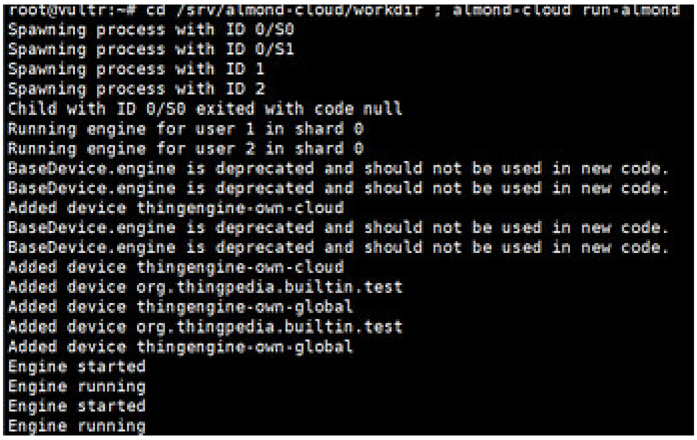


```
Error: ER_ACCESS_DENIED_ERROR: Access denied for user '<smj>'@'localhost' (using password: YES)
    at Handshake.Sequence._packetToError (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/sequences/Sequence.js:47
    at Handshake.ErrorPacket (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/sequences/Handshake.js:123:18)
    at Protocol._parsePacket (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/Protocol.js:291:23)
    at Parser._parsePacket (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/Parser.js:433:10)
    at Parser.write (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/Parser.js:43:10)
    at Protocol.write (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/Protocol.js:38:16)
    at Socket.<anonymous> (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/Connection.js:88:28)
    at Socket.<anonymous> (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/Connection.js:526:10)
    at Socket.emit (events.js:198:13)
    at addChunk (_stream_readable.js:288:12)
    --------------------
    at Protocol._enqueue (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/Protocol.js:144:48)
    at Protocol.handshake (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/protocol/Protocol.js:51:23)
    at Connection.connect (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/Connection.js:116:18)
    at Connection._implyConnect (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/Connection.js:454:10)
    at Connection.query (/usr/local/share/.config/yarn/global/node_modules/mysql/lib/Connection.js:196:8)
    at Promise (/usr/local/share/.config/yarn/global/node_modules/almond-cloud/util/exec_sql.js:49:24)
    at new Promise (<anonymous>)
    at Object.exec (/usr/local/share/.config/yarn/global/node_modules/almond-cloud/util/exec_sql.js:47:15)
root@vultr:~#
```

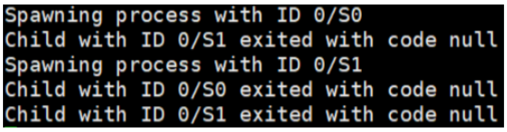
Is there something wrong with our mysql server here?

> 嗨，喬瓦尼，
>
> 我的團隊通過添加密鑰變量以及 `mysql` 服務器鏈接完成了第 `3` 步。 但是，當我們嘗試運行第 `4` 步（`almond-cloud bootstrap`）中的命令時，會出現以下錯誤：
> 我們這裡的`mysql`服務器有問題嗎？

We managed to fix the above problem by reinstalling mysql (it turns out that we accidently installed both mariadb and mysql), but we are now stuck in step 5 and 6.
The output for step 5 is as follows:

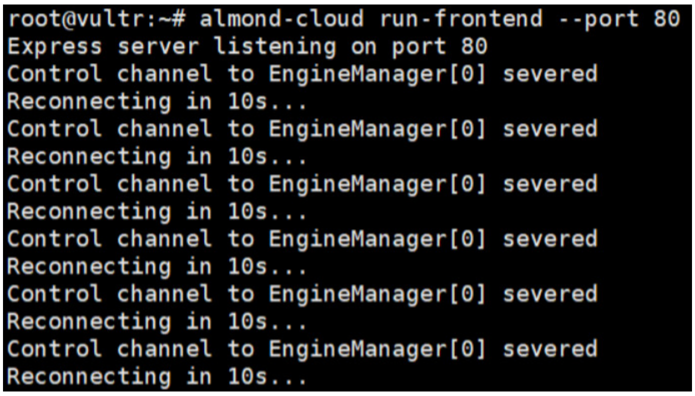


```
root@vultr:~# cd /srv/almond-cloud/workdir ; almond-cloud run-almond
Spawning process with ID 0/S0
Spawning process with ID 0/S1
Spawning process with ID 1
Spawning process with ID 2
Child with ID 0/S0 exited with code null
Running engine for user 1 in shard 0
Running engine for user 2 in shard 0
BaseDevice.engine is deprecated and should not be used in new code.
BaseDevice.engine is deprecated and should not be used in new code.
Added device thingengine-own-cloud
BaseDevice.engine is deprecated and should not be used in new code.
BaseDevice.engine is deprecated and should not be used in new code.
Added device thingengine-own-cloud
Added device org.thingpedia.builtin.test
Added device thingengine-own-global
Added device org.thingpedia.builtin.test
Added device thingengine-own-global
Engine started
Engine running
Engine started
Engine running
```



```
Spawning process with ID 0/S0
Child with ID 0/S1 exited with code null
Spawning process with ID 0/S1
Child with ID 0/S0 exited with code null
Child with ID 0/S1 exited with code null
```

The output for step 6 is as follows:



```
root@vultr:~# almond-cloud run-frontend --port 80
Express server listening on port 80
Control channel to EngineManager[0] severed
Reconnecting in 10s...
Control channel to EngineManager[0] severed
Reconnecting in 10s...
Control channel to EngineManager[0] severed
Reconnecting in 10s...
Control channel to EngineManager[0] severed
Reconnecting in 10s...
Control channel to EngineManager[0] severed
Reconnecting in 10s...
Control channel to EngineManager[0] severed
Reconnecting in 10s...
```

If my understanding is correct, we should do both of these commands at the same time right? Do you have any suggestions on what the problem is here?

我們通過重新安裝`mysql`設法解決了上述問題（結果是我們不小心安裝了`mariadb`和`mysql`），但我們現在卡在步驟5和6中。
步驟 5 的輸出如下：

步驟 6 的輸出如下：
如果我的理解是正確的，我們應該同時執行這兩個命令對嗎？ 您對這裡的問題有什麼建議嗎？

Are you running both processes in the same directory? There should be a “control” socket file in the directory where you’re running the backend process.

Also, you should probably avoid running as root (unless you’re in a docker or similar environment), as that can lead to hard-to-debug permission errors…

您是否在同一目錄中運行兩個進程？ 在您運行後端進程的目錄中應該有一個“控制”套接字文件。

此外，您可能應該避免以 `root` 身份運行（除非您在 `docker` 或類似環境中），因為這會導致難以調試的權限錯誤……