

### Лабораторная работа №3

Настройка автоматического создания бэкапов.

**Задача 1:** На вашем сервере создать файл, создающий резервную копию вашего сервиса.

План действий:

1. Соединиться с вашим сервером
2. Определить корневую папку вашего сервиса и путь к ней.

Данную папку веб-сервер открывает при попытке соединиться с вашим сервером по IP. Допустим это будет папка:

```
droot = /home/www;
```

3. Создать специальную папку для хранения бэкапов, допустим

```
mkdir ${droot}/backup
```

4. Установить права доступа к папке такие, чтобы содержимое папки было доступно через http

5. Перейти в эту папку. Далее создать в ней bash скрипт:

```
vi backup.sh
```

!!!Данный скрипт будет использоваться для автоматического создания бэкапов вашего сервиса. Далее вся работа ведется в этом скрипте. Построчно данный скрипт должен содержать!!!:

- a. Первая строка содержит указание, что данный скрипт исполняет именно bash код:

```
#!/bin/bash.
```

- b. Введите переменную `projname`, которая содержит название вашего сервиса на латинице

- c. Далее вы вводите переменную `droot`, которая содержит путь к папке с вашим сервисом

- d. Создание бэкапа базы данных в корневую папку `droot` с названием `sql_${projname}_dump.sql`

- e. Создайте переменную `fna` которая будет содержать название проекта и текущую дату

- f. Заархивируйте ваш сервис из папки `droot` в папку `/tmp/`, исключая из архива папку бэкапов. Название архива `backup_${fna}_tar.gz`.
- g. Удалить из папки `droot` созданный дамп `.sql`
- h. Переместите созданный архив в папку для бэкапов
- i. Удалите (на всякий случай) архив из папки `/tmp`
- j. Написать условие, если существует файл `${droot}/backup/lbkr.txt`, то сохраняем его содержимое в переменную `prev` (данный файл содержит название последнего бэкапа). Далее удаляем бэкап по названию из файла `${droot}/backup/${prev}`
- k. Удалите файл `lbkr.txt` из вашей папки `backup`
- l. Запишите название нового бэкапа в файл `lbkr.txt` в папку `backup`
- 6. Ваш файл для бэкапов создан. Протестируйте его функционирование.
- 7. Установить выполнение данного скрипта по крону каждые 10 минут.

**Задача 2. На вашем сервере создать файл, забирающий резервные копии с чужих серверов.**

План действий:

- 1. Создайте на сервере `/backup_storage/`
- 2. Перейдите в папку `/backup_storage/` и создайте файл `projects`
- 3. В файл `projects` пропишите IP адреса серверов ваших сокурсников, на каждой строчке по одному IP
- 4. В папке `/backup_storage` создайте папку `backups` (в нее будут складываться бэкапы)
- 5. Вернитесь в папку `/backup_storage`. Создайте в ней `bash` скрипт:  
`vi get_backup.sh`
  - a. Первая строчка содержит указание, что данный скрипт исполняет именно `bash` код:

`#!/bin/bash.`

b. Напишите конструкцию:

```
while read line # читаем построчно
```

```
do # исполняем кода
```

*# здесь будет основной исполняющий код в цикле*

```
done < /backup_storage/projects # эта строчка заполняет цикл на чтение
```

из файла построчно

далее должно идти описание основного исполняющего кода цикла:

c. Заполняем переменную `projname` значением из первой строчки файла;

```
projname=$line;
```

d. Далее сохраните url проекта в переменную `projurl`

e. Напишите условие, что если не существует папки `/backup_storage/backups/$projname`, то надо ее создать (это папка бэкапов проекта).

f. Далее удалите все файлы \*.txt из папки бэкапов проекта

g. Скачайте файл `lbpk.txt` по url `projurl` и сохраните этот файл в папку бэкапов проекта

h. Сохраните в переменную `lback` содержимое скаченного файла

i. Так как мы знаем, что файл `lbpk.txt` хранил название последнего бэкапа, и данное название хранится теперь в переменной `lback`, то необходимо скачать с `projurl` последний бэкап с названием `lback`

6. Вы создали скрипт, скачивающий все бэкапы к вам на сервер. Теперь установите, чтобы по крону он каждые полчаса запускался и протестируйте.