Практическое задание №2. Мониторинг, PHP и Bash-автоматизация

1. Диагностическая страница средствами PHP

У вас уже настроен виртуальный хост и есть файл `/var/www/<ваш_логин>/public_html/index.php`.
Замените его содержимое на PHP-скрипт, который будет выводить:

- Приветствие: `Панель управления сервером <ваш_логин>`.
- Версию PHP (функция `phpversion()`).
- Имя сервера (функция `gethostname()`).
- **Объем свободной оперативной памяти** (прочитайте файл `/proc/meminfo` средствами PHP, найдите строку `MemAvailable`, переведите в мегабайты).
- **Среднюю нагрузку на систему (Load Average)** (прочитайте `/proc/loadavg`).
- Текущее время в формате `ДД.ММ.ГГГГ ЧЧ:ММ`.

Проверьте работу скрипта в браузере по адресу `http://<ваш_логин>.bolzens.ru/`.

2. Анализ конфигурации и логов Apache

- Выведите в терминал содержимое вашего конфигурационного файла: `cat /etc/apache2/sites-available/<ваш_логин>.bolzens.ru.conf`.
- Обратите внимание на пути к логам (ErrorLog и CustomLog).
- Перейдите в директорию с логами и посмотрите последние 5 строк лога доступа: `sudo tail -n 5 /var/log/apache2/<ваш_логин>-access.log`.
- Сделайте пару запросов к вашему сайту из терминала (например, `curl -I http://localhost/`) и убедитесь, что в логе появились новые записи.

3. Написание Bash-скрипта (Резервное копирование)

Вам нужно автоматизировать бэкап файлов вашего сайта.

- Перейдите в свою домашнюю директорию (`cd ~`).
- Создайте файл `backup.sh` (используйте `nano`).
- Сделайте его исполняемым (`chmod +x backup.sh`).
- Напишите скрипт, который:
  - Создает папку `backups` в домашней директории (если её нет).
  - Узнает текущую дату в формате `ГГГГ-ММ-ДД` (команда `date +%F`).
  - Упаковывает папку `/var/www/<ваш_логин>/public_html` в архив `tar.gz`. Имя архива должно быть `backup_<ДАТА>.tar.gz`.
  - Перемещает этот архив в папку `backups`.
  - Выводит в консоль сообщение об успешном создании бэкапа и его размер (команда `du -h`).
- Запустите скрипт: `./backup.sh` и проверьте, что архив создался (`ls -lh ~/backups`).

4. Оптимизация: контроль ресурсов

Так как память ограничена, важно следить за процессами.

Выведите в терминал текущее потребление памяти: `free -h`.

Выведите топ-5 процессов по потреблению памяти: `ps aux --sort=-%mem | head -6`.

Узнайте, сколько памяти занимает веб-сервер Apache (запустите `ps aux | grep apache` и просуммируйте объем памяти `RSS` вручную или через `awk`).

Вопрос для размышления (ответьте в отчете текстом): Почему при лимите в 256 МБ на контейнер не стоит запускать множество тяжелых процессов одновременно? Как оптимизация кода на PHP (например, отказ от тяжелых фреймворков) влияет на потребление памяти?