**Практическая работа "Деплой приложения на сервис amvera"**

**Цель**: закрепить упаковку веб-приложения в docker, закрепить начальные практические навыки по автодеплою и поработать с сервисом, позволяющим разворачивать веб-приложения, упакованные в docker.

**Задание:** реализовать веб-приложение согласно техническому заданию, упаковать данное приложение в docker и выполнить деплой на облачный сервис amvera.

**Техническое задание**

Необходимо реализовать JSON WEB API приложение на ASP.NET со следующими обработчиками:

1) GET / -> 200

{

“message”: “server is running”

}

2) GET /ping -> 200

{

“message”: “pong”

}

3) GET /time-to-new-year -> 200

{

“days”: <кол-во целых дней до ближайшего НГ>,

“hours”: <кол-во целых часов до ближайшего НГ>,

“minutes”: <кол-во целых минут до ближайшего НГ>

}

Значения полей не связаны между собой, каждое поле хранит общее кол-во времени до ближайшего НГ в указанных единицах (дни, часы, минуты)

**План работы**

1. Создать проект пустого приложения ASP.NET в Visual Studio. Реализовать приложение согласно техническому заданию. Настроить запуск на порт 8080. Запустить приложение локально из хостовой ОС. Протестировать работу приложения в Postman. Приложение должно быть реализовано максимально простым и удобным для вас способом.

2. Упаковать данное приложение в docker. Для этого сгенерировать Dockerfile в Visual Studio (во избежание проблем с Visual Studio сразу переключить профиль запуска с Dockerfile на http).

Развернуть приложение в docker любым удобным для вас способом:

* использование команд docker build / docker run для сборки и запуска контейнера
* использование docker-compose.yml, в котором прописать конфигурацию сервиса приложения, ссылающегося на сборку Dockerfile (как в предыдущей практике но без БД) - РЕКОМЕНДУЕТСЯ

Для всей работы, не касающейся кода (работа с docker-compose, git, команд docker) рекомендуется открыть корневую папку решения Visual Studio в Visual Studio Code, в нем же открыть терминал для выполнения команд.

Протестировать работу приложения, упакованного в docker.

Если у вас нет возможности использовать docker на локальной машине, то можно подготовить dockerfile и перейти к п.3. т.к. вы можете все равно задеплоить приложение.

3. Зарегистрироваться на сервисе <https://amvera.ru/>, создать в нем новый проект приложения (использовать начальный баланс, по завершению работы не забудьте отключить проект чтобы не списывался начальный баланс).

В открытом в корне решения Visual Studio Code терминале заинициализировать локальный git-репозиторий: git init

Если у вас главная ветка main (или иная, отличная от master), то предварительно переименуйте ее в master: git branch -m master

Создать в корне решения файл .gitignore и вставить туда шаблон содержимого для Visual Studio: <https://github.com/github/gitignore/blob/main/VisualStudio.gitignore>

Добавить файлы изменений в коммит: git add .

Точка тоже часть команды!

Сделать коммит: git commit -m “Init project files”

Добавить удаленный репозиторий amvera к вашему репозиторию: git remote add <ссылка на ваше репозиторий amvera> - данную команду можно найти в разделе “Репозиторий” вашего проекта amvera.

Запушить проект на удаленный репозиторий amvera: git push amvera master

Убедиться в том, что файлы проекта появились на amvera в разделе “Репозиторий”

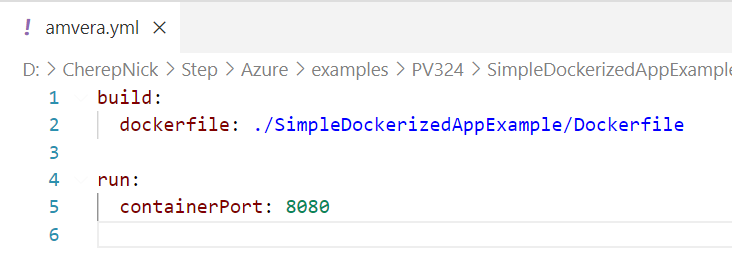
В своем проекте добавить файл amvera.yml в корне решения VS. Прописать в нем конфиг сборки и разворачивания:

build:

dockerfile: <относительный путь к Dockerfile>

run:

containerPort: 8080



Выполнить коммит и пуш:

git add .

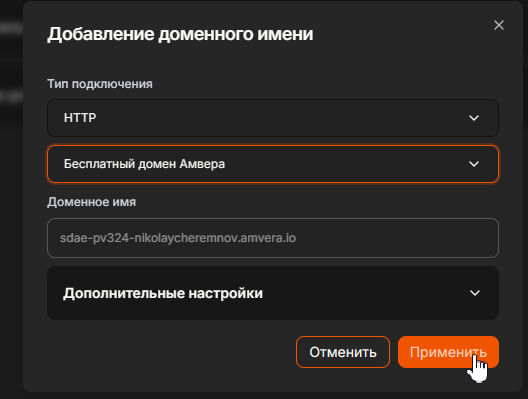
git commit -m “Add amvera.yml”

git push amvera master

Убедиться в том, что файл запушился на amvera и приложение начало разворачиваться.

Мониторить логи разворачивания и дождаться сборки и запуска проекта.

После успешной сборки и запуска (либо во время разворачивания) добавить публичный бесплатный домен для своего проекта в разделе “Домены” - HTTP бесплатный



Протестировать запущенное приложение, используя публичный домен. Поделиться ссылка на приложение в чат собрания.

На данном этапе основная практика завершена.

4\*. Данный пункт не является обязательным, но рекомендуемым.

Внести любое изменение в код вашего веб-приложения (добавить новый обработчик, поменять существующий и пр.)- максимально простое, но видимое со стороны клиента. Протестировать изменение локально, затем выполнить обновление приложения на сервисе:

git add .

git commit -m “Add feature”

git push amvera master

Начнет выполняться новая сборка!

Есть ненулевая вероятность, что amvera повиснет :с

Но попробовать стоит!