

ДОДАТОК А  
Слайди презентації

Харківський національний університет радіоелектроніки  
Кафедра Інформаційно-мережної інженерії  
Атестаційна робота магістра на тему:

**Проектування системи автоматизованого розгортання  
інфраструктури веб-проекту**

Студент: Лисенко Максим Ігорович  
Група: ІМІм-18-1  
Керівник: доц. к.т.н. Костромицький Андрій Іванович  
Харків - 2019

## Вступ

Актуальність автоматизованого підходу полягає у скороченні використання людських ресурсів за рахунок відмови від ручних дій, а отже, є можливість виділити більше часу на впровадження нових функцій до продукту, засобів безпеки веб – проекту.

При впровадженні автоматизації, компанія стає більш конкурентоспроможною, за рахунок використання новітніх технологій та підходів, за рахунок більш гнучкого розподілу людських ресурсів.

## Мета роботи

- Огляд загальних принципів побудови веб-проекту.
- Аналіз та порівняння найпоширеніших засобів автоматизації інфраструктури веб-проекту.
- Аналіз та порівняльна характеристика інструментів оркестрації відповідних ресурсів.
- Побудова веб-проекту на основі вхідних даних.

3

## Переваги та недоліки автоматизації

### переваги:

- підвищення продуктивності;
- поліпшення якості;
- економія витрат часу;
- спрямування заощадженого часу на більш важливі завдання;
- збільшення прибутку.

### недоліки:

- складність впровадження;
- несприйняття колективом.

4

## Структура розробки веб - проекту



Рисунок 1 – Структура розробки веб-проекту

5

## Оркестрація

Оркестрація – це автоматизована конфігурація, координація та управління комп'ютерними системами та програмним забезпеченням.

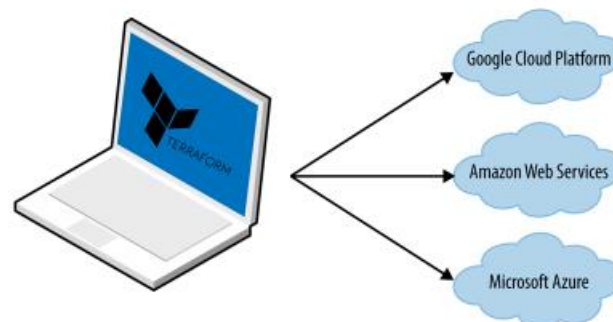


Рисунок 2 – Оркестрація ресурсів

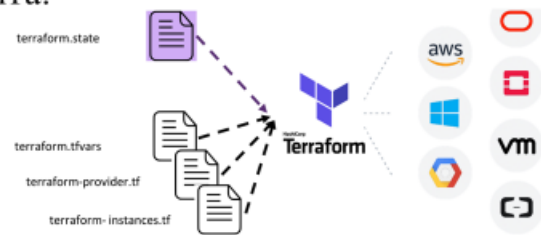
6

## Terraform

Terraform – інструмент, що дозволяє розробляти, змінювати і версіювати інфраструктуру сервісу в форматі коду.

переваги:

- Оркестрування, а не просто конфігурація інфраструктури;
- побудова незмінної інфраструктури;
- декларативний, а не процедурний код;
- архітектура, що працює на стороні клієнта.



7

## Cloud Formation

AWS CloudFormation надає універсальну мову для опису і виділення всіх ресурсів інфраструктури в хмарному середовищі.

переваги:

- Моделювання усієї інфраструктури;
- автоматизація та розгортання;
- просто код.



8

## Terraform vs Cloud Formation

	Terraform	CloudFormation
Тип	Оркестрація	Оркестрація
Хмара	AWS, GCP, Azure та інші	Тільки AWS
Формат конфігурації	HCL/JSON	JSON
Контроль стану	+	-
Запуск без застосування	terraform plan	deploy --no-execute-change-set
Контроль застосування	+	-
Управління вже створеними ресурсами	+	-
Можливість розширення	Провайдери	-
Управління секретними даними	Видимі у state файлі	Не видимі
Перевірка змін	Детальний план перед виконанням	Change Sets

9

## Автоконфігурація

Автоконфігурація – процес безперервної та автоматичної доставки конфігурації на ресурси веб-проекту. Інструменти автоконфігурації потребують уваги тоді, коли зростає кількість ресурсів веб-проекту та ручна конфігурація кожного з них, процес рутинний та потребує багато часу та зусиль.

Відомі інструменти автоконфігурації, такі як: Puppet, Chef в даний час користуються популярністю за рахунок широкої аудиторії користувачів та вкладників.



10

## Puppet

Puppet вважається найбільш використовуваним з інструментів автоконфігурації. Він найбільш повний з точки зору можливих дій, модулів і призначених для користувача інтерфейсів, охоплює майже кожен операційну систему і надає утиліти для всіх основних ОС.

## Chef

Chef схожий на Puppet з точки зору загальної концепції, в ньому також є головний сервер і агенти, які встановлені на керованих вузлах. На додаток до головного сервера, установка Chef також вимагає робочої станції, для управління ним.

11

## Puppet vs Chef

	Переваги	Недоліки
Puppet	Повний інтерфейс користувача	Підхід, керований кодом, означає більшу гнучкість та контроль конфігурацій
	Потужні можливості звітування	Інструмент «knife» зменшує головний біль у установці
	Надає доступ до добре створеної спільноти підтримки	Надає широку колекцію рецептів конфігурації та модулів
Chef	Його модельний підхід дорівнює меншому контролю в порівнянні з підходами, керованими кодами	Якщо ви ще не знаєте Ruby та процедурного кодування, приготуйтеся до крутої кривої навчання
	Розширені завдання вимагають CLI, і оскільки він заснований на Ruby, вам потрібно буде ознайомитися з ним	Це складний інструмент

12

## Побудова веб-проекту

### Вихідні дані до проекту:

- Кластер віртуальних серверів для розгортання додатку;
- Публічна хмара AWS;
- База даних MySQL для зберігання результатів тестування та списку студентів;
- Налаштування мережного захисту за допомогою AWS Security Groups.

Проаналізувавши інструменти автоконфігурації та оркестрації та згідно з вхідними даними до проекту, вибираємо Terraform та Puppet.

13

## Інфраструктура веб-проекту

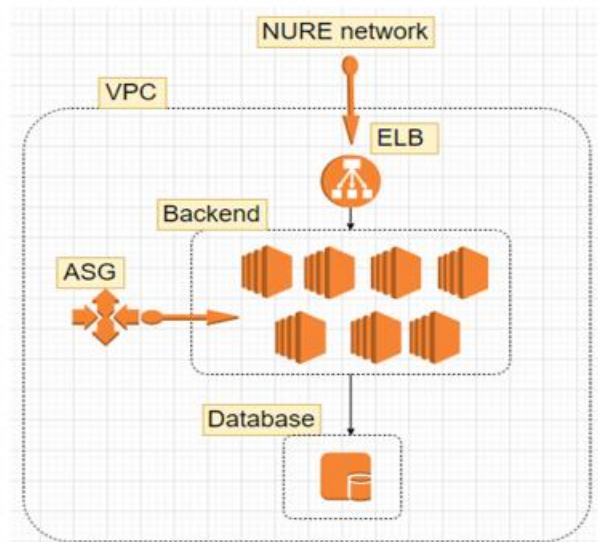


Рисунок 3 – Інфраструктура веб - проекту

14

## Security Groups

	Load Balancer	Backend	MySQL DB
Inbound	80, 8080, 443 з мережі університету (в залежності від налаштувань)	80, 8080, 443 з балансеру (в залежності від налаштувань)	3306 с Security Group бекенд машини
Outbound	80, 8080, 443 на бекенд машини (в залежності від налаштувань)	80, 8080, 443 для доступу до інтернет ресурсів	80, 8080, 443 через NAT Gateway
		3306 до Security Group MySQL DB машини	8140 на Puppetmaster
		8140 на Puppetmaster	

15

## Висновки

- Було описано загальні принципи побудови веб-проекту.
- Проаналізував та порівняв найпоширеніші засоби автоматизації інфраструктури веб-проекту.
- Проаналізував та дав порівняльну характеристику інструментам оркестрації відповідних ресурсів.
- Був побудований веб-проекту на основі вхідних даних.
- Впровадивши автоматизацію до розробки веб-проекту, вдалося підвищити відмовостійкість, гнукчість конфігурації, швидкість доставки оновлень, зменшити людський фактор.

16

**Дякую за увагу!**

