

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Kolaboratívne prototypovanie

[CollabUI]

Metodika nasadzovania

Tím: 24. CollabUI

Vedúci tímu: Ing. Eduard Kuric, PhD.

Členovia tímu: Adrián Nagy, Lukáš Vrba, Ján Kleň, Michal Melúch, Tomáš Mňačko, Peter Písecký,
Miloslav Smetana

Študijný program: Inteligentné softvérové systémy

Ročník: 2017/2018

1. Úvod

Metodika nasadzovania definuje postupy pri nasadzovaní produkčnej verzie projektu, webovej stránky tímového projektu a aktualizácií dokumentov na tímovej stránke. Metodika sa viaže k nasadzovaniu na server s operačným systémom **Ubuntu 17.04**.

Touto metodikou sa riadia všetci členovia tímu č. 24 Collab-UI v rámci predmetu tímový projekt v akademickom roku 2017/2018.

2. Pojmy a skratky

SSH (angl. Secure shell) – zabezpečí prístup k príkazovému riadku

VNC (angl. Virtual network computing) – grafický program, ktorý umožňuje vzdialené pripojenie ku grafickému používateľskému rozhraniu pomocou počítačovej siete

DNS (angl. Domain name system) – systém, ktorý ukladá prístup k informácii o názve stroja a názve domény

HTML – hypertextový značkový jazyk

PHP – skriptovací jazyk

Git – distribuovaný systém riadenia verzií

Virtual host – spôsob hostovania viacerých domén na jednom serveri

Root directory – hlavný adresár hierarchie adresovej štruktúry

DB – databáza

JS - JavaScript

Package manager – kolekcia softvérových nástrojov, ktoré automatizujú proces inštalácie

3. Dôležité odkazy

3.1 Konfigurácia VNC

<http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/html.html> - časť poznámky k inštalácii servera

<https://help.ubuntu.com/lts/serverguide/openssh-server.html> - inštalácia SSH

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-04> - inštalácia Apache

<https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-14-04> - inštalácia Git

<https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-16-04> - nastavenie Virtual host

3.2 Nasadenie produkčnej verzie

<https://book.cakephp.org/3.0/en/installation.html> - inštalácia CakePHP

<https://nodejs.org/en/download/package-manager/#debian-and-ubuntu-based-linux-distributions> - inštalácia NodeJS

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-16-04> - inštalácia PostgreSQL

<https://stackoverflow.com/questions/38975974/how-to-install-bower-on-ubuntu-16-04-lts> - inštalácia Bower

<https://github.com/datastax/php-driver/blob/master/ext/README.md> - PHP driver pre PHP rozšírenia

<https://github.com/datastax/php-driver> - Cassandra PHP driver

4 .Pravidlá a postupy

Pri nasadzovaní je potrebné striktné dodržanie postupu krokov, nakoľko jednotlivé kroky majú medzi sebou rôzne závislosti. Pred samotným nasadením projektu a webovej stránky je dobré vedieť, ako sa potupuje pri úplne novom VNC, ktorý ešte nie je nakonfigurovaný, aby nebol problém ani pri zmene poskytovateľa serverových služieb.

4.1 Konfigurácia VNC

Na prihlásenie do VNC je potrebný program, ktorý s VNC dokáže pracovať, napríklad program [RealVNC](#). Prihlásenie na virtuálny stroj je v tvare teamXX-17.studenti.fiit.stuba.sk, podľa informácií uvedených v tejto [dokumentácii](#), v časti poznámky k inštalácii servera.

Pred samotnou konfiguráciou je potrebné definovať, ako budeme k virtuálnemu stroju v neskorších fázach pristupovať. V našom prípade sme zvolili možnosť prístupu cez SSH. Príkaz na inštaláciu SSH na server:

```
sudo apt install openssh-server
```

V nasledujúcich krokoch budeme vždy na server pristupovať nasledovne:

```
ssh PRIHLASOVACIE_MENO@teamXX-17.studenti.fiit.stuba.sk
```

Aby sme vedeli zabezpečiť nasadenie projektu a webovej stránky je potrebný webový server napríklad Apache. Inštalácia Apache je možná nasledujúcim príkazom:

```
sudo apt-get update  
sudo apt-get install apache2
```

Pred inštaláciou akéhokoľvek rozšírenia je dobré najprv aktualizovať správcu inštalácie balíkov pomocou príkazu `sudo apt-get update`.

Aby sme nemuseli pri nasadzovaní ručne kopírovať súbory, tak na server nainštalujeme git nasledujúcim príkazom:

```
sudo apt-get install git
```

A prihlásime sa do gitu, z ktorého budeme získavať všetky dáta na nasadenie a aktualizáciu nasledovne:

```
git config --global user.name 'POUZIVATELSKE_MENO'
```

```
git config --global user.email 'POUZI VATELSKY EMAIL'
```

4.2 Nasadenie produkčnej verzie projektu

Pred nasadením produkčnej verzie projektu na server je potrebné nainštalovať požiadavky, ktoré sú nutnou súčasťou správneho fungovania projektu.

4.2.1 Konfigurácia virtual hosta

Nakoľko sa na servery nachádza viac adries, ktoré sa používajú je potrebné nastaviť virtual hosta, aby pri zadaní domény, na ktorej sa má nachádzať projekt smeroval do adresára s projektom. V našom prípade sme potrebovali umiestniť projekt, tak aby bol na doméne prvej úrovne, ale podľa zadania sa na tejto adrese musí nachádzať práve webová stránka tímu. Tým pádom sme museli kúpiť doménu, ktorej DNS záznamy sme presmerovali na náš server. Nastavenie virtual hosta prebieha v troch krokoch:

- Vytvorenie priečinka, v ktorom sa bude nachádzať projekt
- Vytvorenie nového virtual hosta v priečinku `/etc/apache2/sites-available`, podľa tohto [dokumentu](#).
- Nastavenie nového virtual hosta, pričom root directory bude smerovať do novo vytvoreného priečinka z prvého bodu

4.2.2 Inštalácia požiadaviek na server

Na zistenie požiadaviek je najprv potrebné overiť, aké technológie boli na projekt použité a či je potrebné doinštalovanie požiadaviek na prácu s nimi. V našom prípade sa jedná o nasledujúce technológie, ktoré si vyžadujú inštaláciu dodatočných požiadaviek:

- [CakePHP](#)
- [Cassandra](#)
- [PostgreSQL + vytvorenie DB](#)
- [NodeJS](#)
- [Bower](#)

4.2.3 Spustenie projektu na servery

Po nastavení servera pomocou inštalácie potrebných požiadaviek je treba postupovať podľa nasledujúcich inštrukcií. Ako prvé je potrebné prihlásiť na server a prejsť do priečinka, v ktorom sa bude do aktuálneho priečinka pomocou gitu:

```
git clone git @bitbucket.org:agileteam/ui-agile-repo.git  
git pull
```

Po skopírovaní repozitára je potrebné nastavenie projektu v niekoľkých krokoch:

- V priečinku `/config/` sa nachádza súbor `app.default.php`, ktorý treba skopírovať do súboru `app.php`
- V súbore `app.php`, treba nastaviť údaje (viď. obrázok nižšie), zodpovedajúce databázovým údajom, ktoré sa vytvorili v časti 4.2.2 (PostgreSQL + vytvorenie DB)

```

229 'Datasources' => [
230     'default' => [
231         'className' => 'Cake\Database\Connection',
232         'driver' => 'Cake\Database\Driver\Postgres',
233         'persistent' => false,
234         'host' => 'localhost',
235         'port' => '5432',
236         'username' => 'Iceweex',
237         'password' => '',
238         'database' => 'test',
239         'encoding' => 'utf8',
240         'timezone' => 'UTC',
241         'flags' => [],
242         'cacheMetadata' => true,
243         'log' => false,
244         'quoteIdentifiers' => false,
245         'url' => env('DATABASE_URL', null),
246     ],

```

Po nastavení pripojenia na relačnú databázu sa môžu spustiť migrácie oboch databáz pomocou príkazom pre PostgreSQL:

```

sudo -i u postgres
psql -U postgres -d postgres -a -f 'Root directory'/src/Dat abase/collabui.sql

```

A pre Cassandra pomocou príkazom:

```

cqlsh -f 'Root directory'/src/Dat abase/collabui.cql

```

Po migrácií databáz sa môžu aktualizovať PHP a JS package managers. PHP package manager Composer sa spustí príkazom z root directory:

```

composer install

```

A JS package manager Bower príkazom z root directory:

```

bower install

```

Nakoniec NPM package manager z priečinka `/node/`

```

npm install

```

Pre kompilovanie js a scss súborov je potrebné nastaviť Gulp a to skopírovaním súboru `gulp.default.js` v priečinku `/node/` do súboru `gulp.js` a nastaviť údaje (viď. Obrázok nižšie).

```

2 //***** CUSTOME SETTINGS *****/
3 var server = {
4   base: '/Users/Icweex/Sites/uiagile_repo', //ROOT FOLDER
5   port: 80
6 };
7
8 var browser = {
9   proxy: 'http://uiagile.com/',
10  port: 3000
11 };

```

Následne sa môže spustiť kompilácia príkazom:

```
gulp watch
```

Ako posledné nastavenie je Socket.io, ktorému je potrebné prepísať adresu hosta na strane klienta (viď. Obrázok nižšie) v súbore `/webroot/js/aoweb/components/ClientComponent.js`.

```

1 ClientComponent = function () {
2
3   aoweb.component.call(this);
4
5   this.host = 'http://localhost:8082/';
6
7   this.loaderActive = undefined;
8   this.socket = undefined;
9 };

```

Na spustenie Node servera z priečinka `/node/` použijeme nasledovný príkaz, ktorý ostane pracovať aj po odhlásení používateľa zo servera.

```
nohup node Server.js &
```

4.3 Nasadenie webovej stránky

Na rozdiel od nasadenia produkčnej verzie webová stránka nemá žiadnu požiadavku na inštaláciu nakoľko sa jedná len o HTML web. Pri nasadení stránky do root directory `/var/www/html` stačí len použiť nasledujúci príkaz Gitu:

```
Git clone git @bitbucket.org:POUZI VATELSKE_MENO/tea_m_website.git
```

Pričom je potrebné mať najprv vytvorený repozitár s požadovanou webovou stránkou

4.4 Aktualizácia dokumentácie

Aktualizácia prebieha najprv úpravou stránky lokálne a následne sa nahrá s aktualizovanými súbormi do repozitára.

Pridanie nových dokumentov prebieha v štyroch krokoch:

- Nahratie nového dokumentu do adresára stránky, v našom prípade do priečinku */assets/docs/*
- Pridanie nového li elementu do index.html, pričom odkaz na dokument bude smerovať na novú adresu dokumentu z bodu 1

```
<li>
  <a class="btn-pdf btn btn-lg btn-common" href="assets/docs/zapisnice/Zapisnica5.pdf" target="_blank" style="width:100%;">
    <i class="fa fa-file-pdf-o"></i>
    27.10.2017
  </a>
  <p class="description hide"><b>Popis</b>: Pokračovanie 2.šprintu</p>
</li>
```

- Publikovanie aktualizácií do repozitára
- Stiahnutie aktualizácií na servery z priečinka */var/www/html* pomocou príkazu *git pull*