

ZÁPADOČESKÁ UNIVERZITA  
FAKULTA APLIKOVANÝCH VĚD  
KATEDRA KYBERNETIKY



## Dokumentace distribuci projektu ITE-YELLOW

*Martin Hamar, Radek Kaupe, Samuel Kokoška*

# 1 Úvod

V rámci předmětu KKY/BSOI jsme si vytvořili a nastavili virtuální stroj s operačním systémem Linux. Tento stroj je momentálně používán k provozu aplikace vyvíjené v rámci předmětu KKY/ITE. Zde je na aplikaci a zde je odkaz na Githubový repozitář, kde jsou k dispozici veškeré zdrojové kódy aplikace jak části ITE, tak části BSOI. Tento referát by měl popsat virtuální stroj, jeho nastavení, činnosti a realizaci způsobů přihlášení do již zmíněné aplikace.

## 2 Řešení

### 2.1 Uživatelské účty

Na virtuálním stroji jsou tři uživatelské účty - každý pro jednoho člena týmu. Tyto účty mohou používat sudo a mohou se na stroj přihlásit přes ssh bez hesla díky jejich veřejným ssh klíčům. Přímé přihlášení na root má povoleno jen zakladatel.

### 2.2 Zálohování

Zálohování databáze bylo implementováno pomocí nástroje Borgbackup, specificky funkcí prune pro automatizaci celého procesu k ukládání každodenní zálohy na účty skokoska a radek, v případě poruchy či potřeby vrátit databázi do předchozího stavu tedy existují dvě zálohy.

Na každém účtě je složka "backup" obsahující zašifrovanou zálohu pod defaultním heslem "gobackImadeamistake".

### 2.3 Provoz distribuované aplikace

Součástí aplikace je PostgreSQL databáze a dva python skripty, jeden z nich řeší MQTT subscribena, abychom mohli data přijímat, validovat a ukládat a druhý tvoří backend naší webové aplikace. K založení PostgreSQL databáze byl vytvořen python skript, který stačí spustit pouze jednou, aby vytvořil tabulky. Skripty na subscribena a backend musí běžet neustále, aby aplikace mohla být v provozu. Subscriber a backend spolu 'komunikují' pouze přes databázi, ke které mají oba skripty přístup. Spouštění skriptů subscriberu a backendu po pádu a logování výstupů zprostředkovávají "launchery"- python skripty, které ty původní provozují v nekonečném cyklu s try-except blokem a přeměňují jejich stdout (výstup na konzoli) do textových souborů. Launchery samotné jsou spuštěny pomocí systemd služeb, které jsou nastaveny, aby se spouštěly při zapnutí/resetu virtuálního stroje. Z důvodu velkého množství výpisů subscribera jsou jeho výpisy uchovávány jen z posledních dvou dní v .txt souborech po hodinách. O to se stará separátní kombinace python skriptu a systemd služby. Nasazení verzí se může provést pomocí programu WinSCP, kterým se nové soubory nahrají na virtuální stroj.

### 2.4 Přihlášení a biometrická autentizace

Při implementaci jsem využil učitelů dodaných zdrojů, ve formě skriptů `faceid_server.py`, `extract_embeddings.py` a `train.sh`.

#### Rozšíření backendu o FaceID

V prvním skriptu se nacházejí Tornado Handleři, jeden zajišťující komunikaci s frontendem v rámci posílání a ukládání fotek a druhý v rámci detekci obličeje ve fotce. Předtím oba dva handleři ???fungovali??? v rámci jedné URL adresy. Já je rozdělil. Aplikace funguje následovně:

1. Nepřihlášený uživatel se může buď přihlásit přes uživatelské jméno a heslo (vstupní stránka), přihlásit se přes FaceID nebo se zaregistrovat

2. Po registraci musí uživatel čekat na schválení adminem, jinak se nepřihlásí (ani v případě, že již fotky má uložené v rámci biometrické autentizace)
3. Přihlášený uživatel má přístup k celé aplikaci, včetně možnosti nafotit si své fotky pro FaceID a natrénovat model pro autentizaci obličejů

### **Uživatelské účty a identity**

Máme celkem 9 uživatelských účtů pro přihlášení, pro:

- 4 vyučující
- 3 členy týmu
- 1 admin účet, ke kterému mají přístup všichni členové týmu

Identit je 8. Jsou vytvořené pro všechny členy týmu, jeden kategorie ‘unknown’ (aby se ‘nové’ tváře nemohli přihlásit, jinak by je systém identifikoval jako validního uživatele) a 3 identity učitelé, aby mohlo dojít k právě zmíněné identitě ‘unknown’. Kategorie ‘unknown’ má celkem 17 fotek, z toho 3 neobsahují lidský obličej, aby se neuronová síť naučila přiřazovat neplatné fotografie neautorizovanému uživateli. Každý z členů má 10 až 25 fotek, které si každý sám nafotil. Od každého vyučujícího jsme dostali 10 - 25 fotek. Každý s různým počtem, aby šlo testovat úspěšnost identifikace. Rozhodovací práh je nastaven na alespoň 90% pravděpodobnost identifikace uživatele. Je nutné podotknout, že identita ‘unkown’ není vůbec uložena v databázi a nemá možnost se do aplikace přihlásit.

### **Funkčnost přihlašování**

V případě, že se člověk přihlásí, je mu nastaven Json Web Token (JWT). Ten řeší zda je uživatelská session je validní či nikoliv. Pokud jsou uživatelská credentials v pořádku a uživatel je zaregistrován a schválený adminem je mu JWT uložen jako secure cookie. Všechny další žádosti nyní automaticky kontrolují zda je JWT neexpiroval nebo s ním nebylo manipulováno. JWT má totiž nastavenou dobu expirace jedné hodiny a obsahuje ‘podepsán’ tajným klíčem, který je znám pouze serveru.

### **Odhlašování**

Pro odhlašování byl vytvořen Tornado Handler, který vymaže token obsahující cookie a přesměruje uživatele zpět na přihlašovací stránku.

## **3 Závěr**