Systém pro kontejnerizaci linuxového školního serveru

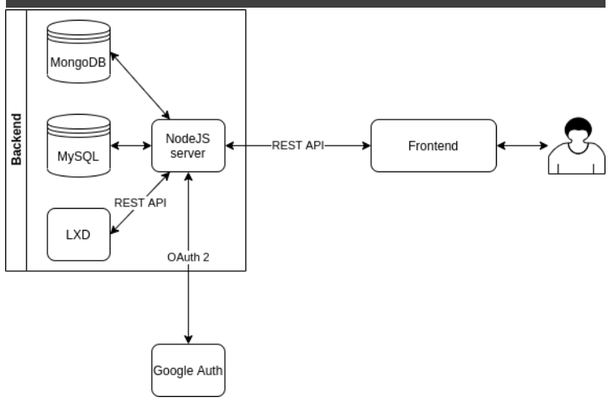
Vladimír Vávra, Kryštof Havránek, Josef Litoš

Gymnázium, Praha 6, Arabská 14, 29.4.2021

Abstrakt

Cílem práce je vytvořit kontejnerizační systém pro operační systém GNU/-Linux. Ten by měl v jednoduchém a přehledném uživatelském prostředí umožnit žákům vytvořit si vlastní linuxový server, na kterém si poté mohou hostovat svoje aplikace, webové stránky, či videohry. Systém by se měl postarat o to, aby zdroje serveru byly spravedlivě rozdělené mezi jednotlivé uživatele.

Architektura



Jádro práce stojí na LXD, což je kontejnerizační systém zabudovaný přímo do linuxového kernelu. Jedná se o velice efektivní řešení, protože jednotlivé kontejnery mohou sdílet společný kernel. Jsou tedy menší a rychlejší, než kdyby se jednalo o samostatný virtuální počítač.

Backend s LXD komunikuje prostřednictvím REST API specifikovaného v dokumentaci LXD. Je napsán v NodeJS a ke svému fungování využívá dva databázové systémy. Většinu informací ukládá do MySQL databáze, ale kvůli komplikacím s LXD byla později zavedena i MongoDB.

Frontend je psán v ReactJS a s backendem komunikuje prostřednictvím vlastního REST API (jehož podrobná specifikace je uvedena v dokumentaci a v souboru USER.yaml.

Na implementaci autentifikace byl poté použit Google Auth, jelikož všichni studenti školy mají svůj Google účet.

Funkce

Hierarchie jednotlivých komponent projektu sestává z uživatelů, projektů, které těmto uživatelům náleží, a kontejnery, což jsou samotné virtuální servery, jež si uživatel může v rámci projektu vytvořit. Na nich poté může uživatel hostovat a testovat webových stránky, aplikace, či jiné.

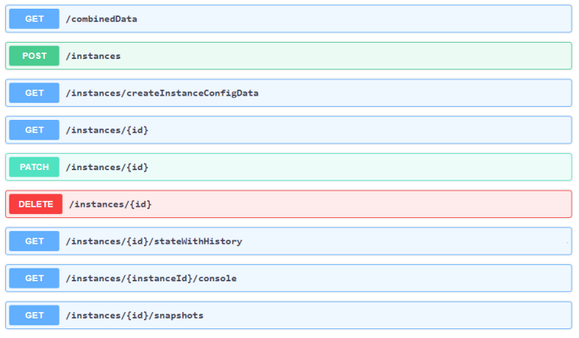
Hlavními funkcemi aplikace jsou:

- Možnost vytvořit a smazat kontejnery a projekty. Kontejnery vždy musí mít specifikované limity, projekty mohou.
- Změna limitů a dalších nastavitelných možností.
- Přehledné zobrazení informací a aktuálního stavu u kontejnerů, projektů i uživatele
- Připojení určitých portů kontejneru k internetu pomocí subdomény

Vedlejšími funkcemi aplikace jsou poté:

- Vytvoření a stažení zálohy kontejneru
- Integrovaný terminál pro správu kontejneru přímo z aplikace.
- Logování stavu každých pro vykreslování dlouhodobého stavu kontejneru

API



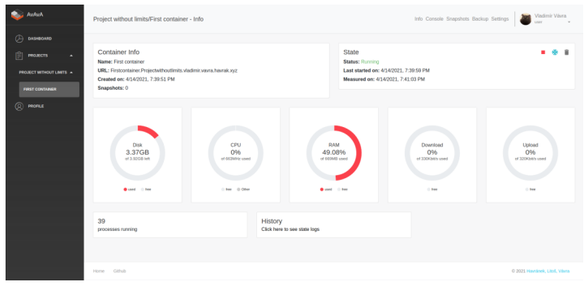
API specifikace pro komunikaci mezi frontendem a backendem bylo vytvořeno ve Swagger editoru a de facto detailněji specifikuje aktuální funkce aplikace. Na obrázku můžete vidět část jeho dokumentace, zbytek je možné nalézt v souboru USER:yaml.

Frontend

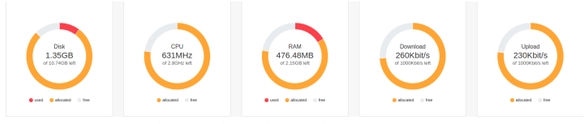
Kód byl napsaný v Javascriptu, konkrétně ReactJS a preprocessoru SASS. Velice důležité jsou i knihovny react-bootstrap a material-ui, pomocí nichž byla napsána většina komponent. Pro základ frontendu byl použit template (viz dokumentace), ze kterého ovšem v aplikaci zůstalo jen velice málo.

Co se technických záležitostí týče, tak o komunikaci s backendem se stará superagent klient vygenerovaný Swagger editorem a konzistence dostupnost dat na frontendu je vyřešena pomocí Redux frameworku.

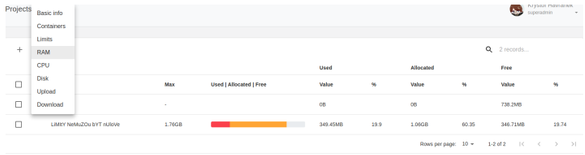
Frontend je rozdělen na tři hlavní kategorie — uživatelská, projektová a kontejnerová. Mezi nimi se dá navigovat pomocí navbaru a sidebaru, kdy logiku navigování zajišťuje React-router.



Na obrázku výše můžete vidět základní pohled na kontejnerovou kategorii. Jsou zde vidět základní informace a aktuální stav. Nahoře je poté možné vybírat mezi jednotlivými pohledy dané kategorie, např. konzole do kontejneru, získání zálohy či změnu nastavení. Na obrázku níže můžete vidět, jak vypadají grafy u projektů.



Na nejspodnějším obrázku můžete poté vidět jeden z pohledů na tabulku projektů. Podobně vypadá i pohled na tabulku kontejnerů.

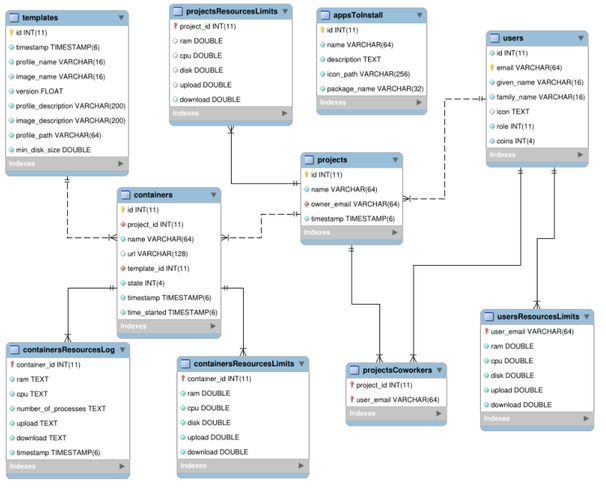


Backend

Centrem Backendu je NodeJS server, na který se klient dotazuje specifikovaným API. Za pomocí LXD, MySQL a MongoDB databází a vlastní logiky serveru se po žádosti seskládá odpověď, která je odeslána na frontend.

Veškerá komunikace projekt s LXD probíhá přes REST API. Pro zjednodušení požadavků obsahuje backend soubor lxdRoute.js, který zbytku backendu umožňuje snadno a efektivně přistupovat k údajům LXD. Poskytuje metody pro každou jednoduchou akci potřebnou pro, jak zjištění/doplnění dat jistých objektů – projekty, kontejnery, jejich stav, zálohy nebo i snapshoty –, tak provedení určitých akcí – změnit stav/limity kontejneru, spustit příkaz/konzoli, nahrát/stáhnout soubory/zálohy... Výsledky jsou zpracovány do využívaných údajů, které se vrátí volajícímu.

Na níže uvedeném diagramu můžete vidět uspořádání MySQL databáze. Jsou v ní uloženy veškerá data, které nelze přímo získat z LXD či jsou přidány naší apliakcí navíc—např. uživatelské limity. MongoDB poté slouží na uchování logování stavu kontejnerů.



Pro připojení k internetu využívá aplikace volně dostupnou haproxy, jejíž instance běží ve stejnojmenném kontejneru. Hostovací stroj totiž standardně nemůže přistupovat na kontejnery prostřednictvím .lxd domény. Do kontejneru haproxy je aktuálně přesměrována traffic ze šesti portů – 80, 443, 2222, 3000, 5000, 9000