



Střední průmyslová škola strojní
a elektrotechnická a Vyšší odborná škola,
Liberec 1, Masarykova 3

APLIKACE PRO SPRÁVU A MONITORING NAPÁJENÍ VIRTUÁLNÍCH SERVERŮ

Maturitní práce

Autor
Obor
Vedoucí práce
Školní rok

Adam Antoš
Informační technologie
Mgr. Michal Stehlík
2021/2022

Přihláška k maturitní práci

Jméno a příjmení studenta

Antoš, Adam

Třída

P4

Název práce

Aplikace pro správu a monitoring napájení virtuálních serverů

Školní rok

MP2021/22

Přidělené role
Vedoucí práce
Oponent

Podpis



Stehlík, Michal

Obecná ustanovení	Vypracování a odevzdání práce proběhne v souladu s platnými normami (vyhláška 177/2009 Sb.) a aktuálním dokumentem "Pokyny k vypracování prací" vydaným školou.
	Práce bude hodnocena z hlediska jejího praktického využití, zvládnutí dokumentace po věcné i formální stránce a obhajoby celé práce. Student byl seznámen s kritérii hodnocení maturitní práce.
	Práce bude odevzdána ve dvou stejnopisech vázaných pevnou nebo kroužkovou vazbou.
	Veškeré náklady na MP včetně vyhotovení obou tištěných kopií si student hradí sám.
Licenční ujednání	Ve smyslu § 60 (Školní dílo) autorského zákona č. 121/2000 Sb. poskytují SPŠSE a VOŠ Liberec výhradní a neomezená práva k využití této mé maturitní práce.
	Bez svolení školy se zdržím jakéhokoliv komerčního využití mé práce.
	Pro výukové účely a prezentaci školy se vzdávám nároku na odměnu za užití díla.

Finanční rozvaha - odhad celkových nákladů

V Kč	Náklady celkem	Hrazené školou
Výrobní	0	0
Na služby	0	0

Jedná se o MP, jejíž vypracování si škola vyžádala? **Ano – Ne**
Podpis studenta (vyjadřuje souhlas s uvedenými údaji a ujednáními)

V Liberci 02.11.2021

Podpis


Konzultant

Práci podporuji

Podpis

Předmětová komise

Práci doporučuji

Podpis

Třídní učitel

Práci doporučuji

Podpis

Garant oboru

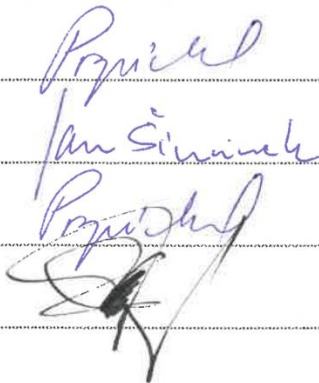
Práci doporučuji

Podpis

Ředitel školy

Práci doporučuji

Podpis



Zadání maturitní práce

Název

Aplikace pro správu a monitoring napájení virtuálních serverů

Předmět

PRG,
WEB

Téma

Cílem je vytvořit mechanismy pro správu a monitoring napájení virtuálních serverů (VMware vSphere). Výsledkem bude aplikace implementující rozhraní pro tyto činnosti.

Použité prostředky

Virtuální stroje, ASP.NET

Cíle práce

1	Průběžný monitoring stavu UPS zařízení napájejících fyzické host servery včetně ukládání zjištěných dat pro pozdější analýzu
2	Reakce na výpadek napájení: dle stavu baterií zahájit postupné vypínání virtuálních serverů (dle nastavených preferencí), následné vypnutí host serverů a odstavení virtuálního úložiště
3	Reakce na obnovení napájení: kontrola stavu host serverů (případné zapnutí), aktivace úložiště, postupné zapnutí virtuálních serverů (dle nastavených preferencí)
4	Ošetření situace, kdy po výpadku je zahájeno vypínání virtuálních serverů a dojde k obnovení napájení. Ukončit vypínání virtuálních serverů a vypnuté virtuální servery opět spustit.

Osnova práce

1	Seznámení s ovládáním stávajících zařízení
2	Komunikace se zadavatelem (specifikace funkcí atd.)
3	Návrh aplikace (databáze, rozvržení, ovládání)
4	Průběžné programování aplikace
5	Nasazení aplikace do produkce

Anotace

Účelem aplikace je vyřešení problematiky řízení napájení virtuálních serverů provozovaných v datovém centru Městského úřadu v Semilech založeném na HW/SW platformě HPE a serverové virtualizaci VMware vSphere.

Vychází z aktuálního stavu, kdy je vypínání/spouštění virtuálních serverů řešeno dvojicí powershellových skriptů, které se spouští na samostatném fyzickém serveru (sloužícím jako NAS), které má značné nedostatky.

Přináší řešení, které dokáže okamžitě reagovat na změny napájení monitorovaných UPS zařízení, takže nedochází ke zbytečnému vypínání virtuálních serverů při krátkém výpadku napájení. Rovněž aplikace zabraňuje neočekávanému vypnutí virtuálních serverů z důvodu vybití baterií UPS zařízení.

Summary

The purpose of the application is to solve the problem of power management of virtual servers operated in the data center of the Municipal Office in Semily based on HW/SW platform of HPE and server virtualization VMware vSphere.

It is based on the current state, when shutting down/starting virtual servers is solved by a pair of powershell scripts, which run on a separate physical server (serving as a NAS), which has significant shortcomings.

It provides a solution that can respond immediately to power changes in monitored UPS devices, so there is no unnecessary shutdown of virtual servers during a short power outage. The application also prevents the virtual servers from shutting down unexpectedly due to the UPS batteries running low.

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne 14.3.2022

.....

Adam Antoš

Obsah

Úvod.....	1
1 Celkové fungování aplikace.....	2
1.1 Scénáře pro spouštění a vypínání.....	2
1.1.1 Spouštění.....	2
1.1.2 Vypínání.....	3
1.2 Celková konfigurace aplikace.....	3
1.3 Uživatelské rozhraní pro konfiguraci.....	3
1.3.1 Stránka pro zobrazení stavu systému.....	4
1.3.2 Konfigurace aplikace (stránka Settings).....	5
1.3.3 Stránky pro správu seznamu UPS.....	5
1.3.4 Stránky pro virtuální servery.....	6
1.3.5 Stránky pro hostitelské servery.....	7
1.3.6 Správa uživatelů.....	8
2 UPS.....	9
2.1 OID.....	9
2.2 Problém při aktualizaci firmwaru síťového modulu UPS.....	9
3 Komunikace se servery.....	11
3.1 Komunikace s vCenter serverem.....	11
3.1.1 Autentifikace.....	11
3.1.2 Získání seznamu virtuálních serverů.....	12
3.1.3 Ovládání napájení jednotlivých virtuálních serverů.....	13
3.2 Komunikace s fyzickými servery.....	13
3.2.1 Komunikace s iLO.....	13
3.2.2 Komunikace se systémem VMWare vSphere ESXi.....	14
4 Služby.....	15
4.1 Služba pro práci s databází (DbService).....	15
4.2 Služba pro e-mailové notifikace (MailService).....	16
4.3 Služba pro definici scénářů (ScenarioService).....	16

4.4	Služba pro práci s UPS (UPSService)	17
4.5	Služba pro práci s virtuálními servery (VMService)	17
4.6	Služba pro práci s fyzickými servery (ServerService)	17
5	Logování dat.....	18
5.1	Logování stavů UPS	18
5.2	Logování stavů serverů a celého systému	19
6	Nasazení aplikace	20
6.1	Konfigurace před nasazením aplikace	20
6.2	Vlastní nasazení aplikace	20
6.3	Ladění a testování aplikace	21
	Závěr.....	22
	Seznam zkratek a odborných výrazů.....	24
	Seznam obrázků.....	26
	Použité zdroje	27

Úvod

Důvodem vzniku této práce je řízení napájení virtuálních serverů Městského úřadu v Semilech, které je nyní řešeno nevyhovujícím způsobem pomocí powershellových skriptů. Což má nedostatek především v případě, když se vypínací skript spustí na základě povelu od SW pro monitoring UPS (HPE Power Protector), tak již nejde zastavit, a i když mezitím výpadek proudu skončí, tak se servery vypnou. Následné zapnutí serverů je při tomto řešení velmi zdoluhavé (skript musí proběhnout opakovaně) a informační systémy úřadu jsou po dlouhou dobu nefunkční. Rovněž nelze řídit dobu, po které se servery začnou vypínat v závislosti na aktuálním vytížení serverů. V praxi tedy může výpadek proudu, který přesáhne o pár sekund nastavený čas zahájení vypínání serverů způsobit hodinový výpadek informačních systémů úřadu.

Proto jsem se rozhodl vytvořit aplikaci, která bude pravidelně kontrolovat stav napájení UPS a dle těchto dat řídit běh virtuálních serverů a diskových polí, která jsou rovněž realizována virtuálně (HPE Store Virtual). Aplikace bude implementována a provozována na nezávislém mini PC (Intel NUC) napájeném nezávislou UPS. Pro komunikaci se serverovými UPS bude využito standardizované rozhraní SNMP. Pro komunikaci s virtuálním serverem VMWare vCenter, který řídí obsluhu virtuálních serverů na obou fyzických serverech, bude využito REST API rozhraní vRealize Automation. Pro ovládání základních funkcí fyzických serverů bude využito REST API rozhraní HPE iLO.

Zdrojový kód této aplikace je umístěn ve verzovacím systému GitHub na adrese: https://github.com/pslib-cz/MP2021-22_Antos-Adam_Aplikace-pro-spravu-a-monitoring-napajeni-virtualnich-serveru.

1 Celkové fungování aplikace

Tato aplikace je navržena pro řízení, a především kontrolu napájení virtuálních serverů, data o stavu jednotlivých UPS jsou vyčítána pomocí protokolu SNMP a unikátních OID, která se dají definovat v grafickém prostředí aplikace (stejně jako jednotlivé UPS), v případě potřeby jsou volány metody pro práci s virtuálními servery nebo jednotlivými ESXi host servery. Aplikace umožňuje také zobrazení dat o UPS ve svém grafickém rozhraní, případně možnost ručního ovládání serverů.

1.1 Scénáře pro spouštění a vypínání

Hlavní výhodou těchto scénářů je oproti původnímu řešení možnost zastavit spouštění nebo start, např. servery se začnou vypínat z důvodu déle trvajícího výpadku elektřiny, ale při vypnutí několika virtuálních serverů se dodávka elektrického proudu obnoví, což by ve stávajícím řešení nemělo vliv na již běžící skript, takže by se všechny servery vypnuly. Tato aplikace ale před ukončením jednotlivých serverů kontroluje pomocí SNMP dotazu stav UPS a v případě změny a dostatečné kapacity baterií volá opačnou metodu spuštění, případně vypnutí (dle aktuálně probíhající operace).

1.1.1 Spouštění

V případě spouštění je nejprve nutné zkontrolovat, zda mají UPS dostatečnou kapacitu pro spuštění, případně i pro vypnutí v případě dalšího výpadku elektřiny. Hostitelské servery ESXi by se měly spouštět automaticky po spuštění UPS, ale vzhledem k tomu, že může nastat stav, kdy se servery vypnou, ale UPS nevybijí (nevypnou) je prováděna kontrola jejich spuštění. V případě, že se servery nespustí, jsou spuštěny pomocí volání metod REST API na rozhraní HPE iLO. Poté se začnou spouštět virtuální servery pro virtuální diskové pole (HPE Store Virtual). Následně je nutné aktualizovat seznam virtuálních diskových polí, aby byly k dispozici (viditelné) pro virtualizační systém a spustit řídicí virtuální server vCenter. Po spuštění vCenter serveru už bude řízení dále probíhat pomocí jeho REST API rozhraní, budou spuštěny jednotlivé virtuální servery dle nastaveného pořadí spouštění (dle jejich důležitosti).

1.1.2 Vypínání

Vypínání virtuálních serverů začne v momentě, kdy serverové UPS poběží na baterie a bude zbývajícím časem na baterii UPS menší než nastavený limit. Postupně se vypnou jednotlivé virtuální servery podle nastaveného pořadí vypínání, poté se vypne server vCenter. Následně se vypnou servery pro virtuální diskové pole HPE Store Virtual (VSA). Poté v závislosti na zbývajícím kapacitě baterií je vypínací proces pozdržen a čeká, zda nedojde k obnově napájení. Pokud se nedočká, tak před vybitím baterií na definovaný práh včas vypne fyzické host servery.

1.2 Celková konfigurace aplikace

Konfigurace aplikace je uložena ve vlastní tabulce v databázi, díky tomu je připravena na více možných konfigurací (např. z důvodu testování), obsahuje adresu a přihlašovací údaje k serveru vCenter, nastavení parametrů SMTP pro odesílání e-mailů o stavu, název řešení, prodlevy mezi kontrolami stavů serverů a aktuální stav systému (probíhající vypínání, spouštění atd.)

Další součástí konfigurace je seznam UPS, jejichž stav je kontrolován pomocí mikroslužby běžící na pozadí na serveru IIS, která v pravidelných intervalech volá vnitřní API. V případě potřeby a dostatečné kapacity baterií je spuštěna služba pro vypínání, případně spouštění.

V databázi je také uložen seznam virtuálních serverů s prioritou spouštění a vypínání. Stejně jako seznam fyzických (hostitelských) serverů, ke kterému jsou přiřazeny virtuální storage servery a seznam jejich datastorů (úložišť virtuálních serverů), jejichž přístupnost je kontrolována při spouštění systému.

1.3 Uživatelské rozhraní pro konfiguraci

Jelikož se jedná o webovou aplikaci ASP.NET Core je její nastavení a kontrola realizována pomocí webového rozhraní s CSS frameworkem Bootstrap. Administrátor může po přihlášení konfigurovat nastavení aplikace. K přihlášení slouží výchozí přihlašovací údaje, které lze později změnit nebo přidat další uživatele.

1.3.1 Stránka pro zobrazení stavu systému

Jedná se o úvodní stránku, kterou administrátor vidí ihned po přihlášení do webové aplikace. Tato stránka slouží pro rychlou kontrolu stavu všech sledovaných zařízení. Nejprve je zde údaj o aktuálním stavu systému (vypnuto, spouštění, vypínání, spuštěno, neznámý) a tlačítko, které odkazuje na stránku se seznamem virtuálních serverů a jejich stavem.

Přehledně jsou zde zobrazeny informace o aktuálním stavu UPS, které jsou pro přehlednost barevně odlišeny (zelená – připojeno k el. síti, oranžová – baterie, červená – chyba komunikace).

Také je zde zobrazen stav fyzických (host) serverů, který je vyčítán pomocí RESP API z HPE iLO, stejně jako u UPS jsou stavy barevně odlišeny (zelená – spuštěný, oranžová – vypnutý, červená – nelze se připojit). U každého fyzického serveru je tlačítko, které přesměruje uživatele na stránku s podrobnými informacemi, kde jsou krom stavu z HPE iLO zobrazeny i stav operačního systému vSphere, virtuálních storage serverů a jejich datastorů, barevně odlišeny jsou velmi podobně jako stav z HPE iLO.

MMNVS
Stav
UPS
Virtuální servery
Host servery
Nastavení
Logy
Správa uživatelů
Změna hesla
Odhlásit

Monitoring a management napájení virtuálních serverů

Stav systému, UPS a host serverů

Aktuální stav systému: Spuštěno

Zobrazit stav virtuálních serverů

UPS 1 (192.168.100.1)	UPS 2 (192.168.100.2)	Test UPS 2 (192.168.100.2)
Připojena k el. síti (100%)	Připojena k el. síti (100%)	Připojena k el. síti (100%)
Zbývajíc čas 1931 s	Zbývajíc čas 1369 s	Zbývajíc čas 1369 s
Zátěž 17%	Zátěž 24%	Zátěž 24%

Stav iLO serveru HPE ESXi 3	Stav iLO serveru HPE ESXi 1
Server je spuštěn	K iLO tohoto serveru se nelze připojit!
Adresa serveru 192.168.100.41	Adresa serveru 192.168.100.11
Adresa iLO 192.168.100.42	Adresa iLO 192.168.100.12
Zobrazit podrobné informace o stavu	Zobrazit podrobné informace o stavu

© 2022 - MMNVS - Adam Antoš

Obrázek 1 Úvodní stránka aplikace – zobrazení stavových dat

1.3.2 Konfigurace aplikace (stránka Settings)

Tato stránka slouží především pro nastavení obecných parametrů využívaných v celé aplikaci, jako je připojení k serveru vCenter nebo odesílání informačních e-mailů správci.

Po uložení adresy REST API, přihlašovacího jména a hesla je aplikací odeslána žádost o přidělení API key a podle její odpovědi je uživateli pomocí informačního oznámení sdělena úspěšnost, případně neúspěšnost připojení k serveru.

Podobné je to v případě nastavení e-mailových notifikací, kdy je na e-mailovou adresu administrátora odeslána testovací zpráva a je zobrazeno informační oznámení s výsledkem. Uživatel tak může zkontrolovat správnost připojení k SMTP serveru pomocí přihlašovacích údajů, adresy a portu, který zadal.

Na tomto místě lze také nastavit čas v sekundách pro pauzu mezi kontrolou stavu serverů při vypínání nebo spouštění. Vzhledem k tomu, že připojení ke každému druhu serverů je řešeno trochu odlišně, tak je u každého možné nastavit jiný čas pauzy mezi kontrolou stavu.

1.3.3 Stránky pro správu seznamu UPS

Přímo na stránce se seznamem UPS, které systém monitoruje je možné přidávat nové UPS, upravit, smazat nebo zobrazit aktuální stav stávajících UPS.

Při přidávání UPS je uživatel vyzván k zadání názvu, IP adresy síťového rozhraní, a jednotlivých SNMP OID pro stav, kapacitu baterie, zbývajících času na baterii a aktuální vytížení. Po přidání je pomocí protokolu SNMP odeslána žádost o informace o modelu a výrobci, které jsou automaticky přiřazeny k UPS. Uživatel je o tomto kroku a jeho úspěšnosti informován pomocí informačního oznámení, tím je zajištěna kontrola funkčnosti připojení.

Pokud uživatel chce upravit (i automaticky načtené) údaje UPS, může tak učinit pomocí stránky pro úpravu, případně UPS zcela vymazat díky stránce pro odstranění.

Stavová stránka vyčítá přes SNMP údaje přímo z UPS pomocí OID identifikátorů uložených do databáze při přidání (stejně jako se děje při automatické kontrole stavu).




Zde uživatel vidí aktuální data, aniž by se musel přihlašovat k webovému rozhraní jednotlivých UPS.

1.3.4 Stránky pro virtuální servery

Nejdůležitější stránkou pro virtuální servery je hlavní stránka s jejich seznamem. Tlačítko „Načíst ze serveru vCenter“ umožňuje pomocí údajů uložených v konfiguraci aplikace načíst z vCenter serveru specifické identifikátory VMID a názvy všech virtuálních serveru do databáze aplikace. Také je k serverům automaticky přiřazena informace o tom, zda se mají automaticky spouštět při zapnutí monitorovaného systému, na základě jejich aktuálního stavu (vypnutý – nebude se spouštět, spuštěný – bude se spouštět). Pomocí metody automatického přidání je také identifikován vCenter virtuální server a označen příslušným záznamem v databázi, v případě chybné identifikace je možné ho ručně změnit uživatelem. Díky tomu nemusí uživatel přidávat seznam všech serverů ručně a je zajištěna 100% správnost všech údajů. V případě, že se nějaké virtuální servery automaticky nenačtou nebo chce uživatel přidat virtuální servery, ke kterým aktuálně nemá k dispozici žádný přístup, může využít možnosti ručního přidání, kde ale není zajištěna kontrola správnosti údajů. Na stránce se seznamem je také možné změnit pořadí spouštění (a vypínání) virtuálních serverů pomocí šipek u jednotlivých položek. U každého virtuálního serveru je zobrazený jeho aktuální stav (vypnuto, spuštěno, nenalezen).

Virtuální servery

[Ruční přidání](#) [Načíst z vCenter](#)

VM id	Název	Pořadí při startu/vypínání	Zapnout?	Stav	
vm-38	vcenter	vCenter	<input checked="" type="checkbox"/>	Spuštěno	Upravit Stav Odstranit
vm-104	Win10-21H1-admin	1  	<input checked="" type="checkbox"/>	Spuštěno	Upravit Stav Odstranit
vm-1904	SysServerW2016	2  	<input checked="" type="checkbox"/>	Spuštěno	Upravit Stav Odstranit
vm-19092	W2016-Tests	3  	<input type="checkbox"/>	Vypnuto	Upravit Stav Odstranit
vm-11207	AthenaSrv2016	4  	<input checked="" type="checkbox"/>	Spuštěno	Upravit Stav Odstranit
x	Spatny-server	5  	<input type="checkbox"/>	Nenalezeno	Upravit Stav Odstranit
vm-62	SQL 2017 standard	6  	<input checked="" type="checkbox"/>	Spuštěno	Upravit Stav Odstranit

Obrázek 2 Seznam virtuálních serverů a informace o jejich stavu

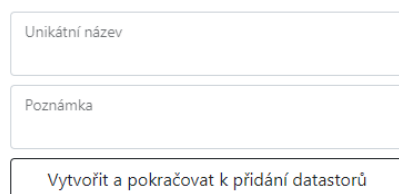
Dále je zde krom stránek pro editaci (zde je možné vybraný server nastavit jako vCenter server – v případě špatné automatické detekce) a odstranění i stránka pro

zobrazení veškerých informací o serveru (od aktuálního stavu až po seznam připojených disků).

1.3.5 Stránky pro hostitelské servery

Tyto stránky jsou navrženy pro ruční přidání jednotlivých hostitelských (fyzických) serverů, které obsahují informace o serveru (název a výrobce), IP adresu, přihlašovací údaje (pro řízení pomocí PowerCLI). Pro hostitelské servery se přidávají virtuální storage servery, kde se eviduje jejich unikátní název. A pro virtuální storage

Přidání virtuálního storage serveru na host serveru HPE ESXi 3

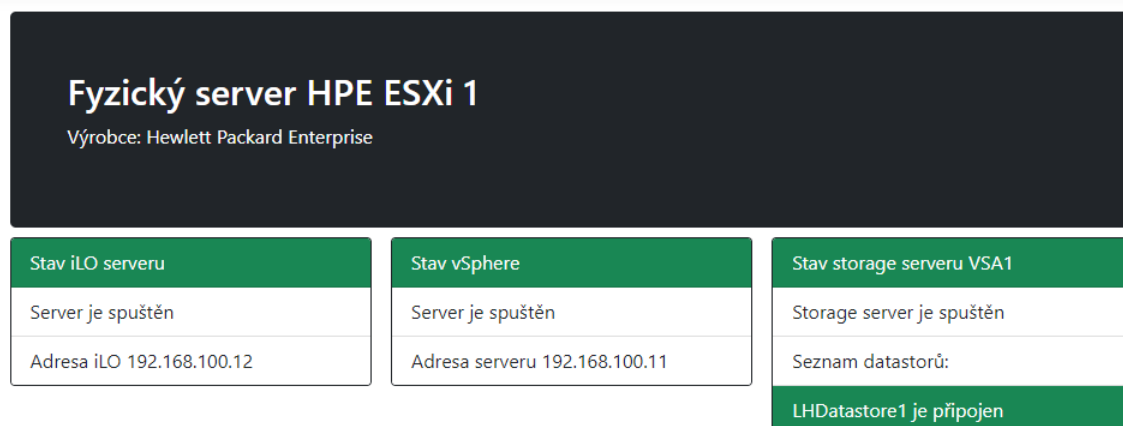


Obrázek 3 Formulář pro přidání VSA serveru

servery se přidávají jejich datastory, kde je důležitý především jejich unikátní název pro kontrolu jejich přístupnosti při spouštění.

Celkově se stránky pro správu těchto tří entit velmi podobají a vzájemně na sebe navazují. Stránky pro přidávání na sebe, pro lepší orientaci uživatele, odkazují.

Při přidávání každé entity, je zavolána metoda pro kontrolu, zda přidávaná entita opravdu existuje a je k dispozici (spuštěna). Např. při přidávání virtuálního storage serveru na hostitelský server je zavolána metoda, která se pomocí PowerCLI připojí na fyzický server a zkontroluje, zda je na něm spuštěný virtuální storage server. Následně je o výsledku uživatel informován pomocí informačního oznámení. Správnost nastavení a stav systému je také možné zkontrolovat pomocí stavové stránky fyzického serveru, kde je zobrazen stav připojení k HPE iLO, systému vSphere a virtuálním storage serverům s jejich datastory.



Obrázek 4 Zobrazení stavu fyzického serveru

Jelikož je ve firmě aktuálně na každém fyzickém serveru pouze jeden virtuální storage server s jedním datastorem, tak by bylo možné vše spojit do jedné entity. Rozhodl jsem se ale pro rozdělení do tří entit z důvodu možných budoucích změn a větší flexibilitnosti aplikace.

1.3.6 Správa uživatelů

Další nezbytnou součástí je stránka pro správu uživatelů, respektive administrátorů celé aplikace, kteří mohou upravovat konfiguraci, přidávat UPS a virtuální i fyzické servery. Jak již bylo zmíněno, v aplikaci je výchozí účet administrator, na něm může uživatel (stejně jako na každém jiném účtu) změnit heslo, případně jeho prostřednictvím založit úplně nový účet (i účty pro ostatní uživatele) a původní administrátorský účet odstranit. Aplikace je do budoucna připravena pro využívání rolí uživatelů, ale v aktuálním nasazení ve firmě toto není potřeba (aplikace je určena pouze pro administrátory).

2 UPS

Aplikace komunikuje s UPS zařízeními přes LAN pomocí NuGet balíčku SNMP Library od společnosti LeXtudio Inc. přes protokol SNMPv1. Získání dat přes protokol SNMP je pomocí jednoznačných OID identifikátorů.

2.1 OID

OID je jednoznačný identifikátor užívaný v protokolu SNMP, existují obecná OID (např. pro získání modelu nebo výrobce zařízení), která jsou automaticky volána při přidávání UPS a v případě jejich nalezení je UPS automaticky pojmenována (lze ručně změnit).

Zde je vidět příklad vyčítání dat z UPS, tučně vyznačené hodnoty jsou OID pro model a výrobce.

```
var dataUps = Messenger.Get(VersionCode.V1,
new IPEndPoint(IPAddress.Parse(UPS.IPAddress), 161),
new OctetString("public"),
    new List<Variable> {new Variable(new
        ObjectIdentifier("1.3.6.1.4.1.232.165.1.2.2.0")),new
        Variable(new
            ObjectIdentifier("1.3.6.1.4.1.232.165.1.2.1.0")) },
    6000);
UPS.Model = dataUps[0].Data.ToString();
UPS.Producer = dataUps[1].Data.ToString();
```

Pro více specifická data má zpravidla každý výrobce vlastní OID, které má většinou uvedeno na svých webových stránkách. Jelikož jsou v tomto konkrétním nasazení obě UPS od společnosti HPE (Hewlett-Packard Enterprise) a jsou téměř identické, tak mají OID stejná, ale systém je připraven i na jiné UPS jiných výrobců, např. pro případ výměny.

2.2 Problém při aktualizaci firmwaru síťového modulu UPS

Před začátkem práce s UPS pomocí protokolu SNMP jsem kontroloval aktuálnost firmwaru síťových modulů, které slouží pro komunikaci s UPS po LAN. Dle webových stránek výrobce (v tomto případě HPE) byla pro jeden ze síťových modulů k dispozici

novější verze firmwaru. Stáhl jsem ji a dle dokumentace k aktualizaci firmwaru a dokumentace od dodavatele UPS jsem zkontroloval kompatibilitu aktualizace UPS, síťových modulů a firmwaru, která vyhovovala. Bez obav jsem provedl aktualizaci firmwaru daného síťového modulu.

Po aktualizaci modulu a restartu UPS jsem zjistil, že síťový modul není schopen komunikovat s UPS. Začal jsem tedy s hledáním řešení na internetu, kde ale o tomto druhu problému bylo velmi málo zmínek a také se jednalo o modely jiných generací této verze UPS. Zkoušel jsem také stáhnout původní verzi firmwaru, ale bohužel dle informace na webových stránkách HPE ji již nešlo stáhnout z důvodu ukončení podpory.

Podle dokumentace od dodavatele serverů (včetně UPS), který prováděl instalaci a fyzické kontroly UPS jsem zjistil, že novější síťový modul, tj. ten se mnou aktualizovaným firmwarem, který se dle dokumentace měl nacházet v novější UPS (Generace 5) se nachází ve starší generaci UPS (Generace 4) a starší modul se nachází v novější generaci UPS. Takže byly oba síťové moduly dodavatelem nainstalovány do UPS obráceně a nebyly s nimi kompatibilní.

Řešení tohoto problému bylo tedy poměrně jednoduché – vzájemná výměna modulů mezi sebou neboli „kus za kus“. Po provedení této výměny a postupném restartu obou UPS začala komunikace fungovat také u aktualizovaného modulu.

3 Komunikace se servery

Pro komunikaci se servery jsou zde použity dvě technologie. Tou první je rozhraní REST API, které zajišťuje komunikaci s vCenter serverem, což je jeden z virtuálních serverů, který kontroluje a monitoruje provoz virtuálních serverů na více fyzických serverech (hostech) včetně jejich případného přesunu na jiného hosta. REST API je také využito pro komunikaci s řídicími moduly iLO fyzických serverů. Druhou technologií je rozhraní PowerCLI, které je použito pro komunikaci s fyzickými (hostitelskými) servery a slouží k ovládání virtuálního diskového pole a virtuálního serveru pro vCenter. Původně jsem počítal, že i na toto použiji rozhraní REST API, ale bohužel dle oficiální dokumentace není pro hostitelské servery dostupné a jediné možné připojení je právě pomocí PowerCLI.

3.1 Komunikace s vCenter serverem

Komunikace s vCenter serverem je realizována pomocí rozhraní REST API, které směřuje na adresu <https://adresavcenter/rest>.

3.1.1 Autentifikace

Základem autentifikace aplikace k API vCenter serveru je vytvoření Session pomocí POST metody, která obsahuje v hlavičce zprávy parametr Authorization s hodnotou Basic a zašifrovanými parametry username a password pomocí Base64. Následně je vrácen **API_key**, který slouží k práci se všemi endpointy. Příklad metody pro šifrování do formátu Base64:

```
private string EncodeTo64(string toEncode)
{
    byte[] toEncodeAsBytes = System.Text.ASCIIEncoding.
    ASCII.GetBytes(toEncode);
    string returnValue =
    Convert.ToBase64String(toEncodeAsBytes);
    return returnValue;
}
```

Příklad metody pro volání POST autentifikační metody v REST API:

```
private async Task<string> GetvCenterApiKey()
{
    string url = _appSettings.Value.vCenterApiUrl +
        "/com/vmware/cis/session";
    string username = _appSettings.Value.vCenterUsername;
    string password = _appSettings.Value.vCenterPassword;
    string header = EncodeTo64(username + ":" + password);

    using (HttpClient httpClient = new HttpClient())
    {
        HttpRequestMessage request = new
            HttpRequestMessage(HttpMethod.Post, url);
        request.Headers.Add("Authorization", "Basic " + header);
        var response = await httpClient.SendAsync(request);
        string jsonResponse = await
            response.Content.ReadAsStringAsync();
        Dictionary<string, string> htmlAttributes =
            JsonConvert.DeserializeObject<Dictionary<string,
                string>>(jsonResponse);
        return htmlAttributes["value"];
    }
}
```

3.1.2 Získání seznamu virtuálních serverů

Díky REST API je možné získat seznam virtuálních serverů přímo z API, takže není nutné jejich seznam ručně přidávat, při změnách přepisovat atd. jako tomu bylo dříve ve skriptovém řešení. Při správně zadaných přihlašovacích údajích a příslušné IP adresy vCenter serveru funguje metoda automatického načtení, kterou lze zavolat z uživatelského rozhraní. Tato metoda přidá seznam všech serverů, případně aktualizuje jejich jména nebo přidá nové do databáze aplikace.

Metoda volá metodou GET příslušný endpoint (dle popisu v dokumentaci), kde do hlavičky zprávy přidá parametr **vmware-api-session-id** s hodnotou při autentifikaci získaného session API key. Je vrácena zpráva, která obsahuje seznam virtuálních serverů ve formátu JSON. Ten je následně pomocí NuGet balíčku JSON.NET od tvůrce Newtonsoft

převeden na seznam objektů a z nich jsou už vytvářeny vlastní objekty virtuálních serverů v databázi aplikace.

3.1.3 Ovládání napájení jednotlivých virtuálních serverů

K ovládání napájení virtuálních serverů slouží také endpointy REST API, které jsou volány pomocí metody GET s klasickou autorizací v hlavičce jako u ostatních metod (např. pro získání seznamu virtuálních serverů).

Příklad kódu pro spuštění virtuálního serveru:

```
string apiKey = GetvCenterApiKey().Result;
string apiUrl = _appSettings.Value.vCenterApiUrl + "/vcenter/vm/"
+ vmId + "/power/start";

using (HttpClient httpClient = new HttpClient())
{
    HttpRequestMessage request = new
    HttpRequestMessage(HttpMethod.Post, apiUrl);
    request.Headers.Add("vmware-api-session-id", apiKey);
    var response = await httpClient.SendAsync(request);
    return response.StatusCode;
}
```

3.2 Komunikace s fyzickými servery

Pro komunikaci s fyzickými servery jsou využity dva různé typy komunikace. První je síťová komunikace s řídícím a servisním modulem iLO, který je vložen ve fyzických serverech a slouží především pro jejich spuštění z vypnutého stavu a získání jejich aktuálního stavu. Druhým typem komunikace je připojení k operačnímu systému vSphere ESXi hostitelských serverů (host serverů), kde se spouštějí a ukončují virtuální storage servery a server vCenter. Rovněž se zde kontroluje připojení datastorů (úložišť virtuálních serverů) při spouštění systému.

3.2.1 Komunikace s iLO

Tato komunikace je realizována pomocí REST API POST a GET metod, kterými jsou volány endpointy REST API iLO fyzických serverů umístěných na adrese <https://ilo->

[ip/redfish/v1](#). Při zahájení komunikace je stejně jako v případě připojení k vCenter serveru nutné získat API token, kterým se aplikace bude nadále identifikovat při volání endpointů API, v tomto případě pro získání stavu napájení a spuštění serveru.

3.2.2 Komunikace se systémem VMWare vSphere ESXi

Ke komunikaci se systémem VMWare vSphere ESXi, který je využíváný pro hostování virtuálních serverů na fyzických serverech, je nutné využití příkazů PowerCLI. Proto jsem využil Nuget balíčku PowerShell.SDK od společnosti Microsoft, díky kterému je používání těchto příkazů v jazyce C# poměrně přívětivé. Pro komunikaci s vSphere ESXi bylo nutné doinstalovat na řídicí mini PC PowerShell balík VMware.PowerCLI.

Zde je příklad volání kódu pro připojení k hostiteli.

```
Runspace runspace = RunspaceFactory.CreateRunspace();
runspace.Open();

Pipeline pipeline = runspace.CreatePipeline();
pipeline.Commands.AddScript("Set-ExecutionPolicy Unrestricted");
pipeline.Commands.AddScript("Import-Module
VMware.VimAutomation.Core")
pipeline.Commands.AddScript("Connect-VIServer -Server adresa
serveru -Protocol https -User uživatelské jméno -Password heslo -
ErrorAction Stop");
pipeline.Invoke().First();
runspace.Close();
```

4 Služby

Veškeré metody v aplikaci jsou rozděleny do služeb, kterých se v aplikaci nachází celkem šest:

- pro práci s databází
- pro posílání e-mailových notifikací
- pro definici scénářů spouštění a vypínání
- pro práci s fyzickými servery a virtuálními hosty na nich – PowerCLI
- pro práci s virtuálními servery pomocí serveru vCenter – REST API
- pro práci s UPS zařízeními

Díky tomu lze jakoukoliv metodu volat odkudkoliv z aplikace (i z jiné služby). Každá služba implementuje metody jejího interface, ze kterého pochází (takže i dědí metody). A díky tomu je možné ji připojit kdekoliv pomocí konstruktoru.

4.1 Služba pro práci s databází (DbService)

Tato služba je určena pro většinu práce s `ApplicationDbContext`. Slouží pro získávání, vkládání, editaci nebo mazání dat z databáze.

4.2 Služba pro e-mailové notifikace (MailService)

V této službě je jediná metoda SendMail, která přijímá parametry předmět e-mailu (subject) a body (tělo – text e-mailu). Slouží pro posílání stavových a notifikačních zpráv na e-mail administrátora. Data pro připojení k SMTP serveru a e-mailovou adresu načítá z globálního nastavení aplikace.

Odesílání je implementováno pomocí NuGet balíčku MailKit, ukázka je níže:

```
AppSettings settings = _dbService.GetSettingsWithoutInclude();
var message = new MimeMessage();
message.From.Add(new MailboxAddress("MMNVS - " + settings.Name,
settings.SmtpUser));
message.To.Add(new MailboxAddress("MMNVS - administrator",
settings.AdministratorE-mail));
message.Subject = subject;

message.Body = new TextPart("plain")
{
    Text = body + @"
-- Tato zpráva byla automaticky vygenerována systémem MMNVS"
};

using (var client = new SmtpClient())
{
    client.Connect(settings.SmtpServer, settings.SmtpPort ?? 25,
settings.SmtpIsSecure);
    client.Authenticate(settings.SmtpUser, settings.SmtpPassword);
    client.Send(message);
    client.Disconnect(true);
}
```

4.3 Služba pro definici scénářů (ScenarioService)

Zde jsou implementovány dvě nejdůležitější metody celého systému, a to metody pro start a vypnutí virtualizačního systému, které v případě jejich zavolání volají další metody ze všech ostatních služeb.

4.4 Služba pro práci s UPS (UPSService)

Účelem této služby je získávání a práce s daty z UPS zařízení napájejících hostitelské servery.

Nacházejí se zde metody pro získání aktuálního stavu UPS, uložení aktuálního stavu do logu UPS (voláno automaticky) nebo metody pro získání logické informace o tom, zda je UPS napájena z elektrické sítě nebo běží na baterii (často využíváno pro kontrolu stavu ve službě scénářů při startu a vypínání).

4.5 Služba pro práci s virtuálními servery (VMService)

Tato služba je určena pro komunikaci s vCenter serverem pomocí REST API. Připojení a komunikace je blíže popsána v kapitole věnované komunikaci s vCenter serverem.

Mezi metody této služby patří metoda pro získání virtuálních serverů ze serveru vCenter, spuštění a vypnutí či získání stavu virtuálního serveru nebo serveru vCenter.

4.6 Služba pro práci s fyzickými servery (ServerService)

Účelem této služby je komunikace s operačním systémem hostitelských serverů vSphere ESXi, kde není možné připojení pomocí REST API, takže je využito technologie PowerCLI. Podrobněji je komunikace popsána v kapitole 3.2.2 věnované komunikaci s hostitelskými servery – komunikace se systémem VMWare vSphere.

Nachází se zde metody pro spouštění a vypínání virtuálních storage serverů, vCenter serveru a hostů, kontrolu dostupnosti datastorů, získání aktuálního stavu serverů. Je zde i metoda pro určení na kterém hostitelském serveru se nachází server vCenter.

Také jsou zde metody pro práci s moduly iLO, které jsou vloženy ve fyzických serverech. V tomto případě probíhá komunikace pomocí REST API, jak je popsáno v kapitole 3.2.1 věnované komunikaci s fyzickými servery – komunikace s iLO.

5 Logování dat

Jako v každém systému je nutné ukládat data o stavu systému v různých krocích, ať už pro účely debuggování (při vývoji), tak při ostrého provozu. Z důvodu rychlého odhalení chyb, které mohly být způsobeny úplně čímkoliv (změna systému, na který je aplikace napojena, aktualizace systému nebo chyba v nové verzi aplikace) nebo jen pro dohledání toho, co se kdy a v jaký čas dělo.

Z tohoto důvodu jsem v databázi této aplikace udělal dvě tabulky právě pro data z logů, které reprezentují data pomocí enumerátorů, celých čísel a ID jiných entit. Ke každému záznamu je samozřejmě přidán údaj o aktuálním datu a času. Data z logů je pro jednodušší přístup možné zobrazit přímo ve stránkách v uživatelském rozhraní, kde jsou seřazeny od nejnovějších a pro jejich rychlejší načítání jsou data rozdělena do několika stránek (stránkována) s možností volby počtu záznamů na stránku.

5.1 Logování stavů UPS

Na toto místo jsou ukládány stavová data z UPS vždy při každém automatickém zavolání vnitřního API. Mezi zde ukládaná data patří mimo aktuální datum a čas: ID (jedinečný identifikátor) UPS, stav (připojeno k síti, provoz na baterii), stav nabití baterie (v procentech), zbývajících čas běhu na baterii (v sekundách) a vytížení UPS (v procentech)

UPS Log

UPS	Čas a datum	Stav	Kapacita baterie	Zbývajících čas	Zátěž	Error
Test UPS 2 (4)	13.03.2022 19:01:36	Napájení ze sítě	100%	1369s	24%	<input type="checkbox"/>
UPS 2 (3)	13.03.2022 19:01:36	Napájení ze sítě	100%	1369s	24%	<input type="checkbox"/>
UPS 1 (2)	13.03.2022 19:01:36	Napájení ze sítě	100%	1931s	17%	<input type="checkbox"/>
Test UPS 2 (4)	13.03.2022 19:01:06	Napájení ze sítě	100%	1369s	24%	<input type="checkbox"/>
UPS 2 (3)	13.03.2022 19:01:06	Napájení ze sítě	100%	1369s	24%	<input type="checkbox"/>
UPS 1 (2)	13.03.2022 19:01:06	Napájení ze sítě	100%	1931s	17%	<input type="checkbox"/>
Test UPS 2 (4)	13.03.2022 19:00:48	Napájení ze sítě	100%	1369s	24%	<input type="checkbox"/>
UPS 2 (3)	13.03.2022 19:00:48	Napájení ze sítě	100%	1369s	24%	<input type="checkbox"/>
UPS 1 (2)	13.03.2022 19:00:47	Napájení ze sítě	100%	2016s	17%	<input type="checkbox"/>
Test UPS 2 (4)	13.03.2022 18:50:53	Napájení ze sítě	100%	1369s	24%	<input type="checkbox"/>

1 2 3 4 5 6 7 8 9 10 11

10



Změnit počet itemů na stránku

Obrázek 5 Stránka pro zobrazení dat z logu UPS

5.2 Logování stavů serverů a celého systému

Do tohoto logu se zapisují především data ze služby scénářů, například vypínání a spouštění serverů (virtuálních i fyzických) a to vždy před začátkem operace a po dokončení (kontrola stavu). Také se zde zapisuje informace o změně stavu systému, který je uložen v tabulce nastavení.

6 Nasazení aplikace

Tato webová aplikace je hostována na malém osobním počítači Intel NUC s operačním systémem Windows 10 a procesorem Intel Pentium Silver J5005, který je nezávislý na zdroji napájení serverů, které řídí (má vlastní UPS). Jako webový server je využita Internetová informační služba IIS, což je jedna ze systémových funkcí systému Windows. Databázový server je realizován pomocí Microsoft SQL Express a jeho správa je zajišťována pomocí Microsoft SQL Management studio. S monitorovaným systémem virtuálních serverů komunikuje přes LAN.

6.1 Konfigurace před nasazením aplikace

Před samotným nasazením jsem musel mezi funkcemi systému Windows aktivovat právě Internetovou informační službu (IIS), nainstalovat modul Runtime .net core, knihovnu .net core a modul ASP.NET Core, který umožňuje běh aplikací ASP.NET Core na IIS. Dále bylo třeba nainstalovat databázový server Microsoft SQL Express 2019, modul PowerCLI pro práci s prostředím VMWare. Pro správné fungování s vCenter serverem jsem také musel nainstalovat certifikát kořenové certifikační autority, který je využíván pro šifrovanou (https) komunikaci na vCenter serveru.

6.2 Vlastní nasazení aplikace

Pomocí konfiguračního profilu pro nasazení v sadě Microsoft Visual Studio 2022 jsem sestavil aplikaci přímo do složky na počítači Intel NUC (pomocí sdílení složek Windows), kterou jsem nadefinoval do webové služby na IIS. Následně jsem v databázi založil uživatele pro práci aplikace s databází. Pomocí upraveného Connection stringu k nové databázi jsem funkcí Update-Database z Entity Framework Core aplikoval veškeré migrace do nově vytvořené databáze, tedy vytvořil všechny tabulky s jednotlivými sloupci. Nasazení jsem dokončil úpravou souboru AppSettings.json.production, kde jsem upravil Connection string k databázi a nastavením aplikace: přidáním UPS, serverů, nastavením údajů k vCenter, SMTP serveru k odesílání e-mailů, časů pro čekání mezi kontrolou stavu a minimální vydrží baterií pro spouštění nebo vypnutí.

6.3 Ladění a testování aplikace

Jelikož jsem aplikaci programoval pro již existující, a především uživateli používané řešení, mohl jsem v produkčním prostředí okamžitě testovat pouze metody pro vyčítání dat a metody pro ukončování a spouštění virtuálních serverů přes server vCenter. Na to jsem využil testovací virtuální servery sloužící pro administrátory. Metody pro ukončování a spouštění fyzických a storage serverů jsem mohl částečně testovat na starším serveru, který ale nemá licenci pro virtuální storage servery, tím pádem ani pro virtuální diskové pole. Z těchto důvodů jsem zde mohl testovat pouze metody pro práci s HPE iLO a operačním systémem VMware vSphere.

Z výše uvedených důvodů jsem test funkčnosti celé aplikace musel provádět o víkendu, kdy výpadek serverů nikoho neomezí. Nejdříve jsem přes uživatelské rozhraní kompletně nastavil veškeré potřebné údaje o serverech, poté jen stačilo odpojit UPS od elektrické sítě a vyčkat, až se baterie vybijí na mezní úroveň pro začátek vypínání. Postupně se vyskytly nějaké problémy, které jsem ale vyřešil úpravou pár řádků kódu. Podobné to bylo i po připojení UPS k elektrické síti, kdy se servery spouštěly. Tento proces jsem zopakoval celkem čtyřikrát, z toho u posledních dvou testů se již nevyskytla žádná chyba. Testování jsem zakončil ověřením, zda je systém schopný reagovat na změny stavu napájení UPS v již běžícím procesu vypínání/spouštění, které se ukázalo jako funkční.

Během testování jsem využil informace ze stránky Logu, díky kterému se mi zjednodušila práce při ladění a testování aplikace.

Závěr

Podařilo se mi vytvořit metody na práci se všemi součástmi řešení (UPS, fyzické i virtuální servery), díky tomu mohu vše komplexně řídit z jednoho místa – této aplikace. Mým původním plánem bylo pracovat se všemi servery pomocí REST API, tedy volání (především) GET a POST metod na webové adresy URL. Dle úvodního průzkumu se toto řešení zdálo jako proveditelné, ale bohužel, jak jsem později zjistil, REST API podporuje pouze řídící virtuální VMWare vCenter server, nikoli fyzické ESXi host servery se systémem pro virtualizaci VMWare vSphere ESXi. U těchto serverů je jejich jediná možnost ovládání pomocí PowerCLI příkazů, takže jsem ji samozřejmě využil a pracuji se servery pomocí těchto příkazů v NuGet balíku PowerShell. Jako poslední součást komunikace se servery jsem implementoval připojení k HPE iLO rozhraní pro řízení fyzických serverů (především kvůli spuštění serverů z vypnutého stavu). Toto rozhraní jsem původně chtěl implementovat pomocí příslušné PowerCLI knihovny od HPE. Později jsem ale zjistil, že komunikace tímto způsobem je velmi nespolehlivá, protože balík HPEiLOCmdlets je poměrně zastaralý a nefunkční. Nakonec se mi podařilo najít možnost implementace pomocí REST API, která funguje mnohem lépe a spolehlivěji.

Ačkoli jsem si práci předem rozvrhl a celkem jasně jsem věděl, jak má celá aplikace fungovat, co má kontrolovat a co v jakých případech provádět, také jsem dle těchto informací navrhl model databáze. Stejně jsem musel postupně při vytváření aplikace databázi i fungování metod upravovat, dokonce jsem do databáze přidal jednu novou tabulku a jednu nepotřebnou zrušil. Celkově se ale nejednalo o žádné markantní změny, takže si myslím, že příprava věnovaná návržení databáze a promýšlení fungování aplikace ještě před začátkem programování se vyplatila a ušetřila čas při programování a ladění aplikace.

Celkově jsem se u projektu snažil o co největší nezávislost na vybraném hardwaru, například přidávání OID k UPS v nastavení aplikace (mohou se u různých modelů lišit), změnu používané verze serveru vCenter nebo možnost využití více virtuálních storage serverů, ačkoliv se v aktuálním nasazení využívá pouze jeden na fyzický (host) server. Díky tomu je aplikace připravena na budoucí výměnu serverů, UPS nebo aktualizaci virtualizačního softwaru VMWare vSphere a vCenter. Také jsem se snažil o co největší přívětivost uživatelského rozhraní pro administrátora a jeho okamžité informování při konfiguraci aplikace o úspěšnosti (případně neúspěšnosti) připojení

k přidávanému (upravovanému) zařízení. Díky tomu má administrátor ověřenu správnost jeho nastavení.

Prostor pro další rozvoj aplikace vidím v přidání dalších druhů zařízení, která bude aplikace řídit v závislosti na stavu napájení UPS, možnosti práce s více systémy pro virtualizaci nebo ještě dalšími verzemi VMWare (pokud dojde ke změnám v ovládání). Nyní je v aplikaci realizováno vše, co je potřeba pro aktuálním nasazení.

Pokud bych si tuto práci měl vybrat znovu, učinil bych tak. Ačkoliv jsem na ní strávil velké množství času (okolo 300 hodin), dosáhl jsem svého cíle a vytvořil jsem to, co jsem očekával – funkční řešení pro automatizovanou správu a kontrolu napájení serverů. Získal jsem mnoho nových zkušeností, znalostí a naučil jsem se nové věci v ASP.NET Core. Porozuměl jsem syntaxi a využívání PowerCLI skriptů, práci s REST API ve frameworku ASP.NET Core jazyka c#. Naučil jsem se získávat informace o zařízeních pomocí protokolu SNMP a pochopil jsem význam unikátních OID sloužících pro získání mnou vyžadovaných údajů.

Seznam zkratk a odborných výrazů

API

Application Programming Interface – rozhraní pro práci se systémem

API endpoint

„Koncový bod API“, URL adresa, na kterou jsou odesílány žádosti typu GET (získání dat), POST (odeslání dat), PUT (úprava dat) dle dokumentace k API

CLI

Command Line Interface – rozhraní, kde uživatel komunikuje s programy zapisováním příkazů do příkazového řádku

Connection string

Textový řetězec, kde jsou uloženy informace k připojení k databázi (adresa databázového serveru, uživatelské jméno a heslo)

ESXi

Elastic Sky X integrated – operační systém hostitelského serveru v prostředí VMWare vSphere

HPE

Hewlett-Packard Enterprise – výrobce profesionálních zařízení

IIS

Internet information service – Internetová informační služba, webový server

iLO

Integrated Lights-Out – je patentovaná technologie pro správu serverů společnosti Hewlett-Packard Enterprise (HPE)

LAN

Local Area Network – místní počítačová síť

NAS

Network Attached Storage – síťové uložště dat

OID

Object identifier – jednoznačný identifikátor pro požadovaný údaj v protokolu SNMP

PC

Personal Computer – osobní počítač

PowerCLI

řešení automatizace úloh pro více platforem, které se volá pomocí příkazů skriptovacího jazyka

REST

Representational State Transfer – architektura pro webové API

SAN

Storage Area Network – dedikovaná datová síť

SMTP

Simple Mail Transfer Protocol – internetový protokol určený pro odesílání internetových zpráv

SNMP

Simple Network Management Protocol – protokol pro získání informací o stavu zařízení

UPS

Uninterruptible Power Supply – záložní baterie, která zajišťuje napájení (především počítačů nebo serverů) i při výpadku elektřiny

vCenter server

jedná se o virtuální server sloužící k řízení několika virtuálních serverů na několika hostitelských serverech zároveň

VSA

Virtual storage appliance – řadič úložiště, který je spuštěn na virtuálním serveru

Seznam obrázků

Obrázek 1 Úvodní stránka aplikace – zobrazení stavových dat.....	4
Obrázek 2 Seznam virtuálních serverů a informace o jejich stavu	6
Obrázek 3 Formulář pro přidání VSA serveru	7
Obrázek 4 Zobrazení stavu fyzického serveru	7
Obrázek 5 Stránka pro zobrazení dat z logu UPS.....	18

Použité zdroje

1. **Wikipedia.** Identifikátor objektu. *Wikipedia*. [Online] 12. leden 2022.
https://cs.wikipedia.org/wiki/Identifikátor_objektu.
2. **VMWare.** *VMWare Developer Documentation*. [Online] 12. leden 2022.
<https://developer.vmware.com/apis>.
3. **Wikipedia.** Zdroj nepřerušovaného napájení. *Wikipedia*. [Online] 12. leden 2022.
https://cs.wikipedia.org/wiki/Zdroj_nepřerušovaného_napájení%C3%AD.
4. **Encyklopedia.** VMware ESXi. *Encyklopedia*. [Online] 13. březen 2022.
https://wikijii.com/wiki/vmware_esxi.
5. **Wikipedia.** Simple Mail Transfer Protocol. *Wikipedia*. [Online] 2022. únor 2022.
https://cs.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol.
6. **TechTarget.** virtual storage appliance (VSA). *TechTarget*. [Online] 12. leden 2022.
<https://searchservvirtualization.techtarget.com/definition/Virtual-storage-appliance-VSA>.
7. **Wikipedia.** Storage Area Network. *Wikipedia*. [Online] 12. leden 2022.
https://cs.wikipedia.org/wiki/Storage_Area_Network.
8. **TechLib.** SNMP. *TechLib*. [Online] 12. leden 2022. <https://tech-lib.eu/definition/snmp.html>.
9. **Wikipedia.** Network Attached Storage. *Wikipedia*. [Online] 12. leden 2022.
https://cs.wikipedia.org/wiki/Network_Attached_Storage.
10. **Parse-error.cz.** Co je to REST API. *Parse error.cz*. [Online] 12. leden 2022.
<https://www.parse-error.cz/nodejs-tutorial/4/co-je-to-rest-api>.
11. **Microsoft.** Co je PowerShell? *Microsoft Docs*. [Online] 8. leden 2022.
<https://docs.microsoft.com/cs-cz/powershell/scripting/overview?view=powershell-7.2>.
12. **Wikipedia.** API. *Wikipedia*. [Online] 12. leden 2022.
<https://cs.wikipedia.org/wiki/API>.
13. **Wikipedia.** HP Integrated Lights-Out. *Wikipedia*. [Online] 13. březen 2022.
https://en.wikipedia.org/wiki/HP_Integrated_Lights-Out.

14. **Wikipedia**. Příkazový řádek. *Wikipedia*. [Online] 20. únor 2022.

https://cs.wikipedia.org/wiki/Př%C3%ADkazový_řádek.

15. **Wikipedia**. Local Area Network. *Wikipedia*. [Online] 14. březen 2022.

https://cs.wikipedia.org/wiki/Local_Area_Network.

A. Seznam příložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

- **MP2021-22_Antos-Adam_Aplikace-pro-spravu-a-monitoring-napajeni-virtualnich-serveru.docx** – editovatelná verze dokumentace maturitní práce
- **MP2021-22_Antos-Adam_Aplikace-pro-spravu-a-monitoring-napajeni-virtualnich-serveru.pdf** – tisknutelná verze dokumentace maturitní práce
- **Aplikace** – zdrojové kódy