PD.TVZ.HR
POLYTECHNIC & DESIGN Vol. 13, No. 4, 2025.
Mrežna inačica: ISSN 2459-6302
Tiskana inačica: ISSN 1849-1995
DOI: 10.19279/TVZ.PD.2025-13-4-06

AUTOMATIZACIJA DOSTUPNOSTI VIRTUALNOG OKRUŽENJA POMOĆU PowerCLI

AUTOMATING THE AVAILABILITY OF A VIRTUAL ENVIRONMENT USING PowerCLI

Ivan Mihaljević[1], Ognjen Mitrović[2], Leon Pongrac[3]

[1] Tehničko veleučilište u Zagrebu, Vrbik 8, 10000 Zagreb, Hrvatska, Student
[2] Tehničko veleučilište u Zagrebu, Vrbik 8, 10000 Zagreb, Hrvatska
[3] Ericsson Nikola Tesla d.d., Krapinska 45, 10000 Zagreb, Hrvatska

SAŽETAK

Razvojem virtualizacijskih tehnologija raste kompleksnost IT sustava, dok korisnici istovremeno zahtijevaju brži pristup resursima, napredne integracije i jednostavnost korištenja. Jedno od ključnih rješenja za ove izazove primjena je skriptnih jezika kojom se postiže automatizacija procesa između različitih hardverskih i softverskih komponenti. Takav pristup predstavlja učinkovito i ekonomično rješenje za transformaciju tradicionalnih podatkovnih centara u proaktivan i dinamičan model rada. U ovom radu analiziran je administratorski alat VMware PowerCLI, temeljen na skriptnom jeziku PowerShell, koji omogućuje sveobuhvatno upravljanje virtualizacijskom platformom. Središnji dio rada prikazuje autorsko rješenje za automatizaciju procesa uključivanja i gašenja virtualnih strojeva. Predloženi model integrira sustav za upravljanje korisničkim zahtjevima Jira s platformom VMware vSphere, čime se postiže besprijekoran tijek rada od korisničkog zahtjeva do izvršenja operacije na infrastrukturi.

Ključne riječi: skriptiranje, automatizacija, PowerShell, PowerCLI, VMware, Jira

ABSTRACT

The emergence of virtualization has significantly increased the complexity of IT systems, while users simultaneously demand faster, more flexible access, higher levels of application integration, and intuitive ease of use. To address these administrative challenges, scripting serves as a key solution for achieving automation across diverse hardware and software components within an ecosystem. Such an approach provides an efficient and cost-effective strategy for transforming traditional, console-managed data centers into proactive and dynamic operational models. This paper analyzes VMware PowerCLI, an administrative automation tool built on the PowerShell scripting language, designed for the comprehensive management of virtualization platforms based on the VMware hypervisor. The core of the paper presents a custom-developed automation solution for the power management of virtual machines. The proposed model integrates the Jira service management platform with the VMware vSphere infrastructure, ensuring a seamless workflow from the initial user request to the final execution of operations within the virtual environment.

Keywords: scripting, automation, PowerShell, PowerCLI, VMware, Jira

1. UVOD
1. INTRODUCTION

Pojavom virtualizacije drastično se povećala kompleksnost IT sustava [1], dok korisnici istovremeno očekuju brži i fleksibilniji pristup resursima, veću integraciju aplikacija te jednostavnost korištenja. Tradicionalni, podatkovni centri kojima se upravlja putem konzole, u kojima se zadaci obavljaju ručno, postaju usko grlo u modernom poslovanju. Kako bi se postigao proaktivan i dinamičan model rada, neophodno je uvođenje automatizacije koja povezuje različite hardverske i softverske komponente sustava [2]. Cilj je automatiziranih podatkovnih centara dinamička izrada virtualnih resursa za krajnje korisnike u što jednostavnijem obliku, pazeći pritom da se zadovolje sigurnosne i poslovne politike definirane uslugom [3].

U suvremenim poslovnim okruženjima, posebno većim, suradnja s vanjskim partnerima postala je standard zbog ekonomske isplativosti i potrebe za specifičnim vještinama koje poduzeće nema unutar svoje strukture. Ti suradnici obično pristupaju mreži putem virtualnih privatnih mreža (VPN) i koriste udaljenu radnu površinu (RDP) na virtualnim strojevima (VM) [4]. Radi optimizacije infrastrukturnih resursa, ti se strojevi ne drže stalno uključenima, nego se aktiviraju isključivo prema potrebi.

U konkretnom primjeru poduzeća obrađenog u ovom radu proces je donedavno bio oslonjen na ljudski faktor i sustav Jira za upravljanje zahtjevima. Neautomatizirani tijek rada sastojao se od izrade korisničkog zahtjeva putem sustava Jira s točnim vremenom uključivanja i gašenja te imenom VM-a. Nakon toga djelatnik korisničke podrške morao bi ručno u sučelju vSphere izvršiti te radnje u zadano vrijeme. Ovakav pristup pokazao se neučinkovitim i podložnim greškama te bi često zahtjev bio zaprimljen, ali VM nije bio uključen ili ugašen na vrijeme zbog preopterećenosti osoblja.

Uvođenje automatizacije u ovakve procese nije samo tehničko unaprjeđenje, već nužnost za održavanje kontinuiteta poslovanja. Prema istraživanjima, automatizacija rutinskih zadataka smanjuje operativne troškove i omogućuje administratorima fokus na kompleksnije arhitektonske izazove, čime se izravno povećava ukupna stabilnost sustava [5].

Cilj je ovog rada prikazati rješenje koje eliminira ljudsku pogrešku korištenjem VMware PowerCLI modula temeljenih na PowerShell jeziku. Fokus će biti na implementaciji skripte koja automatski, neposredno po zadavanju zahtjeva u Jiri, kreira zakazane zadatke (eng. *scheduled tasks*) unutar VMware okruženja. Time se osigurava maksimalna dostupnost resursa uz minimalan utrošak vremena i eliminaciju ručnog rada administratora.

2. VIRTUALIZACIJA I VMWARE
2. VIRTUALIZATION AND VMWARE

Virtualizacija predstavlja tehnologiju koja omogućuje izvršavanje više operacijskih sustava i aplikacija na jednom fizičkom poslužitelju [6], čime se drastično povećava iskoristivost hardvera koja je kod tradicionalnih servera uglavnom bila između 20-50% dok je u nekim slučajevima ta iskoristivnost znala biti ispod 10% [7]. U središtu ovog procesa nalazi se virtualni stroj (VM), koji koristi resurse fizičkog računala (hosta) putem sloja softvera nazvanog hipervizor. Zadaća hipervizora, osim izolacije virtualnih strojeva, je i briga o resursima koji su povezani i definirani za svaki stroj [7].

Kao jedan od globalnih predvodnika u području računalstva u oblaku i virtualizacijskih tehnologija tvrtka VMware (koja od studenog 2023. godine posluje kao dio korporacije Broadcom) nudi širok spektar rješenja za serversku infrastrukturu, upravljanje cloud sustavima te mrežnu sigurnost [8]. Pod okriljem Broadcomovog odjela za izradu modernih virtualnih podatkovnih centara VMware Cloud Foundation ključna je platforma VMware vSphere, čiju arhitekturu čine dvije primarne komponente – ESXi hipervizor i vCenter Server.

Osnovna komponenta virtualizacije je ESXi i on predstavlja sam hipervizor koji se instalira na fizičko računalo [9]. ESXi ima osnovno upravljačko (CLI) sučelje za inicijalno konfiguriranje hipervizora, dok za izradu i upravljanje radom VM-ova koristi se VMware Host Client, sučelje temeljeno na standardu HTML5.

Dok ESXi upravlja radom na razini jednog čvora, vCenter Server služi kao središnja točka za upravljanje svim hostovima i virtualnim strojevima unutar podatkovnog centra. On objedinjuje resurse u zajednički bazen, čime se postiže jednostavnost administrativnog nadzora putem središnje konzole [10]. Za udaljeno povezivanje s vCenterom koristi se vSphere Client, grafičko sučelje također temeljeno na standardu HTML5, koji omogućuje pristup svim funkcijama vSphere platforme iz bilo kojeg preglednika, neovisno o operacijskom sustavu korisnika [11].

Instalacijom vCentera otključava se niz naprednih funkcionalnosti koje virtualizacijsku platformu mogu pretvoriti u moderni softverski definirani podatkovni centar. Neke od ključnih mogućnosti su [10]:

- Upravljanje inventarom i ulogama: Omogućuje detaljno pretraživanje resursa (hostova, VM-ova, mreža i skladišta) te precizno dodjeljivanje uloga i ovlasti korisnicima.
- Kontinuitet poslovanja i mobilnost: Izvršavanje migracija uživo bez zastoja (vMotion), što omogućuje održavanje hardvera bez prekida rada usluga na VM-ovima.
- Optimizacija i skalabilnost: Automatizacija svakodnevnih zadataka i mogućnost spremanja konfiguracije hosta (Host Profiles) radi lakšeg dodavanja novih poslužitelja u sustav.
- Visoka dostupnost i udaljeni pristup: Pruža podršku za udaljene lokacije, čime se eliminira potreba za IT osobljem na svakoj fizičkoj lokaciji gdje se nalaze VM-ovi.
- Sigurnost (SSO): Značajka jedinstvene prijave (eng. *single sign-on*) omogućuje administratorima pristup svim instancama vCenter Servera bez potrebe za višestrukom autentikacijom.

Konačno, vSphere pruža dodatnu fleksibilnost i podršku za integracije te je moguće izgraditi okruženje koje podržava i okvire otvorenog koda (eng. *open source framework*), primjerice OpenStack. VMware opisuje vSphere kao srce modernog softverski definiranog podatkovnog centra koje omogućuje pokretanje, upravljanje, povezivanje i zaštitu aplikacija u zajedničkom okruženju [12]. Jedno od takvih proširenja ključno za programsku automatizaciju infrastrukture je i modul PowerCLI.

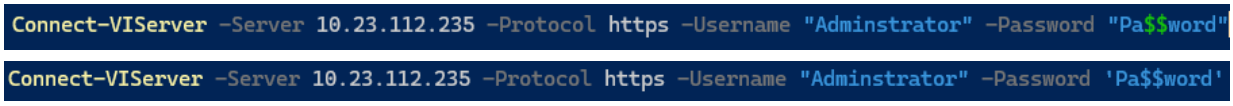
3. POWERSHELL I POWERCLI
3. POWERSHELL AND POWERCLI

Kako bi se ostvarila automatizacija unutar okruženja vSphere, potrebno je koristiti alat koji može programski komunicirati s vCenter API-jem. Primarna zadaća takvog alata isporuka je strukturiranog niza naredbi koje vCenter Server interpretira i potom prosljeđuje na izvršavanje unutar vSphere okruženja, čime se eliminira potreba za ručnim unosom svake pojedinačne operacije putem grafičkog sučelja. Kao odgovor na ovaj izazov koristi se PowerShell, odnosno njegov specijalizirani set modula pod nazivom VMware PowerCLI.

PowerShell je moderni, višeplatformski (eng. *cross-platform*), okvir za automatizaciju zadataka i upravljanje konfiguracijom koji se sastoji od ljuske naredbenog retka (eng. *command-line shell*) i skriptnog jezika izgrađenog na .NET okviru [13]. Za razliku od tradicionalnih ljuski koje procesiraju isključivo tekstualne podatke, PowerShell je objektno orijentiran. To znači da svaka naredba (cmdlet) ne vraća samo tekstualni ispis, već punopravni objekt s pripadajućim svojstvima i metodama. To administratorima omogućuje manipulaciju podacima na vrlo visokoj razini bez gubitka informacija pri prijenosu kroz niz naredbi ulančavanjem (eng. *pipeline*) ili tijekom izvođenja unutar skripti.

Uz osnovne mogućnosti skriptiranja, snaga PowerShella leži u njegovoj proširivosti putem funkcija, klasa i modula, što omogućuje razvoj kompleksnih razvojnih okvira i specijaliziranih alata. Upravo ta arhitektonska podrška, u kombinaciji s izvornom obradom formata podataka poput CSV-a, JSON-a i XML-a te mogućnošću definiranja dinamičkih tipova, stvara idealan temelj za izgradnju modula VMware PowerCLI [14].

PowerCLI je, u svojoj biti, moćna implementacija tih funkcionalnosti koja prevodi objektni model PowerShella u naredbe razumljive okruženju vSphere. On sadrži više



Slika 1 *Prikaz korištenja specijalnih znakova u PowerShellu*

Figure 1 *Usage of special characters in PowerShell*

od 800 cmdleta za adminstraciju različitih VMware modula, od vSphere-a i vSAN-a do mrežnih tehnologija i Cloud servisa. Djelujući kao posrednik između administratora i VMware API-ja, PowerCLI prati modularni model rada. Primjerice, inačica 12.0.0 nudi 23 različita modula, pri čemu su neki, poput VMware.VimAutomation.Common i VMware.VimAutomation.Sdk, neophodni za osnovnu funkcionalnost i razvoj naprednih skripti.

Specifičnost ovog pristupa očituje se i u "Object-by-Name" (OBN) metodi odabira koja omogućuje dohvaćanje objekata iz inventara prosljeđivanjem naziva uz korištenje zamjenskih znakova (wildcards). Važno je naglasiti da sustav posjeduje ugrađenu toleranciju na greške – ako se naredbi preda ime nepostojećeg objekta, generirat će se OBN greška, ali se izvršavanje daljnjih instrukcija nad ispravnim objektima neće prekinuti, što je ključno za pouzdanost automatiziranih procesa.

Ključ razumijevanja rada PowerCLIja leži u načinu na koji on pristupa vCenteru. Svaka interakcija započinje uspostavom veze putem naredbe Connect-VIServer koja kreira objekt sesije s pohranjenim vjerodajnicama. Pri radu s vjerodajnicama, neophodno je koristiti jednostruke navodnike (') kako bi se izbjegli problemi s interpretacijom posebnih znakova unutar PowerShell ljuske što je prikazano na Slici 1.

Interakcija modula PowerCLI s okruženjem vSphere temelji se na objektnom modelu upravljanja (eng. *management object model*). Riječ je o visoko strukturiranom sustavu podataka dizajniranom za upravljanje, praćenje i kontrolu cjelokupnog životnog ciklusa komponenti virtualne infrastrukture. Unutar ovog modela, strukture podataka primarno se dijele na tipove upravljanih objekata (eng. *managed object types*) i tipove podatkovnih objekata (eng. *data object types*). Detaljan uvid u hijerarhiju svih dostupnih objekata i pripadajućih API referenci pruža Broadcomova dokumentacijska platforma na službenom portalu za razvojne inženjere [15].

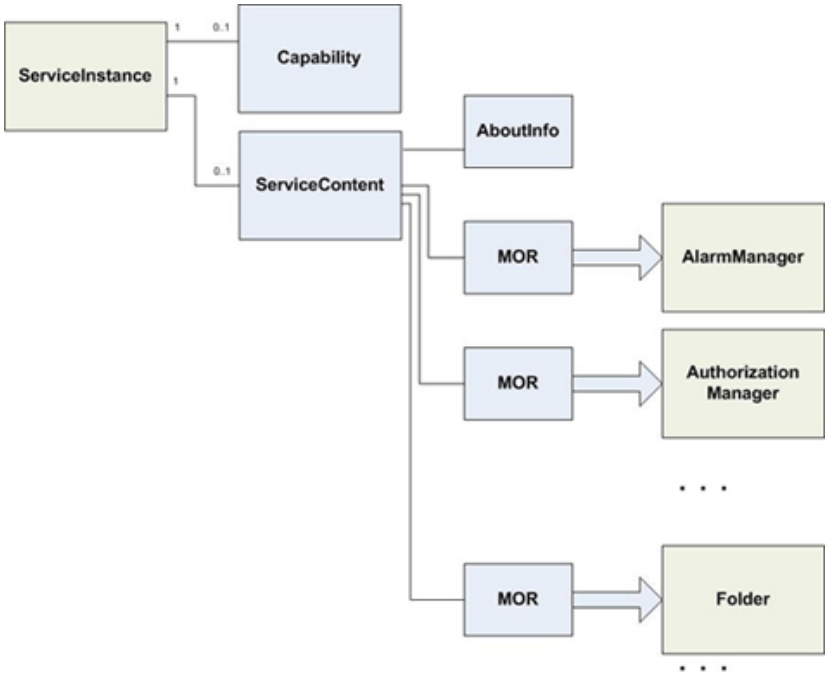
Tip upravljanog objekta predstavlja temeljnu strukturu podataka objektnog modela na strani poslužitelja. Konkretne instance ovih struktura nazivaju se upravljani objekti (eng. *managed objects*) te se primarno dijele u dvije kategorije [16]:

- Upravljani entiteti (eng. *managed entities*): Ovoj kategoriji pripadaju objekti koji nasljeđuju svojstva osnovne klase ManagedEntity. Oni predstavljaju stvarne komponente virtualne infrastrukture, poput fizičkih poslužitelja (host sustava), virtualnih strojeva, klastera ili podatkovnih centara.
- Servisni objekti (eng. *service objects*): Ovi objekti djeluju kao sučelja za upravljanje globalnim uslugama cijelog sustava. Ne predstavljaju fizičke resurse, već omogućuju izvršavanje specifičnih administrativnih funkcija poput upravljanja licencama, nadzora performansi, upravljanja virtualnom pohranom ili mrežnim postavkama.

Svi upravljani objekti definirani su kroz svoja svojstva (podatke) i operacije. Iako se u literaturi vezanoj uz vSphere API češće koristi termin operacija, u kontekstu skriptiranja u PowerShellu on je sinonim za metodu – radnju koju objekt može izvršiti (primjerice,. uključivanje stroja ili kreiranje snimke sustava).

Nakon povezivanja sa serverom klijentska aplikacija dobiva referencu na upravljani objekt. Ta se referenca zove ServiceInstance. Slika 2 prikazuje ServiceInstance i neke povezane podatkovne objekte. Na slici nisu prikazana imena svojstava već samo tipovi podataka. MOR predstavlja ManagedObjectReference, vrstu podatkovnog objekta posebne namjene [16].

U tehničkoj dokumentaciji i razvojnim okruženjima, za Managed Object Reference često se koriste sinonimi poput „MoRef“ ili jednostavnije, „reference“ [18]. Suština



Slika 2 *Primjer ServiceInstance-a s podatkovim objektima [17]*

Figure 2 *Example of a ServiceInstance with data objects [17]*

ovog koncepta leži u tome da MOR služi kao jedinstveni identifikator objekta koji se nalazi na strani poslužitelja, omogućujući njegovo precizno adresiranje. Budući da većina API metoda zahtijeva MOR kao ulazni parametar ili ga vraća kao rezultat operacije, njegovo ispravno dohvaćanje predstavlja temeljnu pretpostavku za uspješnu interakciju sa sustavom i programsku manipulaciju resursima.

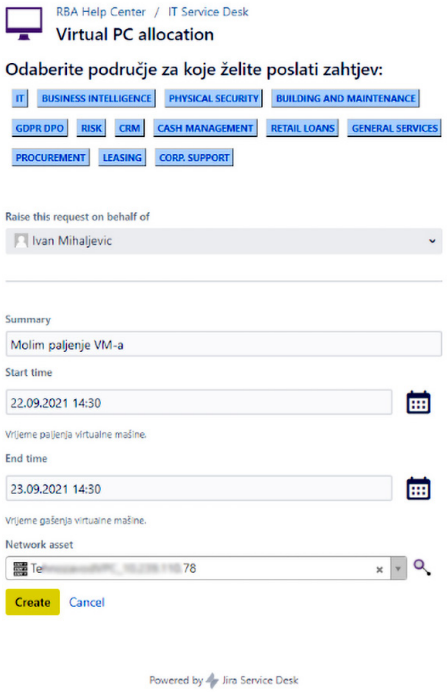
S druge strane, tip podatkovnog objekta osnovna je struktura podataka objektnog modela na strani poslužitelja. Objektni model upravljanja u vSphereu koristi osnove objektno orijentiranih značajki. Primjerice, tipovi upravljanih objekata sastoje se od tipova podatkovnih objekata i primitivnih tipova podataka. Neke vrste upravljanih objekata proširuju druge vrste upravljanih objekata, ali i neke vrste podatkovnih objekata proširuju druge vrste podatkovnih objekata.

4. IMPLEMENTACIJA
4. IMPLEMENTATION

U ovom poglavlju opisan je proces implementacije sustava za automatizaciju dostupnosti virtualnih strojeva. Rješenje je dizajnirano kako bi premostilo jaz između korisničkog zahtjeva za resursima i fizičke aktivacije tih resursa unutar okruženja vSphere. Automatizacija se oslanja na tri ključne točke:

- Jira Service Management: služi kao sučelje za korisnike gdje se definira naziv virtualnog stroja (VM) te točno vrijeme (datum i sat) uključivanja i gašenja.
- PowerShell/PowerCLI skripta: središnji dio automatizacije koji periodički dohvaća podatke iz Jire, obrađuje ih i pretvara u instrukcije za vCenter.
- VMware vCenter Scheduled Tasks: mjesto gdje se instrukcije pohranjuju i izvršavaju u zadano vrijeme.

Iz perspektive sustava Jira, korisnički zahtjevi mogu biti konfigurirani kroz širok spektar elemenata poput polja s ponuđenim opcijama, polja za slobodni unos, radio tipki, kućica za višestruki odabir i sličnih kontrola [19]. Međutim, u specifičnom slučaju automatizacije procesa uključivanja i gašenja virtualnih strojeva, sustav je optimiziran na svega tri ključna podatka, a to su naziv stroja, vrijeme uključivanja te vrijeme gašenja. Svi ovi podaci su unaprijed strogo definirani i korisnik ih bira isključivo iz padajućeg izbornika. Ovakvim pristupom u potpunosti se eliminira mogućnost pogreške u tipkanju, čime se sprječava rizik od neizvršavanja skripte zbog pogrešnih ulaznih parametara.



Slika 3 *Prikaz pripremljenog zahtjeva u Jiri*

Figure 3 *View of prepared request in Jira*

Važan aspekt ovakve implementacije je i automatizirana provjera integriteta zahtjeva koju vrši sama Jira. Sustav provjerava postoji li već ranije zadano vrijeme uključivanja ili gašenja za određeni virtualni stroj te onemogućuje odabir termina koji bi doveli do kolizije. Primjerice, ako je u jednom zahtjevu već odabrano vrijeme uključivanja za 1. prosinca u 8:00 sati i vrijeme gašenja isti dan u 16:00 sati, sustav korisniku neće ponuditi niti dopustiti odabir vremena uključivanja u 9:00 sati. Ključno je naglasiti da se ova provjera vrši isključivo na strani Jire putem njezinih internih pravila i radnih tokova [20] te ona nije dio same PowerShell skripte, čime se osigurava da skripta zaprima samo logički ispravne i provjerene instrukcije.

Naposljetku, sustav uključuje i bitan sigurnosni sloj koji ograničava vidljivost i mogućnost podnošenja ovakvih zahtjeva. Ova vrsta automatizirane usluge vidljiva je samo strogo definiranom, manjem broju korisnika, odnosno isključivo onima koji su evidentirani kao vlasnici specifičnih resursa unutar okruženja VMware. Time je postignuta kontrola nad kritičnom infrastrukturom, osiguravajući da samo ovlaštene osobe mogu upravljati dostupnošću virtualnih poslužitelja kojima upravljaju.

Neposredno nakon zadavanja zahtjeva unutar korisničkog sučelja, poslužitelj Jira putem naredbenog retka inicira pokretanje PowerShell skripte, prosljeđujući joj tri ključna parametra u strogo definiranom redoslijedu. Skripta prvo zaprima naziv virtualnog stroja, a zatim vrijeme uključivanja i vrijeme gašenja, formatirano prema međunarodnom standardu „YYYY-MM-DDTHH:mm:ss“. Postupak pozivanja skripte zahtijeva precizno definiranje sintakse unutar naredbenog retka, gdje se unutar navodnika navodi putanja do same skripte te svaki pripadajući parametar u zasebnim navodnicima. Primjerice, za skriptu naziva VMwareAutoPower.ps1 pohranjenu na lokalnom disku, naredba uključuje punu putanju i argumente za ciljani stroj te pripadajuće vremenske oznake, čime se osigurava ispravna interpretacija unutar operacijskog sustava. Važno je naglasiti kako se za ispravan format vremena ponovno brine Jira, koja korisnički unos pretvara u oblik kompatibilan s PowerCLI modulima prije samog prosljeđivanja skripti.

Kako bi se omogućilo nesmetano izvršavanje automatizacijskih procesa, na poslužitelju s kojeg se skripta pokreće moraju biti instalirani odgovarajući moduli VMware PowerCLI. U ovom konkretnom slučaju, na poslužitelju Jira s Windows operacijskih sustavom, instalacija se provodi putem konzole PowerShell korištenjem naredbe Install-Module VMware.PowerCLI. Pritom je nužno zadovoljiti specifične softverske preduvjete, ovisno o verziji modula. Primjerice, inačica 12.0.0 zahtijeva PowerShell 5.1 ili 7 te minimalno .NET Framework 4.7.2 za Windows okruženja. Uz aktivnu internetsku vezu za preuzimanje paketa, moduli se po završetku procesa pohranjuju unutar korisničkog profila, čime postaju trajno dostupni sustavu za sve buduće automatizacijske zadatke.

Logički slijed same skripte započinje uspostavom komunikacije s poslužiteljem vSphere putem cmdleta Connect-VIServer. Ova naredba inicira novu sesiju ili obnavlja prethodnu, omogućujući sigurnu interakciju s infrastrukturom. Prilikom pozivanja cmdleta, skripti se prosljeđuju mrežna adresa poslužitelja i vjerodajnice servisnog korisnika, kreiranog isključivo za potrebe ove automatizacije radi povećanja sigurnosti i lakšeg

```
Connect-VIServer -server 10.20.100.200 -user imiha -Password Sifra123
$VMname=$args[0]
$TimeToPowerOn=$args[1]
$TimeToPowerOff=$args[2]
```

Slika 4 *Isječak koda skripte za spajanje na server vCenter*

Figure 4 *Script code snippet for connecting to vCenter server*

```
$VM = Get-View -ViewType VirtualMachine -Filter @{"Name" = $VMname}
```

Slika 5 *Isječak koda skripte dohvat objekta VM*

Figure 5 *Script code snippet for retrieving VM object*

```
$entity = New-Object VMware.Vim.ManagedObjectReference
$entity.type = "VirtualMachine"
$entity.Value = $VM.moref.value
```

Slika 6 *Isječak koda skripte referenciranje na stvarni resurs poslužitelja*

Figure 6 *Script code snippet referencing an actual server resource*

nadzora pristupa. Nakon uspješne autentikacije, skripta pohranjuje argumente zaprimljene iz naredbenog retka u interne varijable $VMname, $TimeToPowerOn i $TimeToPowerOff. Pritom se koristi indeksiranje članova niza, gdje brojevi u uglatim zagradama precizno mapiraju predane argumente na pripadajuće varijable, osiguravajući da podaci iz Jire budu ispravno dodijeljeni logici unutar koda.

Nakon uspješnog povezivanja s poslužiteljem i pohrane parametara iz Jire, skripta koristi cmdlet Get-View za dohvaćanje cjelokupnog objekta virtualnog stroja. Ključno je naglasiti razliku između standardne naredbe Get-VM, koja vraća .NET objekt visoke razine, i naredbe Get-View, koja vraća vSphere objekt niske razine izravno povezan s vSphere Web Services API-jem. Upravo su svojstva ovog potonjeg objekta neophodna za daljnji rad, stoga se on pohranjuje u varijablu $VM:

Nadalje, u skripti se deklarira varijabla $entity u koju se sprema objekt klase ManagedObjectReference [21] koja se koristi za upućivanje na upravljani objekt na strani poslužitelja. Ovo je potrebno kako bi se u skripti kasnije moglo definirati na kojem stroju je potrebno kreirati zadatak. ManagedObjectReference ima dva svojstva: tip (type) i vrijednost (value). Tip predstavlja vrstu objekta kojim se upravlja, primjerice, neki host, mapa, profil, podatkovni centar i slično. U našem slučaju, s obzirom da želimo izvršavati operacije nad virtualnim strojem, tip je i postavljen na „VirtualMachine“. Uz tip, potrebno je navesti i vrijednost, a u ovom slučaju ime objekta kojim se upravlja. Ime objekta sadržano je u $VM. moref.value stoga je ta vrijednost iskorištena za definiranje svojstva „value“.

Sljedeća faza implementacije obuhvaća precizno definiranje same akcije i vremenskog rasporeda putem objekta ScheduledTaskSpec. Ovaj složeni objekt služi kao spremnik za sve parametre potrebne za kreiranje zadatka, a unutar vSphere API-ja [22] obuhvaća šest ključnih svojstava koja određuju ponašanje zakazane operacije. To su:

- action: metoda koja će se zapravo izvršiti nad objektom
- description: proizvoljni opis zadatka
- enabled: boolean vrijednost koja definira je li zadatak omogućen ili ne
- name: proizvoljno ime zadatka
- notification: string za upis adrese e-pošte na koju po izvršavanju zadatka dolazi obavijest, mora biti definirana makar i kao prazan string
- scheduler: određuje kada će se zakazani zadatak izvršiti

U skripti se svojstvo scheduler definira kroz objekt OnceTaskScheduler, čime se osigurava da se zadatak izvrši samo jednom u točno zadano vrijeme. OnceTaskScheduler nasljeđuje svojstva od roditelj objekta TaskScheduler,

```
$spec = New-Object VMware.Vim.ScheduledTaskSpec
$spec.name = "$VM Power On $TimeToPowerOn"
$spec.description = "$VM Power On $TimeToPowerOn"
$spec.enabled = $true
$spec.notification = ""
$spec.scheduler = New-Object VMware.Vim.OnceTaskScheduler
$spec.scheduler.activeTime = $TimeToPowerOn
$spec.scheduler.expireTime = "2048-04-15T14:13:00"
$spec.action = New-Object VMware.Vim.MethodAction
$spec.action.name = "PowerOnVM_Task"
```

Slika 7 *Isječak koda skripte priprema planiranog zadataka na poslužitelju*

Figure 7 *Script code snippet preparing a scheduled task on the server*

```
$stm = Get-View -Id 'ScheduledTaskManager-ScheduledTaskManager'
$stm.CreateScheduledTask($entity, $spec)
```

Slika 8 *Isječak koda skripte izrada planiranog zadatka na poslužitelju*

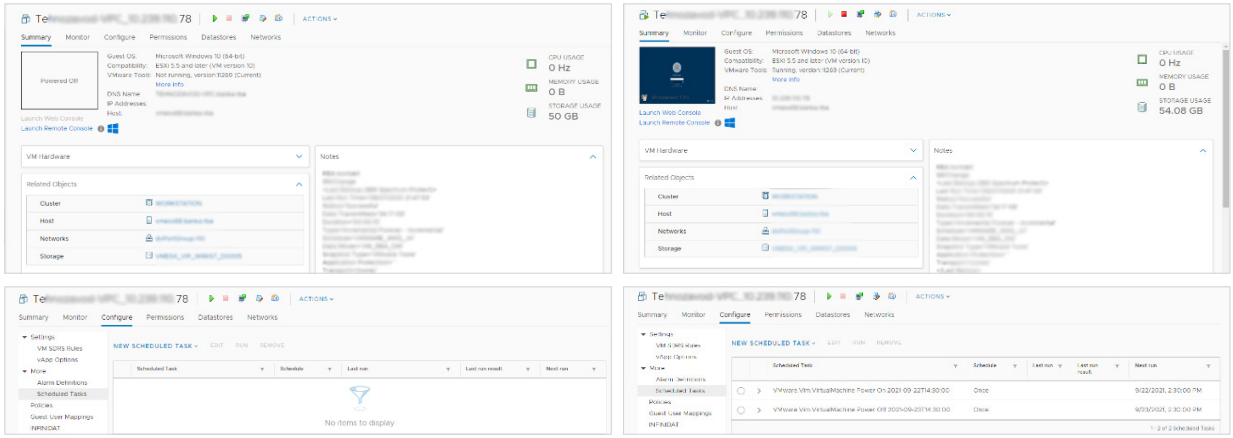
Figure 8 *Script code snippet for creating a scheduled task on the server*

a potrebna svojstva za izvršenje zadatka su svojstvo activeTime, koje definira točan trenutak aktivacije zadatka, te expireTime koji predstavlja krajnji rok do kojeg vCenter smije inicirati operaciju u slučaju privremene nedostupnosti sustava. Postavljanjem ovog parametra na daleku budućnost osigurava se visoka razina pouzdanosti, jer će sustav izvršiti nalog čim se steknu uvjeti, bez rizika od preranog isteka zadatka. Konačno, nakon definiranja vremena, korištenjem objekta MethodAction definirana je metoda koja će se izvršiti. Svojstvo, odnosno metoda uz $spec. action.name, mora odgovarati imenu metode koja se izvršava nad VirtualMachine objektom.

Kada je definirano što će se izvršiti i na kojem stroju, te upute je potrebno proslijediti vCenterovom upravitelju zadataka. To se postiže ponovnim korištenjem Get-View cmdleta, ali ovaj put s ID-jem koji cilja na ScheduledTaskManager. Pozivanjem metode CreateScheduledTask, kojoj se prosljeđuju entitet stroja (definiran kroz unos u Jiri) i pripremljena specifikacija, vCenter preuzima daljnju brigu o izvršenju:

Isto načelo primjenjuje se i za proces gašenja virtualnog stroja. Ključna razlika leži u kreiranju nove specifikacije (primjerice, $spec2) unutar koje se svojstvo action.name postavlja na vrijednost „PowerOffVM_Task“. Ovakvim modularnim pristupom unutar jedne skripte osigurava se kompletna automatizacija životnog ciklusa dostupnosti resursa prema zahtjevu korisnika.

Kako bi se utvrdila vrijednost predloženog rješenja potrebno ga je sagledati u kontekstu komercijalnih alata za automatizaciju clouda, poput VMware Cloud Foundation Automation –



Slika 9 *Stanje planiranih zadataka prije i poslije izrade zahtjeva u Jiri*

Figure 9 *Status of planned tasks before and after creating a request in Jira*

CFA (bivši vRealize Automation) [23]. Korištenje VMware CFA primarno je namijenjeno velikim, heterogenim IT sustavima gdje brojne aplikacije kritično ovise o naprednoj automatizaciji životnog ciklusa resursa. Implementacija takvog rješenja zahtijeva visoko specijalizirano IT osoblje za dugotrajno konfiguriranje cloud templatea i politika zakupa, što često rezultira mjesecima rada prije pune produkcije. S druge strane, predloženi skriptni pristup optimiziran je za specifične funkcije i niske troškove unutar stabilnih okruženja, pružajući agilnost bez potrebe za kompleksnom infrastrukturom. Pregled usporedbe može se vidjeti u tablici ispod.

5. ZAKLJUČAK
5. CONCLUSION

Implementacija sustava za automatizaciju dostupnosti virtualnih resursa putem PowerShella i PowerCLI modula pokazala je kako se moderni IT izazovi mogu uspješno riješiti povezivanjem naizgled heterogenih sustava. Korištenjem Jire kao strogo kontroliranog ulaznog sučelja i vSphere API-ja kao izvršne platforme, stvoren je zatvoreni ciklus koji eliminira potrebu za manualnim intervencijama administratora u rutinskim operacijama uključivanja i gašenja poslužitelja. Posebna vrijednost ovakvog rješenja leži u činjenici da se visoka razina automatizacije može postići korištenjem postojećih razvojnih okvira, čime se izbjegava ulaganje u skupe i složene komercijalne alate trećih strana, kao što su VMware vRealize Automation (vRA) ili Cisco UCS Director, koji često zahtijevaju dugotrajnu implementaciju i znatne troškove licenciranja.

Glavna prednost prikazanog rješenja leži u njegovoj robustnosti i preciznosti. Korištenjem niske razine vSphere objekata, poput ManagedObjectReference i ScheduledTaskSpec, osigurana je visoka pouzdanost izvršenja zadataka unutar same infrastrukture, neovisno o stanju skriptnog poslužitelja nakon inicijalizacije. Ovakav pristup ne samo da smanjuje rizik od ljudske pogreške pri unosu podataka, već i značajno optimizira iskorištenost resursa unutar podatkovnog centra, omogućujući da virtualni strojevi budu aktivni isključivo u razdobljima kada su stvarno potrebni.

Naposljetku, ovaj rad potvrđuje da PowerShell nije samo alat za pomoćnu administraciju, već moćan razvojni okvir sposoban za upravljanje kritičnom infrastrukturom na razini poduzeća. Skalabilnost ovakvog modela otvara prostor za daljnja unaprjeđenja, pri čemu bi logičan sljedeći korak bila potpuna automatizacija životnog ciklusa virtualnih strojeva koja bi uključivala automatsko kreiranje (eng. provisioning) novih VM-ova iz predložaka izravno kroz zahtjeve u sustavu Jira. Iako sam PowerShell podržava deklarativni model putem Desired State Configuration (DSC) mehanizma, za dodatnu razinu automatizacije mogli bi poslužiti Terraform ili Ansible. Ovi alati, u svojim besplatnim verzijama otvorenog koda, mogli bi dodatno obogatiti realizaciju ovog plana, omogućujući precizno upravljanje stanjem resursa i njihovu postinstalacijsku konfiguraciju.

Tablica 1 *Usporedni prikaz predloženog rješenja i komercijalne platforme VMware CFA*

Table 1 *Comparative analysis of the proposed solution and the VMware CFA commercial platform*

Kriterij	**Predloženo rješenje (Jira i PowerShell)**	**Komercijalni alat (VMWare CFA)**
Trošak	Nizak (Jira + vSphere licence)	Visok (dodatna mjesečna pretplata po procesorskoj jezgri za CFA, te hardverski resursi)
Kompleksnost	Niska/Srednja (zahtijeva poznavanje PowerCLI, vSphere API Object Modela)	Visoka (sam postupak integracije i ugađanje platforme CFA u IT sustav)
Fleksibilnost	Visoka (potpuna kontrola nad kodom, uz ograničenja dostupnih funkcija kroz pozive SDK/API)	Ograničena (funkcionalnosti unutar zadanih okvira i grafičkih sučelja platforme ili pisanje naprednih vRealize Orchestrator skripti vRO)
Sigurnost	Primjena servisnih računa s minimalnim pravima (least privilege access)	Ugrađeni napredni RBAC sustavi i politike
Podrška	Interna (vlastiti razvoj i održavanje)	Službena podrška proizvođača

Također moglo bi se ići u smjeru izrade automatiziranih izvještavanja o potrošnji resursa, što bi se moglo realizirati skriptnim prikupljanjem podataka o radnim satima virtualnih strojeva te njihovom automatskom vizualizacijom putem Power BI platforme ili generiranjem interaktivnih HTML dashboarda. Takvi izvještaji omogućili bi menadžmentu jasan uvid u uštede ostvarene gašenjem resursa izvan radnog vremena, otvarajući put za potpunu transformaciju prema samoposlužnom IT modelu (IT as a Service)

6. REFERENCE

6. REFERENCES

[1.] M. S. Liana Fong, "Duality of virtualization: simplification and complexity," ACM SIGOPS Operating Systems Review, vol. 42, no. 1, pp. 96-97, 2008. doi:10.1145/1341312.1341330

[2.] S. Tamane, "A Review on Virtualization: A Cloud Technology," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3, no. 7, pp. 4582-4585, 2015. doi:https://doi.org/10.17762/ijritcc.v3i7.4695

[3.] L. MacVittie, "Automating the Data Center, F5 White Paper," 2008. [Online]. Available: https://cdn.studio.f5.com/files/k6fem79d/production/69af8b6860a8da5ecc3ca46e5788693895cfe0a3.pdf. [Accessed 22 3 2026].

[4.] E. M. Irfan Dhia Irsyad, "A Survey on Designs and Implementations of Virtual Private Network (VPN)," in International Conference on Electrical Engineering and Informatics (ICEEI), Bandung, Indonesia, 2023. doi:10.1109/iceei59426.2023.10346627

[5.] F. Oluwafisayo, "Developing a Framework for Cost Reduction Strategies through Process Automation in SMEs: A United States Perspective," Qeios, 2024. doi:https://doi.org/10.32388/4E7AC7

[6.] S. N. T.-c. Chiueh, "A survey on virtualization technologies," New York, 2005.

[7.] M. Portnoy, Virtualization Essentials, 3rd ed., Hoboken: John Wiley & Sons, 2023, p. 336. doi:10.1002/9781394320608

[8.] Inc., Broadcom, "Broadcom Completes Acquisition of VMware," Broadcom, 2023. [Online]. Available: https://investors.broadcom.com/news-releases/news-release-details/broadcom-completes-acquisition-vmware. [Accessed 22 03 2026].

[9.] B. Đorđević, V. Timčenko, E. Nikolić and N. Davidović, "Comparing Performances of Native Host and Virtualization on ESXi hypervisor," in International Symposium INFOTEH-JAHORINA (INFOTEH), Sarajevo, 2021. doi:10.1109/infoteh51037.2021.9400648

[10.] K. Kuminsky, Implementing VMware VCenter Server, Birmingham: Packt Publishing Ltd, 2013. ISBN 978-1849689984

[11.] J. B. Mark Achtemichuk, "What's New in Performance? VMware vSphere 6.5.," VMware Inc., 2017. [Online]. Available: https://www.vmware.com/docs/whats-new-vsphere65-perf.

[12.] S. L. K. C. Forbes Guthrie, "An Introduction to Designing VMware Environments," in VMware vSphere design, Indianapolis, John Wiley & Sons, 2013, pp. 1-19.

[13.] J. P. T. L. D. J. J. H. Travis Plunk, Learn PowerShell in a Month of Lunches, Fourth Edition: Covers Windows, Linux, and macOS, Manning, 2022.

[14.] L. Dekens, "Work with the vSphere SDK," in VMware vSphere powerCLI reference: automating vSphere administration, Indianapolis, John Wiley & Sons, 2011, pp. 584 - 621.

[15.] Broadcom, "The vSphere Web Services API," Broadcom, 2005-2026. [Online]. Available: https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere-sdks-tools/7-0/virtual-disk-development-kit-programming-guide/practical-programming-tasks/interfacing-with-vmware-vsphere/the-vsphere-web-services-api.html. [Accessed 22 03 2026].

[16.] S. Jin, VMware VI and vSphere SDK: Managing the VMware Infrastructure and vSphere, Boston: Pearson Education, 2009.

[17.] "Managed Object Types Overview," Broadcom, [Online]. Available: https://

developer.broadcom.com/xapis/vsphere-web-services-api/latest/mo-types-landing.html.

[18.] Broadcom, "Managed Object References," [Online]. Available: https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere-sdks-tools/7-0/virtual-disk-development-kit-programming-guide/backing-up-virtual-disks-in-vsphere/design-and-implementation-overview/information-containers-as-managed-objects/managed-object-re. [Accessed 22 03 2026].

[19.] R. Wright, JIRA Strategy Admin Workbook, Northern Virginia: Industry Templates, LLC, 2016, p. 296.

[20.] Atlassian, "Jira Service Management: Conditions and validators," 2026. [Online]. Available: https://confluence.atlassian.com/servicemanagementserver/conditions-and-validators-1044784442.html. [Accessed 22 03 2026].

[21.] Broadcom, "vSphere Web Services API, Data Object - ManagedObjectReference(vmodl.ManagedObjectReference)," [Online]. Available: https://developer.broadcom.com/xapis/vsphere-web-services-api/latest/vmodl.ManagedObjectReference.html. [Accessed 22 03 2026].

[22.] Broadcom, "vSphere Web Services API, Data Object - ScheduledTaskSpec(vim.scheduler.ScheduledTaskSpec)," 2026. [Online]. Available: https://developer.broadcom.com/xapis/vsphere-web-services-api/latest/vim.scheduler.ScheduledTaskSpec.html. [Accessed 22 03 2026].

[23.] Broadcom, "VCF Automation Overview," [Online]. Available: https://techdocs.broadcom.com/us/en/vmware-cis/vcf/vcf-9-0-and-later/9-0/overview-of-vmware-cloud-foundation-9/what-is-vmware-cloud-foundation-and-vmware-vsphere-foundation/vcf-automation-overview.html.

AUTORI · *AUTHORS*

• Ivan Mihaljević - rođen je 17.2.1997. u Zagrebu. Nakon završetka opće gimnazije, upisuje i završava preddiplomski studij informatike na Tehničkom veleučilištu u Zagrebu, a zatim paralelno s pohađanjem diplomskog studija informatike radi u Raiffeisenbank Hrvatska. Nakon završetka studija u istoj tvrtki obavlja posao sistem administratora.

Korespondencija · *Correspondence*

ivanmihaljevic2@gmail.com

• Ognjen Mitrović - Nepromijenjena biografija nalazi u časopisu Polytechnic & Design, Svezak 5, Br. 2 iz 2017. godine.

Korespondencija · *Correspondence*

ognjen.mitrovic@tvz.hr

• Leon Pongrac - rođen 2000. godine u Zagrebu. Preddiplomski i diplomski studij završio je na Fakultetu elektrotehnike i računarstva u Zagrebu na kojem je diplomirao 2025. godine. Tema diplomskog rada je "Vizualizacija dinamičkog nacrta kuće u virtualnoj stvarnosti" Zaposlen je u Ericsson Nikola Tesla gdje radi u odjelu razvoja. Trenutačno je u postupku izbora za vanjskog suradnika na Tehničkom veleučilištu u Zagrebu.

Korespondencija · *Correspondence*

leon.pongrac@gmail.com