

AUTOMATIZACIJA POSLOVA KORIŠTENJEM SYSTEM CENTER ORCHESTRATOR-A

Baruškin, Luka

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:225:389744>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-25**



Repository / Repozitorij:

[Algebra University - Repository of Algebra University](#)



VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

**Automatizacija poslova korištenjem
System Center Orchestrator-a**

Luka Baruškin

Zagreb, siječanj 2020.

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.

U Zagrebu, 10.1.2020.

Predgovor

Nakon ove četiri godine koje su prošle od kako sam upisao Visoko učilište Algebra sa sigurnošću mogu reći da je upis istog bio moja najbolja odluka. Profesori su svojom stručnošću i prijateljskim pristupom omogućili da uspješno upijem što je više moguće znanja. Kako sam već na drugoj godini počeo raditi tvrdim da me niti jedno drugo Učilište ne bi bolje osposobilo za tržište rada. Cijelo školovanje je bilo prožeto stvarnim životnim primjerima i izazovima s kojima su se sreli naši profesori i asistenti. Posebno se zahvaljujem svojem mentoru Vedranu Dakiću koji mi je dočarao i prikazao koliko je zanimljivo i zabavno biti „sistemaš“.

Van akademskog života zahvalio bih svojim roditeljima koji me uvijek s vremenom krenu podržavati u onome što radim zajedno sa ostatkom obitelji, čak i kada se oglušim na njihove dobronamjerne savjete.

Također zahvaljujem kolegama iz Infobip-a sa kojima radim i koji su mi ustupili infrastrukturu na kojoj sam napravio praktični dio ovog rada.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

Ovaj rad govori o načelima automatizacije procesa u informacijskim sustavima. Spominju se rezultati istraživanja provedenih u tvrtkama raznih veličina o utjecaju uvođenja automatizacije. U ovom radu su prikazani prvi koraci na putu prema automatizaciji imaginarne tvrtke Robot d.o.o. Kako ta tvrtka ima informacijski sustav baziran na Microsoft okolini odlučili su koristiti sam System Center Orchestrator alat. Kroz rad je opisan način rada i mogućnosti samog alata. Opisane su različite komponente od kojih se sastoji, objašnjen je postupak instalacije i osnovne konfiguracije, a u zadnjem dijelu je na osam primjera prikazano kako sam alat radi i način na koje je tvrtka Robot d.o.o. zakoračila prema automatizaciji svojih procesa.

Ključne riječi: automatizacija, System Center Orchestrator, Microsoft

Overview

This paper discusses the principles of process automation in information systems. Mention is made of research findings from companies of various sizes on the impact of the introduction of automation. This paper presents the first steps on the path to automation of the imaginary company Robot d.o.o. Because this company has an information system based on the Microsoft environment, they decided to use the System Center Orchestrator tool. Throughout the paper, the method and capabilities of the tool itself are described. The various components of which it is made of are described, the installation procedure and basic configuration are explained, and in the last section, eight examples show how the tool works and how Robot d.o.o. stepped towards the automation of its processes.

Keywords: automation, System Center Orchestrator, Microsoft

Sadržaj

1.	Uvod	1
2.	Microsoft	2
3.	Microsoft System Center	3
4.	Automatizacija.....	4
5.	Microsoft System Center Orchestrator	5
5.1.	Povijest	5
5.2.	Arhitektura sustava.....	5
5.2.1.	Orchestration database.....	6
5.2.2.	Management Server	6
5.2.3.	Runbook Designer	6
5.2.4.	Runbook Server	7
5.2.5.	Orchestration Console and Web Service	7
5.3.	Konkurencija	7
5.4.	Instalacija.....	8
5.4.1.	Preduvjeti.....	8
5.4.2.	Minimalna konfiguracija	8
5.4.3.	Preporučena konfiguracija.....	8
5.4.4.	Postupak instalacije	9
5.4.5.	Konfiguriranje integracije.....	14
5.5.	Integracija s Active Directory-em	17
5.5.1.	Active Directory	17
5.5.2.	Konfiguracija integracije	18
5.6.	Integracija sa Microsoft Exchange-om.....	22

5.6.1.	Microsoft Exchange.....	22
5.6.2.	Konfiguracija integracije	22
5.7.	Integracija sa System Center Operations Mangerom	26
5.7.1.	System Center Operations Manager	26
5.7.2.	Konfiguracija integracije	26
5.8.	Upravljanje Virtualnim Serverima	30
5.8.1.	Kreiranje virtualnog servera prema specifikaciji.....	30
5.8.2.	Brisanje virtualnog servera.....	40
5.8.3.	Upravljanje Checkpointovima.....	42
	Zaključak	45
	Popis kratica	46
	Popis slika.....	47
	Popis tablica.....	49
	Popis kôdova	50
	Literatura	51
	Prilog	52

1. Uvod

Ovaj rad je baziran na imaginarnoj tvrtki Robot d.o.o.. To je tvrtka srednje veličine koja okvirne veličine 300 zaposlenika, 200 stolnih računala, 50 laptopa i 10 stalnih Windows servera.

Kao i svaka IT tvrtka i Robot d.o.o. se bori sa čestim kretanjem ljudi, pozitivnim i negativnim. Jedan od ciljeva je omogućiti tvrtki ubrzanje pripreme IT okoline za dolazak novog zaposlenika.

Za posluživanje svoje web stranice tvrtka je odlučila koristiti svoj server. Kako je web vrlo posjećen samim time server generira veliku količinu logova. Kako bi olakšali upravljanje istima cilj je automatizirati taj proces.

Kako njihovi sistem inženjeri i programski inženjeri imaju svoju testnu okolinu na kojoj često podižu nove virtualne server jedan od ciljeva je olakšati proces upravljanja istima.

Izvori podataka za ovaj rad su stručne knjige i razni internetski izvori, od službene dokumentacija sa do foruma gdje razni informatički stručnjaci dijele svoja iskustva.

This thesis is based on imaginary company Robot d.o.o.. That`s middle sized company that has roughly 300 employees, 200 desktop computers, 50 laptops and 10 permanent Windows servers.

As every IT company Robot d.o.o. is also fighting with frequent people movement, positive and negative. One of goals is to ease up preparing IT infrastructure for new employee.

For hosting of its website company decided to use their own web server. Their website is getting a lot of visits and that causes server to generate large amount of logs. To ease up managing all those logs goal is to automate that process.

Their System Engineers and Developers have test environment where they frequently deploy new virtual machines and one of goals is to make that managing of those servers easier.

Sources for this thesis are professional books and variety of Internet sources, from official documentation pages to forums where IT professionals exchange their experiences.

2. Microsoft

Microsoft je u trenutku pisanja ovog rada najvrjednija informatička tvrtka na svijetu. Njihovo djelovanje se rasprostire na široki spektar područja tako na primjer proizvode hardware, razvijaju vlastite klijentske i serverske operacijske sustave i razne programe za iste.

Najpoznatiji su baš po svojem operacijskom sustavu Microsoft Windows i Microsoft Window Server koji su široko rasprostranjeni, posebice klijentska verzija.

Veliki dio svojih sredstava ulažu u razvoj novih tehnologija poput virtualne stvarnosti ili proširene stvarnosti.

Relativno kratko su se pojavili na tržištu mobilnih uređaja koji su dolazili sa njihovom verzijom Windows Mobile operacijskog sustava.

U sklopu svojeg djelovanja također podupiru razvoj i obrazovanje mladih ljudi, budućih informatičkih stručnjaka, dijelom izravno preko svojeg programa akademija, a dijelom davanje podrške obrazovnim institucijama.

Prepoznaju potrebe modernog svijeta te se uvijek kreću u smjeru budućnosti i te sudjeluju u postavljanju budućih trendova.

Jednostavnost i broj dostupnih aplikacija za Windows operativne sustave im omogućuju veliku prednost nad konkurentima baziranim na otvorenom kodu, no valja napomenuti da ne zanemaruju taj pravac pa su tako unutar svojeg operacijskog sustava ugradili podršku za emulaciju sustava baziranih na otvorenom kodu kako bi korisnici vrlo lako mogli koristiti prednosti oba svijeta.

3. Microsoft System Center

U Microsoftu su prepoznali potrebu centralnog upravljanja informatičkom okolinom, iz tog razloga imaju veliki broj alata koji od samih početaka Windows Server operacijskih sustava omogućuju takav način upravljanja. Administracija se odrađivala korištenjem Microsoft Management konzole i raznih Snap-In-ova, no to i dalje zahtjeva spajanje na svaki pojedini server što je zadovoljavalo u počecima.

Prvi centralizirani proizvod je bio Microsoft System Management Server (SMS) koji je služio upravljanjem i nadzorom konfiguracija korisničkih računala čiji je broj bio znatno veći od broja servera. Oko 2000. godine, predstavljanjem Active Directory-a, pojavile su se mnogo veće mogućnosti, između ostalih Group Policy te se povećao broj potrebnih servera.

U isto vrijeme je Microsoft prepoznao potrebu alata za upravljanje i nadzor servera čiji je broj rastao te su tako 2000. godine kupili prava od NetIQ-a i razvili Microsoft Operations Manager (MOM).

U današnje vrijeme gdje je broj servera velik potrebno nam je nešto s čime možemo odmah biti spojeni na cijelu informatičku okolinu te centralno upravljati s različitim aspektima iste. Prepoznavši tu potrebu Microsoft je razvio paket alata koji pružaju takav način upravljanja informatičkom okolinom, taj set alata se skupno zove Microsoft System Center. Kroz različite verzije sadržavao je različite alate i funkcionalnosti, ovisno o smjeru kretanja tehnologije sve od 2005 počevši sa System Center Essentials paketom[1]. Kako se tehnologija mijenjala javila se potreba za prilagodbom, tako je 2007. godine predstavljen The System Center Suite koji se sastojao od nekoliko alata, svaki za svoju svrhu: System Center Configuration Manager 2007 R2 (SCCM), System Center Operations Manager 2007 R2 (SCOM), System Center Essentials 2007, System Center Data Protection Manager 2007 (SCDPM), System Center Virtual Machine Manager 2008 (), System Center Mobile Device Manager 2008, System Center Capacity Planner 2007 i System Center Service Manager, kasnije su slijedile verzije System Center 2010, System Center 2012, System Center 2012 R2, System Center 2016, System Center 1801, System Center 1807 i System Center 2019.[1]

Za korisnike s velikim informatičkim okolinama baziranim na Microsoft-ovim tehnologijama System Center paket predstavlja veliku vrijednost u vidu uštede vremena i bržeg reagiranja i odrađivanja potrebnih radnji.

4. Automatizacija

S obzirom na povećanje opsega posla informatičkih stručnjaka, koje dolazi sa sve većom informatizacijom društva i potrebom ljudi da budu povezani i da što više olakšaju sve svoje zadatke, prirodno se javila potreba za olakšanjem i ubrzanjem poslova koje obavljaju sami informatički stručnjaci. Tu na scenu dolazi automatizacija, njome se omogućuje veća brzina, preciznost i skalabilnost odrađivanja određenih zadataka. Također nam omogućuje djelomičnu nadoknadu za nedostatak stručnog kadra zbog uštede vremena koju donosi.

Inicijalno zahtjeva napredno poznavanje sustava i procesa kojeg se automatizira te ponekad oduzima veliku količinu vremena, no korištenjem takvih metoda se dugoročno uštedi puno veća količina vremena samim time i novca. No svakako treba pripaziti da se ne radi automatizacija nekog procesa koji se odrađuje jednokratno ili vrlo rijetko zato što bi na automatizaciju tog posla mogli potrošiti znatno više vremena nego što nam je potrebno za inicijalno izvršavanje tog procesa. Isto tako treba pripaziti na procese koji imaju česte iznimke, takvi procesi bi nam oduzeli vrijeme za automatiziranje, a nakon toga još dodatno vrijeme za svaku iznimku.

Najveća prednost automatizacije se vidi u upotrebi Cloud-native servisa gdje tvrtke koje imaju taj način rada koriste automatizaciju za većinu svakodnevnih operacija što im uvelike olakšava i ubrzava sam razvoj softwarea, a samim time i zaradu. Prema istraživanjima tvrtke Capgemini 75% takvih tvrtki kaže da im je automatizacija pomogla kod ostvarenja većeg profita. Uz to 86% tvrdi da im se automatizacijom poboljšalo korisničko iskustvo.[2]

Automatizacija se najčešće radi razvojem potrebnih rješenja prilagođenih za potrebe tvrtke, bilo to razvojem potpunog programa ili samo neke skripte kojom se izvršava određeni niz naredbi sa ulaznim parametrima koje pružimo. No postoji i veliki broj standardiziranih alata za automatizaciju procesa u informatičkom sustavu koji imaju naprednije mogućnosti od samih skripti te pojednostavljaju i ubrzavaju proces automatizacije procesa.

5. Microsoft System Center Orchestrator

Microsoft System Center Orchestrator (SCO) je jedan od standardiziranih alata za automatizaciju. Razvijen je od strane Microsofta i dio je System Center paketa. Samim time ima odličnu gotovu integraciju sa ostalim System Center alatima i ostalim Microsoft servisima.

5.1. Povijest

System Center Orchestrator se prvi puta pojavio unutar System Center paketa sa verzijom System Center Orchestrator 2012, 2012. godine.

Prva verzija je krajem 2012. godine dobila Service Pack 1 te još uvijek ima aktivnu podršku od strane Microsofta sve do 2022. godine.

U 2013. godini sa novom verzijom Windows Servera, 2012 R2, izlazi i istoimena verzija System Center paketa, uključujući System Center Orchestrator.

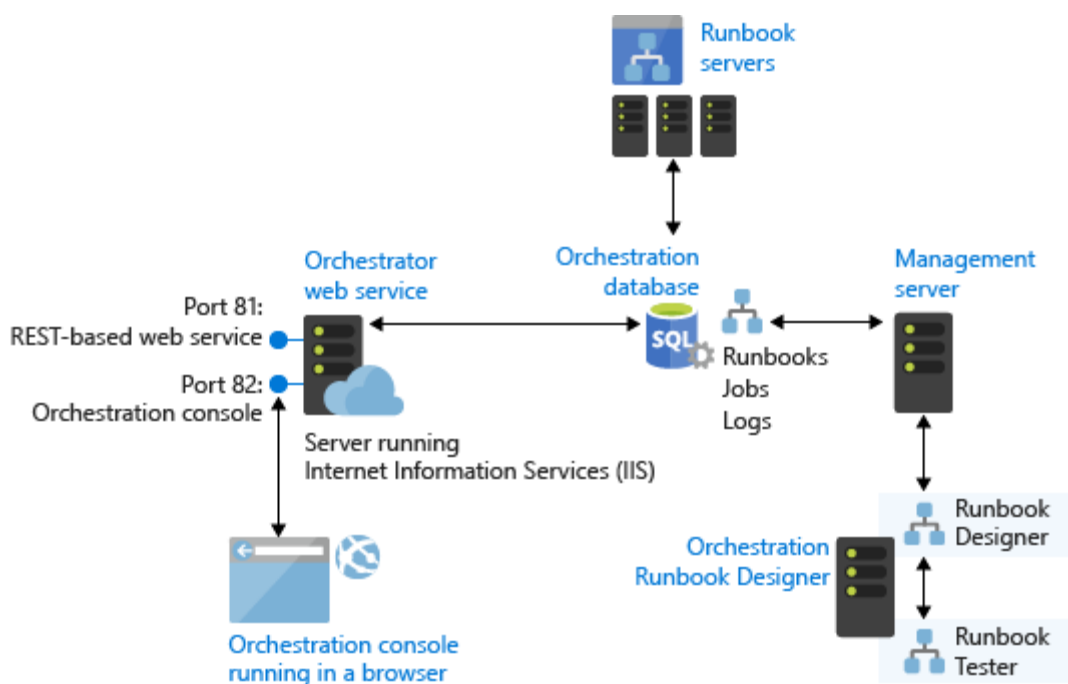
Ista stvar se ponovila 2016. sa verzijom Windows Servera i System Center-a 2016.

Nakon verzije 2016 Microsoft uvodi novo imenovanje te tako izlaze verzije 1801 (veljača 2018.) i 1807 (kolovoz 2018.) da bi se početkom 2019. godine vratili na stari način imenovanja te izlazi trenutno najnovija verzija System Center Orchestrator 2019. [1]

5.2. Arhitektura sustava

Sam sustav se sastoji od četiri glavne komponente koje su potrebne za optimalan rad (Slika 5.1)[3]:

1. Orchestration database
2. Management Server
3. Runbook Designer
4. Runbook Server
5. Orchestration Console and Web Service



Slika 5.1 Arhitektura sustava (izvor: <https://docs.microsoft.com/en-us/system-center/orchestrator/learn-about-orchestrator?view=sc-orch-1801> preuzeto 8.1.2020.)

U nastavku je ukratko opisana svrha svake komponente.

5.2.1. Orchestration database

Orchestratio database je Microsoft SQL baza u kojoj se spremaju sve informacije o Runbook-ovima, konfiguracijama i logovima. Iz tog razloga je preporučeno da bude visoko dostupna te je potreban kvalitetan backup iste.

5.2.2. Management Server

Svrha Management Server-a je da povezuje Runbook Designer servere sa Orchestartion database bazom [3]. Na njemu registramo Integration Pack-ove koje kasnije koristimo na Runbook Designer serverima.

Integration Pack je paket kojim proširujemo mogućnosti System Center Orchestrator, odnosno dobijemo nove funkcije koje možemo koristiti unutar Runbook-ova. Kako im samo ime govori koriste se za integraciju drugih sustava sa System Center Orchestratorom.

5.2.3. Runbook Designer

Runbook Designer koristimo kako bi kreirali nove Runbook-ove.

Runbook je logička cjelina koja se sastoji od jedne ili više akcija koje se izvode po zadanom redoslijedu u određenim uvjetima. U njima zapravo definiramo kako ćemo nešto automatizirati. U Runbook Designer konzoli možemo na principu „drag&drop“ koristiti ugrađene funkcije ili funkcije iz Integration Pack-ova te ih međusobno povezivati.

5.2.4. Runbook Server

Runbook server komunicira izravno s Orchestration database bazom te iz nje dohvaća podatke o novim Runbook-ovima i sprema podatke o poslovima koje odrađuje [3]. To je komponenta koja izvršava sam Runbook i odrađuje automatizaciju.

Svaki Runbook može imati primarni i sekundarne Runbook servere.

5.2.5. Orchestration Console and Web Service

Orchestration Web Service komunicira izravno s bazom i poslužuje Orchestration Console web stranicu [3]. Preko te stranice možemo vidjeti postojeće Runbook-ove i njihove konfiguracije, možemo podesiti primarne i sekundarne Runbook servere te pokretati i pratiti rezultate Runbook-a.

Orchestration Console web stranica zahtjeva Microsoft Silverlight.

5.3. Konkurencija

Očekivano je da Microsoft System Center Orchestrator nije jedini takav proizvod i ostali proizvođači su prepoznali potrebu informatičkih stručnjaka za takvim alatom te su također vrlo aktivni na tržištu.

Glavni konkurent Microsoft System Center Orchestrator-u je VMware Orchestrator koji ima nešto naprednije mogućnosti upravljanja virtualiziranim okolinama baziranim na VMware-ovom vSphere serveru.

Usprkos tome valja napomenuti da Microsoft System Center Orchestrator ima mnogo veći broj gotovih funkcija za automatizaciju te između ostalog može upravljati virtualiziranom okolinom baziranom na VMware vSphere serveru. [4]

5.4. Instalacija

Instalacija Microsoft System Center Orchestrator-a je prilično jednostavan proces, zahtjeva nekoliko servisnih računa, a izvršava se klasično preko čarobnjaka za instalaciju kao i većina aplikacija koje se koriste na Microsoft Windows operacijskim sustavima.

5.4.1. Preduvjeti

Softverski preduvjeti:

Operativni sustav: Windows Server 2016 ili 2019 sa grafičkim sučeljem

SQL: Microsoft SQL Server 2016 ili 2017 (collation: SQL_Latin1_General_CP1_CI_AS)

Microsoft Internet Information Services (IIS)

Microsoft .NET Framework 3.5 Service Pack 1

Microsoft .NET Framework 4

Microsoft SQL Server 2012 Native Client – QFE (u slučaju da koristimo SQL Server 2016)

Hardverski preduvjeti:

Processor: 2.1 GHz, dual-core CPU

RAM: 2GB

Disk: 200MB

5.4.2. Minimalna konfiguracija

Kod malih okolina moguće je sve komponente instalirati na jedan server, zajedno sa Microsoft SQL bazom podataka koja je potrebna za rad System Center Orchestrator-a. [5]

U praktičnom dijelu ovog rada sam se odlučio na takvo rješenje.

5.4.3. Preporučena konfiguracija

Preporučeni način instalacije System Center Orchestrator-a je na više odvojenih servera. I to na način da svaka komponenta bude odvojena na svojem serveru. Kod velikih okolina možda će biti potrebno napraviti klastere određenih komponenti kako bi se raspodijelio teret i postigla visoka dostupnost.

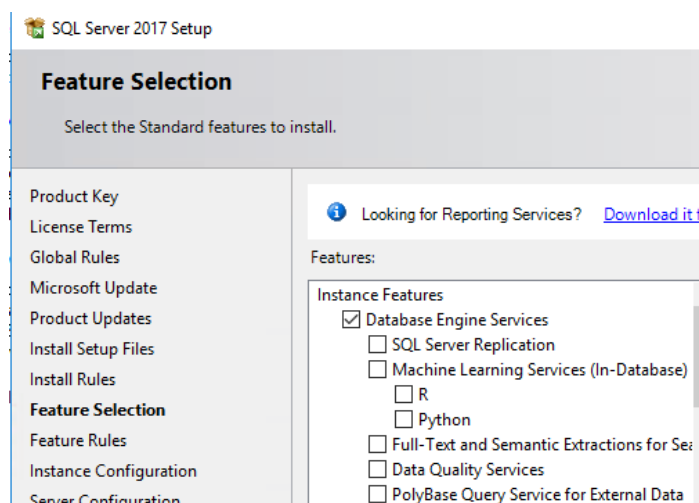
Primjerice bazu koja se koristi za System Center Orchestrator je preporučeno staviti na SQL klaster instancu. Također možemo imati više Runbook servera kako bi u slučaju nedostupnosti jednog naši Runbook-ovi nastavili funkcionirati.

5.4.4. Postupak instalacije

Microsoft SQL Server

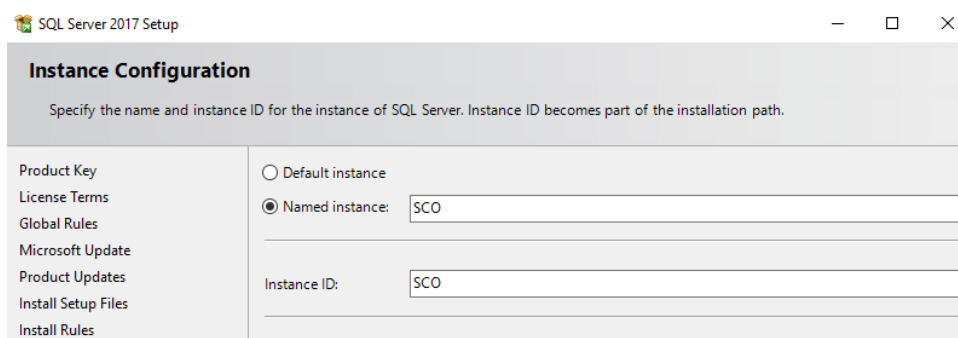
Prije same instalacije preporučeno je da instalirati potrebne preduvjete kako ne bi morali pokretati instalaciju više puta. Iz tog razloga prvo instaliramo SQL Server Standard edition s obzirom da je licenca za isti već uključena sa System Center paketom za potrebe System Center paketa.

Kod instalacije SQL Servera na koraku „Feature Selection“ potrebno je odabrati jedino „Database Engine Services“ (Slika 5.2).



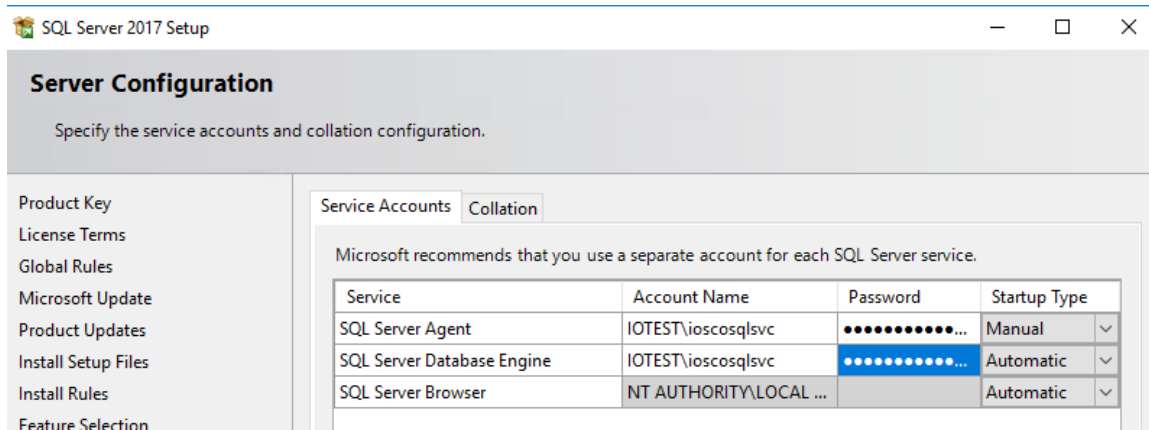
Slika 5.2 SQL Features

Na sljedećem koraku možemo odabrati želimo li koristiti zadano ime instance ili je želimo imenovati, ja sam ju odlučio imenovati (Slika 5.3).

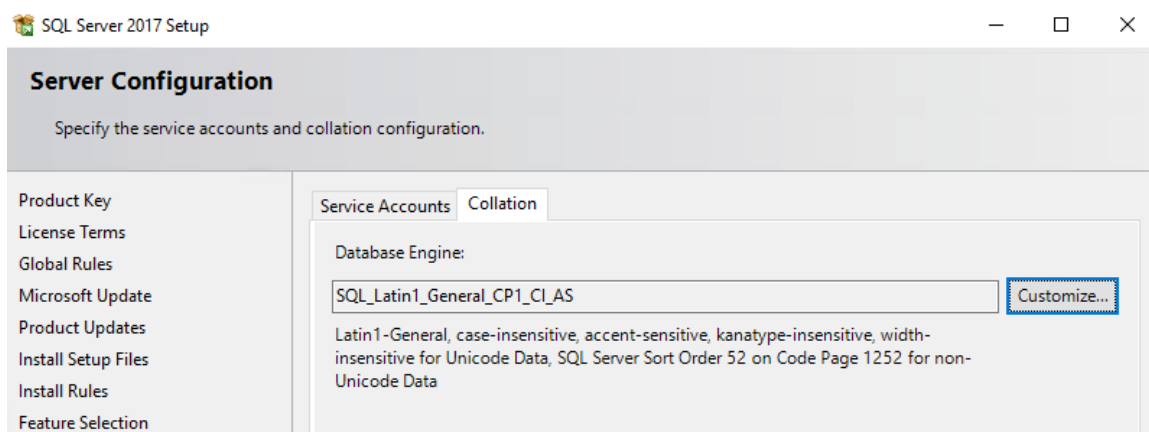


Slika 5.3 Imenovanje SQL instance

Na sljedećem koraku na prvoj kartici definiramo servisne račune za SQL server što je preporučeno da imamo odvojene za Agent i Database Engine, no nije nužno (Slika 5.4), a na drugoj collation, obavezno staviti SQL_Latin1_General_CP1_CI_AS (Slika 5.5).



Slika 5.4 SQL servisni računi

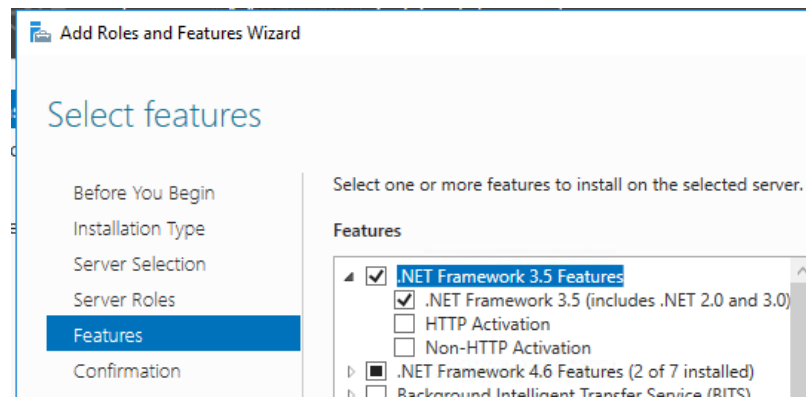


Slika 5.5 SQL collation

Nakon tog još odabiremo način autentikacije, Windows ili Mixed mode te putanje baza što je preporučeno da se nalaze na odvojenim diskovima.

.NET Framework 3.5

Kako bi na Windows Server 2016 instalirali .NET Framework 3.5 potreban nam je instalacijski medij tog operativnog sustava zato što se tamo nalaze potrebne datoteke za instalaciju. Nakon što imamo instalacijski medij dostupan na serveru pokrećemo „Add Roles and Features Wizard“ iz Server Manager aplikacije. U tom čarobnjaku po „Features“ dijelom odaberemo „.NET Framework 3.5 Features“ (Slika 5.6).



Slika 5.6 .NET Framework 3.5 Feature

Čarobnjak nas na sljedećem koraku upozorava da moramo specificirati dodatnu putanju zato što nema potrebne datoteke za instalaciju značajki (eng. Feature) koje smo označili. Tada mu specificiramo dodatnu putanju do „SxS“ mape na instalacijskom mediju koja se obično nalazi u mapi „Sources“ tako je primjer putanje „E:\Sources\SxS“.

Završimo instalaciju .NET Framework 3.5 značajke.

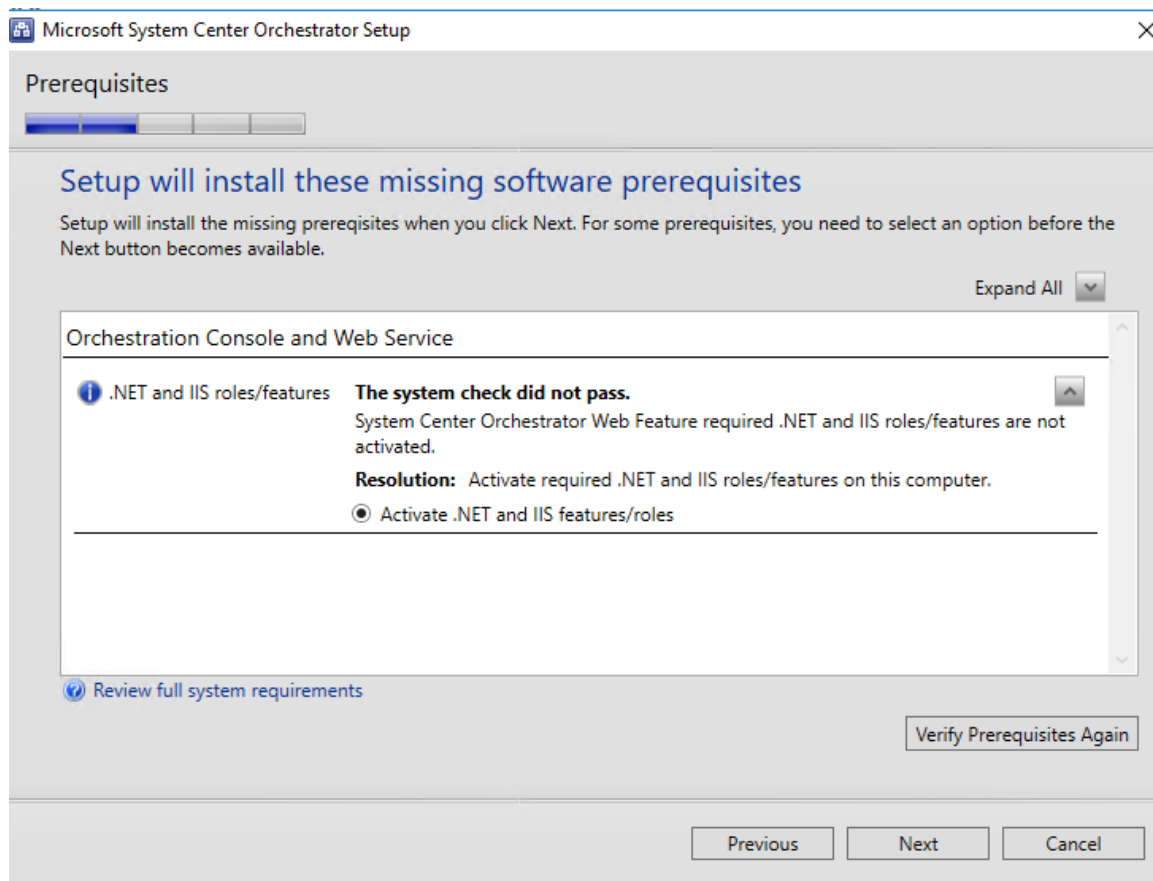
System Center Orchestrator

Sa instalacijskog medija pokrećemo SetupOrchestrator.exe. Na početnom instalacijskom ekranu stisnemo prvi gumb „Install“.

Na „Product Registration“ ekranu unosimo ime odgovorne osobe, ime tvrtke i ključ proizvoda. Sljedeće dvije stranice nam prikazuju uvjete licence i poruku da su slanje dijagnostika i statistika korištenja uključeni kod instalacije i način na koji se mogu ručno isključiti.

Sada odabiremo komponente System Center Orchestrator-a koje želimo instalirati, u mojem slučaju sam odabrao sve komponente s obzirom da se radi o relativno maloj okolini i sve komponente će se nalaziti na jednom serveru.

Zatim slijedi provjera preduvjeta za instalaciju i pokretanje System Center Orchestrator-a. Kako sam unaprijed instalirao Microsoft SQL server i .NET Framework 3.5 ostali preduvjeti će se instalirati kroz sam postupak instalacije System Center Orchestrator-a Slika 5.7.



Slika 5.7 Poruka o instalaciji preduvjeta koji nedostaju

Nakon instalacije i konfiguracije preostalih preduvjeta dobijemo poruku da su svi instalirani i nastavljamo konfiguraciju instalacije samog System Center Orchestrator-a.

Prvi korak je postavljanje servisnog račun, preporuka je imati odvojeni servisni račun. Nakon unosa korisničkog imena i lozinke možemo odmah testirati je li naš unos ispravan.

Zatim postavljamo spajanje na bazu gdje možemo koristiti Windows autentikaciju ili SQL autentikaciju te isto tako testirati konekciju.

Ukoliko se radi o novoj instalaciji na sljedećem koraku odabiremo da želimo kreirati novu bazu, zadano ime je Orchestrator iako ga možemo promijeniti ili koristimo već postojeću bazu.

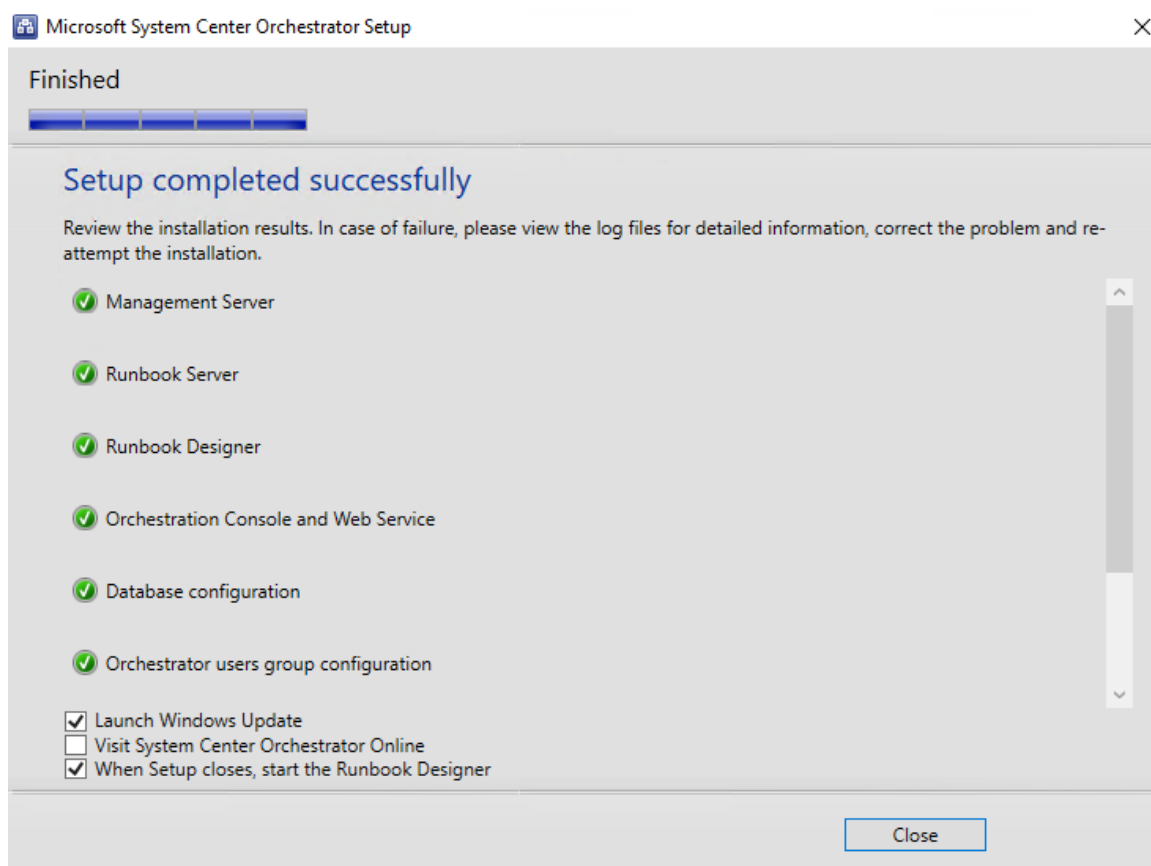
System Center Orchestrator kod instalacije traži dodjelu „Orchestrator users“ grupe koja će imati administrativna prava, preporuka je da grupa bude u Active Directory-u radi lakšeg upravljanja članstvom.

Na sljedećem koraku odabiremo portove za „Web service“ i „Orchestration console“, zadane vrijednosti su 81 i 82, portovi koje odaberemo moraju biti slobodni na našem serveru zato što se na njima kreira nova IIS stranica.

Nakon tog odabiremo instalacijsku putanju, želimo li primati automatska ažuriranja putem servisa Windows Update i želimo li u slučaju greške poslati anonimno izvješće Microsoftu kako bi lakše i brže identificirali i riješili probleme vezane za System Center Orchestrator.

Sljedeća stranica nam sažeto pokazuje sve opcije koje smo odabrali kako bi još jednom potvrdili da je sve u redu prije same instalacije nakon čega pokrećemo instalaciju.

Po završetku instalacije imamo nekoliko izbora, želimo li odmah pokrenuti Windows Update i provjeriti ima li novih ažuriranja, želimo li otvoriti početnu stranicu System Center Orchestrator-a i želimo li pokrenuti Runbook Designer kako bi započeli s radom (Slika 5.8). Kod odabira zadnje opcije automatski nam se otvara Runbook Designer te možemo započeti kreirati vlastite Runbook-ove ili konfigurirati Integration Pack-ove (IP).



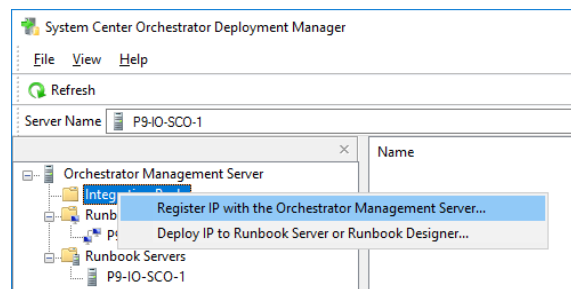
Slika 5.8 Završetak instalacije

5.4.5. Konfiguriranje integracije

Kako bi integrirali bilo koji servis sa System Center Orchestrator-om potrebno je instalirati Integration pack. Integration pack-ovi su dostupni za preuzimanje na Microsoft stranicama (<https://docs.microsoft.com/en-us/system-center/orchestrator/list-of-orchestrator-integration-packs?view=sc-orch-1801>) ili ostalim stranicama ako se ne radio Integration pack-u koji je razvio Microsoft.

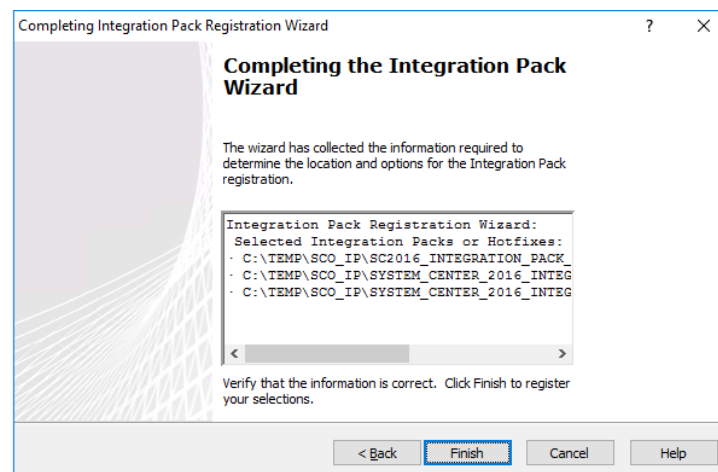
Nakon preuzimanja željenog Integration pack-a potrebno ga je dodati u konfiguraciju System Center Orchestrator-a. Kako bi to napravili koristimo Deployment Manager alat.

Prvo kopiramo željeni Integration pack na lokalni disk Management servera ili u neku dijeljenu mapu. Zatim pokrenemo Deployment Manager, desni klik na Integration Packs i odaberemo opciju „Register IP with the Orchestrator Management server...“ (Slika 5.9)



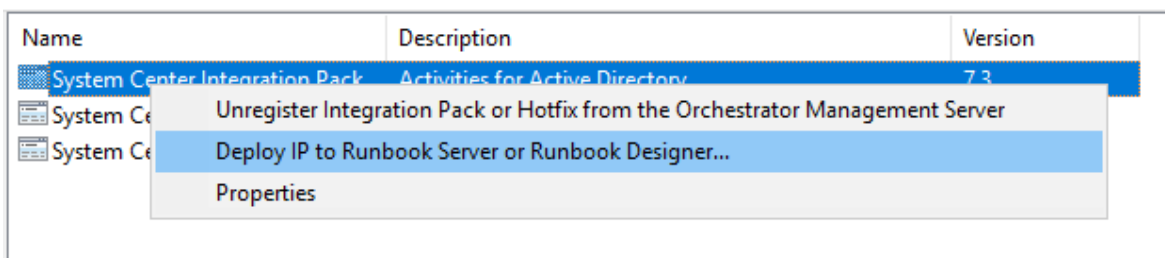
Slika 5.9 Registracija Integration Pack-a

Time se otvara čarobnjak za registraciju Integration pack-ova. Na početnoj stranici kliknemo „Next“, zatim „Add“ i zatim odaberemo .OIP (Orchestrator Integration Pack) datoteke koje želimo registrirati. Nakon odabira odabrani Integration pack-ovi se nalaze na listi i stisnemo „Next“. Po završetku registracije dobijemo potvrdu Slika 5.10.



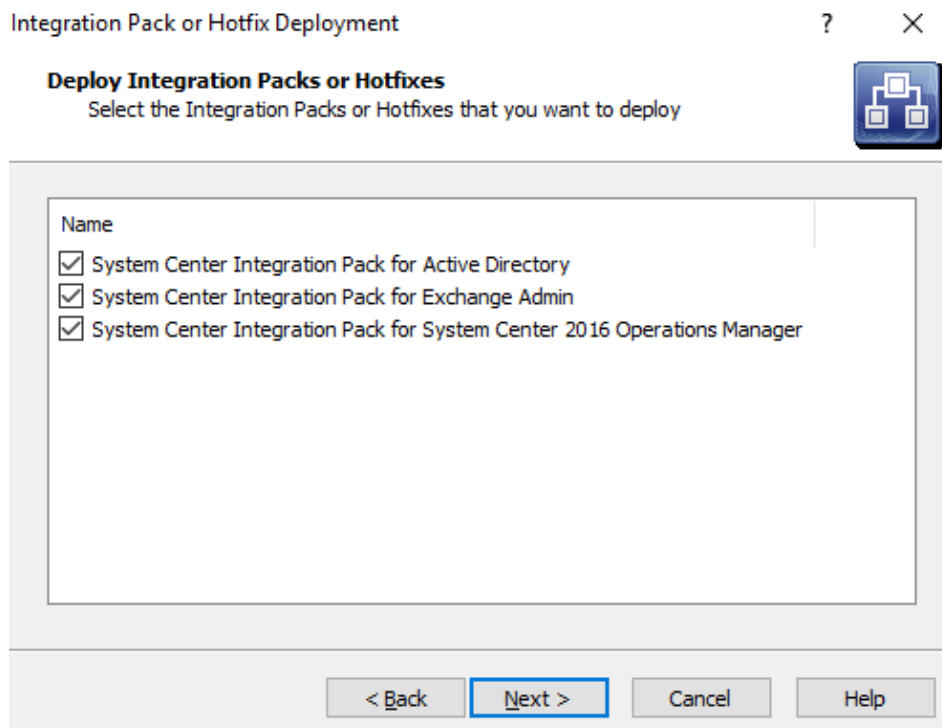
Slika 5.10 Potvrda registracije Integration Pack-a

Nakon tog se registrirani Integration pack-ovi nalaze na listi te možemo desnim klikom kliknuti na bilo koji od njih i odabrati opciju „Deploy IP to Runbook Server or Runbook Designer...” (Slika 5.11).

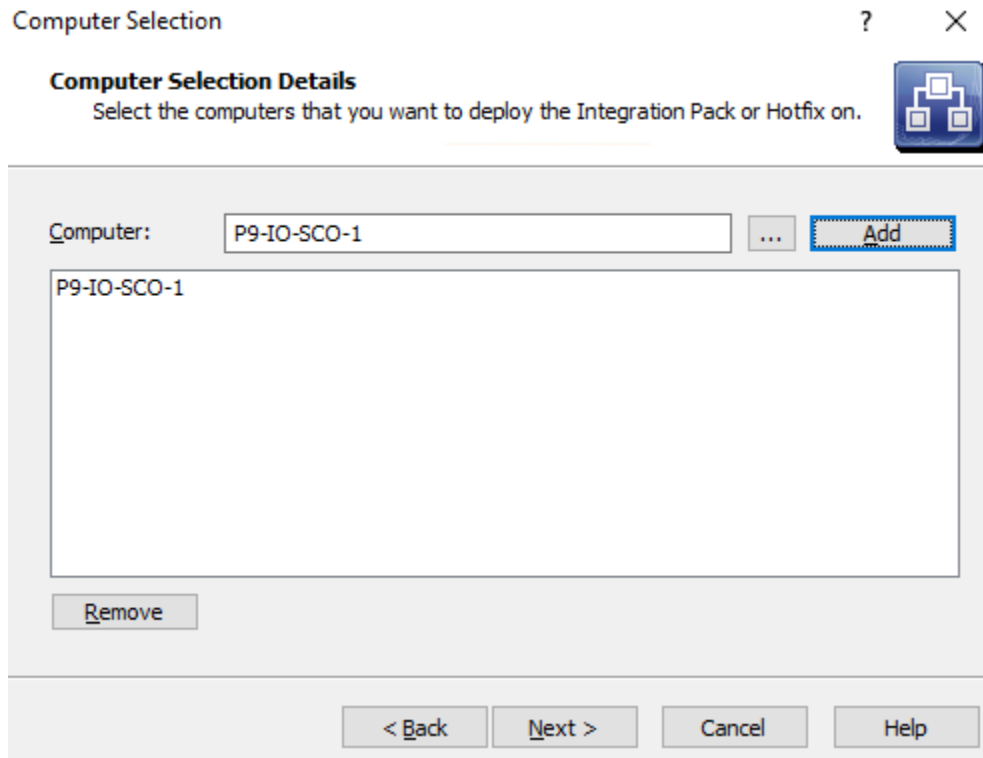


Slika 5.11 Deploy Integration Pack

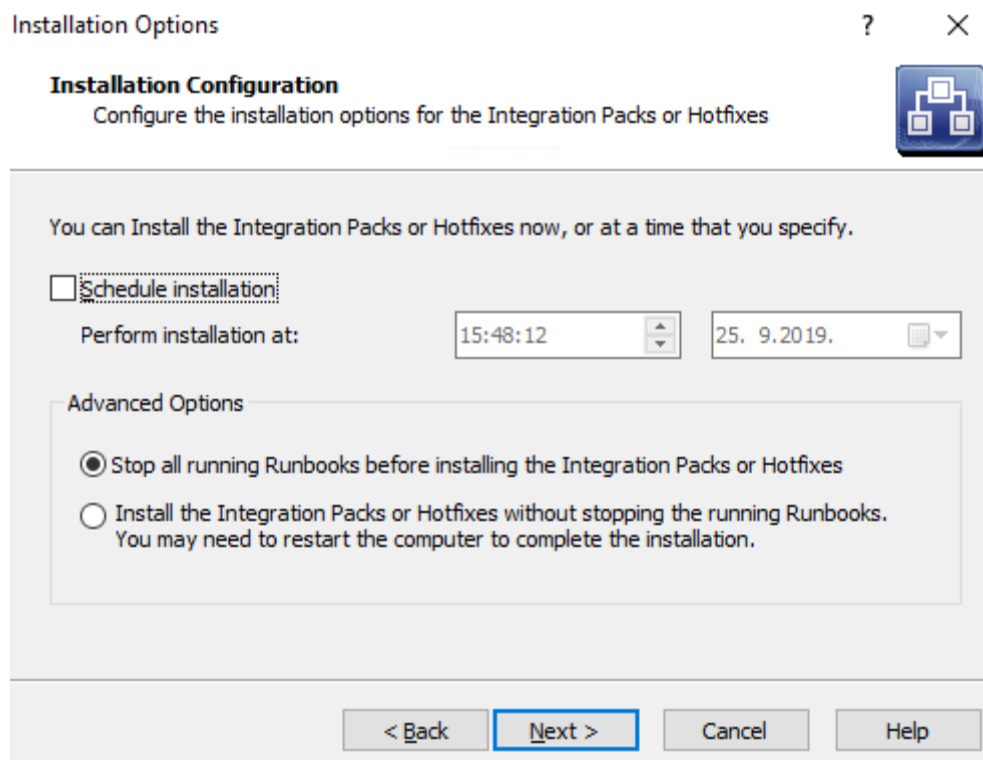
Tada nam se otvara novi čarobnjak gdje možemo odabrati koje Integration pack-ove želimo instalirati (Slika 5.12) na koje servere (Slika 5.13), ako ne želimo svaki Integration pack na svim serverima biti će potrebno nekoliko puta proći ovaj postupak. U mojem slučaju, gdje imam samo jedan server, želim sve Integration pack-ove instalirati na njega kako bih ih mogao koristiti u dizajnu Runbook-ova. Zatim odabiremo opcije instalacije (Slika 5.14) i potvrđujemo odabrane postavke (Slika 5.15). Na kraju pričekamo da završi sama instalacija (Slika 5.16).



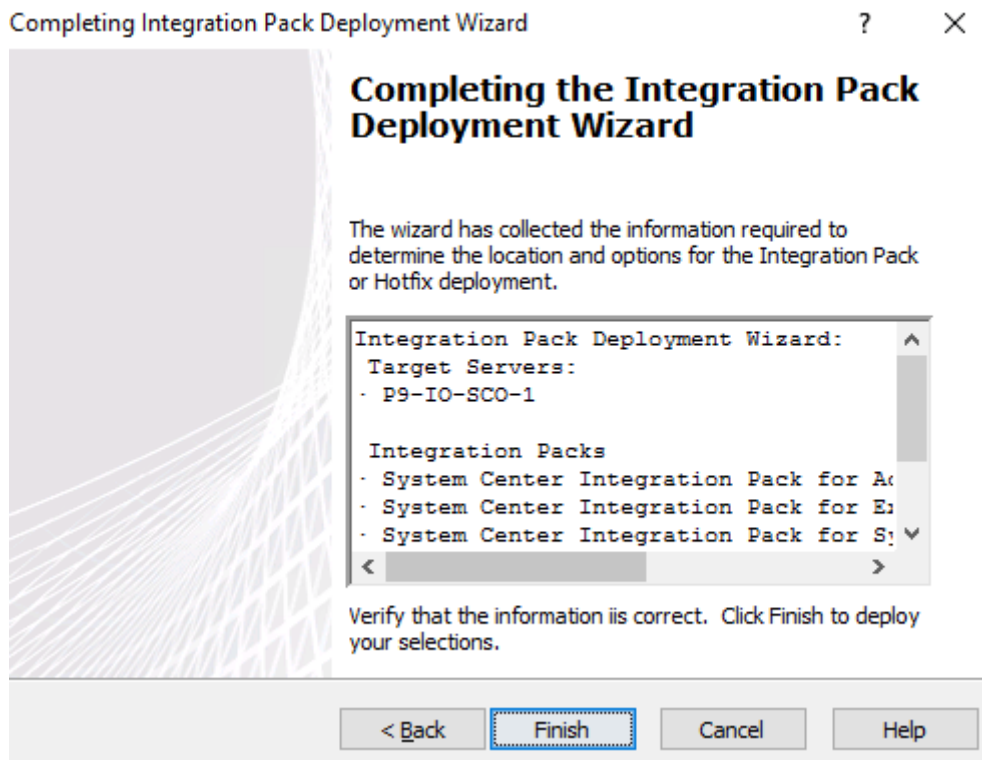
Slika 5.12 Odabir Integration Pack-ova za instalaciju



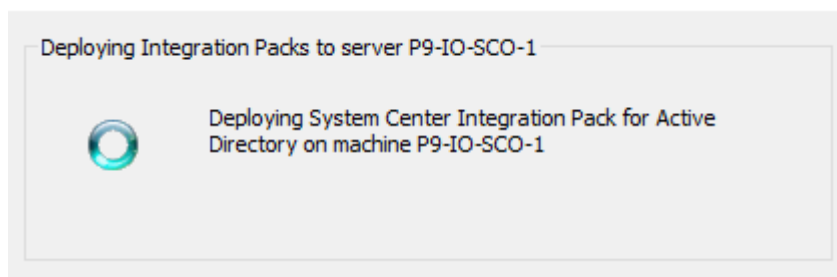
Slika 5.13 Odabir Runbook servera



Slika 5.14 Odabir opcija za instalaciju Integration Pack-ova



Slika 5.15 Potvrda prije instalacije



Slika 5.16 Instalacija Integration Pack-ova

5.5. Integracija s Active Directory-em

Jedna od čestih stvari koje se automatiziraju u tvrtkama je proces koji se odvija kod dolaska novog zaposlenika u tvrtku što najčešće znači minimalno kreiranje korisničkog računa u imeničkom servisu tvrtke, u mojem slučaju Active Directory-u.

5.5.1. Active Directory

Active Directory je Microsoftova izvedba LDAP (Lightweight Directory Access Protocol) imeničkog servisa. Široko je popularan zbog jednostavnosti korištenja i svih mogućnosti koje ima za razliku od nekih drugih implementacija LDAP imeničkog servisa.

Omogućuje centralno upravljanje korisnicima, raspoređivanje u organizacijske jedinice, razne vrste grupa za različite primjene, pruža autentikaciju za praktično bilo koji servis.

5.5.2. Konfiguracija integracije

Kako bi konfigurirali integraciju sa Active Directory-em (AD) potrebno je podesiti postavke Integration pack-a. Moramo mu definirati korisnički račun koji će se koristiti za akcije na Active Directory-u i server na koji će se spajati.

Te akcije izvodimo u Runbook Designer konzoli, klikom na „Options“ i zatim „Active Directory“. Otvara nam se prozor sa konfiguracijama, zato što ih možemo imati više za različite stvari, na taj način možemo granulirati prava. Trenutno nemamo niti jednu te iz tog razloga kliknemo „Add“ nakon čega nam se otvara novi prozor.

Unosimo ime konfiguracije i odabiremo tip, kako se radi o Active Directory Integration pack-u možemo odabrati samo „Microsoft Active Directory Domain Configuration“.

Odabirom tipa otvaraju nam se nova polja koja moramo popuniti, treba nam ime servisnog računa koji ćemo koristiti za interakciju sa AD-om, lozinka tog računa, adresa Domain Controller-a i zadani Organizational Unit (u njemu će se stvarati novi korisnici) (Slika 5.17).

Properties	
Configuration User Name	sco-ad-service
Configuration Password	*****
Configuration Domain Controller Name (FQDN)	ROBO-DC-01.robo.local
Configuration Default Parent Container	OU=Employees,DC=robo,DC=local

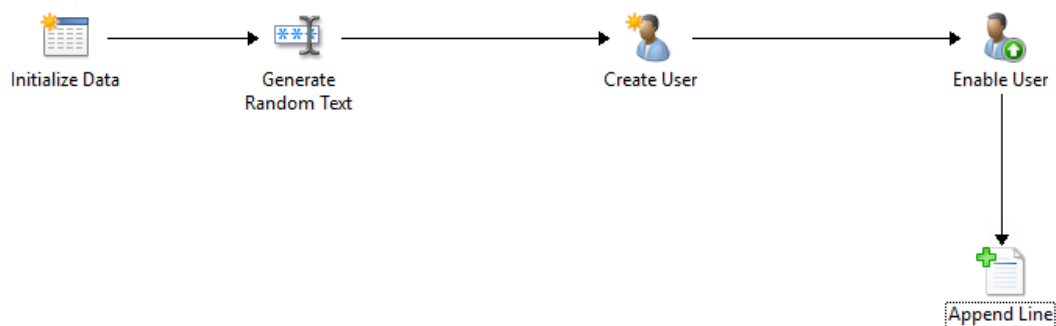
Slika 5.17 Active Directory konfiguracija integracije

Kreiranje korisničkog računa

Sve akcije unutar System Center Orchestrator-a se izvode unutar Runbook-ova zato moramo kreirati novi Runbook koji će služiti za kreiranje korisničkih računa.

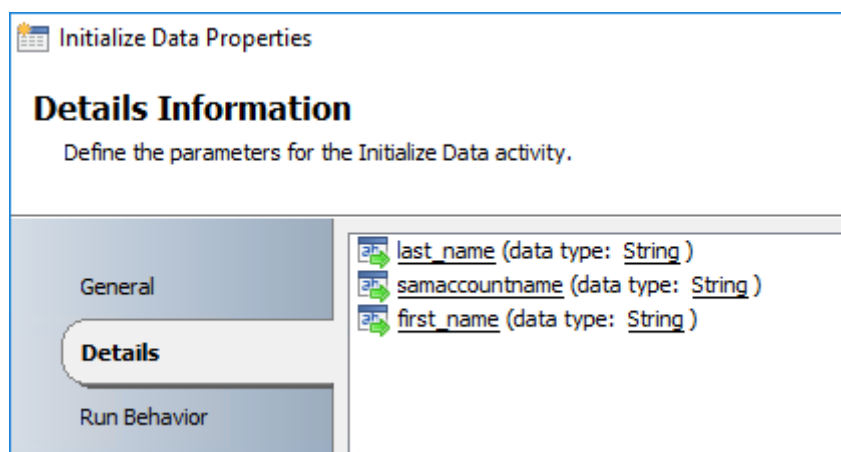
Kako bi uspješno kreirao korisnički račun potrebni su mi određeni parametri, a to su ime, prezime, korisničko ime i lozinka. Te podatke, osim lozinke, ću inicijalizirati kod pokretanja Runbook-a.

Prvi korak je već spomenuta inicijalizacija podataka, nakon tog generiram lozinku, zatim kreiram korisnički račun, uključujem ga i spremam lozinku u tekstualnu datoteku kako bi ju mogao isprintati i dati korisniku prvi radni dan (Slika 5.18).[6]



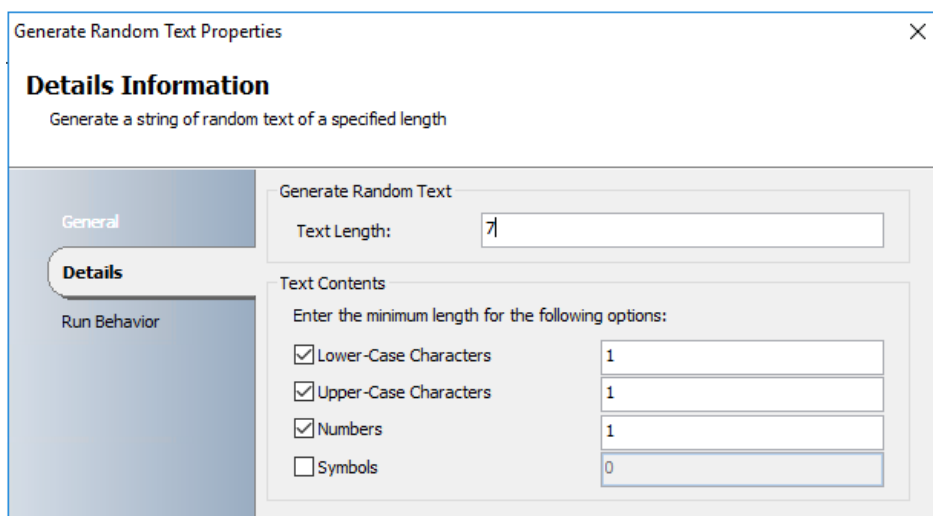
Slika 5.18 Proces kreiranja korisničkog računa

Kod inicijalizacije koristim „Initialize Data“ funkciju u kojoj definiram nazive varijabli koje ću dohvatiti i tip kojeg će biti (Slika 5.19).



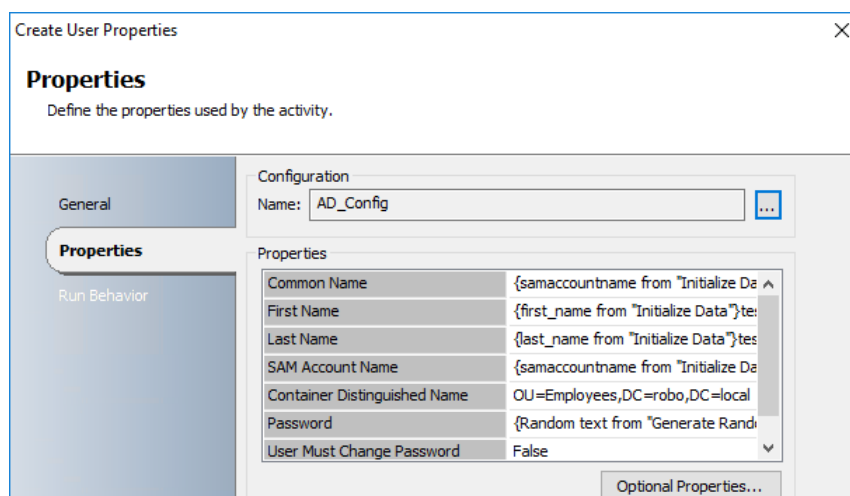
Slika 5.19 Kreiranje korisničkog računa – Initialize Data funkcija

Nakon tog generiram lozinku pomoću „Generate Random Text“ funkcije u kojoj definiram duljinu teksta, broj velikih i malih slova i broj numeričkih znamenki, kako se radi o inicijalnoj lozinki radi jednostavnosti nisu korišteni specijalni znakovi (Slika 5.20).



Slika 5.20 Kreiranje korisničkog računa – Generate Random Text funkcija

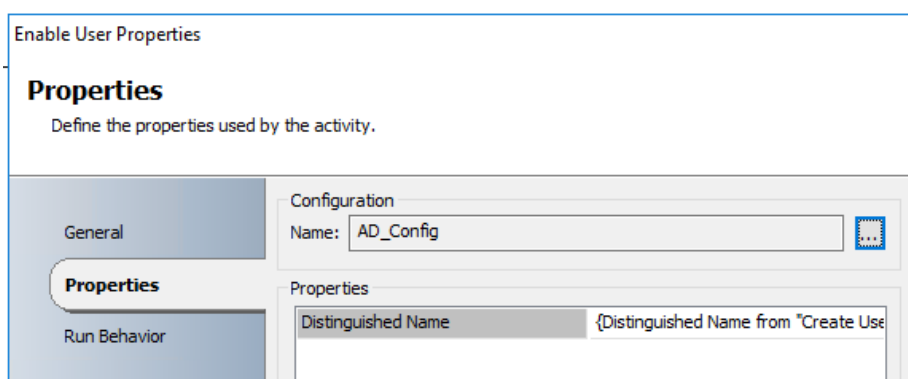
Za kreiranje korisnika unutar Active Directory-a koristim funkciju „Create User“ koja je dio Active Directory Integration pack-a. Za nju sam koristio sljedeće parametre: „Common Name“ koji dohvaćam iz inicijaliziranih podataka, na isti način „First Name“, „Last Name“, „SAM Account Name“, „Container Distinguished Name“ je ručno definiran i označava organizacijsku jedinicu u koju se smještaju svi zaposlenici, „Password“ dohvaćam iz funkcije „Generate Random Text“ i „User Must Change Password“ fiksno postavljam na „False“ kako korisnik ne bi odmah morao promijeniti lozinku već kad bude spreman za to (Slika 5.21).



Slika 5.21 Kreiranje korisničkog računa – Create User funkcija

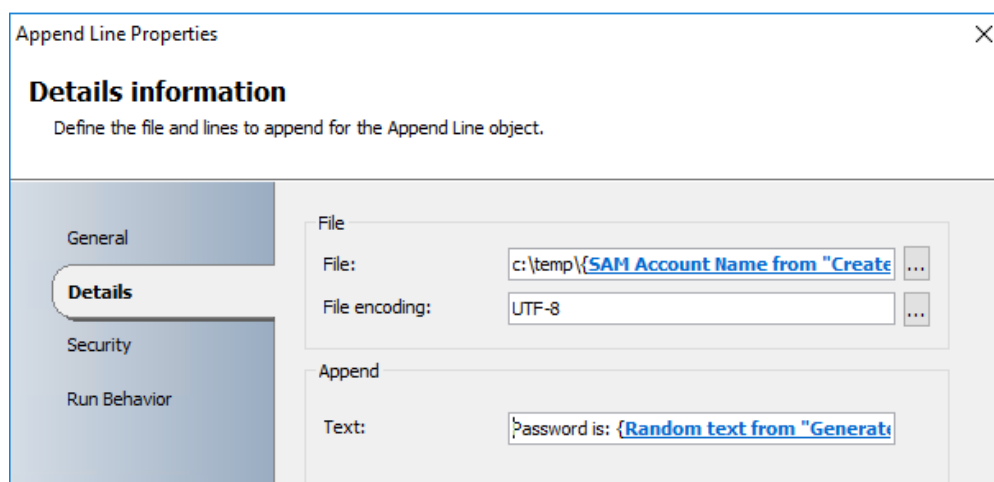
Klikom na gump „Optional Properties...“ otvara se prozor u kojem mogu izabrati koja sve svojstva mogu koristiti kod kreiranja korisničkog računa.

Korisnički račun je nakon kreiranja isključen (eng. Disabled) te ga je potrebno uključiti što se automatski odrađuje sljedećim korakom koji koristi funkciju „Enable User“ ta funkcija zahtjeva samo „Distinguished Name“ parametar koji uzima od izlaza funkcije „Create User“ (Slika 5.22).



Slika 5.22 Kreiranje korisničkog računa – Enable User funkcija

Za kraj cijelog procesa potrebno je spremiti lozinku u tekstualni dokument kako bi ju mogao isprintati i dati korisniku. Za to koristim funkciju „Append Line“ koja dodaje novu liniju u tekstualni dokument. Funkcija je podešena sa sljedećim parametrima: „File“ označava putanju do datoteke, ako ista ne postoji kreira ju, kako bi razlikovao koja je lozinka za kojeg zaposlenika kreiram datoteku za svakog od njih, „Encoding“ se mora mijenjati ako ne podržava znakove koje želim koristiti ili ako upisujem liniju u već postojeću datoteku, „Text“ označava tekst koji će se upisati u tu liniju, u mojem slučaju se uzima tekst generiran funkcijom „Generate Random Text“ (Slika 5.23).



Slika 5.23 Kreiranje korisničkog računa – Append Line funkcija

Dodavanje korisničkog računa u grupe

Korisnici se automatski dodaju u grupu Everybody@robo.local zato što je ista kreirana kao dinamička distribucijska lista.

Ako se u budućnosti pojavi potreba za automatskim dodavanjem u Security grupe biti će potrebno dodati informaciju o odjelu ili timu te funkciju za dodavanje u grupe.

5.6. Integracija sa Microsoft Exchange-om

Sljedeća stvar koja slijedi nakon kreiranja korisničkog računa je kreiranje e-mali pretinca kako bi novi zaposlenik mogao komunicirati sa ostalim kolegama i suradnicima. U mojem radu koristio sam sustav Microsoft Exchange koji je odlično integrirana sa Active Directory-em i System Center Orchestrator-om.

5.6.1. Microsoft Exchange

Microsoft Exchange Server je vjerojatno drugi najpoznatiji Microsoftov proizvod, odmah iza Active Directory-a. To je sustav koji se koristi za razmjenu elektroničkih poruka, e-mail-ova. Microsoft Exchange Server je jedan od najpopularnijih on-premise mail sustava na svijetu.

Usko je povezan sa Active Directory-em te je iz tog razloga jednostavno otvoriti e-mail pretinac za postojećeg Active Directory korisnika.

5.6.2. Konfiguracija integracije

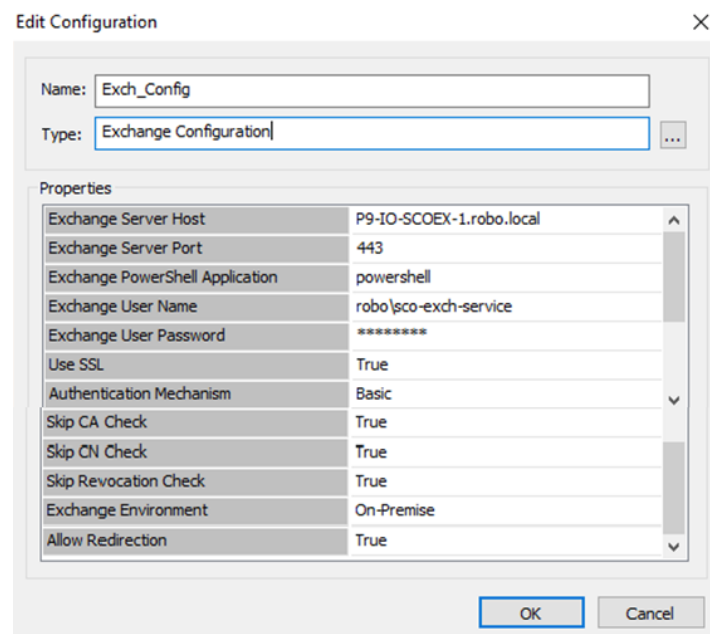
Kao i kod Active Directory integracije potrebno je podesiti postavke Integration pack-a.

U Runbook Designer konzoli kliknemo „Options“ i zatim „Exchange Admin“. Za taj Integration pack nemamo niti jednu konfiguraciju te ju moramo kreirati klikom na „Add“. Otvara se novi prozor gdje upisujemo Naziv konfiguracije i odabiremo tip, koji je u ovom slučaju „Exchange Configuration“.

Nakon odabira tipa pojavljuju se nova polja koja trebamo popuniti (Slika 5.24):

1. „Exchange Server Host“ označava ime Exchange servera na koji ćemo se spajati
2. „Exchange Server Port“ označava port koji ćemo koristiti kod spajanja

3. „Exchange PowerShell Application“ označava naziv stranice na kojoj se nalazi PowerShell
4. „Exchange User Name“ označava korisničko ime koje ćemo koristiti za izvođenje akcija na Exchange serveru i u ovom slučaju sam kreirao posebnog servisnog korisnika.
5. „Exchange User Password“ označava lozinku za tog korisnika
6. „Use SSL“ odabiremo „True“ ili „False“ ovisno o postavkama
7. „Authentication Mechanism“ ima više izbora te također odabiremo prema postavkama servera: Basic, CredSSP, Default, Kerberos, Negotiate, Negotiate with Implicit Credential
8. „Skip CA Check“ ukoliko je „True“ ne provjerava da je certifikat potpisan sa poznatim Certificate Authority-em.
9. „Skip CN Check“ ukoliko je „True“ ne zahtjeva da se Common Name na certifikatu poklapa sa imenom servera.
10. „Skip Revocation Check“ ukoliko je „True“ ne provjerava je li certifikat istekao.
11. „Exchange Environment“ biramo On-Premise ili Online, ovisno o konfiguraciji.
12. „Allow Redirection“ ukoliko je „True“ pratimo redirekciju zahtjeva umjesto da dobijemo grešku.



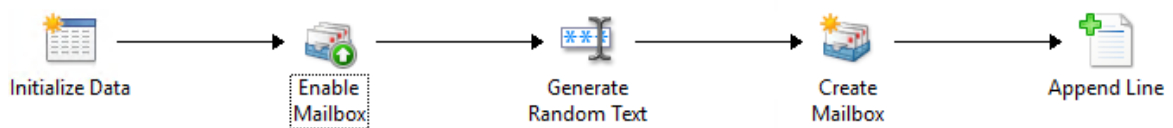
Slika 5.24 Microsoft Exchange konfiguracija integracije

Nakon što smo unijeli sve postavke kliknemo OK i Finnish te možemo započeti sa kreiranjem Runbook-a.

Kreiranje e-mail pretinaca

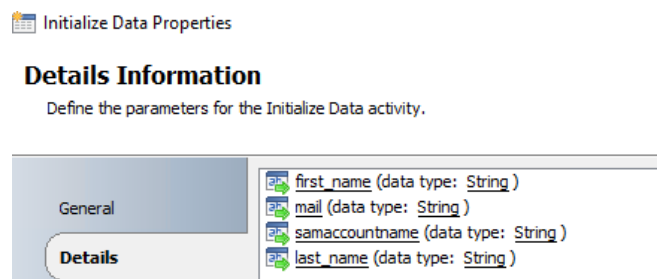
Za kreiranje e-mail pretinca potrebni su mi sljedeći parametri: ime, prezime, e-mail adresa, korisničko ime. Te parametre ću inicijalizirati, osim lozinke, kod pokretanja Runbook-a

Prvi korak je inicijalizacija podataka, zatim pokušavam uključiti e-mail pretinac za tog korisnika, ako korisnik postoji e-mail pretinac se kreira i tu je kraj, ako korisnik ne postoji kreira se te isto kao u primjeru sa Active Directory integracijom generira se lozinka koja se sprema u tekstualnu datoteku, a korisniku se automatski kreira i e-mail pretinac (Slika 5.25).



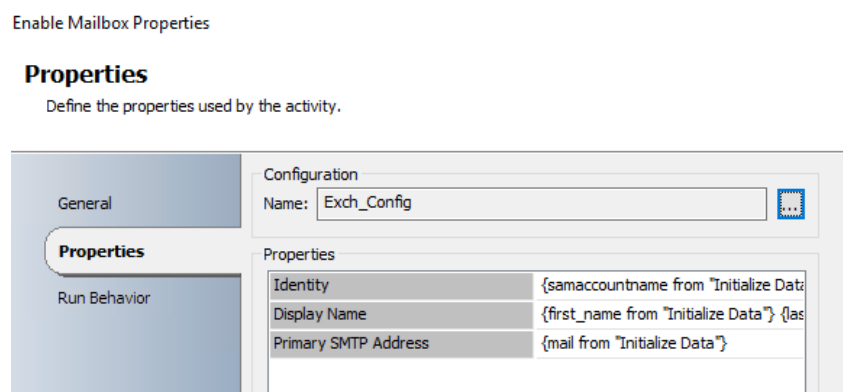
Slika 5.25 Proces kreiranja e-mail pretinca

Kod inicijalizacije koristim „Initialize Data“ funkciju u kojoj definiram nazive varijabli koje ću dohvatiti i tip kojeg će biti (Slika 5.26).



Slika 5.26 Kreiranje e-mail pretinca – Initialize Data funkcija

Nakon inicijalizacije pokušavam uključiti e-mail pretinac pomoću funkcije „Enable Mailbox“ za tog korisnika sama funkcija koristi nekoliko parametara (Slika 5.27) ako korisnik postoji uključi mu se e-mail pretinac i tu završava Runbook.



Slika 5.27 Kreiranje e-mail pretinca – Enable Mailbox funkcija

5.7. Integracija sa System Center Operations Mangerom

Kako bi unaprijedili vrijeme reagiranja na alarme koje izbacuje nadzorni alat System Center Operations Manager pa čak u ovom primjeru i uklonili potrebu za reakcijom odlučio sam integrirati System Center Orchestrator sa System Center Operations Managerom što je odlična sinergija.

5.7.1. System Center Operations Manager

System Center Operations Manager je, kao što mu ime govori, jedan od alata dostupnih u System Center paketu. Primarna zadaća tog alata je nadzor rada infrastrukture i aplikacija. Vrlo je prilagodljiv te ima veliki broj naprednih mogućnosti, iz tog razloga je veoma kompleksan.

Slično kako System Center Orchestrator ima Integration pack-ove tako System Center Operations Manager ima Management pack-ove koji mu omogućuju nadzor određenih komada softwarea i hardwarea.

5.7.2. Konfiguracija integracije

Kao i u dosadašnjim primjerima za konfiguraciju integracije potrebno je otvoriti Runbook Designer konzolu te kliknuti „Options“ i izabrati odgovarajući Integration pack, što je u ovom slučaju „SC 2016 Operations Manager“.

Kako nemamo niti jednu konfiguraciju potrebno je klikom na „Add“ dodati novu. Ta konfiguracija ima nekoliko parametara (Slika 5.29).

1. „Name“ označava ime konfiguracije.
2. „Server“ označava adresu ili DNS ime System Center Operations Manager Management servera.
3. „Domain“ označava ime domene koje je korisnik koji ima prava za spajanje član.
4. „User name“ označava ime korisnika koji će se koristiti za spajanje, preporuča se odvojeni servisni račun.
5. „Password“ je lozinka korisnika koji će se koristiti za spajanje.
6. „Polling“ označava interval u kojem će se provjeravati stanje na serveru.
7. „Reconnect“ označava interval u kojem se veza mora ponovo uspostaviti u slučaju gubitka iste.

Connection ×

Name

Server

Credentials

Domain

User name

Password

Monitoring Intervals

Polling seconds

Reconnect seconds

Slika 5.29 System Center Operations Manager konfiguracija integracije

Klikom na „Test Connection“ možemo provjeriti radi li komunikacija ispravno. Nakon toga možemo klikom na „Ok“ i „Finish“ završiti dodavanje konfiguracije i započeti s kreiranjem Runbook-a.

Brisanje web server logova

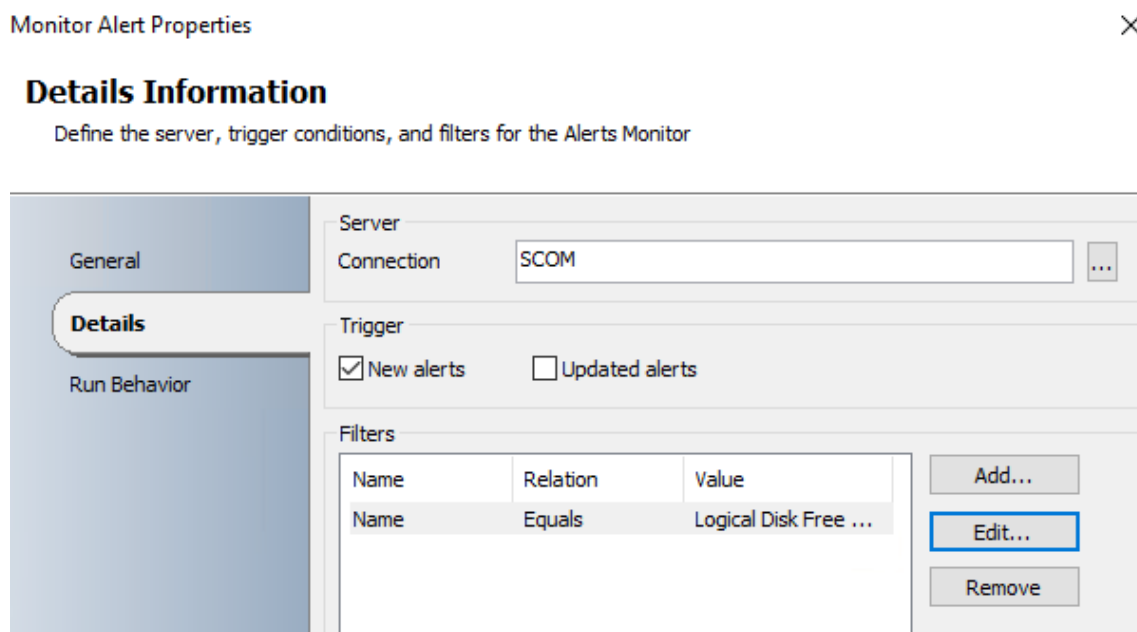
Kako bi automatski brisao stare web server logove kada System Center Operations Manager generira upozorenje potrebno je nekoliko koraka u Runbook-u (Slika 5.30).



Slika 5.30 Proces brisanja web server logova

Prvi korak je praćenje novih upozorenja, drugi korak je samo brisanje logova koje sam izveo pomoću PowerShell skripte koja se spaja na web server i briše sve osim zadnjeg loga, nakon tog radim promjenu upozorenja, odnosno označavam ga kao „Closed“ što znači da je problem riješen, na kraju kako bi ponovo bio spreman kada stigne novo upozorenje automatski ponovo pokrećem Runbook.

Funkciju „Monitor Alert“ koristim kako bi dohvatio informaciju od određenom upozorenju, ona prima sljedeće parametre (Slika 5.31): „Connection“ predstavlja konfiguraciju koju koristim za spajanje na System Center Operations Manager server. Odabirem da želim okidanje samo na nova upozorenja te ih u zadnjem dijelu filtriram, u mojem slučaju prema imenu.



Slika 5.31 Brisanje web server logova - Monitor Alert funkcija

Kada se pojavi novo upozorenje koje odgovara postavljenom filteru prelazi se na sljedeći korak što je izvršavanje skripte koja briše stare logove, sam kod skripte se nalazi u nastavku. Odrađuje spajanje na Web server i brisanje logova, kod spajanja moram koristiti korisnički račun koji ima prava za spajanje i brisanje te nije preporučeno imati lozinke u skriptama. Tu veliku prednost daje System Center Orchestrator pomoću kojeg možemo spremi varijable koje se nakon toga mogu koristiti na bilo kojem mjestu unutar Runbook-a, tako i unutar samog koda skripte. Za spremljene lozinke sam odabrao da vrijednosti budu kriptirane te ih nitko ne može pročitati. Još jedna velika prednost kod takvih varijabli je da centralno možemo promijeniti njihove vrijednosti, na primjer kod promjene lozinke nekog servisnog računara koji se koristi.

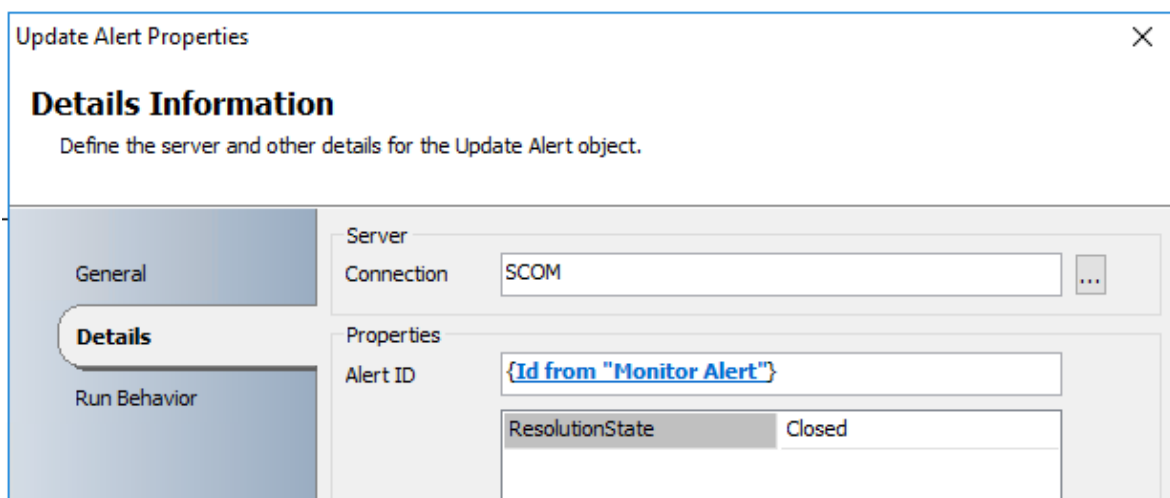
Spremljene varijable su u kodu obojane plavom bojom kako bi se izbjegla zabuna sa Script Block-om.

```
$domainPassword = ConvertTo-SecureString "{Admin Password}" -
AsPlainText -Force
$domainCredential = New-Object
System.Management.Automation.PSCredential ("{Admin
username}", $domainPassword)
Invoke-Command P9-IO-SCOW-1.robo.local -Credential
$domainCredential {
    cd "c:\logs"
    $lastLog = (Get-ChildItem | sort CreationTimeUtc | select
-Last 1).Name
    Get-ChildItem | where Name -ne $lastLog | Remove-Item -
Force
}
```

Kod 5.1 Skripta za brisanje logova sa web servera

Nakon što se skripta izvrši upozorenje generirano od System Center Operations Manager-a ažuriramo sa novim statusom „Closed“ pomoću funkcije „Update Alert“. Ta funkcija također zahtjeva parametar „Connection“ kao i funkcija „Monitor Alert“. Kao drugi parametar treba „Alert ID“ u ovom primjeru koristim ID upozorenja kako bi promijenio samo to upozorenje, a promjena koju radim je da „ResolutionState“ svojstvo postavljam na „Closed“ (Slika 5.32).

Ova funkcija ima brojne druge mogućnosti i može mijenjati velik broj svojstava upozorenja te je iz tog razloga najčešće korištena funkcija u integraciji ova dva proizvoda.



Slika 5.32 Brisanje web server logova – Update Alert funkcija

Zadnji korak je ponovno pokretanje ovog Runbook-a kako bi ponovo bili spremni kada se pojavi novo upozorenje. U tu svrhu koristim funkciju „Invoke Runbook“ koja prima samo dva parametra „Runbook“ koji označava koji Runbook želimo pokrenuti te „Runbook server“ koji označava na kojem Runbook serveru želimo da se taj Runbook izvršava.

Ova funkcija je ugrađena u System Center Orchestrator i njome se omogućuje pozivanje istog ili nekog drugog Runbook-a te iz tog razloga možemo imati bolju granulaciju. Postiže se slična stvar kao sa korištenjem funkcija u programiranju.

5.8. Upravljanje Virtualnim Serverima

Virtualizacija je već dugo vremena jedna od najpopularnijih infrastrukturnih tehnologija. Omogućuje nam da više nemamo zasebne fizičke servere za svaku funkciju ili da imamo preko nekoliko funkcija na jednom fizičkom serveru, unutar jednog operativnog sustava već sada imamo infrastrukturu fizičkih servera koji unutar sebe imaju virtualne. Na taj način svi fizički serveri imaju samo jednu funkciju, a to je virtualizacija, dok virtualni serveri koji se nalaze na fizičkim svaki imaju svoju funkciju, najčešće jedna funkcija po virtualnom serveru. Također svaki virtualni server ima svoj operativni sustav koji je u potpunosti neovisan o operativnom sustavu fizičkog servera.

Microsoft podržava virtualizaciju unutar svojih operacijskih sustava pomoću role Hyper-V. Prva inačica Hyper-V role se pojavila kao dio operacijskog sustava Windows Server 2008, jako je unaprijeđena u 2008 R2 te se u današnje vrijeme koristi u velikom broju tvrtki i svakom verzijom se sve više unaprijeđuje.

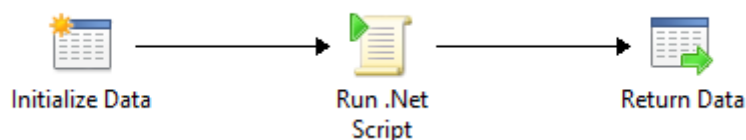
5.8.1. Kreiranje virtualnog servera prema specifikaciji

Kako Robot d.o.o. ima u potpunosti Microsoft infrastrukturu i koristi Hyper-V virtualizaciju System Center Orchestrator je odlično rješenje za automatizaciju kreiranja virtualnih servera za testiranje ili produkciju. Obično proces kreiranja i inicijalnog konfiguriranja virtualnog servera sa Windows Server operacijskim sustavom traje između jednog i dva sata, a sa System Center Orchestrator-om sam to uspio smanjiti na svega desetak minuta.

Kako ova tvrtka nema potrebe za upravljanjem stotinama fizičkih servera nisu se odlučili za implementaciju System Center Virtual Machine Manager-a, još jednog alata iz System Center paketa, koji služi za upravljanje fizičkim virtualizacijskim serverima i ima dostupan

Integration Pack. Tako sam ovaj dio konfiguracije odradio također preko funkcije za pokretanje skripte.

Sam Runbook je prilično jednostavan, prvo se radi inicijalizacija podataka, zatim pokretanje skripte i na kraju se vraća podatak o imenu virtualnog servera čime se označava kraj Runbook-a (Slika 5.33).



Slika 5.33 Proces kreiranja virtualnog servera

U prvom koraku korištenjem funkcije „Initialize Data“ dohvaćam sljedeće podatke:

1. „IPAddressWithSuffix“ predstavlja IP adresu virtualnog servera u CIDR (eng. Classless Inter-Domain Routing) notaciji, npr. 10.116.121.192/22
2. „RAM“ predstavlja količinu radne memorije koja će biti dodijeljena virtualnom serveru, npr. 4
3. „vCPU count“ predstavlja broj virtualnih procesorskih jezgri koje će biti dodijeljene virtualnom serveru, npr. 2
4. „VM Name“ predstavlja ime virtualnog servera.

U sljedećem koraku se pokreće skripta napisana u PowerShell-u, koja se spaja na server koji služi za virtualizaciju te se kreira virtualni server sa zadanim svojstvima. Kod skripte je u nastavku, varijable koje iščitavamo iz System Center Orchestrator-a ili „Intialize Data“ funkcije su obojane plavom bojom kako se ne bi zamijenile sa Script Block-om.

```
$domainPassword = ConvertTo-SecureString "{Service Account  
Passowd}" -AsPlainText -Force  
$domainCredential = New-Object  
System.Management.Automation.PSCredential ("{Service Account  
Username}", $domainPassword)  
  
$localPassword = ConvertTo-SecureString "{Local VM Password}"  
-AsPlainText -Force  
$localCredential = New-Object  
System.Management.Automation.PSCredential ("{Local VM  
Username}", $localPassword)
```



```

$args = @("{VM Name from "Initialize Data"}", "{RAM from
"Initialize Data"}", "{vCPU count from "Initialize
Data"}", "{IPAddressWithSuffix from "Initialize
Data"}", $domainCredential, $localCredential)

Invoke-Command -ComputerName "P9-IO-SCOHV-1" -Credential
$domainCredential {
    $name = $args[0]
    $RAM = $args[1]
    $CPU = $args[2]
    $IPAddressWithSuffix = $args[3]
    $domainCredential = $args[4]
    $localCredential = $args[5]
    Deploy-NewVM -Name $name -IPAddressWithSuffix
$IPAddressWithSuffix -RAM $RAM -CPU $CPU -LocalCredential
$localCredential -DomainCredential $domainCredential

} -ArgumentList $args

```

Kod 5.2 Skripta za kreiranje virtualnog servera

Skripta na samom virtualizacijskom serveru izvršava funkciju „Deploy-NewVM“ unutar PowerShell-a koja je dio posebnog modula napisanog za potrebe tvrtke te se isti nalazi na svim virtualizacijskim serverima. Kod modula, u nastavku, je kompleksniji od samih skripti koje se pokreću kroz System Center Orchestrator te je radi preglednosti i jednostavnosti razvijen kao modul. Kao izlaz funkcije „Deploy-NewVM“ dobije se log file c:\temp\log.txt koji se može koristiti kod otklanjanja mogućih grešaka prilikom kreiranja novog virtualnog servera.

```

function Write-LogStream {
    param (
        # Message that we want to write to log
        [Parameter(Mandatory, Position=0)]
        [string]
        $Message
    )
    $date = get-date -Format "dd.MM.yyyy-hh:mm:ss"
    "$date]" + $Message | out-file "C:\temp\log.txt" -Append
}
function Create-VMFolder {
    param (
        # Name of virtual machine

```

```

        [Parameter(Mandatory)]
        [string]
        $Name
    )
    try {
        $folders = Get-ChildItem "C:\hyper-V" | where
{$_.PSIsContainer -eq $true}
        if ($folders.Name -notcontains $Name){
            $result = New-Item -ItemType Directory -Name
$Name -Path "C:\hyper-v"
        }
        else {
            Write-LogStream "WARNING: Folder with name $Name
already exists."
        }
    }
    catch {
        Write-LogStream "ERROR: Folder with name $Name could
not be created."
    }
    if ($result) {
        Write-LogStream "INFO: Folder successfully created."
        $true
    } else {
        $false
    }
}
function Copy-DiskImage {
    param (
        # Name of virtual machine
        [Parameter(Mandatory,Position=0)]
        [string]
        $Name
    )
    try {
        $folders = Get-ChildItem "C:\hyper-V" | where
{$_.PSIsContainer -eq $true}
        if ($folders.Name -contains $Name){
            Copy-Item -Path (get-childitem "c:\Hyper-
V\image\").FullName -Destination "C:\Hyper-
V\$name\$name.vhdx"

```

```

        $result = Get-Item -Path "C:\Hyper-
V\$name\$name.vhdx"
    }
    else {
        Write-LogStream "WARNING: Folder with name $Name
does not exist."
    }
}
catch {
    Write-LogStream "ERROR: Image disk image could not be
copied."
}
if ($result) {
    Write-LogStream "INFO: Disk image copied."
    $true
} else {
    $false
}
}
function Set-VMNetworkConfiguration {
    Param (
        # VM Network adapter
        [Parameter(Mandatory=$true, Position=0,
ValueFromPipeline=$true)]
        [Microsoft.HyperV.PowerShell.VMNetworkAdapter]
        $NetworkAdapter,
        [Parameter(Mandatory=$true, Position=1)]
        [String]
        $IPAddress,
        [Parameter(Mandatory=$true, Position=2)]
        [String]
        $Subnet,
        [Parameter(Mandatory=$true, Position=3)]
        [String]
        $DefaultGateway,
        [Parameter(Mandatory=$true, Position=4)]
        [String[]]
        $DNSServer
    )
    $countErrors = 0
    try {

```

```

        $VM = Get-WmiObject -Namespace
'root\virtualization\v2' -Class 'Msvm_ComputerSystem' |
Where-Object { $_.ElementName -eq $NetworkAdapter.VMName }
        $VMSettings =
$vm.GetRelated('Msvm_VirtualSystemSettingData') | Where-
Object { $_.VirtualSystemType -eq 'Microsoft:Hyper-
V:System:Realized' }
        $VMNetAdapter =
$VMSettings.GetRelated('Msvm_SyntheticEthernetPortSettingData
')

        $NetworkSettings =
@($VMNetAdapter.GetRelated("Msvm_GuestNetworkAdapterConfigura
tion"))

        $NetworkSettings[0].IPAddresses = $IPAddress
        $NetworkSettings[0].Subnets = $Subnet
        $NetworkSettings[0].DefaultGateways = $DefaultGateway
        $NetworkSettings[0].DNSServers = $DNSServer
        $NetworkSettings[0].ProtocolIFType = 4096
        $NetworkSettings[0].DHCPEnabled = $false

        $Service = Get-WmiObject -Class
"Msvm_VirtualSystemManagementService" -Namespace
"root\virtualization\v2"
        $setIP =
$Service.SetGuestNetworkAdapterConfiguration($VM,
$NetworkSettings[0].GetText(1))

        if ($setip.ReturnValue -eq 4096) {
            $job=[WMI]$setip.job

            while ($job.JobState -eq 3 -or $job.JobState -eq
4) {
                start-sleep 1
                $job=[WMI]$setip.job
            }

            if ($job.JobState -eq 7) {
                Write-LogStream "INFO: IP SET"
            }
        }

```

```

        else {
            $job.GetError()
        }
    } elseif($setip.ReturnValue -eq 0) {
        Write-LogStream "INFO: IP SET"
    }
}
catch {
    Write-LogStream "ERROR: Could not set IP address."
    $countErrors++
}
try {
    Write-LogStream "INFO: Starting VM."
    Start-VM -Name $NetworkAdapter.VmName
}
catch {
    Write-LogStream "ERROR: Could not start VM."
}
if ($countErrors -eq 0) {
    $true
} else {
    $false
}
}
function Create-VM {
    param (
        # Name of virtual machine
        [Parameter(Mandatory,Position=0)]
        [string]
        $Name,
        # RAM size in Gigabytes
        [Parameter(Mandatory,Position=1)]
        [double]
        $RAM,
        # vCPU count
        [Parameter(Mandatory,Position=2)]
        [int]
        $CPU
    )
    $countErrors = 0
    if ((Get-VM).Name -notcontains $Name){

```

```

# Create Virtual Machine
try {
    $vm = New-VM -Path "C:\Hyper-V\$name" -Name $Name
-MemoryStartupBytes $($RAM*1GB) -Generation 2 -BootDevice VHD
-VHDPATH "C:\Hyper-V\$name\$name.vhdx" -SwitchName "External"
-ErrorAction Stop
    if ($vm) {
        Write-LogStream "INFO: VM $Name succesfully
created."
    }
}
catch {
    Write-LogStream "ERROR: VM could not be created."
    $countErrors++
}
# Set correct CPU count
try {
    $vm | Set-VM -ProcessorCount $CPU -ErrorAction
Stop
    Write-LogStream "INFO: CPU count set to $CPU."
}
catch {
    Write-LogStream "ERROR: CPU count could not be
set."
    $countErrors++
}
} else {
    Write-LogStream "WARNING: VM with name: $Name already
exists on this host."
    $countErrors++
}
if ($countErrors -eq 0) {
    $true
} else {
    $false
}
}
function Add-VMtoDomain {
    param (
        # Name of virtual machine

```

```

        [Parameter(Mandatory)]
        [string]
        $Name,
        # Local VM credential
        [Parameter(Mandatory)]
        [pscredential]
        $LocalCredential,
        # Domain credential
        [Parameter(Mandatory)]
        [pscredential]
        $DomainCredential,
        # IP address of virtual machine
        [Parameter(Mandatory)]
        [string]
        $IPAddress
    )
    $countErrors = 0
    Write-LogStream "Adding VM to domain."
    try {
        Add-Computer -ComputerName $IPAddress -
LocalCredential $LocalCredential -Credential
$DomainCredential -NewName $Name -DomainName "robo.local" -
OUPath "OU=Servers,DC=robo,DC=local" -Restart -Force
    }
    catch {
        Write-LogStream "ERROR: Could not join VM to domain."
        $countErrors++
    }
    if ($countErrors -eq 0) {
        $true
    } else {
        $false
    }
}
function Deploy-NewVM {
    [CmdletBinding()]
    param (
        # Name of virtual machine
        [Parameter(Mandatory)]
        [string]
        $Name,

```

```

        # IP address of virtual machine
        [Parameter(Mandatory)]
        [string]
        $IPAddressWithSuffix,
        # RAM size in Gigabytes
        [Parameter(Mandatory)]
        [double]
        $RAM,
        # vCPU count
        [Parameter(Mandatory)]
        [int]
        $CPU,
        # Local VM credential
        [Parameter(Mandatory)]
        [pscredential]
        $LocalCredential,
        # Domain credential
        [Parameter(Mandatory)]
        [pscredential]
        $DomainCredential
    )

    begin {
        $ErrorActionPreference = "Stop"
    }

    process {
        if (Create-VMFolder -Name $Name) {
            if (Copy-DiskImage -Name $Name) {
                if (Create-VM -Name $Name -RAM $RAM -CPU
$CPU) {
                    $IPAddress =
$IPAddressWithSuffix.split("/") [0]
                    if (Get-VMNetworkAdapter -VMName $Name |
Set-VMNetworkConfiguration -IPAddress "$IPAddress" -Subnet
"255.255.252.0" -DNSServer "10.116.121.57" -DefaultGateway
"10.116.120.1") {
                        Start-Sleep "600"
                        if (Add-VMtoDomain -LocalCredential
$LocalCredential -DomainCredential $DomainCredential -
IPAddress $IPAddress -Name $Name) {

```

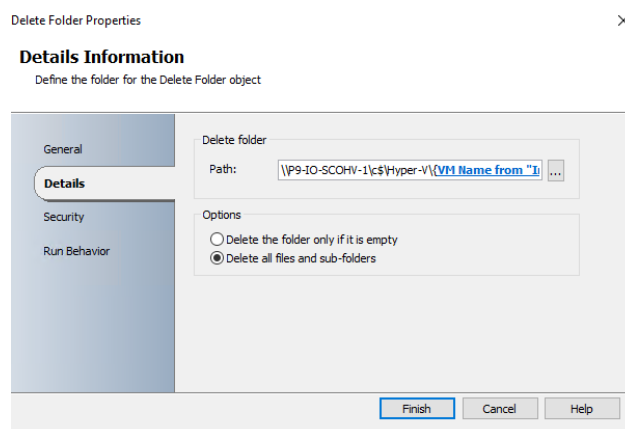

Spremljene varijable su u kodu obojane plavom bojom kako bi se izbjegla zabuna sa Script Block-om.

```
$domainPassword = ConvertTo-SecureString "{Service Account  
Passowd}" -AsPlainText -Force  
$domainCredential = New-Object  
System.Management.Automation.PSCredential ("{Service Account  
Username}", $domainPassword)  
  
$args = @("{VM Name from „Initialize Data“}")  
  
Invoke-Command -ComputerName "P9-IO-SCOHV-1" -Credential  
$domainCredential {  
    $name = $args[0]  
    Get-VM $name | Stop-VM -Force  
    Get-VM $name | Remove-VM -Force  
} -ArgumentList $args
```

Kod 5.4 Skripta za brisanje virtualnog servera

Funkcija „Delete Computer“ služi za brisanje zapisa iz Active Directory-a te prima ima parametre „Configuration Name“ koji označava koju konfiguraciju za spajanje na Active Directory želimo koristiti te „Distinguished Name“ zapisa koji želimo ukloniti što je u ovom slučaju kombinacija vrijednosti imena iz „Initialize Data“ funkcije i fiksnog dijela kojim je usmjeren na dobru organizacijsku jedinicu.

Funkcija „Delete Folder“ ima samo parametar „Path“ kojim označavamo putanju do mape koju želimo izbrisati, u ovom slučaju se radi o putanji na udaljenom serveru te je kombinacija fiksnog dijela putanje i imena virtualnog servera koji se briše. Također je bilo potrebno označiti opciju da se obrišu sve datoteke i podmape(Slika 5.35).



Slika 5.35 "Delete Folder" funkcija

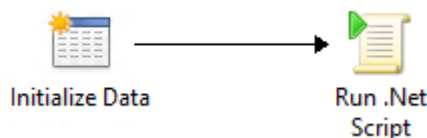
5.8.3. Upravljanje Checkpointovima

Kako bi omogućili brže i lakše testiranje možemo koristiti Checkpointe ili Snapshote, obje riječi su prihvatljive s obzirom da ih sam Microsoft koristi naizmjenice. Prednost korištenja Checkpointa je ta što korisnik može spremiti trenutno stanje virtualnog servera, odraditi neku akciju, npr. ažuriranje, te ako ta akcija uzrokuje nestabilnost ili druge probleme na virtualnom serveru može se brzo vratiti na prijašnje stanje.

Kod upravljanja Checkpointovima imamo tri bitne akcije, a to su:

- Kreiranje
- Primjena
- Brisanje

Svaka akcija ima svoj Runbook te su svi vrlo jednostavni, u „Initialize Data“ funkciji primaju vrijednosti o imenu virtualnog servera i imenu Checkpointa. Nakon toga se izvršava kratka skripta koja odrađuje potrebnu konfiguraciju (Slika 5.36).



Slika 5.36 Proces upravljanja Checkpointovima

Kreiranje Checkpointa

Kreiranjem Checkpointa zapravo spremamo trenutno stanje virtualnog servera kako bi ga kasnije mogli vratiti u to stanje ili pobrisati radi lakšeg upravljanja virtualnim serverom.

Kod skripte za kreiranje Checkpointa se nalazi u nastavku, spremljene varijable su u kodu obojane plavom bojom kako bi se izbjegla zabuna sa Script Block-om.

```
$domainPassword = ConvertTo-SecureString "{Service Account  
Passowd}" -AsPlainText -Force  
$domainCredential = New-Object  
System.Management.Automation.PSCredential ("{Service Account  
Username}", $domainPassword)  
$args = @("{VM Name from „Initialize Data“}", "{Checkpoint  
Name from „Initialize Data“}")  
$return = Invoke-Command -ComputerName "P9-IO-SCOHV-1" -  
Credential $domainCredential {  
    $name = $args[0]
```

```

        $checkpointName = $args[1]
        Get-VM $name | Checkpoint-VM -SnapshotName
$checkpointName
        if ($(Get-VMSnapshot -VMName $name -Name $checkpointName
-ErrorAction SilentlyContinue).Count -eq 1){
            1
        } else {
            0
        }
    } -ArgumentList $args
$return

```

Kod 5.5 Skripta za kreiranje Checkpointa

Primjena Checkpointa

Primjenom Checkpointa se vraćam na neko bivše spremljeno stanje, najčešće se to radi kada nam akcija na virtualnom serveru nije prošla planirano.

Kod skripte za primjenu Checkpointa se nalazi u nastavku, spremljene varijable su u kodu obojane plavom bojom kako bi se izbjegla zabuna sa Script Block-om.

```

$domainPassword = ConvertTo-SecureString "{Service Account
Passowd}" -AsPlainText -Force
$domainCredential = New-Object
System.Management.Automation.PSCredential ("{Service Account
Username}", $domainPassword)
$args = @("{VM Name from „Initialize Data“}", "{Checkpoint
Name from „Initialize Data“}")
$return = Invoke-Command -ComputerName "P9-IO-SCOHV-1" -
Credential $domainCredential {
    $name = $args[0]
    $checkpointName = $args[1]
    Get-VM $name | Restore-VMCheckpoint -Name $checkpointName
-Confirm:$false
} -ArgumentList $args
$return

```

Kod 5.6 Skripta za primjenu Checkpointa

Brisanje Checkpointa

Brisanje Checkpointa izvodimo nakon što utvrdimo da je radnja na virtualnom serveru uspješno odrađena i da neće biti potrebe za vraćanjem u to stanje.

Kod skripte za brisanje Checkpointa se nalazi u nastavku, spremljene varijable su u kodu obojane plavom bojom kako bi se izbjegla zabuna sa Script Block-om.

```
$domainPassword = ConvertTo-SecureString "{Service Account  
Passowd}" -AsPlainText -Force  
$domainCredential = New-Object  
System.Management.Automation.PSCredential ("{Service Account  
Username}", $domainPassword)  
$args = @("{VM Name from „Initialize Data“}", "{Checkpoint  
Name from „Initialize Data“}")  
$return = Invoke-Command -ComputerName "P9-IO-SCOHV-1" -  
Credential $domainCredential {  
    $name = $args[0]  
    $checkpointName = $args[1]  
    Get-VM $name | Get-VMSnapshot | where Name -eq  
$checkpointName | Remove-VMSnapshot  
    if ($(Get-VMSnapshot -VMName $name -Name $checkpointName  
-ErrorAction SilentlyContinue).Count -eq 1){  
        0  
    } else {  
        1  
    }  
} -ArgumentList $args  
$return
```

Kod 5.7 Skripta za brisanje Checkpointa

Zaključak

Automatizacija donosi brojne prednosti za svaku tvrtku, posebice tvrtku koja se bavi informacijskim tehnologijama. Izvršavanje procesa se višestruko ubrzava te se smanjuje broj ponovnih izvođenja radi nenamjernih grešaka. System Center Orchestrator ima široki spektar integracije sa drugim proizvodima te podržava razvoj vlastitih načina integracije što ga čini odličnim izborom za približavanje automatizacije krajnjem korisniku. Posebice korištenjem „Orchestration Console“ web sučelja.

Smatram da će budućnost donijeti sve veću potrebu za automatizacijom i da svatko kod kreiranja nekog procesa mora imati na umu da se taj proces mora moći automatizirati. Takvim razmišljanjem i razvojem svijesti proces se često drugačije gradi od samog početka i podređen je automatizaciji što na kraju donosi bolje prihvaćanje i korisnost istog.

Popis kratica

SMS	<i>System Management Server</i>
MOM	<i>Microsoft Operations Manager</i>
SCCM	<i>System Center Configuration Manager</i>
SCOM	<i>System Center Operations Manager</i>
SCDPM	<i>System Center Data Protection Manager</i>
SCO	<i>System Center Orchestrator</i>
IP	<i>Integration Pack</i>
OIP	<i>Orchestrator Integration Pack</i>
LDAP	<i>Lightweigth Directory Access Protocol</i>
AD	<i>Active Directory</i>

Popis slika

Slika 5.1 Arhitektura sustava (izvor: https://docs.microsoft.com/en-us/system-center/orchestrator/learn-about-orchestrator?view=sc-orch-1801 preuzeto 8.1.2020.)	6
Slika 5.2 SQL Features	9
Slika 5.3 Imenovanje SQL instance	9
Slika 5.4 SQL servisni računi	10
Slika 5.5 SQL collation	10
Slika 5.6 .NET Framework 3.5 Feature	11
Slika 5.7 Poruka o instalaciji preduvjeta koji nedostaju.....	12
Slika 5.8 Završetak instalacije	13
Slika 5.9 Registracija Integration Pack-a	14
Slika 5.10 Potvrda registracije Integration Pack-a	14
Slika 5.11 Deploy Integration Pack.....	15
Slika 5.12 Odabir Integration Pack-ova za instalaciju	15
Slika 5.13 Odabir Runbook servera.....	16
Slika 5.14 Odabir opcija za instalaciju Integration Pack-ova.....	16
Slika 5.15 Potvrda prije instalacije	17
Slika 5.16 Instalacija Integration Pack-ova	17
Slika 5.17 Active Directory konfiguracija integracije.....	18
Slika 5.18 Proces kreiranja korisničkog računa.....	19
Slika 5.19 Kreiranje korisničkog računa – Initialize Data funkcija	19
Slika 5.20 Kreiranje korisničkog računa – Generate Random Text funkcija.....	20
Slika 5.21 Kreiranje korisničkog računa – Create User funkcija	20
Slika 5.22 Kreiranje korisničkog računa – Enable User funkcija.....	21
Slika 5.23 Kreiranje korisničkog računa – Append Line funkcija	21

Slika 5.24 Microsoft Exchange konfiguracija integracije	23
Slika 5.25 Proces kreiranja e-mail pretinca	24
Slika 5.26 Kreiranje e-mail pretinca – Initialize Data funkcija.....	24
Slika 5.27 Kreiranje e-mail pretinca – Enable Mailbox funkcija.....	24
Slika 5.28 Kreiranje e-mail pretinca – Create Mailbox funkcija.....	25
Slika 5.29 System Center Operations Manager konfiguracija integracije.....	27
Slika 5.30 Proces brisanja web server logova.....	27
Slika 5.31 Brisanje web server logova - Monitor Alert funkcija.....	28
Slika 5.32 Brisanje web server logova – Update Alert funkcija	29
Slika 5.33 Proces kreiranja virtualnog servera	31
Slika 5.34 Proces brisanja virtualnog servera.....	40
Slika 5.35 "Delete Folder" funkcija.....	41
Slika 5.36 Proces upravljanja Checkpointovima.....	42

Popis tablica

Popis kôdova

Kod 5.1 Skripta za brisanje logova sa web servera	29
Kod 5.2 Skripta za kreiranje virtualnog servera	32
Kod 5.3 PowerShell modul razvijen za stvaranje virtualnih servera.....	40
Kod 5.4 Skripta za brisanje virtualnog servera.....	41
Kod 5.5 Skripta za kreiranje Checkpointa.....	43
Kod 5.6 Skripta za primjenu Checkpointa	43
Kod 5.7 Skripta za brisanje Checkpointa	44

Literatura

Svaki autor piše popis literature na kraju rada. Popis literature se piše stilom literatura.

- [1] [HTTPS://SUPPORT.MICROSOFT.COM/HR-HR/LIFECYCLE/SEARCH?ALPHA=MICROSOFT%20SYSTEM%20CENTER](https://support.microsoft.com/hr-hr/lifecycle/search?alpha=microsoft%20system%20center) (PREUZETO 8.1.2020.)
- [2] [HTTPS://WWW.CAPGEMINI.COM/THE-AUTOMATION-ADVANTAGE/#](https://www.capgemini.com/the-automation-advantage/#) (PREUZETO 8.1.2020.)
- [3] [HTTPS://DOCS.MICROSOFT.COM/EN-US/SYSTEM-CENTER/ORCHESTRATOR/LEARN-ABOUT-ORCHESTRATOR?VIEW=SC-ORCH-1801](https://docs.microsoft.com/en-us/system-center/orchestrator/learn-about-orchestrator?view=sc-orch-1801) (PREUZETO 3.10.2019.)
- [4] [HTTPS://WWW.GARTNER.COM/REVIEWS/MARKET/IT-PROCESS-AUTOMATION/COMPARE/MICROSOFT-VS-VMWARE](https://www.gartner.com/reviews/market/it-process-automation/compare/microsoft-vs-vmware) (PREUZETO 17.12.2019.)
- [5] KERRIE MEYLER, PETE ZERGER, MARCUS OH, ANDERS BENGTTSSON, KURT VAN HOECKE, SYSTEM CENTER ORCHESTRATOR 2012 UNLEASHED, INDIANAPOLIS, SAMS, 2013
- [6] SAMUEL ERSKINE, ANDREAS BAUMGARTEN (MVP), STEVE BEAUMONT, MICHAEL SEIDL, MICROSOFT SYSTEM CENTER 2016 ORCHESTRATOR COOKBOOK - SECOND EDITION, PACKT, 2017

Prilog



ALGEBRA
VISOKO
UČILIŠTE

SYSTEM CENTER
ORCHESTRATOR

Pristupnik: Luka Baruškin, JMBAG

Mentor: prof. mag. sc. Vedran Dakić