VMware vRealize Orchestrator Automatizacija

Dmitrović, Danijel

Undergraduate thesis / Završni rad

2018

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:856253

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Algebra Univerity College - Repository of Algebra Univerity College

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

VMware vRealize Orchestrator Automatizacija

Danijel Dmitrović

Zagreb, srpanj 2018.

„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, 31.7.2018.

Danijel Dmitrović

Zahvala:

Zahvaljujem profesoru Vedranu Dakiću na savjetima, pomoći i trudu koji je uložio pripremajući infrastrukturu te pomažući kako bi uspješno napisao i realizirao ovaj završni rad.

Također zahvaljujem se Visokom učilištu Algebra na prilici koju sam dobio i konstantnoj podršci kroz cijeli studij bez čega ne bih bio danas ovo što jesam.

Zahvaljujem se svim drugim profesorima, asistentima i kolegama na studiju koji su mi pomogli tokom mog studiranja.

Predgovor

U grani informacijskih tehnologija ljudi su oduvijek gubili jedan dio svog vremena rješavajući ponavljajuće probleme ili radeći ponavljajuće poslove. Sada možemo zamisliti što bi se sve moglo postići kada bi ti sati koji su utrošeni ponovno i ponovno bili usmjereni na učenje i poboljšanje svakog pojedinca u organizaciji. Alati poput Microsoftovog powershell-a i System Center Orchestrator-a te VMware-ovog PowerCLI-a i vRealize Orchestrator-a samo su neki od primjera kako kvalitetno organizirati i isplanirati automatizirano izvođenje tih poslova, skidajući pritom teret sa ljudi koji to kontinuirano rade. Prednosti koje time dobivamo su nebrojene, ne samo da će zaposlenici imati više vremena za učiti i napredovati te time gurati cjelokupni posao dalje, nego će se povećati i razina zadovoljstva unutar same organizacije. VMware vRealize Orchestrator omogućava najveći nivo automatizacije te pomaže kod brzog i efikasnog ispunjenja poslovnih zahtjeva. Izvršenje svakog od workflow-a kreiranih u vRealize Orchestratoru se može zakazati za točno određeno vrijeme na točno određeni datum. Jednom kreiran workflow ne zahtjeva ljudsku interakciju sve do potrebe za određenom promjenom.

Sažetak

Virtualizacija je danas neophodna kod svake organizacije. Virtualizacija omogućava uštedu prostora koji bi u suprotnome bio iskorišten za nagomilane servere, uštedu novaca, povećanje performansi, puno lakšu administraciju i daljinsko upravljanje. Veće organizacije kreiraju nove virtualne mašine na dnevnoj ili tjednoj bazi, uzimaju snapshote, podižu domene i laboratorije u virtualnoj okolini. Svaki od tih zadataka može biti u potpunosti automatiziran. VMware vRealize Orchestrator je VMware-ov produkt koji je proizveden za tu svrhu. On također omogućava korištenje već postojećih skripti (bash ili powershell) za automatizaciju i orkestraciju na smislen način koji zadovoljava potrebe poslovanja. Vrlo često to nalazimo u cloud okolinama gdje se provizioniraju serveri i aplikacije. Omogućena je integracija orchestratora sa postojećom vSphere okolinom. Svaki zadatak koji automatiziramo se naziva workflow (zadatak koji ima tijek odnosno trajanje). Ti zadaci se mogu spajati i kombinirati na način da jedan zadatak može uključivati njih više. Orchestrator uključuje već postojeću kolekciju zadataka koje možemo iskoristiti ili prilagoditi po potrebi. Također, omogućeno je skriptiranje unutar Orchestratora koristeći JavaScript. U praktičnom djelu ovog rada biti će prikazan automatizirani zadatak koji će u sebi uključivati druge zadatke te powershell skripte. Konačni rezultat tog zadatka biti će funkcionalna Active Directory infrastruktura sa 4 servera od kojih je jedan Domain Controller, a ostala 3 su serveri članovi domene. Sve to sa samo jednim klikom na workflow.

Abstract

Virtualization is indispensable today in every organization. Virtualization saves space that would otherwise been used for the accumulated servers, saving money, increasing performance, easier administration and remote control. Larger organizations create new virtual machines on a daily or weekly basis, taking snapshots, raise domain and laboratories in a virtual environment. Each of these tasks can be fully automated. VMware vRealize Orchestrator is a VMware product that is manufactured for the purpose of automation. It also allows the use of already existing scripts (bash or powershell) and their orchestration in a meaningful way that meets business needs. That approach is very often found in cloud environments where servers and apps are being provisioned. Orchestrator allows integration with the existing vSphere environment. Each task that is automated is called a workflow (a task that has a flow and duration). These workflows can be merged and combined in a way that one workflow can involve one or more other workflows. Orchestrator includes an existing collection of workflows that we can use or adjust as needed. Also, scripting within the Orchestrator is done by using JavaScript language. In the practical part of this work a workflow will be created that will involve other workflows as well as powershell scripts. The final result of this workflow will be a functional Active Directory infrastructure with 4 servers, one of which is a Domain Controller, and the other 3 servers are domain members. All this achieved with just one click on the workflow.

Sadržaj

1. Uvod 1

2. Upoznavanje sa VMware Orchestratorom 2

2.1. Što je VMware Orchestrator 2

2.2. Detaljna arhitektura VMware Orchestrator-a 3

3. Instalacija i konfiguracija vRealize Orchestrator-a 5

3.1. Zahtjevi za instalaciju 5

3.2. Mrežne postavke 5

3.3. Instalacija i konfiguracija VMware orchestrator appliance-a 7

4. Integracija vRealize Orchestrator-a sa postojećom vSphere infrastrukturom 15

4.1. Zašto i kako integrirati vSphere i Orchestrator? 15

4.2. Zašto i kako povezati orkstrator sa različitim vrstama infrastruktura? 18

5. Automatizacija posla putem workflow-a 20

5.1. Napredno korištenje orkestrator klijenta 20

5.2. Što su Workflowi? 21

5.2.1. General tab 23

5.2.2. Inputs tab 24

5.2.3. Outputs tab 25

5.2.4. Presentation tab 25

5.2.5. Parameters References tab 26

5.2.6. Workflows Tokens tab 26

5.2.7. Events tab 26

5.2.8. Permissions tab 27

5.3. Što su Akcije? 27

6. Povezivanje više workflow-a u kompleksnije workflowe 29
6.1. Template 29
6.2. Domain Controller 30
6.2.1. Dodavanje nove mape 30
6.2.2. Linked_Clone_Creation workflow 31
6.2.3. Domain_Controller_Script workflow 32
6.2.4. Domain_Controller_Creation workflow 32
6.2.4.1 Linked_Clone_Creation parametri 33
6.2.4.2 Sleep parametar 36
6.2.4.3 Domain_Controller_Script parametri 37
6.3. Three Member Servers 42
6.4. The_Workflow_Of_Genesis 47
6.5. Proces automatizacije 48
7. vRO skriptiranje koristeći JavaScript 53
7.1. JavaScript kao jezgra Orchestrator-a 53
7.2. Primjer skriptabilnog zadatka 54
Zaključak 59
Popis kratica 60
Popis slika 61
Popis tablica 64
Popis kôdova 65
Literatura 66

1. Uvod

Ovaj završni radi objasniti će u detalje što je to VMware vRealize Orchestrator, njegovu instalaciju, konfiguraciju i integraciju sa vSphere okolinom. Biti će objašnjeno kako u pozadini Orchestrator radi, koje plugine koristi i njegov izgled te grafičko sučelje. Na praktičnom primjeru biti će u detalje objašnjeno što je to workflow, kako se planira, dizajnira i kreira. Praktični zadatak koji će se u ovom radu automatizirati biti će kreiranje potpuno nove, funkcionalne Active Directory infrastrukture koja se sastoji od jednog Domain Controller-a te dva servera člana domene. Svaki server će imati konfigurirano odgovarajuće ime, IP adresu i DNS kao i svoju ulogu. Nakon pokretanja tog workflow-a ljudska interakcija neće biti potrebna. Nakon završetka workflowa novokreirana infrastruktura može se koristiti u svrhu laboratorija ili produkcije. Također, biti će demonstrirane i objašnjene osnove JavaScript skriptiranja unutar orkestratora.

2. Upoznavanje sa VMware Orchestratorom

U ovom poglavlju biti će riječi o tome čemu Orchestrator služi, kako se uklapa u VMware okolinu i kako ga koristiti.

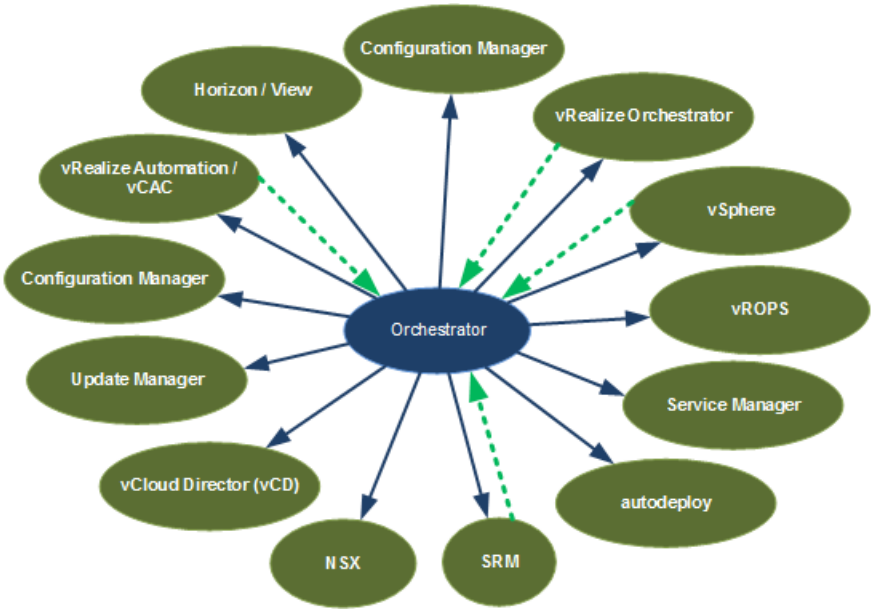
Teme ovog poglavlja su:

- Što je VMware Orchestrator?
- Arhitektura VMware Orchestrator-a

2.1. Što je VMware Orchestrator

VMware vRealize Orchestrator je zadnja iteracija orkestratora proizvedena od VMware kompanije. Orchestrator se rodio 2007 godine od strane male kompanije nazvane Dunes. VMware je tada kupio Dunes, a time i orkestrator i preimenovao ga je u VMware Orchestrator, a zatim u vCenter Orchestrator. Mnogo administratora danas i dalje u potpunosti ne razumije svrhu orkestratora ili podcjenjuje njegovu moć. Ovaj rad će prikazati većinu toga što orkestrator može, a sve ostalo je stvar mašte i kreativnosti osobe koja ga koristi. Orkestrator je započeo svoj život kao Windows instalacija no dolaskom vSphere-a 6 VMware ga je odlučio reinkarnirati kao Linux bazirani appliance. Orkestrator je u suštini vizualni alat za skriptiranje koji u pozadini koristi JavaScript. Svaki workflow kreiran u orkestratoru sastoji se od blokova na koje možemo gledati kao na lego kocke. Orkestrator koristi plug-in arhitekturu kako bi proširio svoje korištenje na razne platforme (HP, Cisco, EMC, NetApp pa čak i integraciju sa Microsoftovim SCOM-om i SCVMM-om). Slika 1. prikazuje sve plug-ine koje orkestrator može koristiti. [1]

[1] D. Langenhan; VMware vRealize Orchestrator Essentials; September 24, 2015; 9781785884245



Slika 2.1 VMware orchestrator plug-ins, D. Langenhan; VMware vRealize Orchestrator Essentials; September 24, 2015; 9781785884245

Matrica kompatibilnosti prikazana u Tablica 2.1, pokazuje koja verzija orkestratora je kompatibilna sa kojom verzijom vSphere-a odnosno vCentra. Najbolje prakse kažu da verzija orkestratora treba pratiti verziju vSphere-a i vCentra.

Tablica 2.1 Matrica kompatibilnosti

vSphere	**vCenter**	**Orchestrator**
6.5 U1	6.5 U1	7.3.0
6.0	6.0	6.0.2
5.5	5.5 U2e	5.5.2.1
5.0	5.1 U3a	5.1.3.1

2.2. Detaljna arhitektura VMware Orchestrator-a

Arhitektura predstavlja sve one čimbenike koji sačinjavaju orkestrator. Kako bi uspješno implementirali orkestrator potrebno je imati novu ili postojeću vsphere infrastrukturu. VMware-ov hipervizor se naziva ESXi, te je on zaslužan za dodjeljivanje resursa virtualnim

mašinama te njihovo postojanje. Kako bi uveli mnoštvo drugih opcija, te samim time i povezali orkestrator sa vmware infrastrukturom, potreban nam je vCenter. On služi za upravljanje esxi hostovima, diskovima, mrežom, autentikacijom te se na njega spaja orkestrator. VMware ima ugrađen SSO (Single Sign On) mehanizam koji je uvelike koristan i za orkestrator budući da je jedan login u vCenter kao autorizirani korisnik (bilo kroz AD ili LDAP) dovoljan da se automatski autenticiramo i na orkestrator. Orkestrator zahtjeva i bazu podataka pa tako po defaultu koristi PostgreSQL no može se spojiti i na Microsoft SQL Server.

Orkestrator ima i svoje limitacije koje su navedene u Tablica 2.2. Limitacije orkestratora

Tablica 2.2. Limitacije orkestratora

Maksimalni broj istovremeno spojenih vCenter servera	20
Maksimalni broj istovremeno spojenih ESXi hostova	1280
Maksimalni broj istovremeno pokrenutih VM-ova	3500
Maksimalni broj istovremeno pokrenutih workflow-a	300

3. Instalacija i konfiguracija vRealize Orchestrator-a

VMware orchestrator je Linux appliance koji se instalira kao virtualna mašina na jedan od ESXi hostova te je već konfiguriran unaprijed. Konfiguracija koja je potrebna sa strane korisnika je instalacija appliance-a, davanje imena i IP adrese orkestratoru te povezivanje i autenticiranje na vCenter gdje će orkestrator živjeti i obavljati sve svoje poslove.

3.1. Zahtjevi za instalaciju

Kako bi uspješno implementirali VMware orchestrator moraju biti ispunjeni zahtjevi iz Tablica 3.1 Zahtjevi orkestratora.

Tablica 3.1 Zahtjevi orkestratora

CPU	2 vCPU-a 2.0 GHz
Memory	6 GB
Disk Space	17 GB
Network	1x Network 1x IP (DHCP omogućen)

Internetski preglednici koje orkestrator podržava su:

- Microsoft Internet Explorer 10 ili noviji
- Mozilla Firefox
- Google Chrome

3.2. Mrežne postavke

Kako bi se orkestratoru moglo pristupiti sa mreže te kako bi se na mreži mogao registrirati njegov naziv, potrebno je postaviti IP adresu te ime orkestratora koje će onda biti registrirano u DNS-u. Orkestrator ima mogućnost dobivanja adrese putem DHCP-a ukoliko ne želimo statički konfigurirati IP adresu. Sa DNS servera ili bilo kojeg servera koji ima isti DNS server

kao i orkestrator možemo provjeriti da li je, nakon mrežne konfiguracije, orkestrator dostupan te možemo li ga pronaći putem njegovog imena.

Tablica 3.2 Komande za provjeru

Tip DNS pretrage	**Komanda**	**Rezultat**
Puni	Ping orchestratortestdd.vmlab.local	172.16.140.131
Kratki	Ping orchestratortestdd	172.16.140.131
Reverzni	Ping -a 172.16.140.131	Orchestratortestdd.vmlab.local

Portovi za pristup orkestratoru putem web sučelja internetskog preglednika su navedeni u Tablica 3.3 Portovi za pristup orkestratoru.

Tablica 3.3 Portovi za pristup orkestratoru

Port	**Broj**	**Protokol**	**Opis**
Virtual Appliance Management Interface	5480	TCP	Pristupni port do orkestratorovog sistemskog sučelja
HTTP server port	8280	TCP	Svi zahtjevi poslani orkestratoru na HTTP port 8280 su automatski redirektani na HTTPS port 8281
HTTPS server port	8281	TCP	Početna stranica orkestratora
Konfiguracijski HTTPS port	8283	TCP	SSL port za konfiguraciju postavki orkestratora putem web sučelja

Portovi koje orkestrator koristi za spajanje na servise koji su mu potrebni za rad su navedeni u Tablica 3.4 Orkestratorovi eksterni portovi

Tablica 3.4 Orkestratorovi eksterni portovi

Port	**Broj**	**Protokol**	**Opis**
SQL Server	1433	TCP	Port putem kojeg se orkestrator spaja na eksternu Microsoft SQL bazu ukoliko je konfiguriran da istu koristi.
PostgreSQL	5432	TCP	Port putem kojeg se orkestrator spaja na PostgreSQL bazu ukoliko je konfiguriran da istu koristi.
Oracle	1521	TCP	Port putem kojeg se orkestrator spaja na eksternu Oracle bazu ukoliko je konfiguriran da istu koristi.
SMTP server port	25	TCP	Port koji orkestrator koristi kako bi putem mail sustava slao notifikacije
vCenter Server API port	443	TCP	Port koji orkestrator koristi kako bi od postojećeg vCentra dobio informacije o infrastrukturi, hostovima i virtualnim mašinama.

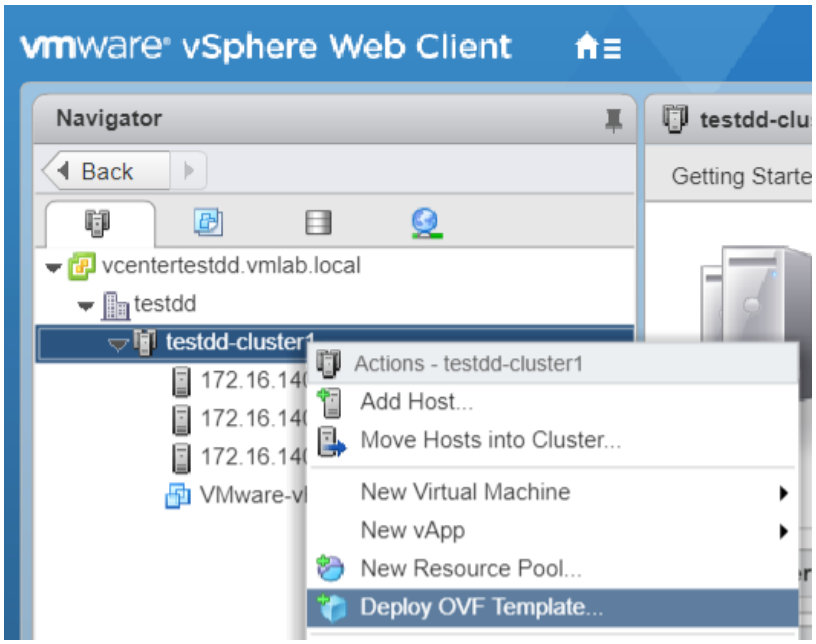
3.3. Instalacija i konfiguracija VMware orchestrator appliance-a

Prilikom preuzimanja orkestratora sa VMware-ovih internetskih stranica možemo uočiti da je isti u .OVA formatu. Taj format predstavlja predložak koji se putem vCentra može jednostavno konvertirati u virtualnu mašinu.

Postupak konvertiranja orkestrator predloška u virtualnu mašinu[2]:

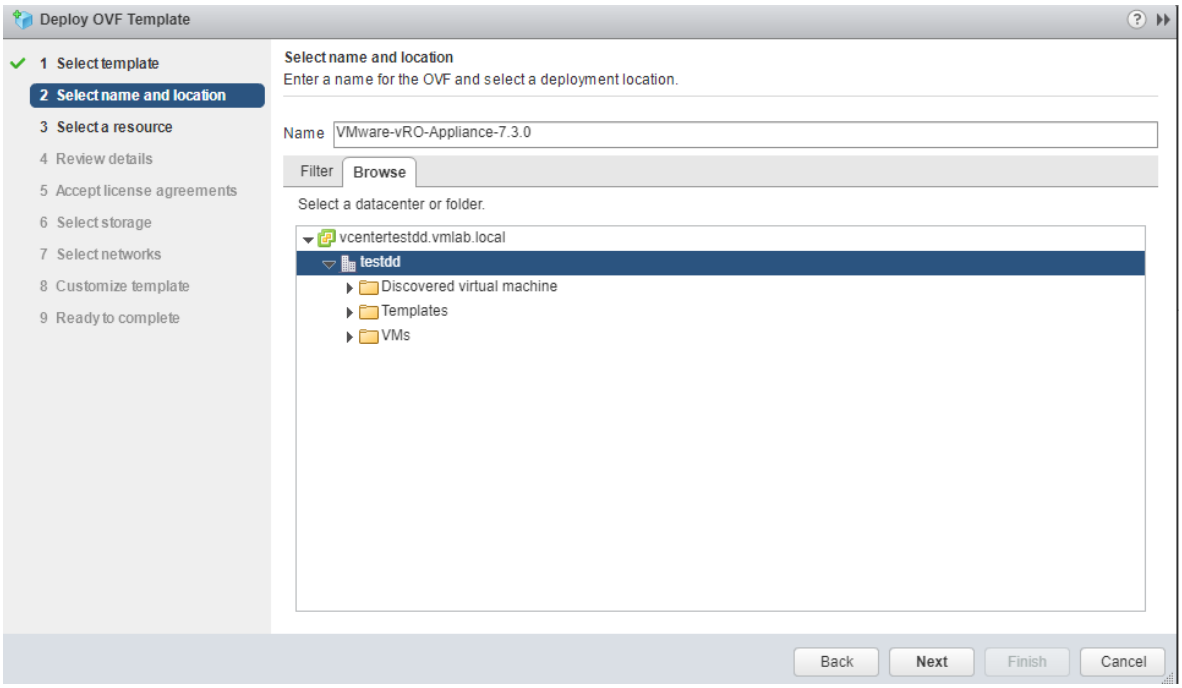
[2] D. Langenhan; VMware vRealize Orchestrator Cookbook - Second Edition; November 11, 2016; 9781786462787

- Spajanjem na vCenter te desnim klikom na host (ili cluster) na koji želimo instalirati orkestrator virtualnu mašinu, otvara se prozor koji sadrži opciju Deploy OVF template.

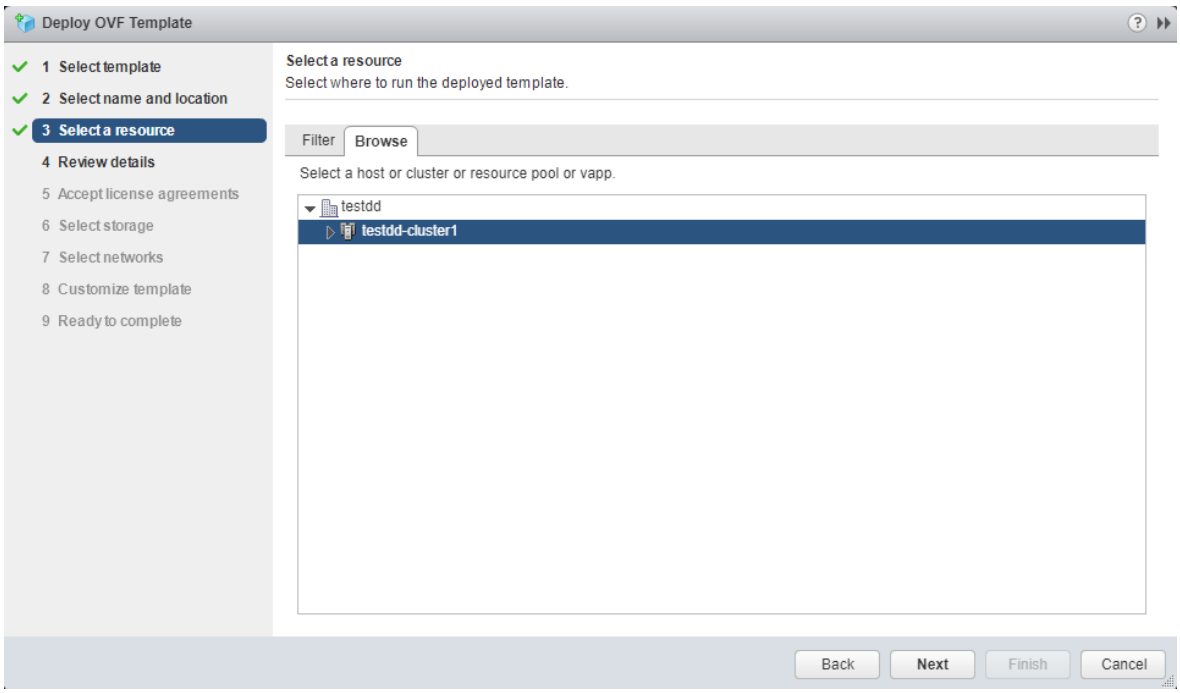


Slika 3.1 Deployment OVF Template-a

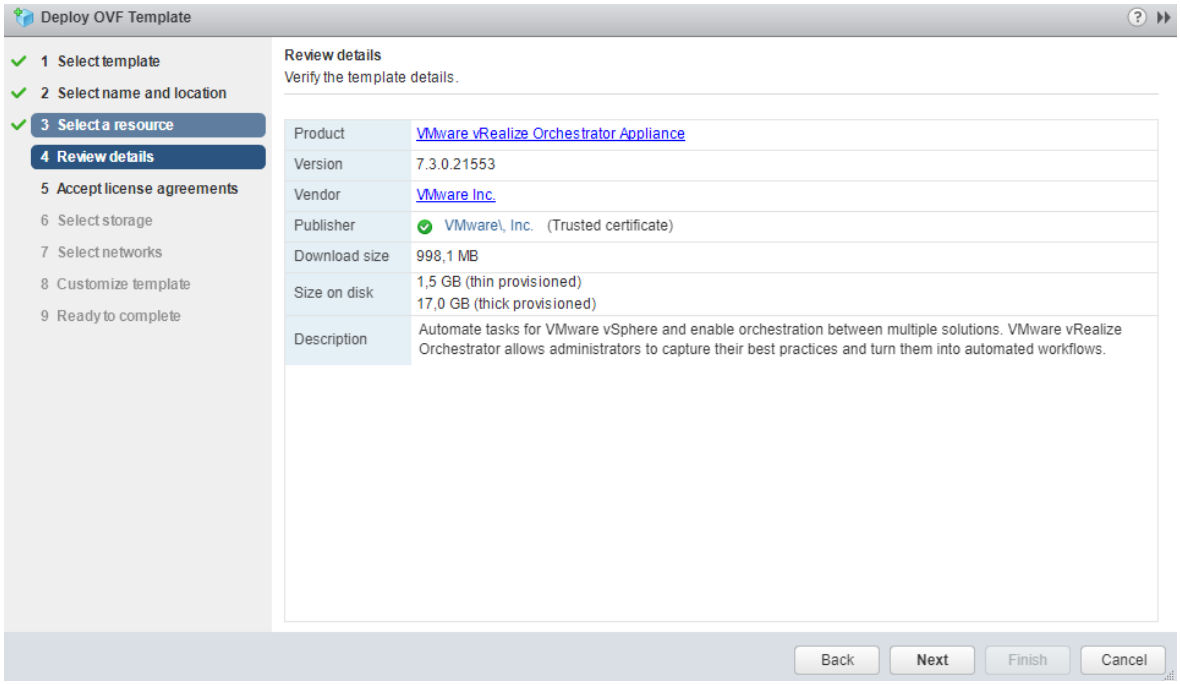
- Slijedi konfiguracija potrebnih postavki kako bi se iz .OVA predloška mogla kreirati virtualna mašina na odabranom hostu. Te postavke uključuju putanju do .OVA predloška kojeg smo preuzeli, ime virtualne mašine, disk odnosno datastore gdje će se virtualna mašina spremiti, specifikacije virtualne mašine kao što su broj virtualnih procesora, memorija, veličina diska i mrežna konfiguracija.



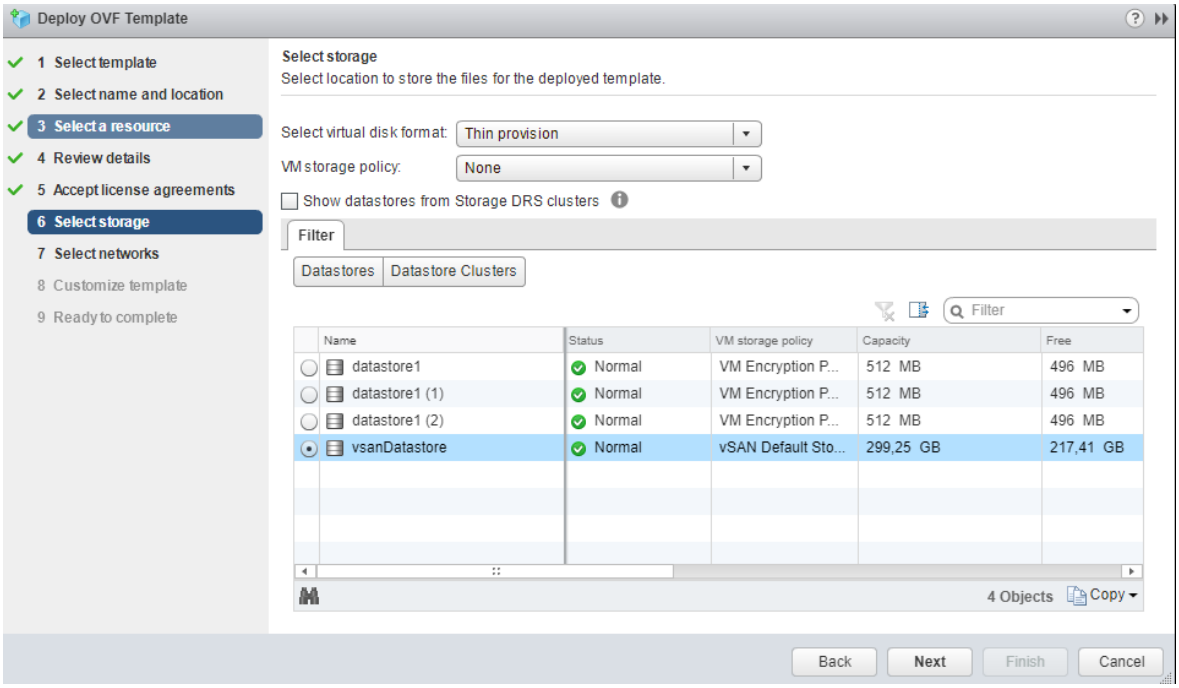
Slika 3.2 Odabir lokacije Deploymenta



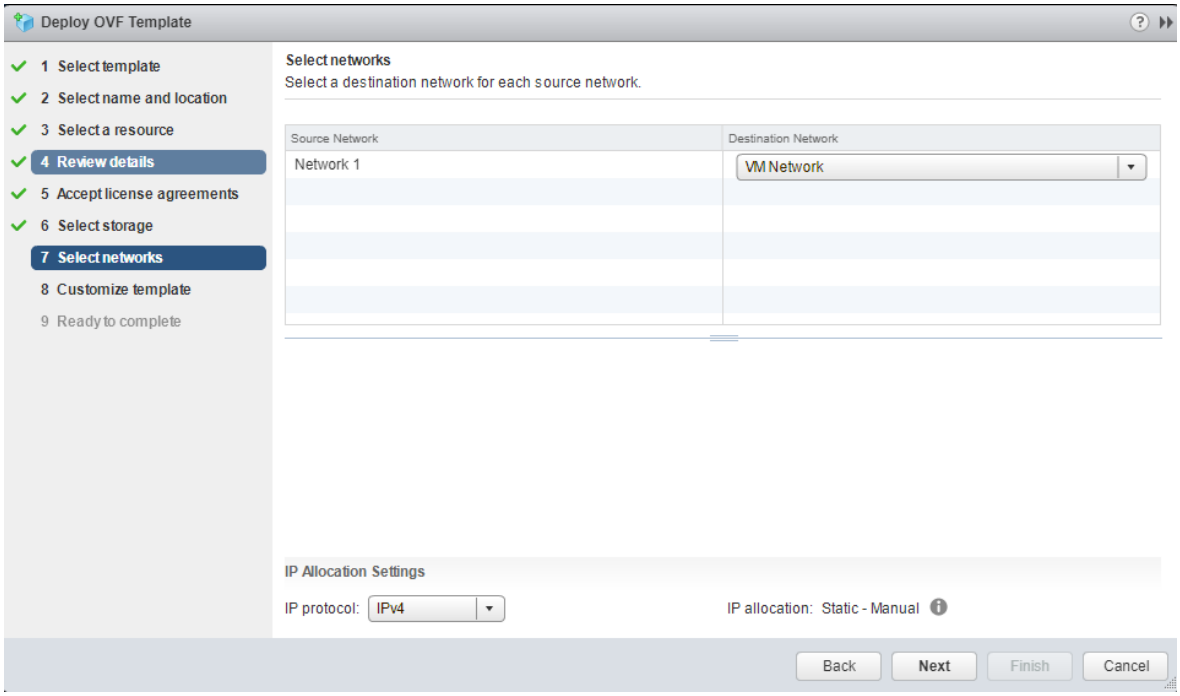
Slika 3.3 Odabir fizičkog ESXi hosta ili cluster-a



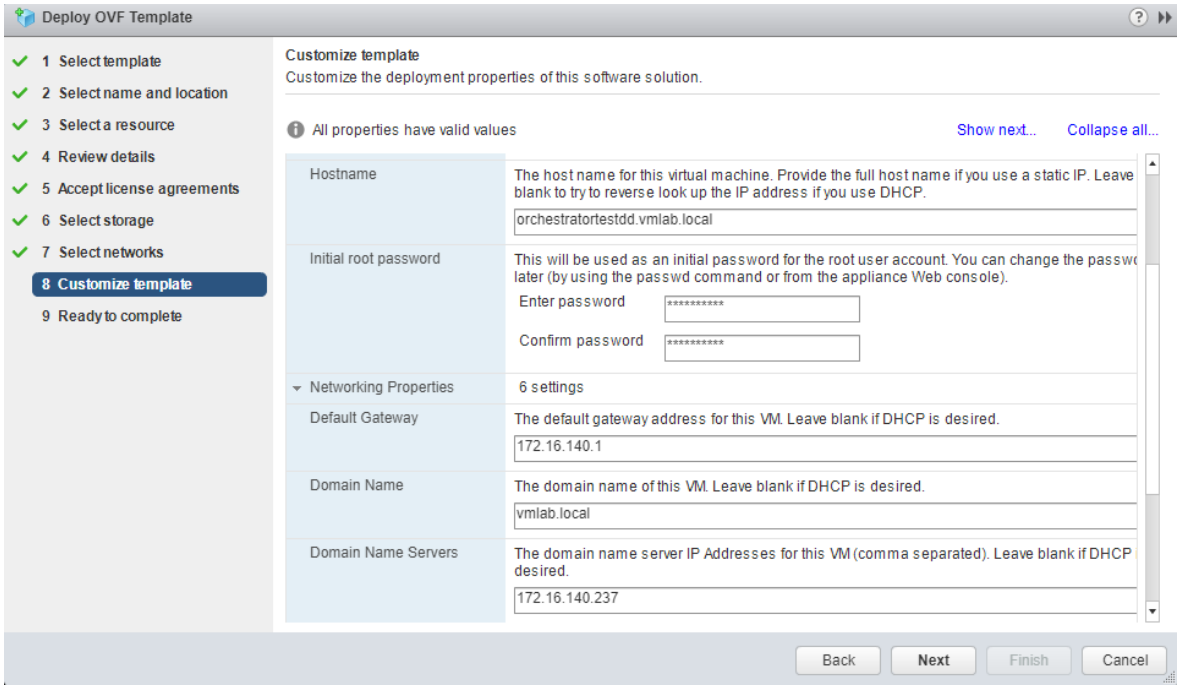
Slika 3.4 Sumarni prikaz detalja



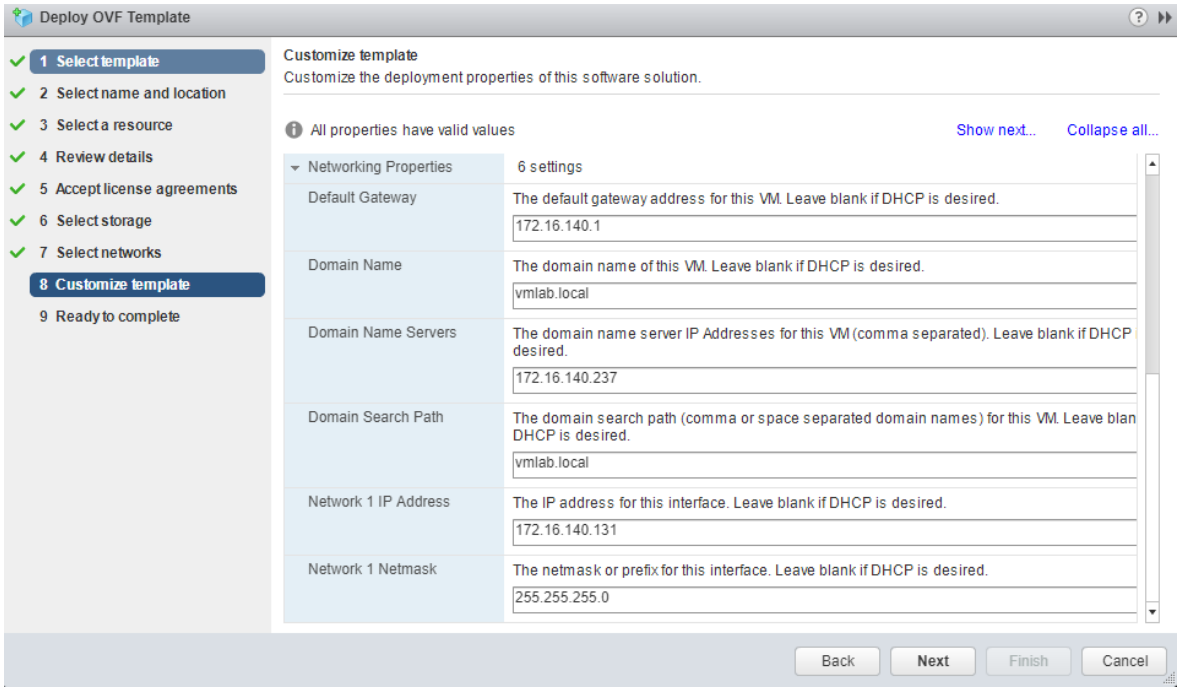
Slika 3.5 Odabir datastore-a



Slika 3.6 Odabir mreže



Slika 3.7 Postavljanje hostname-a, gateway-a i DNS-a



Slika 3.8 Postavljanje DNS suffixa i IP adrese orkestratora

- Nakon što završimo inicijalno postavljenje, odraditi će se instalacija orkestrator virtualne mašine. Virtualnu mašinu naziva VMware-vRO-Appliance-7.3.0 moći ćemo vidjeti na popisu virtualnih mašina u vCentru. Kako bi provjerili da li je virtualna mašina zaista dostupna, možemo to odraditi ping komandom na njeno ime ili na IP adresu koju smo zadali.

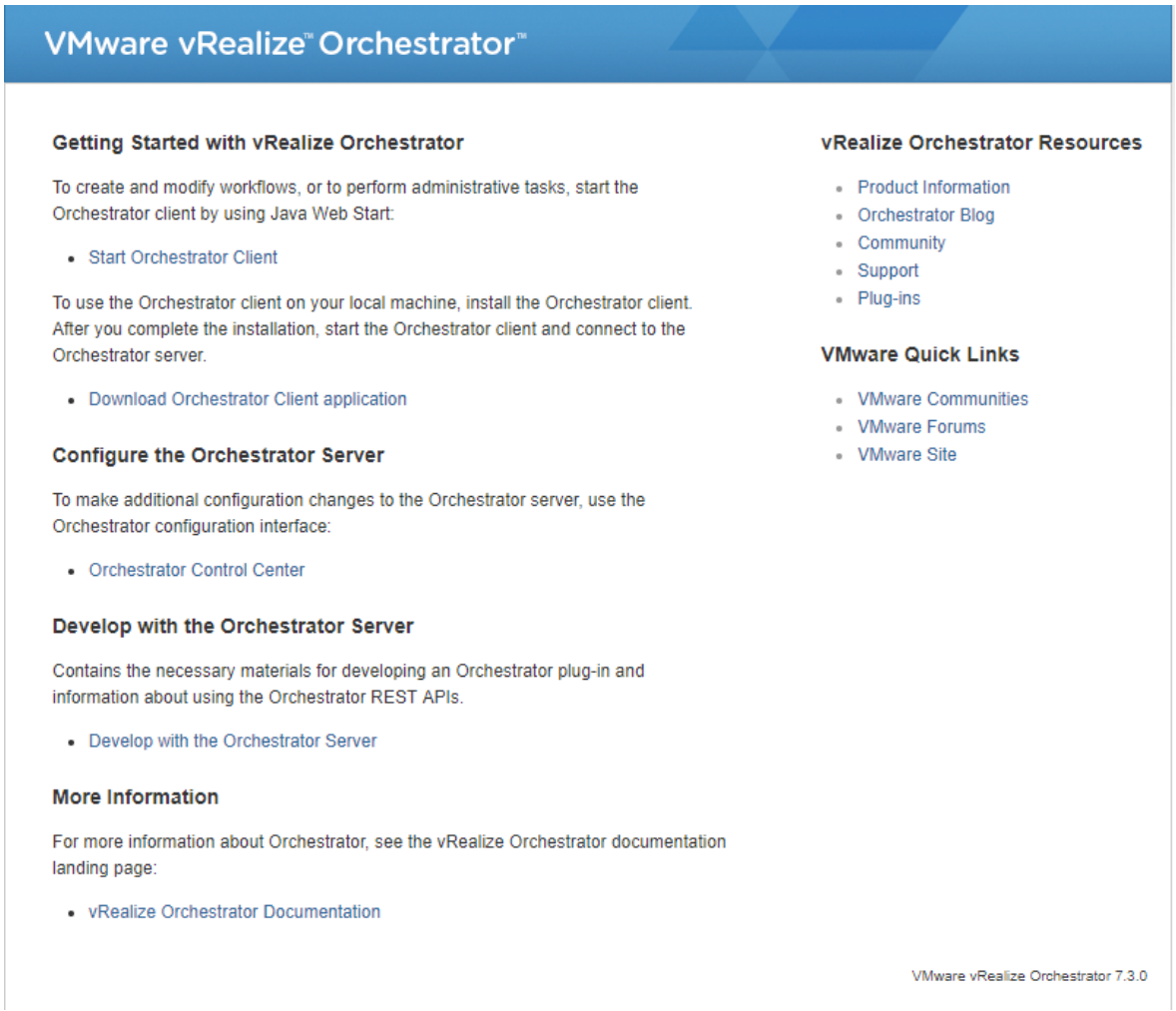
```
PS C:\Windows\system32> ping orchestratortestdd.vmlab.local

Pinging orchestratortestdd.vmlab.local [172.16.140.131] with 32 bytes of data:
Reply from 172.16.140.131: bytes=32 time=81ms TTL=60
Reply from 172.16.140.131: bytes=32 time=58ms TTL=60
Reply from 172.16.140.131: bytes=32 time=62ms TTL=60
Reply from 172.16.140.131: bytes=32 time=58ms TTL=60

Ping statistics for 172.16.140.131:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 58ms, Maximum = 81ms, Average = 64ms
```

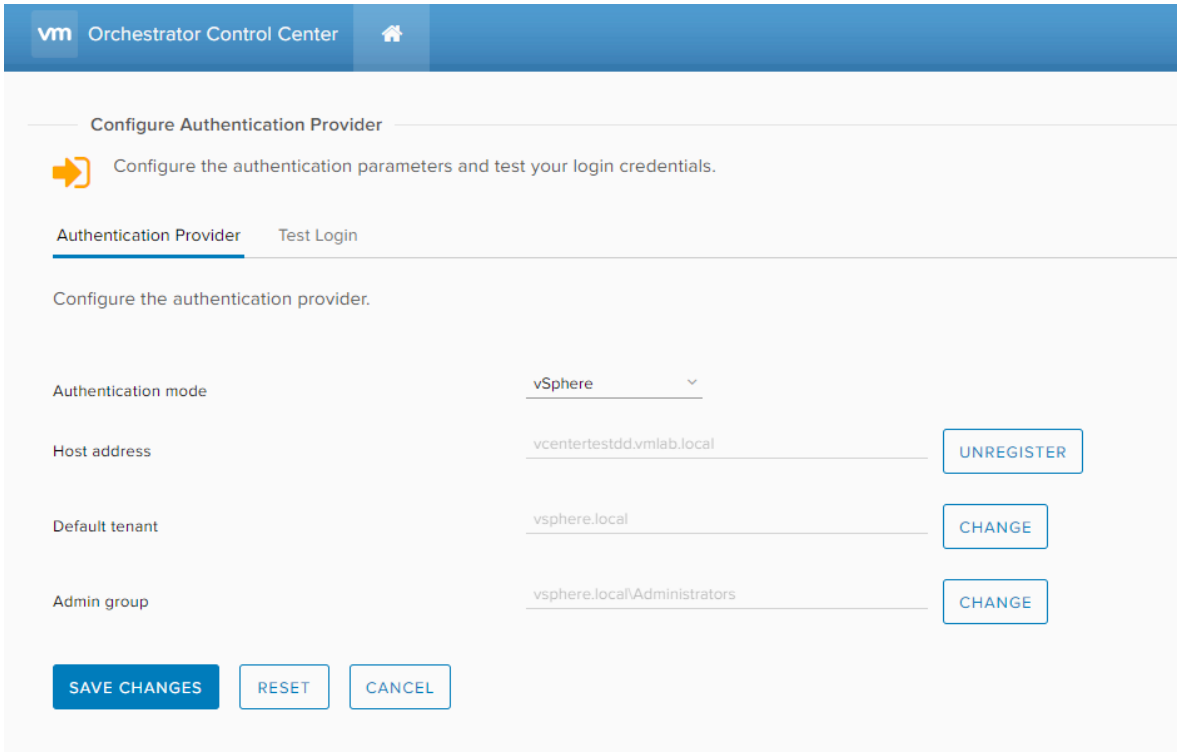
Slika 3.9 Provjera dostupnosti virtualne mašine

- Sada putem web sučelja možemo pristupiti orkestratoru. U web preglednik upišemo ime ili IP adresu orkestratora i učitati će se stranica prikazana na Slika 3.10 Orkestrator web sučelje



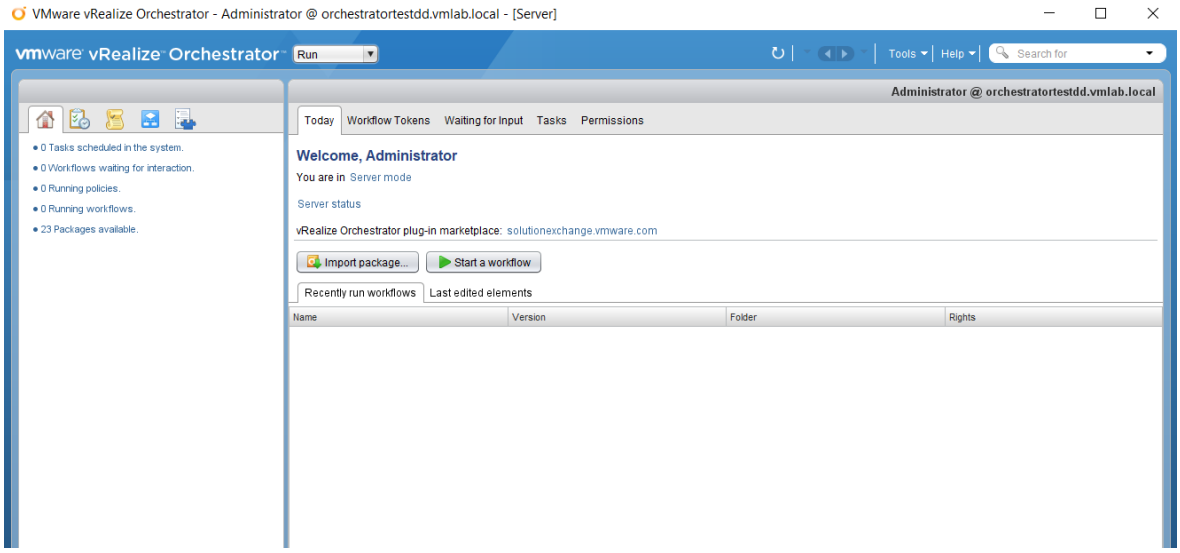
Slika 3.10 Orkestrator web sučelje

- Prije nego pokrenemo orkestrator klijent, moramo konfigurirati autentikaciju sa vSphere-om. To ćemo odraditi preko Orchestrator Control Center-a. Klikom na Orchestrator Control Center otvara se prozor za konfiguraciju Authentication Provider-a. Tu upisujemo postavke našeg vCenter server-a, te se promjene spremaju nakon uspješne autentikacije.



Slika 3.11 Konfiguracija SSO autentikacije

- Uspostavom ove autentikacije (SSO) smo omogućili da se korisničko ime i lozinka s kojom se autenticiramo u vCentar, koristi za autentikaciju i u orkestrator. Povratkom na početni izbornik odabiremo pokretanje orkestrator klijenta. Preuzima se java program kojeg otvaramo, upisujemo korisničko ime i lozinku te dobivamo korisničko sučelje orkestrator-a.



Slika 3.12 Korisničko sučelje orkestratora

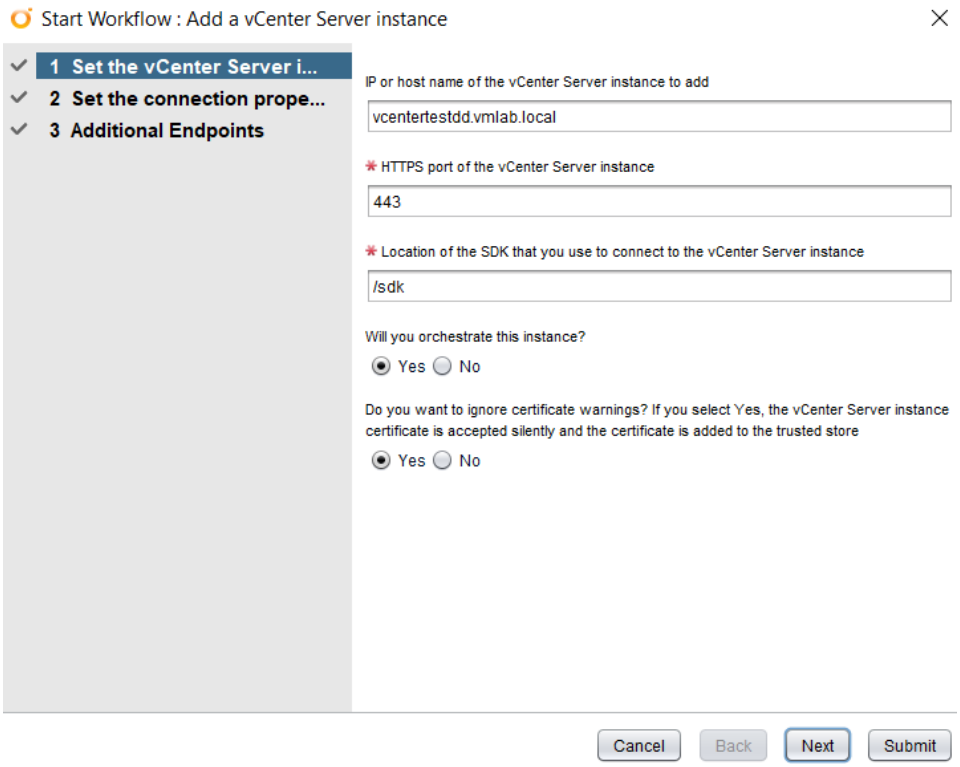
4. Integracija vRealize Orchestrator-a sa postojećom vSphere infrastrukturom

Prije nego krenemo dizajnirati i pokretati prve workflowe, potrebno je orkestratoru reći nad kojim vCentrom (te njegovim pripadajućim resursima) ćemo te workflowe izvršavati.

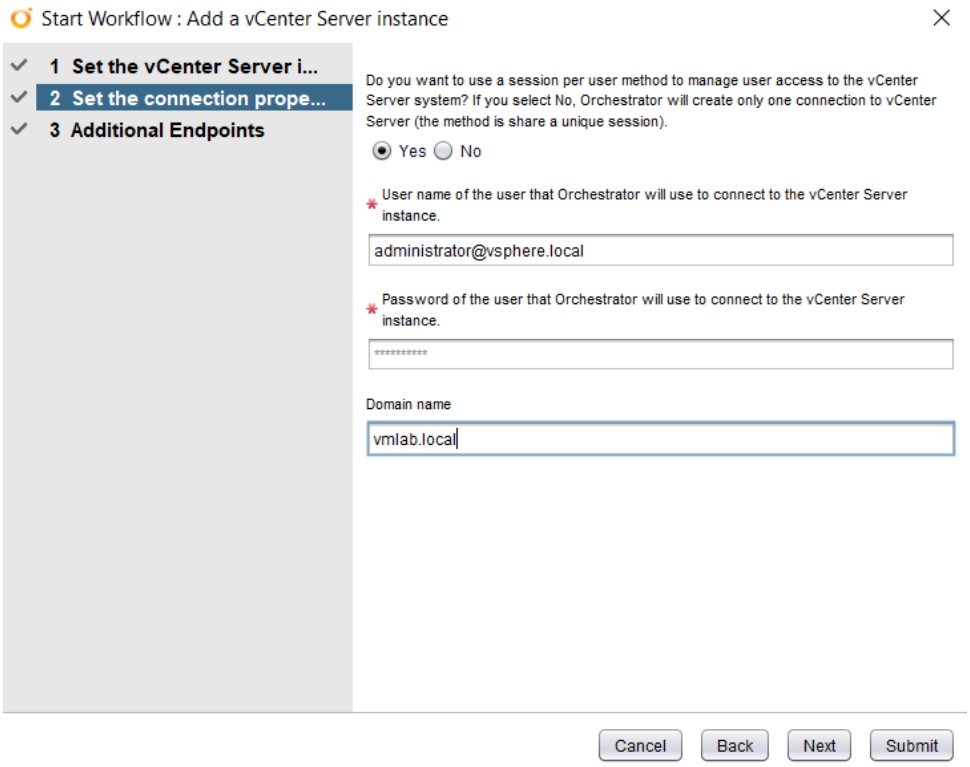
4.1. Zašto i kako integrirati vSphere i Orchestrator?

Kao i prethodni zadaci unutar orkestratora i ovaj se izvršava putem ugrađenog workflow-a.

1. Pozicioniramo se na **Library | vCenter | Configuration**
2. Desni klik na workflow naziva **Add a vCenter Server instance** i odabiremo **Start workflow**
3. Unosimo fqdn vCenter servera, port 443, odabiremo, sdk lokaciju ostavljamo defaultnu, te odabiremo **Yes** na pitanje želimo li orkestrirati ovu instancu i želimo li ignorirati upozorenja za certifikate.
4. U sljedećem koraku upisujemo kredencijale za vCenter korisnika te završavamo workflow

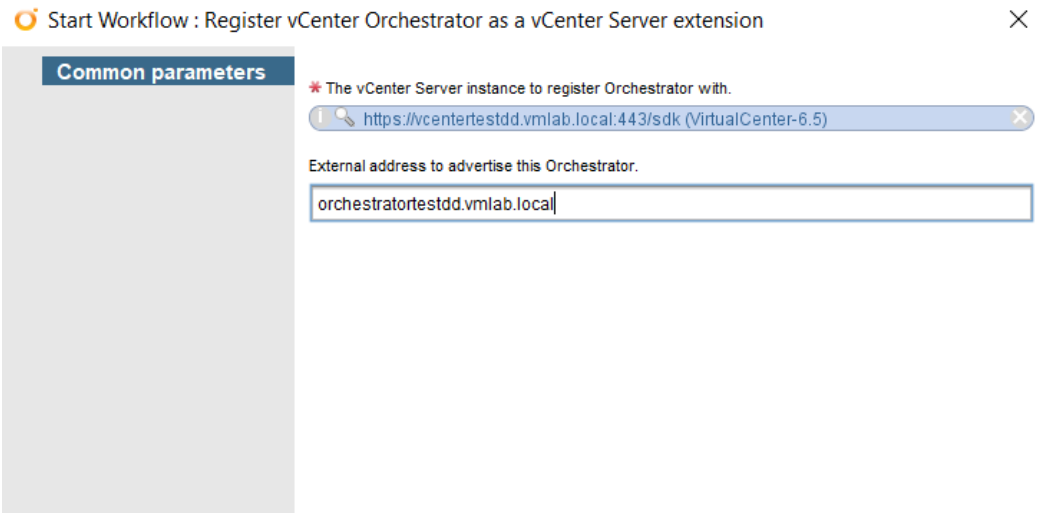


Slika 4.1 Postavke vCentra

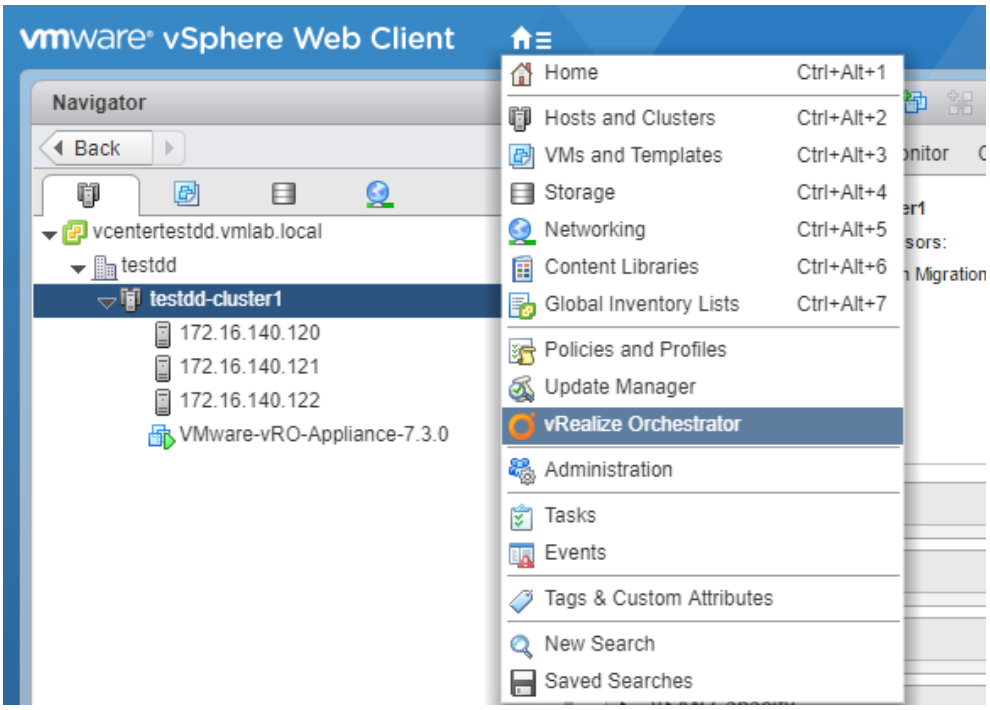


Slika 4.2 Postavke vCenter korisnika

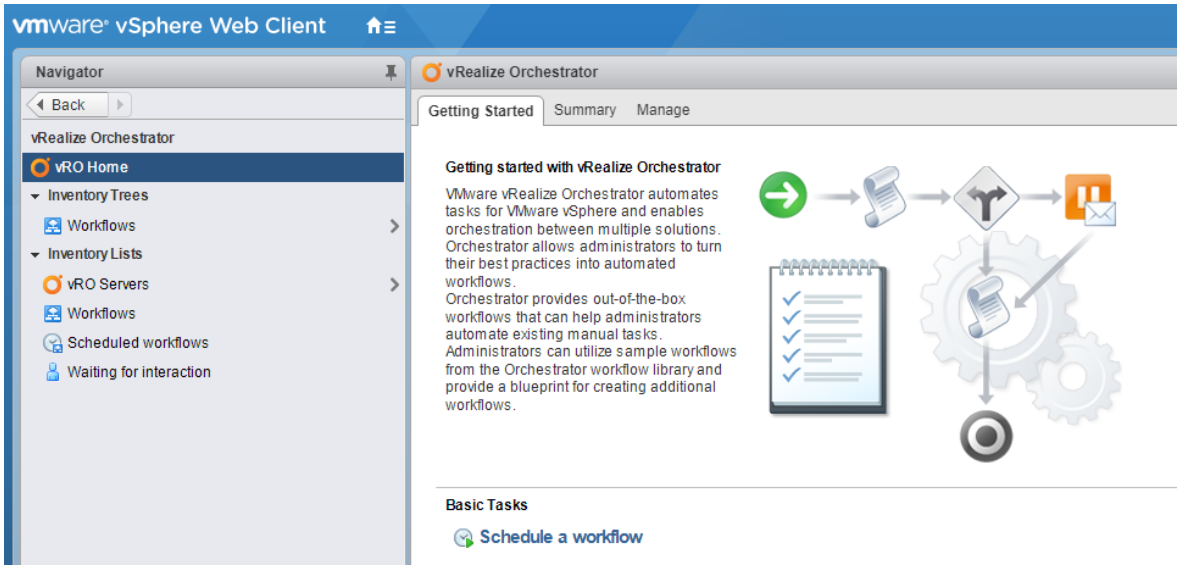
5. Kako bi iz samog vCentra mogli pristupati orkestratoru, moramo ga dodati kao ekstenziju u vCentar.
6. Kako bi to napravili pokrećemo workflow naziva **Register vCenter Orchestrator as a vCenter Server extension** na lokaciji **Library | vCenter | Configuration**
7. Odabiremo vCenter instancu i upisujemo ime orkestratora koje će se pokazivati unutar vCentra



Slika 4.3 Konfiguracija orkestratora kao ekstenzije



Slika 4.4 Pristup orkestratoru iz vCentra



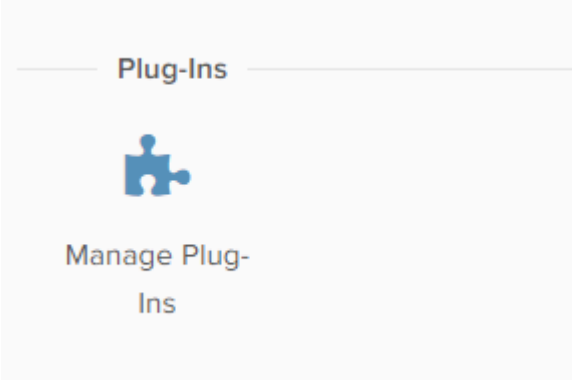
Slika 4.5 Orkestrator integriran u vCenter

4.2. Zašto i kako povezati orkstrator sa različitim vrstama infrastruktura?

Postoje razni pluginovi koji se mogu instalirati u orkestrator kako bi se omogućila povezanost i upravljanje drugim sustavima i tehnologijama. Pluginovi se mogu preuzeti sa VMware internetskih stranica.

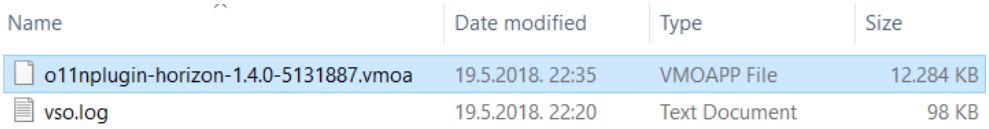
Za primjer ćemo instalirati **VMware Horizon vRealize Orchestrator Plug-in**.

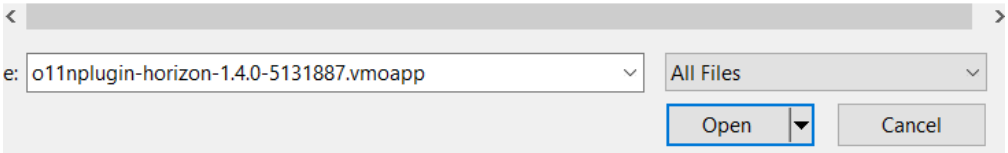
Kako bi instalirali preuzeti plugin u orkestrator, moramo ući u **Orchestrator Control Center**. Unutar **Control Centra**, nalazi se opcija **Manage Plug-Ins**.



Slika 4.6 Upravljanje pluginovima

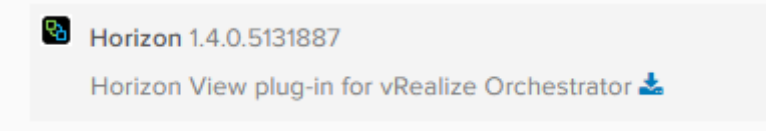
Klikom na **Browse**, odabiremo preuzeti plugin





Odabiremo opciju **Install**, prihvaćamo License Agreement te pričekamo instalaciju plugina.

Plugin će se sada pokazati na listi instaliranih pluginova te će po defaultu biti uključen.



Slika 4.7 Uspješno instaliran Horizon plugin

Instalacijom ovog plugina omogućili smo povezivanje, upravljanje, konfiguraciju i automatizaciju procesa vezanih uz VMware Horizon infrastrukturu.

5. Automatizacija posla putem workflow-a

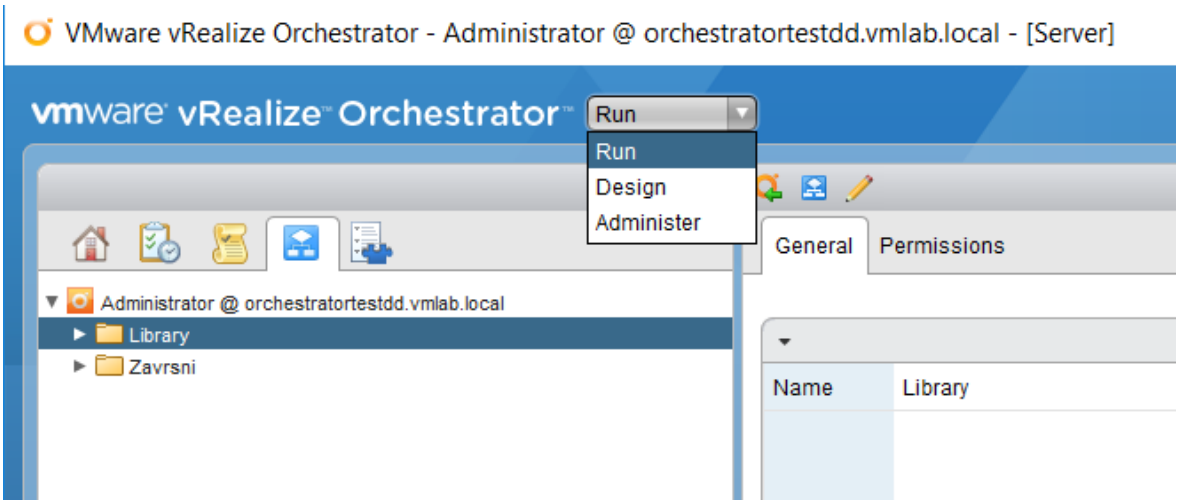
Nakon što smo uspješno instalirali, konfigurirali i integrirali orkestrator sa vSphere infrastrukturom, možemo ga početi koristiti. Posao možemo automatizirati korištenjem workflow-a. Workflow je skup akcija koje orkestrator izvršava a koje skupa djeluju prema cilju kojeg želimo automatizirati. Workflow se može sastojati od više drugih workflow-a koji zajedno čine harmoniju. Orkestrator ovakvim pristupom otvara novi pogled i novi način razmišljanja kod sistem administratora.

5.1. Napredno korištenje orkestrator klijenta

Pristup orkestratoru vrši se u tri koraka.

1. Upisujemo FQDN ili IP adresu u web preglednik.
2. Preuzimamo java bazirani orkestrator klijent na računalo
3. Otvaramo klijent i logiramo se kao domenski ili vSphere korisnik

Kada prvi puta otvorimo orkestrator klijent, biti će postavljen u **Run** način rada kako prikazuje i Slika 5.1 Početni izgled orkestratora



Slika 5.1 Početni izgled orkestratora

Svaki od navedena tri načina rada (pogleda) predstavlja svoju fazu u kreiranju efikasnog workflowa. Najviše vremena kod kreiranja workflowa se provodi u **Design** pogledu jer se

tamo vrši dizajniranje svakog pojedinog elementa, svake akcije, svakog podworkflowa, njihove međusobne komunikacije i interakcije.

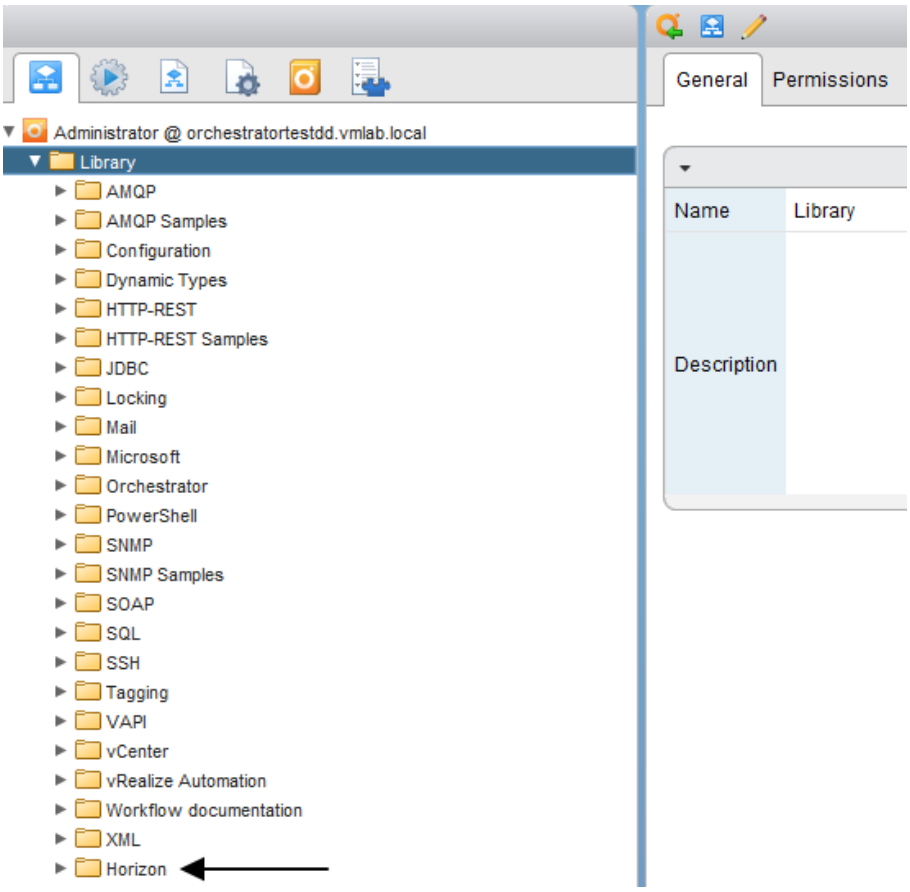
Nakon odabira **Design** pogleda, sa lijeve strane vidimo niz ikona.

Slika 5.2 Ikone u Design pogledu

1. Workflows – Unutar ovog tab-a vrši se upravljanje svega što je povezano sa workflowima.
2. Actions – Unutar ovog tab-a vrši se upravljanje svega što je povezano sa akcijama. Akcije u orkestratoru predstavljaju manje, gradivne elemente svakog workflowa.
3. Resources – Predstavlja datoteke koje se mogu koristiti tokom kreiranja workflowa.
4. Configurations – Centralno definirani atributi koji se mogu koristiti tokom kreiranja workflowa.
5. Packages – Paketi u sebi sadrže workflowe, akcije i sve druge gradivne elemente koji se mogu importati u okrestrator i exportati iz njega.
6. Inventory – Prikazuje sve objekte kojima pojedini plugin ima pristup.

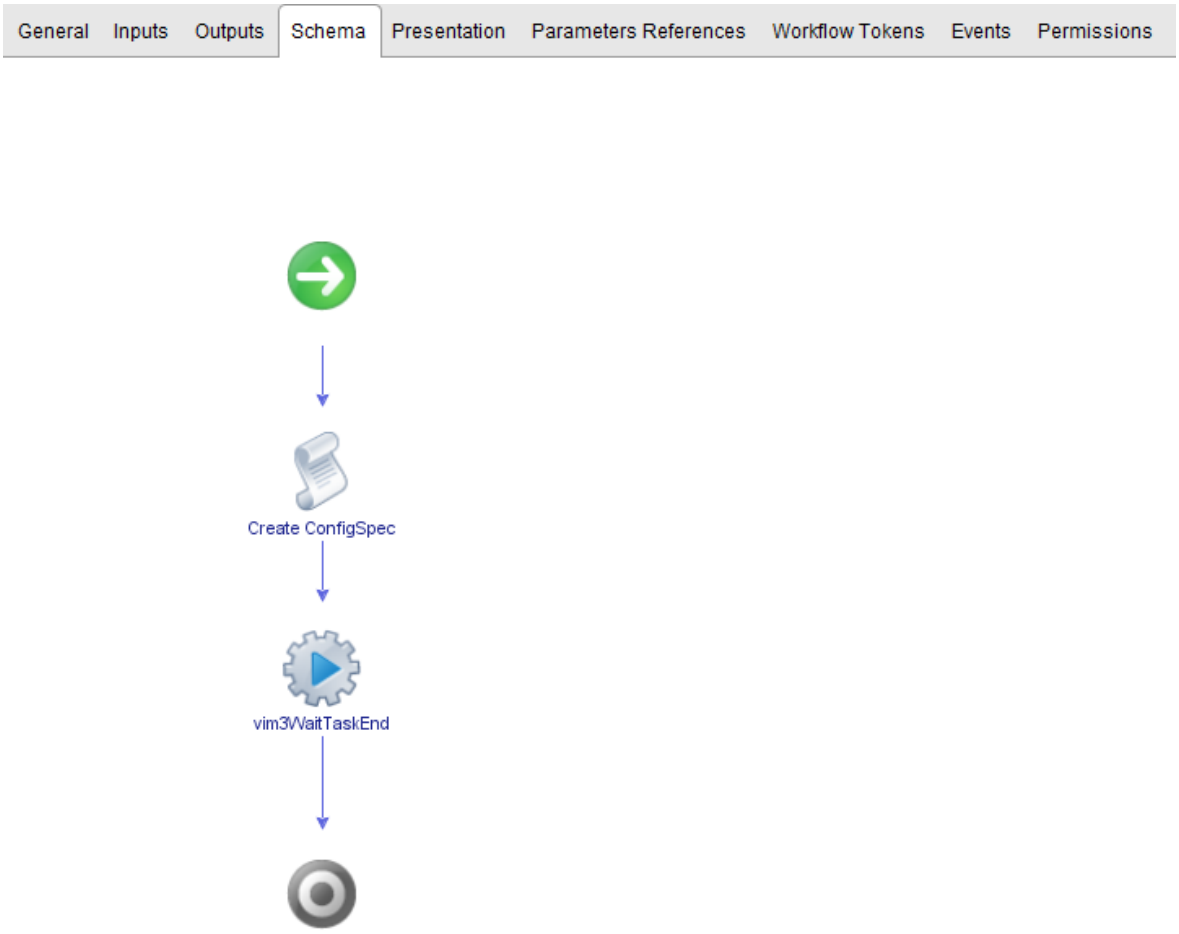
5.2. Što su Workflowi?

Unutar **Design** pogleda klikom na tab **Workflows** dobivamo detaljan pregled svih mapa u kojima se nalaze predefinirani workflowi koji dolaze u sklopu orkestrator instalacije. Također, ovdje se nalaze i workflowi koji se kreiraju prilikom instalacije dodatnih pluginova. S obzirom da smo mi instalirali VMware Horizon plugin, na možemo vidjeti da se automatski kreirao i adekvatni workflow.



Slika 5.3 Workflow tab

Unutar mape **vCenter**, nalazi se nekoliko podmapa. Proširivanjem podmape **Virtual Machine Management** te unutar nje podmape **Basic** možemo vidjeti nekoliko workflowa. Uzmimo za primjer workflow **Create simple virtual machine**. Klikom na taj workflow sa desne strane dobivamo njegove atribute, inpute, outpute i što je najbitnije **Schemu**. Schema prikazuje generalni pogled onoga što će taj workflow zapravo raditi. Schema vizualno prikazuje međusobno povezane objekte, kao i njihov slijed izvršavanja. Na njoj se vidi kada neki objekt ima izravni utjecaj na neki drugi, odnosno kada se rezultat izvršenja nekog objekta ili zadatka može iskoristiti kao ulazni podatak u neki drugi objekt ili zadatak. Upravo je taj vizualni prikaz ono što ljudski mozak brže i lakše interpretira.

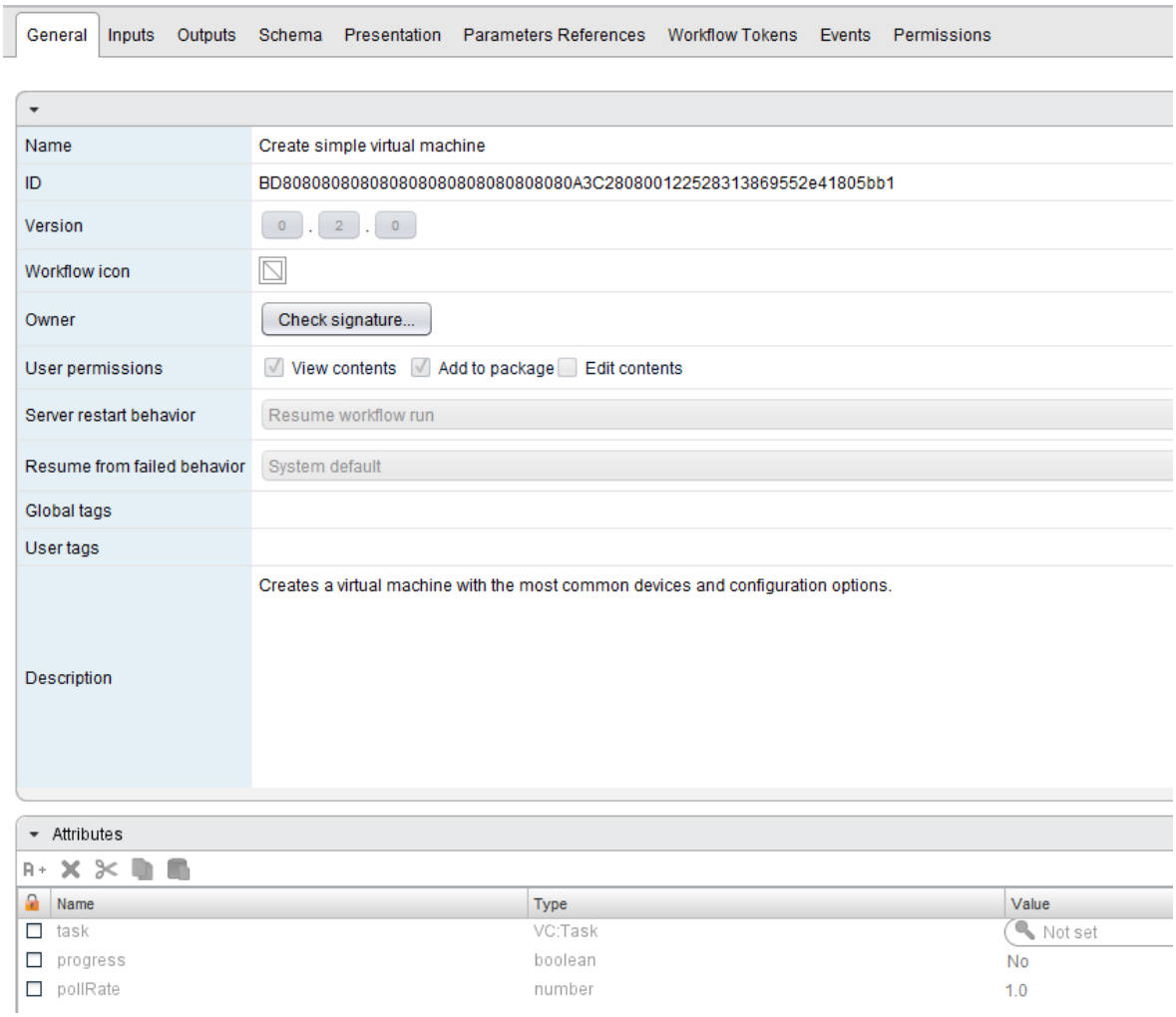


Slika 5.4 Primjer scheme workflowa

Na Slika 5.4 možemo vidjeti da uz Schema tab, workflow sadrži i tabove kao što su **General, Inputs, Outputs, Presentation, Parameters References, Workflow Tokens, Events, Permissions**.

5.2.1. General tab

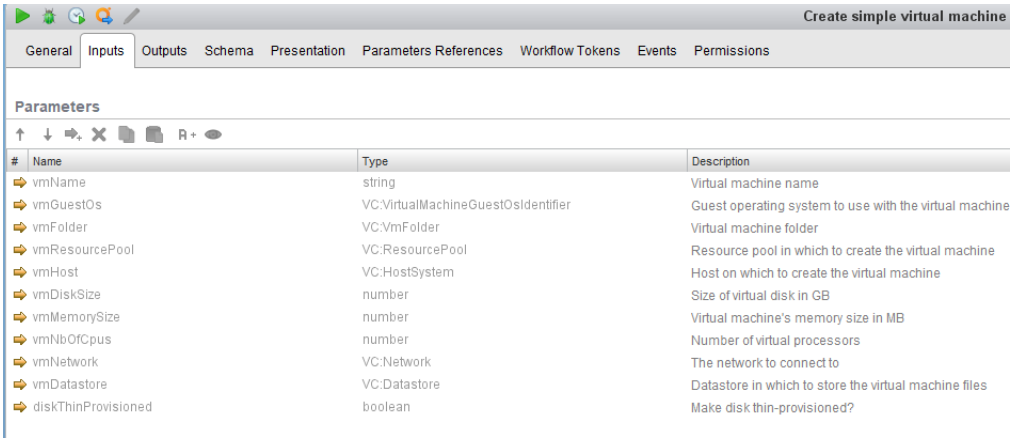
General tab se sastoji od dva glavna područja. Gornje područje prikazuje sve postavke vezane uz workflow kao što su njegovo ime, ID itd. Donje područje prikazuje atribute workflowa. Atributi su globalne varijable koje su definirane za sve elemente tog workflowa.



Slika 5.5 The General tab

5.2.2. Inputs tab

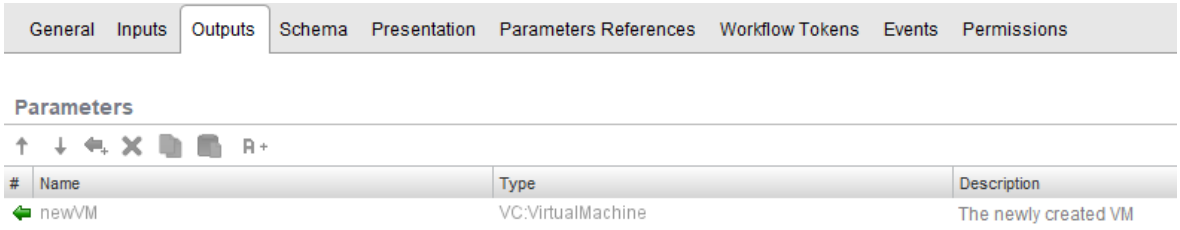
Inputs tab prikazuje sve ulazne parametre za ovaj workflow. Ulazni parametar je varijabla koja treba biti popunjena sa vrijednošću kada workflow započinje.



Slika 5.6 The Inputs tab

5.2.3. Outputs tab

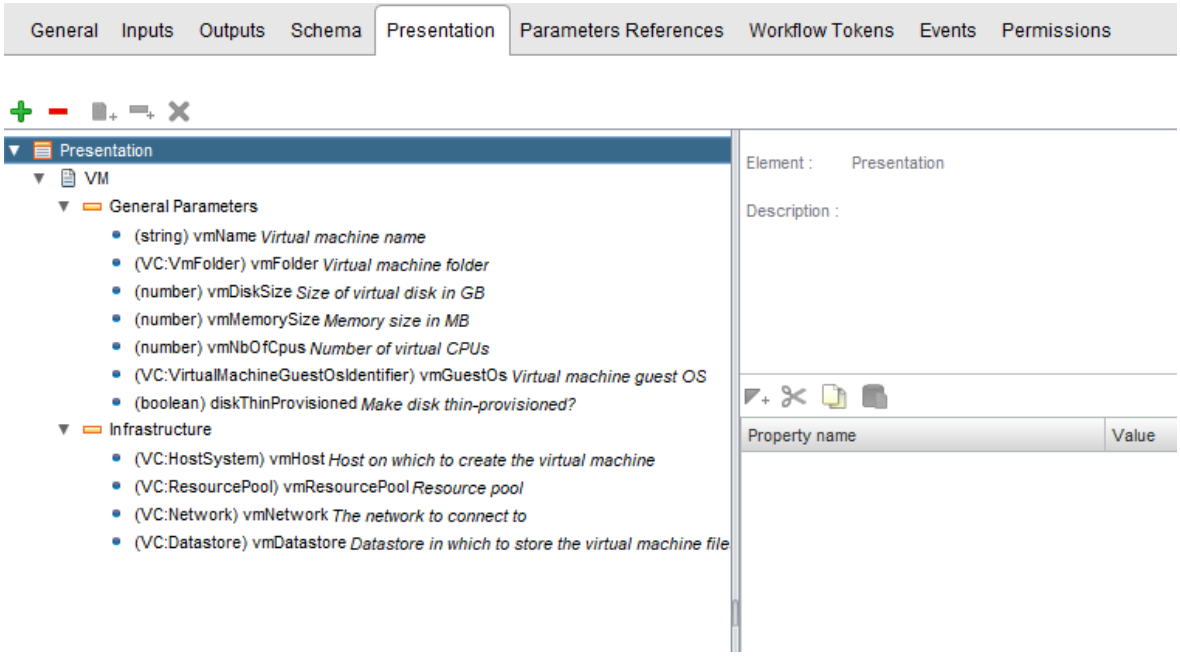
Outputs tab prikazuje sve izlazne parametre ovog workflow-a. Te se vrijednosti dodaju kada workflow završava. Te vrijednosti se onda mogu koristiti od strane drugog workflow-a kao njegovi input parametri.



Slika 5.7 The Outputs tab

5.2.4. Presentation tab

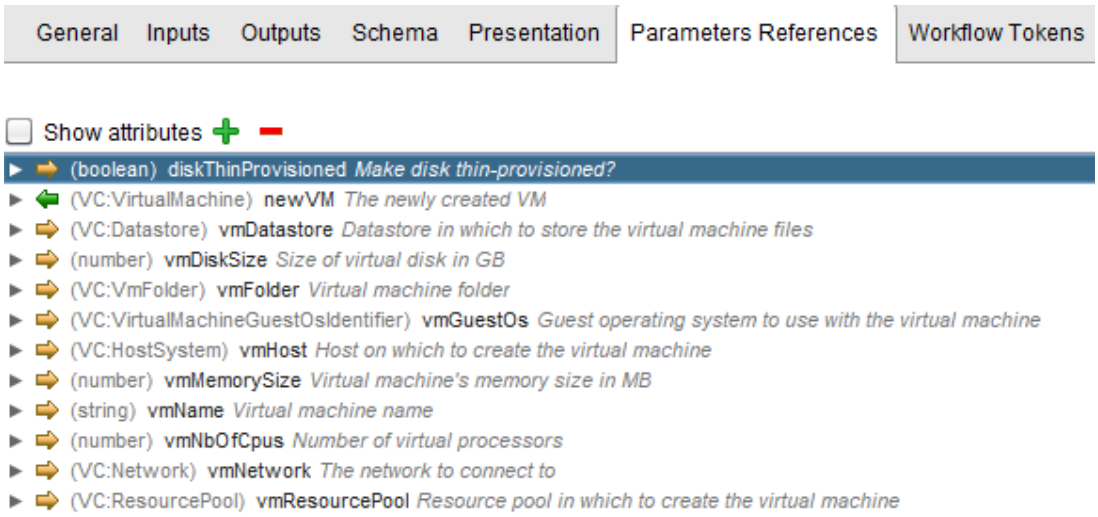
Presentation tab definira kako su input parametri prikazani i formatirani korisniku. Unutar ovog taba se mogu kreirati stranice i sekcije te formatirati načini prikaza, mogu se sortirati parametri te kreirati kustomizirana pravila.



Slika 5.8 The Presentation tab

5.2.5. Parameters References tab

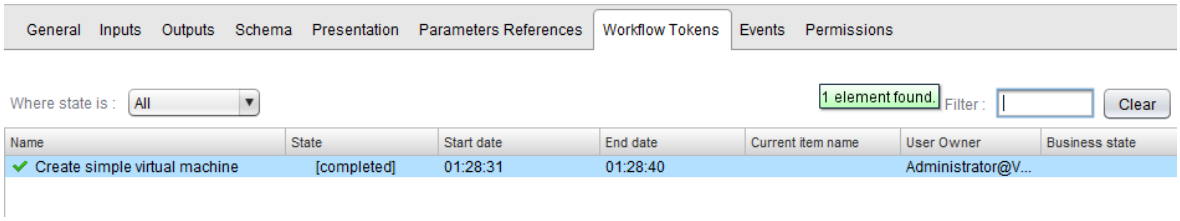
Ovaj tab prikazuje sve parametre unutar workflow-a te postavke odnosno reference za svaki parametar.



Slika 5.9 The Parameters References tab

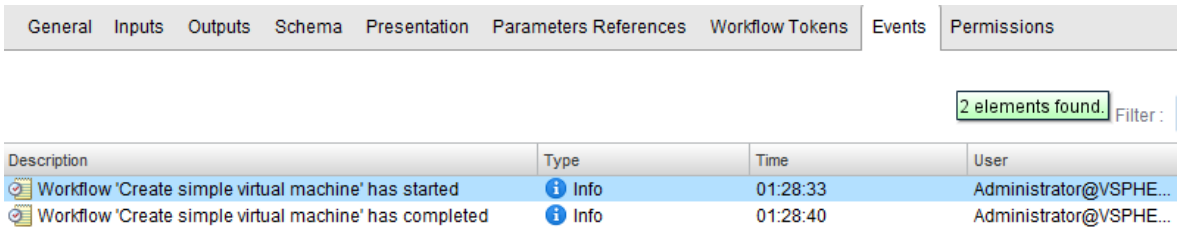
5.2.6. Workflows Tokens tab

Workflow token se kreira prilikom pokretanja workflow-a. Budući da se jedan workflow može pokretati više puta istovremeno, svako pokretanje kreira zasebni unikatni token. Svaki token sadrži status workflow-a, kao i njegovo vrijeme početka i završetka. Također, prikazuje se vlasnik odnosno korisnik koji je workflow pokrenuo što je vrlo korisna informacija.



5.2.7. Events tab

Events tab prikazuje sve događaje vezane uz workflow. Recimo bilježi se događaj kada se workflow pokrenuo, vrijeme pokretanja, kada se završio, vrijeme završetka, greške ukoliko ih je bilo itd.



Slika 5.10 The Events tab

5.2.8. Permissions tab

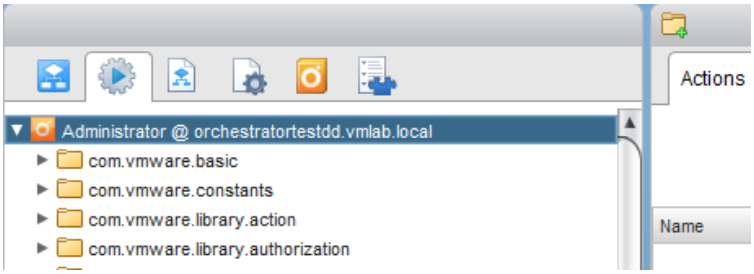
Unutar ovog tab-a reguliraju se sva prava vezana uz workflow.

Slika 5.11 The Permissions tab

5.3. Što su Akcije?

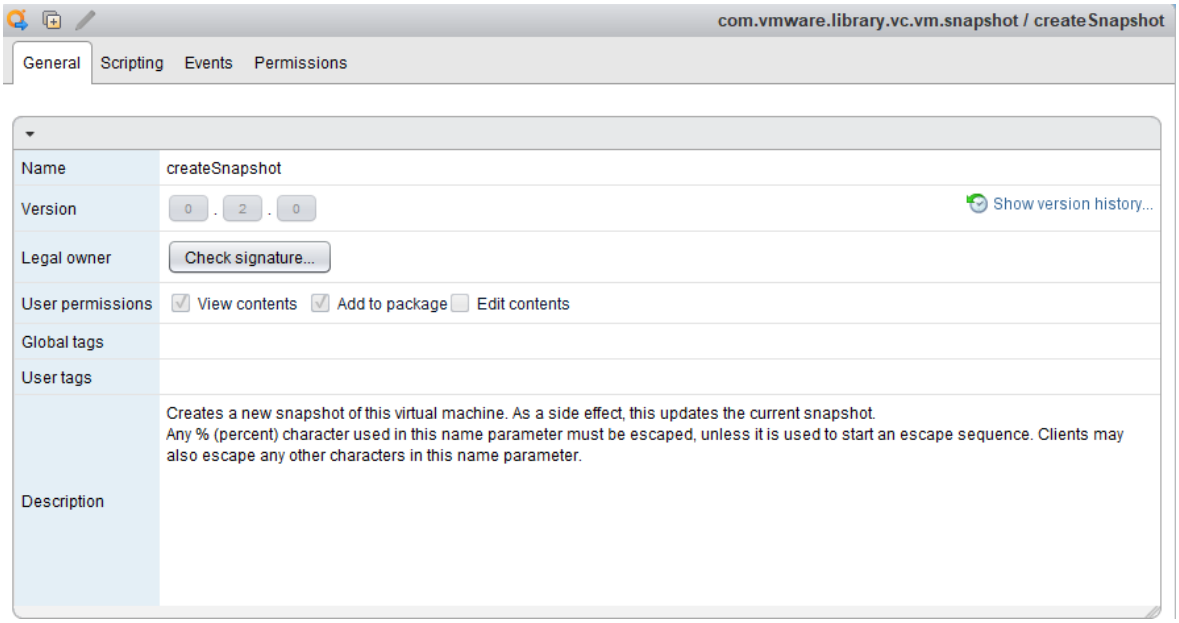
Akcija unutar orkestratora predstavlja skriptu koja izvršava određeni zadatak. Vrlo često akcije se uspoređuju sa funkcijama koje pronalazimo u poznatim programskim jezicima. Glavna distinkcija između workflowa i akcije je ta da akcija vraća samo **jednu** vrijednost ili jedan objekt, dok workflow može vratiti **više** objekata.

Unutar **Design** pogleda klikom na tab **Actions** dobivamo detaljan pregled svih mapa u kojima se nalaze akcije. Te mape se nazivaju **Moduli** te se mogu kreirati kustomizirani moduli klikom na ikonu u gornjem desnom kutu kako je prikazano na Slika 5.12 Actions tab.



Slika 5.12 Actions tab

Svi ovi predefinirani moduli u sebi sadrže predefinirane akcije koje je VMware uključio u sam orkestrator. Ako proširimo modul **com.vmware.library.vc.vm.snapshot** unutar njega vidimo 8 akcija. Klikom na akciju **createSnapshot** akciju sa desne strane dobivamo detaljniji opis te akcije.



Slika 5.13 Detaljan prikaz createSnapshot akcije

Na Slika 5.13 možemo primijetiti pored da postoji i Scripting tab. Klikom na taj tab vidimo skriptu koja se pokreće kako bi se izvršila odabrana akcija. Ovo nam omogućava da dupliciramo akcije i da koristimo kod koji nam VMware daje u orkestratoru za kreiranje kompleksnijih akcija.

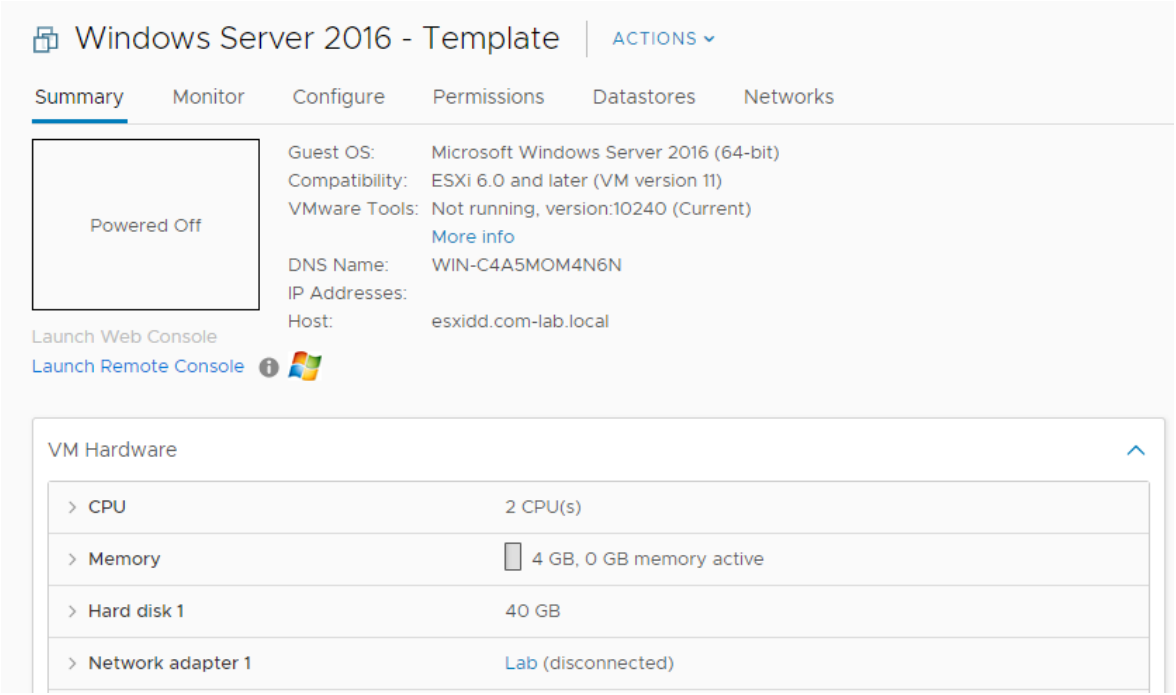
6. Povezivanje više workflow-a u kompleksnije workflowe

VMware orchestrator nam omogućuje povezivanje više workflowa u jedan, te s time dobivamo mogućnost da izvršavamo nekoliko poslova jedan za drugim ili čak paralelno, a usput imamo mogućnost pokretanja svakog tog workflowa individualno ukoliko je potrebno. Koraci praktičnog primjera izvršavanja workflowa unutar workflowa te povezivanje istih u veću strukturu će biti sljedeći:

- Workflow kreiranja linked clone virtualne mašine od template-a naziva SERVERDC
- Promoviranje virtualne mašine SERVERDC u Domain Controller za domenu lab.local
- Kreiranje linked clone virtualne mašine za prvi server SERVER1
- Kreiranje linked clone virtualne mašine za drugi server SERVER2
- Kreiranje linked clone virtualne mašine za treći server SERVER3
- Dodavanje 3 nove mašine u domenu lab.local

6.1. Template

Kako bi automatizirali proces kreiranja virtualnih mašina na najbrži mogući način potrebno je kreirati template virtualnu mašinu. Template će biti referentna virtualna mašina iz koje ćemo kreirati ostale. Ona će sadržavati operacijski sustav koji želimo te konfiguraciju koju želimo. Za primjer je kreirana virtualna mašina naziva Windows Server 2016 – Template i instaliran OS Windows Server 2016. Mašina ima 2 virtualna procesora, 4GB RAM-a te 40GB diska, dodijeljena joj je jedna VMXNET3 mrežna kartica. Mašina je ugašena, sysprep nije potrebno raditi budući da ćemo to odraditi kroz workflow kreiranja linked clone-a.



Slika 6.1 Template VM

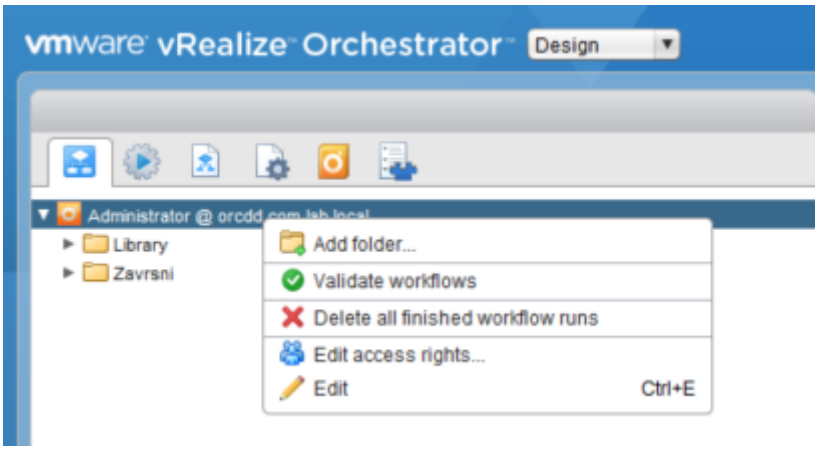
U procesu kloniranja virtualne mašine od template-a, možemo odabrati opciju *full clone* ili *linked clone*. Full clone radi potpuno kopiju, te klonira i disk te ga sprema kao zasebni za tu novu mašinu, što znači da će novi disk zauzimati jednako prostora kao i template disk. Linked clone opcija nam omogućuje da kod kloniranja napravimo novu mašinu koja je povezana na template, da koristi sve podatke sa njenog diska, a na svoj zapisuje samo razlike, pa time štedimo diskovni prostor. U praktičnom primjeru ćemo koristiti linked clone opciju.

6.2. Domain Controller

Prvi workflow kojeg ćemo dizajnirati će biti taj za kreiranje virtualne mašine naziva SERVERDC, dodavanje adekvatnog hostname-a, IP adrese, subnet maske, DNS-a i ostalih postavki kao i promoviranje iste u Domain Controller za lab.local domenu. Workflow će se zvati Domain_Controller_Creation i sastojati će se od dva subworkflowa naziva Linked_Clone_Creation te Domain_Controller_Script.

6.2.1. Dodavanje nove mape

Mapa u koju ćemo spremati workflowe će se zvati „Zavrsni“. Klikom na tab Workflows te desnim klikom na instancu i odabirom opcije Add Folder dodajemo mapu.

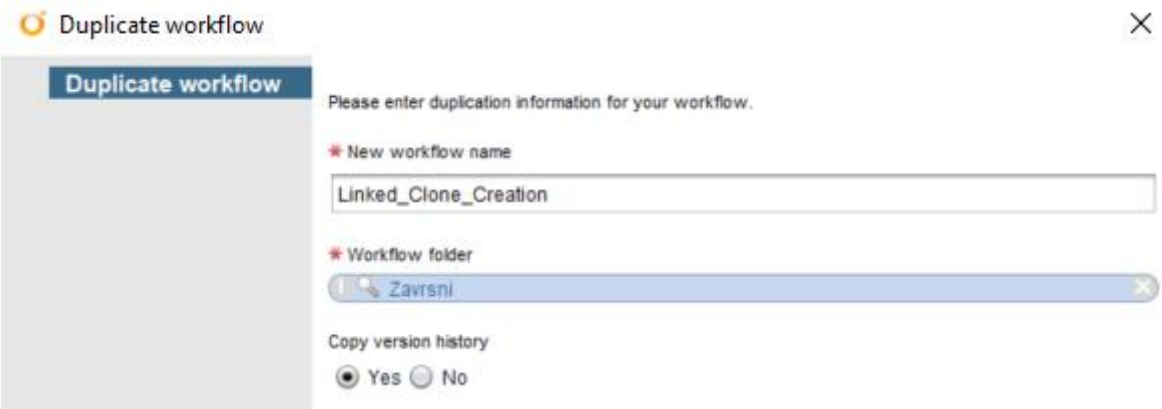


Slika 6.2 Dodavanje mape naziva Zavrsni

6.2.2. Linked_Clone_Creation workflow

Za kreiranje linked clone virtualne mašine od template-a koristiti ćemo već postojeći workflow naziva *Linked clone, Windows with single NIC and credentials.* Navedeni workflow se nalazi na lokaciji **Library | vCenter | Virtual Machine Management | Clone | Linked Clone | Windows Customization.**

Navedeni workflow ćemo kopirati u mapu Zavrsni te mu dodijeliti naziv Linked_Clone_Creation. Desnim klikom te odabirom opcije duplicate workflow, otvara se novi prozor gdje upisujemo naziv i odabiremo mapu Zavrsni te završavamo kopiranje klikom na Submit.



Slika 6.3 Kopiranje workflow-a

6.2.3. Domain_Controller_Script workflow

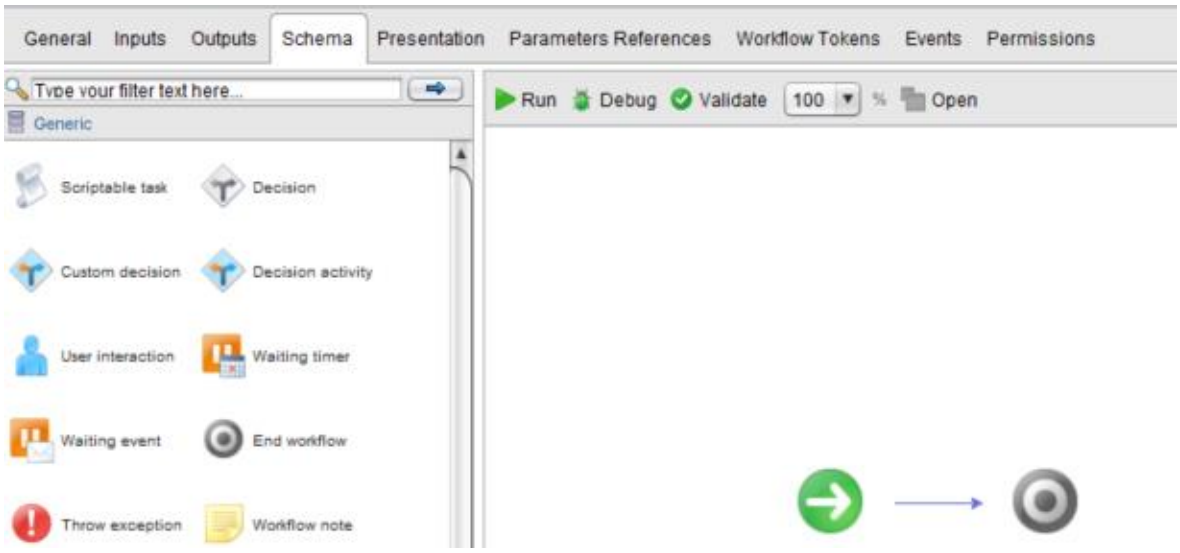
Kako bi na novokreiranoj virtualnoj mašini SERVERDC izvršili skriptu koja će istu promovirati u Domain Controller za domenu lab.local, iskoristiti ćemo workflow naziva *Run program in guest*. Isti se nalazi na lokaciji **Library | vCenter | Guest operations | Processes.**

Na jednak način kao i prijašnji workflow, i ovaj ćemo kopirati u mapu Zavrsni te mu dati naziv Domain_Controller_Script.

6.2.4. Domain_Controller_Creation workflow

Ovaj workflow će u sebi sadržavati dva novokreirana workflowa naziva Linked_Clone_Creation i Domain_Controller_Script. U ovaj workflow ćemo upisati sve potrebne vrijednosti, te povezati sve potrebne objekte za potpunu automatizaciju ovog procesa.

Desnim klikom na mapu Zavrsni, odabiremo opciju New Workflow. Otvara se novi prozor u kojem dajemo ime workflowu te klikom na tab Schema krećemo sa dizajniranjem istoga.



Slika 6.4 Schema tab praznog Domain_Controller_Creation workflowa

Na Slika 6.4, sa lijeve strane vidimo objekte koje možemo dodavati unutar workflowa. S obzirom da želimo pozvati dva workflowa unutar ovoga, objekt kojeg ćemo dodati zove se

Workflow element. Povučemo objekt u prostor iza zelene strelice koja označava početak te se otvara prozor u kojeg upisujemo ime workflowa kojeg želimo pozvati. Upisujemo prvo Linked_Clone_Creation, a zatim Domain_Controller_Script.

Slika 6.5 Umetanje 2 workflowa

Još jedan korak koji moramo napraviti, kako bi se uspješno izvršila ova operacija je dodavanje Sleep objekta, koji će pauzirati izvođenje na kratak period nakon prvog workflowa te nakon tog perioda krenuti na drugi. Ovo se radi zbog stabilnosti izvršavanja te se vrijednost pauze modificira po potrebi, mi ćemo koristiti 120 sekundi. S lijeve strane klikom na Basic tab, odabiremo Sleep objekt te ga povlačimo između dva workflowa kao što možemo vidjeti na Slika 6.6

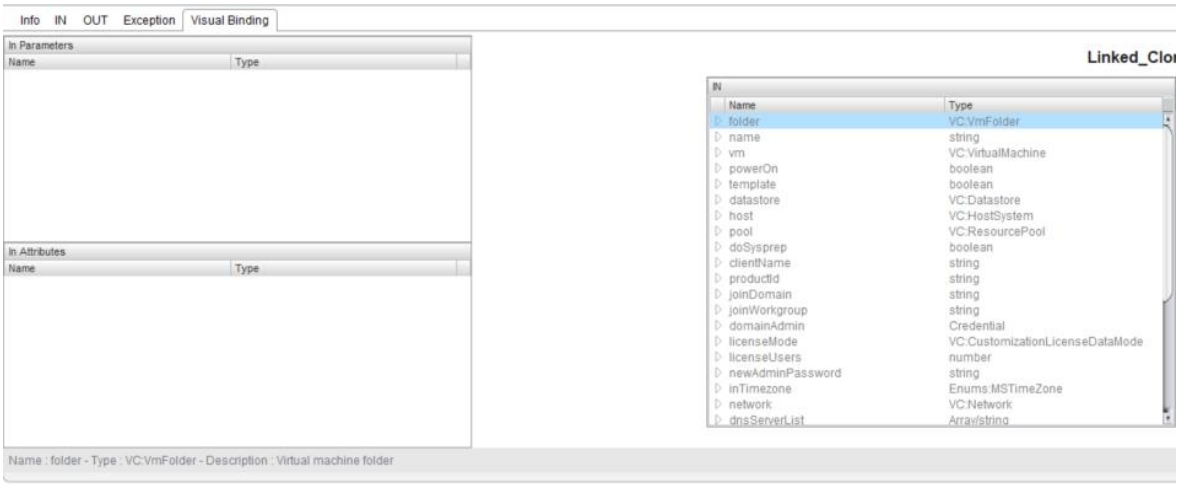
Slika 6.6 Umetanje Sleep objekta

Svaki od ovih objekata ima svoje input i output parametre. Input parametre možemo unijeti odmah ili ih možemo ostaviti prazne pa će nas prilikom pokretanja workflowa orchestrator pitati da ih unesemo jedan po jedan. Output parametar je rezultat uspješno izvršene akcije.

6.2.4.1 Linked_Clone_Creation parametri

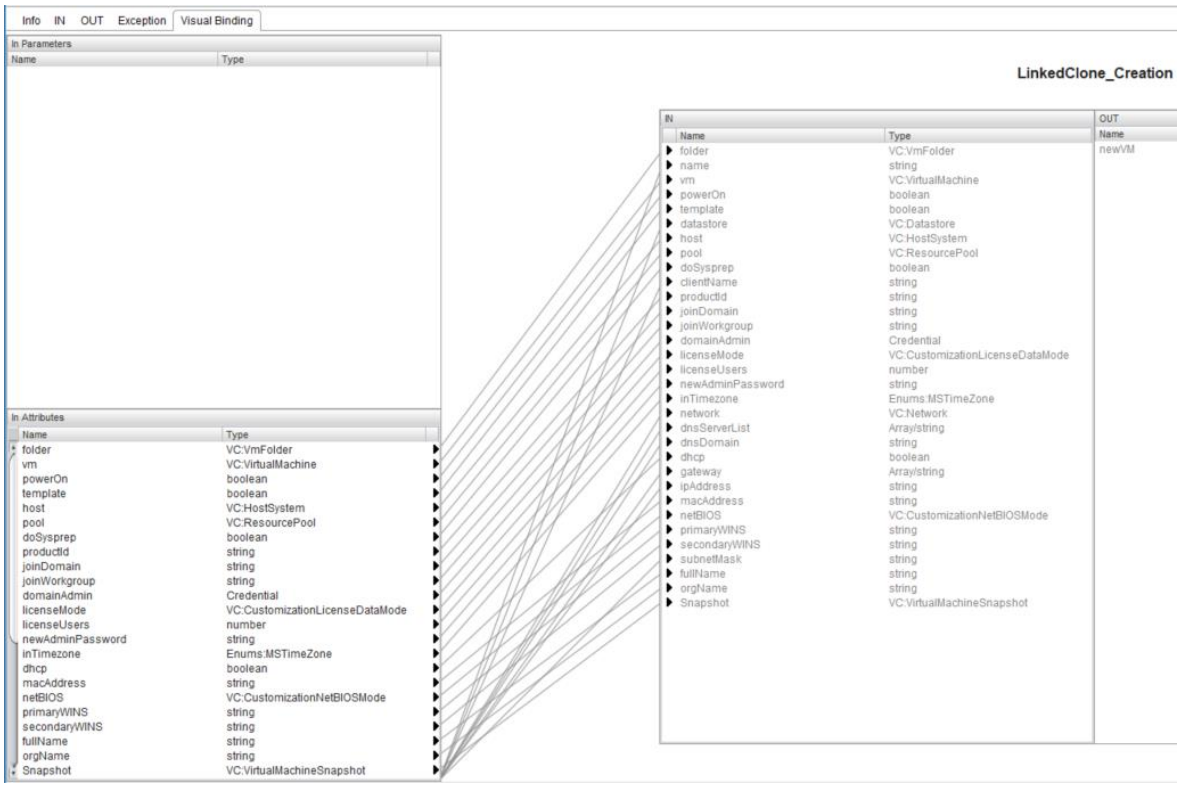
Klikom na Linked_Clone_Creation workflow, otvara se prozor na donjoj polovici ekrana. Odabirom tab-a IN možemo vidjeti sve inpute, na tabu OUT sve outpute, a na tabu Visual Binding njihove vrijednosti i povezivost. Tab Visual Binding sa lijeve strane sadrži In

Parameters, In Attributes, u sredini pokazuje IN i OUT od trenutnog objekta, a sa desne Out Parameters i Out Attributes. In Parameters su input parametri kojima ne želimo odmah dati vrijednost već želimo da nas orchestrator pita prilikom pokretanja, dok su In Attributes input parametri kojima želimo odmah dati vrijednost kako bi workflow izvršili bez potrebe za upisivanjem istih. Prilikom povlačenja objekta u IN attributes otvoriti će se dijaloški okvir u kojem ćemo moći odmah upisati vrijednost.



Slika 6.7 Linked_Clone_Creation parametri

S obzirom da želimo izvršiti workflow bez prekida za upisivanjem vrijednosti, mi ćemo svaki IN parametra povući u In Attributes te mu odmah upisati vrijednost. Konačni rezultat je prikazan na Slika 6.8 Linked_Clone_Creation In Attributes



Slika 6.8 Linked_Clone_Creation In Attributes

Svakom od atributa je dodijeljena vrijednost. Za neke atribute kao što su ime ESXi hosta na kojem će se virtualna mašina kreirati i pokretati, ime Datastore-a gdje će virtualna mašina, njeni diskovi i konfiguracija biti spremljeni, vrijednost se razlikuje od infrastrukture do infrastrukture. U našem slučaju sažetak najvažnijih atributa je prikazan u Tablica 6.1

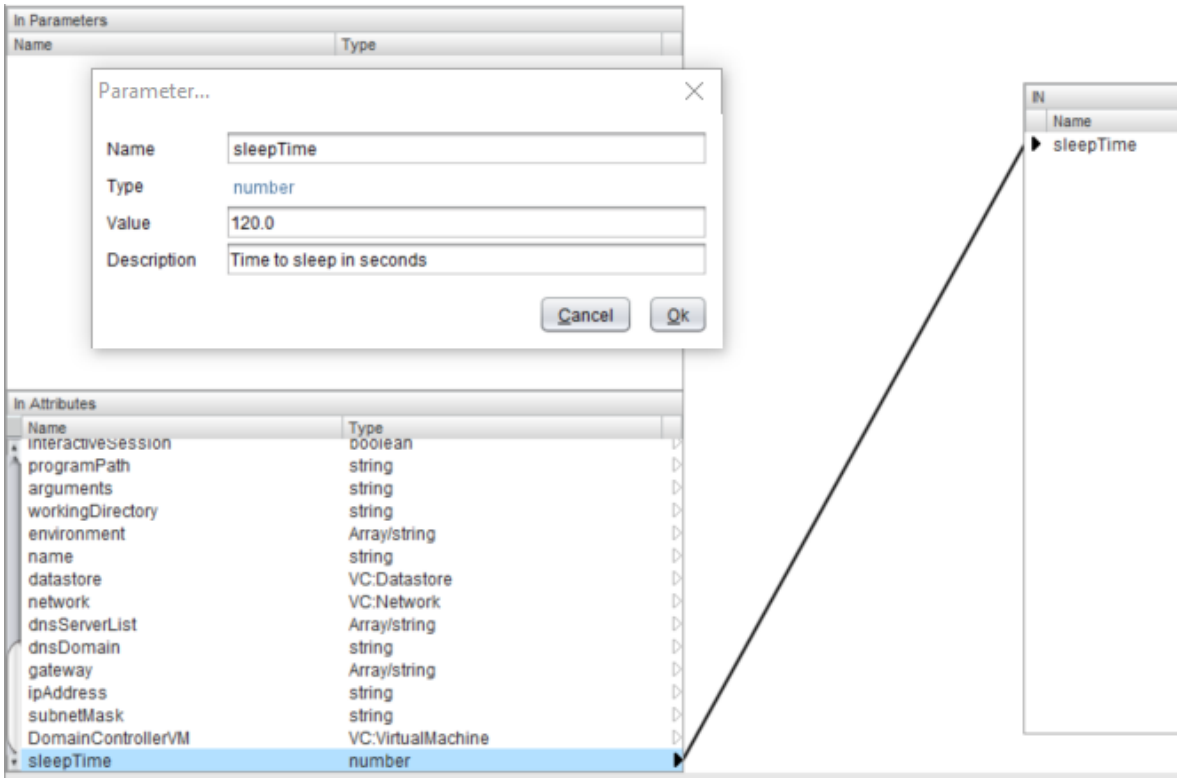
Tablica 6.1 Linked_Clone_Creation najvažniji atributi

In atribut	**Vrijednost**
Name	SERVERDC
ClientName (hostname)	SERVERDC
DoSysprep	Yes
FullName	Administrator
NewAdminPassword	Pa$$w0rd
InTimezone	Central European Standard Time

IpAddress	10.10.10.1
SubnetMask	255.255.255.0
DnsServerList	127.0.0.1

6.2.4.2 Sleep parametar

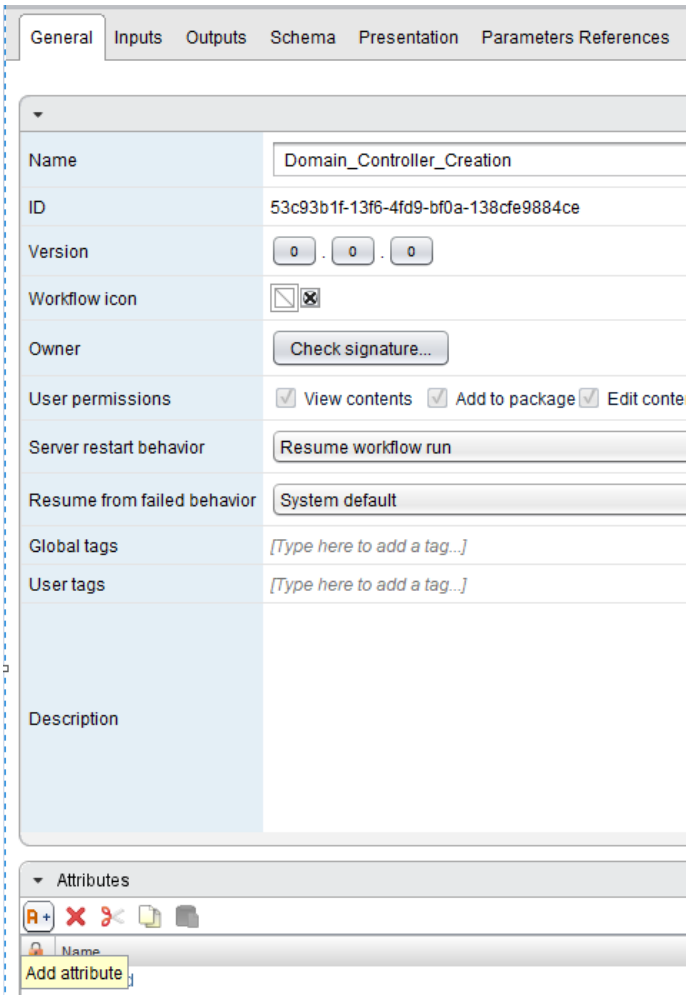
Sleep objekt je vrlo jednostavan te ima samo jedan IN parametar naziva sleepTime koji nam govori na koliko dugo ćemo pauzirati workflow te je izražen u sekundama. Povlačimo ga u InAttributes, dvokliknemo na atribut te mu dodijeljujemo vrijednost 120 kako je prikazano na Slika 6.9 SleepTime



Slika 6.9 SleepTime

6.2.4.3 Domain_Controller_Script parametri

Da bi promovirali novokreiranu virtualnu mašinu u Domain Controller, ovim workflowom ćemo pokretati Powershell skriptu unutar operacijskog sustava virtualne mašine SERVERDC. Kako bi taj workflow znao na kojoj virtualnoj mašini mora pokretati skriptu, on za input parametar virtualne mašine, mora uzeti output parametar prethodnog workflowa. Da bi to postigli, kreirati ćemo novi atribut naziva DomainControllerVM te ćemo ga povezati na output parametar workflowa Linked_Clone_Creation. Kako bi kreirali novi atribut, prebaciti ćemo se na General tab, te u prozoru attributes kliknuti na *Add attribute*. Dati ćemo mu odgovarajući naziv, te mu staviti tip VC:VirtualMachine kao što je prikazano na slikama Slika 6.10Slika 6.11.



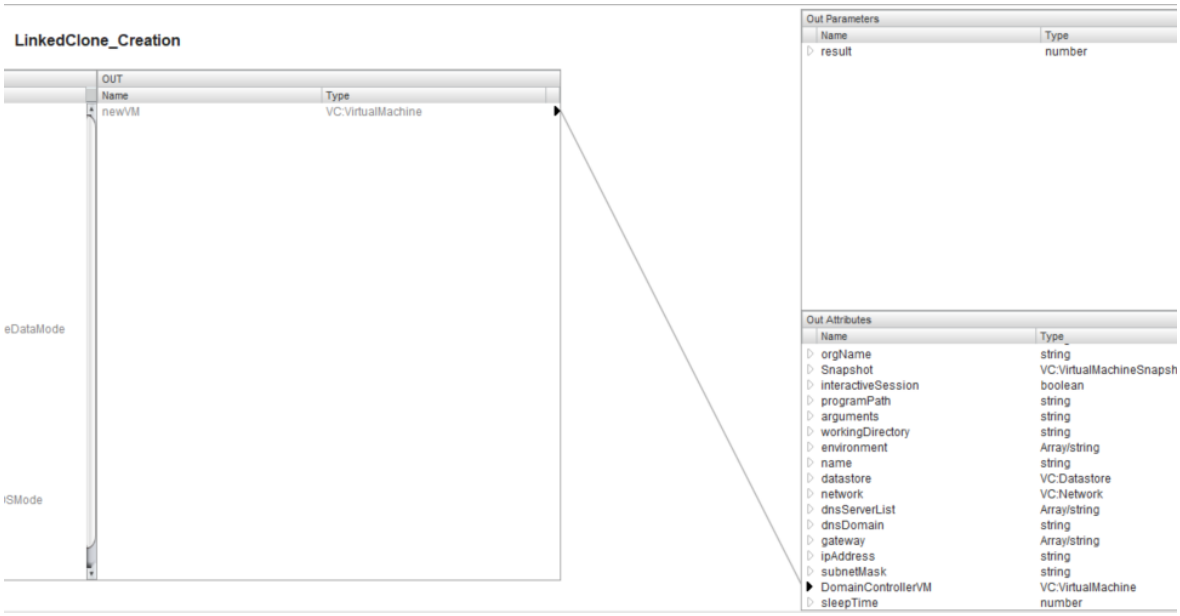
Slika 6.10 Dodavanje atributa

Attributes

Name	Type
secondaryWINS	string
fullName	string
orgName	string
Snapshot	VC:VirtualMachineSnapshot
interactiveSession	boolean
programPath	string
arguments	string
workingDirectory	string
environment	Array/string
name	string
datastore	VC:Datastore
network	VC:Network
dnsServerList	Array/string
dnsDomain	string
gateway	Array/string
ipAddress	string
subnetMask	string
DomainControllerVM	VC:VirtualMachine
sleepTime	number

Slika 6.11 Domain Controller VM atribut

Nazad na Schema tabu, klikom na workflow Linked_Clone_Creation, povezujemo output parametar newVM sa Out atributom DomainControllerVM kao što je prikazano na Slika 6.12.



Slika 6.12 Povezivanje out parametara

Sada kada imamo out atribut DomainControllerVM u koji će, nakon kreiranja SERVERDC VM-a biti spremljena vrijednost te virtualne mašine, taj atribut možemo iskoristiti kao in atribut za workflow Domain_Controller_Script. Klikom na isti, te na tab Visual Binding, povezujemo in parametar vm sa in atributom DomainControllerVM. Sve ostale in parametre povezujemo na način kako je prikazano

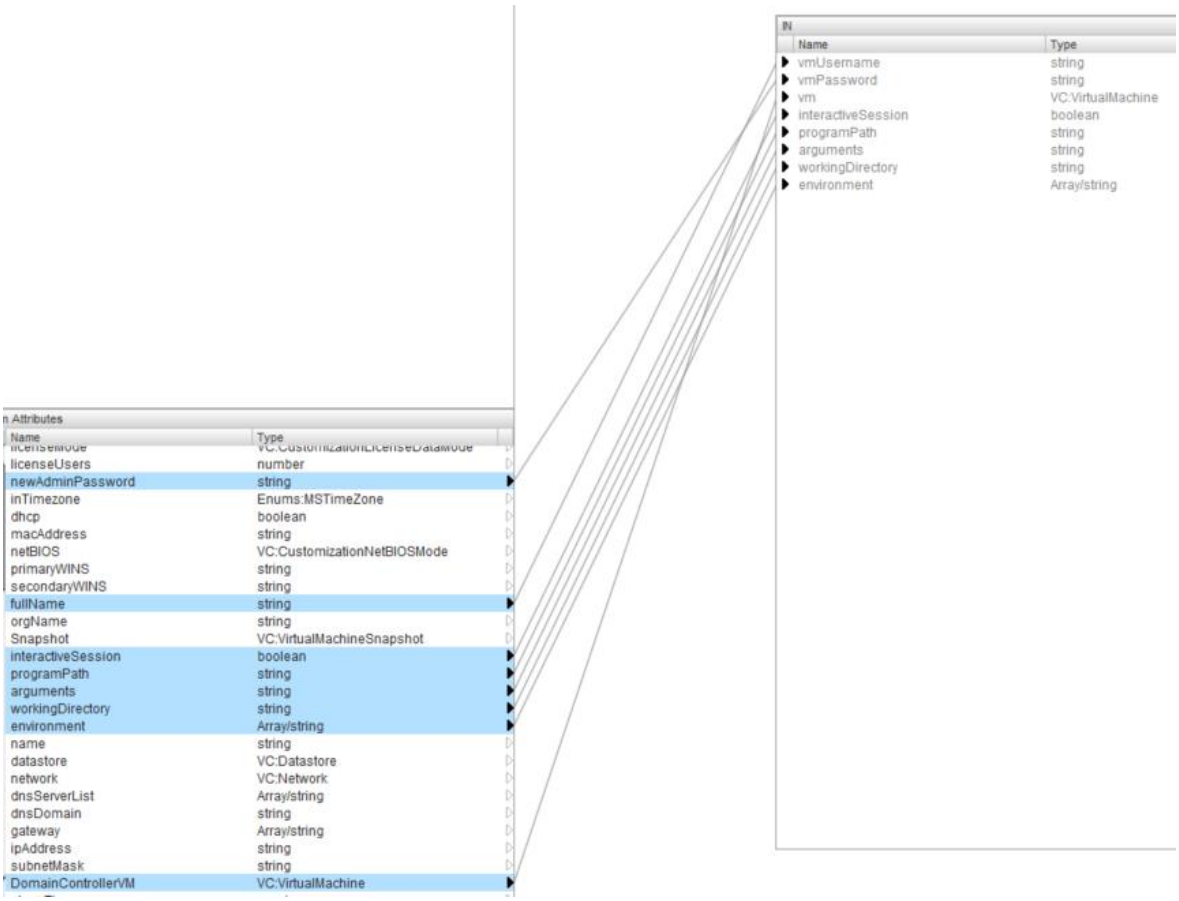
Tablica 6.2 Domain_Controller_Script parametri

In parametar	Objašnjenje	In atribut
vmUsername	Username na serveru koji ima prava izvršiti skriptu	Povezujemo ga sa in atributom fullName u kojem smo upisali vrijednost Administrator
vmPassword	Lozinka od navedenog korisnika Administrator	Povezujemo ga sa in atributom newAdminPassword u kojem smo upisali vrijednost Pa$$w0rd
vm	Na kojoj virtualnoj mašini želimo izvršiti skriptu	Povezujemo ga sa in atributom DomainControllerVM koji će, nakon kreiranja SERVERDC virtualne mašine, u sebi sadržavati link na tu mašinu
programPath	Putanja unutar virtualne mašine na kojoj se nalazi program koji želimo pokrenuti	S obzirom da želimo izvršiti Powershell skriptu, ovdje upisujemo putanju do powershell.exe programa. C:\Windows\System32\ WindowsPowerShell\v1.0\powershell.exe
interactiveSession	Želimo li da se na ekranu virtualne mašine pojavi powershell prozor putem kojeg	Yes

	možemo pratiti izvršenje skripte	
Arguments	Ovdje upisujemo sami kod skripte	Kod je prikazan pod Kôd 6.1 Powershell skripta za promoviranje Domain Controllera
workingDirectory	Gdje želimo da se skripta izvršava	Ovo ostavljamo prazno, to znači da će se izvršavati unutar Administrator home foldera što je u redu
Enviromment	Želimo li koristiti environment varijable	Možemo navesti polje varijabli, u ovom slučaju nisu potrebne.

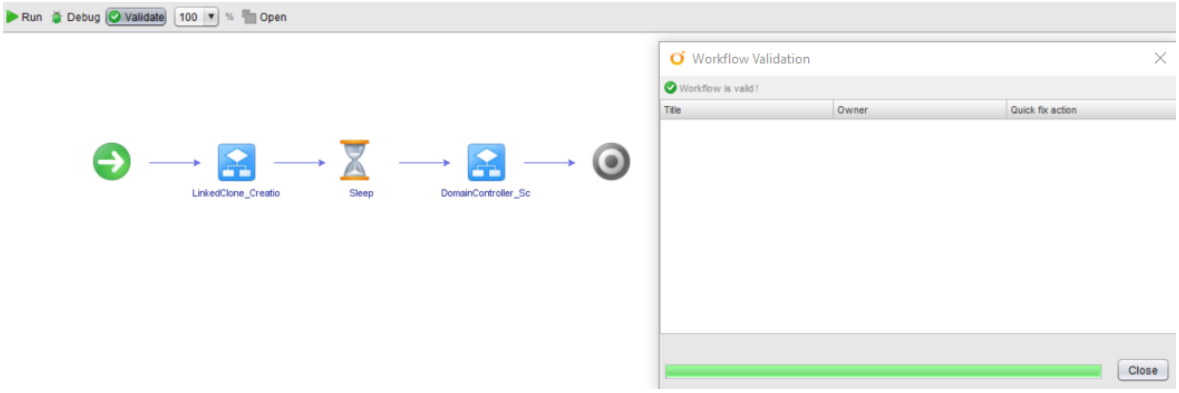
```
powershell.exe -command {install-windowsfeature ad-domain-services,dns `
-IncludeAllSubfeature -IncludeManagementTools};
powershell.exe -command {Import-Module ADDSDeployment};
powershell.exe -command {Install-ADDSForest `
-DatabasePath "C:\Windows\NTDS" `
-DomainMode "WinThreshold" `
-DomainName "lab.local" `
-DomainNetbiosName "LAB" `
-ForestMode "WinThreshold" `
-InstallDns:$true `
-LogPath "C:\Windows\NTDS" `
-NoRebootOnCompletion:$false `
-SysvolPath "C:\Windows\SYSVOL" `
-SafeModeAdministratorPassword (ConvertTo-SecureString 'Pa$$w0rd' -
AsPlainText -Force) `
-Force:$true}
```

Kôd 6.1 Powershell skripta za promoviranje Domain Controllera



Slika 6.13 Domain_Controller_Script konačni rezultat

Povezivanjem svih ovih parametara i atributa svakog pojedinačnog workflowa, te dodavanje odgovarajućih vrijednosti atributima, završili smo sa kreiranjem Domain_Controller_Creation glavnog workflowa. Klikom na gumb *Validate* možemo provjeriti ispravnost workflowa, a ukoliko ima ikakvih grešaka, orchestrator će ih dojaviti te će napisati kako iste riješiti.



Slika 6.14 Domain_Controller_Creation workflow validation

Workflow spremamo klikom na *Save and close* u donjem desnom kutu ekrana.

6.3. Three Member Servers

Kreiranje member servera je proces koji je identičan kao i kreiranje SERVERDC virtualne mašine, jedina razlika je u skripti koja se pokreće nakon kreacije. Prvi korak je kopirati Domain_Controller_Script workflow-a i dati mu ime Member_Server_Script. Nakon toga ćemo kreirati 3 sljedeća workflow-a:

- First_Member_Server_Creation
- Second_Member_Server_Creation
- Third_Member_Server_Creation

Ti workflowi će biti prazni, pa ćemo stoga ući u svaki od njih i dodati objekt *Workflow element* naziva Linked_Clone_Creation, pa objekt *Sleep*, pa zatim objekt *Workflow element* naziva Member_Server_Script. Svaki od 3 nova workflowa izgleda kao što je prikazano na Slika 6.15

Slika 6.15 Izgled 3 nova workflowa

Za Linked_Clone_Creation workflow, postupak dodavanja vrijednosti atributima te povezivanje istih na in i out parametre je u potpunosti identičan kao i kod Linked_Clone_Creation unutar Domain_Controller_Creation workflowa. Ono što se razlikuje su vrijednosti imena virtualne mašine, IP adresa, DNS itd. Sumarni prikaz najbitnijih atributa za svaki od 3 workflowa je prikazan u tablicama Tablica 6.3Tablica 6.4Tablica 6.5.

Tablica 6.3 Linked_Clone_Creation workflow atributi unutar First_Member_Server_Creation workflowa

In atribut	**Vrijednost**
Name	SERVER1
ClientName (hostname)	SERVER1
DoSysprep	Yes
FullName	Administrator
NewAdminPassword	Pa$$w0rd
InTimezone	Central European Standard Time
IpAddress	10.10.10.2
SubnetMask	255.255.255.0
DnsServerList	10.10.10.1

Tablica 6.4 Linked_Clone_Creation workflow atributi unutar Second_Member_Server_Creation workflowa

In atribut	**Vrijednost**
Name	SERVER2
ClientName (hostname)	SERVER2
DoSysprep	Yes
FullName	Administrator
NewAdminPassword	Pa$$w0rd
InTimezone	Central European Standard Time
IpAddress	10.10.10.3
SubnetMask	255.255.255.0

DnsServerList	10.10.10.1

Tablica 6.5 Linked_Clone_Creation workflow atributi unutar Third_Member_Server_Creation workflowa

In atribut	**Vrijednost**
Name	SERVER3
ClientName (hostname)	SERVER3
DoSysprep	Yes
FullName	Administrator
NewAdminPassword	Pa$$w0rd
InTimezone	Central European Standard Time
IpAddress	10.10.10.4
SubnetMask	255.255.255.0
DnsServerList	10.10.10.1

Kako bi putem Member_Server_Creation worflowa mogli izvršiti skriptu na svakom pojedinom serveru unutar svakog od 3 glavna workflowa potrebno je kreirati atribut na koji ćemo povezati out parametar Linked_Clone_Creation, isto kao i kod kreiranja SERVERDC.

Atribute ćemo nazvati respektivno:

- FirstMemberServer
- SecondMemberServer
- ThirdMemberServer

U svakom od 3 glavna workflowa ćemo povezati out parametar Linked_Clone_Creationa naziva newVM sa tim out atributom kao što je bilo prikazano na Slika 6.12 Povezivanje out parametara

Sleep objektu ćemo ponovno dati vrijednost od 120 sekundi, a Member_Server_Script atributi će biti kao što je prikazano u

In parametar	Objašnjenje	In atribut
vmUsername	Username na serveru koji ima prava izvršiti skriptu	Povezujemo ga sa in atributom fullName u kojem smo upisali vrijednost Administrator
vmPassword	Lozinka od navedenog korisnika Administrator	Povezujemo ga sa in atributom newAdminPassword u kojem smo upisali vrijednost Pa$$w0rd
vm	Na kojoj virtualnoj mašini želimo izvršiti skriptu	Povezujemo ga sa in atributom FirstMemberServer (ili SecondMemberServer ili ThirdMemberServer ovisno u kojem se workflowu nalazimo) koji će, nakon kreiranja virtualne mašine, u sebi sadržavati link na tu mašinu
programPath	Putanja unutar virtualne mašine na kojoj se nalazi program koji želimo pokrenuti	S obzirom da želimo izvršiti Powershell skriptu, ovdje upisujemo putanju do powershell.exe programa. C:\Windows\System32\ WindowsPowerShell\v1.0\powershell.exe
interactiveSession	Želimo li da se na ekranu virtualne mašine pojavi powershell prozor putem kojeg	Yes

	možemo pratiti izvršenje skripte	
Arguments	Ovjde upisujemo sami kod skripte	Kod je prikazan na **Pogreška! Izvor reference nije pronađen.**
workingDirectory	Gdje želimo da se skripta izvršava	Ovo ostavljamo prazno, to znači da će se izvršavati unutar Administrator home foldera što je u redu
Enviromment	Želimo li koristiti environment varijable	Možemo navesti polje varijabli, u ovom slučaju nisu potrebne.

```
cmd /c netdom join %computername% /domain:lab.local /userd:administrator
/passwordd:'Pa$$w0rd'; shutdown /r /t 0
```

Kôd 6.2 cmd skripta za dodavanje servera u lab.local domenu

Klikom na *Validate* provjeravamo točnost workflowa te ga spremamo klikom na *Save and Close*.

Trenutna situacija je prikazana u Tablica 6.6

Tablica 6.6 Four main workflows

Main workflows	**Subworkflows**
Domain_Controller_Creation	Sastoji se od Linked_Clone_Creation i Domain_Controller_Script
First_Member_Server_Creation	Sastoji se od Linked_Clone_Creation i Member_Server_Script
Second_Member_Server_Creation	Sastoji se od Linked_Clone_Creation i Member_Server_Script

Third_Member_Server_Creation	Sastoji se od Linked_Clone_Creation i Member_Server_Script

6.4. The_Workflow_Of_Genesis

Finalni korak je kreacija workflowa koji će spajati sva 4 dosadašnja u jedan, te će jedinom klikom omogućiti kreaciju SERVERDC VM-a, njeno promoviranje u Domain Controller za lab.local domenu, paralelnu kreaciju SERVER1, SERVER2 i SERVER3 VM-ova te njihovo paralelno dodavanje u lab.local domenu.

Desnim klikom na mapu Zavrsni, odabiremo opciju New Workflow. Dajemo mu ime The_Workflow_Of_Genesis, dodajemo *Workflow element* Domain_Controller_Creation, zatim Sleep objekt. Za paralelno izvršavanje 3 workflowa odjednom, nakon Sleep objekta dodajemo objekt *Nested Workflows*. U donjem dijelu ekrana, u tabu *Workflows* dodamo respektivno First_Member_Server_Creation, zatim Second_Member_Server_Creation, a zatim i Third_Member_Server_Creation. The_Workflow_Of_Genesis sada izgleda kao na Slika 6.16.

Slika 6.16 The_Workflow_Of_Genesis

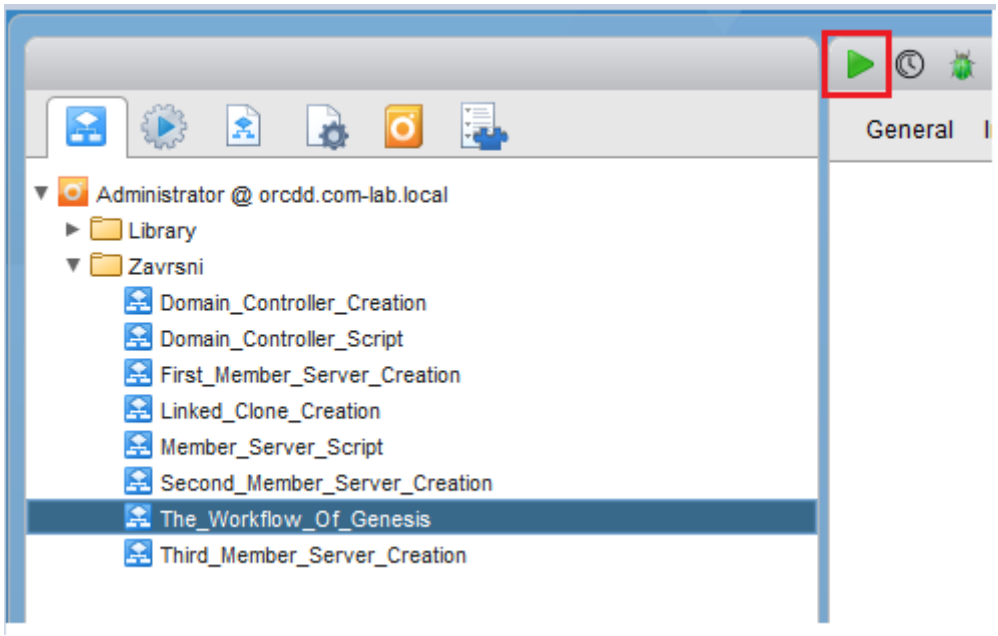
Sve što je još potrebno napraviti je dodati vrijednost sleepTime atributu. U ovom workflowu sleepTime atribut nam daje do znanja koliko je potrebno Domain Controlleru da obavi promoviranje prije nego krenemo kreirati 3 member servera. Vrijednost varira od infrastrukture do infrastrukture, no ovdje ćemo staviti 600 sekundi (10 minuta) kao prosječnu vrijednost.

Klikom na *Validate* provjeravamo ispravnost workflowa, te klikom na *Save and Close* isti spremamo.

S ovime završava proces dizajniranja automatiziranih workflowa, sve što je ostalo je glavni workflow naziva The_Workflow_Of_Genesis pokrenuti te pratiti automatizirani proces.

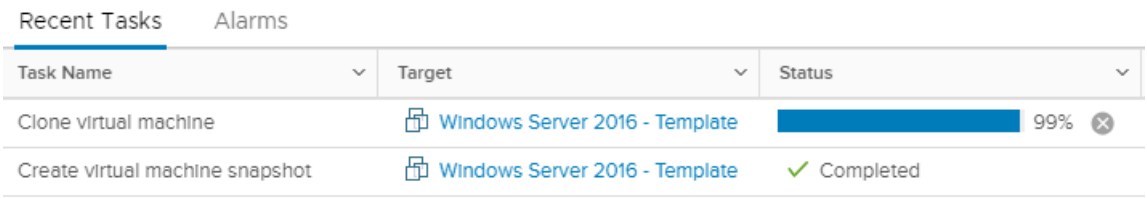
6.5. Proces automatizacije

U mapi Zavrsni, odabrati The_Workflow_Of_Genesis te pritskom na gumb Play kreće njegovo izvršavanje.

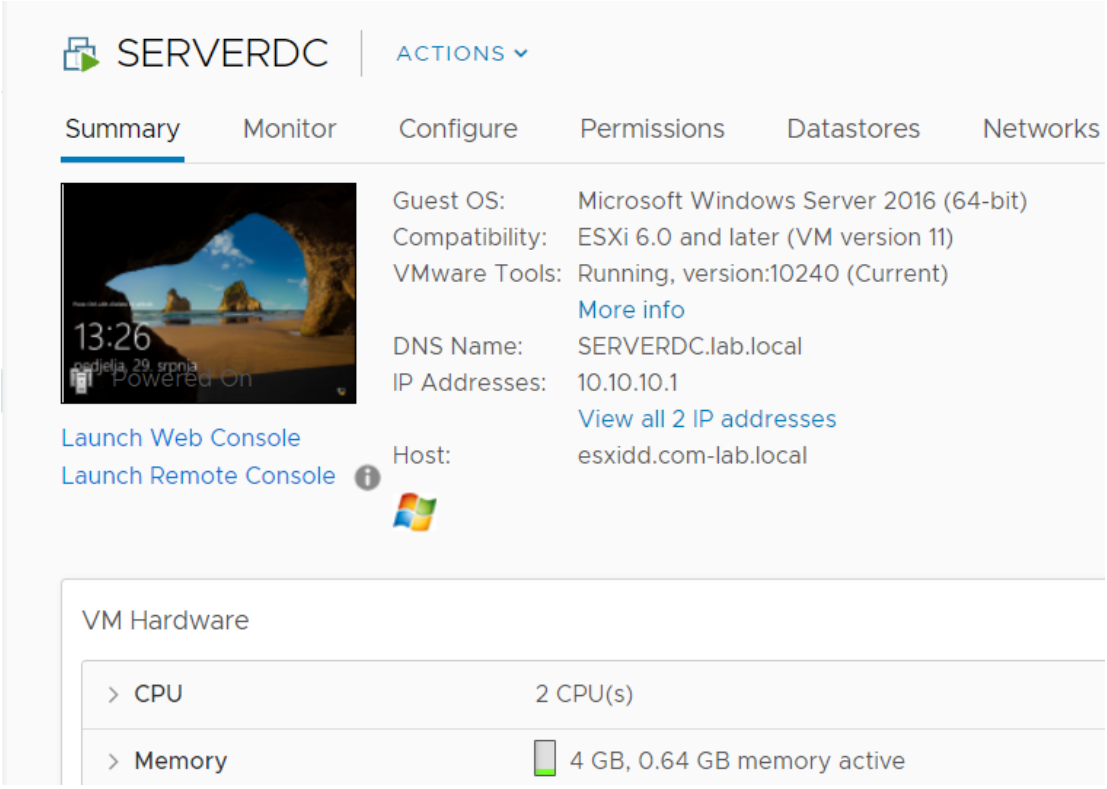


Slika 6.17 The_Workflow_Of_Genesis pokretanje

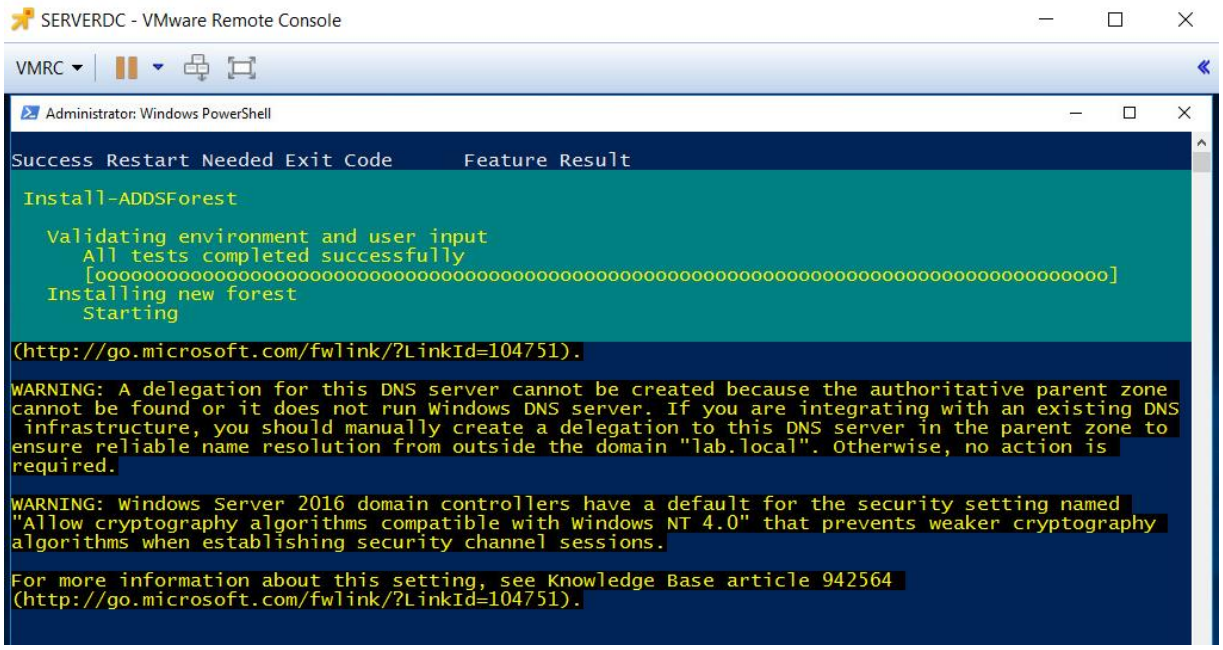
Potpuni proces je prikazan na slikama Slika 6.18Slika 6.19Slika 6.20Slika 6.21, Slika 6.22, Slika 6.23Slika 6.24Slika 6.25 i Slika 6.26.



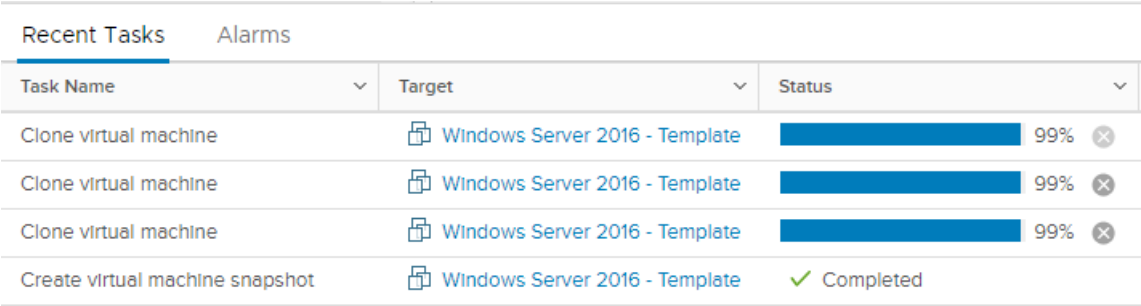
Slika 6.18 Kloniranje Template-a u SERVERDC



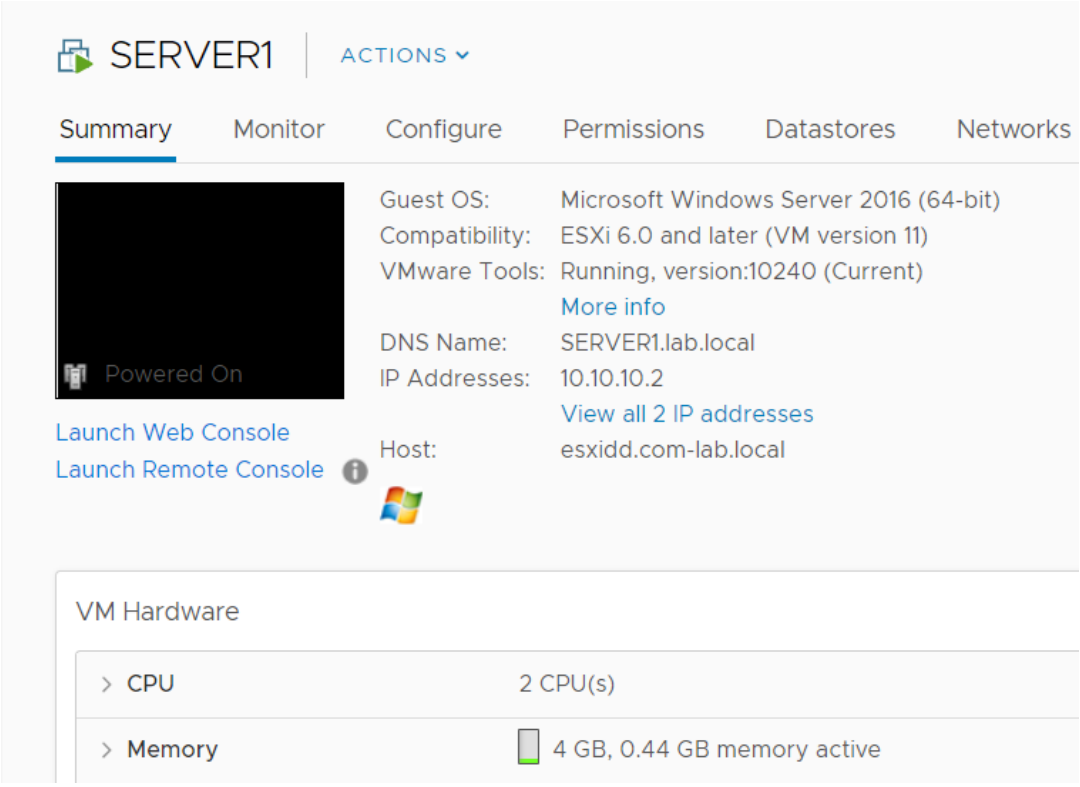
Slika 6.19 SERVERDC



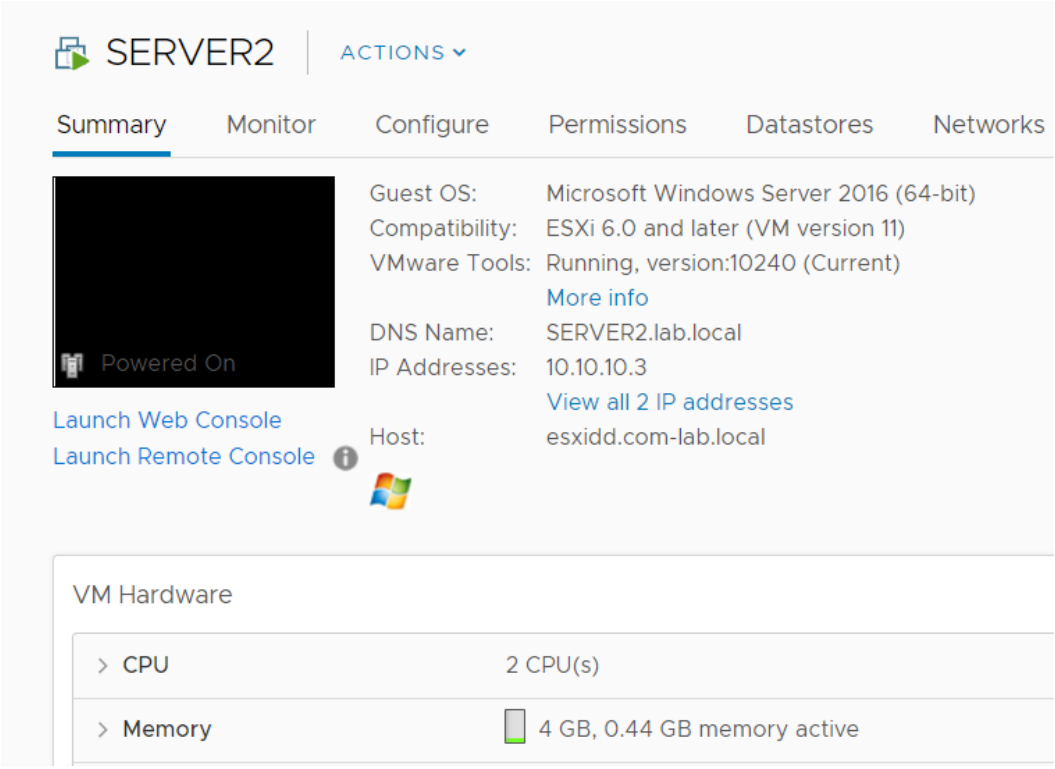
Slika 6.20 Automatsko pokretanje skripte za promoviranje SERVERDC u Domain Controller



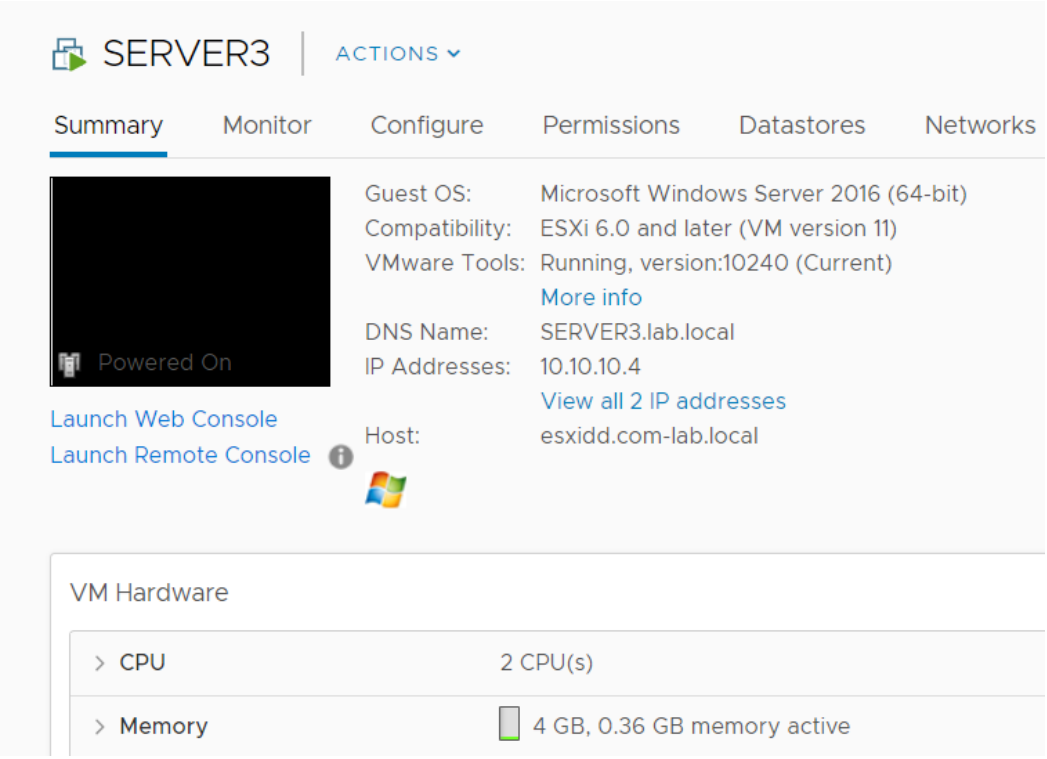
Slika 6.21 Kloniranje Template-a u SERVER1, SERVER2 i SERVER3



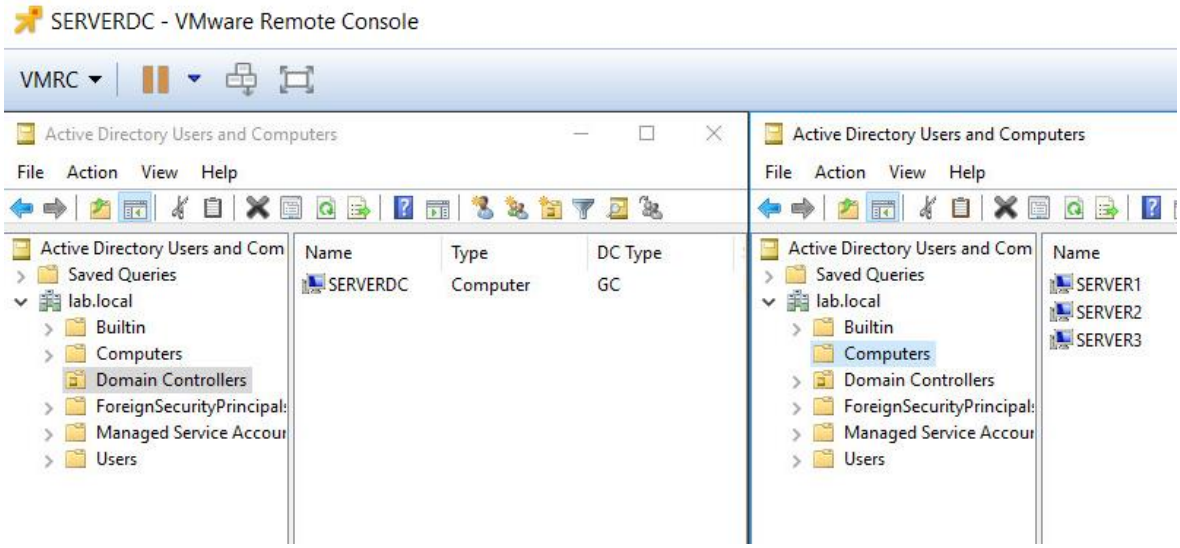
Slika 6.22 SERVER1



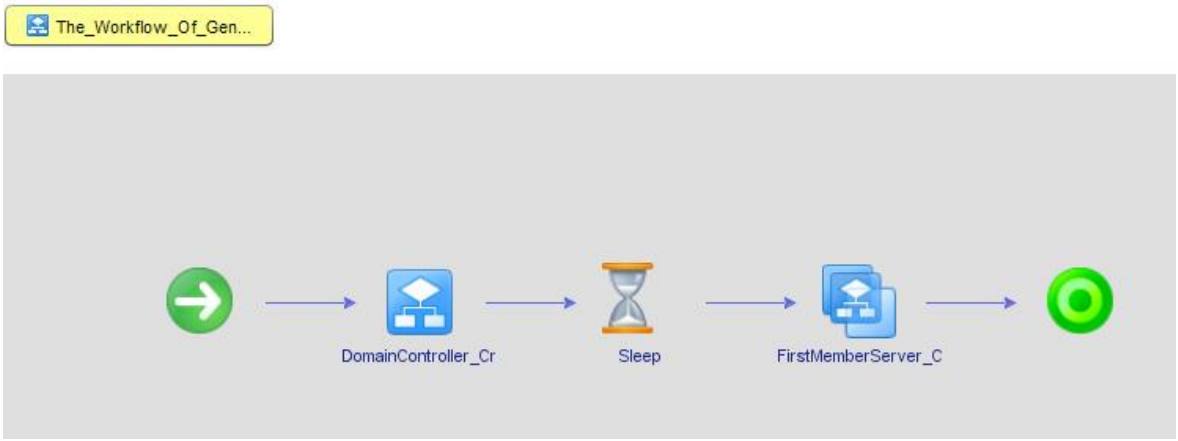
Slika 6.23 SERVER2



Slika 6.24 SERVER3



Slika 6.25 Finalno stanje unutar AD Users & Computers



Slika 6.26 Uspješno izvršen The_Workflow_Of_Genesis

7. vRO skriptiranje koristeći JavaScript

7.1. JavaScript kao jezgra Orchestrator-a

Esencija orchestratora je skriptni jezik imena JavaScript. Orchestrator vizualnom prezentacijom prikazuje razne objekte s kojima administrator može orkestrirati, dok on sam u pozadini vrlo pametno izvršava JavaScript. JavaScript ne treba biti kompajliran prije izvršavanja, odnosno ne zahtjeva kreaciju .exe datoteke. Orchestrator prepoznaje JavaScript kod, te kako bi znao kada je došao završetak pojedine linije, na kraj te linije stavljamo točku-zarez (;). Varijable su tekstualne linije koda koje će u postupku deklaracije u sebi sadržavati neku vrijednost koja može biti promjenjiva. Deklaracija se radi na sljedeći način:

```
var mojBroj = new Number ();
mojBroj = 10
```

Kôd 7.1 JavaScript deklaracija varijable

Svaka komanda unutar JavaScripta je osjetljiva na velika odnosno mala slova, tako da na taj segment također treba obraćati pažnju prilikom pisanja skripti. Isto vrijedi i za imena varijabli.

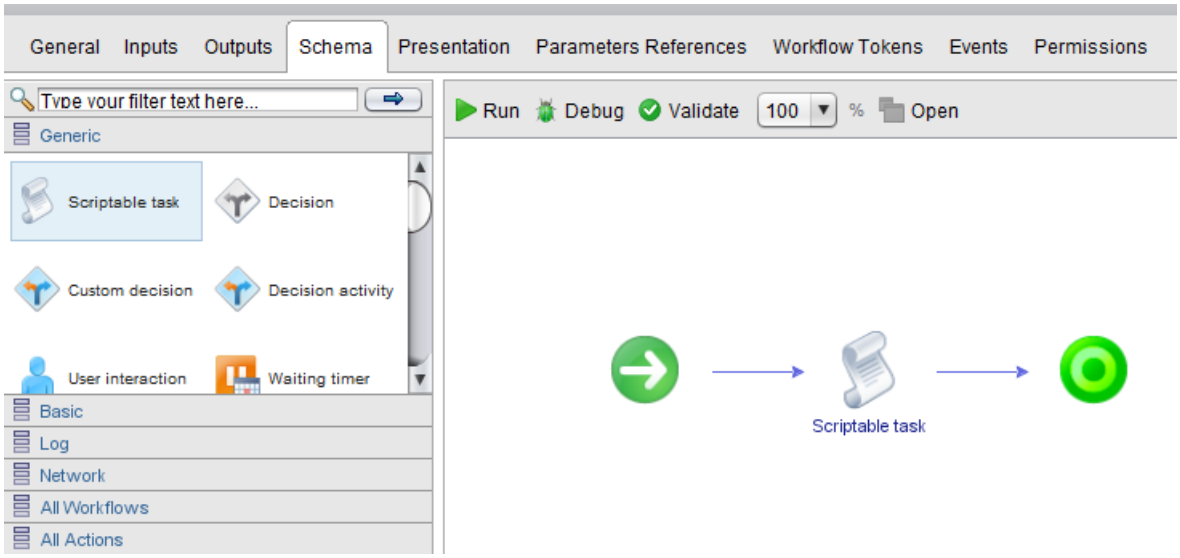
Komentari unutar skripte su linije teksta koje se ne izvršavaju odnosno preskaču, a orkestrator ih prepoznaje jer se ispred njih stavlja oznaka //. Komentari su tu s razlogom, a najbitniji od razloga je dokumentiranje skripte. Svaka linija koda se objašnjava putem komentara te pisac skripte točno zna zašto je tu liniju napisao i to baš na taj način. Isto tako, ukoliko tu skriptu naslijedi netko drugi, kada ju prvi puta pročita on će vrlo lako i uz minimalno utrošenog vremena znati što i na koji način ta skripta radi.

Objekti unutar workflowa u kojima orchestrator omogućava korištenje JavaScript skriptnog jezika nazivaju se skriptabilni zadaci. Na njih možemo gledati kao na elegantan način kako još detaljnije manipulirati workflow-om. Vrlo koristan element kod ove vrste zadataka je taj da s njima možemo obavljati logging i izvještavati točno što, kada i kako se nešto obavlja. [3]

[3] D. Langenhan; VMware vRealize Orchestrator Essentials; September 24, 2015; 9781785884245

7.2. Primjer skriptabilnog zadatka

Kreirati ćemo novi workflow, naziva JavaScriptExample, te ćemo unutar njega dodati objekt Scriptable task koji se nalazi u Generic tabu objekata.

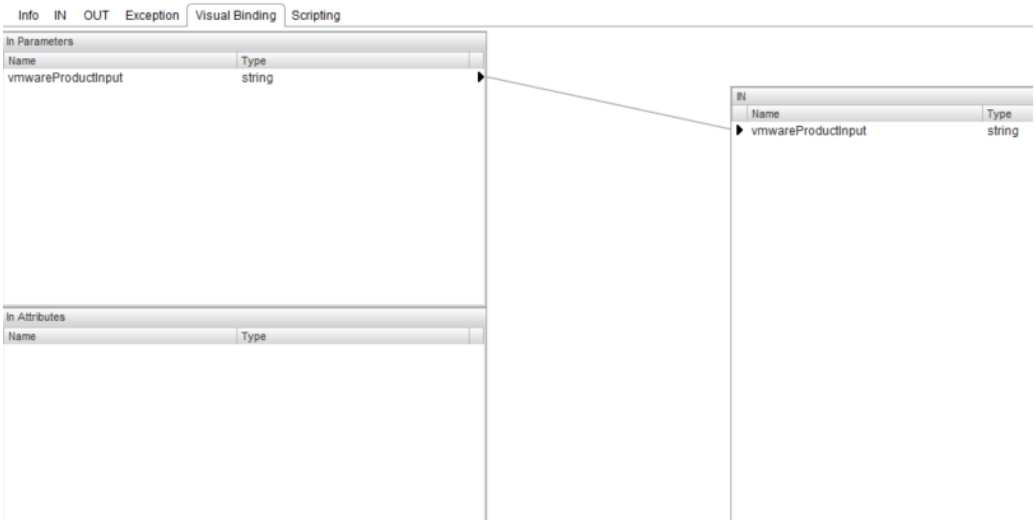


Slika 7.1 JavaScriptExample workflow

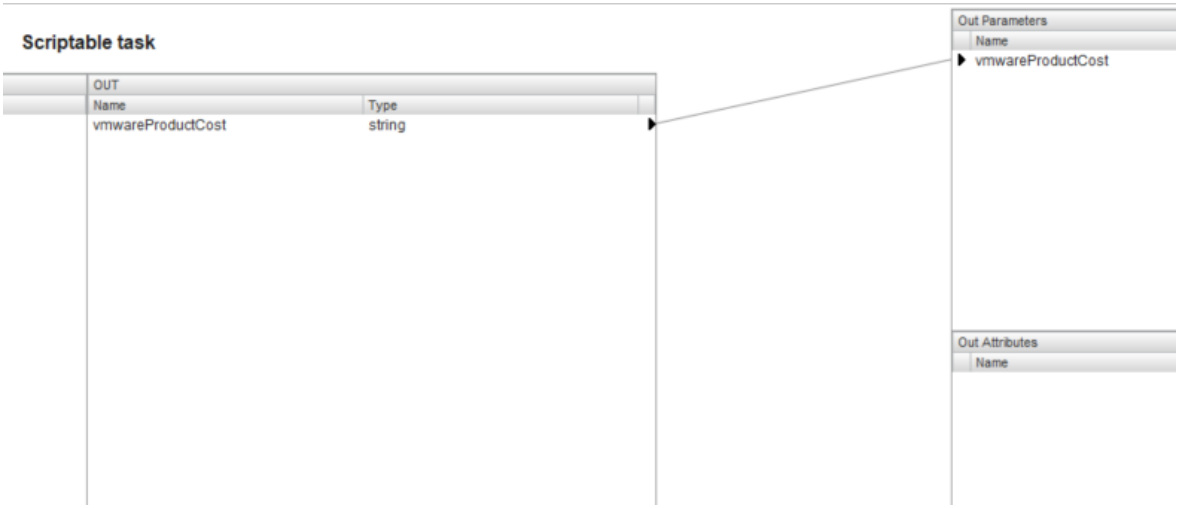
Kao i kada dodamo workflow unutar drugog workflowa, i ovdje nam se pojavljuju tabovi kao što su IN, OUT te Visual Binding. Jedan dodatni tab kojeg Scriptable task ima zove se Scripting i u njemu pišemo JavaScript kod.

Za primjer ćemo dodati atribut naziva vmwareProductAttribute, input parametar naziva vmwareProductInput i output parametar vmwareProductCost.

Unutar Visual Binding taba parametre i atribute spajamo na način prikazan na

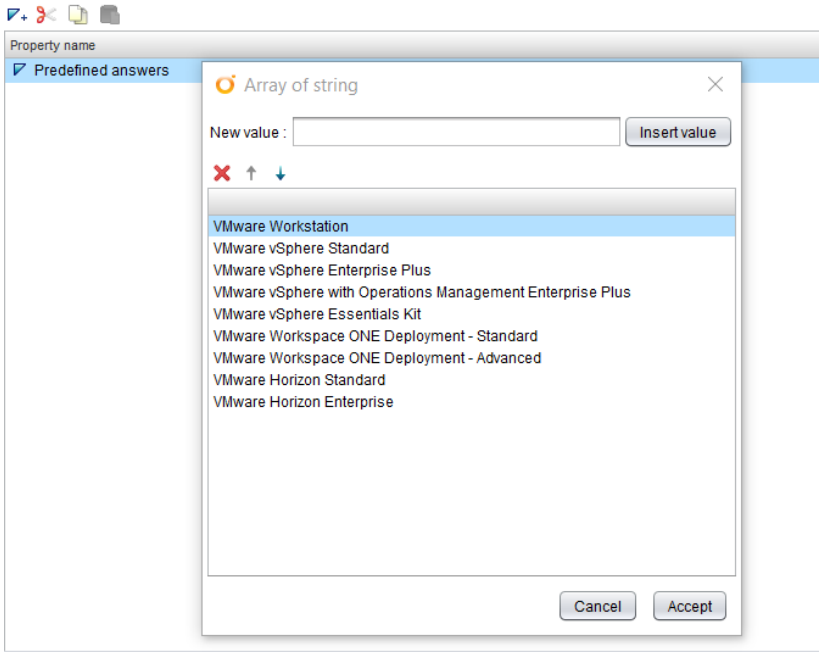


Slika 7.2 JavaScripExample In Parameters



Slika 7.3 JavaScriptExample Out Parameters

Sljedeći korak je definiranje input parametara koje korisnik može odabrati prilikom pokretanja ovo workflowa. Kako korisnik ne bi morao upisivati parametar, kreirati ćemo drop-down meni putem kojeg će korisnik odabrati ono što želi. U ovom primjeru će biti riječi o raznim VMware produktima. Prelaskom na Presentation tab, upisujemo description za input, pa ćemo tu staviti rečenicu: „Odaberi VMware produkt“. Ta linija teksta će se pojaviti iznad drop-down menija. U prozoru ispod description-a nalaze se postavke, te klikom na + znak odabiremo opciju *Predefined answers* kako bi mogli upisati produkte koje će korisnik odabrati. Primjer je prikazan na



Slika 7.4 Upisivanje VMware produkata

Povratkom na Schema tab, odabirom Scripting taska te klikom na Scripting tab, upisujemo JavaScript kod putem kojega ćemo svakom mogućem slučaju odnosno produktu dodijeliti njegovu cijenu. Kod je prikazan pod

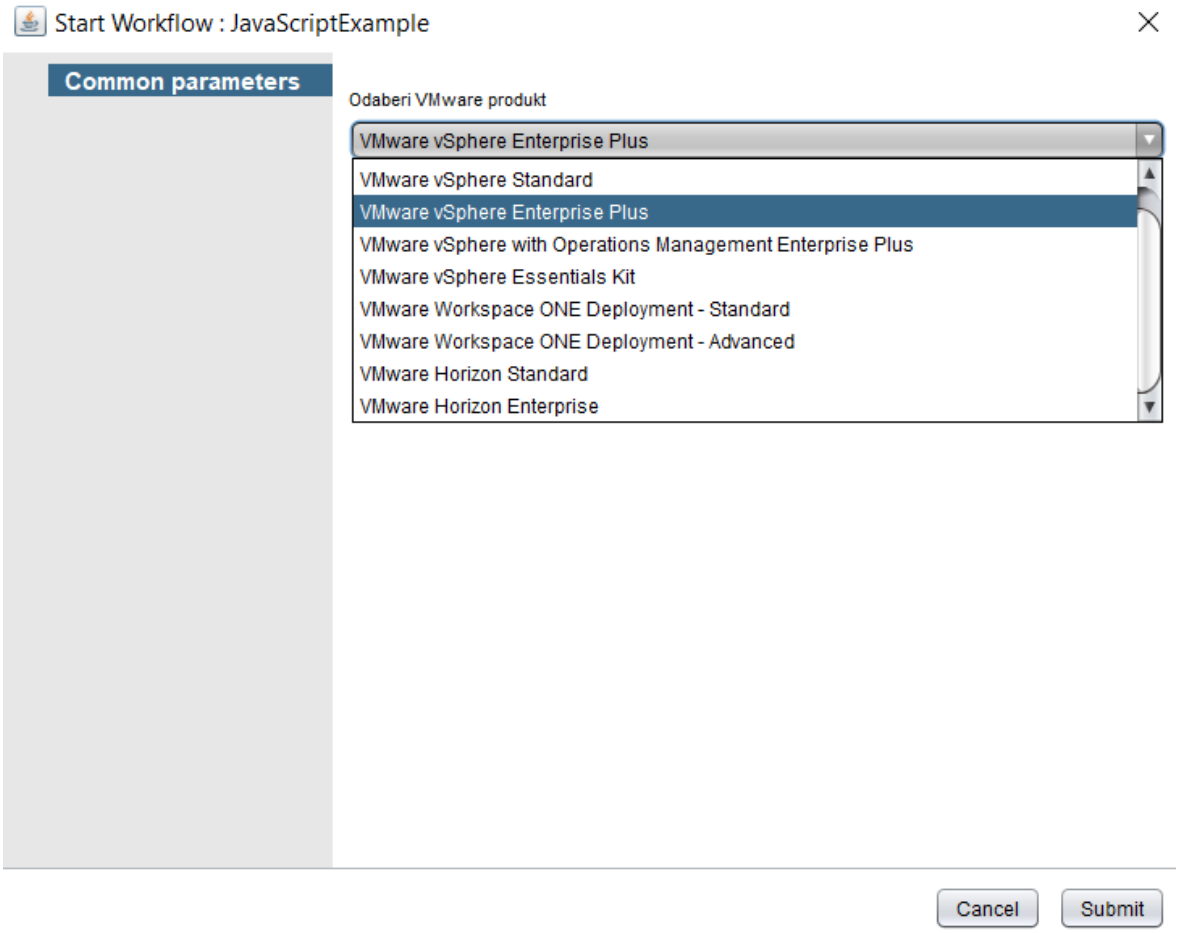
```
switch(vmwareProductInput) {
      case "VMware Workstation":
            vmwareProductCost = 280;
            break;
      case "VMware vSphere Standard":
            vmwareProductCost = 995;
            break;
      case "VMware vSphere Enterprise Plus":
            vmwareProductCost = 3495;
            break;
      case "VMware vSphere with Operations Management Enterprise Plus ":
            vmwareProductCost = 4525;
            break;
      case "VMware vSphere Essentials Kit":
            vmwareProductCost = 650;
            break;
      case "VMware Workspace ONE Deployment - Standard":
            vmwareProductCost = 9440;
            break;
      case "VMware Workspace ONE Deployment - Advanced":
            vmwareProductCost = 13600;
            break;
      case "VMware Horizon Standard":
            vmwareProductCost = 3565;
            break;
      case "VMware Horizon Enterprise":
            vmwareProductCost = 4994;
```

```
            break;
}

System.log ("Cijena " + vmwareProductInput + " produkta je: $" +
vmwareProductCost)
```

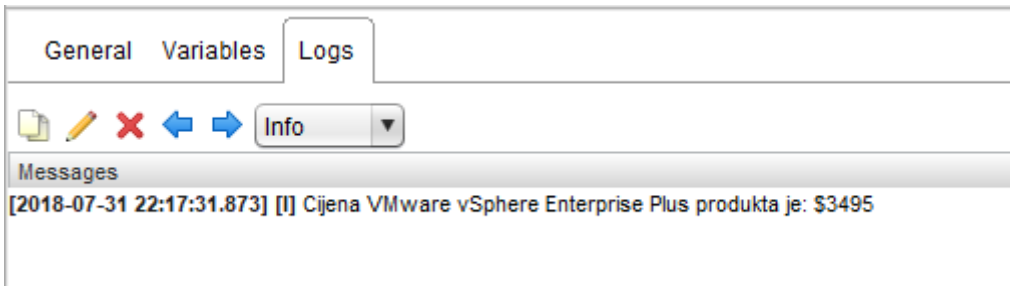
Kôd 7.2 JavaScript kod za dodjelu i ispis cijene produkta

Klikom na validate potvrđujemo da je workflow validan, te ga pokrećemo klikom na Run. Prikazuje se sljedeći meni:



Slika 7.5 Odabir VMware produkta, drop-down meni

Unutar drop-down menija odabiremo željeni proizvod te kliknemo na Submit. Workflow se izvršava te nam u Logu ispisuje vrijednost produkta, kao što je prikazano na



Slika 7.6 Cijena odabranog produkta

S ovim primjerom možemo vidjeti koliko je orkestrator svestran alat, te kada mu se pridoda segment pisanja vlastitih skripti, stvarno ne preostaje puno toga što sa tim alatom ne možemo izvršiti.

Zaključak

vRealize Orchestrator je zbilja nevjerojatan alat koji omogućava neizmjernu količinu sadržaja. Omogućava administratorima kreiranje kompleksnih worklfowa i akcija, njihovo međusobno povezivanje kao i kreiranje vlastitih skripti. Kombiniranjem tih elemenata može se ostvariti ono što bi se bez orchestratora ostvarivalo danima, mjesecima, godinama pa možda čak i nikad. U današnjem svijetu, gdje sve ide prema modelu as a Service, orchestrator je taj koji u pozadini radi sav posao i pomaže poduzećima ne samo opstati na tržištu, već njime i vladati.

Popis kratica

VM	*Virtual Machine*	Virtualna mašina
vRO	*vRealize Orchestrator*	Orkestrator
DC	*Domain Controller*	Domenski kontroler

Popis slika

Slika 2.1 VMware orchestrator plug-ins, D. Langenhan; VMware vRealize Orchestrator Essentials; September 24, 2015 3

Slika 3.1 Deployment OVF Template-a 8

Slika 3.2 Odabir lokacije Deploymenta 9

Slika 3.3 Odabir fizičkog ESXi hosta ili cluster-a 9

Slika 3.4 Sumarni prikaz detalja 10

Slika 3.5 Odabir datastore-a 10

Slika 3.6 Odabir mreže 11

Slika 3.7 Postavljanje hostname-a, gateway-a i DNS-a 11

Slika 3.8 Postavljanje DNS suffixa i IP adrese orkestratora 12

Slika 3.9 Provjera dostupnosti virtualne mašine 12

Slika 3.10 Orkestrator web sučelje 13

Slika 3.11 Konfiguracija SSO autentikacije 14

Slika 3.12 Korisničko sučelje orkestratora 14

Slika 4.1 Postavke vCentra 15

Slika 4.2 Postavke vCenter korisnika 16

Slika 4.3 Konfiguracija orkestratora kao ekstenzije 16

Slika 4.4 Pristup orkestratoru iz vCentra 17

Slika 4.5 Orkestrator integriran u vCenter 17

Slika 4.6 Upravljanje pluginovima 18

Slika 4.7 Uspješno instaliran Horizon plugin 19

Slika 5.1 Početni izgled orkestratora 20

Slika 5.2 Ikone u Design pogledu 21

Slika 5.3 Workflow tab 22

Slika 5.4 Primjer scheme workflowa 23
Slika 5.5 The General tab 24
Slika 5.6 The Inputs tab 24
Slika 5.7 The Outputs tab 25
Slika 5.8 The Presentation tab 25
Slika 5.9 The Parameters References tab 26
Slika 5.10 The Events tab 27
Slika 5.11 The Permissions tab 27
Slika 5.12 Actions tab 27
Slika 5.13 Detaljan prikaz createSnapshot akcije 28
Slika 6.1 Template VM 30
Slika 6.2 Dodavanje mape naziva Zavrsni 31
Slika 6.3 Kopiranje workflow-a 31
Slika 6.4 Schema tab praznog Domain_Controller_Creation workflowa 32
Slika 6.5 Umetanje 2 workflowa 33
Slika 6.6 Umetanje Sleep objekta 33
Slika 6.7 Linked_Clone_Creation parametri 34
Slika 6.8 Linked_Clone_Creation In Attributes 35
Slika 6.9 SleepTime 36
Slika 6.10 Dodavanje atributa 37
Slika 6.11 Domain Controller VM atribut 38
Slika 6.12 Povezivanje out parametara 38
Slika 6.13 Domain_Controller_Script konačni rezultat 41
Slika 6.14 Domain_Controller_Creation workflow validation 41
Slika 6.15 Izgled 3 nova workflowa 42
Slika 6.16 The_Workflow_Of_Genesis 47

Slika 6.17 The_Workflow_Of_Genesis pokretanje 48

Slika 6.18 Kloniranje Template-a u SERVERDC 48

Slika 6.19 SERVERDC 49

Slika 6.20 Automatsko pokretanje skripte za promoviranje SERVERDC u Domain Controller 49

Slika 6.21 Kloniranje Template-a u SERVER1, SERVER2 i SERVER3 50

Slika 6.22 SERVER1 50

Slika 6.23 SERVER2 51

Slika 6.24 SERVER3 51

Slika 6.25 Finalno stanje unutar AD Users & Computers 52

Slika 6.26 Uspješno izvršen The_Workflow_Of_Genesis 52

Slika 7.1 JavaScriptExample workflow 54

Slika 7.2 JavaScripExample In Parameters 54

Slika 7.3 JavaScriptExample Out Parameters 55

Slika 7.4 Upisivanje VMware produkata 55

Slika 7.5 Odabir VMware produkta, drop-down meni 57

Slika 7.6 Cijena odabranog produkta 58

Popis tablica

Tablica 2.1 Matrica kompatibilnosti 3

Tablica 2.2. Limitacije orkestratora 4

Tablica 3.1 Zahtjevi orkestratora 5

Tablica 3.2 Komande za provjeru 6

Tablica 3.3 Portovi za pristup orkestratoru 6

Tablica 3.4 Orkestratorovi eksterni portovi 7

Tablica 6.1 Linked_Clone_Creation najvažniji atributi 35

Tablica 6.2 Domain_Controller_Script parametri 39

Tablica 6.3 Linked_Clone_Creation workflow atributi unutar First_Member_Server_Creation workflowa 43

Tablica 6.4 Linked_Clone_Creation workflow atributi unutar Second_Member_Server_Creation workflowa 43

Tablica 6.5 Linked_Clone_Creation workflow atributi unutar Third_Member_Server_Creation workflowa 44

Tablica 6.6 Four main workflows 46

Popis kôdova

Kôd 6.1 Powershell skripta za promoviranje Domain Controllera 40

Kôd 6.2 cmd skripta za dodavanje servera u lab.local domenu 46

Kôd 7.1 JavaScript deklaracija varijable 53

Kôd 7.2 JavaScript kod za dodjelu i ispis cijene produkta 57

Literatura

[1] D. Langenhan; VMware vRealize Orchestrator Essentials; September 24, 2015; 9781785884245

[2] D. Langenhan; VMware vRealize Orchestrator Cookbook - Second Edition; November 11, 2016; 9781786462787

Algebra
visoka škola za
primijenjeno računarstvo

VMware vRealize Orchestrator Automatizacija

Pristupnik: Danijel Dmitrović, 0321004954

Mentor: dipl. ing. Vedran Dakić