

MICROSOFT AZURE IMPLEMENTACIJA

MICROSOFT AZURE IMPLEMENTATION

Jozo Slejko¹, Ognjen Mitrović²

¹Span d.o.o.

²Tehničko veleučilište u Zagrebu

Sažetak

U članku su dati modeli implementacije Azure IaaS (e. Infrastructure as a Service) resursa koje pruža Microsoft Azure platforma. Nastavno na modele, detaljnije se analizira Azure Resource Manager (ARM) API (e. Application programming interface) i njegove karakteristike i prednosti. Predstavljena je arhitektura ARM API-ja i njegovi sastavni dijelovi. Navedeni su alati koji služe za implementaciju ARM resursa, ARM PowerShell i Azure Cross-Platform (xPlat) CLI (e. command line interface) alat. Među ostalim načinima implementacije resursa kroz ARM API, ARM JSON (e. JavaScript Object Notation) predlošci mogu se koristiti za deklarativnu implementaciju Azure resursnih grupa. Prikazan je primjer implementacije Azure resursa kroz ARM Visual Studio projekt koji koristi JSON predloške.

Ključne riječi: Računarstvo u oblaku, Azure, IaaS, Azure Resource Manager (ARM)

Abstract

This article describes the implementation models of Azure IaaS (Infrastructure as a Service) resources provided by Microsoft Azure platform. Following on the models, it looks at the Azure Resource Manager (ARM) API (Application programming interface) and its features and benefits. Architecture of ARM API and its components is presented in the article, with tools that are used to deploy ARM resources, ARM PowerShell and Azure Cross-Platform (Xplat) CLI (command line interface) tool.

Among other ways of resource implementation through ARM API, ARM JSON (JavaScript Object Notation) templates can be used to declarative deploy Azure resource groups. An example of Azure resource implementation through the use of ARM Visual Studio project that uses JSON templates will be shown.

Keywords: Cloud computing, Azure, IaaS, Azure Resource Manager (ARM)

1. Uvod

1. Introduction

Microsoft Azure je Microsoftova platforma računarstva u javom oblaku, koja pruža široku i rastuću paletu usluga koje se mogu koristiti bez kupovine ikakvog hardvera i često čine temeljne elemente računarstva u oblaku.

2 Azure modeli implementacije

2. Azure implementation models

Trenutno postoje dva modela implementacije resursa u Azureu. Resource Manager model implementacije je različit od klasičnog modela implementacije, te ta dva modela nisu potpuno kompatibilna. [1]

Azure je izvorno nudio samo klasični model implementacije. U tom modelu je svaki resurs nezavisan i nije bilo načina za grupiranje povezanih resursa zajedno. U 2014. godini Azure predstavlja Resource Manager koji donosi koncept grupe resursa. Resursna grupa je spremnik za resurse koji dijele zajednički životni ciklus.

Resource Manager model implementacije nudi nekoliko prednosti:

- Implementacija, upravljanje i nadzor svih servisa nekog rješenja kao jedne grupe.
- Opetovano implementiranje rješenja tijekom životnog vijeka uz sigurnost da su resursi implementirani na konzistentan način.
- Primjenjivanje politike kontrole pristupa nad svim resursima u grupi resursa gdje se te politike automatski primjenjuju na novo dodane resurse.
- Primjenjivanje oznake resursa za logično organiziranje svih resursa u Azure pretplati.
- Korištenje JavaScript objektne notacije (JSON) za definiranje infrastrukture rješenja. JSON datoteka je poznata kao Resource Manager predložak.
- Definiranje ovisnosti među resursima.

3 Azure upravitelj resursa

3.1 Azure Resource Manager

U ranim danima Azurea, Microsoft je razvio REST (e. Representational state transfer) API Azure Service Management (ASM) sučelje u Azure javnom oblaku zvanom Red Dog Front End (RD FE). Ovaj imperativni API je pogonjen XML-om (e. Extensible Markup Language) i omogućuje uspostavljanje resursa Azure oblaka kroz Azure Management portal. [2]

Zbog nekih problema oko performansi, paralelizma i korisničkog iskustva, Microsoft je razvio novi API pod nazivom Azure Resource Manager (ARM) API.

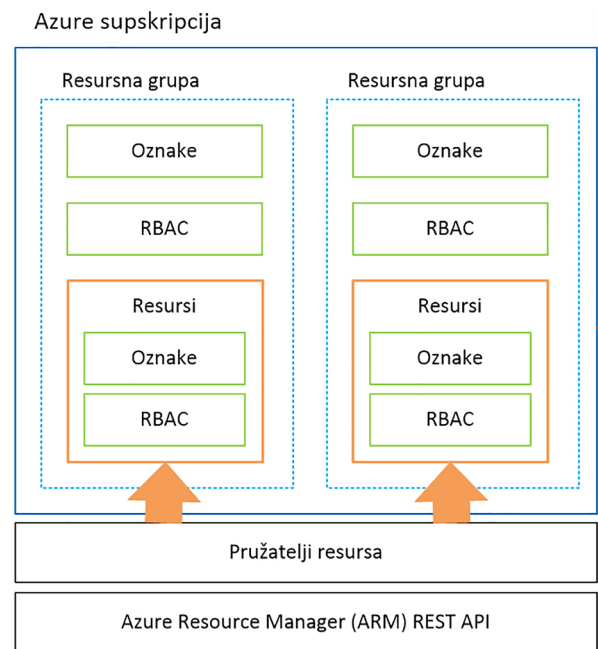
Neke od ključnih prednosti ARM API-ja su:

- Novo grafičko sučelje Azure Portala
- Deklarativno sučelje (ARM JSON predlošci) bazirano na JSON-u
- Podrška za oznake resursa (metapodaci)
- Zaključavanje resursa kako bi se spriječila slučajna izmjena ili brisanje
- Povezivanje resursa
- Role Based Access Control (RBAC) ograničavanje korisničke kontrole nad resursima
- Politike resursa za ograničavanje zahtjeva poslanim na API

3.1 Arhitektura

3.1 Architecture

Slika 1 je dijagram koji pokazuje odnose između objekata u Azure Resource Manager (ARM) API-ju.



Slika 1: ARM odnos objekata [3]

Figure 1: ARM object relation [3]

3.2 Pružatelj resursa

3.2 Resource Provider

Svaki resurs u oblaku je izložen ARM API-ju kroz sloj poznat kao „Pružatelj resursa“ (Resource Provider). Svaki Resource Provider izlaže povezani niz funkcionalnosti kao pojedinačne vrste resursa. U nekim slučajevima se pružatelj resursa mora „registrirati“ s Azure pretplatom, prije nego li se tipovi resursa koje on sadrži mogu koristiti.

3.3 Resursna grupa

3.3 Resource group

ARM resursna grupa je spremnik izrađen u Azure regiji koju odredi korisnik, u kojoj su smješteni resursi Azure oblaka. Grupe resursa su općenito namijenjene za održavanje resursa oblaka koji dijele zajednički životni ciklus, npr. skup PaaS (e. Platform as a Service) usluga jedne aplikacije, ali to nije uvjet.

Postavljanje resursa u resursne grupe moguće je korištenjem imperativnog ARM API-ja ili putem ARM JSON predložaka. Resursne grupe mogu imati dodijeljene oznake.

Resursne grupe dio su nove funkcionalnosti pravila kontrole pristupa temeljena na ulogama (Azure Role-based Access Control, RBAC).

3.4 Resurs

3.4 Resource

U kontekstu ARM API-ja, resurs je instanca resursnog tipa (Resource Type) koja je smještena u ARM resursnu grupu, te je generički prikaz resursa Azure oblaka (npr. Storage Blob, Virtual Machine, Web App).

Resursima se može upravljati kao jednostavnim JSON objektima, ali oni moraju biti u skladu s određenim skupom zajedničkih osobina, kako bi ih ARM REST API ispravno prepoznao. Ovo su zajedničke JSON osobine: [3]

- API verzija – API verzija koja se koristi za upravljanje resursima.
- Naziv – Naziv resursa unutar resursne grupe.
- Vrsta – Pružatelj resursa i vrsta resursa (npr. Microsoft.Compute/virtualMachines).
- Mjesto – Azure regija (lokacija) u kojoj je resurs.
- Svojstva – Skup osobina koji sadrži definicije resursa. Obavezna i izborna JSON svojstva su drugačija za svaki tip resursa.

Uz mogućnost upravljanja resursima Azure oblaka pomoću ARM predložaka, resursima se također može upravljati pomoću imperativnih naredbi. ARM PowerShell modul i cross-platform (xPlat) CLI alat uključuju naredbe za operacije stvaranja, čitanja, ažuriranja i brisanja za većinu vrsta resursa Azure oblaka.

4 ARM PowerShell

4 ARM PowerShell

ARM PowerShell modul dizajniran je za implementaciju i upravljanje resursima Azure pretplate pomoću ARM API-ja.

5 Azure Cross-Platform (xPlat) CLI

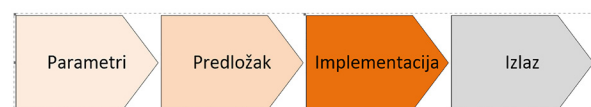
5 Azure Cross-Platform (xPlat) CLI

Ako se koristi Windows ili neki drugi sustav, ne postoji ili se ne želi koristiti PowerShell, mnoge se naredbe za upravljanje Azure pretplatom mogu obaviti pomoću Azure Cross-Platform (xPlat) CLI alata. Azure xPlat CLI je Node.js aplikacija koja komunicira sa oba Azure API-ja, ASM kao i ARM.

6 ARM JSON predlošci

6 ARM JSON templates

ARM API koristi koncept „JSON predložaka“ (tzv. „ARM predlošci“) za implementaciju i upravljanje resursima oblaka kroz deklarativnu perspektivu. ARM predložak se implementira u grupu resursa pomoću inkrementalnog ili potpunog načina.



Slika 2: Implementacija JSON predlošcima [3]

Figure 2: JSON template implementation [3]

6.1 Struktura

6.1 Structure

Svaka datoteka ARM predložka mora slijediti određenu strukturu kako bi se uspješno implementirala na ARM REST API. Postoje dva korijenska JSON svojstva koja moraju biti deklarirana, \$schema i contentVersion.

Struktura glavnog tijela ARM JSON predložka sadrži sljedeće četiri glavne sekcije: ulazni parametri, varijable, resursi i izlazni parametri.

Da se ubrza uspostavljanje resursa oblaka, Microsoft je osigurao repozitorij ARM predložaka na GitHub projektu zvanom „QuickStart templates“. [3] Prilikom izrade ARM predloška, najbolje je koristiti postojeće ARM predloške i kopirati dijelove JSON-a koji su potrebni.

7 Implementacija

7 Implementation

Postoji nekoliko različitih načina implementacije resursa pomoću ARM API-ja:

- „Deploy to Azure“ tipka (preko Azure Portala)
- Korištenje ARM PowerShella
- „Azure Resource Group“ Visual Studio Project predložak
- Azure Cross-Platform (xPlat) CLI alat
- Direktno korištenje ARM REST API

7.1 „Deploy to Azure“ tipka

7.1 „Deploy to Azure“ button

„Deploy to Azure“ tipka se nalazi na README.md stranicama na Azure „Quickstart Templates“ GitHub repozitoriju. Ova tipka omogućuje brzo uspostavljanje ARM predložaka izravno iz izvora.

Nakon što se klikne tipka, korisnik se preusmjerava na Microsoft Azure Portal, gdje se traži unos ulaznih parametara za JSON predložak.

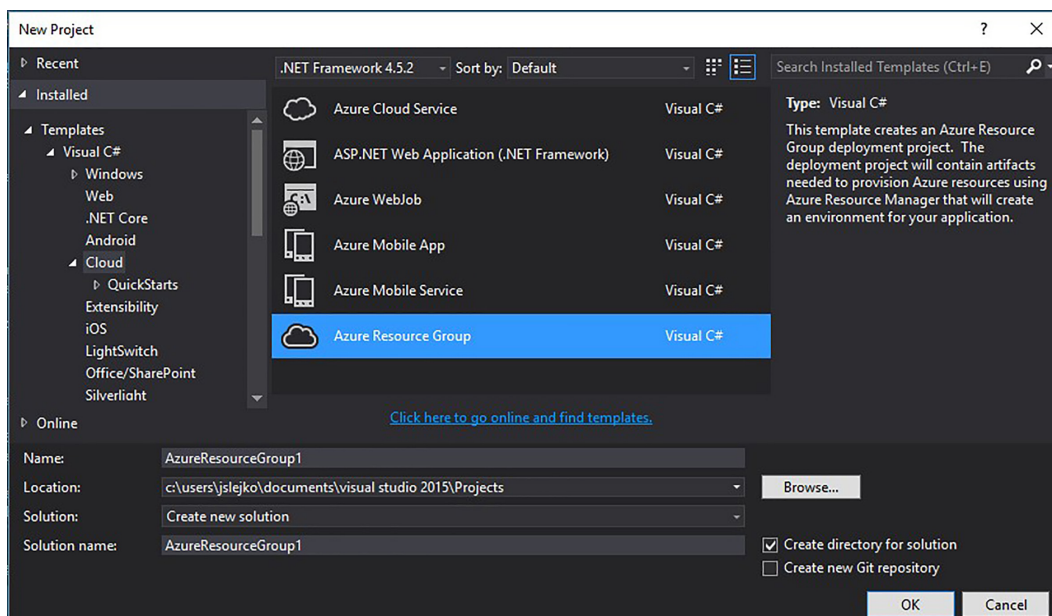
7.2 Implementacija ARM predložaka PowerShellom

7.2 ARM template implementation with PowerShell

New-AzureRmResourceGroup je ARM PowerShell naredba koja može stvoriti i implementirati u resursnu grupu.

Naredba nudi nekoliko bitnih parametara za implementaciju:

- Name – naziv Azure resursne grupe koja se kreira
- Location – Azure regija iz koje se upravlja Azure resursnom grupom
- TemplateUri – URL (e. uniform resource locator) do datoteke ARM JSON predloška koji se implementira
- TemplateParameterObject – PowerShell Hashtable koji sadrži parove imena parametara i vrijednosti koji se prenose u ARM JSON predložak



Slika 3: Visual Studio Azure Resource Group predložak

Figure 3: Visual Studio Azure Resource Group template

7.3 Implementacija IaaS resursa JSON predloščima

7.3 Implementation of IaaS resource with JSON templates

Implementacija ARM resursnih grupa je moguća kroz Microsoft Visual Studio integrirano razvojno okruženje.

Verzije VS-a (Visual Studio) koje podržavaju rad sa upraviteljem resursa (Resource Manager) su 2012, 2013, 2015 i novije. Dodatno je potrebno instalirati Azure SDK (e. software deployment kit) paket. [4]

Visual Studio nudi projektni predložak za kreiranje implementacije Azure resursne grupe (Slika 3). Na primjer, moguće je koristeći kod implementirati okruženje koji ima virtualnu mrežu, neki broj virtualnih mašina i računara za pohranu, razdjeljivač opterećenja (load balancer), SQL server i drugo.

Projekt omogućava kreiranje svih potrebnih specificiranih resursa u jednoj operaciji s mogućnošću ponavljanja implementacije.

7.4 Visual Studio projekt

7.4 Visual Studio project

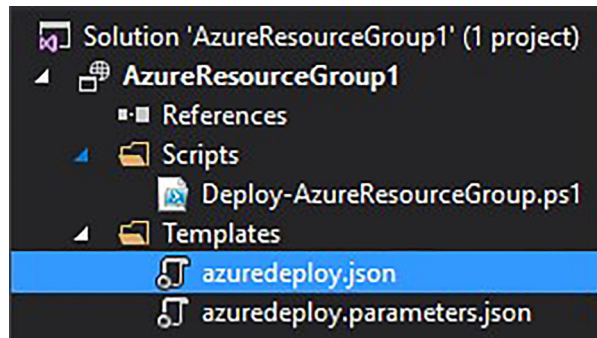
Nakon kreiranja novog projekta, Azure SDK omogućuje biranje vrste predloška npr. Windows Virtual Machine, Web App, SQL, Ubuntu Server, Storage itd.

Kreirati ćemo prazan predložak (Blank Template).

7.4.1 Struktura projekta

7.4.1 Project structure

Struktura novog praznog projekt ima već nekoliko kreiranih datoteka (Slika 4).



Slika 4: Struktura projekta

Figure 4: Project structure

Deploy-AzureResourceGroup Powershell skripta zove Powershell naredbe koje služe za kreiranje i implementaciju resursnih grupa koristeći JSON datoteke predložaka. [5]

Azuredeploy.json datoteka je JSON predložak u kojem se deklariraju resursi koji se dodaju ili kreiraju kroz implementacije resursne grupe. Također definira ovisnosti između resursa kako bi upravitelj resursa implementirao grupu u pravilnom redosljedju. Ona kod praznog templatea ima samo definiranu shemu bez ikakvih resursa, parametara i varijabli (Slika 5).

Struktura datoteke je podijeljena na tri dijela: parametri, varijable i resursi.

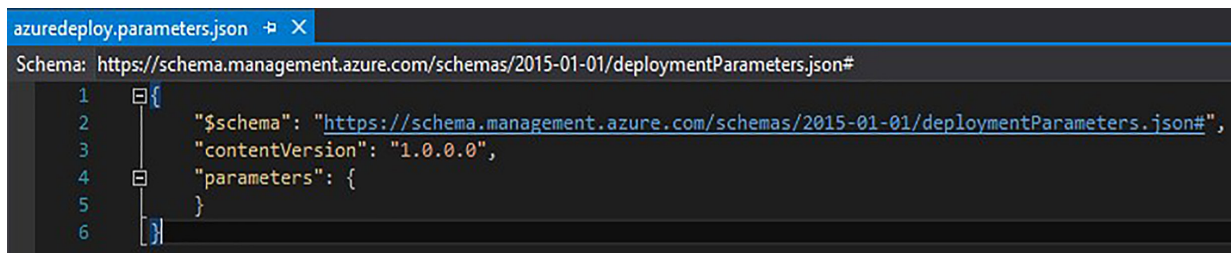
```

azuredeploy.json
Schema: https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#
1  {
2    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3    "contentVersion": "1.0.0.0",
4    "parameters": {
5    },
6    "variables": {
7    },
8    "resources": [
9    ],
10   "outputs": {
11   }
12  }

```

Slika 5: azuredeploy.json datoteka

Figure 5: azuredeploy.json file



Slika 6: azuredeploy.parameters.json datoteka

Figure 6: azuredeploy.parameters.json file

Odjeljak parametara sadrži parametre koji se koriste u predlošku. Vrijednosti parametara se mogu specificirati unutar JSON datoteke ili kroz portal.

Odjeljak varijabli sadrži varijable koje se koriste interno unutar predloška. Korištenjem varijabli izbjegava se kodiranje vrijednosti.

Odjeljak resursa definira sve resurse koje treba dodati ili ažurirati tijekom implementacije.

U odjeljku izlaza mogu se po izboru dobiti povratne vrijednosti implementacije. Na primjer, može se dobiti javna IP adresa virtualnog stroja nakon što se kreira.

Azuredeploy.parameters.json (Slika 6) je datoteka koja specificira vrijednosti ulaznih parametara koji se koriste u glavnom JSON predlošku (azuredeploy.json) za implementaciju resursne grupe.

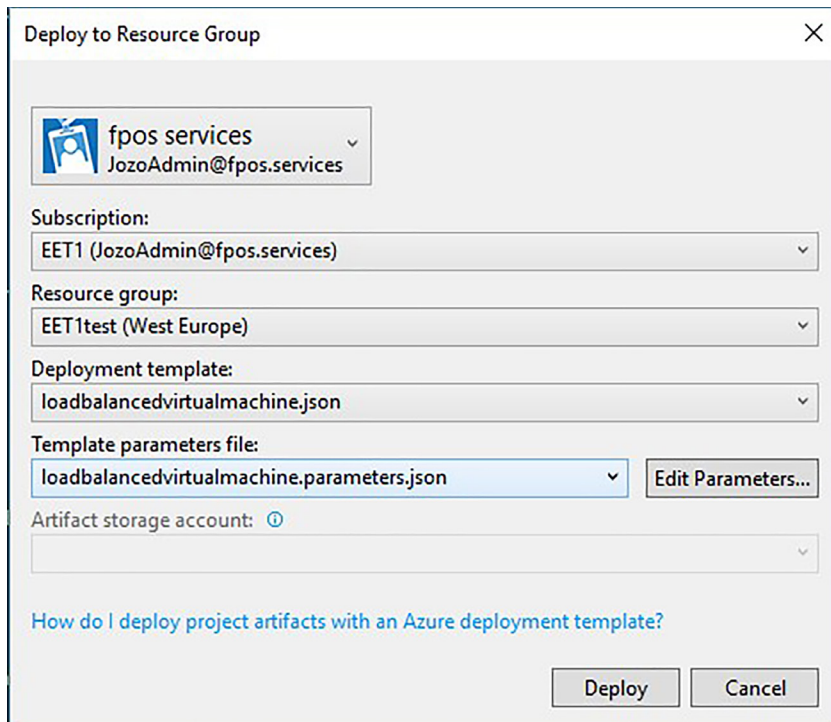
8 Primjer projekta implementacije

8 Implementation project example

JSON datoteka u ovom primjeru deklarativno implementira Windows Server 2012 R2 poslužitelj sa SQL Server 2016 Standard licencom.

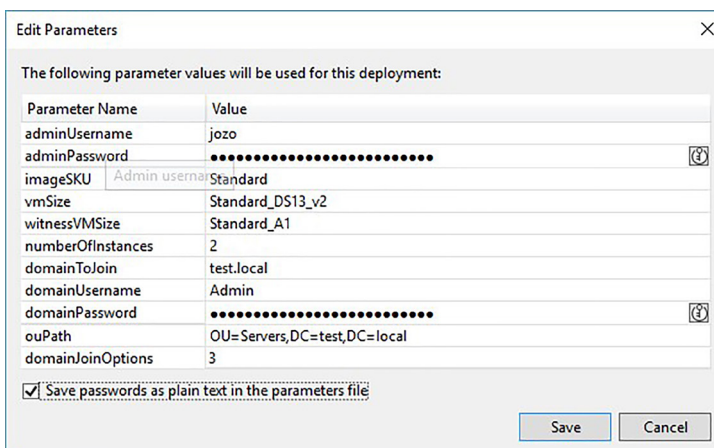
```
{
  "$schema": "https://
schema.management.azure.
com/schemas/2015-01-01/
deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "adminUsername": {
      "type": "string",
```

```
      "metadata": {
        "description": "Admin
korisničko ime"
      }
    },
    "adminPassword": {
      "type": "securestring",
      "metadata": {
        "description": "Admin
lozinka"
      }
    }
  },
  "variables": {
    "sqlVMName": "test-sql1",
    "vmSize": "Standard_DS13_v2",
    "imagePublisher":
"MicrosoftSQLServer",
    "imageOffer": "SQL2016-WS2012R2",
    "imageSKU": "Standard",
    "sqlStorageName":
"testeuwsawsq11",
    "SQLNicName": "test-wsql1-nic",
  },
  "resources": [
    {
      "apiVersion": "2015-06-15",
      "type": "Microsoft.Compute/
virtualMachines",
      "name":
"[variables('sqlVMName')]",
      "location": "[resourceGroup().
location]",
      "tags": {
        "displayName": "SQL"
      },
      "dependsOn": [
        "[concat('Microsoft.
Storage/storageAccounts/',
```

Slika 8: Prozor za implementaciju resursne grupe

Figure 8: Resource group implementation window



Slika 9: Prozor za konfiguraciju vrijednosti parametara

Figure 9: Parameter configuration window

Ukoliko se želi promijeniti neki dio konfiguracije resursne grupe, samo se ažurira predložak i napravi se ponovna implementacija koja izmijeni potrebne stvari na resursnoj grupi.

9 Zaključak

9 Conclusion

Jedna od najkvalitetnijih platformi javnog oblaka ona je Microsofta, Azure platforma. U više kategorija po Gartneru trenutno spada u leader kvadrant. Azure pruža SaaS (e. software as a service), PaaS i IaaS usluge, te podržava široku lepezu Microsoftovih i rješenja ostalih proizvođača, ujedno se fokusirajući na sigurnost, kvalitetu i raspoloživost usluge bilo gdje u svijetu.

Azure IaaS usluge zaista su vrhunske kvalitete i mogućnosti, postavljajući ljestvicu kvalitete visoko iznad konkurencije. Rješenja infrastrukture uspostavljaju se ili ručno kroz grafički internet portal ili u slučaju velikih implementacija kroz JSON predloške, metoda zvana „Infrastructure as a Code“.

Budućnost oblaka, bilo javnog ili privatnog, je svijetla, prednosti su velike, a mane se brzo rješavaju pošto cijela IT zajednica sudjeluje u formiranju budućih usluga oblaka. Vrijeme računalstva kao komunalije je već počelo, samo je pitanje vremena kada će to postati standard.

6. REFERENCE

6. REFERENCES

- [1.] FitzMacken, Tom: Azure Resource Manager vs. classic deployment: Understand deployment models and the state of your resources, <https://azure.microsoft.com/en-us/documentation/articles/resource-manager-deployment-model>, 29. srpnja 2016.
- [2.] FitzMacken, Tom: Azure Resource Manager overview, <https://azure.microsoft.com/en-us/documentation/articles/resource-group-overview>, 18. kolovoza 2016.
- [3.] Sullivan, Trevor: Microsoft Azure Resource Manager, <https://trevorsullivan.net/microsoft-azure/microsoft-azure-resource-manager>, 2015.
- [4.] Sandbu, Marius: Getting started with Azure Resource Manager and visual studio, <https://msandbu.wordpress.com/2015/05/22/getting-started-with-azure-resource-manager-and-visual-studio/>, 22. svibnja 2015.
- [5.] FitzMacken, Tom: Creating and deploying Azure resource groups through Visual Studio, <https://azure.microsoft.com/en-us/documentation/articles/vs-azure-tools-resource-groups-deployment-projects-create-deploy>, 03. kolovoza 2016.

AUTORI · AUTHORS



Jozo Slejko

Jozo Slejko je rođen 1983. godine u Zagrebu, gdje završava osnovnu školu i prirodoslovnu matematičku gimnaziju. Radi kao projektant rješenja u firmi Span d.o.o. gdje se primarno

bavi Microsoft infrastrukturnim rješenjima u oblaku.

Korespodencija

jozo.slejko@gmail.com



Ognjen Mitrović

Rođen je 18.05.1980. u Zagrebu. Diplomirao je na Tehničkom veleučilištu u Zagrebu.

Prije dolaska na Tehničko veleučilište u Zagrebu radio je niz godina u IT sektoru na

poslovima systemske i mrežne administracije, konzultacije korisnika i unaprjeđenja IT sustava te kao predavač službenih Microsoft edukacija. Područja od osobitog interesa uključuju rad s virtualizacijskim tehnologijama te dizajn IT infrastrukture i implementacija visoko dostupnih servisa temeljenih na Microsoft tehnologijama. Posjeduje niz Industrijskih IT certifikata (Cisco, Microsoft, Citrix).

Korespodencija

ognjen.mitrovic@tvz.hr