#GlobalAzure

#CloudGenVerona

@cloudgen_verona

Thanks to all the sponsors

PREMIUM SPONSOR

4WARD

KIRATECH®
WE DEVOPS IT

BASIC SPONSOR

CODICEPLASTICO

dgroove
Tecnologia emozionale.

CloudGen
ominidi verso l'evoluzione

TOPIC

Funziona?! Allora non toccarlo,

ovvero l'analisi d'infrastruttura in esercizio.

Who I am

GIULIANO LATINI

@giulianolatini

github.com/giulianolatini

linkedin.com/in/giulianolatini/

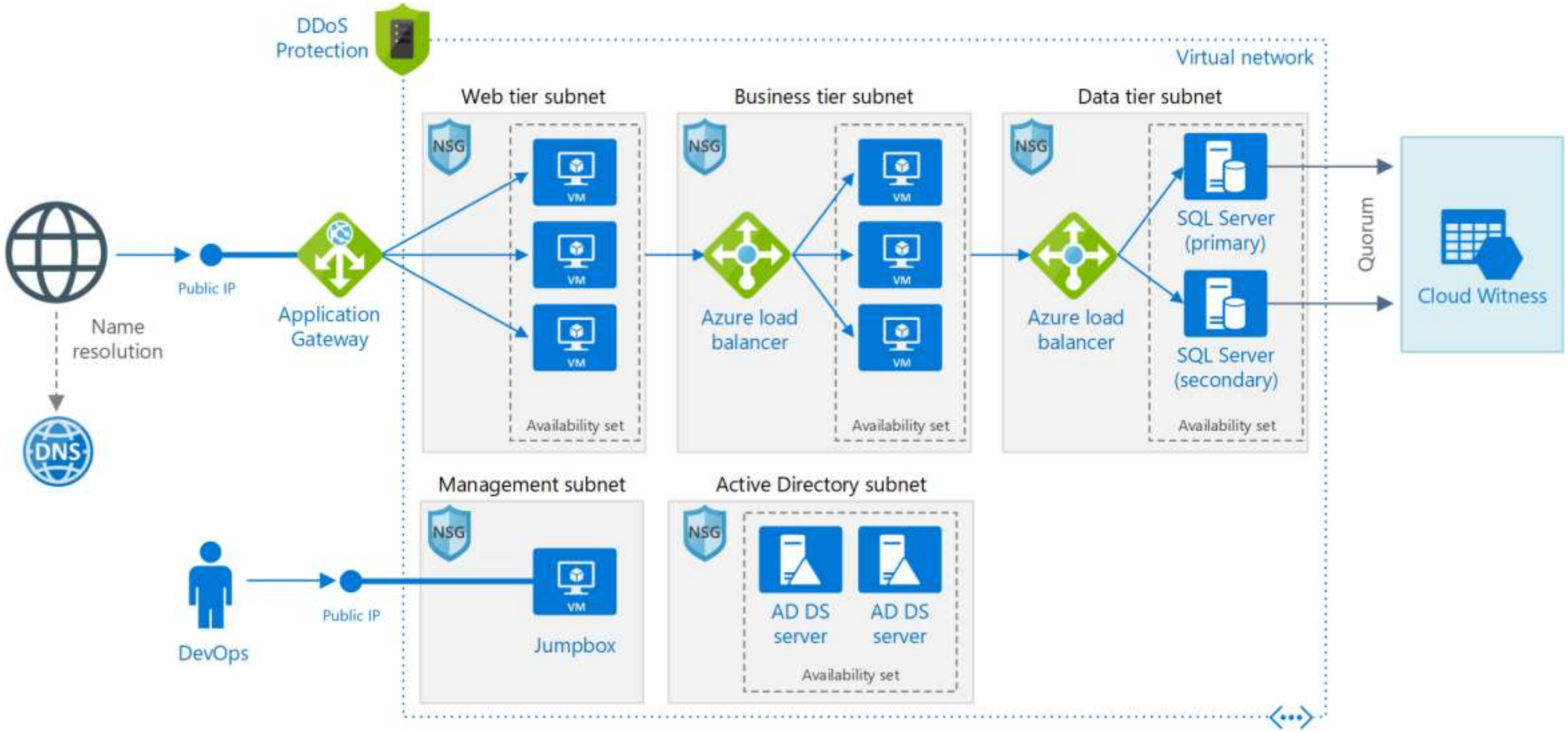
Agenda

- Di cosa parliamo oggi?
- Come esplorare un’infrastruttura in Microsoft Azure
- Quali dati sono interessanti?
- Azure Resource Graph e Kusto Language
- Demo
- Cosa è installato nella VM
- Canali di accesso remote e consultazione dei pacchetti
- Demo
- Prospettive e soluzioni

> Power is nothing without control

Campagna Pirelli 1995

Di cosa parliamo oggi?



Come esplorare un'infrastruttura in Microsoft Azure

- Manualmente/Documentazione ?
- Attraverso l'accesso http://portal.azure.com/ ?
- Azure REST API ?
- Analisi dei Template ARM?
- Scripting ?
- Azure Resource Graph ?
- Excel/Power BI

Quali dati sono interessanti?

- VMs
- Networking
- Risorse
- Images/Sku/Licensing
- States
- Storage location

Azure Resource Graph

Azure Resource Graph

- Azure CLI (as an extension)
- Azure PowerShell Core (Az.ResourceGraph module)
- Azure PowerShell (AzureRM.ResourceGraph module)
- Azure Portal (“All Resources” and “Resource Graph Explorer” view)
- REST API
- Azure Cloud Shell

Kusto Language

Kusto Language

- Linguaggio Query
- Basato sul modello relazionale Tabelle/Colonne
- Operatori query per analisi complesse
- Query aggregabili tramite Pipeline
- In Azure Resource Graph non c'è supporto completo del linguaggio
- Analisi interattiva su BigData
- Es.: `where type =~ 'Microsoft.Compute/virtualMachines' | summarize count()`

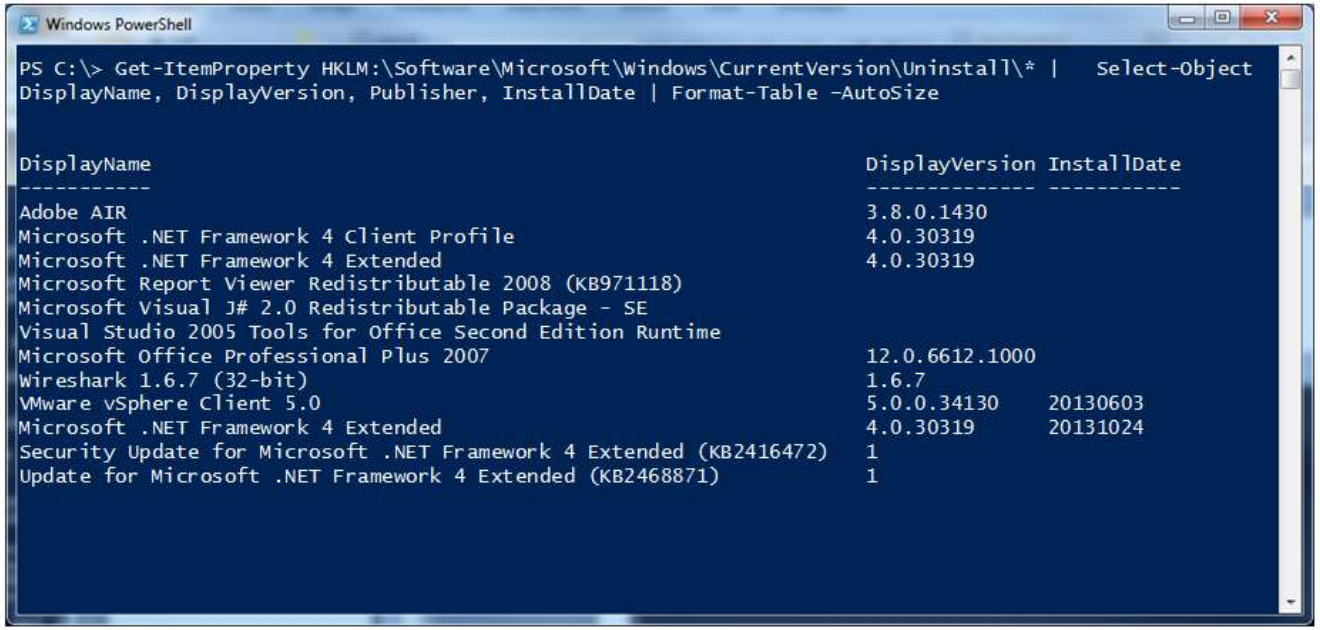
Azure Resource Graph

Cosa è installato nella VM

PowerShell

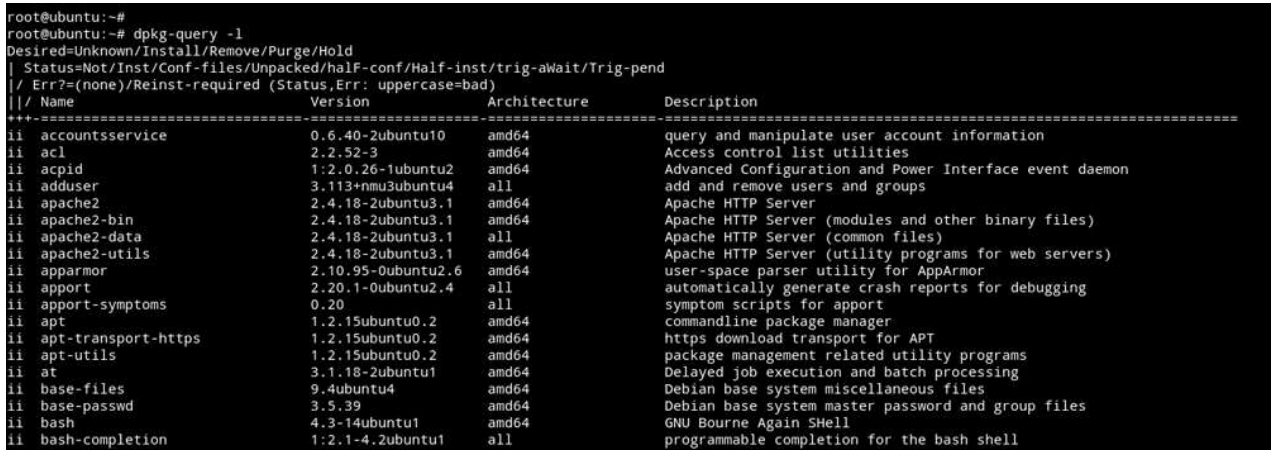
```
Azure:/
PS Azure:\> Enable-AzVMPSRemoting -Name 'vm-win-01' -ResourceGroupName 'azure-cloudshell-demo' -Protocol https -OsType Windows
Azure:/
PS Azure:\> Invoke-AzVMCommand -Name 'vm-win-01' -ResourceGroupName 'azure-cloudshell-demo' -ScriptBlock {get-service win*} -Credential
 $cred

Status   Name               DisplayName                           PSComputerName
------   ----               -----------                           --------------
Running  WinDefend          Windows Defender Antivirus Service    vmwin01.westus2.cloudapp.azure.com
Running  WindowsAzureGuest… Windows Azure Guest Agent             vmwin01.westus2.cloudapp.azure.com
Running  WindowsAzureNetAg… Windows Azure Network Agent           vmwin01.westus2.cloudapp.azure.com
Running  WindowsAzureTelem… Windows Azure Telemetry Service       vmwin01.westus2.cloudapp.azure.com
Running  WinHttpAutoProxyS… WinHTTP Web Proxy Auto-Discovery Serv… vmwin01.westus2.cloudapp.azure.com
Running  Winmgmt            Windows Management Instrumentation    vmwin01.westus2.cloudapp.azure.com
Running  WinRM              Windows Remote Management (WS-Managem… vmwin01.westus2.cloudapp.azure.com

Azure:/
PS Azure:\>
```

Canali di accesso remote

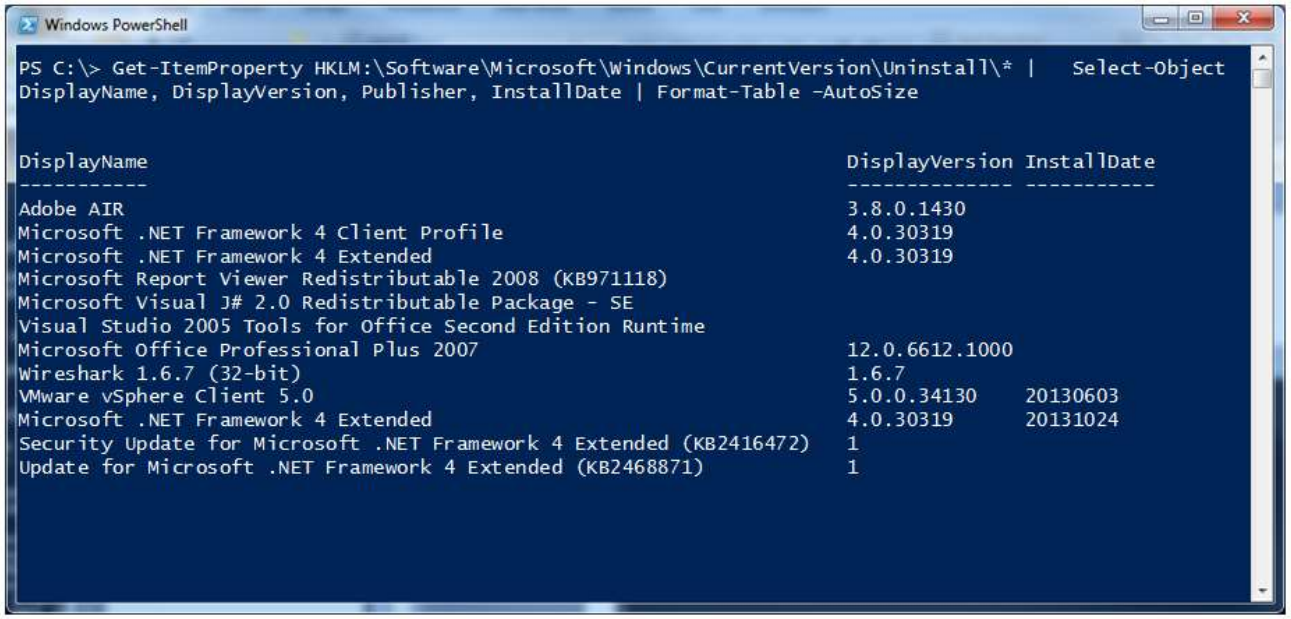



```
Invoke-AzVMCommand -Name
gabcltSrv31 -ResourceGroupName
GAB2019TestLab01 -ScriptBlock
{get-service win*} -Credential
(get-credential)
```

```
ssh -T
giulianolatini@23.100.59.233
'dpkg-query -l'
```

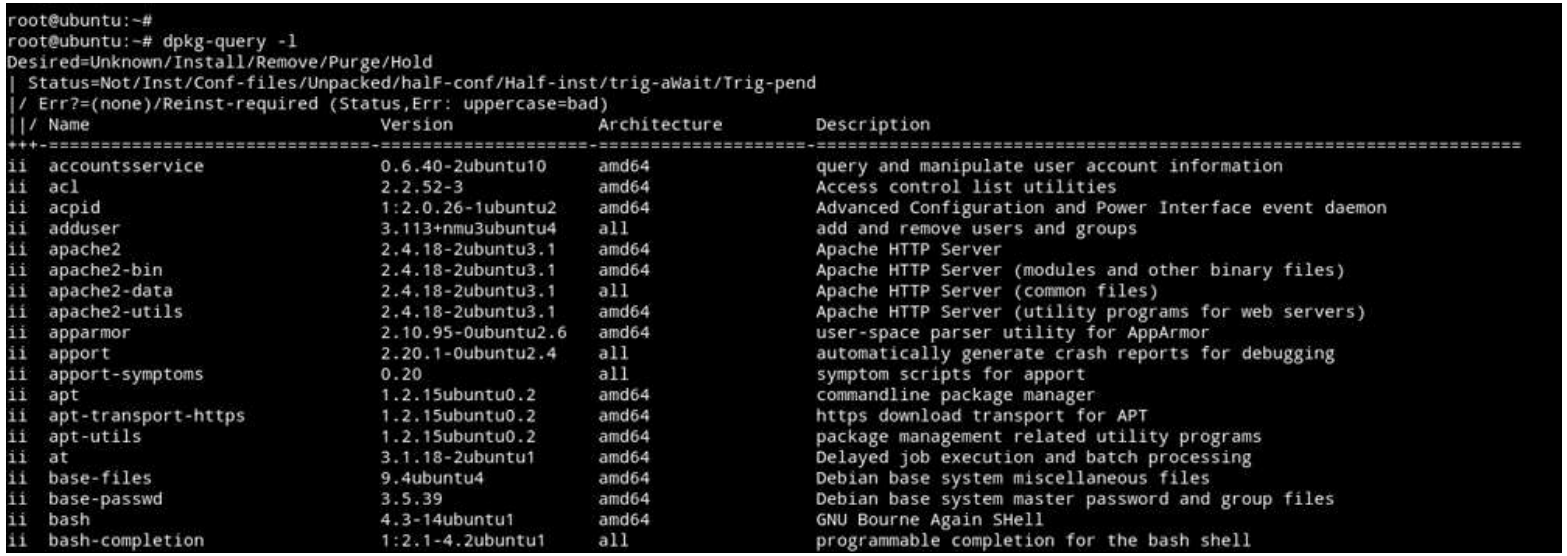


Canali di accesso remote e consultazione dei pacchetti



Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall* | Select-Object DisplayName, DisplayVersion, Publisher, InstallDate | Format-Table –AutoSize

dpkg-query -l



Lista del Software Installato

Prospettive e soluzioni

- Interroghiamo l'infrastruttura
- Inspezionabilità per soluzioni Infrastructure as Code
- Esportabilità verso Excel/Power BI
- Rilevamento di variazioni per infrastruttura

- Documentazione?
- Dashboard di stato per live monitor?
- Query pinnabili sulla dashboard Azure ?

Bibliografia 1/3

- Azure Resource Graph Documentation - Tutorials, API Reference
- Reference - Azure Data Explorer - Kusto Query Language
- Getting started with Kusto - Azure Data Explorer
- Kusto Tools - Azure Data Explorer
- Documentazione di Resource Manager in Azure - Esercitazioni, esempi, informazioni di riferimento e risorse
- Saving time with Azure Resource Graph
- Explore Azure Resources with Resource Graph

Bibliografia 2/3

- MicrosoftDocs/azure-docs/articles/governance/resource-graph/
- MicrosoftDocs/azure-docs/articles/governance/resource-graph/samples/
- Azure Resource Graph Introduction
- Explore Azure Resources With Resource Graph - DZone Cloud
- Azure Resource Graph: How many CPU cores are used
- Top 10 PowerShell Script to check the Windows Service on remote server - OnClick360
- Remoting into Azure ARM Virtual Machines using Powershell
- PowerShell Basics: Connecting to VMs with Azure PSRemoting

Bibliografia 3/3

- Check Azure VM status with PowerShell
- Configure PowerShell remoting for Azure VMs - Powershellbros.com
- Enable PowerShell Remoting to an Azure Virtual Machine, without Domain Membership | Azure | SignalWarrant.com - Practical PowerShell for Humans
- Use PowerShell to Quickly Find Installed Software | Scripting
- Use PowerShell to Find Installed Software | Scripting
- How to Create a List of Your Installed Programs on Windows

Codice

Abilita Resource Graph

- PowerShell: `Install-Module -Name AzureRm.ResourceGraph -AllowPrerelease`
- PowerShell: `Install-Module -Name Az -AllowClobber`
- CLI: `az extension add --name resource-graph`

Codice

Esempi di Query

Powershell

```
Search-AzGraph -Query "project name, type, location | order by name asc"

search-AzureRmGraph -Query "where type =~ 'Microsoft.Compute/virtualMachines' and name == 'gabcltSrv31' "
```

CLI

```
az graph query -q "project name, type, location | order by name asc"

az graph query -q "where type =~ 'Microsoft.Compute/virtualMachines' | project name, type, location | order by name asc" --output tsv |
out-file numberofcores.csv

az graph query -q "where type =~ 'Microsoft.Compute/virtualMachines' and name == 'gabcltSrv31' "

az graph query -q "where type =~ 'Microsoft.Compute/virtualmachines' and properties.hardwareProfile.vmSize matches regex 'Standard_DS[0-
9]*_v2'| project name"
```

Codice

Accesso VM

Abilitazione Remoting

```
Enable-AzVMPSRemoting -Name 'gabcltSrv31' -ResourceGroupName 'GAB2019TestLab01' -Protocol https -OsType Windows

Enable-AzVMPSRemoting -Name 'gabwebSrv32' -ResourceGroupName 'GAB2019TestLab01' -Protocol https -OsType Windows

Enable-AzVMPSRemoting -Name 'gabwebSrv21' -ResourceGroupName 'GAB2019TestLab01' -Protocol ssh -OsType Linux
```

Connessione con Prompt

```
Enter-AzVM -name 'gabwebSrv32' -ResourceGroupName 'GAB2019TestLab01' -Credential (get-credential)

Enter-AzVM -name 'gabcltSrv31' -ResourceGroupName 'GAB2019TestLab01' -Credential (get-credential)

choco install vscode

ssh giulianolatini@23.100.59.233
```

Codice

Lista Software Installati

```
Invoke-AzVMCommand -Name gabcltSrv31 -ResourceGroupName GAB2019TestLab01 -ScriptBlock {get-service win*} -Credential (get-credential)

Invoke-AzVMCommand -Name 'gabcltSrv31' -ResourceGroup 'GAB2019TestLab01' -ScriptBlock {Get-ItemProperty
HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Select-Object DisplayName, DisplayVersion, Publisher, InstallDate | Format-
Table –AutoSize} -Credential (get-credential)

$cred = get-credential

Invoke-AzVMCommand -Name 'gabcltSrv31' -ResourceGroup 'GAB2019TestLab01' -ScriptBlock {get-service win*} -Credential $cred

Invoke-AzVMCommand -Name 'gabcltSrv31' -ResourceGroup 'GAB2019TestLab01' -ScriptBlock {Get-ItemProperty
HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Select-Object DisplayName, DisplayVersion, Publisher, InstallDate | Format-
Table –AutoSize} -Credential $cred

ssh -T giulianolatini@23.100.59.233 'dpkg-query -l'
```

Codice

Conclusione

```powershell
$cred = get-credential

$machine = az graph query -q "where type =~ 'Microsoft.Compute/virtualMachines' and name == 'gabcltSrv31' | project name " --output tsv

foreach ($computer in $machine) { Invoke-AzVMCommand -Name $computer -ResourceGroup GAB2019TestLab01 -ScriptBlock {Get-ItemProperty
HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Select-Object DisplayName, DisplayVersion, Publisher, InstallDate | Format-
Table -AutoSize} -Credential $cred }

az graph query -q "where type =~ 'Microsoft.Compute/virtualMachines' and name == 'gabcltSrv31' | project name " --output tsv | foreach {
Invoke-AzVMCommand -Name $_ -ResourceGroup GAB2019TestLab01 -ScriptBlock {Get-ItemProperty
HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Select-Object DisplayName, DisplayVersion, Publisher, InstallDate | Format-
Table -AutoSize} -Credential (get-credential) }
```

Thanks

Questions?

github.com/giulianolatini

giulianolatini

linkedin.com/in/giulianolatini/