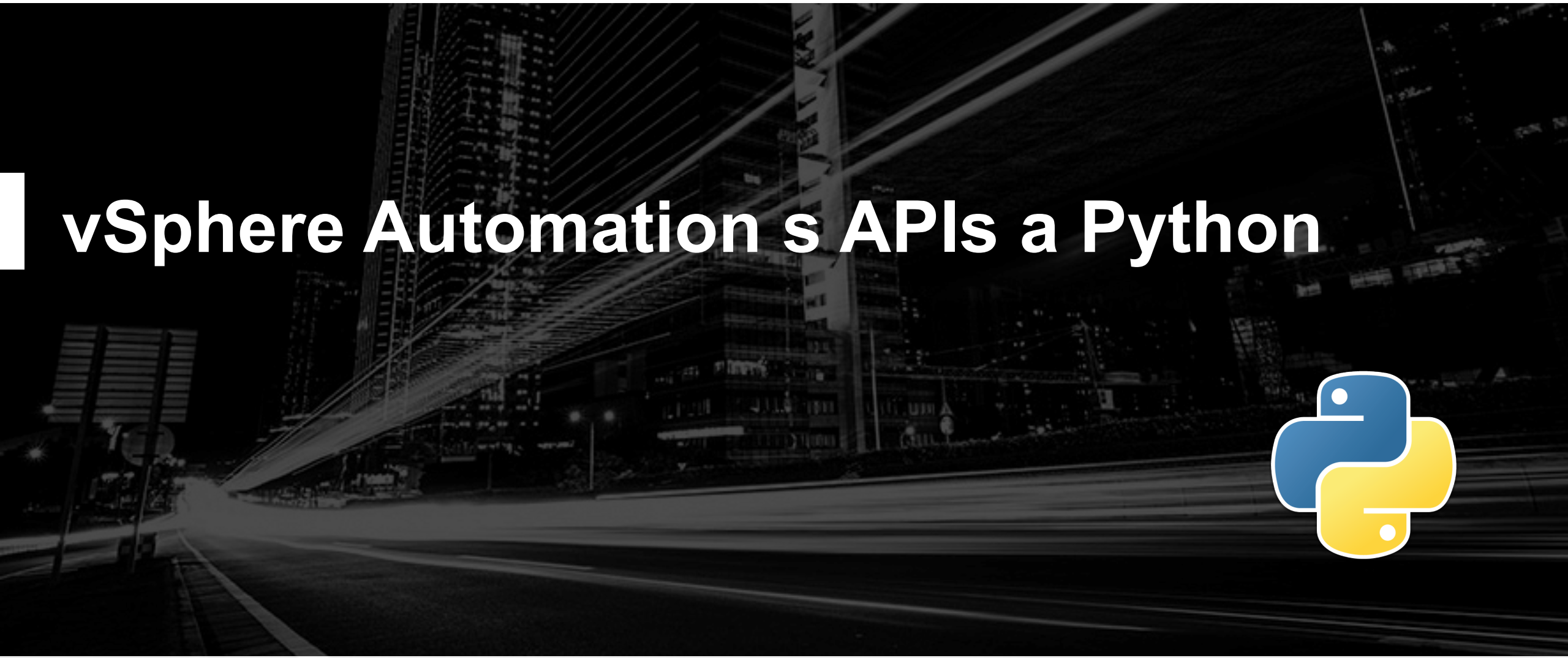
# vSphere Automation s APIs a Python



# Vladan Laxa

- Arrow ECS
- v IT cca 16 let
- Konzultant pro systémové integrátory
- Infinity, AG COM, AUTOCONT
- VMware zaměření 14 let
- VCP-DCV od verze 2
- Cloud Management and Automation
- vladan.laxa@arrow.com

https://github.com/vlaxa

https://twitter.com/latraxa

# Představení účastníků:

- Jméno a příjmení
- Firma
- Profesní zaměření
- Zkušenost s automatizací, znalost Python

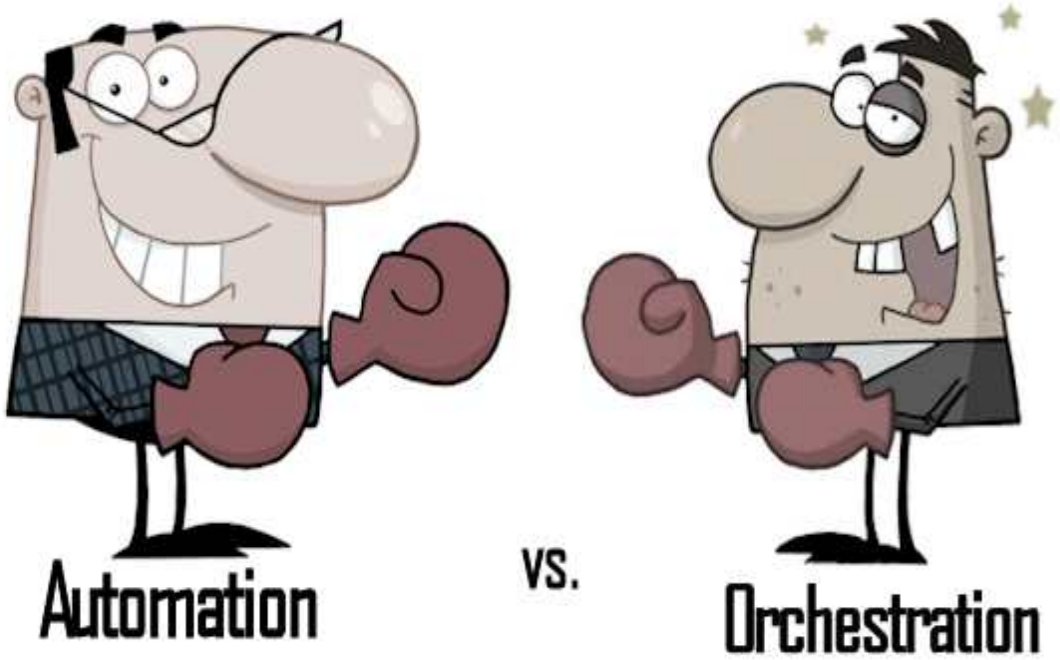
# Agenda:

- Automatizace vs. Orchestrace
- Praktické seznámení s vSphere APIs
- Nástroje pro práci s vSphere API
- Interakce s vSphere API s využitím jazyka Python
- Automatizace – Python knihovna pyVmomi
- Orchestrace vSphere pro DevOps s Python
- Závěr

# Automatizace vs. Orchestrace

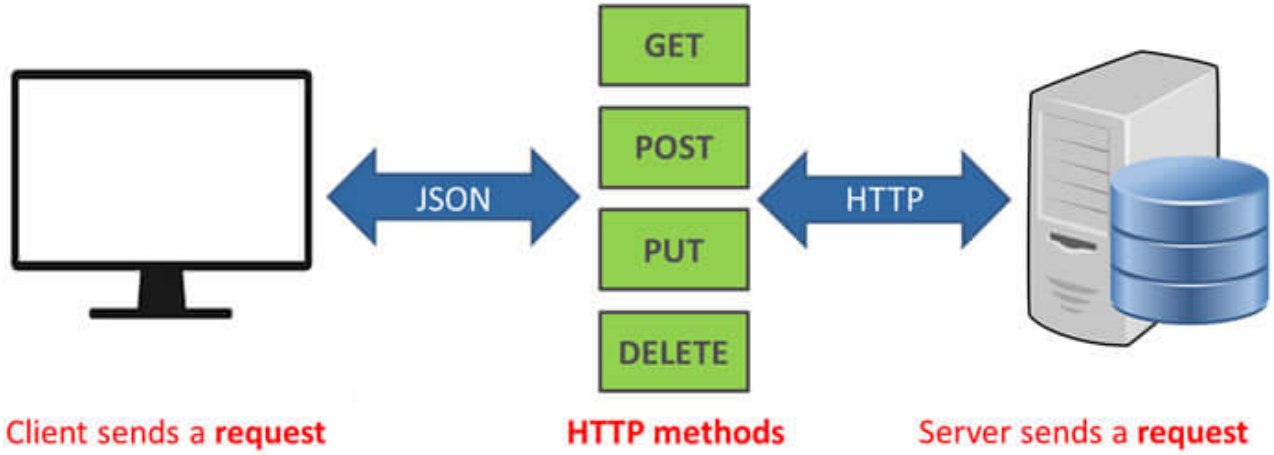
- **Automatizace** je ošetření jedné úlohy tak, aby byla vykonána bez zásahu člověka
- **Orchestrace** je optimalizace procesu / workflow sestávajícího z více automatizovaných úloh

- Šetří **čas**
- Šetří **zdroje**
- Minimalizuje riziko vzniku **chyby**
- **DR** scénáře s využitím **IaaC**



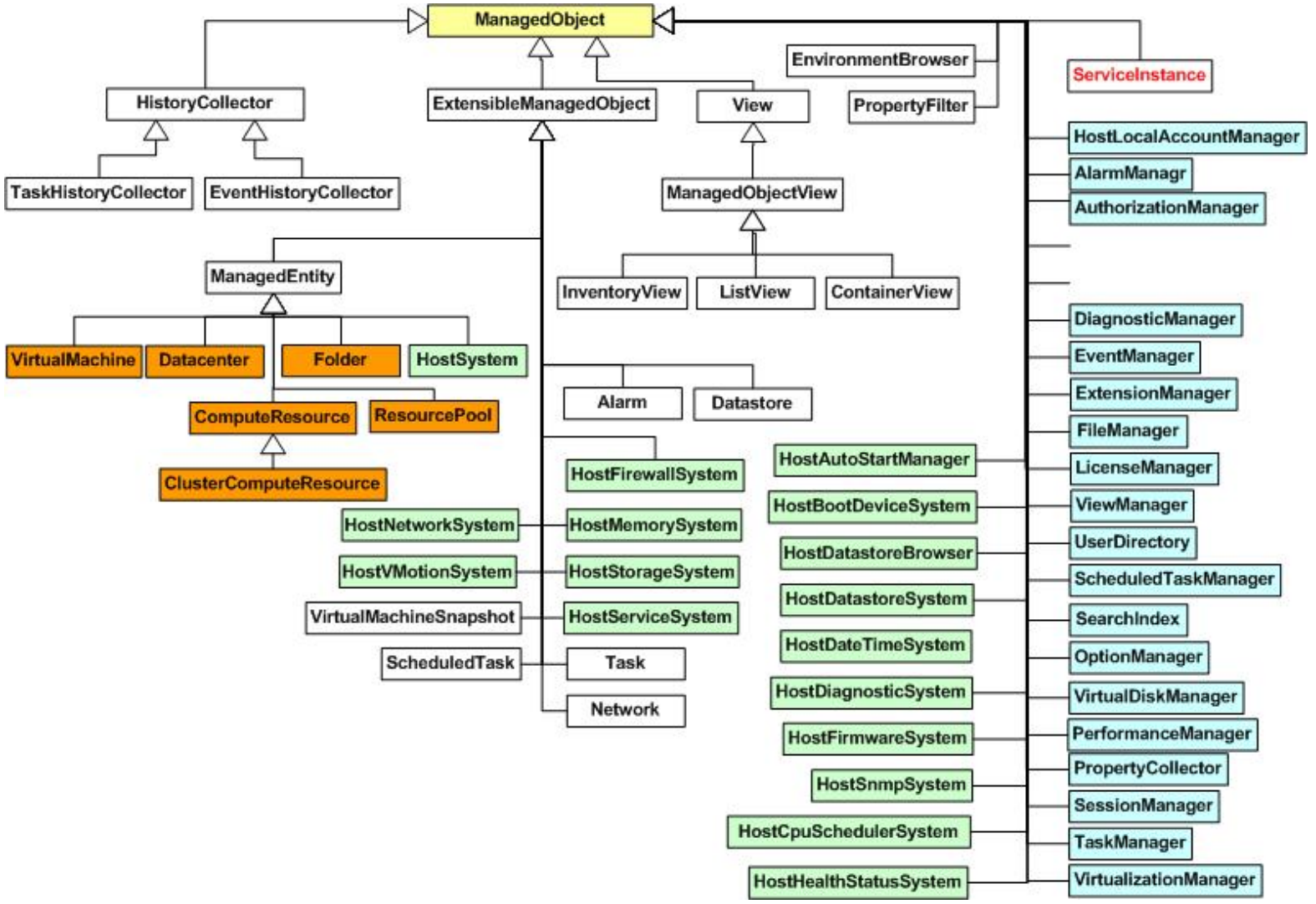
# Co je to API?

- **API** (application programming interface)
- Je software, který umožňuje propojení jedné aplikace na data a služby druhé aplikace
- Výměna dat mezi aplikacemi
- Softwarová integrace
- Využití JSON datového formátu
- vSphere APIs
  https://code.vmware.com/apis



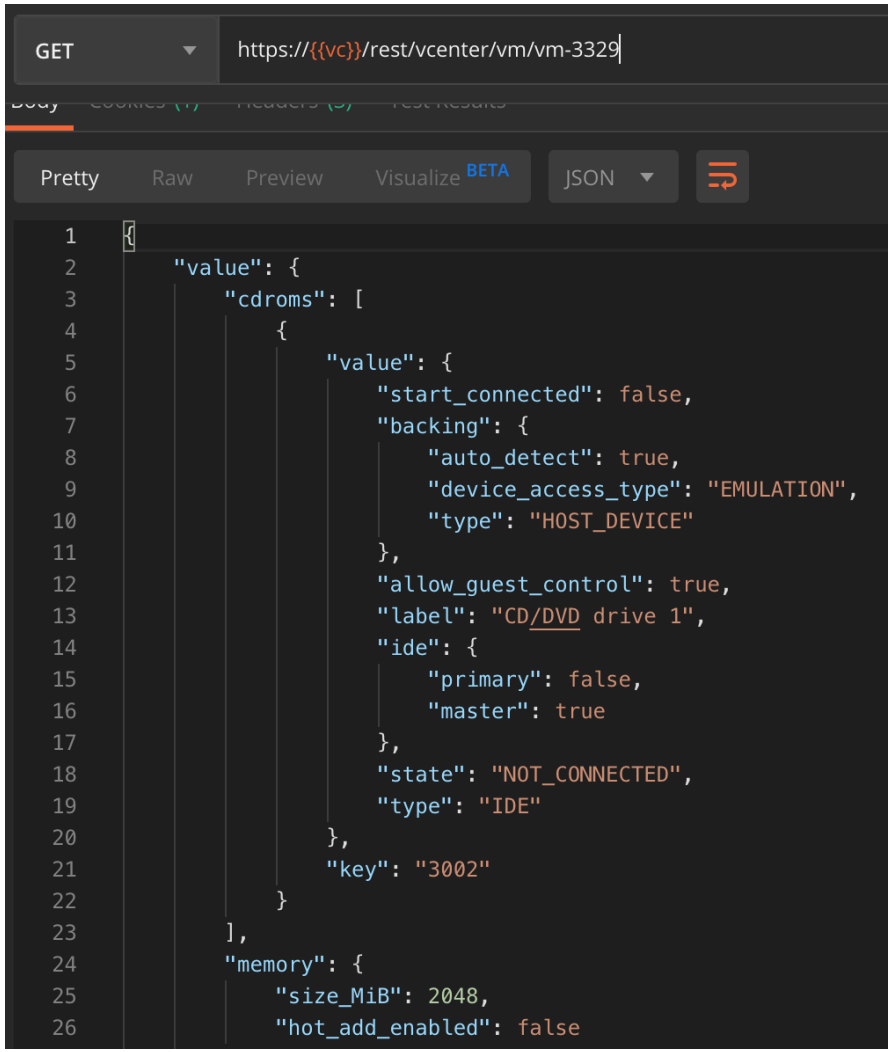
# vSphere API Object Model

- **Managed Object -** server side object
- **Managed Object Reference -** client reference to a Managed Object
- **Data Object** – obsahuje informace o Managed Object a využívá se pro přenos dat mezi klientem a serverem



# JSON datový formát

- **JSON** (JavaScript Object Notation)
- Odlehčený formát pro výměnu dat založený na syntaxi JavaScript objektů
- Textového typu
- Čitelný pro běžného uživatele
- Nezávislý na konkrétním programovacím jazyku
- Reprezentuje strukturovaný datový objekt
- Ukládá se do souborů *.json application/json MIME type
- JSON data v Python, využívá  se vestavěný **json** module
- Odlehčená  alternativa k **XML**



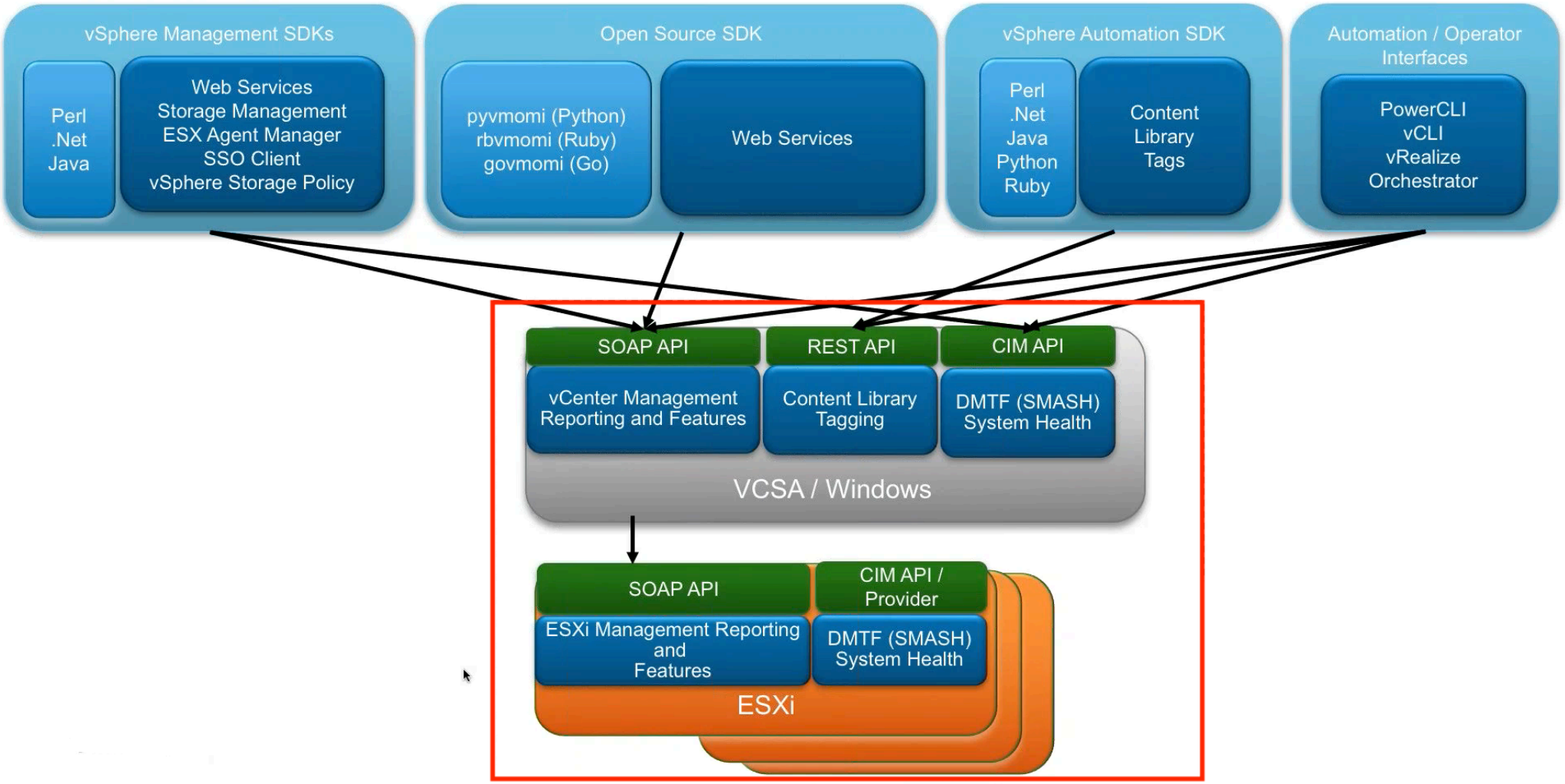
# JSON datový formát v Pythonu

- JSON existuje jako řetězec —posloupnost (série bytů). Ke konverzi objektu (slovníku) do JSON reprezentace, je třeba objekt encodovat do “série bytů”, tento proces se nazývá **serializace**.
- **Deserializace** reverzní operace k serializaci. Dekóduje data získaná v JSON formátu

- dumps()  serializace objektu do JSON formátovaného řetězce.
- dump()  serializace objektu do JSON formátovaného streamu (pro zápis do souboru).

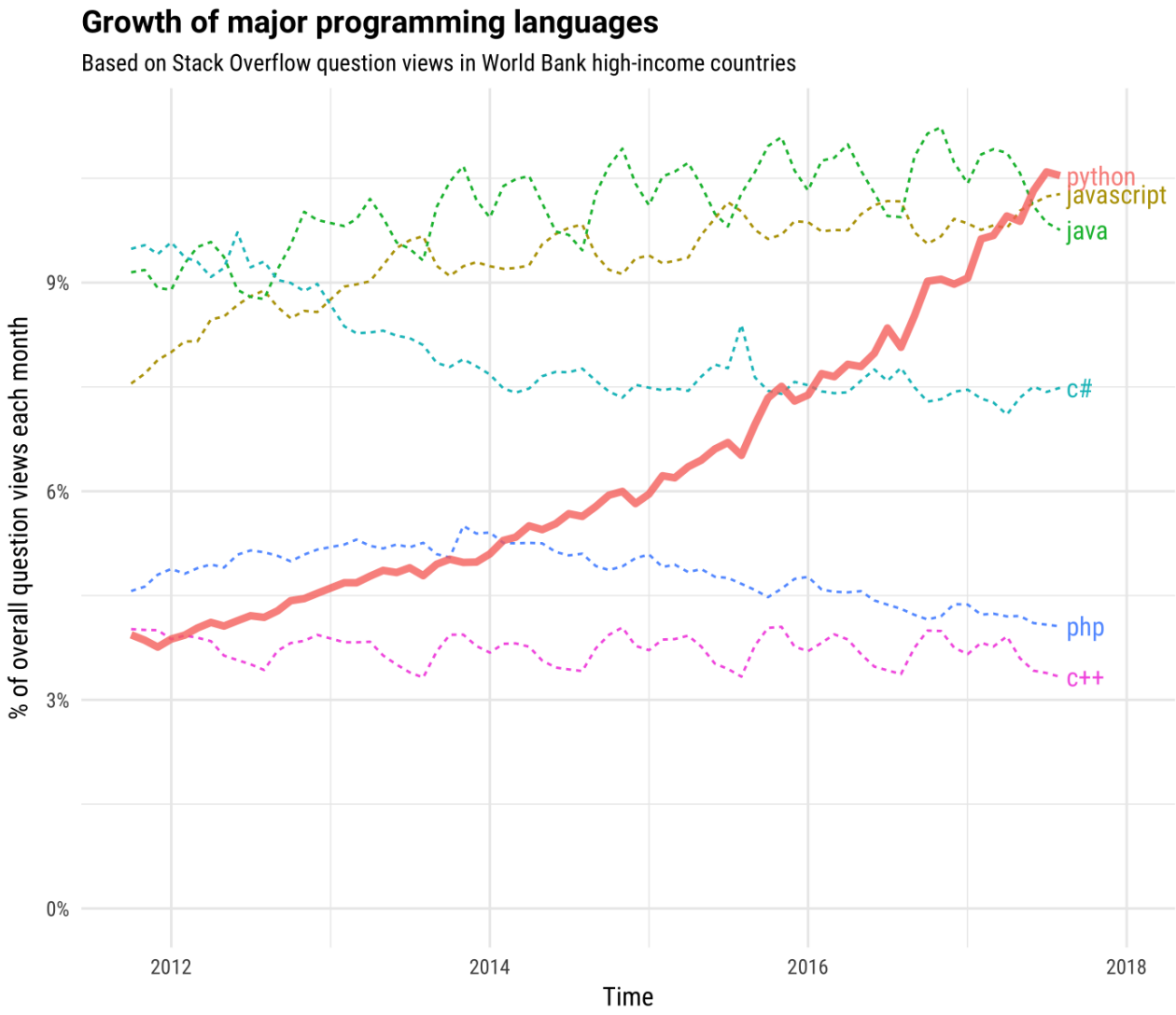
- loads()  deserializuje  JSON data do Python objektu.
- load()  deserializuje JSON formátovaný stream (pro čtení ze souboru) do Python objektu.

# Praktické seznámení s vSphere APIs



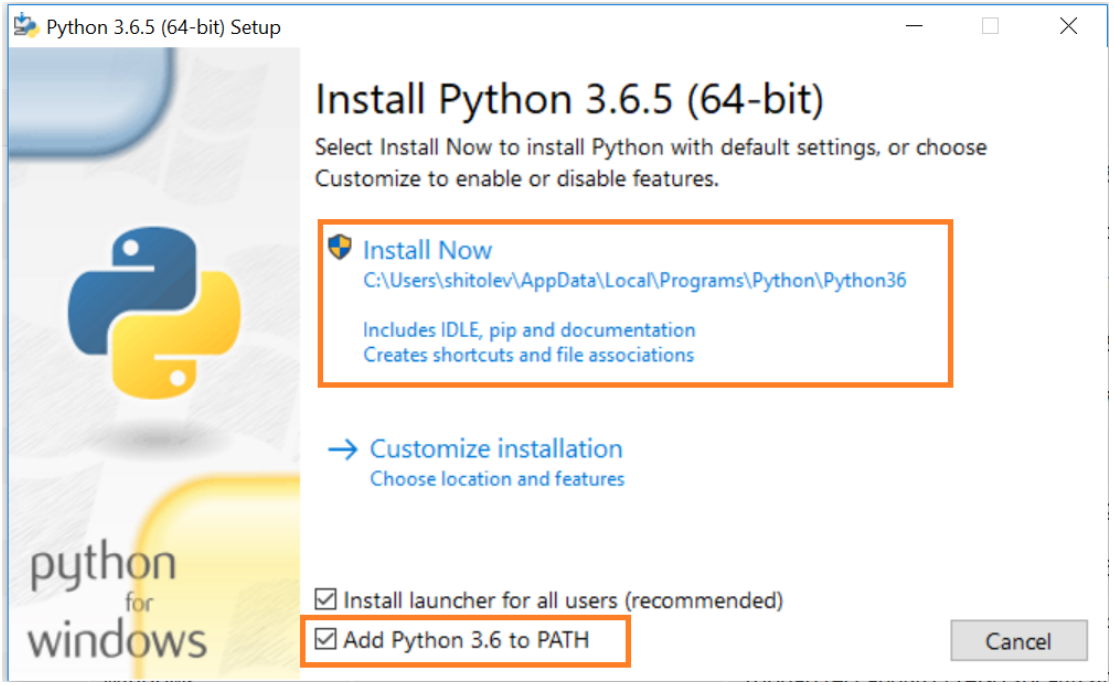
# Proč Python?

- Velmi populární jazyk
- Poměrně snadné na naučení
- Interpretovaný jazyk
- Často využívaný pro config management, automatizaci, ML, AI
- Multiplatformní
- Množství informačních zdrojů
- Vstřícná komunita



# Instalace Pythonu a příprava prostředí

- Multiplatformní https://www.python.org
- Linux, macOS, iOS, Windows
- Izolace vývojového prostředí do **virtuálního** prostředí
- Práce s externími moduly (knihovnami)
- brew install python3
- sudo apt-get install python3.7




- Virtuální prostředí – založení a aktivace:
- **Windows**
- py -3 -m venv venv
- venv\Scripts\activate
- **Linux, macOS**
- python3 -m venv venv
- source venv/bin/activate

git clone https://github.com/vlaxa/vsphere-automation-workshop

# SOAP vs. REST API

- **SOAP (Simple Object Access Protocol)**
- Je vlastní protokol
- Lepší zabezpečení
- Plná podpora u vSphere
- Složitější, vyžaduje důkladnou znalost API
- SOAP má built-in ACID compliance. (ACID (Atomicity, Consistency, Isolation, Durability) compliance.)
- SOAP-based calls nelze využít cache

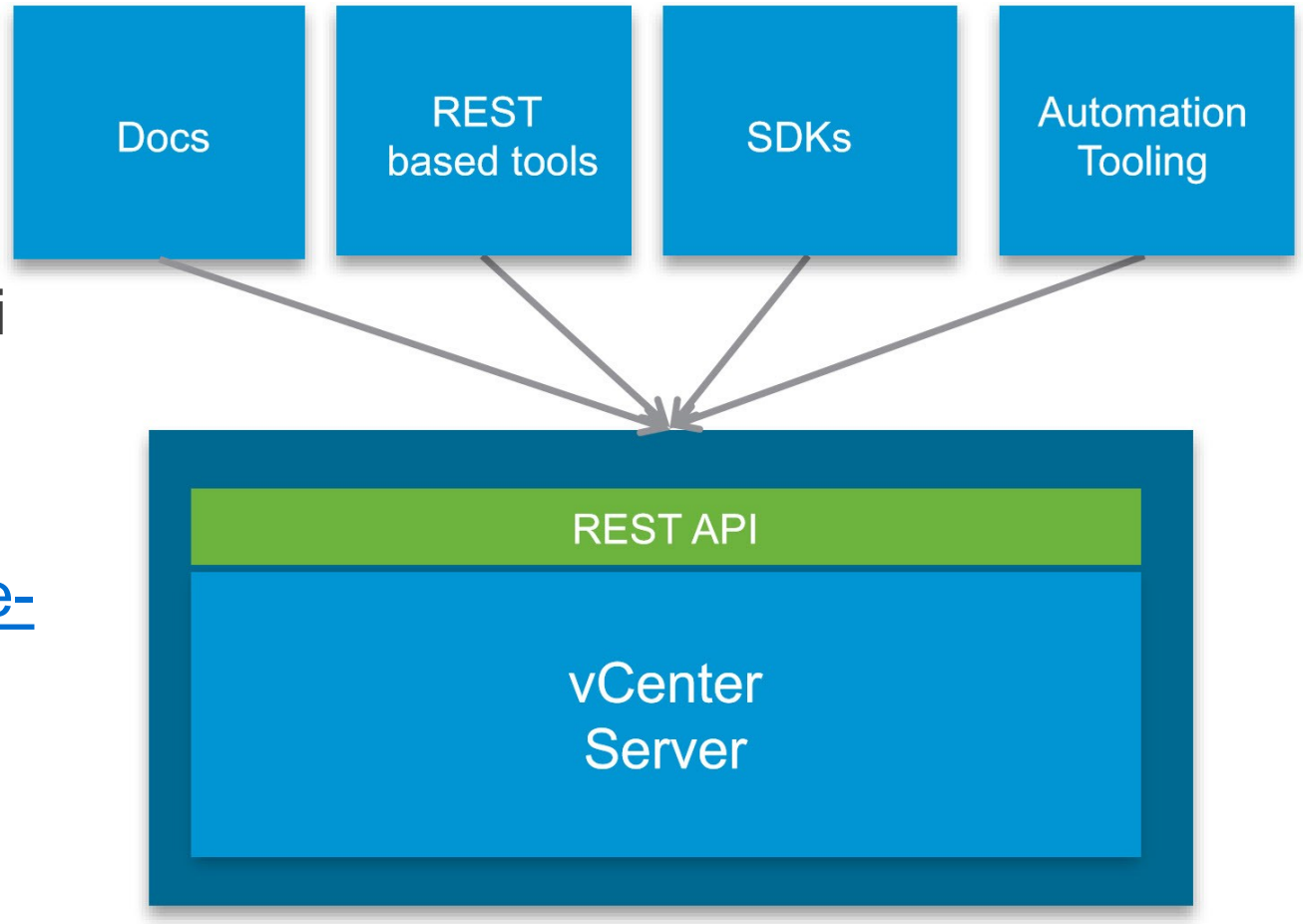
- **REST (Representational State Transfer)**
- Styl Architektury
- Optimalizováno pro web
- Moderní API design
- Jednoduchý přístup k automatizaci s využitím standardních nástrojů
- Vyžaduje méně zdrojů
- Management VM a VCSA
- Reprezentuje webové služby založeny na URIs (Uniform Resource Identifier)

# RESTful API Metody (CRUD)

- **POST** vytvoření dat (Create)
- **GET** získání požadovaných dat (Retrieve) = **nedestruktivní**
- PUT změnu (Update)
- DELETE smazání (Delete)

# vCenter REST API

- REST API pro VM management
- Moderní API design
- Jednoduchý přístup k automatizaci s využitím standardních nástrojů
- Management VCSA
- https://github.com/vmware/vsphere-automation-sdk-python
- https://vmware.github.io/vsphere-automation-sdk-rest/vsphere/index.html



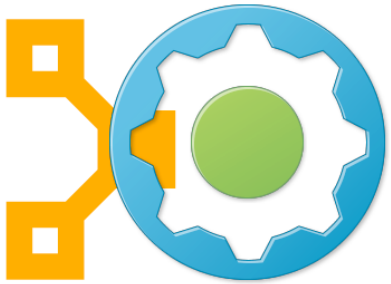
# vCenter REST API

- Dostupné APIs na vCenter **API exploreru**
- **https://vcenter_ip_or_fqdn/apiexplorer**
- Využívá **Swagger** technologie
- **Appliance** – Zahrnuje Vmware VCSA appliance
- **CIS** – Common Infrastructure Services, týká se tagování
- **Content** – obsluhuje volání související s Content Library
- **vAPI** – vSphere API, API endpoint
- **vCenter** – obsluhuje vCenter, VM management

# Nástroje pro práci s vSphere APIs

- CURL
- **Postman**
- API Explorer
- **MOB**
- **IDE** (Integrated Development Environment)
- Python **requests**
- Ansible, Ansible Tower
- Terraform
- vRealize Automation
- vRealize Orchestrator
- Python **pyVmomi**









# CURL

- https://curl.haxx.se
- **C**ommand line tool and library for transferring data with **URL**s

```
curl -k -i -u administrator@vsphere.local:password -X POST -c cookie-jar.txt
https://vcenter/rest/com/vmware/cis/session
curl -k -i -b cookie-jar.txt https://vcenter/rest/vcenter/vm
```

# Postman


- https://www.getpostman.com
- API klient
- API development testování
- Kolekce pro vSphere REST API
- https://github.com/vmware/vsphere-automation-sdk-rest/tree/master/samples/postman

# API Explorer

- https://vcenter_fqdn_or_ip/apiexplorer
- vSphere REST API development

# MOB

- https://host_fqdn_or_ip/mob
- **MOB** = Managed Object Browser
- Dostupné i na ESXi hostovi (Advanced Setting)
- Config.HostAgent.plugins.solo.enableMob

Home

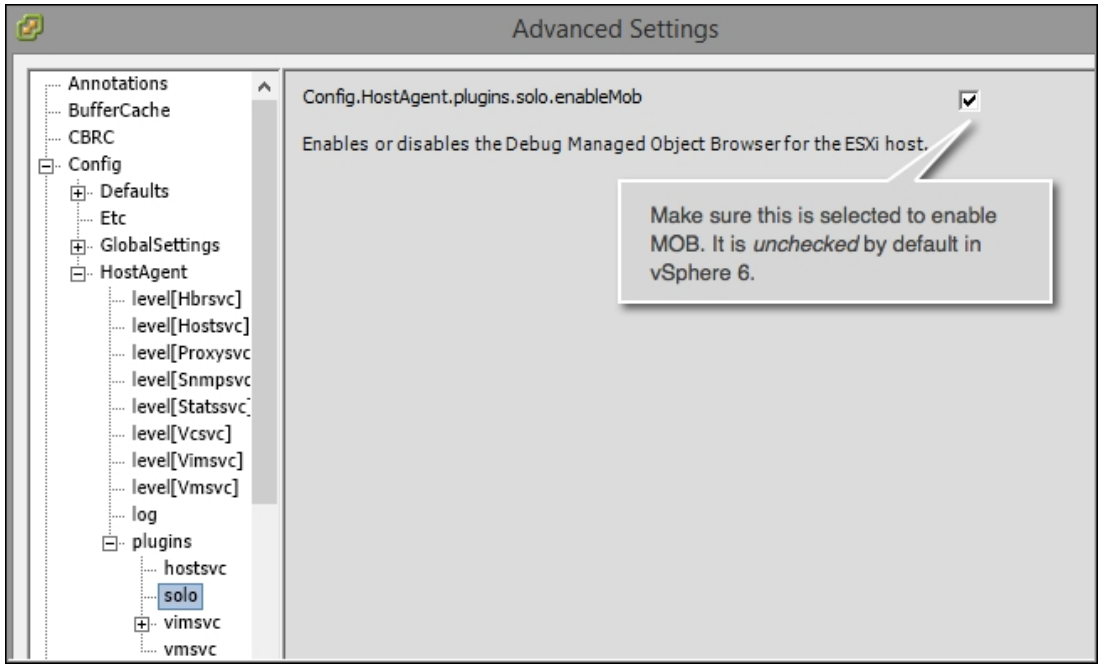
Managed Object Type: **ManagedObjectReference:ServiceInstance**
Managed Object ID: **ServiceInstance**

**Properties**

| NAME | TYPE | VALUE |
|---|---|---|
| capability | Capability | capability |
| content | ServiceContent | content |
| serverClock | dateTime | "2019-11-13T09:09:50.190465Z" |

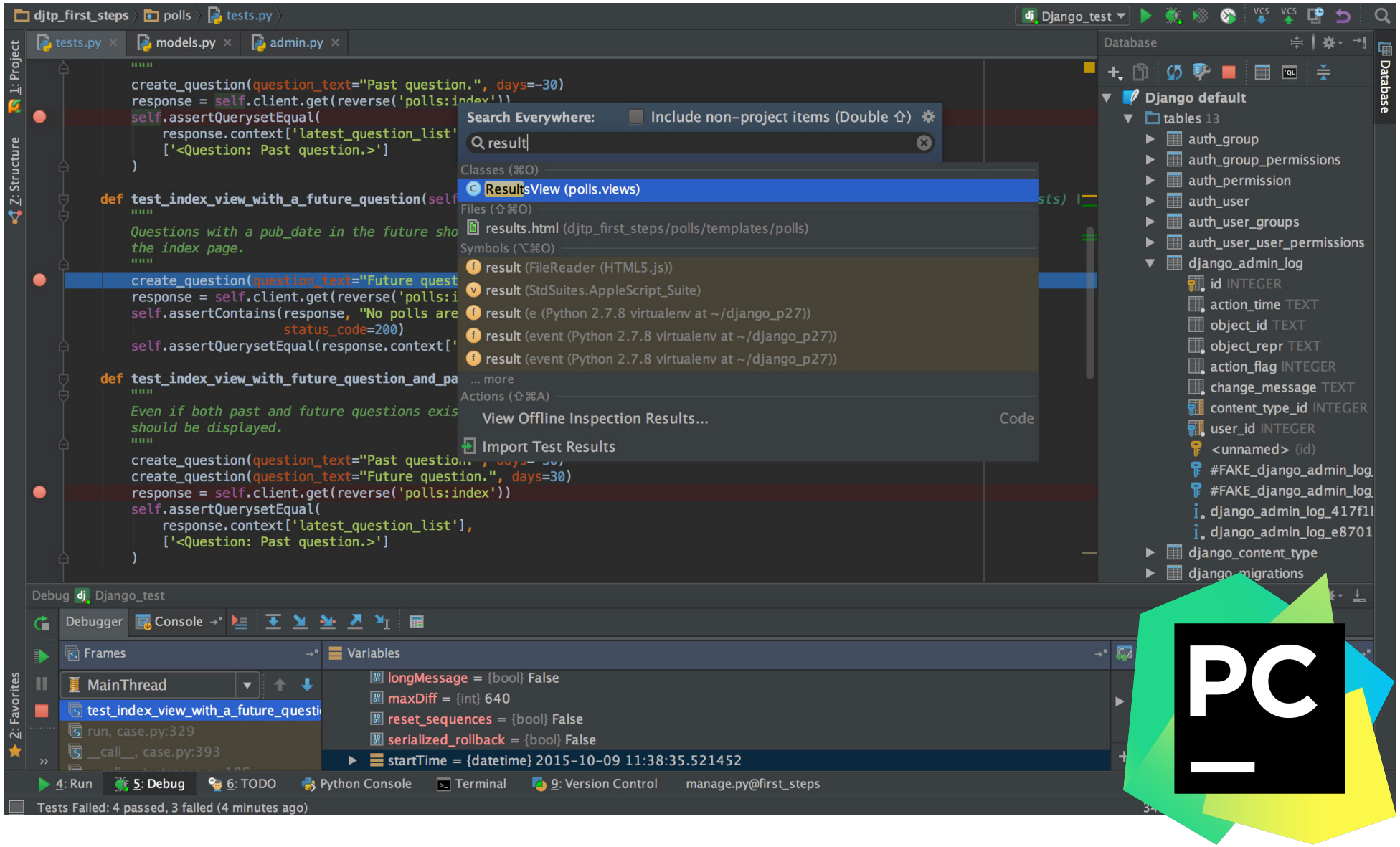
**Methods**

| RETURN TYPE | NAME |
|---|---|
| dateTime | CurrentTime |
| HostVMotionCompatibility[] | QueryVMotionCompatibility |
| ServiceContent | RetrieveServiceContent |
| ProductComponentInfo[] | RetrieveProductComponents |
| Event[] | ValidateMigration |



# IDE

- Integrated Development Environment
- **PyCharm**
- VS Code
- Atom



# Python requests

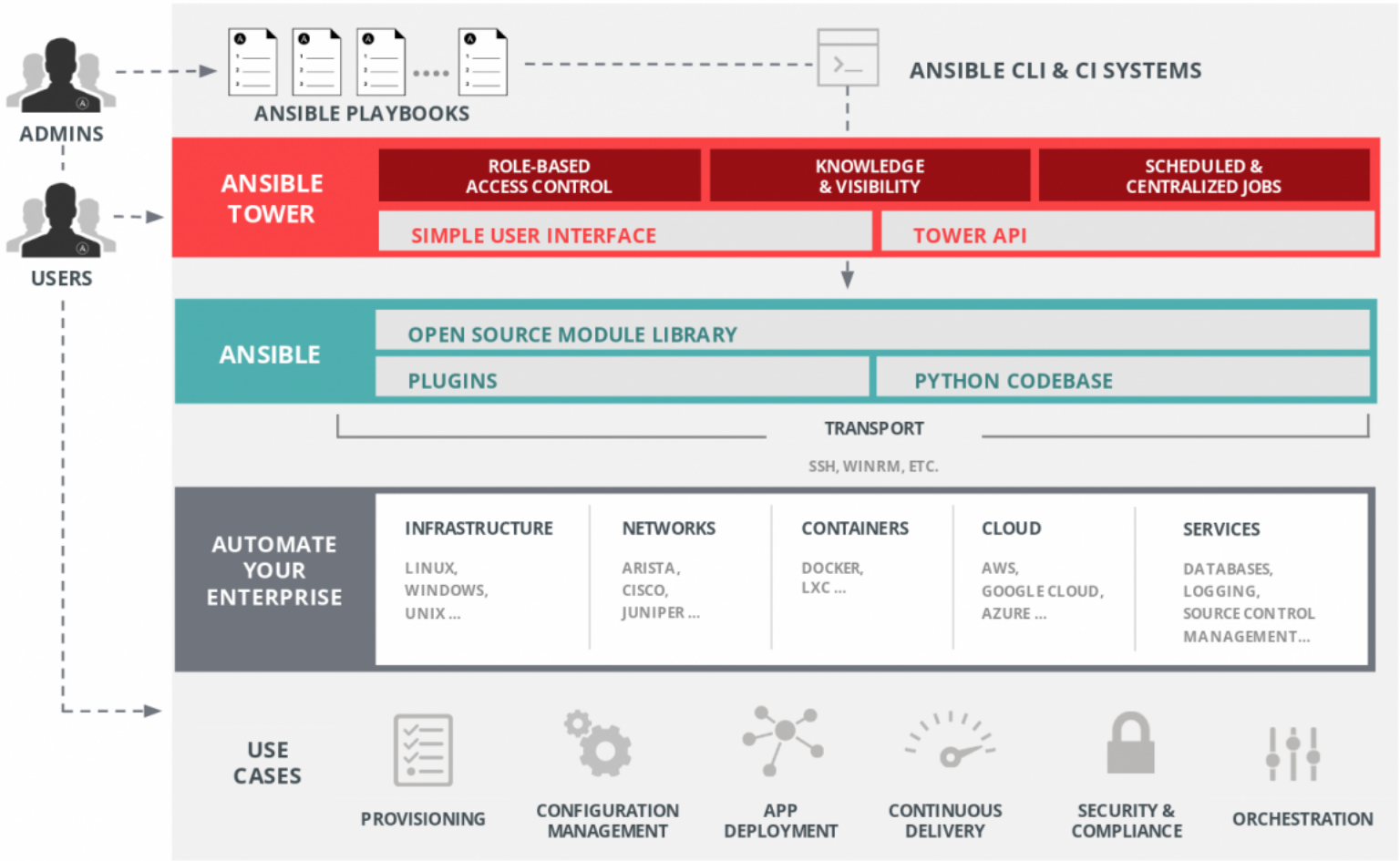

- Python knihovna **requests**
- pip install requests
- import requests
- Vytváří dotazy s využitím HTTP metod: **GET**, POST, PUT, DELETE
- Response status code 200 OK, 404 Not Found
- Content
- Headers

# Interakce s vSphere REST API s využitím Python


- Python
- Knihovna **Requests**
- **Header** – metadata k řízení dotazu a odpovědi
  - Authentication – token nebo login data
  - Content-Type – specifikuje IN/OUT typy (JSON, XML, txt,…)
- pip install requests
- **DEMO**

# Ansible, Ansible Tower

- https://docs.ansible.com/ansible/latest/scenario_guides/guide_vmware.html
- Ansible vSphere module
- Integrace Ansible Tower s vRA (není podpora u vRA 8.0)
- Postavené na Python
- Možné instalovat jako python modul
- Psaní vlastních modulů v pythonu
- https://docs.ansible.com/ansible/latest/dev_guide/developing_modules_general.html



# vRealize Automation

- https://github.com/kovarus/vrealize-pysdk
- vRA Python příklady:
- https://github.com/codyde/vrealize-auto-python-examples
- vRA API Samples pro Postman:
- https://code.vmware.com/samples/2412/vrealize-automation-api-samples-for-postman

# vRealize Orchestrator

- Modul: vmwvro
- https://pypi.org/project/vmwvro/

# Python SDK for vCloud Director

- https://vmware.github.io/pyvcloud/
- pyvcloud

# Automatizace - Python a knihova pyVmomi

- **pyVmomi**
- Prosinec 2013 => 6 let
- Využíváme **MOB**
- Open Source Python SDK
- datacenter = c.content.rootFolder.childEntity[0]
- https://code.vmware.com/apis/704/vsphere
- **DEMO**

Home

Data Object Type: **AboutInfo**
Parent Managed Object ID: **ServiceInstance**
Property Path: **content.about**

**Properties**

| NAME | TYPE | VALUE |
|---|---|---|
| apiType | string | "VirtualCenter" |
| apiVersion | string | "6.7.3" |
| build | string | "14792544" |
| fullName | string | "VMware vCenter Server 6.7.0 build-14792544" |
| instanceUuid | string | "985c286e-f373-4817-a999-851745987373" |
| licenseProductName | string | "VMware VirtualCenter Server" |
| licenseProductVersion | string | "6.0" |
| localeBuild | string | "000" |
| localeVersion | string | "INTL" |
| name | string | "VMware vCenter Server" |
| osType | string | "linux-x64" |
| productLineId | string | "vpx" |
| vendor | string | "VMware, Inc." |
| version | string | "6.7.0" |

# Automatizace – Python - vytvoření session

- **pyVmomi**
- **Python** Virtualization Management Object Management Infrastructure
- pyVim.connect module
- SmartConnect funkce
- ssl module
- SSLContext
- help(SmartConnect)

vminfo.py

```
1   from pyVim.connect import SmartConnect, Disconnect
2   import ssl
3
4   # For VC 6.5/6.0
5   #s = ssl.SSLContext(ssl.PROTOCOL_TLSv1)
6   # For VC 6.7
7   s = ssl.SSLContext(ssl.PROTOCOL_SSLv23)
8   s.verify_mode = ssl.CERT_NONE
9
10  try:
11      c = SmartConnect(host="10.188.140.23", user="root", pwd='VMware1!')
12      print('Valid certificate')
13  except:
14      c = SmartConnect(host="10.188.140.23", user="root", pwd='VMware1!', sslContext=s)
15      print('Invalid or untrusted certificate')
16
```

"c" je object (pyVmomi.VmomiSupport.vim.ServiceInstance)

# MOB – příklad pro VM

- Property
- „Content“

Home

Managed Object Type: **ManagedObjectReference:ServiceInstance**
Managed Object ID: **ServiceInstance**

**Properties**

| NAME | TYPE | VALUE |
|---|---|---|
| capability | Capability | capability |
| content | ServiceContent | content |
| serverClock | dateTime | "2019-11-26T07:18:40.292274Z" |

**Methods**

| RETURN TYPE | NAME |
|---|---|
| dateTime | CurrentTime |
| HostVMotionCompatibility[] | QueryVMotionCompatibility |
| ServiceContent | RetrieveServiceContent |
| ProductComponentInfo[] | RetrieveProductComponents |
| Event[] | ValidateMigration |

# MOB – příklad pro VM

- Property „rootFolder“

| | | |
|---|---|---|
| localizationManager | ManagedObjectReference:LocalizationManager | ha-l10n-manager |
| overheadMemoryManager | ManagedObjectReference:OverheadMemoryManager | Unset |
| ovfManager | ManagedObjectReference:OvfManager | ha-ovf-manager |
| perfManager | ManagedObjectReference:PerformanceManager | ha-perfmgr |
| propertyCollector | ManagedObjectReference:PropertyCollector | ha-property-collector |
| rootFolder | ManagedObjectReference:Folder | ha-folder-root (root) |
| scheduledTaskManager | ManagedObjectReference:ScheduledTaskManager | Unset |
| searchIndex | ManagedObjectReference:SearchIndex | ha-searchindex |
| serviceManager | ManagedObjectReference:ServiceManager | ha-servicemanager |
| sessionManager | ManagedObjectReference:SessionManager | ha-sessionmgr |

# MOB – příklad pro VM

- Property „childEntity“ = ManagedObjectReference (MOR)
- Všechny DC
- Počítá se od [0]

Home

Managed Object Type: **ManagedObjectReference:Folder**
Managed Object ID: **ha-folder-root**

**Properties**

| NAME | TYPE | VALUE |
|---|---|---|
| alarmActionsEnabled | boolean | Unset |
| availableField | CustomFieldDef[] | |
| childEntity | ManagedObjectReference:ManagedEntity[] | ha-datacenter (ha-datacenter) |
| childType | string[] | "vim.Datacenter" |
| configIssue | Event[] | |
| configStatus | ManagedEntityStatus | "green" |
| customValue | CustomFieldValue[] | |
| declaredAlarmState | AlarmState[] | |
| disabledMethod | string[] | |

# MOB – příklad pro VM

- Property „vmFolder“

Home

Managed Object Type: **ManagedObjectReference:Datacenter**
Managed Object ID: **ha-datacenter**

**Properties**

| NAME | TYPE | VALUE |
|---|---|---|
| alarmActionsEnabled | boolean | Unset |
| availableField | CustomFieldDef[] | |
| configIssue | Event[] | |
| configStatus | ManagedEntityStatus | "green" |
| configuration | DatacenterConfigInfo | configuration |
| customValue | CustomFieldValue[] | |
| datastore | ManagedObjectReference:Datastore[] | 5812a8b2-4c17ace0-1909-f44d306206eb (VMFS_SSD)<br>583c4ea8-9334f768-1414-f44d306206eb (VMFS_SSD_NVME) |
| datastoreFolder | ManagedObjectReference:Folder | ha-folder-datastore (datastore) |
| declaredAlarmState | AlarmState[] | |
| disabledMethod | string[] | |
| effectiveRole | int[] | -1 |
| hostFolder | ManagedObjectReference:Folder | ha-folder-host (host) |
| name | string | "ha-datacenter" |
| network | ManagedObjectReference:Network[] | HaNetwork-VCHA (VCHA)<br>HaNetwork-VM Network (VM Network)<br>HaNetwork-VM_test (VM_test) |
| networkFolder | ManagedObjectReference:Folder | ha-folder-network (network) |
| overallStatus | ManagedEntityStatus | "green" |
| parent | ManagedObjectReference:Folder | ha-folder-root (root) |
| permission | Permission[] | |
| recentTask | ManagedObjectReference:Task[] | |
| tag | Tag[] | |
| triggeredAlarmState | AlarmState[] | |
| value | CustomFieldValue[] | |
| vmFolder | ManagedObjectReference:Folder | ha-folder-vm (vm) |

# MOB – příklad pro VM

- Property „childEntity“ (MOR)
- vms = datacenter.vmFolder.childEntity

Home

Managed Object Type: **ManagedObjectReference:Folder**
Managed Object ID: **ha-folder-vm**

**Properties**

| NAME | TYPE | VALUE |
|---|---|---|
| alarmActionsEnabled | boolean | Unset |
| availableField | CustomFieldDef[] | |
| childEntity | ManagedObjectReference:ManagedEntity[] | 1 (ZCS)<br>11 (VCSA)<br>2 (VBR)<br>3 (UBNT)<br>4 (SRV)<br>5 (MGW)<br>6 (LTSP)<br>7 (IOT)<br>8 (CONT)<br>(less...) |
| childType | string[] | "vim.VirtualMachine" |
| configIssue | Event[] | |
| configStatus | ManagedEntityStatus | "green" |
| customValue | CustomFieldValue[] | |
| declaredAlarmState | AlarmState[] | |
| disabledMethod | string[] | |
| effectiveRole | int[] | -1 |
| name | string | "vm" |
| overallStatus | ManagedEntityStatus | "green" |
| parent | ManagedObjectReference:Datacenter | ha-datacenter (ha-datacenter) |

# MOB – příklad pro VM

- Property „name“

```
datacenter = c.content.rootFolder.childEntity[0]
vms = datacenter.vmFolder.childEntity

for i in vms:
    print(i.name)
```

Home

Managed Object Type: **ManagedObjectReference:VirtualMachine**
Managed Object ID: **11**

**Properties**

| NAME | TYPE | VALUE |
|---|---|---|
| alarmActionsEnabled | boolean | Unset |
| availableField | CustomFieldDef[] | |
| capability | VirtualMachineCapability | capability |
| config | VirtualMachineConfigInfo | config |
| | Event[] | |
| | ManagedEntityStatus | "green" |
| | CustomFieldValue[] | |
| | ManagedObjectReference:Datastore[] | 5812a8b2-4c17ace0-1909-f44d306206eb (VMFS_SSD) |
| | AlarmState[] | |
| | string[] | "vim.ExtensibleManagedObject.setCustomValue"<br>"vim.ManagedEntity.destroy"<br>"vim.ManagedEntity.addTag"<br>"vim.ManagedEntity.removeTag"<br>"vim.ManagedEntity.retrieveCustomValues"<br>(more...) |
| effectiveRole | int[] | -1 |
| environmentBrowser | ManagedObjectReference:EnvironmentBrowser | 11-envmgr |
| guest | GuestInfo | guest |
| guestHeartbeatStatus | ManagedEntityStatus | "green" |
| layout | VirtualMachineFileLayout | layout |
| layoutEx | VirtualMachineFileLayoutEx | layoutEx |
| name | string | "VCSA" |
| network | ManagedObjectReference:Network[] | HaNetwork-VM Network (VM Network) |
| overallStatus | ManagedEntityStatus | "green" |
| parent | ManagedObjectReference:Folder | ha-folder-vm (vm) |
| parentVApp | ManagedObjectReference:ManagedEntity | Unset |
| permission | Permission[] | |
| recentTask | ManagedObjectReference:Task[] | |

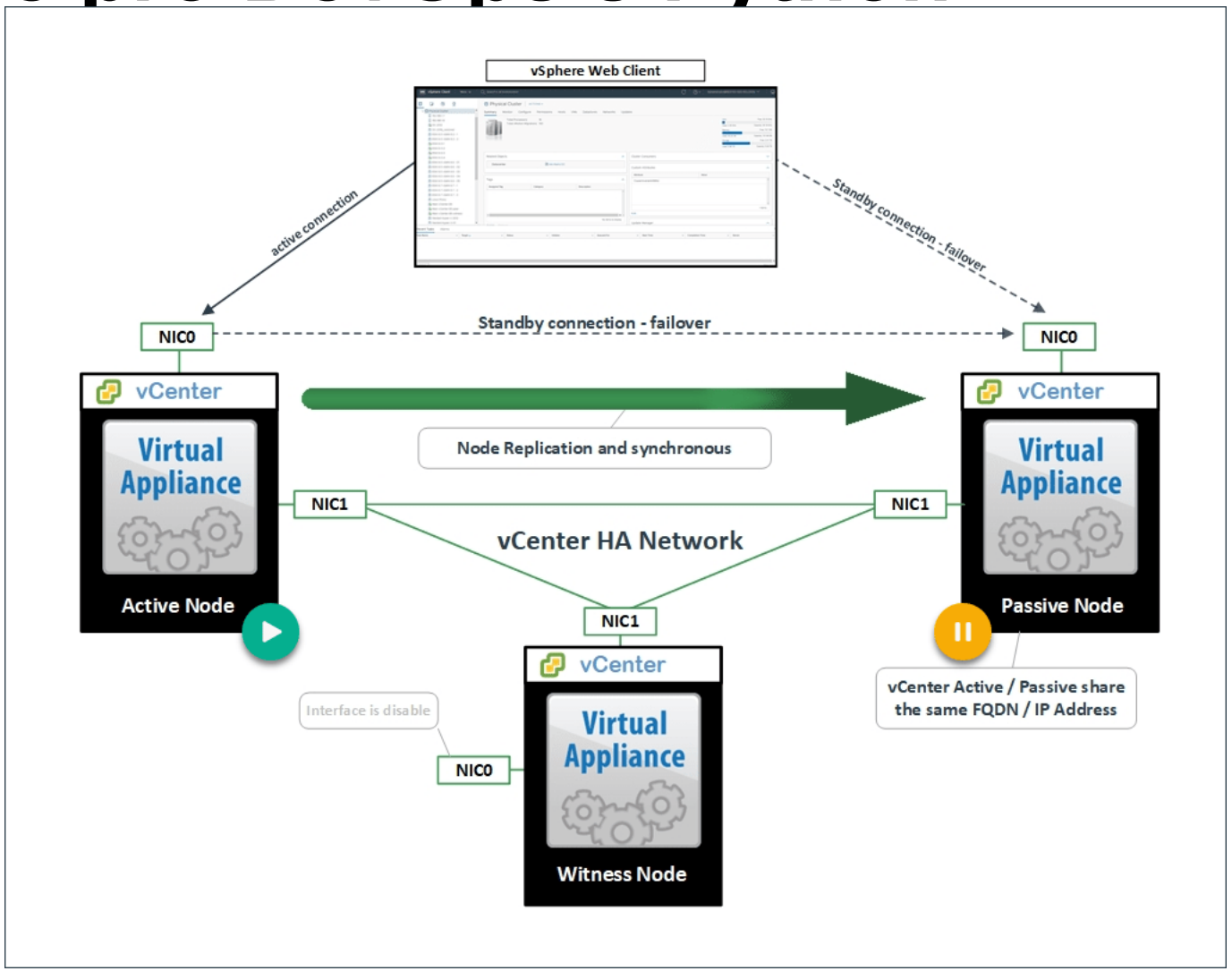
# MOB – příklad pro VM report

- vmreport.py
- **DEMO**

- **Community Samples**
- **https://github.com/vmware/pyvmomi-community-samples/tree/master/samples**

# Orchestrace vSphere pro DevOps s Python

- Orchestrace prostředí vSphere
- Deployment vCenter HA
- Znát funkcionalitu před automatizací
- Znát konfigurační požadavky
  - VCHA network
  - SSL thumbprint
- Prozkoumat API týkající se automatizované funkcionality
- **DEMO**

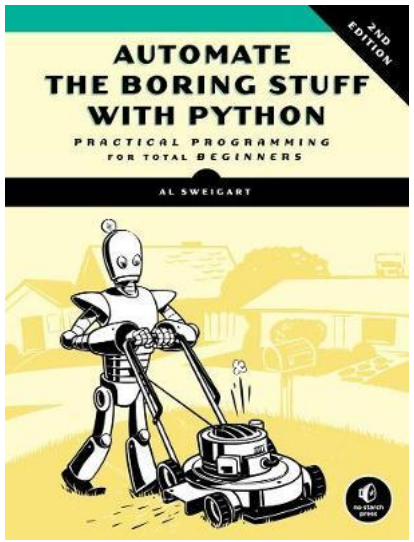


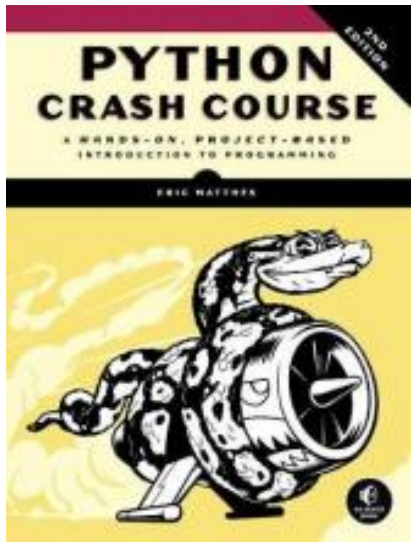
# Orchestrace vSphere vCenter HA

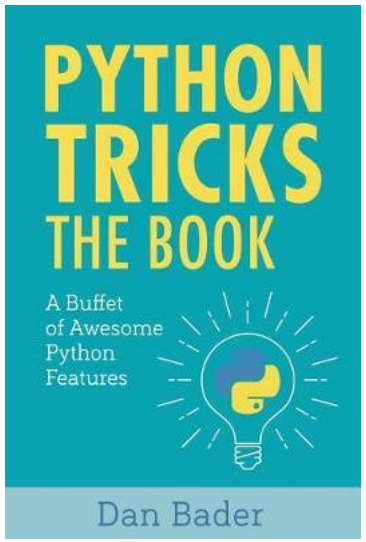
- Orchestrace konfigurace VCHA
- API týkající se konfigurované funkcionality
- **vim.vcha.FailoverClusterManager**
- **https://vdc-download.vmware.com/vmwb-repository/dcr-public/723e7f8b-4f21-448b-a830-5f22fd931b01/5a8257bd-7f41-4423-9a73-03307535bd42/doc/vim.vcha.FailoverClusterManager.html**
- **DEMO - failover**

# Jak na další vzdělávání v Pythonu?

- Knihy:
  - Automate the boring stuff with Python
  - https://automatetheboringstuff.com
  - Python Crash Course
  - Python Tricks the Book
  - Ponořme se do Pythonu 3
  - Python 3 – výukový kurz
- Školení pro začátečníky Pyladies
- https://pyladies.cz
- Setkání Python komunity Pyvo
- https://pyvo.cz






# Thank You


*„Není důležité si všechno pamatovat,*
*je však vhodné vědět, kde dohledat!“*

#automateitall