

Zerto **CON**

Resilient & Ready

May 21–23, 2018

REST Easy! Fear APIs, PowerShell, and Scripting No More

Justin Paul, Technical Alliances Architect

Mike Nelson, Cloud Architect, MVP



What is an API?

“...a set of clearly defined methods of communication between various software components.” --Wikipedia

Why Zerto Cmdlets and REST API's?

- Single script installations
- VRA, VPG, site and failover automation
- Metrics and Analytics extraction
- Export and integrate with 3rd party tools

Two Different Ways to Interact with Zerto

- PowerShell Cmdlets & REST APIs
 - Can be used in many different programming languages
 - PowerShell
 - Bash
 - Python
 - PHP
 - and others

PowerShell cmdlets

- Pros
 - Most IT Admins have already used PowerShell
 - Simple to get started
 - Can run on ZVM (or your desktop) with minimal new components
 - HUGE PowerShell Community
- Cons
 - Windows based (for the most part)
 - Must have Zerto PowerShell Cmdlets installed
 - Not all Zerto APIs are available

What is a REST API?

Two parts – REST and API

- API is an Application Programming Interface, which is a set of rules that allow programs to talk to each other
- REST is for Representational State Transfer, which determines what the API looks like. It is also a set of rules that developers follow when creating an API.
- Basically, it allows you to query an API via a URL and request a response, or send a change to the API via a URL and modify the data.

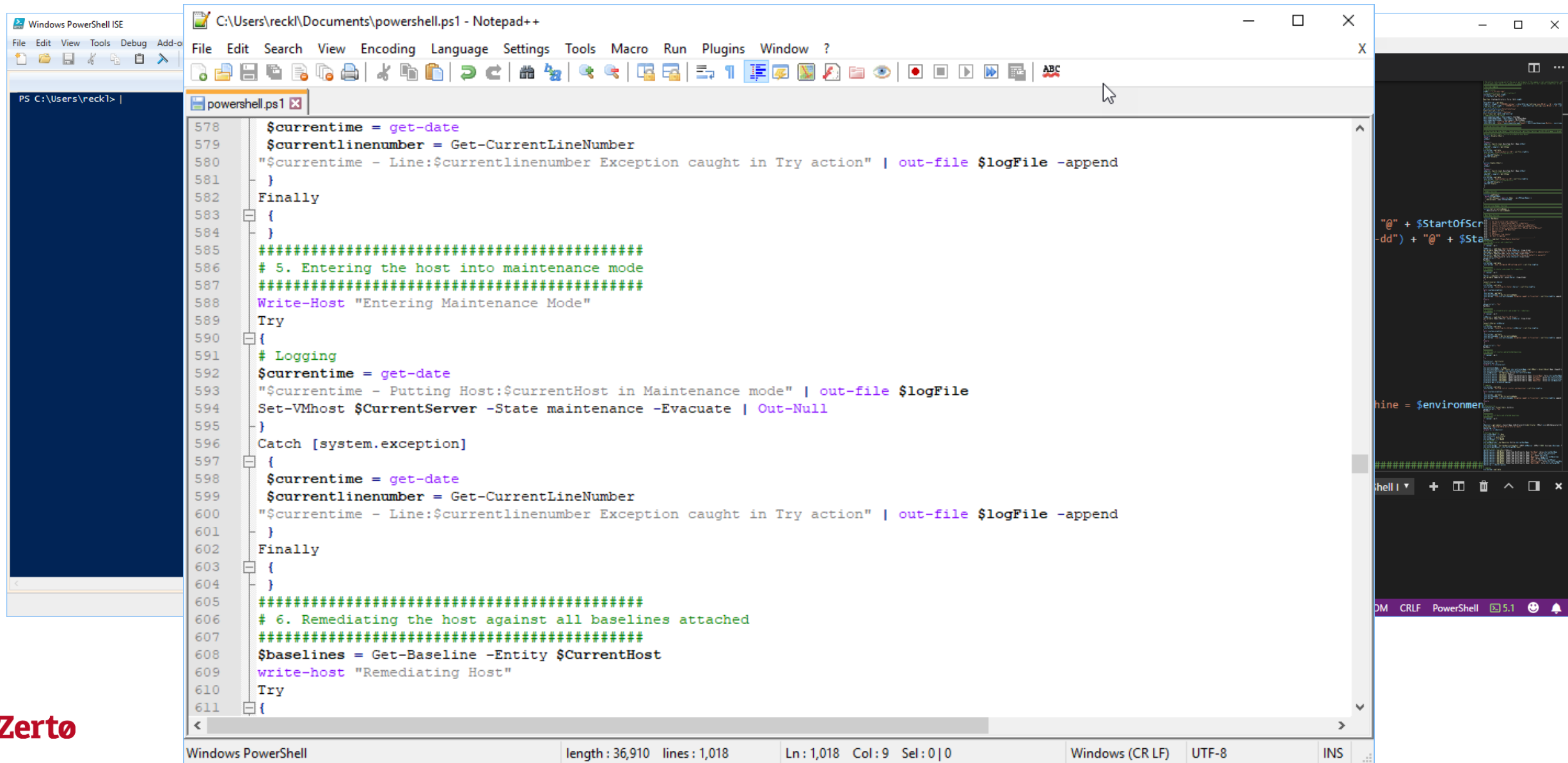
REST API

- Cons
 - More complex than PowerShell Cmdlets
 - Can take more time / longer scripts
- Pros
 - Many programming language choices (FLEXIBILITY!)
 - Native scripts for Linux (Bash, Python, etc)
 - Deeper automation capabilities with Zerto
 - Can be used in PowerShell with Zerto PowerShell API
 - No API software to install

How to Get Started

Editors and Prerequisites

Getting Started with PowerShell – Editors

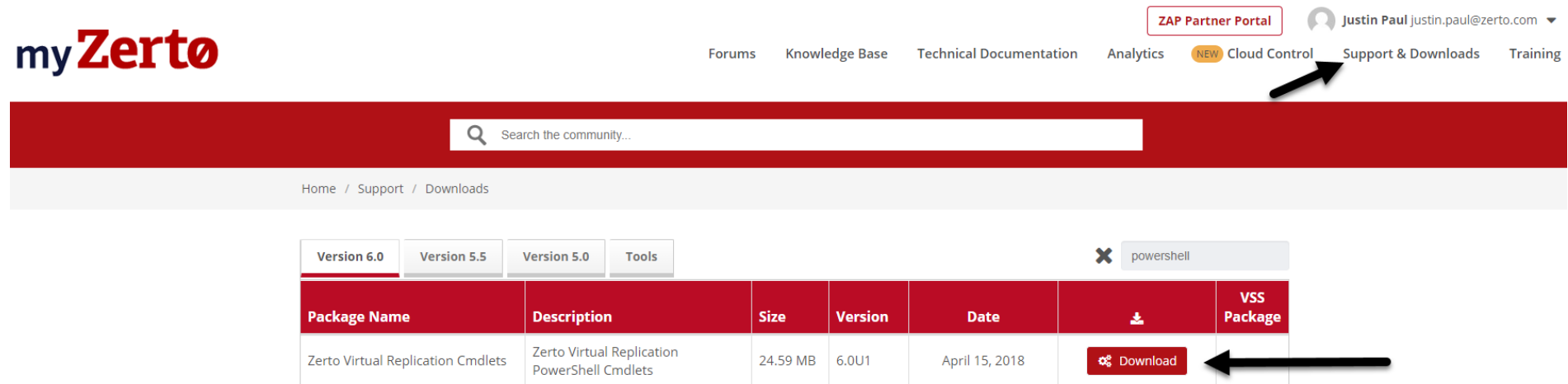


The image displays two side-by-side windows. The left window is the Windows PowerShell ISE, showing a command prompt with the prompt 'PS C:\Users\reck>'. The right window is Notepad++, editing a file named 'powershell.ps1'. The script in Notepad++ is a PowerShell script that handles exceptions and performs maintenance tasks. It includes comments in green and code in black. The script is divided into two main sections, one starting at line 578 and another at line 590. Both sections use a Try-Finally block to ensure logging and cleanup. The first section logs an exception and then enters maintenance mode. The second section logs an exception and then remediates the host against all baselines attached. The status bar at the bottom of Notepad++ shows 'Windows PowerShell', 'length: 36,910 lines: 1,018', 'Ln: 1,018 Col: 9 Sel: 0 | 0', 'Windows (CR LF)', 'UTF-8', and 'INS'.

```
578 $currenttime = get-date
579 $currentlinenumber = Get-CurrentLineNumber
580 "$currenttime - Line:$currentlinenumber Exception caught in Try action" | out-file $logFile -append
581 }
582 Finally
583 {
584 }
585 #####
586 # 5. Entering the host into maintenance mode
587 #####
588 Write-Host "Entering Maintenance Mode"
589 Try
590 {
591 # Logging
592 $currenttime = get-date
593 "$currenttime - Putting Host:$currentHost in Maintenance mode" | out-file $logFile
594 Set-VMhost $CurrentServer -State maintenance -Evacuate | Out-Null
595 }
596 Catch [system.exception]
597 {
598 $currenttime = get-date
599 $currentlinenumber = Get-CurrentLineNumber
600 "$currenttime - Line:$currentlinenumber Exception caught in Try action" | out-file $logFile -append
601 }
602 Finally
603 {
604 }
605 #####
606 # 6. Remediating the host against all baselines attached
607 #####
608 $baselines = Get-Baseline -Entity $CurrentHost
609 write-host "Remediating Host"
610 Try
611 {
```

Getting Started with PowerShell – Prerequisites

- Install Zerto PowerShell Cmdlets
 - Download from <https://zerto.com/myZerto/>
- Needs to be installed on whatever machine will run the script
 - Needs to be on the machine you will debug the script from too.



The screenshot shows the myZerto website interface. At the top, there is a navigation bar with links for Forums, Knowledge Base, Technical Documentation, Analytics, Cloud Control (marked as NEW), Support & Downloads, and Training. A user profile for Justin Paul is visible in the top right. Below the navigation bar is a search bar with the placeholder text "Search the community...". The main content area shows a breadcrumb trail: Home / Support / Downloads. There are tabs for Version 6.0, Version 5.5, Version 5.0, and Tools. A search filter "powershell" is applied. A table lists the available packages:

Package Name	Description	Size	Version	Date	Download	VSS Package
Zerto Virtual Replication Cmdlets	Zerto Virtual Replication PowerShell Cmdlets	24.59 MB	6.0U1	April 15, 2018	Download	

Two arrows are present: one pointing to the "Support & Downloads" link in the top navigation bar, and another pointing to the "Download" button in the table.

PowerShell Usage Tips

- PowerShell cmdlet port is 9080; not 9669!
- Change your default PowerShell administrator password
 - <http://bit.ly/2JPpeNH>
 - By default administrator password is “password”
- Refer to Zerto’s PowerShell cmdlet documentation for help
 - <http://bit.ly/2H5hJVn>
- Use a template to get started – “Git” it from here – <https://github.com/Zerto-TA-Public/Script-Templates/>

PowerShell Usage Tips

- `Add-PSSnapin Zerto.PS.Commands`
- `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Force`
- Use Try, Catch, and Finally
- Create re-usable code
- Don't reinvent the wheel
- Document your code

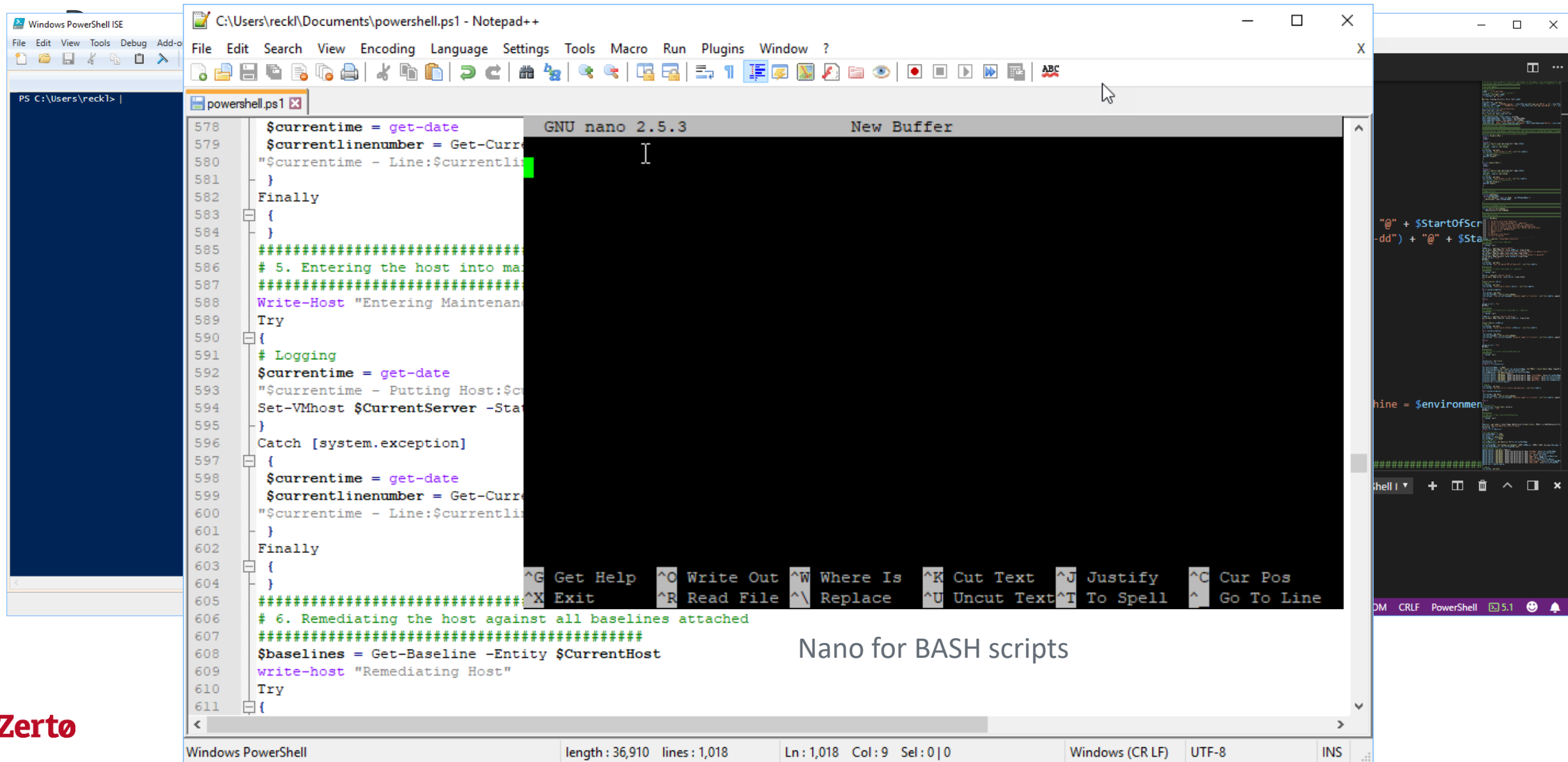
A Quick Test of Functionality

- Add-PSSnapin Zerto.PS.Commands
- Get-ProtectionGroups

```
PS C:\Users\Administrator> Add-PSSnapin Zerto.PS.Commands
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator> Get-ProtectionGroups

cmdlet Get-ProtectionGroups at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
SiteName: AWS-Beta
ZVMIP: 172.16.1.20
ZVMPort: 9080
Username: administrator
Password: password
AWS-Test-4x10GB
PS C:\Users\Administrator> _
```

Getting Started with REST API – Editors



Getting Started with REST APIs – Prerequisites

- Requirements depend on the programming language
 - for BASH we might use cURL to call our REST commands
 - For PowerShell “Invoke-RestMethod” is built in!
 - Google “REST API calls in <insert your programming language here>”
- Proper Authentication credentials
 - vSphere based credentials
 - HyperV / Domain credentials

REST API Usage Tips

- REST API uses familiar https port – 9669
- Commands cannot be ran on their own, Authentication Session must exist.
 - Sessions timeout after 30 minutes
- Automation with PowerShell and REST API whitepaper
 - <http://bit.ly/2qKazKO>
- Zerto REST API Online Help
 - <http://bit.ly/2qIESBH>
 - <http://bit.ly/2vvbk0c>

Accelerate Your Script Usage

Zerto has templates to get you started!

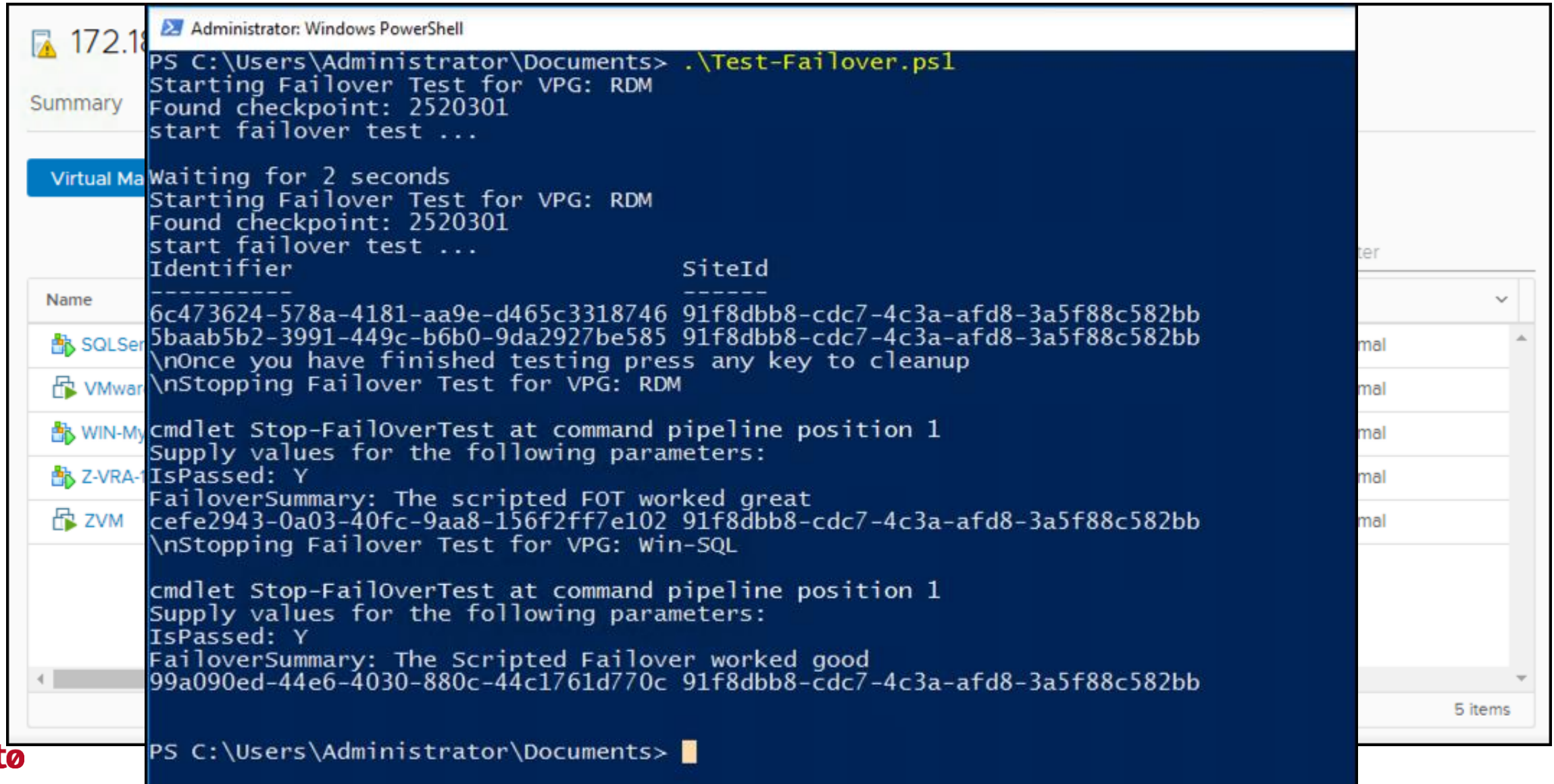
Online Template resources

- Tech Alliances team publishes to GitHub – <https://github.com/Zerto-TA-Public/>
- Tech Marketing team also publishes on GitHub – <https://github.com/Zerto-Tech-marketing>
- Great PowerShell Module conversion of REST APIs - <https://github.com/ChristopherGLewis/ZertoModule>
- Zerto API and PowerShell Guides on MyZerto
- Ask your SE!

Automating Failover Tests with Scripts

- <http://github.com/Zerto-TA-Public/Script-Templates/>
 - VPG-Test-Failover.ps1
- 100% PowerShell based
- Can be customized to F.O.T. as many VPGs as you wish
- Easily cleanup the test as well

Script Walk-through



```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Documents> .\Test-Failover.ps1
Starting Failover Test for VPG: RDM
Found checkpoint: 2520301
start failover test ...

Waiting for 2 seconds
Starting Failover Test for VPG: RDM
Found checkpoint: 2520301
start failover test ...
Identifier                               SiteId
-----
6c473624-578a-4181-aa9e-d465c3318746 91f8dbb8-cdc7-4c3a-afd8-3a5f88c582bb
5baab5b2-3991-449c-b6b0-9da2927be585 91f8dbb8-cdc7-4c3a-afd8-3a5f88c582bb
\nOnce you have finished testing press any key to cleanup
\nStopping Failover Test for VPG: RDM

cmdlet Stop-FailOverTest at command pipeline position 1
Supply values for the following parameters:
IsPassed: Y
FailoverSummary: The scripted FOT worked great
cefe2943-0a03-40fc-9aa8-156f2ff7e102 91f8dbb8-cdc7-4c3a-afd8-3a5f88c582bb
\nStopping Failover Test for VPG: Win-SQL

cmdlet Stop-FailOverTest at command pipeline position 1
Supply values for the following parameters:
IsPassed: Y
FailoverSummary: The Scripted Failover worked good
99a090ed-44e6-4030-880c-44c1761d770c 91f8dbb8-cdc7-4c3a-afd8-3a5f88c582bb

PS C:\Users\Administrator\Documents>
```

Summary

Virtual Machines

Name
SQLServer
VMware
WIN-MY
Z-VRA-1
ZVM

5 items

Installing VRAs using REST APIs

- Get the script from Github
<https://github.com/Zerto-Tech-Marketing/Bulk-VRA-Deployment/>
- Modify the CSV file with information for your environment
- Modify the PowerShell Script with credentials for Vmware and vCenter
- Run the PowerShell Script

Installing VRAs using REST APIs

Step 1

	A	B	C	D	E	F	G	H
1	ESXiHostName	DatastoreName	PortGroupName	VRAGroupName	MemoryInGB	DefaultGateway	SubnetMask	VRAIpAddress
2	172.16.1.11	SAN-1	VM Network	default	3	172.16.1.1	255.255.255.0	172.16.1.21
3	172.16.1.12	SAN-1	VM Network	default	3	172.16.1.1	255.255.255.0	172.16.1.22
4								
5								
6								

```
The cert policy original is TrustAllCertsPolicy
The cert policy is now TrustAllCertsPolicy
Executing {
  "DatastoreIdentifier": "d444d13e-ac24-403b-b956-3ffe5fadcc14.datastore-11",
  "GroupName": "default",
  "HostIdentifier": "d444d13e-ac24-403b-b956-3ffe5fadcc14.host-10",
  "HostRootPassword": null,
  "MemoryInGb": "3",
  "NetworkIdentifier": "d444d13e-ac24-403b-b956-3ffe5fadcc14.network-13",
  "UsePublicKeyInsteadOfCredentials": true,
  "VraNetworkDataApi": {
    "DefaultGateway": "172.16.1.1",
    "SubnetMask": "255.255.255.0",
    "VraIpAddress": "172.16.1.21",
    "VraIPConfigurationTypeApi": "Static"
  }
}
cb7063a2-dc87-4008-be02-2464f55bed36.be14a4c1-f2a7-4505-824d-44414ef1665c
Waiting 180 seconds before deploying the next VRA
```

Step 2

Step 3

Name	Target	Status
Deploy OVF template	Cluster1	0% <input type="text"/>
Uninstall	172.16.1.11	Completed
Delete datastore file	SAN-1	Completed
Delete datastore file	SAN-1	Completed
Install	172.16.1.11	Completed
Start service	172.16.1.11	Completed
Install VRA	172.16.1.11	10% <input type="text"/>

Installing VRAs using REST APIs

Zerto Site1-VMware 172.16.1.20 Unconfigured location 16:23 Unconfigured contact info

DASHBOARD VPGs VMs SITES **SETUP** OFFSITE BACKUP MONITORING REPORTS

Setup NEW VRA | EXPORT | MORE ▾ | ↗

VRAs ✓ INSTALLED | 2

DATASTORES ✓ AVAILABLE | 3

REPOSITORIES NO REPOSITORIES

☐ Show only hosts with VRA installed Views: GENERAL SETTINGS WORKLOAD PROTECTION ⚙

<input type="checkbox"/>	Host Address	Host Version		VRA Name	VRA Status	VRA Version	VRA Address	# VPGs	# VMs
<input type="checkbox"/>	172.16.1.11	6.0	✓	Z-VRA-172....	✓ Installed	Latest	172.16.1.21	0	0
<input type="checkbox"/>	172.16.1.12	6.0	✓	Z-VRA-172....	✓ Installed	Latest	172.16.1.22	0	0

Creating an automated recovery plan

- Use the template from FOT script
 - swap out “test failover” cmdlets for “live failover”
- Place on recovery site ZVM
- Probably can't script every scenario

Quiescent Checkpoints

- Zerto has both a REST API and PowerShell Cmdlets for inserting User Checkpoints.
- When used with Database (or application) specific scripts you can achieve a “clean” checkpoint
- Scripts already exist for MySQL on Linux, MySQL on Windows, MongoDB, and SAP Hana
- Get them from <https://github.com/Zerto-TA-Public/>

Questions?

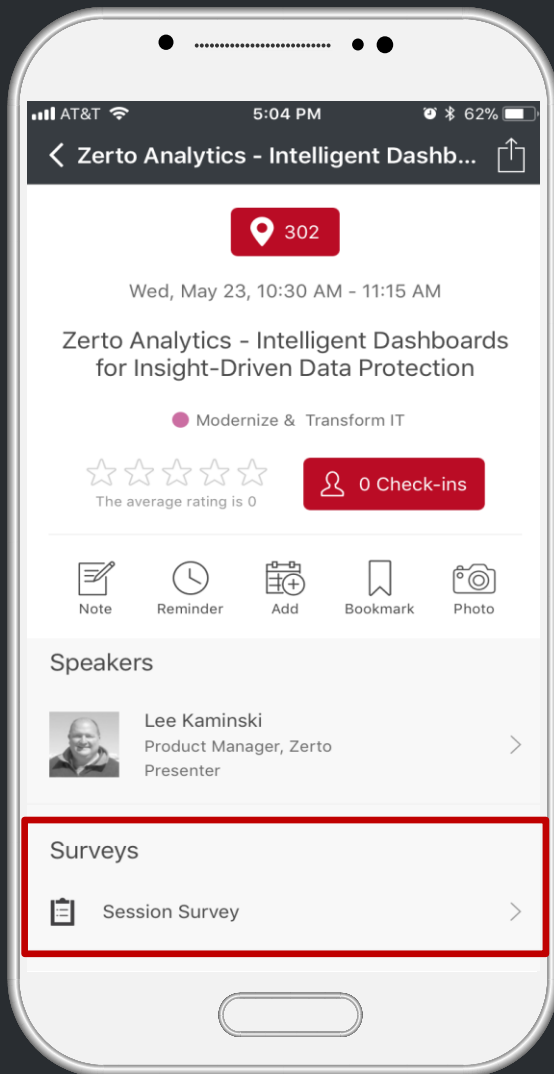
This session is part of the Zerto API Specialist Certification

Doing the certification?

Enter **NoFear** in the "test" on myZerto University to receive credit for this session.

Next step for certification

Attend: Beyond the Basics: API, Powershell, & Scripting.



Tell Us What You Think

1. Open the ZertoCON mobile app
2. Select the schedule
3. Find this session
4. Take the brief session survey

Thank You

2 0 1 8

Zerto **CON**

B O S T O N