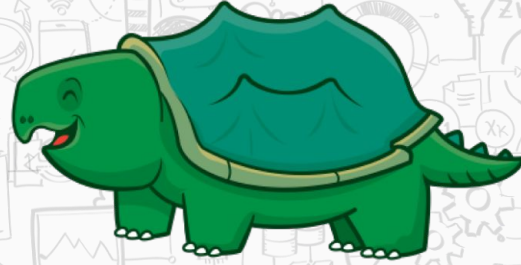




snap



Snap for Windows Capstone

Jesse Millar, Mat Kuhn, Phillip Anderson, Devin Durtschi, McKade Clements

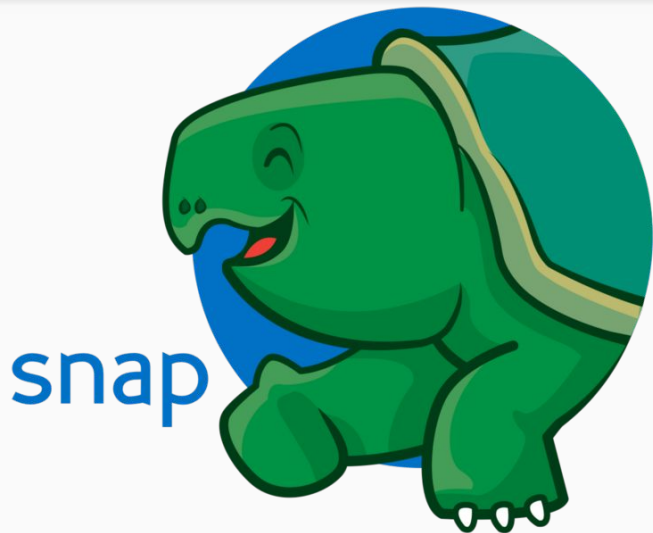
Doctor J. Ekstrom

What is it?

Snap is an open source project started by Intel in July of 2015

Written in Go, but recently made expandable to encompass the usage of other languages

A single API for collecting telemetry data



the open telemetry framework

\$ go get github.com/intelsdi-x/snap

Objective Statement

Automate the Snap build process and create three Snap data collection plugins for Windows, including Perfmon, Sysinternals, and Active Directory by March 20th.

Snap Build Process

Prior to this project, only supported Unix based systems

Built in Go which cross compiles

Makefiles don't work natively on Windows Server

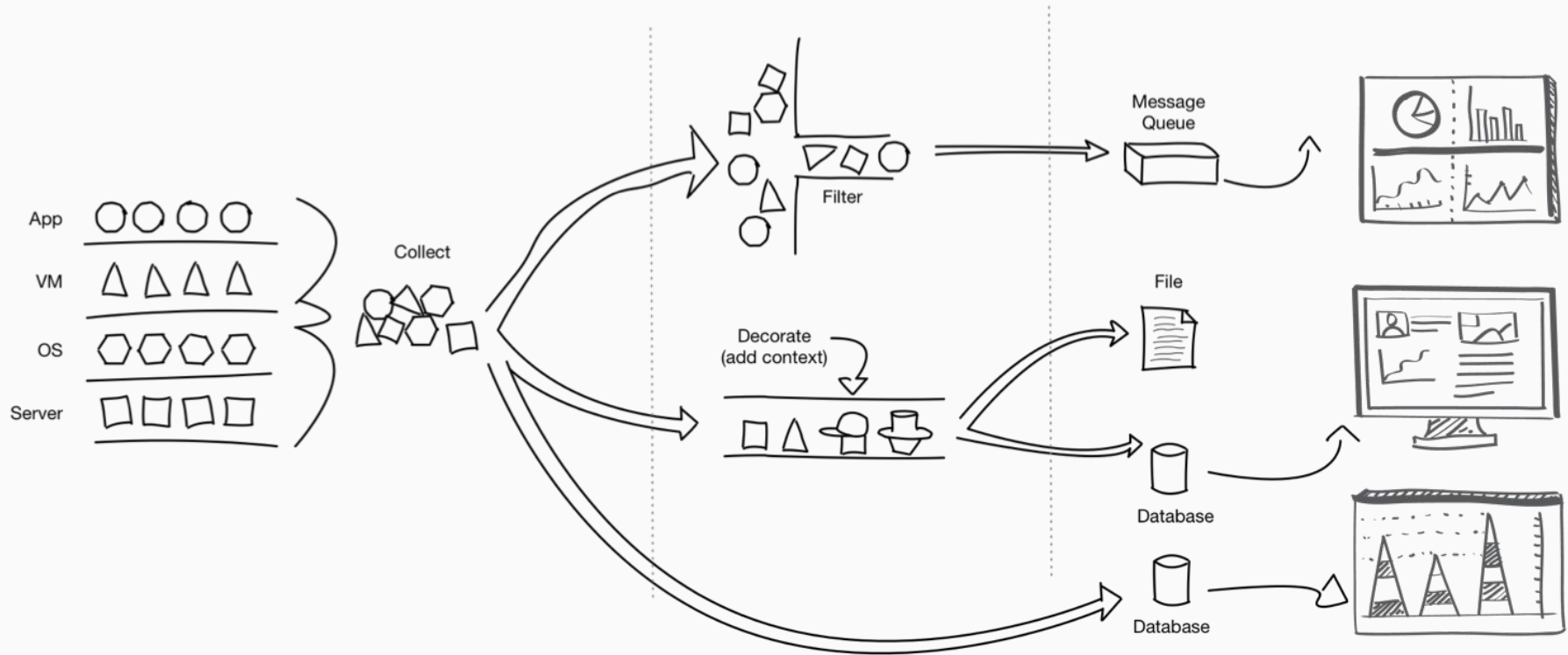
Looked at CMake or Windows batch script as a possible solution
(Successfully accomplished this, but wanted an automated solution as well)

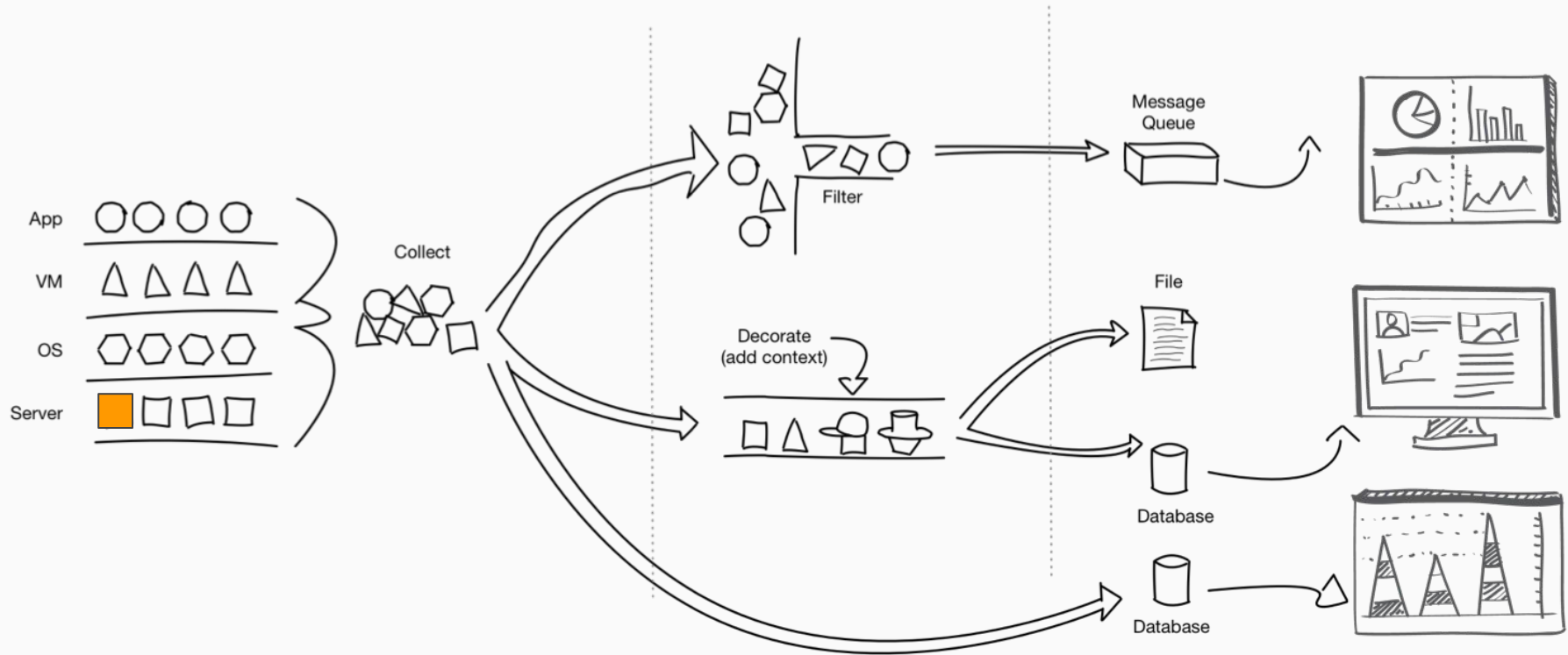
Snap Build Process (cont'd)

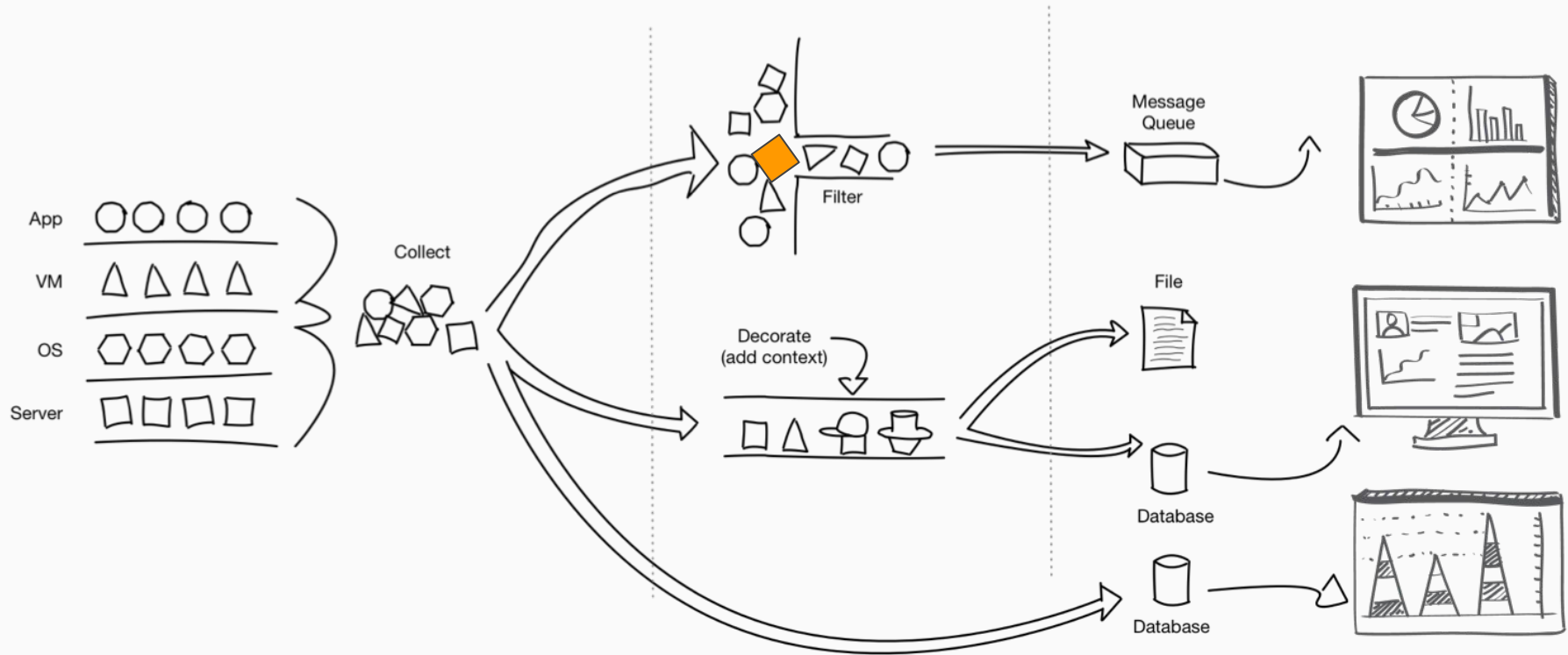
Used WiX and Visual Studio to create an MSI

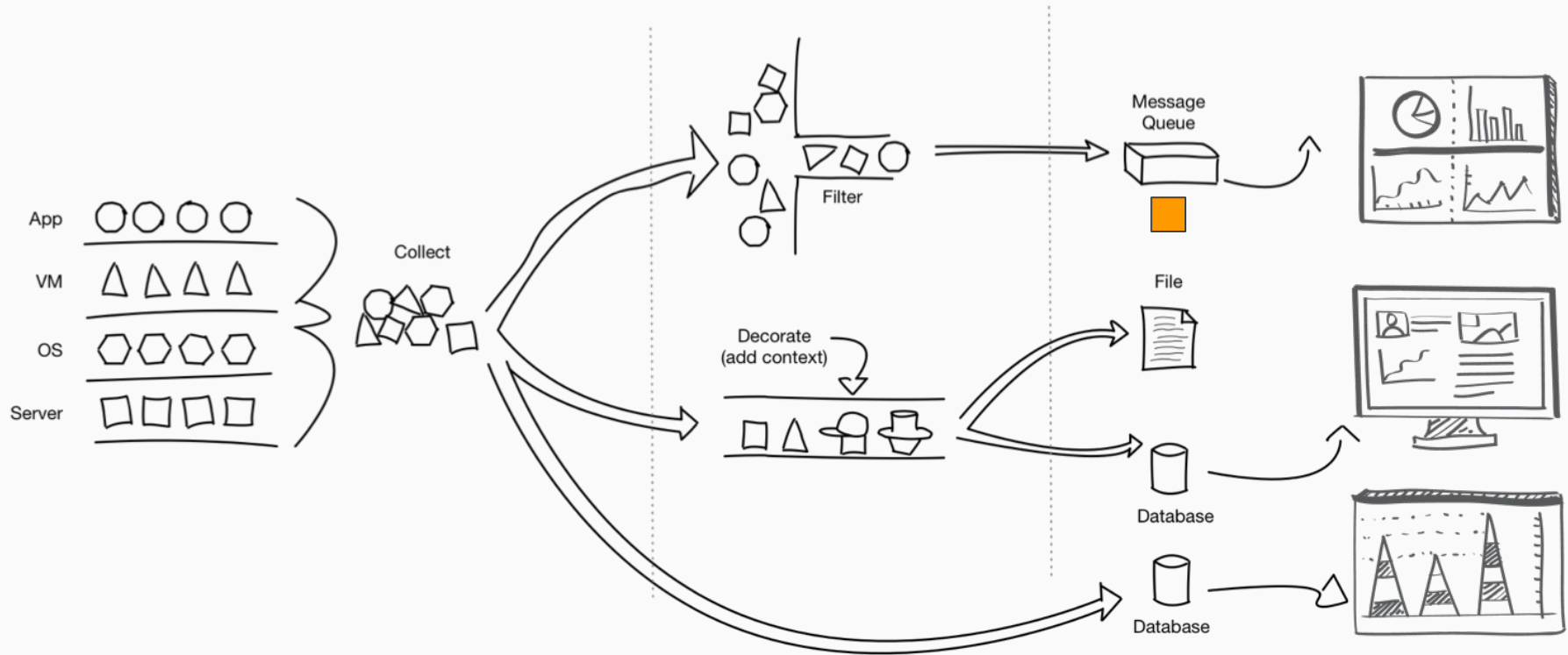
- Puts the compiled binaries in the right places
- Creates a service to launch the Snap agent automatically on startup





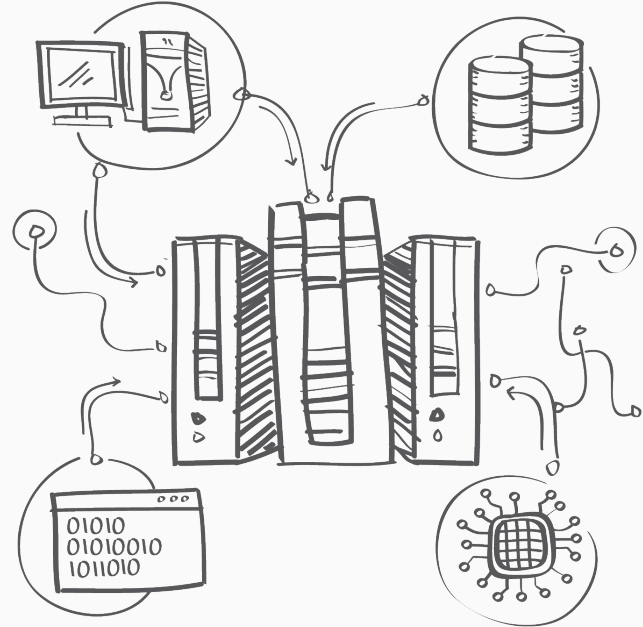






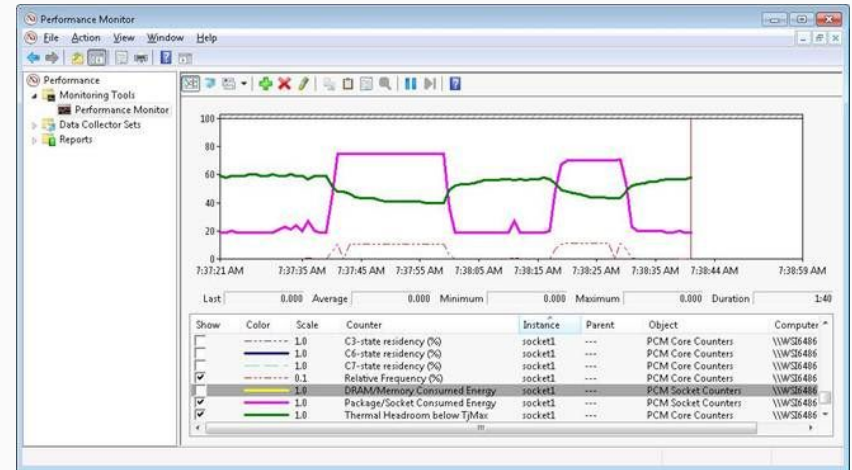
Plugins

1. Perfmon
2. SysInternals
3. Active Directory



Perfmon

- Powershell v3.0+: Get-Counters
- Concurrency
 - Prevented most timeouts
 - Mutex around map
- Glide for package management
- 12 commonly used metrics
 - Easily extensible
- GRPC library switch

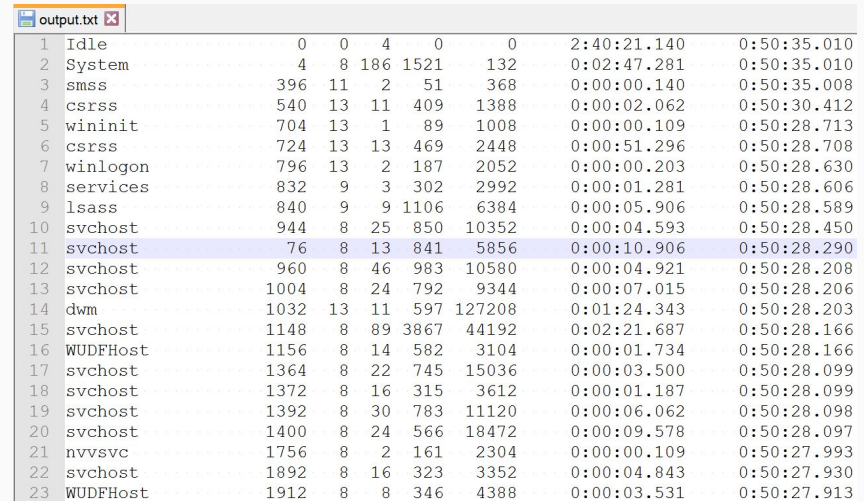


SysInternals - PsList

Runs an executable to collect the number of running processes, threads, and handles

Uses PsTools - PsList.exe

Glide for package management

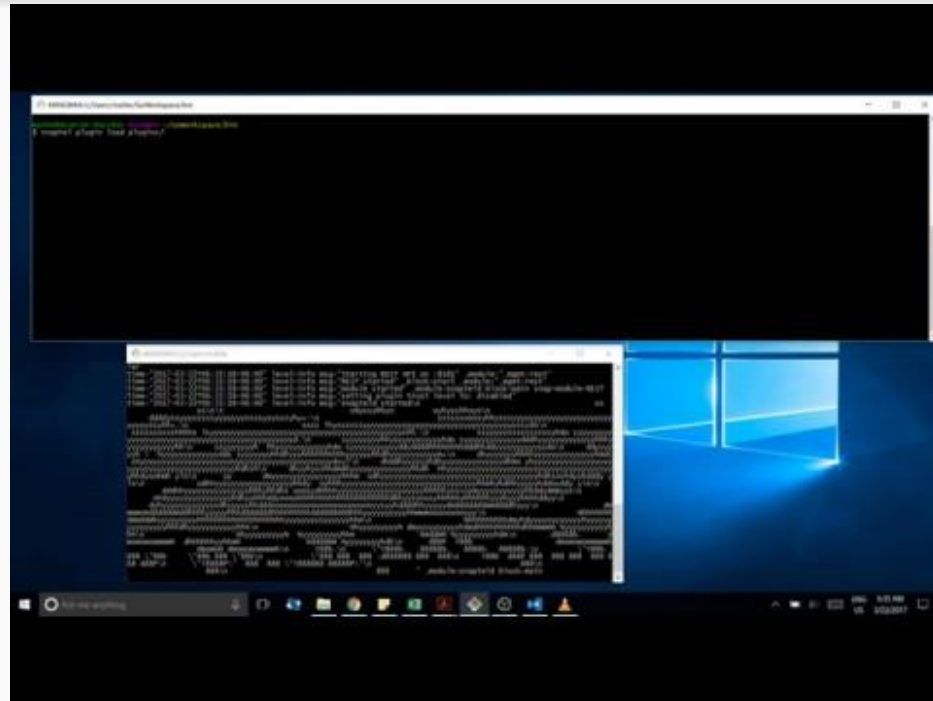


PID	Name	Private Bytes	Working Set	Paged Pool	Non-paged Pool	Session ID	Process ID	Thread ID	Start Time
1	Idle	0	0	4	0	0			2:40:21.140
2	System	4	8	186	1521	132			0:02:47.281
3	smss	396	11	2	51	368			0:00:00.140
4	csrss	540	13	11	409	1388			0:00:02.062
5	wininit	704	13	1	89	1008			0:00:00.109
6	csrss	724	13	13	469	2448			0:00:51.296
7	winlogon	796	13	2	187	2052			0:00:00.203
8	services	832	9	3	302	2992			0:00:01.281
9	lsass	840	9	9	1106	6384			0:00:05.906
10	svchost	944	8	25	850	10352			0:00:04.593
11	svchost	76	8	13	841	5856			0:00:10.906
12	svchost	960	8	46	983	10580			0:00:04.921
13	svchost	1004	8	24	792	9344			0:00:07.015
14	dwm	1032	13	11	597	127208			0:01:24.343
15	svchost	1148	8	89	3867	44192			0:02:21.687
16	WUDFHost	1156	8	14	582	3104			0:00:01.734
17	svchost	1364	8	22	745	15036			0:00:03.500
18	svchost	1372	8	16	315	3612			0:00:01.187
19	svchost	1392	8	30	783	11120			0:00:06.062
20	svchost	1400	8	24	566	18472			0:00:09.578
21	nvsvc	1756	8	2	161	2304			0:00:00.109
22	svchost	1892	8	16	323	3352			0:00:04.843
23	WUDFHost	1912	8	8	346	4388			0:00:03.531

Active Directory

- Similar to Perfmon plugin
- Powershell v3.0+: Get-Counters
- Concurrency
 - Prevented most timeouts
 - Mutex around map
- Glide for package management
- 19 requested metrics
 - From LDAP to Kerberos

Demo - Perfmon Plugin



Motivation

- Expand Snap's impact
- Greater ease in system management
- Telemetry that runs best on Intel Architecture (IA)
- Attempts to modernize gathering, processing, publishing telemetry data

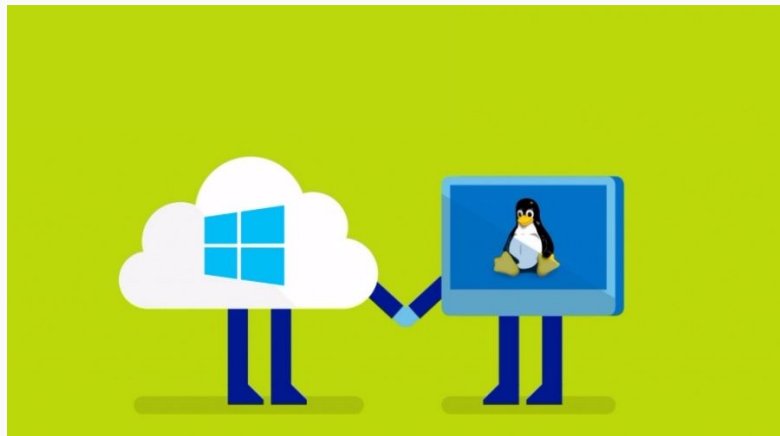
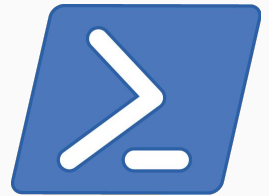
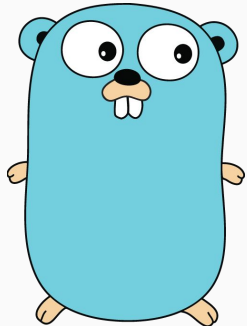
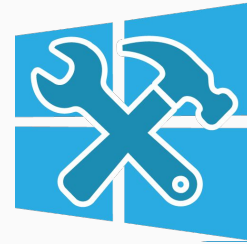


Image source: <https://fosbytes.com/microsoft-ubuntu-linux-windows-10-creators-update/>

Technologies Involved



Future Work

- Linux plugin conversions -> Windows (E.g. Docker, etc.)
- Convert Powershell commands to Windows Management Instrumentation
- Build scripts and MSI installers for each plugin

Lessons Learned

- Understanding pre-built framework and requirements
- Nature of open-source projects
 - Changing documentation
 - Changing code bases
 - Deprecated library
 - Code review process
- Risk Management
 - Bugs in code
 - Timeout factors
 - Native build script to bash
- Go language, Git/GitHub, PowerShell

Questions?