A scenic photograph of a waterfall flowing over large, light-colored rocks. The water is white and turbulent as it falls. The background is filled with dense green trees and foliage. A blue rectangular overlay with the letters "STM" in white is positioned in the upper center of the image.

STM

# Splunk App for Stream

David Shpritz, Aplura LLC.

Baltimore Area User Group

3/21/2016

```
32616  splunkd
32695  tcpdump
590   named
1602   vim
1243   python
```

```
%CPU  USER          UID      0 used.  615976 avail Mem
0.3  splunk        1000    0:02.80  4.3  20  0  135356  43956
0.3  tcpdump       72     0:02.45  0.7  20  0  28588  6872
0.3  named         25     7:31.12  1.8  20  0  548272  18872
0.3  dan           1028    0:00.08  0.6  20  0  152828  6248
```

# Agenda

- What is Splunk App for Stream?
- Why use Steam?
- Where to use Stream?
- Deploying Stream
- Questions

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUID RUSER	ST
32616	splunkd												
32695	tcpdump												
590	named												
1602	vim												
1243	python												

# What Is Splunk App for Stream?



32616 splunkd  
32695 tcpdump  
590 named  
1602 vim  
1243 python

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RD	WR	RS	WS
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	1800	0	1800	0
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	72	72	72
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	25	25	25
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1821	1821	1821	1821



# Some history

- Splunk acquires Cloudmeter, December 2013
- Renamed Splunk App for Stream
- Released with Splunk 6.0 (August, 2014)
- Now at version 6.4.3 (January, 2016)

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUID RUSER	ST
32616	splunkd												
32695	tcpdump												
590	named												
1602	vim												
1243	python												

# Purpose of Stream

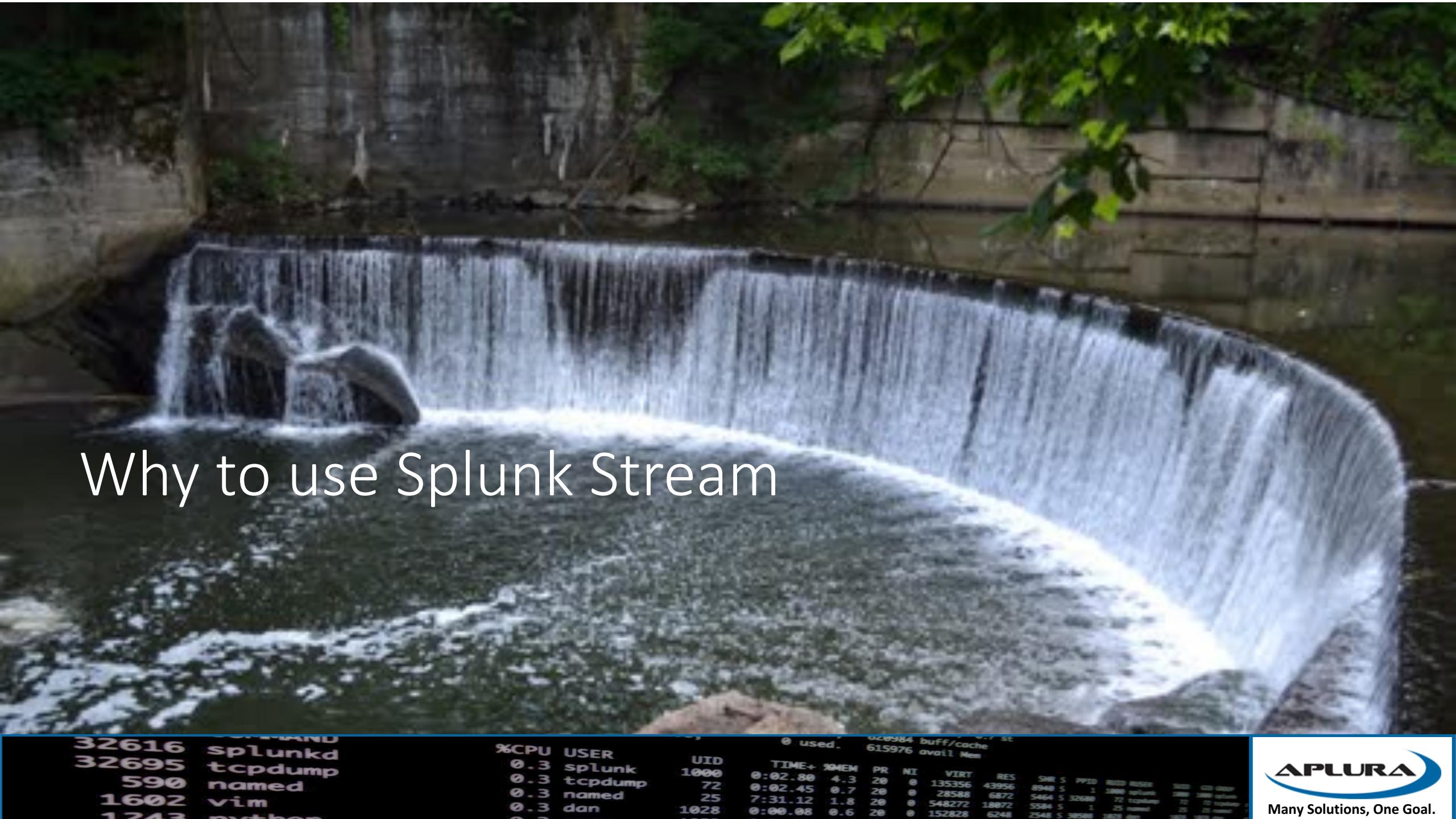
- Rapid deployment
- Rapid configuration
- Capture wire data
- Interpret wire data
- Summarize/filter/aggregate
- Index
- Kind of like Bro, but more Splunky, and GUI

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RSS	RSER	ST
32616	splunkd													
32695	tcpdump													
590	named													
1602	vim													
1243	python													

# So what can we capture?

- Well, we aren't really capturing and indexing packets
- Forwarders capture packets, analyze the protocols
- What protocols (a lot):
  - TCP/UDP
  - Application protocols (HTTP, databases, email, file sharing, chat)
  - About 30 different protocols currently
  - <http://docs.splunk.com/Documentation/StreamApp/latest/DeployStreamApp/Whattypeofdatadoesthisappcollect>

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RD	WR	RSER	SLRU	GRU	DR
32616	splunkd																
32695	tcpdump																
590	named																
1602	vim																
1243	python																



# Why to use Splunk Stream

		%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RD	WR	RSER	SAQD	GD	DR
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	1800	splunk	3400	2000	0	
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump	3400	2000	0	
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	72	25	0	
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1829	dan	3400	2000	0	
1243	python	0.3																	

# No logs

- No ownership
  - No visibility
  - No forwarders (as endpoints)
  - No logging options



# Poor logs

- Logging is high overhead
- Logs make no sense
- Key events are not logged



		%CPU	USER	UID	% used.	0 used.	buff/cache	avail Mem
<b>32616</b>	<b>splunkd</b>	<b>0.3</b>						
<b>32695</b>	<b>tcpdump</b>	<b>0.3</b>	<b>splunk</b>	<b>1000</b>	<b>0:02.80</b>	<b>4.3</b>	<b>20</b>	<b>0 135356 43956</b>
<b>590</b>	<b>named</b>	<b>0.3</b>	<b>tcpdump</b>	<b>72</b>	<b>0:02.45</b>	<b>0.7</b>	<b>20</b>	<b>0 28588 6872</b>
<b>1602</b>	<b>vim</b>	<b>0.3</b>	<b>named</b>	<b>25</b>	<b>7:31.12</b>	<b>1.8</b>	<b>20</b>	<b>0 548272 18872</b>
<b>1243</b>	<b>python</b>	<b>0.3</b>	<b>dan</b>	<b>1028</b>	<b>0:00.08</b>	<b>0.6</b>	<b>20</b>	<b>0 152828 6248</b>

# Cloud

- Many cloud services don't offer logs on things
- No chokepoints



		%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SRR	S	PPID	NAME	ST
32616	splunkd														
32695	tcpdump	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	18000 splunk	
590	named	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	3	32688 tcpdump	
1602	vim	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25 named	
1243	python	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	3	38588 1828 dan	

# VS. Bro IDS

- Lower CPU usage
- Lower RAM usage
- More OS support (Linux, Windows, OSX)

**But**

- High traffic requires network packet brokers (Gigamon, Ixia, etc.)
- Can't write your own interpreters
- No Snort rules



%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SRR	PPID	NIOS RUSER	SILO	GRD	CHUNK	SHM	PPID	NIOS RUSER	SILO	GRD	CHUNK
32616	splunkd																			
32695	tcpdump																			
590	named																			
1602	vim																			
1243	python																			

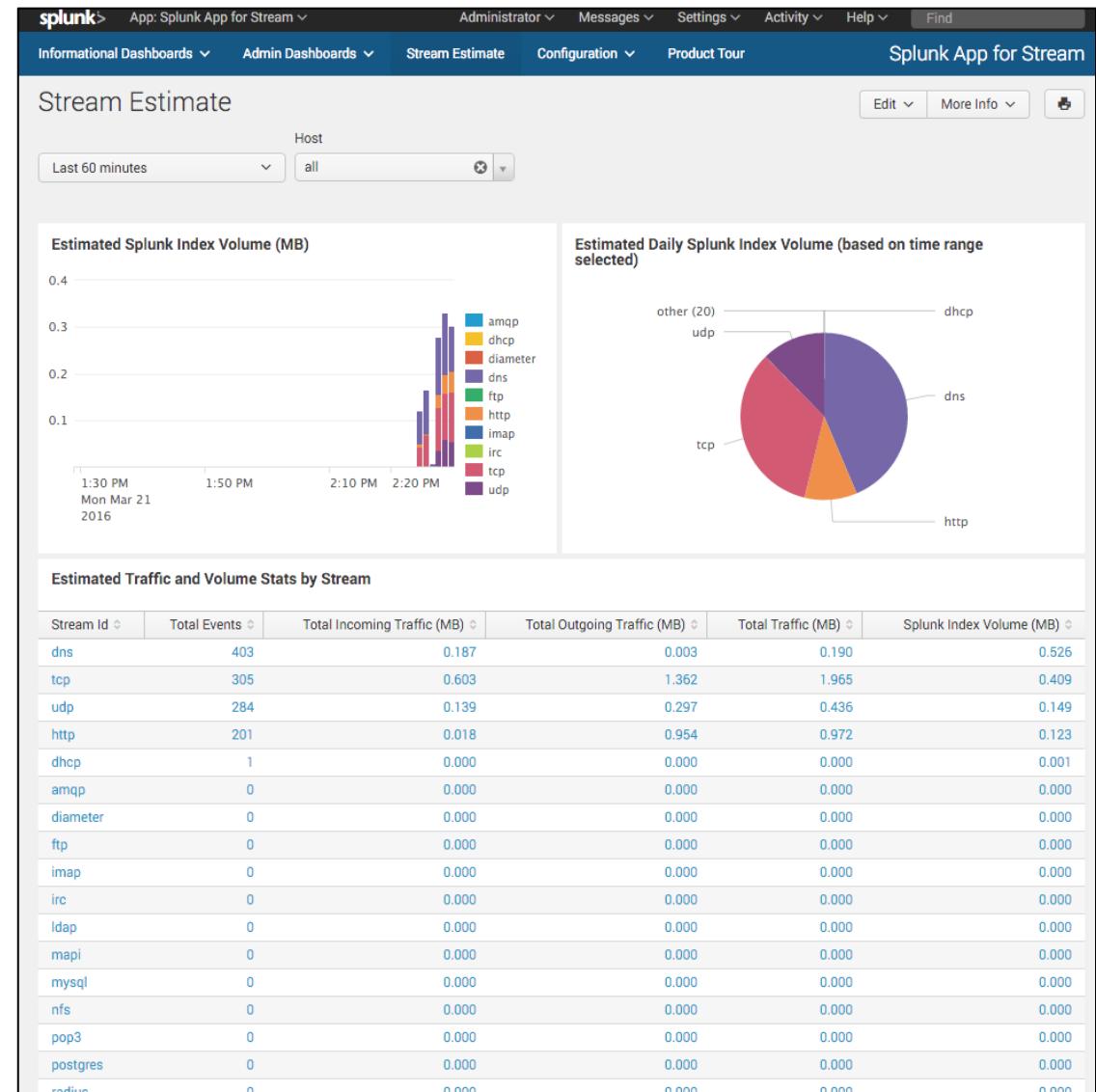
# Other features

- Filtering
- Aggregation
- Ephemeral Streams (short term)
- SSL decrypt
- Centralized management
- Integration with ES
  - Start a stream after Notable event
  - Protocol analysis dashboards

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUUD RUUDER	ST
32616	splunkd												
32695	tcpdump	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1 1800 splunk
590	named	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32695 72 tcpdump
1602	vim	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1 25 named
1243	python	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588 1629 dan

# Data Estimation

- “What if I turn this on?”
- Tells you how much data you would be indexing



```
32616  splunkd
32695  tcpdump
590   named
1602   vim
1243   python
```

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	PPID	NIUSER	ST
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	5	1800 splunk	615976
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	5	32689 72 tcpdump	buff/cache
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	1	25 named	avail Mem
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	5	38588 1829 dan	used.

# Granular control of the data

- Not just which systems, but also what data, which fields

Configure Stream - http  
HTTP Protocol Events

< Back to streams

Mode: Enabled | Estimate | Disabled

Splunk Index: default

Protocol: HTTP

Aggregation: No | Yes, every [ ] seconds

Filters: 0 filters configured | View Filters

Fields:

Enable	Name	Description	Type	Term	Action
<input checked="" type="checkbox"/>	bytes	The total number of bytes transferred	Original	flow.bytes	
<input checked="" type="checkbox"/>	bytes_in	The number of bytes sent from client to server	Original	flow.cs-bytes	
<input checked="" type="checkbox"/>	bytes_out	The number of bytes sent from server to client	Original	flow.sc-bytes	
<input checked="" type="checkbox"/>	cookie	The Cookie HTTP request header	Original	http.cookie	
<input type="checkbox"/>	dest_ip	IP address of the server in dot-quad notation	Original	flow.sip	

```
32616  splunkd
32695  tcpdump
590    named
1602   vim
1243   python
```

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUSS	ST
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	1800	splunk
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1829	dan

# Global Filters

- Filter out noise from the enterprise
- Things like vulnerability scanners

## IP Address Filters

Use whitelist/blacklist filter rules to capture/ignore network data based on IP address. [?](#)

### Whitelist IP Addresses

Define a whitelist to capture data from IP addresses on that list only.

Add Ip Address  Save

### Blacklist IP Addresses

Define a blacklist to ignore those IP addresses, and allow data capture from all other IP addresses.

123.4.56.78

Add Ip Address  Save

%CPU	USER	UID	% used.	TIME+ / 34MEM	PR	NI	VIRT	RES	SRR	S	PPID	RUUD RUSER	ST	
32616	splunkd			0:02.80	4.3	20	0	135356	43956	8948	S	1 18000 splunkd		
32695	tcpdump			0:02.45	0.7	20	0	28588	6872	5464	S	32689 72 tcpdump		
590	named			25	7:31.12	1.8	20	0	548272	18872	5584	S	25 named 72 25 named	
1602	vim			0:00.08	0.6	20	0	152828	6248	2548	S	38588 1629 vim		
1243	python													

# Distributed Forwarder Management

- Set up groups for capture
- Uses regex for groups on the “Forwarder ID”
- Forwarder ID is configurable via XML config file
- Yes, it’s another Splunk deployment/control mechanism

Distributed Forwarder Management							
Create Stream Forwarder groups using pattern match.							
2 groups		<a href="#">X</a>					
i	Name	Description	Rule	Include Ephemeral Streams?	Contains Streams	Actions	
>	AppServers	These are my app servers that get fun stuff	^.*Pro.*\$	Yes	3	<a href="#">▼</a>	
>	defaultgroup	Used when there is no matching group found for a given stream forwarder ID		Yes	41	<a href="#">▼</a>	

```
32616  splunkd
32695  tcpdump
590    named
1602   vim
1243   python
```

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SRR	S	PPID	RUED RUSER	ST
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	18000 splunk	615976 avail Mem
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72 tcpdump	
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25 named	
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1828 dan	



# Where to use Splunk Stream

```
32616 splunkd  
32695 tcpdump  
590 named  
1602 vim  
1243 python
```

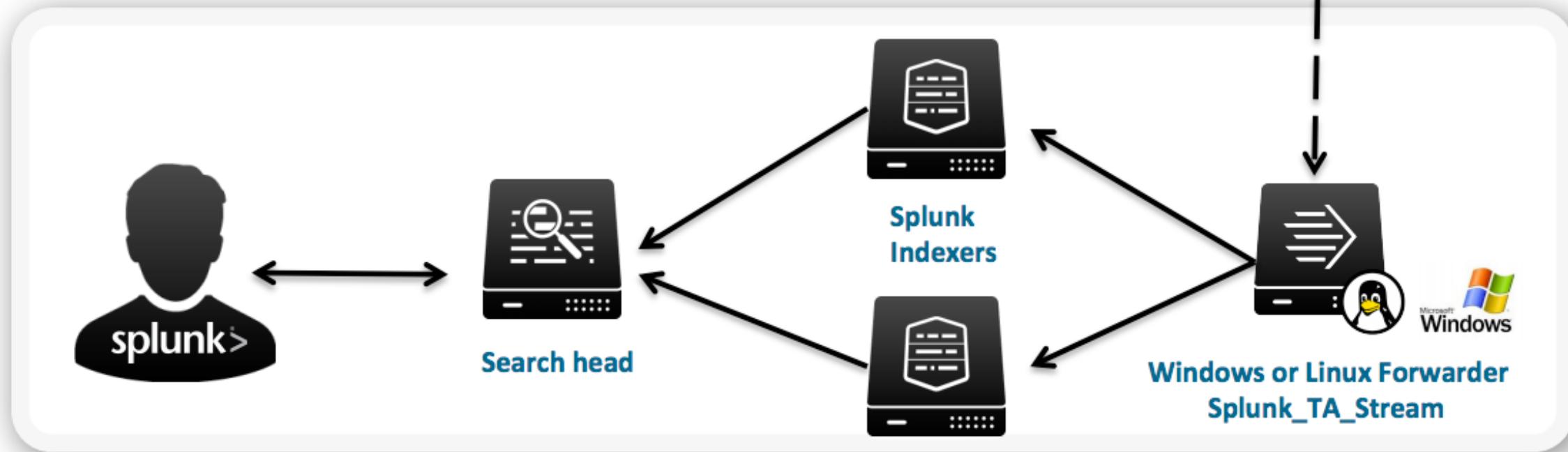
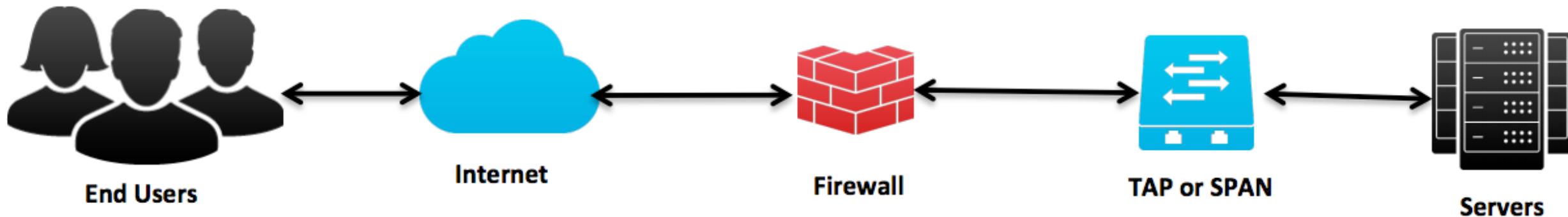
%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SRR	PPID	RD RUSER	SWAP	GD RUSER	
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1800 splunk	0	1800	
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	32689	
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25 named	72	25 named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1821	38588	1821 dan



# Dedicated Stream Forwarders

- Send data off of a switch Span or Tap
- Tools like Gigamon, Ixia, Etc.
  - You need these for really big pipes to spread the love
- Purpose built
  - Higher CPU and RAM
  - Better network cards
- Also a good option is you want to perform SSL decrypt
- Note that if you do this you will want to change some of your kernel settings (buffer sizes)
- Make sure to monitor your forwarders for thruput warnings!

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RD NRSER	SWAP	GRD	OPERS
32616	splunkd														
32695	tcpdump														
590	named														
1602	vim														
1243	python														

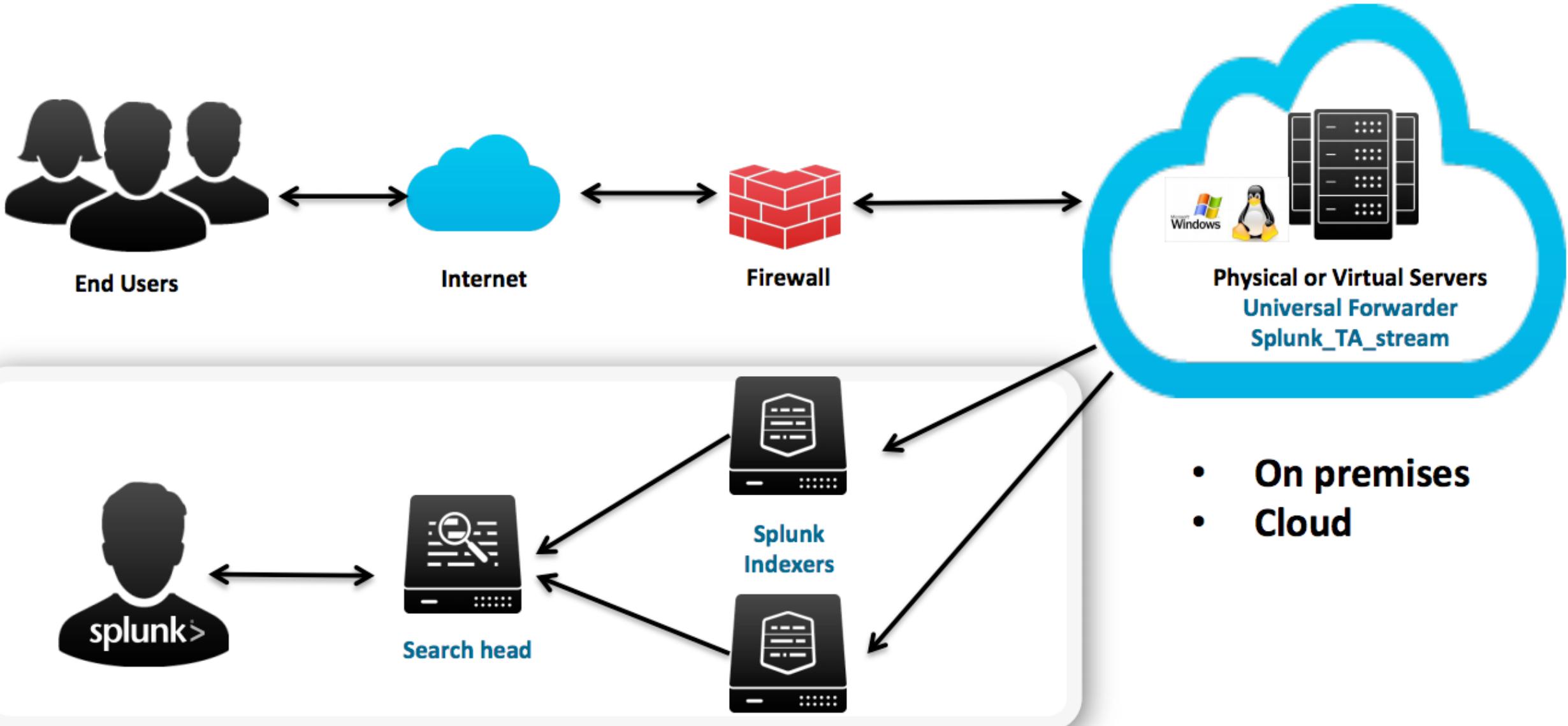


32616	COMMAND							
32695	splunkd							
32695	tcpdump							
590	named							
1602	vim							
1243	python							
		%CPU	USER	UID	TIME+	%MEM	PR	NI
		0.3	splunk	1000	0:02.80	4.3	20	0
		0.3	tcpdump	72	0:02.45	0.7	20	0
		0.3	named	25	7:31.12	1.8	20	0
		0.3	dan	1028	0:00.08	0.6	20	0
		0 used.						
		615976	buff/cache					
		avail Mem						

# Deploy to the Endpoints

- Deploy directly to the systems you want to monitor
- Good for application debugging
- Nice option for Splunk ES
- Can be done from Deployment Server
- Granular control over groups
- Could mean a lot of “hand on”

		%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUID	RUSER	ST
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	1860	splunk	0000
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump	0000
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	72
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1829	dan	0000
1243	python	0.3														



```

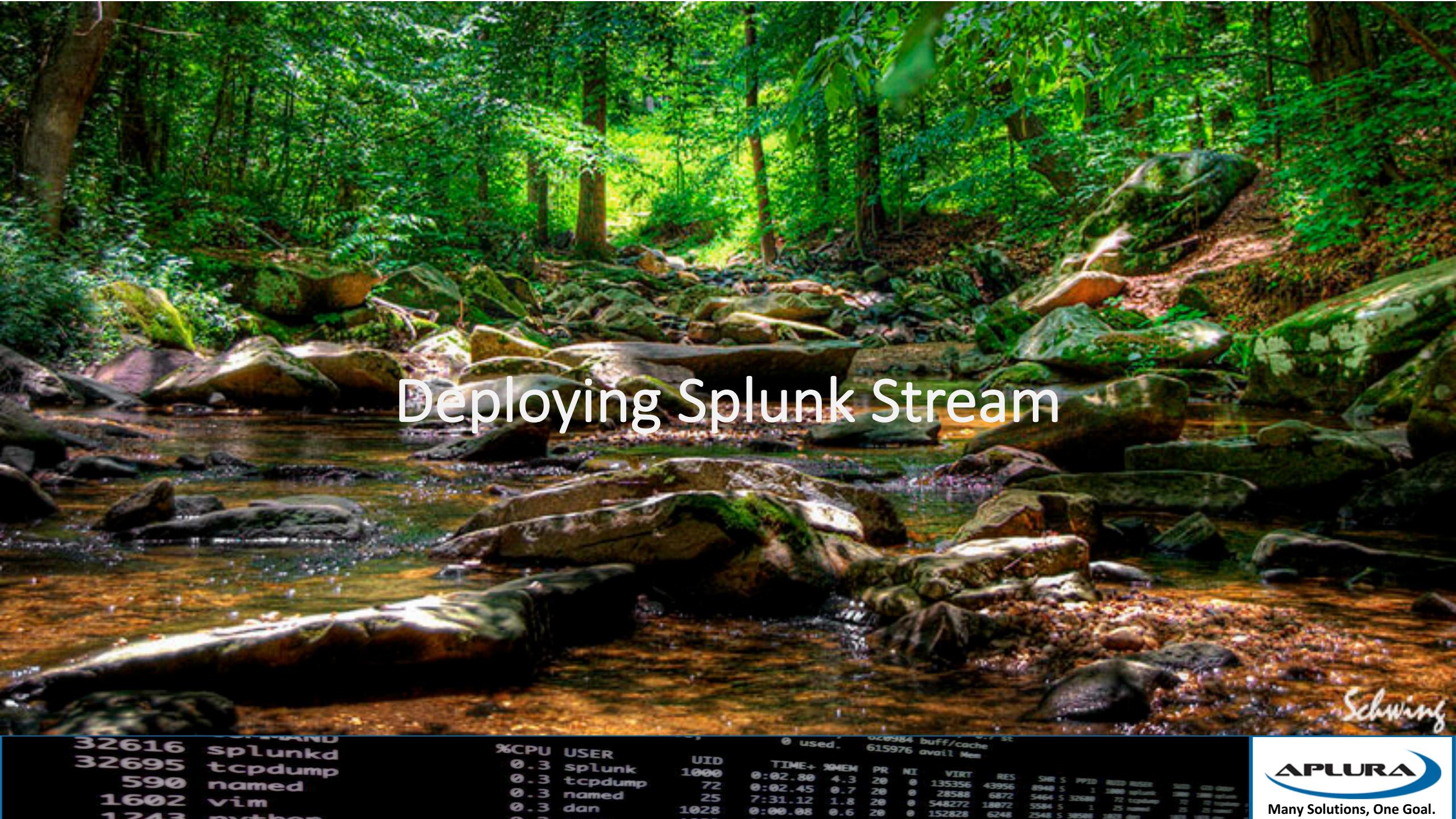
32616  splunkd
32695  tcpdump
590    named
1602   vim
1243   python

```

```

%CPU  USER      UID      TIME+  %MEM    PR  NI  VIRT      RES      SHR      S  PPID      NI  RES
0.3   splunk    1000     0:02.80  4.3    20  0  135356  43956  8948  S  2  1800  splunk
0.3   tcpdump   72       0:02.45  0.7    20  0  28588  6872   5464  S  32689  2  1800  tcpdump
0.3   named     25       7:31.12  1.8    20  0  548272  18872  5584  S  1  25  named
0.3   dan       1028     0:00.08  0.6    20  0  152828  6248   2548  S  38588  1829  dan

```



# Deploying Splunk Stream

Schwing

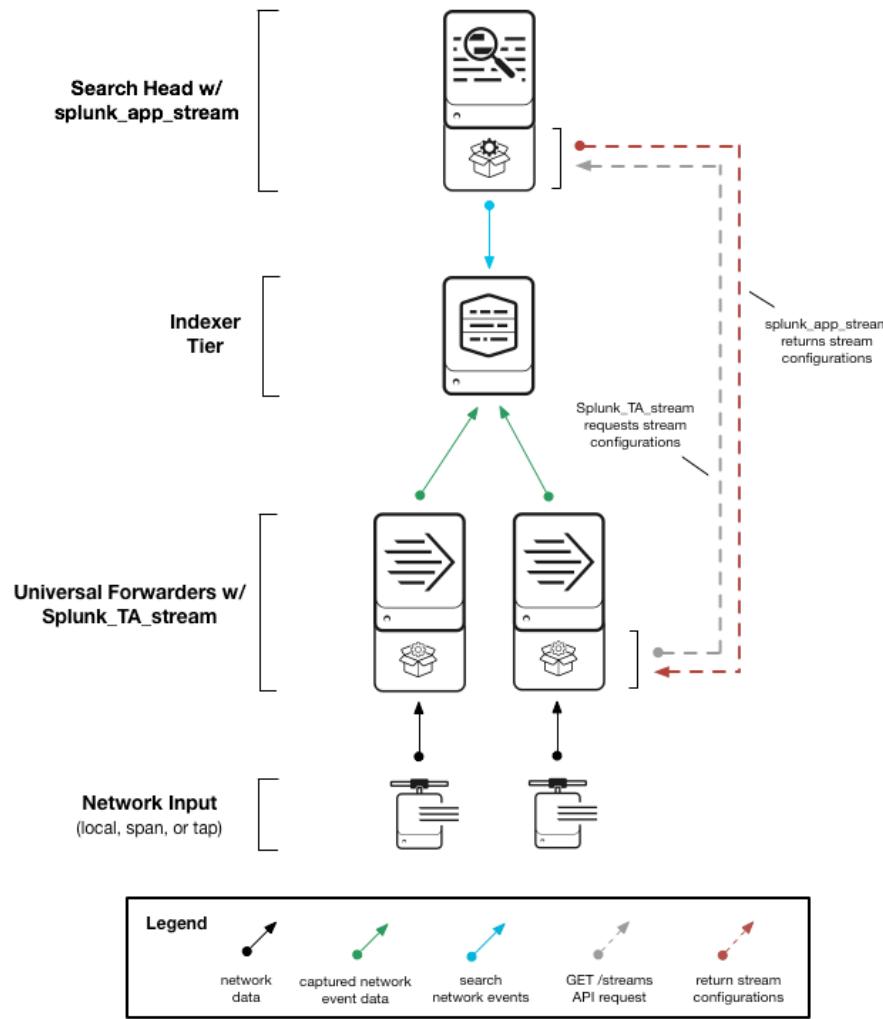
32616	splunkd		
32695	tcpdump		
590	named		
1602	vim		
1243	python		

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	PPID	RD	WR	SWAP	GRD	Other
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	5	2	1800	1800	1800	1800
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	5	32689	72	72	72	72
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	5	1	25	named	25	25
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	5	38588	1829	1829	1829	1829



Many Solutions, One Goal.

## Splunk App for Stream Distributed Deployment Architecture



```

32616  splunkd
32695  tcpdump
590   named
1602   vim
1243   python

```

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	PPID	NIOSR	ST
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	5	1800	splunk
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	5	32689	tcpdump
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	1	25	named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	5	38588	dan

# Two parts

- The Splunk App for Stream
  - Dashboards for analytics on protocols
  - Administrative panels for configuration
  - Stream Estimate (really cool, more later)
  - Goes on Search Head/Controller
- Splunk Stream Add-on
  - Binaries
  - Index-time operations (linebreaking, timestamping)
  - Goes on Indexers and Forwarders (UF or HF)

		%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RD	WUSER	SWAP	GD	DR
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	1800	splunk	3400	400	1000
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump	3400	200	1000
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	72	25	1000
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1829	dan	3400	1000	1000
1243	python	0.3																

# Install the Splunk App for Stream

- Can co-locate with ES
- Can co-locate with DMC
- In smaller (less than 100 forwarders) don't use with the DS
  - Possible exhausted connections (DS and Stream poll separately)
- Installs just like any other Splunk app

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RSS	RSER	SWAP	GD	CD
32616	splunkd															
32695	tcpdump															
590	named															
1602	vim															
1243	python															

# Harvest the Add On

- Installs to a few places
- \$SPLUNK\_HOME/etc/apps/Splunk\_TA\_stream
- \$SPLUNK\_HOME/etc/apps/splunk\_app\_stream/install/Splunk\_TA\_stream
- \$SPLUNK\_HOME/etc/deployment-apps/Splunk\_TA\_stream
  - Will create the local inputs.conf with the app server location

```
Davids-MacBook-Pro-2:local root# cat inputs.conf
[streamfwd://streamfwd]
splunk_stream_app_location = https://Davids-MacBook-Pro-2.local:8000/en-us/custom/splunk_app_stream/
stream_forwarder_id =
disabled = 0
```

Harvest the

\* Skip this is your SH is your DS

%CPU	USER	UID	% used	0 used.	buff/cache	ST
32616	splunkd			615976	avail Mem	
32695	tcpdump					
590	named					
1602	vim					
1243	python					

# Make sure your forwarders can talk back

- Your forwarders will need to be able to talk to the SH with `splunk_app_stream` installed
- The port is the same as the GUI for your SH

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SRR	S	PPID	RUID RUSER	SHRD	GRD GUSER
32616	splunkd													
32695	tcpdump													
590	named													
1602	vim													
1243	python													

# Configure your forwarders

- Don't have to be root on Linux
  - Use the included setuid.sh script
- Must be local admin or local system on Windows
- On UFs you should monitor your thruput limits

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RSS	RSER	SWAP	GID	GRPID
32616	splunkd															
32695	tcpdump															
590	named															
1602	vim															
1243	python															

# Inputs.conf

- Remember that the inputs.conf is layerable
- Just like other Splunk configs
- Doesn't have to be in the Splunk\_TA\_stream
- On the DS you can deploy two apps, one with the input to point back to the splunk\_app\_stream
- Then also deploy the Splunk\_TA\_stream

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUID RUSER	ST
32616	splunkd												
32695	tcpdump												
590	named												
1602	vim												
1243	python												

# Configure your streams

- The defaults may send more fields than you need
- Can tell forwarders which parts of the data you want
- You can have different configs for different groups!

Configure Stream - http  
HTTP Protocol Events

< Back to streams

Mode: Enabled

Splunk Index: default

Protocol: HTTP

Aggregation: No

Filters: 0 filters configured [View Filters](#)

Fields:

Enable	Name	Description	Type	Term	Action
<input checked="" type="checkbox"/>	bytes	The total number of bytes transferred	Original	flow.bytes	<a href="#">▼</a>
<input checked="" type="checkbox"/>	bytes_in	The number of bytes sent from client to server	Original	flow.cs-bytes	<a href="#">▼</a>
<input checked="" type="checkbox"/>	bytes_out	The number of bytes sent from server to client	Original	flow.sc-bytes	<a href="#">▼</a>
<input checked="" type="checkbox"/>	cookie	The Cookie HTTP request header	Original	http.cookie	<a href="#">▼</a>

```
32616  splunkd
32695  tcpdump
590    named
1602   vim
1243   python
```

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	PPID	RUSS	ST
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	5	1000	spunk
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	5	32689	tcpdump
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	1	25	named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	5	38588	dan

# Configure your forwarder groups

- Uses good ol' regex
- Lets you say ahead of time if Ephemeral Streams should be allowed

Create New Forwarder Group

Name: DBServer

Description: Database servers in the App group

Include Ephemeral Streams?  Yes  No

[Cancel](#) [Next >](#)

Matched Forwarders ( New Rule )

Regex Rule: appdbhost[\d]+

Preview of Matched Forwarders: No matches found.

Forwarder ID	Last Known Event

[Cancel](#) [< Back](#) [Next >](#)

PPID	PID	COMMAND	%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	NAME	ST
32616	splunkd		0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	18000 splunk	RUNN
32695	tcpdump		0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72 tcpdump	RUNN
590	named		0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25 named	RUNN
1602	vim		0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1629 dan	RUNN
1243	python		0.3													

# Gotcha with Groups

- Just regex on the Stream forwarder ID (not IP, hostname)
- This is configured in an XML file
- Messy
- The “defaultgroup” forwarder group for all unmatched hosts will gather ALL THE THINGS

Distributed Forwarder Management						
Create Stream Forwarder groups using pattern match.						
3 groups		<input type="text"/> <span style="color: red;">X</span>				
i	Name	Description	Rule	Include Ephemeral Streams?	Contains Streams	Actions
>	AppServers	These are my app servers that get fun stuff	^.*Pro.*\$	Yes	3	<span style="color: blue;">▼</span>
>	DBServer	Database servers in the App group	appdbhost[\d]+	No	0	<span style="color: blue;">▼</span>
>	defaultgroup	Used when there is no matching group found for a given stream forwarder ID		Yes	41	<span style="color: red;">▼</span>

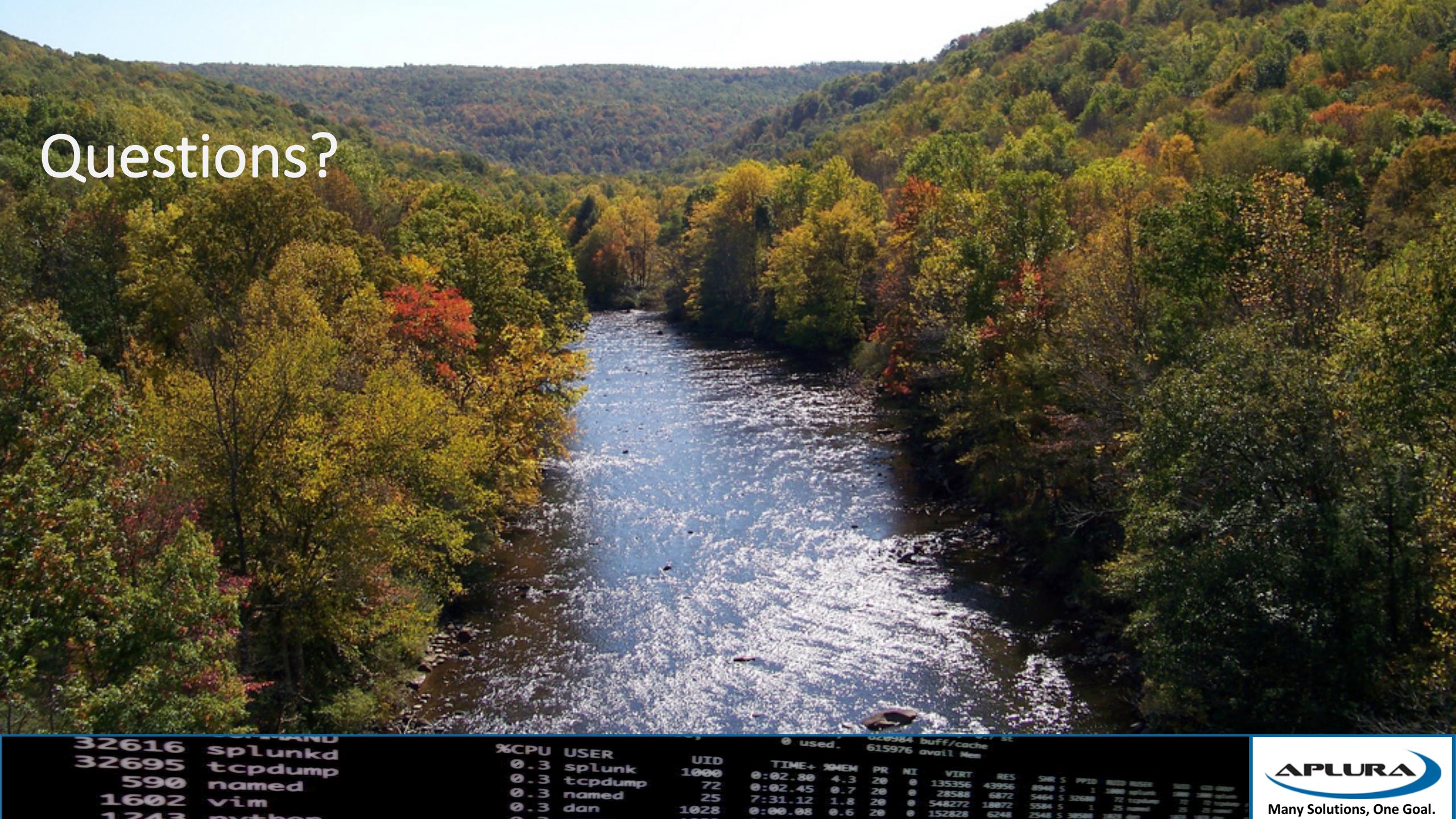
```
32616  splunkd
32695  tcpdump
590    named
1602   vim
1243   python
```

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	NAME	ST
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	18000 splunk	615976 avail Mem
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72 tcpdump	
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25 named	
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1828 dan	

# Wait for data to flow in

- That's pretty much it!
- Docs make it look a lot harder

		%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	RD	RSER	SLICE	GRD	Color
<b>32616</b>	<b>splunkd</b>	<b>0.3</b>	<b>splunk</b>	<b>1000</b>	<b>0:02.80</b>	<b>4.3</b>	<b>20</b>	<b>0</b>	<b>135356</b>	<b>43956</b>	<b>8948</b>	<b>S</b>	<b>1</b>	<b>1800</b>	<b>splunk</b>	<b>1000</b>	<b>0000</b>	
<b>32695</b>	<b>tcpdump</b>	<b>0.3</b>	<b>tcpdump</b>	<b>72</b>	<b>0:02.45</b>	<b>0.7</b>	<b>20</b>	<b>0</b>	<b>28588</b>	<b>6872</b>	<b>5464</b>	<b>S</b>	<b>32699</b>	<b>72</b>	<b>tcpdump</b>	<b>72</b>	<b>20</b>	<b>0000</b>
<b>590</b>	<b>named</b>	<b>0.3</b>	<b>named</b>	<b>25</b>	<b>7:31.12</b>	<b>1.8</b>	<b>20</b>	<b>0</b>	<b>548272</b>	<b>18872</b>	<b>5584</b>	<b>S</b>	<b>1</b>	<b>25</b>	<b>named</b>	<b>72</b>	<b>20</b>	<b>0000</b>
<b>1602</b>	<b>vim</b>	<b>0.3</b>	<b>dan</b>	<b>1028</b>	<b>0:00.08</b>	<b>0.6</b>	<b>20</b>	<b>0</b>	<b>152828</b>	<b>6248</b>	<b>2548</b>	<b>S</b>	<b>38598</b>	<b>1629</b>	<b>dan</b>	<b>1028</b>	<b>20</b>	<b>0000</b>
<b>1243</b>	<b>python</b>	<b>0.3</b>																



# Questions?

```
32616  splunkd
32695  tcpdump
590    named
1602   vim
1243   python
```

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SRR	S	PPID	NAME
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	2	1860 splunk
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72 tcpdump
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25 named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1821 dan

# Credits

- Thanks to the Baltimore Area Splunk User Group
- Cover Slide: Upper Swallow Falls in Oakland, MD, Chris Flees, <http://fineartamerica.com/profiles/chris-flees.html?tab=artwork&page=7>
- Slide 3: Potomac River in Maryland, Terry J. Adams, <http://www.fhwa.dot.gov/byways/byways/60807/photos>
- Slide 7: Timanus Mill on the Jones Falls in Baltimore, “Monument City”, <http://www.panoramio.com/photo/57148558>
- Slide 8: “Missing Homework Log” by “Red Beetle RB”. <https://www.teacherspayteachers.com/Product/Missing-Homework-Log-4112>
- Slide 9: Rotten log, National Wildlife Foundation, <https://www.nwf.org/kids/family-fun/outdoor-activities/investigate-a-rotten-log.aspx>
- Slide 10: The Simpsons, <http://i.imgur.com/91sn32Q.jpg?fb>
- Slide 11: Bro Network Security Monitor, <https://www.bro.org/>
- Slide 17: Ian Adams Photography, <http://ianadamsphotography.com/news/galleries/bridges/>
- Slides 19 and 21: Splunk Conf 2015, “Splunk App for Stream Deployments in the Real World: Enhance Operational Intelligence Across Application Delivery, IT Ops, Security and More”,  
[http://conf.splunk.com/session/2015/conf2015\\_SUdovicic\\_CChing\\_MDickey\\_SplunkEntWhatsNew\\_StreamDeploymentsInTheReal.pdf](http://conf.splunk.com/session/2015/conf2015_SUdovicic_CChing_MDickey_SplunkEntWhatsNew_StreamDeploymentsInTheReal.pdf)
- Slide 22: Gunpowder Falls in Baltimore County, MD, <http://hdrcreme.com/photos/1818-gunpowder-falls>
- Slide 23: Splunk Docs, <http://docs.splunk.com/Documentation/StreamApp/latest/DeployStreamApp/DeploymentArchitecture>
- Slide 34: Youghiogheny River at Friendsville, MD by Joe Dawson, <https://www.flickr.com/photos/jmd41280/5066756138>

Top 10 processes using CPU									
%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR
32616	splunkd								
32695	tcpdump								
590	named								
1602	vim								
1243	python								