Your Time Is Now
Do IT With DevOps

Chris Jackson, Distinguished Systems Engineer
BRKNMS-2446
Agenda

- Introduction
- DevOps Overview
- How Can DevOps Benefit Me?
- What Tools/Skills are Needed?
- Examples of Programmatic Interfaces
- Conclusion
Why are you here?
(In this Session)
Ideas
Compare
Sharpen
Skills
DevOps Overview
“[DevOps is] a set of cultural norms and technical practices that enable this fast flow of work from dev through test through operations while preserving world class reliability”

Gene Kim - author of “The Phoenix Project"
“If you are doing quarterly releases and your competitor is doing daily releases you will fall so far behind”

Adrian Cockcroft – former Netflix Architect
http://a16z.com/2016/09/01/microservices/
Developers World

• Care About
  • Writing Software
  • Working Code
  • APIs
  • Libraries
  • Sprints

• Success
  • Software works – Laptop and Test
  • Finished Sprint
Waterfall Development Process

Around since the 50s
Sequential Design Approach
Requirements and scope are fixed
Before Agile Methodologies

Time

Analysis  Design  Code  Test

Half way finished? 50% Complete 100% Unusable

Sacrificial Lamb
Lean and Agile

Lean – Management philosophy to eliminate waste across all aspects of business
• Derived from Toyota Production System

Agile – Implementation of Lean for software development
• Short sprints
• Continuous incremental value
Benefits of Agile Development

Time

Analysis
Design
Code
Test

40% Complete
100% Usable Code
SCRUM Agile Project Management

Product Backlog

Sprint Backlog

24 hrs Daily Scrum

2 weeks Sprint

Potentially Shippable Product Increment

Stories
- Create account
- Book reservation
- Print report
- Login
- Search

Create Permit

Approve Permit

Cancel Permit
Extreme Programing (XP)

- Focuses on Software Quality and Responding quickly to changing customer requirements
- Frequent releases and short dev cycles
- Improve productivity and interaction with customer

- Paired Programing
- Test Driven Development
- Continuous Integration
- Continuous Delivery
Kanban

• Japanese for “Signal Board”
• 5 Core Principles
  • Visualize the workflow
  • Limit Work In Process
  • Manage Flow
  • Make Process Policies Explicit
  • Improve Collaboratively
Pulling it all together

Lean

Agile

Scrum  Kanban

Extreme Programming

IT Level

Business Level

Software Project and Team Management

Software Engineering Practices
Operations World

- Care About
  - Everything is stable
  - Standards
  - Templates
  - Not getting bothered at 2:00 am

- Success
  - Software is stable
  - Backup and restore works
  - Systems are operating within defined thresholds
Traditional IT service delivery: Slow, manual, and error prone

- **Requirements**
  - Define strategy & required features

- **Architecture**
  - Ensure service updates are aligned with enterprise architecture

- **Development**
  - Write software / integrate underlying services

- **QA & BAT**
  - Test that software works

- **OPS**
  - Pushes updates to production and manage service

The more complex a project becomes, the longer the schedule, and the higher the probability of scope and schedule surprises.
What is a Microservice?

- A small and focused piece of software
- Stateless loosely coupled
- Language and technology independent
- Highly scalable and fault tolerant
Microservice = LOTS of east west traffic
Operating the Death Star

450 microservices

500+ microservices

500+ microservices

Source:
Netflix: [http://www.slideshare.net/BruceWong3/the-case-for-chaos](http://www.slideshare.net/BruceWong3/the-case-for-chaos)
Twitter: [https://twitter.com/adriancost/status/441883572618948608](https://twitter.com/adriancost/status/441883572618948608)
Hail-o: [https://sudo.hailoapp.com/services/2015/03/09/journey-into-a-microservice-world-part-3/](https://sudo.hailoapp.com/services/2015/03/09/journey-into-a-microservice-world-part-3/)
Operating Microservice Apps is Not Easy.
Infrastructure as Code
(Configuration Management)

• Writing high level code that automates the provisioning and deployment of infrastructure components

• Not just script writing or infrastructure automation

• Uses software development practices
  • Versioning control
  • Design Patterns
  • Testing

• Infrastructure is DEFINED by the code

• Vagrant, Puppet, Chef, Ansible, Docker, etc
Immutable Infrastructure

- Unchanging over time or unable to be changed

- Once you instantiate something, it never changes
- Instead you replace it with a newly built instance
- Only works in a true cloud environment with API control over all aspects of Configuration and Monitoring
Immutable Infrastructure Stack

Mutable Server

Build

Patch App

Patch OS

Immutable Server

App v1
OS v1

App v2
OS v1

App v2
OS v2

BRKNMS-2446
Traditional VM vs Immutable Delivery Model

- VM = 2-5 minutes
- Software Entropy
- Infrastructure as code upwards of 10 minutes to restore
- Container 500ms
- New instantiation each time with latest packages
- Immutable delivery model 2-3 secs
Containers Enable Microservices

• Micro-services Architecture
  • Small loosely coupled and purpose built services
  • Efficient Distribution/packaging vehicle

• Better aligning to the Dev and Ops of DevOps
  • Puppet, Chef, Ansible are popular with Ops but not Devs
  • Containers are equally popular with both
  • Containers create a natural segmentation of effort
    • Devs inside the container, Ops outside the container
The Impact of Containers on DevOps

- Guaranteed consistency for CI/CD
  - Simplified light weight image files
    - Only what the app needs and nothing else
  - What is built on a laptop will work the same in production

- Open community built best of breed containers
  - Public container repository - Docker Hub

- SPEED
  - Deploy in seconds instead of minutes
PETS vs. CATTLE vs. CHICKENS
Continuous Integration, Delivery, and Deployment

- **Continuous Integration**: Merging of development work with code base constantly so that automated testing can catch problems early.

- **Continuous Delivery**: Software package delivery mechanism for releasing code to staging for review and inspection.

- **Continuous Deployment**: Relies on CI and CD to automatically release code into production as soon as it is ready. Constant flow of new features into production.
Continuous Deployment
How Can It Benefit My Organization?
DevOps Hype

• DevOps has reached maximum hype
• No longer a counter culture revolution for hipsters
• Moving target - Is defined, but loose standards
• Casualties abound
• Many successes too!

Source: Gartner
DevOps Darlings

- 100 releases a day
- Fully automated build tools to test and make packages • Fully automated machine image bakery • Fully automated image deployment
- Developers deploy when they want • manage their own capacity and auto scaling • and fix anything that breaks

- 95% of eBay marketplace traffic is powered by OpenStack cloud
- App-provisioning time is now 30 minutes (used to be weeks)
- Moving to cloud and virtualization saved million of dollars in capex

- 45+ APIs facilitating in store experience, supply chain, and back office
- 80 deployments a week
- Monthly API volume over 1.5 billion hits
- Less than 10 incidents a month
Areas Where DevOps Benefits Network IT

- Provisioning – Config, SWIM
- Operational State Checks
- Assisted Troubleshooting
- Monitoring/Dashboards/Analytics
- Auditing/Compliance Checks
Lean, Agile, and DevOps Combined
What Tools/Skills/Technology are Needed?
So... Are All Network Engineers Becoming Programmers?

NO

NO

NO

NO

Yes

NO

Cisco Live!
But…Are Some Network Engineers Embracing Programming?
Languages
Remember This Inflection Point?

Telephony in 1998

- IP Telephony struggled until we got ‘hybrid engineers’ to translate between the Circuit Switch ‘Tip & Ring’ and Packet Switch ‘Bits & Bytes’ camps

- Likewise, now we need the next generation of ‘hybrid engineers’ to translate between traditional network domain engineers and software/application developers
• DevNet – https://developer.cisco.com
With Cisco’s DevNet You Can… Browse APIs
<table>
<thead>
<tr>
<th>Continuous Integration</th>
<th><img src="image1" alt="Icons" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Management</td>
<td><img src="image2" alt="Icons" /></td>
</tr>
<tr>
<td>Collaboration</td>
<td><img src="image3" alt="Icons" /></td>
</tr>
<tr>
<td>Working Environment</td>
<td><img src="image4" alt="Icons" /></td>
</tr>
<tr>
<td>Source/Image Control</td>
<td><img src="image5" alt="Icons" /></td>
</tr>
<tr>
<td><strong>PaaS</strong></td>
<td><img src="image6" alt="Icons" /></td>
</tr>
<tr>
<td><strong>IaaS</strong></td>
<td><img src="image7" alt="Icons" /></td>
</tr>
</tbody>
</table>
# Devops Periodic Table

## Periodic Table of DevOps Tools (V1)

<table>
<thead>
<tr>
<th>No.</th>
<th>Tool</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open Source</td>
<td>Database, SCM</td>
</tr>
<tr>
<td>2</td>
<td>Free</td>
<td>Build</td>
</tr>
<tr>
<td>3</td>
<td>Freemium</td>
<td>Testing</td>
</tr>
<tr>
<td>4</td>
<td>Paid</td>
<td>Containerization</td>
</tr>
<tr>
<td>5</td>
<td>Enterprise</td>
<td>Collaboration</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Security</td>
</tr>
</tbody>
</table>

### Tools:

- **Open Source**
  - Database
  - SCM
  - Build
  - Testing
  - Containerization
  - Collaboration
  - Security

- **AWS**
  - Amazon Web Services

- **DevOps Tools**
  - Git
  - Jira
  - Jenkins
  - Docker
  - Kubernetes
  - JUnit
  - TestNG
  - Selenium
  - JMeter
  - Artifacts

- **Build Tools**
  - Maven
  - Gradle
  - Ant
  - Bابل
  - Babel
  - Buildkite
  - TeamCity
  - Travis CI

- **Testing Tools**
  - JUnit
  - TestNG
  - Selenium
  - JMeter
  - Artifacts

- **Deployment Tools**
  - Git
  - Jira
  - Jenkins
  - Docker
  - Kubernetes
  - JUnit
  - TestNG
  - Selenium
  - JMeter

- **Configuration Management**
  - Ansible
  - Puppet

- **Cloud Platforms**
  - AWS
  - Azure
  - Google Cloud
  - IBM Cloud

- **Virtualization**
  - VMware
  - Hyper-V

- **Database Tools**
  - MySQL
  - PostgreSQL

- **Version Control**
  - Git
  - SVN

- **CI/CD**
  - Jenkins
  - CircleCI

- **Reporting**
  - Artifacts
  - JUnit
  - TestNG

- **Monitoring**
  - Nagios
  - Zabbix

- **Security**
  - OWASP
  - NIST

- **Automation**
  - Jenkins
  - Travis CI

- **Virtualization**
  - VMware
  - Hyper-V

- **Testing**
  - JUnit
  - TestNG
  - Selenium

- **Telecommunication**
  - New Relic

- **Visualization**
  - Grafana

- **Networking**
  - Docker
  - Kubernetes

- **DevOps Tools**
  - Docker
  - Kubernetes

- **Data Management**
  - MySQL
  - PostgreSQL

- **Configuration Management**
  - Ansible
  - Puppet
Need a Place to Store Code

- A place to store current and past versions of code
  - Ability to merge, branch, fork
  - Ability to see who has changed which file
  - Highly available service
  - Ability to revert to prior versions
  - Public or Private / OnPrem / OffPrem
  - Social
Git

$ git add ciscolive.py

$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
# (use "git rm --cached <file>..." to unstage)
#
# new file: ciscolive.py
Network Engineers using git

- git - a version control system
- Archive configs in a NMS like Prime Infrastructure
- Consider supplementing this with archiving device configs onto a TFTP/FTP/SCP server, then use git to create a tracked repository

```
[cloudsevtuser1@git configs]$ ls -l
total 64
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4507 Dec 5 20:05 London.virl.info.RUNNINGCONFIG.cfg
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4508 Dec 5 09:41 London.virl.info.STARTUPCONFIG.cfg
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4724 Dec 5 09:41 Richardson.virl.info.RUNNINGCONFIG.cfg
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4724 Dec 5 09:41 Richardson.virl.info.STARTUPCONFIG.cfg
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4949 Dec 5 20:05 RTP.virl.info.RUNNINGCONFIG.cfg
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4926 Dec 5 09:41 RTP.virl.info.STARTUPCONFIG.cfg
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4728 Dec 5 09:42 San_Jose.virl.info.RUNNINGCONFIG.cfg
-rw-r--r--. 1 cloudsevtuser1 cloudsevtuser1 4704 Dec 5 09:42 San_Jose.virl.info.STARTUPCONFIG.cfg
[cloudsevtuser1@git configs]$ git add *.cfg
[cloudsevtuser1@git configs]$ git status
On branch master
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
modified:   London.virl.info.RUNNINGCONFIG.cfg
modified:   RTP.virl.info.RUNNINGCONFIG.cfg
```

Ciscolive!
Network Engineers using git

```
[Cloudsevtuser1@git configs]$ git commit -m "Config Archive Dec 5 11:08:15 PM"
[master 618786d] Config Archive Dec 5 11:08:15 PM
  2 files changed, 1 insertion(+), 1 deletion(-)
[Cloudsevtuser1@git configs]$ 
```
Network Engineers using git

```
[cloudsevtuser1@git configs]$ git commit -m "Config Archive Dec 5 11:08:15 PM"
[master 618786d] Config Archive Dec 5 11:08:15 PM
2 files changed, 1 insertion(+), 1 deletion(-)
[cloudsevtuser1@git configs]$ git show 618786d
commit 618786d47788c5d9a714db10f0e04fbc0c15becc
Author: CloudSEVTUser1 <CloudSEVTUser1@SEVT.cisco.com>
Date: Mon Dec 5 20:08:33 2016 -0800

Config Archive Dec 5 11:08:15 PM

diff --git a/London.virl.info.RUNNINGCONFIG.cfg b/London.virl.info.RUNNINGCONFIG.cfg
index 2f8deec..b014c98 100644
--- a/London.virl.info.RUNNINGCONFIG.cfg
+++ b/London.virl.info.RUNNINGCONFIG.cfg
@@ -175,7 +176,6 @@ line vty 0 4
    transport input telnet ssh
    
  no scheduler allocate
  -ntp server 10.122.6.110
  +
  
end

diff --git a/RTP.virl.info.RUNNINGCONFIG.cfg b/RTP.virl.info.RUNNINGCONFIG.cfg
index d1904bf..75b2454 100644
--- a/RTP.virl.info.RUNNINGCONFIG.cfg
+++ b/RTP.virl.info.RUNNINGCONFIG.cfg
@@ -150,6 +150,7 @@ ip route 10.122.6.0 255.255.254.0 19.0.0.2
  ip ssh server algorithm authentication password
  
  logging host 10.122.7.139
+logging host 10.1.2.3
!snmp-server community CloudSEVT RO
snmp-server chassis-id 23
```
Jenkins

- A CI app that monitors execution of automated jobs

- **Build/test software projects continuously**
  Jenkins provides an easy-to-use system to help developers integrate changes to a project, and makes getting a fresh build easier for users

- **Monitors executions of externally-run jobs**
  cron/at jobs, procmail jobs, and those run on a remote machine
  Jenkins can automate the collection/review of the periodic output from compilers or build servers and make it easier to notice when something failed
Image Repository Management

• Provide highly configurable proxies between your organization and public repositories - reduces network traffic by caching commonly used libraries and other software components

• Provide an organization with a deployment destination for your own generated output

• Common Tools: Sonatype Nexus, Apache Archiva, Artifactory, Docker Hub
http://www.sonatype.com/nexus
Communication

- Integration with Development Environment
  - Code commits should appear
- Chat with History
- Team Rooms
- Kanban Board
- Multi-Platform Clients
- “Chat-Ops”
1. Developer Pulls From Trunk

2. Make incremental changes on local environment and unit tests

3. Developer pushes commits

4. Continuous Integration Server

5. Changes Kick off Test Build Environment

6. Jenkins runs Multiple Tests Integration, Smoke, etc.

7. Reports back test results

8. If tests pass Deploy code to artifact repository

9. Continuous Delivery

Artifact Repository

Production
Examples of Programmatic Interfaces
Network Device Instrumentation / Programmatic Interfaces

- VTY - telnet/SSH
- SNMP
- EEM
- NETCONF
- Python
- OpFlex
- Via Controller (REST)
Why Change?

• Familiar Manual, CLI-driven, device-by-device approach is inefficient

• Increased need for programmatic interfaces which allow faster and automated execution of processes and workflows with reduced errors

• In SDN - Need for a ‘central source of truth’ and touch-point
Embedded Event Manager (EEM)

- Extremely flexible and powerful subsystem within Cisco IOS Software
- Adapt device behavior and insert custom logic without IOS upgrade
- 24 Event Detectors (ED) integrated with IOS modules for wide range of system event detection
- CLI and Tcl based policy provides consistent programmability interface
- Powerful event engine supporting multi-event correlation, advance scheduling and more
NETCONF (XML)

- IETF working group http://www.ietf.org/html.charters/netconf-charter.html
- Chartered to produce a network configuration protocol which can:
  - Differentiate between configuration and non-configuration data
  - Provide extensibility so vendors will provide all configuration data through a single protocol
  - Provide a programmatic interface that uses a textual data representation for ease of manipulation
  - Supports integration with configuration database systems
  - Supports feature-rich configuration transactions such as locking and rollback
  - Is as transport-independent as possible
  - Provides support for asynchronous notifications
NETCONF (Cont.)

- NETCONF configuration protocol

- SSH—Secure Shell, well known and supported, considered mandatory in NETCONF

- BEEP—Blocks Extensible Exchange Protocol, not well known, considered optional - not updated with latest RFCs
NETCONF IETF Working Group

• Settled on Extensible Markup Language (XML) based data encoding for configuration data and protocol messages

• Defined that NETCONF protocol operations use simple Remote Procedure Calls (RPC)

• The NETCONF protocol defines a simple mechanism to manage configs and retrieve operation state data

• The protocol requires that the mapping over SSH is mandatory to implement (while the mapping over BEEP and SOAP are optional)
NETCONF

• NETCONF access for configuration over SSH first released IOS 12.4(9)T

  • IOS

  • IOS-XE

  • NX-OS
## NETCONF Conceptual Layers

<table>
<thead>
<tr>
<th>Layer</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Configuration data (interface GigabitEthernet0/0)</td>
</tr>
<tr>
<td>Operations</td>
<td><code>&lt;get-config&gt;</code>, <code>&lt;edit-config&gt;</code></td>
</tr>
<tr>
<td>RPC</td>
<td><code>&lt;rpc&gt;</code>, <code>&lt;rpc-reply&gt;</code></td>
</tr>
<tr>
<td>Transport Protocol</td>
<td>SSH, BEEP, SSL, console</td>
</tr>
</tbody>
</table>

Source: RFC 4741, 1.1 Protocol Overview
Supported NETCONF Operation-Types

- `<close-session>`
- `<commit>`
- `<copy-config>`
- `<delete-config>`
- `<discard-changes>`
- `<edit-config>`
- `<get>`
- `<get-config>`

- `<kill-session>`
- `<lock>`
- `<unlock>`
- `<validate>`
- `<notification-on>`
- `<notification-off>`

The XML Schema (XSD) Provides Insight to Valid Formatting
get-conf Operation

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="4" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config><source><running/></source>
   <filter type="cli"><config-format-xml options="all"></config-format-xml></filter>
  </get-config>
</rpc>

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="4" xmlns="urn:ietf:params:netconf:base:1.0"><data><xml-config-data>
Building configuration...
<Device-Configuration>
  <version><Param>12.4</Param></version>
  <parser><cache/></parser><service><slave-log/></service>
  <service><pad/></service>
  <service><timestamps><debug><datetime><msec/></datetime><debug></timestamps></service>
  <service><timestamps><log><datetime><msec/></datetime><log></timestamps></service>
</xml-config-data></data></rpc-reply>
```
REST

- Representational State Transfer
- Stateless, Lightweight alternative to Web Services and RPC.

A REST service is:
- Platform & Language independent
- Standards-based (runs on top of HTTP)
- Can easily be used in the presence of firewalls

- No built-in security features, encryption, session management, QoS guarantees, etc. but can be added by building on top of HTTP
- For security, username/password tokens are often used
- For encryption, REST can be used on top of HTTPS (secure sockets)....
REST Data Elements

- Resources and Resource Identifiers
- Uniform Interface (GET, PUT, POST, DELETE)
- Resource Oriented
- Simple

<table>
<thead>
<tr>
<th>HTTP</th>
<th>Method</th>
<th>CRUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>CREATE</td>
<td>Create</td>
</tr>
<tr>
<td>GET</td>
<td>RETRIEVE</td>
<td>Retrieve</td>
</tr>
<tr>
<td>PUT</td>
<td>UPDATE</td>
<td>Update</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE</td>
<td>Delete</td>
</tr>
</tbody>
</table>
Tools

Any REST Client can be used

Browser based Plugins
Chrome ‘REST Console’
Firefox ‘REST Client’, ‘Poster’, ‘HttpRequester’

Command line tools
CURL
wget
NXOS Python Interpreter

AE-N5596-4# python
Python 2.7.2 (default, Jun 14 2012, 16:37:41)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Loaded cisco NxOS lib!

>>> 2 + 2
4

>>> quit()

AE-N5596-4# show version
Cisco Nexus Operating System (NX-OS) Software

Software
    BIOS:      version 3.5.0
    loader:    version N/A
    kickstart: version 5.2(1)N1(1a)
    system:    version 5.2(1)N1(1a)

Hardware
    cisco Nexus5596 Chassis ("O2 48X10GE/Modular Supervisor")
    Intel(R) Xeon(R) CPU         with 8263848 kB of memory.
    Processor Board ID FOC154330NL
And now, a practical example
Building an Availability Dashboard like the CiscoLive NOC

Step 1 – Get device reachability status from Prime Infrastructure’s REST API

- Prime Infrastructure has its API documentation available within the app
Building an Availability Dashboard like the CiscoLive NOC

Step 1 – Get device reachability status from Prime Infrastructure’s REST API

- Look for the ’GET Device Reachability Status’ method

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET Device Availability Status</td>
<td>Retrieves the device reachability status.</td>
</tr>
<tr>
<td>GET Device Availability Message</td>
<td>Retrieves the device availability message.</td>
</tr>
<tr>
<td>GET Device Availability Summary</td>
<td>Retrieves the device availability summary.</td>
</tr>
<tr>
<td>GET Device Details</td>
<td>Fetches the details of devices.</td>
</tr>
<tr>
<td>GET Device Down Message</td>
<td>Retrieves the device down message.</td>
</tr>
<tr>
<td>GET Device Health Info</td>
<td>Retrieves the device health information.</td>
</tr>
<tr>
<td>GET Device Metric Data</td>
<td>Fetches the metric data for the device.</td>
</tr>
<tr>
<td>GET Device Metrics Supported</td>
<td>Fetches the URLs for various metrics of the device.</td>
</tr>
<tr>
<td>GET Device Port Summary</td>
<td>Retrieves the device port summary.</td>
</tr>
<tr>
<td>GET Device Reachability Status</td>
<td>Retrieves the device reachability status.</td>
</tr>
<tr>
<td>GET Host Details</td>
<td>Fetches the details of Hosts</td>
</tr>
</tbody>
</table>
Building an Availability Dashboard like the CiscoLive NOC

Step 1 – Get device reachability status from Prime Infrastructure’s REST API

GET statisticsService/device/reachabilityStatus

Retrieves the device reachability status.
Since Product Version: 1.2

Resource URL
/websys/api/v1/statisticsService/device/reachabilityStatus

Request Parameters

<table>
<thead>
<tr>
<th>Type</th>
<th>Attribute Name</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>topN</td>
<td>query</td>
<td>Indicates the number of devices to get. By default, when not specified returns all available results.</td>
</tr>
<tr>
<td>String</td>
<td>source</td>
<td>query</td>
<td>An optional filter parameter that can be used to provide the IP Address or a group ID to get a filtered response.</td>
</tr>
<tr>
<td>String</td>
<td>type</td>
<td>query</td>
<td>A filter criteria that can take values such as ALL, REACHABLE or UNREACHABLE.</td>
</tr>
</tbody>
</table>
Building an Availability Dashboard like the CiscoLive NOC

Step 1 – Get device reachability status from Prime Infrastructure’s REST API

• We recognize that the request URL needs to be:
  
  https://(servername_or_IP)/webacs/api/v1/op/statisticsService/device/reachabilityStatus

• Now we can use something like POSTMAN, HttpRequester, or Cisco Process Orchestrator to execute the REST call

• The output appears as the next slide
Building an Availability Dashboard like the CiscoLive NOC

Step 1 – Get device reachability status from Prime Infrastructure’s REST API
Building an Availability Dashboard like the CiscoLive NOC

Step 2 – Transform the data

• We noted that the information was returned in ‘records’ defined by
  /magmtResponse/statisticsDTO/childStatistics/childStatistic

```xml
<?xml version="1.0" ?>
<mgmtResponse responseType="operation" requestUrl="https://10.122.:
  <statisticsDTO>
    <childStatistics>
      <childStatistics>
        </childStatistics>
      <statisticEntries>
        <statisticEntry>
          <attributeName>deviceIp</attributeName>
          <displayName>Device IP Address</displayName>
          <entryValue>19.0.0.21</entryValue>
        </statisticEntry>
        <statisticEntry>
          <attributeName>deviceName</attributeName>
          <displayName>Device Name</displayName>
          <entryValue>RTP.virl.info</entryValue>
        </statisticEntry>
        <statisticEntry>
          <attributeName>reachability</attributeName>
          <displayName>Reachability</displayName>
          <entryValue>DOWN</entryValue>
        </statisticEntry>
        <statisticEntries>
      </childStatistics>
    </statisticsDTO>
  </mgmtResponse>
```
Building an Availability Dashboard like the CiscoLive NOC

Step 2 – Transform the data

- We can iterate over the output line-by-line and processing it with loops and conditional logic checks…

- ...Or use XSL Transforms, which lend well to converting XML to HTML output

- This is “A GOOD Skill To Have ™”
Building an Availability Dashboard like the CiscoLive NOC

Step 2 – Transform the data; Example XSL Transform

```xml
<?xml version="1.0"?><!DOCTYPE xsl:stylesheet PUBLIC "Unofficial XSLT 1.0 DTD" "http://www.w3.org/1999/11/xslt10.dtd">
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html" indent="yes"/>
<xsl:template match="/">
  <html>
    <body>
      <table border="1" cellpadding="2" cellspacing="2" width="80%">
        <tbody>
          <xsl:for-each select="//childStatistic">
            <tr><td>
              <xsl:value-of select="statisticEntries/statisticEntry[attributeName='deviceName']/entryValue"/>
            </td><td>
              <xsl:value-of select="statisticEntries/statisticEntry[attributeName='deviceIp']/entryValue"/>
            </td><td>
              <xsl:value-of select="statisticEntries/statisticEntry[attributeName='reachability']/entryValue"/>
            </td></tr>
          </xsl:for-each>
        </tbody>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```
Building an Availability Dashboard like the CiscoLive NOC

Step 3 – Write the results to a file

```html
<html>
  <body>
    <table border="1" cellpadding="2" cellspacing="2" width="80%">
      <tbody>
        <tr>
          <td>RTP.virl.info</td>
          <td>19.0.0.21</td>
          <td>DOWN</td>
        </tr>
        <tr>
          <td>Richardson.virl.info</td>
          <td>19.0.0.23</td>
          <td>DOWN</td>
        </tr>
        <tr>
          <td>London.virl.info</td>
          <td>19.0.0.22</td>
          <td>UP</td>
        </tr>
        <tr>
          <td>San_Jose.virl.info</td>
          <td>19.0.0.20</td>
          <td>UP</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```
Building an Availability Dashboard like the CiscoLive NOC

Rendering in HTML as:

<table>
<thead>
<tr>
<th>Domain</th>
<th>Status</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP.virl.info</td>
<td>DOWN</td>
<td>19.0.0.21</td>
</tr>
<tr>
<td>Richardson.virl.info</td>
<td>DOWN</td>
<td>19.0.0.23</td>
</tr>
<tr>
<td>London.virl.info</td>
<td>UP</td>
<td>19.0.0.22</td>
</tr>
<tr>
<td>San_Jose.virl.info</td>
<td>UP</td>
<td>19.0.0.20</td>
</tr>
</tbody>
</table>
# Network Programmability Cisco Education Offerings

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
<th>Cisco Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Integrating Business Applications with Network Programmability (NIPBA); Integrating Business Applications with Network Programmability for Cisco ACI (NPIBAACI)</strong></td>
<td>Learn networking concepts, and how to deploy and troubleshoot programmable network architectures with these self-paced courses.</td>
<td>Cisco Business Application Engineer Specialist Certification</td>
</tr>
<tr>
<td><strong>Developing with Cisco Network Programmability (NPDEV); Developing with Cisco Network Programmability for Cisco ACI (NPDEVACI)</strong></td>
<td>Learn how to build applications for network environments and effectively bridge the gap between IT professionals and software developers.</td>
<td>Cisco Network Programmability Developer Specialist Certification</td>
</tr>
<tr>
<td><strong>Designing with Cisco Network Programmability (NPDES); Designing with Cisco Network Programmability for Cisco ACI (NPDESACI)</strong></td>
<td>Learn how to expand your skill set from traditional IT infrastructure to application integration through programmability.</td>
<td>Cisco Network Programmability Design Specialist Certification</td>
</tr>
<tr>
<td><strong>Implementing Cisco Network Programmability (NPENG); Implementing Cisco Network Programmability for Cisco ACI (NPENPGACI)</strong></td>
<td>Learn how to implement and troubleshoot open IT infrastructure technologies.</td>
<td>Cisco Network Programmability Engineer Specialist Certification</td>
</tr>
</tbody>
</table>

For more details, please visit: [http://learningnetwork.cisco.com](http://learningnetwork.cisco.com)

Questions? Visit the Learning@Cisco Booth or contact [ask-edu-pm-dcv@cisco.com](mailto:ask-edu-pm-dcv@cisco.com)
In Conclusion

- Remember - DevOps is a professional industry driven movement that is influencing a cultural shift in programming and operational support

- You CAN do IT
  …but go in with your eyes open and recognize the Changes required and the Benefits expected

- Use new programmatic interfaces to stream-line your operations – Move away from ‘Finger Defined Networks (FDN)’!
Complete Your Online Session Evaluation

• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Continue Your Education

• Demos in the Cisco campus
• Walk-in Self-Paced Labs
• Lunch & Learn
• Meet the Engineer 1:1 meetings
• Related sessions
Thank You
Your Time Is Now
Backup Slides
## Business Transformation Cisco Education Offerings

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
<th>Cisco Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>For IT and Network Professionals:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Building Business Specialist Skills</td>
<td>• Builds non-technical skills key to ensure business impact and influence. Topics include: business analysis, finance, technology adoption and effective communications. • Bridges IT and business impacts of mature and emerging solutions including cloud plus Internet of Everything</td>
<td>Cisco Enterprise IT Business Specialist</td>
</tr>
<tr>
<td><strong>For Technology Sellers:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applying Cisco Specialized Business Value Analysis Skills</td>
<td>Builds skills to discover and address technology needs using a business-focused, consultative sales approach</td>
<td>Cisco Business Value Specialist</td>
</tr>
<tr>
<td>Executing Advanced Cisco Business Value Analysis and Design Techniques</td>
<td>Enables customer transformation through business architecture and solution selling expertise</td>
<td>Cisco Certified Business Value Practitioner</td>
</tr>
<tr>
<td>Performing Cisco Business-Focused Transformative Architecture Engagements</td>
<td>Provides skills and an approach to build a strategic roadmap of IT initiatives, aligned to business priorities</td>
<td>Cisco Transformative Architecture Specialist</td>
</tr>
</tbody>
</table>

For more details, please visit: [http://learningnetwork.cisco.com](http://learningnetwork.cisco.com)

Questions? Visit the Learning@Cisco Booth or contact ask-edu-pm-dcv@cisco.com
# Cloud Cisco Education Offerings

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
<th>Cisco Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding Cloud Fundamentals (CLDFND)</td>
<td>Learn how to perform foundational tasks related to Cloud computing, and the essentials of Cloud infrastructure</td>
<td>CCNA Cloud</td>
</tr>
<tr>
<td>Introducing Cloud Administration (CLDADM)</td>
<td>Learn the essentials of Cloud administration and operations, including how to provision, manage, monitor, report and remediate.</td>
<td></td>
</tr>
<tr>
<td>Implementing and Troubleshooting the Cisco Cloud Infrastructure (CLDINF)</td>
<td>Learn how to implement and troubleshoot Cisco Cloud infrastructure: compute, network, storage.</td>
<td></td>
</tr>
<tr>
<td>Designing the Cisco Cloud (CLDDES)*</td>
<td>Learn how to design private and hybrid Clouds including infrastructure, automation, security and virtual network services</td>
<td>CCNP Cloud</td>
</tr>
<tr>
<td>Automating the Cisco Enterprise Cloud (CLDAUT)*</td>
<td>Learn how to automate Cloud deployments – provisioning IaaS (private, private with network automation and hybrid) and applications, life cycle management</td>
<td></td>
</tr>
<tr>
<td>Building the Cisco Cloud with Application Centric Infrastructure (CLDACI)*</td>
<td>Learn how to build Cloud infrastructures based on Cisco Application Centric Infrastructure, including design, implementation and automation</td>
<td></td>
</tr>
<tr>
<td>UCS Director Foundation (UCSDF)</td>
<td>Learn how to manage physical and virtual infrastructure using orchestration and automation functions of UCS Director.</td>
<td></td>
</tr>
</tbody>
</table>

* Available Q2CY2016

For more details, please visit: [http://learningnetwork.cisco.com](http://learningnetwork.cisco.com)

Questions? Visit the Learning@Cisco Booth or contact ask-edu-pm-dcv@cisco.com
Catalyst – The ‘Agile’ Movement & Scrum

• Some environments approach Agile methodology through ‘Scrum’

• Emphasizes:
  - Empirical feedback
  - Team self-management
  - Building properly tested product increments within short iterations

• Structure and feedback model of Scrum sometimes conflicts with status quo at established organizations not using Agile.
Lean Startup Concepts

- Minimum viable product
- Continuous deployment
- Split testing
- Actionable metrics
- Pivot
- Innovation accounting
- Build-Measure-Learn
Catalyst – The ‘Agile’ Movement

• Scrum Roles: Product Owner, Scrum Master, and the Team

• The responsibilities of the traditional project manager role are split up among these three Scrum roles.

• Scrum has five meetings:
  - Backlog Grooming (aka Backlog Refinement),
  - Sprint Planning,
  - Daily Scrum (AKA 15-minute standup)
  - Sprint Review Meeting
  - Sprint Retrospective Meeting.
### Waterfall

<table>
<thead>
<tr>
<th>Process</th>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>Shifting at each step</td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>Detect and remediate at last phase of project</td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td>Lessons learned from one release flow to next release</td>
<td></td>
</tr>
<tr>
<td>Risk</td>
<td>Occurs at end - ‘firefights’ common</td>
<td></td>
</tr>
<tr>
<td>Review</td>
<td>After release</td>
<td></td>
</tr>
<tr>
<td>Feedback</td>
<td>At end of project</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Planning</th>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals</td>
<td>Entry/Exit criteria defined for each step - entry/exit criteria</td>
<td></td>
</tr>
<tr>
<td>Definition of Ready</td>
<td>Analysis/Design for all use-cases before programming begins</td>
<td></td>
</tr>
<tr>
<td>Scoping</td>
<td>Product Owner decides</td>
<td></td>
</tr>
<tr>
<td>Effort</td>
<td>Project Manager estimates - PO approves</td>
<td></td>
</tr>
<tr>
<td>Scheduling</td>
<td>Determined by effort at each phase</td>
<td></td>
</tr>
</tbody>
</table>
# Waterfall

<table>
<thead>
<tr>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaming</td>
<td>Analysis and design complete before proceeding to development</td>
</tr>
<tr>
<td>Resourcing</td>
<td>Specific to each phase</td>
</tr>
<tr>
<td>Responsibility</td>
<td>Changes from phase to phase with PO overall responsible</td>
</tr>
<tr>
<td>Assignment</td>
<td>PM assigns work for whole project</td>
</tr>
<tr>
<td>Status</td>
<td>Periodically scheduled - generally focused on % completion</td>
</tr>
<tr>
<td>Effort tracking</td>
<td>Follows project plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td>Developed and performed by testing-specific team members</td>
</tr>
<tr>
<td>Testing Scope</td>
<td>Defined after Implementation phase</td>
</tr>
<tr>
<td>Regression</td>
<td>Identified after system test</td>
</tr>
<tr>
<td>Gap Handling</td>
<td>Addressed in next release</td>
</tr>
</tbody>
</table>
## Waterfall

<table>
<thead>
<tr>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Completion measured by deliverables (HLD, LLD, Test Plan, etc) at each phase</td>
</tr>
<tr>
<td>Closure</td>
<td>Delivery/Publishing of deliverables at each phase</td>
</tr>
<tr>
<td>Delivery</td>
<td>Publishing of deliverables at each phase and production of software/product at end</td>
</tr>
</tbody>
</table>
Waterfall

Pros

• Simplest to Implement
• Most well understood
• Minimal resources required to implement
• Longest history

Cons

• Difficult to revert - if requirements phase is wrong, then design and implementation will suffer
• Customer not engaged until late in the process - no feedback until the end
• Higher amount of risk and uncertainty
• Considered slow, dated
• ‘Firefight’ at end
Iterative

- Another common and ‘comfortable’ method
- Cyclical in flow - advance when current step *for a feature/function* is considered complete
- Each phase has specific deliverables
- Several iterations may be required
- Quality focus shift from Requirements/Design to Implementation/Testing
- Project Manager consults with Product Owner for iteration scope
- “Waterfall in loops”
### Iterative

#### Process

<table>
<thead>
<tr>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>Shifts from Analysis/Design to Implementation/Testing</td>
</tr>
<tr>
<td>Control</td>
<td>Detect and remediate at each iteration for new features - regression test</td>
</tr>
<tr>
<td>Improvement</td>
<td>Lessons learned from one iteration flow to next iteration</td>
</tr>
<tr>
<td>Risk</td>
<td>Occurs in Implementation/Test</td>
</tr>
<tr>
<td>Review</td>
<td>After each iteration/milestone</td>
</tr>
<tr>
<td>Feedback</td>
<td>At end of iteration</td>
</tr>
</tbody>
</table>

#### Planning

<table>
<thead>
<tr>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals</td>
<td>Completion of Analysis &amp; Design by one team - turned over to Implement/Test by another</td>
</tr>
<tr>
<td>Definition of Ready</td>
<td>Analysis/Design for some use-cases before programming begins</td>
</tr>
<tr>
<td>Scoping</td>
<td>PM decides scope for iteration - consults with PO</td>
</tr>
<tr>
<td>Effort</td>
<td>PM estimates</td>
</tr>
<tr>
<td>Scheduling</td>
<td>Based on iteration-specific delivery commitments</td>
</tr>
</tbody>
</table>
### Iterative

#### Execution

<table>
<thead>
<tr>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaming</td>
<td>Analysts and architects pass requirements to Dev/Test</td>
</tr>
<tr>
<td>Resourcing</td>
<td>Roles stay within specialty - extensions on as-needed basis</td>
</tr>
<tr>
<td>Responsibility</td>
<td>PM responsible for iteration</td>
</tr>
<tr>
<td>Assignment</td>
<td>PM sets feature/task focus per iteration</td>
</tr>
<tr>
<td>Status</td>
<td>Periodically scheduled - generally focused on % completion</td>
</tr>
<tr>
<td>Effort tracking</td>
<td>Follows iteration plan</td>
</tr>
</tbody>
</table>

#### Testing

- Developed and performed by testing-specific team members

#### Testing Scope

- Prep and execution starts after A&D phase

#### Regression

- Partially addressed within each iteration - fully addressed in ‘stabilization iteration’

#### Gap Handling

- Full regression test after development iterations - no new features developed - defect fixes only
### Iterative

<table>
<thead>
<tr>
<th>Focus</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Measure by iteration delivery of feature-set - A&amp;D of one iteration followed by Dev/Test in same iteration</td>
</tr>
<tr>
<td>Closure</td>
<td>Completion of A&amp;D use-cases. Completion of Dev/Test use-cases.</td>
</tr>
<tr>
<td>Delivery</td>
<td>Production of working software at end of each iteration</td>
</tr>
</tbody>
</table>
**Iterative**

**Pros**
- Simple to Implement
- Well understood
- Resources required to implement are reasonable
- Provides more opportunity to adjust plans

**Cons**
- Allows reversion at risk of inefficiency - if requirements phase is wrong, then it will be addressed in next iteration
- Customer engaged in every iteration, but feedback comes at end of iteration
Agile to DevOps Value Curve

Inspired by: http://www.collab.net/solutions/devops

Continuous Improvement: Increase productivity in small increments and measure.
Getting an Insider’s View with EEM

- **RELIABLE** — Captures reliable information within the box when connectivity to external systems are not available or reliable

- **QUICK** — Onboard logic provides instant reaction when certain condition is detected and wins precious time to capture critical information

- **DETAILED** — An insider’s view allows you to get more granularity information than you could have afforded through external communication

- **EVENT-DRIVEN** — EEM supports many event detectors integrated with IOS modules to generate events and allow you to avoid constant polling

- **DISTRIBUTED** — Scripts are distributed to each network device and runs locally when triggered, supported distributed and collaborative processing for complex network management tasks
What Can EEM Do for Me?

**Challenge 1:** Every few weeks a router is running low on memory around 2 am, and I want to find out what’s happening

- **Solution:** EEM script could be triggered based on the memory utilization, capture the memory information and send the output with Syslog or Email

**Challenge 2:** If my ACL configuration gets changed, I want to get notified, but I can’t sit there monitor it all the time

- **Solution:** EEM script could be triggered by CLI command, take a snapshot of the logged in user, changed configuration, and send an email to you

**Challenge 3:** I want to save energy, but I can’t go around turn off everyone’s IP phone everyday

- **Solution:** Timer ED can be used to trigger the execution of an EEM script to turn off your IP phone at 7pm everyday and turn it back on 7am the next day
EEM/EOT Example (Applet Policy)

! track object 8 stub
! interface FastEthernet2/0
  ip address 10.1.99.2 255.255.255.0
  duplex full
  no clns route-cache
  standby 1 ip 10.1.99.10
  standby 1 preempt
  standby 1 track 8
!

event manager applet memory-demo
  event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type
    exact entry-op lt entry-val 5120000 poll-interval 10 action 1.0 syslog
    priority critical msg "Memory exhausted; current available memory is
    $_snmp_oid_val bytes"
    action 2.0 track set 8 state down
!

When ciscoMemoryPoolFree is less than 5MB, switch HSRP traffic to Standby router.
# Where Can I Find It?

## -- EEM Version/Product Support Matrix

**CISCO ACCESS ROUTERS - Current Models**

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco 800 Series</th>
<th>Cisco 1800 Series</th>
<th>Cisco 2800 Series</th>
<th>Cisco 3800 Series</th>
<th>Cisco 1900 Series</th>
<th>Cisco 2900 Series</th>
<th>Cisco 3900 Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td>12.3(11)T</td>
<td>12.3(11)T</td>
<td>12.3(11)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td></td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td>12.4(20)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td>12.4(22)T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
<td>15.0(1)M</td>
</tr>
<tr>
<td>3.2</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
<td>15.1(3)T</td>
</tr>
<tr>
<td>4.0</td>
<td></td>
<td>15.2(2)T</td>
<td>15.2(2)T</td>
<td>15.2(2)T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CISCO ACCESS ROUTERS - Old Models**

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco 1700 Series</th>
<th>Cisco 2600 Series</th>
<th>Cisco 2600XM Series</th>
<th>Cisco 2691 Series</th>
<th>Cisco 3600 Series</th>
<th>Cisco 3700 Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>12.3(4)T</td>
<td>12.3(4)T</td>
<td>12.3(4)T</td>
<td></td>
<td>12.3(4)T</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
<td>12.3(14)T1</td>
</tr>
<tr>
<td>2.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Where Can I Find It?
-- EEM Version/Product Support Matrix

### CISCO SERVICE AGGREGATION/CORE ROUTERS

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Cisco ASR1000 Series</th>
<th>Cisco 7200 Series</th>
<th>Cisco 7301</th>
<th>Cisco 7304</th>
<th>Cisco 7600 Series</th>
<th>Cisco UBR 10000</th>
<th>Cisco UBR 7200</th>
<th>Cisco 12000 Series</th>
<th>Cisco XR 12000</th>
<th>Cisco CRS-1</th>
<th>Cisco ASR 9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>12.0(26)S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>FM</td>
<td>FM</td>
<td>FM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>12.2(27)SBC</td>
<td>12.2(28)SB</td>
<td>12.2(18)SXF5</td>
<td>12.2(28)SB</td>
<td>12.2(28)SB</td>
<td>FM</td>
<td>FM</td>
<td>FM</td>
<td>FM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.4(2)T</td>
<td>12.4(2)T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>2.1XE</td>
<td>12.4(11)T</td>
<td>12.2(33)SB</td>
<td>12.2(33)SRB</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>12.2(33)SB</td>
<td>FM</td>
<td>FM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>12.2(33)XN RLS7</td>
<td>12.4(20)T</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>FM</td>
<td>FM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>12.2(33)XN RLS7</td>
<td>12.4(22)T</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>12.2(33)SRE</td>
<td>FM</td>
<td>FM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>15.1(3)S3</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
</tr>
<tr>
<td>3.2</td>
<td>Planning</td>
<td>15.1(3)T</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
</tr>
<tr>
<td>4.0</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
</tr>
</tbody>
</table>

### CISCO CATALYST SWITCHES

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Catalyst 3000 Switches</th>
<th>Cisco 3400ME Switches</th>
<th>Catalyst 4500 Switches</th>
<th>Catalyst 4900 Switches</th>
<th>Catalyst 6500 Switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>12.2(40)SE</td>
<td>12.2(40)SE</td>
<td>12.2(44)SG</td>
<td>12.2(44)SG</td>
<td>12.2(33)SXH</td>
</tr>
<tr>
<td>2.3</td>
<td>12.2(50)SE</td>
<td>12.2(50)SE</td>
<td>12.2(52)SG</td>
<td>12.2(52)SG</td>
<td>12.2(33)SXI</td>
</tr>
<tr>
<td>3.0</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td>12.2(54)SG1</td>
<td>12.2(54)SG1</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>12.2(52)SE</td>
<td>12.2(52)SE</td>
<td>15.0(2)SG</td>
<td>15.0(2)SG</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
<td>Planning</td>
</tr>
<tr>
<td>Event Detector</td>
<td>Description (ED Triggers, based on ...)</td>
<td>EEM Version in IOS</td>
<td>IOS XR</td>
<td>NX-OS</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------</td>
<td>--------------------</td>
<td>---------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td>2.0</td>
<td>2.1</td>
<td>2.2</td>
</tr>
<tr>
<td>Syslog</td>
<td>RegEx match of local syslog message</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNMP Notif</td>
<td>SNMP MIB Variable Threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Watchdog</td>
<td>IOS process or subsystem activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface Counter</td>
<td>(Interface) Counter Threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timer</td>
<td>Designated Time or Interval</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counter</td>
<td>Change of a designated counter value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application specific</td>
<td>An IOS subsystem or policy script</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLI</td>
<td>RegEx match of input via command</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OIR</td>
<td>Hardware online insertion and removal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>none</td>
<td>No trigger, used in conjunction with</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERM</td>
<td>Embedded Resource Manager (ERM) events</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EOT</td>
<td>Enhanced Object Tracking variable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>IOS Redundancy Facility (switchover)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOLD</td>
<td>Generic Online Diagnostics (GOLD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNMP Proxy</td>
<td>Incoming remote SNMP Notification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML RPC</td>
<td>Incoming XML message</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routing</td>
<td>State change of Routing Protocols</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netflow</td>
<td>Traffic Flow information from Netflow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPSLA</td>
<td>IPSLA events (supersedes EOT for EEM / IPSLA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLI enhanced</td>
<td>Integrates CLI Ed with the XML PI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNMP Object</td>
<td>Intercept SNMP GET/SET requests</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neighbor Disco</td>
<td>CDP, LLPD, Link up/down events</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identity</td>
<td>802.1x and MAB authentication events</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td>MAC Address Table entry changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enhanced CLI ED</td>
<td>Provide AAA-like info to CLI policies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESM enhancements for syslog ED</td>
<td>Filter syslog messages in line</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>Register for environmental monitoring</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statistics</td>
<td>Threshold crossing of a statistical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fan (absent / bad)</td>
<td>Presence and State of a Fan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module failure</td>
<td>Occurrence of a Module Failure Event</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storm Control</td>
<td>Occurrence of a Storm Control Event</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>Temperature Sensor Thresholds</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For Your Reference

BRKNMS-2446