

ZENOSS SERVICE DYNAMICS SERVICE IMPACT TRAINING



Copyright and Trademark Notices

Copyright © 2015–2019 Zenoss, Inc. All rights reserved.
Confidential and Proprietary Information of Zenoss.

Zenoss and the Zenoss logo are trademarks or registered trademarks of Zenoss, Inc. in the United States and other countries. All other trademarks, logos, and service marks are the property of Zenoss or other third parties. Use of these marks is prohibited without the express written consent of Zenoss, Inc. or the third-party owner.

Cisco, Cisco UCS, and Cisco Unified Computing System are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

UNIX is a registered trademark of The Open Group.

Linux is a registered trademark of Linus Torvalds.

Oracle, the Oracle logo, MySQL, and Java are registered trademarks of the Oracle Corporation and/or its affiliates.

Tomcat is a trademark of the Apache Software Foundation.

VMware, VMware vCloud, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

RabbitMQ is a trademark and/or registered trademark of Pivotal Software, Inc. in the United States and/or other countries.

Active Directory, Microsoft, SQL Server, Windows, Windows PowerShell, and Windows Server are registered trademarks of Microsoft Corporation in the United States and other countries.

All other trademarks, registered trademarks, product names, and company names or logos mentioned in this document are the property of their respective owners.

Revision: 5.2.3-ILT-V1.1

Service Impact Overview

- Service Impact provides a service-centric view of your IT infrastructure
- In Service Impact, a service consists of the infrastructure elements that support a specific business-level application
- Events from Resource Manager are interpreted in the context of the service(s) they impact
- Where Resource Manager answers “What happened?”, Service Impact addresses “So what?”

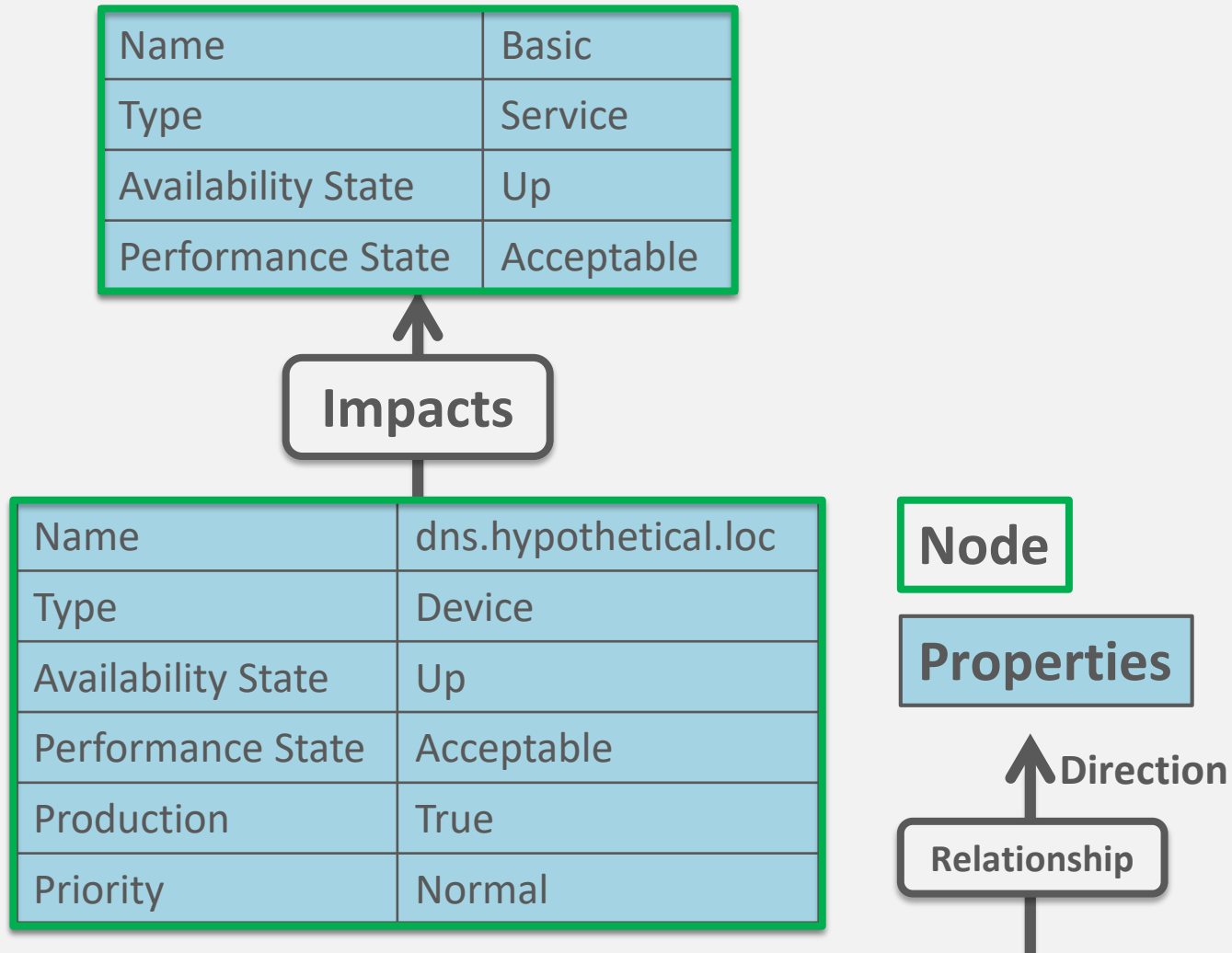
Benefits

- Constantly updated service status
- Automated service assurance and dynamic service impact analysis
- Automated root cause analysis
- Enhanced event management

How Service Impact Works

- Requires Resource Manager
- Event-Driven
- Adds two new services:
 - **Impact**: Performs state propagation, and provides service alerting and root-cause analysis
 - **zenimpactstate**: Performs state calculations and event filtering for the **Impact** service
- Adds Graph Database
 - Incorporates a copy of the object database device model
 - Adds relationships between service elements, and the state of each element

Graph Database

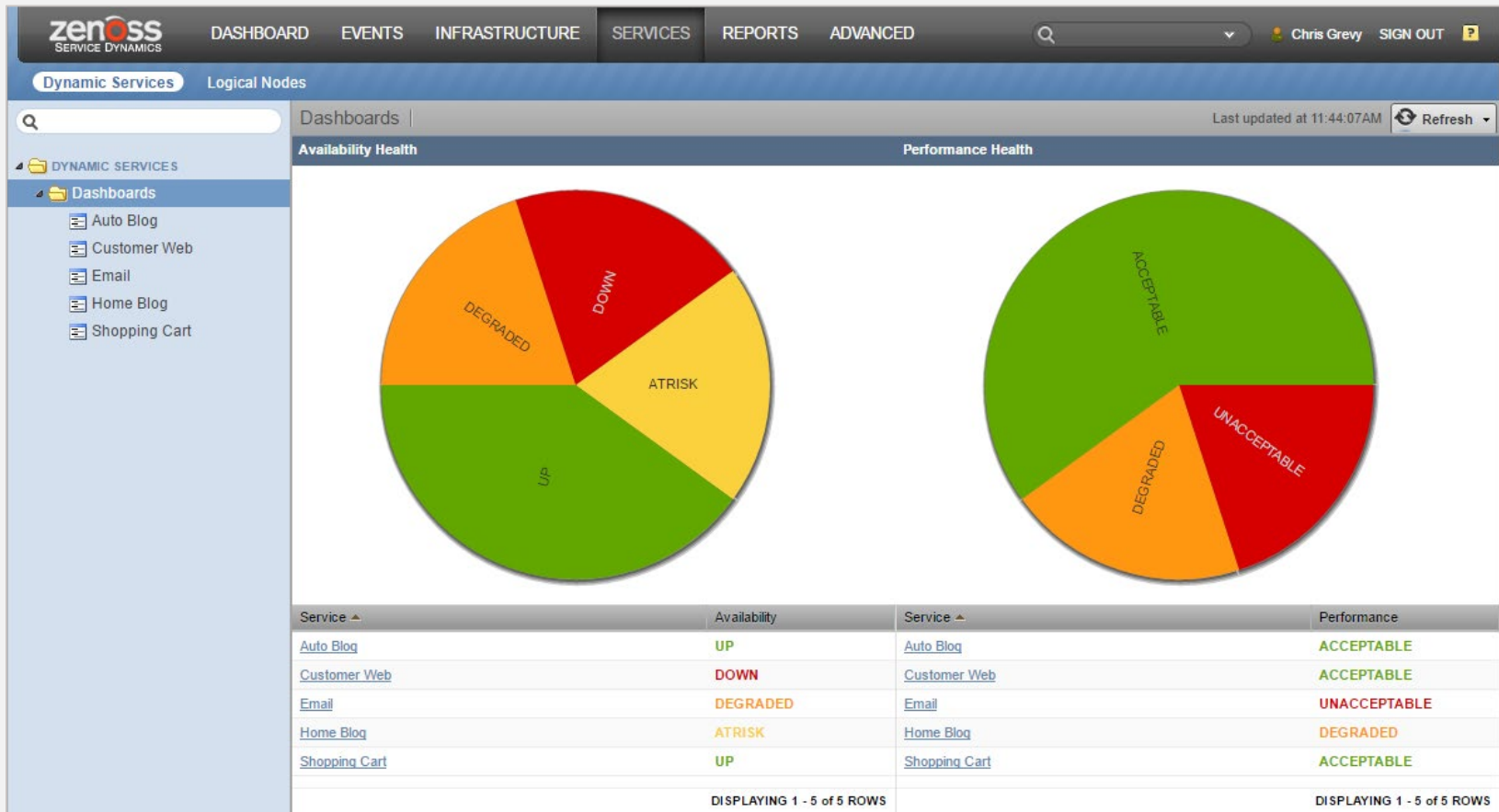


Service States

- **Availability** Health
 - Up, At Risk, Degraded, Down
- **Performance** Health
 - Acceptable, Degraded, Unacceptable
- Select a Dynamic Service Organizer on the **Services** page to see the status of all the Dynamic Services it contains

Dynamic Service Organizer View

View the state of each service in the selected organizer



Service Views: Members

Add or remove service elements

The screenshot displays the Zenoss Service Dynamics interface. The top navigation bar includes links for DASHBOARD, EVENTS, INFRASTRUCTURE, SERVICES, REPORTS, and ADVANCED. The left sidebar shows a tree view of Dynamic Services, with 'Agent Blogs - Common' selected. The main content area shows the 'MySQL' service under 'Agent Blogs - Common'. The service health is indicated by a red 'A' and a green 'P' button. A dropdown menu labeled 'Members' is open, showing a list of service elements. The 'Add to Service' dialog box is also visible, showing search results for 'eth0'.

Dynamic Services Logical Nodes

MySQL /Agent Blogs - Common

SERVICE HEALTH: A P

Members ▼

Currently in Service: + -

Health	Type
A P	OSProcess
A P	FileSystem
A P	IpService

Add to Service

Search: eth0

Interface(3)

Search Results

Icon	Description
eth0	(dns.hypothetical.loc)
eth0	(mysql.hypothetical.loc)
eth0	(wordpress.hypothetical.loc)

Page 1 of 1 | DISPLAYING 1 - 3 of 3 ROWS

ADD CLOSE

Service Views: Impact Events

View impacting events and root cause analysis

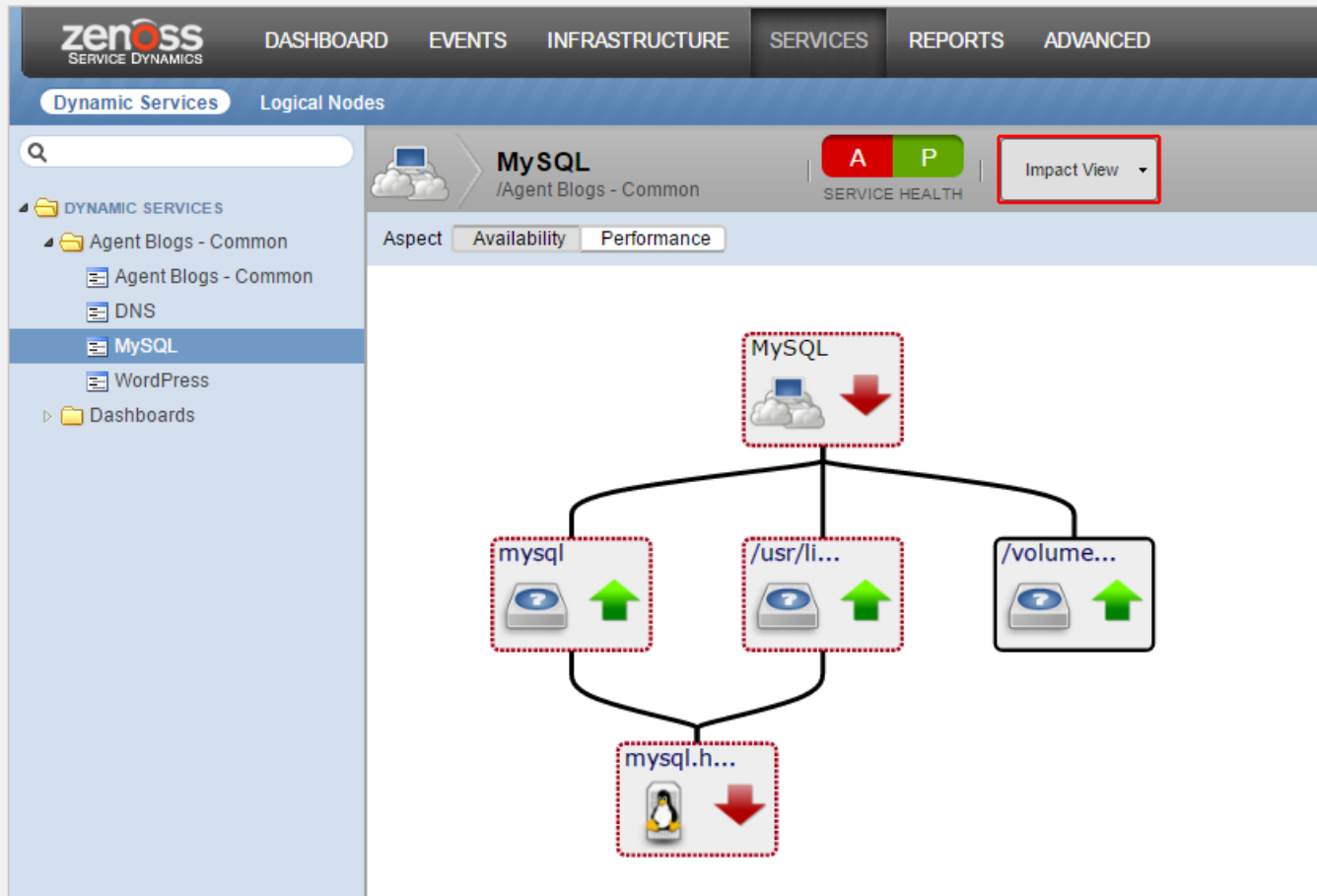
The screenshot displays the Zenoss Service Dynamics interface. The top navigation bar includes links for DASHBOARD, EVENTS, INFRASTRUCTURE, SERVICES, REPORTS, and ADVANCED. The left sidebar shows a tree view of Dynamic Services, with 'MySQL' selected under 'Agent Blogs - Common'. The main content area shows the 'MySQL' service view. A red box highlights the 'Impact Events' dropdown menu. Below this, a table lists impact events with columns for Status, Severity, Service, Event Class, Summary, First Seen, Last Seen, and Count. The table shows one event with a status of 'Critical' and a count of 52. Below the table, a detailed view of the event is shown, including a table with columns for Confidence, Status, Severity, Resource, Component, Event Class, Summary, and First Seen. The detailed view shows three events with confidence levels of 55, 28, and 19.

Status	Severity	Service	Event Class	Summary	First Seen	Last Seen	Count
Critical	High	MySQL	/Service/State/Availability	Se...	2013-10-09 09:00:28	2013-10-09 09:47:49	52

Confidence	Status	Severity	Resource	Component	Event Class	Summary	First Seen
55	Critical	High	vmhost12.hy...		/Status/Pi...	vmhost12.hypothetica...	2013-10-09 09:00:27
28	Critical	High	vmhost12.hy...	VM-mysql...	/Status/Pi...	VM-mysql.hypothetic...	2013-10-09 09:46:43
19	Critical	High	mysql.hypoth...		/Status/Pi...	mysql.hypothetical.lo...	2013-10-09 09:47:49

Service Views: Impact View

View the service graph



Exercise #0: Connect to Your Zenoss Instance

- ❑ Connect to your Resource Manager instance:

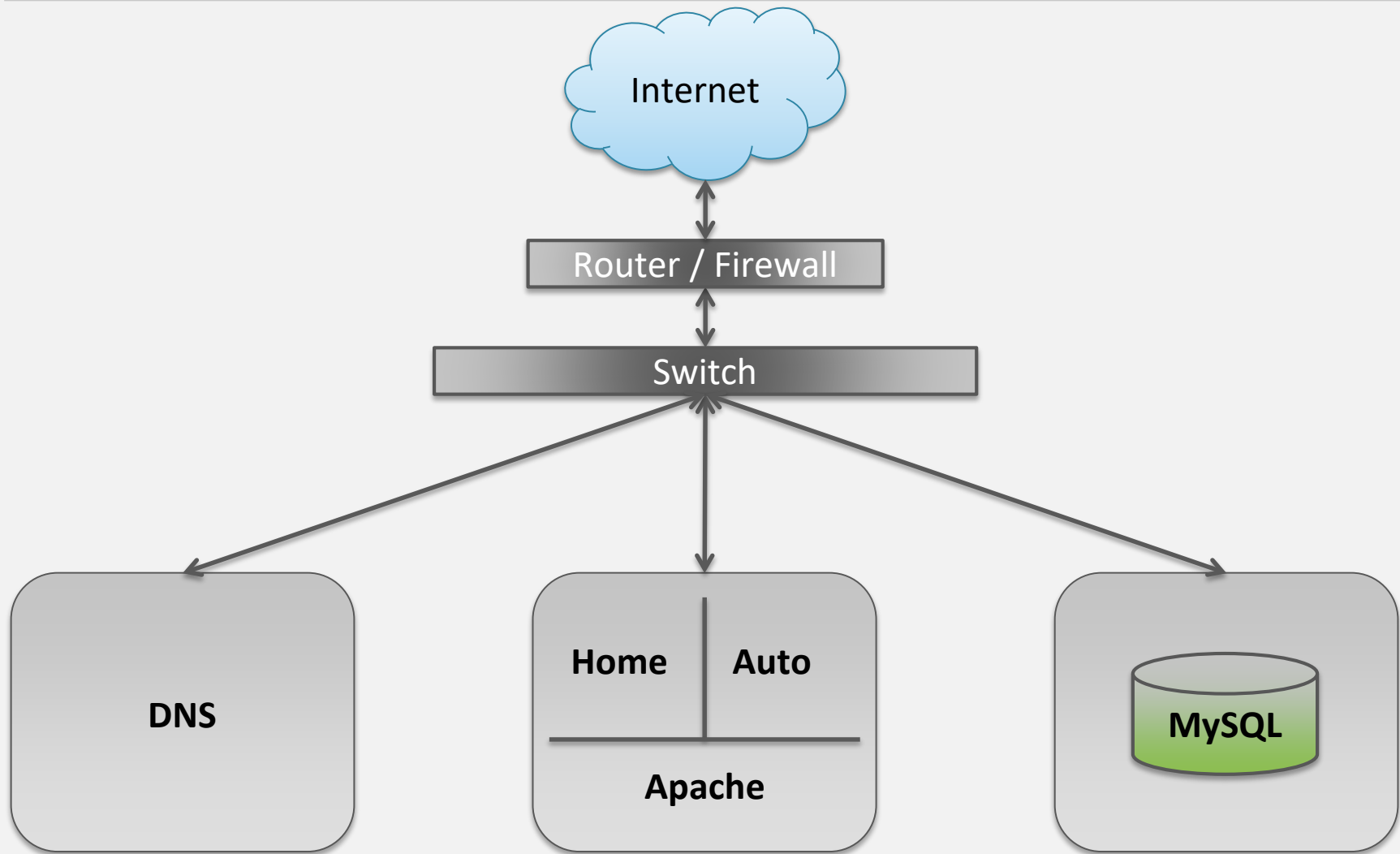
`https://zenoss5.region-trainX.zenoss.com` where:

- **region** is either blank (in which case you should omit the hyphen ("-")), or one of **in**, **ir**, **pre**, **sa**, or **sing**
- **X** is the number of your assigned lab instance

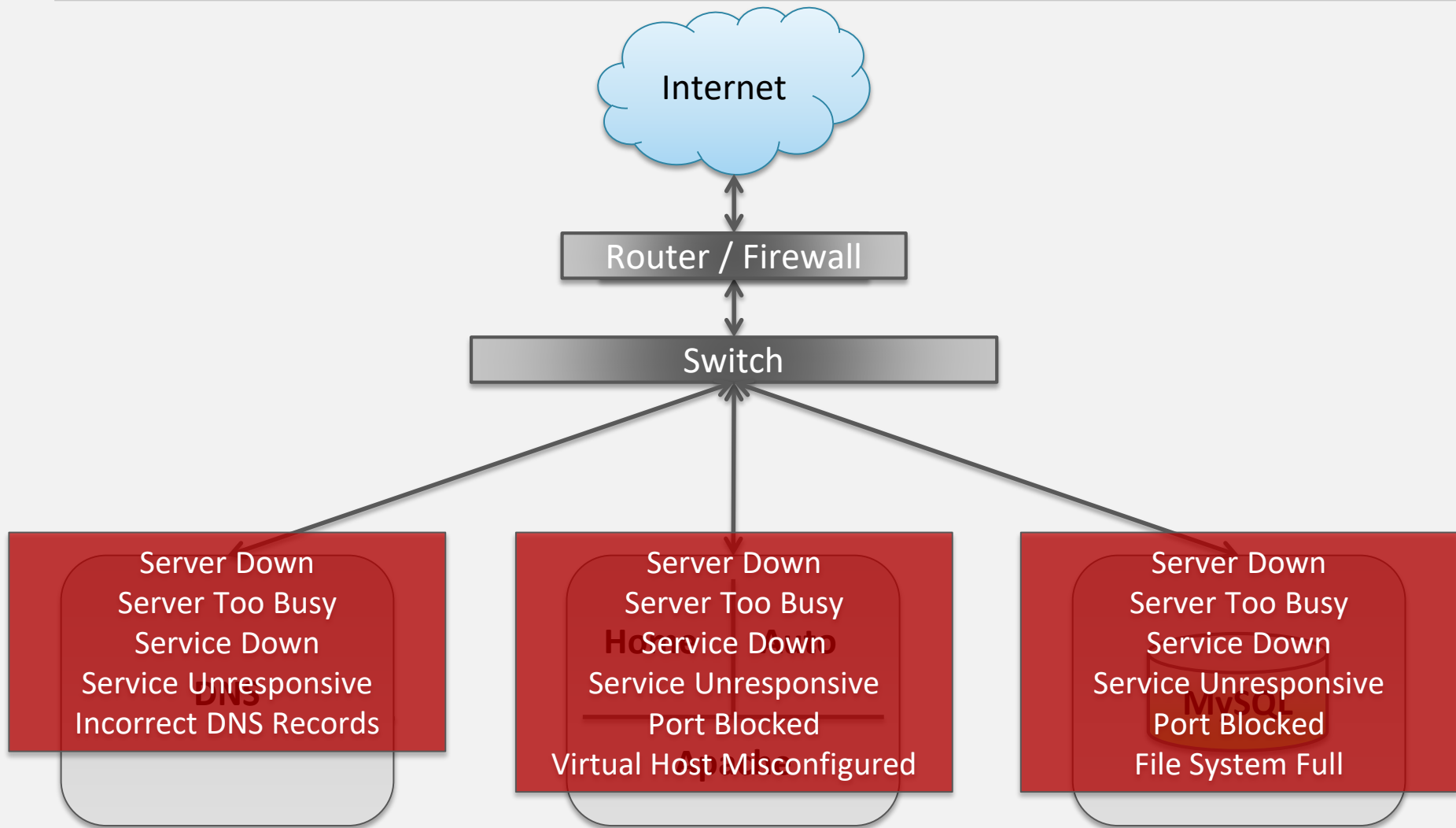
If you are unsure of which region your lab instance is in, or which instance has been assigned to you, check with your instructor before proceeding

- ❑ Your browser may post a privacy or security warning; it is safe to continue
- ❑ Enter **training** as the *Username*
- ❑ Enter the password provided by your instructor

Case Study: Hypothetical Insurance Blogs



Case Study: What Can Go Wrong



Case Study: Device Monitoring Examples

- Device Status
- Device Components
 - Processes
 - IP Services
 - File Systems

Case Study: Application Monitoring Examples

- Apache
- HTTP
 - Time to Retrieve / Page Size
 - Synthetic Web Transactions (Twill)
- MySQL
- DNS

Impact Dynamic Services

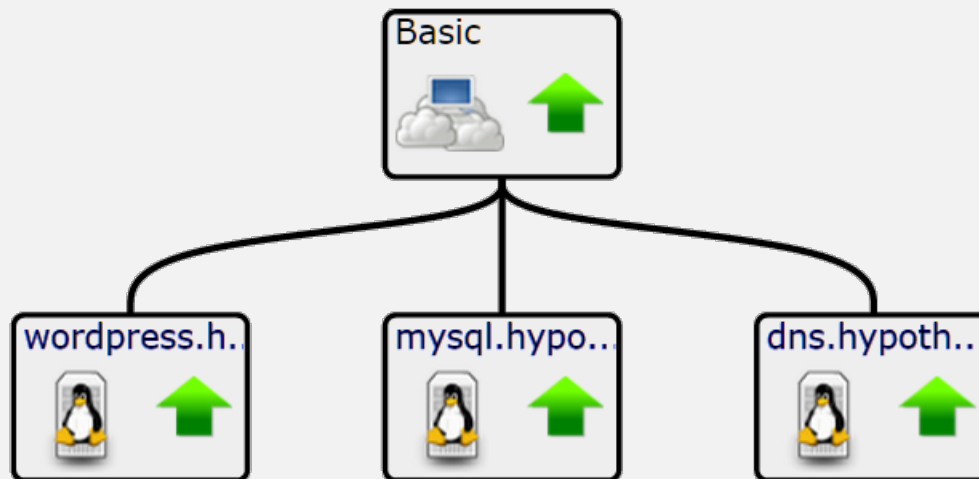
A Dynamic Service is a collection of members:

- **Devices**
- Device Components
- Other Dynamic Services
- Logical Nodes

Exercise #1: Create a Basic Service

- ❑ Create a new service named **Basic**
- ❑ Add the **dns**, **wordpress**, and **mysql** devices to the service

Locate the devices in the **Add to Service** dialog by searching for “hyp” and then clicking on **Devices**



State Mappings

The state of a device or component service element is determined by two properties of incoming events:

- The type of impact, **Availability** or **Performance**, is derived from the **event class**
- The state of the impacted element is derived from the **event severity**
- Well-configured Monitoring Template **Thresholds** are critical to successfully implementing Impact
- By default, a service's state is the most severe state of any of its constituent members

Event Class Mapping

The top-level event class of a new event determines the impact type:

- Event Class **/Status/*** ➞ Service **Availability**
(excludes /Status/Snmp and /Status/WMI)

Status	Severity	Resource	Component	Event Class	Summary
...	...				
	!!	Basic		/Service/State/Availability	Service Basic is DOWN.
	!!	mysql.hypothetical.loc		/Status/Ping	mysql.hypothetical.loc is DOWN!

- Event Class **/Perf/*** ➞ Service **Performance**

Status	Severity	Resource	Component	Event Class	Summary
...	...				
	!	Basic		/Service/State/Performance	Service Basic is DEGRADED.
	!	wordpress.hypothetical.loc		/Perf/CPU	threshold of low CPU idle not me

Event Severity Mapping

The severity of a new event determines the impact state of matching device or component members:

Event Severity	Impact State			
		Availability		Performance
Clear		UP		ACCEPTABLE
Debug		UP		(no mapping)
Info		UP		(no mapping)
Warning		AT RISK		DEGRADED
Error		DEGRADED		DEGRADED
Critical		DOWN		UNACCEPTABLE

Monitoring Templates (Review)

Monitoring templates provide:

- **Data Sources**
 - Each data source functions as an individual query using a specific protocol or command
 - Results are parsed and returned as one or more **Data Points**
- **Thresholds**
 - Define limits on performance metric values which generate events when breached
 - Implemented by the collector
- **Graph Definitions**
 - Render one or more **Data Points** graphed against collection time

Exercise #2a: How Events Impact Service States

Use manually created “synthetic events” to explore the relationship between new events and the resulting service state

- ❑ Close any existing events on the event console
- ❑ Create an event with these properties:

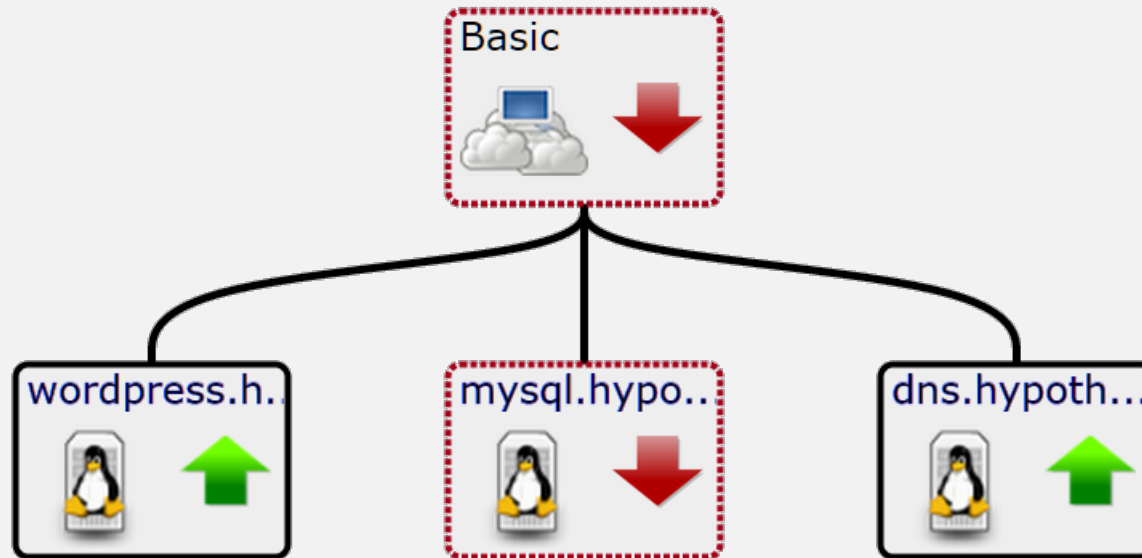
Summary	Host Down
Device	mysql.hypothetical.loc
Component	[<i>leave blank</i>]
Severity	Critical
Event Class Key	[<i>leave blank</i>]
Event Class	/Status/Ping
Collector	localhost

Exercise #2a: How Events Impact Service States

Observe:

- ❑ New events displayed on the **Event Console**: the synthetic event and the Impact service event
- ❑ Changes to the pie charts on the top-level **Services** page
- ❑ The **Health** column of the affected service element (device) on the service's **Members** page
- ❑ The Impact service event and corresponding impacting event on the service's **Impact Events** page
- ❑ The state of the graph on the service's **Impact View** page

Exercise #2a: How Events Impact Service States



Exercise #2b: How Events Impact Service States

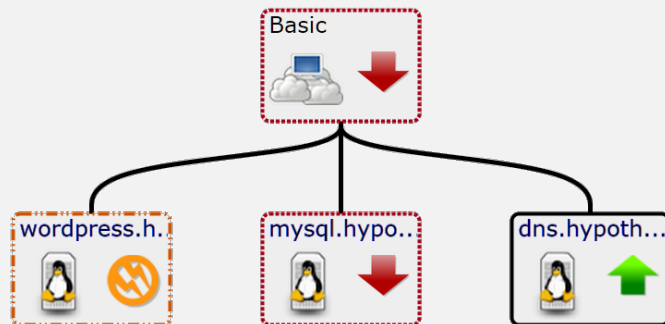
- ❑ Create two additional events as shown below and observe the changes on the Event Console

Summary	HTTP worker process died
Device	wordpress.hypothetical.loc
Component	[leave blank]
Severity	Error
Event Class Key	[leave blank]
Event Class	/Status/HTTP
Collector	localhost

Summary	Host Down
Device	mysql.hypothetical.loc
Component	[leave blank]
Severity	Critical
Event Class Key	[leave blank]
Event Class	/Status/Ping
Collector	localhost

Exercise #2c: Impact Event Management

- ❑ Click on the **Basic** service in the *Resource* row in the Event Console to go to the service's Members page
- ❑ Go to the **Impact View** page to see the service graph:



- ❑ Go to the **Impact Events** page and select the service event
- ❑ Close the *Host Down* event
- ❑ Observe the changes in the service event

Exercise #2d: Clear Events and Service States

- ❑ Create an event to clear the open */Status/HTTP* event:

Summary	HTTP worker process died
Device	wordpress.hypothetical.loc
Component	[leave blank]
Severity	Clear
Event Class Key	[leave blank]
Event Class	/Status/HTTP
Collector	localhost

- ❑ On the **Impact Events** page, verify that the */Status/HTTP* event and the corresponding service event have been cleared

Exercise #2e: Simulate a Threshold Event

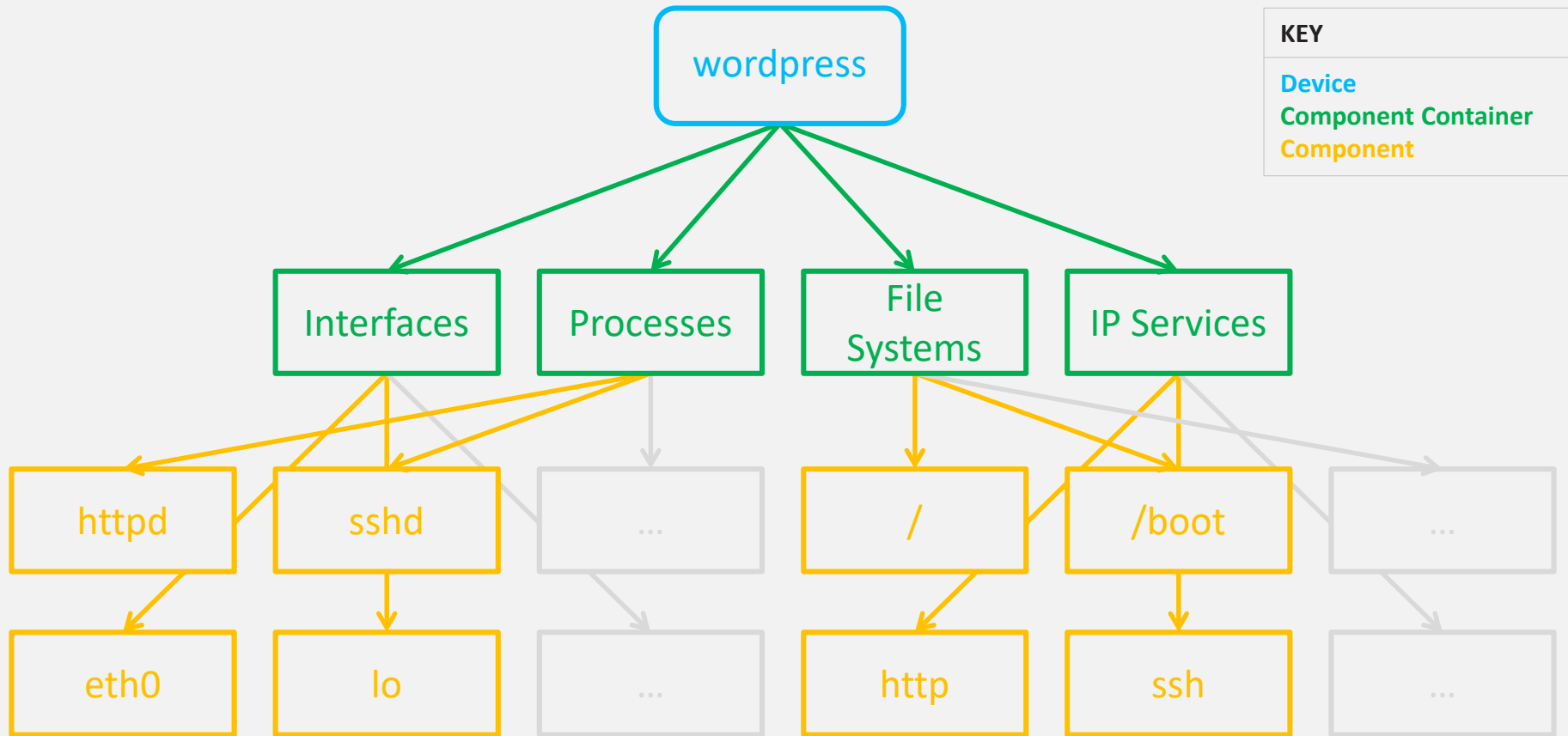
- ❑ Navigate to the **Device** Monitoring Template for the **dns.hypothetical.loc** device
- ❑ Open the **high load** Threshold
- ❑ Note the values for the *Severity* and *Event Class* properties
- ❑ Create a matching synthetic event for the **dns.hypothetical.loc** device
- ❑ Observe the results on the Event Console
- ❑ Create a matching synthetic event for the **mysql.hypothetical.loc** device
- ❑ Observe the results on the Event Console noting the *Count* field of the service event

Impact Dynamic Services

A Dynamic Service is a collection of elements:

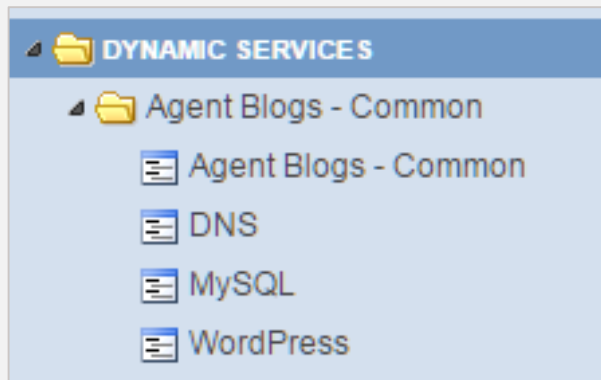
- Devices
- **Device Components**
- Other Dynamic Services
- Logical Nodes

Server Component Monitoring



Exercise #3: Prepare for Component Monitoring

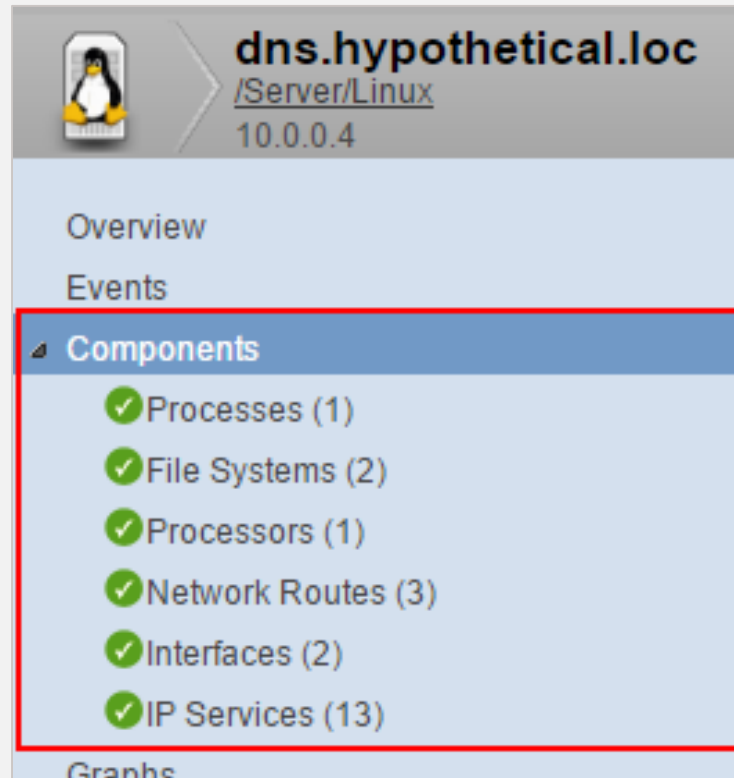
- ❑ Delete the **Basic** service created in the last exercise
- ❑ Create a new top-level service organizer named **Agent Blogs - Common**
- ❑ Underneath the new organizer, create four new services named **Agent Blogs - Common**, **DNS**, **MySQL**, and **WordPress**



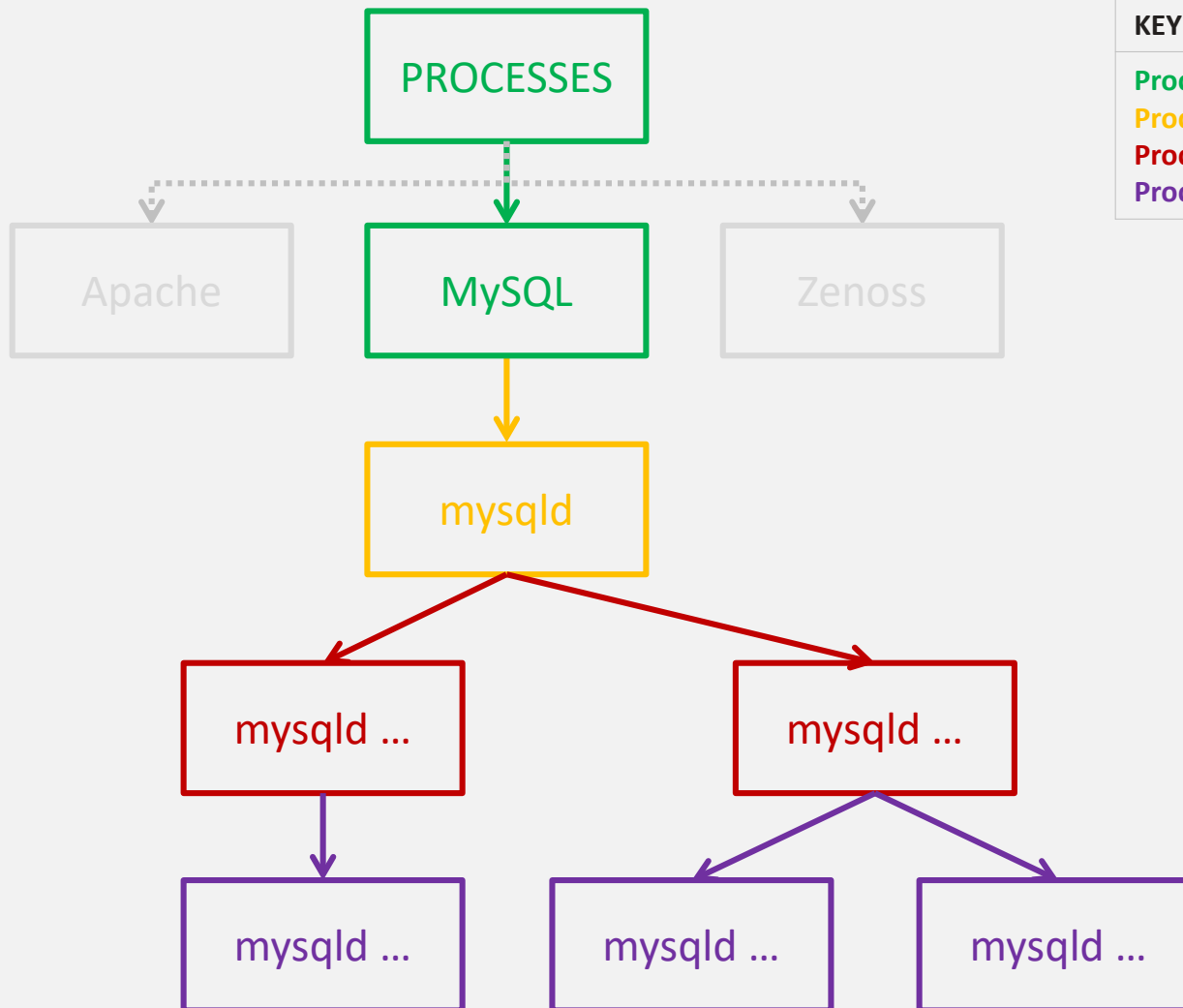
Server Component Monitoring: Overview

- Device components are properties of a device that usually consist of multiple instances
- Components are organized by type (*Interfaces, File Systems, Processes, ...*)
- The set of available component types can vary between device classes
- Components are discovered at model time; if an existing component — a process, for example — is not present when a device is remodeled, it will be removed, **but** ...
- Components can be locked to prevent changes or deletion

Server Component Monitoring: Overview



Process Monitoring



KEY

Process Class Organizer

Process Class

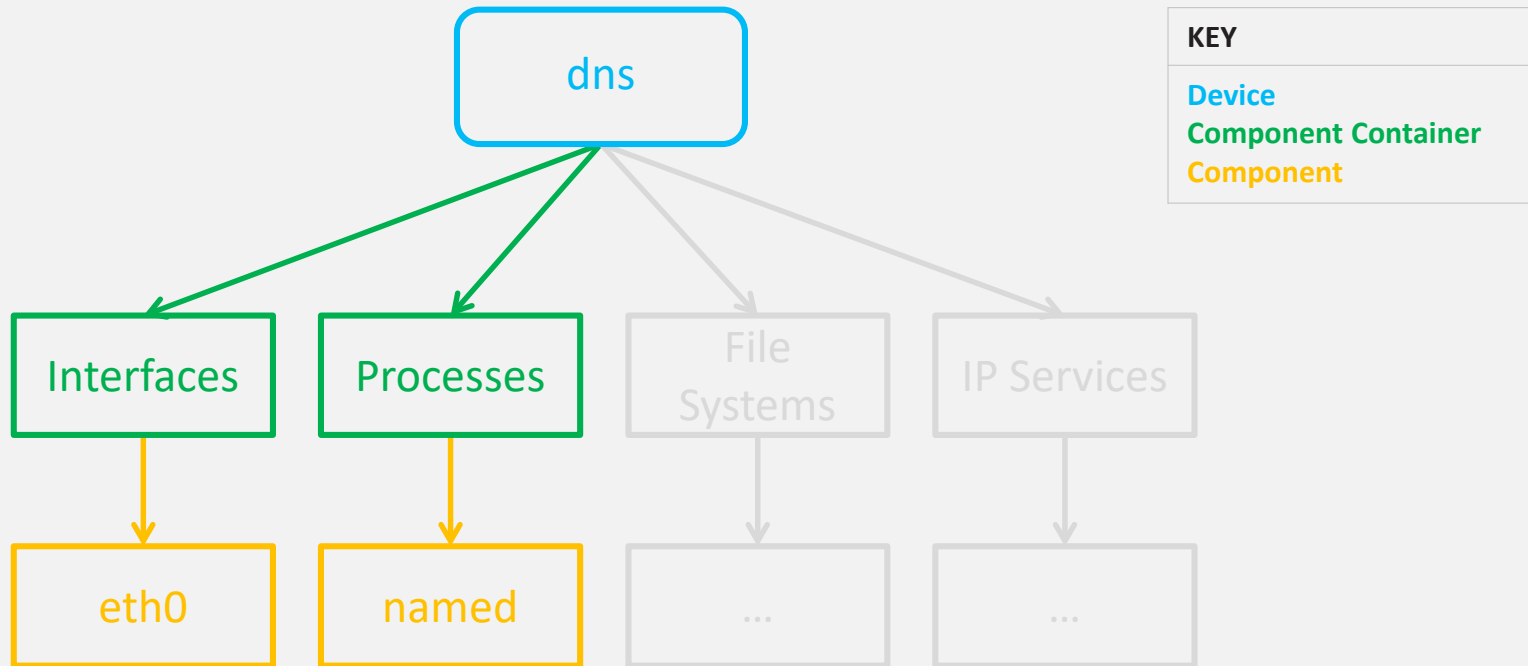
Process Set

Process

Sample Process List

```
automount --pid-file /var/run/autofs.pid
/usr/sbin/snmpd -LS0-6d -Lf /dev/null -p /var/run/snmpd.pid
/bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql
--socket=/var/lib/mysq
/usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql
--user=mysql --l
/usr/sbin/abrt
crond
/usr/sbin/atd
/usr/bin/rhsmcertd
/usr/sbin/certmonger -S -p /var/run/certmonger.pid
```

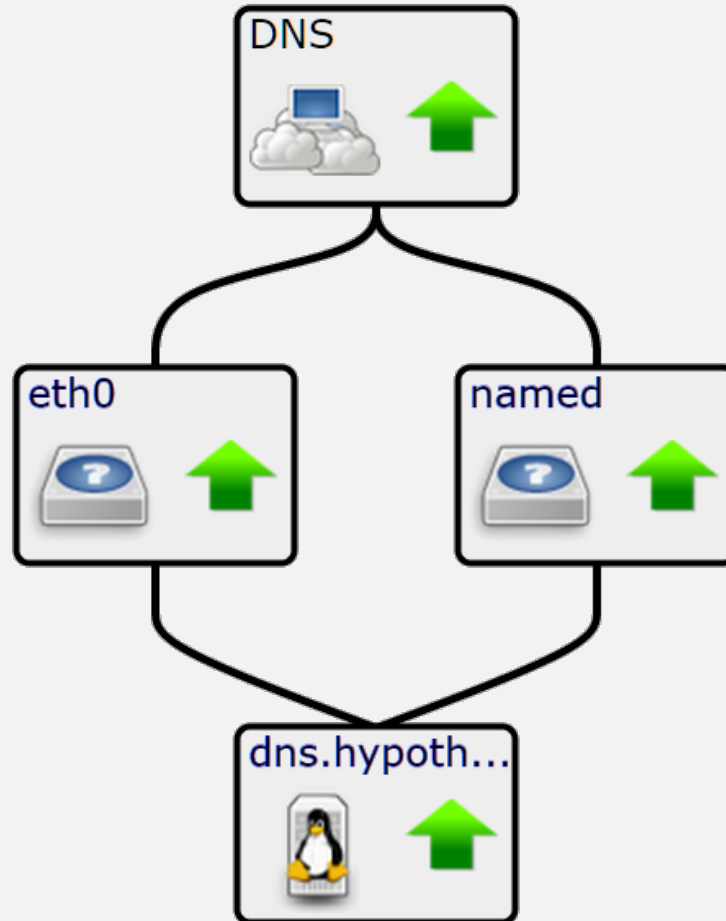
Component Monitoring: DNS Server



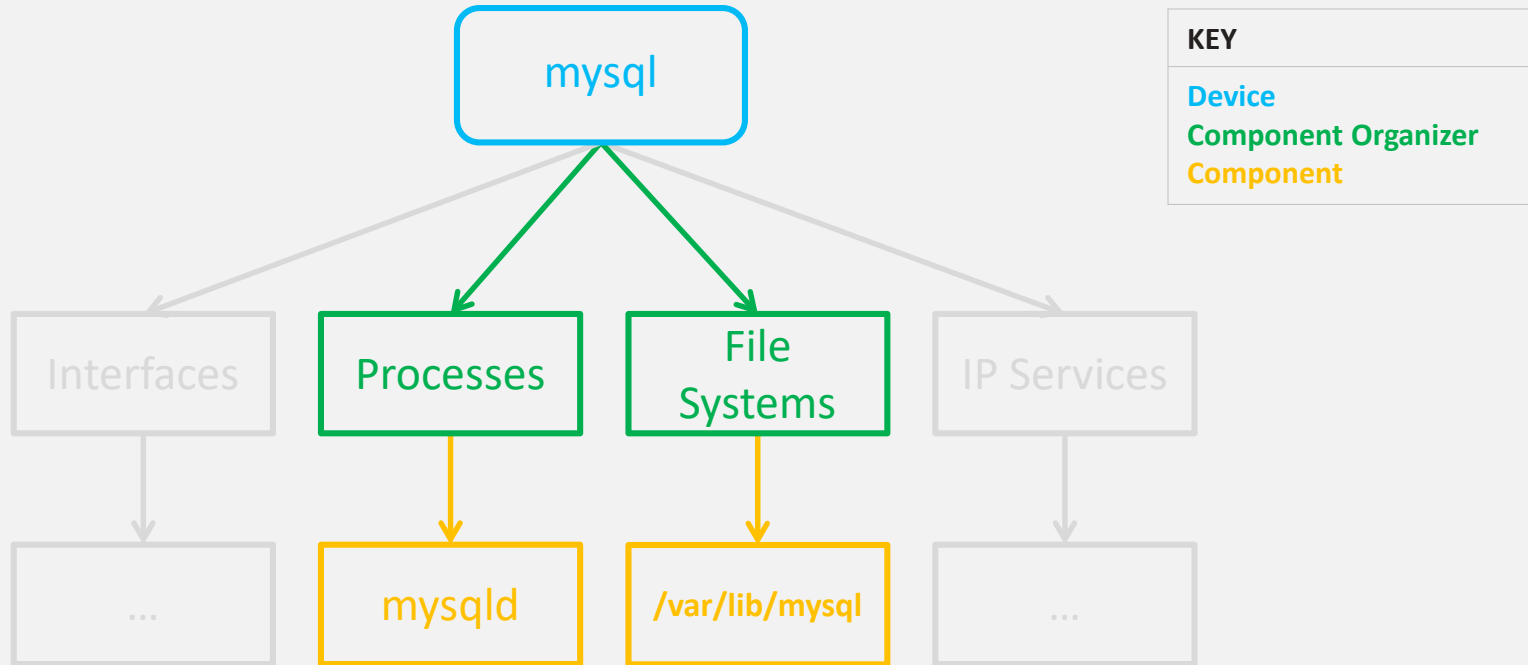
Exercise #4: Add DNS Service Members

- ❑ In the **/Agent Blogs - Common/DNS Service**, add the **dns.hypothetical.loc**
 - ❑ **eth0** Interface
 - ❑ **/usr/sbin/named** Process

DNS Service



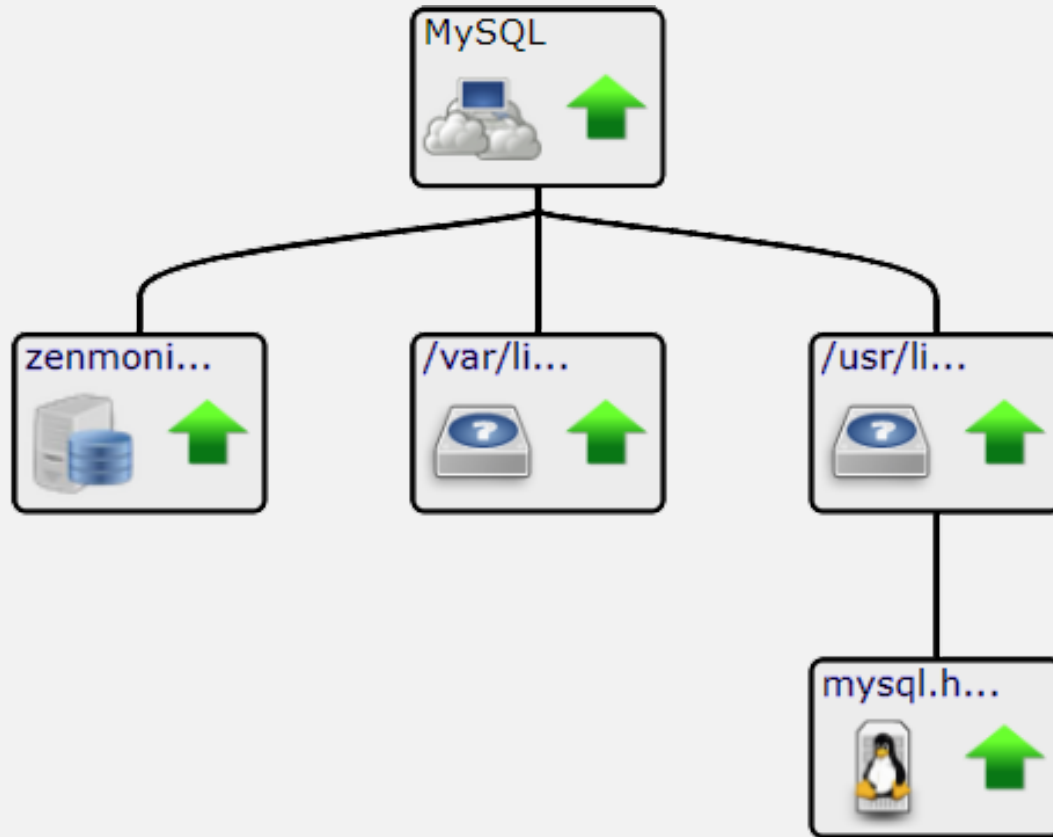
Component Monitoring: MySQL Server



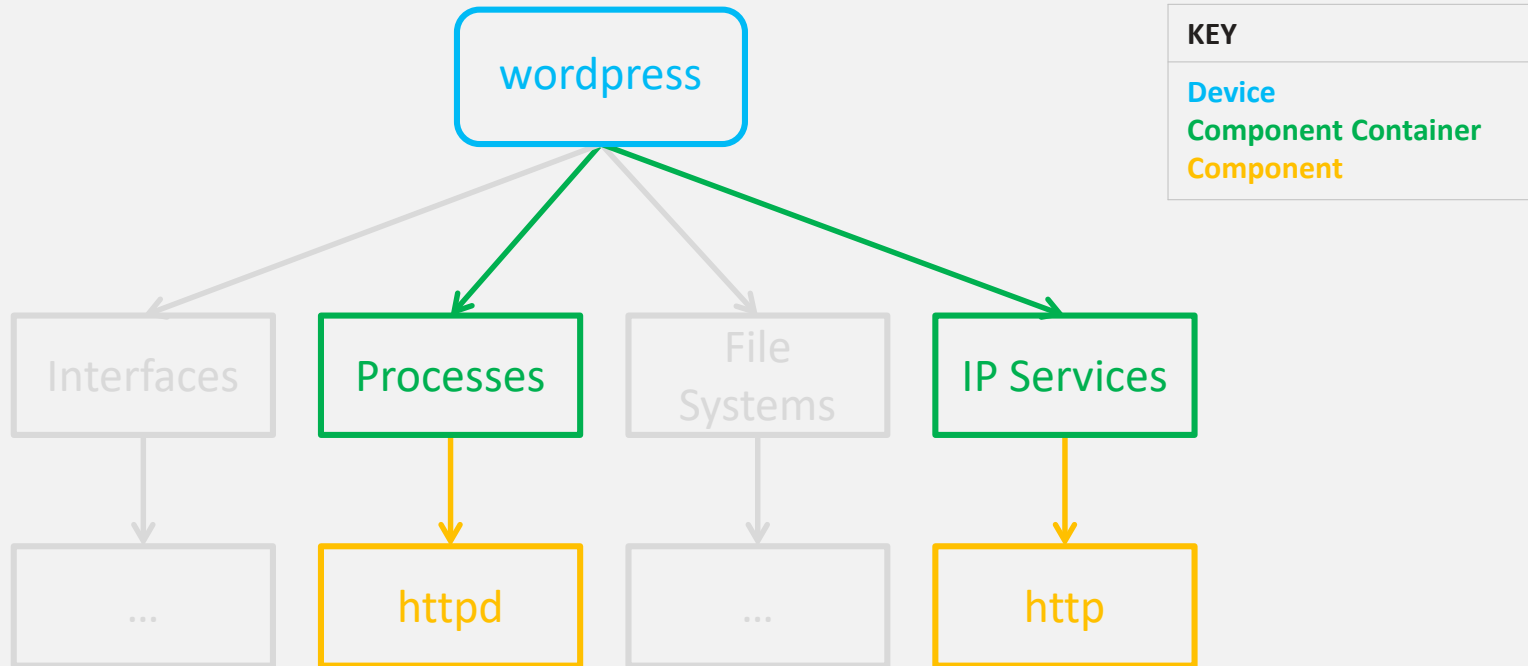
Exercise #5: Add MySQL Service Members

- ❑ In the **/Agent Blogs - Common/MySQL** Service, add the **mysql.hypothetical.loc**
 - ❑ **/usr/libexec/mysqld** Process
 - ❑ **/var/lib/mysql** File System

MySQL Service



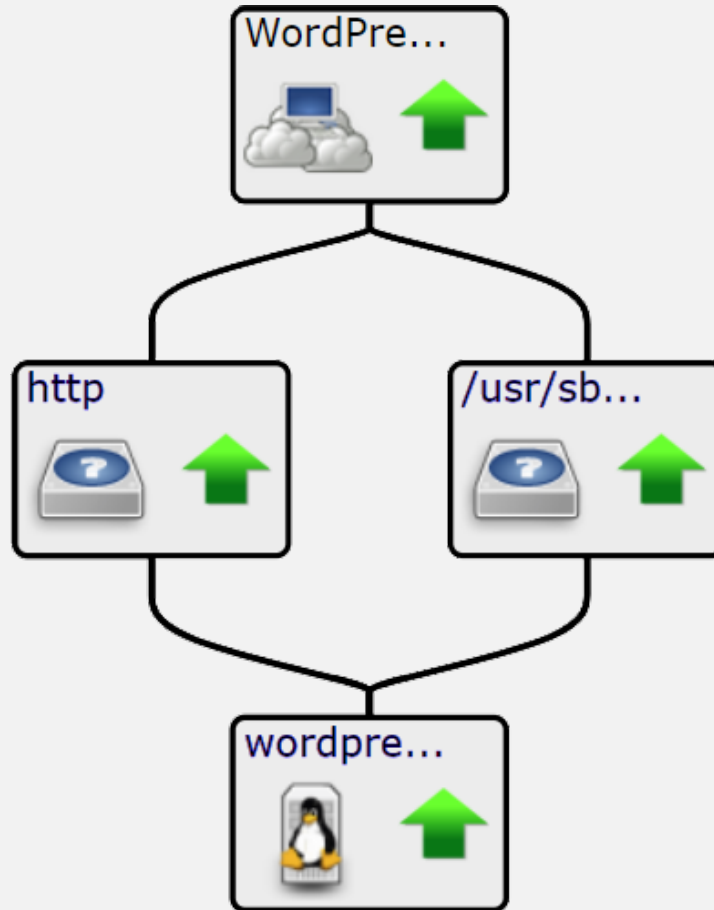
Component Monitoring: WordPress Server



Exercise #6: Add WordPress Service Members

- ❑ In the **/Agent Blogs - Common/WordPress** Service, add the **wordpress.hypothetical.loc**
 - ❑ **/usr/sbin/http** Process
 - ❑ **http** IP Service

WordPress Service



Application Monitoring

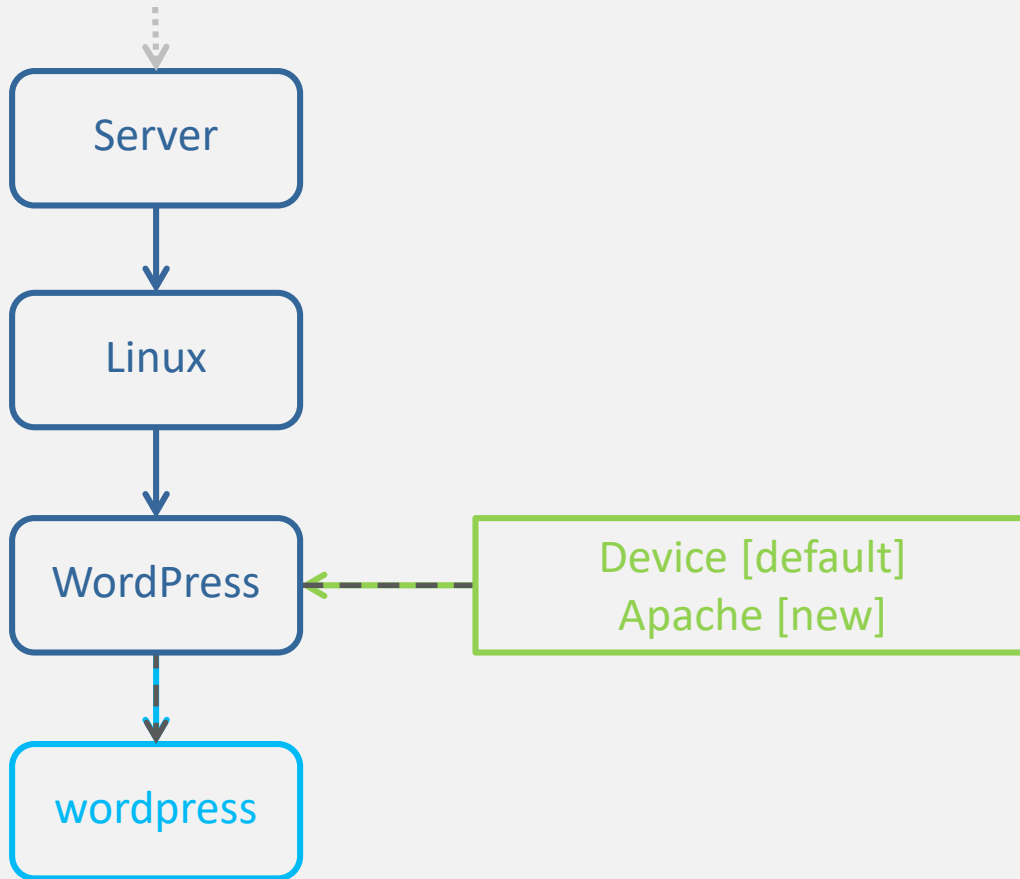
- Zenoss provides ZenPacks for monitoring a wide variety of applications
- Depending on the ZenPack, application monitoring may be implemented at the device level or the component level



See the ZenPack Directory at

<https://www.zenoss.com/product/zenpacks>

Adding Apache Monitoring



KEY

Device Class

Template Binding

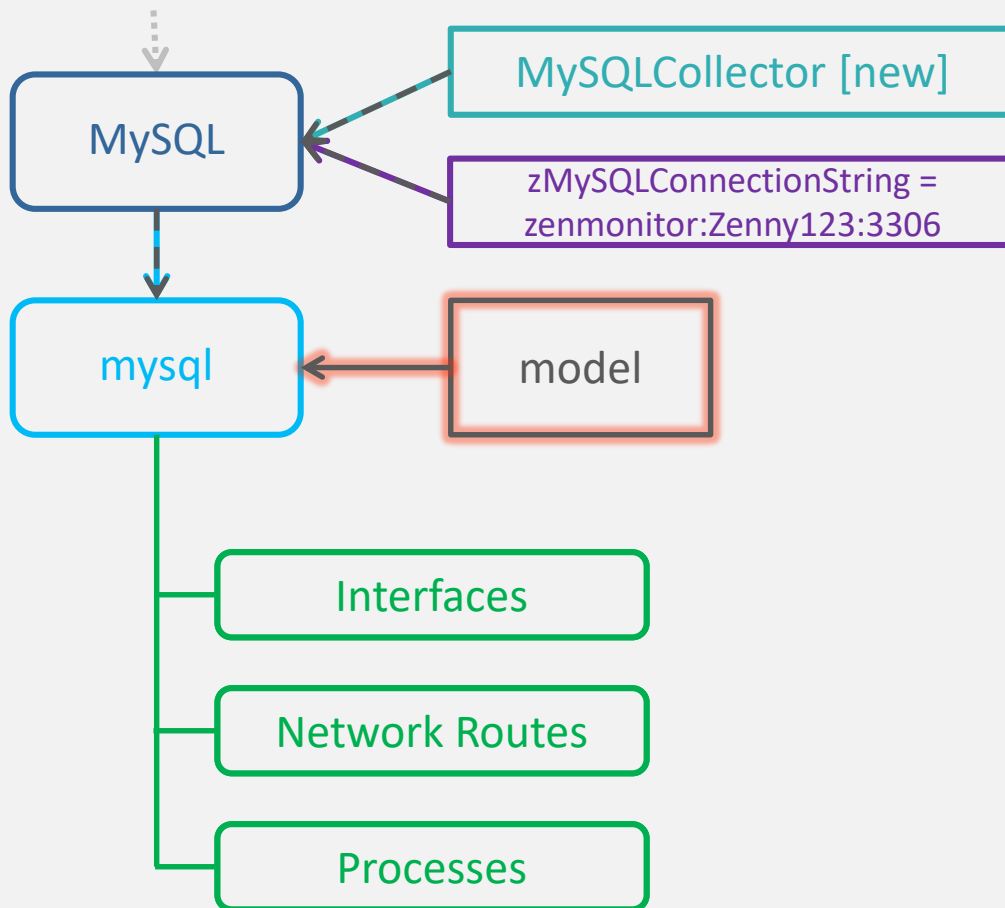
----> Inheritance

Device

Configuring Apache Monitoring

- ☑ Enable extended status in your Apache server
- ☑ Bind the **Apache** template to the WordPress device class
- ☐ Verify that the four **Apache** graphs are present for the **wordpress.hypothetical.loc** device
- 📖 See <https://www.zenoss.com/product/zenpacks/apache-http-server> for more information

Adding MySQL Monitoring



KEY

Device Class

Modeler Plugin

Configuration Property

----> Inheritance

Device

Component

Configuring MySQL Monitoring

- ❑ Verify that the **MySQL Servers** and **MySQL Databases** components are present
- 📖 See <https://www.zenoss.com/product/zenpacks/mysql-monitor> for more information

Impact Dynamic Services

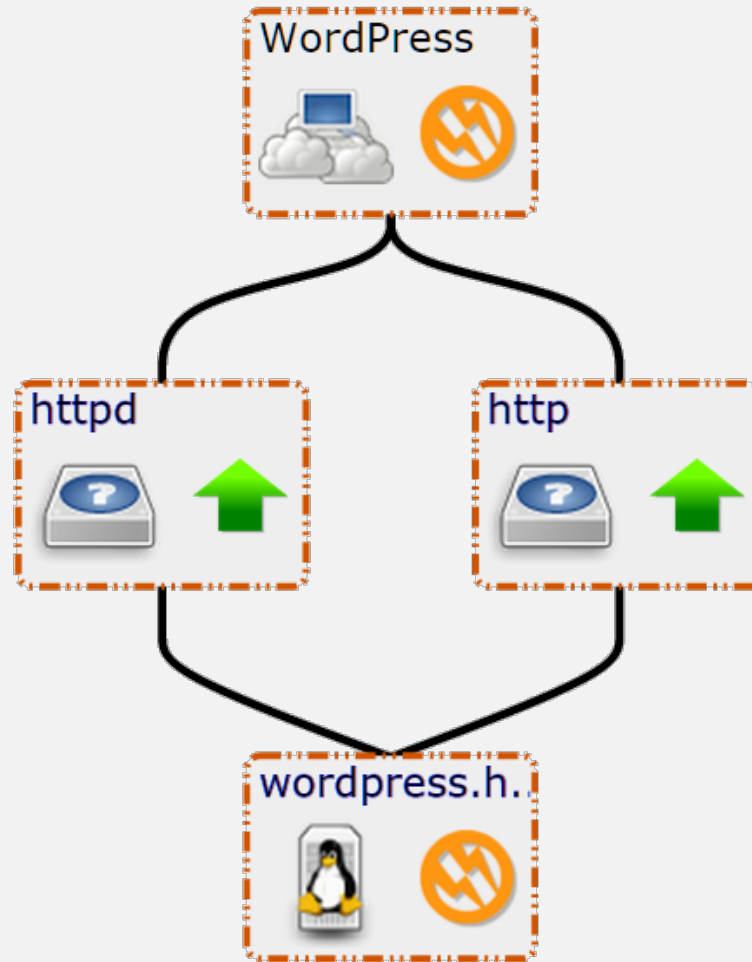
A Dynamic Service is a collection of elements:

- Devices
- Device Components
- **Other Dynamic Services**
- Logical Nodes

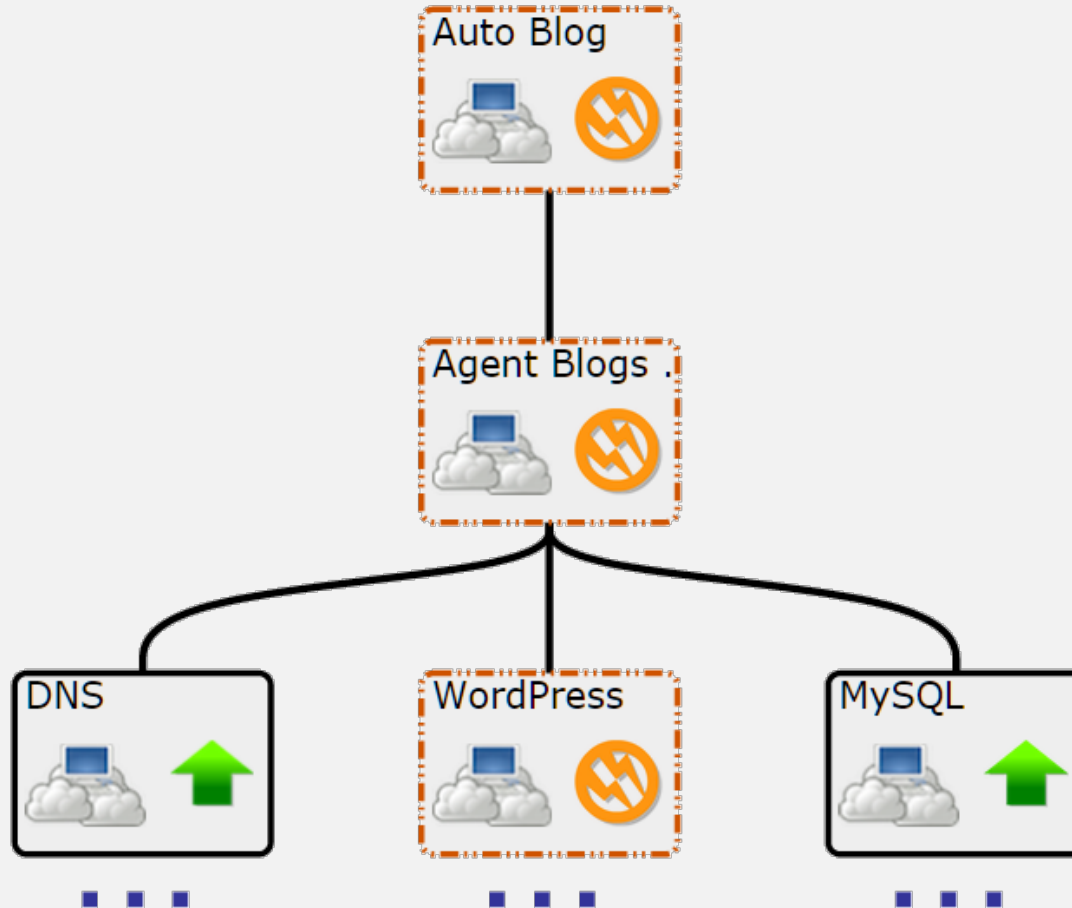
Exercise #7: Create Top-Level Services

- ❑ Create a new top-level service organizer named **Dashboards**
- ❑ Underneath the new organizer, create two new services: **Auto Blog**, **Home Blog**
- ❑ Add the **DNS**, **WordPress**, and **MySQL** services to the **Agent Blogs - Common** service
- ❑ Add the **Agent Blogs - Common** service to the **Auto Blog** and **Home Blog** services
- ❑ Create a new event to simulate the **CPU over 90 percent** threshold in the Apache monitoring template and observe the results

Exercise #7: WordPress Service

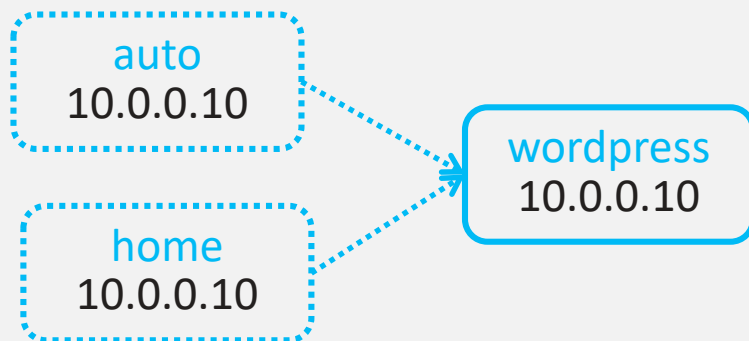


Exercise #7: Auto Blog Service



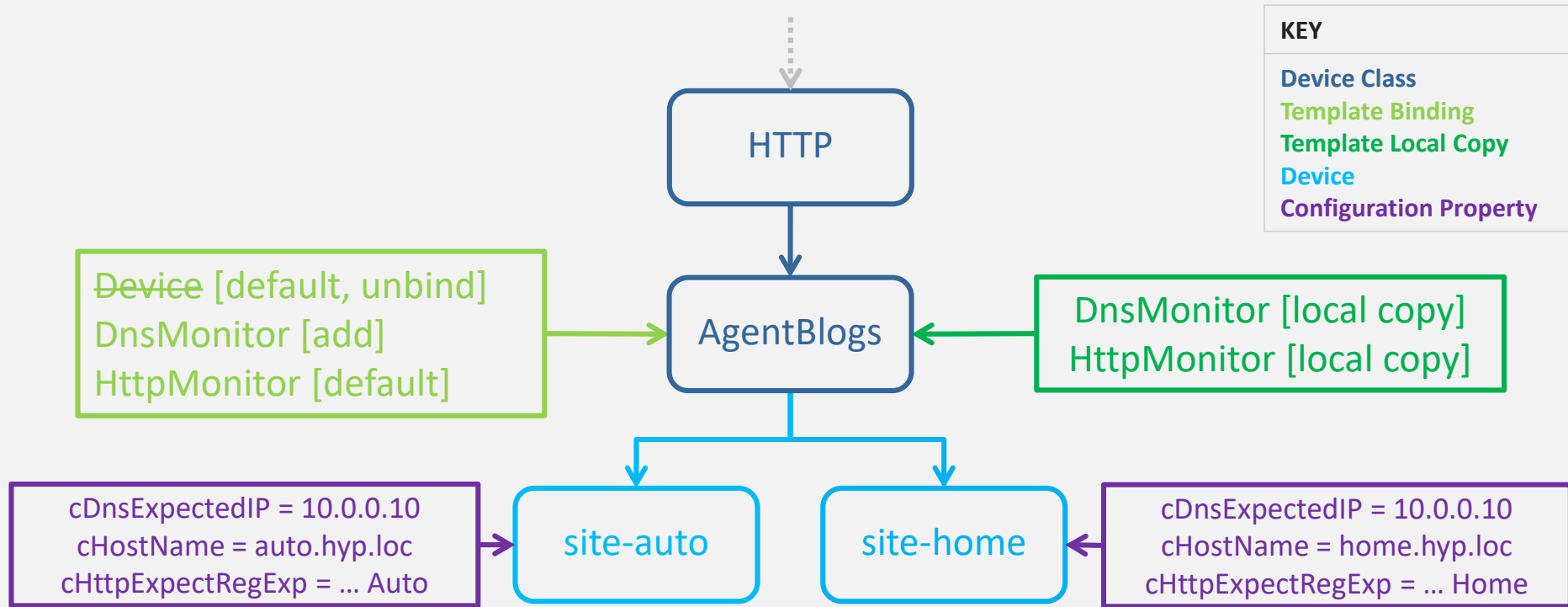
Virtual Web Host Monitoring

- Zenoss does not allow multiple devices to share the same IP address
- Challenge: Monitoring “virtual” web hosts that resolve to the same IP address



- Solution: Create placeholder devices with host names that do not resolve in DNS: for example, **site-auto.hypothetical.loc** and **site-home.hypothetical.loc**

Virtual Web Host Monitoring



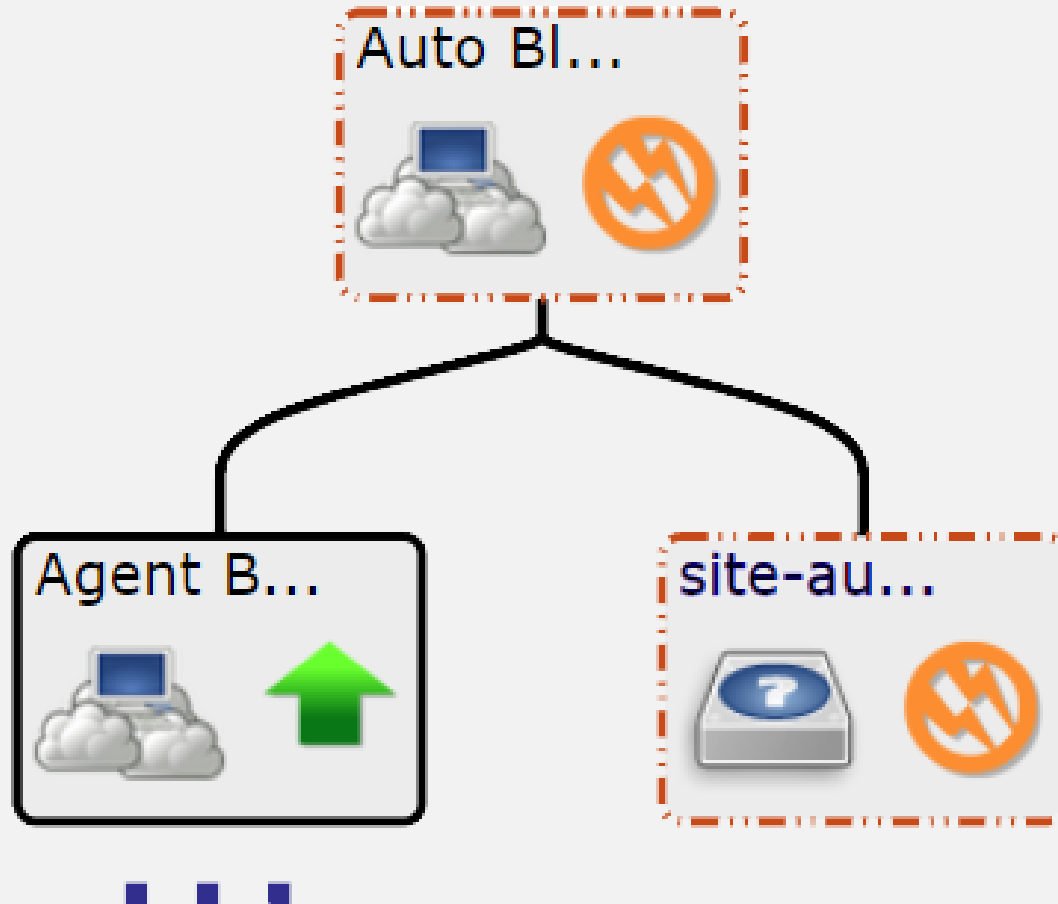
Exercise #8a: Update the Blog Services

- ❑ Add the **site-auto.hypothetical.loc** device to the **/Dashboards/Auto Blog** service
- ❑ Add the **site-home.hypothetical.loc** device to the **/Dashboards/Home Blog** service

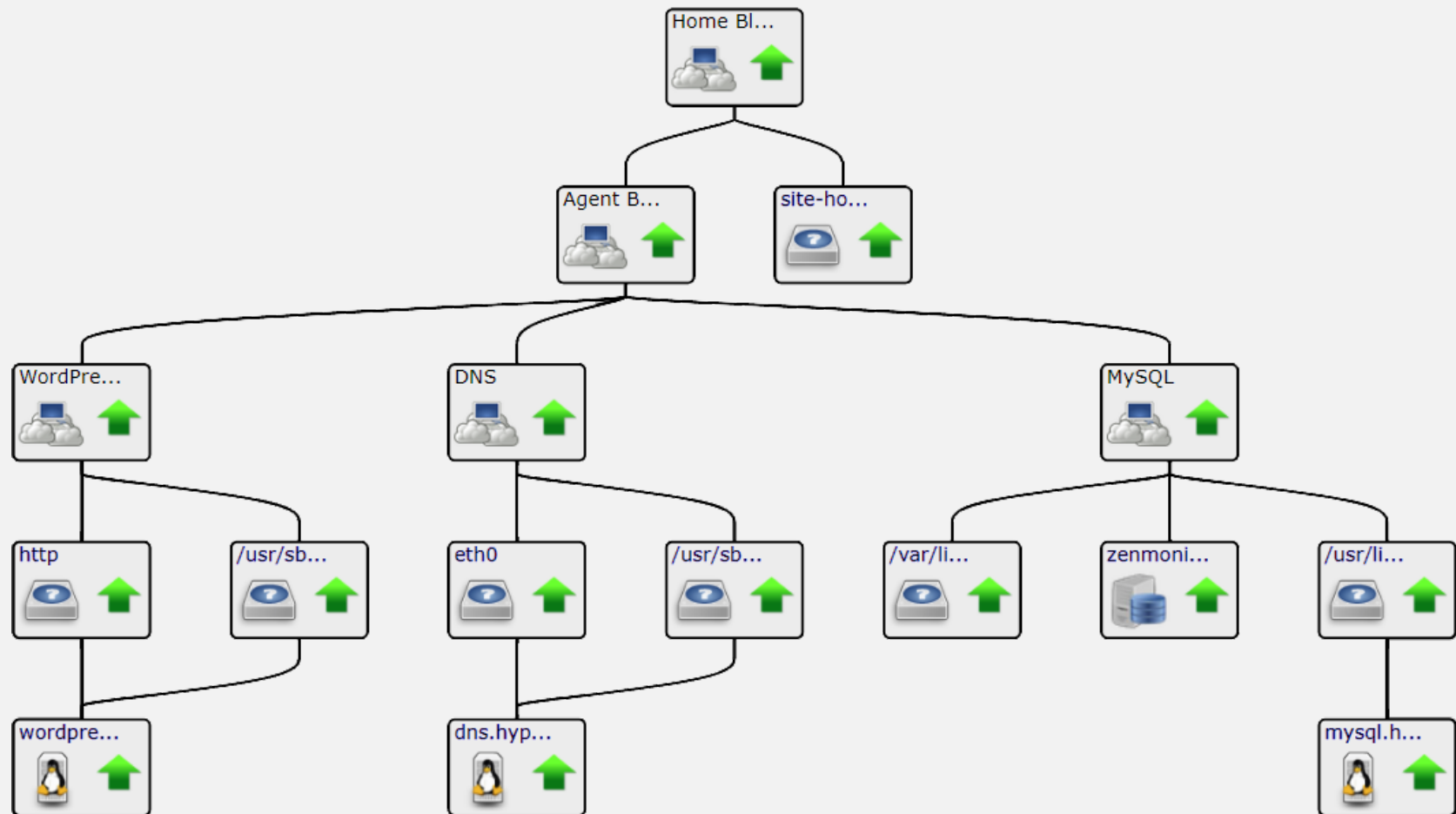
Exercise #8b: Generate a Test Event

- ❑ Close any open events
- ❑ Create a **/Status/HTTP** or **/Status/DNS** event on one of the new devices and observe the results

Exercise #8: Auto Blog Service (Collapsed)



Exercise #8: Home Blog Service (Expanded)



Synthetic Web Transactions Using Twill

- Twill is a scripting language designed for testing web sites
- It allows you to simulate a browser session: follow links, fill out and submit forms, check for specific content, etc.
- This approach is often referred to as monitoring with *synthetic web transactions*
- Monitoring a web site with Twill often starts with a trial and error process using the interactive Twill shell

 The Twill documentation is available at <http://twill.idyll.org>

Root Cause Analysis

Go to a service's Impact Events view to get a list of open service events and their impacting events with a root cause confidence ranking

The screenshot shows the Zenoss MySQL Impact Events view. At the top, there's a header for 'MySQL /Agent Blogs - Common' with a 'SERVICE HEALTH' indicator showing 'A' (Alerts) and 'P' (Problems). A dropdown menu shows 'Impact Events' and a 'Refresh' button is present.

Below the header, there's a table with the following columns: Status, Severity, Service, Event Class, Summary, First Seen, Last Seen, and Count. The table displays one row for the MySQL service, indicating a critical status (red exclamation mark) and a count of 52.

Below this table, there's a section for 'Impact Events' with a 'Show historical events' checkbox. A table of impact events is shown, with columns: Confidence, Status, Severity, Resource, Component, Event Class, Summary, First Seen, Last Seen, and Count. The table displays three rows of impact events, each with a confidence ranking (55, 28, and 19) and a critical status (red exclamation mark).

The confidence ranking is highlighted with a red box in the original image. The table also includes a 'Select' dropdown and a 'DISPLAYING 1 - 3 of 3 ROWS' indicator.

Status	Severity	Service	Event Class	Summary	First Seen	Last Seen	Count
!!		MySQL	/Service/State/Availability	Service Age...	2013-10-09 09:00:28	2013-10-09 09:47:49	52

Confidence	Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
55	!!		vmhost12.hy...		/Status/Pi...	v...	2013-10-09 09:00:27	2013-10-09 09:00:27	
28	!!		vmhost12.hy...	VM-mysql...	/Status/Pi...	V...	2013-10-09 09:46:43	2013-10-09 09:46:43	
19	!!		mysql.hypoth...		/Status/Pi...	m...	2013-10-09 09:47:49	2013-10-09 09:47:49	

Impact Dynamic Services

A Dynamic Service is a collection of elements:

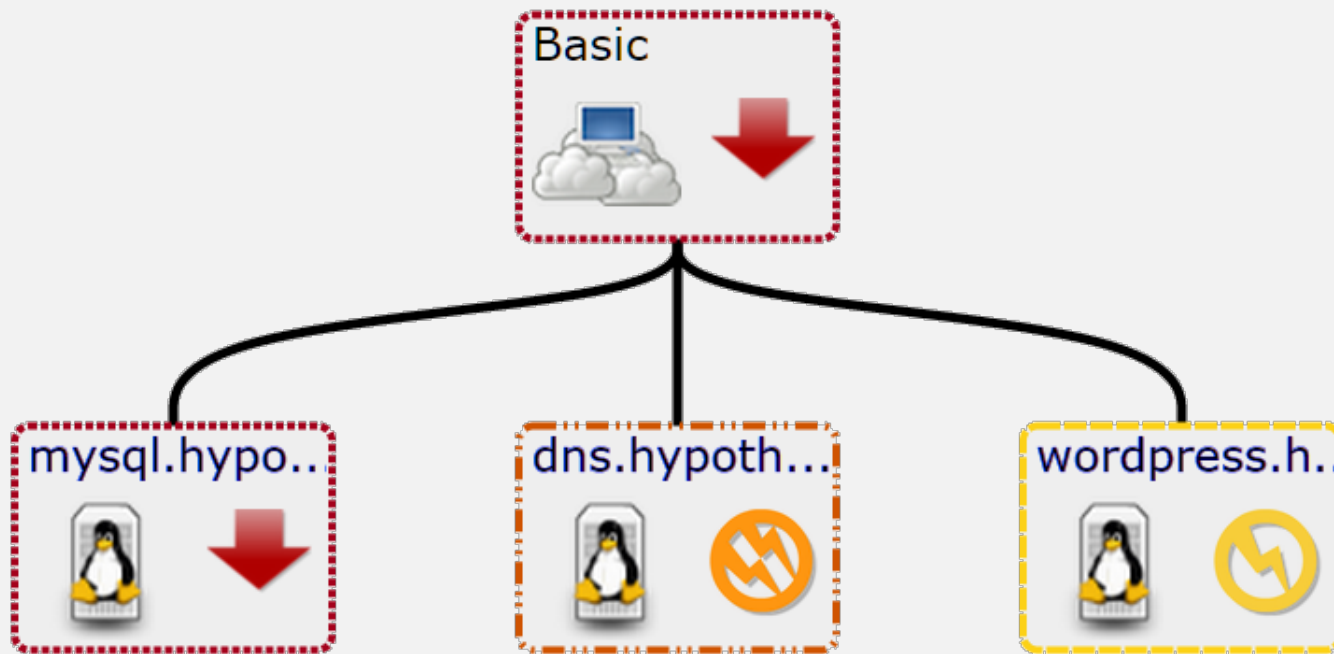
- Devices
- Device Components
- Other Dynamic Services
- **Logical Nodes**

Logical Nodes

- Represent groups of events meeting a set of criteria (similar to a trigger)
- Consist of an event filter and state provider
- State provider maps event severity to Impact state
- Can only be added to services

Default Service Policy

The state of impacting elements are rolled up into the state of the impacted element; by default, the most severe resulting state takes precedence



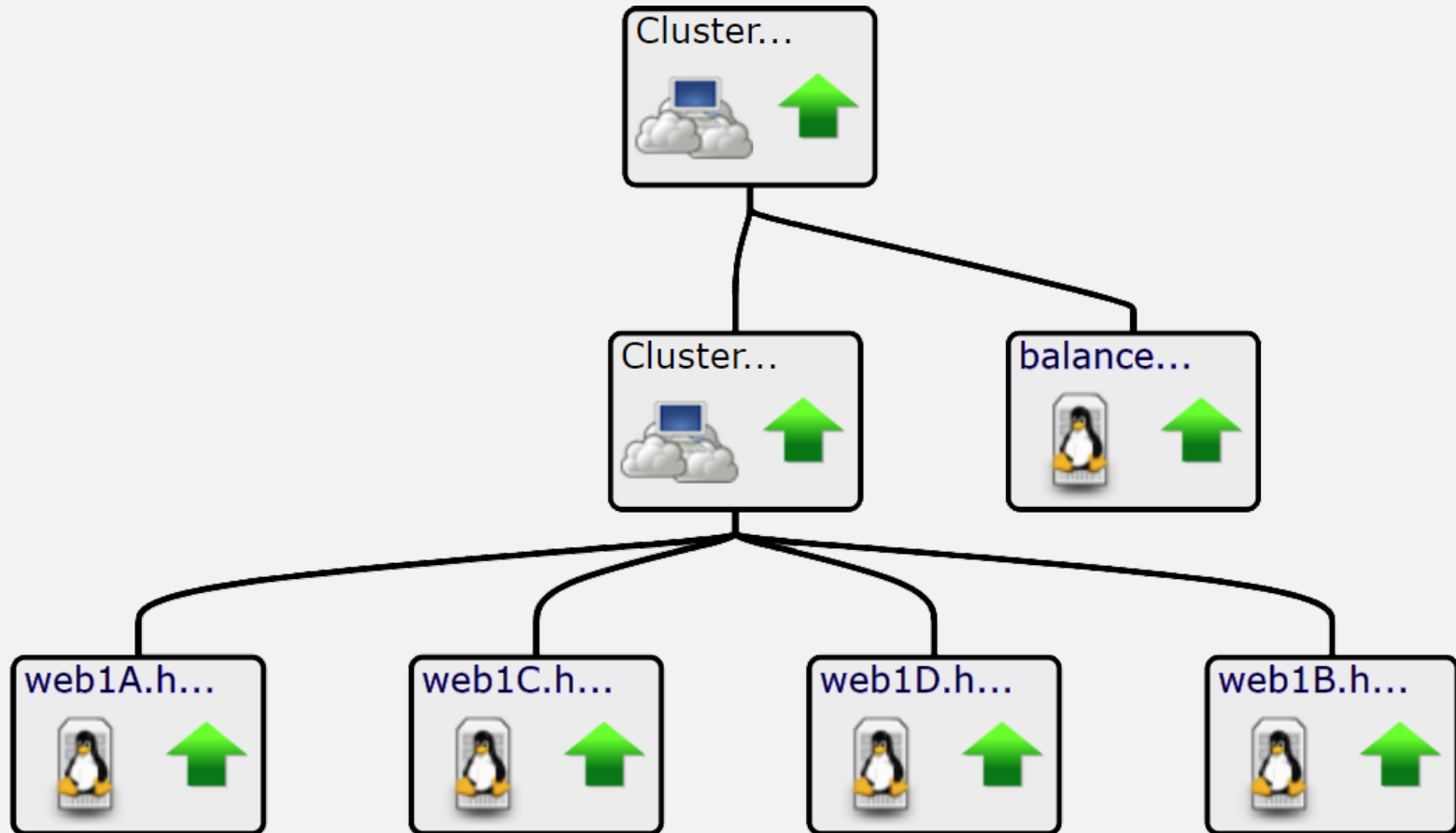
Service Policies

- Collection of State Triggers: the conditions under which the state of a service or element changes
- Applied to a service or any of its elements
- Assist in root cause identification
- Provide multiple availability and performance states
- Provide event storm filtering, roll-up, and windowing
- Can be applied globally or contextually
- State of impacting elements rolled up into state of impacted element, worst resulting state takes precedence

Exercise #9a: Create the Web Cluster Services

- ❑ Create a new top-level service organizer named **Web Clusters**
- ❑ Underneath the new organizer, create two new services:
Cluster 1 and **Cluster 1 Servers**
- ❑ Add the four web server demo devices (**web1A – web1D**) to the **Cluster 1 Servers** service
- ❑ Add the **balancer1** device and the **Cluster 1 Servers** service to the **Cluster 1** service

Exercise #9a: Cluster 1 Service



Exercise #9b: Configure the Service Policy

- ❑ Go to the **Impact View** for the **Cluster 1 Servers** service
- ❑ Right click on the **Cluster 1 Servers** node and select **Edit Impact Policies**
- ❑ **Add a Global Policy** with these state triggers:

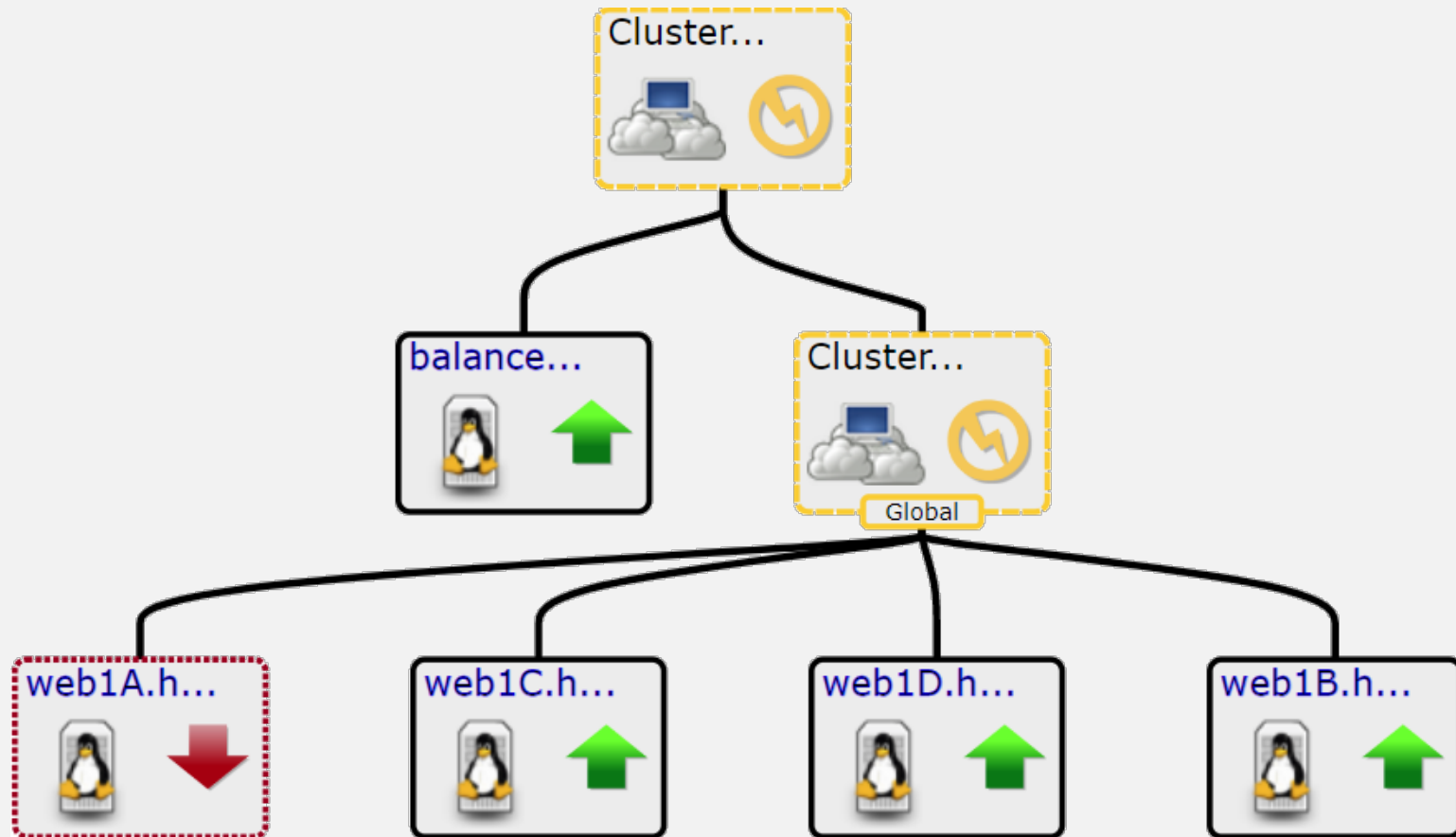
My state will be	If	%	Of type	Are	
ATRISK	>=	25	☒	Any	DOWN
DEGRADED	>=	50	☒	Any	DOWN
DOWN	>=	100	☒	Any	DOWN

Exercise #9c: Test the Service Policy

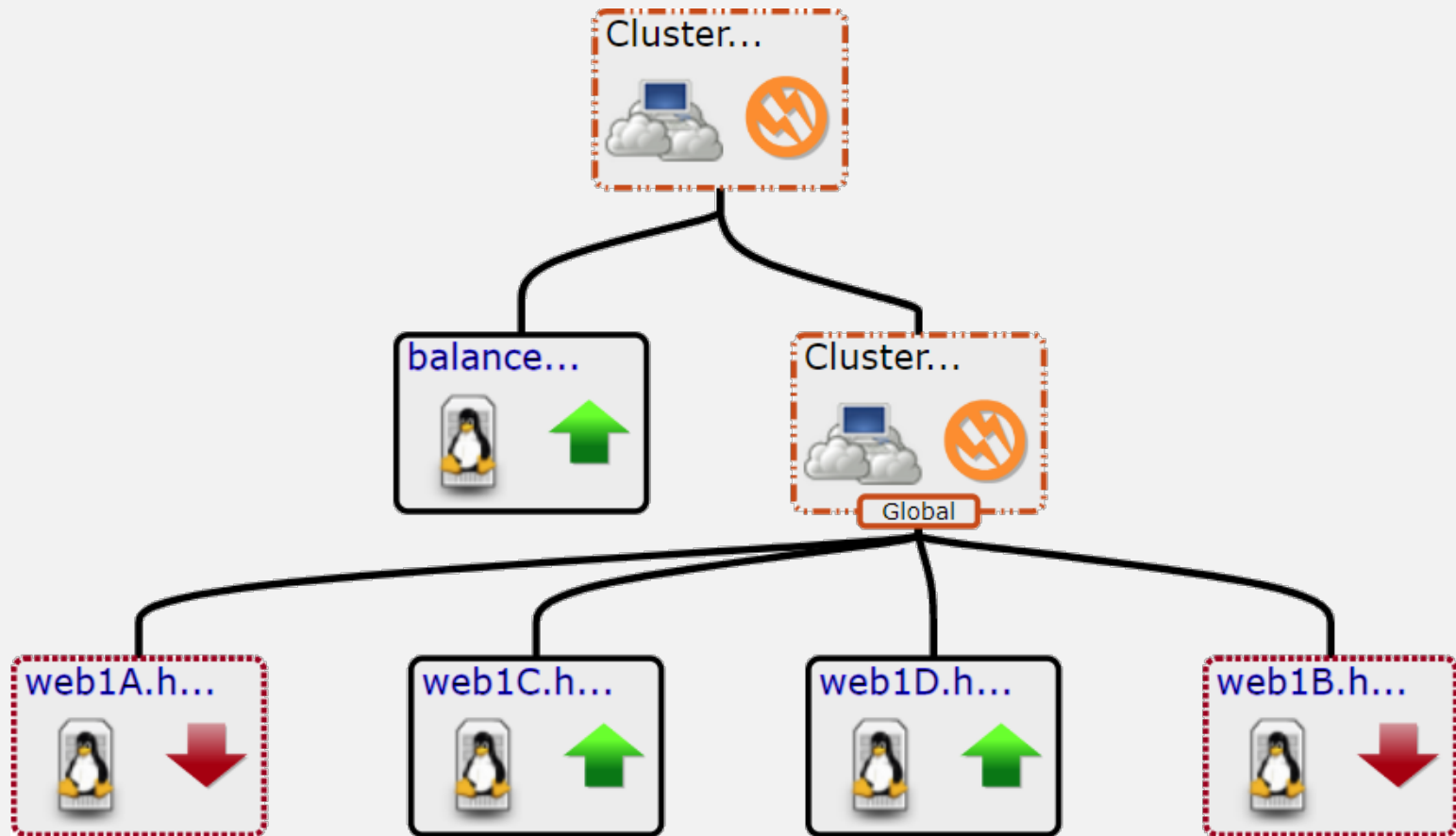
- ❑ Create a critical event as shown for each of the four web server devices in turn, and observe the results

Summary	Host Down
Device	web1X.hypothetical.loc where X = A, B, C, or D
Component	[leave blank]
Severity	Critical
Event Class Key	[leave blank]
Event Class	/Status/Ping
Collector	localhost

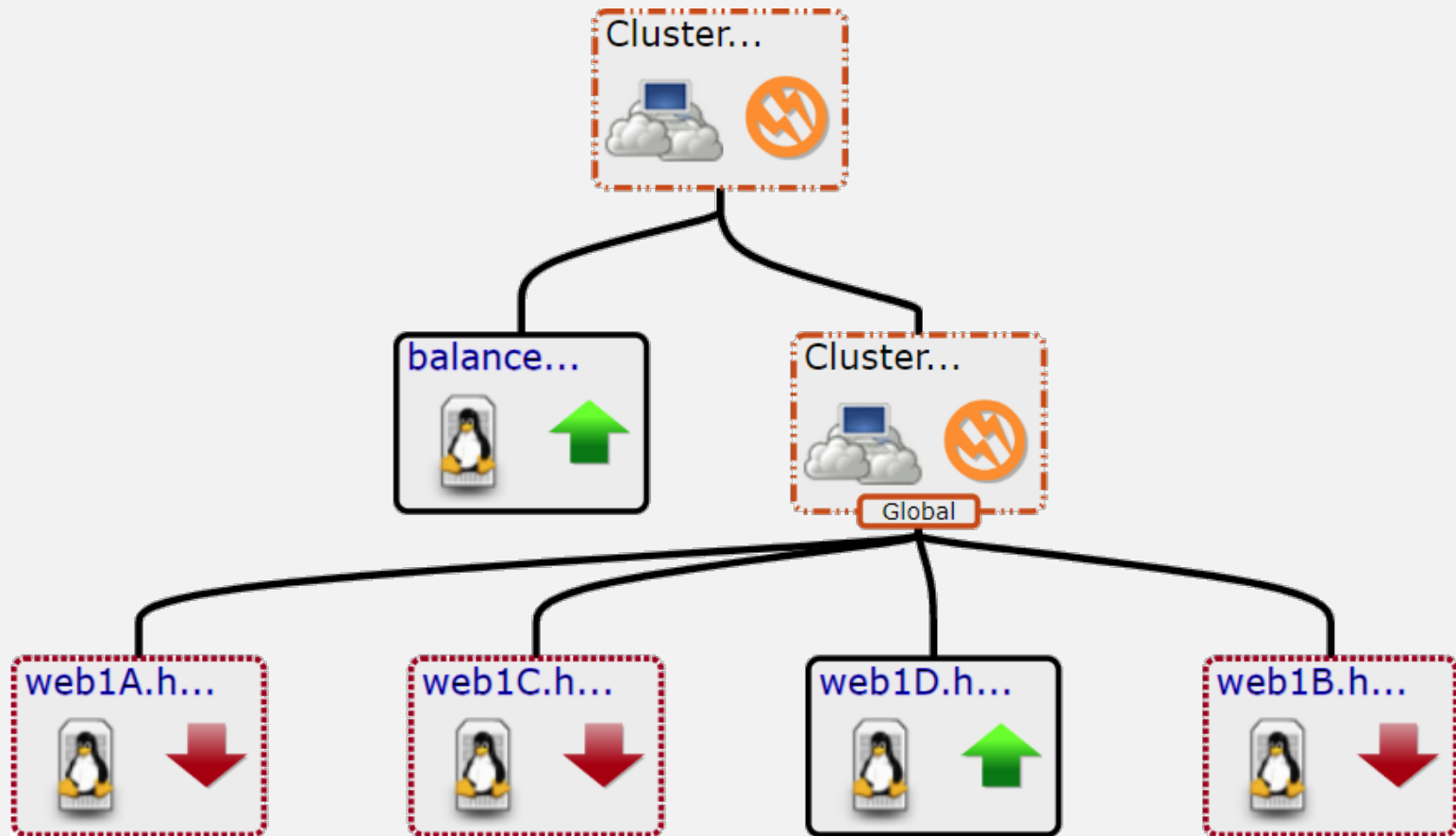
Exercise #9: One (1) Web Server Down



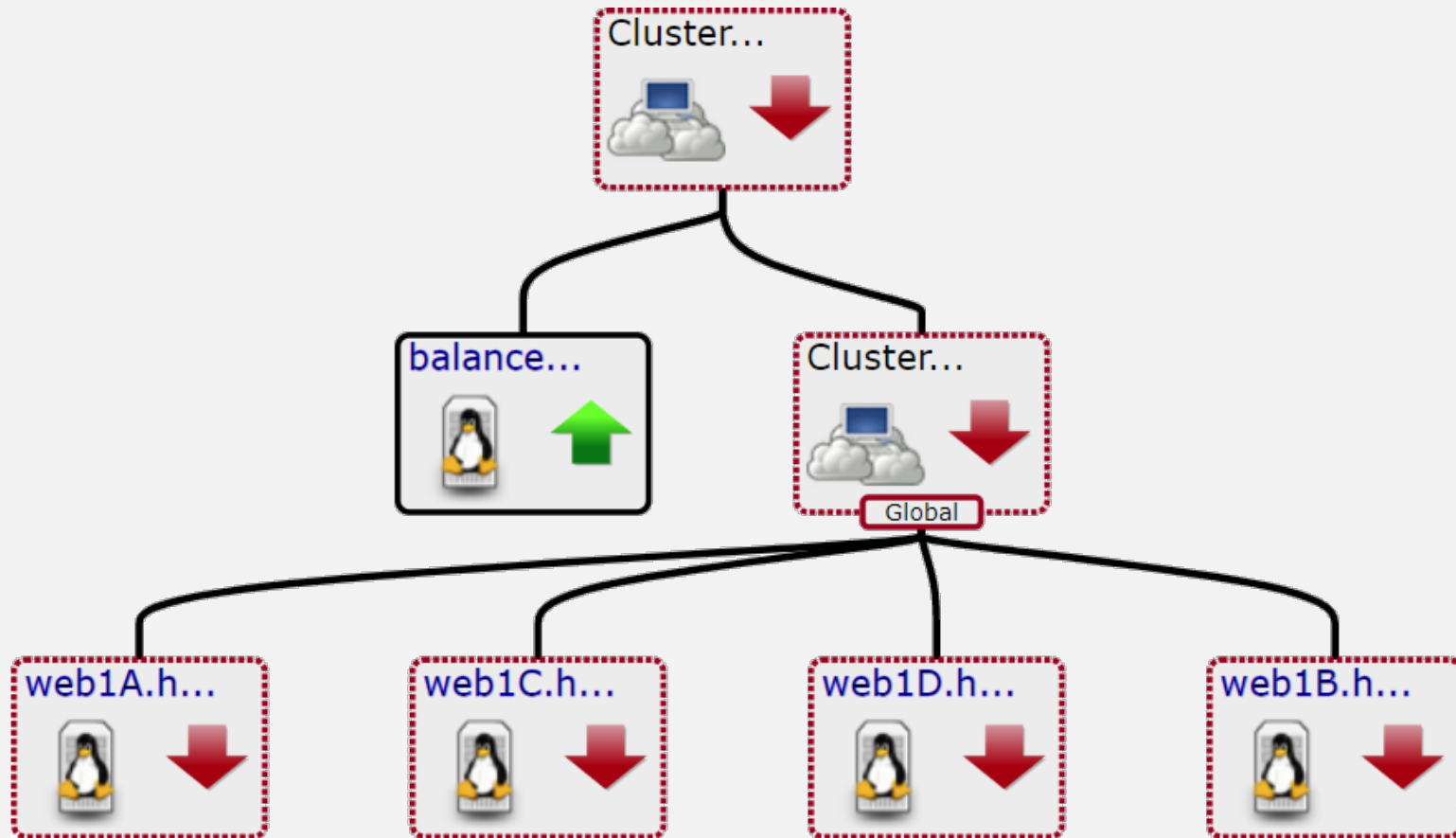
Exercise #9: Two (2) Web Servers Down



Exercise #9: Three (3) Web Servers Down



Exercise #9: Four (4) Web Servers Down



Impact Services Dashboard Portlet

- To add the Impact portlet to your Zenoss Dashboard, click on the **Add portlet** link and select **Impact Services** from the pop-up menu
- You can select the service class (organizer) to be displayed



The screenshot shows a portlet titled "Impact Services" with a settings gear icon in the top right corner. Below the title is a table with two columns: "Name" and "Health". The table lists six services, each with a link in the "Name" column and a health status in the "Health" column. The health status is represented by a colored box containing the letters "A" and "P".

Name	Health
Email	A P
agent.hypothetical.loc	A P
blogs.hypothetical.loc	A P
claims.hypothetical.loc	A P
hypothetical.loc	A P
store.hypothetical.loc	A P

Triggers and Notifications

- Triggers can test for service event classes:

Edit Trigger - Example Service Trigger

Trigger | Users

Name:

Enabled: ☒

Rule: all of the following rules:

starts with

- New notification action: Send SNMP Trap using Service Impact MIB
- Impact-specific notification fields (context)

Class Survey

- We value your feedback!
- Please take a few minutes now to complete the class survey at:
<https://www.surveymonkey.com/s/HXV35KK>
- Your responses will be anonymous unless you choose to leave your name

THANK YOU!