

Data Collected by Discovery

ServiceNow

Overview

Data Collected by Discovery



Note: This article applies to Fuji. For more current information, see *Discovery* ^[1] at <http://docs.servicenow.com> The ServiceNow Wiki is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

General Data

Discovery collects the following general data in a network.

Computers, Clusters, and Virtual Machines:

- AIX Computers
- ESX Servers
- HPUNIX Computers
- Hyper-V
- Linux Computers
- Linux Kernel-based Virtual Machines (KVM)
- Mac Computers
- Netware Computers
- Solaris Computers
- Solaris Zones
- Virtual Machines
- VMware vCenter
- Windows Computers

Hardware Devices:

- Dell DRAC
- Load Balancers and Load Balancer Proxy Software
- Network Printers
- Routers
- Storage Devices
- Switches
- Uninterruptible Power Supplies (UPS)

Web and Database Servers:

- Apache Web Servers
- General Software Packages
- HBase on UNIX
- JBoss Servers
- Microsoft IIS Servers
- Microsoft SQL Servers
- MongoDB Instances
- MySQL Servers
- NGINX Web Servers
- Oracle Databases
- PostgreSQL Instances
- Tomcat Servers
- WebLogic Application Servers
- WebSphere Servers

Other Technologies:

- Amazon Web Services (AWS) Cloud
 - Connections
 - IP Networks
 - Puppet Automation Software
 - Relationships
 - Services/Daemons
-

Enhancements

Fuji

The following probe types are included with the Fuji release. Within each type is a bundle consisting of one or more probes. To view the description of a probe, click a link within a bundle.

Database instances:

- MySQL on Windows
- Oracle instance on Windows
- PostgreSQL
- MongoDB on UNIX and Windows
- HBase on Linux

Web application servers:

- JBoss
- Weblogic
- WebSphere

Load balancer applications for UNIX:

- Apache JK module
- Apache proxy module
- HAProxy
- NGINX

Configuration automation for UNIX:

- Puppet Masters

Server repair tools:

- SNMP DRAC for Dell

Eureka

- Allows discovery of storage device relationships.
- Automatically validates and updates relationships for vCenter CIs.

Dublin

- Discovery collects data on F5 BIG-IP load balancers.
- Discovery collects data on Linux Kernel-based Virtual Machines (KVM).
- The Running Processes [cmdb_running_process] table contains new fields to track the list of TCP ports a process listens on and connects to.
- A new table TCP Connections [cmdb_tcp] replaces the cmdb_tcp_connection and cmdb_tcp_half tables.
- The Application Instance [cmdb_ci_appl] table contains new fields to track the classifier that created the instance record and the running process that matched the classifier.
- For data collected on Linux computers, the **OS Version** field displays:
 - The Linux distribution version if the CI is running a Red Hat, Fedora, CentOS, or SUSE distribution of Linux.
 - The kernel version if the CI is running any other GNU or Linux version.

References

- [1] <https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/reference/r-discovery.html>
-

Computers, Clusters, and Virtual Machines

Windows

Overview

Discovery can classify and discover Windows servers and workstations that use the following operating systems:

- Windows NT Server
- Windows 2000 Server
- Windows 2003 Server
- Windows 2008 Server†
- Windows 2012 Server
- Windows XP
- Windows Vista
- Windows 7
- Windows 8

You may need to set Windows permissions on the systems you want to discover.

† For fiber channel discovery on a Windows 2008 host, the Microsoft Fibre Channel Information Tool (*fcinfo.exe*) must be installed on that machine. The fcinfo executable should be available on the environment path. The fcinfo tool is available for download at <http://www.microsoft.com>.

Data Collected

Discovery stores information about Windows servers and workstations in the following tables and fields.

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_computer	os	wmi
OS version	cmdb_ci_computer	os_version	wmi
OS service pack	cmdb_ci_computer	os_service_pack	wmi
Name	cmdb_ci_win_server	name	DNS, NBT
Hostname	cmdb_ci_win_server	host_name	DNS, NBT
DNS domain	cmdb_ci_win_server	dns_domain	DNS
OS domain	cmdb_ci_computer	os_domain	NBT
Assigned to	cmdb_ci_win_server	assigned_to	wmi
Department	cmdb_ci_win_server	department	Internal (User)
Short description	cmdb_ci_win_server	short_description	wmi
Manufacturer	cmdb_ci_win_server	manufacturer	wmi
Serial number	cmdb_ci_win_server	serial_number	wmi
CPU name	cmdb_ci_computer	cpu_name	wmi
CPU manufacturer	cmdb_ci_computer	cpu_manufacturer	wmi

CPU speed (MHz)	cmdb_ci_computer	cpu_speed	wmi
CPU count *	cmdb_ci_computer	cpu_count	wmi
CPU core count *	cmdb_ci_computer	cpu_core_count	wmi
CPU core thread * (Calgary release)	cmdb_ci_computer	cpu_core_thread	wmi
Model ID	cmdb_ci	model_id	wmi
RAM (MB)	cmdb_ci_computer	ram	wmi
Disk space (GB)	cmdb_ci_computer	disk_space	wmi
Type	cmdb_ci_disk	type	wmi
Description	cmdb_ci_disk	short_description	wmi
Disk space (GB)	cmdb_ci_disk	disk_space	wmi
Free space (GB)	cmdb_ci_disk	free_space	wmi
Name	cmdb_ci_disk	name	wmi
Volume serial number	cmdb_ci_disk	volume_serial_number	wmi
Name	cmdb_running_process	name	wmi
Command	cmdb_running_process	command	wmi
Connects to	cmdb_running_process	connects_to	wmi
Listening on	cmdb_running_process	listening_on	wmi
Type	cmdb_running_process	type	wmi
PID	cmdb_running_process	pid	wmi
Parameters	cmdb_running_process	parameters	wmi
Name	cmdb_ci_network_adapter	name	wmi
IP address	cmdb_ci_network_adapter	ip_address	wmi
MAC address	cmdb_ci_network_adapter	mac_address	wmi
Netmask	cmdb_ci_network_adapter	netmask	wmi
DHCP enabled	cmdb_ci_network_adapter	dhcp_enabled	wmi
Vendor	cmdb_ci_network_adapter	vendor	wmi
Default gateway (Calgary release)	cmdb_ci_hardware	default_gateway	wmi

* Core counts and threads per core might not be accurate, due to issues with Microsoft reporting. See [http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373(v=vs.85).aspx) ^[1] for details.

Versions Prior to the Dublin Release

Discovery stores information about Windows servers and workstations in the following tables and fields on versions prior to Dublin.

View tables and fields on versions prior to Dublin

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_win_server	os	wmi
OS version	cmdb_ci_win_server	os_version	wmi
OS service pack	cmdb_ci_win_server	os_service_pack	wmi
Name	cmdb_ci_win_server	name	DNS, NBT
Hostname	cmdb_ci_win_server	host_name	DNS, NBT
DNS domain	cmdb_ci_win_server	dns_domain	DNS
OS domain	cmdb_ci_win_server	os_domain	NBT
Assigned to	cmdb_ci_win_server	assigned_to	wmi
Department	cmdb_ci_win_server	department	Internal (User)
Short description	cmdb_ci_win_server	short_description	wmi
Manufacturer	cmdb_ci_win_server	manufacturer	wmi
Serial number	cmdb_ci_win_server	serial_number	wmi
CPU name	cmdb_ci_computer	cpu_name	wmi
CPU manufacturer	cmdb_ci_computer	cpu_manufacturer	wmi
CPU speed (MHz)	cmdb_ci_computer	cpu_speed	wmi
CPU count *	cmdb_ci_computer	cpu_count	wmi
CPU core count *	cmdb_ci_computer	cpu_core_count	wmi
CPU core thread * (Calgary release)	cmdb_ci_computer	cpu_core_thread	wmi
Model ID	cmdb_ci	model_id	wmi
RAM (MB)	cmdb_ci_computer	ram	wmi
Disk space (GB)	cmdb_ci_computer	disk_space	wmi
Type	cmdb_ci_disk	type	wmi
Description	cmdb_ci_disk	short_description	wmi
Disk space (GB)	cmdb_ci_disk	disk_space	wmi
Free space (GB)	cmdb_ci_disk	free_space	wmi
Name	cmdb_ci_disk	name	wmi
Volume serial number	cmdb_ci_disk	volume_serial_number	wmi
Name	cmdb_running_process	name	wmi
Command	cmdb_running_process	command	wmi
Type	cmdb_running_process	type	wmi
PID	cmdb_running_process	pid	wmi
Parameters	cmdb_running_process	parameters	wmi
Name	cmdb_ci_network_adapter	name	wmi
IP address	cmdb_ci_network_adapter	ip_address	wmi
MAC address	cmdb_ci_network_adapter	mac_address	wmi
Netmask	cmdb_ci_network_adapter	netmask	wmi
DHCP enabled	cmdb_ci_network_adapter	dhcp_enabled	wmi
Vendor	cmdb_ci_network_adapter	vendor	wmi

Default gateway (Calgary release) cmdb_ci_hardware default_gateway wmi

* Core counts and threads per core might not be accurate, due to issues with Microsoft reporting. See [http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373(v=vs.85).aspx) ^[1] for details.

Enhancements

Dublin

- The Running Processes [cmdb_running_process] table contains new fields to track the list of TCP ports a process listens on (listening_on) and connects to (connects_to).

References

[1] [http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373(v=vs.85).aspx)

Solaris

Discoverable Solaris Operating Systems

Discovery can classify and discover Solaris servers and workstations that use the following operating systems:

- Oracle Solaris 10
- Oracle Solaris 11

You must provide SSH credentials for the systems you want to discover.

Tables and Fields

Discovery stores information about Solaris computers in the following tables and fields.

Label	Table	Field Name	Source
Operating System	cmdb_ci_computer	os	uname
Short description	cmdb_ci_solaris_server	short_description	uname
Name	cmdb_ci_solaris_server	name	DNS, NBT
Hostname	cmdb_ci_solaris_server	host_name	DNS, NBT
DNS domain	cmdb_ci_solaris_server	dns_domain	DNS
Start date	cmdb_ci_solaris_server	start_date	uptime
CPU type	cmdb_ci_computer	cpu_type	kstat
CPU speed (MHz)	cmdb_ci_computer	cpu_speed	kstat
CPU count	cmdb_ci_computer	cpu_count	kstat
CPU core count	cmdb_ci_computer	cpu_core_count	kstat
CPU core thread (Calgary release)	cmdb_ci_computer	cpu_core_thread	kstat
Model number	cmdb_ci_solaris_server	model_number	suntime
Model ID	cmdb_ci_solaris_server	model_id	suntime

RAM (MB)	cmdb_ci_computer	ram	prtconf
Disk space (GB)	cmdb_ci_solaris_server	disk_space	iostat
Serial Number	cmdb_ci_solaris_server	serial_number	sneep †
Manufacturer	cmdb_ci_disk	manufacturer	iostat
Model ID	cmdb_ci_disk	model_id	iostat
Volume serial number	cmdb_ci_disk	volume_serial_number	iostat
Disk space (GB)	cmdb_ci_disk	disk_space	iostat
Name	cmdb_ci_disk	name	iostat
Name	cmdb_ci_file_system	name	df
Capacity (MB)	cmdb_ci_file_system	capacity	df
Available Space (MB)	cmdb_ci_file_system	available_space	df
Mount point	cmdb_ci_file_system	mount_point	df
Name	cmdb_ci_patches	name	showrev
Name	cmdb_running_process	name	ps
Command	cmdb_running_process	command	ps
Connects to	cmdb_running_process	connects_to	lsof
Listening on	cmdb_running_process	listening_on	lsof
Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	ifconfig
IP address	cmdb_ci_network_adapter	ip_address	ifconfig
MAC address	cmdb_ci_network_adapter	mac_address	ifconfig
Netmask	cmdb_ci_network_adapter	netmask	ifconfig
Default gateway (Calgary release)	cmdb_ci_hardware	default_gateway	netstat

† **Note:** To discover Fujitsu PRIMEPOWER devices, you must install Oracle SNEEP and run Solaris discovery with root credentials.

Versions Prior to the Dublin Release

Discovery stores information about Solaris computers in the following tables and fields in versions prior to Dublin.

View tables and fields on versions prior to Dublin

Label	Table	Field Name	Source
Operating System	cmdb_ci_solaris_server	os	uname
Short description	cmdb_ci_solaris_server	short_description	uname
Name	cmdb_ci_solaris_server	name	DNS, NBT
Hostname	cmdb_ci_solaris_server	host_name	DNS, NBT
DNS domain	cmdb_ci_solaris_server	dns_domain	DNS
Start date	cmdb_ci_solaris_server	start_date	uptime
CPU type	cmdb_ci_solaris_server	cpu_type	kstat
CPU speed (MHz)	cmdb_ci_solaris_server	cpu_speed	kstat
CPU count	cmdb_ci_solaris_server	cpu_count	kstat
CPU core count	cmdb_ci_computer	cpu_core_count	kstat
CPU core thread (Calgary release)	cmdb_ci_computer	cpu_core_thread	kstat
Model number	cmdb_ci_solaris_server	model_number	suntype
Model ID	cmdb_ci_solaris_server	model_id	suntype
RAM (MB)	cmdb_ci_solaris_server	ram	prtconf
Disk space (GB)	cmdb_ci_solaris_server	disk_space	iostat
Serial Number	cmdb_ci_solaris_server	serial_number	sneep
Manufacturer	cmdb_ci_disk	manufacturer	iostat
Model ID	cmdb_ci_disk	model_id	iostat
Volume serial number	cmdb_ci_disk	volume_serial_number	iostat
Disk space (GB)	cmdb_ci_disk	disk_space	iostat
Name	cmdb_ci_disk	name	iostat
Name	cmdb_ci_file_system	name	df
Capacity (MB)	cmdb_ci_file_system	capacity	df
Available Space (MB)	cmdb_ci_file_system	available_space	df
Mount point	cmdb_ci_file_system	mount_point	df
Name	cmdb_ci_patches	name	showrev
Name	cmdb_running_process	name	ps
Command	cmdb_running_process	command	ps
Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	ifconfig
IP address	cmdb_ci_network_adapter	ip_address	ifconfig
MAC address	cmdb_ci_network_adapter	mac_address	ifconfig
Netmask	cmdb_ci_network_adapter	netmask	ifconfig
Default gateway (Calgary release)	cmdb_ci_hardware	default_gateway	netstat

Enhancements

Dublin

- The Running Processes [cmdb_running_process] table contains new fields to track the list of TCP ports a process listens on (listening_on) and connects to (connects_to).

Linux

Data Collected by the SSH probe or Linux - Identity multiprobes on Linux Computers

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_linux_server	os	uname
OS Version	cmdb_ci_computer	os_version	uname -a or cat /etc/*release
Short description	cmdb_ci_linux_server	short_description	uname
Name	cmdb_ci_linux_server	name	DNS, NBT
Hostname	cmdb_ci_linux_server	host_name	DNS, NBT
DNS domain	cmdb_ci_linux_server	dns_domain	DNS
Start date	cmdb_ci_linux_server	start_date	uptime
Manufacturer	cmdb_ci_computer	manufacturer	dmidecode
Serial number	cmdb_ci_computer	serial_number	dmidecode
CPU type	cmdb_ci_linux_server	cpu_type	/proc/cpuinfo
CPU speed (MHz)	cmdb_ci_linux_server	cpu_speed	/proc/cpuinfo
CPU count	cmdb_ci_linux_server	cpu_count	/proc/cpuinfo
CPU core count	cmdb_ci_computer	cpu_core_count	/proc/cpuinfo
CPU core thread (Calgary release)	cmdb_ci_computer	cpu_core_thread	/proc/cpuinfo
CPU manufacturer	cmdb_ci_linux_server	cpu_manufacturer	/proc/cpuinfo
Model number	cmdb_ci_computer	model_number	dmidecode
Model ID	cmdb_ci_computer	model_id	dmidecode
RAM (MB)	cmdb_ci_linux_server	ram	meminfo
Disk space (GB)	cmdb_ci_linux_server	disk_space	/proc/ide, /proc/scsi, /var/log/dmesg
Type	cmdb_ci_disk	type	/proc/ide, /proc/scsi, /var/log/dmesg
Model ID	cmdb_ci_disk	model_id	/proc/ide, /proc/scsi, /var/log/dmesg
Disk space (GB)	cmdb_ci_disk	disk_space	/proc/ide, /proc/scsi, /var/log/dmesg
Name	cmdb_ci_disk	name	/proc/ide, /proc/scsi, /var/log/dmesg
Name	cmdb_ci_file_system	name	df
Capacity (MB)	cmdb_ci_file_system	capacity	df
Available Space (MB)	cmdb_ci_file_system	available_space	df
Mount point	cmdb_ci_file_system	mount_point	df
Name	cmdb_running_process	name	ps

Command	cmdb_running_process	command	ps
Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	ifconfig
IP address	cmdb_ci_network_adapter	ip_address	ifconfig
MAC address	cmdb_ci_network_adapter	mac_address	ifconfig
Netmask	cmdb_ci_network_adapter	netmask	ifconfig
Default gateway (Calgary release)	cmdb_ci_hardware	default_gateway	route

Linux Kernel-based Virtual Machines (KVM)

Overview

Discovery identifies Linux kernel-based virtual machines (KVM) when the process classifier detects **libvirtd** running on a Linux server (starting with the Dublin release). The classification triggers the creation of a `cmdb_ci_kvm` record, and launches the SSHCommand probes to explore the Linux server with `virsh`, `lvert` utility, and virtual machine configuration data.

Discovery creates a `cmdb_ci_kvm_instance` record for each virtual machine on the server. Discovery matches the `cmdb_ci_kvm_instance` record to a corresponding `cmdb_ci_computer` record using the MAC addresses of installed network adapters.

Tables used by Discovery on Linux KVM

Discovery uses the following tables to store configuration records for kernel-based virtual machines.

Table Name	Extends	Description	Source
<code>cmdb_ci_kvm</code>	<code>cmdb_ci_vm</code>	A hypervisor that manages kernel-based virtual machines (KVMs)	Process classifier detects libvirtd running on Linux servers
<code>cmdb_ci_kvm_vm_instance</code>	<code>cmdb_ci_vm_instance</code>	A virtual machine instance on this hypervisor	<code>virsh list -all</code> and <code>dumpxml</code> command
<code>cmdb_ci_kvm_object</code>	<code>cmdb_ci_vm_object</code>	An object connected to a virtual machine instance	<code><network></code> , <code><storage pool></code> , and <code><storage volume></code> elements from the <code>dumpxml</code> command
<code>cmdb_kvm_device</code>	N/A	A device connected to a virtual machine instance	<code><devices></code> element from the <code>dumpxml</code> command

Data Collected by Discovery on Linux KVM

Discovery finds the following information for kernel-based virtual machines.

Label	Table Name	Field Name	Source
Linux Host	cmdb_ci_kvm	linux_host	Reference to the cmdb_ci_linux_server that is running this virtual machine
Details	cmdb_ci_kvm	details_xml	dumpxml
Object ID	cmdb_ci_kvm_vm_instance	object_id	virsh dumpxml
State	cmdb_ci_kvm_vm_instance	state	virsh list -all
CPUs	cmdb_ci_kvm_vm_instance	cpus	virsh dumpxml
Memory	cmdb_ci_kvm_vm_instance	memory	virsh dumpxml
Disks	cmdb_ci_kvm_vm_instance	disks	virsh dumpxml
Disks size	cmdb_ci_kvm_vm_instance	disks_size	virsh domblkinfo
Network adapters	cmdb_ci_kvm_vm_instance	nics	virsh dumpxml
Name	cmdb_ci_kvm_vm_instance	name	virsh dumpxml
Short description	cmdb_ci_kvm_vm_instance	short_description	virsh desc
Details	cmdb_ci_kvm_object	details_xml	XML element from dumpxml
KVM instance	cmdb_kvm_device	kvm_instance	Reference to cmdb_ci_kvm_instance
Device	cmdb_kvm_device	device	disk, controller, interface, etc.
Type	cmdb_kvm_device	type	depends on device
Details	cmdb_kvm_device	details_xml	XML element from dumpxml

KVM Relationships

Discovery collects the following relationship data.

Relationship	Parent Table	Child Table
Registered On::Has Registered	KVM [cmdb_ci_kvm]	KVM Virtual Machine Instance [cmdb_ci_kvm_vm_instance]
Provided By::Provides	KVM [cmdb_ci_kvm]	Network [cmdb_ci_kvm_network]
Defines resource for::Gets resources from	KVM [cmdb_ci_kvm]	Storage Pool [cmdb_ci_kvm_storage_pool]
Connected By::Connects	KVM Virtual Machine Instance [cmdb_ci_kvm_vm_instance]	Network [cmdb_ci_kvm_network]
Instantiated By::Instantiates	KVM Virtual Machine Instance [cmdb_ci_kvm_vm_instance]	Computer [cmdb_ci_computer]
Virtualized By::Virtualizes	Computer [cmdb_ci_computer]	KVM [cmdb_ci_kvm]
Provides storage for::Stored on	Storage Pool [cmdb_ci_kvm_storage_pool]	KVM Virtual Machine Instance [cmdb_ci_kvm_vm_instance]

HPUX

Data Collected by Discovery on HPUX Computers

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_hpx_server	os	uname
Short description	cmdb_ci_hpx_server	short_description	uname
Name	cmdb_ci_hpx_server	name	DNS, NBT
Hostname	cmdb_ci_hpx_server	host_name	DNS, NBT
DNS domain	cmdb_ci_hpx_server	dns_domain	DNS
Start date	cmdb_ci_hpx_server	start_date	uptime
Manufacturer	cmdb_ci_computer	manufacturer	dmidecode
Serial number	cmdb_ci_hpx_server	serial_number	uname
CPU type	cmdb_ci_hpx_server	cpu_type	cpuinfo
CPU speed (MHz)	cmdb_ci_hpx_server	cpu_speed	adb
CPU count	cmdb_ci_hpx_server	cpu_count	cpuinfo
Model ID	cmdb_ci_hpx_server	model_id	model
RAM (MB)	cmdb_ci_hpx_server	ram	adb
Name	cmdb_ci_file_system	name	df
Capacity (MB)	cmdb_ci_file_system	capacity	df
Available Space (MB)	cmdb_ci_file_system	available_space	df
Mount point	cmdb_ci_file_system	mount_point	df
Name	cmdb_ci_patches	name	swlist
Name	cmdb_running_process	name	ps
Command	cmdb_running_process	command	ps
Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	hifconfig
IP address	cmdb_ci_network_adapter	ip_address	hifconfig
MAC address	cmdb_ci_network_adapter	mac_address	hifconfig
Netmask	cmdb_ci_network_adapter	netmask	hifconfig

AIX

Tables and Fields

Discovery stores information about AIX computers in the following tables and fields.

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_computer	os	uname
OS version	cmdb_ci_computer	os_version	oslevel
OS service pack	cmdb_ci_computer	os_service_pack	oslevel
Short description	cmdb_ci_aix_server	short_description	uname
Name	cmdb_ci_aix_server	name	DNS, NBT
Hostname	cmdb_ci_aix_server	host_name	DNS, NBT
DNS domain	cmdb_ci_aix_server	dns_domain	DNS
Start date	cmdb_ci_aix_server	start_date	uptime
CPU type	cmdb_ci_computer	cpu_type	lsdev, lsattr
CPU speed (MHz)	cmdb_ci_computer	cpu_speed	lsdev, lsattr
CPU count	cmdb_ci_computer	cpu_count	lsdev, lsattr
Manufacturer	cmdb_ci_aix_server	manufacturer	lsattr
Model ID	cmdb_ci_aix_server	model_id	lsattr
RAM (MB)	cmdb_ci_computer	ram	lsdev, lsattr
Name	cmdb_ci_file_system	name	df
Capacity (MB)	cmdb_ci_file_system	capacity	df
Available Space (MB)	cmdb_ci_file_system	available_space	df
Mount point	cmdb_ci_file_system	mount_point	df
Name	cmdb_ci_patches	name	instfix
Name	cmdb_running_process	name	ps
Command	cmdb_running_process	command	ps
Connects to	cmdb_running_process	connects_to	lsof
Listening on	cmdb_running_process	listening_on	lsof
Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	ifconfig, netstat
IP address	cmdb_ci_network_adapter	ip_address	ifconfig, netstat
MAC address	cmdb_ci_network_adapter	mac_address	ifconfig, netstat
Netmask	cmdb_ci_network_adapter	netmask	ifconfig, netstat

Versions Prior to the Dublin Release

Discovery stores information about AIX computers in the following tables and fields in versions prior to Dublin.

View tables and fields on versions prior to Dublin

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_aix_server	os	uname
OS version	cmdb_ci_aix_server	os_version	oslevel
OS service pack	cmdb_ci_aix_server	os_service_pack	oslevel
Short description	cmdb_ci_aix_server	short_description	uname
Name	cmdb_ci_aix_server	name	DNS, NBT
Hostname	cmdb_ci_aix_server	host_name	DNS, NBT
DNS domain	cmdb_ci_aix_server	dns_domain	DNS
Start date	cmdb_ci_aix_server	start_date	uptime
CPU type	cmdb_ci_aix_server	cpu_type	lsdev, lsattr
CPU speed (MHz)	cmdb_ci_aix_server	cpu_speed	lsdev, lsattr
CPU count	cmdb_ci_aix_server	cpu_count	lsdev, lsattr
Manufacturer	cmdb_ci_aix_server	manufacturer	lsattr
Model ID	cmdb_ci_aix_server	model_id	lsattr
RAM (MB)	cmdb_ci_aix_server	ram	lsdev, lsattr
Name	cmdb_ci_file_system	name	df
Capacity (MB)	cmdb_ci_file_system	capacity	df
Available Space (MB)	cmdb_ci_file_system	available_space	df
Mount point	cmdb_ci_file_system	mount_point	df
Name	cmdb_ci_patches	name	instfix
Name	cmdb_running_process	name	ps
Command	cmdb_running_process	command	ps
Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	ifconfig, netstat
IP address	cmdb_ci_network_adapter	ip_address	ifconfig, netstat
MAC address	cmdb_ci_network_adapter	mac_address	ifconfig, netstat
Netmask	cmdb_ci_network_adapter	netmask	ifconfig, netstat

Enhancements

Dublin

- The Running Processes [cmdb_running_process] table contains new fields to track the list of TCP ports a process listens on (listening_on) and connects to (connects_to).

Mac

Tables and Fields

Discovery stores information about Mac (OS/X) computers in the following tables and fields.

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_computer	os	uname
OS Version	cmdb_ci_computer	os_version	system_profiler
OS Service pack	cmdb_ci_computer	os_service_pack	system_profiler
Short description	cmdb_ci_computer	short_description	uname
Name	cmdb_ci_computer	name	DNS, NBT
Hostname	cmdb_ci_computer	host_name	DNS, NBT
DNS domain	cmdb_ci_computer	dns_domain	DNS
Start date	cmdb_ci_computer	start_date	uptime
Manufacturer	cmdb_ci_computer	manufacturer	Assumed to be Apple
Serial number	cmdb_ci_computer	serial_number	system_profiler
CPU type	cmdb_ci_computer	cpu_type	system_profiler
CPU speed (MHz)	cmdb_ci_computer	cpu_speed	system_profiler
CPU count	cmdb_ci_computer	cpu_count	system_profiler
Model ID	cmdb_ci_computer	model_id	system_profiler
RAM (MB)	cmdb_ci_computer	ram	system_profiler
Disk space (GB)	cmdb_ci_comptuer	disk_space	system_profiler
Volume name	cmdb_ci_disk	volume_name	system_profiler
Volume serial number	cmdb_ci_disk	volume_serial_number	system_profiler
Disk space (GB)	cmdb_ci_disk	disk_space	system_profiler
Name	cmdb_ci_disk	name	system_profiler
Device ID	cmdb_ci_disk	device_id	system_profiler
Free space (GB)	cmdb_ci_disk	free_space	system_profiler
File system	cmdb_ci_disk	file_system	system_profiler
Name	cmdb_running_process	name	ps
Command	cmdb_running_process	command	ps
Connects to	cmdb_running_process	connects_to	lsof
Listening on	cmdb_running_process	listening_on	lsof

Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	system_profiler
IP address	cmdb_ci_network_adapter	ip_address	system_profiler
MAC address	cmdb_ci_network_adapter	mac_address	system_profiler
Netmask	cmdb_ci_network_adapter	netmask	system_profiler
MAC manufacturer	cmdb_ci_network_adapter	mac_mfr	Assumed to be Apple
DHCP enabled	cmdb_ci_network_adapter	dhcp_enabled	system_profiler

Versions Prior to the Dublin Release

Discovery stores information about Mac (OS/X) computers in the following tables and fields on versions prior to Dublin.

View tables and fields on versions prior to Dublin

Label	Table Name	Field Name	Source
Operating System	cmdb_ci_computer	os	uname
OS Version	cmdb_ci_computer	os_version	system_profiler
OS Service pack	cmdb_ci_computer	os_service_pack	system_profiler
Short description	cmdb_ci_computer	short_description	uname
Name	cmdb_ci_computer	name	DNS, NBT
Hostname	cmdb_ci_computer	host_name	DNS, NBT
DNS domain	cmdb_ci_computer	dns_domain	DNS
Start date	cmdb_ci_computer	start_date	uptime
Manufacturer	cmdb_ci_computer	manufacturer	Assumed to be Apple
Serial number	cmdb_ci_computer	serial_number	system_profiler
CPU type	cmdb_ci_computer	cpu_type	system_profiler
CPU speed (MHz)	cmdb_ci_computer	cpu_speed	system_profiler
CPU count	cmdb_ci_computer	cpu_count	system_profiler
Model ID	cmdb_ci_computer	model_id	system_profiler
RAM (MB)	cmdb_ci_computer	ram	system_profiler
Disk space (GB)	cmdb_ci_comptuer	disk_space	system_profiler
Volume name	cmdb_ci_disk	volume_name	system_profiler
Volume serial number	cmdb_ci_disk	volume_serial_number	system_profiler
Disk space (GB)	cmdb_ci_disk	disk_space	system_profiler
Name	cmdb_ci_disk	name	system_profiler
Device ID	cmdb_ci_disk	device_id	system_profiler
Free space (GB)	cmdb_ci_disk	free_space	system_profiler
File system	cmdb_ci_disk	file_system	system_profiler

Name	cmdb_running_process	name	ps
Command	cmdb_running_process	command	ps
Type	cmdb_running_process	type	ps
PID	cmdb_running_process	pid	ps
Parameters	cmdb_running_process	parameters	ps
Name	cmdb_ci_network_adapter	name	system_profiler
IP address	cmdb_ci_network_adapter	ip_address	system_profiler
MAC address	cmdb_ci_network_adapter	mac_address	system_profiler
Netmask	cmdb_ci_network_adapter	netmask	system_profiler
MAC manufacturer	cmdb_ci_network_adapter	mac_mfr	Assumed to be Apple
DHCP enabled	cmdb_ci_network_adapter	dhcp_enabled	system_profiler

Enhancements

Dublin

- The Running Processes [cmdb_running_process] table contains new fields to track the list of TCP ports a process listens on (listening_on) and connects to (connects_to).

Solaris Zones

Label	Table Name	Field Name	Source
Version	cmdb_ci_vm_zones	version	zoneadm, zonename
Correlation ID	cmdb_ci_vm_zones	correlation_id	zoneadm, zonename
Name	cmdb_ci_vm_instance	name	zoneadm, zonename
Parent	cmdb_ci_vm_instance	parent	Internal
CMDB CI	cmdb_ci_vm_instance	cmdb_ci	Internal
Correlation ID	cmdb_ci_vm_instance	correlation_id	zoneadm, zonename

Netware

Data Collected by Discovery on Netware

Label	Table Name	Field Name	Source
Name	cmdb_ci_netware_server	name	snmp
Serial number	cmdb_ci_netware_server	serial_number	snmp
OS Version	cmdb_ci_netware_server	os_version	snmp
RAM	cmdb_ci_netware_server	ram	snmp
CPU count	cmdb_ci_netware_server	cpu_count	snmp

ESX Servers




Discovery

Orchestration

Related Topics

- Cloud Provisioning
- Help the Help Desk
- Help the Help Desk Login Script
- ECC Queue
- Useful Related Lists in CI Forms
- Creating a Workflow
- Using Workflow Activities

Get the Book

-  Discovery
-  Data Collected by Discovery
-  Orchestration for VMWare



Note: This article applies to Fuji and earlier releases. For more current information, see *Discovery of ESX Servers* ^[1] at <http://docs.servicenow.com> **The ServiceNow Wiki is no longer being updated. Visit <http://docs.servicenow.com> for the latest product documentation.**

Overview

ServiceNow Discovery can explore the VMware vCenter process running on a Windows host machine and can discover the ESX servers ^[2] that host vCenter. See VMware Component Relationships for a description of the VMware architecture and component relationships.

Required Roles

Users with the itil and asset roles can access ESXi and ESX configuration item (CI) records. To run discovery on vCenter servers, users must have the discovery_admin role.

Credential Requirements

Three sets of credentials are needed to run a complete Discovery of vCenter/ESX servers:

- **Windows credentials:** Allows Discovery to access the Windows host on which the vCenter server runs
- **vCenter credentials:** Allows a vCenter probe to explore a vCenter server
- **VMWare CIM credentials:** Allows Discovery to access the serial numbers of discovered ESX servers

Refer to the VMware knowledge base for the credentials that are required to connect to a vCenter or ESX server.

ESX Server Discovery Components

ESX Server discovery uses the following components to classify and explore virtual machines (starting with the Dublin release):

Component	Name	Description
Classifier	vCenter	Classifies stand-alone vCenter servers.
Probe	VMWare - vCenter	Exploration probe that extracts the list of ESX hosts, resource pools, and virtual machines from the vCenter host. Implementation details for this probe are located in a MID server script include called VMWarevCenterProbe.
Probe	CIM - ESX Chassis Serial Number	Exploration probe that queries each ESX server for its serial number.

When Discovery runs, a classifier called **vCenter**, classifies the process running on a Windows machine and launches the **VMware - vCenter** probe. This probe logs into the vCenter instance with the credentials provided and uses the vCenter API to return information about ESX machines, virtual machines, and resource pools. If using a domain account to access vCenter, specify the domain with the username in the credential record in one of the supported formats such as **Domain\UserName**. In order to discover ESX server serial numbers, Discovery launches the **CIM - ESX Chassis Serial Number** probe against each ESX server. This probe is required because the vCenter API does not provide a way to retrieve the ESX serial numbers.

[Click the plus for versions prior to Dublin](#)

In versions prior to Dublin, Discovery used SSH to classify ESX servers.

Component	Name	Description
Classifier	ESX	Classifies stand-alone vCenter servers.
Multiprobe	UNIX - Classify	Retrieve information to classify the type of UNIX-based CI. Includes these probes: <ul style="list-style-type: none"> • ESX - OS • UNIX - OS
Probe	ESX - OS	Determine if the target is an ESX

vCenter and ESX Data Collected by Discovery

When Discovery detects the vCenter process running on a host machine, it launches exploration probes to gather typical server CI data as well as the following data specific to ESX Servers.

Tables and Fields

Label	Table	Field Name	Source
CPU expandable	ESX Resource Pool [cmdb_ci_esx_resource_pool]	cpu_expendable	VMWare - vCenter probe
CPU limit (MHz)	ESX Resource Pool [cmdb_ci_esx_resource_pool]	cpu_limit_mhz	VMWare - vCenter probe
CPU reserved (MHz)	ESX Resource Pool [cmdb_ci_esx_resource_pool]	cpu_reserved_mhz	VMWare - vCenter probe
CPU shares	ESX Resource Pool [cmdb_ci_esx_resource_pool]	cpu_shares	VMWare - vCenter probe
Full path	ESX Resource Pool [cmdb_ci_esx_resource_pool]	fullpath	VMWare - vCenter probe
Memory expandable	ESX Resource Pool [cmdb_ci_esx_resource_pool]	mem_expandable	VMWare - vCenter probe
Memory limit (MB)	ESX Resource Pool [cmdb_ci_esx_resource_pool]	mem_limit_mb	VMWare - vCenter probe
Memory reserved (MB)	ESX Resource Pool [cmdb_ci_esx_resource_pool]	mem_reserved_mb	VMWare - vCenter probe
Memory shares	ESX Resource Pool [cmdb_ci_esx_resource_pool]	mem_shares	VMWare - vCenter probe
Owner	ESX Resource Pool [cmdb_ci_esx_resource_pool]	owner	VMWare - vCenter probe
Owner Managed Object Reference ID	ESX Resource Pool [cmdb_ci_esx_resource_pool]	owner_morid	VMWare - vCenter probe
URL	ESX Server [cmdb_ci_esx_server]	url	VMWare - vCenter probe
Managed object reference ID	VMware vCenter Server Object [cmdb_ci_vcenter_server_obj]	morid	VMWare - vCenter probe
Serial Number	Serial Number [cmdb_serial_number]	serial_number	CIM - ESX Chassis Serial Number probe

Relationships

Discovery collects the following relationship data for ESX Servers.

Base Class	Relationship	Dependent Class
ESX Resource Pool [cmdb_ci_esx_resource_pool]	Defines resources for	ESX Server [cmdb_ci_esx_server]
ESX Server [cmdb_ci_esx_server]	Gets resources from	ESX Resource Pool [cmdb_ci_esx_resource_pool]
Virtual Machine Instance [cmdb_ci_vm_instance]	Registered on	ESX Server [cmdb_ci_esx_server]
ESX Server [cmdb_ci_esx_server]	Has registered	Virtual Machine Instance [cmdb_ci_vm_instance]
VMware vCenter Instance [cmdb_ci_vcenter]	Manages	ESX Server [cmdb_ci_esx_server]
ESX Server [cmdb_ci_esx_server]	Managed by	VMware vCenter Instance [cmdb_ci_vcenter]

Enhancements

Dublin

- Discovery uses vCenter and CIM probes instead of SSH to explore ESX servers. Discovery will continue to classify an ESX server if it finds the SSH port open, but it will not launch SSH probes to collect additional information.

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/reference/r_DiscoverESXServers.html
- [2] https://docs.servicenow.com/bundle/helsinki-it-operations-management/page/product/vmware-cloud-provisioning/concept/c_ESXServers.html

Network Gear

Routers



Note: This article applies to Fuji. For more current information, see *Data Collected by Discovery on Network Routers and Switches* ^[1] at <http://docs.servicenow.com>. The ServiceNow Wiki is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

Overview

Network routers and switches often have similar capabilities. It is very common for some switches known as *Layer 3 switches* to have IP routing capability. Larger routers with optional modules might accept switching modules. Because of these overlaps and the resulting ambiguity of a particular device's classification, Discovery collects the same data for both routers and switches.

Turning off the Collection of Port Data

In some environments where the network devices contain an extremely large number of ports and each individual port's information is not critical to the overall management of the devices, you may not want to collect the port data for SNMP network devices.

To disable the collection of port data:

1. Navigate to **Discovery > Discovery Definition > Probes**.
2. Go to the **SNMP - Switching** probe.
3. In the **SNMP Fields** tab, find the field named **ports**. Double-click the value in the **Active** column. Change the value to **false**.
4. Click the check mark to save the change.

SNMP Credentials

Discovery explores many kinds of devices, such as switches, routers, and printers, using the SNMP protocol. Credentials for SNMP do not include a user name, just a password, which is the community string. The default read-only community string for many SNMP devices is public, and Discovery will try that automatically. Enter the appropriate SNMP credentials if they differ from the public community string.

System Properties

You can configure many aspects of Discovery to work according to your personal preferences. The following properties apply to network routers and switches. To view the properties, navigate to **Discovery Definition > Properties** unless indicated otherwise.

Property	Description
glide.discovery.L3_mapping	<p>Provides a logical mapping of the TCP/IP layer for network gears, starting in Fuji. This is not Layer 2 mapping,</p> <ul style="list-style-type: none"> • Type: string • Default value: true • Location: System Property [sys_properties] table
glide.discovery.max_concurrent_invocations_per_schedule	<p>Sets a maximum number of scheduled invocations of the same Discovery schedule, starting in Fuji. It prevents a backlog of scheduled runs if Discovery does not finish before the next invocation is scheduled to run. The value is an integer that represents the max number of automated invocations of the same schedule that may proceed at one time. If the limit has been reached, subsequent scheduled invocations are canceled. A value of 0 or any negative number disables this property.</p> <ul style="list-style-type: none"> • Type: integer • Default value: 3 • Location: Discovery Definition > Properties

Router and Switch Data

Table Name	Field and Label Name	Source
cmdb_ci	Serial number [serial_number]	SNMP, various MIBs
cmdb_ci	Start date [start_date]	SNMP, RFC1213 MIB
cmdb_ci	Manufacturer [manufacturer]	SNMP, RFC1213 MIB
cmdb_ci	Model ID [model_id]	SNMP, RFC1213 MIB
cmdb_ci_network_adapter	IP address [ip_address]	SNMP, IP MIB
cmdb_ci_network_adapter	MAC address [mac_address]	SNMP, IF MIB
cmdb_ci_netgear	Can route IP [can_route]	SNMP, IP MIB, BGP MIB
cmdb_ci_netgear	Can switch IP [can_switch]	SNMP, dot1dBridge MIB
cmdb_ci_netgear	Can partition VLANs [can_partitionvlans]	SNMP, dot1dBridge MIB
cmdb_ci_netgear	Can hub [can_hub]	SNMP, IP MIB
dscy_router_interface	Name [name]	SNMP, IP MIB
dscy_router_interface	Type [type]	SNMP, IP MIB
dscy_router_interface	Number [number]	SNMP, IP MIB
dscy_router_interface	IP address [ip_address]	SNMP, IP MIB
dscy_router_interface	MAC address [mac_address]	SNMP, IP MIB
dscy_route_interface	Destination network [dest_ip_network]	SNMP, IP MIB
dscy_route_interface	Type [type]	SNMP, IP MIB
dscy_route_next_hop	Destination network [dest_ip_network]	SNMP, IP MIB
dscy_route_next_hop	Type [type]	SNMP, IP MIB
dscy_route_next_hop	Next hop [next_hop_ip_address]	SNMP, IP MIB
dscy_swch_partition	Base IP address [base_ip_address]	SNMP, dot1dBridge MIB
dscy_swch_partition	Base MAC address [base_mac_address]	SNMP, dot1dBridge MIB
dscy_swch_partition	Base netmask [base_netmask]	SNMP, dot1dBridge MIB

dscy_swch_partition	Type [type]	SNMP, dot1dBridge MIB
dscy_swch_partition	Transparent [transparent]	SNMP, dot1dBridge MIB
dscy_swch_partition	Sourceroute [sourceroute]	SNMP, dot1dBridge MIB
dscy_swch_partition	Name [name]	SNMP, dot1dBridge MIB
dscy_swch_partition	Status [status]	SNMP, dot1dBridge MIB
dscy_swch_partition	Interface number [interface_number]	SNMP, dot1dBridge MIB
dscy_switchport	Type [type]	SNMP, dot1dBridge MIB
dscy_switchport	Status [status]	SNMP, dot1dBridge MIB
dscy_switchport	MAC address [mac_address]	SNMP, dot1dBridge MIB
dscy_switchport	Port number [port_number]	SNMP, dot1dBridge MIB
dscy_switchport	Interface name [interface_name]	SNMP, dot1dBridge MIB
dscy_switchport	Interface number [interface_number]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	MAC address [mac_address]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	MAC manufacturer [mac_mfr]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	Status [status]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	IP address [ip_address]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	Netmask [netmask]	SNMP, dot1dBridge MIB

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/reference/r_DataCollDiscoNWRouteAndSwitch.html

Switches



Note: This article applies to Fuji. For more current information, see *Data Collected by Discovery on Network Routers and Switches* [1] at <http://docs.servicenow.com>. The ServiceNow Wiki is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

Overview

Network routers and switches often have similar capabilities. It is very common for some switches known as *Layer 3 switches* to have IP routing capability. Larger routers with optional modules might accept switching modules. Because of these overlaps and the resulting ambiguity of a particular device's classification, Discovery collects the same data for both routers and switches.

Turning off the Collection of Port Data

In some environments where the network devices contain an extremely large number of ports and each individual port's information is not critical to the overall management of the devices, you may not want to collect the port data for SNMP network devices.

To disable the collection of port data:

1. Navigate to **Discovery > Discovery Definition > Probes**.
2. Go to the **SNMP - Switching** probe.
3. In the **SNMP Fields** tab, find the field named **ports**. Double-click the value in the **Active** column. Change the value to **false**.
4. Click the check mark to save the change.

SNMP Credentials

Discovery explores many kinds of devices, such as switches, routers, and printers, using the SNMP protocol. Credentials for SNMP do not include a user name, just a password, which is the community string. The default read-only community string for many SNMP devices is public, and Discovery will try that automatically. Enter the appropriate SNMP credentials if they differ from the public community string.

System Properties

You can configure many aspects of Discovery to work according to your personal preferences. The following properties apply to network routers and switches. To view the properties, navigate to **Discovery Definition > Properties** unless indicated otherwise.

Property	Description
glide.discovery.L3_mapping	<p>Provides a logical mapping of the TCP/IP layer for network gears, starting in Fuji. This is not Layer 2 mapping,</p> <ul style="list-style-type: none"> • Type: string • Default value: true • Location: System Property [sys_properties] table
glide.discovery.max_concurrent_invocations_per_schedule	<p>Sets a maximum number of scheduled invocations of the same Discovery schedule, starting in Fuji. It prevents a backlog of scheduled runs if Discovery does not finish before the next invocation is scheduled to run. The value is an integer that represents the max number of automated invocations of the same schedule that may proceed at one time. If the limit has been reached, subsequent scheduled invocations are canceled. A value of 0 or any negative number disables this property.</p> <ul style="list-style-type: none"> • Type: integer • Default value: 3 • Location: Discovery Definition > Properties

Router and Switch Data

Table Name	Field and Label Name	Source
cmdb_ci	Serial number [serial_number]	SNMP, various MIBs
cmdb_ci	Start date [start_date]	SNMP, RFC1213 MIB
cmdb_ci	Manufacturer [manufacturer]	SNMP, RFC1213 MIB
cmdb_ci	Model ID [model_id]	SNMP, RFC1213 MIB
cmdb_ci_network_adapter	IP address [ip_address]	SNMP, IP MIB
cmdb_ci_network_adapter	MAC address [mac_address]	SNMP, IF MIB
cmdb_ci_netgear	Can route IP [can_route]	SNMP, IP MIB, BGP MIB
cmdb_ci_netgear	Can switch IP [can_switch]	SNMP, dot1dBridge MIB
cmdb_ci_netgear	Can partition VLANs [can_partitionvlans]	SNMP, dot1dBridge MIB
cmdb_ci_netgear	Can hub [can_hub]	SNMP, IP MIB
dscy_router_interface	Name [name]	SNMP, IP MIB
dscy_router_interface	Type [type]	SNMP, IP MIB
dscy_router_interface	Number [number]	SNMP, IP MIB
dscy_router_interface	IP address [ip_address]	SNMP, IP MIB
dscy_router_interface	MAC address [mac_address]	SNMP, IP MIB
dscy_route_interface	Destination network [dest_ip_network]	SNMP, IP MIB
dscy_route_interface	Type [type]	SNMP, IP MIB
dscy_route_next_hop	Destination network [dest_ip_network]	SNMP, IP MIB
dscy_route_next_hop	Type [type]	SNMP, IP MIB
dscy_route_next_hop	Next hop [next_hop_ip_address]	SNMP, IP MIB
dscy_swch_partition	Base IP address [base_ip_address]	SNMP, dot1dBridge MIB
dscy_swch_partition	Base MAC address [base_mac_address]	SNMP, dot1dBridge MIB
dscy_swch_partition	Base netmask [base_netmask]	SNMP, dot1dBridge MIB

dscy_swch_partition	Type [type]	SNMP, dot1dBridge MIB
dscy_swch_partition	Transparent [transparent]	SNMP, dot1dBridge MIB
dscy_swch_partition	Sourceroute [sourceroute]	SNMP, dot1dBridge MIB
dscy_swch_partition	Name [name]	SNMP, dot1dBridge MIB
dscy_swch_partition	Status [status]	SNMP, dot1dBridge MIB
dscy_swch_partition	Interface number [interface_number]	SNMP, dot1dBridge MIB
dscy_switchport	Type [type]	SNMP, dot1dBridge MIB
dscy_switchport	Status [status]	SNMP, dot1dBridge MIB
dscy_switchport	MAC address [mac_address]	SNMP, dot1dBridge MIB
dscy_switchport	Port number [port_number]	SNMP, dot1dBridge MIB
dscy_switchport	Interface name [interface_name]	SNMP, dot1dBridge MIB
dscy_switchport	Interface number [interface_number]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	MAC address [mac_address]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	MAC manufacturer [mac_mfr]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	Status [status]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	IP address [ip_address]	SNMP, dot1dBridge MIB
dscy_swch_fwd_rule	Netmask [netmask]	SNMP, dot1dBridge MIB

Other Hardware Devices

Network Printers

Data Collected by Discovery on Network Printers

Label	Table Name	Field Name	Source
Serial number	cmdb_ci	serial_number	SNMP, various MIBs
Start date	cmdb_ci	start_date	SNMP, RFC1213 MIB
Manufacturer	cmdb_ci	manufacturer	SNMP, RFC1213 MIB
Model ID	cmdb_ci	model_id	SNMP, RFC1213 MIB
IP address	cmdb_ci_network_adapter	ip_address	SNMP, IP MIB
MAC address	cmdb_ci_network_adapter	mac_address	SNMP, IF MIB
Printer type	cmdb_ci_printer	print_type	SNMP, PRINT MIB
Use count	cmdb_ci_printer	use_count	SNMP, PRINT MIB
Use count units	cmdb_ci_printer	use_units	SNMP, PRINT MIB
Colors	cmdb_ci_printer	colors	SNMP, PRINT MIB
Horizontal resolution	cmdb_ci_printer	horizontal_resolution	SNMP, PRINT MIB
Vertical resolution	cmdb_ci_printer	vertical_resolution	SNMP, PRINT MIB
Resolution units	cmdb_ci_printer	resolution_units	SNMP, PRINT MIB
Description	discovery_printer_supplies	description	SNMP, PRINT MIB
Supply type	discovery_printer_supplies	supply_type	SNMP, PRINT MIB
Supply class	discovery_printer_supplies	supply_class	SNMP, PRINT MIB
Current level	discovery_printer_supplies	current_level	SNMP, PRINT MIB
Max capacity	discovery_printer_supplies	max_capacity	SNMP, PRINT MIB

Uninterruptible Power Supplies (UPSs)

Data Collected by Discovery on Uninterruptible Power Supplies (UPSs)

Label	Table Name	Field Name	Source
Serial number	cmdb_ci	serial_number	SNMP, various MIBs
Start date	cmdb_ci	start_date	SNMP, RFC1213 MIB
Manufacturer	cmdb_ci	manufacturer	SNMP, RFC1213 MIB
Model ID	cmdb_ci	model_id	SNMP, RFC1213 MIB
IP address	cmdb_ci_network_adapter	ip_address	SNMP, IP MIB
MAC address	cmdb_ci_network_adapter	mac_address	SNMP, IF MIB
UPS software version	cmdb_ci_ups	ups_software_version	SNMP, UPS MIB
Agent software version	cmdb_ci_ups	agent_software_version	SNMP, UPS MIB
Attached devices	cmdb_ci_ups	attached_devices	SNMP, UPS MIB
Battery status	cmdb_ci_ups	battery_status	SNMP, UPS MIB
Seconds remaining on battery	cmdb_ci_ups	seconds_on_battery	SNMP, UPS MIB
Estimated minutes remaining on battery	cmdb_ci_ups	est_mins_remaining	SNMP, UPS MIB
Estimated charge % remaining	cmdb_ci_ups	est_charge_remaining	SNMP, UPS MIB
Battery voltage	cmdb_ci_ups	battery_voltage	SNMP, UPS MIB
Battery current	cmdb_ci_ups	battery_current	SNMP, UPS MIB
Battery temperature (C)	cmdb_ci_ups	battery_temperature	SNMP, UPS MIB
Input line bads	cmdb_ci_ups	input_line_bads	SNMP, UPS MIB
Output source	cmdb_ci_ups	output_source	SNMP, UPS MIB
Output frequency	cmdb_ci_ups	output_freq	SNMP, UPS MIB
Bypass frequency	cmdb_ci_ups	bypass_freq	SNMP, UPS MIB
Nominal input voltage	cmdb_ci_ups	nom_input_volt	SNMP, UPS MIB
Nominal input frequency	cmdb_ci_ups	nom_input_freq	SNMP, UPS MIB
Nominal output voltage	cmdb_ci_ups	nom_output_volt	SNMP, UPS MIB
Nominal output frequency	cmdb_ci_ups	nom_output_freq	SNMP, UPS MIB
Rated output VA	cmdb_ci_ups	rated_output_va	SNMP, UPS MIB
Rated output power	cmdb_ci_ups	rated_output_power	SNMP, UPS MIB
Low battery threshold minutes	cmdb_ci_ups	low_battery_threshold_mins	SNMP, UPS MIB
Audible alarm status	cmdb_ci_ups	audible_alarm_status	SNMP, UPS MIB
Low voltage transfer point	cmdb_ci_ups	low_voltage_transfer_point	SNMP, UPS MIB
High voltage transfer point	cmdb_ci_ups	high_voltage_transfer_point	SNMP, UPS MIB
Input index	cmdb_ci_ups_input	input_index	SNMP, UPS MIB
Input frequency (Hz)	cmdb_ci_ups_input	input_freq	SNMP, UPS MIB
Input voltage (RMS VAC)	cmdb_ci_ups_input	input_volt	SNMP, UPS MIB
Input current (RMS AAC)	cmdb_ci_ups_input	input_current	SNMP, UPS MIB

Input power (Watts)	cmdb_ci_ups_input	input_power	SNMP, UPS MIB
Output index	cmdb_ci_ups_output	output_index	SNMP, UPS MIB
Output load (%)	cmdb_ci_ups_output	output_load	SNMP, UPS MIB
Output voltage (RMS VAC)	cmdb_ci_ups_output	output_volt	SNMP, UPS MIB
Output current (RMS AAC)	cmdb_ci_ups_output	output_current	SNMP, UPS MIB
Output power (Watts)	cmdb_ci_ups_output	output_power	SNMP, UPS MIB
Bypass index	cmdb_ci_ups_bypass	bypass_index	SNMP, UPS MIB
Bypass voltage (RMS VAC)	cmdb_ci_ups_bypass	bypass_volt	SNMP, UPS MIB
Bypass current (RMS AAC)	cmdb_ci_ups_bypass	bypass_current	SNMP, UPS MIB
Bypass power (Watts)	cmdb_ci_ups_bypass	bypass_power	SNMP, UPS MIB
Alarm index	cmdb_ci_ups_alarm	alarm_index	SNMP, UPS MIB
Alarm type	cmdb_ci_ups_alarm	alarm_type	SNMP, UPS MIB
Alarm time	cmdb_ci_ups_alarm	alarm_time	SNMP, UPS MIB

Common Information Model (CIM) storage devices

Discovery

Orchestration

Related Topics

- Cloud Provisioning
- Help the Help Desk
- Help the Help Desk Login Script
- ECC Queue
- Useful Related Lists in CI Forms
- Creating a Workflow
- Using Workflow Activities

Get the Book



Discovery



Data Collected by Discovery



Orchestration for VMWare



Note: This article applies to Fuji. For more current information, see *How CIM Discovery Works* ^[1] at <http://docs.servicenow.com>. The Wiki page is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

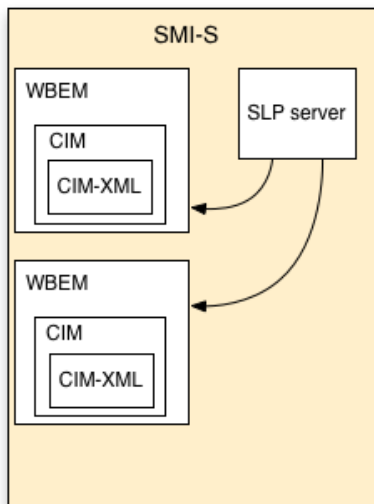
Overview

CIM probes can explore any device based on the Common Information Model (CIM) by querying a CIM server, also referred to as a CIMOM - Common Information Model Object Manager. By default, Discovery uses CIM probes to explore storage systems as well as to get the serial numbers of ESX servers.

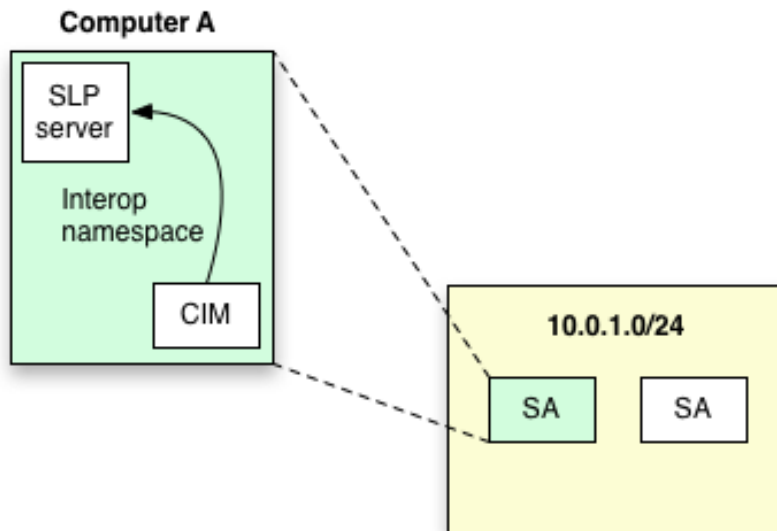
Architecture

The following components are part of CIM:

- **Common Information Model (CIM):** CIM ^[2] allows multiple parties to exchange information about managed elements. CIM represents these managed elements and the management information, while providing the mechanism to actively control and manage the elements.
- **Storage Management Initiative Specification (SMI-S):** SMI-S ^[3] is a standard of use that describes methods for storage discovery on the vendor's side. ServiceNow uses SMI-S to determine how to discover CIM. SMI-S is based on the Common Information Model (CIM) and the Web-Based Enterprise Management (WBEM) standards, which define management functionality via HTTP. The main objective of SMI-S is to enable management of dissimilar storage products. ServiceNow supports SMI-S version 1.5 or higher.

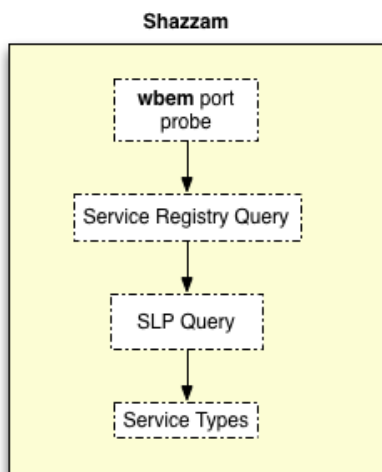


- **Web-Based Enterprise Management (WBEM):** WBEM ^[4] defines a particular implementation of CIM, including protocols for discovering and accessing each CIM implementation.
- **Service Location Protocol (SLP):** SLP ^[5] is an *ad hoc* protocol for retrieving and associating configuration information about CIM servers, such as default paths, capabilities, and the exact interop namespace ^[6]. ServiceNow Discovery retrieves the interop namespace of a CIM server via SLP and passes that information to the CIM Classify probe. SLP, referred to here as the *SLP server*, uses service agents (SA) to gather and disseminate information about a CIM server on a subnet. A subnet can have multiple service agents.



How CIM Discovery Works

1. The Shazzam probe launches the **wbem** port probe as part of network discovery.
2. The wbem port probe detects activity on target ports SLP 427, CIM 5989 and 5988, and then examines the **Service Registry Queries** related list, at **Discovery Definition > Port Probes**, for the SLP query. The base system provides this query is provided to detect the **service:wbem** service type, which indicates the presence of an SLP server.
3. The Shazzam probe launches a scanner for the WBEM service type. The scanner retrieves:
 - The attributes of the service from the SLP server.
 - The interop namespaces^[6] of CIM servers in the network.
4. The scanner appends the namespace values it finds to the port probe results.
5. The wbem port probe appends the SLP data it carries to the CIM Classify probe.
6. The CIM Classify probe uses that information to explore the CIM servers.



wbem Port Probe

The **wbem** probe stores the data it retrieves in the CIM Classification [discovery_classy_cim] table. The wbem port probe is defined in **Discovery Definition > Port Probes**.

Discovery Port Probe

UpdateDelete

Name:wbem

Description:WBEM (CIM)

Scanner:Generic TCP

Conditional:☐

Active:☒

Cls:☒

IPs:☐

Triggered by services:wbem_http, wbem_https

Triggers probe:CIM Classify

Use classification:CIM Classification [discovery_classy_cim]

Classification priority:1

Script:

1

UpdateDelete

Service Registry Queries

New

Go toType

1 to 1 of 1

Port Probe = wbem

Type

☐ SLP Query

Actions on selected rows...

1 to 1 of 1

SLP Query

The SLP query detects the wbem service (**service:wbem**) on an SLP server and gathers the attributes of the service.

SLP Query

UpdateDelete

Port Probe:wbem

Type:SLP Query

UpdateDelete

Service Types

New

Go toService Type

1 to 1 of 1

SLP Query = bb551dbd37331000627d8c00dfbe5d89

Service Type

☐ service:wbem

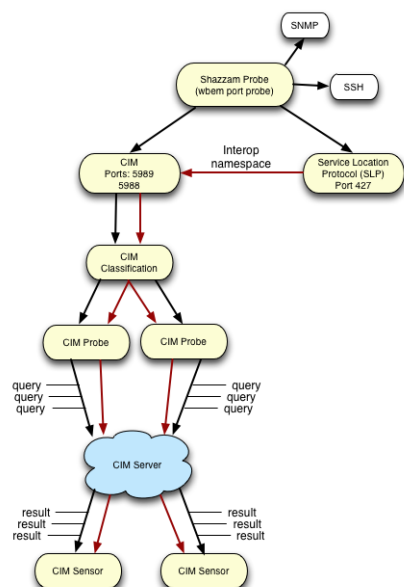
Actions on selected rows...

1 to 1 of 1

To access the SLP query:

1. Navigate to *Discovery Definition > Port Probes*.
2. Select the **wbem** port probe record.
3. From the **Service Registry Queries** related list, select the **SLP Query**.

The wbem port probe appends the SLP data it carries to the CIM Classify probe before launching it. The CIM Classify probe extracts VMware ESX serial numbers and connector relationships between the SAN and NAS components from CIM Servers in the network.



1. Navigate to **Discovery Definition > Probes**.
2. Select the **CIM - Classify** probe.

CIM Probe
Update Delete

Name:
Probe Type:

Description:

Classifies a system via CIM/SLP.
 Queries are dynamically built by collecting all CIM CI Classifications and aggregating their Classification Criteria.
 Duplicate CIM queries will be merged.
 CIM Queries are dynamically created by merging all CIM CI Classification Criteria.

Used by Discovery: ☒

Used by Runbook: ☐

Update
Delete

Related Links

[Test probe](#)

CIM Queries |
 Probe parameters (1) |
 Sensors (1) |
 Included by MultiProbe |
 Responds to Probes

Probe parameters ▾ New Go to
« 1 to 1 of 1 »

Name	Value	Value script	Active
queries		var cimQueries = []; var classifierReco...	true

☐ Actions on selected rows...
 « 1 to 1 of 1 »

Enhancements

Eureka

- Provides more efficient querying of SMI-S compliant storage devices using CimIQL queries.

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/reference/r_HowCIMDiscoveryWorks.html
 - [2] [http://en.wikipedia.org/wiki/Common_Information_Model_\(computing\)](http://en.wikipedia.org/wiki/Common_Information_Model_(computing))
 - [3] http://www.snia.org/tech_activities/standards/curr_standards/smi
 - [4] http://en.wikipedia.org/wiki/Web-Based_Enterprise_Management
 - [5] http://en.wikipedia.org/wiki/Service_Location_Protocol
 - [6] http://sfdoccentral.symantec.com/ccstor/5.1.1/win_unix/html/cc_hscg/apas01.htm
-

Software

Apache Web Servers

Overview

Starting in Fuji, Discovery can find information about Apache Web Servers. The Unix - Active Processes probe captures the following information for Apache web servers. When the Apache web server process is detected during discovery, if the mod_jk module or the Apache mod_proxy module is running on the target Apache web server, additional probes trigger for each module. The probes gather configuration parameters from each module configuration for additional sensor processing.

Data Collected by Discovery on All Apache Web Servers

Table Name	Field and Label Name	Probe
cmdb_ci_web_server	Name [name]	apcfg
cmdb_ci_web_server	Version[version]	httpd
cmdb_ci_web_server	Description [short_description]	httpd

Adding sudo access for the Unix - Active Processes Probe

To elevate sudo privileges for the Unix - Active Processes probe to access the Apache web server:

1. Navigate to **Discovery > Probes**.
2. In the **Search** field, search for Unix - Active Processes.
3. Click the Unix - Active Processes probe.
4. In the **Probe Parameters** related list, click **New**.
5. Click **New**.
6. Use the following information to fill out the form:
 - **Name:** *must_sudo*
 - **Value:** *true*
7. Click **Submit**.

Mod_jk Module

The Apache mod_jk module forwards requests from the Apache web server to a Servlet container, such as Tomcat. Additional mod_jk directives can also manage load balancing. Discovery populates the CMDB when it detects an Apache Server. When the Apache Web Server process is detected, if the mod_jk module is running on the web server as a load balancer, the related information populates to the CMDB.

Requirements for Discovery

Consider the following requirements for discovering an Apache server that contains the mod_jk module:

- The MID Server user account to explore the target server must have access to the *httpd.conf* configuration file in the */etc/httpd/conf/* folder.
- Discovery uses secure shell (SSH) commands to identify the following associated elements:
 - Apache Get Configuration
 - Apache Version
 - Apache Get JK Module
- The following probes require execute privileges to run commands:

Probe	Commands
Apache – Get Configuration	echo, sed, httpd, cut, grep, egrep (within the Bourne shell script)
Apache – Version	httpd
Apache – Get JK Module	echo, sed, httpd, cut, grep, egrep (within the Bourne shell script)

Probes and Sensors

Discovery uses the Unix - Active Processes probe to identify an Apache server that contains the mod_jk module:

1. The Unix - Active Processes probe detects a running process that matches one of the following criteria:
 - The name of the process is *httpd*.
 - The name of the process is *apache*.
2. If there is a match on one of these criteria, a record is created in the Web Server table [cmdb_ci_web_server] if one does not already exist for that running process. The following probes are also triggered:
 - **Apache – Version:** the sensor of this probe populates the Apache version information in the Web Server record.
 - **Apache – Get Configuration:** this probe contains a Bourne shell script and an argument that determines the path of the Apache configuration file. The sensor of this probe populates some additional information in the Web Server record.
3. The sensor processing of Apache – Get configuration probe results triggers the following probes if the mod_jk module is running on the web server:
 - **Apache – JK Module:** if the mod_jk module is running as a load balancer on the server, the sensor of this probe populates the information in the Load Balancer Service [cmdb_ci_lb_service], Load Balancer Pool [cmdb_ci_lb_pool] and Load Balancer Pool Member [cmdb_ci_lb_pool_member] tables.

Data Collected

For the mod_jk module with no load balancer, the following data is collected by default:

Default Data Collected by Discovery

Table Name	Field and Label Name	Probe
cmdb_ci_web_server	Name [name]	Apache – Get Configuration
cmdb_ci_web_server	Version [version]	Apache – Version
cmdb_ci_web_server	Description [short_description]	Apache – Version

If the mod_jk module is enabled for load balancing, Discovery connects the following data:

Data Collected by Discovery When the mod_jk Module is Identified as a Load Balancer

Table Name	Field and Label Name	Probe
cmdb_ci_lb_appl	Name [name]	Apache – Apache - Get JK Module
cmdb_ci_lb_appl	IP Address [ip_address]	Apache – Apache - Get JK Module
cmdb_ci_lb_appl	Last Discovered [last_discovered]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool_member	Name [name]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool_member	Last Discovered [last_discovered]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool_member	IP Address [ip_address]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool_member	Load Balancer [load_balancer]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool_member	Port [port]	Apache – Apache - Get JK Module
cmdb_ci_lb_service	Input URL [input_url]	Apache – Apache - Get JK Module
cmdb_ci_lb_service	Last Discovered [last_discovered]	Apache – Apache - Get JK Module
cmdb_ci_lb_service	IP Address [ip_address]	Apache – Apache - Get JK Module
cmdb_ci_lb_service	Name [name]	Apache – Apache - Get JK Module
cmdb_ci_lb_service	Load Balancer [load_balancer]	Apache – Apache - Get JK Module
cmdb_ci_lb_service	Port [port]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool	Last Discovered [last_discovered]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool	Load balancing Method [load_balancing_method]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool	Load Balancer [load_balancer]	Apache – Apache - Get JK Module
cmdb_ci_lb_pool	Name [name]	Apache – Apache - Get JK Module

Relationships

In addition to data population, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

- The records in the cmdb_ci_lb_appl table run on the cmdb_ci_web_server table records.
- The records in the cmdb_ci_lb_service table use the cmdb_ci_lb_pool table records.
- The records in the cmdb_ci_pool table are used by the cmdb_ci_service table record.
- The records in the cmdb_ci_pool table are members of the cmdb_ci_pool_member table.
- The records in the cmdb_ci_pool_member table are members of the cmdb_ci_pool table.

Mod_proxy Module

The Apache mod_proxy module implements a proxy, gateway, or cache for the Apache web server. An additional mod_proxy_balancer can also manage load balancing. Discovery populates the CMDB when it detects an Apache server. When the Apache Web Server process is detected, if the mod_proxy module is running on the web server as a load balancer the related information populates to the CMDB.

Requirements for Discovery

Consider the following requirements for discovering an Apache server that contains the mod_proxy module:

- The MID Server user account to explore the target server must have access to the *httpd.conf* configuration file in the */etc/httpd/conf/* folder.
- Discovery uses secure shell (SSH) commands to identify the following associated elements:
 - Apache Get Configuration
 - Apache Version
 - Apache Get Proxy Module
- The following probes require execute privileges to run commands:

Probe	Commands
Apache – Get Configuration	echo, sed, httpd, cut, grep, egrep (within the Bourne shell script)
Apache – Version	httpd
Apache – Get Proxy Module	grep, egrep (within the Bourne shell script)

Probes and Sensors

Discovery uses the Unix - Active Processes probe to identify an Apache server that contains the mod_proxy module. The probes and sensors operate in the following manner:

1. The Unix - Active Processes probe detects a running process that matches one of the following criteria:
 - The name of the process is *httpd*.
 - The name of the process is *apache2*.
2. If there is a match on one of these criteria, a record is created in the Web Server table [cmdb_ci_web_server] if one does not already exist for that running process. The following probes are also triggered:
 - **Apache – Version:** the sensor of this probe populates the Apache version information in the Web server [cmdb_ci_web_server] record.
 - **Apache – Get Configuration:** this probe contains a Bourne shell script and an argument that determines the path of the Apache configuration file. The sensor of this probe populates some additional information in the Web server [cmdb_ci_web_server] record.
3. The sensor processing of the Apache – Get configuration probe results triggers the following probes if the mod_proxy module is running on the web server:

- **Apache - Get Proxy Module:** if the mod_proxy module is running as a load balancer on the server, the sensor of this probe populates the information in the Load Balancer Service [cmdb_ci_lb_service], Load Balancer Pool [cmdb_ci_lb_pool] and Load Balancer Pool Member [cmdb_ci_lb_pool_member] tables.

Data Collected

For the mod_proxy module with no load balancer, the following data is collected by default:

Data Collected by Discovery by Default

Table Name	Field and Label Name	Probe
cmdb_ci_web_server	Name [name]	Apache – Get Configuration
cmdb_ci_web_server	Version [version]	Apache – Version
cmdb_ci_web_server	Description [short_description]	Apache – Version

If the mod_proxy module is enabled for load balancing, Discovery connects the following data:

Data Collected by Discovery if the mod_proxy Module is Identified as a Load Balancer

Table Name	Field and Label Name	Probe
cmdb_ci_lb_appl	Name [name]	Apache - Get Proxy Module
cmdb_ci_lb_appl	IP Address [ip_address]	Apache - Get Proxy Module
cmdb_ci_lb_appl	Last Discovered [last_discovered]	Apache - Get Proxy Module
cmdb_ci_lb_pool_member	Name [name]	Apache - Get Proxy Module
cmdb_ci_lb_pool_member	Last Discovered [last_discovered]	Apache - Get Proxy Module
cmdb_ci_lb_pool_member	IP Address [ip_address]	Apache - Get Proxy Module
cmdb_ci_lb_pool_member	Load Balancer [load_balancer]	Apache - Get Proxy Module
cmdb_ci_lb_pool_member	Port [port]	Apache - Get Proxy Module
cmdb_ci_lb_service	Input URL [input_url]	Apache - Get Proxy Module
cmdb_ci_lb_service	Last Discovered [last_discovered]	Apache - Get Proxy Module
cmdb_ci_lb_service	IP Address [ip_address]	Apache - Get Proxy Module
cmdb_ci_lb_service	Name [name]	Apache - Get Proxy Module
cmdb_ci_lb_service	Load Balancer [load_balancer]	Apache - Get Proxy Module
cmdb_ci_lb_service	Port [port]	Apache - Get Proxy Module
cmdb_ci_lb_pool	Last Discovered [last_discovered]	Apache - Get Proxy Module
cmdb_ci_lb_pool	Load balancing Method [load_balancing_method]	Apache - Get Proxy Module
cmdb_ci_lb_pool	Load Balancer [load_balancer]	Apache - Get Proxy Module
cmdb_ci_lb_pool	Name [name]	Apache - Get Proxy Module

Relationships

In addition to data population, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

- The records in the cmdb_ci_lb_appl table run on the cmdb_ci_web_server table records.
- The records in the cmdb_ci_lb_service table use the cmdb_ci_lb_pool table records.
- The records in the cmdb_ci_pool table are used by the cmdb_ci_service table records.
- The records in the cmdb_ci_pool are members of the cmdb_ci_pool_member table records.
- The records in the cmdb_ci_pool_member is a member of the cmdb_ci_pool table records.

JBoss Servers

Overview

Discovery can detect JBoss Application Servers running on Linux and Windows systems. Discovery creates or updates CMDB records about the servers, hosted web applications, and web services, starting with the Fuji release.

To view the main JBoss server record, navigate to **Configuration > Application Servers > JBoss**. You can view the server's associated web applications and web services from the **Related Lists** tab of the record.

Linux

Discovery can collect data about JBoss Application Servers running on Linux systems.

Requirements

For JBoss Application Servers running on Linux systems:

- Set probe permissions to use these Bourne shell commands: find, cat, and dirname.
- Enable SSH on the JBoss Application server. The SSH credential must also have read permissions on the *web.xml* and *jboss-service.xml* files.

Probes and Sensors

Discovery identifies a Linux JBoss Application Server with the following probes and sensors:

1. The Unix - Active Processes probe detects a running process that matches an `org.jboss.main` entry point parameter.
2. If there is a match, a record is created in the JBoss [cmdb_ci_app_server_jboss] table. The following probes are also triggered:
 - **JBoss - Find web.xml list:** the sensor of this probe populates information in the Web Application [cmdb_ci_web_application] table if applicable.
 - **JBoss - Get jboss-service.xml:** the sensor of this probe populates information in the Web Service [cmdb_ci_web_service] table.
3. The JBoss - Find web.xml list probe searches for the *web.xml* files of JBoss Application Server. The probe uses the `classpath` parameter in the running process, and then searches in the related *server\default\deploy* directory for the JBoss installation.
4. If associated web applications reside in the *server\default\deploy* directory, the JBoss - Get web.xml probe triggers for each application. This probe reads the *web.xml* file for each web application and the sensor populates additional information to the Web Service table.

5. The Boss - Get jboss-service.xml probe uses the `classpath` parameter in the running process to search for the *jboss-service.xml* file in the related *server\default\conf* directory for the JBoss installation.
6. If the probe successfully finds the *jboss-service.xml* file in the *server\default\conf* directory, the sensor reads the contents of the XML file and creates additional records in the Web Service table as necessary.

Data Collected

The following data is collected from Linux JBoss Application Servers.

Data Collected by Discovery on JBoss Servers

Table Name	Field and Label Name	Source
cmdb_ci_web_service	Name [name]	jboss-service.xml
cmdb_ci_web_service	App server [app_server]	Internal reference
cmdb_ci_web_application	Description [short_description]	web.xml
cmdb_ci_web_application	Name [name]	web.xml
cmdb_ci_web_application	Document base [document_base]	web.xml
cmdb_ci_web_application	App server [app_server]	web.xml

Relationships

For Linux, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

- The records in the Web Application [cmdb_ci_web_application] table run on the Linux Server [cmdb_ci_linux_server] table records.
- The records in the Web Service [cmdb_ci_web_service] table run on the Linux Server [cmdb_ci_linux_server] table records.

Windows

Discovery can collect data about JBoss Application Servers running on Windows systems.

Requirements

For JBoss Application Servers running on Windows systems, enable PowerShell on the MID Server.

Probes and Sensors

Discovery identifies a Windows JBoss Server with the following probes and sensors:

1. The Windows - Active Processes probe detects a running process that matches an *org.jboss.main* entry point parameter.
2. If there is a match, a record is created in the JBoss Application Server [cmdb_ci_app_server_jboss] table. The following probes are also triggered:
 - **Windows - JBoss Find web.xml list:** the sensor of this probe populates information in Web Application [cmdb_ci_web_application] records if applicable.
 - **Windows – JBoss Get jboss-service.xml:** the sensor of this probe populates information in the Web Service [cmdb_ci_web_service] table.

3. The Windows - JBoss find web.xml list probe searches for the JBoss Application Server *web.xml* files. The probe uses the `classpath` parameter in the running process and then searches in the related *server\default\deploy* directory for the JBoss installation.
4. If associated web applications reside in the *server\default\deploy* directory, a Windows – JBoss Get web.xml probe triggers for each application. This probe reads the *web.xml* for each web application and the sensor populates additional information to the Web Service table.
5. The Windows - JBoss Get jboss-service.xml probe uses the `classpath` parameter in the running process to search for the *jboss-service.xml* file in the related *server\default\conf* directory for the JBoss installation.
6. If the probe successfully finds the *jboss-service.xml* file in the *server\default\conf* directory, the sensor reads the contents of the XML file and creates additional records in the Web Service table as necessary.

Data Collected

The following data is collected from Windows JBoss Application Servers.

Data Collected by Discovery on JBoss Servers

Table Name	Field and Label Name	Source
cmdb_ci_web_service	Name [name]	jboss-service.xml
cmdb_ci_web_service	App server [app_server]	Internal reference
cmdb_ci_web_application	Description [short_description]	web.xml
cmdb_ci_web_application	Name [name]	web.xml
cmdb_ci_web_application	Document base [document_base]	web.xml
cmdb_ci_web_application	App server [app_server]	web.xml
cmdb_ci_web_application	Servlet Name [servlet_name]	web.xml
cmdb_ci_web_application	Servlet Class [servlet_class]	web.xml

Relationships

For Windows, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

- The records in the Web Application [cmdb_ci_web_application] table run on the Windows Server [cmdb_ci_windows_server] table records.
- The records in the Web Service [cmdb_ci_web_service] table run on the Windows Server [cmdb_ci_windows_server] table records.

MySQL Servers

Overview

Starting in Fuji, Discovery creates or updates a CMDB record when it detects a running instance of MYSQL on UNIX or Windows systems.

Requirements for Discovery

Discovery searches for the MySQL configuration file location from the following areas:

- **UNIX:** Discovery searches for the MySQL configuration file location from the `mysqld` process, or port 3306.
- **Windows:** Discovery searches for the MySQL configuration file location from the `mysqld.exe` process, or port 3306.

For each process, the following process parameters are explored in the following order:

1. `--defaults-extra-file`
2. `--defaults-file`

If the MYSQL configuration file location is not found from that search, then the following occurs:

- **UNIX:** The configuration file location defaults to `/etc/my.cnf`.
- **Windows:** No default configuration file location exists, and the probe to read the configuration file location is skipped.

Data Collected

The following data is collected.

Data Collected by Discovery on MySQL Servers

Label	Table Name	Field Name	Source
MySQL configuration	cmdb_ci_db_mysql_instance	myconf	my.cnf
TCP port(s)	cmdb_ci_db_mysql_instance	tcp_port	running_process
Version	cmdb_ci_db_mysql_instance	version	mysqld

PostgreSQL Instances

Starting in Fuji, Discovery creates or updates a CMDB record when it detects a running instance of PostgreSQL on UNIX systems. The following data is collected.

Data Collected by Discovery on PostgreSQL Instances

Label	Table Name	Field Name	Source
Name	cmdb_ci_db_postgresql_instance	name	PostgreSQL Instance@hostname
Data Directory	cmdb_ci_db_postgresql_instance	data_dir	running process
TCP port	cmdb_ci_db_postgresql_instance	tcp_port	running process
SQL Configuration	cmdb_ci_db_postgresql_instance	postgres_conf	data_directory/postgresql.conf
Version	cmdb_ci_db_postgresql_instance	version	postmaster/postgres

Oracle Databases



Note: This article applies to Fuji. For more current information, see *Oracle Database Discovery* ^[1] at <http://docs.servicenow.com>. The ServiceNow Wiki is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

Overview

Starting in Fuji, Discovery can identify an Oracle database instance that is running on UNIX or Windows operating systems.

Discovery Requirements

UNIX Operating System

- Discovery requires credentials that allow read permission to the oratab file.
- Discovery requires credentials that allow read permission to the Server Parameter file.
- Discovery identifies a running instance of an Oracle database from the process that starts with *ora_pmon_*. Ensure this process is running in the IP range you designate.

Windows Operating System

- Discovery requires credentials that allow read permission to the Server Parameter file.
- Discovery identifies a running instance of an Oracle database from the *oracle.exe* process. Ensure this process is running in the IP range you designate.

Data Collected by Discovery

UNIX Operating System

The following data is collected:

Data Collected by Discovery on Oracle Databases

Label	Table Name	Field Name	Source
SID (UNIX)	cmdb_ci_db_ora_instance	sid	Extracts SID from the name of the process that starts with <i>ora_pmon_</i>
Version	cmdb_ci_db_ora_instance	version	<p>Tries to extract the version in this order</p> <ul style="list-style-type: none"> From the output of the <i>ORA_HOME/bin/sqlplus /NOLOG</i> command From the output of the <i>ORA_HOME/bin/lsnrctl status</i> command From the path of <i>ORA_HOME</i>
Oracle Home	cmdb_ci_db_ora_listener	oracle_home	From the <i>ORATAB</i> file
Server Parameter File (SPfile)	cmdb_ci_db_ora_instance	spfile	<p>The following locations are explored for the location of the Server Parameter File. If this file does not exist in one of the explored locations, Discovery does not find the file and reports an error.</p> <ul style="list-style-type: none"> <i>oracle_home/dbs/spfileSID.ora</i> <i>oracle_home/dbs/spfile.ora</i> <i>oracle_home/dbs/initSID.ora</i>

Windows Operating System

The following data is collected:

Data Collected by Discovery on Oracle Databases

Label	Table Name	Field Name	Source
SID (Windows)	cmdb_ci_db_ora_instance	sid	Extracts SID from the process parameter that is passed to the <i>oracle.exe</i> process
Version	cmdb_ci_db_ora_instance	version	From the output of the <i>ORA_HOME/bin/sqlplus.exe -V</i> command
Oracle Home	cmdb_ci_db_ora_listener	oracle_home	Parsed from the path of <i>oracle.exe</i>
Server Parameter File (SPfile)	cmdb_ci_db_ora_instance	spfile	<p>The following locations are explored for the location of the Server Parameter File. If this file does not exist in one of the explored locations, Discovery does not find the file and reports an error.</p> <ul style="list-style-type: none"> <i>oracle_home\database\spfileSID.ora</i> <i>oracle_home\database\spfile.ora</i> <i>oracle_home\database\initSID.ora</i>

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/concept/c_OracleDatabaseDiscovery.html

HBase

Starting in Fuji, Discovery creates or updates a CMDB record when it detects a running instance of HBase on a UNIX server. The following data is collected.

Data Collected by Discovery on HBase Instances

Label	Table Name	Field Name	Source
Name	cmdb_ci_db_hbase_instance	name	HBase Instance@ <i>hostname</i> ¹
Root Directory	cmdb_ci_db_hbase_instance	root_dir	hbase-site.xml
TCP port	cmdb_ci_db_hbase_instance	tcp_port	running process
Site XML	cmdb_ci_db_hbase_instance	site_xml	hbase-site.xml
Version	cmdb_ci_db_hbase_instance	version	HBase shell
HBase Home	cmdb_ci_db_hbase_instance	hbase_home	running process
ZooKeeper Quorum	cmdb_ci_db_hbase_instance	zookeeper	hbase-site.xml

¹The HBase process classifier obtains the master or slave name from the ps command output, if full output is returned. The name contains Hmaster if the instance is a master instance. The name contains HRegionServer if the name is a slave instance.

MongoDB Instances

Starting in Fuji, Discovery creates or updates a CMDB record when it detects a running instance of MongoDB. The following data is collected on UNIX, Windows or both operating systems.

Data Collected by Discovery on MongoDB Instances

Label	Table Name	Field Name	Source
Version	cmdb_ci_db_mongodb_instance	version	mongod (UNIX) or mongod.exe (Windows)
Mongo configuration	cmdb_ci_db_mongodb_instance	mongodb_conf	mongod.conf
TCP port(s)	cmdb_ci_db_mongodb_instance	tcp_port	Process Classification or mongod.conf

Tomcat Servers

Discoverable Web Applications

Discovery can identify and classify Web applications present in either the CATALINA_BASE or CATALINA_HOME directories.

Tables and Fields

Discovery stores information about Tomcat servers and Web applications in the following tables and fields.

Data Collected by Discovery on Tomcat Servers

Label	Table Name	Field Name	Source
Server port	cmdb_ci_app_server_tomcat	server_port	server.xml
Version	cmdb_ci_app_server_tomcat	version	server.info
Tomcat	cmdb_ci_tomcat_connector	tomcat	server.xml
Port	cmdb_ci_tomcat_connector	port	server.xml
App server	cmdb_ci_web_service	app_server	Internal reference
Description	cmdb_ci_web_application	short_description	web.xml
Document base	cmdb_ci_web_application	document_base	web.xml
App server	cmdb_ci_web_application	app_server	web.xml

Weblogic Application Servers



Note: This article applies to Fuji. For more current information, see *WebLogic Application Server Discovery*^[1] at <http://docs.servicenow.com>. The ServiceNow Wiki is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

Overview

Discovery can detect Oracle or BEA WebLogic Application Servers running on Linux and Windows systems, starting in Fuji. Discovery creates or updates a CMDB record when it detects a running instance of a WebLogic Application Server. Navigate to **Configuration > Application Servers > BEA Weblogic** to view the main Weblogic server record. You can view the servers web applications from the **Related Lists** tab of the record.

Linux

Requirements for Discovery

- The Linux - Weblogic - Find config.xml probe requires the use of these Bourne shell commands: find, cat, test, and dirname. The SSH credential must also have read permissions on the *config.xml* file.
- WebLogic administration server instances started via NodeManager must have the `-Dweblogic.RootDirectory=<path>` parameter defined and visible through the Linux ps process stat command (for each AdminServer) in order for the rest of the Linux WebLogic Application Server and web application information to be populated in the CMDB.

Probes and Sensors

For Linux, Discovery identifies the Linux WebLogic Server using the following probes and sensors:

1. The Unix - Active Processes probe detects a running process that matches one of the follow criteria:
 - The parameters of the process contain `weblogic.Server`.
 - The parameters of the process contain `-Dweblogic.name`.
2. If there is a match on one of the criteria:
 - A record is created in the BEA Weblogic [cmdb_ci_app_server_weblogic] table. The record is populated with the server name and TCP port, which is gathered from the running process.
 - The Linux - Weblogic - Find config.xml probe triggers. The sensor of this probe populates additional information in the BEA Weblogic [cmdb_ci_app_server_weblogic] record and the Web Application [cmdb_ci_web_application] record if applicable.
3. The Linux - Weblogic - Find config.xml probe attempts to find the related *config.xml* file for the server by either:
 - Using the `-Dweblogic.RootDirectory=<path>` parameter defined in the running process.
 - Searching for the parent process that started the WebLogic server (only viable if the *weblogic jvm* was started via the *startWeblogic.sh* or related custom script and not the `init` process).

Adding sudo access for the Weblogic - Find config.xml Probe

To elevate sudo privileges for the Linux - Weblogic - Find config.xml probe to access the Apache web server:

1. Navigate to **Discovery Definition > Probes**.
2. Search for and select the **Linux - Weblogic - Find config.xml** probe.
3. In the **Probe Parameters** related list, click **New**.
4. Enter the following information on the Probe Parameter form:
 - **Name:** *must_sudo*
 - **Value:** *true*
5. Click **Submit**.

Data Collected

For Linux, Discovery collects and stores default information from the WebLogic Application Server in the following table fields:

Default Data Collected by Discovery on Linux Weblogic Application Servers

Table Name	Field Label and Name	Source
cmdb_ci_app_server_weblogic	Name [name]	running process
cmdb_ci_app_server_weblogic	TCP port(s) [tcp_port]	running process

Discovery collects and stores information from the *config.xml* file of the WebLogic Application Server in the following table fields:

Data Collected by Discovery on Linux Weblogic Application Servers

Table Name	Field Label and Name	Source
cmdb_ci_app_server_weblogic	Version [version]	config.xml
cmdb_ci_app_server_weblogic	Weblogic Domain [weblogic_domain]	config.xml
cmdb_ci_web_application	Name [name]	config.xml
cmdb_ci_web_application	Context path [context_path]	config.xml
cmdb_ci_web_application	App server [app_server]	config.xml

Relationships

For Linux, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

Parent Class	Relationship Type	Child Class
cmdb_ci_app_server_weblogic	Runs on::Runs	cmdb_ci_linux_server
cmdb_ci_web_application	Runs on::Runs	cmdb_ci_app_server_weblogic
cmdb_ci_app_server_weblogic	Depends on::Used by	cmdb_ci_app_server_weblogic†

†This relationship is made between an AdminServer and any managed servers it encapsulates.

Windows

Requirements for Discovery

- PowerShell must be enabled on the MID Server, in order for the WebLogic probes to gather full server and web application information.
- The WebLogic Administration Server instances that start via WebLogic NodeManager must have the `-Dweblogic.RootDirectory=<path>` parameter defined upon server startup. The Windows credential must also have read permissions on the *config.xml* file.

Probes and Sensors

For Windows, Discovery identifies the Windows WebLogic Server using the following process:

1. The Windows - Active Processes probe detects a running process that matches one of the follow criteria:
 - The parameters of the process contain `weblogic.Server`.
 - The parameters of the process contain `-Dweblogic.name`.
2. If there is a match on one of the criteria:
 - A record is created in the BEA Weblogic [cmdb_ci_app_server_weblogic] table. The record is populated with the server name and TCP port, which is gathered from the running process.
 - The Windows - Weblogic - Find config.xml probe triggers. The sensor of this probe populates additional information in the BEA Weblogic [cmdb_ci_app_server_weblogic] record and the Web Application [cmdb_ci_web_application] record if applicable.
3. The **Windows - Weblogic - Find config.xml** probe attempts to find the related *config.xml* file for the server by:
 - Using the `-Dweblogic.RootDirectory=<path>` parameter defined in the running process.
 - Searching for *config.xml* files under the `-Dplatform.home=<path>` parameter defined in the running process (not as efficient using the parameters of the process).
4. If there are associated web applications found in the WebLogic *config.xml* file, the Windows – Weblogic find web.xml probe triggers for each application. This probe reads the WebLogic *web.xml* file for each web application and the sensor then populates additional information.

Data Collected

For Windows, Discovery collects and stores default information from the WebLogic Application Server in the following table fields:

Default Data Collected by Discovery on Windows WebLogic Application Servers

Table Name	Field Label and Name	Source
cmdb_ci_app_server_weblogic	Name [name]	running process
cmdb_ci_app_server_weblogic	TCP port(s) [tcp_port]	running process

Discovery collects and stores information from the Windows WebLogic Application Server configuration files in the following table fields:

Data Collected by Discovery from the Windows WebLogic Application Server configuration files

Table Name	Field Label and Name	Source
cmdb_ci_app_server_weblogic	Version [version]	config.xml
cmdb_ci_app_server_weblogic	Weblogic Domain [weblogic_domain]	config.xml
cmdb_ci_web_application	Name [name]	config.xml
cmdb_ci_web_application	Document base [document_base]	config.xml
cmdb_ci_web_application	Description [short_description]	web.xml
cmdb_ci_web_application	Servlet class [servlet_class]	web.xml
cmdb_ci_web_application	Servlet name [servlet_name]	web.xml
cmdb_ci_web_application	App server [app_server]	config.xml
cmdb_ci_web_application	TCP port(s) [tcp_port]	web.xml

Relationships

For Windows, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

Parent Class	Relationship Type	Child Class
cmdb_ci_app_server_weblogic	Runs on::Runs	cmdb_ci_linux_server
cmdb_ci_web_application	Runs on::Runs	cmdb_ci_app_server_weblogic
cmdb_ci_app_server_weblogic	Depends on::Used by	cmdb_ci_app_server_weblogic†

†This relationship is made between an AdminServer and any managed servers it encapsulates.

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/concept/c_DataCollDiscoWebLogicServers.html

Websphere Servers

Overview

The IBM WebSphere Application Server is a software framework with middleware that hosts Java-based web applications. Discovery can detect a WebSphere Application Server and populate the CMDB with information about the server and its web applications, starting in Fuji.

To view the main record for a WebSphere Application Server, navigate to **Configuration > Application Servers > IBM Websphere**. You can also view the server's web applications and web services from the **Related Lists** tab of the record.

Linux

Discovery can collect data about WebSphere Application Servers running on Linux systems.

Discovery Requirements

For WebSphere Application Servers running on Linux systems:

- Enable SSH on the WebSphere Application Server.
- Set execute privileges to enable the following probes to run commands:

Probe	Commands
WebSphere - Get serverindex.xml	cat/read permissions on the <i>serverindex.xml</i> file
WebSphere - Get cell.xml	cat/read permissions on the <i>cell.xml</i> file
WebSphere - Get server.xml	cat/read permissions on the <i>server.xml</i> file

Probes and Sensors

For Linux, Discovery identifies the WebSphere Application Server using the following process:

1. The Unix - Active Processes probe detects a running process that matches the `com.ibm.ws.runtime.WsServer`. For example, the `...com.ibm.ws.bootstrap.WSLauncher com.ibm.ws.runtime.WsServer` process output has a parameter of `/opt/IBM/WPS/profiles/ccmdb/config localhostNode01Cell ccmdb01 server1`. The values are:
 - last parameter = server_name = server1
 - last parameter - 1 = node_name = ccmdb01
 - last parameter - 2 = cell_name = localhostNode01Cell
 - last parameter - 3 = config_path = '/opt/IBM/WPS/profiles/ccmdb/config'
2. If there is a match, a record is created in the IBM Websphere [cmdb_ci_app_server_websphere] table. The following probes are triggered:
 - **WebSphere - Cell**
 - **WebSphere - Web Applications**
 - **WebSphere - Web Services**
3. The WebSphere - Cell probe searches for the *cell.xml* file for the instance by using the parameters in the running process, and then searching in the related `<config_path>\cells\<cell_name>\` directory.

4. If the probe successfully finds the *cell.xml* file, the sensor reads its contents and populates additional Websphere Cell [cmdb_ci_websphere_cell] table records as necessary.
5. The WebSphere - Web Applications probe searches the *serverindex.xml* file for the instance by using the parameters in the running process, and then searching in the related `<config_path>\cells\<cell_name>\nodes\<node_name>\` directory.
6. If the probe successfully finds the *serverindex.xml* file, the sensor read its contents and populates additional Web Application [cmdb_ci_web_application] table records as necessary.
7. The WebSphere - Web Services probe searches for the *server.xml* file for the instance by using the parameters in the running process, and then searching in the related `<config_path>\cells\<cell_name>\nodes\<node_name>\servers\<server_name>` directory.
8. If the probe successfully finds the *server.xml* file, the sensor reads its contents and populates additional Web Service [cmdb_ci_web_service] table records as necessary.

Data Collected

Discovery gathers information from WebSphere Application Servers running on Linux systems and stores it in the following table fields:

Data Collected by Discovery on Linux-Based WebSphere Application Servers

Table Name	Field Label and Name	Source
cmdb_ci_websphere_cell	Name [name]	server.xml
cmdb_ci_web_service	App server [app_server]	Internal reference
cmdb_ci_web_service	Name [name]	server.xml
cmdb_ci_web_application	Name [name]	serverindex.xml
cmdb_ci_web_application	App server [app_server]	serverindex.xml

Relationships

For Linux, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

Parent Class	Relationship Type	Child Class
cmdb_ci_app_server_websphere	Runs on::Runs	cmdb_ci_linux_server
cmdb_ci_web_application	Runs on::Runs	cmdb_ci_app_server_websphere
cmdb_ci_web_service	Runs on::Runs	cmdb_ci_app_server_websphere
cmdb_ci_websphere_cell	Contains::Contained by	cmdb_ci_app_server_websphere

Windows

Discovery can collect data about WebSphere Application Servers running on Windows systems.

Discovery Requirements

For WebSphere Application Servers running on Windows systems, enable Powershell on the MID Server.

Probes and Sensors

For Windows, Discovery identifies the WebSphere Application Server using the following process:

1. The Windows - Active Processes probe detects a running process that matches the `'com.ibm.ws.runtime.WsServer'`. For example, the `...com.ibm.ws.bootstrap.WSLauncher com.ibm.ws.runtime.WsServer` process output has a parameter of `/opt/IBM/WPS/profiles/ccmdb/config localhostNode01Cell ccmdb01 server1`. The values are:
 - last parameter = server_name = server1
 - last parameter - 1 = node_name = ccmdb01
 - last parameter - 2 = cell_name = localhostNode01Cell
 - last parameter - 3 = config_path = '/opt/IBM/WPS/profiles/ccmdb/config'
2. If there is a match, a record is created in the IBM Websphere [cmdb_ci_app_server_websphere] table. The following probes are triggered:
 - Windows - WebSphere - Cell
 - Windows - WebSphere - Web Applications
 - Windows - WebSphere - Web Services
3. The Windows - Websphere - Cell probe searches for the `cell.xml` file for the instance by using the parameters in the running process, and then searching in the related `<config_path>\cells\<cell_name>\` directory.
4. If the probe successfully finds the `cell.xml` file, the sensor reads its contents and populates additional Websphere Cell [cmdb_ci_websphere_cell] table records as necessary.
5. The Windows - WebSphere - Web Applications probe searches the `serverindex.xml` file for the instance by using the parameters in the running process, and then searching in the related `<config_path>\cells\<cell_name>\nodes\<node_name>\` directory.
6. If the probe successfully finds the `serverindex.xml` file, the sensor read its contents and populates additional Web Application [cmdb_ci_web_application] table records as necessary.
7. The Windows - Websphere - Web Services probe searches for the `server.xml` file for the instance by using the parameters in the running process, and then searching in the related `<config_path>\cells\<cell_name>\nodes\<node_name>\servers\<server_name>` directory.
8. If the probe successfully finds the `server.xml` file, the sensor reads its contents and populates additional Web Service [cmdb_ci_web_service] table records as necessary.

Data Collected

Discovery gathers information from WebSphere Application Servers running on Windows systems and stores it in these table fields:

Data Collected by Discovery on Windows-based WebSphere Application Servers

Table Name	Field Label and Name	Source
cmdb_ci_app_server_websphere	Name [name]	Running process
cmdb_ci_websphere_cell	Name [name]	cell.xml
cmdb_ci_websphere_cell	Cell ID [cell_id]	cell.xml
cmdb_ci_websphere_cell	Cell type [cell_type]	cell.xml
cmdb_ci_websphere_cell	Cell discovery protocol [cell_discovery_protocol]	cell.xml
cmdb_ci_web_service	Name [name]	server.xml
cmdb_ci_web_service	Service ID [service_id]	server.xml
cmdb_ci_web_application	Name [name]	serverindex.xml

Relationships

For Windows, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

Parent Class	Relationship Type	Child Class
cmdb_ci_app_server_websphere	Runs on::Runs	cmdb_ci_win_server
cmdb_ci_web_application	Runs on::Runs	cmdb_ci_app_server_websphere
cmdb_ci_web_service	Runs on::Runs	cmdb_ci_app_server_websphere
cmdb_ci_websphere_cell	Contains::Contained by	cmdb_ci_app_server_websphere

Microsoft IIS Servers

Data Collected by Discovery on Microsoft IIS Servers

Label	Table Name	Field Name	Source
Version	cmdb_ci_web_server	version	Windows registry
Name	cmdb_ci_web_site	name	wmi
Log directory	cmdb_ci_web_site	log_directory	wmi
Description	cmdb_ci_web_site	short_description	wmi
Correlation ID	cmdb_ci_web_site	correlation_id	Internal
IP address	cmdb_ci_web_site	ip_address	wmi
TCP port	cmdb_ci_web_site	tcp_port	wmi



Note: You must install *IIS Management Scripts and Tools* on a Microsoft IIS Server in order for Discovery to collect data from it.

General Software Packages

Data Collected by Discovery on General Software Packages

Label	Table Name	Field Name	Source
Name	cmdb_ci_spkg	name	Various
Version	cmdb_ci_spkg	version	Various
Install count	cmdb_ci_spkg	install_count	Various
License count	cmdb_ci_spkg	license_count	Various
Microsoft SMS ID	cmdb_ci_spkg	msft_sms_id	Various
Installed on	cmdb_software_instance	installed_on	Various
Software	cmdb_software_instance	software	Various

VMware vCenter

Discovery

Orchestration

Related Topics

- Cloud Provisioning
- Help the Help Desk
- Help the Help Desk Login Script
- ECC Queue
- Useful Related Lists in CI Forms
- Creating a Workflow
- Using Workflow Activities

Get the Book



Discovery



Data Collected by Discovery



Orchestration for VMWare



Note: This article applies to Fuji and earlier releases. For more current information, see *Discovery for VMware vCenter* ^[1] at [http://docs.servicenow.com](http://docs.servicenow.com/docs.servicenow.com) **The ServiceNow Wiki is no longer being updated. Visit <http://docs.servicenow.com> for the latest product documentation.**

Overview

These options are available for getting VMware vCenter data:

- ServiceNow Discovery ^[2] can run the VMWare - vCenter probe when it identifies a VMware vCenter process running on a Windows machine.
- ServiceNow Orchestration can run the VMWare - vCenter probe from a workflow.

See VMware Component Relationships for a description of the VMware architecture and component relationships.

ServiceNow Discovery supports these vCenter versions:

- vCenter versions 4.1 and 5
- vSphere version 5.5
- vCenter appliance version 5.1

Discovery Configuration and Process Flow

Discovery of vCenter CIs requires at least two credentials: the Windows credential for the host on which the vCenter server runs and the VMware credential for the vCenter server. When Discovery runs, a classifier called **vCenter** classifies the process running on a Windows machine and launches the **VMware - vCenter** probe. This probe logs into the vCenter instance with the credentials provided and uses the vCenter API to return information about ESX machines, virtual machines, and resource pools. If using a domain account to access vCenter, specify the domain with the user name in the credential record in one of the supported formats such as **Domain\UserName**.



Note: Make sure to select a credential **Type** of **VMware**.

Orchestration Configuration and Workflow

Administrators can configure instances with Orchestration to run a workflow called **Discover vCenter** to populate the CMDB with vCenter data without having to activate the Discovery plugin. This workflow is particularly useful for an administrator who has just installed vCenter and wants to begin using it for cloud provisioning, but has not purchased Discovery. This workflow is available starting with the Calgary release.

The workflow resolves the DNS name and populates the CMDB by running the standard VMware - vCenter probe (installed by default). You must have a properly configured MID Server connected to the ServiceNow instance to use this feature.

You can discover a vCenter instance using one of these methods available on the VMware vCenter Instance form:

- **Discover vCenter Details:** This related link is available to users with the `vmware_operator` role.
- **Scheduled discoveries:** Users with the `cloud_admin` role can create and edit scheduled discoveries from this related list. Users with the `vmware_operator` role can view scheduled discoveries.



Note: For more information about these roles and capabilities, see *Cloud Provisioning Security*.

To discover a single vCenter instance:

1. Navigate to **VMware > Configuration > vCenter Instances**.
2. Select the vCenter instance you want to discover.
3. Under **Related Links**, click **Discover vCenter Details**.

VMware vCenter Instance [Update] [Delete] [Refresh] [Download]

Name: ServiceNow vCenter

Full name: VMware vCenter Server 5.0.0 build-455964

API version: 5.0

URL: <https://10.15.235.24/sdk>

Related Items: [Add] [List] [Grid] [Table] [Show 3 Levels]

Manages - VMware vCenter Datacenters

- San Diego Dev Data Center
- Seattle Staging Data Center
- Virginia QA Data Center
- San Jose Prod Data Center

[Update] [Delete]

Related Links

[Discover vCenter Details](#)

[Subscribe](#)

Scheduled discoveries [New] [Edit...] Go to [Scheduled discovery] [Search]

1 to 1 of 1

vCenter instance = ServiceNow vCenter

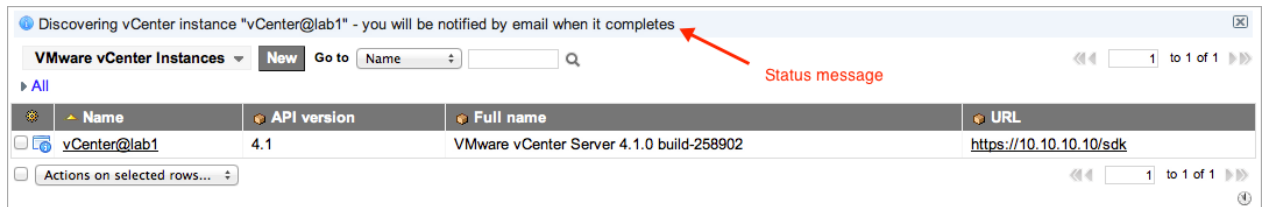
Scheduled discovery

Scheduled discovery
ServiceNow vCenter

Actions on selected rows...

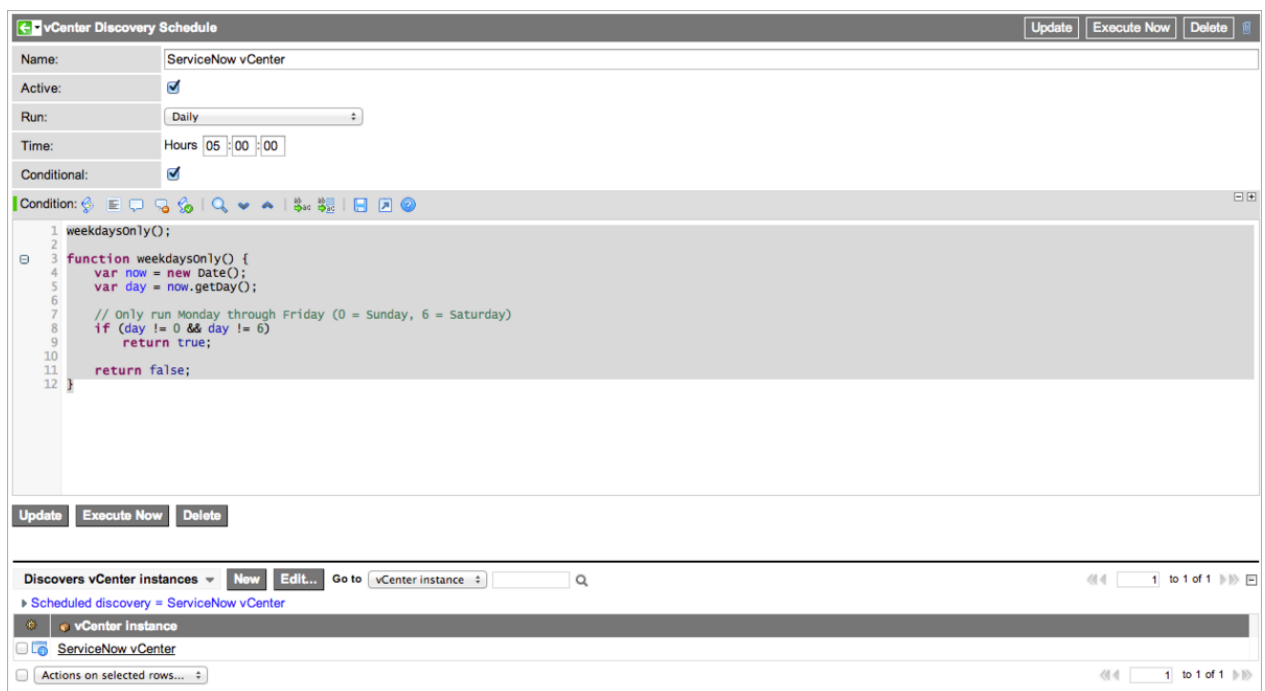
1 to 1 of 1

The list of vCenter instances appears. ServiceNow sends you an email confirmation when the scan is complete.



To create a scheduled discovery:

1. Navigate to **VMware > Configuration > vCenter Instances**.
2. Select the vCenter instance you want to discover.
3. Click **New** in the **Scheduled discoveries** related list.
4. Enter a unique and descriptive name for this schedule.
5. Select the day and time to run this discovery, using the field descriptions from the Discovery Schedules form as a reference.
6. [Optional] Select the **Conditional** check box to script a condition that defines a custom run time.
7. Click **Submit**.



The Discovery vCenter workflow runs in *silent mode* for this type of discovery and does not send notifications.

8. Click **Execute Now** to run this discovery immediately.

Configuring an Alternative Port for vCenter

By default, the vCenter probe runs on port 443, which is the standard port for the https protocol. You can specify an alternate port using one of the following procedures.

- Hard code the port information in the URL
- Use the `vcenter_port` probe parameter to specify the alternate port number.

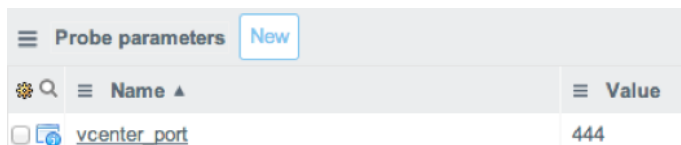
Hard code the port information in the URL

1. Navigate to **Configuration > VMware > vCenter**.
2. Select the specific instance from the list of instances.
3. Edit the **URL** field to include the port.
For example, **https://10.0.0.1:444**
4. Click **Submit**.

Specify the alternate port number in the vCenter probe

To specify an alternate port number for vCenter in the vCenter probe:

1. Navigate to **Discovery > Probes**.
2. Use the Name search to find the probe named **VMWare - vCenter**.
3. In the **ProbeVMWare - vCenter** page, in the Probe Parameters section, click **New**.
4. In the **Name** field, enter `vcenter_port`.
5. In the **Value** field, enter your alternate port number.



6. Click **Submit**.

vCenter Data Collected

Discovery collects the following data for vCenter, including data on the datastores used by the ESX servers.

vCenter

Field Label	Table Name	Column Name	Source
API version	VMware vCenter Instance [cmdb_ci_vcenter]	api_version	VMWare - vCenter probe
Full name	VMware vCenter Instance [cmdb_ci_vcenter]	fullname	VMWare - vCenter probe
Instance UUID	VMware vCenter Instance [cmdb_ci_vcenter]	instance_uuid	VMWare - vCenter probe
URL	VMware vCenter Instance [cmdb_ci_vcenter]	url	VMWare - vCenter probe
Effective CPU	VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	effectivecpu	VMWare - vCenter probe
Effective memory	VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	effectivememory	VMWare - vCenter probe
Number of effective hosts	VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	effectivehosts	VMWare - vCenter probe
Number of hosts	VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	numhosts	VMWare - vCenter probe
Total CPU	VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	totalcpu	VMWare - vCenter probe
Total memory	VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	totalmemory	VMWare - vCenter probe

Top level folder for hosts	VMware vCenter Datacenter [cmdb_ci_vcenter_datacenter]	host_morid	VMWare - vCenter probe
Top level folder for VMs	VMware vCenter Datacenter [cmdb_ci_vcenter_datacenter]	folder_morid	VMWare - vCenter probe
Full path	VMware vCenter Folder [cmdb_ci_vcenter_folder]	fullpath	VMWare - vCenter probe
Accessible	VMware vCenter Network [cmdb_ci_vcenter_network]	accessible	VMWare - vCenter probe
Managed object reference ID	VMware vCenter Object [cmdb_ci_vcenter_object]	morid	VMWare - vCenter probe
vCenter Instance UUID	VMware vCenter Object [cmdb_ci_vcenter_object]	vcenter_uuid	VMWare - vCenter probe
vCenter Reference	VMware vCenter Object [cmdb_ci_vcenter_object]	vcenter_ref	VMWare - vCenter probe
Managed object reference ID	VMware vCenter Server Object [cmdb_ci_vcenter_server_obj]	morid	VMWare - vCenter probe
Folder	vCenter Folder Type [vmware_vcenter_folder_type_m2m]	folder	VMWare - vCenter probe
Type	vCenter Folder Type [vmware_vcenter_folder_type_m2m]	type	VMWare - vCenter probe

Datstores

Discovery identifies each datastore in the system and creates the relationships with the virtual machines and the ESX servers that use these datastores. Discovery collects the following data from datastores.

Field Label	Table Name	Column Name	Source
Accessible	VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	accessible	VMWare - vCenter probe
Capacity (GB)	VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	capacity	VMWare - vCenter probe
Free space (GB)	VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	freespace	VMWare - vCenter probe
Type	VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	type	VMWare - vCenter probe
URL	VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	url	VMWare - vCenter probe

Relationships

Discovery automatically creates relationships for vCenter components using data from a key class. Subsequent Discoveries use the same key class to automatically validate and remove relationships that are no longer valid starting with the Eureka release.

See also VMware Component Relationships

Parent Class	Relationship Type	Child Class	Relationship Key Class
Computer [cmdb_ci_computer]	Virtualized by::Virtualizes	ESX Server [cmdb_ci_esx_server]	Child
VMware Virtual Machine Instance [cmdb_ci_vmware_instance]	Registered on::Has registered	ESX Server [cmdb_ci_esx_server]	Child
VMware Virtual Machine Instance [cmdb_ci_vmware_instance]	Connected by::Connects	VMware vCenter Network [cmdb_ci_vcenter_network]	Parent
VMware vCenter Network [cmdb_ci_vcenter_network]	Provided by::Provides	ESX Server [cmdb_ci_esx_server]	Parent
VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	Provides storage for::Stored on	VMware Virtual Machine Instance [cmdb_ci_vmware_instance]	Parent
VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	Used by::Uses	ESX Server [cmdb_ci_esx_server]	Child
VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	Members::Member of	ESX Server [cmdb_ci_esx_server]	Child

ESX Resource Pool [cmdb_ci_esx_resource_pool]	Defines resources for::Get resources from	VMware vCenter Cluster [cmdb_ci_vcenter_cluster]	Parent
ESX Resource Pool [cmdb_ci_esx_resource_pool]	Defines resources for::Get resources from	ESX Server [cmdb_ci_esx_server]	Parent
VMware vCenter Folder [cmdb_ci_vcenter_folder]	Contains::Contained by	VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	Parent
VMware vCenter Folder [cmdb_ci_vcenter_folder]	Contains::Contained by	VMware vCenter Folder [cmdb_ci_vcenter_folder]	Parent
VMware vCenter Folder [cmdb_ci_vcenter_folder]	Contains::Contained by	Virtual Machine Template [cmdb_ci_vmware_template]	Parent
VMware vCenter Folder [cmdb_ci_vcenter_folder]	Contains::Contained by	VMware Virtual Machine Instance [cmdb_ci_vmware_instance]	Parent
VMware vCenter Datacenter [cmdb_ci_vcenter_datacenter]	Contains::Contained by	VMware vCenter Network [cmdb_ci_vcenter_network]	Parent
VMware vCenter Datacenter [cmdb_ci_vcenter_datacenter]	Contains::Contained by	VMware Virtual Machine Instance [cmdb_ci_vmware_instance]	Parent
VMware vCenter Datacenter [cmdb_ci_vcenter_datacenter]	Contains::Contained by	ESX Server [cmdb_ci_esx_server]	Parent
VMware vCenter Datacenter [cmdb_ci_vcenter_datacenter]	Contains::Contained by	VMware vCenter Datastore [cmdb_ci_vcenter_datastore]	Parent
VMware vCenter Datacenter [cmdb_ci_vcenter_datacenter]	Contains::Contained by	VMware vCenter Folder [cmdb_ci_vcenter_folder]	Parent

Enhancements

Eureka

- Automatically validates and updates relationships for vCenter CIs.

Calgary

- An Orchestration workflow **Discover vCenter** can populate the CMDB with vCenter data without having to install the Discovery plugin. See Orchestration Configuration and Workflow.

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/concept/c_DiscoveryForVMwareVCenter.html
- [2] <https://docs.servicenow.com/bundle/helsinki-it-operations-management/page/product/discovery/reference/r-discovery.html>

NGINX

Overview

NGINX is an open source web server with a load balancer. Starting in Fuji, Discovery identifies the web server and information related to the load balancer.

Requirements for Discovery

For NGINX servers:

- Ensure that the NGINX software is installed and running on the server.
- Grant the MID Server has access to the NGINX configuration file, which is */etc/nginx/nginx.conf* by default.
- Enable secure shell (SSH) commands to identify the following associated elements:
 - NGINX Version
 - NGINX Get Configuration

The following probes require execute privileges to run commands:

Probe	Commands
Nginx – Version	nginx
Nginx – Get Configuration	echo, sed, httpd, cut, grep, egrep (within the Bourne shell script)

Probes and Sensors

Discovery identifies NGINX server software using the following process:

1. The Nginx Process Classifier detects a running process that matches the following criteria during the exploration of a UNIX server:
 - The name of the process starts with *nginx*.
 - The name of the process contains *master*.
2. If there is a match:
 - A record is created in the Web Server [cmdb_ci_web_server] table.
 - A *Runs on* relationship is created in the CI Relationship [cmdb_rel_ci] table for the Linux Server [cmdb_ci_linux_server] table and the Web Server [cmdb_ci_web_server] table.
3. The following two probes are triggered:
 - **Nginx – Version:** this probe contains a Bourne shell script. It determines the version of NGINX and populates the Web Server [cmdb_ci_web_server] table.
 - **Nginx – Get Configuration:** this probe contains a Bourne shell script and an argument that determines the path of the NGINX configuration file. The probe identifies configuration parameters based on keywords within the configuration file and returns them as a single payload result.
4. The sensor on the ServiceNow instance processes the payload and populates the CMDB.

Data Collected

Discovery creates or updates CMDB records when it detects a running NGINX process. The following data is collected.

Data Collected by Discovery for NGINX

Field Label	Table and Field Name	Probe
Name	cmdb_ci_lb_appl [name]	Nginx– Get Configuration
IP Address	cmdb_ci_lb_appl [ip_address]	Nginx– Get Configuration
Last Discovered	cmdb_ci_lb_appl [last_discovered]	Nginx– Get Configuration
Version	cmdb_ci_lb_appl [version]	Nginx– Version
Name	cmdb_ci_lb_pool_member [name]	Nginx– Get Configuration
Last Discovered	cmdb_ci_lb_pool_member [last_discovered]	Nginx– Get Configuration
IP Address	cmdb_ci_lb_pool_member [ip_address]	Nginx– Get Configuration
Load Balancer	cmdb_ci_lb_pool_member [load_balancer]	Nginx– Get Configuration
Fully Qualified Domain Name	cmdb_ci_lb_pool_member [fqdn]	Nginx– Get Configuration
Port	cmdb_ci_lb_pool_member [port]	Nginx– Get Configuration
Last Discovered	cmdb_ci_lb_service [last_discovered]	Nginx– Get Configuration
IP Address	cmdb_ci_lb_service [ip_address]	Nginx– Get Configuration
Name	cmdb_ci_lb_service [name]	Nginx– Get Configuration
Load Balancer	cmdb_ci_lb_service [load_balancer]	Nginx– Get Configuration
Input URL	cmdb_ci_lb_service [input_url]	NGINX – Get Configuration

Relationships

In addition to data population, the following relationships are created in the CI Relationship [cmdb_rel_ci] table:

- The records in the cmdb_ci_lb_appl table run on the cmdb_ci_web_server table records.
- The records in the cmdb_ci_lb_service table use the cmdb_ci_lb_pool table records.
- The records in the cmdb_ci_pool table are used by the cmdb_ci_service table records.
- The records in the cmdb_ci_pool table are members of the cmdb_ci_pool_member table.
- The records in the cmdb_ci_pool_member table are members of cmdb_ci_pool table.

Puppet Automation Software

Overview

Starting in Fuji, Discovery populates the CMDB when it detects Puppet automation software.

Requirements

By default, Discovery identifies Puppet Masters running on UNIX servers. Discovery uses secure shell (SSH) commands to collect information.

With the addition of the Puppet Configuration Management plugin, Discovery identifies the following associated elements:

- Puppet Certification Requests
- Puppet Manifests
- Puppet Modules

The credentials used to discover the UNIX server must have privileges to execute the following commands:

Probe	Commands
Puppet – Master Info	puppet, echo, hostname (within the Bourne shell script)
Puppet – Certificate Requests	puppet
Puppet – Manifests	echo, sed, find (within the Bourne shell script)
Puppet – Modules	puppet

The use of sudo is supported, but you must add the `must_sudo` parameter to the probe.

Probes

Discovery identifies the Puppet Master using the following process:

1. The **UNIX - Active Processes** probe detects a running process that matches one of the following criteria:
 - The name of the process is *pe-httpd*.
 - The name of the process is *ruby* and the parameters of the process contain *puppet master*.
2. If there is a match on one of these criteria:
 - A record is created in the Puppet Master table `[cmdb_ci_puppet_master]` if one does not already exist for that running process.
 - The following probe is triggered:
 - **Puppet - Master Info:** The sensor of this probe populates additional information in the Puppet Master `[cmdb_ci_puppet_master]` record.
3. With the activation of the **Puppet Configuration Management** plugin, the sensor processing of **Puppet - Master Info** triggers the following simultaneously:
 - **Puppet – Certificate Requests:** The sensor of this probe populates the Puppet Certificate Request `[puppet_certificate_request table]` with open requests. Open requests are requests that are not already signed or rejected.
 - **MultiProbe Puppet – Resources:** This probe contains the following probes:

- **Puppet – Module:** The sensor of this probe populates records within the Puppet Module [puppet_module] table.
- **Puppet – Manifests:** The sensor of this probe populates records within the Puppet Manifest [puppet_manifest], Puppet Class [puppet_class], and Puppet Parameter [puppet_parameter] tables.

Accessing Puppet Masters

Navigate to **Configuration > Automation Servers > Puppet Masters** to view the main Puppet server record, Puppet Master. You can view the Puppet Master resources from the Related Lists tab of the record.

Adding the must_sudo Parameter to the Puppet Probe

To use the Puppet probe with sudo, you must add the *must_sudo* parameter to *each probe that requires it*.

To add the *must_sudo* parameter to the Puppet probe:

1. Navigate to **Discovery > Probes**.
2. In the **Search** field, search on the name **Puppet**.
3. Click the **Puppet Master - Info** probe.
4. In the **Related Links** pane, select the **Probe parameters** tab.
5. Click **New**.
6. Use the following information to fill out the form:
 - **Name:** must_sudo
 - **Value:** true
7. Click **Submit**.
8. Repeat these steps for the other Puppet probes.

Data Collected

The following data is collected by default:

Data Collected by Discovery by Default

Label	Table Name	Field Name	Source
Name	Puppet Master [cmdb_ci_puppet_master]	name	Puppet - Master Info
Configuration directory	Puppet Master [cmdb_ci_puppet_master]	config_directory	Puppet - Master Info
Manifest directory	Puppet Master [cmdb_ci_puppet_master]	manifestdir	Puppet - Master Info
Module path	Puppet Master [cmdb_ci_puppet_master]	modulepath	Puppet - Master Info
Fully qualified domain name	Puppet Master [cmdb_ci_puppet_master]	fqdn	Puppet - Master Info
IP Address	Puppet Master [cmdb_ci_puppet_master]	ip_address	Puppet - Master Info
Version	Puppet Master [cmdb_ci_puppet_master]	version	Puppet - Master Info

The following additional data is collected with the Puppet Configuration Management plugin:

Data Collected by Discovery with the Puppet Configuration Management plugin

Label	Table Name	Field Name	Source
Name	Module [puppet_module]	name	Puppet - Modules
Path	Module [puppet_module]	path	Puppet - Modules
Name	Manifest [puppet_manifest]	name	Puppet - Manifests
Path	Manifest [puppet_manifest]	path	Puppet - Manifests
Content	Manifest [puppet_manifest]	content	Puppet - Manifests
Name	Class [puppet_class]	name	Puppet - Manifests
Inherits class	Class [puppet_class]	inherits	Puppet - Manifests
Selectable	Class [puppet_class]	selectable	Puppet - Manifests
Name	Parameter [puppet_parameter]	name	Puppet - Manifests
Default value	Parameter [puppet_parameter]	default_value	Puppet - Manifests
Name	Certificate Request [puppet_certificate_request]	name	Puppet - Certificate Requests
Node request	Certificate Request [puppet_certificate_request]	node_request	Puppet - Certificate Requests

Other Stuff

Amazon Web Services (AWS) Cloud



Note: This article applies to Fuji and earlier releases. For more current information, see *Discovery Data for an Amazon Web Services (AWS) Cloud* ^[1] at <http://docs.servicenow.com> **The ServiceNow Wiki is no longer being updated. Visit <http://docs.servicenow.com> for the latest product documentation.**

Overview

Starting in Fuji, Discovery can find information about Amazon Web Services. The discovery of Amazon Web Services cloud is based on account information rather than an IP range. MID Servers are not used in this type of discovery.

In AWS, a web service account is a master account that has many subscriptions, where each subscription is a set of login credentials. Each subscription has views into the resources available in the master account to that subscription. To discover the entire web service account, you must have the credentials for each subscription.

A Discovery schedule can discover one or more Amazon web service accounts.

To perform host-based discovery of the virtual hosts contained within an AWS Virtual Private Cloud (VPC):

- A MID server must be installed and configured on a node within the VPC.
- Each VPC that is discovered must have a separate discovery schedule for the IP addresses in that VPC.

See *Installed with Amazon Web Services* for a list of items installed with AWS, including plugins.

Requirements to Discover Amazon Web Services

Amazon Web Services account credentials.

Data Collected

You can view the data that is collected by the AWS Discovery in the following tabs:

1. Navigate to **AWS Discovery > Accounts**.
2. Click the AWS account you would like to view.
3. View the following tabs.
 - EC2 Virtual Machine Instances
 - AWS VPCs
 - AWS Subnets
 - AWS Elastic Load Balancers
 - AWS EBS Volumes
 - AWS Elastic Block Store Snapshots

AWS Auto Scaling Group table

Data Collected by Discovery for AWS Auto Scaling Group table

Label	Table Name	Field Name	Source
Account ID	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	account_id	AWS ASG - DescribeAutoScalingGroups
Auto Scaling group ARN	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	resource_name	AWS ASG - DescribeAutoScalingGroups
Correlation ID	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	correlation_id	AWS ASG - DescribeAutoScalingGroups
Created time	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	created_time	AWS ASG - DescribeAutoScalingGroups
Default cooldown	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	default_cooldown	AWS ASG - DescribeAutoScalingGroups
Desired capacity	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	desired_capacity	AWS ASG - DescribeAutoScalingGroups
Enabled metrics	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	enabled_metrics	AWS ASG - DescribeAutoScalingGroups
Health check grace period	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	health_check_grace_period	AWS ASG - DescribeAutoScalingGroups
Health check type	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	health_check_type	AWS ASG - DescribeAutoScalingGroups
Launch configuration	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	launch_config	AWS ASG - DescribeAutoScalingGroups
Load balancers	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	load_balancers	AWS ASG - DescribeAutoScalingGroups
Max size	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	max_size	AWS ASG - DescribeAutoScalingGroups
Min size	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	min_size	AWS ASG - DescribeAutoScalingGroups
Name	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	name	AWS ASG - DescribeAutoScalingGroups
Region	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	region	AWS ASG - DescribeAutoScalingGroups
Termination policies	AWS Auto Scaling Group [cmdb_ci_aws_asgrp]	termination_policies	AWS ASG - DescribeAutoScalingGroups
VPC zone identifier	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	vpc_zone_identifiers	AWS ASG - DescribeAutoScalingGroups

AWS Auto Scaling Group Launch Config table

Data Collected by Discovery for AWS Auto Scaling Group Launch Config table

Label	Table Name	Field Name	Source
Account ID	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	account_id	AWS ASG - DescribeLaunchConfigurations
Associate Public IP Address	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	associate_public_ip_addr	AWS ASG - DescribeLaunchConfigurations
Block Device Mappings	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	block_device_mappings	AWS ASG - DescribeLaunchConfigurations
Created time	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	created_time	AWS ASG - DescribeLaunchConfigurations
EBS Optimized	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	ebs_optimized	AWS ASG - DescribeLaunchConfigurations
Image ID	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	image_id	AWS ASG - DescribeLaunchConfigurations
Instance Monitoring	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	instance_monitoring	AWS ASG - DescribeLaunchConfigurations
Instance Type	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	instance_type	AWS ASG - DescribeLaunchConfigurations
Kernel ID	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	kernel_id	AWS ASG - DescribeLaunchConfigurations
Key Name	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	key_name	AWS ASG - DescribeLaunchConfigurations
Launch Configuration ARN	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	resource_name	AWS ASG - DescribeLaunchConfigurations
Launch Configuration Name	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	name	AWS ASG - DescribeLaunchConfigurations
Ramdisk ID	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	ramdisk_id	AWS ASG - DescribeLaunchConfigurations
Security Groups	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	security_groups	AWS ASG - DescribeLaunchConfigurations
User data	AWS Auto Scaling Group Launch Config [aws_asgrp_launch_cfg]	user_data	AWS ASG - DescribeLaunchConfigurations

AWS Availability Zone table

Data Collected by Discovery for AWS Availability Zone table

Label	Table Name	Field Name	Source
Account ID	AWS Availability Zone [aws_availability_zone]	account_id	AWS EC2 - DescribeAvailabilityZones
Message	AWS Availability Zone [aws_availability_zone]	message	AWS EC2 - DescribeAvailabilityZones
Region	AWS Availability Zone [aws_availability_zone]	region	AWS EC2 - DescribeAvailabilityZones
Zone name	AWS Availability Zone [aws_availability_zone]	zone_name	AWS EC2 - DescribeAvailabilityZones
Zone state	AWS Availability Zone [aws_availability_zone]	state	AWS EC2 - DescribeAvailabilityZones

AWS EBS Volume table

Data Collected by Discovery for AWS EBS Volume table

Label	Table Name	Field Name	Source
Account ID	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	account_id	AWS EC2 - DescribeVolumes
Assigned to	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	assigned_to	AWS EC2 - DescribeVolumes
Assignment group	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	assignment_group	AWS EC2 - DescribeVolumes
Attachment status	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	attachment_status	AWS EC2 - DescribeVolumes
Availability zone	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	availability_zone	AWS EC2 - DescribeVolumes
Correlation ID	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	correlation_id	AWS EC2 - DescribeVolumes
Delete On Termination	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	delete_on_termination	AWS EC2 - DescribeVolumes
Device Name	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	device_name	AWS EC2 - DescribeVolumes
Instance	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	instance	AWS EC2 - DescribeVolumes
Name	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	name	AWS EC2 - DescribeVolumes
Region	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	region	AWS EC2 - DescribeVolumes
Snapshot	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	snapshot	AWS EC2 - DescribeVolumes
Volume ID	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	volume_id	AWS EC2 - DescribeVolumes
Volume IOPs	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	volume_iops	AWS EC2 - DescribeVolumes
Volume size (GiBs)	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	volume_size	AWS EC2 - DescribeVolumes
Volume status	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	volume_status	AWS EC2 - DescribeVolumes
Volume type	AWS EBS Volume [cmdb_ci_aws_ebs_volume]	volume_type	AWS EC2 - DescribeVolumes

AWS Elastic Block Store Snapshot table

Data Collected by Discovery for AWS Elastic Block Store Snapshot table

Label	Table Name	Field Name	Source
Account ID	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	account_id	AWS EC2 - DescribeSnapshots
Name	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	name	AWS EC2 - DescribeSnapshots
Owner ID	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	owner_id	AWS EC2 - DescribeSnapshots
Progress	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	progress	AWS EC2 - DescribeSnapshots
Region	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	region	AWS EC2 - DescribeSnapshots
Short description	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	short_description	AWS EC2 - DescribeSnapshots
Snapshot ID	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	snapshot_id	AWS EC2 - DescribeSnapshots
Snapshot size (GB)	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	snapshot_size	AWS EC2 - DescribeSnapshots
Start time	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	start_time	AWS EC2 - DescribeSnapshots
State	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	state	AWS EC2 - DescribeSnapshots
Volume ID	AWS Elastic Block Store Snapshot [aws_ebs_snapshot]	volume_id	AWS EC2 - DescribeSnapshots

AWS Elastic Load Balancer table

Data Collected by Discovery for AWS Elastic Load Balancer table

Label	Table Name	Field Name	Source
Account ID	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	account_id	AWS ELB - DescribeLoadBalancers
Canonical hosted zone name	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	canonical_hosted_zone_name	AWS ELB - DescribeLoadBalancers
Canonical hosted zone name ID	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	canonical_hosted_zone_name_id	AWS ELB - DescribeLoadBalancers
Comments	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	comments	AWS ELB - DescribeLoadBalancers
Correlation ID	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	correlation_id	AWS ELB - DescribeLoadBalancers
Created time	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	created_time	AWS ELB - DescribeLoadBalancers
DNS Domain	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	dns_domain	AWS ELB - DescribeLoadBalancers
DNS name	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	dns_name	AWS ELB - DescribeLoadBalancers
Instances	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	instance	AWS ELB - DescribeLoadBalancers
Name	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	name	AWS ELB - DescribeLoadBalancers
Region	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	region	AWS ELB - DescribeLoadBalancers
Scheme	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	scheme	AWS ELB - DescribeLoadBalancers

Security groups	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	security_groups	AWS ELB - DescribeLoadBalancers
Source security groups	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	source_security_group	AWS ELB - DescribeLoadBalancers
Subnets	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	subnet	AWS ELB - DescribeLoadBalancers
VPC	AWS Elastic Load Balancer [cmdb_ci_aws_elb]	vpc	AWS ELB - DescribeLoadBalancers

AWS Resource table

Data Collected by Discovery for AWS Resource table

Label	Table Name	Field Name	Source
Account ID	AWS Resource [cmdb_ci_aws_resource]	account_id	AWS CF - ListStackResources
Logical ID	AWS Resource [cmdb_ci_aws_resource]	logical_id	AWS CF - ListStackResources
Name	AWS Resource [cmdb_ci_aws_resource]	name	AWS CF - ListStackResources
Region	AWS Resource [cmdb_ci_aws_resource]	region	AWS CF - ListStackResources
Resource ID	AWS Resource [cmdb_ci_aws_resource]	resource_id	AWS CF - ListStackResources

AWS Subnet table

Data Collected by Discovery for AWS Subnet table

Label	Table Name	Field Name	Source
Account ID	AWS Subnet [cmdb_ci_aws_subnet]	account_id	AWS EC2 - DescribeSubnets
Availability zone	AWS Subnet [cmdb_ci_aws_subnet]	availability_zone	AWS EC2 - DescribeSubnets
Available IP address count	AWS Subnet [cmdb_ci_aws_subnet]	available_ip_address_count	AWS EC2 - DescribeSubnets
CIDR block	AWS Subnet [cmdb_ci_aws_subnet]	cidr_block	AWS EC2 - DescribeSubnets
Correlation ID	AWS Subnet [cmdb_ci_aws_subnet]	correlation_id	AWS EC2 - DescribeSubnets
Default for region	AWS Subnet [cmdb_ci_aws_subnet]	default_for_az	AWS EC2 - DescribeSubnets
Map public IP on launch	AWS Subnet [cmdb_ci_aws_subnet]	map_public_ip_on_launch	AWS EC2 - DescribeSubnets
Name	AWS Subnet [cmdb_ci_aws_subnet]	name	AWS EC2 - DescribeSubnets
Region	AWS Subnet [cmdb_ci_aws_subnet]	region	AWS EC2 - DescribeSubnets
Subnet ID	AWS Subnet [cmdb_ci_aws_subnet]	subnet_id	AWS EC2 - DescribeSubnets
Subnet state	AWS Subnet [cmdb_ci_aws_subnet]	state	AWS EC2 - DescribeSubnets
VPC	AWS Subnet [cmdb_ci_aws_subnet]	vpc_id	AWS EC2 - DescribeSubnets

AWS VPCs table

Data Collected by Discovery for AWS VPCs table

Label	Table Name	Field Name	Source
Account ID	AWS VPCs [cmdb_ci_aws_vpc]	account_id	AWS EC2 - DescribeVpcs
CIDR block	AWS VPCs [cmdb_ci_aws_vpc]	cidr_block	AWS EC2 - DescribeVpcs
Correlation ID	AWS VPCs [cmdb_ci_aws_vpc]	correlation_id	AWS EC2 - DescribeVpcs
Default	AWS VPCs [cmdb_ci_aws_vpc]	is_default	AWS EC2 - DescribeVpcs
DHCP options ID	AWS VPCs [cmdb_ci_aws_vpc]	dhep_options_id	AWS EC2 - DescribeVpcs
Instance tenancy	AWS VPCs [cmdb_ci_aws_vpc]	instance_tenancy	AWS EC2 - DescribeVpcs
Name	AWS VPCs [cmdb_ci_aws_vpc]	name	AWS EC2 - DescribeVpcs
Region	AWS VPCs [cmdb_ci_aws_vpc]	region	AWS EC2 - DescribeVpcs
VPC ID	AWS VPCs [cmdb_ci_aws_vpc]	vpc_id	AWS EC2 - DescribeVpcs
VPC state	AWS VPCs [cmdb_ci_aws_vpc]	state	AWS EC2 - DescribeVpcs

AWS VPC Security Group table

Data Collected by Discovery for AWS VPC Security Group table

Label	Table Name	Field Name	Source
Account ID	AWS VPC Security Group [aws_vpc_security_group]	account_id	AWS EC2 - DescribeSecurityGroups
Correlation ID	AWS VPC Security Group [aws_vpc_security_group]	correlation_id	AWS EC2 - DescribeSecurityGroups
Description	AWS VPC Security Group [aws_vpc_security_group]	description	AWS EC2 - DescribeSecurityGroups
Group ID	AWS VPC Security Group [aws_vpc_security_group]	group_id	AWS EC2 - DescribeSecurityGroups
Group name	AWS VPC Security Group [aws_vpc_security_group]	group_name	AWS EC2 - DescribeSecurityGroups
Name	AWS VPC Security Group [aws_vpc_security_group]	name	AWS EC2 - DescribeSecurityGroups
Region	AWS VPC Security Group [aws_vpc_security_group]	region	AWS EC2 - DescribeSecurityGroups
VPC	AWS VPC Security Group [aws_vpc_security_group]	vpc	AWS EC2 - DescribeSecurityGroups

EC2 Image table

Data Collected by Discovery for EC2 Image table

Label	Table Name	Field Name	Source
Account ID	EC2 Image [ec2_image]	account_id	AWS EC2 - DescribeImages
Architecture	EC2 Image [ec2_image]	architecture	AWS EC2 - DescribeImages
Description	EC2 Image [ec2_image]	description	AWS EC2 - DescribeImages
Image ID	EC2 Image [ec2_image]	image_id	AWS EC2 - DescribeImages
Image location	EC2 Image [ec2_image]	image_location	AWS EC2 - DescribeImages
Kernel ID	EC2 Image [ec2_image]	kernel_id	AWS EC2 - DescribeImages
Location	EC2 Image [ec2_image]	location	AWS EC2 - DescribeImages
Manufacturer	EC2 Image [ec2_image]	manufacturer	AWS EC2 - DescribeImages
Name	EC2 Image [ec2_image]	name	AWS EC2 - DescribeImages
Platform	EC2 Image [ec2_image]	platform	AWS EC2 - DescribeImages
Public	EC2 Image [ec2_image]	is_public	AWS EC2 - DescribeImages
Ramdisk ID	EC2 Image [ec2_image]	ramdisk_id	AWS EC2 - DescribeImages
Region	EC2 Image [ec2_image]	region	AWS EC2 - DescribeImages
Root device type	EC2 Image [ec2_image]	root_device_type	AWS EC2 - DescribeImages
State	EC2 Image [ec2_image]	state	AWS EC2 - DescribeImages
Virtualization	EC2 Image [ec2_image]	virtualization	AWS EC2 - DescribeImages

EC2 Key Pairs table**Data Collected by Discovery for EC2 Key Pairs table**

Label	Table Name	Field Name	Source
Account ID	EC2 Key Pairs [ec2_keypairs]	account_id	AWS EC2 - DescribeKeyPairs
Finger print	EC2 Key Pairs [ec2_keypairs]	finger_print	AWS EC2 - DescribeKeyPairs
Name	EC2 Key Pairs [ec2_keypairs]	name	AWS EC2 - DescribeKeyPairs
Region	EC2 Key Pairs [ec2_keypairs]	region	AWS EC2 - DescribeKeyPairs

EC2 Virtual Machine Instance table

Data Collected by Discovery for EC2 Virtual Machine Instance table

Label	Table Name	Field Name	Source
Account ID	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	account_id	AWS EC2 - DescribeInstances
Availability zone	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	availability_zone	AWS EC2 - DescribeInstances
Category	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	category	AWS EC2 - DescribeInstances
Correlation ID	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	correlation_id	AWS EC2 - DescribeInstances
DNS Domain	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	dns_domain	AWS EC2 - DescribeInstances
IP Address	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	ip_address	AWS EC2 - DescribeInstances
Key pair	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	key_pair	AWS EC2 - DescribeInstances
Manufacturer	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	manufacturer	AWS EC2 - DescribeInstances
Name	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	name	AWS EC2 - DescribeInstances
Region	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	region	AWS EC2 - DescribeInstances
Root Device Name	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	root_device_name	AWS EC2 - DescribeInstances
Root Device Type	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	root_device_type	AWS EC2 - DescribeInstances
State	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	state	AWS EC2 - DescribeInstances
Subnet ID	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	subnet_id	AWS EC2 - DescribeInstances
VPC ID	EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	vpc_id	AWS EC2 - DescribeInstances

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/reference/r_DataCollectedOnAWScloud.html

Services/Daemons

Data Collected by Discovery on Services and Daemons

Label	Table Name	Field Name	Source
Name	cmdb_ip_service	name	Various
Port	cmdb_ip_service	port	Various
Description	cmdb_ip_service_ci	description	Various
CI	cmdb_ip_service_ci	ci	Internal reference
Service	cmdb_ip_service_ci	service	Internal reference

IP Networks

Data Collected by Discovery on IP Networks

Label	Table Name	Field Name	Source
Discover	cmdb_ci_ip_network	discover	Various internal
Subnet	cmdb_ci_ip_network	subnet	Various internal
Network discovery	cmdb_ci_ip_network	network_discovery	Various internal
Last discovered	cmdb_ci_ip_network	last_discovered	Various internal
MID server	cmdb_ci_ip_network	mid_server	Various internal
Router	cmdb_ci_ip_network	router	Various internal
State	cmdb_ci_ip_network	state	Various internal

Relationships

Overview

Discovery can find the following information about relationships between CIs:

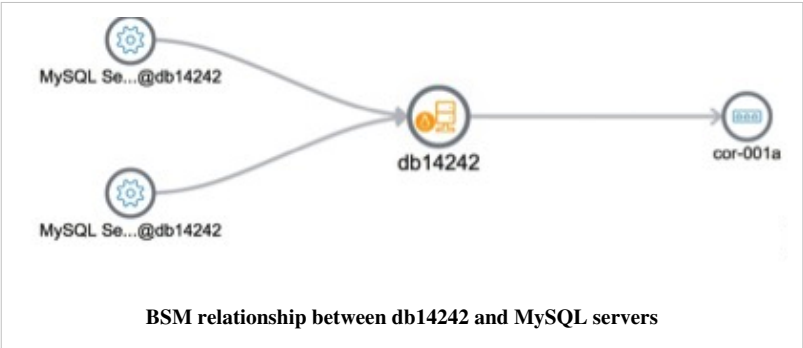
Data Collected by Discovery on Relationships

Label	Table Name	Field Name	Source
Parent	cmdb_rel_ci	parent	Internal
Child	cmdb_rel_ci	child	Internal
Type	cmdb_rel_ci	type	Internal

Discovery for Servers and Network devices

Starting in Fuji, the Layer 3 BSM Discovery feature will map Layer 3 relationships between a server and other network devices. The relationship begins at the router or switch and shows relationship and IP address information for associated servers and network devices.

The system property glide.discovery.L3_mapping toggles the Layer 3 BSM discovery of these devices. When glide.discovery.L3_mapping is *true*, the relationship created is of type IP Connection::IP Connection. In this example, the BSM resulting from the discovery of Linux Server db14242 shows that it has an IP Connection::IP Connection relationship to switch cor-001a.



Prerequisites

- The server or network device needs to have IP Address information.
- The system property glide.discovery.L3_mapping is set to true to discover routers and switches.
- The router or Layer 3 switch that provides the IP Address needs to

have been successfully discovered with populated Exit Interface Routing Rules.

System Properties

Property	Description
glide.discovery.L3_mapping	<p>Starting in Fuji, provides a logical mapping of the TCP/IP layer for network gears. This is not Layer 2 mapping.</p> <ul style="list-style-type: none">• Type: string• Default value:true• Location: The System Property [sys_properties] table

Connections

Tables and Fields

Discovery stores information about TCP connections in the following tables and fields.

Label	Table Name	Field Name	Source
Absent	cmdb_tcp	absent	Internal
Computer	cmdb_tcp	computer	Internal
IP	cmdb_tcp_connection	ip	Internal
PID	cmdb_tcp	pid	Internal
Port	cmdb_tcp	port	Internal
Process	cmdb_tcp	process	Internal
Type	cmdb_tcp	type	Internal

Versions Prior to the Dublin Release

Discovery stores information about TCP connections in the following tables and fields on releases prior to Dublin.

View tables and fields on versions prior to Dublin

Label	Table Name	Field Name	Source
Application	cmdb_tcp_half	application	Internal
Computer	cmdb_tcp_half	computer	Internal
PID	cmdb_tcp_half	pid	Internal
Command	cmdb_tcp_half	command	Internal
From IP	cmdb_tcp_half	from_ip	Internal
From port	cmdb_tcp_half	from_port	Internal
To IP	cmdb_tcp_half	to_ip	Internal
To port	cmdb_tcp_half	to_port	Internal
Type	cmdb_tcp_half	type	Internal
Computer	cmdb_tcp_connection	computer	Internal
PID	cmdb_tcp_connection	pid	Internal
Command	cmdb_tcp_connection	command	Internal
From IP	cmdb_tcp_connection	from_ip	Internal
From port	cmdb_tcp_connection	from_port	Internal
To IP	cmdb_tcp_connection	to_ip	Internal
To port	cmdb_tcp_connection	to_port	Internal
State	cmdb_tcp_connection	state	Internal

Enhancements

Dublin

- The Running Processes [cmdb_running_process] table contains new fields (listening_to and connecting_to) to track the list of TCP ports a process listens on and connects to.
- A new table TCP Connections [cmdb_tcp] replaces the cmdb_tcp_connection and cmdb_tcp_half tables.
- The Application Instance [cmdb_ci_appl] table contains new fields to track the classifier that created the instance record and the running process that matched the classifier.

F5 BIG-IP



Note: This article applies to Fuji. For more current information, see *Load Balancer: F5 BIG-IP* ^[1] at <http://docs.servicenow.com>. The ServiceNow Wiki is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

Overview

Discovery of F5 BIG-IP load balancers is performed via SNMP. VMware images of BIG-IP for testing are available with free 90-day keys: <https://www.f5.com/trial>.

Discovery for F5 BIG-IP load balancers is available starting with the Dublin release.

Source of Load Balancer Data

Discovery uses the following SNMP MIBs to collect data for F5 BIG-IP load balancers:

- F5-BIGIP-COMMON-MIB
- F5-BIGIP-LOCAL-MIB
- F5-BIGIP-SYSTEM-MIB

For more information see SNMP Probe.

Model

The F5 BIG-IP load balancer model represents a generic load balancer and its components. The abstract class is Load Balancer (cmdb_ci_lb). The implementation class, extended from Load Balancer, is F5 BIG-IP (cmdb_ci_lb_bigip). The load balancer components are modeled as follows.

Component	Table Name †	Description
Load Balancer Service	cmdb_ci_lb_service	A virtual service that the device balances by forwarding requests to members within a pool.
Load Balancer Pool	cmdb_ci_lb_pool	A collection of host-to-port mappings to be balanced.
Load Balancer Pool Member	cmdb_ci_lb_pool_member	A host-to-port mapping of a request to be balanced.
Load Balancer VLAN	cmdb_ci_lb_vlan	A virtual LAN segment.
Load Balancer Interface	cmdb_ci_lb_interface	A network interface.

† On these tables, the field referring to the parent cmdb_ci_lb has been updated to also allow references to cmdb_ci_lb_appl, starting with Fuji.

Relationships

An F5 discovery creates relationships between the application and the load balance service if the application CI exists. However, if the application CI was not discovered, the F5 BIG-IP sensor will map between the computer and the load balance service instead.

In this example, load balancer lba-001a has multiple services. The vip_repo_sea1-http service distributes the Apache Server on Port 80. Because there are three services (vip_ns_internal.sea1, vip_ldaps.sea1, vip_ldap.sea1) where the sensor could not locate the application, relationships to the Linux server ops01 are created instead.



References

[1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/concept/c_LoadBalancerF5BIGIP.html

Dell DRAC

Overview

The Dell™ Remote Assistant Card (DRAC) provides users with tools and functionality to monitor, troubleshoot, and repair servers. Discovery can detect a DRAC devices and populate the CMDB with pertinent information, starting in Fuji.

To view Dell DRAC information, navigate to **Configuration > Base Items > Out-of-Band Devices**.

Discovery Requirements

For Dell DRAC:

- Configure the device for SNMP communications. The SNMP – DRAC probe supports SNMPv1 and SMNPv2c.
- Add SNMP credentials to Discovery Credentials.
- Ensure that the Dell DRAC device has an SNMP Object Identifier (OID) *sysDescr* value that matches **1.3.6.1.4.1.674.10892.2** or **1.3.6.1.4.1.674.10892.5**.

Probes and Sensors

By default, during a CI discovery, Discovery identifies Dell DRAC devices while finding network devices that communicate with SNMP. Discovery identifies Dell DRAC devices using the following process:

- During a CI Discovery, a network device communicating via SNMP is detected.
- During SNMP Classification, the Shazzam probe obtains the **sysDescr** value to classify the CI as Dell DRAC device.
- The SNMP - DRAC probe is triggered to explore the device.

Data Collected

For the Dell DRAC, the following data is collected:

Data Collected by Discovery for Dell DRAC

Label	Table Name	Field Name	Source
Firmware version	cmdb_ci_outofband_device	firmware_version	SNMP walk: drsFirmwareVersion (racFirmwareVersion for iDRAC7)
Host	cmdb_ci_outofband_device	host	SNMP walk: drsSystemServiceTag (systemServiceTag for iDRAC7) *
IP Address	cmdb_ci_outofband_device	ip_address	DNS probe
Name	cmdb_ci_outofband_device	name	SNMP - Identity probe
Product version	cmdb_ci_outofband_device	product_version	SNMP walk: drsProductVersion (racVersion for iDRAC7)
Type	cmdb_ci_outofband_device	type	SNMP walk: drsProductType (racType for iDRAC7)
URL	cmdb_ci_outofband_device	url	SNMP walk: drsProductURL (racURL for iDRAC7)

* Host is a reference to the `cmdb_ci_computer` table via the serial number. Therefore, in order for the `cmdb_ci_outofband_device.host` field to be populated correctly, the host machine needs to be discoverable or exist within the CMDB with the appropriate serial number.

HAProxy

Overview

HAProxy is an open-source load balancer that can manage any TCP service. It is particularly suited for HTTP load balancing because it supports session persistence and Layer 7 processing. Discovery supports HAProxy for HTTP load balancing. However, TCP load-balancing is not supported.

Requirements for Discovery

Consider the following requirements for discovering the HAProxy:

- The HAProxy software is installed and running on a Linux server.
- The MID Server is deployed to explore the server and the MID Server has access to the server HAProxy configuration file.
- The configuration probe checks for the `haproxy.cfg` file using one of the following methods:
 - Using the `f` parameter for the HAProxy process output.
 - Using the default `/etc/haproxy/haproxy.cfg` path.
- The following probes require credentials to execute commands:

Probe	Commands
HAProxy – Version	haproxy
HAProxy – Get Configuration	echo, sed, cut, grep, egrep (within the Bourne shell script)

Probes

The probes and sensors operate in the following manner:

1. The UNIX - Active Processes probe detects a running process that matches the following criteria:
 - The name of the process is `haproxy`.
2. If there is a match on this criterion:
 - A record is created in the Load Balancer Application [`cmdb_ci_lb_appl`] table if one does not already exist for that running process.
3. The following probes are triggered:
 - **HAProxy – Get Configuration:** the sensor of this probe populates additional information in the Load Balancer Application [`cmdb_ci_lb_appl`] table. The probe also populates information in the Load Balancer Service [`cmdb_ci_lb_service`], Load Balancer Pool [`cmdb_ci_lb_pool`] and Load Balancer Pool Member [`cmdb_ci_lb_pool_member`] tables.
 - **HAProxy – Version:** the sensor of this probe populates the HAProxy version in the Load Balancer Application [`cmdb_ci_lb_appl`] table.

Data Collected

Discovery creates or updates CMDB records when it detects a running HAProxy process. The following data is collected.

Data Collected by Discovery for HAProxy

Table Name	Field and Label Name	Probe
cmdb_ci_lb_appl	Name [name]	HAProxy – Get Configuration
cmdb_ci_lb_appl	IP Address [ip_address]	HAProxy – Get Configuration
cmdb_ci_lb_appl	Last Discovered [last_discovered]	HAProxy – Get Configuration
cmdb_ci_lb_appl	Version [version]	HAProxy – Get Configuration
cmdb_ci_lb_pool_member	Name [name]	HAProxy – Get Configuration
cmdb_ci_lb_pool_member	Last Discovered [last_discovered]	HAProxy – Get Configuration
cmdb_ci_lb_pool_member	IP Address [ip_address]	HAProxy – Get Configuration
cmdb_ci_lb_pool_member	Load Balancer [load_balancer]	HAProxy – Get Configuration
cmdb_ci_lb_pool_member	Port [port]	HAProxy – Get Configuration
cmdb_ci_lb_pool_member	Fully Qualified Domain Name [fdqn]	HAProxy – Get Configuration
cmdb_ci_lb_service	Last Port [last_port]	HAProxy – Get Configuration
cmdb_ci_lb_service	Last Discovered [last_discovered]	HAProxy – Get Configuration
cmdb_ci_lb_service	IP Address [ip_address]	HAProxy – Get Configuration
cmdb_ci_lb_service	Name [name]	HAProxy – Get Configuration
cmdb_ci_lb_service	Load Balancer [load_balancer]	HAProxy – Get Configuration
cmdb_ci_lb_service	Port [port]	HAProxy – Get Configuration
cmdb_ci_lb_pool	Last Discovered [last_discovered]	HAProxy – Get Configuration
cmdb_ci_lb_pool	Load Balancing Method [load_balancing_method]	HAProxy – Get Configuration
cmdb_ci_lb_pool	Load Balancer [load_balancer]	HAProxy – Get Configuration
cmdb_ci_lb_pool	Name [name]	HAProxy – Get Configuration

Relationships

In addition to populating the data, the following relationships records are created in Ci Relationships [cmdb_rel_ci] table:

- The records in the cmdb_ci_lb_appl table run on the cmdb_ci_web_server table records.
- The records in the cmdb_ci_lb_service table use the cmdb_ci_lb_pool table records.
- The records in the cmdb_ci_pool table are used by the cmdb_ci_service table records.
- The records in the cmdb_ci_pool table are members of the cmdb_ci_pool_member table records.
- The records in the cmdb_ci_pool_member table are members of the cmdb_ci_pool table records.

Storage Devices



Note: This article applies to Fuji and earlier releases. For more current information, see *Storage Discovery*^[1] at <http://docs.servicenow.com>. **The ServiceNow Wiki is no longer being updated. Visit <http://docs.servicenow.com> for the latest product documentation.**

Overview

Discovery finds and maps dependencies for the following types of storage:

- Direct-attached storage (DAS), network-attached storage (NAS), or storage area network (SAN) storage that is discovered via a host.
- NAS or SAN storage that is discovered via a Storage Management Initiative Specification (SMI-S) provider and Common Information Model (CIM).
- Virtual storage for VMware ESX servers and Linux Kernel-based Virtual Machines (KVM). Discovery maps this storage to the underlying physical storage.

Discovery can identify SAN storage devices that use Fibre Channel (FC) or Internet Small Computer System Interface (iSCSI).

The Discovery storage data model has been updated to standardize how storage is represented in the CMDB, starting with the Fuji release. For more information, see Fuji enhancements.

Enhancements

Fuji

- The ability to discover Fibre Channel disks, pools, volumes, and adapters as well as the dependencies between each CI. **Note:** Fibre Channel is not supported on Windows 2012.
- The Disk [cmdb_ci_disk] table extends the new Storage Device [cmdb_ci_storage_device] table. The original Disk table now represents physical devices only. Child tables now represent each type of storage device.
 - Fibre Channel Disk [cmdb_ci_fc_disk] table.
 - iSCSI Disk [cmdb_ci_iscsi_disk] table.
- The File System table [cmdb_ci_file_system] was changed to be a more generic Storage Volume table [cmdb_ci_storage_volume].
- The storage array probes no longer create relationships between every subcomponent. Instead, they follow the standardized storage model when creating CIs.
- The **Size** and **Size in Bytes** fields are normalized across Disk, Volume, and Pool tables.
- The **Free-space** and **Free-space in Bytes** are fields normalized on Disk and Pool tables.

Click to view a list of potential customer impact of these changes:

- Changes to UI Forms and Lists. UI Forms and Lists have been updated to reflect the model changes on Disk and File System. Remember that you will not receive any updates for forms or lists you have configured when your system is upgraded.
 - Deprecated fields due to the model changes are still populated in Fuji. During an upgrade, these fields are migrated to the new fields.
 - Storage Arrays now use reference fields to map dependencies between internal components. Previously relationships were used. Relationships are now created only between independent systems. The customer will still see pre-existing relationships after an upgrade.
 - In Linux systems, configuration of the `/etc/sudoers` file may need more commands added (such as `dmsetup`) to access list.
-

Eureka

- The Disk [cmdb_ci_disk] table extends the Configuration Item[extended cmdb_ci] table.

References

- [1] https://docs.servicenow.com/bundle/jakarta-it-operations-management/page/product/discovery/concept/c_Storage.html

Article Sources and Contributors

Data Collected by Discovery *Source:* <http://wiki.servicenow.com/index.php?oldid=250474> *Contributors:* CapaJC, Dawn.bunting, Doogiesd, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, Joseph.messerschmidt, Phillip.salzman, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

Windows *Source:* <http://wiki.servicenow.com/index.php?oldid=246434> *Contributors:* Aleck.lin, G.yedwab, Joseph.messerschmidt, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

Solaris *Source:* <http://wiki.servicenow.com/index.php?oldid=245474> *Contributors:* Aleck.lin, Dawn.bunting, Fuji.publishing.user, Julie.phaviseth, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Linux *Source:* <http://wiki.servicenow.com/index.php?oldid=190269> *Contributors:* Aleck.lin, George.rawlins, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Linux Kernel-based Virtual Machines (KVM) *Source:* <http://wiki.servicenow.com/index.php?oldid=189789> *Contributors:* Vaughn.romero

HPUX *Source:* <http://wiki.servicenow.com/index.php?oldid=160154> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

AIX *Source:* <http://wiki.servicenow.com/index.php?oldid=246878> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

Mac *Source:* <http://wiki.servicenow.com/index.php?oldid=191195> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Solaris Zones *Source:* <http://wiki.servicenow.com/index.php?oldid=160162> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Netware *Source:* <http://wiki.servicenow.com/index.php?oldid=160163> *Contributors:* Steven.wood, Vaughn.romero

ESX Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=191305> *Contributors:* Dawn.bunting, Joe.zucker, John.ramos, Joseph.messerschmidt, Phillip.salzman, Steven.wood, Vaughn.romero, Virginia.kelley

Routers *Source:* <http://wiki.servicenow.com/index.php?oldid=250477> *Contributors:* Dawn.bunting, Fuji.publishing.user, George.rawlins, Ishrath.razvi, John.ramos, Joseph.messerschmidt, Ludwig.adriaansen, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

Switches *Source:* <http://wiki.servicenow.com/index.php?oldid=250477> *Contributors:* Dawn.bunting, Fuji.publishing.user, George.rawlins, Ishrath.razvi, John.ramos, Joseph.messerschmidt, Ludwig.adriaansen, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

Network Printers *Source:* <http://wiki.servicenow.com/index.php?oldid=160168> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Uninterruptible Power Supplies (UPSs) *Source:* <http://wiki.servicenow.com/index.php?oldid=160170> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Common Information Model (CIM) storage devices *Source:* <http://wiki.servicenow.com/index.php?oldid=250096> *Contributors:* John.ramos, Joseph.messerschmidt, Rachel.sienko, Steven.wood, Vaughn.romero

Apache Web Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=246647> *Contributors:* Dawn.bunting, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

JBoss Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=247171> *Contributors:* Dawn.bunting, Fuji.publishing.user, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

MySQL Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=246596> *Contributors:* Fuji.publishing.user, Shankara.meenkeri, Virginia.kelley

PostgreSQL Instances *Source:* <http://wiki.servicenow.com/index.php?oldid=246597> *Contributors:* Dawn.bunting, Fuji.publishing.user, Virginia.kelley

Oracle Databases *Source:* <http://wiki.servicenow.com/index.php?oldid=240484> *Contributors:* Dawn.bunting, Fuji.publishing.user, John.ramos, Phillip.salzman, Virginia.kelley

HBase *Source:* <http://wiki.servicenow.com/index.php?oldid=247152> *Contributors:* Dawn.bunting, Fuji.publishing.user, Virginia.kelley

MongoDB Instances *Source:* <http://wiki.servicenow.com/index.php?oldid=246595> *Contributors:* Dawn.bunting, Fuji.publishing.user, Virginia.kelley

Tomcat Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=240494> *Contributors:* Dawn.bunting, Fuji.publishing.user, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Weblogic Application Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=247166> *Contributors:* Dawn.bunting, Fuji.publishing.user, John.ramos, Virginia.kelley

Websphere Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=247169> *Contributors:* Dawn.bunting, Fuji.publishing.user, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

Microsoft IIS Servers *Source:* <http://wiki.servicenow.com/index.php?oldid=203911> *Contributors:* Doogiesd, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

General Software Packages *Source:* <http://wiki.servicenow.com/index.php?oldid=160189> *Contributors:* Doogiesd, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

VMware vCenter *Source:* <http://wiki.servicenow.com/index.php?oldid=250531> *Contributors:* Cheryl.dolan, Dawn.bunting, Fuji.publishing.user, Ishrath.razvi, Joe.zucker, John.ramos, Joseph.messerschmidt, Josh.gerbasi, Phillip.salzman, Steven.wood, Vaughn.romero

NGINX *Source:* <http://wiki.servicenow.com/index.php?oldid=247611> *Contributors:* Dawn.bunting, Fuji.publishing.user, Vaughn.romero, Virginia.kelley

Puppet Automation Software *Source:* <http://wiki.servicenow.com/index.php?oldid=246598> *Contributors:* Dawn.bunting, Fuji.publishing.user, Virginia.kelley

Amazon Web Services (AWS) Cloud *Source:* <http://wiki.servicenow.com/index.php?oldid=251177> *Contributors:* Dawn.bunting, Fuji.publishing.user, John.ramos, Phillip.salzman, Virginia.kelley

Services/Daemons *Source:* <http://wiki.servicenow.com/index.php?oldid=160192> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

IP Networks *Source:* <http://wiki.servicenow.com/index.php?oldid=160194> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

Relationships *Source:* <http://wiki.servicenow.com/index.php?oldid=249373> *Contributors:* Phillip.salzman, Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne, Virginia.kelley

Connections *Source:* <http://wiki.servicenow.com/index.php?oldid=191200> *Contributors:* Steven.wood, Tom.dilatush, Vaughn.romero, Vhearne

F5 BIG-IP *Source:* <http://wiki.servicenow.com/index.php?oldid=250475> *Contributors:* John.ramos, Vaughn.romero, Virginia.kelley

Dell DRAC *Source:* <http://wiki.servicenow.com/index.php?oldid=247233> *Contributors:* Dawn.bunting, Fuji.publishing.user, Vaughn.romero, Virginia.kelley

HAProxy *Source:* <http://wiki.servicenow.com/index.php?oldid=246440> *Contributors:* Dawn.bunting, Fuji.publishing.user, Virginia.kelley

Storage Devices *Source:* <http://wiki.servicenow.com/index.php?oldid=250522> *Contributors:* Dawn.bunting, Fuji.publishing.user, John.ramos, Phillip.salzman, Steven.wood, Virginia.kelley

Image Sources, Licenses and Contributors

Image:Warning.gif *Source:* <http://wiki.servicenow.com/index.php?title=File:Warning.gif> *License:* unknown *Contributors:* CapaJC

Image:Knowledge.gif *Source:* <http://wiki.servicenow.com/index.php?title=File:Knowledge.gif> *License:* unknown *Contributors:* G.yedwab, Joseph.messerschmidt, Publishing.user

Image:CIM_SMI-S_Standard_Diagram.png *Source:* http://wiki.servicenow.com/index.php?title=File:CIM_SMI-S_Standard_Diagram.png *License:* unknown *Contributors:* Steven.wood

Image:CIM_Agents_Diagram.png *Source:* http://wiki.servicenow.com/index.php?title=File:CIM_Agents_Diagram.png *License:* unknown *Contributors:* Joseph.messerschmidt, Steven.wood

Image:CIM_Shazzam_Processing_Diagram.png *Source:* http://wiki.servicenow.com/index.php?title=File:CIM_Shazzam_Processing_Diagram.png *License:* unknown *Contributors:* Steven.wood

Image:CIM_wbem_Port_Probe.png *Source:* http://wiki.servicenow.com/index.php?title=File:CIM_wbem_Port_Probe.png *License:* unknown *Contributors:* Steven.wood

Image:CIM_SLP_Query.png *Source:* http://wiki.servicenow.com/index.php?title=File:CIM_SLP_Query.png *License:* unknown *Contributors:* Steven.wood

Image:Shazzam_with_CIM_Diagram.png *Source:* http://wiki.servicenow.com/index.php?title=File:Shazzam_with_CIM_Diagram.png *License:* unknown *Contributors:* Steven.wood

Image:CIM_Classify_Probe.png *Source:* http://wiki.servicenow.com/index.php?title=File:CIM_Classify_Probe.png *License:* unknown *Contributors:* Steven.wood

File:Discover_vCenter_UIAction.png *Source:* http://wiki.servicenow.com/index.php?title=File:Discover_vCenter_UIAction.png *License:* unknown *Contributors:* Steven.wood

File:Discover_vCenter_Message.png *Source:* http://wiki.servicenow.com/index.php?title=File:Discover_vCenter_Message.png *License:* unknown *Contributors:* Steven.wood

File:Scheduled_Discovery.png *Source:* http://wiki.servicenow.com/index.php?title=File:Scheduled_Discovery.png *License:* unknown *Contributors:* Steven.wood

Image:VCenterAlternatePortParameter.png *Source:* <http://wiki.servicenow.com/index.php?title=File:VCenterAlternatePortParameter.png> *License:* unknown *Contributors:* Fuji.publishing.user

Image:Layer3_BSM.png *Source:* http://wiki.servicenow.com/index.php?title=File:Layer3_BSM.png *License:* unknown *Contributors:* Virginia.kelley

Image:F5_LB_BSM.png *Source:* http://wiki.servicenow.com/index.php?title=File:F5_LB_BSM.png *License:* unknown *Contributors:* Virginia.kelley