



Inventarisierung VMWare Backend 2017

Inhaltsverzeichnis

Installation	3
Konfiguration	4

Installation

Vorraussetzungen

- KIX Professional 17 - aktuelle Version
- Ein installiertes vSphere Perl SDK für vSphere 5.5+ auf dem KIX-Server
- Einen vCenter-Nutzer mit read-only Rechten (gewöhnlich ist ein Nutzer mit der im Datacenter level zugewiesenen Standard "read-only"-Rolle ausreichend)

Paket Installation

Installieren Sie das Paket "*kixpro-inventorysync-vmware*" mit der "*Paketverwaltung*" (Menü "*Admin*" Bereich "*Systemverwaltung*") oder über die Kommandozeile.

Konfiguration

Grundlagen

Das VMWare-Backend arbeitet basierend auf Objekttypen. Jeder Objekttyp hat seinen eigenen Satz von Attributen und liefert die Daten, die zu den ConfigItem-Klassenattributen in der ConfigItem-Klassendefinition gemappt werden.

Es ist immer notwendig, einen Objekttyp zuzuordnen, entweder auf der Ebene der Datenquelle selbst oder auf der Ebene des ConfigItem-Klassennamen-Mappings oder basierend auf den Namen der ConfigItem-Klasse selbst.

Die folgenden Objekttypen sind möglich:

- Datacenter
- Datastore
- Cluster
- Host
- VM

Beispiel:

Wenn Sie eine ConfigItem-Klasse mit dem Namen *"VMWare-Servers"* haben, müssen Sie den Objekttyp zuordnen, weil dies die Daten für ein VMWare-Hostsystem bereit stellt.

- Möglichkeit 1: Auf der Ebene der Datenquelle

Definieren Sie eine Datenquelle und fügen Sie *";ObjectType=Host"* an das Ende der Parameterzeile in der SysConfig-Einstellung: *ITSMConfigItem::InventorySync###Parameters*.

- Möglichkeit 2: Definieren eines Klassennamen-Mapping

Erzeugen Sie ein ConfigItem-Klassenmapping, wo der Schlüssel *"VMWare-Servers"* lautet und der Wert den Objekttyp definiert.

SysConfig-Einstellung: *InventorySync::Backend::VMWare###CIClassObjectTypeMapping* .

- Möglichkeit 2: Die ConfigItem-Klasse umbenennen

Wenn Sie nicht die Möglichkeit 1 oder 2 nutzen möchten, bleibt noch die Möglichkeit, die ConfigItem-Klasse von *"VMWare-Servers"* umzubenennen in *"Host"*. In diesem Fall repräsentiert der Name der ConfigItem-Klasse den zu nutzenden Objekttyp.

SysConfig-Optionen

Um die Einstellungen für diese Funktion zu konfigurieren, wählen Sie in der SysConfig die Gruppe "KIX Professional". Nachdem die Seite neu geladen wurde, wählen Sie die Untergruppe "ITSMConfigItem::InventorySync".

Source

- SysConfig-Einstellung: *ITSMConfigItem::InventorySync###Sources*

ITSMConfigItem::InventorySync###Sources	
Schlüssel	Inhalt
<input type="text"/>	<input type="text"/>

Definition der Quellen für die externen Inventardaten. Der Schlüssel wird für alle folgenden Einstellungen benötigt. Schlüsselbezeichnung kurz halten.

Eine Quelle benötigt einen benannten Kommunikationspunkt. Zur Konfiguration wird ein Hash benutzt. Der Hash-Schlüssel ist der interne Name der Quelle. bspw. "VMWare". Der Hash-Wert ist der Anzeigename für diese Quelle, bspw. "my VMWare environment". Sie können mehrere Quellen für jeden externen Inventory-Server hinzufügen.

Source-Backend-Mapping

- SysConfig-Einstellung: *ITSMConfigItem::InventorySync###Backend*


ITSMConfigItem::InventorySync###Backend	
Schlüssel	Inhalt
<input type="text"/>	

Definiert das zu nutzende Backend für die Inventarquelle an (entspricht dem Modulnamen im Backend-Verzeichnis).

Diese Einstellung definiert das zu nutzende Backend. Zur Konfiguration wird ein Hash benutzt. Der Hash-Schlüssel ist der Identifikator der Quelle (in unserem Beispiel "VMWare"). Der Hash-Wert muss mit "VMWare" angegeben werden, damit das Modul InventorySync das VMWare-Backend für diese Datenquelle benutzt.

Source-Parameter

- SysConfig-Einstellung: *ITSMConfigItem::InventorySync###Parameters*

<input checked="" type="checkbox"/> ITSMConfigItem::InventorySync###Parameters Specify the parameters of the inventory data source (backend specific).		
Schlüssel	Inhalt	
		

In dieser Einstellung konfigurieren Sie die Parameter zur Nutzung des VMWare-Backends. Alle Parameter müssen in der folgenden Form angegeben werden:

<parameter>=<value>,<value>,...

Parameter	Benötigt	Beschreibung
URL	x	Der URL zum Webservice-Interface des vCenter servers (gewöhnlich http://<host>:443/sdk/webService)
User	x	Nutzername des Datenbank-Nutzers.
Password	x	Passwort des Datenbank-Nutzers.
ObjectType		Optionaler Parameter, um die Datenquelle bezüglich eines speziellen Objekttyps einzuschränken. Mögliche Werte sind: Datacenter Datastore Cluster Host VM

Alternatives ObjectType mapping

- SysConfig-Einstellung: *InventorySync::Backend::VMWare###CIClassObjectTypeMapping*

Diese Einstellung finden Sie in der Untergruppe "ITSM::ConfigItem::InventorySync::VMWare". Wenn in der Datenquelle selbst keine Restriktionen bezüglich eines bestimmten Objekttyps konfiguriert sind, können Sie hier den zu nutzenden Objekttyp für jede ConfigItem-Klasse zuordnen. Der Schlüssel des Hashs repräsentiert den Name der ConfigItem-Klasse und der Wert repräsentiert den Objekttyp. Folgende Objekttypen sind möglich:

- *Datacenter*
- *Datastore*
- *Cluster*
- *Host*
- *VM*

ObjectType fallback

Wenn weder die Datenquelle auf einen bestimmten Objekttyp beschränkt wurde, noch ein klassenbasiertes Mapping existiert, wird der Name der CI-Klasse vom System zur Bestimmung des Objekttyps verwendet.

SSL Verification

SysConfig-Untergruppe *"ITSMConfigItem::InventorySync::opsi*

- SysConfig-Einstellung: *InventorySync::Backend::VMWare###UserAgentVerifyHostname*

Mit dieser Einstellung können Sie die SSL-Zertifikatsprüfung für den konfigurierten Host ein- bzw. ausschalten.

InventoryContentXPath-Struktur

Die Erweiterung von ConfigItem-Klassen und die Nutzung der InventoryContentXPath Parameter werden im Kapitel *"KIX SysConfig/KIX Professional/InventorySync"* beschrieben.

Der XPath für VMWare hat die folgende Struktur:

- Hardware XPath: *HW/<Container>/<Attribute>*
- Software XPath: Derzeit existiert kein Software XPath

Welche Attribute und Container verfügbar sind, hängt von den Rechten des Nutzers ab, den Sie zur Verbindung zum vCenter-Server benutzen. Weiterhin hängt es ab vom Status der übereinstimmenden Objekte. Es werden nicht immer alle dieser Attribute verfügbar sein.

Gewöhnlich sind die folgenden Attribute verfügbar:

ObjectType	Container	Attribute
Datacenter	Info	configStatus
		numVMDKs
		numHosts
		numClusters
		numDatastores
		numVMs
		overallStatus
	Clusters	name
	Datastores	name
Host	name	
Networks	name	
Datastore	Info	LUN
		build
		capacity
		capacityGB
		capacityKB
		capacityMB
		freeSpace
		freeSpaceGB
		freeSpaceKB

ObjectType	Container	Attribute
		freeSpaceMB
		instanceUuid
		isSSD
		numHosts
		numVMs
		type
		url
		version
		vmfsVersion
		Hosts
VMs	name	
Cluster	Info	actionHistory
		alarmActionsEnabled
		apiType
		apiVersion
		build
		configStatus
		disableMethod
		effectiveCpuGhz
		effectiveCpuMhz
		effectiveMemory
		effectiveMemoryGB
		effectiveMemoryKB
		effectiveMemoryMB
		fullName
		hBDatastoreCandidatePolicy
		instanceUuid

ObjectType	Container	Attribute	
		licenseProductName	
		licenseProductVersion	
		localeBuild	
		localeVersion	
		name	
		numCpuCores	
		numCpuThreads	
		numDatastores	
		numEffective Hosts	
		numHosts	
		numVMs	
		osType	
		overallStatus	
		productLineId	
		resourcePool	
		totalCpuGhz	
		totalCpuMhz	
		totalMemory	
		totalMemoryKB	
		totalMemoryMB	
		totalMemoryGB	
		vendor	
		version	
		HA	admissionControl
			configuredFailoverCapacity
			dasConfig
			dasIsolationAddress

ObjectType	Container	Attribute
		defaultVmSettings.dasIsolationResponse
		defaultVmSettings.restartPriority
		defaultVmSettings.vmToolsMonitoringSettings
		defaultVmSettings.vmToolsMonitoringSettings.clusterSettings
		defaultVmSettings.vmToolsMonitoringSettings.failureInterval
		defaultVmSettings.vmToolsMonitoringSettings.maxFailureWindow
		defaultVmSettings.vmToolsMonitoringSettings.maxFailures
		defaultVmSettings.vmToolsMonitoringSettings.minUpTime
		defaultVmSettings.vmToolsMonitoringSettings.vmMonitoring
		failover.Level
		hostMonitoring
		status
		vmMonitoring
	DRS	defaultVmBehavior
		status
	Datastores	extent.diskName
		extent.partition
		name
		url
	Hosts	name
VMs	name	
Hosts	Info	bootTime
		build
		cluster

ObjectType	Container	Attribute
		datacenter
		DNS.dhcp
		DNS.hostname
		DNS.nameserver
		DNS.searchDomains
		defaultGateway
		cpuMhz
		cpuGhz
		cpuModel
		datacenter
		fullname
		hostMaxVirtualDiskCapacity
		hostMaxVirtualDiskCapacityKB
		hostMaxVirtualDiskCapacityMB
		hostMaxVirtualDiskCapacityGB
		licenseProductName
		memorySizeGB
		memorySizeKB
		memorySizeMB
		model
		name
		numCpuCores
		num CpuPkgs
		numCpuThreads
		numDatastores
		numHBAs
		numNICs

ObjectType	Container	Attribute
		numVMs
		port
		powerState
		uptime
		uuid
		vendor
		version
		vmotion.ipAdress
		vmotion.subnetMask
	vSwitches	mtu
		name
		numPorts
		numPortsAvailable
		portgroup.name
		portgroup.usedPorts
		portgroup.vlanId
		usedPorts
	NICs	autoNegotiateSupported
		dhcp
		driver
		duplex
		ipAdress
		macAdress
		name
		pci
		resourcePoolSchedulerAllowed
		speedMb

ObjectType	Container	Attribute
		subnetMask
		wakeOnLanSupported
	VMs	name
VM	Info	cluster
		faultToleranceState
		guestId
		guestMemoryUsage
		guestMemoryUsageKB
		guestMemoryUsageMB
		guestMemoryUsageGB
		guestOS
		host
		hostMemoryUsage
		hostMemoryUsageKB
		hostMemoryUsageMB
		hostMemoryUsageGB
		host
		instanceUuid
		ipAddress
		isTemplate
		memorySize
		memorySizeKB
		memorySizeMB
		memorySizeGB
		name
numCPU		
numEthernetCards		

ObjectType	Container	Attribute
		numVirtualDisks
		overallStatus
		resourcePool
		storageFree
		storageFreeKB
		storageFreeMB
		storageFreeGB
		storageTotal
		storageTotalKB
		storageTotalMB
		storageTotalGB
		storageUsed
		storageUsedKB
		storageUsedMB
		storageUsedGB
		toolsStatus
		uptime
		vmPathName
	vmState	
	CDRoms	label
		summary
	Datastores	name
	VDisks	backingObjectId
		capacity
		capacityKB
		capacityMB
		capacityGB

ObjectType	Container	Attribute
		datastore
		diskMode
		filename
		label
		size
		type
		uuid
		writeTrough
	VNICs	ipAddress
		label
		macAddress
		network
	Snapshots	createTime
		description
		name
		quiesced
		state

Bitte beachten: Die vCenter API definiert Virtual Machine Templates ebenfalls als Virtual Machine objects. Ob eine VM ein Template ist oder nicht, können Sie feststellen, wenn Sie auf das Attribut "isTemplate" schauen.

Beispiel ConfigItem-Klassen für das VMWare-Backend

Das Paket enthält Beispiel ConfigItem-Klassendefinitionen für alle Objekttypen. Diese Beispiele finden Sie im Verzeichnis:

`/opt/kix/var/packagesetup/InventorySyncVMWareBackend`

Bitte beachten Sie, dass dies nur Beispiele sind, die alle verfügbaren Attribute enthalten, inklusive einiger Attribute mit Nutzungswerten (bspw. aktueller Speicher oder Prozessorauslastung) oder verschiedener Einheiten (kB,MB,GB,Mhz, Ghz,...) . Sie können diese Definitionen nach Ihren Wünschen anpassen.

Wichtig: Bitte beachten Sie, dass gemappte Nutzungswerte gewöhnlich eine neue Version eines ConfigItem erzeugen, weil sie häufigen Änderungen unterliegen. Sie sollten sie aus Ihrer ConfigItem-Klassendefinition entfernen, wenn diese Werte in Ihrer CMDB nicht von Interesse sind.

Sie müssen ebenfalls sicherstellen, dass zwingend erforderliche ConfigItem-Attribute während des Synchronisationsprozesses Werte erhalten. Andernfalls wird durch KIX Professional eine Fehlermeldung angezeigt, weil die Attribute keinen Wert haben.

Diese Beispiele nutzen das ConfigItem-Attribut "*CICReference*", das bereits in KIX enthalten ist. Es ist sehr hilfreich, um automatisch Links zwischen verschiedenen ConfigItems zu erzeugen.

Um die Vorteile dieser Konfigurationsbeispiele voll nutzen zu können, sollten Sie zunächst importieren und anschließend synchronisieren, damit alle Objekte im Synchronisationsprozess bereits verfügbar sind und somit verknüpft werden können.