正在收集SNMP集成数据

OnCommand Insight

NetApp
April 01, 2024

This PDF was generated from https://docs.netapp.com/zh-cn/oncommand-insight/howto/importing-snmp-integration-data.html on April 01, 2024. Always check docs.netapp.com for the latest.

目录

正在收集SNMP集成数据 1
集成软件包 1
集成文件格式 1
导入SNMP集成软件包 2
创建SNMP集成数据源 3
integration.json文件信息 5

正在收集SNMP集成数据

通过集成SNMP数据源、您可以在OnCommand Insight 中收集通用SNMP数据。

集成软件包

SNMP集成数据源使用"`Integration Pack`"来定义收集的集成值以及提供这些值的SNMP对象。

集成软件包包括：

- 一个JSON配置文件(integration.json)、根据特定设备类型(交换机、路由器等)的SNMP对象定义集成有效负载内容。
- 集成软件包所依赖的MIB文件列表。

一个集成软件包可以定义多种数据类型。例如、在集成RHEL主机时、可以为正常运行时间、用户数和正在运行的进程数等常规系统信息定义数据类型、还可以为内存和文件系统使用情况数据定义第二种数据类型。通常、每个数据类型都必须为"`flat`"、并且不能包含嵌套数据。

一个集成软件包不应定义24种以上的数据类型。Insight会限制收集的集成数据量。如果尝试在一分钟内载入24个以上的报告、则会导致速率错误。

集成类型的名称必须遵循以下规则：

- 名称不能以以下字符开头：_、-或+
- 名称不能包含以下字符：#、\、/、*、？ 、 "、<、>、|、"、 `、
- 长度不能超过100个UTF-8编码字节
- 不能命名为。或。

集成文件格式

集成软件包是一个ZIP文件、其中包含一个JSON配置文件(integration.json)、用于根据SNMP对象定义集成有效负载内容。它还包含一个MIB文件夹、其中包含所有MIB文件及其MIB依赖项。

。 `integration.json` 文件必须位于ZIP文件的顶层、MIB文件必须位于ZIP中的"resources/mibs"子目录中。如果需要、此ZIP文件还可能包含readme.txt等文件。集成ZIP结构的一个示例是：

```
.
├── integration.json
└── resources
    └── mibs
        ├── IANAifType-MIB.txt
        ├── IF-MIB.txt
        ├── INET-ADDRESS-MIB.txt
        ├── IP-MIB.txt
        ├── RFC1155-SMI.txt
        ├── RFC-1212.txt
        ├── RFC1213-MIB.txt
        ├── SNMPv2-CONF.txt
        ├── SNMPv2-MIB.txt
        ├── SNMPv2-SMI.txt
        ├── SNMPv2-TC.txt
        └── TCP-MIB.txt
```

导入SNMP集成软件包

您可以使用Web UI将SNMP集成软件包导入到OnCommand Insight 中。集成软件包通过中定义的"integrationPackName"值进行标识 `integration.json` ZIP文件中包含的配置文件。

开始之前

您必须已创建格式正确的ZIP文件、其中包含要导入到OnCommand Insight 服务器的集成软件包。

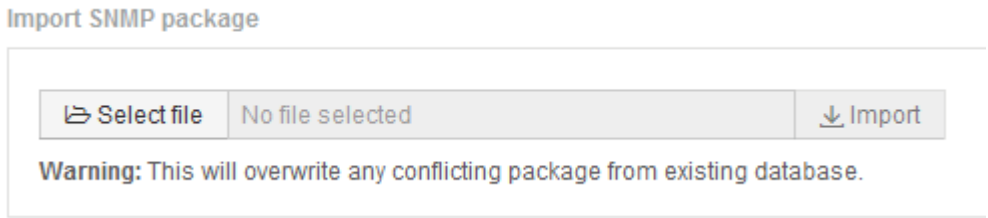
关于此任务

按照以下步骤将SNMP集成软件包导入到Insight服务器中。

步骤

1. 单击*管理*>*设置*>* SNMP集成*

   系统将显示导入SNMP软件包屏幕：

2. 单击*选择文件*以选择包含SNMP软件包的本地文件。

   您选择的文件将显示在文件框中。

任何同名的现有集成软件包都会被覆盖。

3. 单击*导入*

   此文件将导入到Insight服务器中。

创建SNMP集成数据源

集成SNMP数据源可提供与Brocade和Cisco的OnCommand Insight 数据源附带的其他基于SNMP的数据源类似的通用SNMP配置属性。

开始之前

要成功使用集成SNMP数据源进行收集、必须满足以下条件：

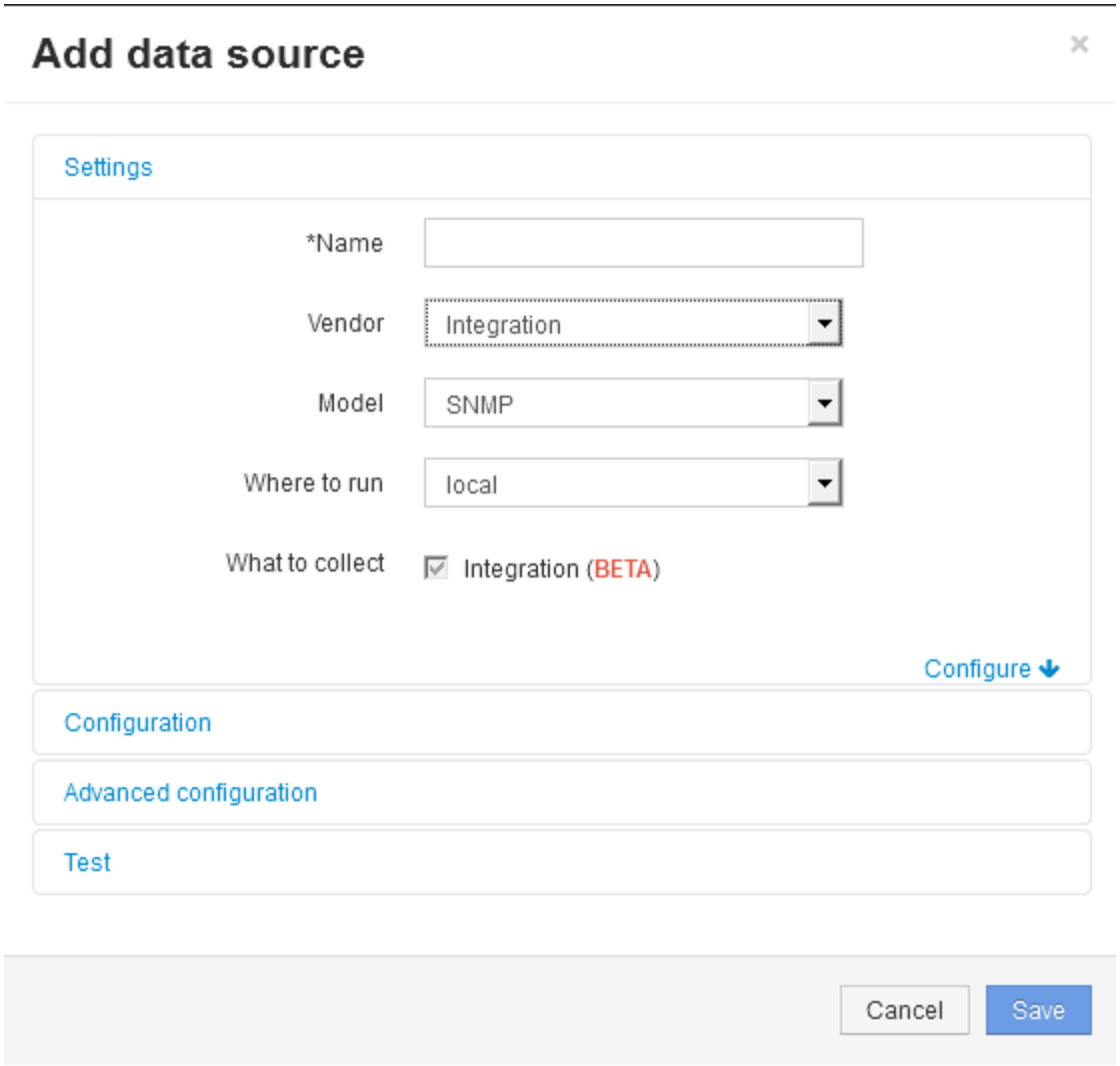
- 您必须已导入要用于此SNMP数据源的集成软件包。
- 所有目标设备都共享相同的凭据。
- 所有目标设备都会实施所配置的集成软件包引用的SNMP对象。

关于此任务

要创建SNMP集成数据源、请在数据源创建向导中选择供应商"Integration"和型号"SNMP"。

步骤

1. 在OnCommand Insight Web UI中、单击*管理*>*数据源*
2. 单击 * + 添加 *
3. 输入数据源的名称
4. 对于Vendor、选择*集成*
5. 对于"Model"、选择"* SNMP*"



6. 有关要收集的内容、请选中*集成*

   这是此数据源上的唯一软件包、默认情况下会进行检查：

7. 单击*配置*
8. 输入要从中收集SNMP数据的系统的IP地址
9. 选择导入的SNMP集成软件包
10. 设置集成轮询间隔
11. 选择SNMP版本
12. 输入SNMP社区字符串

    适用于SNMP V1和V2。

13. 为要从中收集数据的系统添加用户名和密码。

    对于SNMP V3。

14. 单击*高级配置*

此时将显示高级配置默认设置。根据需要对这些设置进行任何更改。

integration.json文件信息

integration.json文件用于标识有效负载。

下图提供了一个简单integration.json文件的颜色编码表示。随附的表标识了文件中对象的功能。

```
{
 "integrationPackName": "WindowsSnmp",
 "description": "Generic integration for mibs supported by the default
SNMP  Agent for Windows 2012, including HOST-RESOURCES",
 "acquisitionType": "SNMP",
 "integrationTypes": [
   {
    "integrationType":      "snmp_win2012_host",
    "name": {
     "mibModuleName":     "RFC1213-MIB",
     "objectName":       "sysName"
    },
    "identifiers": {
     "hostname": {
      "mibModuleName":     "RFC1213-MIB",
     }
    },
    "attributes": {
     "description": {
      "mibModuleName":     "RFC1213-MIB",
      "objectName":       "sysDescr"
     },
     "snmp_sys_obj_id": {
      "mibModuleName":     "RFC1213-MIB",
      "objectName":       "sysObjectID"
     }
    },
    "dataPoints": {
     "uptime": {
      "num": {
       "mibModuleName":   "RFC1213-MIB",
       "objectName":      "sysUpTime"
      }
     }
    }
   }]}
```

Blue	Reserved
Red	User customizable strings and IDs
Green	MIB names
Purple	MIB object
Black	JSON structure

关于integration.json文件

每个字段都具有以下特征：

- "identifiers"部分是一个唯一的复合密钥、用于在Insight中创建新的"object"
- "attributes"提供了有关对象的支持元数据。

  在这两种情况下、仅会保留该对象(通过标识符标识)的最新报告值。

- "dataPoints"是时间序列数据、必须为数值。Insight会将此处报告的每个值保留90天(默认情况下)、并将其时间序列与标识的对象关联起来。

数字表达式

默认情况下、所有值表达式均在集成有效负载中报告为字符串。"identifiers"和"attributes"只能定义字符串值。"dataPoints"可以定义字符串或数字值。数值使用以下修饰符键之一进行定义：

- num—自上次初始化计数器以来接收的总字节数
- delta—在轮询间隔期间接收的字节数
- rate—轮询间隔期间的平均接收速率、以每秒字节为单位

  可以通过速率和数学运算的组合来实现轮询间隔期间的平均接收速率(以MB/秒为单位)

数学运算

。 `integration.json` 文件支持以下数学运算：加、减、乘、除。以下示例显示了JSON文件中的乘法、除法和求和操作。

```
"network_utilization":
{
  "mult": [
    {
      "div": [
        {
          "sum": [
            "rate": {
              "mibModuleName":  "IF-MIB",
              "objectName":     "ifHCOutOctets",
              "comment":        "bytes per second out"
            },
            "rate": {
              "mibModuleName":  "IF-MIB",
              "objectName":     "ifHCInOctets",
              "comment":        "bytes per second in"
            }
          ]
        },
        {
          "num": {
            "mibModuleName":  "IF-MIB",
            "objectName":     "ifSpeed",
            "comment":        "1,000,000 bits per second"
          }
        }
      ]
    },
    {
      "const":                0.0008,
      "comment":        "normalize to ratio of bits and convert to percent:
                        8 * 100 / 1,000,000 = 0.0008"
    }
  ]
}
```

关键字

集成软件包关键字string用于强制将通常以十六进制格式呈现的八位字节字符串或从八位字节字符串派生的专有类型呈现为ASCII字符。

八位字节字符串通常包含二进制数据、例如MAC地址和WWN：

```
"interface_mac": {
          "mibModuleName":       "IF-MIB",
          "objectName":          "ifPhysAddress"
         }
```

ifPhysAddress类型为PhysAddress、它只是一个八位字节字符串：

```
  PhysAddress ::= TEXTUAL-CONVENTION
                DISPLAY-HINT "1x:"
                STATUS       current
                DESCRIPTION
                                "Represents media- or physical-level
addresses."
                SYNTAX       OCTET STRING
```

如果ifPhysAddress默认呈现为十六进制、则结果为：

```
"interface_mac": "00:50:56:A2:07:E7"
```

但是、如果要将八位字节字符串或从八位字节字符串派生的专有类型解释为ASCII、则可以使用"string"关键字：

```
        "string_test_1": {
          "string": {
            "mibModuleName":      "IF-MIB",
            "objectName":         "ifPhysAddress"
          }
        },

        "string_test_2": {
          "string": [
            {
              "mibModuleName":      "IF-MIB",
              "objectName":         "ifPhysAddress"
            },
            {
              "const": "JSD"
            },
            {
              "mibModuleName":      "IF-MIB",
              "objectName":         "ifPhysAddress"
            }
          ]
        }
```

此关键字遵循现有字符串串联规则、在以下示例中的术语之间插入一个空格：

```
      "string_test_1": "PV¢ç",
              "string_test_2": "PV¢ç JSD PV¢ç"
```

"string"关键字对单个术语或一系列术语起作用、但不对嵌套表达式起作用。只有DataPoint表达式才支持嵌套表达式。如果尝试在dataPoint表达式中使用"string"表达式、则会出现类似以下内容的错误：

_java.lang.IllegalArgumentException：集成软件包"GenericSwitch32"索引"nmp_generic_interface_32"部分"ataPoints"键"tring_test_3"不受支持的JSON数字表达式" {"string"： {"mibModuleName"："if-mib"、"objectName"："PhysiftName"}

某些派生的八位字节字符串类型(例如DisplayString、SnmpAdminString)的优先级为硬编码、高于"string"关键字。这是因为SnmpAdminString是专门采用UTF-8编码的、我们希望正确处理它、而"string"关键字则强制使用snmp_framework返回的默认字符串表示形式、即假定每个字符具有单字节ASCII代码点。

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。