DDN ExaScaler ZenPack

Support

This ZenPack is developed by DDN for modelling and monitoring of DDN's ExaScaler storage solution.

Releases

Version: 0.0.7
Summary of changes: First release
Released on: 04/03/2015
Compatible with: Zenoss 4 & zenoss 5

Background

DDN ExaScaler Zenpack will provide modelling and monitoring functionality for DDN's ExaScaler storage solutions.

Prerequisites

1. This zenpack works only on Exascaler version 1.6 and above, since it uses DDN exascaler apis for collecting metrics on each cluster node.
2. This zenpack has a dependency on custom script get_lustre_config.py. This should be placed in this path /cm/shared/ddn/dm/exascaler/get_lustre_config.py for zenpack to model properly. get_lustre_config.py file is available in the src folder.

Installed Items

Installing the ZenPack will add the following items to your Zenoss system.

Device Classes

Following device classes will be created once this ZenPack is installed.

- /Storage
- /Storage/DDN
- /Storage/DDN/Exascaler

Configuration Properties

Following configuration properties to be added which is required for this ZenPack.

- **zCommandUsername**
  - name of the user through which Zenoss system communicates with the device (Ex: **root**).

- **zCommandPassword**
  - password of that particular user through which Zenoss system communicates with the device (Ex: **root**).
- **ZKeyPath**
  - If passwordless SSH is configured provide the full path for private key file. (Ex: **~/.ssh/id_rsa**).

Modeler Plugins

List of modeler plugins for Exascaler.

- ddn.ModelExaScaler

Monitoring Templates

Defines the metrics, events and thresholds for modeled components.

Component Level

1. **MetaDataServer**
   a. Close
   b. Getattr
   c. Link
   d. MetaOps
   e. Mkdir
   f. Mknod
   g. Open
   h. Rename
   i. ReqActive
   j. ReqQdepth
   k. ReqWaittime
   l. Rmdir
   m. Setattr
   n. Statfs
   o. Unlink

2. **ObjectStorageServer**
    a. create
    b. destroy
    c. ReadBytes
    d. Statfs
    e. WriteBytes

Graphs

Component Level

1. **MetaDataServer**
    a. MetaOps
2. **ObjectStorageServer**
    a. IO
    b. MetaOps

Events

Component Level

1. **MetaDataServer**
    a. Events for metadata server will be get triggered when the metadata server status is not in PASS state.
2. **ObjectStorageServer**
    a. Events for object storage server will be get triggered when the object storage server status is not in PASS state.

Events Classes

- /Perf

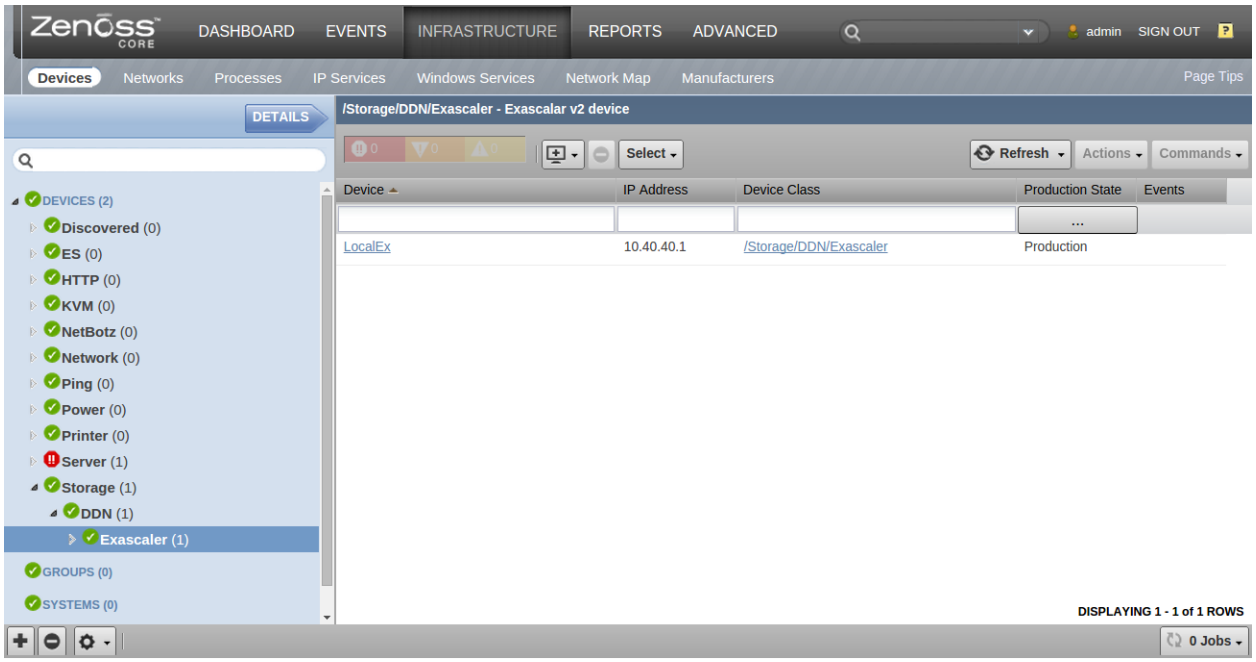
Thresholds

No Thresholds defined.

Detailed Overview

Device class

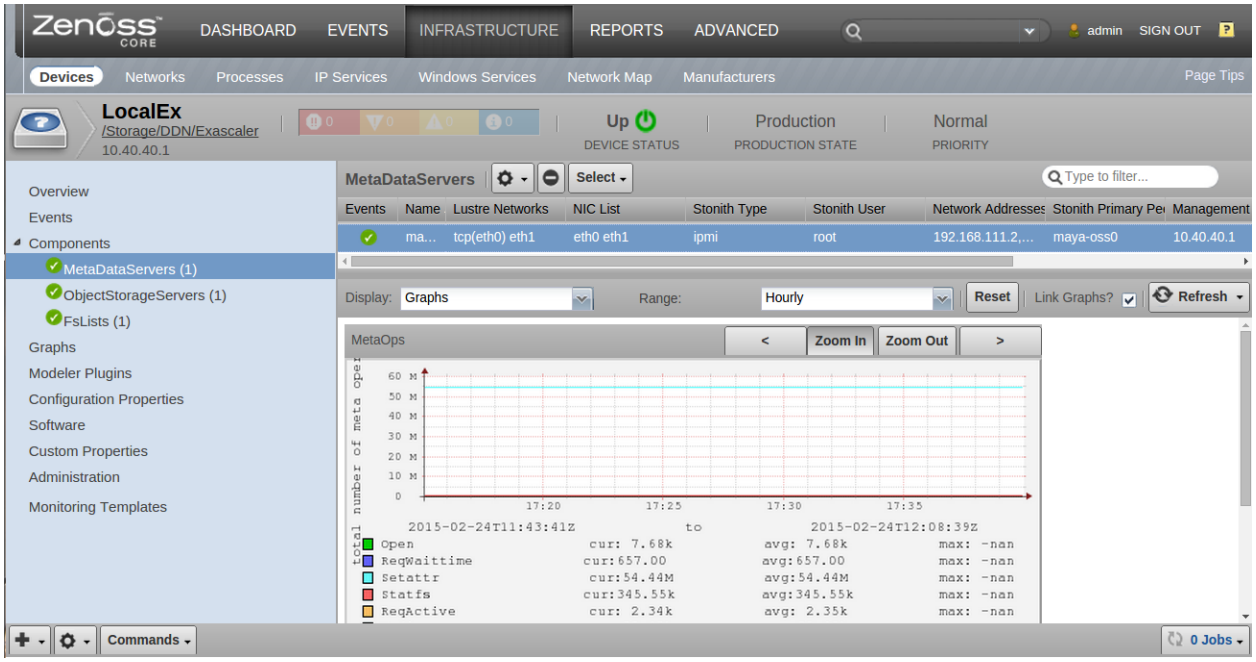
Below screenshot give the list of device class available in this ZenPack.



Device Components

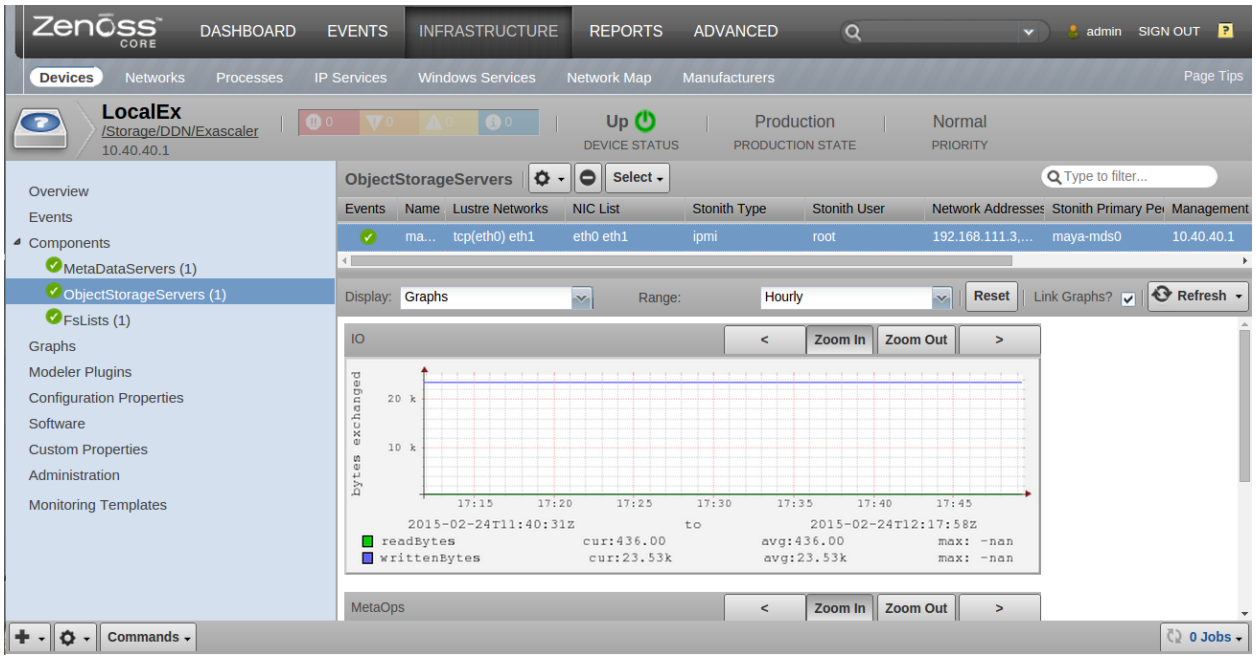
1. List of MetaData Server

Below screenshot give the list of ExaScaler MetaData Server in this ZenPack.



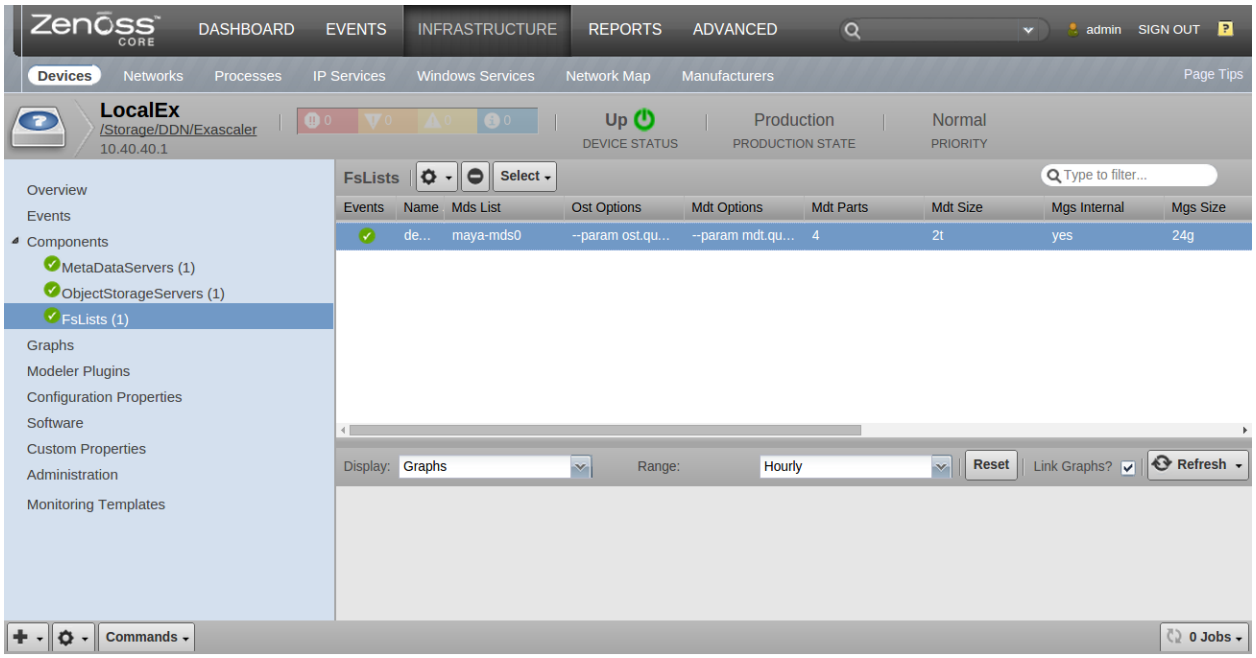
2. List of ObjectStorageServer

Below screenshot give the list of ExaScaler Object Storage Server in this ZenPack.



3.FsLists

Below screenshot give the list of ExaScaler FsLists in this ZenPack.



ZenPack Installation

1. Download the appropriate egg file for the version of the Zenoss you are running.
2. Ensure you are logged in as Zenoss user.
   >>> su - zenoss

3. Install ZenPack

   a. Zenoss 4
      >>> zenpack -- install ZenPacks.DDN.Exascaler-*-*.egg

   b. Zenoss 5
      >>> sudo serviced service run zenpack install ZenPacks.DDN.Exascaler-*-*.egg

4. Restart Zenoss
   a. Zenoss 4
      >>> zenoss restart
   b.Zenoss 5
      >>> sudo serviced service stop <service name>
            [Get the service name by running command "sudo serviced service status"]
      >>> sudo serviced service start <service name>

To list all installed ZenPacks

a. Zenoss 4
   >>> zenpack -- list
b.Zenoss 5
   >>> sudo serviced service run zope zenpack list

Steps to uninstall ZenPacks

a. Zenoss 4
   >>> zenpack -- remove <zen pack>
b.Zenoss 5
   >>> sudo serviced service run zope zenpack uninstall <zen pack>

Instructions to model Exascaler through Zenoss

A Exascaler solution is a cluster of devices. So it cannot be modeled and monitored like other devices. This zenpack expects a pseudo network device to be created locally. This pseudo network device shall be used to register the cluster. Two zProperty (‘zESMdsNodes’,’zESOssNodes’) is defined to map the network address of MetaData Server and Object Storage Server in the cluster.

Follow the below instructions to model a Exascaler solution through zenoss:

1. Create a pseudo network device:
    a. sudo ip link add link em1 address 44:44:44:44:44:44 em1:10 type macvlan
    b. sudo ifconfig em1:10 10.1.1.4 netmask 255.255.224.0

    Note : Change the ip and MAC address accordingly
2. Create a device using this newly created IP (ie. 10.1.1.4) or you can choose to use any local private IP instead of 10.1.1.4 .
3. Update the zProperty accordingly before modeling. For ex. if you have 2 MetaData Server reachable at address (IP1, IP2) [management interfaces] and 2 Object Storage Server reachable at address (IP3, IP4) [management interfaces], provide all those ip on appropriate zProperty (ie. ‘zESMdsNodes’ and ’zESOssNodes’) will be used for device modeling.