Virtualization Release 1.0 - Release notes

Contents

Document Creation

Author: Jeff Walker

Release Version: 1.0

Date: 3/31/17

Technical Announcements

- VMware dropping support for all 3rd party Virtual Switches. VMware intends to deprecate the APIs used by third party virtual switches, and expects to remove the APIs as part of a future update to VMware vSphere 6.5 as well as future releases of the vSphere platform. Customers are encouraged to begin migrating from third party vSwitches including Cisco Nexus 1000V, Cisco VM-FEX, Cisco AVS, HPE 5900v and IBM DVS 5000v to vSphere Distributed Switch. VMware will continue to support the 3rd party virtual switch APIs, up to VMware vSphere 6.5 Update 1.
- VSAN Licensing now includes vDS as it is the recommended virtual switch based on technical feature set to support VSAN traffic.

Helpful/Interesting Resources

https://blogs.vmware.com/virtualblocks/2017/01/19/designing-vsan-networks-use-distributed-switch/

https://blogs.vmware.com/virtualblocks/2017/03/23/cisco-ucs-blades-powered-vmware-vsan/

Release Versioning

Overview

With each release the versions of the major components of the environment will be defined. Not every release will contain new versions.

Release Versions

Product	N-1	N
UCS Firmware	2.2(8f)(legacy 6100 generation)	3.1(2e)(future state)
ESXi Hypervisor	5.5 U3	6.0.0 U2 (3620759)
vCenter Server	5.5 U3	6.0.0 U2 (3634793)
vRealize Operations Manager	---	6.4.0 (4506154)
vSphere Replication	5.8.0.2	6.1.1 (3849281)
Site Recovery Manager	5.8.0.2	6.1.1 (3884620)
VSAN	6.0	6.2
vRealize Log Insight	---	4.3.0

Virtual Machine Hardware Version	8-10	11

Upgrade Timing and Future Version Direction Expectations

There are no hard dates set regarding implementation of release versions, but as a general guideline, if TLM releases follow a quarterly cycle, it shouldn't be unreasonable that in timeframe of a cycle that the environment be upgraded. It should be understood that the Virtualization TLM will not be targeting back to back release version changes. All things considered, the Virtualization team should understand that by year end vSphere 6.5 and related components is the target version. Getting all components to the versions defined in release 1.0 is critical to ensure the supported upgrade path and plan is available from Release 1.0 version to the 6.5 versions release sometime later this year.

Upgrade Sequence

Overview

Per the Update Sequence for vSphere 6.0 documentation in the provided link, the Virtualization team's update sequence is as follows for the Virtualization Infrastructure:

1. Cisco UCS Infrastructure Firmware
2. Cisco UCS Server Firmware (staged)
3. External vCenter SSO (this is not applicable in all locations)
4. vRA (cloud team; ensure their version supports target vSphere versions)
5. vCenter Server
6. vSphere Replication
7. VUM
8. vRealize Operations Manager
9. Site Recovery Manager
10. ESXi & UCS Server Firmware
11. Upgrade VSAN/On Disk format to v3
12. VM Hardware Version

Resources

VSAN Upgrade Requirements:

https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2145248

vSphere Upgrade Sequence:

https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2109760

Cormac Hogan VSAN Upgrade Steps:

http://cormachogan.com/2016/04/05/vsan-6-2-part-12-vsan-6-1-6-2-upgrade-steps/

Gitlab

Access

Understanding the basics of Git and Git Repositories will be crucial for all automation and script management moving forward for the Virtualization space. All TLM releases for Virtualization will be distributed to the Operations team through Git.

Access to Gitlab is open to all employees, using your ad credentials will allow you to log in. You can see any project that is open publically, the “virtualization-automation” project is the project owned by the Virtualization TLM. You can contribute to the project, any additions to the project will be added in a branch and that branch will have to be submitted for a merge request.

To fully engage with Git it is recommended that you download and install Git for Windows, you can download Git from https://git-scm.com/download/win . Ensure that during installation you select the Windows Explorer Integration options and to associate .git and .sh file typs as it suggests. Accept the remainder of the defaults for the install. After installation you can either launch Git Bash to work with Repos, or you can launch any shell, such as CMD or Powershell and the Git commands should be available.

Having Git installed on your system allows you the Check Out a Git Repository and work with a local copy. It will also allow you push your changes to Repo to the Gitlab server. To finalize your setup with Gitlab and Git installed, you will need to generate a SSH key locally and paste your public key into Gitlab under your Profile Settings > SSH Keys menu option. This will allow for connectivity to Gitlab without credentials.

URLs

General URL: http://gitlab.dstcorp.net

Virtualization Repo URL: http://gitlab.dstcorp.net/dt203162/virtualization-automation

Virtualization Repo Zip Download: http://gitlab.dstcorp.net/dt203162/virtualization-automation/repository/archive.zip?ref=master

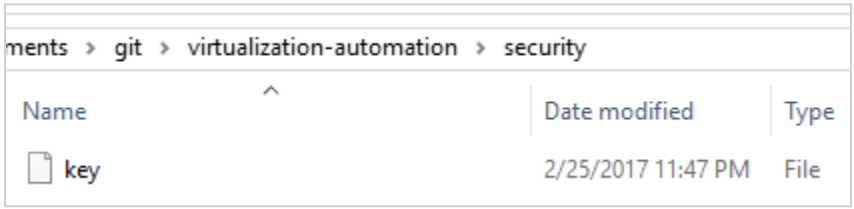
Automation

Automation Usage

After each release, the Virtualization team will need to download the virtualization-automation repo zip and put it on the Automation Server(s), which at the time of this writing is DSKCVMWARENFS. Using the wss-master_vcadmin account, unpackage the Zip container and place the folder structure on the D:\ drive under the “virtualization-automation” folder.

Automation Security Key

Once in place, a critical part to the functionality of the automation in this new structure is that we place a security key inside of the security folder in a file called “key” with no file extension or file type. See screenshot below for clarity:



Once the empty key file is in place, the team will need to retrieve the actual key from the team Keepass database under the Service Accounts folder. The entry in Keepass is called “key”, you will copy the key exactly as is from

the password field, and place it into the empty key files on the automation server. Ensure that the key occupies only the first line in the file and that there are no lingering blank spaces trailing the key. This key is used to decrypt the passwords for accounts used in the automated scripts.

Configuring the Automation Jobs

With the Repository's contents and key now in place on the automation server, the Virtualization team will need to review to the "virtualization-automation/config/virt_config_standards.json" description file. In this file all of the supported automation jobs for the release are described along with their corresponding script and schedule to run on. Ensure these automated jobs are configured to run properly using the scripts from the repo. All automation jobs need to be configured to run as wss-master_vcadmin, this account both has local administration access on the automation server and also has admin access to the team's file share where many output files are written to. The scheduled tasks/automation jobs can be found under the following nodes:

standardsDefinitions.general.automation.scheduledtasks.tasks

Automation Repo Directory Structure

The folder and file structure of the virtualization-automation repo is critical to the automation process, by strictly following the folder and file structure the Virtualization team's automation can run from any server, with the correct account access, as all of the automated process have been updated to use relative pathing. Below are some of the guidelines for the repository:

Virtualization-Automation Repo Major Directory

- Config
    - Key configuration files that many automated process import for usage, also contains environment configuration standards.
- Documentation
    - Any related documentation created during the release time period
- Functions
    - Script functions importable and consumable by any automation process, example: remedy integration functions.
- General
    - Scripts that are for general use that do not specifically fit within a relation to ucs or vmware. Scripts in this directory are not automation ready and although they can be ran, they should not be expected to be ran without the user having a knowledge of the script, what changes may need to be made for successful execution, and what the script actually does.
- Output
    - General storage point for the output of any script, should be considered "short-term storage" as any data that should be kept for an extended or indefinite amount of time should be pushed to the team's file share
- UCS and VMware
    - Automation related specifically to Cisco UCS and VMware. **IMPORTANT**, any script directly under the root of these directories is considered automation ready and can run with or without user interaction. Scripts in the root of these directories cannot be relocated or their ability to run without error will fail. Scripts designed for automation are promoted from the "dev" sub directories of these major directories to the root when they are considered production ready and have been tested.

Virtualization-Automation Repo Sub Directories

- Dev
    - Staging directory for the development of scripts which are targeted to be fully automated. When the scripts have been fully developed/updated/changed AND have been tested they are promoted from dev to the root of their parent Major directory

- General
    - Scripts that are related to their parent directory topic but are not for a specific project and are not targeted for unassisted automation. Scripts in this directory are not automation ready and although they can be ran, they should not be expected to be ran without the user having a knowledge of the script, what changes may need to be made for successful execution, and what the script actually does.
- Project Named Directories
    - Scripts/automation developed for a specific project but may not be applicable to general consumption and do not fit into the fully automated model although they may or may not be fully automated processes. Could have been created for a one of execution, but could still be executed if needed whether fully or partially automated. Kept in the repo for reference purposes to assist in building out future scripts/automation.
- Retired
    - Should not be executed. Are considered deprecated. Kept in the repo for reference purposes to assist in building out future scripts/automation.

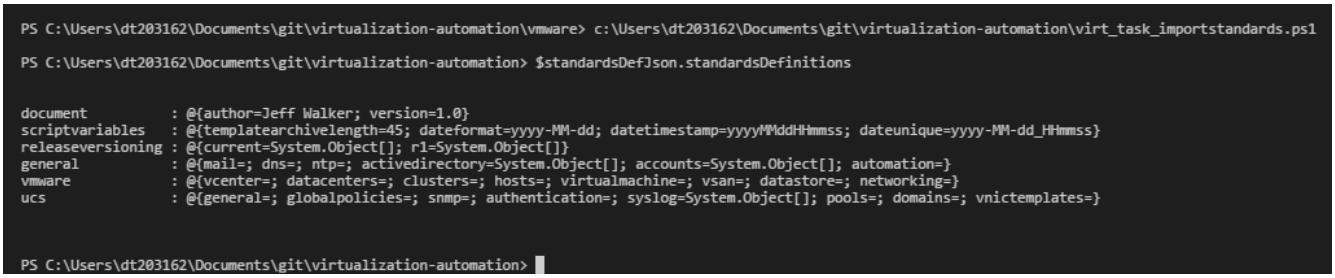
Standards Definitions File

A file is being introduced to manage standards across the Virtualization Team's ecosystem, including virtualization and underlying hardware. The standards definitions file will be considered as always under development as standards will continue to be documented and also defined when they are not. The standards definition file can be found in the Virtualization-Automation Git Repository under: virtualization-automation/config/virt_config_standards.json. JSON format was chosen because of ability to be read and understood easily, its ease of use to define both objects and arrays, and also its ability to be easily consumable by many programming languages.

The standards definitions file serves two major purposes:

1. Document standards for the Virtualization domain in a single centralized location. An easily accessible central location is key for building consistency. This standards document will both be pulled during repo download and also can be pulled directly from the repo for the most up-to-date version through the Gitlab web portal or through command line.

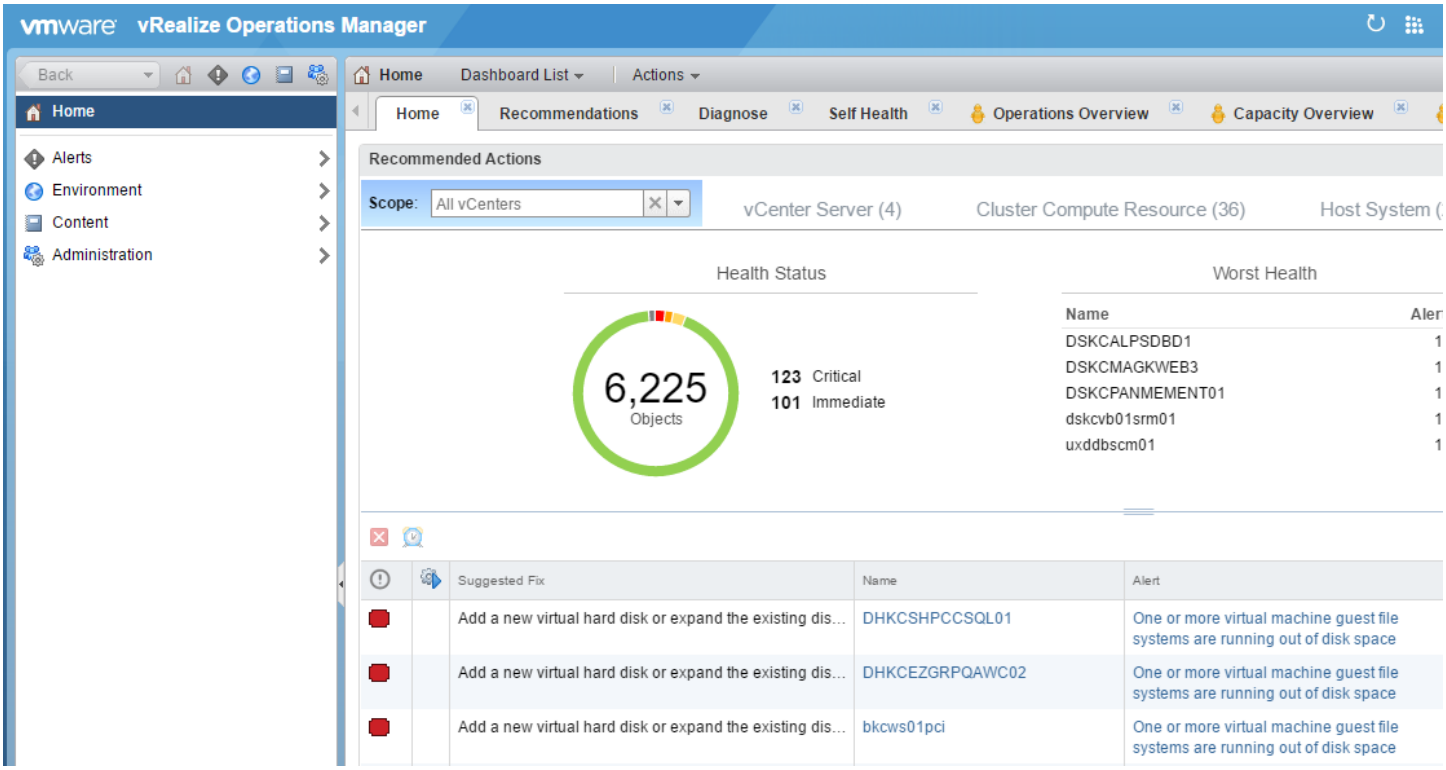
   A small script exists in the root of the virtualization-automation directory called virt_task_importstandards.ps1, when the script is ran it will populate a variable within Powershell and you can thing view the defined standards in an object/member method through Powershell or you can open it in a text editor for viewing as well. See below screenshot for an example of importing the standards using the import script:



2. A central location where automation can pull defined configuration information to automate processes, deployments, and to manage configuration drift. Together this is known as Infrastructure as Code (IaC).

For release 1.0 the standards definition file contassssins a large amount of data but is not in a finalized state, and again, this file will always be in a state of underdevelopment. Some minor portions of the automated process the team has are pulling data from the standards definitions file but expect that this will be expanded upon in a major way in future releases, to the point of script variables and alerting even being defined.

vRealize Operations Management



Overview

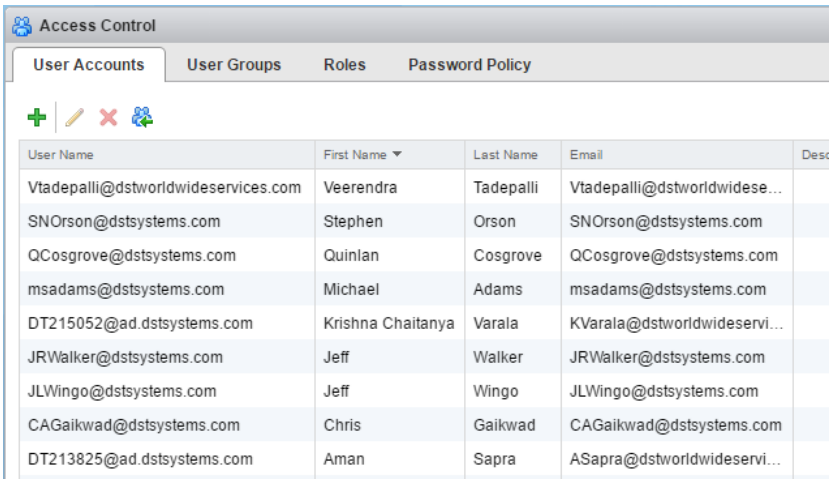
Operations Management has now been introduced into the Virtualization landscape on an official level. There have been previous installations that were treated as throw away, but this official release of the product into our environment means that it needs to be full supported by the virtualization team and will be upgraded, instead of deleted and deployed as new, so that collected data can be maintained. Spend time in the interface and become familiar, it is a powerful tool and with its integration into LogInsight it will become an essential troubleshooting and planning tool.

Access

Virtualization, Cloud, and TLMs Teams

Accounts

- Admin entry in Keepass
- User accounts are added individually to the system by the Admin user account
- You must log in using the name listed under the “User Name” column below:



URLs

Administration URL (https://vcops-vsphere/admin), used for administration of the Operations Collectors, Operations Node Clustering, and Updates

User Interface URL (https://vcops-vsphere/ui/login.action), used for viewing all dashboards, collected data, user permissions, capacity planning, etc.

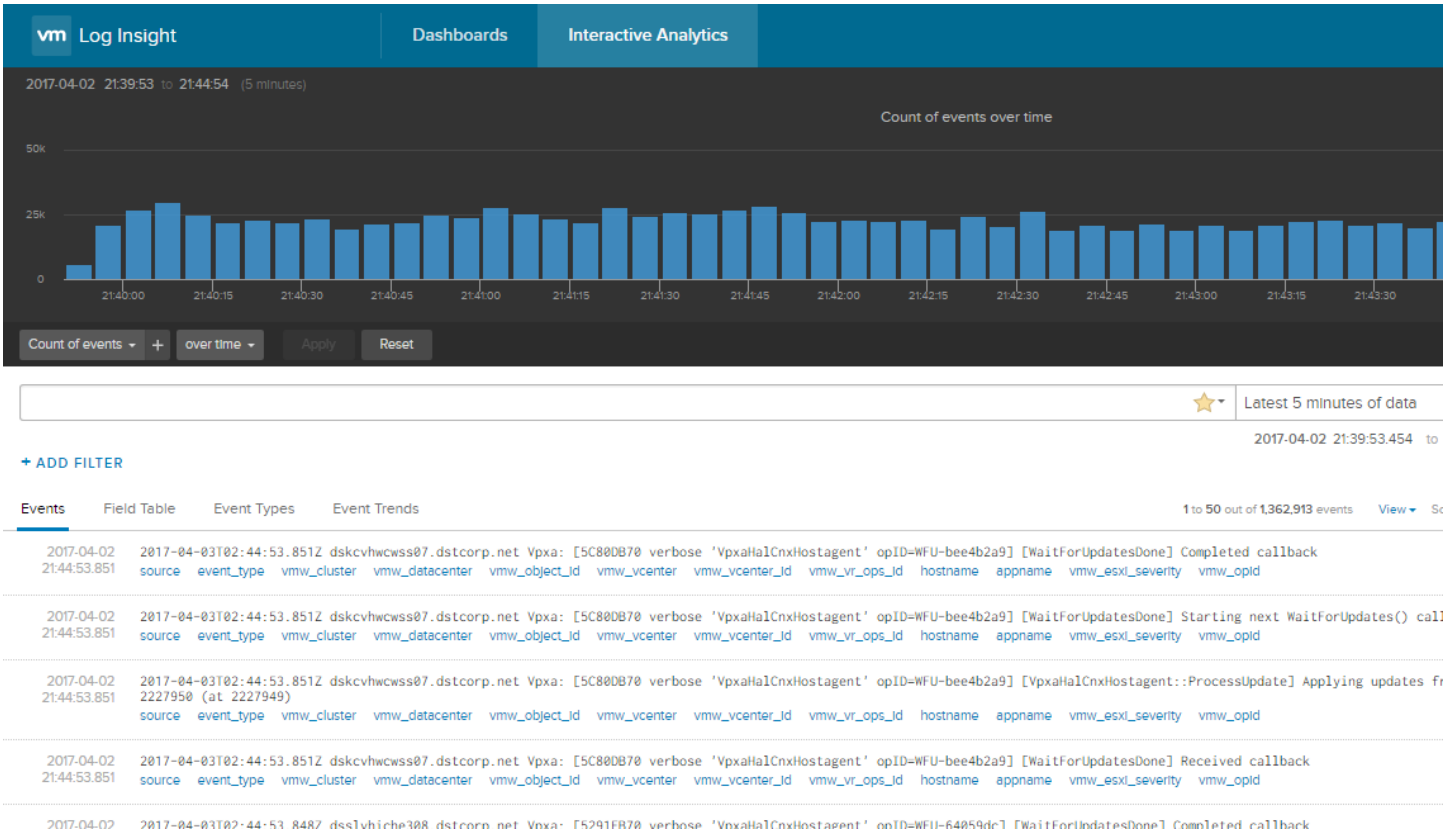
Future Goals

Dashboards for the respective teams with access, and updates to thresholds and limits defined in product to support our desired state, such as HA buffer space, density and consolidation ratios, etc.

Training

Many free training resources are available from VMware for Ops Manager, but in addition as the Virtualization TLM also learns more about the product information will be passed along.

vRealize LogInsight



Overview

LogInsight being included into Release 1.0 was a strategic call and was thought to be included close the end of the Release 1.0 development cycle. With its close integration with vROps it made sense to deploy and integrate both products together for the release. It is officially deployed, and is therefore under the management and administration of the Virtualization Team. LogInsight is a very clean and simple product to use but its ability to assist in troubleshooting and data aggregation is essential for the team to troubleshoot and proactively take control of the ever expanding Virtualization environment. For Release 1.0 log insight has been connected to Server based Production vCenters, it has been integrated into our vROps deployment, the content pack for Cisco UCS was deployed and all UCS domains have been directed to log to LogInsight, and the content pack for vRealize Orchestrator has also been installed and vRO is logging to the system as well. With aggregation of the logs from many sources in the Virtualization stack this will enable a deep level of troubleshooting and point in time discovery to better analyze and discover root cause. LogInsight will also be critical for the Virtualization team to discover reoccurring issues. Become familiar with the Interactive Analytics area of LogInsight, you can create custom queries and get deep into analytical troubleshooting. Administrator guide in the repo Documentation directory.

Access

Virtualization Team, Cloud Team, TLM Team

Accounts

admin entry in Keepass, full access full control

vmteam entry in Keepass, full view access, cannot manage admin settings

URLs

Administration and User Interface URL (https://vloginsight), used for both administration of LogInsight and also view access for dashboard and analytics engine.

Future Goals

Integration with the new EMC Unity an VMAX arrays, this will enable full stack troubleshooting for the virtualization team.

Training

Many free training resources are available from VMware for LogInsight, but in addition as the Virtualization TLM also learns more about the product information will be passed along.