PURESTORAGE®

Pure Storage and Open Source

Simon Dodsley

Director, Open Source Integrations
(theansibleguy.com)

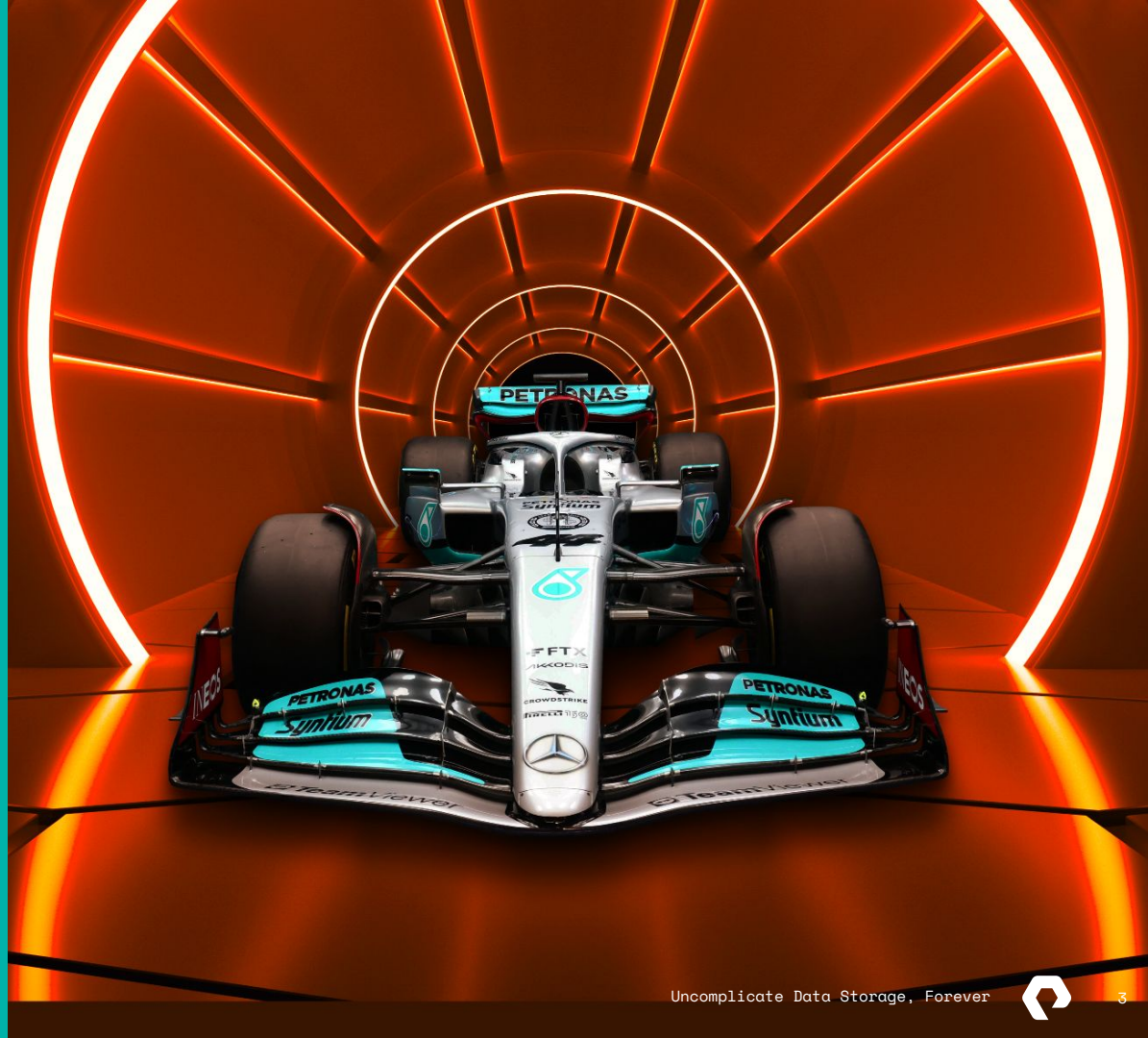
Agenda

Orchestration

Automation

Observability

Orchestration



openstack.

ONE OF THE MOST ACTIVE OPEN SOURCE PROJECTS WITH 40 MILLION CORES IN PRODUCTION

- OpenStack community completes the alphabet with Zed, and moves on to date-based releases
  - **2024.1 (Caracal)** just released
- OpenStack production deployments hit 40 Million Cores
- More than 450 organizations have contributed upstream to OpenStack, making it one of the most active open source projects in the world
- Over 9000 contributors across 180 countries
- OpenStack public cloud footprint exceeds 300 data centers worldwide

OpenStack

An IaaS Platform – without all those pesky licensing costs...

Cinder Block Storage (FlashArray)

Block QoS

Block Replication

Multi-platform

Manila File Storage (FlashBlade)

Simple Management

iSCSI, FC, NVMe, NFS

Enhanced user experience

Red Hat, Canonical, IBM, Mirantis & Platform9

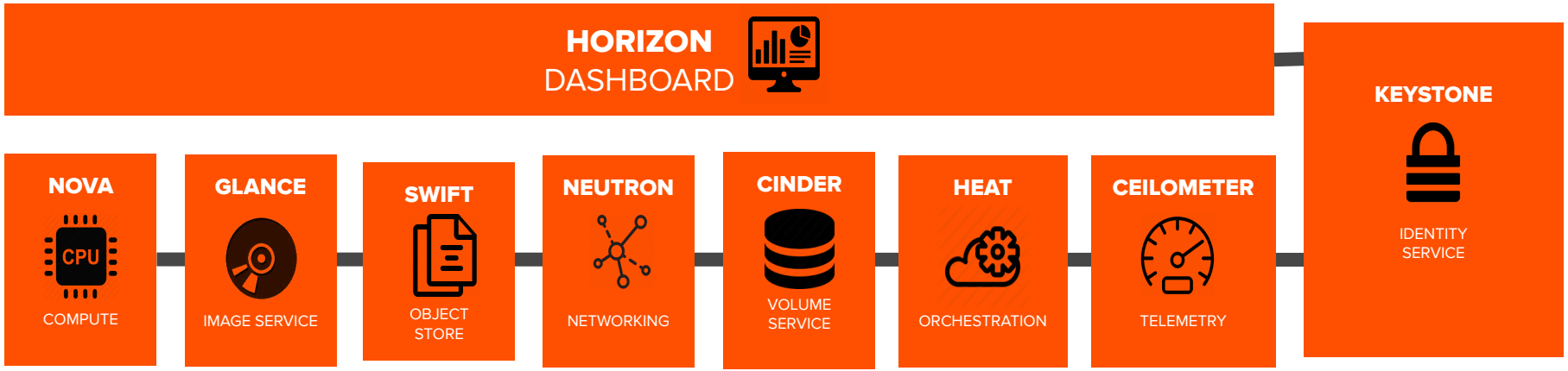
Release Cycles

Six monthly cycle

- Releases correspond to Design Summits
- Alphabetic release naming extended to date-based
- Latest Release: 2024.1 (Caracal)
- Next Release: 2024.2 (Dalmatian)
- Upgrade paths now available, including skip-level
- Less delta between releases now
- More feature rich

Series	Status	Initial Release Date	Next Phase	EOL Date
2024.2 Dalmatian	Development	2024-10-02 *estimated* (schedule)	Maintained *estimated 2024-10-02*	
2024.1 Caracal (SLURP)	Maintained	2024-04-03	Unmaintained *estimated 2025-10-03*	
2023.2 Bobcat	Maintained	2023-10-04	End Of Life *estimated 2025-04-04*	
2023.1 Antelope (SLURP)	Maintained	2023-03-22	Unmaintained *estimated 2024-09-22*	
Zed	Maintained	2022-10-05	Unmaintained *estimated 2024-05-02*	
Yoga	Unmaintained	2022-03-30	End Of Life *TBD*	
Xena	Unmaintained	2021-10-06	End Of Life *TBD*	
Wallaby	Unmaintained	2021-04-14	End Of Life *TBD*	
Victoria	Unmaintained	2020-10-14	End Of Life *TBD*	
Ussuri	End Of Life	2020-05-13		2024-02-20
Train	End Of Life	2019-10-16		2024-02-20
Stein	End Of Life	2019-04-10		2024-01-04
Rocky	End Of Life	2018-08-30		2023-08-16
Queens	End Of Life	2018-02-28		2023-01-18
Pike	End Of Life	2017-08-30		2022-10-13
Ocata	End Of Life	2017-02-22		2021-06-15
Newton	End Of Life	2016-10-06		2017-10-25
Mitaka	End Of Life	2016-04-07		2017-04-10
Liberty	End Of Life	2015-10-15		2016-11-17
Kilo	End Of Life	2015-04-30		2016-05-02
Juno	End Of Life	2014-10-16		2015-12-07
Icehouse	End Of Life	2014-04-17		2015-07-02
Havana	End Of Life	2013-10-17		2014-09-30
Grizzly	End Of Life	2013-04-04		2014-03-29
Folsom	End Of Life	2012-09-27		2013-11-19
Essex	End Of Life	2012-04-05		2013-05-06
Diablo	End Of Life	2011-09-22		2013-05-06
Cactus	End Of Life	2011-04-15		
Bexar	End Of Life	2011-02-03		
Austin	End Of Life	2010-10-21		

Architecture



- Modular architecture
- Designed for easy scale-out
- Based of (growing) set of core services

Feature Comparison - OpenStack / VMware

Management Console

OpenStack - Horizon
VMware - vCenter Server

Compute

OpenStack
Nova

KVM | ESXi
Hyper-V
XenServer

VMware
ESXi

Image Service

OpenStack
Glance

VMware
Content Library

Object Store

OpenStack
Swift

VMware
N/A
(FlashBlade)

Networking

OpenStack
Neutron

VMware
NSX

Volume Service

OpenStack
Cinder

VMware
vVols | VMFS
RDM

Orchestration

OpenStack
Heat

VMware
Aria Orchestrator
Aria Automation
vSphere Plugin
Powershell

Telemetry

OpenStack
Ceilometer

VMware
Skyline

Identity Service

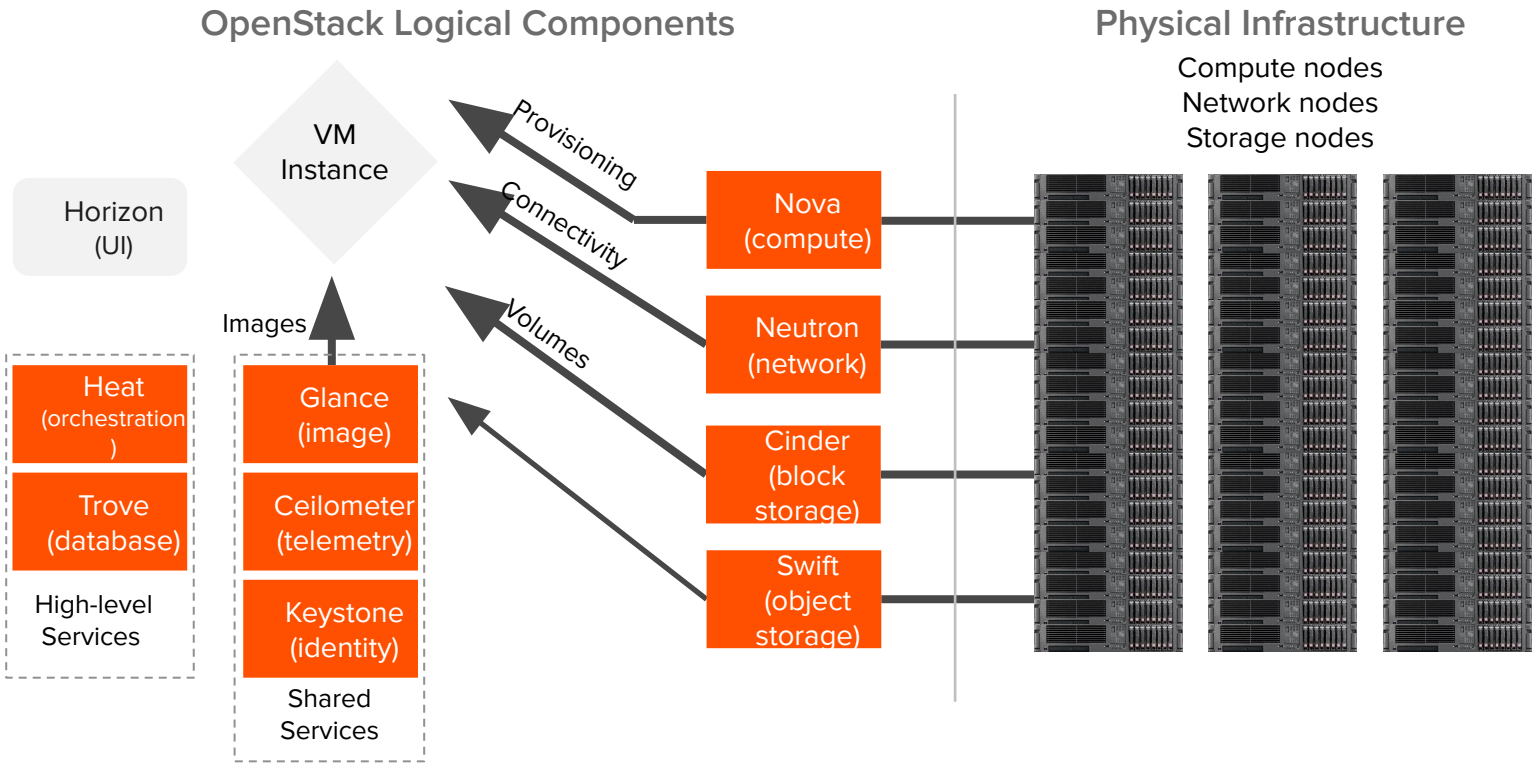
OpenStack
Keystone

VMware
Single-Sign On

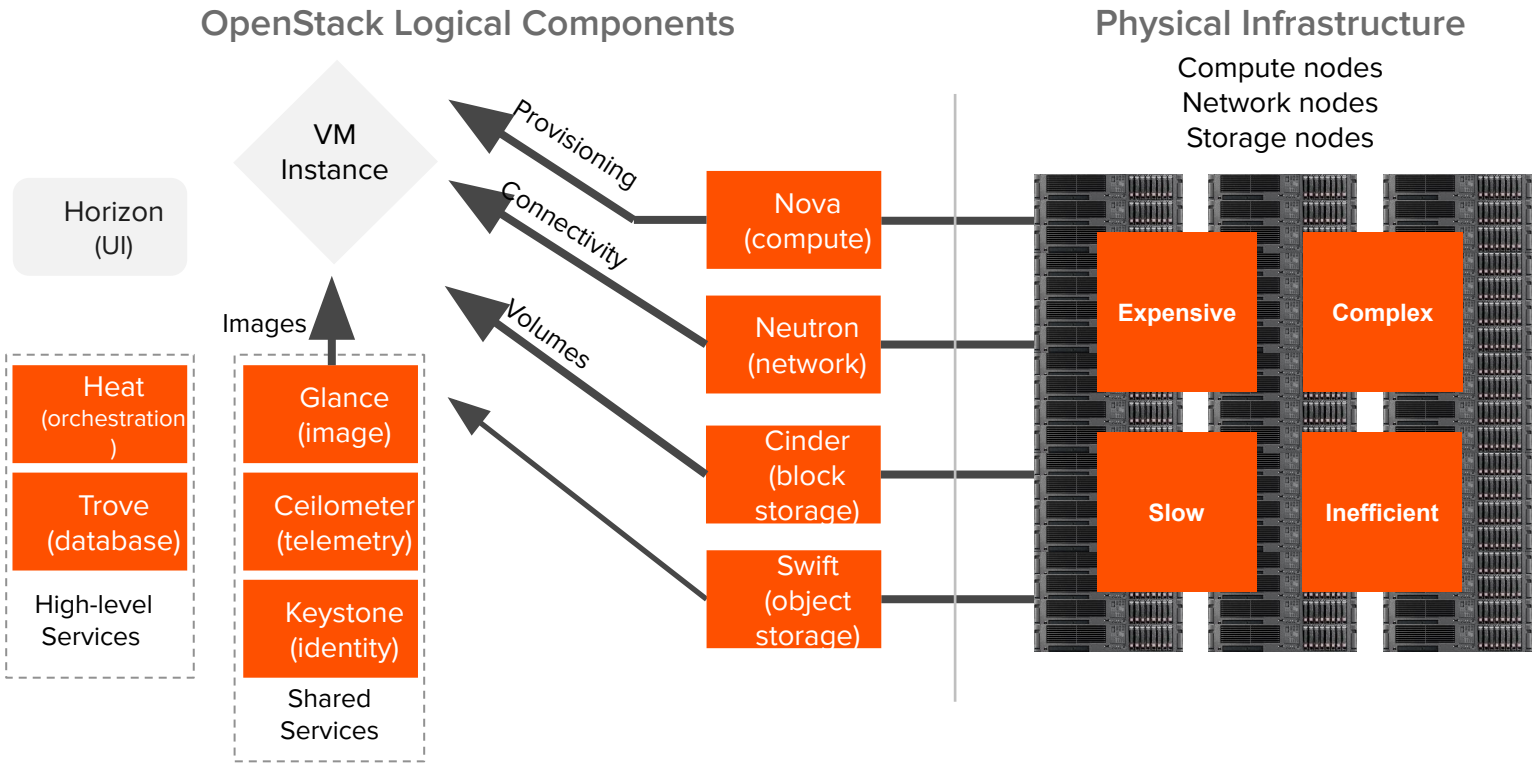
Projects

- **Important Others:**
  - Manila – Filesystem service
  - Trove – DBaaS
  - Ironic – Bare Metal
  - Magnum - Container orchestration

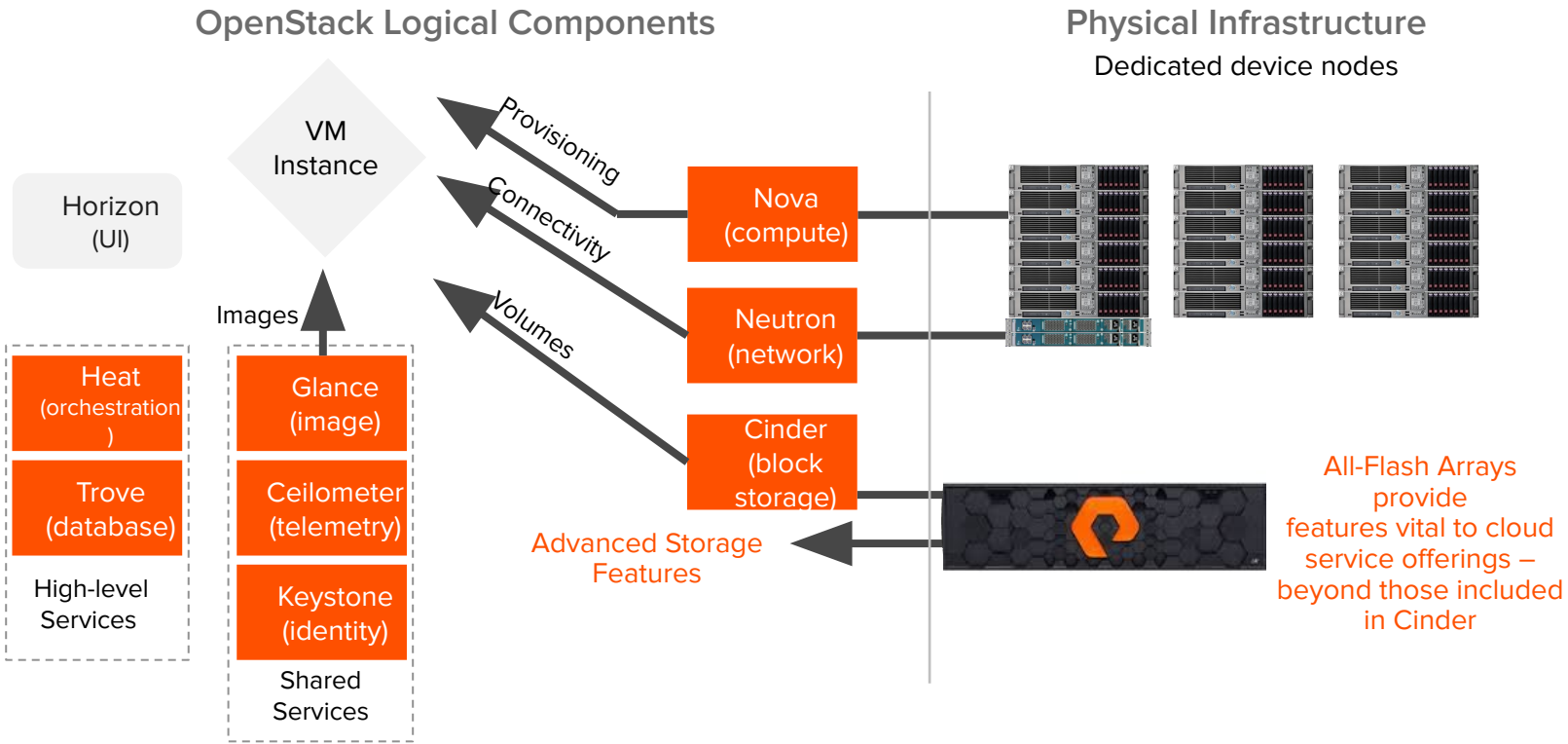
Typical OpenStack Deployment With White Box Systems



Typical OpenStack Deployment With White Box Systems



Typical OpenStack Deployment With White Box System



PURESTORAGE®

Pure and OpenStack

OpenStack Contributions

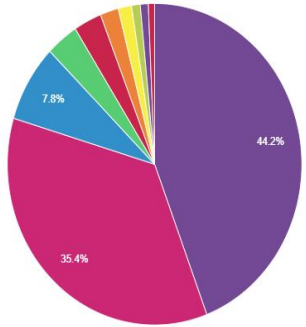
- Over 60,900 lines of code (27,000 in Cinder)
- Continuous Contributor to Cinder
- Code to 26 projects
- High-Level Architectural Contributions
- Supported Vendor Driver

Lines of code by Module

Show 10 entries Search

#	Module	Lines of code
1	cinder	26972
2	third-party-ci-tools	21583
3	fuel-plugin-purestorage-cinder	4744
4	cinder-specs	1877
5	manila	1227
6	os-brick	893
7	openstack-manuals	581
8	cinder-tempest-plugin	535
9	nova-specs	414
10	puppet-cinder	373

Showing 1 to 10 of 26 entries Previous Next



Contribution Summary

Commits: **310**
LOCs: **60981**
Do not merge (-2): **2**
Patch needs further work (-1): **93**
Commented on patch (0): **0**
Looks good (+1): **391**
Looks good for core (+2): **33**
Approve: **13**
Abandon: **0**
Change Requests: **246** (46 of them abandoned)
Patch Sets: **874**
Draft Blueprints: **18**
Completed Blueprints: **7**
Filed Bugs: **41**
Resolved Bugs: **31**
Emails: **31**
Translations: **0**

Pure’s Investment In OpenStack

- Full in-tree Cinder driver for FlashArray since Juno (2014 H2)
- **Unique** Multi-protocol Cinder support – iSCSI, FC, NVMe-RoCE & NVMe-TCP
- Full in-tree Manila driver for FlashBlade since Xena (2021 H2)
- Active contributor to multiple core projects
- Continuing development of driver – new feature add and feature support
- Certified with Red Hat, Canonical, Mirantis and IBM OpenStack deployments
- Dedicated Corporate team of developers and architects
- Sponsorship of Open Infrastructure Summits and Days

OpenStack Cinder Support Matrix

Pure Storage Related

•Required Driver Functions:

- •Create Volume
- •Delete Volume
- •Attach Volume
- •Detach Volume
- •Extend Volume
- •Create Snapshot
- •Delete Snapshot
- •Create Volume from Snapshot
- •Create Volume from Volume (clone)
- •Create Image from Volume
- •Volume Migration (host assisted)

•Optional Driver Functions Supported:

- •Extend an Attached Volume
- •Volume Replication (async, sync & trisync)
- •Consistency Groups
- •Replicated CG Snapshots
- •Thin Provisioning
- •Multi-Attach Support
- •Revert to Snapshot
- •iSCSI & NVMe CIDR support (IPv4 & IPv6)
- •iSCSI & NVMe CIDR list support (IPv4 & IPv6)
- •Host Personality
- •Quality-of-Service (back-end)
- •Active/Active High Availability Support
- •Enhanced support for PowerVC

OpenStack Features Roadmap

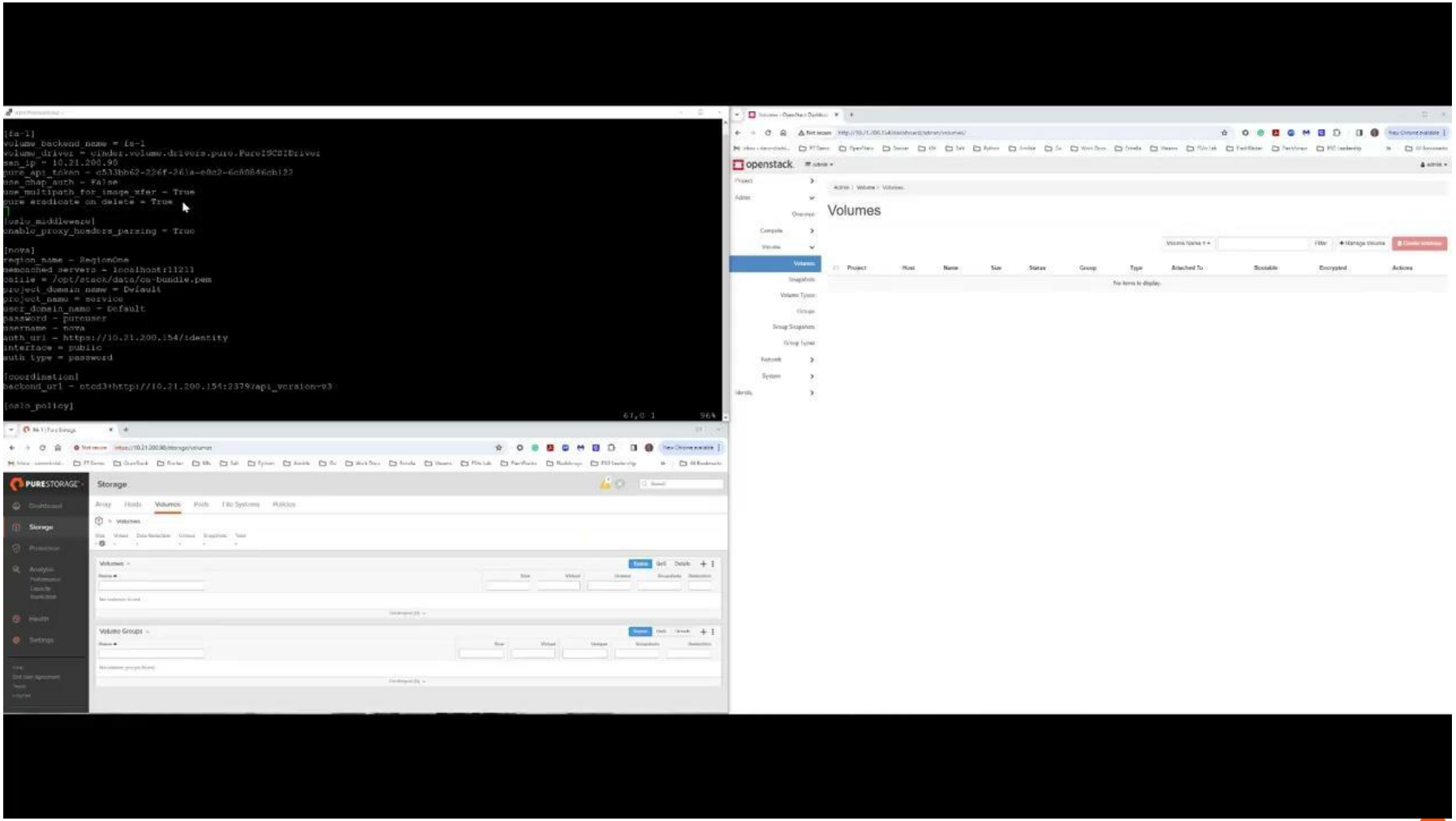
- Full REST v2 compliance (in review)
- SafeMode Support (in review)
- Volume Group Support (in progress)
- ActiveDR Support
- NVMe-FC Cinder driver for FlashArray (pending os-brick support)
- Manila driver for FlashArray File Services (2H 2025)
- Cinder NFS Drivers for FlashBlade and FlashArray

VMware to OpenStack Migration Options

- Utilise vVOLs
  - convert VMDK to vVOL
  - clone vVOL to full volume
  - import volume into OpenStack using `cinder manage`

- Use qemu-img
  - convert VMDK to QCOW2 using `qemu-img convert`
  - import to OpenStack as image using `openstack image create`

- Use V2V toolset (incl. backup/restore)
  - eg: Linux `virt-v2v`, Trilio, Commvault



Automation

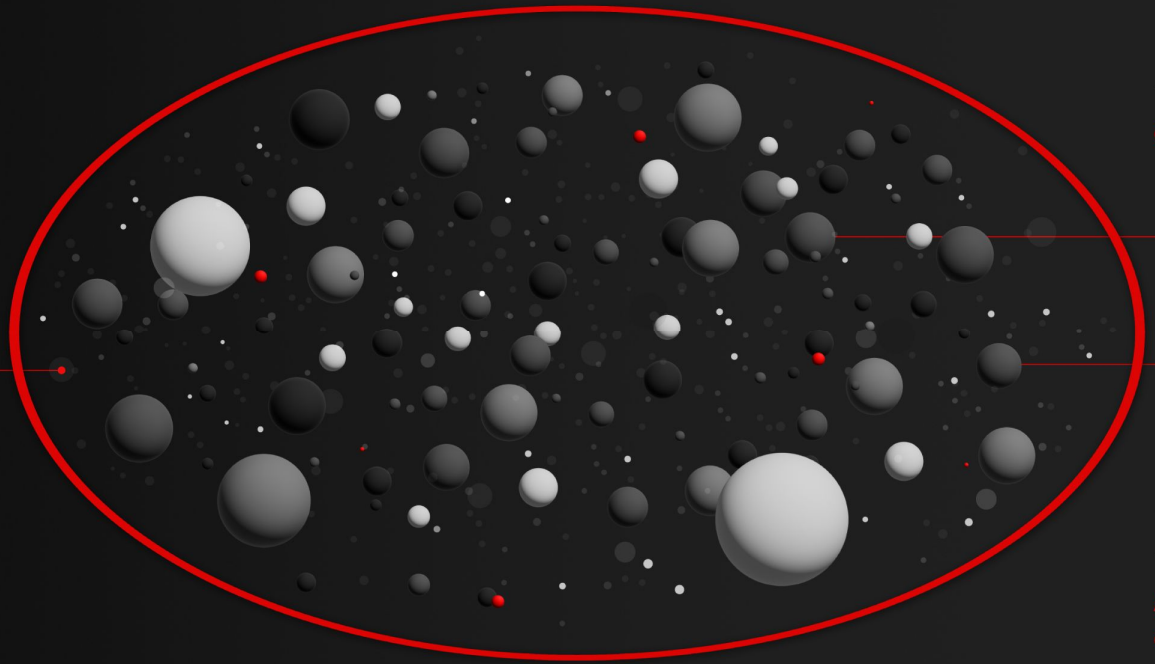
PURESTORAGE®

Ansible

PURESTORAGE®

#1

IT automation community in the world

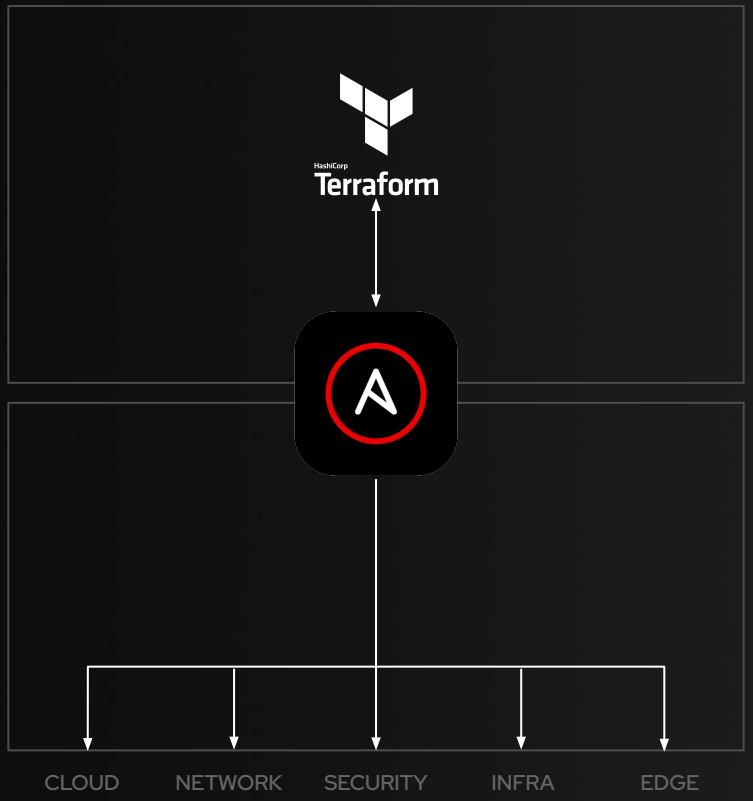


55,000+

GitHub stars

3,550+

Active open source contributors



Everything as code

Everything is possible

- **Infrastructure and configuration as code**
  - Ansible creates infrastructure manifests and triggers the provisioning and deprovisioning of infrastructure.
  - Dynamic inventory management provides access to newly provisioned infrastructure without manual intervention.
  - Combine tool chains to deliver infrastructure as code and configuration as code. Allowing for complete management of infrastructure life cycles with post-provisioning tasks.

One tool to manage multiple storage platforms

Day 0/1 deployment through Day 2 maintenance

Manage users, policies, volumes, snapshots, etc. with certified content collections

Reduce system inconsistencies and cost by optimizing storage management

Configure directory services

Simplify provisioning

IBM

PURESTORAGE®

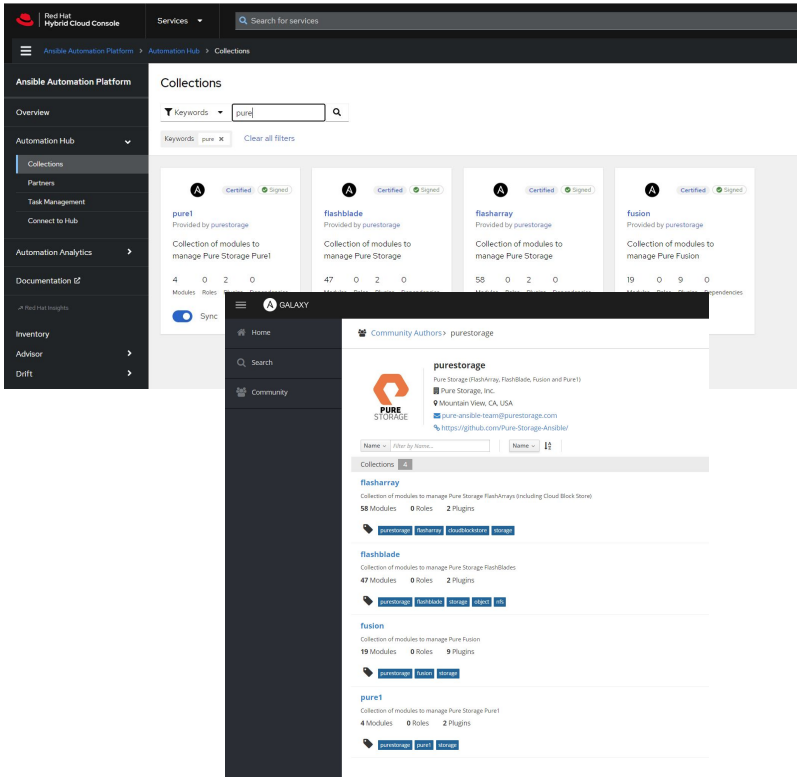
NetApp

Hewlett Packard Enterprise

DELL EMC

WHAT DOES PURE PROVIDE

- Modules and Collections Fully Certified by Red Hat
- Users can log support calls directly with Red Hat or Pure
- … or raise GitHub issues
- Available through Collections
  - Ansible Galaxy
  - Ansible Automation Hub (RH account needed)
  - Containerized Execution Environments



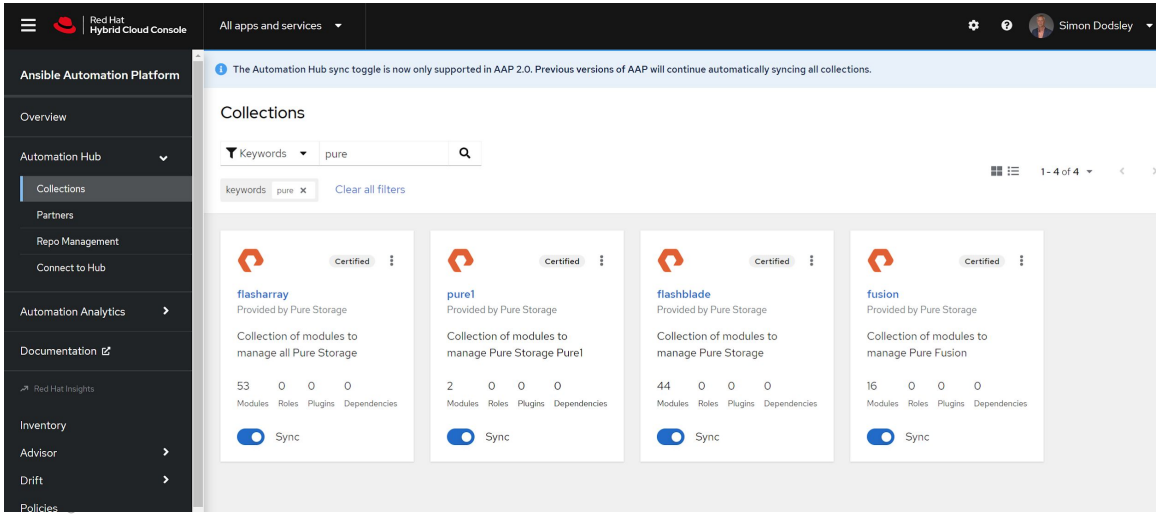
Collections Fully Certified by Red Hat

- 117 modules and counting

Users can log support calls directly with Red Hat or Pure

- ... or raise GitHub issues

FA, FB and Fusion* included in latest Ansible versions



Red Hat Certified Technology

ANSIBLE

Ansible

PURE INFO MODULES

USED TO BE CALLED FACTS MODULES

Everything you could ever want to know about your platform.

Get all the facts or just a subset (sometimes much faster response)

`purefa_info`, **`purefb_info`** & **`pure1_info`**

FlashArray subsets available*:

all, minimum, config, performance, capacity, network, subnet, interfaces, hgroups, pgroups, hosts, admins, volumes, snapshots, replication, vgroups, offload, apps, arrays, certs, kmip, clients, policies, dir_snaps, filesystems, virtual_machines, hosts_balance and subscriptions

FlashBlade subsets available*:

all, minimum, config, performance, capacity, network, subnets, lags, filesystems, snapshots, buckets, replication, policies, arrays, accounts, admins, ad, kerberos and drives

Pure1 subsets available*:

all, minimum, appliances, subscriptions, contracts, environmental and invoices

Use the created `{dict}` to **set facts** to use later in the playbook and to do clever things with Jinja2

*Not a comprehensive list

DEFINING YOUR TARGET PLATFORM

TELLING MODULES WHICH FA OR FB TO TALK TO...

- All modules need to have **`fa_url`** / **`fb_url`** and **`api_token`** provided as in the previous examples, except...
- Define shell environmental variables:
  - `PUREFA_URL=10.21.200.210`
  - `PUREFA_API=c6033033-fe69-2515-a9e8-966bb7fe4b40`
  - `PUREFB_URL=10.21.200.5`
  - `PUREFB_API=T-9f276a18-50ab-446e-8a0c-666a3529a1b6`
- These will be applied to **ALL** `purefa` or `purefb` modules, unless they have local definitions in the task.
- Or, use variables in the playbook, or as part of a variables file, or even pass in through the ansible command line.

PURE1 AS A TARGET

TELLING MODULES HOW TO TALK TO PURE1...

- All **`pure1`** modules need to have **`app_id`** and **`key_file`** provided with **`password`**, if set, except...
- Define shell environmental variables:
  - `PURE1_APP_ID=pure1:apikey:bRyL4DfW2KaItKMj`
  - `PURE1_PRIVATE_KEY_FILE="/directory/file.pem"`
  - `PURE1_PRIVATE_PASSWORD=password`
- These will be applied to **ALL** **`pure1`** modules, unless they have local definitions in the task.
- Or, use variables in the playbook, or as part of a variables file, or even pass in through the ansible command line.

ANSIBLE FACTS FOR EXTERNAL STORAGE

ADDED BY PURE STORAGE

- The following core facts for hosts are available to help automate storage provisioning when using `gather_facts: true` in your playbook or role
  - `ansible_iscsi_iqn` – provides the host iSCSI IQN
  - `ansible_hostnqn` – provide the host NMVe NQN
  - `ansible_fibre_channel_wwn` – provides all the host fibre channel WWNs, specifically the WWPNs
- Examples would look like this:

```
"ansible_iscsi_iqn": "iqn.1994-05.com.redhat:58acc9c1b3ca"
"ansible_hostnqn": "nqn.2014-08.com.example:nvme:nvm-subsystem-sn-d78432"
"ansible_fibre_channel_wwn": [
            "21000024ff52a9ba",
            "21000024ff52a9bb"
        ]
```

- Coincidentally these are formatted as required by the Pure FlashArray module

SIMPLE EXAMPLES

USING COLLECTIONS

- Create a FlashArray volume and attach to the localhost:

```
- name:
  hosts: localhost
  gather_facts: true
  tasks:
  - name: Create a volume
    purestorage.flasharray.purefa_volume:
      name: testvolume
      size: 13G
      bw_qos: 25M
      iops_qos: 200K
      fa_url: <FlashArray Mgmt VIP>
      api_token: <API token>

  - name: Create host and attach volume
    purestorage.flasharray.purefa_host:
      name: "{{ ansible_hostname }}"
      iqn: "{{ ansible_iscsi_iqn }}"
      volume: testvolume
      fa_url: <FlashArray Mgmt VIP>
      api_token: <API token>
```

- Create a FlashBlade NFS v4 filesystem with quotas, hard limit and snapshot support:

```
- name:
  hosts: localhost
  tasks:
  - name: Create a filesystem
    purestorage.flashblade.purefb_fs:
      name: testshare
      size: 1T
      nfsv4: True
      user_quota: 10k
      group_quota: 100k
      hard_limit: True
      snapshot: True
      nfs_rules: "10.10.28.78/32(rw,no_root_squash)"
      fb_url: <FlashBlade Mgmt VIP>
      api_token: <API token>
```

FURTHER EXAMPLES
USING COLLECTIONS

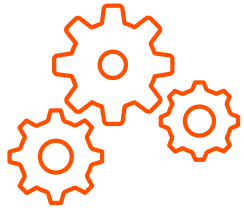
- Connect two FlashArrays in ActiveCluster and create a stretched pod:

```
- name:
  hosts: localhost
  tasks:
  - name: Connect FlashArrays
    purestorage.flasharray.purefa_connect:
      connection: sync
      target_url: <Target FA Mgmt VIP>
      target_api: <Target FA API token>
      fa_url: <Source FA Mgmt VIP>
      api_token: <Source FA API token>

  - name: Create Active Cluster pod
    purestorage.flasharray.purefa_pod:
      name: pod_1
      fa_url: <FlashArray Mgmt VIP>
      api_token: <FlashArray API token>

  - name: Stretch Active Cluster pod
    purestorage.flasharray.purefa_pod:
      name: pod_1
      stretch: <target array name>
      fa_url: <FlashArray Mgmt VIP>
      api_token: <FlashArray API token>
```

- Connect two FlashBlades and create a filesystem replica:

```
- name:
  hosts: localhost
  tasks:
  - name: Connect FlashBlades
    purestorage.flashblade.purefb_connect:
      target_url: <Target FB Mgmt VIP>
      target_api: <Target FB API token>
      fb_url: <Source FB Mgmt VIP>
      api_token: <Source FA API token>

  - name: Create File System Replica
    purestorage.flashblade.purefb_fs_replica:
      name: fs_1
      fb_url: <FlashBlade Mgmt VIP>
      api_token: <FlashBlade API token>
```

USE CASES WITH PURE + ANSIBLE…



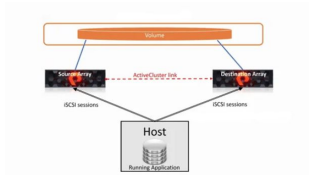
DAILY PROVISIONING, CONFIGURATION MANAGEMENT, ETC…

- Configure new arrays to the corporate standards
- Manage infrastructure configuration drift on fleets of arrays
- Update global configuration settings across multiple arrays
- Perform repetitive tasks, eg clone DB volumes for development work
- Perform DR tasks in a consistent and reproducible fashion
- Deploy hosts with a consistent external storage config

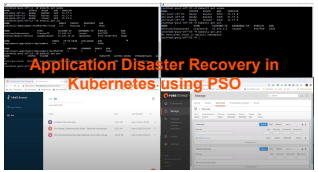
IF YOU CAN DO IT ON THE GUI OR CLI, YOU CAN DO IT WITH ANSIBLE

REAL WORLD EXAMPLES…

[Manage Infrastructure Drift]

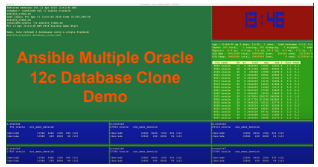


[Live Application Data Migration]



[Application Disaster Recovery in Kubernetes*]

[Load Balancing FlashArrays]



[Oracle Multi-Database Refresh]

Red Hat Solution Brief

Red Hat | PURESTORAGE®

Automate your storage environment with Red Hat and Pure Storage

Data management is a growing concern

87% of IT leaders believe that managing unstructured data growth is a top priority.[1]

68% of organizations spend more than **30%** of their IT budget on data storage, back-ups, and disaster recovery.[1]

Data growth brings new IT challenges

Modern enterprise IT organizations manage enormous amounts of data. In fact, more than 50% of enterprises have more than 5 petabytes (PB) of data.[1] This quantity of data can make it difficult for IT teams to operate efficiently and respond rapidly to issues, even in highly available and security-focused data storage environments.

As a result, many organizations want to streamline data resilience and security processes and simplify high-volume, repetitive tasks like provisioning and backup. Red Hat and Pure Storage offer solutions to help you increase the efficiency and accuracy of day-to-day operations and improve response time.

Build automated data storage workflows with Red Hat and Pure Storage

Red Hat and Pure Storage combine advanced automation and data storage platforms into integrated, streamlined solutions. As a foundation for building and scaling automation across an organization, Red Hat® Ansible® Automation Platform includes all of the tools you need to implement enterprise-wide automation in hybrid cloud environments.

Pure Storage delivers a modern data experience with flash-based storage and policy-driven services that let you run an automated, Storage-as-a-Service model across your storage platforms. With Pure Storage, you can build an all-flash datacenter, based on FlashBlade, FlashArray, and Pure Fusion's storage-as-code capabilities to reduce the complexity and expense of managing your environment.

Ansible Pure Storage Certified Content collections—predefined automation content developed, tested, and maintained by Red Hat and Pure Storage—help you automate more rapidly. Published to Ansible automation hub, these collections integrate Ansible Automation Platform and Pure Storage products, including FlashArray, FlashBlade, Pure1, and Fusion. You can perform nearly every operation in the Pure Storage command line and graphical user interfaces (GUIs) directly from Ansible Automation Platform as part of advanced automation workflows for managing your entire data storage environment.

Snapshot production databases

When integrating and testing new application features, developers need to work with the latest production data. IT staff typically use manual processes to copy production databases for developers to use. However, these databases can quickly diverge, causing issues when deploying new features into production. Ansible Automation Platform and Pure Storage let you build automated workflows that clone production databases each morning and delete old copies late at night. You can keep your production database safe, provide up-to-date copies for developers, and manage storage—all without manual intervention.

1 Komprise. "The state of unstructured data management 2022." August 2022.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

redhat.com

Ansible-foo

Troubleshooting Ansible for Beginners

01 **Latest Versions**
Ensure you are at the latest versions of Ansible*, Collections and Pure SDKs

02 **Multiple Versions**
Make sure you don't have multiple Collections in different locations - `ansible-galaxy collection list`

03 **Check your YAML**
YAML syntax is finicky - test with a YAML lint validator online

04 **Verbosity is your friend**
Rerun failures with `-vvvv` to get extended traceback information

05 **Devil is in the Detail**
When raising support calls or GitHub issues, provide versions, logs and example playbooks

VERSIONS ARE IMPORTANT

GETTING THE DETAILS

- Ansible version

```
$ ansible --version
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/home/simon/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /home/simon/.local/lib/python3.10/site-packages/ansible
  ansible collection location =  /home/simon/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/simon/.local/bin/ansible
  python version =  3.10.12  (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
```

- Ansible Collections versions

```
$ ansible-galaxy collection list | grep purestorage
purestorage.flasharray        1.21.0
purestorage.flashblade         1.13.1
purestorage.pure1             1.3.0
purestorage.flashblade         1.12.1
```

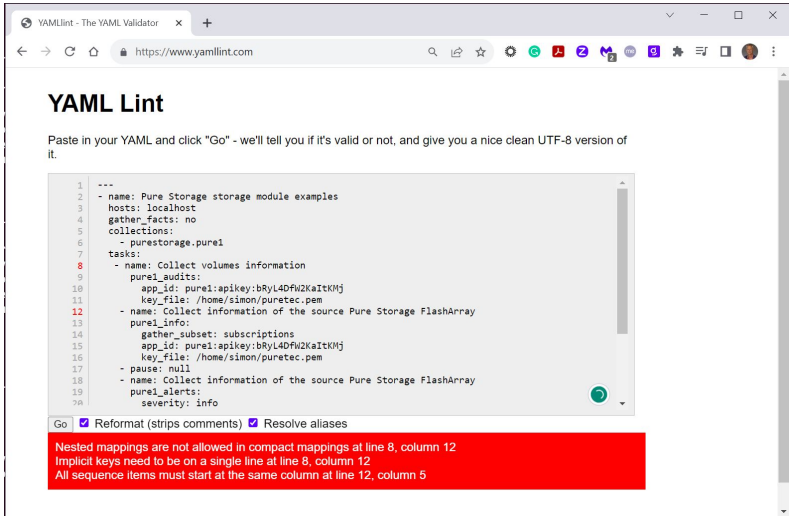
- SDK versions

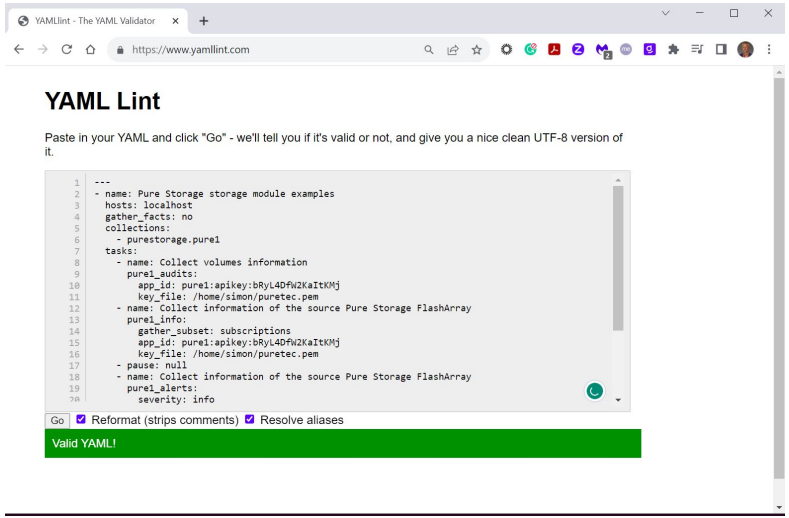
```
$ pip list | grep pur
purestorage                   1.19.0
purity-fb                     1.12.3
py-pure-client                1.41.1
```

YAML IS “FUN”

SPOT THE DIFFERENCE

```
simon@simon-X1: $ ansible-playbook pure1.yaml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
ERROR! We were unable to read either as JSON nor YAML, these are the errors we got from each:
JSON: Expecting value: line 1 column 1 (char 0)

Syntax Error while loading YAML.
  mapping values are not allowed in this context

The error appears to be in '/home/simon/pure1.yaml': line 8, column 17, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

- name: Collect volumes information
    pure1_audits:
                ^ here
```

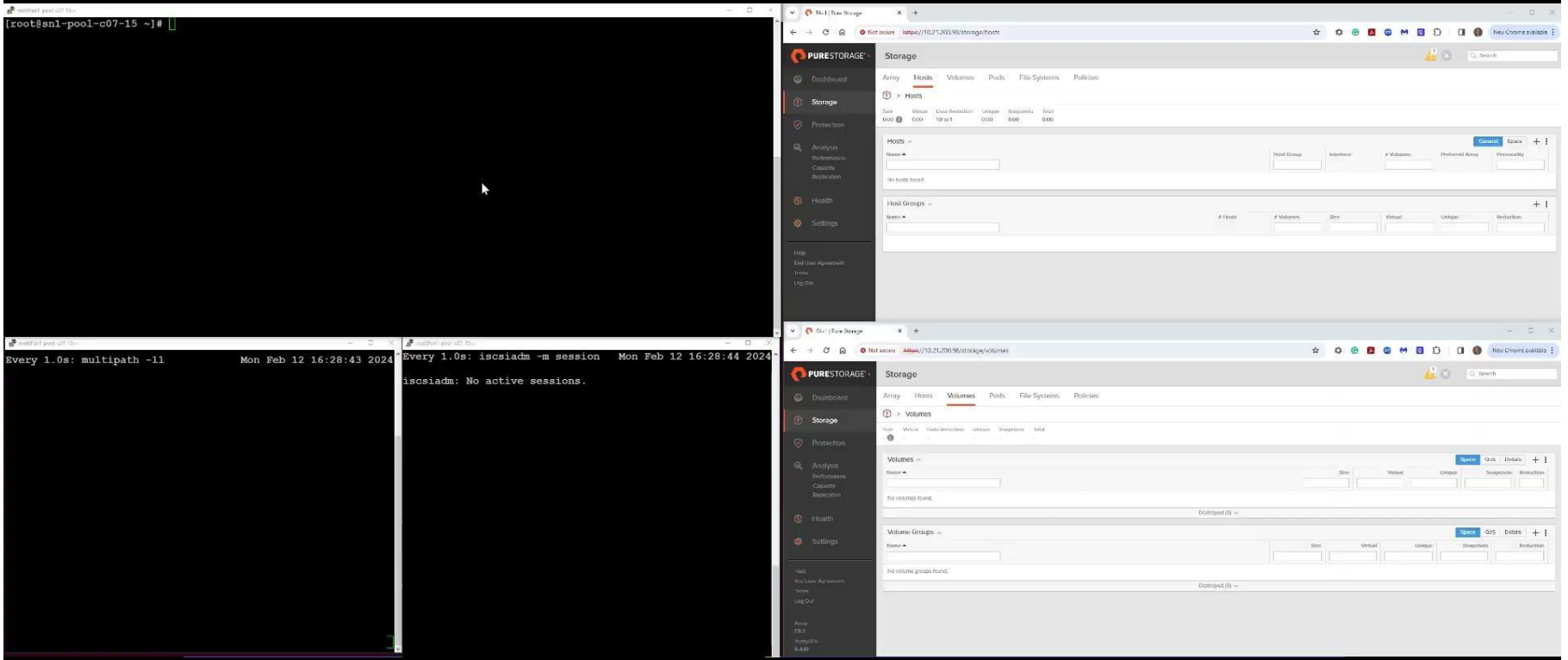




USING -vvvv

ansible-playbook -vvvv <playbook>

```
TASK [aaa]
*************************************************************************************************************************************************
task path: /etc/ansible/roles/purefa_snap_repl_kddia.yml:4
<127.0.0.1> ESTABLISH LOCAL CONNECTION FOR USER: root
<127.0.0.1> EXEC /bin/sh -c 'echo ~root && sleep 0'
<127.0.0.1> EXEC /bin/sh -c '( umask 77 && mkdir -p "` echo /root/.ansible/tmp `"&& mkdir "` echo
/root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121 `" && echo ansible-tmp-1695703335.1349988-35950-135452677291121="` echo
/root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121 `" ) && sleep 0'
Using module file /usr/local/lib/python3.9/site-packages/ansible_collections/purestorage/flasharray/plugins/modules/purefa_snap.py
<127.0.0.1> PUT /root/.ansible/tmp/ansible-local-35893ae9lkvwl/tmpmwg8xmbg TO
/root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/AnsiballZ_purefa_snap.py
<127.0.0.1> EXEC /bin/sh -c 'chmod u+x /root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/
/root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/AnsiballZ_purefa_snap.py && sleep 0'
<127.0.0.1> EXEC /bin/sh -c '/usr/bin/python3 /root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/AnsiballZ_purefa_snap.py &&
sleep 0'
<127.0.0.1> EXEC /bin/sh -c 'rm -f -r /root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/ > /dev/null 2>&1 && sleep 0'
fatal: [localhost]: FAILED! => {
    "changed": false,
    "module_stderr": "/usr/lib/python3.9/site-packages/urllib3/connectionpool.py:1013: InsecureRequestWarning: Unverified HTTPS request is being made
    to host '10.229.192.60'. Adding certificate verification is strongly advised. See:
    https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings\n
    warnings.warn(\n/usr/lib/python3.9/site-packages/urllib3/connectionpool.py:1013: InsecureRequestWarning: Unverified HTTPS request is being made to
    host '10.229.192.60'. Adding certificate verification is strongly advised. See:
    https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings\n
    warnings.warn(\n/usr/lib/python3.9/site-packages/urllib3/connectionpool.py:1013: InsecureRequestWarning: Unverified HTTPS request is being made to
    host '10.229.192.60'. Adding certificate verification is strongly advised. See:
    https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings\n
    warnings.warn(\n/usr/lib/python3.9/site-packages/urllib3/connectionpool.py:1013: InsecureRequestWarning: Unverified HTTPS request is being made to
    host '10.229.192.60'. Adding certificate verification is strongly advised. See:
    https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings\n
    warnings.warn(\n/usr/lib/python3.9/site-packages/urllib3/connectionpool.py:1013: InsecureRequestWarning: Unverified HTTPS request is being made to
    host '10.229.192.60'. Adding certificate verification is strongly advised. See:
    https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings\n  warnings.warn(\nTraceback (most recent call last):\n  File
    \"/root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/AnsiballZ_purefa_snap.py\", line 107, in <module>\n
    _ansiballz_main()\n  File \"/root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/AnsiballZ_purefa_snap.py\", line 99, in
    _ansiballz_main\n    invoke_module(zipped_mod, temp_path, ANSIBALLZ_PARAMS)\n  File
    \"/root/.ansible/tmp/ansible-tmp-1695703335.1349988-35950-135452677291121/AnsiballZ_purefa_snap.py\", line 47, in invoke_module\n
    runpy.run_module(mod_name='ansible_collections.purestorage.flasharray.plugins.modules.purefa_snap',
    init_globals=dict(_module_fqn='ansible_collections.purestorage.flasharray.plugins.modules.purefa_snap', _modlib_path=modlib_path),\n  File
    \"/usr/lib64/python3.9/runpy.py\", line 210, in run_module\n    return _run_module_code(code, init_globals, run_name, mod_spec)\n  File
    \"/usr/lib64/python3.9/runpy.py\", line 97, in _run_module_code\n    _run_code(code, mod_globals, init_globals,\n  File
    \"/usr/lib64/python3.9/runpy.py\", line 87, in _run_code\n    exec(code, run_globals)\n  File
    \"/tmp/ansible_purestorage.flasharray.purefa_snap_payload_38xwzc6b/ansible_purestorage.flasharray.purefa_snap_payload.zip/ansible_collections/pures
    torage/flasharray/plugins/modules/purefa_snap.py\", line 660, in <module>\n  File
    \"/tmp/ansible_purestorage.flasharray.purefa_snap_payload_38xwzc6b/ansible_purestorage.flasharray.purefa_snap_payload.zip/ansible_collections/pures
    torage/flasharray/plugins/modules/purefa_snap.py\", line 642, in main\n  File
    \"/tmp/ansible_purestorage.flasharray.purefa_snap_payload_38xwzc6b/ansible_purestorage.flasharray.purefa_snap_payload.zip/ansible_collections/pures
    torage/flasharray/plugins/modules/purefa_snap.py\", line 295, in create_snapshot\nTypeError: post_remote_volume_snapshots() got an unexpected
    keyword argument 'allow_throttle'\n",
    "module_stdout": "",
    "msg": "MODULE FAILURE\nSee stdout/stderr for the exact error",
    "rc": 1
}
```


[root@sn1-pool-c07-15 ~]#
Every 1.0s: multipath -ll Mon Feb 12 16:28:43 2024
Every 1.0s: iscsiadm -m session Mon Feb 12 16:28:44 2024
iscsiadm: No active sessions.
PURESTORAGE
Storage
Dashboard
Storage
Protection
Analysis
Health
Settings
Array
Hosts
Volumes
Pods
File Systems
Policies
Hosts
Host Groups
No hosts found.
Volumes
Volume Groups
No volumes found.
No volume groups found.

Observability

Pure and O11y

Meet you where your data is...

- Whatever platform you choose
- Integrations based on standards
  - OpenMetrics
  - OpenTelemetry

OpenMetrics Exporter

- Open Metrics Exporters for FlashArray and FlashBlade
  - Runs in Docker container
  - Best Efforts support only
- Integrates into all major O11y platforms
- Get them for free...
  - https://github.com/PureStorage-OpenConnect/pure-fa-openmetrics-exporter
  - https://github.com/PureStorage-OpenConnect/pure-fb-openmetrics-exporter
- Future: Integration directly into Purity//FA 6.6.8 and //FB
  - Full support at this point

OpenTelemetry Receivers

- Open Telemetry Receivers for FlashArray and FlashBlade
- Merged into otel-collector-contrib repo
- Can be used by any OTel compliant Observability toolset
  - Already merged into SumoLogic and OTel Collector Contrib distributions
- More details...
  - https://github.com/open-telemetry/opentelemetry-collector-contrib/tree/main/receiver/purefareceiver
  - https://github.com/open-telemetry/opentelemetry-collector-contrib/tree/main/receiver/purefbreceiver

Current OME Integrations

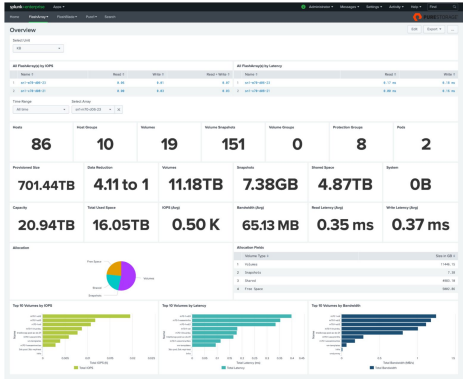
All use the same source data feeds from OpenMetrics Exporters

	Cloud-based (SaaS)	On-Prem (Darksite)
	✗	✓
	✓ Enterprise + Cloud	✓ Open Source (requires an on-prem TSDB)
	✓	✗
	✓	✗

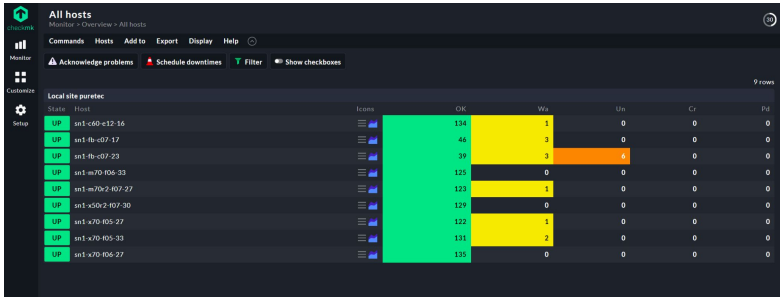
Other Monitoring Tools

Using proprietary connection agents

splunk>enterprise



checkmk



STOR2RRD