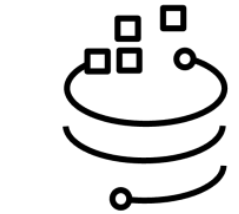
Modernization of Storage Scale

IBM Storage Scale User Meeting @ ISC2024

May 13, 2024 | Hotel Reichshof Hamburg

Norbert Schuld

Disclaimer



- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.

Modernize?

Moving from the past...

The current control plane (mm-commands)

- Designed for humans
- It is the single control plane interface
- Hard to impossible for applications to leverage
- Slow - implemented in shell scripting
- Little flexibility

Existing REST API

- Hosted on the Scale GUI
- User authentication limited to user id/password
- Interfaces with Scale via mm-commands
- Rigid set of roles for access control
- No support for domain and tenancy separation

...to the future

The new control plane

- API designed for applications
  - Ease of 3rd party integration with Scale
  - Versioning
- CLI designed for humans
  - CLI is a client application of the API
  - Behavioral consistency between API and CLI
  - Familiarity with other softwares (k8s, Openshift, git...)
- Improved security
  - Multiple authn methods (UDS, user certs, OIDC)
  - Authz based on RBAC
  - Domain separations and multi-tenancy with native domains
- Performance / Scalability *
- Developer Productivity
  - Leverage modern tools and languages
  - Code generation (45.5k LoC to date)
  - Skill building

Tech Preview 2024

Modernization of Scale: Security

Security Improvements

Removal of SSH dependency

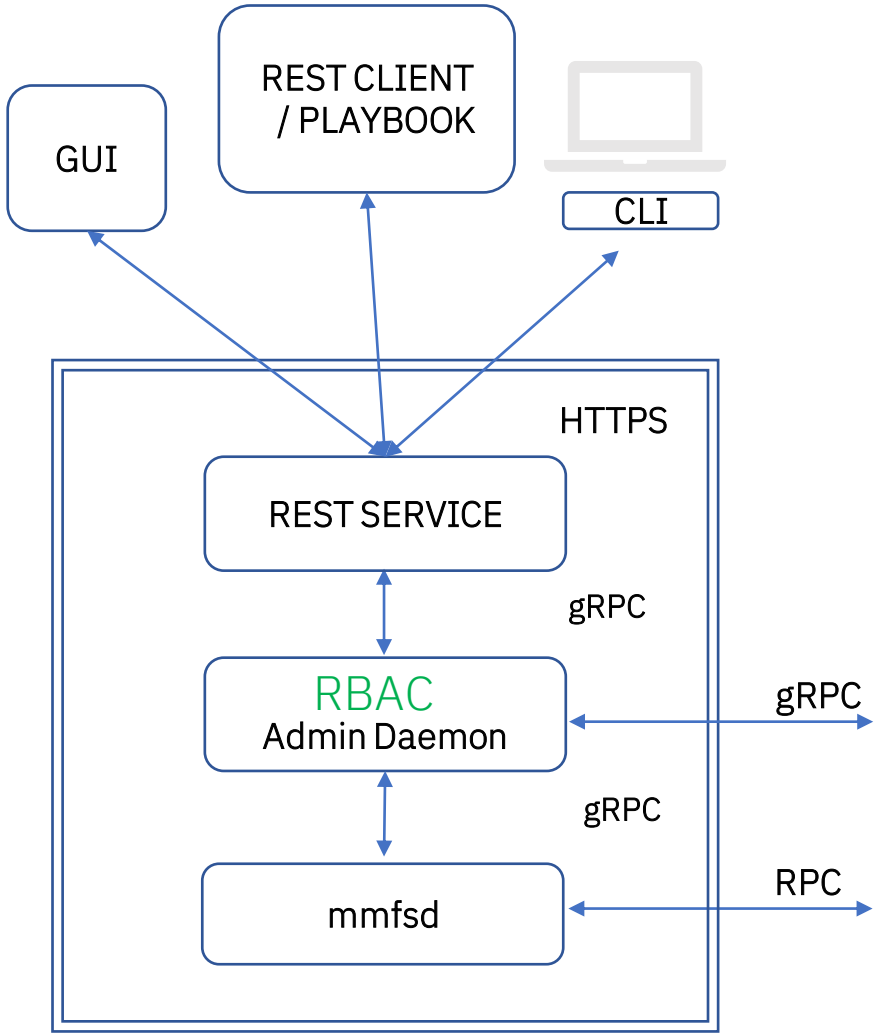
Removal of root requirement for control plane

Remote Administration

Fine-Grained Role Based Access Control
Declarative policy rules based on
Open Policy Agent

Control Plane Designed For Applications / Operators

Retain CLI for human management



Role-Based Access Control

Authentication

- scalectl via UDS – the admin daemon uses the id returned by the OS as the user's authenticated identity
- scalectl via URL or cURL – the admin daemon uses PAM for authenticating users (pam_unix module)
- users must exist on the host, GPFS does not create them
- additional authentication methods will be added with each release (OAUTH, LDAP/AD)

Authorization RBAC core components for enforcing authorization

- Based on Open Policy Agent (Graduated project from Cloud Native Computing Foundation)
- Domains – logical groupings of resources, users/roles, and actions
  (no built-in restrictions on which resources can be within a domain)
- Resources – represented as a URL endpoints (effectively represents a filesystem, fileset, disk, nsd, etc.)
  - A wildcard (*) can be used to match on any resource,
    e.g., /scalemgmt/v3/filesystems/fs0/filesets/* will match on any filesets in filesystem fs0
- Action – various operations that can be performed on a resource (not all actions apply to every resource)
  - Currently supported: create, delete, get, list, update, link, unlink, mount, unmount and cani / impersonate
- User – who makes the request
- Effect – "allow" or "deny", with following ordered rule evaluation:
  1. Default deny
  2. Find allow rule that matches request
  3. Check if explicit deny rule exists that overrides allow

Role-Based Access Control

Authorization (continued)

- RBAC components for enforcing authorization
  - Membership – relationship between a user and role within a domain
  - Permission – relationship between a role, action, effect, and resource within a domain
  - Resource Group – a collection of Resources that can be reused in various roles
  - Time – the time of the request
  - Attribute – advanced use case to define custom attributes that can be evaluated for Attribute Based Access Control (ABAC)

Default Domain (StorageScaleDomain)

- Contains the memberships, permissions and resource groups required by Scale components to function properly
- Cannot be deleted
- Roles cannot be modified (for now), but new roles can be added
- Root user is in the domain, but other users can be added
- If the domain is not specified, RBAC will be evaluated against the default domain
  - with scalectl --domain option; or in REST header as “X-StorageScaleDomain“

Tech Preview in Scale 5.1.9 + 5.2.0

Installation

Available on Linux x86-64 only in distinct packages

- Look for gpfs.scaleapi* packages and install all
- The base GPFS functionality was extended with a gRPC service in the mmfsd daemon

Post install…

- A new functional id and group
  - scaleapiadmd:scaleapiadm
- A new daemon running – scaleadmd
  - Currently running as root, but will move to running as scaleapiadmd:scaleapiadm in GA version
- Some files and directories in /var/mmfs have new ownership of scaleapiadmd:[scaleapiadm|root]

```
[root@dev-22 ~]# ls -l /var/mmfs
total 0
drwxr-xr-x. 2 root         root          64 Nov  1 11:13 ces
drwxr-xr-x. 2 root         root          35 Nov  1 11:13 etc
drwxr-xr-x. 4 scaleapiadmd root          40 Nov  1 11:13 gen
drwxr-xr-x. 2 root         root           6 Nov  1 11:13 mmbackup
drwx------. 2 root         root           6 Nov  1 11:13 mmpmon
drwxr-xr-x. 2 root         root         119 Nov  1 11:13 mmsysmon
lrwxrwxrwx. 1 root         root          13 Nov  1 11:13 run -> /var/run/mmfs
drwxr-sr-x. 5 scaleapiadmd scaleapiadm  42 Nov  1 11:13 scaleadm
drwxr-s---. 4 scaleapiadmd scaleapiadm  34 Nov  1 11:13 ssl
drwxr-xr-x. 3 root         root          26 Nov  1 11:14 tmp
[root@dev-22 ~]#
```

- Ready to create a new one-node cluster as root or non-root with additional setup

Command Examples

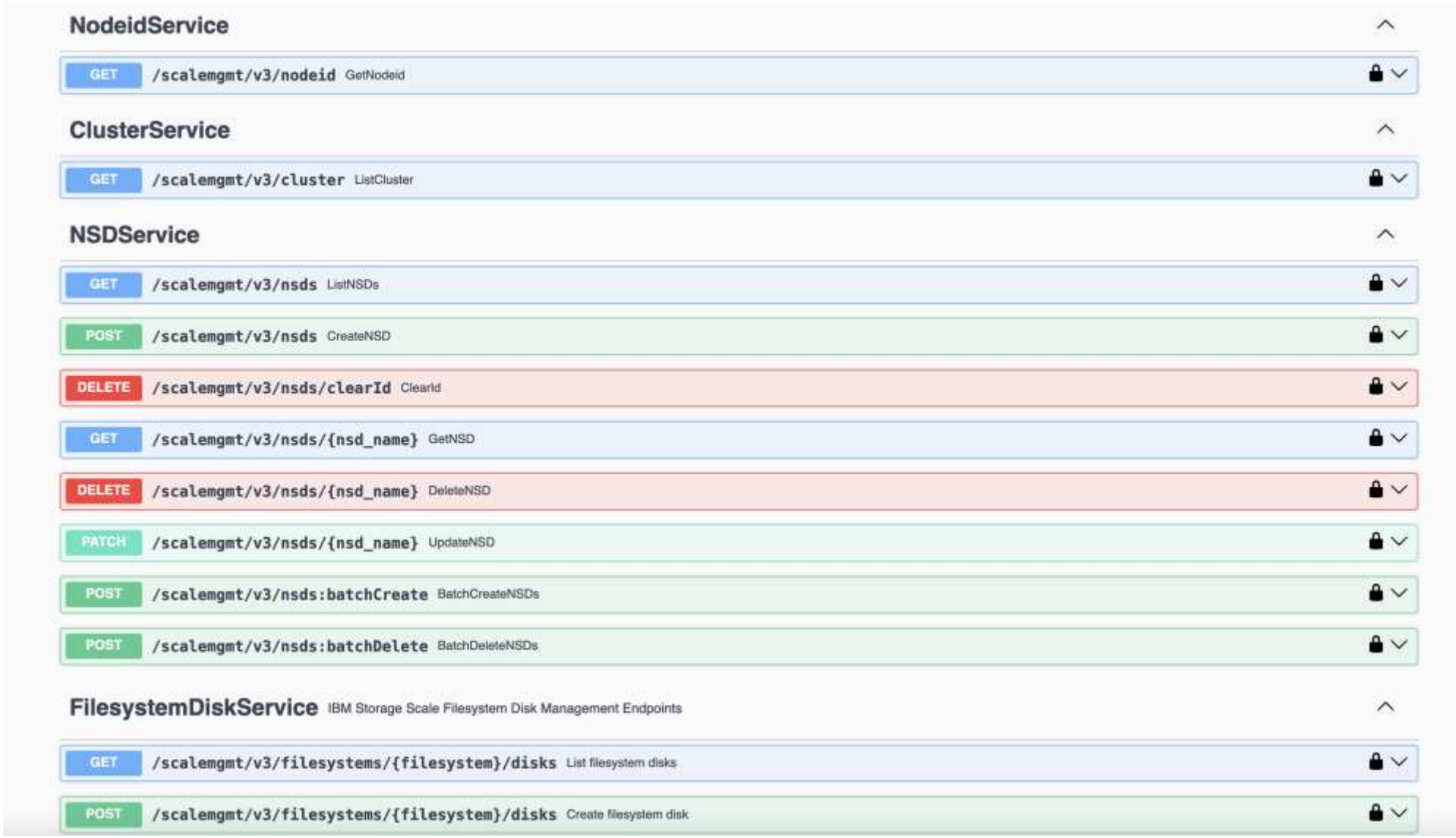
```
[root@mosdev-11 ~]# scalectl -h
Storage Scale Admin CLI interface

Usage:
  scalectl [flags]
  scalectl [command]

Available Commands:
  authorization Authorization commands
  cluster       Cluster commands
  fileset       Fileset commands
  filesystem    Filesystem commands
  node          Node commands
  nodeid        NodeId commands
  nsd           NSD commands
  operations    Operation commands

Flags:
      --debug string[="stderr"]    enable debug logging for the current request. Accepts an absolute file path to store
 the logs in the form of --debug=<file>. If no file path is provided, stderr will be used
      --domain string              Sets the domain for the request (default "StorageScaleDomain")
  -h, --help                       help for scalectl
      --insecure-skip-tls-verify   if true, the server's certificate will not be checked for validity. This will make y
our HTTPS connections insecure
      --json                       display output in json format
      --url string                 send the request over https to the specified endpoint <FQDN/IP>:<port>. An IPv6 addr
ess must be wrapped in square brackets such as [IPv6]:<port>. If a port is not specified, 46443 will be used
      --version                    scalectl build information

Additional help topics:
  scalectl config        config commands

Use "scalectl [command] --help" for more information about a command.
```

Command Examples

```
[root@mosdev-11 ~]# scalectl filesystem -h
Filesystem commands

Usage:
  scalectl filesystem [command]

Available Commands:
  create      Create a new filesystem
  delete      Delete an existing filesystem
  get         Describe an existing filesystem
  list        List existing filesystems
  mount       Mount existing filesystem
  mountAll    Mount all existing filesystems
  unmount     Unmount existing filesystem
  unmountAll  Unmount all existing filesystems
  update      Update an existing filesystem

Flags:
  -h, --help   help for filesystem

Global Flags:
      --debug string[="stderr"]    enable debug logging for the current request. Accepts an absolute file path to store
 the logs in the form of --debug=<file>. If no file path is provided, stderr will be used
      --domain string              Sets the domain for the request (default "StorageScaleDomain")
      --insecure-skip-tls-verify   if true, the server's certificate will not be checked for validity. This will make y
our HTTPS connections insecure
      --json                       display output in json format
      --url string                 send the request over https to the specified endpoint <FQDN/IP>:<port>. An IPv6 addr
ess must be wrapped in square brackets such as [IPv6]:<port>. If a port is not specified, 46443 will be used

Use "scalectl filesystem [command] --help" for more information about a command.
```

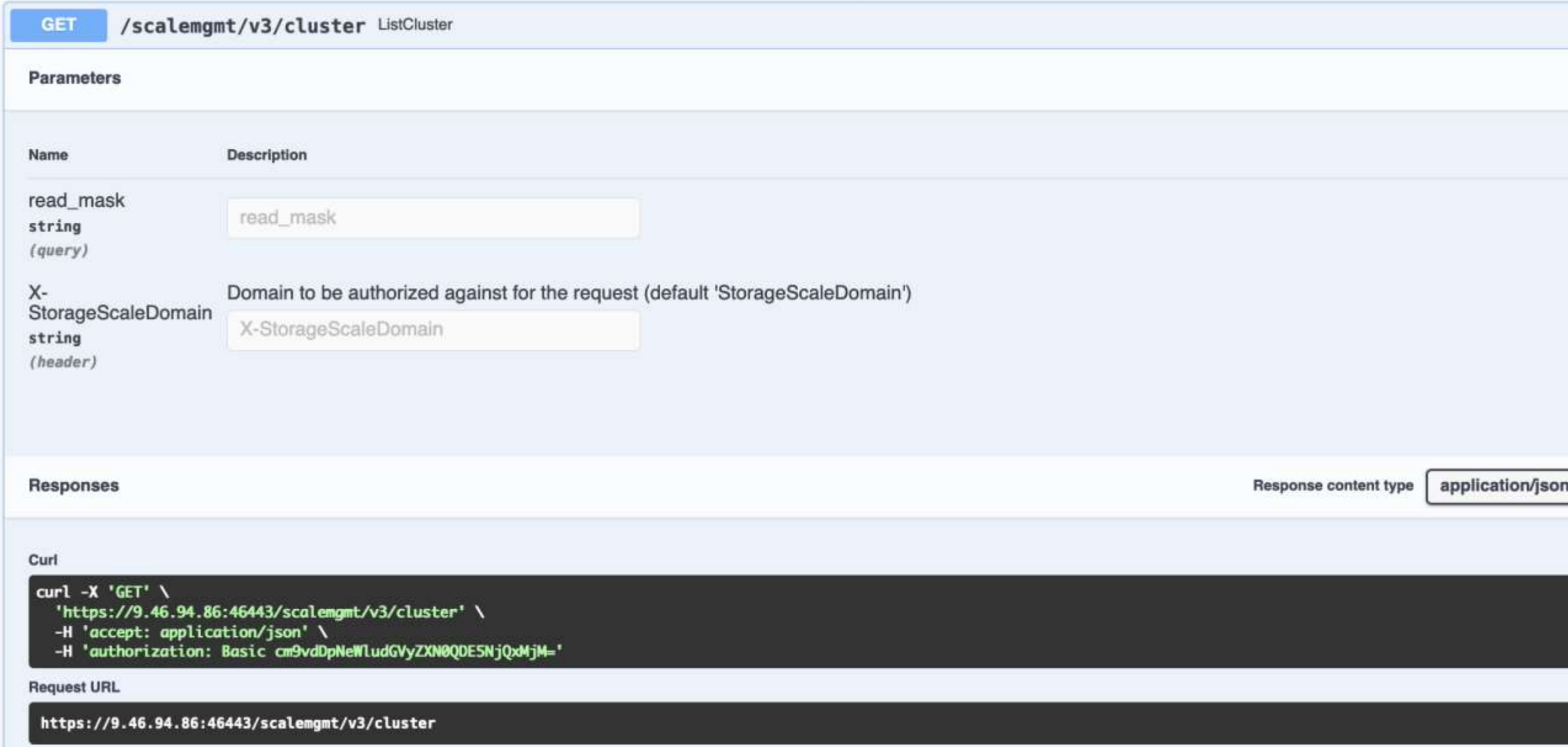
REST Overview

After the admin daemon has been started, you can access the swagger documentation by going to the endpoint (ending slash is important):https://x.x.x.x:46443/openapi/ where x.x.x.x is the public IP of the API server node



REST Example

Swagger UI:



GET /scalemgmt/v3/cluster ListCluster

Parameters

Name	Description
read_mask **string** *(query)*	read_mask
X-StorageScaleDomain **string** *(header)*	Domain to be authorized against for the request (default 'StorageScaleDomain') X-StorageScaleDomain

Responses Response content type application/json

Curl

```
curl -X 'GET' \
  'https://9.46.94.86:46443/scalemgmt/v3/cluster' \
  -H 'accept: application/json' \
  -H 'authorization: Basic cm9vdDpNeWludGVyZXN0QDE5NjQxMjM='
```

Request URL

```
https://9.46.94.86:46443/scalemgmt/v3/cluster
```

REST Example

Swagger UI:

Server response

Code	Details
200	**Response body**

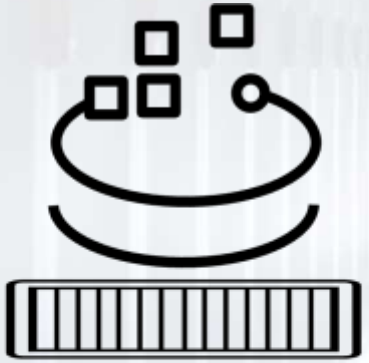

```
{
        "clusters":  [
                {
                        "cluster_name":  "ansibleCluster2153-1.fyre.ibm.com",
                        "cluster_id":  "4623696294882986386",
                        "rsh_path":  "/usr/bin/ssh",
                        "uid_domain":  "ansibleCluster2153-1.fyre.ibm.com",
                        "rsh_sudo_wrapper":  "NO",
                        "rcp_path":  "/usr/bin/scp",
                        "rcp_sudo_wrapper":  "NO",
                        "repository_type":  "CCR",
                        "primary_server":  "ansibleCluster2153-1.fyre.ibm.com",
                        "nodes":  [
                                {
                                        "node_number":  "1",
                                        "daemon_node_name":  "ansibleCluster2153-1.fyre.ibm.com",
                                        "ip_address":  "10.21.106.34",
                                        "admin_node_name":  "ansibleCluster2153-1.fyre.ibm.com",
                                        "designation":  {
                                                "quorum":  true,
                                                "manager":  true
                                        }
                                },
                                {
```

What is implemented today

- NSD Services
- Node Services (add, delete)
- Node Identities (TLS communication across admin daemons)
- Cluster Creation / Information
- Storage Pool Services
- Fileset Services
- RBAC
- Filesystem Services
- Filesystem Policy Services (mmlspolicy / mmchpolicy)
- LRO Management (long running operations)
- Non-Root Admin Daemon
- Manager services (mmlsmgr, mmchmgr)

Thank you for using

Storage Scale



Storage Scale System