Cloudify 4.0.0
Release Notes

Table of Contents

What's New?

Cloudify Manager

- **Multi-Tenancy** This version of Cloudify introduces the concept of multiple tenants in Cloudify Manager, for Premium customers. Multi-tenancy enables you to create multiple independent logical groups as isolated environments that can be managed by a single Cloudify Manager. A tenant is a logical entity that contains all its resources, for example blueprints, deployments and workflows. All operations are performed within a tenant.

  A single default tenant is created during Cloudify Manager installation, and you add other tenants as required. You can assign users and user groups to each tenant and assign user roles, enabling you to specify the content within Cloudify Manager that users are able to access and manage. User management can be integrated with your LDAP/AD setup.

  In the case of Community Edition users, a single tenant with a single user is created during installation. When you log into Cloudify Manager, the built-in credentials are used.

  The default tenant cannot be deleted.
  .
- **Clustering Cloudify Managers for High Availability** Premium customers now have the ability to cluster multiple Cloudify Manager instances, to ensure resilience in the event of a manager becoming unavailable.

  One Cloudify Manager is designated as the active manager and the others (two) are designated as hot standbys that are constant mirrors of the data of the active manager. In the event that the active Cloudify Manager fails, an automatic failover switch activates one of the hot standbys.

- **User Interface** A new user interface is displayed on Cloudify Manager. The user interface enables you to create dashboards that are:

- ○ Easily and intuitively configured, to build a personalized view of the specific data that most interests you
  - ○ Write your own widgets, whether their data is Cloudify based or from external systems, and to integrate them as part of your Cloudify dashboard.
  - ○ Support different views, according to the user role.

- ● **LDAP Integration** LDAP/AD (a Premium user feature) integration is improved from previous versions. You configure Cloudify with the LDAP configuration during the bootstrap process, or via the API (on a clean Cloudify Manager.)

  When a user logs in to Cloudify Manager, their credentials are passed to the LDAP/AD service for authentication.Following authentication, the user is able to access any tenants in the Cloudify Manager to which they have been assigned, either as an individual or as part of an assigned LDAP group.

  LDAP passwords are no longer stored in the Cloudify database. Authentication is performed via LDAP and a token is generated and used for the user session.

Web User Interface

- ● The Web user interface, for Premium edition users, has been upgraded to provide a more streamlined user experience.
  - ○ Out-of-the-box dashboards display widgets for the most commonly required data and viewing preferences.
  - ○ The dashboards are customizable so that you can easily and intuitively configure them to build a personalized view for the specific data that interests you most.
  - ○ You can create your own dashboard pages.
  - ○ You can write your own widgets, regardless of whether their data is Cloudify-based or derived from external systems, and integrate them as part of your Cloudify dashboards.
  - ○ The display of different views according to the user roles (users/admins) is supported.

Cloudify Command-Line Interface

- ● The following commands have been added to the CLI:
  - ○ `profiles`
  - ○ `tenants`
  - ○ `cluster`
  - ○ `users`
  - ○ `user-groups`

- ○ `ldap`
- The following commands have been removed from the CLI:
  - ○ `recover`
  - ○ `teardown`
- **Technical Preview Feature**
  In the CLI, you can create resources that are private to your account that will not be shared across the tenant. The CLI optional argument is `--private-resource`.

Security

- Cloudify includes built-in user roles with which users are associated. Each role has different permissions, ensuring a role-based access control operation.

  Cloudify supports the concept of users, user groups, and tenants. These elements can be either defined locally in Cloudify, or taken from an external user management system (LDAP integration is native). In the latter case, passwords are not stored in Cloudify, authentication is performed via LDAP and a token is generated and used for the user session.

  Communication from the outside world to Cloudify Manager and its SSL/TLS configuration is the user's responsibility (CA/host verification, etc.), where the endpoints include the UI and REST API.

  If an self-signed certificate is used, you must use the following commands from the UI machine in order for the UI to be available:

  ```
  /etc/sysconfig/cloudify-stage
  NODEJS_HOME=/opt/nodejs
  STAGE_HOME=/opt/cloudify-stage
  NODE_TLS_REJECT_UNAUTHORIZED=0
  ```

  Then run `sudo systemctl restart cloudify-stage` to restart the cloudify-stage service.

  Communication between Cloudify agents and Cloudify Manager (and within Cloudify Manager) is the responsibility of Cloudify, and is determined by Cloudify. Cloudify generates the necessary certificates for internal communication.

- SSL is enabled for agent-manager communication. All communications between the client the server are encrypted. When a connection is established, Cloudify Manager presents a signed certificate to the client. The client can use that certificate to validate the authenticity of the manager.

- A secret store is implemented inside the Cloudify PostgreSQL database, which provides a tenant-wide variable store.

- Security operations, such as authenticating success or failure and user details, are audited in dedicated log file on the management server.

Migrating from Cloudify 3.4.x to Cloudify 4.0.0

To migrate from Cloudify 3.4.x to Cloudify 4.0.0, you must create a snapshot of your 3.4.x machine, to upload to Cloudify 4.0.0. Follow this procedure perform the migration.

Migration Procedure

PREREQUISITE

Before taking the snapshot, verify that there are not any instances of node ID or deployment ID that include an underscore. For example, change `node_ID` to `node-ID`.

1. Take a snapshot of Cloudify 3.4.x according to the [installation instructions](#) in the Cloudify 3.4.x documentation.
2. Install Cloudify 4.0.0 according to the [installation instructions](#) in the Cloudify 4.0.0 documentation.
3. On the newly installed manager, upload and restore the 3.4 snapshot.
   You can restore a snapshot to a manager that does not have any data on it (clean), or to one with existing content.

   Note: The snapshot is uploaded to the tenant on which you performed the upload operation.

   Restore the uploaded snapshot into a specific tenant by specifying a new (unique) tenant name. The tenant is created as part of the restore process, and is populated with the snapshot content.
   - Restoration using the CLI is described [here](#).
   - Restoration from the UI is described [here](#).

New Features

The following new features are available in this release:

Bootstrapping/Installation

- [CFY-6380](#) - Cloudify agent installations on Windows are now located under %PROGRAM FILES%.
- [CFY-6092](#) - Elasticsearch has been replaced by PostgreSQL, which is installed during Cloudify Manager bootstrapping.
- [CFY-4837](#) - Bootstrapping can now run on ports other than port 22.
- [CFY-6145](#) - The bootstrap `admin` user cannot be deleted or modified.
- [CFY-6146](#) - Only the bootstrap admin can restore v4.x snapshots.

Cloudify Manager

- [CFY-6129](#) - `logstash jdbc` and `postgresql` plugins have been added to the Cloudify Manager resources file.
- [CFY-6000](#) - High Availability functionality has been added to Cloudify MANAGER to the manager
- [CFY-6075](#) - Database replication has been added as part of the HA configuration.

Security

- [CFY-6184](#) - LDAP passwords are no longer stored in the Cloudify DB.
- [CFY-6497](#) - Generate internal SSL certificates on Cloudify manager image first boot
- [CFY-6514](#) - Sync security configuration files
- [CFY-6365](#) - Automatically-generated SSL certificates are passed from Cloudify Manager to the agent during the agent's creation.
- [CFY-6370](#) - Permissions are verified for every file server request.
- [CFY-6435](#) - To enable secure communication between the agent and the Cloudify Manager, file server credentials are passed to the agent.
- [CFY-6364](#) - An SSL certificate for Cloudify Manager/agent communication is automatically generated during bootstrapping.

- [CFY-6369](#) - An API call has been created that validates the permission of a user to access a file server resource.
- [CFY-6367](#) - All internal (agent/mgmtworker to rest service) REST communications are now SSL encrypted over port 53333.
- [CFY-6492](#) - File server access is now possible only through port 53333 (with the suffix `/resources`) and is also SSL encrypted
- [CFY-6319](#) - You can generate random salt and secret key during bootstrapping.
- [CFY-6236](#) - Validate manager credentials when using `cfy profiles set/unset`
- [CFY-5948](#) - Enforce user authentication and authorization by default (replace flask_securest).

CLI

- [CFY-6236](#) - Cloudify Manager credentials are validated when using `cfy profiles set/unset`
- [CFY-6472](#) - A default maximum for workflow operation retries (60) has been specified.
- [CFY-6502](#) - The `cfy cluster set-active` command is supported
- [CFY-6349](#) - During boot-up, if the `set_manager_ip_on_boot` parameter is set to `true`, the Cloudify Manager image updates the private IP address of Cloudify Manager, assuming that `admin_password` has been specified.
- [CFY-6350](#) - The `get Cloudify version`, `get user tenants` and `get user token` requests can be implemented without a specific tenant having to be specified.
- [CFY-6382](#) - The service_name property has been added to the Cloudify agent `AgentConfig` data type
- [CFY-5821](#) - Snapshots are managed via PostgreSQL, but it is still possible to load older Elasticsearch-based snapshots.
- [CFY-5509](#) - cfy plugins upload support use of a URL
- [CFY-5296](#) - `cfy * list` supports sorting.
- [CFY-5451](#) - `cfy events list` now includes logs by default. You can use the `--no-logs` flag to hide logs.
- [CFY-6130](#) - `cfy profile` commands have been rearranged.
- [CFY-6135](#) - `cfy use` and `cfy profile's` have been consolidated.
- Context-dependent commands are now available, according to whether a manager is being used.
- `cfy init` is no longer required before `cfy use` or `cfy bootstrap`.

- The `cfy use` flag has been added to set user and ssh-key path.
- `cfy local` has been removed, to consolidate its commands with "manager" commands.
- `.cloudify` now resides under `~` unless an environment variable called `CFY_WORKDIR` is set, in which case it is switched to that dir.
- A profile is now created per manager. To that end, `cfy profiles` is introduced. It provides the `list`, `delete`, `import` and `export` commands.
- Importing and exporting profiles can include SSH keys via the `--include-keys` flag.
- `cfy init -r` no longer resets logger and coloring configuration. Use the `--hard` flag.
- Replace flags with positional arguments where necessary.
- `--skip-logging` is removed from `cfy init`.
- The `--debug` flag, which was an alias for `-vvv` is removed..
- Names for blueprints and deployments are always automatically generated, unless there is a blueprint or deployment already with the same name. Names can still be explicitly specified.
- Task retry related flags have been added to `cfy teardown`.
- `cfy blueprints publish-archive` has been deleted. `cfy blueprints upload` can now receive a path to a local yaml, a path to a local archive, or the URL of an archive. If an archive is is provided, the `--blueprint-filename` flag is now mandatory.
- `cfy inputs` has been added to enable local context inputs.
- `output` flags have all been renamed to `output-path`.
- all `--force`-like flags have been renamed to `--force`.
- Bootstrap validations are ignored when `--skip-validations` is provided in `cfy upgrade`.
- A `--skip-sanity` flag is now available for to the bootstrap process.
- `cfy install` and `cfy blueprints upload` can now receive either a local path to a yaml, local path to an archive, URL to an archive and a Docker/Vagrant-like string to a repository.
- `cfy profiles export` can now receive the `--include-keys` flag, which includes the SSH keys (if applicable) of the profiles in the archive. `cfy profiles import` imports any keys in the archive and places them in their original paths.
- `cfy profiles get-active` prints out the currently used profile. `cfy profiles list` now shows an asterisk next to the active profile.

- `cfy blueprints upload` and `cfy install` now auto-generates blueprint and deployment names only once per blueprint.
- `cfy blueprints package` has been added. This command packages a blueprint into a tar.gz or ZIP archive according to the OS. This archive is installable or uploadable via `cfy install` and `cfy blueprints upload`.
- Setting the env var CFY_MULTIPLE_BLUEPRINTS to `true` enables you to initialize multiple blueprints locally.

Orchestration

- [CFY-6389](#) - The `CloudifySettings` blueprint has been removed from Cloudify Manager images.
- [CFY-5665](#) - Remove transient deployment workers mode configuration from manager blueprints
- [CFY-6319](#) - You can generate random salt and secret key during bootstrapping.
- [CFY-6336](#) - The token API now returns a user's role.
- [CFY-6340](#) - You can work with the bootstrap admin role, even when LDAP is configured.
- [CFY-6206](#) - Support has been added to enable deletion of tenants/groups/users.
- [CFY-6096](#) - Support has been added for private resources and resource permissions.
- [CFY-6193](#) - `tenant` and `permission` responses have been added to the CLI.

API

- [CFY-6369](#) - An API call has been created that validates the permission of a user to access a file server resource.
- [CFY-6193](#) - `tenant` and `permission` responses have been added to the API.
- [CFY-6162](#) - To optimize performance, as a response from the REST API, the maximum value that can be set for a pages is 1000 records.

Issues Addressed

The following known issues have been addressed in this release:

Bootstrapping/Installation

- [CFY-6052](#) - Bootstrap no longer fails if a Cloudify Manager VM image doesn't contain `/root/.ssh`.
- [CFY-6510](#) -[CFY-6504](#) - Offline bootstrap no longer fails due to `dsl_resources` input.
- [CFY-5745](#) - If you attempt to bootstrap without `auth env` variables, an error message is displayed.

CLI

- [CFY-6003](#) - The redundant deployment-id argument has been deleted from `cfy deployments create`.
- [CFY-6005](#) - Error messages that are displayed when executing `cfy teardown -f --ignore-deployments` have been made clearer.
- [CFY-6006](#) - Windows CLI package upload blueprint from secured URL no longer fails on certificate verification.
- [CFY-5995](#) - When using `get_attribute` in secure mode, `dispatch evaluate_functions` no longer attempts to use `ctx` before it is set.
- [CFY-6510](#) - The message displayed in the event of a bug in which the manager does not return an argument that the CLI expected to get in the list response has been clarified.

Orchestration

- [CFY-6496](#) - Issues with security groups in Cloudify Manager blueprints have been addressed.
- [CFY-6473](#) - An infinite loop no longer occurs in dispatch if the workflow executor reaches an invalid state.
- [CFY-4233](#) - Unicode characters are not supported in blueprints. Strings in blueprints are now checked to ensure there are no unicode characters.
- [CFY-6156](#) - You can now successfully use `cfy blueprints upload` from GitHub.

- [CFY-5699](#) - The blueprint parsing error in which a circular `get_property` was detected even though it didn't exist has been addressed.
- [CFY-5971](#) - Backwards compatibility for installing agents on Windows hosts for pre-3.3.0 blueprints is now available.
- [CFY-5631, CFY-5561](#) - Dereferencing of anchors in blueprints has been fixed. The behavior was inconsistent, and different values were set each time.
- [CFY-6476](#) - The issue related to source plugin TAR creation when uploading a blueprint has been addressed.
- [CFY-6388](#) - The situation in which `logstash` failed to fetch logs and events that were written after exchanges from logs and events were deleted has been addressed.
- [CFY-6333](#) - The situation in which the `create snapshot` command failed when the snapshot file was too big has been addressed.
- [CFY-6437](#), [CFY-6341](#), [CFY-6128](#) - The issue related to creating and restoring snapshots has been addressed.
- [CFY-5493](#) - The scale workflow for scaling a node template that is part of a group and is contained in another node not part of that group has been fixed.
- [CFY-5589](#) - Workflow parameters types are validated before attempts to execute the workflow.
- [CFY-5711](#) - The validation of workflow parameters now operates as expected.

API

- [CFY-5371](#) - The issue in which filtering was not supported in REST service snapshots endpoints, has been addressed.
- [CFY-5838](#) - The issue in which `workflow_test decorator` (for uni-tests) was checking for the wrong directory path has been addressed.

General

- [CFY-6498](#) - It is now possible to specify a file path for `manager_resources_package`.
- [CFY-6196](#) - The error message returned when creating a user with an existing name has been clarified.
- [CFY-3902](#) - The Diamond agent restarts when Cloudify Manager is restarted.

- [CFY-5877](#) - The error message when the agent installer fails to connect to VM has been clarified.
- [CFY-5464](#) - Descending sort by time is supported in sorting search results.
- [CFY-5782](#) - Fix deployment update broken for deployments migrated from Cloudify 3.2.1
- [CFY-5453](#) - The issue in which a deployment's `updated_at` field was not updated during deployment update has been addressed.
- [CFY-5698](#) - The issue in which a management worker sporadically doesn't restart after reboot has been addressed.

Known Issues

The following issues are known to exist in this release:

Cloudify Manager

Multi-Tenancy

Limitations
- Roles can only be assigned to individual users, not user groups.
- Resource sharing between tenants is not available.
- You can only delete tenants or user groups if they are empty, and users if they are unassigned.
- LDAP is not supported for nested groups.
- You cannot customize password strength rules.

Issues
- [CFY-6660](#) - REST calls do not support `private resource` and `resource creator` fields.

Web User Interface

Limitations
- The interface only supports uploads of public resources.
- The `created-by` field is not displayed.
- High availability is not supported in the user interface.

- Multiple Cloudify Managers are not supported in the user interface.

- [CFY-6673](#) - Deletion of plugins is implemented using the `--force` flag. A plugin will be deleted even if it is in use.

CLI

- [CFY-6616](#) - If you delete a Cloudify Manager and you are using the CLI as a client, you must delete the CLI profile of the manager.
- [CFY-6646](#) - In the event that the private key used for installing and uninstalling an agent is passed in a blueprint as a file path on Cloudify Manager, it is not replicated in a cluster. Therefore, using that key to install from one Cloudify Manager and uninstall from another (after failover) will fail.
  Workaround:
  Copy the private key to the Cloudify Manager that is becoming activate.

REST API

- [CFY-6660](#) - If you make a REST API call using the `_include=created_by` argument, you will receive a corrupted response.