

# Cloudify 4.1

## Release Notes

### Contents

What's New?	2
Cloudify Manager	2
Web User Interface	2
Cloudify Composer	2
Upgrading to Cloudify 4.1	3
New Features	4
Cloudify Manager	4
Cloudify Manager User Interface	4
Security	4
CLI	5
Cloudify Composer	5
API	5
Issues Addressed	5
Bootstrapping/Installation	5
CLI	5
Orchestration	5
Cloudify Composer	6
API	6
General	6
Known Issues	7
Upgrading Cloudify Manager	7
Cloudify Web Interface	7
High Availability	7
Cloudify Composer	9

## What's New?

### Cloudify Manager

- You can now explicitly specify whether plugins are to be installed from source during deployment creation.
- You can use the [teardown](#) command in the command line interface to remove Cloudify Manager and all its resources from a VM.
- Cloudify Manager no longer requires a root user for its operation.
- You can [specify resources as private](#) to increase isolation within tenants.
- To enhance multi tenancy isolation and boundaries, agents belonging to one tenant cannot trigger operations on another tenant. Achieved through RabbitMQ isolation between tenants.

### Web User Interface

- You can now create your own custom widgets to assist you in displaying your data in a custom manner, or can integrate the Cloudify UI with other components in your architecture.

Widgets can be written using two different methods.

- Using the React Utility is the recommended method, and requires a build operation to be executed. You can build the `widget.js` file yourself, or use the Cloudify build system.
- Pure Vanilla JavaScript which enables attachment of an HTML template file. The callbacks for this method are described later in this topic.
- A custom widget environment is now available on which you can develop and test your widgets.

### Cloudify Composer

- Cloudify Composer is now integrated into Cloudify Manager, eliminating the need for separate installation and making the process of uploading composed blueprints to the Manager easy and intuitive.

## Upgrading to Cloudify 4.1

To upgrade from Cloudify 4.0.x to Cloudify 4.1, you must create a snapshot of your 4.0.x machine, and upload it to a tenant on Cloudify 4.1.0. Before you start, it is important that you [review the upgrade options](#). Note also that snapshots can only be restored to clusters that have only a single node.

---

**Note:**

Before beginning the upgrade process, please review the [Known Issues](#) section at the end of these release notes.

---

- Restoration using the CLI is described [here](#).
- Restoration from the UI is described [here](#).

### Migration Procedure

If you are migrating **from Cloudify 3.4.x to Cloudify 4.1**, follow this procedure to perform the migration.

#### 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 4.1.0 according to the [installation instructions](#) in the Cloudify 4.1.0 documentation.
3. On the newly installed manager, upload and restore the 3.4.x 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:

### Cloudify Manager

- [CFY-6876](#) - To enhance multi tenancy isolation and boundaries, agents belonging to one tenant cannot trigger operations on another tenant.
- [CFY-6900](#) - Snapshots from versions 3.4 and later can now be restored to newer versions of Cloudify.
- [CFY-6474](#) - You can now explicitly specify whether plugins should be installed from source on deployment creation.
- [CFY-6870](#) - When a snapshot that includes live agents is restored, the certificates are replaced, to enable communication with pre-existing agents.
- [CFY-6899](#) - When you restore a snapshot with a certificate, the VM is automatically rebooted, unless the `no_reboot` flag was supplied.
- [CFY-7017](#) - Validation of the minimal available memory can now be configured or disabled.

### Cloudify Manager User Interface

- [CFY-6931](#) - Changes made using the Manager UI that are related to custom widgets and images are saved in snapshots, and therefore reflected in an upgraded Manager.
- [CFY-6908](#) – User interface files, such as widgets and images are saved in HA replication and snapshots.
- [STAGE-32](#) – A custom widget environment is now available.
- [STAGE-126](#) - You can now select the main blueprint file from a list during the upload process.

### Security

- [STAGE-237](#) – Blueprints and deployments can be defined as private so they are only available to the user who created the resource.
- Cloudify Manager no longer requires a root user for its operation.
- Resources can be defined as private for further isolation within tenants.
- [CFY-6876](#) - To enhance multi tenancy isolation and boundaries, agents belonging to one tenant cannot trigger operations on another tenant. Achieved through RabbitMQ isolation between tenants.

## CLI

- [CFY-6722](#) - Multiple local profiles are supported by default.

## Cloudify Composer

- [COMPOSER-731](#) - Cloudify Composer is now integrated into the Cloudify Manager, eliminating the need for separate installation and making the process of uploading composed blueprints to the Manager easy and intuitive.

## API

- [CFY-6474](#) - You can now explicitly specify whether plugins should be installed from source on deployment creation.

## Issues Addressed

The following known issues have been addressed in this release:

### Bootstrapping/Installation

- [CFY-6948](#) - When enabling SSL on bootstrap, `CLOUDIFY_SSL_TRUST_ALL` is no longer ignored for CLI commands.
- [CFY-6901](#), [CFY-6894](#), [CFY-6934](#) - The issue in which you could not bootstrap a second Manager on a cluster, or add an existing second Manager to a cluster, has been addressed.

## CLI

- [CFY-6934](#) - ``cfy profiles use`` breaks without `--rest-port`.
- [CFY-6930](#) – The issue in which you could not rejoin a cluster using the CLI has been addressed.

## Orchestration

- [CFY-6869](#) - Migration file created with `drop_index`
- [CFY-6874](#) - The issue in which the `get_property` intrinsic function did not look inside data types when `address` is of a declared type, has been addressed.
- [CFY-6933](#) - The issue in which a small number (~30) of deployments could not be created simultaneously has been addressed.

## Cloudify Composer

- [COMPOSER-506](#) - The issue in which changes to a custom node type were not reflected in an already existing instance of that type is now resolved.
- [COMPOSER-598](#) - The issue in which, when renaming floating IP/security group names, the new node names were not updated in attached compute network section, is now resolved.
- [COMPOSER-602](#) - An autosave indicator has been added.
- [COMPOSER-626](#) - Container type instances are now able to reside inside a compute node type.
- [COMPOSER-659](#) - Composer now prevents the addition of a group as a member of itself.
- [COMPOSER-706](#) - The issue in which changes to derived properties of custom types were not being saved is now resolved.
- [COMPOSER-710](#) - The issue in which changes to a custom node's operations that are derived from a parent node was not saved is now resolved.
- [COMPOSER-747](#) - The issue in which an input or a custom relationship's property that was defined with a default Boolean value of "false" was saved with a "true" value, is now resolved.
- [COMPOSER-709](#) - Custom nodes that derive from a node type defined in an imported file are now deleted when that import is removed from the blueprint.

## API

- [CFY-6898](#) - The `get version` API call now returns the correct edition value.

## General

- [CFY-6864](#) - The issue in which a cluster comprising a single node could not be restored has been resolved.
- [CFY-6867](#) - The issue in which the cluster `start` command did not work has been resolved.
- [CFY-6914](#) - The `create` and `restore snapshot` commands are now working correctly.
- [CFY-6913](#) - The issue in which a snapshot of a Cloudify 3.4 instance that was restored to Cloudify 4.0.1 reported success but was not successfully restored has been addressed.
- [CFY-6652](#) - The timestamp in logs has been changed system time.

- [CFY-6962](#) - The issue in which a Diamond plugin did not use the correct configuration following a restart has been addressed.
- [CFY-6954](#) - The issue in which when upgrading an agent, the Diamond plugin service file was not updated to point to the new agent has been addressed.
- [CFY-6942](#) - When SSL is enabled on the Manager, the issue in which, in some circumstances, traffic was not redirected internally from port 80 to 443 has been addressed.
- [CFY-6932](#) - Handle restore snap with plugins to 4.0.1 (script)

## Known Issues

The following issues are known to exist in this release:

### Upgrading Cloudify Manager

- SSH key files on tenants other than the default tenant are not included in the snapshot creation and restoration process. It is recommended that you [use secrets](#) to store this data, ensuring a proper upgrade.  
**Workaround:** When you are creating your snapshot of Cloudify Manager 4.x, you can pass the `-exclude-credentials` flag. This prevents the store/retrieve process from being touched. You can then recreate the keys from the original Cloudify Manager in the same locations on the new Cloudify Manager.
- An agent upgrade is not automatically performed on tenants, other than the default tenant.  
**Workaround:** To upgrade the agent on a tenant other than the default, run  

```
cfy agents install -t TENANT_NAME
```
- Before taking a snapshot of a Cloudify Manager version 4.0.x, it is important that you run the process described in the documentation including tearing down existing Manager. During the process, essential patches are applied.

### Cloudify Web Interface

- [STAGE-397](#) – A user without `admin` permissions cannot see a tenant's secrets in the user interface. To use secrets, a user without `admin` permissions must use the CLI.

### High Availability

- [CFY-7039](#) – If, when you start a cluster use the floating IP of a Manager(default), the required ports are not open, an error message is not displayed. It will appear that the cluster has started correctly.  
**Workaround:** After you have started a cluster, view the Nodes list and verify that the node is online and does not have a `FAILED DB` status.

- [CFY-7030](#) – If you start a cluster with a single Manager instance, and that Manager becomes unavailable, when you attempt to connect a new (second) Manager to the cluster, an error message is returned that there is no active node in the cluster.  
**Workaround:** Address the issues that made the Manager unavailable. You can now connect to the original second Manager.
- [CFY-6906](#) – This issue relates to the situation in which you have created a cluster from an image and you are joining a second Manager, which was bootstrapped to the cluster. When you join the second Manager to the cluster and run `cfy --version`, the version list *incorrectly* shows that you are connect to the bootstrapped (passive) Manager. If you run `cfy cluster node list`, the list *correctly* shows the active Manager as being active and that you are connected to it.
- [CFY-6868](#) – This issue occurs when upgrading from an earlier version to a later version. If you bootstrap a Manager and start a cluster, then tear down the Manager, and bootstrap a new Manager, when you attempt to start a cluster on the new Manager, an error message is returned.
- [CFY-6859](#) – If you have a cluster of two Cloudify Manager instances and you remove the active Manager, when you join a third Manager to the cluster (of which the second instance is now the active Manager), an internal server error is returned.
- [CFY-6813](#) – This issue relates to the situation in which you have created a cluster and have then torn down that active Manager. If you use the CLI to switch back to a profile that includes the recently torn down Manager, you erroneously receive a message that the Manager cannot be used because there is no active node in the cluster.
- [CFY-6822](#) – This issue relates to the situation in which you have two Managers in a cluster. If you remove the passive Manager from the cluster, and then remove the active Manager, you can still upload a blueprint and create resources on what was the active Manager, even though it is now outside of the cluster. If you attempt to start a new cluster with this Manager, or to join it to another cluster, an error message is returned



informing you that the Manager is already part of a cluster, even though the cluster node list is empty.

- [CFY-7019](#) – When you are uploading a plugin to a Manager in a cluster, instead of the plugin upload confirmation message, you erroneously get a message that there was a timeout in the process.
- [CFY-6821](#) – You can join a Manager that already has resources (users, tenants, or plugins) on it to a cluster and set it as the active Manager, although a Manager should be clean before it can be joined to a cluster.

## Cloudify Composer

- In the blueprints catalog on the **Imports** tab, the default `types.yaml` file is located in the `http://www.getcloudify.org/spec/cloudify/4.1m2/types.yaml` directory. There are no changes to the functionality of the YAML file.
- In the out-of-the-box catalog, the OpenStack plugin is not included. You can add it by clicking **Import** and entering the link to the appropriate plugin. To identify the correct link, open the [Cloudify downloads page](#).
- You cannot add an output using Cloudify Composer. Open the blueprint in a text editor and add the output directly in the blueprint.