Cloudify 6.2.0 Release Notes

September 20th, 2021

Cloudify 6.2.0 release contains new features and fixes on top of Cloudify's v6.0.
Version 6.2.0 can be deployed as a new installation, as an in-place update over a 5.2.x/6.x.x Cloudify manager and as an upgrade version for any supported previous Cloudify release.
See the upgrade section below for more details.

6.2 New Functionality And Improvements

New Functionality

Version 6.2 contains functionality improvements and extensions to features introduced in version 6.1. Improvements were performed in the following areas:

1. UI & UX Improvements
    - Enhancements to Dashboard
    - Blueprint marketplace UX update
    - Blueprints Have a User friendly display label
    - Added ability to hide inputs when creating deployments
    - Filters autocomplete for default filter and deployment display name search
    - Enhancements to deployments view
    - Automated plugin upload on blueprint upload
2. Security
    - Security auditing – auditing activity in Cloudify
3. Blueprints DSL
    - Intrinsic functions to handle properties of scaled nodes
4. Kubernetes:
    - New Kubernetes ‘Getting Started’ guide
    - Discovery for GKE
    - Deploy-On example with EKS
    - Additional examples in catalog
5. Plugin Enhancements
    - Plugins upload UI and installation UI improvements
    - Ansible plugin support for Galaxy collections
    - AWS Plugin Enhancements
6. Installation & maintenance
    - Helm enhancements and documentation

Other Corrections & Fixes

1. Authentication & Audit
   Correction to enable support for OCTA groups
   Example section for LDAP configuration
2. Orchestration at Scale
   Enchantments to Deploy-On deployment naming
3. Deployment Lifecycle
   Enhancements to Deployment update
4. Performance improvements
   Optimization of workflow task handling

5. Workflow Engine
   Correction to resuming workflows with subgraphs
6. Installation & Maintenance:
   Correction of process installation with existing database;
   Snapshot corrections for deployments with large amount of node instances;
   Correction to uninstall cleanup.

6.1 New functionality and improvements

New functionality

Version 6.1 contains functionality improvements and extensions to features introduced in version 6.0, the improvements were performed in the following areas:

1. Filters
   Filters now can be updated and defined dynamically on the environments and service view pages
2. Batch Executions
   Improvements of batch executions: backend and Interface
   Performance improvements of executions
3. Deployment display name
   Extended support for UI widgets to display the deployments display name
4. Authentication improvements
   Documentation, improved error messages, and roles improvements for LDAP
5. Environment Discovery
   Discovered Kubernetes environments now support token refresh
6. General
   IPV6 Support - adjustment of internal service for IPV6 environments

Other Corrections & Fixes

1. High Availability and Load Balancing Clustering
   Fixed: cluster teardown fails with SSL
   Improved logging for cluster manager
   Fixed: Patroni permissions on the data directory
   Remove cluster manager RPM on clean up
   Fixed installing cluster without SSL
   Fixed log download issue in 3 node cluster configuration
2. Manager UI
   Improvements to deployment display, topology and map view
   Topology zooming improvements

3. Plugins Related
    Terraform plugins directory structure support
    Fixed: rendering issues for Terraform topology
4. Manager Backup Snapshots
    Fixed: snapshot issue resulting in an error with high node count

6.0 New Functionality

The key theme of v6.0 is **Workload orchestration at scale**.
Providing complete governance over single service or environment orchestration had been the focus of Cloudify in previous versions, and we take pride in our abilities to provide complete automation and day 2 management to any service, simple or complex, over every combination of infrastructure, platform, and scenario.
Real life scenarios, however, don't stop at that. With new Cloud native and agile approaches, services are deployed over thousands of locations and require ongoing maintenance regardless of the underlay.
Cloudify 6.0 takes that to the next level with the ability to deploy, manage, and maintain workloads over thousands of locations in one action. Installation, update, configuration, modification, removal, and all other workflows can be executed over any number of services, based on a smart placement policy that takes into account any meaningful criteria - OS, architecture, location, version are just a few examples.

Environment as a service 2.0

Hierarchy-based environments/services new pages allow for a clear view of your entire system, with the ability to tune it to your organization needs and set it up automatically. Automatic discovery of environments and the ability to deploy a service onto multiple environments (potentially tens of thousands) unleash the operator’s power to run multi-geography services as easily as deploying a local one.

Characterize your services and environments and apply smart placement policies

This major theme introduces enhancements to the way services and environments are described in the Cloudify database. By automatically/manually assigning labels to the objects, one may assign any set of criteria to their deployments and use that info to slice and dice their objects by any criteria that make sense. This allows for smart placement policies as well as mass day2 operations over any group of services.

Operability

With Cloudify 6.x, managing Cloudify is more granular and easy than before. Recurrent workflows, usability improvements, onboarding wizard, and extended developer options are just some of the enhancements applied.

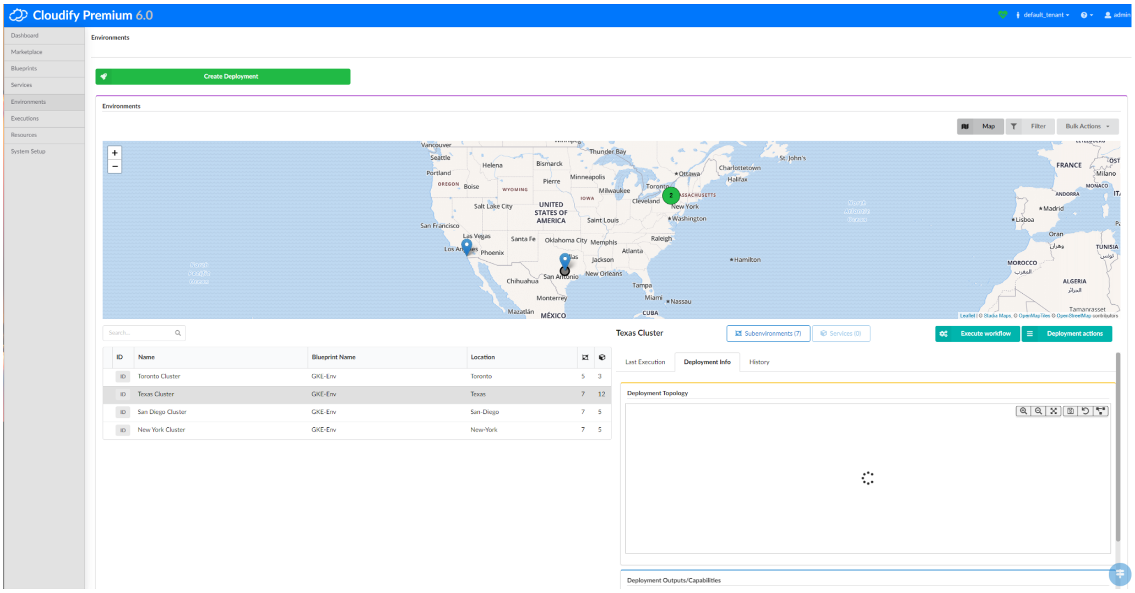
Cloudify 6.0 is a major Cloudify release containing over 300 developed stories and 350 resolved issues and tasks, improving all aspects from functionality to robustness and security.

Environment as a Service (EaaS) 2.0

Building on top of the foundations set in previous releases, Cloudify v6.0 includes dedicated `Environments` views and actions.

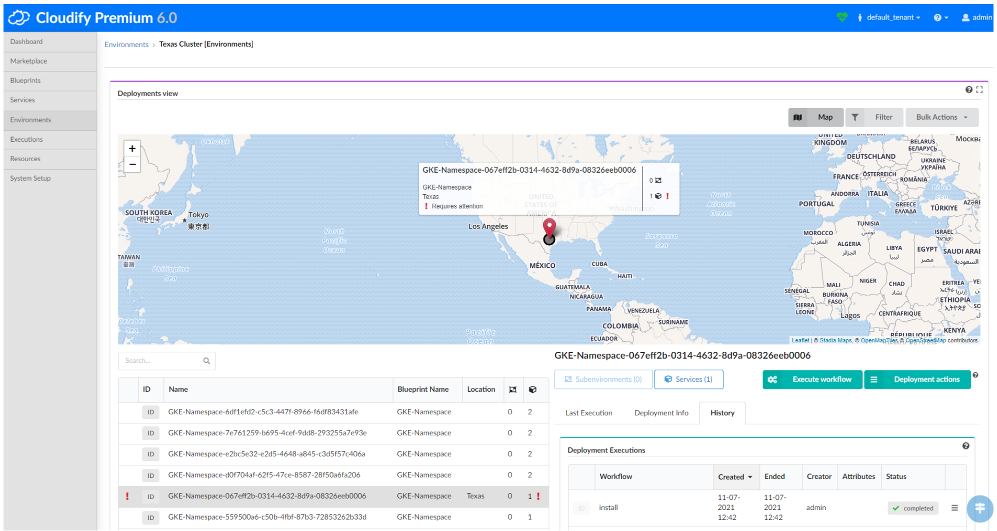
Hierarchy based environments view

The new environments page was designed to be the operator’s one-stop-shop for data, monitoring and actions. It features three synchronized panes allowing quick identification of any service requiring attention based on its state, detecting data-center or regional issues, quick access to the complete details of the environment or service deployment details, and immediate action and workflow execution.



ID	Name	Blueprint Name	Location		
ID	Toronto Cluster	GKE-Env	Toronto	5	3
ID	Texas Cluster	GKE-Env	Texas	7	12
ID	San Diego Cluster	GKE-Env	San-Diego	7	5
ID	New York Cluster	GKE-Env	New-York	7	5

Each environment shares the aggregated state of all of its sub-environments and services which allows quick identification of issues. Drill-down to the root cause is done via the same page, and corrective actions are taken via the embedded workflow/action options.



	ID	Name	Blueprint Name	Location		
	ID	GKE-Namespace-6df1efd2-c5c3-447f-8966-f6df83431afe	GKE-Namespace		0	2
	ID	GKE-Namespace-7e761259-b695-4cef-9dd8-293255a7e93e	GKE-Namespace		0	2
	ID	GKE-Namespace-e2bc5e32-e2d5-4648-a845-c3d5f57c406a	GKE-Namespace		0	2
	ID	GKE-Namespace-d0f704af-62f5-47ce-8587-28f50a6fa206	GKE-Namespace		0	2
!	ID	GKE-Namespace-067eff2b-0314-4632-8d9a-08326eeb0006	GKE-Namespace	Texas	0	1 !
	ID	GKE-Namespace-559500a6-c50b-4fbf-87b3-72853262b33d	GKE-Namespace		0	1

GKE-Namespace-067eff2b-0314-4632-8d9a-08326eeb0006

	Workflow	Created	Ended	Creator	Attributes	Status
ID	install	11-07-2021 12:42	11-07-2021 12:42	admin		✓ completed

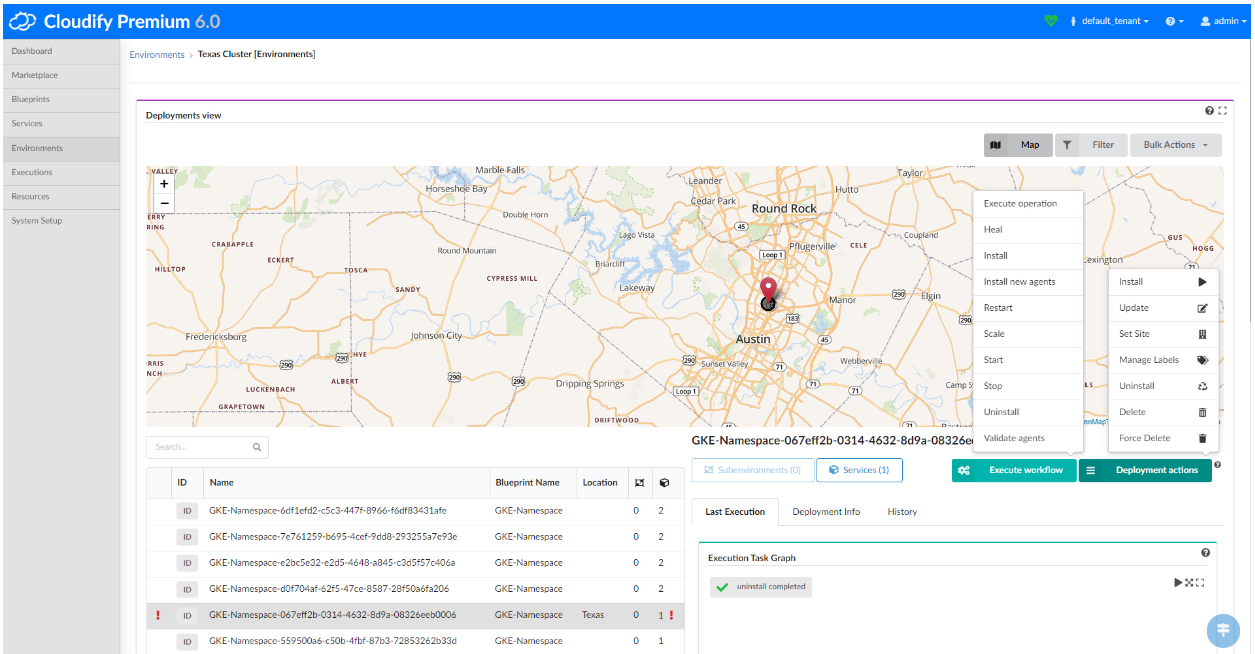
Map view

Calgary
Regina
Winnipeg
Vancouver
Seattle
Thunder Bay
Helena
Bismarck
Portland
Ottawa
OREGON
Boise
Minneapolis
Pierre
WYOMING
Milwaukee
Toronto
IOWA
2
Cleveland
Salt Lake City
UNITED STATES OF AMERICA
New York
Washingt
San Francisco
Saint Louis
Las Vegas
Santa Fe
Oklahoma City
Memphis
Raleigh
Los Angeles
Phoenix
Atlanta
Dallas
Jackson
San Antonio
New Orleans
Chihuahua
Tampa
Monterrey
Miami
Nassau

The map view was adjusted to support tens of thousands of locations and hundreds of thousands of services and environments. It is optimized for anything from several regions, branches or data centers, to huge edge networks.
The operator view allows quick focus on the area or location of interest while reducing map clutter by aggregating and grouping items.

Embedded actions

Run any day1 or day2 operation directly from the main view. See your updates take place in real-time. Resolve issues and track the impact from a single view.



	ID	Name	Blueprint Name	Location		
	ID	GKE-Namespace-6df1efd2-c5c3-447f-8966-f6df83431afe	GKE-Namespace		0	2
	ID	GKE-Namespace-7e761259-b695-4cef-9dd8-293255a7e93e	GKE-Namespace		0	2
	ID	GKE-Namespace-e2bc5e32-e2d5-4648-a845-c3d5f57c406a	GKE-Namespace		0	2
	ID	GKE-Namespace-d0f704af-62f5-47ce-8587-28f50a6fa206	GKE-Namespace		0	2
!	ID	GKE-Namespace-067eff2b-0314-4632-8d9a-08326eeb0006	GKE-Namespace	Texas	0	1 !
	ID	GKE-Namespace-559500a6-c50b-4fbf-87b3-72853262b33d	GKE-Namespace		0	1

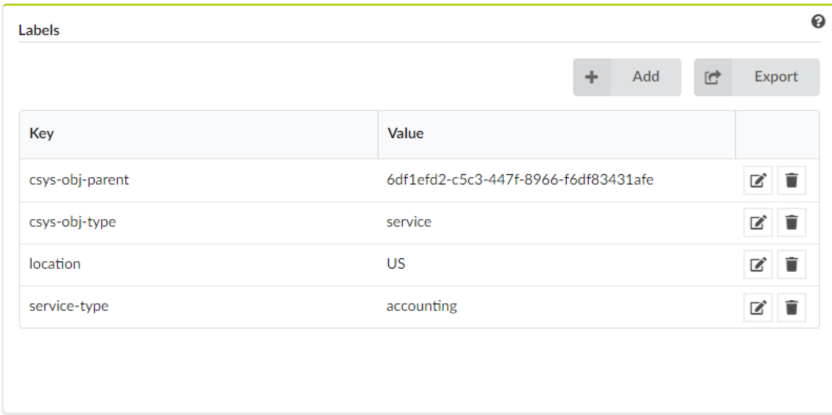
Learn more about the new Deployments view

Orchestration at scale

Installing a single instance of an application or a service can be as easy or as complex as the service design and the infrastructure used. Running that service or app over 10K locations is more challenging.
With Cloudify 6.0 you can build your own placement policies by any criteria, and push both day1 and day 2 operations over any set of environments and services.
Installing a new service in all of your US branches, or running a certificate refresh over all of your production load balancers are just two examples of the new bulk actions.

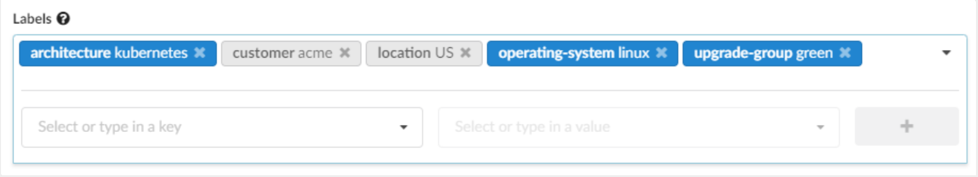
Labels & filters

Labels are sets of key+value which can be assigned with every Cloudify deployment. Labels are completely granular and any set of labels can be assigned to a deployment or to a set of deployments.
For example: one may assign labels to characterize the geo-details (region="US East", location=NY), the architecture (platform=K8S, OS=linux), metadata (type=environment, upgrade-group=green), and so on.



Key	Value
csys-obj-parent	6df1efd2-c5c3-447f-8966-f6df83431afe
csys-obj-type	service
location	US
service-type	accounting

Labels can be assigned and maintained at any phase of the lifecycle - A developer may set them as part of the blueprint, a user may specify them during deployment, and an admin may manage labels post-deployment during its existence.

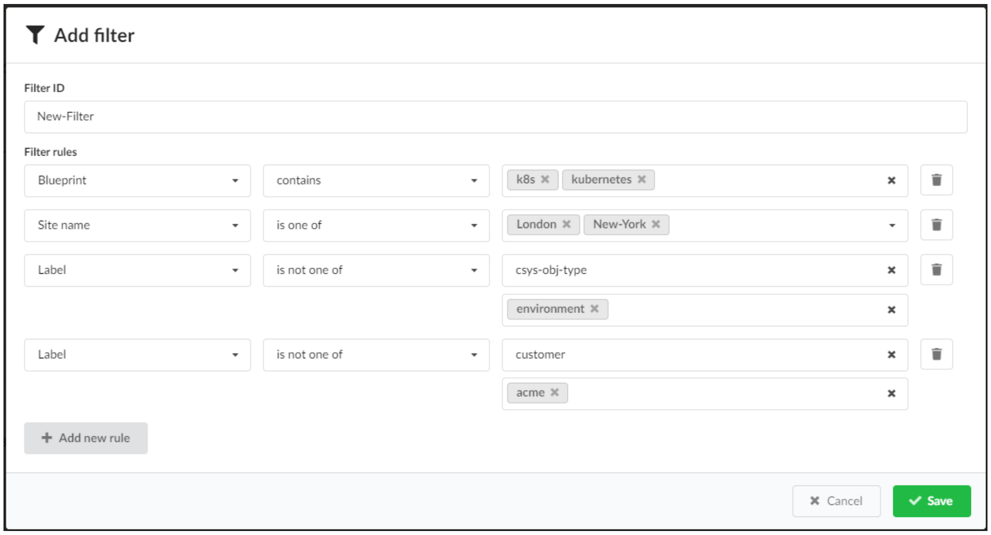

Labels

architecture kubernetes ✖ customer acme ✖ location US ✖ operating-system linux ✖ upgrade-group green ✖

Labels allow easy characterization and make Cloudify a single source of truth regarding the criteria, removing the need to maintain tables/files with various lists of services.
Learn more about labels and the different options to manage them

Once labels are applied, **Filters** allow the user to group deployments by any combination of properties and thus slice and dice their services and environments to detect all items that comply with the filter's logic.

Filters support multiple rules by various aspects such as labels, site, blueprint & owner, and a complete set of operators and logic. The user may save the filters and reuse them.



Add filter

Filter ID: New-Filter

Filter rules

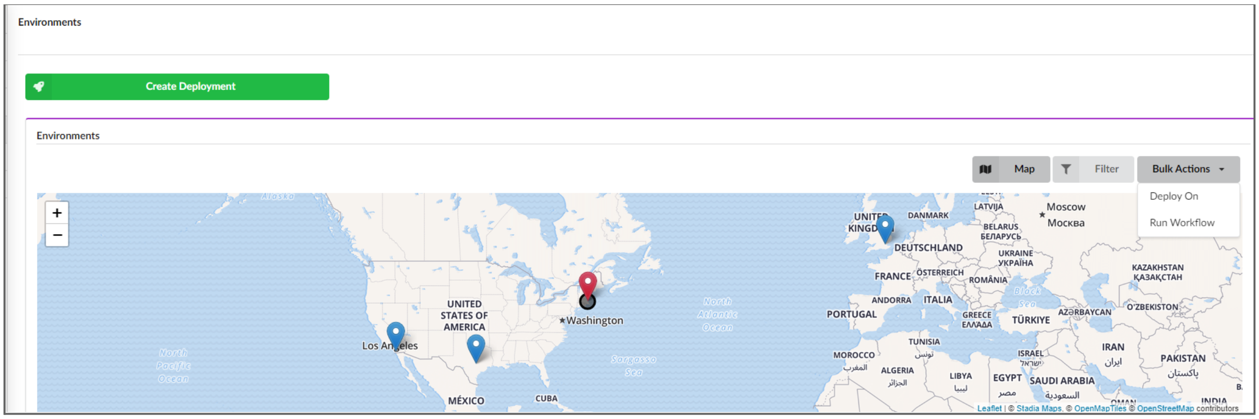
Blueprint	contains	k8s ✖ kubernetes ✖
Site name	is one of	London ✖ New-York ✖
Label	is not one of	csys-obj-type environment ✖
Label	is not one of	customer acme ✖

Filters can be used in the environments and Services views, but more importantly, filters are the basis for the bulk actions, allowing the user to deploy a service onto multiple environments or run a day2 operation over multiple services.
By selecting a filter, the user defines the list of systems onto which they wish to run the bulk actions, which makes for an extremely flexible placement policy.

Learn more about [filters]

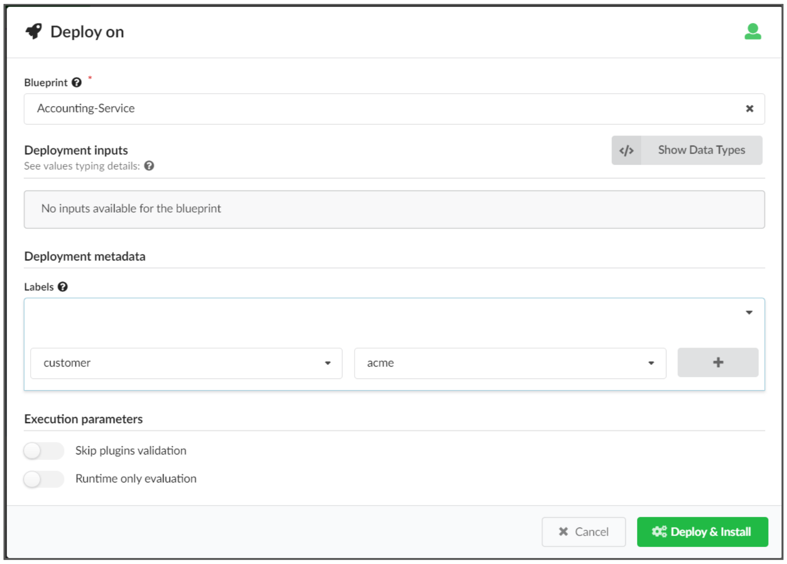
Bulk actions

6.0 introduces two types of bulk actions - Deploy on & Run workflow



Deploy on allows the user to deploy a service onto multiple environments. For example, the user may select all of their EKS clusters that are used for staging as the list of environments, and use the *Deploy on* bulk action to deploy the latest version of their application using a helm chart over these clusters.

Cloudify will leverage the filter and the labels to generate the proper list of environments, and then iterate the installation of the service while making the correlation between each environment and the service deployed on it. Using this approach, the user will not need to maintain the different environments credentials or details. Each environment will declare its properties such that the placement will be done automatically.


Deploy on

Blueprint *
Accounting-Service

Deployment inputs
See values typing details:

No inputs available for the blueprint

Deployment metadata

Labels
customer
acme

Execution parameters
Skip plugins validation
Runtime only evaluation

Run Workflow allows the user to apply a day2 operation over a large set of services/environments. After selecting the target list of deployments leveraging the new filters, the user selects the required day 2 operation, for example - run a security patch, and Cloudify will apply the relevant workflow on each of the environments based on the specific workflow defined for that service/environment.

Deploy on & Run Workflow bulk actions are available through API, CLI, and Web UI. The results of both can be tracked via the execution widget.

Learn more about bulk actions.

Environments discovery

Populating the Cloudify DB with hundreds or thousands of environments is an important step on the path towards large-scale orchestration. Of course, such actions can be done manually or via a scripted flow, however, these options are cumbersome and error-prone.
Cloudify 6.0 extends the options for auto-population of environments by adding an environment discovery automated workflow.
The new workflow scans existing services and retrieves the environment information into the Cloudify DB, thus allowing the user to discover their set of Amazon Kubernetes clusters (for example) and creating an environment in Cloudfy per EKS cluster. Each discovered system is then labeled automatically for its region, location, flavor, and more. Information such as the

connection-token is also retrieved and securely stored to allow direct and seamless orchestration of services over this environment.

Once environment discovery is done, the system has all the required details to allow for a smart placement policy over these clusters (by criteria), and the required credentials allowing seamless installation or day 2 operation over these environments.

Discovery can be executed as a recurrent workflow, it will detect newly added systems and import them into Cloudify while identifying already imported systems to keep an updated picture at all times.

Version 6.0 provides out-of-the-box support for EKS and StarlingX discovery. These can easily be extended for more platforms and infrastructures via their dedicated plugins and follow-up releases will include support for AKS discovery, OpenShift discovery, and multiple more platforms.

Operability

Scheduled workflows

Scheduling a workflow had been introduced in Cloudify 4.6, allowing users to set expiration time to installed systems (scheduled uninstall), run workflows at specific time windows, and more.
Cloudify 6.x adds recurrent schedules - allowing the user to setup any workflow to be executed on a regular basis at a specified time. Backing up systems, refreshing keys and tokens, scheduling maintenance patches, or setting up systems to be deployed and removed every weekend can now be easily set.
This provides both additional required functionality, and potentially a large cost reduction by using resources only at the time they are needed.

6.0 further extends the developer abilities to set the scheduling as part of the blueprint definition, thus any deployment generated from that blueprint will spawn its scheduled workflows during its deployment.

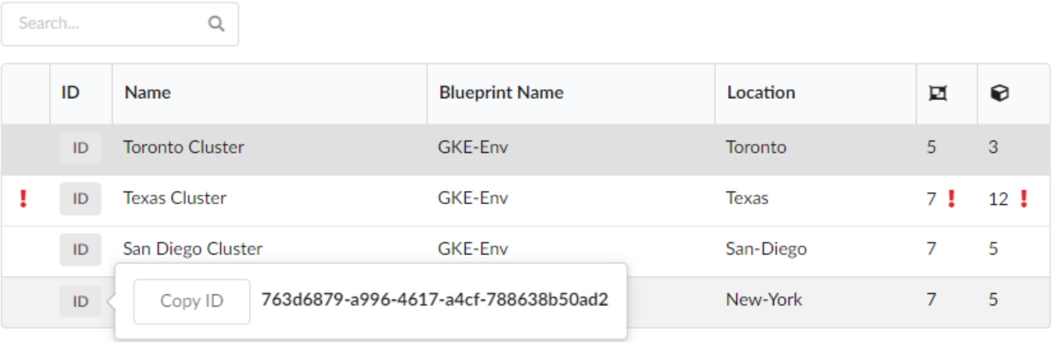
```
deployment_settings:
  default schedules:
    scheduled-install:
      workflow: install
      since: '2023-1-1 18:00'
      timezone: 'EST'
      count: 1
    scan-twice-a-week:
      workflow: update-scan
      since: '+4h'
      until: '2022-1-1 15:00'
      recurrence: 1w
      weekdays: [tu, we]
      count: 5
      slip: 30
    backup:
      workflow: backup
      since: '2022-1-1 15:00'
      timezone: 'Israel'
      recurrence: 1mo
      weekdays: [l-fr]
      count: 10
```

The scheduling structure is extremely flexible and allows for frequent (every minute) recurrences, minutely, hourly, daily, weekly, and monthly options with smart flags (e.g. last monday of the month, how to handle failure and retry, how to handle downtime during scheduled execution, and more).

The workflow scheduling mechanism was updated and it is now running as a dedicated service.

Deployment Display name

Cloudify deployment objects are identified by a unique ID through which other deployments may access the details of this deployment, users may search and detect that deployment.
In version 6.0 an additional property - Display Name - is added to the deployment and presented next to the ID.
The display name may consist of any UTF8 character and contain spaces and special characters that are not allowed as part of the ID.
In various reports and views, the user may now filter by the display name as well as the ID.

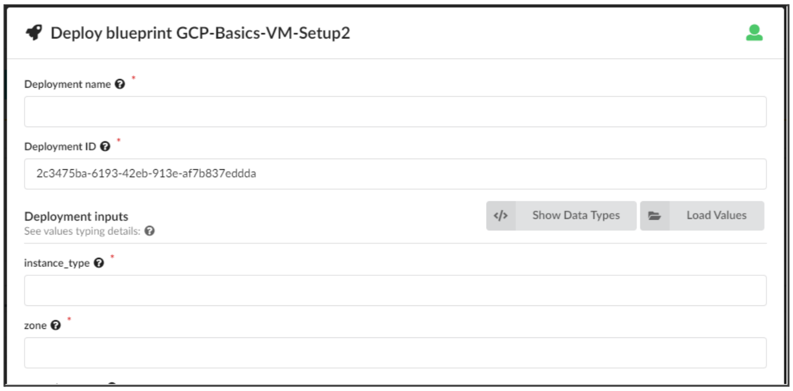


Search...

	ID	Name	Blueprint Name	Location		
	ID	Toronto Cluster	GKE-Env	Toronto	5	3
!	ID	Texas Cluster	GKE-Env	Texas	7 !	12 !
	ID	San Diego Cluster	GKE-Env	San-Diego	7	5
	ID			New-York	7	5

763d6879-a996-4617-a4cf-788638b50ad2

Upon deployment creation, it is optional for the user to submit a unique ID for the deployment. If they do not, the system allocates a UUID.



Deploy blueprint GCP-Basics-VM-Setup2

Deployment name *

Deployment ID *

2c3475ba-6193-42eb-913e-af7b837eddda

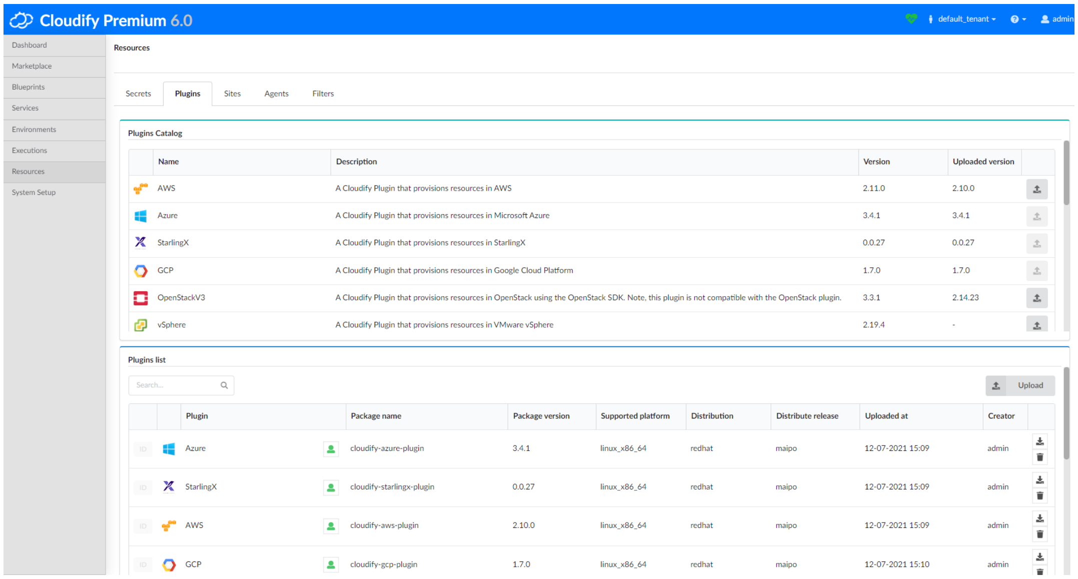
Deployment inputs
See values typing details:

instance_type *

zone *

Menu updates

Version 6.0 introduces an update to the settings and resources options on the left menu.
All system resources options are now grouped and managed as tabs under the ***Resources*** menu option. The tabs include Secrets, Plugins, Sites, Agents, and Filters.



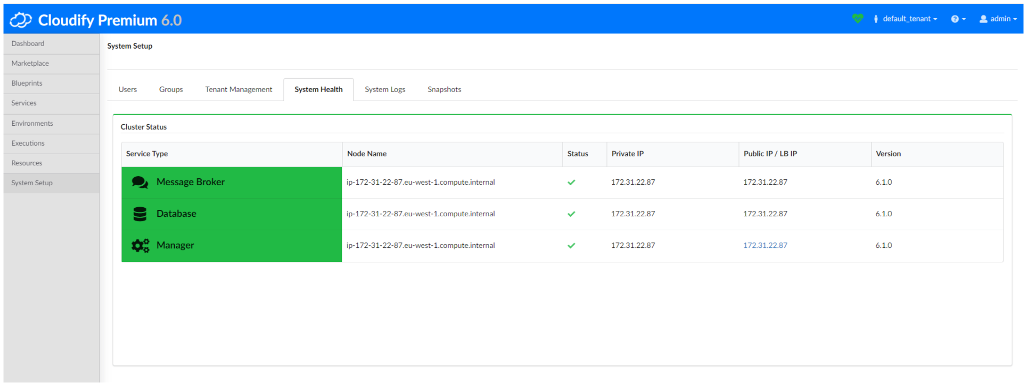
Plugins Catalog

Name	Description	Version	Uploaded version
AWS	A Cloudify Plugin that provisions resources in AWS	2.11.0	2.10.0
Azure	A Cloudify Plugin that provisions resources in Microsoft Azure	3.4.1	3.4.1
StarlingX	A Cloudify Plugin that provisions resources in StarlingX	0.0.27	0.0.27
GCP	A Cloudify Plugin that provisions resources in Google Cloud Platform	1.7.0	1.7.0
OpenStackV3	A Cloudify Plugin that provisions resources in OpenStack using the OpenStack SDK. Note, this plugin is not compatible with the OpenStack plugin.	3.3.1	2.14.23
vSphere	A Cloudify Plugin that provisions resources in VMware vSphere	2.19.4	-

Plugins list

Plugin	Package name	Package version	Supported platform	Distribution	Distribute release	Uploaded at	Creator
Azure	cloudify-azure-plugin	3.4.1	linux_x86_64	redhat	maipo	12-07-2021 15:09	admin
StarlingX	cloudify-starlingx-plugin	0.0.27	linux_x86_64	redhat	maipo	12-07-2021 15:09	admin
AWS	cloudify-aws-plugin	2.10.0	linux_x86_64	redhat	maipo	12-07-2021 15:09	admin
GCP	cloudify-gcp-plugin	1.7.0	linux_x86_64	redhat	maipo	12-07-2021 15:10	admin

The new ***System Setup*** menu item includes tabs for the following: Users, Groups, Tenant Management, System Health, System Logs, and Snapshots.



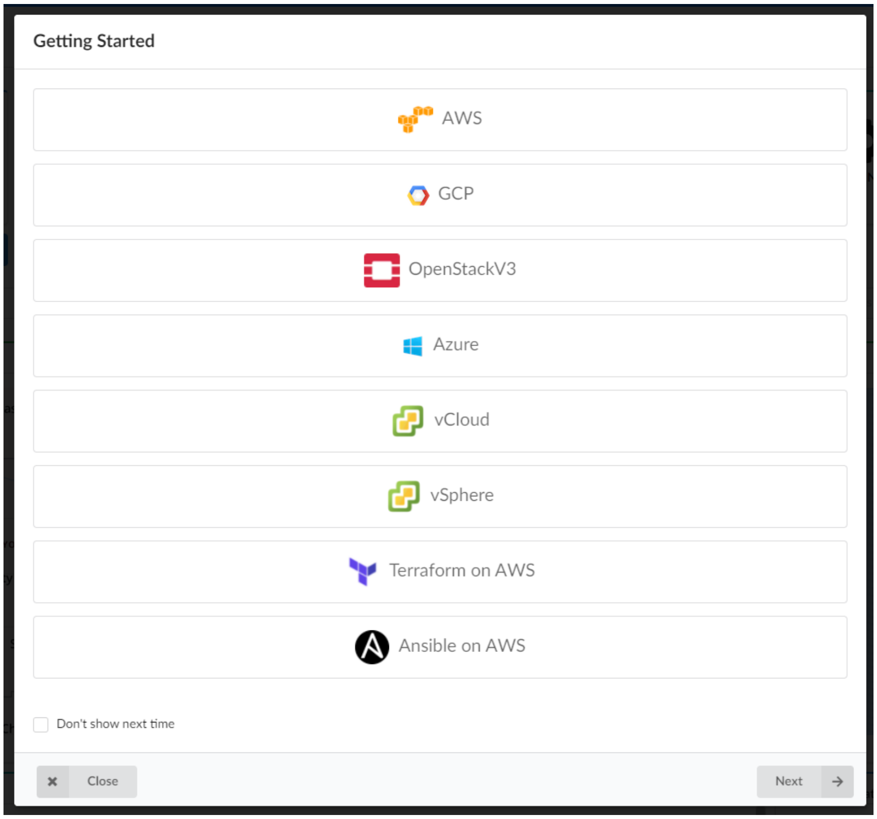
Cluster Status

Service Type	Node Name	Status	Private IP	Public IP / LB IP	Version
Message Broker	ip-172-31-22-87.eu-west-1.compute.internal	✓	172.31.22.87	172.31.22.87	6.1.0
Database	ip-172-31-22-87.eu-west-1.compute.internal	✓	172.31.22.87	172.31.22.87	6.1.0
Manager	ip-172-31-22-87.eu-west-1.compute.internal	✓	172.31.22.87	172.31.22.87	6.1.0

Getting started wizard

For first time users, Cloudify 6.x includes a ***Getting Started*** wizard, designed to simplify the first time experience of setting up the Cloudify manager.
The wizard walks the users through selecting the target infrastructures, uploading the relevant plugins, setting up the proper secrets (for access credentials), and uploading example blueprints.

The wizard appears upon login, unless opted out by the user. The admin may set this option to enabled/disabled.



StarlingX

The StarlingX plugin is introduced as part of Cloudify 6.0. The new plugin provides discovery capabilities over StarlingX based platforms, with the ability to scan central controllers for subcloud objects, retrieve all related subcloud platforms, import their details into Cloudify, and label them based on existing properties. It further allows deployment of Kubernetes/Openstack based services over the relevant platforms.

Learn more about the [StarlingX plugin]

Developer enhancements

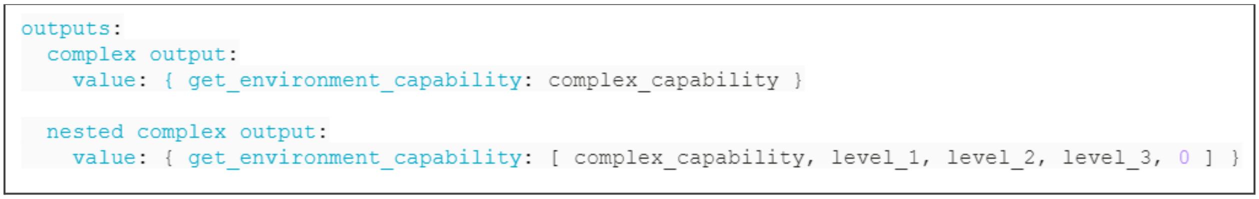
The ‘apply’ command

The ***cfy apply*** command simplifies the development lifecycle by providing an easy update & testing flow in one command. It is used to install/update a deployment using Cloudify manager without having to manually go through the process of uploading a blueprint, creating a deployment, and executing a workflow
Upon introduced changes, the apply command will either deploy & install the new blueprint if such an instance does not exist yet, or will take the existing instance and update it with the new changes.

Learn more about the apply command

Intrinsic functions

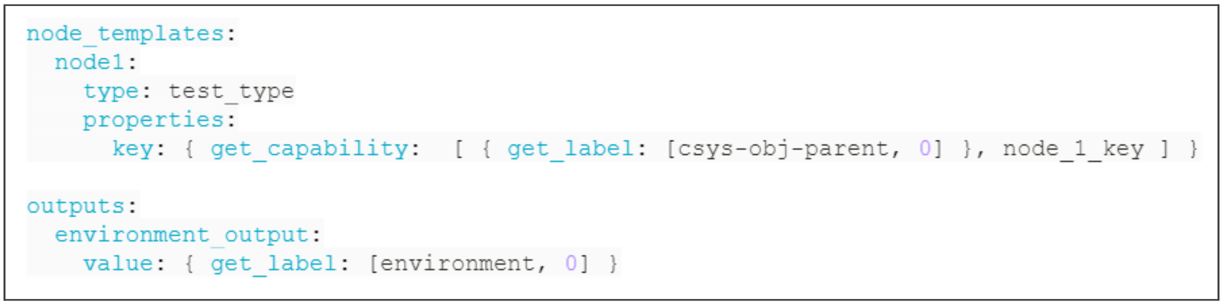
Cloudify v6.0 provides two new intrinsic functions allowing the blueprint developers to extract more information during deployment/runtime. get_label retrieves labels value based on its referred key, and get_environment _capability retrieves a capability from the parent environment onto which that service is being deployed.



```
outputs:
  complex output:
    value: { get_environment_capability: complex_capability }

  nested complex output:
    value: { get_environment_capability: [ complex_capability, level_1, level_2, level_3, 0 ] }
```

Using these intrinsic functions, the developer can set services to auto retrieve environment details and seamlessly deploy itself over that environment with zero user inputs. It allows for easy placement policy over different systems with different credentials or onboarding flows, by letting the service pick up the required info automatically from the environment.



```
node_templates:
  node1:
    type: test_type
    properties:
      key: { get_capability:  [ { get_label: [csys-obj-parent, 0] }, node_1_key ] }

outputs:
  environment_output:
    value: { get_label: [environment, 0] }
```

Learn more about the intrinsic functions

Propagation of workflow parameters to sub-components.

When executing a workflow that is propagated to deployment components, The workflow parameters are propagated as well. This is true for all workflows and all workflow parameters. If such parameters do not exist they are ignored.

Blueprint labels

Similar to the deployment labels, a developer may now assign labels blueprints, which allows for search through the blueprints by category or any criteria.

Learn more about blueprint labels

Security & Compliance

As with every Cloudify release, security improvements are included in v6.0. All used packages have been updated to accommodate all latest security patches, and all known issues handled in all previous Cloudify patches have been resolved in v6.0.

ISO compliance

Cloudify is ISO 27001 (Security) and ISO 27701 (Data Privacy) certified.

Maintenance

Cloudify Cluster Manager

The Cloudify cluster manager is a tool simplifying the installation and upgrade of a Cloudify management cluster. Using a configuration file, the entire installation flow and setup (SSL certificates and more) is applied automatically. Once used for the installation, all following upgrades can be applied using the cluster manager running just a few commands executed from one node to upgrade the entire cluster.
It is capable of installing fully distributed clusters (>9 nodes) and compact clusters (3 nodes) with both embedded and external Database.

Learn more about the cluster manager.

Agents

Cloudify 5.2 extends the agents OS support to include

- RHEL 8.x agent.
- CentOS 8.x agent.

LDAP Authentication improvements

LDAP properties can now be configured allowing support for non-standard user directories through customization of:

- Base DN
- Group DN
- Bind format,
- User filter
- Group member filter
- Attributes used for first name, last name, email, group membership, and uid

Providing admin credentials is no longer required. If credentials are configured they will be used, otherwise, the user login credentials are used to retrieve the group info.
Learn more

Okta IdP SSO support

Cloudify supports several methods of user authentication to the Cloudify Management Console. One of the methods is allowing Single Sign-On (SSO) experience leveraging Identity Providers (IdP).

In v5.1.2 the framework was updated to accommodate the latest changes and Okta’s latest release is now fully supported.

Learn more

Other Improvements

v6.0 includes many medium/minor enhancements and improvements.

- Helm plugin - add support for verify_ssl.
- Sites map widget updated to reflect new states - Good, In progress, Requires attention.
- Asynchronous deployment creation - starting v6.0 nodes and node instances are generated using a workflow in an asynchronous mode.
- Performance enhancements
    - Asynchronous blueprint upload
        - Blueprint upload to the system is now done in an asynchronous manner. The upload state is updated in the UI and can be queried through the CLI. The change is backward compatible.
    - API results caching was introduced for repeatable UI requests.
    - Index optimization was applied
- Deployment groups (API/CLI only) - Deployments may be grouped leveraging a filter or through manual addition, and the groups can be used for bulk actions.
- Jenkins logging - Improved visibility to errors in Jenkins nodes.
- White labeling and customization
    - Composer customization was added
    - String localization support was added

Deploying Cloudify 6.2.0

v6.2.0 can be deployed as a new installation, as an in-place update over a 5.2.x / 6.0.x Cloudify manager, or as an upgrade version for any supported previous Cloudify release.

Upgrading from 5.1.x / 5.2.x / 6.0.x

NOTE!
Upgrading from 5.1.0 is supported only via snapshots, if you want to avoid the upgrade via the snapshot process, please upgrade first to any of the following 5.1.1, 5.1.2, 5.1.3, 5.1.4, and then continue the upgrade to 6.2.0.

Before the Upgrade

As a best practice, we recommend taking a snapshot of the system before the update. Read this page for more details.

NOTE!
If the Cloudify cluster you are upgrading from was deployed using the *Cloudify Cluster Manager* (which is the recommended approach), you can simplify the update process to 6.2.x by running it through the Cluster Manager.
Below, you can find the procedures for either using the Cluster Manager, or a manual flow.

Tip: the upgrade steps require yum installation of the 6.2.0 RPM. This can be done either by downloading the rpm package to the local Cloudify nodes and directing the command to the rpm path or by referencing the URL of the package. The second option requires a live connection to the package path. Here are usage examples for both:

```
# Downloading the RPM to the current directory
curl -O
https://repository.cloudifysource.org/cloudify/6.2.0/ga-release/cloudify-manager-install-6.2.0-ga.el7.x86_64.rpm
# Installing the downloaded RPM
sudo yum install -y cloudify-manager-install-6.2.0-ga.el7.x86_64.rpm
```

```
# Installing the RPM directly
sudo yum install -y
https://repository.cloudifysource.org/cloudify/6.2.0/ga-release/cloudify-manager-ins
```

```
tall-6.2.0-ga.el7.x86_64.rpm
```

Upgrading Cloudify All-In-One

Update steps:

1. Install the new 6.2.0 cloudify-manager-install RPM, by using the command:
   `sudo yum install -y <6.2.0 RPM>`

2. To start the upgrade, run the command
   `cfy_manager upgrade`

3. If Cloudify agents are used in your deployments, run
   `cfy agents install`

4. When opening the Cloudify Management Console after the upgrade, you might see “This page is empty”, this happens because of cached data. To solve this, press CTRL + Shift + R.

Upgrading a Cloudify Compact Cluster (3 nodes)

If the initial cluster installation was done using the Cloudify Cluster Manager, follow this simplified process.

Updating a Cloudify compact cluster leveraging the Cloudify Cluster Manager

You can use the Cloudify Cluster Manager tool to upgrade a compact cluster:

Upgrade your Cloudify Cluster Manager by running

```
sudo yum install -y
https://repository.cloudifysource.org/cloudify/cloudify-cluster-manager/1
.0.14/ga-release/cloudify-cluster-manager-1.0.14-ga.el7.x86_64.rpm
```

On the host that has Cloudify Cluster Manager installed, run `cfy_cluster_manager upgrade`.

Optional Arguments:

--config-path The completed cluster configuration file path. Default: `./cfy_cluster_config.yaml`

--upgrade-rpm Path to a v6.2.0 cloudify-manager-install RPM. This can be either a local or remote path.
Default:
`http://repository.cloudifysource.org/cloudify/6.2.0/ga-release/cloudify-manager-install-6.2.0-ga.el7.x86_64.rpm`

-v, --verbose Show verbose output

Running this command will automatically run the upgrade procedure on the cluster.

If the Cluster was manually deployed, please follow this procedure instead:

Manually updating a Cloudify compact cluster

1. Install the new 6.1.0 cloudify-manager-install RPM on all 3 nodes of the cluster, by using the command:
   `sudo yum install -y <6.1.0 RPM>`
   Repeat this step on all 3 nodes.

2. On each of the cluster nodes, run `cfy_manager upgrade -c <path to DB config>`.
   Do it one after the other, not in parallel.

   Tip: If you used the cloudify-cluster-manager tool to generate the Cloudify cluster, the path to the DB config file is `/etc/cloudify/postgresql-<node number>_config.yaml`*. If the cluster was manually installed, please direct the command to the path of the file you generated.*

3. On each of the cluster nodes, run `cfy_manager upgrade -c <path to rabbitmq config>`.
   Do it one after the other, not in parallel.

   Tip: If you used the cloudify-cluster-manager tool to generate the Cloudify cluster, the path to the RabbitMQ config file is `/etc/cloudify/rabbitmq-<node number>_config.yaml`*. If the cluster was manually installed, please direct the command to the path of the file you generated.*

4. On each one of the cluster nodes, run `cfy_manager upgrade -c <path to manager config>`
   Do it one after the other, not in parallel.

   Tip: If you used the cloudify-cluster-manager tool to generate the Cloudify cluster, the path to the manager config file is `/etc/cloudify/manager-<node number>_config.yaml`*. If the cluster was manually installed, please direct the command to the path of the file you generated.*

5. If Cloudify agents are used in your deployments, run the following command from **just one** of the cluster nodes:
   `cfy agents install`

6. When opening the Cloudify Management Console after the upgrade, you might see “This page is empty”, this happens because of cached data. To solve this, press CTRL + Shift + R.

Upgrading a Cloudify Fully Distributed Cluster (9+ nodes)

If the initial cluster installation was done using the Cloudify Cluster Manager, follow this simplified process.

Updating a Cloudify Fully Distributed Cluster leveraging the Cloudify Cluster Manager

You can use the Cloudify Cluster Manager tool to upgrade a fully distributed cluster:

Upgrade your Cloudify Cluster Manager by running

```
sudo yum install -y https://repository.cloudifysource.org/cloudify/cloudify-cluster-manager/1.0.14/ga-release/cloudify-cluster-manager-1.0.14-ga.el7.x86_64.rpm
```

On the host that has Cloudify Cluster Manager installed, run `cfy_cluster_manager upgrade`.

Optional Arguments:

--config-path The completed cluster configuration file path. Default: `./cfy_cluster_config.yaml`

--upgrade-rpm Path to a v6.20 cloudify-manager-install RPM. This can be either a local or remote path.
Default: `http://repository.cloudifysource.org/cloudify/6.2.0/ga-release/cloudify-manager-install-6.2.0-ga.el7.x86_64.rpm`

-v, --verbose Show verbose output

Running this command will automatically run the upgrade procedure on the cluster.

If the cluster was manually deployed, please follow this procedure instead:

Manually updating a Fully Distributed Cluster

Update steps:

1. Install the new 6.2.0 cloudify-manager-install RPM on all the cluster nodes, by using the command:
   `sudo yum install -y <6.2.0 RPM>`
   Repeat this step on all 9 nodes.

2. On all three database nodes run `cfy_manager upgrade`
   Do it one after the other, not in parallel.

3. On all three RabbitMQ nodes run `cfy_manager upgrade`
   Do it one after the other, not in parallel.

4. On all manager nodes, run `cfy_manager upgrade`
   Do it one after the other, not in parallel.

5. If Cloudify agents are used in your deployments, run the following command from **just one** of the manager nodes:

```
cfy agents install
```

7. When opening the Cloudify Management Console after the upgrade, you might see “This page is empty”, this happens because of cached data. To solve this, press CTRL + Shift + R.

Upgrading from Previous Versions (4.x - 5.0.5) to 6.x.x

The upgrade flow from versions 4x - 5.0.5 to versions 5.1 and above requires additional steps. This is due to the Python 3 migration introduced in 5.1. This migration requires updating plugin code.
Please review the 5.1 upgrade procedure carefully and consult with the Cloudify support team to assure a smooth and successful upgrade.

New installation

- To deploy a single - All-In-One manager, please follow the AIO manager installation guide.
- To deploy a highly available Compact Cluster - a distributed cluster of 3 nodes - please refer to the 3 nodes cluster installation guide.
- To deploy a highly available Fully Distributed Cluster - a distributed cluster of 9 nodes - please refer to the 9 nodes cluster installation guide.
  NOTE! You can simplify the cluster deployment and automate the provisioning by leveraging the Cloudify Cluster Manager Package.
- **NOTE! When using your own signed certificates you must include the *external_ca_cert_path* as it will be used for all other certificates.**

Download Cloudify 6.2.0

Manager install:

- RPM - cloudify-manager-install-6.2.0ga.rpm
- OpenStack image - cloudify-manager-6.2.0.qcow2
- Docker containers
    - All-in-one manager - Cloudify manager aio
    - Manager - Cloudify manager worker
    - Database - Cloudify postgresql
    - Messaging queue - Cloudify rabbitmq

Premium CLI packages:

- RPM (CentOS/RHEL) - Cloudify cli centos (.rpm)
- Debian - Cloudify cli debian (.dep)
- Windows - Cloudify cli Windows (.exe)

Support

Support Discontinuance

With the official end of life for CentOS 6.x on November 30th, 2020, starting Cloudify 5.1.1 CentOS 6 agent will no longer be supported (nor provided in the package).

Supported Versions

Listed below are the support discontinuance dates for the recent Cloudify versions. As of these dates, the respective versions will no longer be supported under the standard Cloudify support agreement.

Version	**Support Discontinuance Date**
Cloudify Premium & Community Editions v4.6.x	Apr 17th, 2021
Cloudify Premium & Community Editions v5.0.5	Feb 3rd, 2022
Cloudify Premium & Community Editions v5.1.x	Oct 19, 2022
Cloudify Premium & Community Editions v5.2.x	Apr 5, 2023
Cloudify Premium & Community Editions v6.0.x	May 26, 2023
Cloudify Premium & Community Editions v6.1.x	Aug 10 2023