



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Setting up Genesys Multicloud CX Private Edition

Upgrade strategies

---

## Contents

- 1 Blue/Green strategy
- 2 Canary strategy
- 3 Rolling Update strategy
- 4 Next steps

---

Talks about different Upgrade strategies supported by Genesys Multicloud CX services.

### Related documentation:

- 
- 

### RSS:

- [For private edition](#)

Genesys supports industry standard upgrade strategies to upgrade Genesys Multicloud CX services. Our services are designed to support specific strategies based on the business function it fulfills.

Currently supported Upgrade strategies are:

- Blue/Green
- Canary
- Rolling Update

The following sections describe the fundamentals of each upgrade strategy.

## Blue/Green strategy

This is a release management technique that is employed to reduce the risk and provide zero-downtime for your services during upgrades. This method involves the following high level steps:

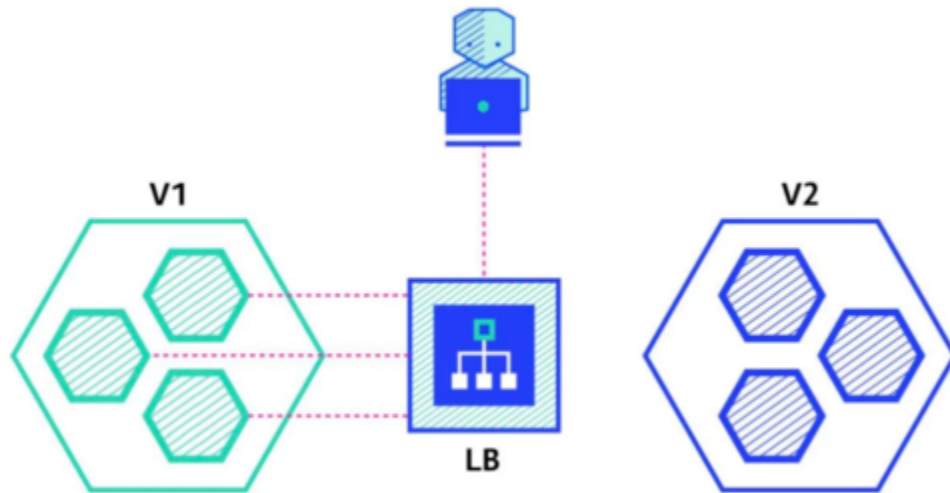
- You will create two identical production environments called '**blue**' and '**green**' and identify any one of them as '**active**' and another one as '**idle**'. You can also imagine the active and idle environments as production and pre-production respectively.
- You deploy and validate the new release in the blue (idle) environment by running quality assurance and user acceptance tests.
- Once you are satisfied and there are no critical issues found, switch all the user traffic from green (active) environment to blue (idle) environment with the help of a router.
- Your blue environment becomes active now and the green environment becomes idle.

### Important

You can keep the green (idle) environment as a fallback option for some time until you gain confidence with the new release running on the blue (active) environment. In case of unexpected issues arise with the newly deployed blue environment, you can always rollback to the last version by switching back to green.

---

In Kubernetes environments, this can be easily achieved by orchestrating the new resources like pods, containers, etc. and killing them when they are not needed. Services like GWS, and WebRTC support Blue/Green upgrade method.

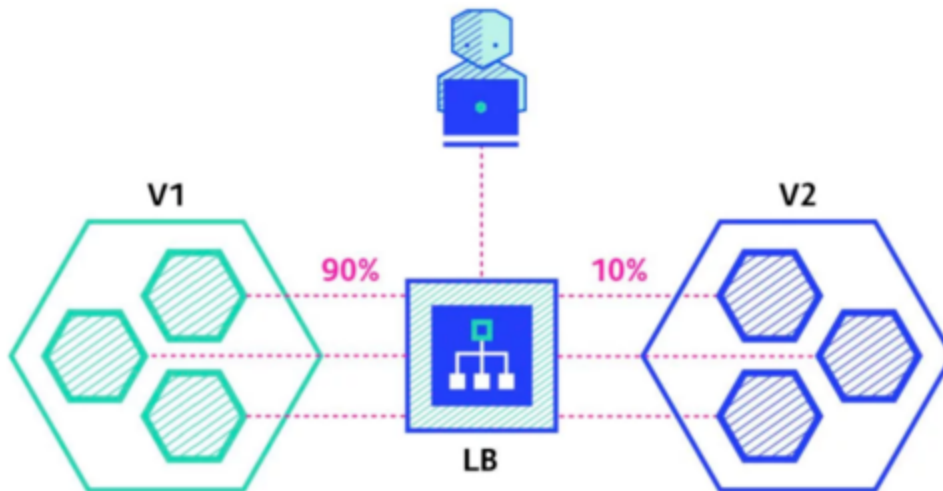


## Canary strategy

In Canary deployments, you will upgrade only a subset of pod instances with the new release and make it available for limited number of users. In this upgrade strategy, both the subset of pod instances (with the new release) and the production pod instances (with the previous release) receives the live production user traffic. You can monitor the user behavior for bugs or performance issues from the upgraded pod instances. When the results are satisfying, you can incrementally roll out the new release to the wider group of pod instances in batches.

Canary deployment also offers easy rollback options to a previous version of the service.

Services like Voice supports Canary upgrade method.



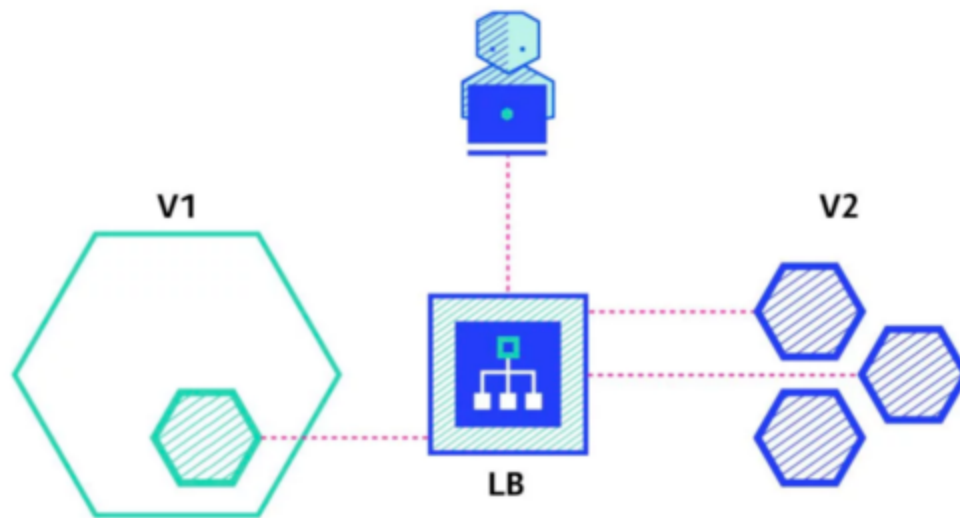
## Rolling Update strategy

This upgrade strategy is similar to Canary. In Rolling Update strategy, you will replace the old pod instances running in production one-by-one with the new ones gradually. You will select a pod instance, deactivate it from the node, update with the new software, and then connect to the node.

During upgrade, the load shared by the pod instance being updated will be shared by other pod instances actively running in the ecosystem. Observe the behavior of the new pod instance, if it is satisfactory, rollout the update to all other pod instances in a similar fashion. This method ensures that at any point of time, the users are served with maximum number of pod instances.

Rolling Update strategy is suitable for upgrading a complex application that runs on multiple Kubernetes nodes (server cluster). It is also suitable for applications that directly interface with a load balancer so that the traffic in the absence of a pod instance (undergoing upgrade) is shared by the remaining active pod instances.

Services like GAuth, and Designer support Rolling Update method.



## Next steps

Learn how to upgrade a Genesys Multicloud CX service from the Upgrade process page.