Types of upgrades

Major and minor versions

Major and minor changes can be consensus breaking. These upgrades usually go through a governance proposal and happen at specific heights.

Patch versions

Patch versions are backwards compatible changes. These typically can be applied in a rolling fashion and don't need to go through a governance proposal.

Hard-forks

One of the limitations of the normal upgrade procedure via governance is that it requires waiting for the entire voting period, which makes them unsuitable for emergency situations. For such cases, hard forks are usually required.

The high-level strategy for coordinating an upgrade is as follows:

1. The vulnerability is fixed on a private branch that contains breaking changes.
2. A new patch release (e.g.v8.0.0
3. ->v8.0.1
4.) needs to be created that contains a hard fork logic and performs an upgrade to the next breaking version (e.g.v9.0.0
5.) at a predefined block height.
6. Validators upgrade their nodes to the patch release (e.g.v8.0.1
7.). In order to perform the hard fork successfully, it's important that enough validators upgrade to the patch release so that they make up at least 2/3 of the total validator voting power.
8. Before the upgrade time (corresponding to the upgrade block height), the new major release (e.g.v9.0.0
9.) including the vulnerability fix is published.
10. Upgrades happen in a similar fashion asMsgSoftwareUpgrade
11. .

From a node operator's perspective, hard forks are essentially a combination of a patch version (v8.0.1) followed by a major version (v9.0.0). Please use the instructions fromPerforming Upgrades to perform the corresponding upgrades.

Snapshots Performing Upgrades