Table of Contents

* [What is yum history command? Do I need this?](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [Why should I use yum history if I can use yum downgrade](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [Any risks of using yum history, when should I use yum history rollback](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [1. List all the yum transactions](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [2. Get list of packages installed or removed in individual transactions](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [3. Get yum transaction history for individual rpm](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [4. Undo yum transaction](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [5. YUM redo transaction](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [6. YUM rollback to a certain transaction in history](https://www.golinuxcloud.com/yum-history-rollback-updates/)
* [Conclusion](https://www.golinuxcloud.com/yum-history-rollback-updates/)

**What is yum history command? Do I need this?**

* In an enterprise level we perform time to time security updates and apply patches to make sure all the security vulnerabilities are addressed
* Normally this is performed using online or offline repositories using [YUM or DNF](https://www.golinuxcloud.com/another-app-is-currently-holding-the-yum-lock/)
* Now imagine as part of some patching your server broke or it is possible that you wish to revert to the last installed environment. In production it is very important that [you always have a fallback or backup option](https://www.golinuxcloud.com/backup-linux-partition/) assuming if something fails.
* In such case if you have a backup then yes you can restore from backup but then we are "not" talking about files, we are talking about rpms so a backup and restore won't be helpful here.
* This is where yum history comes for the rescue.
* YUM or DNF maintains and stores a sqlite database of information about each transaction.
* The history is organized terms of transaction ids and is updated every time a yum transaction affects the package configuration of the system.
* Mostly this database can be found in the /var/lib/yum/history/ directory.
* So yum history provides an option to rollback to previous successful transactions without you having to worry about downgrading individual packages.

**Why should I use yum history if I can use yum downgrade**

* Yes this is a valid question.
* With recent Linux distribution yum is mapped to dnf which is far more stable and robust compared to earlier versions to handle downgrades but in some cases you would still face problems
* For example while updating X rpm the yum will install certain rpms as part of dependencies but while downgrading this X rpm, it will fail to remove those additionally installed dependent packages.
* You can refer the [official yum wiki](http://yum.baseurl.org/wiki/DownGradeProblems.html) to learn more about the problems with downgrades
* With yum history you can leave this handling to yum to make sure it will revert all the transactions.

**Any risks of using yum history, when should I use yum history rollback**

* Now here comes the bad news, we do have some risks involved with yum history rollback and history related arguments.
* Rollback of selinux, selinux-policy-\*, kernel, glibc (dependencies of glibc such as gcc) packages to older version is not supported.
* Thus, downgrading a system to minor version is not recommended as this might leave the system in undesired state.
* You should use the yum history option for **small update rollbacks only**.
* I have personally faced problems with selinux when yum rollback failed to revert the transaction. Although your system would be still in usable state as yum will not proceed unless all the dependency criteria are fulfilled.

**WARNING:**

You have to be careful when you use history options such as undo and rollback. Yum does its best to comply, but it cannot restore configurations, and it will not restore previous versions of your configuration files if you have edited them. This is not a fail-safe option if you don't have any backups. Secondly if you have used rpm command to add/remove/modify any rpm then those transactions will not be visible with yum history and can also cause conflict while trying to perform undo, redo or rollback action with yum.

**1. List all the yum transactions**

Use yum history command without any additional arguments to list all the transactions

bash

# yum history

yum history

* The **first column** contains the ID of each transaction. You will use these ID to perform any operations with yum history
* The **second column** shows the command which was executed with yum or dnf on the terminal
* The **third column** provides the date and time information when the respective command was executed
* The **fourth column** provides the action used with yum or dnf. The "I,U" means both install and update was done. In that line I had performed yum install kernel but yum internally decided to also upgrade few packages dependent on kernel
* The **fifth column** shows the number of rpms which were altered i.e. installed/removed/modified

**2. Get list of packages installed or removed in individual transactions**

Now the above output is brief as we know that with transaction id 38, 3 rpms were altered but to get more information on these rpms and the action performed we will use:

bash

# yum history info <transaction id>



**3. Get yum transaction history for individual rpm**

If you wish to get the list of transaction ids when a particular package was altered then you can use

bash

# yum history info <pkg\_name>

This will give you a list of all the transaction history along with other relevant details from the respective transaction. For example here I want to get the list of transaction IDs when kernel package was altered.

yum history info kernel

To only get the transaction ID we can add grep to this command, as shown below:

Get transaction ID from yum history

**4. Undo yum transaction**

*Now to go back in time is not possible in real life but is possible with yum and dnf*🙂

yum history undo will require access to all the previous RPM version; thus, need to ensure the older RPM versions are available to the system. It is recommended that prior to doing updates, you closely inspect the output of package-cleanup --orphans to know what currently installed RPMs are no longer available in the enabled repositories.

I had installed abrt package with ID 42

bash

42 | install abrt | 2020-08-22 14:04 | Install | 18

which now I don't need any more so let me undo this transaction:

bash

# yum history undo 42

yum history undo

**5. YUM redo transaction**

This command will re-perform a specific transaction (as defined by the transaction ID), as follows:

bash

# yum history redo 42

So in this command we will re-perform the action as defined with transaction ID 42 i.e. to install abrt and all it's dependency packages

yum history redo

**6. YUM rollback to a certain transaction in history**

Now this command is quite useful and equally dangerous as it will revert back all the yum related transactions upto the ID which you provide with YUM. In this example we will revert back our Linux server to the state of transaction ID 45 where in I had downgraded python3-bind. If you try to rollback to a very old transaction then it is also possible that the rollback would never go through due to one or the other dependencies.

So I will try to do rollback to a recent yum transaction:

bash

45 | downgrade python3-bind-9 | 2020-08-22 14:13 | Downgrade | 5

In this example I will rollback to transaction ID 45,

bash

# yum history rollback 45

as you see yum is about to do 5 upgrades, 21 downgrades and will remove 31 additional packages with this rollback.

yum history rollback

When I tried to do a yum rollback to much more previous transaction, I was getting below error:

