

Versioning your JIRA and Confluence app directories

I recently published https://hg.sr.ht/~redradishtech/versioned_directories, a script to manage the seemingly trivial task of keeping a `/opt/atlassian/jira/current` symlink pointing to the correct `/opt/atlassian/jira/$version`; or more generally, managing a directory containing N versions of something.

Background

We know JIRA's application files usually go in:

```
/opt/atlassian/jira
```

and the home directory goes in:

```
/var/atlassian/application-data/jira
```

but what happens below those directories?

We could just dump JIRA in `/opt/atlassian/jira`, like the Atlassian installer does, but then on upgrade there is no safe rollback path. Likewise with the home directory: if we only have one copy in `/var/atlassian/application-data/jira`, and an upgrade goes wrong, we're left having to restore from backup.

Versioned directories

The standard I follow is to keep the current and previous releases in versioned directories, with `current/` and `previous/` symlinks:

```
root@jturner-desktop /opt/atlassian/jira # ls -l
drwxr-sr-x+ 27 root          redradish_jira 36 Jul  8 22:28 8.10.0
drwxr-sr-x+ 27 root          redradish_jira 37 Jul  8 22:28 8.5.1
drwxr-xr-x   3 root          root             6 Jul  8 22:28 old
lrwxrwxrwx   1 root          root             6 Jul  8 22:28 current -> 8.10.0
lrwxrwxrwx   1 root          root             5 Jul  8 22:27 previous -> 8.5.1
```

All versions older than `previous` get moved to `old/`, and eventually deleted.

Each old version has an `UPGRADED_TO_$newver.txt` marker file, to clearly indicate that it is no longer current:

```
root@jturner-desktop /opt/atlassian/jira # cat previous/UPGRADED_TO_8.10.0.txt
This is the old directory for 8.5.1, prior to the 8.10.0 upgrade on Wed 08 Jul 2020 22:28:45 AEST
```

Upgrading involves deploying a new release to `/opt/atlassian/jira/$newver`, adjusting `current` and `previous` symlinks, and moving the old `previous` directory into `old/`.

The JIRA home directory is just the same:

```
root@jturner-desktop /var/atlassian/application-data/redradish_jira # ls -l
drwxr-xr-x+ 26 redradish_jira redradish_jira 33 Jul 11 12:28 8.10.0
drwxr-xr-x+ 26 redradish_jira redradish_jira 33 Jul  8 22:28 8.5.1
drwxrwx--- 35 root            root             36 Jul  7 07:01 backups
drwxr-xr-x   3 redradish_jira redradish_jira   6 Jul  8 22:28 old
lrwxrwxrwx   1 root            root             6 Jul  8 22:28 current -> 8.10.0
lrwxrwxrwx   1 root            root             5 Jul  8 22:27 previous -> 8.5.1
```

If I have to roll back, then the abandoned new version directory is pointed to with a `next/` symlink, and get a `DOWNGRADED_TO_$oldver.txt` marker.

What do the scripts do?

Say JIRA 9.0 arrives. I would unpack JIRA 9.0 into `/opt/atlassian/jira/9.0`, and then:

```

root@jturner-desktop /opt/atlassian/jira # switchver . upgrade 9.0
root@jturner-desktop /opt/atlassian/jira # ls -l
drwxr-sr-x+ 27 root redradish_jira 36 Jul 11 13:12 8.10.0
drwxr-xr-x   2 root root              3 Jul 11 13:12 9.0
drwxr-xr-x   4 root root              7 Jul 11 13:11 old
lrwxrwxrwx   1 root root              3 Jul 11 13:11 current -> 9.0
lrwxrwxrwx   1 root root              6 Jul  8 22:28 previous -> 8.10.0

```

8.5.1 is stashed away in old/

Say 9.0 is a lemon. To downgrade:

```

root@jturner-desktop /opt/atlassian/jira # $ATL_MANAGE/lib/versioned_directories/switchver . downgrade
root@jturner-desktop /opt/atlassian/jira # ls -l
drwxr-sr-x+ 27 root redradish_jira 36 Jul 11 13:12 8.10.0
drwxr-sr-x+ 27 root redradish_jira 37 Jul  8 22:28 8.5.1
drwxr-xr-x   2 root root              3 Jul 11 13:12 9.0
drwxr-xr-x   2 root root              4 Jul 11 12:31 database_views
drwxr-xr-x   3 root root              6 Jul 11 13:12 old
lrwxrwxrwx   1 root root              6 Jul  8 22:28 current -> 8.10.0
lrwxrwxrwx   1 root root              3 Jul 11 13:11 next -> 9.0
lrwxrwxrwx   1 root root              5 Jul 11 13:12 previous -> 8.5.1

```

Notice how the script restored the `previous/` symlink, figuring out what it should be by looking at the marker files. In fact, I can `upgrade` and `downgrade` through the entire sequence of versions available.

Is this rocket science?

No, but having marker files, an always-consistent structure, and a command to rollback or rollforward quickly is nice. The structure works well in non-obvious ways:

- You can make `old/` a separate partition, and rollback will still be fast, because the `previous/` version isn't yet moved to `old/`
- You can have replication, and upgrade your replication standby safely. `prod:/opt/atlassian/jira/8.10.0/` can keep replicating to `sandbox:/opt/atlassian/jira/9.0/` even if you deployed `sandbox:/opt/atlassian/jira/9.0/`
- Your backups will be consistent even if they occur halfway through an upgrade, at least if you hardcode versions. E.g. if `rsnapshot` is backing up `/var/atlassian/application-data/jira/8.10.0/`, it doesn't matter if `/var/atlassian/application-data/jira/9.0` goes live.

It's all wrapped up in a nice script. Give it a try!