

Workflow

Leveraging environments and version control

Using Git/SVN

- Puppet manifests are code
- Use revision control to make it versionable
- Visibility/control over changes

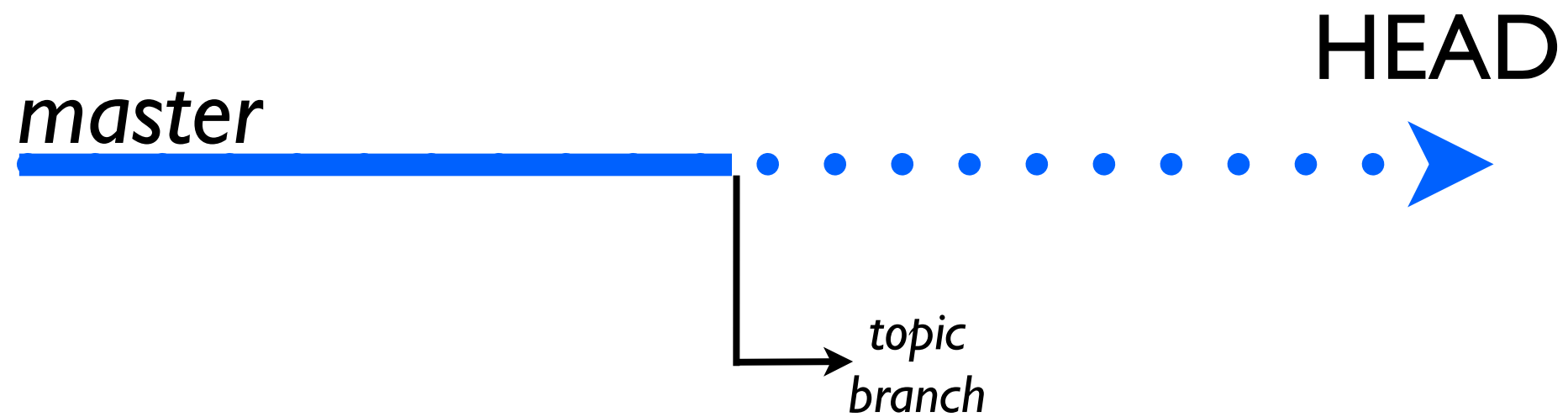
Working with branches

- Raise a ticket to describe the work
- Work in a topic branch
- Merge changes back to the master branch

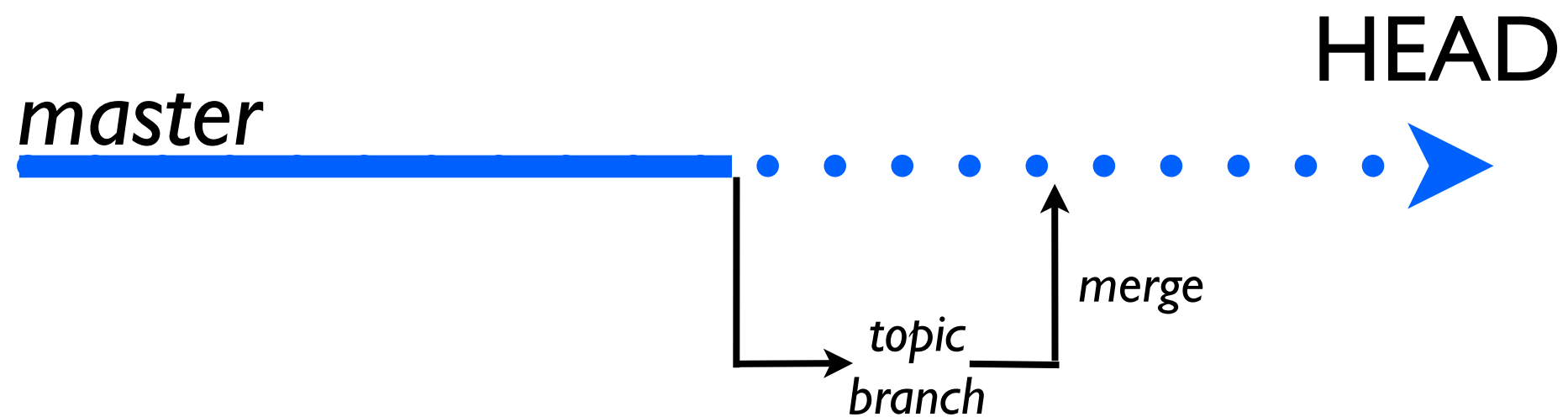
Keep all your code in master



Keep all your code in master



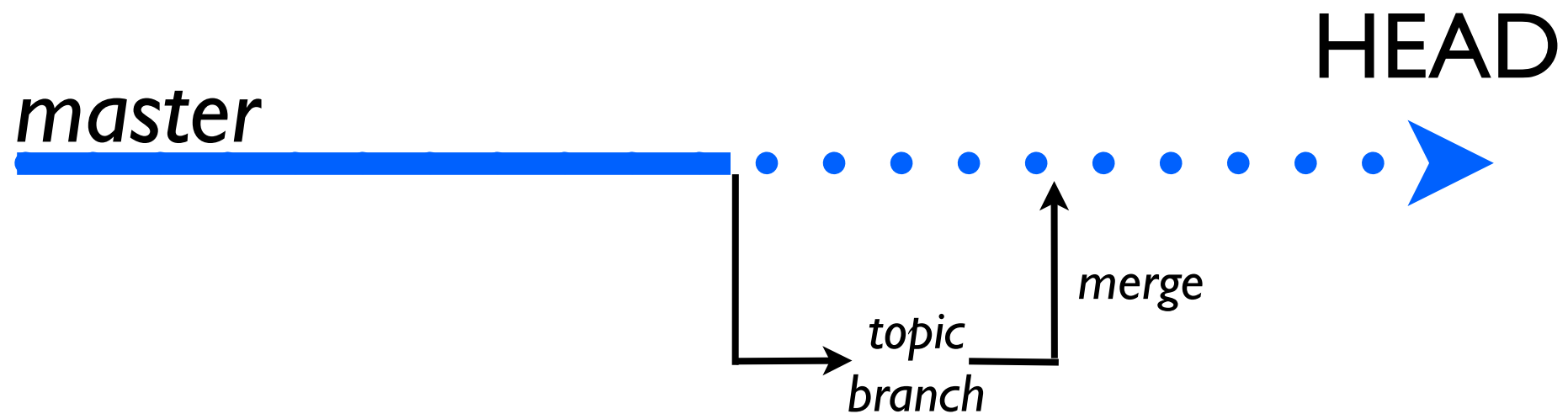
Keep all your code in master



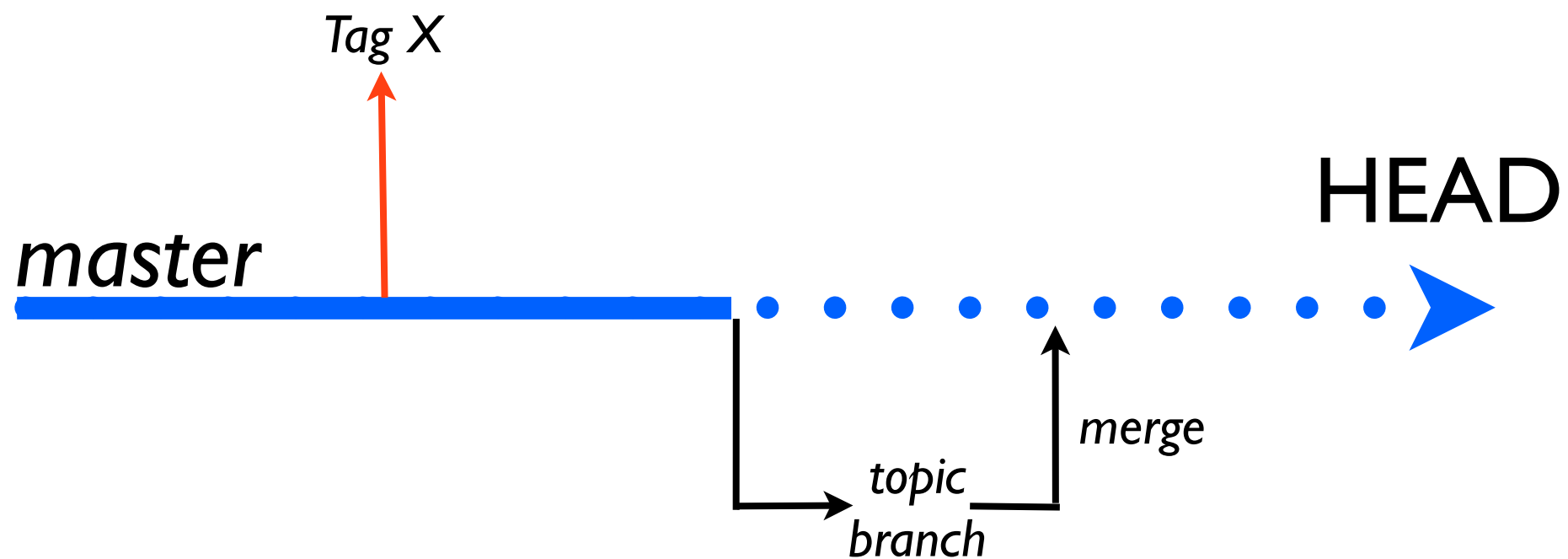
Working with tags

- Use tags as Puppet code “releases”

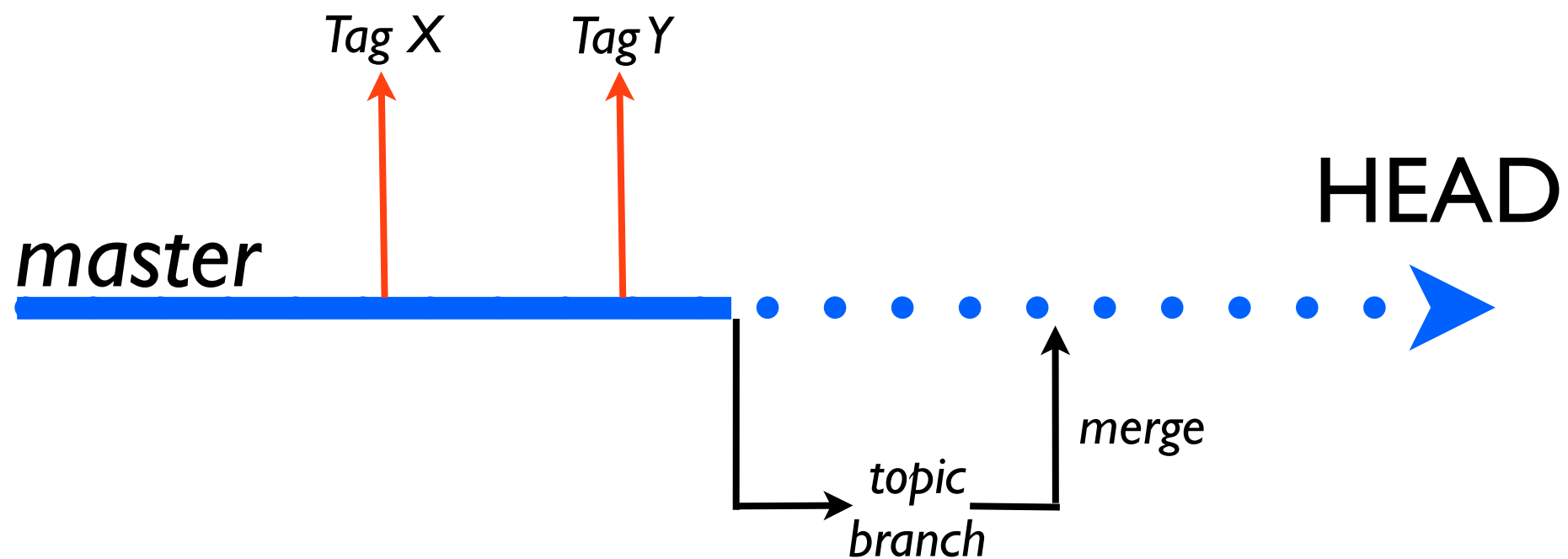
Keep all your code in master



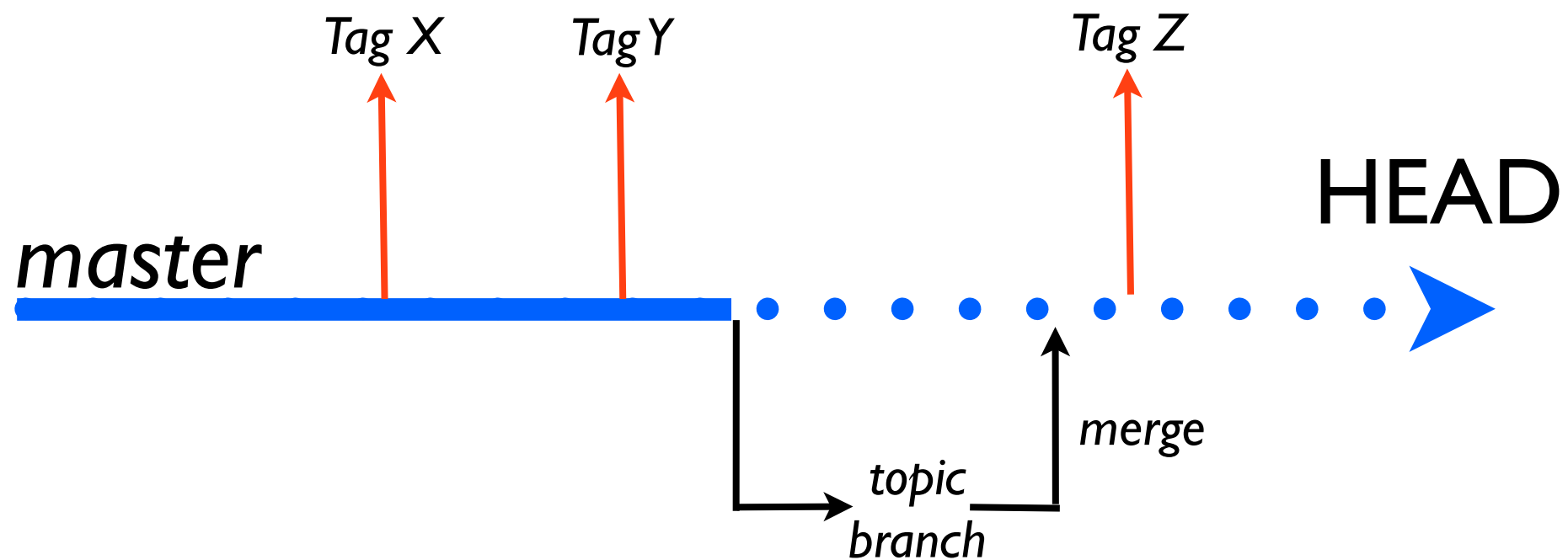
Keep all your code in master



Keep all your code in master



Keep all your code in master



Tags

- Tags are immutable
- If changes are required, a new tag must be created

Releasing to Puppet Master

```
# cd /etc/puppet/modules  
# git checkout releaseY
```

Releasing to Puppet Master

```
# ls -las /etc/puppet/  
total 16  
4 drwxr-xr-x    4 root root 4096 Mar 13 19:48 .  
4 drwxr-xr-x. 84 root root 4096 Mar 13 19:47 ..  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseX  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseY
```

Releasing to Puppet Master

```
# ls -las /etc/puppet/  
total 16  
4 drwxr-xr-x    4 root root 4096 Mar 13 19:48 .  
4 drwxr-xr-x.  84 root root 4096 Mar 13 19:47 ..  
0 lrwxrwxrwx    1 root root    8 Mar 13 19:48 production -> releaseX  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseX  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseY
```

Leveraging environments

```
# cat /etc/puppet.conf  
[...]  
modulepath=/etc/puppet/$environment/modules
```


Leveraging environments

```
# cat /etc/puppet.conf  
[...]  
modulepath=/etc/puppet/$environment/modules  
  
# puppet agent -t
```

Leveraging environments

```
# cat /etc/puppet.conf  
[...]  
modulepath=/etc/puppet/$environment/modules  
  
# puppet agent -t  
# puppet agent -t --environment releaseX --noop
```

The deployment

```
# ls -las /etc/puppet/  
total 16  
4 drwxr-xr-x    4 root root 4096 Mar 13 19:48 .  
4 drwxr-xr-x. 84 root root 4096 Mar 13 19:47 ..  
0 lrwxrwxrwx    1 root root    8 Mar 13 19:48 production -> releaseY  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseX  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseY
```

The rollback

```
# ls -las /etc/puppet/  
total 16  
4 drwxr-xr-x    4 root root 4096 Mar 13 19:48 .  
4 drwxr-xr-x.  84 root root 4096 Mar 13 19:47 ..  
0 lrwxrwxrwx    1 root root    8 Mar 13 19:48 production -> releaseX  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseX  
4 drwxr-xr-x    3 root root 4096 Mar 13 19:47 releaseY
```

Other benefits

- Change control over what is merged into master
- Commits in between tags show exactly what has been released.

Fear driven release management myths

- Making changes is dangerous
- Don't update unless it's really important
- Make changes as infrequently as possible

Fear driven release management facts

- Making many changes at once is dangerous
- Change is good
- If you fear making changes the problem is likely elsewhere (eg: testing!)

Fear driven release
management
increases risk

Release little and often

- Minor incremental updates are easier to manage
- Finding bugs easier
- Rollback less impacting
- Greater visibility
- Good testing practices make this easy