**Proposed solution for init/config patches**

**8 cases for all init/config patches**

1. Add configuration files to system folder
2. Add scripts to system folder
3. Add service files to /etc/systemd/system/
4. Change configuration files with “noreplace” attribute in system folder
5. Change files including configuration file, script and service without “noreplace” attribute in system folder
6. Change service files in usr/lib/systemd/system/ that will be started by SM after power on
7. Change service files in usr/lib/systemd/system/ that will be started by systemd after power on
8. Use puppet instead of existed patch

**4 proposed solutions for all these cases except for case 5/6**

1. Add configuration package
2. Add a configuration package folder under the folder where the original package sit.

 Name package like nfs-utils-config.

1. Use this configuration package to package configuration files to system folder.
2. Add “requires: original package” in spec file like “Requires: nfs-utils” to make sure original package is installed before configuration package.
3. If we need to add files, we can package customized files to system folder
4. If we need to change files with “noreplace”, we can add customized files to /usr/share/starlingx/stx.\*\*\*.conf,

Then in %post section of the spec, copy it to destination folder as \*\*\*.conf during initiation installation.

1. For case 7, services are enabled by systemctl command in spec file. We can package customized service file to /etc/systemd/system/, like we did for memcached.

Refer to https://review.openstack.org/#/c/610459/

1. Use Puppet
2. We can do some modifications on existed pp files in

stx/stx-config/puppet-manifests/src/modules/platform/manifests/

1. For some basic configuration file change with “noreplace” attribute, we can also add a pp file in above folder
2. Rework current patches

Some META patches includes not only configuration related patch, but also src patch or spec change like build flag.

After handle remove configuration part out, we need rework META patch.

Propose to use [spec-include-TiS-changes.patch](https://review.openstack.org/#/c/610459/10/filesystem/nfs-utils/centos/meta_patches/spec-include-TiS-changes.patch) to including all spec change or src related patch adding if there is no corresponding META patch

**Solution for case 5 and 6**

1. Change files including configuration file, script and service without “noreplace” attribute in system folder

If we need to change this kind of file to customized one, after in-service patching on this package, the file will be overwritten to original one. How to handle this patching scenario? Any proposal?

From my point, we can utilize existed in-service patching-script mechanism to call in-service script to copy customized file to destination folder after patching. Any comment?

1. Change service files in usr/lib/systemd/system/ that will be started by SM after power on

We cannot do it like we do for case 7. Can we use patching-script as well for this in-service patching?

If we use puppet, we can only resolve boot-required patching.

**Analysis on all init/configuration patches.**

Please see my comment in 2 columns of the attachment

“Comments from zhipeng” and “proposed solution”

