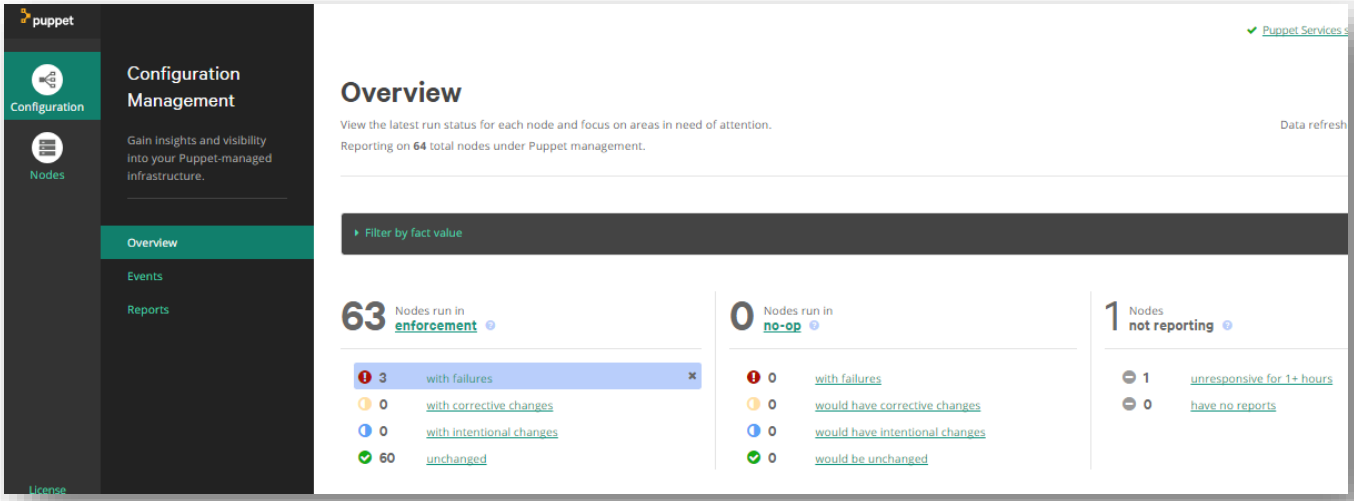
National MI Puppet Documentation

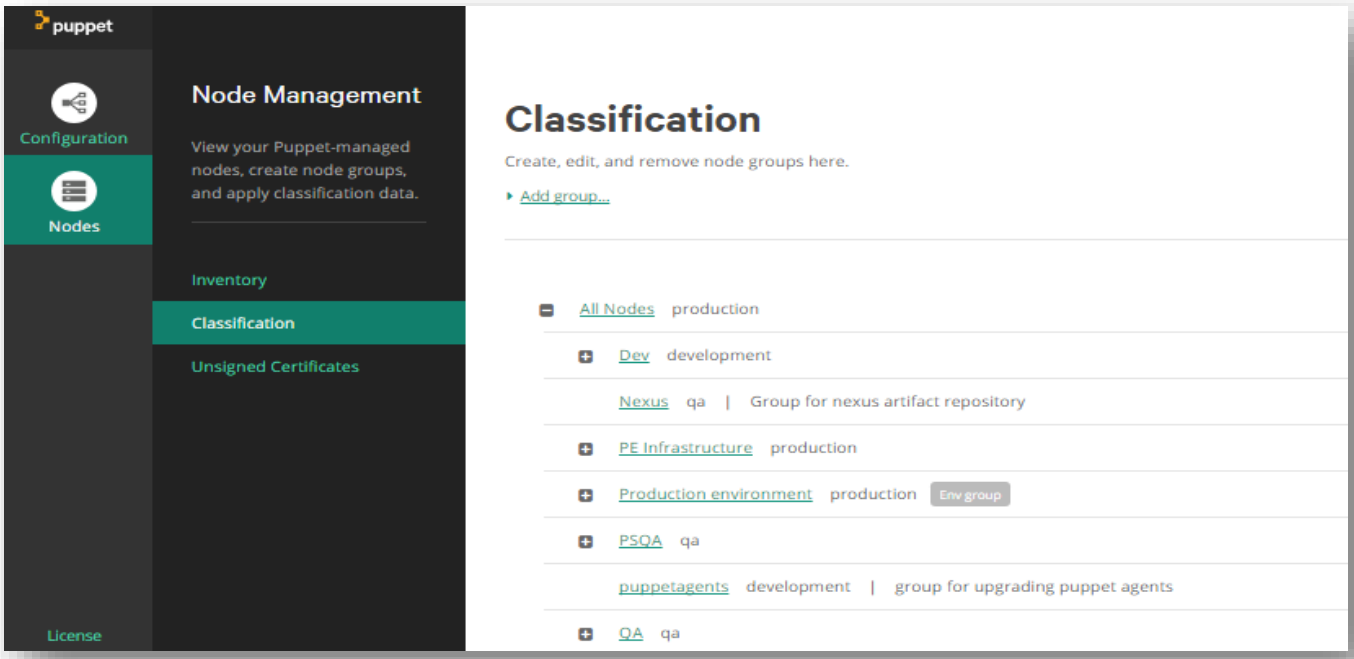
AXIS Application Deployment Procedure:

1. Login to Puppet Enterprise Console for a specific environment.
   https://10.1.160.53/
2. Login using your active directory logins to access the console.

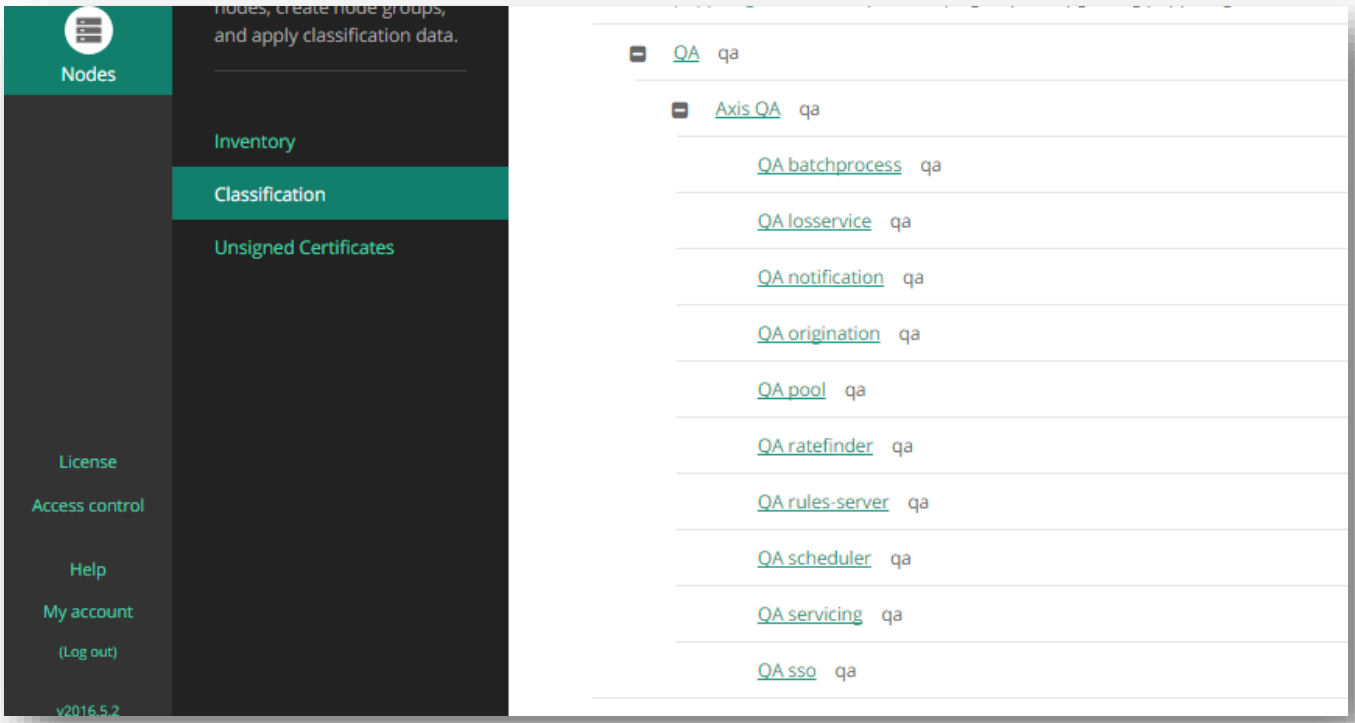
You will the see the following screen after login:



3. Navigate to the **Nodes → Classification** section and expand the environment you will be deploying to.
   https://10.1.160.53/#/node_groups/groups

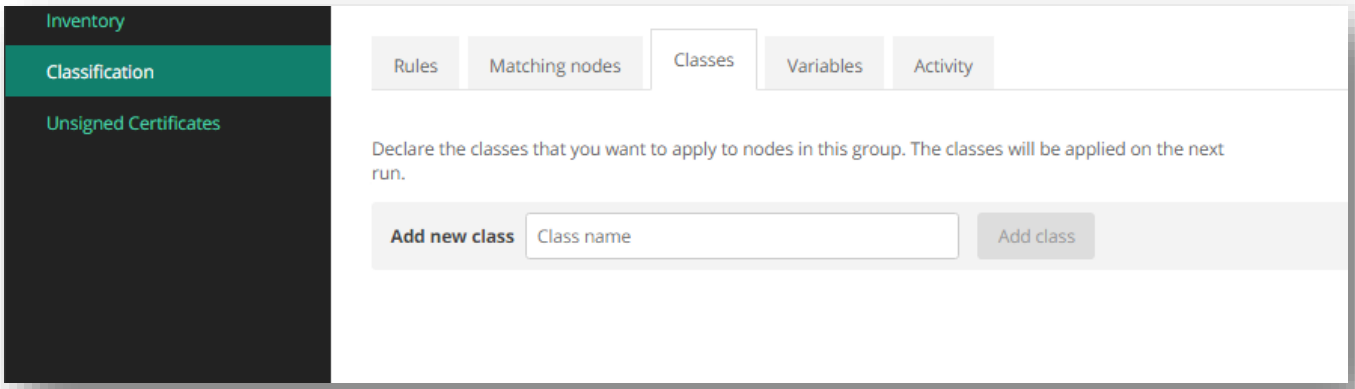


4. Let's take for example we are deploying to QA, navigate to **+QA → +Axis QA**
   You can notice the QA <Application Name>.

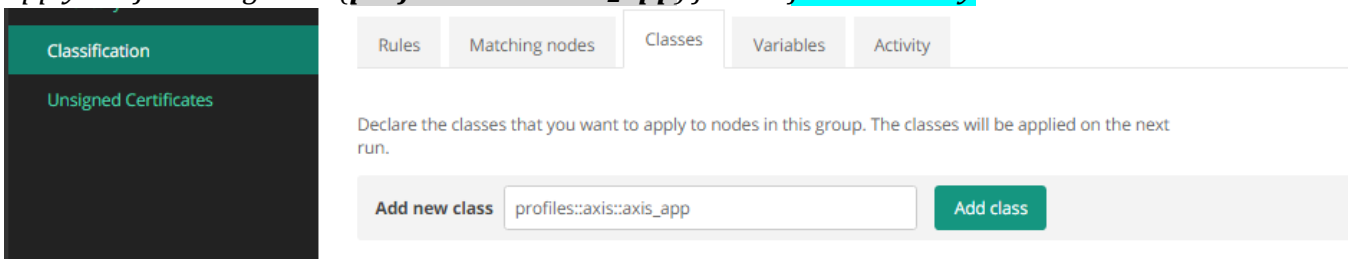


5. Click on each application you want to deploy and look for matching nodes first, and verify the nodes listed are the correct nodes belonging to the application.
   +Axis QA → QA batchprocess → Matching nodes
6. Once confirmed, get into Classes and apply the following classes and parameters.
   +Axis QA → QA batchprocess → Classes

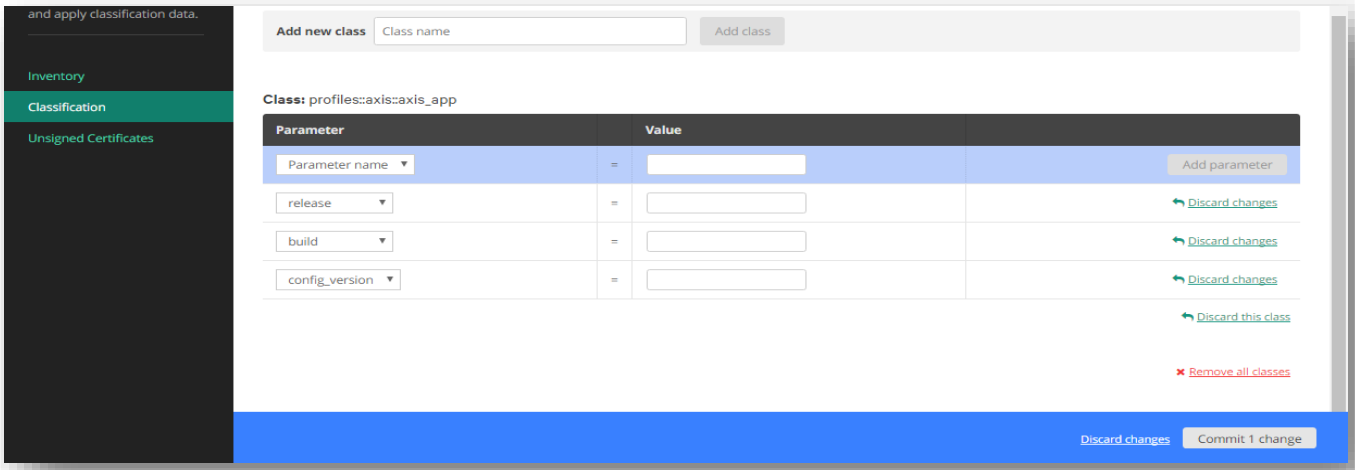
Classes will be empty for the first time:



*Apply the following class (**profiles::axis::axis_app**) for the first time only.*



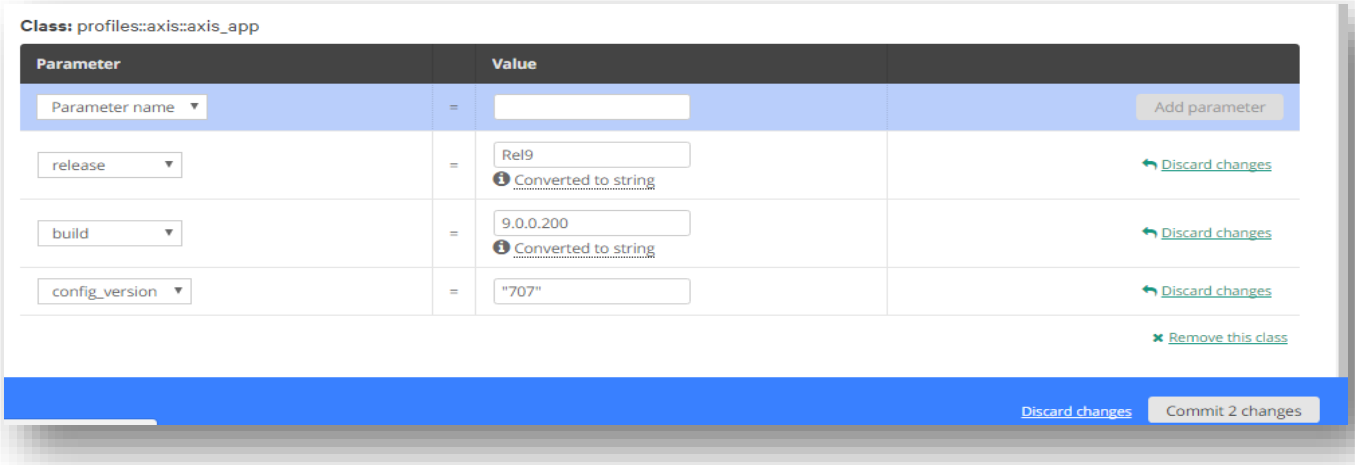
Add the Class and add the following parameters.
Select **parameter name** <parameter> and **Add parameter**.
Add these 3 parameters:



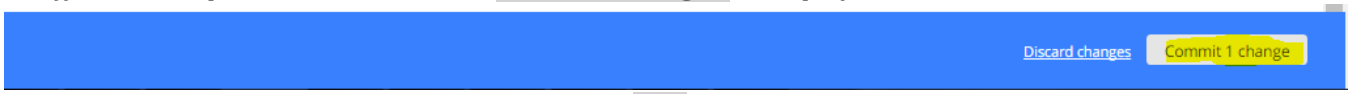
Now, apply the values to the parameters added.

- **Config_version** must be quoted Integer value, eg., → "707"
- **Build** is a 4 digit version, eg., → 9.0.0.200
- **Release** is a string starting with Rel, eg., → Rel9 or Rel9.1

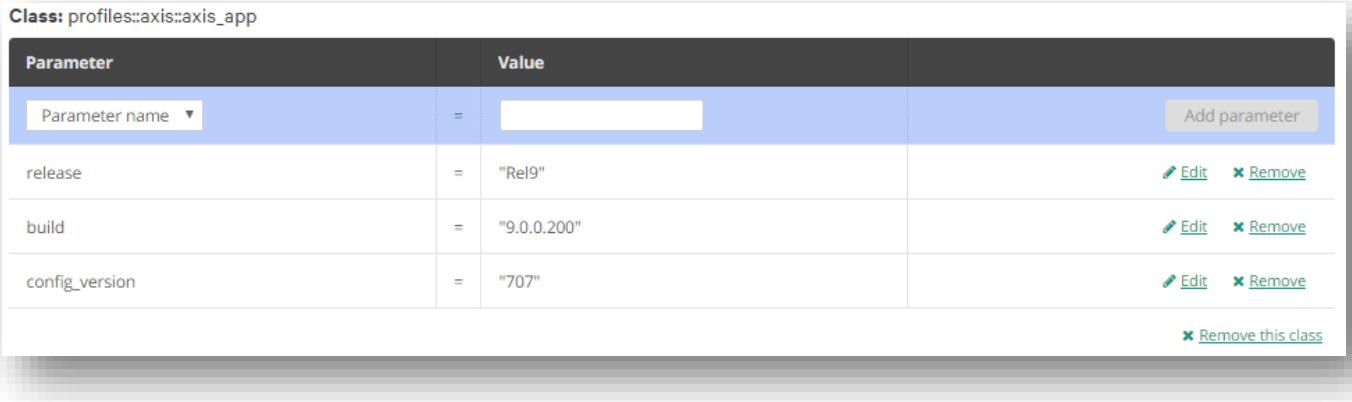
Input those parameters:



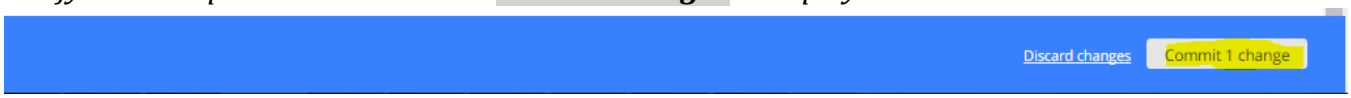
*Verify all those parameters and click **commit changes** to deploy.*



7. If the class is already applied, then just edit the parameters.



*Verify all those parameters and click **commit changes** to deploy.*



8. Continue the same steps (4, 5, 6, and 7) for all other applications except **QA Notification**. **Classification → QA <Application name>**

NOTE:

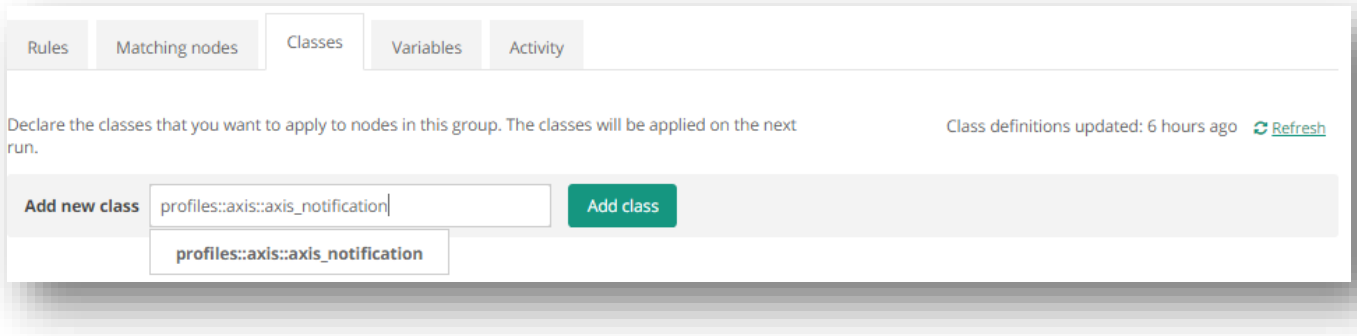
We can deploy one or more specific applications also. For example, we are planning to do a new axis deployment on Origination. So, we can only update the parameters for QA Origination and apply the changes.

By implementing this, we can achieve the *Application-Specific Deployments*.

Notification Server

9. For notification server, the class is different from all other applications.
   Classification → +QA → +Axis QA → QA notification

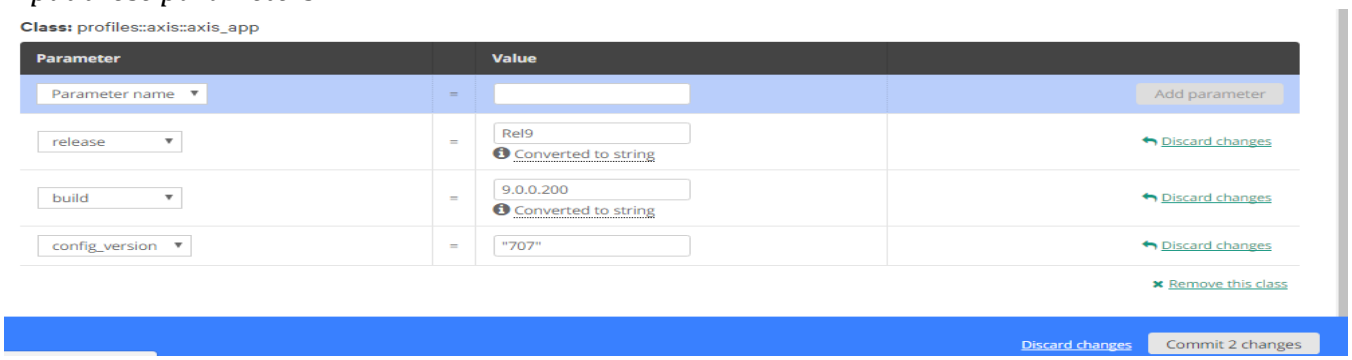
Verify Matching node, then Click classes and add the following class (instead of axis_app) for the first time only. Add new class: **profiles::axis::axis_notification → Add class**



Now, apply the values to the parameters added.

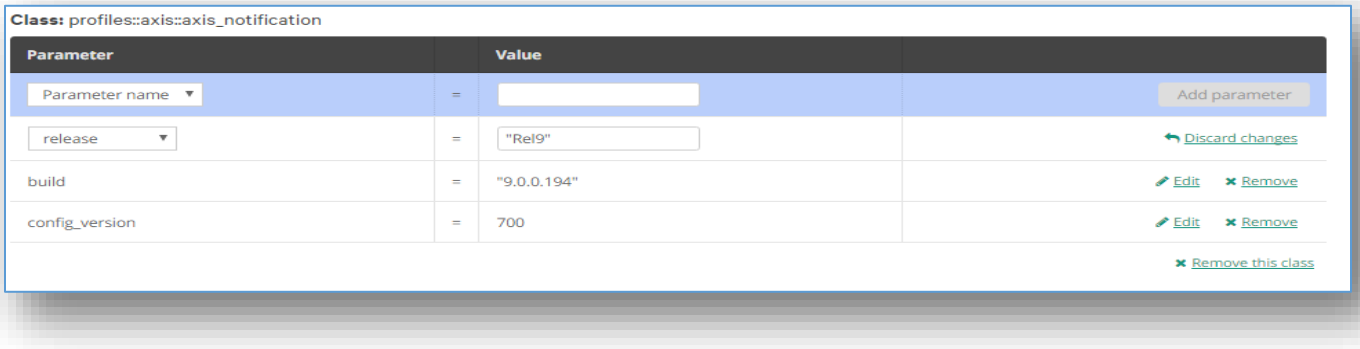
- **Config_version** must be quoted Integer value, eg., → "707"
- **Build** is a 4 digit version, eg., → 9.0.0.200
- **Release** is a string starting with Rel, eg., → Rel9 or Rel9.1

Input those parameters:



*Verify all those parameters and click **commit changes** to deploy.*

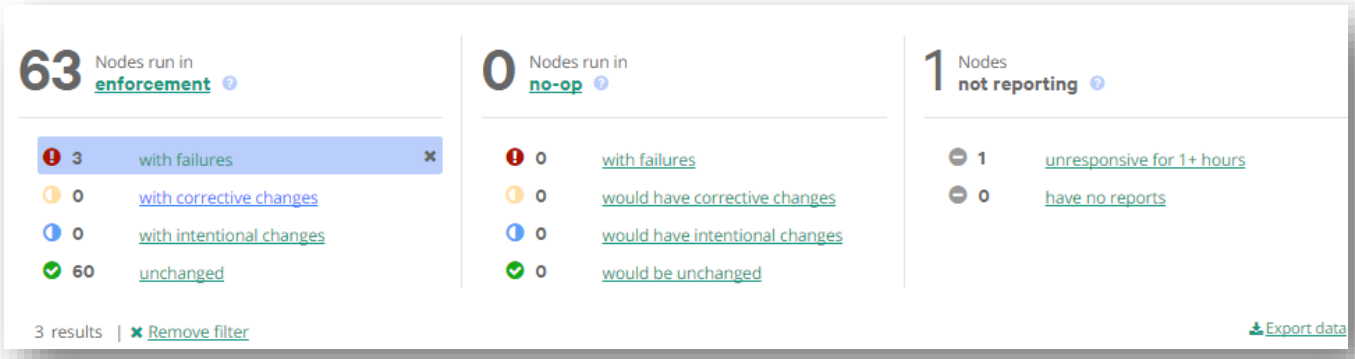
10. If the class is already applied, then just edit the parameters.

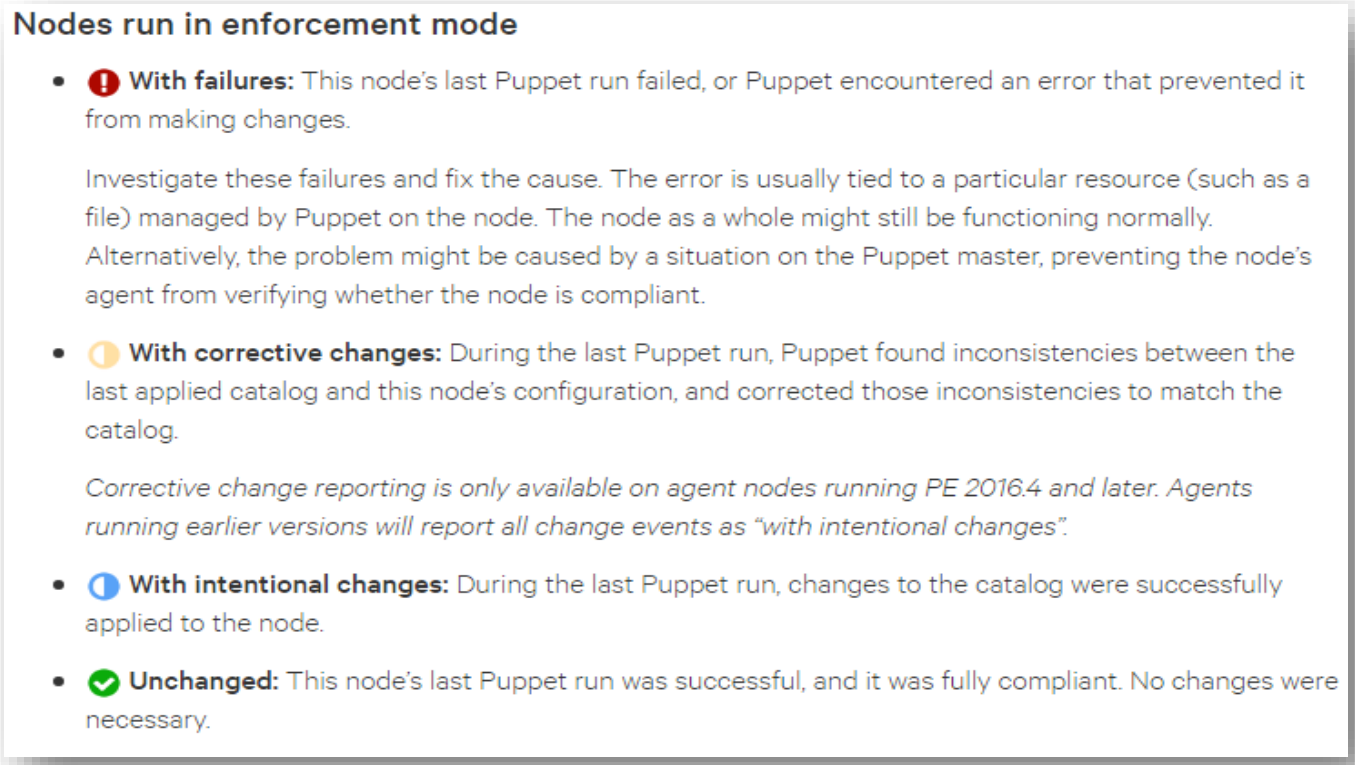


*Verify all those parameters and click **commit changes** to deploy.*

Events and Reports:

11. Once all the values are updated, look for changes in the console in Configuration tab.
    https://docs.puppet.com/pe/2016.5/CM_overview.html





For viewing reports and events, walk through the following documentation in puppet docs.
https://docs.puppet.com/pe/2016.5/CM_events.html

You can also look into event changes with respect to state (failed, changes, unchanged)
Configuration → Events → State (Failure, Corrective, Intentional Changes)

For viewing reports:
https://docs.puppet.com/pe/2016.5/CM_reports.html

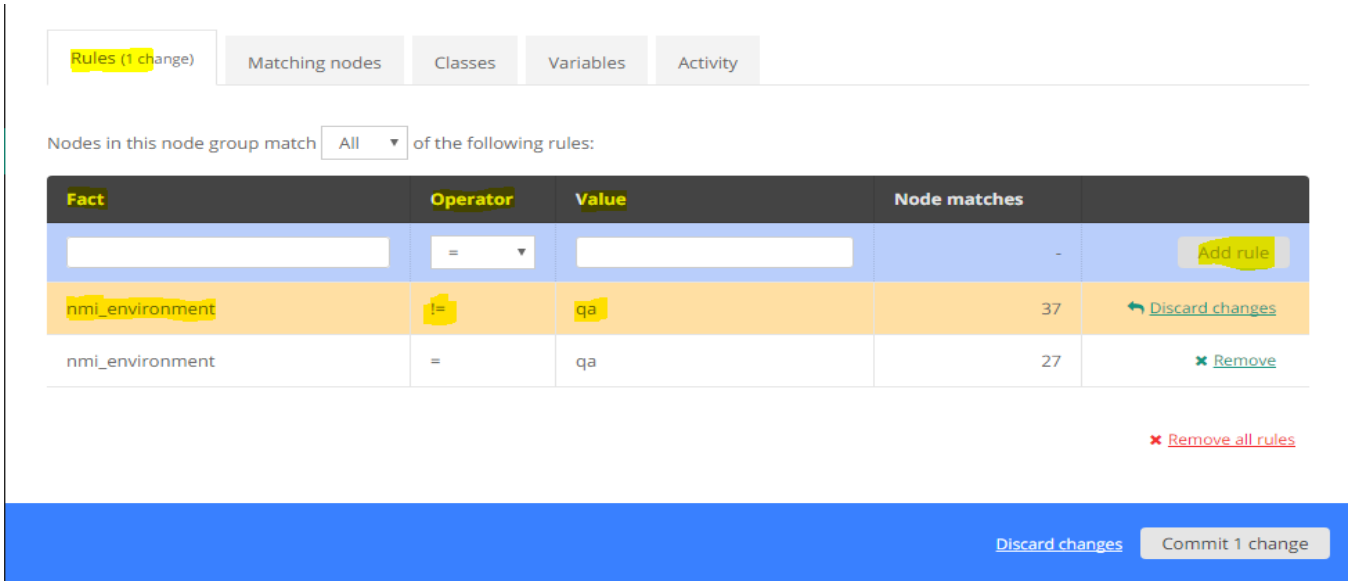
To ensure puppet won't start the tomcat while DB patches are applied:

This can be achieved by not managing QA Nodes to apply *axis* class. Follow these steps to do so:

Classification → QA → Rules

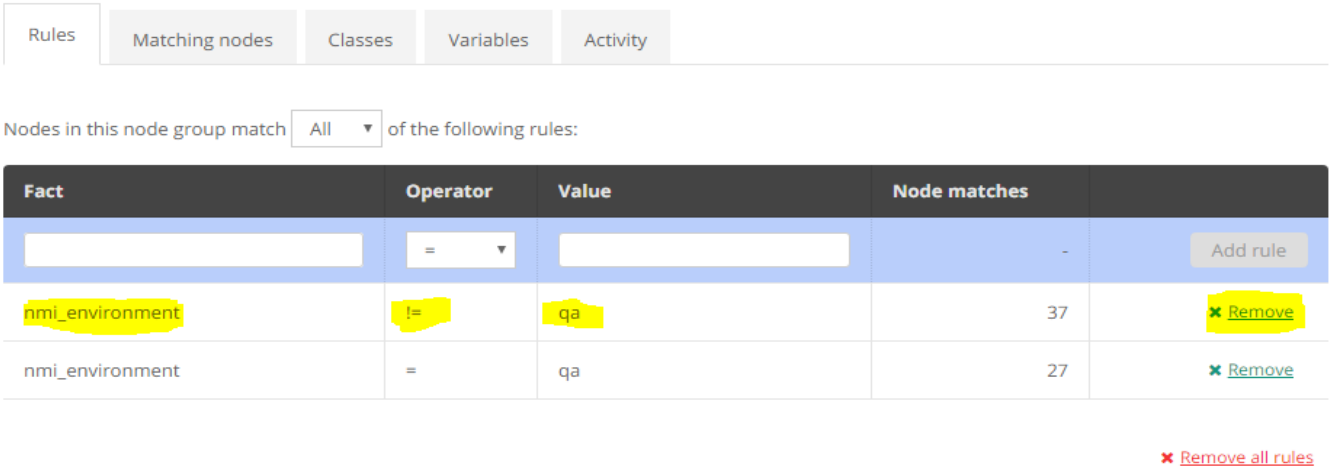
- Add a parameter as mentioned:
  - Fact → **nmi_environment**
  - Operator → **!=**
  - Value → **qa**

  Add rule.



- Before commit, get into **Matching nodes** and you should notice **0 nodes**.
- Commit Changes.
- Now, you can continue with DB patches.
- Once the Db patches are completed.

Classification → QA → Rules → Remove the Parameter just added → Commit.



Look for Matching Nodes → Commit Changes.

To ensure puppet won't start the tomcat and to apply the axis class on specific application:

Instead of disabling puppet and not to manage the axis class on all nodes in particular environment, we can also disable to a specific application also.

Classification → +QA →+Axis QA → QA <Application>

Under rules:

Fact → **nmi_role**

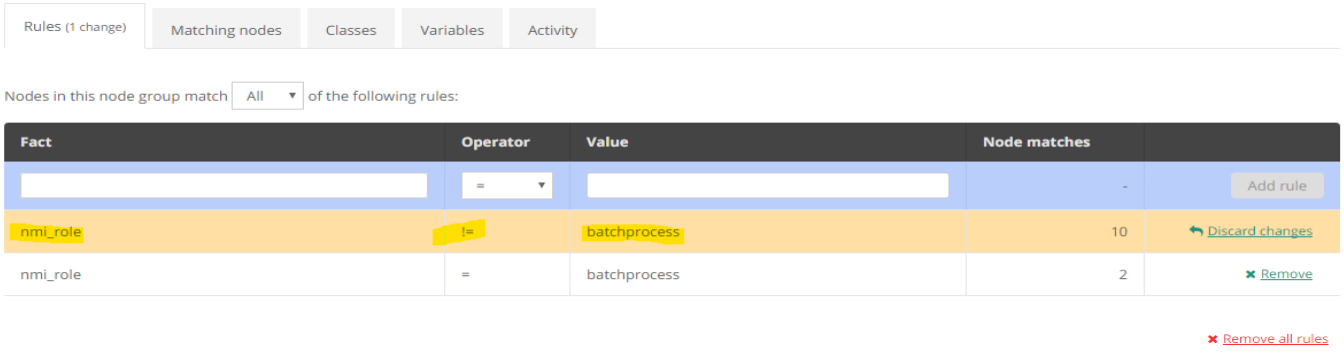
Operator → **!=**

Value → **Application-name**

Add rule, then

Look for 0 Matching Nodes.

Commit Changes.



To enable Puppet again, follow the reverse process.

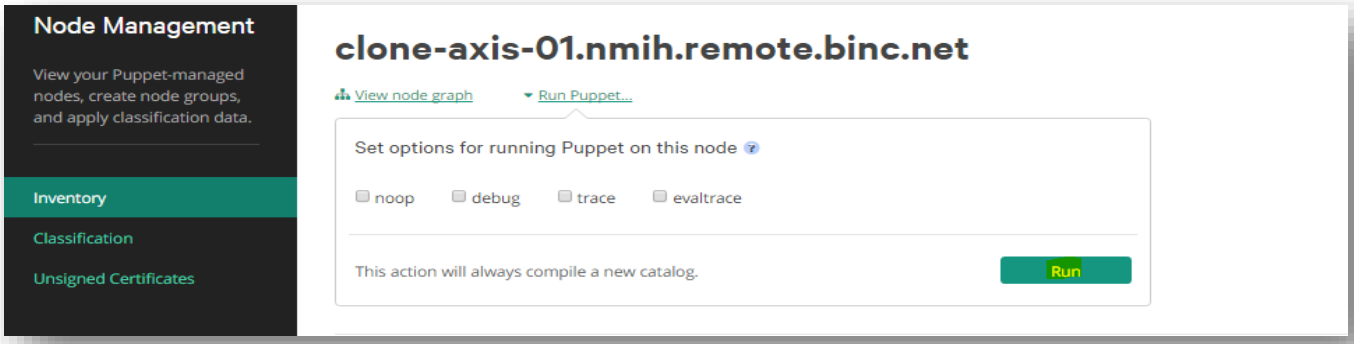
Remove the rule applied before and commit changes.

Remove → nmi_role != batchprocess

Commit Changes

To apply forceful changes without waiting for the puppet agent for another 15min.

Node name → Run Puppet → Run



Puppet/Axis Troubleshooting

AXIS-PUPPET:

Things to remember:

- Make sure to update the right version, build and release parameter, failing to do so would result in deploying the wrong version of deployment.
- Puppet agent will run for every 15min randomly on each server. So, wait until the next report generated to see the events (failure, changes).
- Make sure SVN has the right config version and right builds on artifact server (Nexus).
- Make sure the matching nodes in the classification belongs to respective application.
- Make sure puppet service on nodes is stopped, as puppet agent run is schedule in cron job.
- Make sure only one tomcat is running during the Axis deployments.

Troubleshooting:

- To disable the puppet on specific application or environment, please refer to Doc: *Above.*
- If nodes are unresponsive,
  - They might be disabled or an old puppet instance is running in background.
  - If any old instance running, just kill the older instance and run puppet again.
- Errors like Connection refused may come, if the nodes are not able to access the puppet master.
  - Look into the nodes and make sure all the required ports are opened.

SVN-CONFIG'S:

Troubleshooting:

- *If Puppet failed to deploy the Axis-config's during the Axis deployments?*
  - Go the application config directory (/opt/axis/<app-name>/conf)
  - In /opt/axis/<app-name>/conf/common/ - execute these commands.
    - ***#svn info*** – Check the latest Revision# matches the config_version specified in Puppet parameters.
  - If the config's are still not updated, do this:
    - Remove the /opt/axis/<app-name>/conf directory
    - #***puppet agent –t*** → look for the changes/config's updated during the puppet agent run or wait for next puppet run (15min).
  - To check out the Latest version of the svn config's
    - cd /opt/axis/<app-name>/conf/<common> or <app-name>
    - #***svn update*** → Checkout's the latest version that's in svn server.
    - #***svn info*** → Checkout for the latest revision no.
  - In case to download the svn config's manually, do this:
    - *cd /opt/axis/<app-name>/conf/common*
    - ***#svn --non-interactive switch -r <CONFIG_VERSION> <URL> --force --accept mine-full***
      (or)
    - ***#svn co –r <CONFIG_VERSION> <URL>***
      - CONFIG_VERSION → Puppet Config version
      - SVN URL → URL that's in output of *#svn info*

- **SVN URL**:
  - DEV/QA/PSQA → *svn://10.1.160.31:3690/branches/<Rel #>/<branch>/common*
    → *svn://10.1.160.31.3690/branches/<Rel #>/<branch>/<app-name>*
  - STG/PSS/CI → *svn://10.20.89.10:3690/branches/<Rel #>/<branch>/common*
    → *svn://10.20.89.10:3690/branches/<Rel #>/<branch>/<app-name>*
  - PROD → *svn://10.1.92.121:3690/branches/<Rel #>/<branch>/common*
    → *svn://10.1.92.121:3690/branches/<Rel #>/<branch>/<app-name>*

ARTIFACTS-NEXUS:

Troubleshooting:

- *If Puppet failed to deploy the artifacts (war's) during the Axis deployments?*
  - Remove the old war and run puppet agent –t
  - Still the war's not updating? Download manually, do this:
    - *Go the nexus url in browser, https://nexus-ip:8081/nexus/*
    - Nexus login → No need user/account → Anonymous access enabled.
    - *Repositories → Releases → Browse Storage → Releases → com → nmi → axis → <app-name> → build# → <app-name-build#>.war*
    - *Click on it, In right section → Artifact → Look for Repository Path:*
    - *Right click on Repository Path and copy the link address.*
    - *Use this copied url for the war to download.*
    - *Login the App server to download*
    - cd /usr/share/apache-tomcat/webapps/
    - remove the old-war and <app-directory>
    - ***wget <Repository-path copied>*** → *you can use* ***curl*** *if wget doesn't work.* ***http://<nexus-ip>:8081/nexus/service/local/repositories/releases/content/com/nmi/axis/ <app-name>/<build#> /<app-name-build#>.war***
    - rename the war from *<app-name-build#>.war to <app-name>.war*
    - *Restart the tomcat, if needed.*
  - You can download the war's to a local machine and scp/winscp over to the <app-server> if either wget/curl doesn't work.
- *Sample nexus-path copied:*
  http://10.1.160.133:8081/nexus/service/local/repositories/releases/content/com/nmi/axis/batchprocess/10.0.0.100/batchprocess-10.0.0.100.war

- **NEXUS-URL**:
  - DEV/QA/PSQA → http://10.1.160.133:8081/nexus/
  - STG/PSS/CI → http://10.10.92.133:8081/nexus/
  - PROD → http://10.1.92.129:8081/nexus/

Things to report to DevOps Team:

- ⊗ If any axis application server in an unresponsive state.
- ⊗ If the matching nodes not listed for a specific application.
- ⊗ If Puppet enterprise console is not accessible.
- ⊗ If nodes are unable to apply the latest deployment.
- ⊗ If any errors while applying the axis, might be because of network, ACL, firewall and etc.,
- ⊗ If any new servers need to be added.
- ⊗ If SVN config is not able to update, might be because of SVN_MIRROR.
- ⊗ If war are not able to update, might be because of Nexus
- ⊗ If any file permission changes needed.

JIRA:

Create a ticket for any other puppet error, failure, and other requests.

While creating a ticket for Puppet, include the following components and assign it to the right group.

Application: **Puppet**

Component: **Task: Server- Linux**

Assignee: **Puppet group** (*Larry Ramos*)