

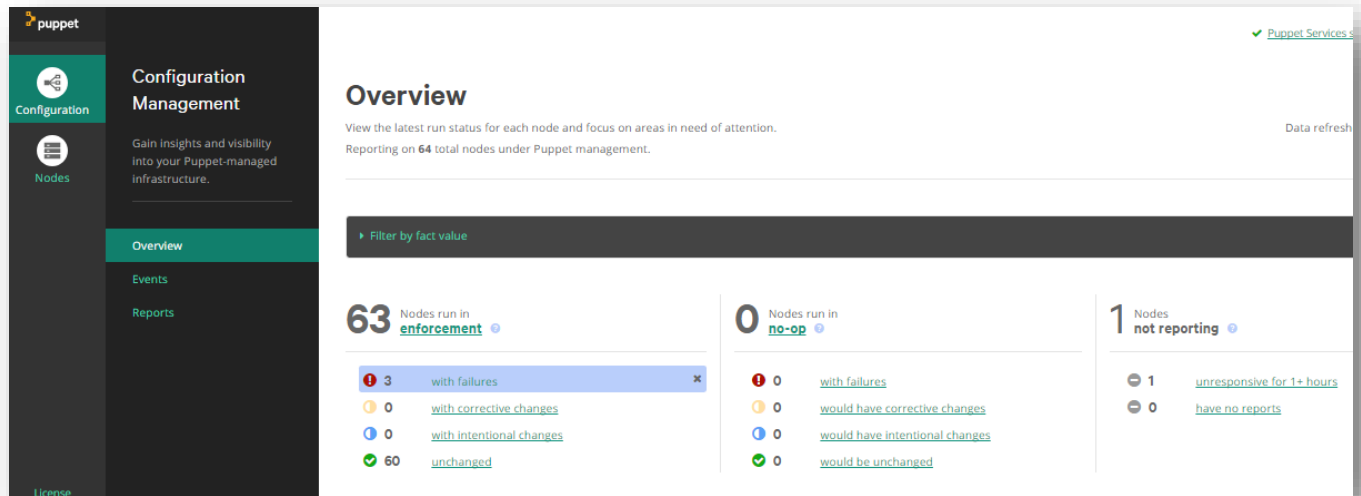
## AXIS Application Deployment Procedure:

1. Login to Puppet Enterprise Console for a specific environment.

<https://10.1.160.53/>

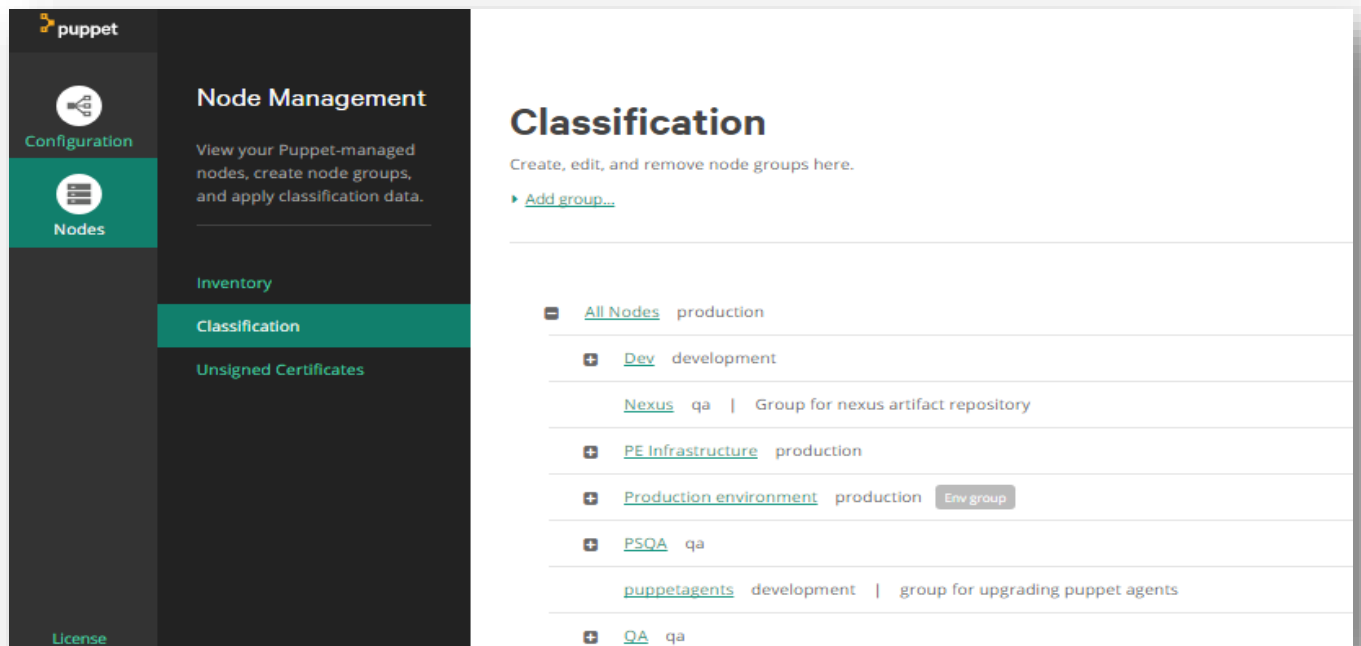
2. Login using your active directory logins to access the console.

You will see the following screen after login:

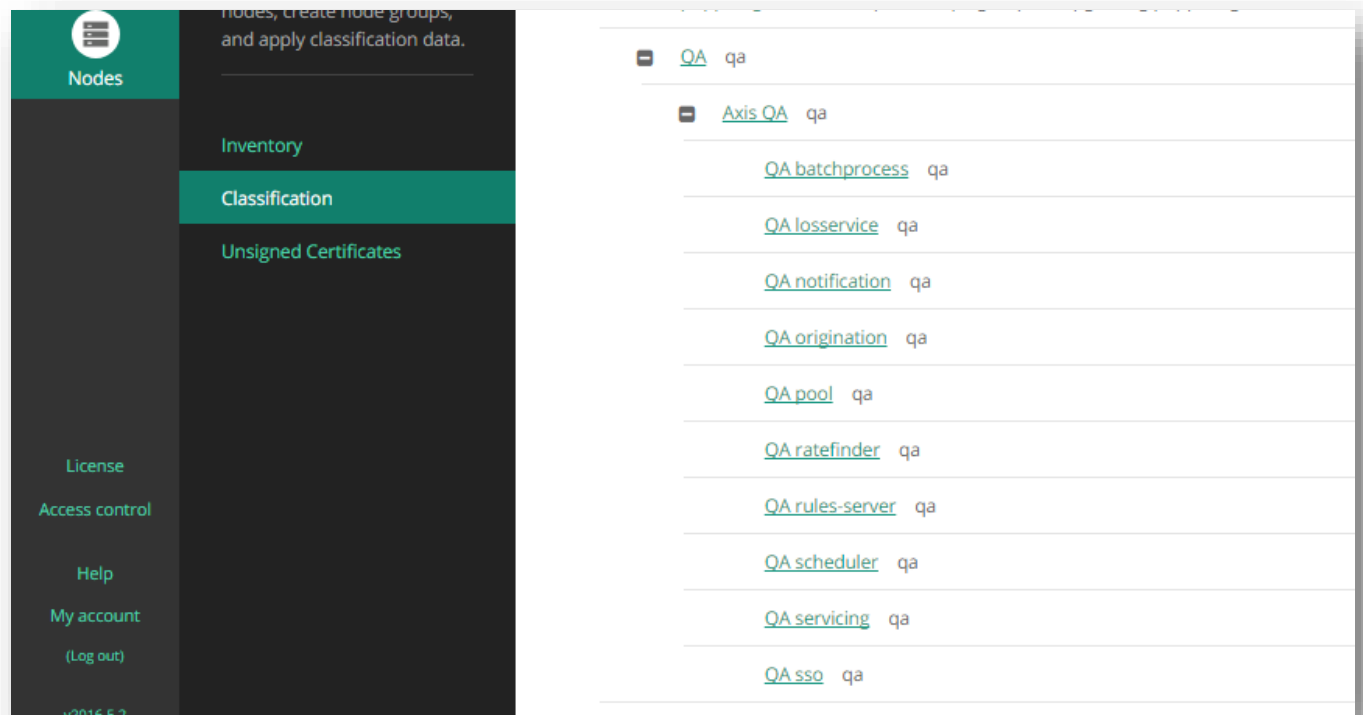


3. Navigate to the **Nodes → Classification** section and expand the environment you will be deploying to.

[https://10.1.160.53/#/node\\_groups/groups](https://10.1.160.53/#/node_groups/groups)

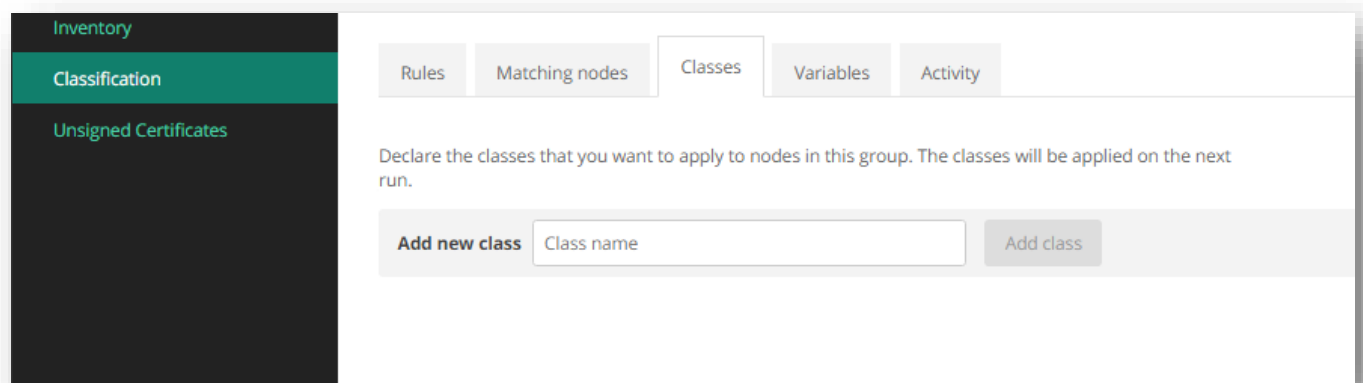


- Let's take for example we are deploying to QA, navigate to **+QA → +Axis QA**  
You can notice the QA <Application Name>.



- 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**
- 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**.

Classification

Unsigned Certificates

Rules Matching nodes **Classes** Variables Activity

Declare the classes that you want to apply to nodes in this group. The classes will be applied on the next run.

Add new class profiles::axis::axis\_app Add class

**Add the Class and add the following parameters.**

Select **parameter name** <parameter> and **Add parameter**.

*Add these 3 parameters:*

and apply classification data.

Inventory

**Classification**

Unsigned Certificates

Add new class Class name Add class

Class: profiles::axis::axis\_app

Parameter		Value	
Parameter name ▼	=		Add parameter
release ▼	=		Discard changes
build ▼	=		Discard changes
config_version ▼	=		Discard changes

Discard this class

Remove all classes

Discard changes Commit 1 change

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:*

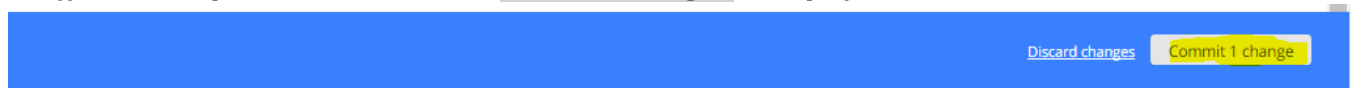
Class: profiles::axis::axis\_app

Parameter		Value	
Parameter name ▼	=		Add parameter
release ▼	=	Rel9	Discard changes
build ▼	=	9.0.0.200	Discard changes
config_version ▼	=	"707"	Discard changes

Remove this class

Discard changes Commit 2 changes

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



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

Class: profiles::axis::axis\_app

Parameter		Value	
Parameter name ▼	=	<input type="text"/>	<button>Add parameter</button>
release	=	"Rel9"	<a href="#">Edit</a> <a href="#">Remove</a>
build	=	"9.0.0.200"	<a href="#">Edit</a> <a href="#">Remove</a>
config_version	=	"707"	<a href="#">Edit</a> <a href="#">Remove</a>
<a href="#">Remove this class</a>			

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**

Rules Matching nodes **Classes** Variables Activity

Declare the classes that you want to apply to nodes in this group. The classes will be applied on the next run. Class definitions updated: 6 hours ago [Refresh](#)

Add new class  [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:

Class: profiles::axis::axis\_app

Parameter		Value	
Parameter name ▼	=	<input type="text"/>	<a href="#">Add parameter</a>
release ▼	=	<input type="text" value="Rel9"/> <small>Converted to string</small>	<a href="#">Discard changes</a>
build ▼	=	<input type="text" value="9.0.0.200"/> <small>Converted to string</small>	<a href="#">Discard changes</a>
config_version ▼	=	<input type="text" value="'707'"/>	<a href="#">Discard changes</a>
<a href="#">Remove this class</a>			

[Discard changes](#) [Commit 2 changes](#)

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

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

Class: profiles::axis::axis\_notification

Parameter		Value	
Parameter name ▼	=	<input type="text"/>	<a href="#">Add parameter</a>
release ▼	=	<input type="text" value="'Rel9'"/>	<a href="#">Discard changes</a>
build	=	<input type="text" value="'9.0.0.194'"/>	<a href="#">Edit</a> <a href="#">Remove</a>
config_version	=	<input type="text" value="700"/>	<a href="#">Edit</a> <a href="#">Remove</a>
<a href="#">Remove this class</a>			

[Discard changes](#) [Commit 1 change](#)

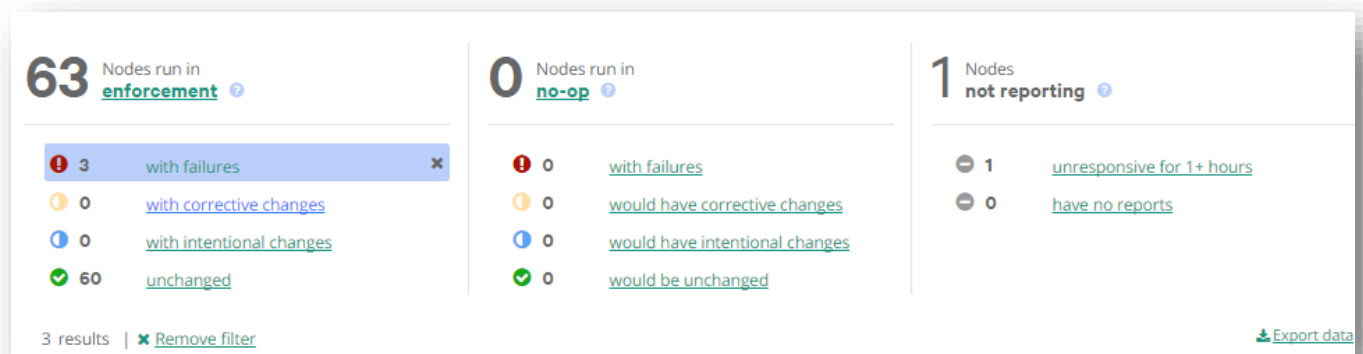
Verify all those parameters and click **commit changes** to deploy.

[Discard changes](#) [Commit 1 change](#)

## 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](https://docs.puppet.com/pe/2016.5/CM_overview.html)



### Nodes run in enforcement mode

- **With failures:** This node's last Puppet run failed, or Puppet encountered an error that prevented it from making changes.  
  
Investigate these failures and fix the cause. The error is usually tied to a particular resource (such as a file) managed by Puppet on the node. The node as a whole might still be functioning normally. Alternatively, the problem might be caused by a situation on the Puppet master, preventing the node's agent from verifying whether the node is compliant.
- **With corrective changes:** During the last Puppet run, Puppet found inconsistencies between the last applied catalog and this node's configuration, and corrected those inconsistencies to match the catalog.  
  
*Corrective change reporting is only available on agent nodes running PE 2016.4 and later. Agents running earlier versions will report all change events as "with intentional changes".*
- **With intentional changes:** During the last Puppet run, changes to the catalog were successfully applied to the node.
- **Unchanged:** This node's last Puppet run was successful, and it was fully compliant. No changes were necessary.

For viewing reports and events, walk through the following documentation in puppet docs.

[https://docs.puppet.com/pe/2016.5/CM\\_events.html](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](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.

Rules (1 change)

Matching nodes

Classes

Variables

Activity

Nodes in this node group match 

All

 of the following rules:

Fact	Operator	Value	Node matches	
<input type="text"/>	<div>=</div>	<input type="text"/>	-	<div>Add rule</div>
<b>nmi_environment</b>	<b>!=</b>	<b>qa</b>	37	<div>Discard changes</div>
nmi_environment	=	qa	27	<div>Remove</div>

Remove all rules

Discard changes

Commit 1 change

- 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.

Rules

Matching nodes

Classes

Variables

Activity

Nodes in this node group match 

All

 of the following rules:

Fact	Operator	Value	Node matches	
<input type="text"/>	<div>=</div>	<input type="text"/>	-	<div>Add rule</div>
<b>nmi_environment</b>	<b>!=</b>	<b>qa</b>	37	<div>Remove</div>
nmi_environment	=	qa	27	<div>Remove</div>

Remove all rules

### 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.**

The screenshot shows the Puppet Enterprise interface. At the top, there are tabs: 'Rules (1 change)', 'Matching nodes', 'Classes', 'Variables', and 'Activity'. Below the tabs, it says 'Nodes in this node group match All of the following rules:'. A table displays the rule configuration:

Fact	Operator	Value	Node matches
nmi_role	!=	batchprocess	10
nmi_role	=	batchprocess	2

Below the table, there are links: 'Discard changes' and 'Remove'. At the bottom right, there is a link: 'Remove all rules'.

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**

The screenshot shows the Puppet Enterprise interface. On the left, there is a sidebar with 'Node Management' and 'Inventory'. The main content area shows the node 'clone-axis-01.nmih.remote.binc.net'. There are links for 'View node graph' and 'Run Puppet...'. Below the links, there is a section 'Set options for running Puppet on this node' with checkboxes for 'noop', 'debug', 'trace', and 'evaltrace'. At the bottom, there is a 'Run' button.



## 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*)