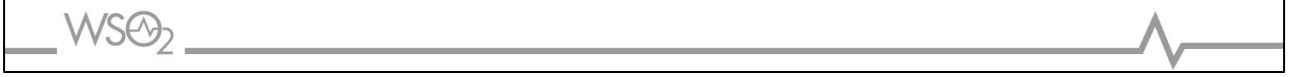

WSO2
WSO2 Product Administration
WSO2 Training
Module 10 - Deployment Automation

Configuration Management

- Maintain and reuse configuration files / templates
- Store using source control
- Configuration version control
- Automate application of configuration changes
- Use of Puppet

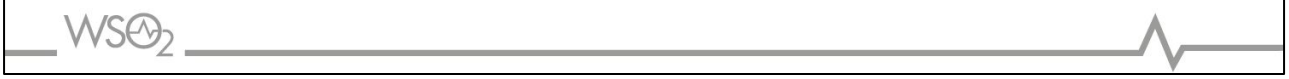


Configuration files directly affect deployment as they contain server specific details and tuning parameters of a setup.
There are several important product configuration files.
carbon.xml contains server settings including port and proxy related information. axis2.xml contains configurations to enable various transports such as HTTP, JMS and FTP. registry related configurations including remote instances and registry mount configurations are available in registry.xml.
In setting up a deployment, we may need to re-use Configuration files.

As a best practise configurations should be stored using a **source control tool**, so that configuration versioning will be applied automatically.
You can either maintain configured configuration files or templates of configuration files and replace the template with appropriate values during deployment setup.
You can have **automated scripts** to copy configuration files / make the configuration changes in the deployment.
Hiera is a key/value lookup tool for configuration data that is used with Puppet. It helps keep specific data out of your manifests. Hiera data act as a site-wide config file which can be requested by Puppet classes when needed.

Artifact Management

- Runtime configurations
  - File system based Deployable artifacts
    - Packaged into Carbon archives (.car)
    - Synapse-configs - ESB configurations (Proxy / API / Endpoint / Sequences)
- Maintain in an artifact repository
- Build with Maven
- Builder integration with source control
- Automated deployment using Puppet



Runtime configuration artifacts such as synapse configurations for proxy services, REST APIs , endpoints and sequences are recommended to be packaged and maintained in Carbon archive files.
In setting up deployment it'll be required to deploy artifacts repeatedly or as required.

You can maintain these artifacts in an artifact repository
The artifact repository can be pointed to jenkins and jenkins will produce build artifacts by building the artifact repository using maven.
Then you can use tools such as puppet to automatically deploy these artifacts on your product deployment.

Automation and Orchestration Tools

- Puppet, Chef, Ansible, CFEngine
- MCollective
- AWS CloudFormation, OpsWork
- Salt

WSO2

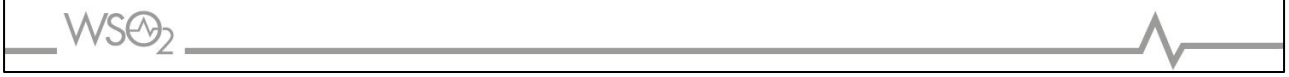
Automation enables performing the deployment without human intervention. Orchestration refers to arranging and coordination of automated tasks, resulting in a consolidated process or workflow
Automation combined with orchestration can be used to setup and manage deployments using tools.

Puppet, Chef, Ansible, CFengine, Salt are all configuration management tools. MCollective (Marionette Collective) is a framework for building orchestration / parallel job execution systems. This supports executing administrative tasks on clusters of servers.
AWS CloudFormation is a tool to create and manage AWS resources, provision and update them where as AWS OpsWorks is again a configuration management tool.

Deployment Orchestration with Puppet

- Set up product instances
- Fetch the deployable artifacts and configurations
- Copy to relevant environment
- Copy to relevant runtime

Using Puppet Modules to Set up WSO2 API-M



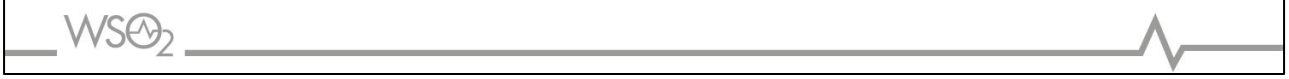
You can orchestrate WSO2 product deployments using tools such as Puppet and Chef.

When the correct parameters are specified, these tools will allow you to set up product instances, push configuration changes, new deployable artifact versions and any other changes to your product clusters without much effort.

Update Management

Updates may contain:

- Java archives (*.jar)
- .war and .jag files
- Instructions for configuration changes
    - Newly added configurations will be optional



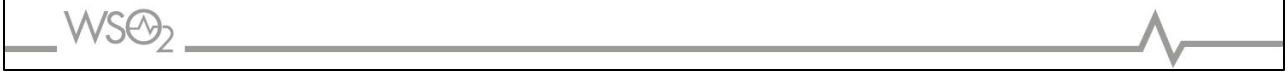
WSO2 provides issue fixes in the form of 'Patches'
These patches may contain Jar archives, .war or .jag files , database scripts, configuration files or changes in configuration files.
To apply a patch you need to follow the README file that is provided with each patch that contains all the necessary instructions.

Product patching can be automated based on different models using tools such as Puppet or mCollective.
Patches can be stored at a single location.
One approach is to deploy patches is to push the patches to all applicable nodes via a patch distribution coordinator at the same time.
Another approach is to deploy the patches on applicable nodes based on the round robin algorithm.

Patches are provided directly for old products, newer products receive patches via WSO2 update manager (WUM).

Procedure

Task 1	Task 2	Task 3
Set up two instances	Set up Puppet master and Puppet agent	Set facter variables and perform a Puppet agent run



THANK YOU

wso2.com