PreTUPS v 7.3.0.

AnsibleUser Manual v 1.7

1. COPYRIGHT STATEMENT AND RESTRICTION

Copyright © 2015Comviva Technologies Ltd, Registered Office at A-26, Info City, Sector 34, Gurgaon-122001, Haryana, India.

All rights reserved. This document shall not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, by any means electronic, mechanical, photographic, optic recording or otherwise without prior consent, in writing, of Comviva Technologies Ltd.

The information in this document is subject to change without notice and describes only the product defined in the introduction of this documentation. This document is intended for the use of prospective Comviva customers for the sole purpose of the transaction for which the document is submitted. No part of it may be reproduced or transmitted in any form or manner whatsoever without the prior written permission of Comviva. The intended audience for this document is the target customer, who/which assumes full responsibility for using the document appropriately. Comvivawelcomes customer comments as part of the process of continuous development and improvement.

Comvivahas made all reasonable efforts to ensure that the information contained in the document is adequate, sufficient and free of material errors and omissions. Comvivawill, if necessary, explain issues, which may not be covered by the document. However, Comviva does not assume any liability of whatsoever nature, for any errors in the document exceptthe responsibility to provide correct information when any such error is brought to Comviva’s knowledge. Comvivawill not be responsible, in any event, for errors in this document or for any damages, incidental or consequential, including monetary losses that might arise from the use of this document or of the information contained in it.

This document and the products, solutions and services it describes are intellectual propertyof Comviva and/or of the respective owners thereof, whether such IPR is registered, registrable, pending for registration, applied for registration or not.

The only warranties for Comviva products, solutions and services are set forth in the express warranty statements accompanying its products and services. Nothing herein should be construed as constituting an additional warranty. Comvivashall not be liable for technical or editorial errors or omissions contained herein.

The Comviva logo is a trademark of Comviva Technologies Ltd. Other products, names, logos mentioned in this document, if any, may be trademarks of their respective owners.

1. Copyright © 2015Comviva Technologies Limited. All rights reserved.

Contents

[1 Document Overview 1-1](#_Toc430271786)

[1.1 Scope 1-1](#_Toc430271787)

[1.2 Audience 1-1](#_Toc430271788)

[1.3 Organization 1-1](#_Toc430271789)

[1.4 Conventions 1-1](#_Toc430271790)

[1.5 Related Documents 1-2](#_Toc430271791)

[1.6 Feedback 1-2](#_Toc430271792)

[2 Ansible Overview 2-1](#_Toc430271793)

[2.1 Ansible Overview 2-1](#_Toc430271794)

[2.2 Control Machine Prerequisites for Installing Ansible 2-1](#_Toc430271795)

[2.3 Host Machine Prerequisites for Installing Ansible 2-1](#_Toc430271796)

[2.4 Ansible Installation 2-2](#_Toc430271797)

[2.5 PreTUPS Installation Using Ansible 2-2](#_Toc430271798)

[2.6 Database Configuration Using Ansible 2-3](#_Toc430271799)



Figures

**No table of figures entries found.**

Tables

[Table 1–1: Conventions 1-1](#_Toc430271800)

[Table 1‑2: Related Documents 1-2](#_Toc430271801)

[Table 2‑1: PreTUPS Installation 2-2](#_Toc430271802)

[Table 2‑2: Database Configuration 2-3](#_Toc430271803)



# Document Overview

1. This chapter is divided into the following sections:

* [Scope](#_Scope)
* [Audience](#_Audience)
* [Organization](#_Organization)
* [Conventions](#_Conventions)
* [Related Documents](#_Related_Documents)
* [Feedback](#_Feedback_1)

## Scope

1. This user manual provides information on using the Ansible platform to install PreTUPS and configure the database. This manual is intended to be used along with the PreTUPS installation manual, which provides information on PreTUPS installation prerequisites and postinstallation activities.

## Audience

1. This user manual is intended for the System Administrators or Operators who need to install PreTUPS and configure the database. The audience should be well versed in Linux, Oracle database maintenance, basic telecommunication concepts, and troubleshooting skills.

## Organization

1. The manual is organized into the following chapters:

* [**Chapter 1— Document Overview**](#_Document_Overview_1):This chapter provides information on the document scope, audience profile, and related documents.
* [**Chapter 2 —Ansible Overview**](#_Ansible_Overview):This chapter provides information on using the Ansible platform to install PreTUPS and configure the database.

## Conventions

1. The following table lists the conventions used in this manual.

Table 1–1: Conventions

| Information | Convention |
| --- | --- |
| UI elements  (such as names of windows, buttons, and fields) | 1. **Bold** |
| References  (such as names of files, sections, paths, and parameters) | 1. *Italic* |
| Hyperlinks | 1. [Hyperlink](http://www.comviva.com) |
| Input  (such as commands and entered text) | Text |
| Output  (such as command output and sample files) | 1. **Output**   Text |
| Keystrokes | 1. **<Key Name>** |
| Examples | 1. **Example** 2. Text |
| Notes | * Text |
| Warnings | * Text |

## Related Documents

1. The following table lists the related documents.

Table 1‑2: Related Documents

| S. No. | Document Name | File Name | Description |
| --- | --- | --- | --- |
| 1 | PreTUPS installation manual | MahindraComviva\_PreTUPS\_v6.8\_IMv1.0 | The PreTUPS installation manual provides information on PreTUPS installation prerequisites and postinstallation activities. |



## Feedback

1. It is our goal to provide you with accurate and useful documentation. Send your comments, suggestions, and feedback to [techwriters@mahindracomviva.com](mailto:techwriters@mahindracomviva.com).

# Ansible Overview

This chapter is divided into the following sections:

* [Ansible Overview](#_Challenges)
* [Control Machine Prerequisites for Installing Ansible](#_Control_Machine_Prerequisites)
* [Host Machine Prerequisites for Installing Ansible](#_Host_Machine_Prerequisites)
* [Ansible Installation](#_Ansible_Installation)
* [PreTUPS Installation Using Ansible](#_PreTUPS_Installation_Using)
* [Database Configuration Using Ansible](#_Database_Configuration_Using)

## Ansible Overview

1. Ansible is a radically simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

Designed for multi-tier deployments, Ansible models your IT infrastructure by describing how all of your systems inter-relate, rather than just managing one system at a time.It uses no agents and no additional custom security infrastructure, so it is easy to deploy. Ansible uses a very simple language (YAML, in the form of Ansible Playbooks) that enables you to describe your automation jobs in a simple way.

For more information on how Ansible works, refer to the following link:

<http://www.ansible.com/how-ansible-works>

Ansible is installed on a control machine, from where it ships contents (such as .war files and configuration files) to host machines to install and configure PreTUPS.

## Control Machine Prerequisites for Installing Ansible

1. Ansible can be run from any machine with Python 2.6 or 2.7 installed (Windows is not supported for the control machine).This includes Red Hat, Debian, CentOS, OS X, any of the BSDs, and so on.
2. For PreTUPS Ansible Installer to run successfully, Ansible Version 2.2.0.0 must be installed on Control machine. PreTUPS Ansible Installer may NOT support Ansible versions prior to this.
3. Paramiko must be Installed on Control Machine. Paramiko is a Python (2.7, 3.4+) implementation of the SSHv2 protocol , providing both client and server functionality

For more information on control machine prerequisites, refer to the following link:

1. <http://docs.ansible.com/ansible/intro_installation.html#control-machine-requirements>

## Host Machine Prerequisites for Installing Ansible

1. On the host machines, you only need Python 2.6 or later, but if you are running less than Python 2.6, you also need python-simplejson.

For more information on host machine prerequisites, refer to the following link:

1. <http://docs.ansible.com/ansible/intro_installation.html#managed-node-requirements>

## Ansible Installation

1. Refer to the following link and install Ansible:
2. <http://docs.ansible.com/ansible/intro_installation.html#installing-the-control-machine>

## PreTUPS Installation Using Ansible

1. Note :

If remote machine (Where application is to be shipped) is AIX machine then home directory must be created manually in advance.

Home directory (ex. pretupshome) should be created manually. The same name must be entered in input variable file in “HOME\_DIR\_NAME” variable.

Ex. Suppose home directory name is to be “pretupshome” on remote machine then pretupshome must be created at “/” location manually and inside inout file (ex. WEB.yml) entry for HOME\_DIR\_NAME should be

HOME\_DIR\_NAME: pretupshome

* Any instance (WEB, SMSR, SMSC) can be shipped independently or all at once on remote machine.
* Number of input variable files will be as many as number of instances i.e. If 3 instances needs to be shipped then there will be 3 different instance\_variable.yml files inside host\_vars
* If all the instances need to be shipped at same location (say /pretupsHome) and logs of every instance needs to be kept at same location (say /pretupsVar) at same remote machine then GROUP\_NAME , USER\_NAME, HOME\_DIR\_NAME and LOG\_DIR\_NAME must be same in all the instance\_variable files.
* Use driver Class “**org.postgressql.Driver**” for **Postgres Database**. Similarly

Use driver Class “**oracle.jdbc.driver.OracleDriver**” for **Oracle Database**

1. The following table lists the steps to install PreTUPS using Ansible.

Table 2‑1: PreTUPS Installation

| Step | Action |
| --- | --- |
|  |  |
|  | Open the */etc/hosts* file on the control machine. |
|  | Mention as many unique <IP><hostname> combinations as the number of Tomcat to be shipped on each host machine.  **Sample**  etc_hosts.png  Example :  C:\Users\satakshi.gaur\Desktop\Capture.PNG |
|  | Save and close the file. |
|  | Copy the following files to the control machine:   * *pretups\_app.tar.gz* * *pretups.war* * *jdk1.8.0\_74.tar.gz* * (tomcats).tar.gz |
|  | Untar *pretups\_app.tar.gz*. |
|  | Navigate to pretups\_app |
|  | Open “site.yml” file |
|  | comment : - name: "Running the WEB playbook"  hosts: appservers  - include: WEB\_site.yml  if WEB instance needs NOT to be shipped  comment : - name: "Running the SMSR playbook"  hosts: smsr\_servers  - include: SMSR\_site.yml  if SMSR instance needs NOT to be shipped  comment : - name: "Running the SMSP playbook"  hosts: smsp\_servers  - include: SMSP\_site.yml  if SMSP instance needs NOT to be shipped  Example :  C:\Users\satakshi.gaur\Desktop\Capture.PNG  Save and close the file |
|  | Open the “*hosts”* file. |
|  | Enter instance\_name mentioned in /etc/hosts according to their group like this :  Under [appservers] group mention the instance\_name for WEB instance  Under [smsr\_servers] group mention the instance\_name for SMSR instance  Under [smsp\_servers] group mention the insance\_name for SMSP group  **Sample**  C:\my data\Sumit_Work\Projects\Ansible\hosts_appservers_2.png  Example :  C:\Users\satakshi.gaur\Desktop\Capture.PNG  Save and close the “hosts” file |
|  | Copy *pretups.war* to the below path:  roles/copy\_war/files |
|  | Copy *jdk1.8.0\_74.tar.gz* to the below path:  roles/copy\_java/files |
|  | Copy SMSCGateway.tar.gz to the below path: (If SMSCGateway is needed to configure)  roles\copy\_SMSCGateway\files |
|  | Navigate to the *host\_vars* directory. |
|  | Copy the existing WEB*.yml* file to create as many variable files as the number of Tomcats to be shipped for WEB instances, with file name as *<hostname>.yml*, where *<hostname>* is the hostname defined in the */etc/hosts* file and same defined in “hosts” file  **Sample**: *IP1\_instance1.yml*, *IP1\_instance2.yml ….*  Example : cp WEB.yml tomcat\_web1.yml  cp WEB.yml tomcat\_web2.yml  Similarly , for SMSR Copy the existing “SMSR*.yml”* file to create as many variable files as the number of Tomcats to be shipped for SMSR instances, with file name as *<hostname>.yml*, where *<hostname>* is the hostname defined in the */etc/hosts* file and same defined in “hosts” file  Example : cp SMSR.yml tomcat\_smsr1.yml  cp SMSR.yml tomcat\_smsr2.yml  And again for SMSP, Copy the existing “SMSP*.yml”* file to create as many variable files as the number of Tomcats to be shipped for SMSP instances, with file name as *<hostname>.yml*, where *<hostname>* is the hostname defined in the */etc/hosts* file and same defined in “hosts” file  Example : cp SMSP.yml tomcat\_smsp1.yml  cp SMSP.yml tomcat\_smsp2.yml |
|  | Open each variable file created in above step and modify the value of variables. For help regarding variables open attached file in notepad++.        Note1: In case of Sudo User(root is not present), provide sudo user details in ansible\_ssh\_user and ansible\_ssh\_pass. Sudo Users must be able to run all the root commands. In order to avoid Linux Prompt asking for password, allow Sudo User to run commands without password in sudoers (visudo) file. This is prerequisite for executing Ansible using sudo user.  Note2 : SHUTDOWN\_PORT, CONNECTOR\_PORT and AJP\_CONNECTOR\_PORT must be unique in each variable file because they are for Server.xml otherwise there will be error in starting tomcat. |
|  | Save and close each variable file. |
|  | Copy tomcat inside pretups\_app/roles/copy\_tomcat/files having same name as specified in *<hostname>.yml file for variable tomcat\_name.*  *Note : For every instance there must be different tomcats inside p*retups\_app/roles/copy\_tomcat/files  If only one tomcat is kept here and same tomcat is tried to be shipped for each instance then there will be problem while shipping tomcat to remote location because tomcat of same name cannot exist at same location for each instance (if / HOME\_DIR\_NAME is same for every instance)  Note 2 : If only one tomcat is provided then follow these steps :   * 1. untar it using following command :   tar –xvzf tomcat\_name.tar.gz   * 1. Now create copies of this tomcat for each instance by following command :   cp –R source\_tomcat\_name new\_tomcat\_name   * 1. Make tar of this tomcat by using following command :   tar –cvzf tomcat\_name.tar.gz tomcat\_name  Here tomcat\_name must be same mentioned in each variable file created in above steps. |
|  | Navigate to the *pretups\_app* directory. |
|  | Install the PreTUPS application by executing following command :  ansible-playbook -i hosts site.yml –vvvv –c paramiko  Sometimes, for some machines after executing this command, it asks for (yes/no) to continue operation on that machine for fingerprint. In that case, enter “yes”  Sample logs that are generated on running the preceding command are attached. |

## Database Configuration Using Ansible

1. The following table lists the steps to configure the database using Ansible.
2. **A.** **ORACLE DATABASE**

Table 2‑2: Database Configuration

| Step | Action |
| --- | --- |
|  | Ensure that the following prerequisites are met:   * Oracle is installed. * IP, port, and SID are available. * *oracle* user is created with *oinstall* as the group. * Permission should be granted to orbain folder through root. * /srv folder permission should be granted to import the db dumb properly. * Database dump is available. * Manual Scripts File (DDL/DML) inside roles/db\_task/templates folder should contain exit statement at the end * *.bash\_profile* content is copied to the *.bashrc* file of *oracle* user. * .bash profile parameters ORACLE\_HOSTNAME ,ORACLE\_BASE and ORACLE\_HOME should be present in .bashrc file. * *Import command : If export is done using expdp then impdp command should be used and if export is done using emp then imp command should be used for dump.* * Following lines are added to the *.bashrc* file of *oracle* user:   http\_proxy=  export http\_proxy   * The *http\_proxy* variable must not be assigned any value. |
|  | Navigate to the *pretups\_app* directory. |
|  | Open db\_site.yml file. |
|  | if database is RAC then comment following lines in db\_site.yml file :    - name: "Running the dbConf.yml for standalone Database"  hosts: dbservers  - include: dbconf.yml  if database is standalone then comment following lines :  - name: "Running the dbconfRAC.yml for RAC Database"  hosts: dbservers  - include: dbconfRAC.yml if database is not RAC  if NO dump import is required and only Scripts are to be executed then comment following :  - name: "Running the dbConf.yml for standalone Database"  hosts: dbservers  - include: dbconf.yml  - name: "Running the dbconfRAC.yml for RAC Database"  hosts: dbservers  - include: dbconfRAC.yml  And uncomment following :  # - name: "Running Manual Scripts Only and No Dump"  # hosts: dbservers  # user: "{{ ansible\_ssh\_user }}"  # gather\_facts: no  # hosts: dbservers  # serial: 1  # roles:  # - db\_task  Example:  Save and close db\_site.yml |
|  | Copy the database dumps to the following path:  roles/db\_ship\_scripts/files  If DB dump needs NOT to be imported and only DB scripts need to be executed on existing Schema then this step is not needed. |
|  | If Only DB Scripts are to be executed (Without importing DB dump ) then follow this step otherwise skip it.  copy and paste all the scripts file(DDL/DML) inside pretups\_app\roles\db\_task\templates folder.  And enter “true” in DB\_instance\_variable\_file in following key:  IsManualScripts: true  And provide Manual Scripts Name in below format without extension:  ManualScriptsName: [SQLScripts1,SQLScripts2,SQLScripts3]   * Name of the scripts mentioned here should match to the sql scripts kept in above mentioned path. * Extension of the scripts must be “.sql”. * Last word in each script file must be “exit;”     These scripts will be executed in the same order (left to right) after importing all the DB dumps. |
|  | If DB dump import is also needed and after that some extra scripts need to be executed then follow this step :  If any additional scripts are required to execute then copy and paste all such scripts in following mentioned file name :  pretups\_app/roles/db\_task/files/ SQLScripts.sql  and enter “true” in DB\_instance\_variable\_file in following key :  AdditionalQueries: true  These scripts will be executed after importing all the DB dumps. |
|  | Navigate to the *host\_vars* directory. |
|  | Rename the existing DB*.yml* file to *<database IP>.yml*, where *<database IP>* is the IP of the server where Oracle is installed.   * Ensure that the renamed file ends with *.yml*(and not *.yml\_ORACLE*) extension.   USE THIS **SOURCE SCHEMA** in the variable file: **PRETUPS670\_ITR3** |
|  | Open the variable file renamed in the preceding step and update the value of the variables as appropriate.  For reference open following file in notepad++     * Ensure that the database dump file name is assigned to the *Dump\_name* variable in the variable file. |
|  | Save and close the variable file. |
|  | Navigate to the *pretups\_app* directory. |
|  | Open the *hosts* file. |
|  | Under the *[dbservers]* group name, overwrite the existing IP or instance name with the database IP. |
|  | Save and close the *hosts* file. |
|  | Configure the database:  ansible-playbook -i hosts db\_site.yml -vvvv -c paramiko  Sample logs that are generated on running the preceding command are attached. |
|  | If Rest Configuration is needed then For Rest Configuration refer to “[Rest Configuration.docx](javascript:void(0)) “ from RFR attachments. |
|  | Restart the tomcat that has been shipped in the application stage. |

1. **B.** **POSTGRESQL DATABASE**

Table ‑3: Database Configuration

| Step | Action |
| --- | --- |
| 1. | Ensure that the following prerequisites are met:   * Postgres is installed. * IP, port, are available. * *postgres* super user is created. * Manual Scripts File (DDL/DML) inside roles/db\_task/templates folder should contain exit statement at the end * *.bash\_profile* content is copied to the *.bashrc* file of *postgres* user.   .Dump should be available only of following format.SQL  DMP  NOTE!!!: DUMP should be exported using below command only  ‘pg\_dump -v -C { SOURCE DATABASE } -Fc -f { DUMP NAME }’ |
| 2 | Navigate to the *pretups\_app* directory. |
| 3 | Open db\_site.yml file. |
| 4. | For Postgres uncomment the following  - name: "Running dbConfPostgres for postgres database"  hosts: dbservers  - include: dbconfPostgres.yml    Save and close db\_site.yml |
| 5. | Copy the database dumps to the following path:  roles/db\_ship\_scripts\_postgress/files  If DB dump needs NOT to be imported and only Schema is created then this step is not needed. |
| 6. | If Only Schema Scripts are to be executed (Without importing DB dump ) then follow this step otherwise skip it.  copy and paste all the scripts file(DDL/DML) inside pretups\_app/roles/db\_ship\_scripts\_postgress/files/Postgress.  And enter “false” in DB\_instance\_variable\_file in following key:  ISDUMP: false  And provide Schema Scripts Name in below format without extension:  PRETUPS\_POSTGRES\_SchemaFile: [pretups\_schema1, pretups\_schema2]   * Name of the scripts mentioned here should match to the sql scripts kept in above mentioned path. * Extension of the scripts must be “.sql”.   These scripts will be executed in the same order (left to right) after importing all the DB dumps. |
| 7. | If DB dump import is also needed and after that some extra scripts need to be executed then follow this step :  If any additional scripts are required to execute then copy and paste all such scripts in following mentioned file name :  pretups\_app/roles/db\_ship\_scripts\_postgress/files/Postgress  and enter “true” in DB\_instance\_variable\_file in following key :  IsManualScripts: true  And provide Manual Scripts Name in below format without extension:  ManualScriptsName: [SQLScripts1,SQLScripts2,SQLScripts3]   * Name of the scripts mentioned here should match to the sql scripts kept in above mentioned path. * Extension of the scripts must be “.sql”.   These scripts will be executed in the same order (left to right) after importing all the DB dumps. |
| 8 | Navigate to the *host\_vars* directory. |
| 9 | Rename the existing DB\_postgres*.yml* file to *<database IP>.yml*, where *<database IP>* is the IP of the server where Postgresql is installed.   * Ensure that the renamed file ends with *.yml*(and not *.yml\_POSTGRES*) extension. |
| 10 | Open the variable file renamed in the preceding step and update the value of the variables as appropriate.  For reference open following file in notepad++    Ensure that the database dump file name is assigned to the PRETUPS\_POSTGRES\_DUMPvariable in the variable file. |
| 11 | Save and close the variable file. |
| 12 | Navigate to the *pretups\_app* directory. |
| 13 | Open the *hosts* file. |
| 14 | Under the *[dbservers]* group name, overwrite the existing IP or instance name with the database IP. |
| 15 | Save and close the *hosts* file. |
| 16 | Configure the database:  ansible-playbook -i hosts db\_site.yml -vvvv -c paramiko  Sample logs that are generated on running the preceding command are attached. |
| 17 | If Rest Configuration is needed then For Rest Configuration refer to “[Rest Configuration.docx](javascript:void(0)) “ from RFR attachments. |
| 18 | Restart the tomcat that has been shipped in the application stage. |

Document Change History

The following table provides the document change history.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change Type | Description | Date |
| 1.0 | Initial | Initial | 24 June, 2016 |
| 1.1 | update | Added SMSC gateway steps | 10 Feb, 2016 |
| 1.2 | update | RAC DB related Changes done | 14 March, 2017 |
| 1.3 | update | Manual Scripts instead of Dump | 20 March, 2017 |
| 1.4 | update | Directory creation at home of Oracle User, configurable Table spaces name and datafile location | 2nd May, 2017 |
| 1.5 | update | Ansible scripts updated to support postgres database installation | 7th June, 2017 |