IMPLEMENTION GUIDE

Oracle Patching Automation Solution using Ansible

Version	Date
Version 1.0	16 April 2022

Author: Srivastava, Ayush

Table of Contents

(Click on the topic to directly navigate to them)

Section I: Introduction 3

Purpose: 3

Target Audience: 3

Technical Pre-requisites of Target server for automated solution: 3

Section II: Technical Components Used in Patching Ansible Solution 4

Section III: What we are automating in Oracle Patching Process? 5

Manual Efforts in the current patching process: 5

Ansible Job templates and their role to resolve manual efforts in process 6

Section IV: Steps of Performing Patching through Ansible 7

Phase A: Planning Phase (server consolidation) 7

Operational Steps of Phase A (Planning):- 8

Phase B: Prepatch Phase (PSU analyze) 11

Operational Steps of Phase B (Precheck):- 12

Phase C: Patch Phase 14

Operational Steps of Phase C (Patch):- 14

Phase D: DataPatch Phase 16

Operational Steps of Phase D (Datapatch):- 16

Section V: Miscellaneous 17

A. Frequently Asked Questions (FAQs): 17

B. General Document of Technical issues during Patching and their Resolutions 19

Section I: Introduction

Purpose:

Patching Oracle servers through Ansible Solution is designed to minimal the DBA intervention which requires as part of Oracle GRID and RDBMS patching activity.

Target Audience:

The document is designed as implementation guide for DBA team member, who is involved in patching activity for Oracle servers. It is to note automation is designed as part of removing manual efforts in patching however Oracle patching knowledge for DBA is still mandatory.

Technical Pre-requisites of Target server for automated solution:

- Database setup type: 19c Grid (Single node or RAC). Non Grid and 12c Grid are not supported.
- Database version: 19c or 12cR2 (12.2) or both.
- OS type: Linux Only.

Section II: Technical Components Used in Patching Ansible Solution

Ansible code is used as for automation in configuration and patch provisioning across servers.

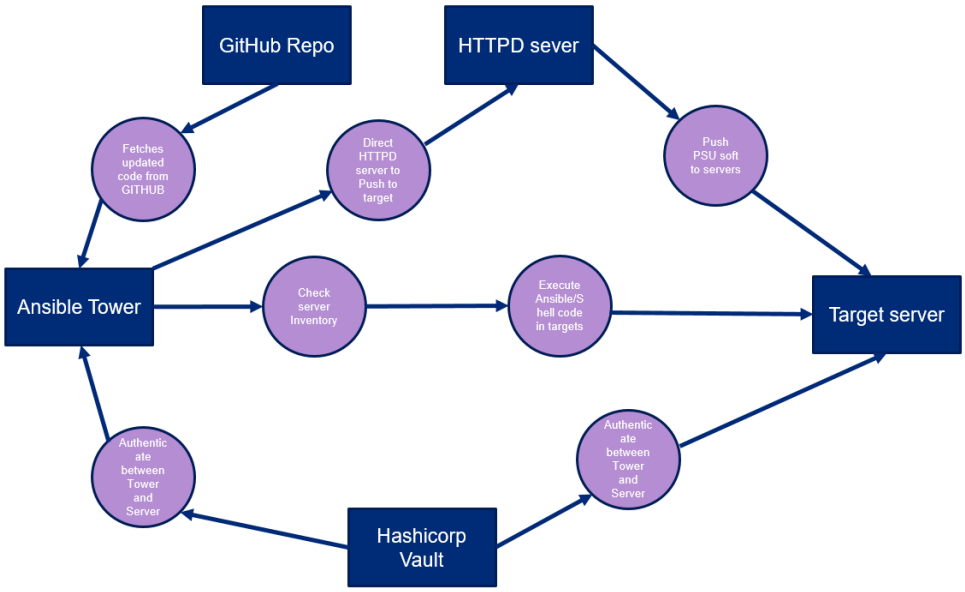
Shell script used as native OS language of Linux

A web based automation controller which centralizes and controls Ansible infrastructure

HashiCorp Vault is a secrets management tool which secures, stores, and tightly controls access to tokens, passwords, certificates etc

GitHub is the distributed version control and source code management service.

Apache HTTP Service, an open-source software, used for hosting PSU software and pushing the software to target server.



MarshMcLennan

Author: Srivastava, Ayush

Section III: What we are automating in Oracle Patching Process?

Manual Efforts in the current patching process:

- Moving PSU software from OEM/OMS server to target server, which is required to be patched.
- Replacing OPatch directory from GRID and RDBMS home with latest one provided by Oracle.
- Running opatch analyze (pre-checks) on GRID/RDBMS before day of patching scheduled.
- Logging to target server individually and taking snapshot of the server resources before patching the server.
- Setting OEM blackout to suppress alerts during patching of the GRID/RDBMS.
- Patching the GRID/RDBMS binaries manually.
- Taking database registry details before running datapatch individually.
- Running Datapatch manually on each database.
- Taking database registry details post datapatch individually.
- Removing OEM blackout from OEM.
- Removing PSU software directory which is copied to target server, after patching completed.

Ansible Job templates and their role to resolve manual efforts in process

The Patching solution through Ansible is divided among 6 Job templates of which 4 are essential and 2 are optional.

ORCL DBS Copy PSU Binaries Playbook (OPTIONAL)

- Moving PSU software from Master servers (server maintain master copy of PSU software) to target server, which is required to be patched, using httpd service through script.
- Identify Oracle Binaries before pushing the required PSU in the server.

ORCL DBS Precheck Analyze Playbook (ESSENTIAL)

- Replacing OPatch directory from GRID and RDBMS home with latest one provided by Oracle using script.
- Running OPatch analyze (pre-checks) on GRID/RDBMS before day of patching scheduled using script.
- Logging to target server individually and taking snapshot of the server resources before patching the server.

ORCL DBS PSU Patching Playbook (ESSENTIAL)

- Setting OEM blackout to suppress alerts during patching of the GRID/RDBMS.
- Patching the GRID/RDBMS binaries using script.

ORCL DBS Data patch apply Playbook (ESSENTIAL)

- Taking database registry details before running datapatch individually.
- Running Datapatch using script on each database on each database.
- Taking database registry details post datapatch individually.

ORCL DBS Housekeeping PSU (ESSENTIAL)

- Removing OEM blackout from OEM.
- Removing PSU software directory which is copied to target server, after patching completed.

ORCL DBS PSU Patching Resume (OPTIONAL)

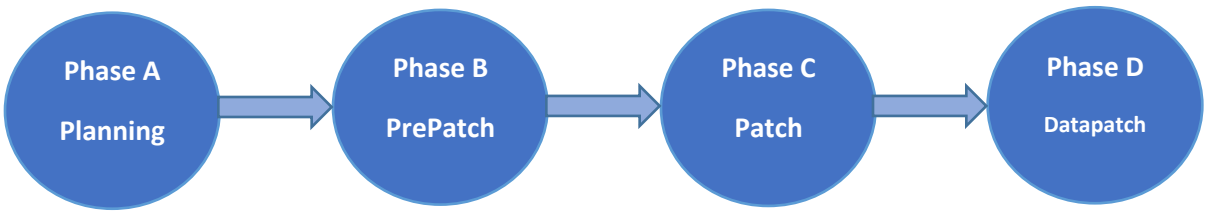
- Execute opatchauto resume on the server followed by other patching step at version and component level (JDK/OJVM) through script.

Author: Srivastava, Ayush

Section IV: Steps of Performing Patching through Ansible

Oracle Patching solution is divided logically under 4 phases which are mentioned below.

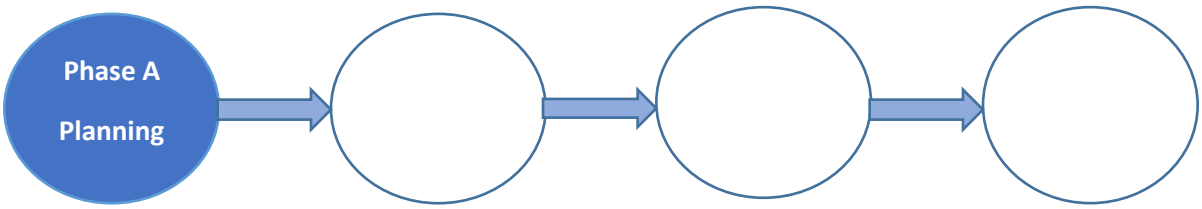
1) Phase A: Planning Stage
2) Phase B: Prepatch (Analyze) Stage
3) Phase C: Patch Stage
4) Phase D: Datapatch Stage



Phases during patching activity through Ansible

Phase A: Planning Phase (server consolidation)

- This step is needed to select and consolidate the servers for Patching through Ansible.



Phase A of patching activity through Ansible

Actions during Phase A (Planning)	
Downtime needed	No
Job Template required	None
Goal achieved during this phase	Add servers in inventory groups, which are needed patching, in Ansible Tower inventory.
Logfile location (located at Target server)	Not applicable

- To decrease the downtime of patching activity we are leveraging the parallelism feature of Ansible.

- In this phase we would consolidate multiple servers which have common downtime or falls in similar 3-4 hours activity window time frame.

- Ansible Tower has feature of Inventory Group which is logical way of grouping the servers under single logical unit. Hence in this phase we would consider a approach to group the servers.

- For explanation of planning phase we are taking below example, where 4 servers or 2 clusters are need to be patched simultaneously.

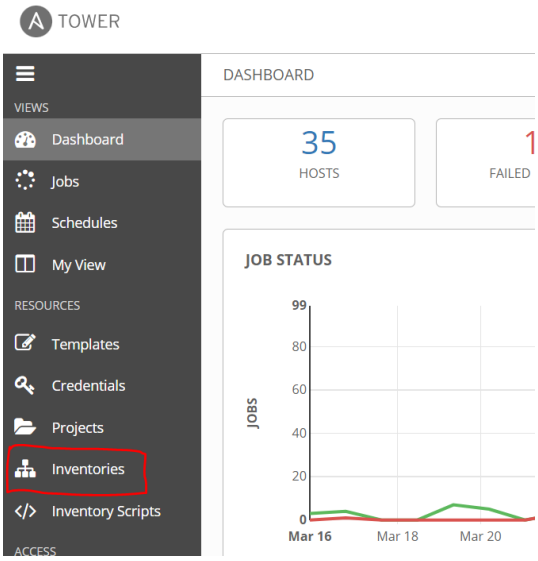
  Example:
  iedub26db03vcn1/2
  iedub24db01vcn1/2

Approach While consolidating the servers in inventory group of Ansible Tower	
WRONG APPROACH	**RIGHT APPROACH**
Both nodes should not be part of same inventory group	Club one node of all RAC servers in one inventory group and other node of all servers as different inventory group
GROUP 1 iedub26db03vcn1 iedub26db03vcn2 **GROUP 2** iedub24db01vcn1 iedub24db01vcn2	**GROUP 1** iedub26db03vcn1 iedub24db01vcn1 **GROUP 2** iedub26db03vcn2 iedub24db01vcn2

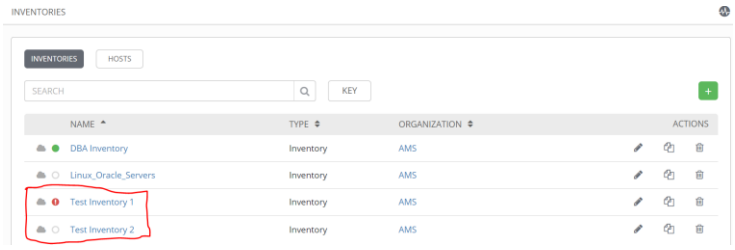
Operational Steps of Phase A (Planning):-

1. Login to Ansible Tower through OKTA.

2. Go to Inventory location to update the servers required for patching as shown below.



3. As shown below two new inventory groups are created (Test Inventory 1 and Test Inventory 2), which are needed to consolidate the database servers under logical single unit, where the Ansible script would run. It is to note during any job execution phase (which would be again highlighted in later part of document) we need to provide the inventory group name and the Ansible job would run in all the servers which are part of specific inventory group.



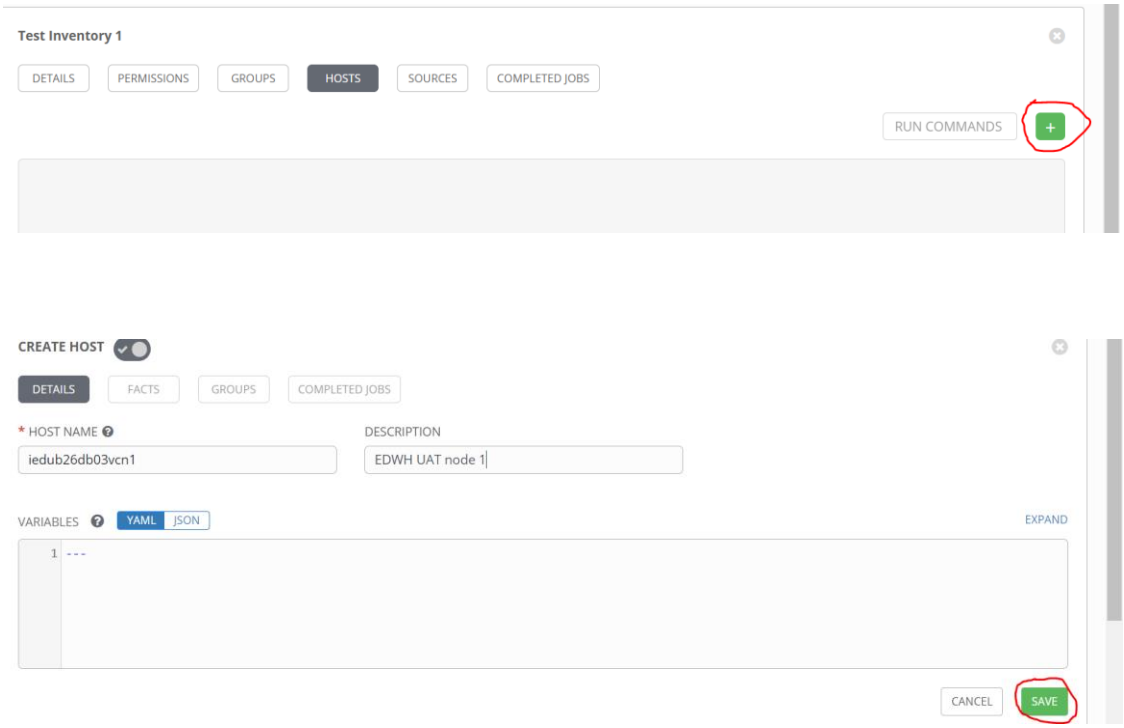
4. Add all node 1 of servers in the inventory group 1 i.e. Test Inventory 1 and all node 2 of servers in the inventory group 2 i.e. Test inventory 2.

   Example: - List of server required to patched are iedub26db03vcn1/2 and iedub24db01vcn1/2 however please note the way servers are grouped/consolidated under inventory group.

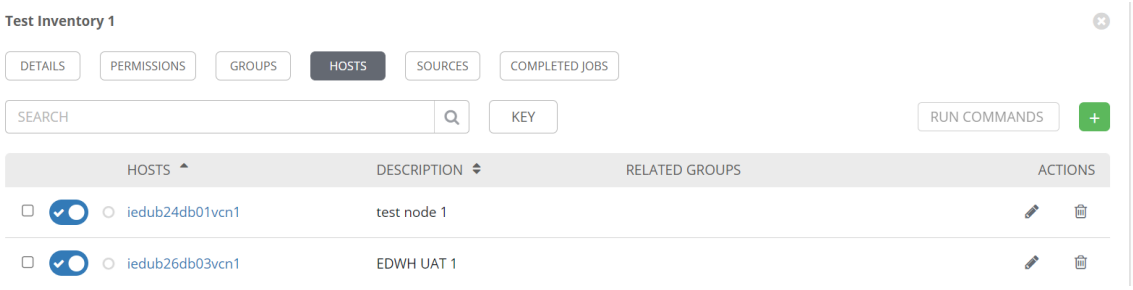
   iedub26db03vcn1 + iedub24db01vcn1 in one group (Test inventory 1) → **Right**
   iedub26db03vcn2 + iedub24db01vcn2 in second group (Test inventory 2) → **Right**

   consolidating both nodes of single cluster in one inventory group is wrong i.e.
   iedub26db03vcn1 + iedub26db03vcn2 in one group → **Wrong**
   OR
   iedub24db01vcn1 + iedub24db01vcn2 in second group → **Wrong**

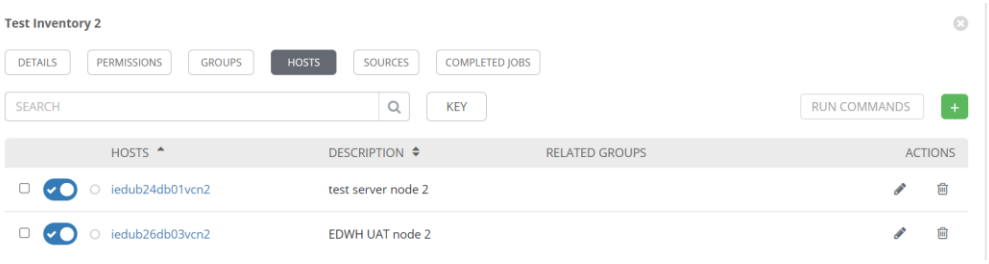
Inventory group name shown here is for representational purpose and name could also be different during activity time.



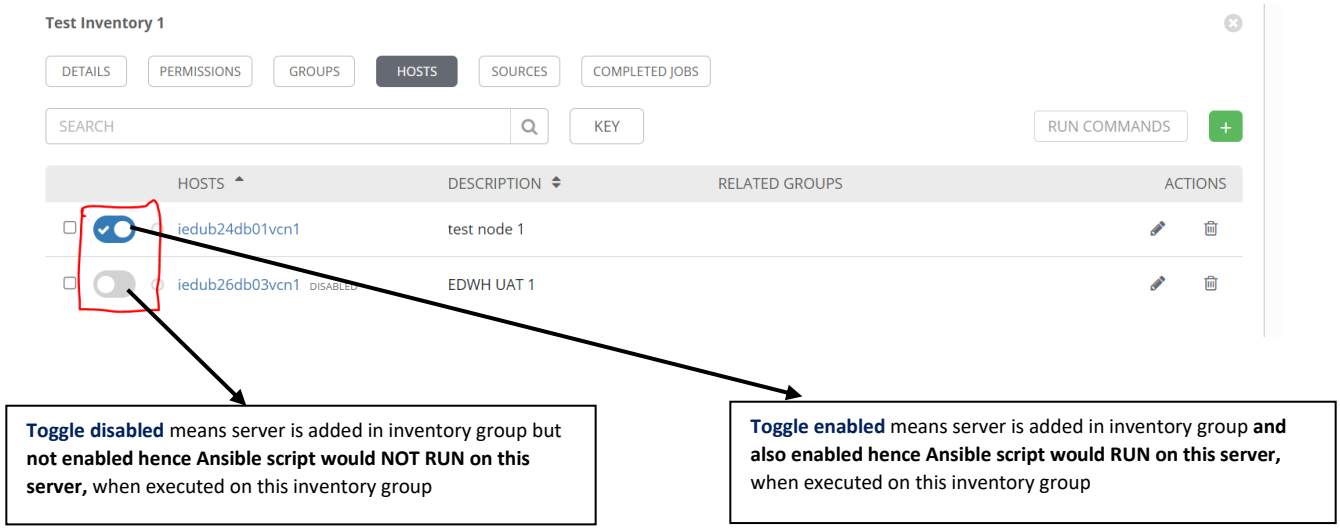
5. Once the servers (node 1 of both clusters) are added in the Inventory group Test Inventory 1 it would be reflected as below.



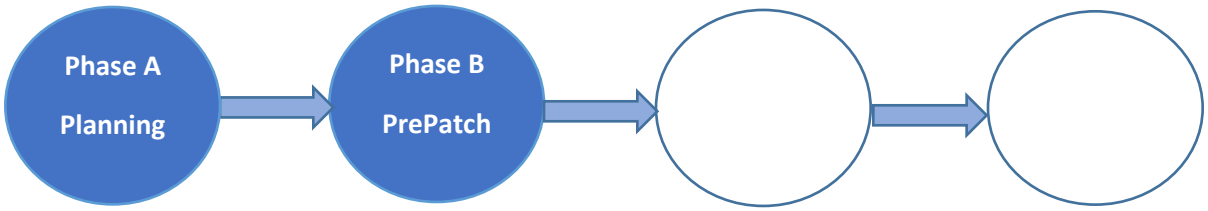
6. Similar information is reflected when servers (node 1 of both clusters) are added in the Inventory group Test Inventory 2



7. Below example shows even the servers are part of one inventory group how they can be toggled between disabled and enabled status hence Ansible script would run accordingly.



Phase B: Prepatch Phase (PSU analyze)



Phase B of patching activity through Ansible

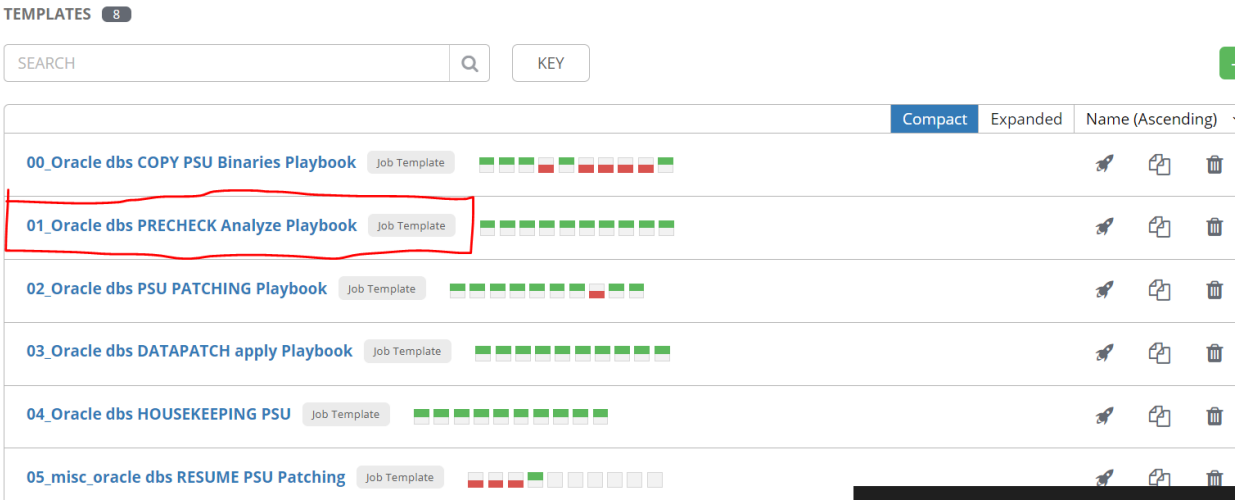
Actions during Phase B (Prepatch / Analyze)	
Downtime needed	No
Job Template required	ORCL DBS Precheck Analyze Playbook
Goals achieved during this phase	a. Take snapshot of server. b. Analyze database processes under server. c. Copy PSU (RU + OJVM + JDK) software from central server to target servers. d. Replace Opatch directory from RDBMS/GRID home. e. Execute analyze for different patches
Logfile location (located at Target server)	a. /tmp/ansible_psu/ospatching_snapshot_<<Date>>.log b. /tmp/ansible_psu/prepatching_ansible_<<server>>_<<Date>>.log

Operational Steps of Phase B (Precheck):-

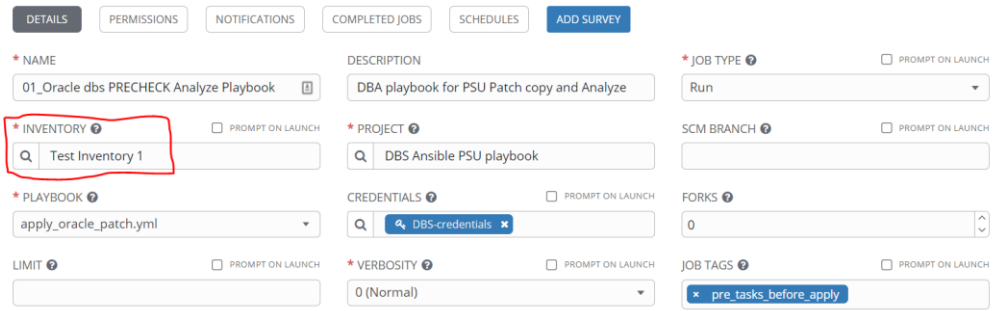
a) The given phase consider Phase A of planning and grouping the database servers in Ansible inventory is already completed.

b) This phase requires to be executed on both nodes of cluster hence to be executed at inventory group level (i.e. considering this example one for Test Inventory 1 and another same execution for Test Inventory 2)

c) Click on the below Analyze Job template

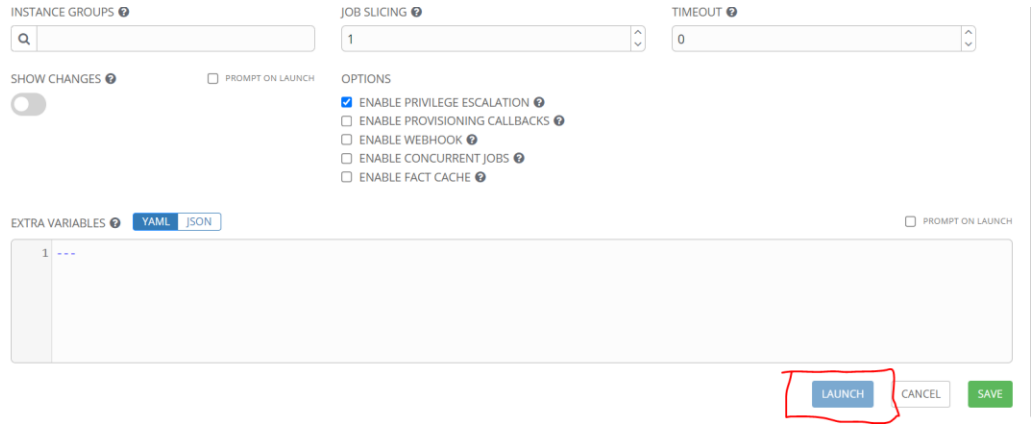


d) The below highlighted screenshot shows the Ansible inventory group on which the Preanalyze job would execute (in this case it is Test Inventory 1). In case the Inventory group need to be changed then click on highlighted field and select another inventory and save the Job template.

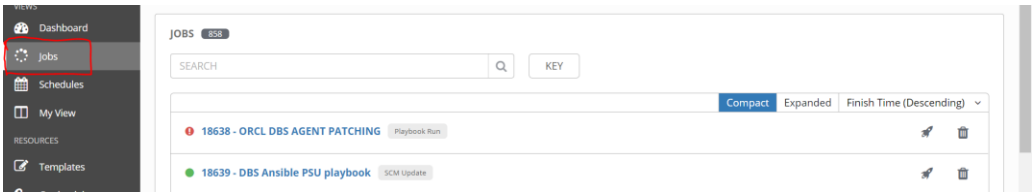


e) Once the required inventory group if changed, make sure to save it before launching the Job template as below.

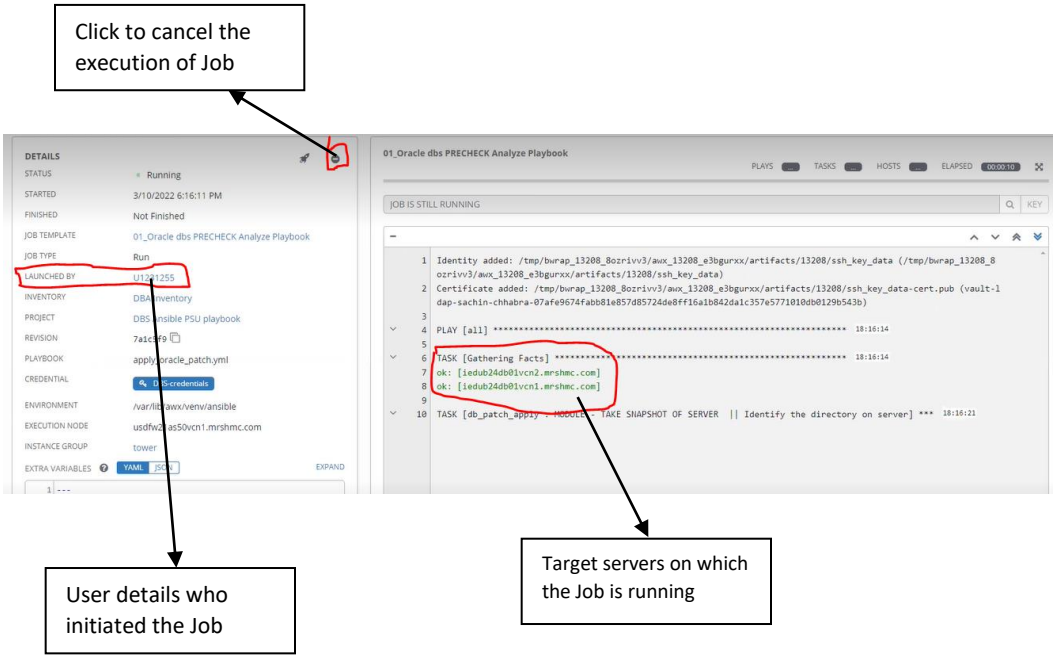
Author: Srivastava, Ayush



f) Once the Job, automatically the page of job execution would be visible else click on highlighted Job link as shown below which current as well as past job status has.



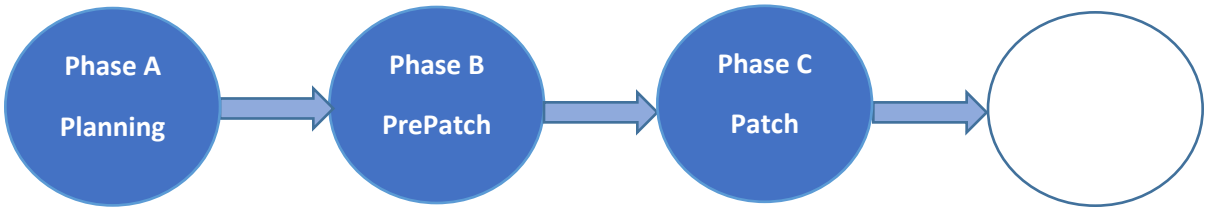
g) Once we are in Job execution phase we can find many useful information as marked below.



h) The monitoring logs would not only be present on Ansible server but Job can also be monitor on target server from → /tmp/ansible_psu location.

Author: Srivastava, Ayush

Phase C: Patch Phase

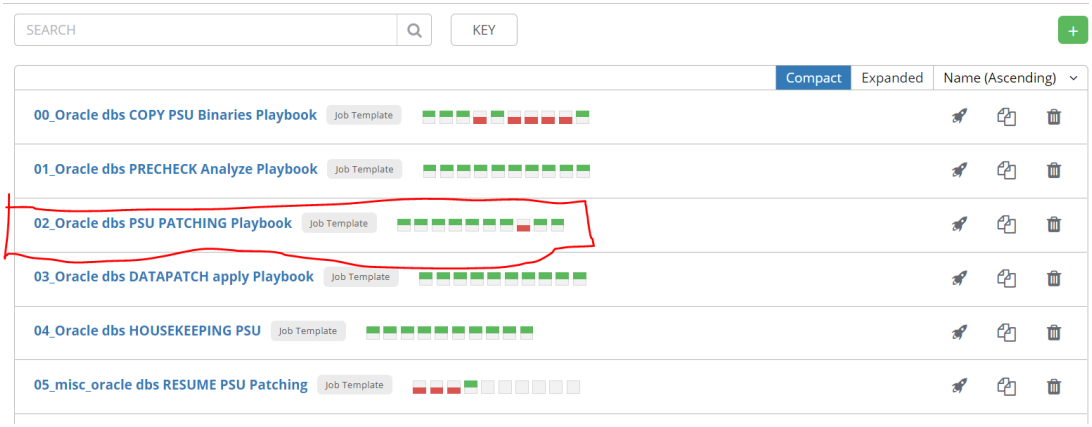


Phase C of patching activity through Ansible

Actions during Phase C (Patch)	
Downtime needed	YES (node level)
Job Template required	**Oracle dbs PSU PATCHING Playbook** **Oracle dbs HOUSEKEEPING PSU**
Goals achieved during this phase	**Oracle dbs PSU PATCHING Playbook** a. Take snapshot of server. b. Set OEM blackout at node level c. Shutdown and start database services as needed d. Patch binaries with RU/OJVM/JDK on the server. **Oracle dbs HOUSEKEEPING PSU** a. Remove Blackout from the OEM at node level b. Remove PSu software directory copied from central server.
Logfile location (located at Target server)	a. /tmp/ansible_psu/ospatching_snapshot_<<Date>>.log b. /tmp/ansible_psu/patching_ansible_<<server>>_<<Date>>.log

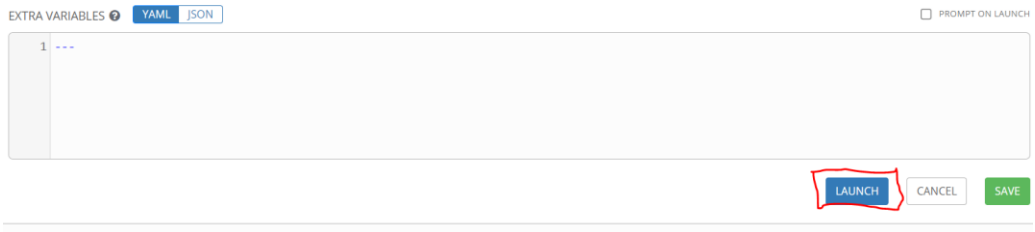
Operational Steps of Phase C (Patch):-

a) The given phase consider Phase A and Phase B (i.e. Planning and Prepatch stage) is already completed.
b) This Phase requires downtime at node level.
c) This phase requires to be executed at each inventory group level (i.e. considering this example one for Test Inventory 1 and another same execution for Test Inventory 2).
d) Click on the highlighted Job template to start execution of patching process.



e) Verify the inventory group mentioned in the Job template, which is same as discussed in Precheck Phase (Phase B) above.

f) Once the inventory group is verified, initiate the Job



g) As soon as job is initiated, monitor the Job execution, either window comes automatically after initiating job or navigate to Job tab and click on the latest job.

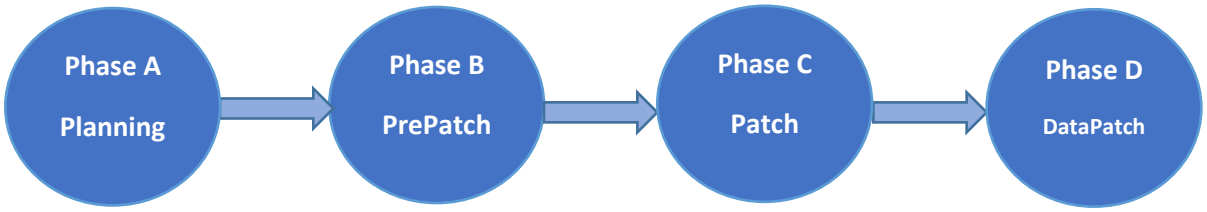
h) Blackout would be set at node level.

i) Patching would start at this point using updated OPatch directory and databases would be started and stopped multiple times.

j) Monitoring of activity can also be done at local server level by logging to target server and checking log under /tmp/ansible_psu location.

k) Once the patching is completed on the node with logfile at local file system showing no error, we can execute Oracle dbs HOUSEKEEPING PSU job which would remove the PSU software and OEM Blackout from the node

Phase D: DataPatch Phase



Phase D of patching activity through Ansible

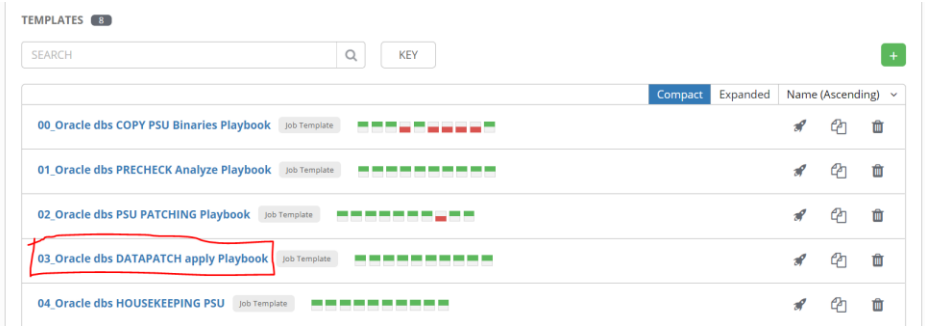
Actions during Phase D (Datapatch)	
Downtime needed	NO
Job Template required	**Oracle dbs DATAPATCH apply Playbook**
Goals achieved during this phase	a. Databases should already be running on the server b. Queries DBA registry and DBA SQL Patch table as Pre-task c. Executes datapatch verbose on each database d. Queries DBA registry and DBA SQL Patch table as Post-task
Logfile location (located at Target server)	e. /tmp/ansible_psu/datapatch_ansible_<<server>>_<<Date>>.log

Operational Steps of Phase D (Datapatch):-

a) The given phase consider Phase A, Phase B and Phase C (i.e. Planning and Prepatch and Patch stage) is already completed for all nodes.

b) This phase requires to be executed ONLY at one node of cluster databases i.e. considering this example Node 1 of databases are grouped in -> Test Inventory 1 and Node 2 of databases are grouped in -> Test Inventory 2. Hence in this case we only need to run datapatch job template for one inventory group only (either Test Inventory 1 OR Test Inventory 2).

c) Click on the highlighted Job template to start execution of datapatch process.



Author: Srivastava, Ayush

d) Verify the inventory group mentioned in the Job template, which is same as discussed in Precheck Phase (Phase B) above.

e) Once the inventory group is verified, initiate the Job.

f) Monitoring of activity can through Ansible and also at local server level by logging to target server and checking log under /tmp/ansible_psu location.

Section V: Miscellaneous

A. Frequently Asked Questions (FAQs):

1) Would it be fine if we run Patching Job from Ansible directly and skip Prepatch (or Analyze) Job from Ansible Tower?

Ans: *No, Precheck/Analyze job is also essential part of database patch process and should be executed before running Patch Job template. As Precheck/Analyze Job requires no downtime hence can be done 1-2 days before the main patching activity day.*

2) Would it be fine If I don't run Precheck/Analyze Job from Ansible Tower but executed manually but the Patching I want to do through Ansible Tower?

Ans: *Though the Analyze job in Precheck module of Ansible Tower executes similar steps what is required to run manually however Precheck module also does many other steps like copying PSU software to target server / replace OPatch and many other hence it is advisable to follow complete cycle from Ansible Tower itself.*

3) Precheck Analyze Job is taking too much time from Ansible Tower to execute is there any way to reduce it?

Ans: *Precheck Analyze Job would take similar time what a manual analyze of Patches should take on specific server. As Ansible is only orchestration tool and the activity runs on server itself hence the time taken by any job depends on target server itself.*

However one point to notice of the Total time most time consuming part in Precheck Analyze Job is the part where PSU software are getting copied from central server to target server hence to reduce Precheck Analyze time we can use ***ORCL DBS Copy PSU Binaries Playbook*** *Template which would only copy the PSU software to target servers. Hence once PSU software are copied now when Precheck Analyze Job would be executed, it would skip copying the PSU software again and would be faster in job execution.*

4) What is a criteria for the database server to qualify for patching from Ansible Tower?

Ans: *The database server should be running 19c Grid in Linux server, though the RDBMS version could be 19c or 12cR2 (12.2) or both, which is also mentioned in initial of this document.*

5) What is the average time to complete a precheck analyze job on a server from Ansible Tower?

Ans: *The time though varies from server to server however even the complex server, which contains multiple Oracle homes (i.e. 4 or 5) and databases (5 + atleast) should not take more than 45 minus at most, considering the PSU software is already pushed to that server. If any server is taking more than hour in Precheck Analyze job then that server should not be clubbed with other servers, as it would slow complete patch cycle in parallel, and this slow server should be either patched manually or independently from Ansible Tower.*

6) Let's example we have 2 node cluster and user has already approved complete downtime for both nodes, Can I run patching on both nodes of RAC in single Job in parallel ?

Ans: *No the nodes of the RAC server should not be patched in single Job, the solution is designed to avoid complete outage hence should be done one node at time.*

7) For any reason I only want to execute only specific Patch on specific Home, say JDK only, and don't want other patches like OJVM, RU to be patched, would it be possible from Ansible?

Ans: *Currently the solution is designed where the script would automatically identify all 19c and 12cR2 Oracle homes and patch it with all patches hence selective patch application is not supported yet.*

8) When we say we can leverage the Ansible feature of parallelism by consolidating the servers for patching, does it mean we can club any set of servers together and run through Ansible would that impact the patching time?

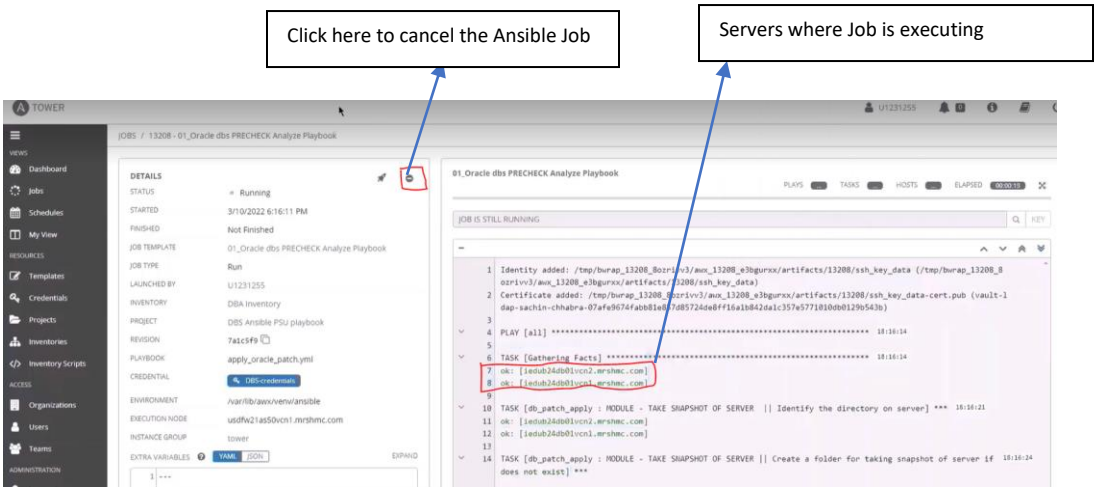
Ans: *Ansible is only an orchestration tool and the job actually runs and take resource from local servers, hence any slowness on target server would impact patching activity time. Example among 3 servers A, B, C servers one server C is very slow and upto a point while A and B complete the patch step in 20 mins the C server take 60 minutes hence in this case complete patch activity would be stretched and would take total time of 60 minutes for all servers hence server selecting in grouping plays important role.*

9) Example, there are 4 instances of databases are running on 2 node cluster however among 4 only 3 are RAC while one database instance is single node database what should I do in such case?

Ans: *Execute the datapatch job from the server which contain all the database instances i.e. RAC and Non RAC or execute the datapatch specifically only for remaining Non RAC database which is left.*

10) While starting the Ansible Job I found out something is missed or wrong, How can I cancel the Job from Ansible Tower side now?

Ans: *It is always advisable to double check the inventory etc on which the job need to be executed, however still in case of any miss / error we need to click on below button to cancel the Ansible job this would stop the Ansbile Job execution for all the servers.*



11) While patching the GRID/RDBMS binary I got some error which is not related to Ansible, is there way Ansible could help to resolve it?

Ans: *Ansible is only the orchestration tool and patching related error should be handles similarly as handled during manual patching, though as with experience a document is maintained which has listed issues and its solutions which could help in troubleshooting issue as mentioned in below section i.e. General Document of Technical issues during Patching and their Resolutions*

B. General Document of Technical issues during Patching and their Resolutions

Patching on Oracle binary could fail due to multiple reasons however many of the issues are consolidated along with the solutions which is kept at below location.

AMSI Project UNITY - DBA KT TRACKER - Problem Solution Guide - All Documents (sharepoint.com)