ORACLE

Patch Your Exadata Infrastructure Using Fleet Maintenance

Enhance Security Posture and Lower OpEx

Sravanth Mouli
Senior Product Manager

Exadata infrastructure patching challenges

- Exadata patching is complex due to its interconnected components
- Manual patching with homegrown scripts are :
  - Error-prone
  - Time-consuming
  - Hard to scale
- Requires dedicated resources leading to higher cost
- Increased risk of breaches due to delayed and improper patching activity
- Separate tools for database/grid and Exadata patching adds complexity and cost

Introducing Exadata Infrastructure patching using Fleet Maintenance

Unified Management

Unified patching process for databases, grid, and Exadata Infrastructure.

Simplified Patching

Gold image based patching process providing flexibility and control

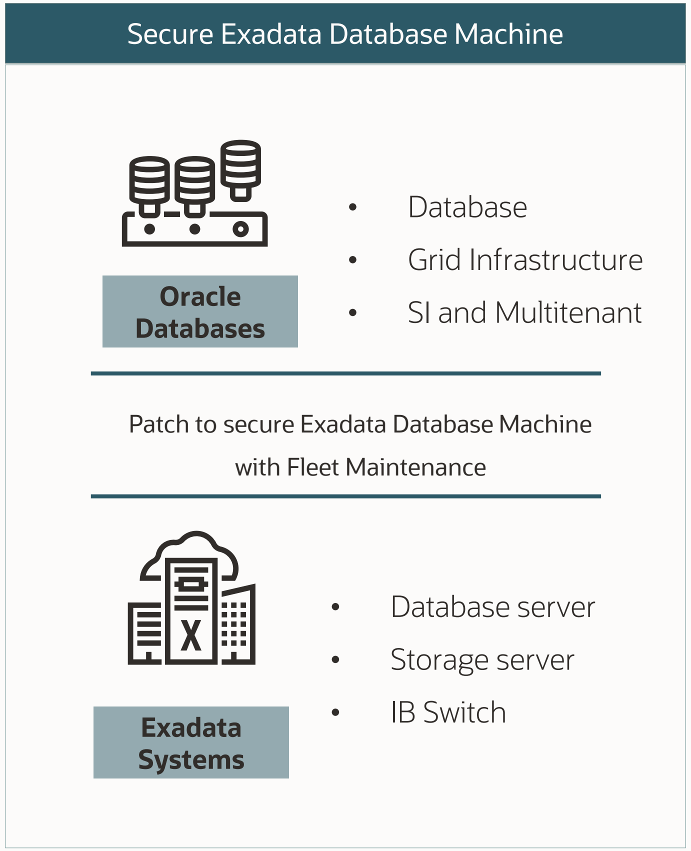
Automate for efficiency

Automated patching for cost savings and operational efficient

Streamline quarterly patching to strengthen security

Unified Management

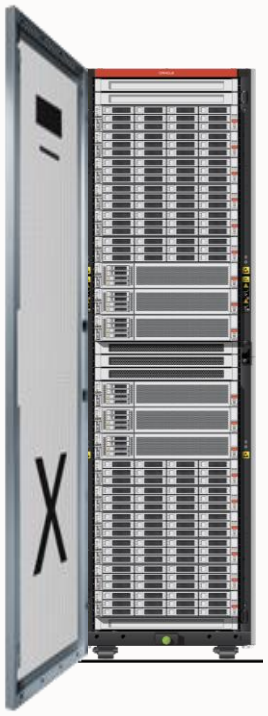
Streamlining patching with Unified Management

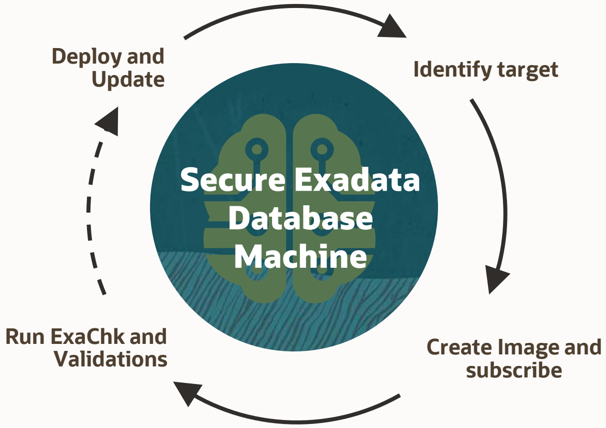


- **Reduce complexity** and administrative overhead by consolidating patching processes into a single platform
- Streamline patching cycles, minimize downtime, and improve overall **operational efficiency**
- Ensure consistent patch levels across the entire Exadata environment, **reducing security vulnerabilities**
- **Optimize resource allocation** and reduce the need for specialized skills by using a unified tool
- Minimize the risk of **human error and inconsistencies** associated with managing patches using multiple tools

Fleet-wide patching of Exadata stack

Using Enterprise Manager Fleet Maintenance





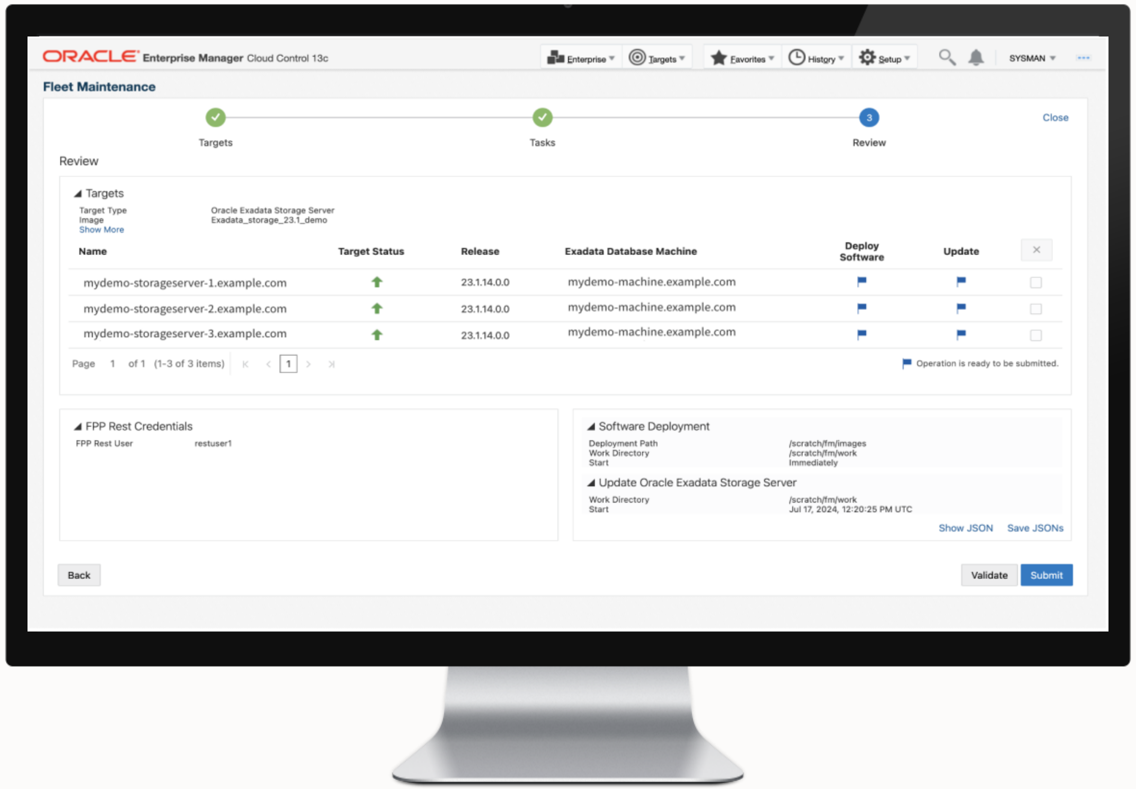
- Monitor, manage secure complete Exadata stack including database, grid and Exadata components
- Exadata Infrastructure patching support for
  - Database server
  - KVM guest, Xen DomU
  - Storage server
  - InfiniBand switch
- Schedule patching windows and customize deployments to fit organizational needs
- Patching support through API, EMCLI and UI

Simplified Patching

Exadata Infrastructure patching

Patch components of choice using Fleet Maintenance

- Gold image based in-place patching solution
- Deploy and update components at scale for a faster patching cycle
- Validate update and rollback to minimize failures
- Pre-check mode to minimise potential failures
- Customise patching with optional parameters
- Optional rollback to last known good state if necessary
- Integrate with automation tools like Ansible, Chef, Terraform using REST APIs



Solution Overview

One time Setup/ First Run

Setup Infrastructure and discover targets in EM

Assess and identify relevant patches for each target type

Create Image(s) for each target type.

Subscribe targets to relevant image to achieve automation at scale.

Deploy image to subscribed targets

Update Exadata Infrastructure

Every Quarter (Recommended)

Refresh Image with latest patch. Mark latest version current

Deploy image to subscribed targets

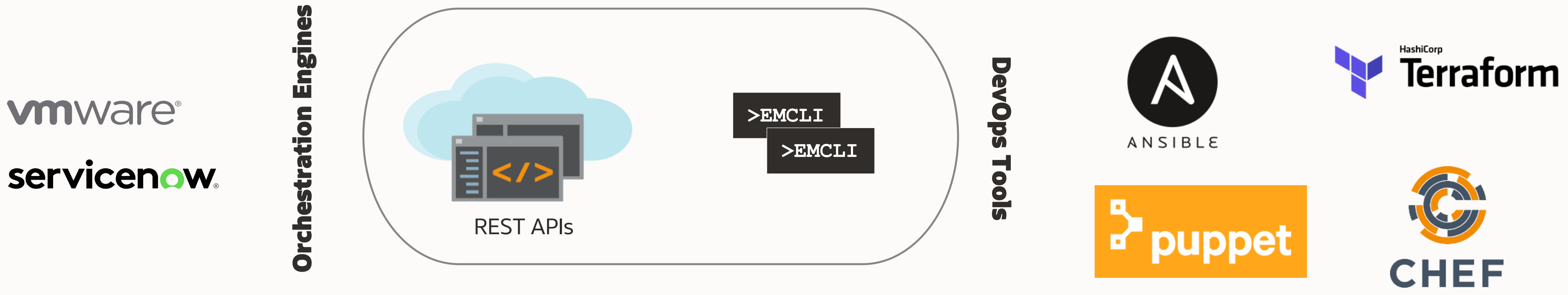
Update Exadata Infrastructure

Exadata Administrator

Exadata Administrator/Patching Operator

Automate for Efficiency

Fleet Maintenance automation



- Integration with broad ecosystem with tools like Ansible, Chef, Terraform
- Integrate with 3rd party Orchestration Engines
- Easy to build, run and manage

Operational efficiency with automation

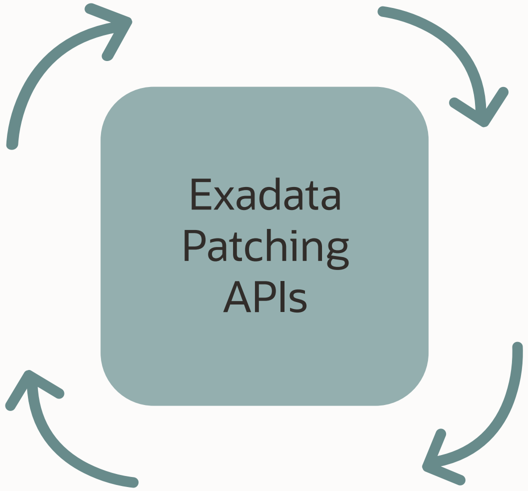
Automate with API/CLI to reduce cost and human error flow

Infrastructure setup (1)

- Discover and Register targets in EM
- Automated FPP server and client setup

Create image & Subscribe targets (2)

- List Images
- Create Image
- Update an Image
- Subscribe targets to an Image



Deploy and Validate (3)

- Deploy Images
- Validation mode to avoid failures
- Include custom options

Update and Rollback (4)

- Update Exadata components
- Rollback validate
- Rollback

REST API flow

Create an Image → Subscribe targets → Deploy Image to target → Update

Create an image for storage nodes

URL - https://<EM URL>:<EM PORT>/em/api/goldImages

```
{
"imageName":  "Exadata_Storage_23.1_demo",
"versionName":  "Exadata_storage_23.1.15_demo",
"imageVersion":  "23.1.15.0",
"targetType":  "oracle_exadata",
"imageLocation":  "/u01/staging/db_2126",
"workingDirectory":  "/tmp",
"referenceHostCredential":  ”DEMO:SYSMAN",
"imageDescription":  "Gold image for exadata storage node ",
"storageFile":  ”p1234567_66778_Linux-x86-64.zip"
}
```

REST API flow

Create an Image → Subscribe targets → Deploy Image to target → Update

Create an image for storage nodes

URL -
```
https://<EM URL>:<EM PORT>/em/api/goldImages
```

```
{
"imageName":  "Exadata_Storage_23.1_demo",
"versionName":  "Exadata_storage_23.1.15_demo",
"imageVersion":  "23.1.15.0",
"targetType":  "oracle_exadata",
"imageLocation":  "/u01/staging/db_2126",
"workingDirectory":  "/tmp",
"referenceHostCredential":  ”DEMO:SYSMAN",
"imageDescription":  "Gold image for exadata storage node ",
"storageFile":  ”p1234567_66778_Linux-x86-64.zip"
}
```

Subscribe targets to an Image

URL -
```
https://<EM URL>:<EM PORT>/em/api/goldImages/{ImageID}/actions/subscribeTargets
```

Body

```
{
"targetList" : [”mystorage-demo1.us.oracle.com”,
                ”mystorage-demo2.us.oracle.com”,
                ”mystorage-demo3.us.oracle.com”],
"targetType" : "oracle_exadata"
}
```

REST API flow

Create an Image → Subscribe targets → Deploy Image to target → Update

Deploy Image

URL : `https://<EM URL>:<EM PORT>/em/api/fleetOperations`

Body

```
{
”deployExadata”:{
        "name": "Deploy Storage node patch",
        "targetList":[”mystorage-demo1.us.oracle.com”,
                      “mystorage-demo2.us.oracle.com",
                      “mystorage-demo3.us.oracle.com”],
        "targetType":” oracle_exadata",
        "patchExtractLocation":"/usr/images",
        "workingDirectory":"/usr/01/fleet_maintenace” }
}
```


REST API flow

Create an Image → Subscribe targets → Deploy Image to target → Update

Deploy Image

URL : `https://<EM URL>:<EM PORT>/em/api/fleetOperations`

Body

```
{
”deployExadata”:{
        "name": "Deploy Storage node patch",
        "targetList":[”mystorage-demo1.us.oracle.com”,
                      “mystorage-demo2.us.oracle.com",
                      “mystorage-demo3.us.oracle.com”],
        "targetType":” oracle_exadata",
        "patchExtractLocation":"/usr/images",
        "workingDirectory":"/usr/01/fleet_maintenace” }
}
```

Update Targets

URL : `https://<EM URL>:<EM PORT>/em/api/fleetOperations`

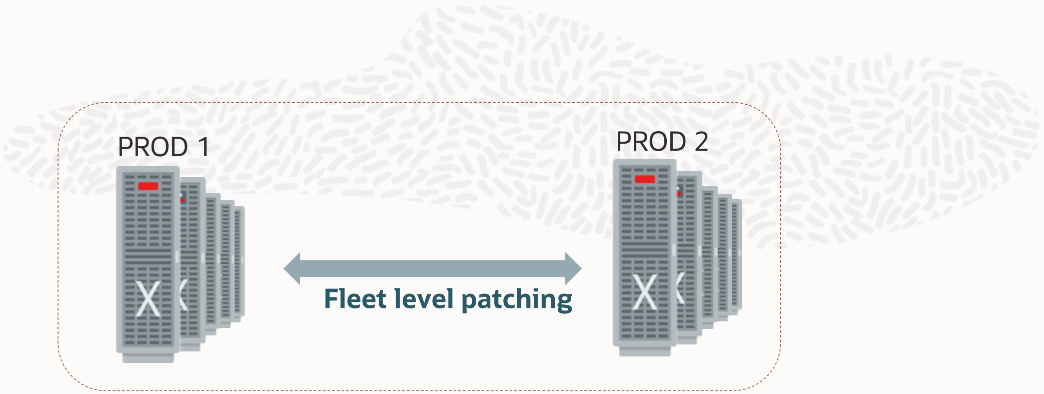
Body

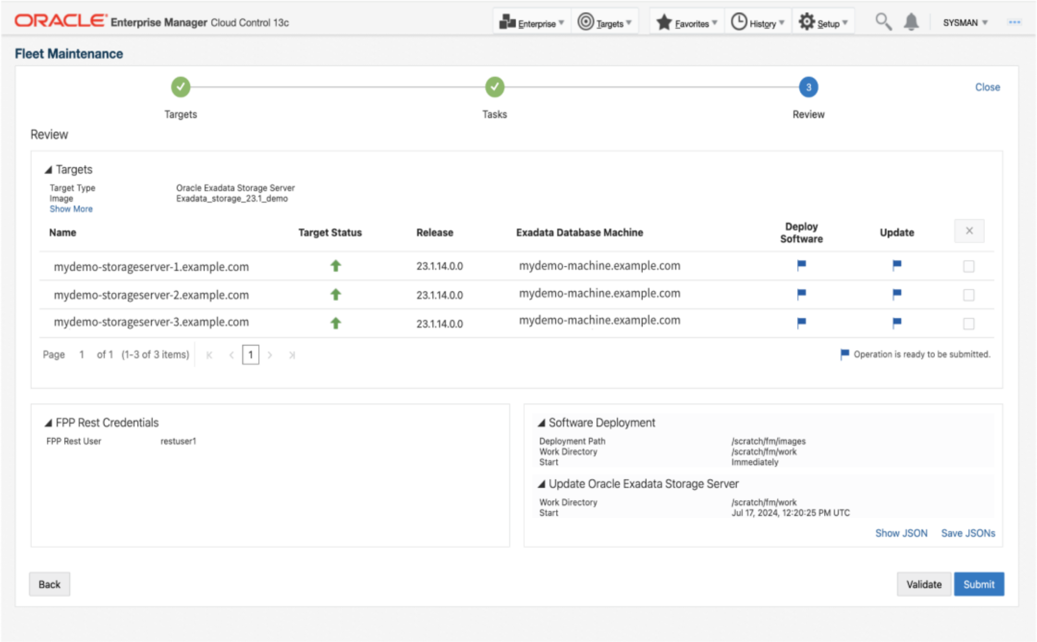
```
{
    "updateExadata":{
        "name":"Update Exadata cell patch",
        "targetList":[”mystorage-demo1.us.oracle.com”,
                      “mystorage-demo2.us.oracle.com",
                      “mystorage-demo3.us.oracle.com”],
        "targetType":"oracle_exadata ",
        "workingDirectory":"usr/01/fleet_maintenace” }
}
```


Summary

Summary

- Exadata software patching for
  - Bare Metal database servers
  - KVM and Xen DomU guests
  - Storage servers
  - InfiniBand network switches
- Patching features
  - **Fleet** level patching of multiple components
  - Rollback to previous version, if required
  - Schedule deploy and update operations to align with preferred maintenance window
- Patching support through API, EMCLI and UI
- **License requirement: Exadata Management pack**





Q&A
Learn More

Web: oracle.com/enterprisemanager

Videos: youtube.com/OracleEnterpriseMgr

Blogs: blogs.oracle.com/observability

Docs: http://docs.oracle.com/en/enterprise-manager

Try it now

Hands-on-labs

Oracle Cloud Free Tier

Always Free

Services you can use for unlimited time

+

30-Day Free Trial

Free credits you can use for more services

www.oracle.com/cloud/free

