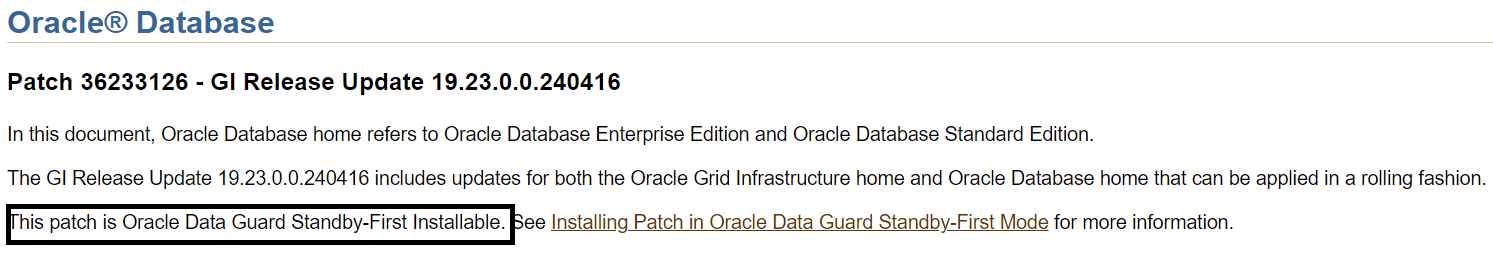
Zero or Near-Zero Downtime Oracle Maintenance Options :

1. Switchover to any of the standby:

We have our primary instance available in **ca-central-1a (pws1e-db-r12)** with two standbys – **pws1e-db2-r12(DR)** in ca-central-1b and **pws1e-db-r12-dr** (DR1) in ca-central-1a but in us-east-1 AZ.

Having two standby databases in place with Oracle Data Guard provides an excellent foundation for achieving near-zero downtime during patching or maintenance activities. Here’s how you can utilize them effectively:

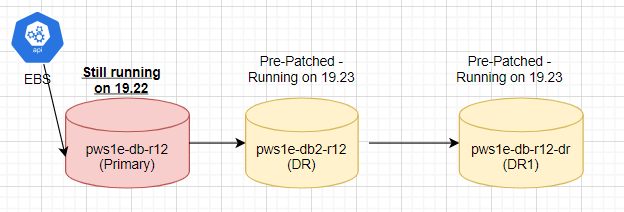


**Note** : This approach will take 2 short downtimes due to double switchovers that we have to do between the primary and one of the standby in this case.

**Preparing for Near-Zero Downtime Patching with Oracle Data Guard**

1. **Pre-Patching of Standbys:**

We can apply patches to both standby databases (one at a time) for pws1e-db2-r12 (DR) and pws1e-db-r12-dr (DR1) ahead of time without requiring any downtime. This approach works because RURs/PSU patches are typically designed to be applied to standby databases first, allowing them to be patched while the primary remains operational and at a different patch level.



1. **Verify Data Guard Configuration**:

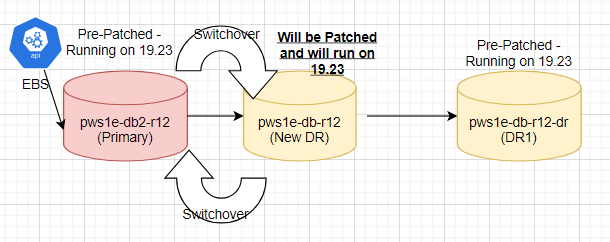
Ensure that the Data Guard configuration is correctly set up with synchronous or asynchronous replication between the primary and standby databases. Both standby databases should be in sync with the primary.

1. **Stop EBS application-related services**: (1st downtime starts from this point - Declare a planned maintenance window): This step is taken to prevent any new changes from being made to the database and to ensure consistency across both the application and database layers.
2. **Backups for Rollback purposes:**

* Take a snapshot at the EBS storage level or a quick incremental backup to obtain a consistent image of the system/database.
* I also recommend creating a Guaranteed Restore Point (GRP) for additional assurance. This GRP will capture the database state just before the switchover.
* We can also take a backup of ORACLE\_HOME as that’s where we have all binaries sitting and goes through the modifications.

1. **Perform a Switchover**:

Perform a controlled switchover from the primary database **ca-central-1a (pws1e-db-r12)** to **pws1e-db-r12-dr** **(DR1)** which is already patched. This involves using command lines or using DGMGRL utility. After the switchover, the former primary **ca-central-1a (pws1e-db-r12)** becomes a standby database, and the former standby **pws1e-db-r12-dr** **(DR1)** becomes the new primary.



1. **Start EBS application-related services (1st downtime ends here)**

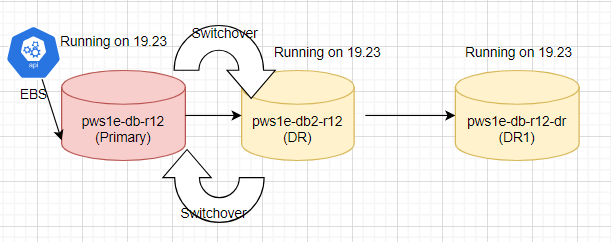
Now when the above switchover is completed, we have a new Primary database opened in READ WRITE mode to cater application requests, we can point all the application related strings to the new primary pws1e-db-r12-dr (DR1) and this will resume the application connectivity and availability. This ends the first downtime.

1. **Apply Patches to the New Standby - ca-central-1a (pws1e-db-r12) :**

With the new standby (**pws1e-db-r12)** now we can apply patches, updates, or perform maintenance tasks on it. This can be done without affecting the availability of the application. At this point we have all three databases patched with the same version/RU/PSU/CPU etc.

1. Run Datapatch from current Primary : Now as the final step we can run the datapatch from the current primary instance pws1e-db-r12-dr (DR1) and the changes will be automatically transported to both standbys.
2. **Stop EBS application-related services**: (2nd downtime starts from this point - Declare a planned maintenance window): This step is taken to prevent any new changes from being made to the database and to ensure consistency across both the application and database layers.
3. **Switchover Back to the Original Primary**:

Once the patching on the former primary (**pws1e-db-r12)** is complete and verified, perform the second switchover to revert the roles back. This involves using command lines or using DGMGRL utility. Perform a controlled switchover from the primary database **pws1e-db-r12-dr** **(DR1**) tocurrent standby **ca-central-1a (pws1e-db-r12)** **which** is now patched. This involves using command lines or using DGMGRL utility. After the switchover, the former primary **ca-central-1a (pws1e-db-r12)** becomes the primary again, and the former standby **pws1e-db-r12-dr** **(DR1)** enters the original state of being a standby.



1. **Start EBS application-related services**: (Downtime ends at this point – End of the planned maintenance window).

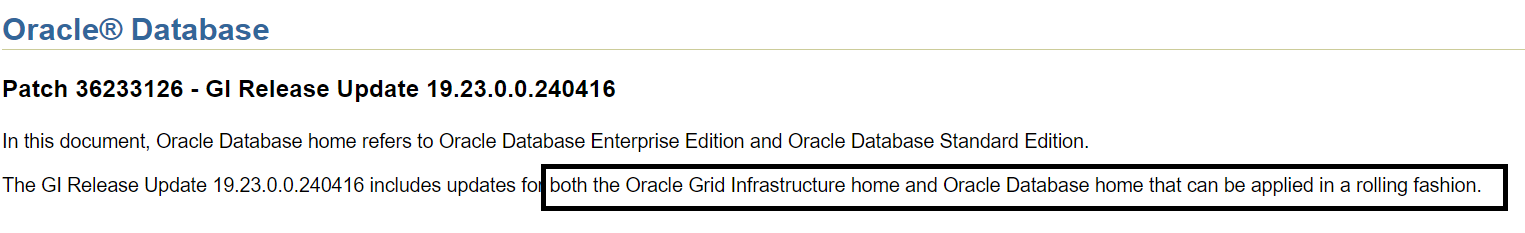
Note : I haven’t added any sanity or pre and post patching steps here.

1. **Migrating to Oracle Cloud Infrastructure (OCI Cloud)**

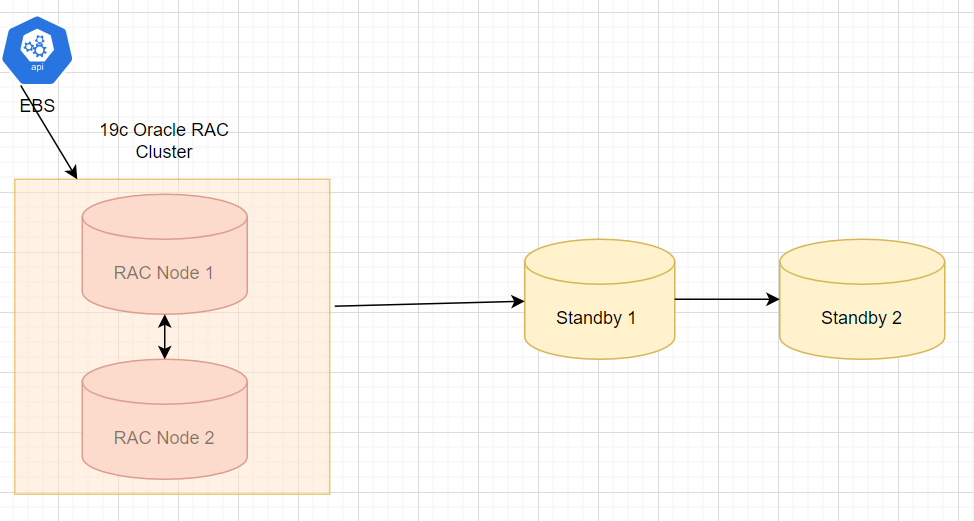
Using Oracle on EC2 Linux (Amazon Elastic Compute Cloud) offers flexibility and control over the Oracle deployments, but it comes with certain limitations compared to Oracle Cloud Infrastructure (OCI). In our environment where we have a single instance 19c hosted on an EC2 instance it becomes a single point of failure and at the same time lead to higher downtimes as rolling - patches approach is not possible in absence of Oracle Real Application Clusters (RAC).

Key advantages of using OCI over AWS :

* Oracle Workloads: OCI is designed to optimize Oracle Database and application workloads, offering seamless integration with Oracle technologies like Autonomous Database, Exadata Cloud Service, and Oracle E-Business Suite.
* With OCI we can use Autonomous Database on Exadata which is a fully managed service that eliminates the need for manual maintenance tasks such as patching, tuning, and backups. This being build on Exadata Cloud Service which leverages Oracle Exadata infrastructure, known for its performance, scalability, and reliability features. Alongside we can enjoy some of the great features of Exadata Smart Scan, In-Memory Processing, and encryption at rest and in transit.
* Using OCI allows us to use RAC which allows multiple Oracle Database instances to run on different nodes within a cluster, sharing a single database. This architecture provides high availability (HA) by eliminating single points of failure and improving scalability. Patching Oracle Real Application Clusters (RAC) in a rolling fashion allows you to apply patches or upgrades with minimal downtime to your Oracle database environment.



The advantage of a RAC rolling PSU patching is that it enables at least some instances of the RAC installation to be available during the scheduled outage required for patching. Only the RAC instance that is currently being patched needs to be brought down. The other instances can continue to remain available. This means that the impact on the application downtime required for such scheduled outages is further minimized.



Some of the high-level steps that we can do to perform the rolling PSU patching.

* **Backup Your Database:** Before starting any patching activities, ensure you have a recent and valid backup of your Oracle RAC database. This backup will be crucial in case of any unforeseen issues during the patching process.
* **Patch the First Node**: We can patch any of the one node to begin rolling PSU Patching. We should use ‘opatchauto’ as it has automated the patch application for the Oracle Grid Infrastructure (GI) home and the Oracle RAC database homes. It operates by querying existing configurations and automating the steps required for patching each Oracle RAC database home of same version and the GI home.

There will be no downtime as one of the Node is still up and running and can service all application requests.

