

# Running Oracle EBS in the cloud

**Andrejs Prokopjevs**  
**The Pythian Group**  
**Riga, Latvia**

## **Keywords:**

oracle ebs r12 apps cloud amazon aws

## **Introduction**

This presentation is based on a real-life experience migrating Oracle E-Business Suite R12.1 production to Amazon AWS, and additional proof-of-concept effort done getting various client systems upgraded to R12.2 and migrated to main cloud vendor platforms on the market. We are going to cover here various areas, like:

- Certification basics. Overview look into supported configurations.
- How to architect. Basic recommendations based on migration and 2+ year production runtime experience. We will mainly cover Amazon AWS use case.
- Advanced configurations outline.
- R12.2 and features / nuances coming with it.
- Microsoft Azure and Oracle Cloud review. Quick comparison outline of main alternative platforms.
- Cloud deployment automation and the most common scenario - auto-scaling.

This is a very client demanding topic and many are looking into cloud migration options and how they can optimize the cost comparing to the on-premise hardware hosting. And many are still misunderstanding the complexity of Oracle EBS stack being capable for cloud deployment.

## **Certification basics**

What is cloud? What benefits it provides us?

Cloud vendor platform services can be classified in three main areas: Infrastructure-as-a-service (IaaS), Software-as-a-service (SaaS) and Platform-as-a-service (PaaS).

You can deploy your IT systems completely in Public cloud, virtualize your own in-house Private cloud, or even deploy a Hybrid cloud architecture deciding what system or component is going to reside on each side.

There are certain benefits of utilizing the public cloud services. To mention few of them:

- You do not need to think about your hardware, it's maintenance, electricity, UPS or cooler in your DC room, hardware lifecycle and its replacement. This is your cloud service provider responsibility.
- Hardware pool - sharing and distribution of the load.
- Transparency. You run only what you need to run.
- "Pay per use" model.
- No termination fees.

But with all these benefits there are always certain risks:

- Security - your data and its processing is not happening in your "safe zone", like your enterprise local network.

- Design of your virtual private network, its access, and security of the data is the main key point for success.
- Encryption - can be covered on application layer, or encryption at rest option available for the storage layer.
- AWS console and privileged API access - Single Point of Disaster. Design of IAM and role based access is important. Root account access must be isolated and secured.
- Possible network latency with cloud service provider. Research for closest availability zone.

But what about Oracle EBS? What cloud deployment model it can support?

SaaS and PaaS are ready solutions, where you cover your own configuration, but you don't manage the software. IaaS - full infrastructure and software management as it would be within a local deployment. There is no PaaS / SaaS option available for Apps tiers, only IaaS where you are building your own Apps tier manually.

For DB tier, it's only IaaS again on AWS as Amazon RDS for Oracle is not supported. Microsoft Azure in general for Oracle deployments is only IaaS. Oracle Cloud has a differentiate feature coming with Database Cloud Service support for DB tier.

Licensing is primary covered within this document - <http://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf>. It covers Amazon EC2 / RDS, and Microsoft Azure, that is called as Authorized Cloud Environments.

Examples for running Database Standard Edition:

- Less than 4 vCPUs: counted as 1 socket = 1 processor licensed.
- More than 4 vCPUs: "closest vCPU multiple of 4" counted as number of sockets = number of processors licensed.
- Azure - replace 4 vCPUs with 2 Azure CPU cores within your calculations.

Oracle EBS database requires Database Enterprise Edition deployment: 2 vCPUs = 1 processor licensed.

Licensing can be processed using same "Pay per use" model, but only for Amazon RDS with SE1 / SE2. For other deployments, including Oracle EBS, it is BYOL (Bring Your Own License). It is possible to cover the licensing using Named User Plus licensing approach.

Any VMware public or private deployment, or any other cloud vendor not part of "Authorized Cloud Environments" is still not certified by this policy and you have to license the whole hardware pool.

You must do your own calculations to confirm that the selected cloud deployment approach is cost efficient for you and your organization.

## How to architect

We will cover several main technical architecture areas outlining basic concepts with some recommendations we have found worth to mention.

### 1. Network

Everything should start with a proper Network design. That includes:

- Network design based on regions and Availability Zones, configuration of subnets, network interfaces and routing tables.

- Security setup that is built from several layers. First layer is Network ACLs, that are configured for the whole Virtual Private Cloud. Then Security Groups, that is, basically, an additional set of firewall rules that can be assigned to specific one or group of EC2 instances. Also, there is an Internet Gateway which is responsible for public internet access of EC2 instance. Using this assignment, you are controlling what instances have and are exposed to the public internet access.
- Virtual Private Gateway - IPsec VPN tunnel connection with your on-premise network.

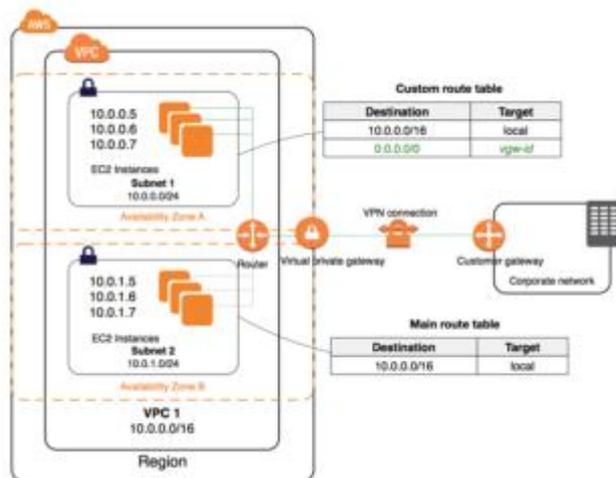


Illustration. 1: VPC architecture basics

Full Amazon VPC documentation reference can be found here: [http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC\\_Introduction.html](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Introduction.html)

IP address assignment has Private and Public IP address spaces.

Private IP addresses – dynamic. Can't be reserved and may change during instance restart.

Public IP addresses – dynamic, but a static IP address can be assigned via Elastic IP feature. Public IP traffic is routed through the public internet.

DNS service is provided by Amazon by default, and automatically managed by internal DHCP. Each region has its own sub-domain. By default, it contains a Private IP address in the name, for example: ec2-10-10-10-1.eu-central-1.compute.amazonaws.com. Route 53 cloud service is available to configure your own DNS, and assign the required host and domain names to your instances.

## 2. Instances

In AWS language, instance is the actual virtual machine. There are various instance types available, like: General purpose, Compute optimized, Memory optimized, Storage optimized, Accelerated Computing, etc. It's all about your "hardware" power and requirement - vCPU, T2 instances with CPU credits / burst, memory, GPU units, storage capabilities.

Full Amazon EC2 instance documentation reference can be found here: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html>

There is also another instance type separation, like: On-Demand which is default "pay per use" model, Spot Instances for quick short term workload, Reserved Instances and Dedicated Hosts for long term resource pre-allocation. This mainly affects pricing.

Instance can be built from an AMI (Amazon Machine Image) template located in Public or Private repository. You can always create your own AMI from an already built and running instance and store it in your personal private repository for future re-use. EC2 comes with RHEL, SLES, Windows Server official package selection, available with "pay per use" licensing model. Oracle Linux, for example, is available via Public AMI repository. Use for free, but don't forget about ULN licensing terms.

This is an example of most common instance configuration that can be used for Oracle EBS deployment.

Apps Tier: m3.xlarge, 4 vCPU, 15 GB Memory.

DB Tier: r3.4xlarge, 16 vCPU, 122 GB Memory.

Sometimes memory size is not the primary criteria for instance type selection. Each instance type has different IOPS and IO throughput limits. In the example above, 122 GB memory instance is selected not because of the memory requirements, but because of IOPS capabilities.

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
c1.xlarge		1,000	125	8,000
c3.xlarge		500	62.5	4,000
c3.2xlarge		1,000	125	8,000
c3.4xlarge		2,000	250	16,000
c4.large	Yes	500	62.5	4,000
c4.xlarge	Yes	750	93.75	6,000
c4.2xlarge	Yes	1,000	125	8,000
c4.4xlarge	Yes	2,000	250	16,000
c4.8xlarge	Yes	4,000	500	32,000
d2.xlarge	Yes	750	93.75	6,000
d2.2xlarge	Yes	1,000	125	8,000
d2.4xlarge	Yes	2,000	250	16,000
d2.8xlarge	Yes	4,000	500	32,000
g2.2xlarge		1,000	125	8,000
i2.xlarge		500	62.5	4,000
i2.2xlarge		1,000	125	8,000
i2.4xlarge		2,000	250	16,000

Illustration. 2: Table illustrating IOPS and IO throughput limits for different instance types.

### 3. Storage

There are several storage types available in EC2, and most standard is Amazon EBS (Elastic Block Store). This is standard SAN-like disk volume storage type. It can be mounted to only one EC2 instance at a time. There are several sub-types available, like gp2 and io1 as SSD, or st1 and sc1 as HDD. These different sub-types do have different throughput and IOPS limits, and cost.

Second type available there is EC2 Instance Store (Ephemeral), that is, basically, a local disk storage. Once your instance is stopped or restarted - all your data is lost. The best use case for this storage type is temporary storage.

Third type - Amazon EFS (Elastic File System). This is NAS analogue from Amazon – NFS based storage. But it's not yet available in all the regions.

Fourth type - Amazon S3. By default, used for storing AMIs and Amazon EBS disk snapshots. It's not a classical file system based storage, more object based.

Full Amazon EC2 storage documentation reference can be found here: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Storage.html>

This is an example of most common storage configuration that can be used for Oracle EBS deployment.

Apps Tier: Elastic Block Store – standard option for single tier approach. Elastic File System is preferable, if scale-out plans are there. If EFS is not available in your region yet, make your own "NAS server" instance, mount EC2 EBS disks, and export them via NFS. “gp2” standard type is absolutely enough, which is based on IO credits / burst. Cheap HDD sub-types should be considered only for secondary data access mounts with low IO workload, like interface and concurrent log/out data.

DB Tier: SSD based Elastic Block Store with or without ASM. No universal recommendation on IOPS / throughput. You MUST test, benchmark, and evaluate your own system performance per workload requirements.

Database Smart Flash Cache feature can greatly improve your database performance. Configure it on Instance Store (Ephemeral) disks.

```
SQL> show parameter db_flash_cache

NAME                                     TYPE          VALUE
-----
db_flash_cache_file                     string        /dev/xvda1
db_flash_cache_size                     big integer   200G
SQL>
```

4. IOPS

This is the most difficult area for architects to understand. IOPS / Throughput depends not only on EC2 Instance or Storage volume type, but also on the size of the storage volume. Basically, there are 3 tables to align to get the clear numbers of your target IO sub-system being designed.

API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS*/Volume	10,000	20,000	500	250
Max. Throughput/Volume†	160 MiB/s	320 MiB/s	500 MiB/s	250 MiB/s
Max. IOPS/Instance	65,000	65,000	65,000	65,000
Max. Throughput/Instance	1,250 MiB/s	1,250 MiB/s	1,250 MiB/s	1,250 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

Illustration. 3: Table illustrating IOPS and IO throughput limits for different storage types.

Provisioned IO (io1) EC2 EBS volumes are not always the most efficient option for IO intensive workload. Larger “gp2” volumes can provide similar or even better IO performance than smaller “io1” volumes, and for less cost. You can experiment by putting and merging multiple “gp2” volumes into single LVM disk mounts, and get similar storage amounts, but for less cost and faster.

Volume size (GiB)	Baseline performance (IOPS)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	100	1852	54,000
100	300	2,000	18,000
214 (Min. size for max. throughput)	642	2,290	8,412
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*
3,334 (Min. size for max. IOPS)	10,000	N/A*	N/A*
16,384 (16 TiB, max. volume size)	10,000	N/A*	N/A*

Illustration. 4: Table illustrating IOPS and IO throughput limit differences depending on volume size.

Wrong types chosen can introduce huge performance issues. Do not put cheap HDD “sc1” storage to Apps tier APPL\_TOP file system, and especially for database storage. As an example, using “sc1” 200 GB mount for DB tier data files can lead a simple 40 000 invalid objects compilation operation to take 2 days to complete. Experiment, benchmark, and find better setup that will suit your requirement and budget.

IOPS in EC2 EBS are based on 16 KiB I/O size for SSD (gp2, io1), or 1 MiB I/O size for HDD (st1, sc1). Also keep in mind that Oracle Database IOPS is not EC2 IOPS.

## 5. Backup / Recovery

EC2 provides EBS storage snapshot feature only. For Apps tier / DB tier file system standard snapshots can be scheduled. Database RMAN backups can be targeted on dedicated volumes with similar periodic snapshots, that can be automated in scripts via CLI. Or EFS (NAS).

Full EC2 storage snapshot documentation reference can be found here: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html>

## 6. Monitoring

AWS has a dedicated monitoring service called CloudWatch. It is available for free, but very limited. Paid option has a variation of checks and graphs available that can be used for full scale instance monitoring. It can be integrated with AWS Simple Workflow and Simple Notification services to process the alarms and on-call paging when required.

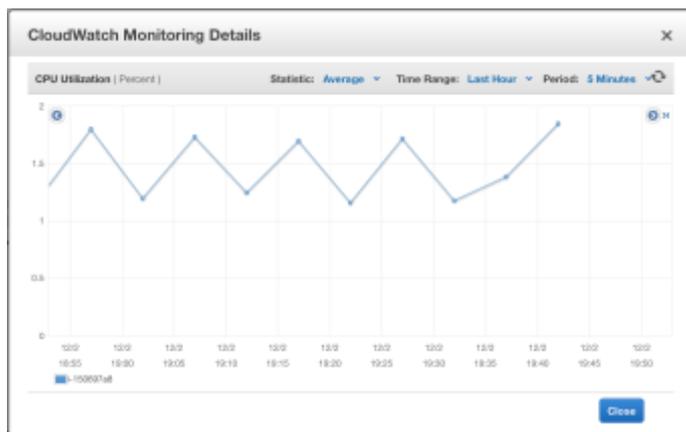


Illustration. 5: Example of Cloud Watch CPU utilization graph.

But it's not very versed for monitoring Oracle. Instead you can use your own custom monitoring solution, or setup EM agents for OEM monitoring.

## 7. Amazon RDS

RDS DBaaS service is not supported for Oracle EBS. But you can still use this service for other components integrated with your environment, like Identity Management, or Business Intelligence. Oracle Fusion Middleware 12c has an official support for Metadata Repository to be loaded into Amazon RDS instance. With older releases, like Oracle Fusion Middleware 11g, it's not supported, but with some "tweaks" you can still get your Metadata Repository loaded into the RDS instance. Main issue is that SYSDBA access is required during installation process, which is replaced with ORACLE\_MASTER and Amazon RDS PL/SQL API.

### Advanced configurations

We will cover several main advanced configuration examples outlining AWS deployment specifics with some recommendations we have found worth to mention.

#### 1. Oracle RAC

RAC cluster configuration is not officially supported. Main reason - there is no shared storage available, there is no multicast support for interconnect. There are non-official solutions available. Few to mention: Deploying Scalable Oracle RAC on Amazon EC2 (<https://aws.amazon.com/articles/7455908317389540>), Flashgrid VSAN solution, or do your own interconnect via OpenVPN, try to adopt EC2 EFS via direct NFS.

But in general, this is quite complex and not recommended. So only vertical scaling is possible.

#### 2. Apps Multi-Tier / PCP

Shared file-system can be implemented via EFS or your own NFS solution. So, basically, there are no limits. Or use non-shared file system approach using separate EC2 EBS storage volumes.

For network front-end purpose there is Elastic Load Balancing service available, that has all requirements to front-end Oracle E-Business Suite. There is support for basic items like sticky session and SSL.

Full ELB documentation reference can be found here:  
[https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html?icmpid=docs\\_elbv2\\_console](https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html?icmpid=docs_elbv2_console)

### 3. External Tier

Similar to standard Apps Multi-Tier deployment. DMZ network restrictions can be implemented via a dedicated VPC security group.

### 4. SSL

Native SSL can be implemented in the same way as for on-premise deployment. SSL termination is also possible with Elastic Load Balancer. AWS Certificate Manager is available to maintain and provision the certificates.

### 5. Integrations

Having Hybrid cloud deployment, please evaluate your network latency with the target availability zone. For heavy data exchange processes and systems integrated it might be good idea to move them along with your Oracle E-Business Suite on the same side. Very good example are: Data Warehouse, ETL. Or adjust your expectations and SLA.

### 6. Disaster Recovery

With AWS, there is no automation option available for managing DR for Oracle EBS, as both Apps/DB tiers are built via “IaaS” approach. Build your own DR tiers based on different Availability zones, or even Regions. Amazon RDS for Oracle (DBaaS) has a built-in automation support.

To speed up the initial clone, you can create EC2 EBS storage volume snapshots, and push them to other zones or even regions, as these are transferable.

### 7. Encryption

As it was mentioned above, encryption can be implemented on applications level, like TDE implementation in your Oracle Database instance, or implement storage layer encryption at rest. Amazon RDS for Oracle supports both approaches.

## **R12.2**

Oracle EBS R12.2 is released, and it has many new features introduced. We can just even mention, at least, dual filesystem and online patching feature.

This is a list of specifics we noted while implementing R12.2 on cloud:

- All items mentioned above for R12.1 are still relevant.
- ADOP dependency: Private IP addresses. Number of FND tables must contain valid IP addresses of all nodes. And EC2 Instance restart might change the private IP assigned. There can be an automated startup service deployed which can cover the required updates, and even communicate with cloud platform through CLI to get or set the required values.

- ADOP dependency: Validation of the /etc/hosts which, basically, is not in use by default, and recommended being not in use. Same private IP address issue affects this, and similar startup service can be deployed which can cover the required updates.
- IOPS / Throughput planning is important for storage. For Apps tier, that's ADOP fs\_clone and file system sync management, that is managed by online cycles. For DB tier, that's workload handling edition objects. For example, "fs\_clone force=yes" run duration on 220 GB standard "gp2" EC2 EBS volume is 42 minutes. "sc1" - didn't complete in 3 days and was terminated.

## Microsoft Azure

With Microsoft Azure cloud platform, everything can be built in the same manner using similar "IaaS" approach. In respect to Oracle E-Business Suite it has all main analogues available for Virtual Network, Load Balancing, Storage, Site Recovery, and even Linux Virtual Machine support.

At the time this paper was prepared, main disadvantage comparing EC2 and Azure was the compute limits. EC2 provided more powerful capacity options that are almost 10 times powerful. There is a 20 vCPU max quota (30 West Europe region), that can be extended via support request, 100+ GB RAM support, enough storage options.

## Oracle Cloud review

With Oracle Public Cloud platform, everything can be built in the same manner using "IaaS" approach. There are still some disadvantages and not all cloud services are supported from infrastructure service availability point of view, but there are many key differentiators why OPC can be considered for Oracle EBS deployment.

Some items to mention:

- Completely based on Oracle's own stack.
- "IaaS", but with some additions, like PaaS support for DB tier: Database Cloud Service and Exadata Cloud Service. You can deploy service based database, and use it as a supported database instance for your Oracle EBS, which supports RAC as well.
- "Lift and Shift" feature to transfer (migrate) your instance between on-premise and cloud deployment. Basically, this is a collection of tools that can automate this process.
- Easy multi-node provisioning for Oracle EBS from templates. Quick deployment for rapid development needs (provisioning).



Illustration. 6: Diagram illustration possible EBS deployment in OPC.

- EBS Cloud Admin Tool central management.

During OpenWorld 2017 there was a number of very interesting announcements in regard to automatic management for Oracle EBS environments deployed in OPC. That includes:

- Managed Oracle EBS environment via GUI console or CLI.
- Full scaled and managed Backup / Restore for Oracle EBS instance.
- Fast Cloning feature available with DBaaS.
- Automated provisioning and cloning, with post provisioning custom framework support.
- Managed scaling.
- Automated DMZ and External application configuration.
- Automated DB tier PSU patching.
- Fully automated DR creation and failover.

With other cloud vendors, you need to build your own custom framework to deliver similar process automation capabilities. Here we can get this out of the box available, although most of the critical items above are still on the roadmap.

As we know, Oracle EBS is really bound to Oracle's own products in terms of integration requirements. With other cloud service providers, it's "IaaS", you need to install, patch, and maintain products integrated yourself. In OPC Oracle is introducing Integrations Cloud Service.



*Illustration. 7: Example Diagram of available end-points supported by ICS.*

But still there are items that you need to build yourself, where cloud service based solution is not yet supported. The best example here is Identity Cloud Service. You still need to build "IaaS" based OID/OUD/OAM instances and maintain them.

There is an ongoing shift to SaaS based Applications. But implemented module list is still small, and still the process is in active development. It is confirmed and promised, that Oracle E-Business Suite is still on the radar and roadmap till 2030.

All the certification and feature updates are tracked under this reference list:  
 Getting Started with Oracle E-Business Suite on Oracle Cloud (Doc ID 2066260.1)  
<https://www.facebook.com/groups/EBS.SysAdmin/>  
[https://twitter.com/Oracle\\_EBS](https://twitter.com/Oracle_EBS)

Our recommendation: Do not rush! Implement, test, evaluate, compare costs, migrate.

### **Cloud deployment automation**

One of the most used cloud deployment automation is auto scaling. We can scale our applications Vertically and Horizontally.

Classical Vertical scaling is an increase of the compute resources. We can change our instance types, add more RAM, more CPU power. But this might dramatically increase runtime cost, and still has limits.

For Oracle EBS server we can increase JVM memory heap size, increase your JVM process count, or add more oacore\_server WebLogic servers in R12.2 case. Plan the downtime, implement the resource increase, and restart for changes to take effect.

Horizontal scaling is an increase of the compute resources, basically, addition of more compute nodes when it's needed, and at the same time, release of compute nodes when we do not need these additional resources anymore. In other words - flexibility. For Oracle EBS - we are adding more apps tiers.

Comparing to a simple LAMP stack or HTML based web server, the main challenge with Oracle E-Business Suite is its deep dependency on internal configuration repository (FND tables, like FND\_NODES). With R12.2 an additional complexity is introduced by ADOP.

Also, there is no universal method how to add a new node for your Oracle EBS instance, that will work for any environment. It's a manual effort to run "perl adclonctx.pl addnode", and find the best combination of pairs file to minimize any post manual correction inside context file and instance configuration.

Node deletion is not a straight forward process. It involves a downtime, setup\_clean, re-run of AutoConfig on each node in 2 rounds. In R12.2 we also need to handle patch file system cleanup.

Amazon EC2 supports an automatic scaling processing based on CloudWatch monitoring statistics. For example, we can add a new Oracle EBS web node on-demand, if existing stack is overloaded. This is being configured using Auto Scaling groups.

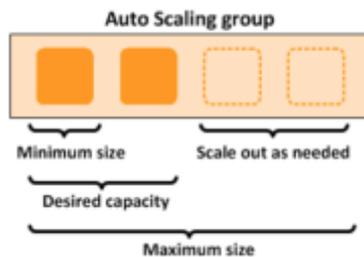


Illustration. 8: AWS EC2 Auto Scaling group illustration.

EC2 Auto Scaling documentation reference can be found here: <https://aws.amazon.com/documentation/autoscaling/>

Auto-Scaling group is created, defining minimum, maximum, and desired node count parameters. Scaling plan is built defining when and how to scale out, and rules how to scale in. It can be defined for a specified time window, or, as mentioned above, can be based on CloudWatch monitoring.

New instances are created based on AMI, that can be created from existing Oracle EBS Apps tier, and stored inside private repository. Once created, references automatically can be added to existing Elastic Load Balancer configuration and enabled. Once new instance is started, you create custom first boot hooks to initiate an automated script to cover all the required post configuration changes (cloud-init method).

Auto-Scaling is free. You pay only standard rates running your additional EC2 instances. But it has limits.

Resource	Default Limit
Launch configurations	100
Auto Scaling groups	20
Scaling policies per Auto Scaling group	50
Scheduled actions per Auto Scaling group	125
Lifecycle hooks per Auto Scaling group	50
SNS topics per Auto Scaling group	10
Classic load balancers per Auto Scaling group	50*
Target groups per Auto Scaling group	50*
Step adjustments per scaling policy	20

*Illustration. 9: AWS EC2 Auto Scaling limits.*

An example of actions required to cover while adding a new Web Node:

- OS setup: Software, kernel, limit configuration is coming from created AMI. No additional changes are required, except the hostname.
- Hostname: We need to generate a unique hostname, set it on the server, DNS, update /etc/hosts.
- Storage: Mount the Apps Base storage layer.
- Apps setup: Generate required pairs file, run “perl adclonectx.pl addnode”, run AutoConfig on all existing nodes (or, at least, “generatetns” step), reload Apps TNS listeners, any other custom post configuration required.
- Service restart on new created node.
- Elastic Load Balancer is handled automatically. Will enable the node once the TCP socket ping succeeds.

An example of actions required to cover while removing an existing Web Node:

- Instance termination.
- Elastic Load Balancer will drop the removed instance from its configuration automatically.
- Apps setup: Master node to trigger the node deletion process.
- R12.1: No process of node deletion. Concurrent Service Managers configured for new node can be just disabled from back-end, INST\_TOP can be deleted from file system. No way to delete a node without a complete downtime and setup\_clean process.
- R12.2: Abandon the node, “adProvisionEBS.pl ebs-delete-node”, “txkSetAppsConf.pl -configuration=removeMS” on existing nodes if OHS is pointed to all WebLogic nodes in the cluster, including the removed one, file system cleanup.

The node deletion process is quite costly. In theory, we do not need to always remove the node that we do not need anymore, rather put it into disabled state, and re-use once scale out is happening again.

An example of actions required to cover while re-using a removed Web Node:

- R12.1: We just don’t need to wipe out the INST\_TOP. Just launch the instance, configure the OS part, and restart the services.
- R12.2: There is no way to recover from abandoned node, only a complete re-add even if we do not delete the node and do not delete the file system content. At the same time, we can’t allow not setting the abandoned node state, as any ADOP online patching related phase is going to fail referencing and validating a non-existing node configuration.
- Periodic housekeeping: Once you have a downtime window, it’s worth to go through “setup\_clean” process to have everything re-registered clean and drop garbage.

One more important area for R12.2 - ADOP online patching cycles, how it is affected while performing auto scaling. While performing test scenarios, we didn’t track any significant impact for scale out process. But there is no 100% guarantee that this process is completely transparent. Scale in process is

an Abandon process. It is handled by ADOP during the cycle correctly. There are more high risks if scaling is happening in the middle of cutover, and that might affect the system configuration dramatically. CloudWatch doesn't know anything about what's going on in Oracle EBS and is triggered by the monitoring results. There is a low chance that this is going to be triggered during a maintenance window, but still Auto-Scale group can be disabled during these maintenance activities.

## **Summary**

Oracle EBS implementation on cloud is more less straightforward, and is a "IaaS" deployment. Oracle Public Cloud offers "PaaS" based Oracle Database available as a service. IO layer performance is the main bottleneck risk. Good design, evaluation of estimates and testing are very important. Security is important. Cost efficiency is a subject of detailed calculation. Cloud automations, like auto-scaling, are possible, but still are too complicated from configuration management.

## **Contact address:**

### **Andrejs Prokopjevs**

The Pythian Group

319 McRae Ave, Suite 700

K1Z 0B9, Ottawa, Ontario, Canada

Phone: +1-613-565-8696

Email [info@pythian.com](mailto:info@pythian.com)

Internet: [www.pythian.com](http://www.pythian.com)