

The following text is partly taken from the Oracle book “*Middleware and Cloud Computing*”
It is available from Amazon: <http://www.amazon.com/dp/0980798000>

Cloud Databases and Oracle

When designing your cloud architecture, sooner or later the question about the database will arise. There are many options, and in order to make an informed decision as to which will best suit your architecture, you should know the pros and cons of at least four:

- You can start installing your database on an AMI with the operating system of your choice, or even select an AMI provided by Oracle and set up the included Standard or Enterprise Edition.
- SimpleDB is an option if you prefer the scalability and availability of a non-relational database.
- The relational database service from AWS offers a convenient and easy way to create and manage an Oracle MySQL database as a cloud service.
- Starting May 2011 you can use RDS to create an Oracle database.

So let's look at the details.

1.1 Oracle Database AMIs

Running an Oracle database in the cloud can be as simple as selecting the right AMI from the list.

Oracle Provided AMIs

Preconfigured AMIs are available from Oracle. These AMIs come with Oracle Enterprise Linux, the Oracle database, Enterprise Manager Database Control and the web based rapid development tool Applications Express (APEX).

Here is a link to Oracle's cloud offering which includes the AMIs provided by Oracle:

*Oracle
documentation*

<http://www.oracle.com/technology/tech/cloud/index.html>

As you may already know, I always recommend not to rely on the posted AMIs. Although they are certainly a good place to start, you have to accept that they can sometimes disappear.

If you start with one of the AMIs provided by Oracle, be sure to create your own image so that you can start it anytime, even if Oracle were to update the set of publicly available AMIs with a newer version. Yes, you do have to pay for the storage of the image snapshot.

In summary, the AMIs published by Oracle are a convenient start if you are looking for the current version of the database and the Oracle Enterprise Linux is your preferred operating system. Always use these images as a basis from which to create your own.

Custom AMIs

If Oracle doesn't provide the database version that you like to use in the cloud, or if you are not happy with Oracle Enterprise Linux, you can always start with an AMI containing just the supported operating system of your choice and install the database yourself (be sure to select a supported operating system).

Even if you choose to start with an S3-backed AMI, you will need an EBS volume for the data file.

Oracle VM with Database on AWS

Since early 2011 Oracle VM based images are available on AWS. Personally, I would like to see Oracle provide a wider range of products based on Oracle VM, including a larger variety of versions and the availability of all AMIs for all regions.

Again, don't rely on an image provided by Oracle; create your own so you can stay independent.

1.2 AWS SimpleDB

Amazon's SimpleDB service is highly scalable, highly available and even includes automatic data indexing. It is based on EC2 and S3 and therefore well suited to applications that have traditionally required clustered relational databases because of availability and performance requirements.

SimpleDB is a cloud service, so no installation is necessary, and the database does not run within your EC2 instance. You can access SimpleDB using an API (there is no integration with the AWS management console). Unlike a clustered database such as Oracle RAC, SimpleDB requires no CAPEX and no DBA for maintenance and administration.

AWS SimpleDB is a *non-relational* data store. The equivalent of a relational database's table is called domain. A domain stores data but, unlike with tables in a relational database, it is not possible to query data across domains in SimpleDB.

SimpleDB is another example of how a cloud service can potentially replace a complex and expensive infrastructure such as a clustered database.

Tools

There is a Firefox plugin called sdbtool for querying and updating Simple DB database domains, which can be installed directly from the following location:

<http://code.google.com/p/sdbtool>

AWS officially provides a Java library to access SimpleDB. Coding SimpleDB in Java is easy and compact.

A lot of folk use Typica (a Java client library) for a variety of AWS services including SimpleDB. You can obtain it from the following site:

<http://code.google.com/p/typica>

Costs

There is a free usage tier for SimpleDB, but do keep in mind that the total cost only covers the following items:

- SimpleDB machine hours
- Data transfer in/out
- Storage space

The first 25 machine hours and 1 GB of storage is free. On average this enables 2,000,000 GET or SELECT API requests to be completed per month free of charge.

Thereafter, \$0.154 per SimpleDB machine hour consumed and \$0.275 per GB-month is added to your bill.

Additional costs for the data transfer are as listed in the table below e.g. in the EU you would have to pay \$0.15 per GB for up to 10 TB of data transfer.

Another popular non-SQL database is MongoDB, check their website at:

<http://www.mongodb.org>

1.3 AWS Relational Database Service

Amazon's Relational Database Service (RDS) is a fully featured database instance provided as a service. You can get started without having to use any command-line tool. The AWS management console enables the creation of new instances and the scaling of compute and storage capacity for the instances.

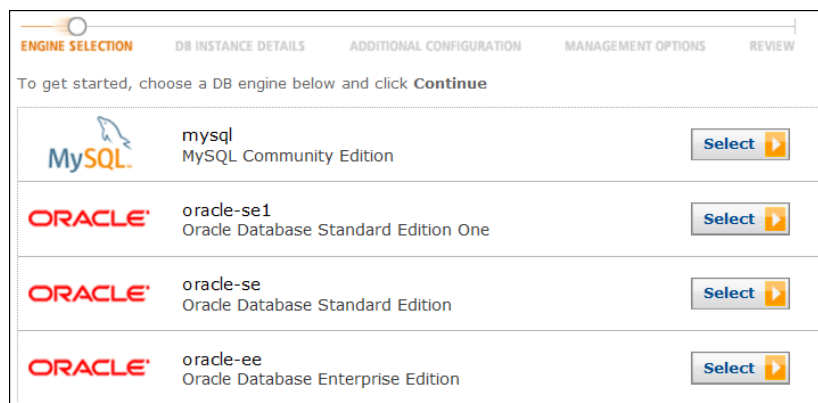


Figure 1: RDS Engine selection

AWS currently supports the community edition of MySQL 5.1 with InnoDB as the default database storage engine and the provisioning of Oracle databases as described in the following chapter. There is support for accessing the instances via SSL. The maximum number of database instances is limited to 20 per account. However, it is possible to request this limit be raised, by completing the form at the following URL:

<http://aws.amazon.com/contact-us/request-to-increase-the-amazon-rds-db-instance-limit>

At first glance, instance classes appear to match the EC2 standard and high memory instances, however there are some subtle differences.

All database instances are 64-bit (whereas the EC2 m1.small instance type is only 32). Also, the high memory db.m2.xlarge instance provides high I/O performance (yet the EC2 instance type m2.xlarge provides only medium I/O performance). There is no micro or HPC database instance class option for the RDS service:

Table 1: RDS Instance Types

Type	bit	GB RAM	Cores	ECU	I/O perf
Standard					
db.m1.small	64	1.7	1	1	Mod
db.m1.large	64	7.5	2	4	High
db.m1.xlarge	64	15	4	8	High
High Memory					
db.m2.xlarge	64	17.1	2	6.5	High
db.m2.2xlarge	64	34.2	4	13	High
db.m2.4xlarge	64	68.4	8	26	High

It is always possible to resize an instance at a later stage. You can therefore complete all your configurations and initial tests with a cheap, small instance and then resize the instance when it is ready to serve the production load.

On the instance details screen, you can also select whether the RDS service should *automatically* perform minor version upgrades of your instances.

Multi-availability zone

The next option is yet another of those wonderful cloud features: you only need to tick the “Multi-AZ Deployment” and AWS will then silently do all the magic to get a standby copy of your instance in another availability zone. This instance runs in another data center within the same region, completely isolated and independent from the first one.

The standby copy is synchronously updated. Also, the standby copy is standby only - as the name suggests - and doesn't serve any database requests at all, not even read requests.

RDS fails over to the standby instance when the primary instance or even the whole availability zone fails, when the instance class of the database instance is changed, or when software is updated during the maintenance window provided.

This failover is automatically initiated. During the failover procedure, the CNAME record of the main instance is changed so that it points to the standby copy.

There is no button on the AWS management console forcing a failover, but changing the instance class will do so. The failover process is also logged by the RDS event log, which can be viewed from the management console.

It is not possible to assign an elastic IP address to a database instance.

I guess you will not be surprised if I say that you have to pay for the standby instance. We will look at the costs at the end of this section, but in a nutshell your costs will double as you will be charged the same fees as for the primary instance.

Next decide how much storage you want to allocate for your instance. The minimum size is 5 GB, although you can create instances as large as 1 TB.

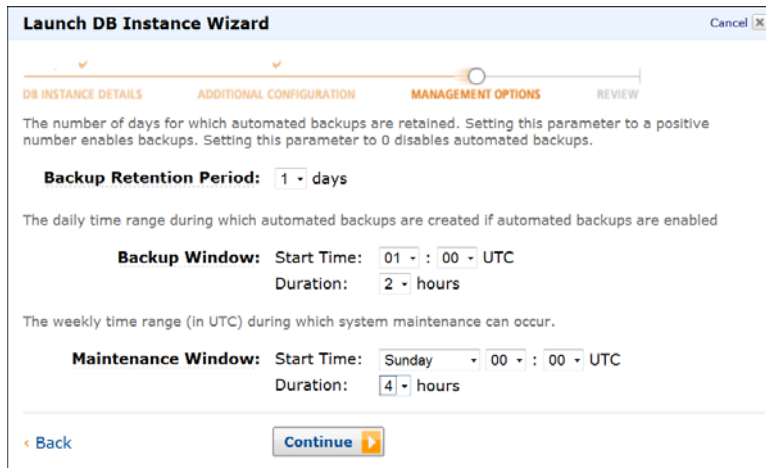


Figure 2: RDS Configure Backup Window

Instance Management

Once the instance is created it will display its endpoint. The syntax of the endpoint is as follows:

```
instancename.uniqueidentifier.region.rds.amazonaws.com
```

The identifier of the instance that I just created for this example is:

```
surfinstance.cawztecsrizb.eu-west-1.rds.amazonaws.com
```

This endpoint DNS name remains the same even if you delete your instance and recreate it at a later stage with the same instance name - although it might map to a different IP address.

With the default settings, no network access is allowed to the database instances. In other words, it is secure but unusable. To be able to access the database from an EC2 instance, allow access rights for an EC2 security group by adding it to the database security group.

Security groups

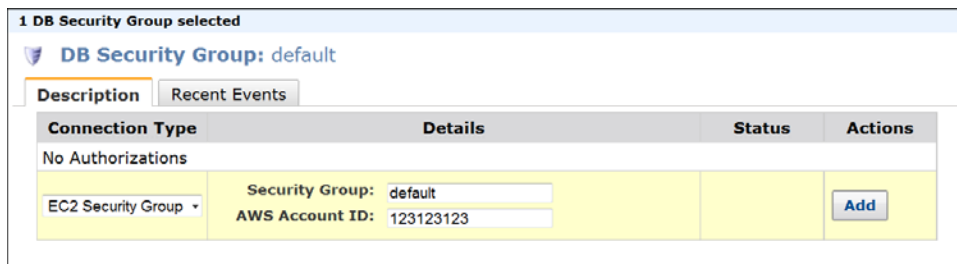


Figure 3: RDS Security Group

Be careful when adding the account-ID and don't copy any spaces, since here the AWS management console is fussy with leading or trailing space. Alternatively, you can authorize an IP address or a range of IP addresses.

Then switch to the monitoring tab. The built-in monitoring for RDS instances is done with AWS CloudWatch. Metrics such as CPU, storage and memory utilization, as well as the requests, throughput and latencies for read and write requests, are all graphically displayed for a configurable time range.

See "Middleware and Cloud Computing" to learn more about CloudWatch details such as all available metrics for monitoring, and how to retrieve statistics over time using the command-line API.

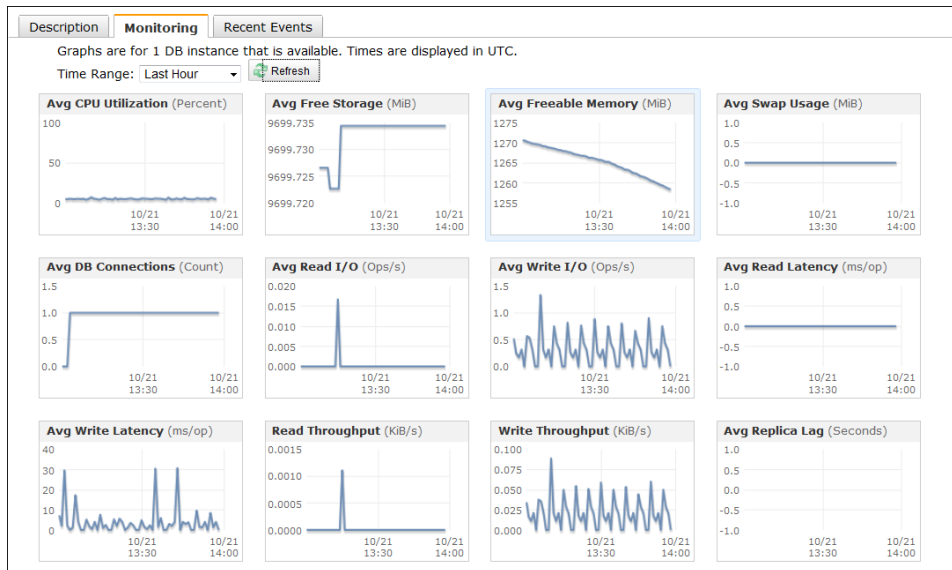


Figure 4: RDS CloudWatch Monitoring

Let's take a look at read replicas. RDS supports read replicas that take advantage of the native and asynchronous MySQL replication. Read replicas improve the throughput for applications with a high database read ratio. You should deploy a read replica when you need to scale beyond the I/O capacity of the primary instance class. It is currently possible to create up to five read replicas for any database instance.

To create a read replica, select one of the primary database instances and click on "Create Read Replica". It is not possible to create a read replica from read replica.

Figure 5: RDS Create Read Replica

The read replicas will automatically use the master database security group settings. For the instance class, it is recommended that the replica's instance class isn't smaller than the primary instance.

You should be aware of the conceptual differences between multi-zone instances and read replicas: multi-availability zone instances increase the availability, whereas read replicas are used for scalability.

Table 2: Comparison Multi-Availability Zone Instances and Read Replicas

	Multi-availability zone instances	Read replicas
Purpose	<i>Availability</i>	<i>Scalability</i>
Number of instances	1 primary + 1 secondary	1 primary + n secondaries
Secondaries used during normal operation?	No	Yes
Synchronization	Synchronous	Asynchronous
Use with ELB	No (only one active)	Not supported
Use with elastic IPs	Not supported	Not supported
Cost	Doubles	Per Instance

Read replicas can lag behind if the write I/O requests of the primary instance cannot be replicated early enough to the replica. The CloudWatch metric “Replica Lag” can help you to monitor the time lag.

If you combine read replicas with a multi-availability zone instance you will benefit from both. When creating a read replica from a multi-availability zone instance, you can even avoid the short I/O suspension that usually occurs because a snapshot of the primary has to be taken first.

To explore the snapshot feature, change to the instances view under Databases / DB Instance, select the image *Snapshot and restore* and take a snapshot of the instance.

Once you have taken a snapshot, you can restore the state of the database to the time of the snapshot.

Delete Instance

RDS Pricing

RDS is a great cloud service that comes with all the cloud benefits. Billing for RDS is complicated, so it is essential that you understand how much you are paying for this service. For multi-availability zone instances your bill will be itemized as follows:

- Per instance-hour cost primary
- Per instance-hour cost secondary
- Database storage primary
- Database storage secondary
- Backup storage
- Data transfer costs

1.4 RDS and Oracle database

Amazon’s offer to provision Oracle databases with RDS is quite exciting. For the first time *real* cloud computing is possible with an Oracle database. Here is why: Cloud computing is nowadays often used as a synonym for what was named “online” a decade ago (see webmail, or cloud storage offering). One of the key criteria for cloud computing is that you only pay what you use - and historically Oracle wasn’t very good in this discipline. Try to buy a database or OFM license for exactly a three week marketing campaign from your sales rep, then you know what I mean.

Yet when using RDS to provision an Oracle database this has changed now. You can either bring your own existing license to the cloud or you can use the RDS service based on an hourly rate:

- License included: Only Oracle Standard Edition One database is supported. Pricing starts at US\$ 0.17 per hour for Europe.

Region: EU (Ireland)	
	Price Per Hour
DB Instance Class (On-Demand)	
Small DB Instance	\$0.17
Large DB Instance	\$0.68
High-Memory DB Instance Class	
Extra Large DB Instance	\$0.94
Double Extra Large DB Instance	\$1.88
Quadruple Extra Large DB Instance	\$3.76

Figure 6: Oracle RDS Pricing with license included

- BYO License: Supports Enterprise Edition, Standard Edition, and Standard Edition One Oracle databases and starts at \$ 0.11 per hour.

Same as for MySQL you can benefit from preconfigured database parameters, automatic backups, software patches and monitoring with Amazon CloudWatch. In addition you can create snapshots at any time.

Launch DB Instance Wizard

ENGINE SELECTION | **DB INSTANCE DETAILS** | ADDITIONAL CONFIGURATION | MANAGEMENT OPTIONS

To get started, choose a DB Instance engine and class below

Engine: oracle-se1

License Model: bring-your-own-license

DB Engine Version: 11.2.0.2.v2 (default)

DB Instance Class: db.m1.small

Multi-AZ Deployment: No

Auto Minor Version Upgrade: Yes No

Provide the details for your RDS Database Instance.

Allocated Storage: 500 GB (Minimum: 10 GB, Maximum 1024 GB)

DB Instance Identifier: surfInstance (e.g. mydbinstance)

Master User Name: surfuser (e.g. awsuser)

Master User Password: ●●●●● (e.g. mypassword)

Figure 7: RDS Oracle instance creation wizard

However, with the Oracle database engine provisioned by RDS there is no support for replication at the moment, such as the multi-AZ deployments and read replicas for MySQL instances.

1.5 WebLogic with RDS

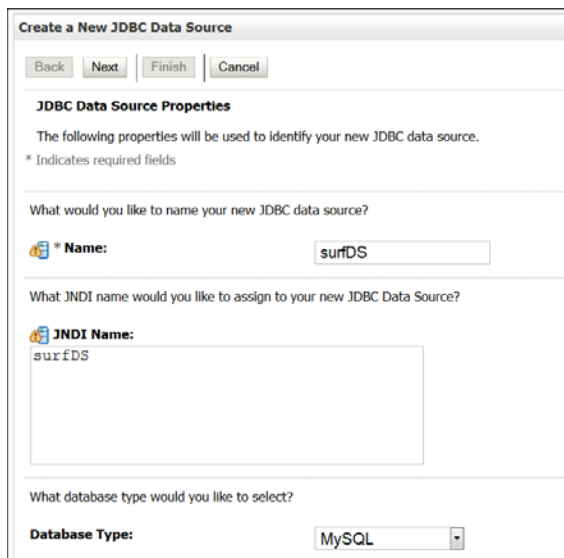
Have you ever wondered how you can use the RDS instance with WebLogic, the Oracle Service Bus or any other middleware product? Well, all you need is a data source.

To try it, start your favorite AMI with WebLogic. In case you don't have an AMI, check out the different options for creating one as described in the "Middleware and Cloud Computing" chapter about AWS basics.

Configuring a WLS Data Source

Once WebLogic is running, connect to the admin server using its public DNS address. Each of the following steps for the creation of the JDBC data source are carried out using the WebLogic admin console, rather than with the AWS management console. *Creating a data source*

Using the WebLogic admin console, navigate to Domain Structure / Domain Name / Services / JDBC / Data Sources. Under JDBC create a new data source and specify the name and the JNDI name of the data source as shown in the screenshot in Figure 8.



Create a New JDBC Data Source

Back Next Finish Cancel

JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

* Indicates required fields

What would you like to name your new JDBC data source?

Name: surfDS

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name: surfDS

What database type would you like to select?

Database Type: MySQL

Figure 8: WLS Create Data Source

On the following two screens, leave all the settings for the driver class and the transactions settings with their default values, and keeping clicking 'next' until you arrive at the data source settings as shown in Figure 9.

Fill in the data source properties exactly as you specified them when creating the RDS instance. For the JDBC host name use the RDS endpoint.

Create a New JDBC Data Source

Back Next Finish Cancel

Connection Properties
Define Connection Properties.

What is the name of the database you would like to connect to?

Database Name: surfDB

What is the name or IP address of the database server?

Host Name: surfinstance.cawztecsriz

What is the port on the database server used to connect to the database?

Port: 3306

What database account user name do you want to use to create database connections?

Database User Name: surfuser

What is the database account password to use to create database connections?

Password: ●●●●●●

Confirm Password: ●●●●●●

Back Next Finish Cancel

Figure 9: WLS Data Source Properties

JDBC URL The settings for the driver class name, the JDBC URL, the user name and the password are already pre-populated. The JDBC URL will display as follows (your hostname will be different):

```
jdbc:mysql://surfinstance.cawztecsrizb.eu-west1.rds.amazonaws.com:3306/surfDB
```

After checking these values, click on the “Test Configuration” button at the top of the page. If the connection to the RDS database can be established you will see a green success message in the top left of the screen.

Messages

✓ Connection test succeeded.

Create a New JDBC Data Source

Test Configuration Back Next Finish Cancel

Test Database Connection
Test the database availability and the connection properties you provided.

What is the full package name of JDBC driver class used to create database connections in the connection pool?
(Note that this driver class must be in the classpath of any server to which it is deployed.)

Driver Class Name: com.mysql.jdbc.Driver

What is the URL of the database to connect to? The format of the URL varies by JDBC driver.

URL: jdbc:mysql://surfinstance

What database account user name do you want to use to create database connections?

Database User Name: surfuser

Figure 10: WLS Test JDBC Configuration

Then click finish. Re-select the data source and target the data source to the admin server as shown below:

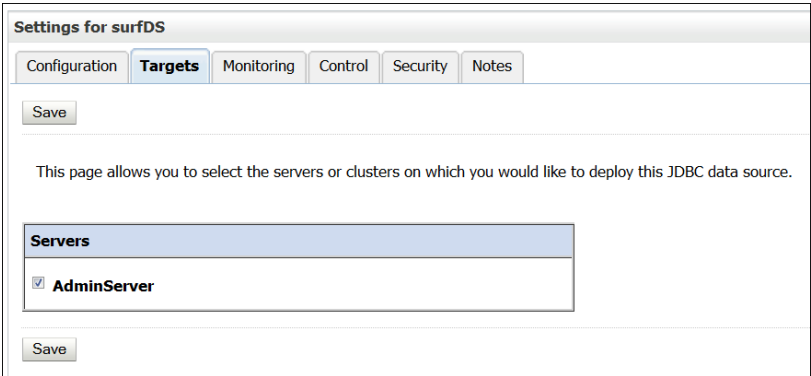


Figure 11: WLS Set JDBC Data Source Target

Verifying the RDS Connectivity

For final proof that the connection pool is actually connected to the RDS instance, increase the initial number of connections in the JDBC connection pool from 1 to 10.

Then switch back to the AWS management console and select the RDS instance's monitoring tab. Because you have increased the number of connections in the JDBC pool to 10, you will now see 10 DB connections open in CloudWatch monitoring:

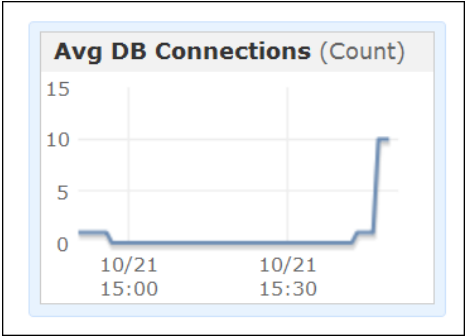


Figure 12: AWS CloudWatch WebLogic Connections to RDS

1.6 Conclusion

To conclude, you could operate and maintain your own Oracle or non-Oracle database on an EC2 instance with an EBS volume for the data file. However, apart from the AWS skills this option requires database administration knowledge and you have to look after backups yourself.

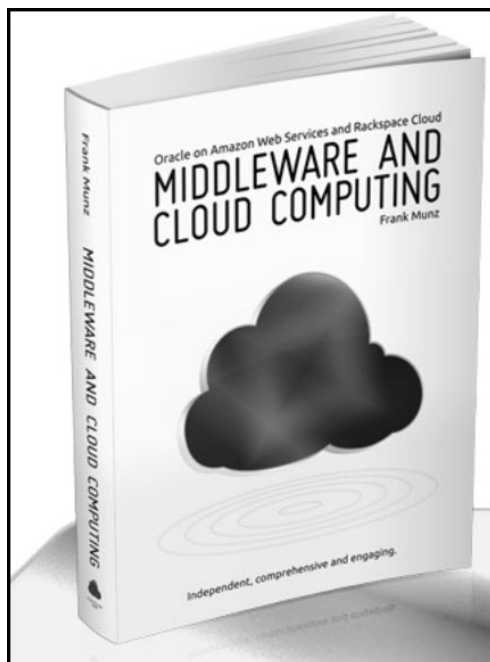
If you are looking for an easy start with an Oracle database, it is possible check if there is an AMI provided by Oracle. Although at the time of writing, it is not possible to run Oracle's flagship database product RAC on any of the industry leading IaaS platforms.

Amazon's relational database service can be used without any database administration knowledge and requires no installation. RDS is convenient to use, easy to scale and provides failover and scheduled backups. The costs of RDS can quickly mount up, so to avoid any nasty surprises, be sure to do calculations for a month or longer using the AWS calculator.

Don't forget the other database service offered by AWS. Should a non-relational database service match your requirements, then it's worth considering the AWS SimpleDB. SimpleDB has a free usage-tier, which makes it an ideal location to store configuration data and information for bootstrapping instances.

ORACLE MIDDLEWARE AND CLOUD COMPUTING

An Introduction to Cloud Computing and Oracle Middleware on Amazon Web Services (AWS), Rackspace Cloud and the RightScale Cloud Management Platform.



Reader Reviews:

“Great book: Hands-on tutorial, up-to-date and well researched for Oracle Middleware Architectures in the cloud”

“Clear, objective and brand new - I highly recommend it!”

“Excellent for practical use“

Contents and more information about the book: <http://cloudbook.munzandmore.com/>

Get it from Amazon.com: <http://www.amazon.com/dp/0980798000>

High-end Oracle Middleware/SOA/Cloud Courses and Consulting (worldwide): <http://www.munzandmore.com/courses>

Excerpt of “Middleware and Cloud Computing”, by Frank Munz

munz & more