Configuring Amazon RDS as an Oracle PeopleSoft Database

April 2017

amazon web services

DLZP GROUP

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

Amazon Web Services (AWS) provides a comprehensive set of services and tools for deploying enterprise-grade solutions in a rapid, reliable, and cost-effective manner. Oracle Database is a widely used relational database management system that is deployed and used with many Oracle applications of all sizes to manage various forms of data in many phases of business transactions. In this whitepaper, we describe the preferred method for configuring an Amazon Relational Database Service (Amazon RDS) for Oracle Database as a backend database for Oracle PeopleSoft Enterprise, a widely used enterprise resource planning (ERP) application.

Introduction

An Amazon RDS for Oracle Database (Amazon RDS for Oracle or RDS for Oracle) provides scalability, performance, monitoring, and backup and restore support. Deploying an RDS for Oracle Database in multiple Availability Zones (AZs) simplifies creating a highly available architecture because a Multi-AZ deployment contains built-in support for automated failover from your primary database to a synchronously replicated secondary database in an alternative Availability Zone. RDS for Oracle always provides the latest version of Oracle Database with the latest patch set updates (PSUs) and manages the database upgrade process on your schedule, eliminating manual database upgrade and patching tasks.

You can use Oracle PeopleSoft Enterprise with Amazon RDS and the preferred Oracle Database edition (using your own license, or a license managed by AWS) to create a production RDS for Oracle Database (DB) instance, or the Standard Edition/Standard Edition One/Standard Edition Two to create RDS for Oracle pre-production environments. Before you can use the PeopleSoft components, you must create and populate schemas for them in your RDS for Oracle Database. To do so, use the AWS RDS console or command line interface (CLI) to launch your DB instance. After the instance is created you need to modify the delivered PeopleSoft DB Creation Scripts and run them against the RDS for Oracle Database instance.

After completing the procedures described in this whitepaper, you can leverage the manageability features of Amazon RDS for Oracle—such as multiple Availability Zones for high availability, hourly pricing of an Oracle Database, and a virtual private cloud (VPC) for network security—while operating the PeopleSoft Enterprise application on AWS.

Database Compatibility and Oracle PeopleSoft Components

Before getting started with Amazon RDS for Oracle, check that your Oracle Database edition and version are compatible with Amazon RDS and the corresponding PeopleSoft components.

The following Oracle editions and versions are supported on Amazon RDS:

- Oracle Database Standard Edition (BYOL) - 11.2.0.4.v1+
- Oracle Database Standard Edition One (BYOL/LI) - 11.2.0.4.v1+
- Oracle Database Standard Edition TWO (BYOL/LI) - 12.1.0.2.v2+
- Oracle Database Enterprise Edition (BYOL) - 11.2.0.4.v1+, 12.1.0.2.v1+

We have validated compatibility between the PeopleSoft versions listed in the following table and RDS for Oracle. Most of these versions are already being used in production by our customers.

PeopleSoft Release	PeopleSoft Components	Database Releases	Notes
PeopleTools 8.55	Application Server Process Scheduler Server	Oracle Database 12.1.0.2.0 11.2.0.4.0	Works with all PeopleSoft Applications PeopleSoft HCM PeopleSoft FSCM PeopleSoft CRM PeopleSoft IAH PeopleSoft Campus Solutions
PeopleTools 8.54	Application Server Process Scheduler Server	Oracle Database 12.1.0.2.0 12.1.0.1.0 11.2.0.4.0	Works with all PeopleSoft Applications PeopleSoft HCM PeopleSoft FSCM PeopleSoft CRM PeopleSoft IAH PeopleSoft Campus Solutions
PeopleTools 8.53	Application Server Process Scheduler Server	Oracle Database 12.1.0.2.0 12.1.0.1.0 11.2.0.4.0 11.2.0.3.0 11.1.0.7.0 10.2.0.5.0	Works with all PeopleSoft Applications PeopleSoft HCM PeopleSoft FSCM PeopleSoft CRM PeopleSoft IAH PeopleSoft Campus Solutions

Prerequisites

Before you create an Amazon RDS for Oracle Database instance, you need to make some decisions about your configuration and complete some basic tasks. You will use the decisions you make in this section later to configure your RDS for Oracle Database instance.

Decide Which AWS Region to Use

Decide which of the available AWS Regions you want to use for your workload. When choosing a Region, consider the following factors:

- Latency between the end users and the AWS Region.[1]
- Latency between your data center and the AWS Region. This is one of the most critical factors when you have PeopleSoft running in the cloud and backends running on premises.
- AWS cost: The AWS service cost varies depending on the Region.
- Legislation and compliance: There might be restrictions on which country your customers' data can be stored in.

Create a VPC, Subnets, and EC2 Instance Security Groups

Create your VPC, subnets, and Amazon Elastic Compute Cloud (Amazon EC2) instance security groups by following steps 1 through 4 in the Amazon Virtual Private Cloud Getting Started Guide.[2]

Use a Supported PeopleSoft Version

You must use an RDS for Oracle Database instance version supported by PeopleSoft Enterprise:

- The PeopleSoft DB version must match your RDS for Oracle Database instance version.
- You must also ensure that the PeopleSoft components are supported by Amazon RDS for Oracle. As of the writing of this document all PeopleSoft components are supported on Amazon RDS.

Determine Which Oracle Database Licenses You Have

When creating your DB instance, you can select either Bring Your Own License (BYOL) or License Included (LI). Not all editions are available for License Included. Before creating the DB instances, make sure you understand which licenses your organization has. For more details, see

https://aws.amazon.com/rds/oracle/faqs/

Ensure That You Have Sufficient IAM Permissions

You must have sufficient AWS Identity and Access Management (IAM) permissions to perform the actions described in this paper. You will need permissions to configure the following AWS services:

- Amazon Virtual Private Cloud (VPC) [3]
- Amazon EC2 [4]
- Amazon RDS [5]
- Amazon Route 53 [6]

Determine the Size of the Database You Will Require

Determine the database size you will require for the installation.

The following table lists various DB instance classes by size of PeopleSoft Environment. Note that this table is provided only as a guideline. You should validate your individual environment against its actual usage.

DB Instance Class	Notes
db.m3.medium	Ideal for a small PeopleSoft demo/dev environment.
db.t2.medium	Ideal for a small PeopleSoft demo/dev environment.
db.m4.xlarge	Ideal for a medium PeopleSoft environment - <100 users.
db.r3.large	Ideal for a medium PeopleSoft environment - <100 users.

DB Instance Class	Notes
db.r3.xlarge	Ideal for a medium PeopleSoft environment - <1000 users.
db.r3.2xlarge	Ideal for a Medium PeopleSoft environment - <10,000 users.
db.r3.4xlarge	Ideal for a Large PeopleSoft environment - <50,000 users.
db.r3.8xlarge	Ideal for a very large PeopleSoft environment - <250,000 users.

Set Up the AWS Command Line Interface (Optional)

You can use either the AWS Management Console or the AWS CLI to perform the tasks described in this whitepaper. To use the AWS CLI, make sure that you have installed the CLI and that you have either an EC2 instance that has an AWS IAM role associated with it (recommended) or an access key ID and secret key.

Configuring a Highly Available Amazon RDS for Oracle Database Instance

We show you how to configure Amazon RDS for Oracle for use with PeopleSoft by using both the AWS Management Console and the AWS CLI. If you plan to use the CLI, we recommend that you follow the AWS Management Console procedure first because it provides context. The AWS CLI commands map directly to the tasks executed using the AWS Management Console.

Create Security Groups

A security group allows you to control access to specific inbound/outbound ports and to restrict access based on source address. You can represent source addresses either in CIDR notation (for example, where 192.168.1.1/32 represents a single IP address, and 192.168.1.0/24 represents 254 IP addresses under the 192.168.1.0 network) or in another security group.

In this paper, you will use an additional security group to define access to source addresses because this approach provides more flexibility and is easier to manage. It allows you to simply give network access from the PeopleSoft

Application security group to the PeopleSoft Demo database, instead of specifying each of the source IP addresses.

Using a security group ensures that only the required ports are exposed to specific hosts. That helps prevent, for example, a PROD PeopleSoft DB instance from accessing a TEST PeopleSoft DB instance.

Create the DB security groups using the procedures specified in the following table:

Security Group Name	Inbound Rules	Source	Notes
PeopleSoft-DEMO-DB-SG	CIDR/IP	IP addresses that can connect to the DB server.	Open only the security groups/ IP addresses / ranges that you want to allow to access the DB instance. You must specify the IP address using CIDR notation. (XX.YY.XX.YY/XX) You can specify the source security groups that you want to allow access from.
PeopleSoft-DEMO-APP-SG	EC2	Security group attached to your EC2 PeopleSoft DEMO instances.	Attach this security group to your EC2 instances; on the DB security group, allow access to the DB port from this security group.

Create the DEMO Database Security Group

You can use either the AWS Management Console or the CLI to create the TEST database OEM security group.

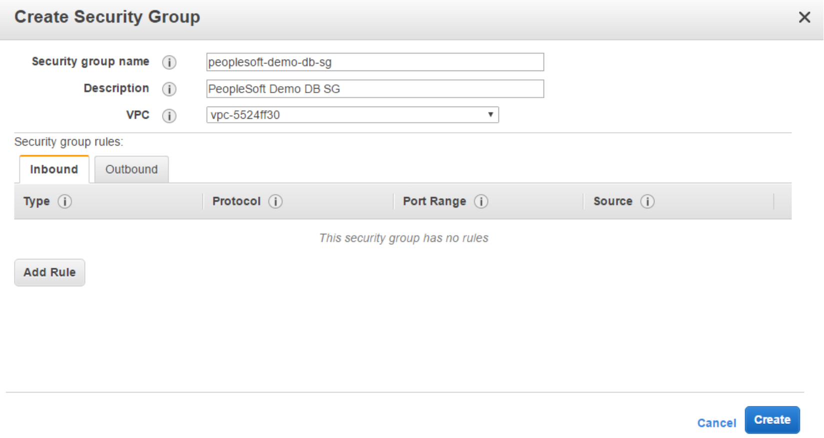
Create the PeopleSoft Demo DB Security Group (Console)

You can create the security group in the AWS Management Console by performing the following steps:

1. In the AWS Management Console, choose **Services**, choose **VPC**, choose **Security Groups**, and then choose **Create Security Group**.

2. Type a **Security group name** and **Description** as shown in Step 3. For this example, use “peoplesoft-demo-db-sg” for the **Security group name** and “PeopleSoft Demo DB SG” for the **Description**. Select the appropriate **VPC ID** for your account.

3. Choose **Create**.



Create Security Group

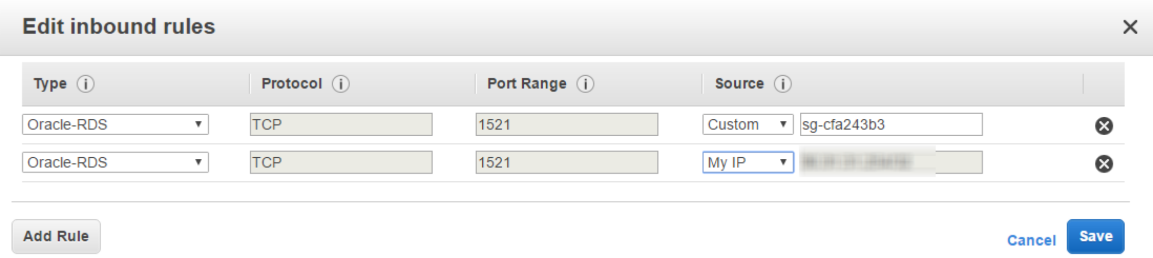
Security group name: peoplesoft-demo-db-sg
Description: PeopleSoft Demo DB SG
VPC: vpc-5524ff30

Security group rules:

Type	Protocol	Port Range	Source

This security group has no rules

4. Update the **Inbound Rules**, set the Type to “Oracle-RDS”, and specify the **Source** security group that you want to give access to. Choose **Save** to save the rule. Add another rule to allow DB access from your local network and/or IP address using the CIDR notation for your IP address range.



Edit inbound rules

Type	Protocol	Port Range	Source
Oracle-RDS	TCP	1521	Custom sg-cfa243b3
Oracle-RDS	TCP	1521	My IP [redacted]

Create the PeopleSoft Demo DB Security Group (CLI)

You can also create the security group using the CLI by executing the following commands:

```
ORALCRDSDBSG=$(aws ec2 create-security-group --group-name
peoplesoft-demo-db-sg --description "PeopleSoft Demo DB SG" --
vpc-id vpc-****** --output text)
```

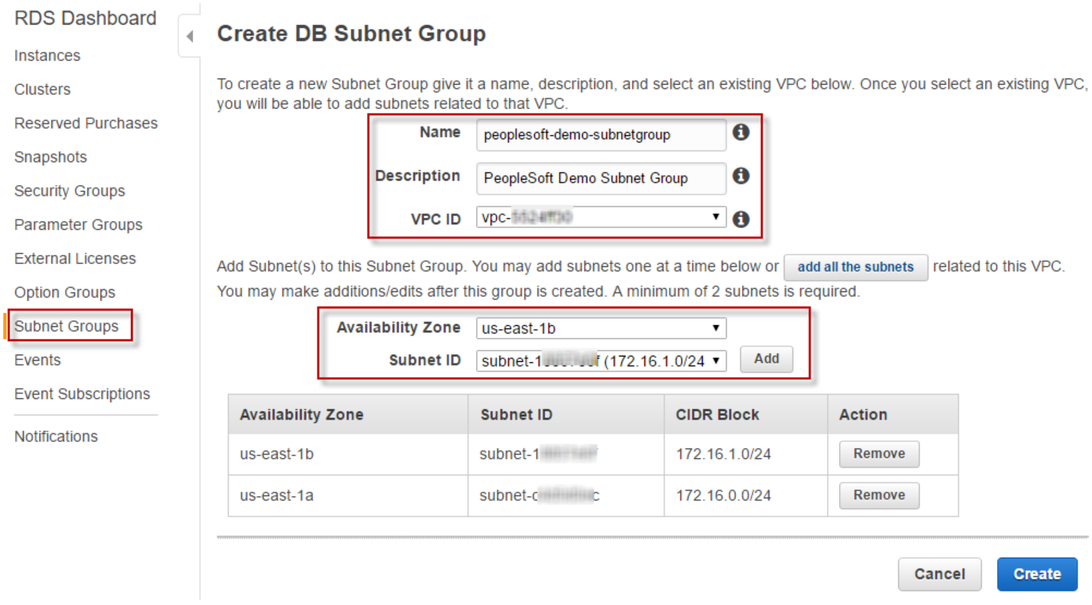
```
aws ec2 authorize-security-group-ingress --group-id
$ORALCRDSDBSG --protocol tcp --port 1521 --source-group sg-
******21

aws ec2 authorize-security-group-ingress --group-id
$ORALCRDSDBSG --protocol tcp --port 1521 --cidr xx.xx.xx.xx/xx

aws ec2 create-tags --resources $ORALCRDSDBSG --tags
"Key=Name,Value=PeopleSoft Demo DB SG"
```

Define a Database Subnet Group

Before you can create an RDS for Oracle Database instance, you must define a subnet group. A subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when creating DB instances using the CLI or API. For an RDS for Oracle Database, you must select two subnets, each in a different Availability Zone.

Define a DB Subnet Group (Console)

1. In the AWS Management Console, choose **Services**, choose **RDS**, choose **Subnet Groups**, and then choose **Create DB Subnet Group**.
2. Specify the following:
    - **Name** and **Description**
    - **VPC ID**: Choose the VPC that the database will be deployed to.
    - **Availability Zone**
    - **Subnet IDs**: Choose two subnets that are in different Availability Zones. If you have multiple subnets per Availability Zone, be sure to select the correct subnets.



Create DB Subnet Group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

Name	peoplesoft-demo-subnetgroup
Description	PeopleSoft Demo Subnet Group
VPC ID	vpc-[redacted]

Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Availability Zone	us-east-1b
Subnet ID	subnet-1[redacted]f (172.16.1.0/24

Availability Zone	Subnet ID	CIDR Block	Action
us-east-1b	subnet-1[redacted]	172.16.1.0/24	Remove
us-east-1a	subnet-c[redacted]c	172.16.0.0/24	Remove

Define a DB Subnet Group (CLI)

You can create the DB subnet group using the CLI as follows. Note that the subnet ID values are masked.

```
aws rds create-db-subnet-group --db-subnet-group-name
peoplesoft-demo-subnetgroup --db-subnet-group-description
"PeopleSoft Demo Subnet Group" --subnet-ids "subnet-
******e1" "subnet-******8e"
```

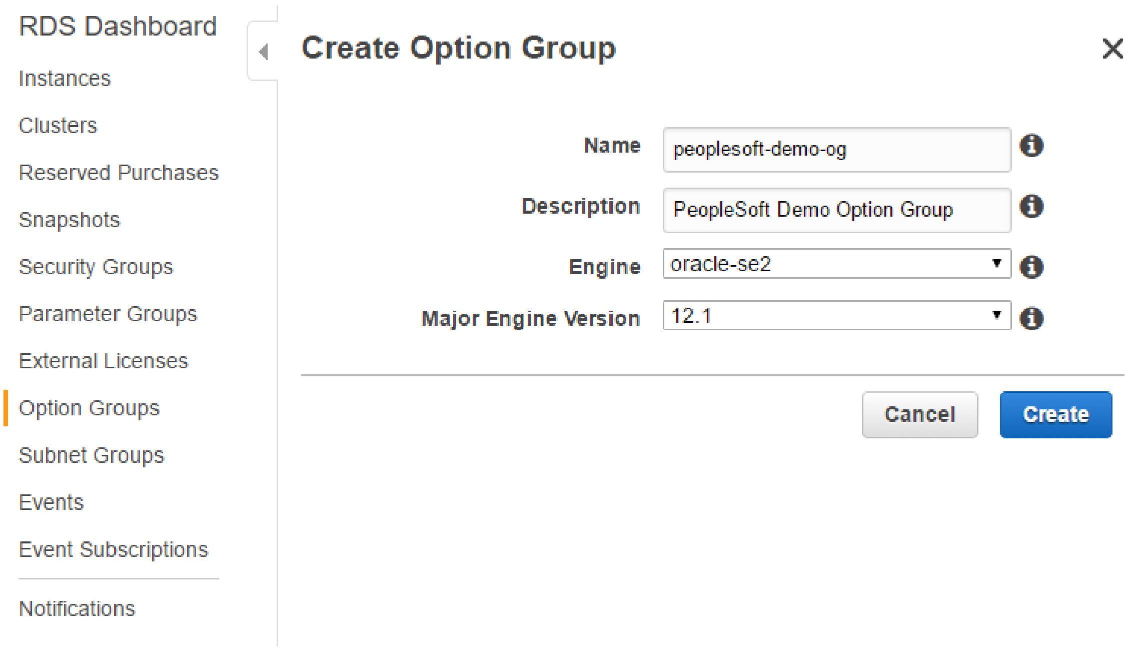
Create an Option Group

An option group provides additional feature definitions that you might want to add to your RDS for Oracle DB instance. Amazon RDS provides default option groups, but you cannot change them. For this reason, create a new option group so you can add a few features (options) later. You can assign an option group to multiple RDS for Oracle DB instances.

Create an Option Group (Console)

1. In the AWS Management Console, choose **Services**, choose **RDS**, choose **Option Groups**, and then choose **Create Option Group**.
2. Specify values for the following fields:

- **Name** and **Description**
- **Engine**: Choose the database edition you want to use in your RDS for Oracle DB instances.
- **Major Engine version**: Choose the database version.



Create Option Group

Name: peoplesoft-demo-og

Description: PeopleSoft Demo Option Group

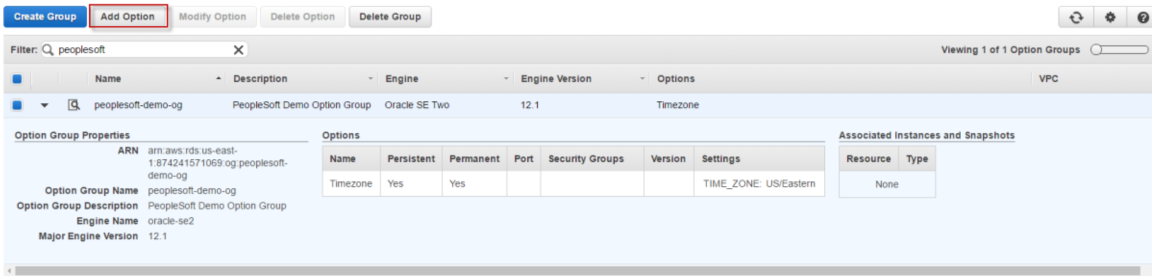
Engine: oracle-se2

Major Engine Version: 12.1

3. Specify the desired options.

   For your PeopleSoft Demo Option Group you need to specify the following options:

   - Timezone (Required)
   - Native Network Encryption or SSL (Optional, but recommended)
   - Statspack (Optional)
   - OEM and OEM Agent (Optional)



Filter: peoplesoft

Viewing 1 of 1 Option Groups

Name	Description	Engine	Engine Version	Options	VPC
peoplesoft-demo-og	PeopleSoft Demo Option Group	Oracle SE Two	12.1	Timezone	

Option Group Properties

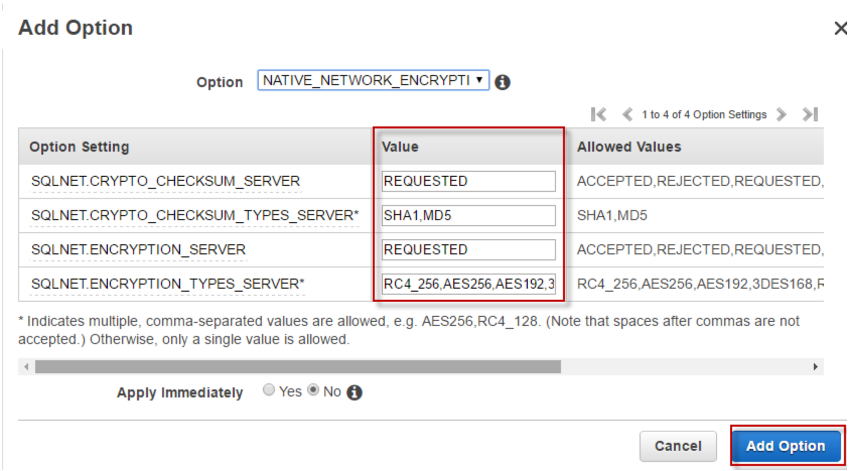
ARN	arn:aws:rds:us-east-1:874241571069:og:peoplesoft-demo-og
Option Group Name	peoplesoft-demo-og
Option Group Description	PeopleSoft Demo Option Group
Engine Name	oracle-se2
Major Engine Version	12.1

Options

Name	Persistent	Permanent	Port	Security Groups	Version	Settings
Timezone	Yes	Yes				TIME_ZONE: US/Eastern

Associated Instances and Snapshots

Resource	Type
None	


Add Option

Option: NATIVE_NETWORK_ENCRYPTI

1 to 4 of 4 Option Settings

Option Setting	Value	Allowed Values
SQLNET.CRYPTO_CHECKSUM_SERVER	REQUESTED	ACCEPTED,REJECTED,REQUESTED,
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER*	SHA1,MD5	SHA1,MD5
SQLNET.ENCRYPTION_SERVER	REQUESTED	ACCEPTED,REJECTED,REQUESTED,
SQLNET.ENCRYPTION_TYPES_SERVER*	RC4_256,AES256,AES192,3	RC4_256,AES256,AES192,3DES168,F

* Indicates multiple, comma-separated values are allowed, e.g. AES256,RC4_128. (Note that spaces after commas are not accepted.) Otherwise, only a single value is allowed.

Apply Immediately ○ Yes ● No

Create an Option Group (CLI)

To create the option group, execute the following commands:

```
aws rds create-option-group --option-group-name peoplesoft-
demo-og --engine-name oracle-se2 --major-engine-version
12.1 --option-group-description "PeopleSoft Demo Option
Group"
```

Create a Parameter Group

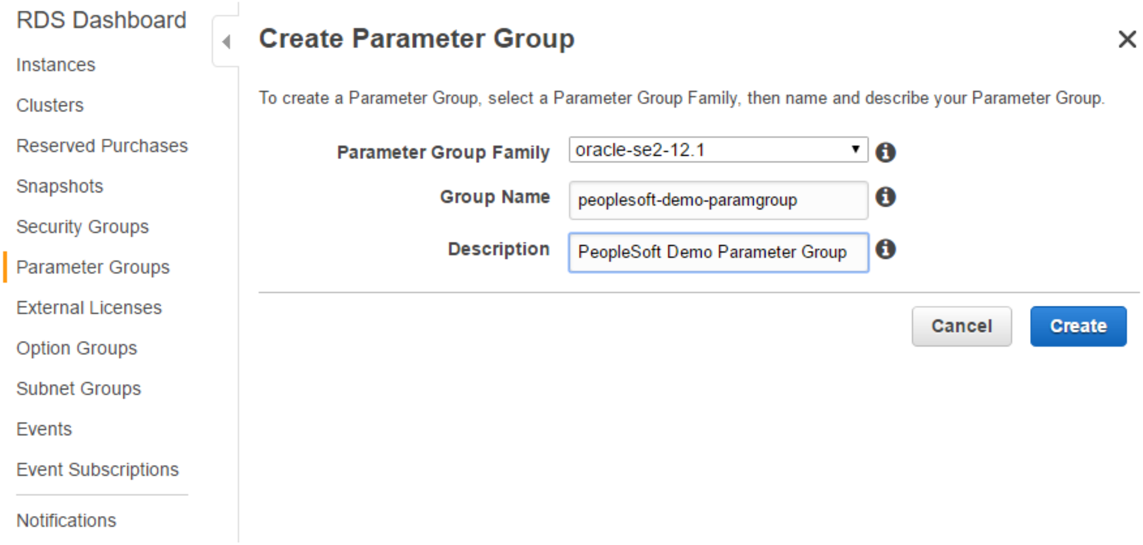
A parameter group is a container of parameters that are applied to one or more database instances. When you launch an Amazon RDS for Oracle DB instance, a default parameter group is created. You can't update the default parameter group, so you need to create a new parameter group for the demo environment.

We recommend that you create a new parameter group, even if you don't need to customize any parameters at this point. It is highly likely that you will eventually need to customize some of the parameters.

We also recommend that you consider how you will reuse the parameter groups among multiple RDS for Oracle DB instances. Your need to reuse parameters depends on how the DB parameters could differ among the different DB instances. For a PeopleSoft deployment it is recommended that you use a unique Parameter Group for each Environment (DEV, TST, PROD), since parameters may need to be modified to suit a particular use case and/or provide you the ability to test a change in the configuration prior to applying it to a new environment.

Create a Parameter Group (Console)

1. In the AWS Management Console, choose **Services**, choose **RDS**, choose **Parameter Groups**, and then choose **Create Parameter Group**.
2. Define:
    - **Parameter Group Family**: Specify the database edition you want to use in your RDS for Oracle DB instances.
    - **Group Name** and **Description:** Give your new a group an identifier of your choice and a description.



Create Parameter Group

To create a Parameter Group, select a Parameter Group Family, then name and describe your Parameter Group.

Parameter Group Family: oracle-se2-12.1

Group Name: peoplesoft-demo-paramgroup

Description: PeopleSoft Demo Parameter Group

Create a Parameter Group (CLI)

Create the parameter group by executing the following commands:

```
aws rds create-db-parameter-group --db-parameter-group-name
peoplesoft-demo-paramgroup --db-parameter-group-family
oracle-se2-12.1 –description “PeopleSoft Demo Parameter
Group"
```

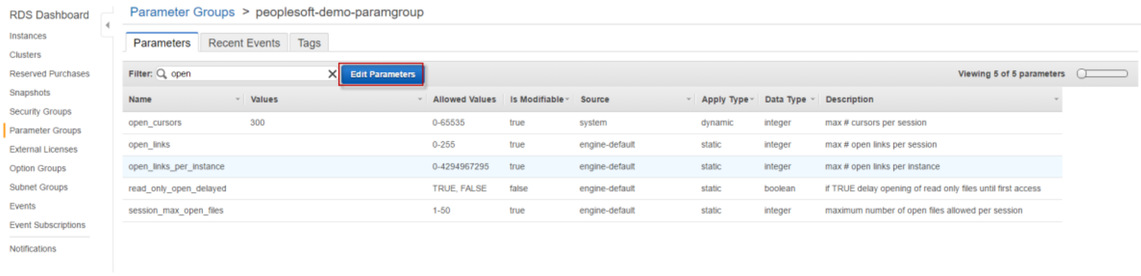
Change a Parameter

For PeopleSoft there are some recommended values when installing the DB. Set the following Parameters in the Parameter Group:

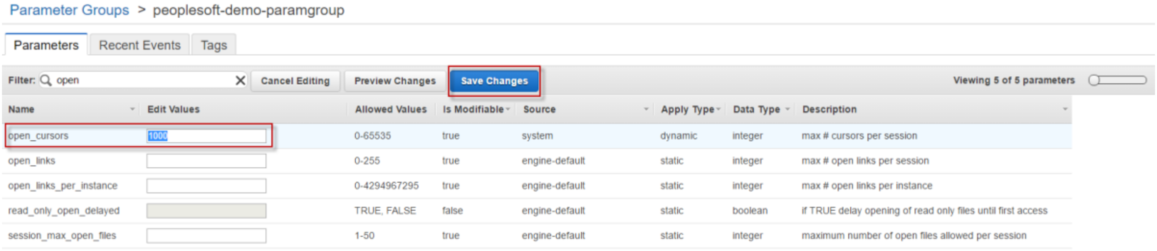
- Open Cursors = 1000
- DB Block Size = automatically set by AWS RDS (Creation of nonstandard block size tablespaces and setting DB_nK_CACHE_SIZE parameters is supported.)
- DB Files = 1021, but can leave default from AWS RDS
- NLS Length Semantics = CHAR for Unicode, BYTE for non-Unicode
- Memory Target = defaults to {DBInstanceClassMemory*3/4}; change if you have a specific requirement, otherwise the default is fine.

Change a Parameter (Console)

1. Choose the Parameter Group you created.
2. Choose **Edit Parameters**.
3. Search for the **Parameter** you want to edit.
4. Change the number of open_cursors to **1000**, and then choose **Save Changes**.
5. Repeat step 3-4 for each parameter you want to edit.



Name	Values	Allowed Values	Is Modifiable	Source	Apply Type	Data Type	Description
open_cursors	300	0-65535	true	system	dynamic	integer	max # cursors per session
open_links		0-255	true	engine-default	static	integer	max # open links per session
open_links_per_instance		0-4294967295	true	engine-default	static	integer	max # open links per instance
read_only_open_delayed		TRUE, FALSE	false	engine-default	static	boolean	if TRUE delay opening of read only files until first access
session_max_open_files		1-50	true	engine-default	static	integer	maximum number of open files allowed per session



Name	Edit Values	Allowed Values	Is Modifiable	Source	Apply Type	Data Type	Description
open_cursors	1000	0-65535	true	system	dynamic	integer	max # cursors per session
open_links		0-255	true	engine-default	static	integer	max # open links per session
open_links_per_instance		0-4294967295	true	engine-default	static	integer	max # open links per instance
read_only_open_delayed		TRUE, FALSE	false	engine-default	static	boolean	if TRUE delay opening of read only files until first access
session_max_open_files		1-50	true	engine-default	static	integer	maximum number of open files allowed per session

Change a Parameter (CLI)

Because the `open_cursors` parameter is a static parameter, you need to specify the method as `pending-reboot`.

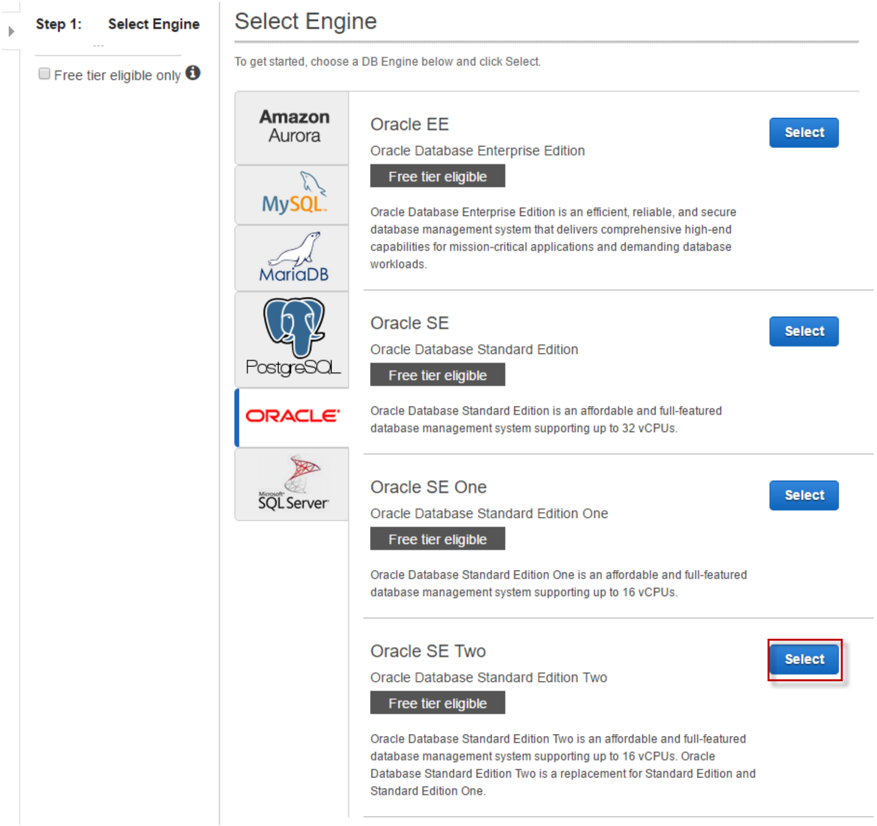
```
aws rds modify-db-parameter-group --db-parameter-group-name
peoplesoft-demo-paramgroup --parameters
"ParameterName=open_cursors,ParameterValue=1000,ApplyMethod=
pending-reboot"
```

Create the Database Instance

Now create an Oracle SE Two Database to see how simple it is to create a highly available database across two Availability Zones. Depending on your SLA requirements, you might want to run the database in a single Availability Zone. Running it in multiple Availability Zones increases cost.

Create the DB Instance (Console)

1. In the RDS console, select the **Oracle SE Two** Database engine.



Select Engine

To get started, choose a DB Engine below and click Select.

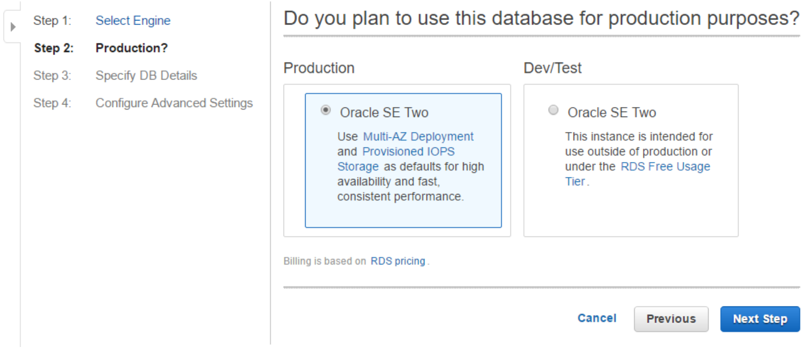
Oracle EE
Oracle Database Enterprise Edition
Free tier eligible
Oracle Database Enterprise Edition is an efficient, reliable, and secure database management system that delivers comprehensive high-end capabilities for mission-critical applications and demanding database workloads.

Oracle SE
Oracle Database Standard Edition
Free tier eligible
Oracle Database Standard Edition is an affordable and full-featured database management system supporting up to 32 vCPUs.

Oracle SE One
Oracle Database Standard Edition One
Free tier eligible
Oracle Database Standard Edition One is an affordable and full-featured database management system supporting up to 16 vCPUs.

Oracle SE Two
Oracle Database Standard Edition Two
Free tier eligible
Oracle Database Standard Edition Two is an affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

2. Choose **Dev/Test**, which is the Single-AZ deployment option. Choose **Production** if you are going to create a Production DB instance.



Step 1: Select Engine
Step 2: **Production?**
Step 3: Specify DB Details
Step 4: Configure Advanced Settings

Do you plan to use this database for production purposes?

Production

● Oracle SE Two

Use Multi-AZ Deployment and Provisioned IOPS Storage as defaults for high availability and fast, consistent performance.

Dev/Test

○ Oracle SE Two

This instance is intended for use outside of production or under the RDS Free Usage Tier.

Billing is based on RDS pricing.

3. Specify the DB details:

- **License model**: For Oracle SE Two, choose either **bring-your-own-license** or **license-included**.

- **DB Engine Version**: Choose **12.1.0.2.v6**.

- **DB Instance Class**: Because the load is just for test purposes for one PeopleSoft Demo, choose a relatively small DB instance class. You can change the DB instance class at any point, but it requires restarting the DB instance.

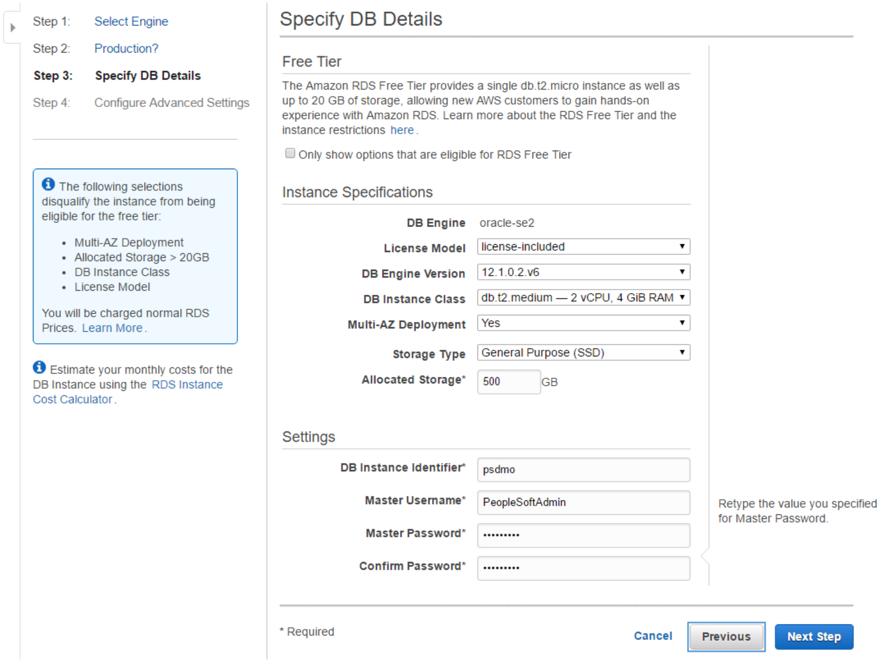
- **Multi-AZ Deployment**: Choose **Yes** so that you can have a second standby instance running on a second Availability Zone.

- **Storage Type**: Because you selected Multi-AZ Deployment, the **Storage Type** is set to **Provisioned IOPS** by default, which provides more consistent throughput but at a higher cost. Because we are not too interested in consistent performance for the dev/test environment, change it to **General Purpose (SSD)**. We suggest you run performance tests to make sure that the selected option meets your requirements.

- **Allocated Storage**: Allocate **500 GB**. Note that for the General Purpose SSD storage type, the number of IOPS is proportional to volume size (it's a 3:1 ratio).

- **DB Instance Identifier**: Enter the DB instance identifier. Amazon RDS uses this identifier to define the database hostname. In our example we use **psdmo**.

- **Master Username**: This is similar to a SYS user, but with fewer privileges because Amazon RDS does not allow you to use either a SYS user or the SYSDBA role.
- **Master Password**: Enter the password for your master user.



Step 1: Select Engine
Step 2: Production?
Step 3: Specify DB Details
Step 4: Configure Advanced Settings

The following selections disqualify the instance from being eligible for the free tier:

- Multi-AZ Deployment
- Allocated Storage > 20GB
- DB Instance Class
- License Model

You will be charged normal RDS Prices. Learn More.

Estimate your monthly costs for the DB Instance using the RDS Instance Cost Calculator.

Specify DB Details

Free Tier

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions here.

□ Only show options that are eligible for RDS Free Tier

Instance Specifications

DB Engine	oracle-se2
License Model	license-included
DB Engine Version	12.1.0.2.v6
DB Instance Class	db.t2.medium — 2 vCPU, 4 GiB RAM
Multi-AZ Deployment	Yes
Storage Type	General Purpose (SSD)
Allocated Storage*	500 GB

Settings

DB Instance Identifier*	psdmo
Master Username*	PeopleSoftAdmin
Master Password*	•••••••••
Confirm Password*	•••••••••

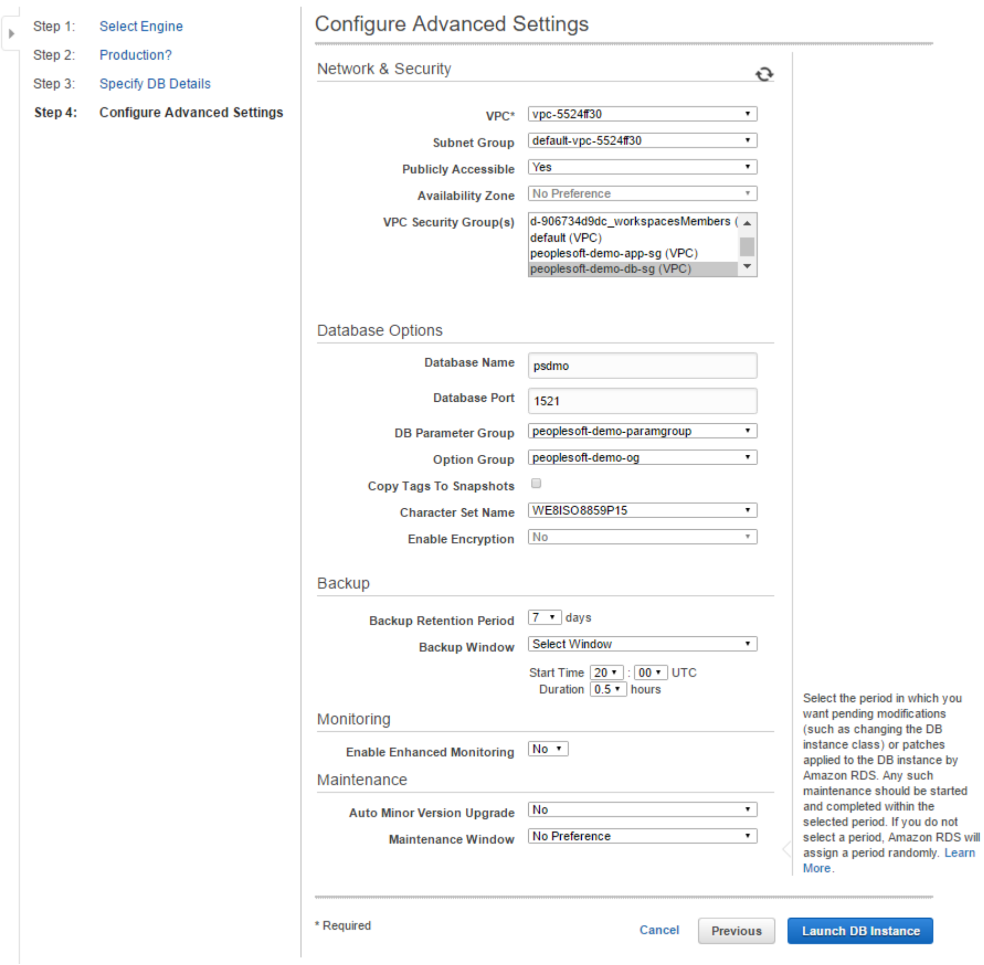
Retype the value you specified for Master Password.

* Required

4. Specify the DB instance advanced settings:

- **VPC**: Choose the VPC where the database will be deployed.
- **Subnet Group**: Choose the subnet group you created earlier. The DB instance is deployed against the subnets associated with the subnet group.
- **Publicly Available**: Allows external access to the database. Usually you should choose **No** because you want to give access only to addresses in your private subnets.
- **Availability Zone**: The option is available when you deploy a DB instance in a single Availability Zone. This option is disabled for Multi-AZ deployments.
- **VPC Security Group**: Choose the security group that will be associated with your DB instance. This security group provides access to the database listener. You created this security group **peoplesoft-demo-db-sg** earlier.

- **Database Name**: Choose the Database Service Name, which your database clients will use to connect. In our example, we used **psdmo**.
- **Database Port**: Choose the TCP port that the database listener listens on. In our example, we choose **1521**, which is a default port for Oracle.
- **DB Parameter Group**: The database engine parameters. Choose the **peoplesoft-demo-paramgroup** DB parameter group that you created earlier.
- **Option Group**: The features that will be enabled in the database. Choose the **peoplesoft-demo-og** option group, which you created earlier.
- **Copy Tags to Snapshots**: When this option is enabled, Amazon RDS copies any tag associated with your DB instance to the database snapshots. This is useful for tracking usage and cost, so select it.
- **Character Set Name**: Choose the character set for your database. In our example, we use **WE8ISO8859P15**, a non-Unicode database. Use the character set that is required for your installation of PeopleSoft.
- **Enable Encryption**: If you want your database data to be encrypted, choose **Enable Encryption**.
- **Backup Retention Period**: Set the backup retention period in days (maximum of 35) for your database.
- **Backup Window**: Set the backup window for your daily backup.
- **Enable Enhanced Monitoring**: Select **Yes** to enable enhanced DB Monitoring. In our example, we selected **No**.
- **Auto Minor Version Upgrade**: Select **Yes** to enable automatic minor version upgrades.
- **Maintenance Window:** Set the timing for the minor maintenance window.



Configure Advanced Settings

Network & Security

VPC* vpc-5524ff30
Subnet Group default-vpc-5524ff30
Publicly Accessible Yes
Availability Zone No Preference
VPC Security Group(s) d-906734d9dc_workspacesMembers (
default (VPC)
peoplesoft-demo-app-sg (VPC)
peoplesoft-demo-db-sg (VPC)

Database Options

Database Name psdmo
Database Port 1521
DB Parameter Group peoplesoft-demo-paramgroup
Option Group peoplesoft-demo-og
Copy Tags To Snapshots □
Character Set Name WE8ISO8859P15
Enable Encryption No

Backup

Backup Retention Period 7 days
Backup Window Select Window
Start Time 20 : 00 UTC
Duration 0.5 hours

Monitoring

Enable Enhanced Monitoring No

Maintenance

Auto Minor Version Upgrade No
Maintenance Window No Preference

Select the period in which you want pending modifications (such as changing the DB instance class) or patches applied to the DB instance by Amazon RDS. Any such maintenance should be started and completed within the selected period. If you do not select a period, Amazon RDS will assign a period randomly. Learn More.

* Required

5. Launch the DB instance.

Create the DB Instance (CLI)

Execute the following command:

```
aws rds create-db-instance --db-name psdmo --db-
instanceidentifier psdmo --db-instance-class db.t2.medium -
engine oracle-se2  --master-username PeopleSoftAdmin --
masteruser-password ******* --vpc-security-group-ids sg-******5a
--db-subnet-group-name peoplesoft-demo-subnetgroup --db-
parameter-group-name peoplesoft-demo-paramgroup --port
1521 --multi-az --engine-version 12.1.0.2.v6 --license-
model license-included  --option-group-name peoplesoft-
demo-og --character-set-name WE8ISO8859P15 --no-publicly-
```

```
accessible --storage-type gp2 --allocatedstorage 500 --
copy-tags-to-snapshot
```

Create a DNS Alias for the Database Instance

When you create an RDS for Oracle DB instance, Amazon RDS creates a unique DNS hostname for your instance (e.g., psdmo.c6jc3rya3ntd.us-east-1.rds.amazonaws.com:1521). You can use that hostname to connect to the database. However, you have no control over the hostname, so you end up with a database URL that is not easy to remember.

In addition, you might at some point need to restore the database from a snapshot. For example, you would need to do so if an operator makes a mistake in manipulating the data, or if a bug in your application corrupts the data. But you can’t restore a snapshot to an existing RDS DB instance. When you restore a database from a snapshot, Amazon RDS creates a new DB instance and generates a new hostname.

To avoid affecting existing applications and having to update their database endpoints, we strongly recommend that you create a DNS alias for your RDS DB instance. Depending on your architecture, you might register the DNS alias either in your corporate DNS server running on premises, or in a DNS server running in AWS.

We will show you how to register a DNS alias in AWS as a private hosted zone using Amazon Route 53. When you use a private hosted zone, only hosts in your VPC can resolve the DNS names for your database. (There is a way to extend the name resolution outside of the VPC, but that's beyond the scope of this whitepaper.)

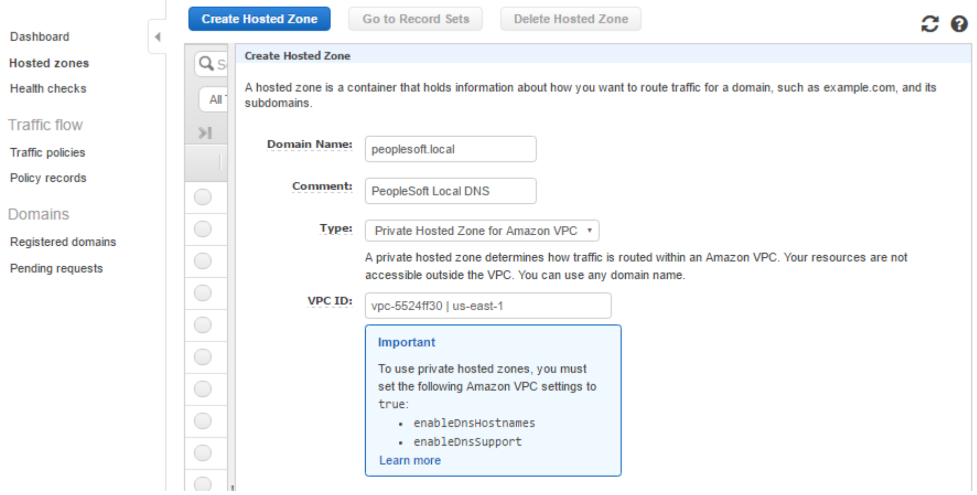
Create an Amazon Route 53 Private Hosted Zone

Create a private hosted name if you haven't yet already done so.

Create an Amazon Route 53 Private Zone (Console)

1. In the AWS Management Console, choose **Services**, choose **Route 53**, choose **Hosted zones**, and then choose **Create Hosted Zone**.
2. Provide the details of your hosted zone:

- **Domain Name**: Type the private domain name. We used **peoplesoft.local** in our example.
- **Type**: Choose **Private Hosted Zone for Amazon VPC**.
- **VPC ID**: Choose the ID of the VPC used by your PeopleSoft infrastructure in AWS.



Create Hosted Zone

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain Name: peoplesoft.local

Comment: PeopleSoft Local DNS

Type: Private Hosted Zone for Amazon VPC

A private hosted zone determines how traffic is routed within an Amazon VPC. Your resources are not accessible outside the VPC. You can use any domain name.

VPC ID: vpc-5524ff30 | us-east-1

Important

To use private hosted zones, you must set the following Amazon VPC settings to `true`:

- `enableDnsHostnames`
- `enableDnsSupport`

Learn more

Create an Amazon Route 53 Private Zone (CLI)

Execute the following command:

```
aws route53 create-hosted-zone --name peoplesoft.local --
vpc '{"VPCRegion":"us-east-1", "VPCId": "vpc-******21"}' --
hosted-zone-config '{"PrivateZone": true}' --caller-
reference 112017
```

You then get a confirmation similar to the following:

```
{
   "ChangeInfo": {
      "Status": "PENDING",
      "SubmittedAt": "2016-06-28T00:22:08.317Z",
      "Id": "/change/C2R8IM38ABCDEF"
   },
   "HostedZone": {
      "ResourceRecordSetCount": 2,
      "CallerReference": "112017",
      "Config": {
            "PrivateZone": true
      },
      "Id": "/hostedzone/Z234334ABCDEF",
      "Name": "peoplesoft.local."
   },
   "Location": "https://route53.amazonaws.com/2013-04-
01/hostedzone/Z234334ABCDEF",
   "VPC": {
      "VPCId": "vpc-******21",
      "VPCRegion": "us-east-1"
   }
}
```

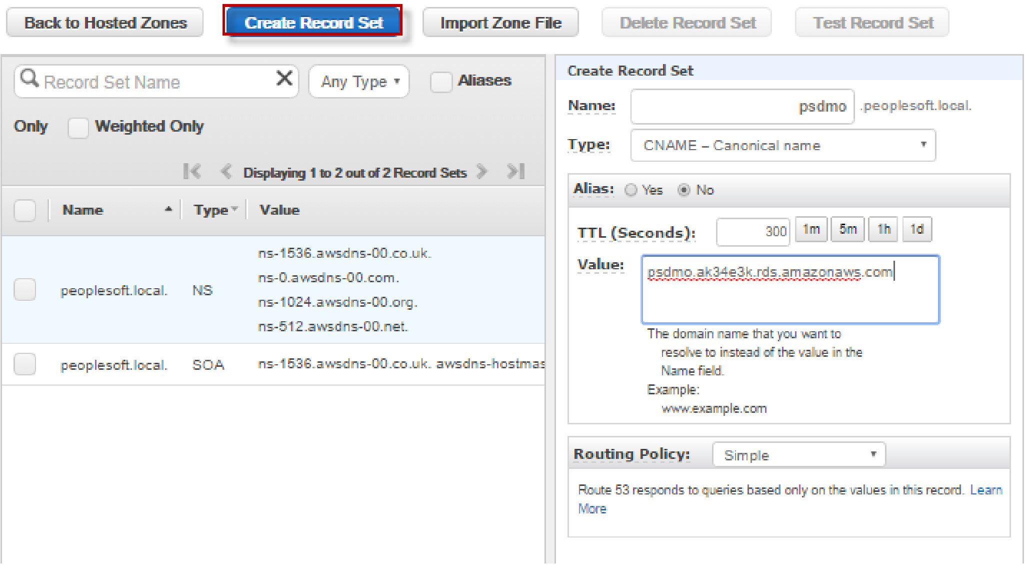
Note the HostedZone Id because you will need it to create the record sets in the next section. In the snippet, the HostedZone Id is: Z234334ABCDEF.

Create a DNS Alias

By creating a DNS alias, you can easily manage your database's endpoint, and you don't have to change your existing application's code.

Create a DNS Alias (Console)

1. Choose **Create Record Set**, and then provide the following details:
    - **Name**: The fully qualified domain name, which in our example is **psdmo**. The value you type will be prepended to the domain name. In our example, it is **psdmo.peoplesoft.local**.
    - **Type**: Choose **CNAME**.
    - **Alias**: Choose **No**.
    - **TTL (Seconds)**: Type **300**.
    - **Value**: The RDS instance hostname (don't add the port information, 1521).



Name	Type	Value
peoplesoft.local.	NS	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.
peoplesoft.local.	SOA	ns-1536.awsdns-00.co.uk. awsdns-hostmas

Create Record Set

Name: psdmo .peoplesoft.local.

Type: CNAME – Canonical name

Alias: ○ Yes ● No

TTL (Seconds): 300 1m 5m 1h 1d

Value: psdmo.ak34e3k.rds.amazonaws.com

The domain name that you want to resolve to instead of the value in the Name field.
Example:
www.example.com

Routing Policy: Simple

Route 53 responds to queries based only on the values in this record. Learn More

Create a DNS Alias (CLI)

Execute the following command:

```
aws route53 change-resource-record-sets --hosted-zone-id
Z234334ABCDEF --change-batch '{"Changes": [{"Action":
"CREATE", "ResourceRecordSet": {"Name":
"psdmo.peoplesoft.local","Type": "CNAME", "TTL": 300,
"ResourceRecords": [{"Value":
"psdmo.ak34e3k.rds.amazonaws.com"}]}}]}'
```

After creating a DNS alias, you can connect successfully to the database using the following URL:

```
psdmo.peoplesoft.local:1521/psdmo
```

Running the PeopleSoft DB Creation Scripts

Now that you have successfully created the database, you can proceed to the normal steps in the PeopleSoft DB Creation Process. The following steps are excerpts from the PeopleTools 8.55 Installation for Oracle Guide. Please use the appropriate PeopleTools install guide for your installation and make note of the following changes in the Manual Installation steps.

Editing the Database Scripts

The first step is to modify the following delivered Database Creation scripts as required in the PeopleSoft installation.

Create DB SQL

This script doesn't need to be run because you already created the database.

UTL Space Script

Edit this script to remove the file path references for creating the PSDEFAULT tablespace.

This is what the default file looks like:

```
REM * Create a temporary tablespace for database users.
REM *
CREATE TEMPORARY TABLESPACE PSTEMP
TEMPFILE                '<drive>:\oradata\<SID>\pstemp01.dbf'
SIZE 300M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
;
REM * Create a tablespace for database users default
tablespace.
REM *
CREATE TABLESPACE       PSDEFAULT
DATAFILE                '<drive>:\oradata\<SID>\psdefault.dbf'
SIZE 100M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO
```

```
;

spool off
```

This is what the edited file should look like:

```
set echo on
spool C:\HRDMO\scripts\nt\edited\utlspace.log
REM * Create a temporary tablespace for database users.
REM *
CREATE TEMPORARY TABLESPACE PSTEMP TEMPFILE EXTENT MANAGEMENT
LOCAL UNIFORM SIZE 128K;
REM * Create a tablespace for database users default
tablespace.
REM *
CREATE TABLESPACE PSDEFAULT EXTENT MANAGEMENT LOCAL
AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO;
spool off
```

Application-Specific Table Space Creation

Depending on the PeopleSoft Application you are installing, you will need to modify the appropriate tablespace file to remove the file path references. Please refer to the PeopleTools Installation for Oracle for details on the DDL scripts that are appropriate for your application.

The following is an example of the delivered XXDDL.sql structure:

```
CREATE TABLESPACE AMAPP DATAFILE
'<drive>:\oradata\<SID>\amapp.dbf' SIZE 108M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO
/
```

You need to edit the file to remove the Datafile criteria and set it to auto-allocate space:

```
CREATE TABLESPACE AMAPP EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO
/
```

DB Owner Script

The DB Owner script must be edited to apply the Connect User ID and Password for the database.

PS Roles Script

No edits to this script are necessary.

PS Admin Script

Update the Access ID and default tablespace as normal.

Connect Script

Update the Connect ID and Password as normal.

Create and Run Data Mover Import Scripts

Follow the PeopleSoft Installation Guide for creating and running the Data Mover Import scripts for your PeopleSoft Application.

Conclusion

We have now completely set up a PeopleSoft demo database that is fully managed on Amazon RDS and ready to perform. This whitepaper described how to configure RDS for Oracle as a backend database for an Oracle PeopleSoft Enterprise demo application. By using these procedures, you can use RDS for Oracle to set up and operate many different PeopleSoft application databases. As a result, you can easily run your PeopleSoft applications on an Amazon RDS for Oracle Database.

References

- PeopleTools 8.55 Installation for Oracle
- Amazon Web Services API Reference

Contributors

The following individuals and organizations contributed to this document:

- David Brunet, VP Research and Development, DLZP Group
- Muhammed Sajeed, PeopleSoft Architect, DLZP Group
- Yoav Eilat, Senior Product Marketing Manager, AWS
- Tsachi Cohen, Software Development Manager, AWS
- Michael Barras, Senior Database Engineer, AWS

Document Revisions

Date	Description
March 2017	First publication

Notes

[1] https://docs.aws.amazon.com/general/latest/gr/rande.html

[2] https://docs.aws.amazon.com/AmazonVPC/latest/GettingStartedGuide/GetStarted.html

[3] https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_IAM.html

[4] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/UsingIAM.html

[5] https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAM.AccessControl.IdentityBased.html

[6] https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/auth-and-access-control.html