

# 데이터를 AWS로!

## ORACLE-RDS MIGRATION

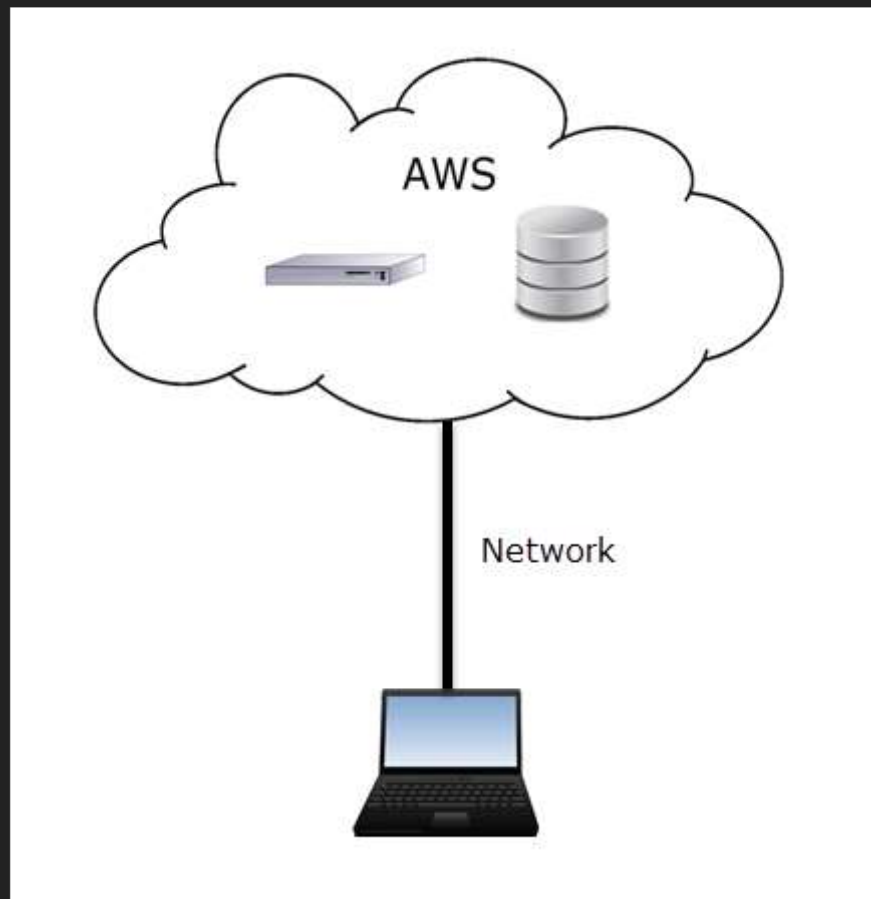
Created by [Jongwon](#)

# AWS?

Amazon Web Services

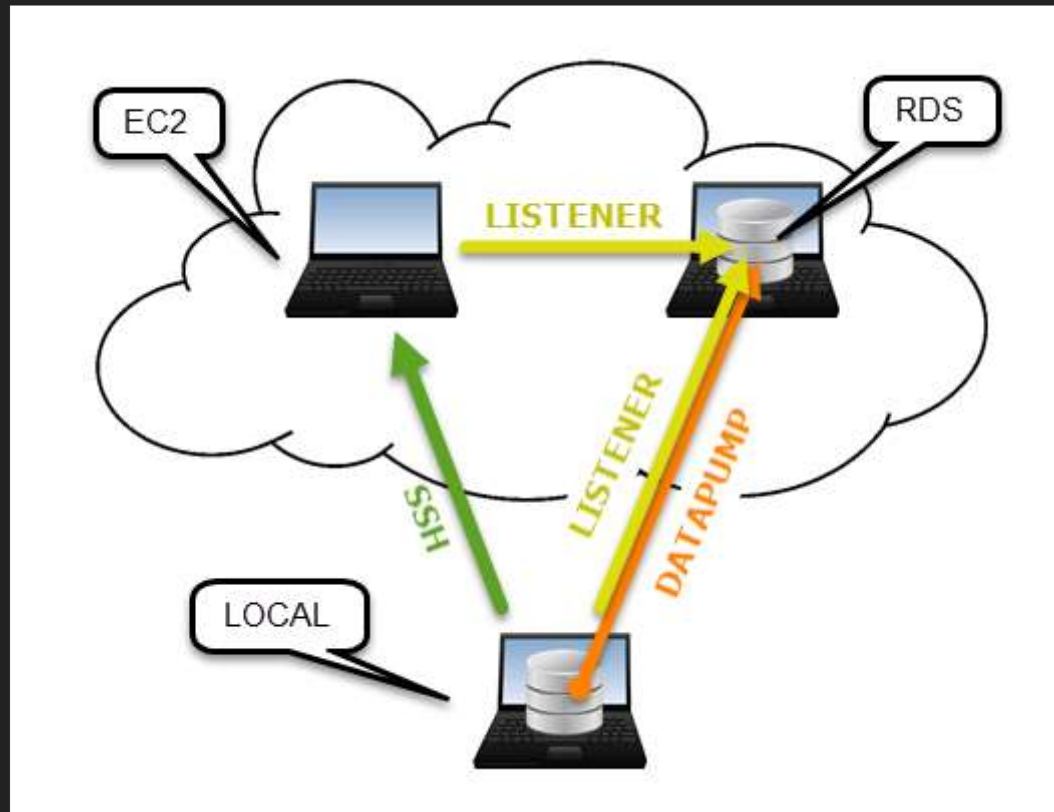
클라우드 컴퓨팅의 대표 주자

# CONCEPT



네트워크만 있으면 어디서나 접속!

# BUILDING PLAN

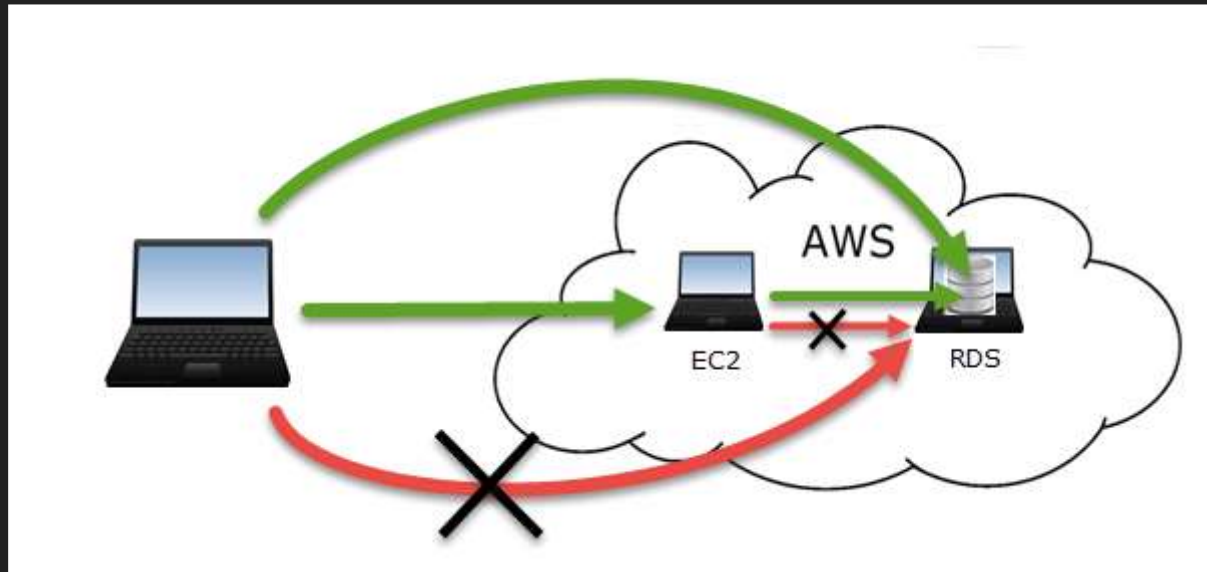


# 인스턴스 생성 및 연결

블로그에 정리해 봤습니다!

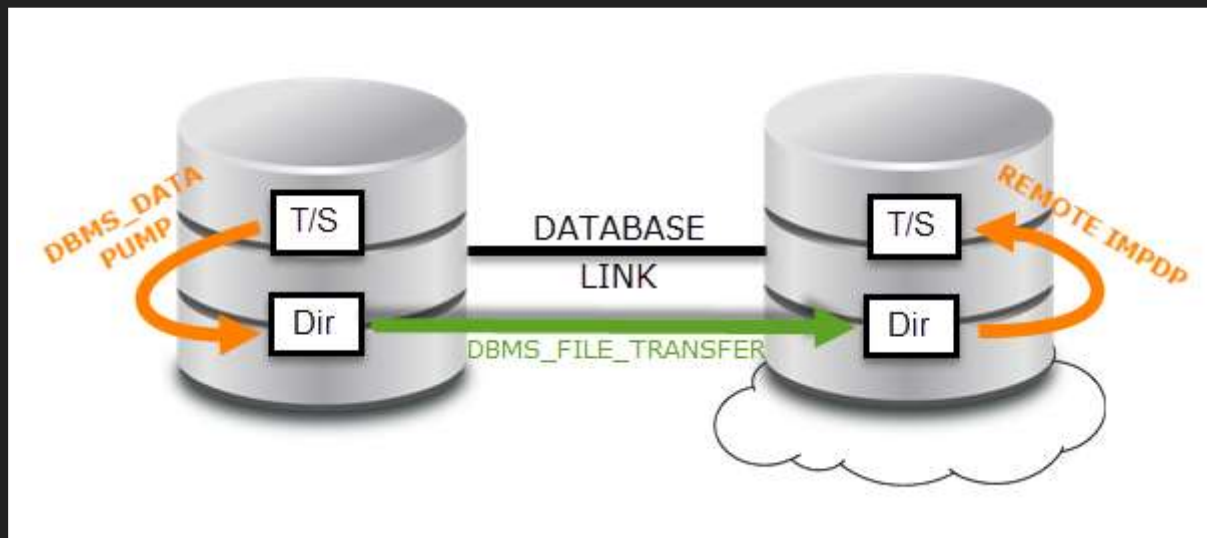
- [EC2\(Amazon Linux\)](#)
- [RDS\(Oracle EE\)](#)

# CONNECTION PROBLEM



RDS 인스턴스가 설치되어있는 운영체제로는  
접근할 수 없습니다.

# POINT OF VIEW



Database Link의 External Directory를 통해 이동합니다.

# DATA PUMP STRATEGY

1. 원본 데이터베이스의 사용자에게 권한 부여
2. DBMS\_DATAPUMP를 사용하여 덤프파일 생성
3. 대상 DB 인스턴스의 Database Link 생성
4. DBMS\_FILE\_TRANSFER로 내보낸 덤프파일을  
RDS 인스턴스로 복사
5. 대상 인스턴스에 필요한 테이블 공간 생성
6. 덤프파일을 RDS 인스턴스의 데이터베이스로 가져오기
7. 뒤처리하기



# PREREQUISITE

1. DBMS\_FILE\_TRANSFER 패키지에 대한 실행 권한이 있어야 합니다.
2. 대상 DB 인스턴스가 버전 11.2.0.2.v6 이상이어야 합니다.
3. 원본 DB 인스턴스의 DATA\_PUMP\_DIR 디렉터리에 대한 쓰기 권한이 있어야 합니다.
4. 원본 인스턴스 및 대상 DB 인스턴스에 덤프 파일을 저장할 수 있는 충분한 스토리지 공간이 있는지 확인해야 합니다.

# 원본 데이터베이스의 사용자에게 권한 부여

```
-- Tablespace 생성
```

```
SYS> create tablespace DPTS
```

```
2 datafile '$ORACLE_BASE/oradata/orcl/dpts01.dbf'
```

```
3 size 200M;
```

```
-- User를 생성하고 Default Tablespace로 지정
```

```
SYS> create user DPUSER identified by dpuser default tablespace DPTS;
```

```
SYS> grant create session, create table to DPUSER;
```

```
SYS> alter user DPUSER quota 100M on users;
```

```
SYS> grant read, write on directory data_pump_dir to DPUSER;
```

```
SYS> grant execute on dbms_datapump to DPUSER;
```

```
-- Test Table
```

```
DPUSER> create table test as select * from all_objects;
```

# DBMS\_DATAPUMP를 사용하여 덤프파일 생성

```
-- DPUSER
DECLARE
n1 NUMBER;
BEGIN
n1 := DBMS_DATAPUMP.open(operation=>'EXPORT', job_mode=>'SCHEMA', job_name=>'DPUSER');
DBMS_DATAPUMP.add_file(handle=>n1, filename=>'test.dmp', directory=>'TEST');
DBMS_DATAPUMP.add_file(handle=>n1, filename=>'test.log', directory=>'TEST');
DBMS_DATAPUMP.metadata_filter(n1, 'SCHEMA_EXPR', 'IN (''DPUSER'')');
DBMS_DATAPUMP.start_job(n1);
END;
/
```

# 대상 DB 인스턴스의 DATABASE LINK 생성

권한을 제한하기 위해 Dummy 유저를 RDS에 생성합니다.

```
-- RDS에서 Dummy 유저 생성
```

```
RDS> create user LINKDP identified by linkdp;
```

```
RDS> grant create session to LINKDP;
```

```
RDS> grant read, write on directory data_pump_dir to LINKDP;
```

```
-- 원본 Database에서 Database Link 생성
```

```
DPUSER> create database link to_rds connect to LINKDP identified by l  
2      using '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=<dns or ip ac
```

# DBMS\_FILE\_TRANSFER로 내보낸 덤프파일을 RDS 인스턴스로 복사

```
-- SYS 또는 DBMS_FILE_TRANSFER 실행권한이 있는 DPUSER
BEGIN
DBMS_FILE_TRANSFER.PUT_FILE(
source_directory_object      => 'DATA_PUMP_DIR',
source_file_name             => 'test.dmp',
destination_directory_object => 'DATA_PUMP_DIR',
destination_file_name        => 'test_copied.dmp',
destination_database         => 'TO_RDS'
);
END;
/
```

# 대상 인스턴스에 필요한 테이블 공간 생성

물리적인 파일 이름을 지정할 수 없기 때문에 [Oracle Managed Files\(OMF\)](#) 방식으로 동일한 이름의 테이블스페이스를 생성하거나, impdp 유틸리티로 REMAP하기 위한 테이블 스페이스를 생성합니다.

```
-- 기본크기 100MB인 BIGFILE 테이블스페이스로 생성됩니다.  
RDS> create tablespace DPTS;  
RDS> alter tablespace DPTS resize 200M;
```

# 덤프파일을 RDS 인스턴스의 데이터베이스로 가져오기

```
# 덤프파일이 있는 운영체제에서 실행
```

```
impdp jongwon@mydb dumpfile=test.dmp directory=DATA_PUMP_DIR full=y
```

```
-- 테스트 테이블 확인
```

```
RDS> select count(*) from dpuser.test;
```

# 뒤처리하기

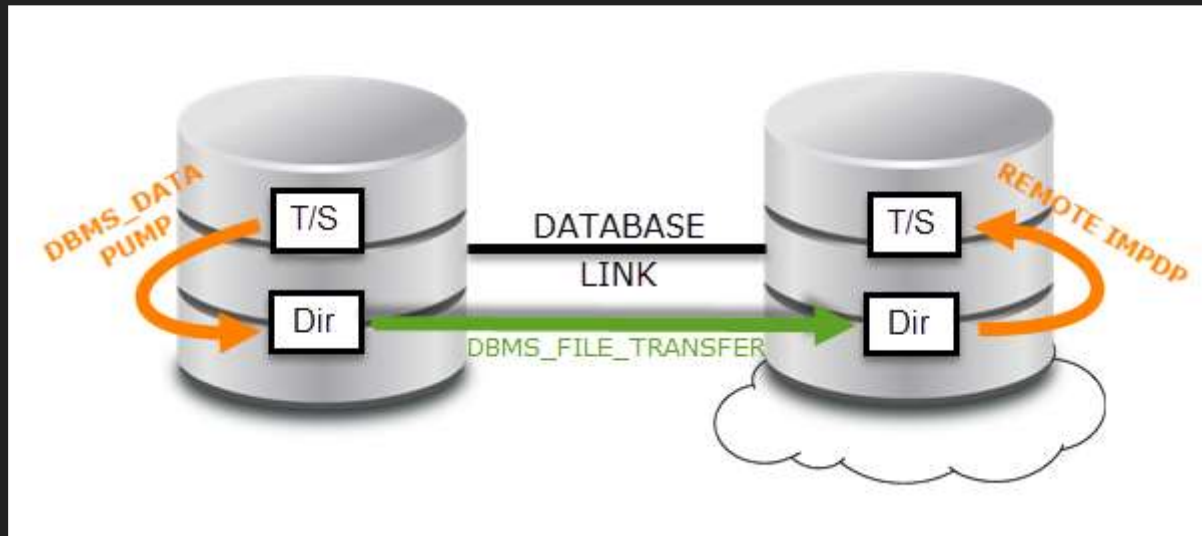
AWS에서는 운영체제에 접근하기 위한 DBA를 위해 rdsadmin이라는 패키지를 제공합니다.

```
-- RDSADMIN.rds_file_util.listdir은 11.2.0.3.v1 이상에서만 사용할 수 있습니다
-- Directory 내의 파일 확인
RDS> select * from table(RDSADMIN.rds_file_util.listdir('DATA_PUMP_DIR'))
```

```
-- 더 이상 필요하지 않은 파일 삭제
RDS> exec utl_file.fremove('DATA_PUMP_DIR','test_copied.dmp');
```



# SUMMARY



1. 원본 데이터베이스의 사용자에게 권한 부여
2. DBMS\_DATAPUMP를 사용하여 덤프파일 생성
3. 대상 DB 인스턴스의 Database Link 생성
4. DBMS\_FILE\_TRANSFER로 내보낸 덤프파일을  
RDS 인스턴스로 복사
5. 대상 인스턴스에 필요한 테이블 공간 생성
6. 덤프파일을 RDS 인스턴스의 데이터베이스로 가져오기
7. 뒤처리하기

# REFERENCE

- AWS Docs : Oracle을 위한 공통 DBA 작업
- AWS Docs : 데이터를 Amazon RDS의 Oracle로 가져오기
- Oracle Docs : DBMS\_DATAPUMP