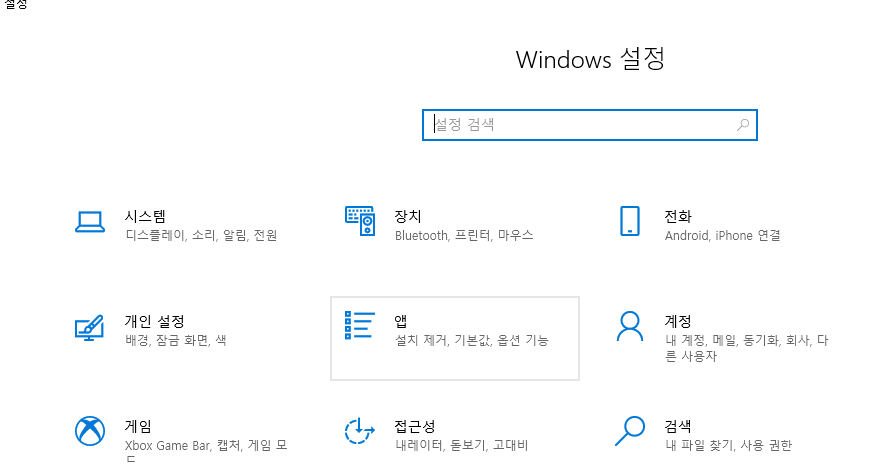
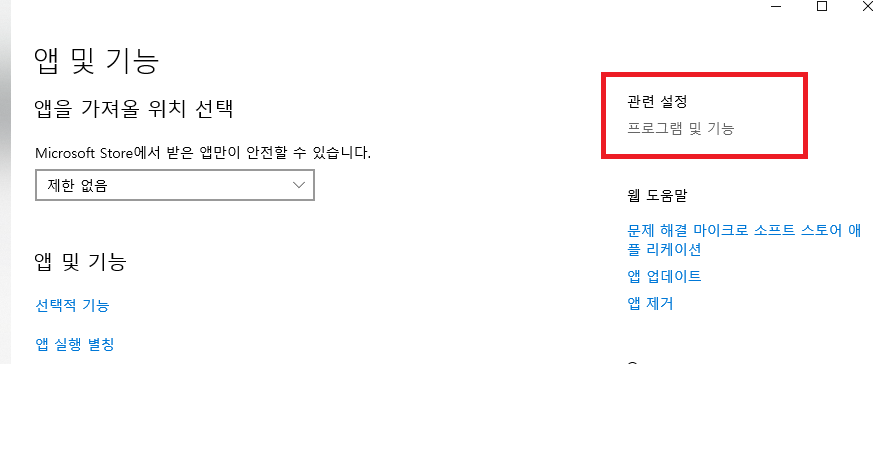
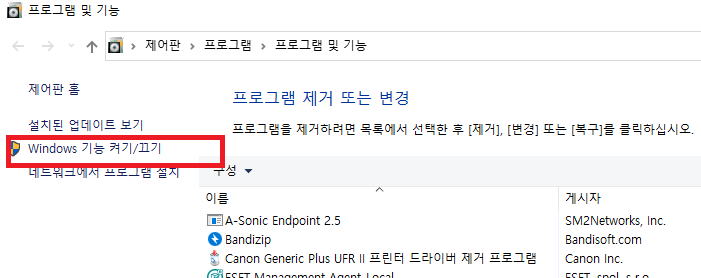
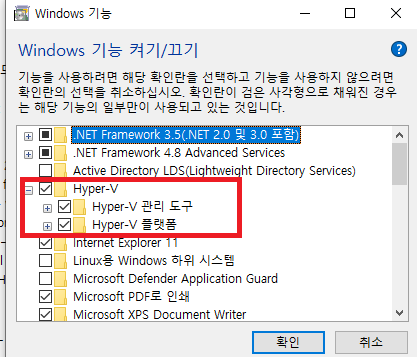
# Hyper V 설치

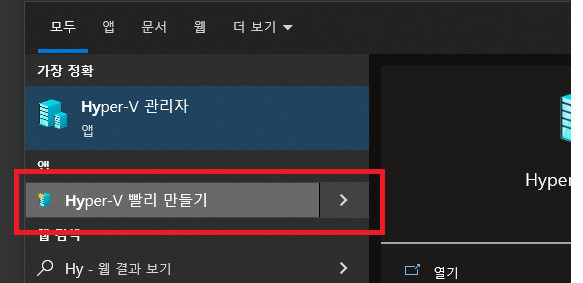
블로그 참조: <https://mainia.tistory.com/5976>

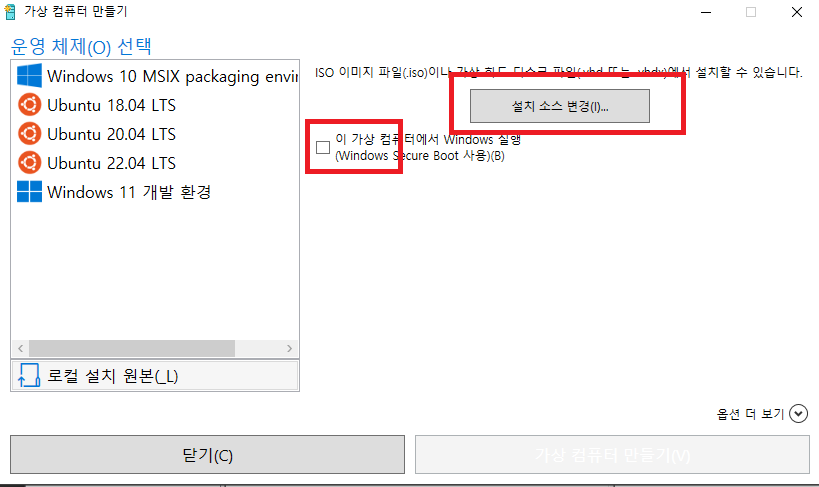


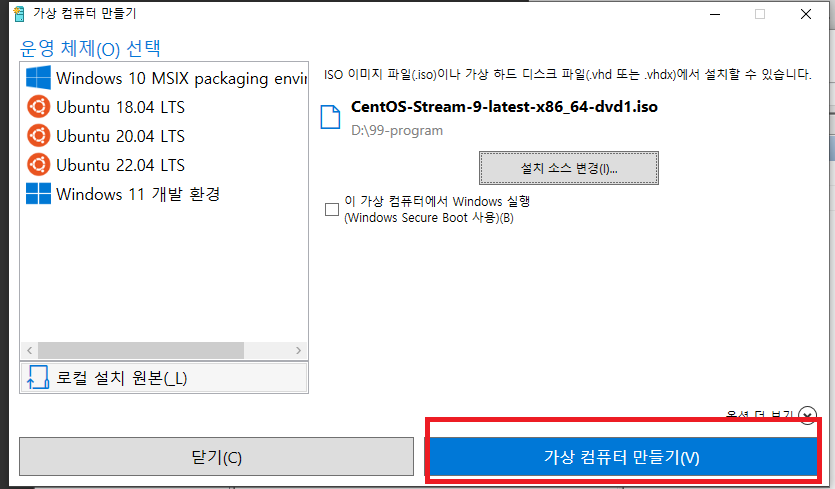












리눅스 설치

<https://veneas.tistory.com/entry/Windows-CentOS-Stream-9-%EC%84%A4%EC%B9%98-Virtual-Box>

# Cent OS 9 Stream에 Dotnet 설치

<https://learn.microsoft.com/ko-kr/dotnet/core/install/linux-rhel#supported-distributions>

| **CentOS Stream 9 ✔️** .NET은 CentOS Stream 9용 AppStream 리포지토리에 포함되어 있습니다. **SDK 설치** .NET SDK를 사용하면 .NET으로 앱을 개발할 수 있습니다. .NET SDK를 설치하면 해당 런타임을 설치할 필요가 없습니다. .NET SDK를 설치하려면 다음 명령을 실행합니다.  Bash  복사  sudo dnf install dotnet-sdk-7.0 **런타임 설치** ASP.NET Core 런타임을 사용하면 런타임을 제공하지 않는 .NET으로 만든 앱을 실행할 수 있습니다. 다음 명령은 .NET에 가장 호환되는 런타임인 ASP.NET Core 런타임을 설치합니다. 터미널에서 다음 명령을 실행합니다.  Bash  복사  sudo dnf install aspnetcore-runtime-7.0  ASP.NET Core 런타임 대신 ASP.NET Core 지원이 포함되지 않은 .NET 런타임을 설치할 수 있습니다. 이전 명령에서 aspnetcore-runtime-7.0을 dotnet-runtime-7.0으로 바꿉니다.  Bash  복사  sudo dnf install dotnet-runtime-7.0 |
| --- |

# 서버 빌드

dotnet clean Plutus.Server.sln

dotnet publish Plutus.Server.sln

1. albatross 빌드

| dotnet publish src/Plutus.Resource -c Release -r linux-x64 -o publish/Plutus.Resource  dotnet publish src/Plutus.Server.Game -c Release -r linux-x64 -o publish/Plutus.Server.Game  dotnet publish src/Plutus.Server.Dove -c Release -r linux-x64 -o publish/Plutus.Server.Dove  dotnet publish src/Plutus.Server.WebAgent -c Release -r linux-x64 -o publish/Plutus.Server.WebAgent  dotnet publish src/Plutus.Server.Albatross -c Release -r linux-x64 -o publish/Plutus.Server.Albatross |
| --- |

dotnet publish src/Plutus.Server.Albatross -c Debug -r linux-x64 -o publish/Plutus.Server.Albatross

albatross 실행

| ./Plutus.Server.Albatross |
| --- |

1. albatross 빌드 및 실행 - **(1번 방법 보다는 이 방법을 추천합니다.)**

| [pg@localhost Plutus.Server.Albatross]$ pwd /home/pg/git/projectg\_server/src/Plutus.Server.Albatross [pg@localhost Plutus.Server.Albatross]$ dotnet run |
| --- |

| [pg@localhost logs]$ netstat -ntpl | grep Plutus (Not all processes could be identified, non-owned process info  will not be shown, you would have to be root to see it all.) tcp 0 0 0.0.0.0:21001 0.0.0.0:\* LISTEN 125258/Plutus.Serve  tcp 0 0 0.0.0.0:5080 0.0.0.0:\* LISTEN 125518/./Plutus.Ser  tcp 0 0 172.24.213.29:6080 0.0.0.0:\* LISTEN 125258/Plutus.Serve  tcp 0 0 172.24.213.29:7080 0.0.0.0:\* LISTEN 125258/Plutus.Serve  tcp 0 0 0.0.0.0:22001 0.0.0.0:\* LISTEN 125258/Plutus.Serve |
| --- |

# Hyper V에 서버 Ping 관련 오류

<https://stackoverflow.com/questions/142614/traceroute-and-ping-in-c-sharp/45565253#45565253>

| static IEnumerable<IPAddress> TraceRoute(string hostNameOrAddress, byte[] dummy, int ttl, int hope) {  if (0 >= hope)  return ImmutableArray<IPAddress>.Empty;   var pinger = new Ping();  PingOptions pingerOptions = new PingOptions(ttl, true);    int timeout = 1000;  var reply = pinger.Send(hostNameOrAddress, timeout, dummy, pingerOptions);   Console.WriteLine($"Status:{reply.Status}");   List<IPAddress> result = new List<IPAddress>();  if (reply.Status == IPStatus.Success)  {  result.Add(reply.Address);   }  else if (reply.Status == IPStatus.TtlExpired || reply.Status == IPStatus.TimedOut)  {  //add the currently returned address if an address was found with this TTL  if (reply.Status == IPStatus.TtlExpired) result.Add(reply.Address);   //recurse to get the next address...  IEnumerable<IPAddress> tempResult = default(IEnumerable<IPAddress>);  tempResult = TraceRoute(hostNameOrAddress, dummy, ttl + 1, hope - 1);  result.AddRange(tempResult);   }  else  {  //failure   }  return result;  } |
| --- |

| [pg@localhost FBData]$ traceroute 8.8.8.8 traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets  1 SUPERCAT-0496-1.mshome.net (172.24.208.1) 0.070 ms 0.056 ms 0.097 ms  2 192.168.122.1 (192.168.122.1) 0.216 ms 0.208 ms 0.190 ms  3 106.250.172.1 (106.250.172.1) 4.406 ms 4.389 ms 4.351 ms  4 10.18.248.109 (10.18.248.109) 2.174 ms 10.18.248.105 (10.18.248.105) 2.164 ms 2.153 ms  5 1.208.4.129 (1.208.4.129) 3.323 ms 1.208.4.165 (1.208.4.165) 3.299 ms 1.208.4.133 (1.208.4.133) 2.663 ms  6 \* 1.208.178.45 (1.208.178.45) 2.705 ms \*  7 1.208.112.77 (1.208.112.77) 2.688 ms \* \*  8 1.208.148.1 (1.208.148.1) 3.356 ms 1.213.113.41 (1.213.113.41) 2.507 ms 1.208.167.5 (1.208.167.5) 2.396 ms  9 1.208.145.46 (1.208.145.46) 45.542 ms 61.42.0.118 (61.42.0.118) 36.867 ms ysnaba7-s10-0-7-25-c.rt.bora.net (203.233.12.118) 45.420 ms 10 1.208.106.106 (1.208.106.106) 38.227 ms 1.208.106.210 (1.208.106.210) 37.428 ms 1.208.148.206 (1.208.148.206) 37.195 ms 11 72.14.215.29 (72.14.215.29) 39.791 ms 74.125.118.154 (74.125.118.154) 41.781 ms 72.14.215.29 (72.14.215.29) 40.203 ms 12 \* \* \* 13 dns.google (8.8.8.8) 37.334 ms 37.249 ms 38.023 ms |
| --- |

# [git] 대용량 파일 저장 git-lfs (Git Large File Storage)

* **lfs를 설치하지 않으면 기획 파일 로드가 되지 않습니다.**

<https://antimidal.tistory.com/30>

| ## RHEL/CentOS  #] curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.rpm.sh | sudo bash  #] sudo yum install git-lfs  #] git lfs install |
| --- |

# Redis 설치

설치 명령어 및 실행 명령어

| $ sudo yum install redis  $ systemctl enable redis |
| --- |

실행 명령어

| $ systemctl start redis  $ systemctl enable redis  $ systemctl status redis  $ systemctl restart redis |
| --- |

레디스 연결 확인

| $ redis-cli ping |
| --- |

## 연결 설정

| vi /etc/redis/redis.conf |
| --- |

| # IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES  # JUST COMMENT OUT THE FOLLOWING LINE.  # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  #bind 127.0.0.1 -::1  bind 0.0.0.0 -::1 |
| --- |

| $ systemctl restart redis |
| --- |

# MySql 설치

<https://www.youtube.com/watch?v=qqECoKYvRNI>

| cat /etc/centos-release  sudo yum update  sudo yum install mysql-server  sudo systemctl start mysqld.service  sudo systemctl status mysqld  sudo systemctl enable mysqld |
| --- |

| #] sudo mysql\_secure\_installation  Securing the MySQL server deployment.  Connecting to MySQL using a blank password.  VALIDATE PASSWORD COMPONENT can be used to test passwords  and improve security. It checks the strength of password  and allows the users to set only those passwords which are  secure enough. Would you like to setup VALIDATE PASSWORD component?  Press y|Y for Yes, any other key for No: n  Please set the password for root here.  New password:  Re-enter new password:  Sorry, passwords do not match.  New password:  Re-enter new password:  By default, a MySQL installation has an anonymous user,  allowing anyone to log into MySQL without having to have  a user account created for them. This is intended only for  testing, and to make the installation go a bit smoother.  You should remove them before moving into a production  environment.  Remove anonymous users? (Press y|Y for Yes, any other key for No) : y  Success.  Normally, root should only be allowed to connect from  'localhost'. This ensures that someone cannot guess at  the root password from the network.  Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y  Success.  By default, MySQL comes with a database named 'test' that  anyone can access. This is also intended only for testing,  and should be removed before moving into a production  environment.  Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y  - Dropping test database...  Success.  - Removing privileges on test database...  Success.  Reloading the privilege tables will ensure that all changes  made so far will take effect immediately.  Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y  Success.  All done! |
| --- |

| mysqladmin -u root -p version  mysql -u root -p |
| --- |

<https://www.server-world.info/en/note?os=CentOS_Stream_9&p=mysql8&f=1>

## 유저추가

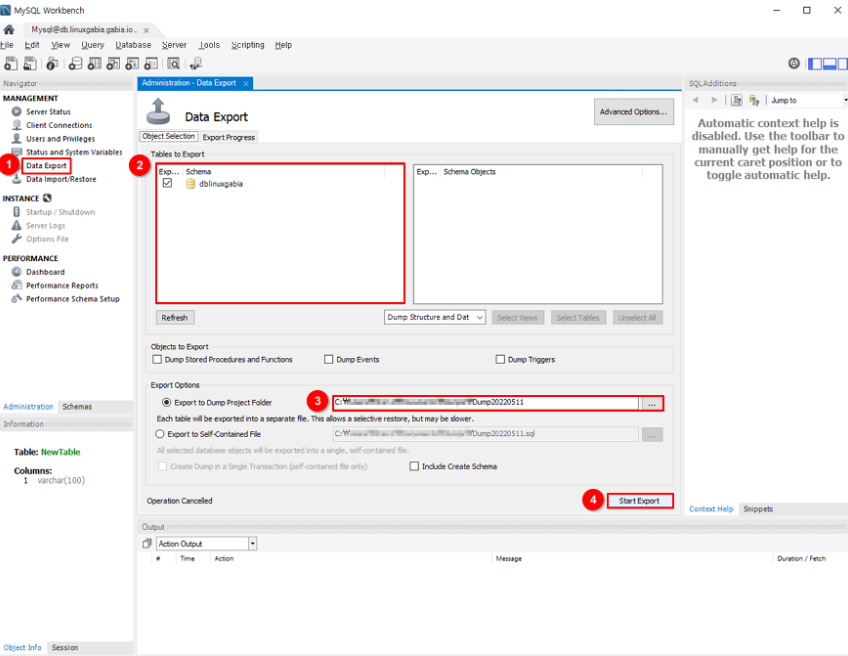
<https://nickjoit.tistory.com/144>

| $ mysql -u root -p  mysql > create user 'pg'@'%' identified by '1111'; // '%' 의 의미는 외부에서의 접근을 허용  mysql > grant all privileges on \*.\* to pg@'%' with grant option; // 모든 데이터베이스 권한 주기  mysql > flush privileges; // 변경된 내용을 메모리에 반영(권한 적용) |
| --- |

| id: pg  pass: 1111 |
| --- |

## Export DB - MySQL 서버 접속/백업/복원하기

<https://customer.gabia.com/manual/hosting/15640/15861>

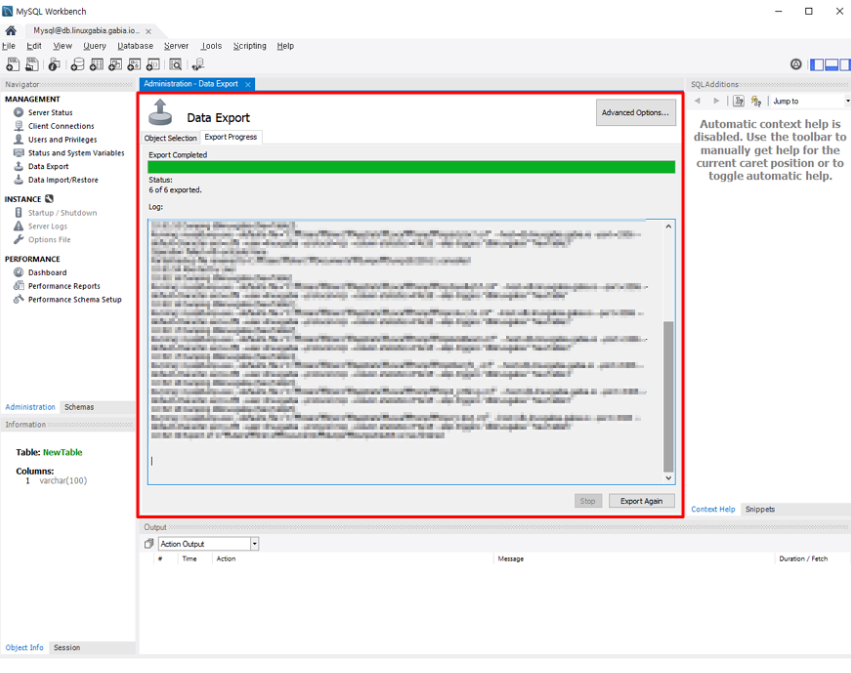


① [Data Export]를 선택합니다.

② 백업할 DB를 선택합니다.

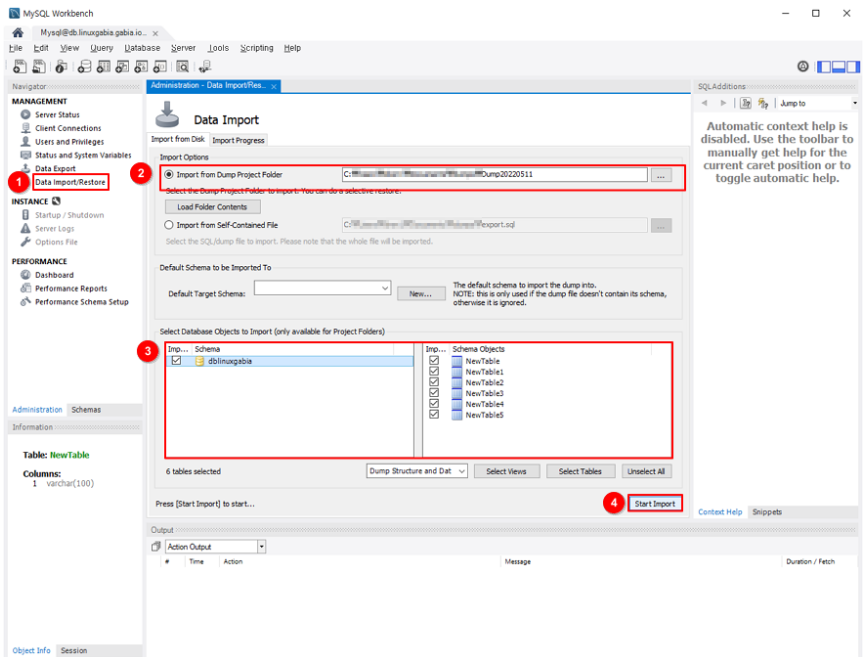
③ 백업 파일을 저장할 디렉터리를 지정합니다.

④ [Start Export]를 클릭합니다.



DB 백업 진행 상황과 완료 여부를 확인할 수 있습니다.

## Import DB - MySQL 서버 접속/백업/복원하기

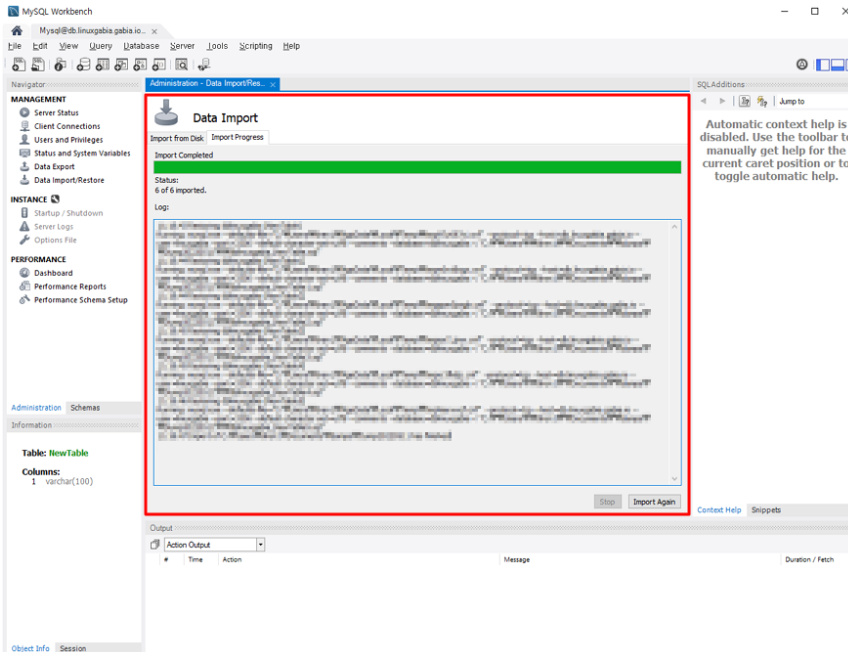


① [Data Import/Restore]를 선택합니다.

② 복원할 DB 데이터가 위치한 디렉터리를 지정합니다.

③ 복원할 데이터베이스를 선택합니다.

④ 복원할 DB 구성요소를 선택합니다.



DB 복원 진행 상황과 완료 여부를 확인할 수 있습니다.

# MySql 서버 내부에 설치

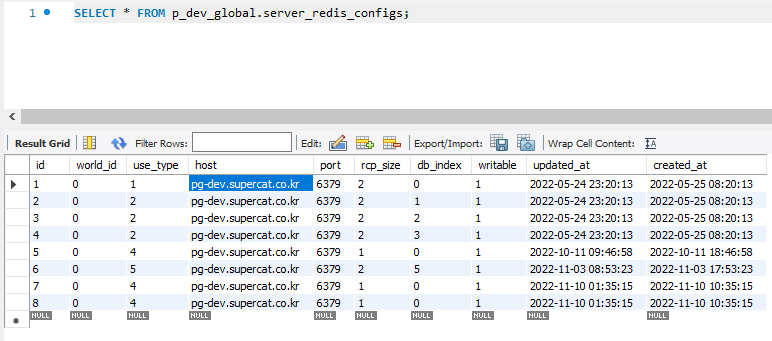
## 기본 schema 생성

| CREATE DATABASE `p\_dev\_global` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_dev\_local\_game0` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_dev\_local\_world` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_dev\_log` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_dev\_world` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_stg\_game0` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_stg\_global` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_stg\_log` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;  CREATE DATABASE `p\_stg\_world` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/; |
| --- |

## 

## redis 접속 정보 수정

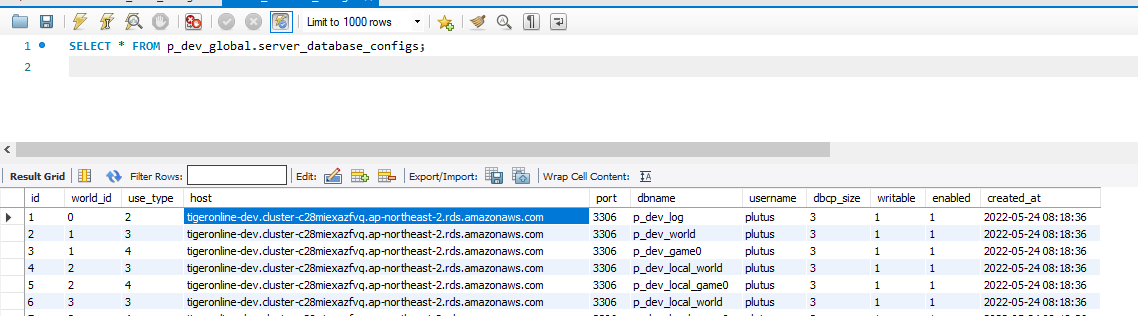
SELECT \* FROM p\_dev\_global.server\_redis\_configs;



pg-dev.supercat.co.kr => 172.24.213.29

로 수정

## db 접속 정보 수정



tigeronline-dev.cluster-c28miexazfvq.ap-northeast-2.rds.amazonaws.com

=> 172.24.213.29

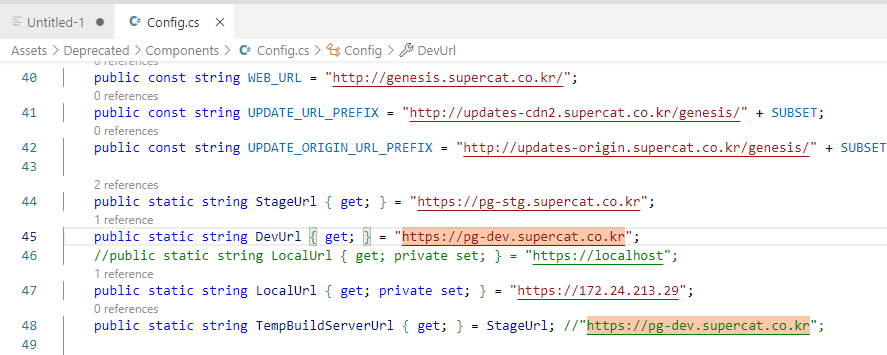
로 수정

# 

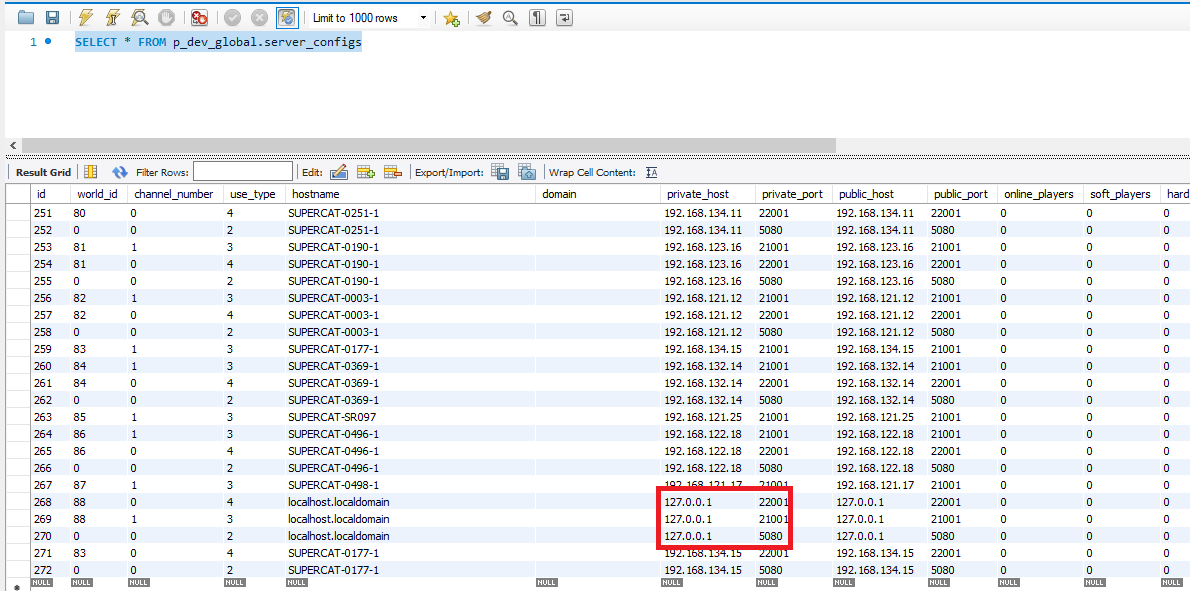
# 

# 

# Client 에 서버 리스트 서버 추가 하기



| public static string DevUrl { get; } = "https://pg-dev.supercat.co.kr";  //public static string LocalUrl { get; private set; } = "https://localhost";  public static string LocalUrl { get; private set; } = "https://172.24.213.29";  public static string TempBuildServerUrl { get; } = StageUrl; //"https:/ |
| --- |



127.0.0.1 -> 172.24.213.29 로 변경

# Yum confict 시 해결법

<https://webinformation.tistory.com/86>

| Transaction Check Error:  file /usr/lib64/mysql/libmysqlclient.so.16.0.0 from install of mysql-community-libs-compat-5.6.14-3.el6.x86\_64 conflicts with file from package compat-mysql51-5.1.54-1.el6.remi.x86\_64  file /usr/lib64/mysql/libmysqlclient\_r.so.16.0.0 from install of mysql-community-libs-compat-5.6.14-3.el6.x86\_64 conflicts with file from package compat-mysql51-5.1.54-1.el6.remi.x86\_64  해결방법  의외로 해결방법이 간단합니다. 아래 명령어를 입력 하시면 자동으로 업데이트를 진행합니다.  [root@localhost ~]# yum shell  > remove compat-mysql51  > install mysql-community-libs-compat  > run |
| --- |

# 방화벽 제거 - SELinux

| [root@dlp ~]# getenforce Enforcing  [root@dlp ~]# setenforce 0  [root@dlp ~]# getenforce  Permissive [root@dlp ~]# setenforce 1  [root@dlp ~]# getenforce  Enforcing |
| --- |

<https://www.lesstif.com/system-admin/centos-selinux-6979732.html>

| [root@dlp ~]# vi /etc/selinux/config  This file controls the state of SELinux on the system.  # SELINUX= can take one of these three values:  # enforcing - SELinux security policy is enforced.  # permissive - SELinux prints warnings instead of enforcing.  # disabled - No SELinux policy is loaded.  # See also:  # https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-selinux-selinux-states-and-modes  #  # NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also  # fully disable SELinux during boot. If you need a system with SELinux  # fully disabled instead of SELinux running with no policy loaded, you  # need to pass selinux=0 to the kernel command line. You can use grubby  # to persistently set the bootloader to boot with selinux=0:  #  # grubby --update-kernel ALL --args selinux=0  #  # To revert back to SELinux enabled:  #  # grubby --update-kernel ALL --remove-args selinux  #  # change value you'd like to set  #SELINUX=enforcing  SELINUX=disabled  # SELINUXTYPE= can take one of these three values:  # targeted - Targeted processes are protected,  # mls - Multi Level Security protection.  SELINUXTYPE=targeted |
| --- |

# NginX 설치

<https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-open-source/>

| sudo yum install epel-release  sudo yum update  sudo yum install nginx |
| --- |

실행 명령어

| $ systemctl start nginx  $ systemctl enable nginx  $ systemctl status nginx  $ systemctl restart nginx |
| --- |

## HTTPS 구성

<https://www.runit.cloud/2020/04/https-ssl.html#openssl%20version%20%EB%AA%85%EB%A0%B9%EC%96%B4%EB%A1%9C%20%ED%99%95%EC%9D%B8%ED%95%98%EA%B8%B0>

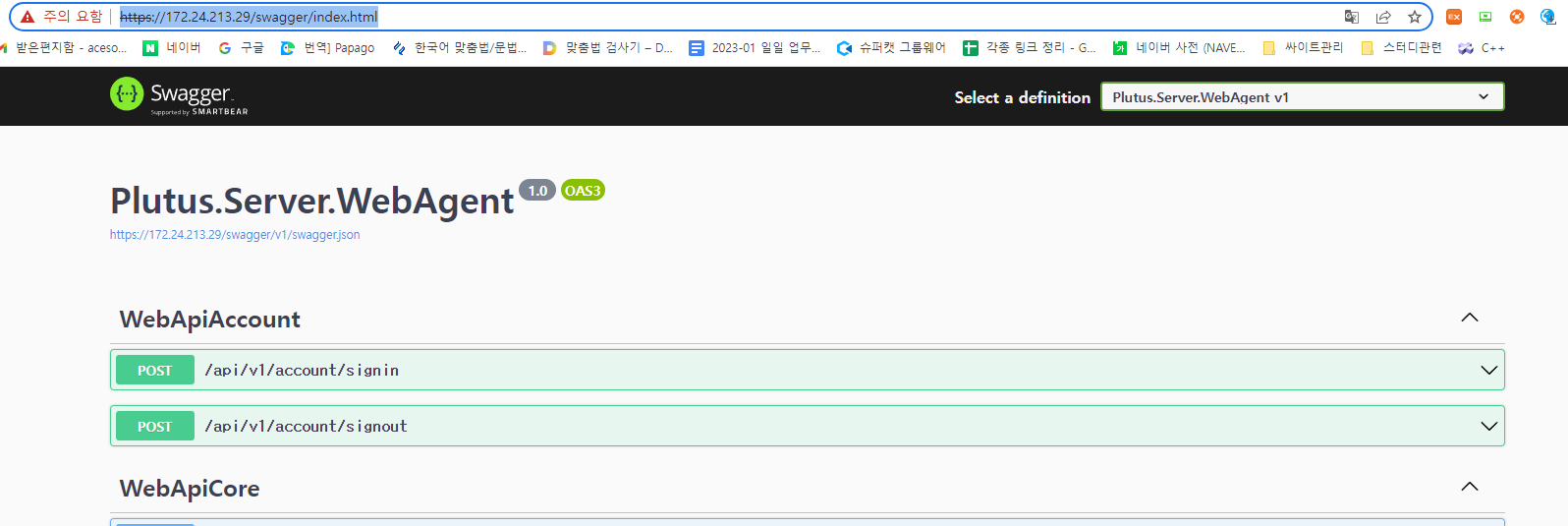
**/etc/nginx/conf.d/default.conf**

| server {  listen 443 ssl http2 default\_server;  listen [::]:443 ssl http2 default\_server;  server\_name localhost;  root /usr/share/nginx/html;  ssl\_certificate "/etc/nginx/ssl/server.pem";  ssl\_certificate\_key "/etc/nginx/ssl/server.key";    ssl\_session\_cache shared:SSL:1m;  ssl\_session\_timeout 10m;  ssl\_ciphers PROFILE=SYSTEM;  ssl\_prefer\_server\_ciphers on;  # Load configuration files for the default server block.  include /etc/nginx/default.d/\*.conf;  location / {  proxy\_pass http://localhost:5080;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection keep-alive;  proxy\_set\_header Host $host;  proxy\_set\_header X-Forwarded-For $remote\_addr;  proxy\_set\_header X-Forwarded-Proto $scheme;  proxy\_cache\_bypass $http\_upgrade;  proxy\_connect\_timeout 300;  proxy\_send\_timeout 300;  proxy\_read\_timeout 300;  send\_timeout 300;  }    error\_page 404 /404.html;  location = /40x.html {  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  }  } |
| --- |

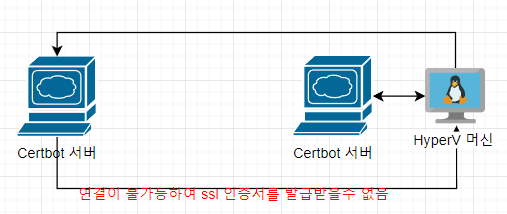
## Swagger 서버 구성

<https://172.24.213.29/swagger/index.html>

* :5080 reverse proxy 로 구성



# 인증키 발급



* **오프 라인으로 인증하는 방법으로 해결!!!**
  + 사설 ssl 로 해결

<https://www.runit.cloud/2020/04/https-ssl.html#openssl%20version%20%EB%AA%85%EB%A0%B9%EC%96%B4%EB%A1%9C%20%ED%99%95%EC%9D%B8%ED%95%98%EA%B8%B0>

* 오류 해결법
  + <https://blogjuso.tistory.com/339>

**아래와 같은 에러 발생시 해결법**

| nginx: [emerg] BIO\_new\_file("/etc/nginx/sslKey/server.crt") failed (SSL: error:0200100D:system library:fopen:Permission denied:fopen('/etc/nginx/sslKey/server.crt','r') error:2006D002:BIO routines:BIO\_new\_file:system lib)  nginx: configuration file /etc/nginx/nginx.conf test failed  -> 파일복사시 보안컨텍스트가 잘못되었나봄...  -> restorecon -v -R /etc/nginx  -> https://serverfault.com/questions/540537/nginx-permission-denied-to-certificate-files-for-ssl-configuration |
| --- |

# 

# 무료 도메인

* 도메인 없이 사설 ssl만으로도 https 서버 생성 가능

<https://taewooblog.tistory.com/83>

<https://xn--220b31d95hq8o.xn--3e0b707e/page/member_login_step1.php>

| bonghoson.kro.kr |
| --- |

http://son.bonghoson.kro.kr/

<https://xn--220b31d95hq8o.xn--3e0b707e/page/domain_conf_view.php?id=1366542>

