# BaroPAM 가이드(PHP)

## 목차


목차 ..... 0

1. BaroPAM 연동 API ..... 1

1.1 연동 API 함수 ..... 1

1.2 인증키 검증 부분 ..... 2

2. BaroPAM 적용 ..... 5

2.1 BaroPAM 적용 프로세스 ..... 5

2.2 BaroPAM 적용 화면 ..... 5

2.3 본인확인 적용 프로세스 ..... 6

2.4 본인확인 적용 화면 ..... 7

2.5 BaroPAM 앱 설치 및 정보 설정 ..... 8

3. NTP(Network Time Protocol) 설정 ..... 9

3.1 Linux 환경 ..... 9

4. About BaroPAM ..... 13

# 1. BaroPAM 연동 API

PHP 환경에서 어플리케이션 로그인 시 비밀번호를 대체 또는 2차 인증(추가 인증)하기 위하여 일회용 인증키(OTA key, One-Time Authentication key)를 검증하는 기능을 제공한다.

## 1.1 연동 API 함수

### 1) system() 함수를 사용하는 경우

- NAME
  system

- SYNOPSIS
  system(string $command, int &$result_code = null): string|false

- DESCRIPTION
  system()함수는 외부 프로그램을 실행하고 출력을 표시.

  command: 실행될 명령
  result_code: IF result_code 인수가 존재한 다음 실행된 명령의 반환 상태가 이 변수에 기록

system() 함수는 주어진 command 실행하고 결과를 출력한다는 점에서 함수의 C 버전과 같다.
system() 함수 호출은 PHP가 서버 모듈로 실행되고 있는 경우에 자동 출력의 각 행 후 웹 서버의 출력 버퍼를 플러시하도록 시도한다.

- RETURN VALUES
  성공하면 명령 출력의 마지막 줄을 반환하고 실패하면 false를 반환

### 2) tomcat과 연동하는 경우

#### ① verifyKEY() 메소드

- NAME
  verifyKEY

- SYNOPSIS
  bool verifyKEY(String login_id, String phone_no, String cycle_time, String ota_key)

- DESCRIPTION
  입력한 일회용 인증키가 맞는지 검증하는 함수.

  login_id: 로그인-ID 항목에 입력한 ID를 설정.
  phone_no: 사용자별 스마트 폰 번호를 숫자만 설정.
  cycle_time: 사용자별로 지정한 일회용 인증키의 생성 주기(3~60초)를 설정.
  ota_key: BaroPAM 앱에서 생성하여 입력한 일회용 인증키를 설정.

  만약, 사용자별로 스마트 폰 번호 및 개인별로 지정한 일회용 인증키의 생성 주기가 일회용 인증키의 생성기와 다른 경우 일회용 인증키가 달라서 검증에 실패할 수 있다. 반드시 정보를 일치시켜야 한다.

- RETURN VALUES
  성공 시에는 true을 반환하며, 실패 시는 false을 반환.

## 1.2 인증키 검증 부분

### 1) system() 함수를 사용하는 경우

```
<?php
// phone_no, cycle_time은 login_id별로 사용자DB에서 가져옴.
$phone_no   = "01027714076"
$cycle_time = "30"

$cmd        = "/usr/baropam/barokey $login_id $phone_no $cycle_time $ota_key"
echo "Command = [$cmd]"

// barokey 실행
$ret = system($cmd, $retval);

// barokey 실행 성공
if ($ret) {
    echo 'barokey 실행 성공!<br>'
    if ($retval == 1) {
        echo '인증 성공!<br>'
    } else {
        echo '인증 실패!<br>'
    }
// barokey 실행 실패
} else {
    echo 'barokey 실행 실패!<br>'
}
?>
```

### 2) tomcat과 연동하는 경우

curl의 명칭은 "**client URL**"을 대표하는 것으로 1997년에 처음 출시되었다. 즉 클라이언트가 스크립트로써 서버에 데이터를 요청하는 것으로 curl로 http/https 되어 있는 **일회용 인증키**의 검증하기 위하여 **일회용 인증키**의 검증을 요청한다.

**일회용 인증키 검증 요청)**

```
<?php
// phone_no, cycle_time은 login_id별로 사용자DB에서 가져옴.
$phone_no   = "01027714076"
$cycle_time = "30"

// cURL 세션을 초기화하는 PHP 함수
$ch = curl_init();

// cURL 전송 옵션 설정
curl_setopt($ch, CURLOPT_URL,"http://nuriapp.com/baropam/web/result_ota.jsp");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
$post = array('login_id' => $login_id, 'phone_no' => $phone_no, 'cycle_time' => $cycle_time,
'ota_key' => $ota_key);
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($post));

// http request 수행
$server_output = curl_exec($ch);

// cURL 세션 닫기
curl_close($ch);

// OTA key 검증(실패)
if (strops($server_output, "00") == false) {
    echo "Authentication faild.";

// OTA key 검증(성공)
} else {
    echo "Authentication success.";
}
?>
```

**일회용 인증키 검증 결과)**

BaroPAM에서 사용하는 인증 코드인 **일회용 인증키**는 Java를 기반으로 작성되었기 때문에 반드시 최신 **JDK 6.x 이상**이 설치되어 있어야 한다. 만약, 설치되어 있지 않으면 최신 JDK를 설치해야 한다.

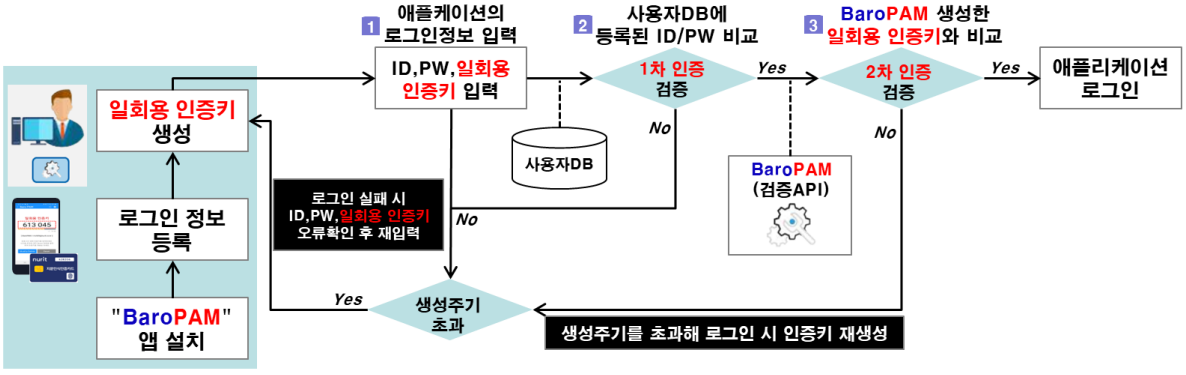
**일회용 인증키**를 검증하는 API는 "**barokey.jar**"로 제공되며, WAS(Web application Server)의 lib 디렉토리 "**barokey.jar**"를 위치시키거나 classpath에 "**barokey.jar**"가 존재하는 디렉토리를 포함해서 설정해 주면 된다.

```
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8" %>
<%@ page import="org.apache.log4j.*"%>
<%@ page import="java.text.*"%>
<%@ page import="java.util.*"%>
<%@ page import="com.barokey.*"%>
<% request.setCharacterEncoding("utf-8"); %>
<% response.setContentType("text/html; charset=utf-8"); %>
<%!
private Logger logger = Logger.getLogger("JspLogger");
%>
<%
/*-------------------------------------------------------------------------*/
/* 변수선언 및 초기화.                                                     */
/*-------------------------------------------------------------------------*/
int     ii = 0, jj = 0, kk = 0, ll = 0;                 // Index

String result   = "";                                  // Result
boolean bota_key = false;                               // 인증키 검증
/*-------------------------------------------------------------------------*/
/* Request에서 데이터를 얻어옴(로그인 정보).                               */
/*-------------------------------------------------------------------------*/
String login_id   = request.getParameter("login_id"  );
String phone_no   = request.getParameter("phone_no"  );
```

```
String  cycle_time = request.getParameter("cycle_time");
String  ota_key    = request.getParameter("ota_key"   );

logger.info("(result_ota.jsp)Starting.....................");
String  param = request.getServerName() + request.getRequestURI()
              + "?remote_addr="         + request.getRemoteAddr()
              + "&login_id="            + login_id
              + "&phone_no="            + phone_no
              + "&cycle_time="          + cycle_time
              + "&ota_key="            + ota_key
              ;
logger.info(param);
/*-------------------------------------------------------------------------*/
/* Begin.                                                                  */
/*-------------------------------------------------------------------------*/
try {
    /*---------------------------------------------------------------------*/
    /* 인증키 검증.                                                        */
    /*---------------------------------------------------------------------*/
    bota_key = barokey.verifyKEY(login_id, phone_no, cycle_time, ota_key);
    /*---------------------------------------------------------------------*/
    /* 인증키 검증(성공).                                                  */
    /*---------------------------------------------------------------------*/
    if (bota_key == true) {
        result = "00";
    /*---------------------------------------------------------------------*/
    /* 인증키 검증(실패).                                                  */
    /*---------------------------------------------------------------------*/
    } else {
        result = "99";
    }
    out.println(result);
/*-------------------------------------------------------------------------*/
/* 예외사항 처리(Exception).                                               */
/*-------------------------------------------------------------------------*/
} catch(Exception e) {
    logger.info("Exception = [" + e + "]");
    e.printStackTrace();
/*-------------------------------------------------------------------------*/
/* Finally.                                                                */
/*-------------------------------------------------------------------------*/
} finally {
    logger.info("(result_ota.jsp)Ending.....................");
}
%>
```

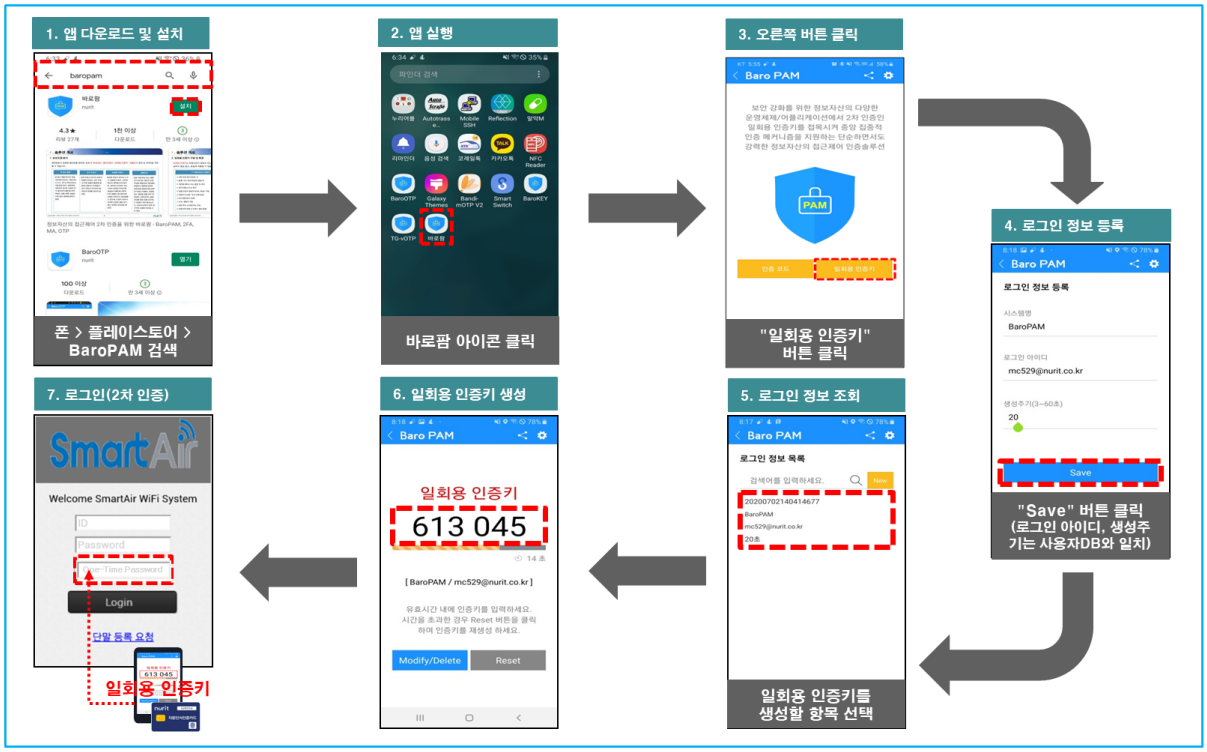
# 2. BaroPAM 적용

## 2.1 BaroPAM 적용 프로세스




| No | 구분 | 내용 | 설명 |
|---|---|---|---|
| 1 | 2차 인증 앱 (BaroPAM) | BaroPAM 앱 다운로드 및 설치 | -안드로이드폰: 구글 플레이스토어 접속 및 "바로팜" 앱 다운로드<br>-아이폰: 애플 앱스토어 접속 및 "BaroPAMs" 앱 다운로드 |
| 2 | | BaroPAM 앱 실행 | |
| 3 | | "일회용 인증키" 버튼 선택 | -실행된 앱 하단 오른쪽 버튼 선택 |
| 4 | | 로그인 정보 등록 | -시스템명: 애플리케이션 명칭(임의 등록 가능)<br>-로그인 아이디: 애플리케이션의 로그인 -ID<br>-생성주기: 3~60초 사이 입력 |
| 5 | | 저장 후 목록 조회 시 나오는 항목 클릭, 일회용 인증키 생성 | -아이폰 사용자는 본인 확인 절차를 거치지 않은 경우 본인 인증 절차를 진행 |
| 6 | | 애플리케이션에 접속해 ID, PW, 일회용 인증키를 입력해 로그인 | -로그인-ID, 생성주기, 일회용 인증키를 이용해 2차 인증 검증 |

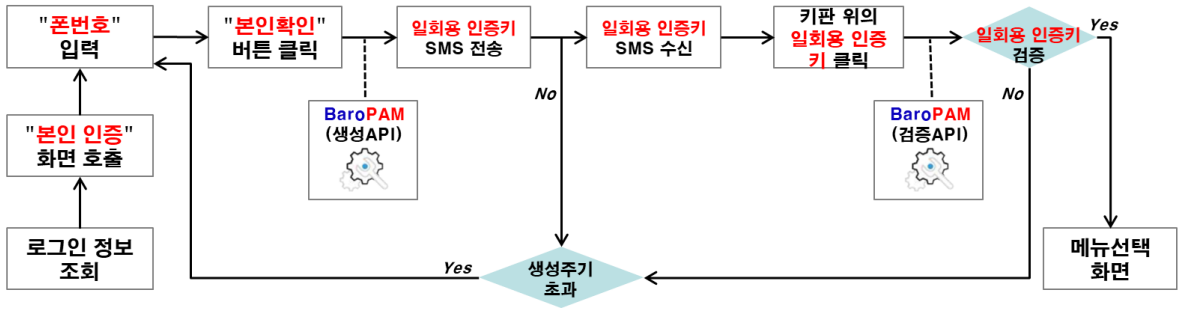
## 2.2 BaroPAM 적용 화면




## 2.3 본인확인 적용 프로세스

아이폰(iPhone)의 기기정보를 얻지 못해서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 로그인 정보 항목을 선택했을 때 "**일회용 인증키**" 생성 화면으로 이동하지 않은 경우가 발생할 수 있다.

또한, 타인의 폰번호를 부정으로 사용하지 못하도록 하기 위해서 별도의 본인확인 기능을 적용할 필요가 있는데, "**BaroPAM**" 앱에서는 자체 알고리즘을 적용하여 자체적으로 본인확인 절차를 실행하고 있다.

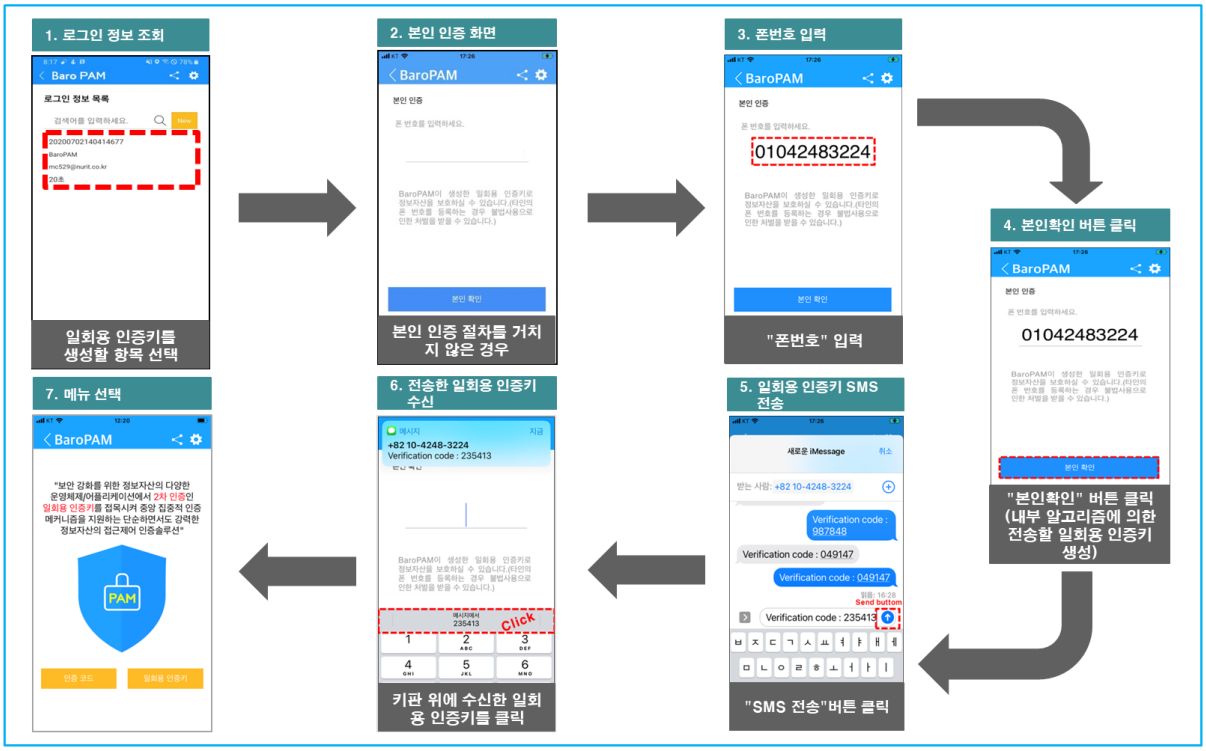



| No | 구분 | 내용 | 설명 |
|---|---|---|---|
| 1 | 아이폰 본인확인 | 로그인 정보 목록에서 일회용 인증키를 생성할 항목 클릭 | -본인확인 절차를 거치지 않은 경우 본인인증 절차를 진행 |
| 2 | | "본인인증" 화면 호출 | |
| 3 | | "폰번호" 입력 | -본인인증할 폰번호를 입력 |
| 4 | | "본인확인" 버튼 클릭 | -내부 알고리즘에 의하여 "Verification code"인 일회용 인증키를 생성<br>-생성한 일회용 인증키를 전송하기 위한 SMS 연동 화면 호출 |
| 5 | | "Verification code" SMS 전송 | -일회용 인증키를 확인한 후 "SMS 전송" 버튼을 클릭 |
| 6 | | "Verification code" SMS 수신 | -본인 폰으로 전송한 일회용 인증키가 키판 위에 표시<br>-폰번호가 틀린 경우 타인 폰으로 일회용 인증키가 전송됨 |
| 7 | | 키판 위의 "Verification code" 클릭 | -내부 알고리즘에 의하여 일회용 인증키를 검증<br>-검증에 성공하면 메뉴선택 화면, 실패하면 본인인증 화면이 호출 |

## 2.4 본인확인 적용 화면

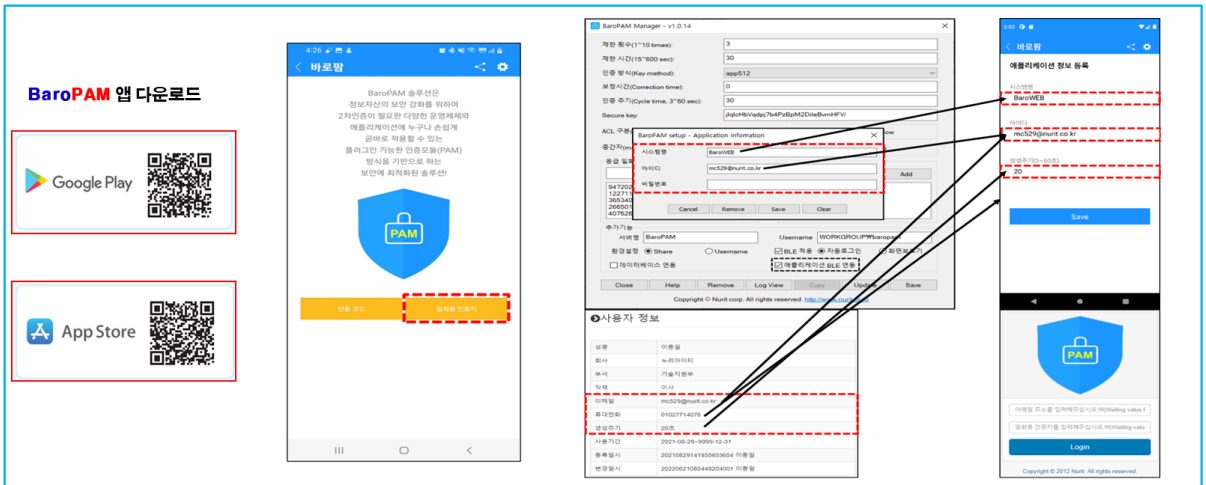
아이폰(iPhone)의 기기정보를 얻지 못해서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 로그인 정보 항목을 선택했을 때 "**일회용 인증키**" 생성 화면으로 이동하지 않은 경우가 발생할 수 있다.

또한, 타인의 폰번호를 부정으로 사용하지 못하도록 하기 위해서 별도의 본인확인 기능을 적용할 필요가 있는데, "BaroPAM" 앱에서는 자체 알고리즘을 적용하여 자체적으로 본인확인 절차를 실행하고 있다.




## 2.5 BaroPAM 앱 설치 및 정보 설정

정보자산에 로그인 시 Verification code에 입력할 일회용 인증키의 생성기인 BaroPAM 앱의 다운로드 (https://play.google.com/store/apps/details?id=com.baro.pam)는 구글의 "Play 스토어"나 Apple의 "App 스토어"에서 가능하며, 설치는 일반 앱의 설치와 동일하다.




**BaroPAM 앱은 Android 6.0 (Marshmalliw) API 23, iOS 13.0 이상에서 사용 가능하며, 가로보기 모드를 지원하지 않는다.**

**BaroPAM 앱을 설치한 후 BaroPAM 앱을 실행하여 메뉴 선택화면에서 "일회용 인증키" 버튼을 클릭하여 애플리케이션의 사용자 정보에 설정한 "인증주기, 아이디, 시스템명"을 BaroPAM 앱의 "애플리케이션 정보 등록" 화면에서 동일하게 입력해야 한다.**

**현상 : 안드로이폰 또는 아이폰의 날짜와 시간이 현재 시간과 차이가 발생하여 "일회용 인증키"가 맞지 않은 경우**
**원인 : 안드로이폰 또는 아이폰의 날짜와 시간을 네트워크에서 제공하는 시간을 사용하지 않아서 발생.**
**조치 : 안드로이폰인 경우는 폰의 "설정" -> "일반" -> "날짜 및 시간" -> "날짜 및 자동 설정"과 "시간대 자동 설정" -> "허용"**
**아이폰인 경우는 폰의 "설정" -> "날짜 및 시간" -> "자동으로 설정" -> "허용"**

# 3. NTP(Network Time Protocol) 설정

최근에는 서버/네트워크 장비에 대한 시간 동기화(타임서버 시간 동기화)하는 방법으로 NTP(Network Time Protocol)을 이용하여 관리자 계정에서 시스템의 시각을 현재 시각으로 설정할 수 있다.

## 3.1 Linux 환경

최근에는 Windows/서버/데이터베이스/네트워크 장비/저장장치에 대한 시간 동기화(타임서버 시간 동기화)하는 방법으로 NTP(Network Time Protocol)을 이용하여 **root** 계정에서 시스템의 시각을 현재 시각으로 설정할 수 있다.

NTP를 사용하기 위해서는 기본적으로 NTP 패키지가 반드시 설치되어 있어야 한다. 설치 확인은 다음의 명령어를 실행하여 확인한다. 만약, 설치되어 있지 않으면 Redhat, CentOS 8 이하 버젼은 "**yum install ntp**" 그외는 "**sudo apt-get install ntp**" 명령어로 설치하면 된다.

```
[root]# rpm -qa | grep ntp
ntp-4.2.2p1-18.el5.centos
chkfontpath-1.10.1-1.1
```

ntpd 서비스를 서버 부팅 시 시작프로그램에 등록 및 ntp 활성화 여부 확인은 다음과 같은 명령어로 확인할 수 있다.

```
[root]# chkconfig ntpd on
[root]# chkconfig —list | grep ntp
ntpd            0:해제  1:해제  2:활성  3:활성  4:활성  5:활성  6:해제
```

chkconfig 이용하여 서버 부팅시 ntpd 데몬 활성화 여부 확인 3, 5 level에 off(해제) 가 되어 있으면 자동 활성화되지 않는다. 자동 활성화하기 위해서는 3, 5에 on(활성)으로 다음과 같은 명령어로 변경해야 한다.

```
[root]# chkconfig —level 3 ntpd on
[root]# chkconfig —level 5 ntpd on
```

우리나라에서 운영되고 있는 NTP 서버는 다음과 같다.

```
server kr.pool.ntp.org
server time.bora.net
server time.kornet.net
```

우리나라에서 운영되고 있는 NTP 서버를 ntpd 데몬 설정을 위한 설정 파일인 "**/etc/ntp.conf**"에 다음과 같이 설정한다.

```
[root]# vi /etc/ntp.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.centos.pool.ntp.org
#server 1.centos.pool.ntp.org
#server 2.centos.pool.ntp.org
#server 3.centos.pool.ntp.org
```

```
server kr.pool.ntp.org iburst
server time.bora.net   iburst
server time.kornet.net iburst
```

iburst 옵션은 일종의 옵션 설정으로써 동기화 하는데 걸리는 시간을 짧게 줄여주는 옵션임.

ntpd 데몬 설정을 위한 설정이 끝나면 반드시 NTP 설정이 제대로 추가되었는지 확인한 후 NTP 데몬의 Restart 작업이 반드시 필요하다.

```
[root]# /etc/init.d/ntpd restart
ntpd를 종료 중: [  OK  ]
ntpd (을)를 시작 중: [  OK  ]
```

ntpd 시간 확인은 다음과 같은 명령어로 확인할 수 있다.

```
[root]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 static.betaidc. 106.247.248.106  3 u    7   64    1    2.884  287.718   0.001
 time.bora.net   .INIT.          16 u    -   64    0    0.000    0.000   0.000
 183.110.225.61  .INIT.          16 u    -   64    0    0.000    0.000   0.000
 LOCAL(0)        .LOCL.          10 l    4   64    1    0.000    0.000   0.001
```

* 표시된 ip 가 현재 시간을 가져오고 있는 ntp 서버임

NTP를 사용하기 위해서는 기본적으로 NTP 패키지가 반드시 설치되어 있어야 한다. 설치 확인은 다음의 명령어를 실행하여 확인한다. 만약, 설치되어 있지 않으면 Redhat, CentOS 8 이상 버젼은 "**yum install chrony**" 명령어로 설치하면 된다.

```
[root@baropam ~]# rpm -qa | grep chrony
chrony-3.5-1.el8.x86_64
```

우리나라에서 운영되고 있는 NTP 서버는 다음과 같다.

```
server kr.pool.ntp.org
server time.bora.net
server time.kornet.net
```

우리나라에서 운영되고 있는 NTP 서버를 ntpd 데몬 설정을 위한 설정 파일인 "**/etc/chrony.conf**"에 다음과 같이 설정한다.

```
[root@baropam ~]# vi /etc/chrony.conf

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#pool 2.centos.pool.ntp.org iburst
server kr.pool.ntp.org iburst
server time.bora.net   iburst
server time.kornet.net iburst

# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift

# Allow the system clock to be stepped in the first three updates
```

```
# if its offset is larger than 1 second.
makestep 1.0 3

# Enable kernel synchronization of the real-time clock (RTC).
rtcsync

# Enable hardware timestamping on all interfaces that support it.
#hwtimestamp *

# Increase the minimum number of selectable sources required to adjust
# the system clock.
#minsources 2

# Allow NTP client access from local network.
allow 192.168.0.0/16

# Serve time even if not synchronized to a time source.
#local stratum 10

# Specify file containing keys for NTP authentication.
keyfile /etc/chrony.keys

# Get TAI-UTC offset and leap seconds from the system tz database.
leapsectz right/UTC

# Specify directory for log files.
logdir /var/log/chrony

# Select which information is logged.
#log measurements statistics tracking
```

ntpd 데몬 설정을 위한 설정이 끝나면 반드시 NTP 설정이 제대로 추가되었는지 확인한 후 NTP 데몬의 Restart 작업이 반드시 필요하다. (chrony 서비스 시작 및 부팅시 구동 등록)

```
[root@baropam ~]# systemctl start chronyd
[root@baropam ~]# systemctl enable chronyd
```

ntpd 시간 확인은 다음과 같은 명령어로 확인할 수 있다.

시간을 받아오는 서버 리스트 / chrony.conf 파일에 등록한 server 리스트)

```
[root@baropam ~]# chronyc sources
210 Number of sources = 2
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* ec2-54-180-134-81.ap-nor>     2   6   377    43   -349us[-1059us] +/-   24ms
^- time.bora.net                 2   6   377    42  +1398us[+1398us] +/-   90ms
```

시간을 받아 오는 서버 정보)

```
[root@baropam ~]# chronyc tracking
Reference ID    : 36B48651 (ec2-54-180-134-81.ap-northeast-2.compute.amazonaw)
Stratum         : 3
```

```
Ref time (UTC)  : Sun Mar 22 07:07:43 2020
System time     : 0.000130027 seconds slow of NTP time
Last offset     : -0.000710122 seconds
RMS offset      : 0.000583203 seconds
Frequency       : 19.980 ppm fast
Residual freq   : +0.142 ppm
Skew            : 3.235 ppm
Root delay      : 0.013462566 seconds
Root dispersion : 0.017946836 seconds
Update interval : 65.0 seconds
Leap status     : Normal
```

시간 상태 및 동기화 등 정보 확인)

```
[root@baropam ~]# timedatectl status
               Local time: Sun 2020-03-22 16:08:45 KST
           Universal time: Sun 2020-03-22 07:08:45 UTC
                 RTC time: Sun 2020-03-22 07:08:44
                Time zone: Asia/Seoul (KST, +0900)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

# 4. About BaroPAM




제 조 사 : 주식회사 누리아이티
등록번호 : 258-87-00901
대표이사 : 이종일
대표전화 : 02-2665-0119(영업문의/기술지원)
이 메 일 : mc529@nurit.co.kr
주 소 : 서울시 강서구 마곡중앙2로 15, 913호(마곡동, 마곡테크노타워2)