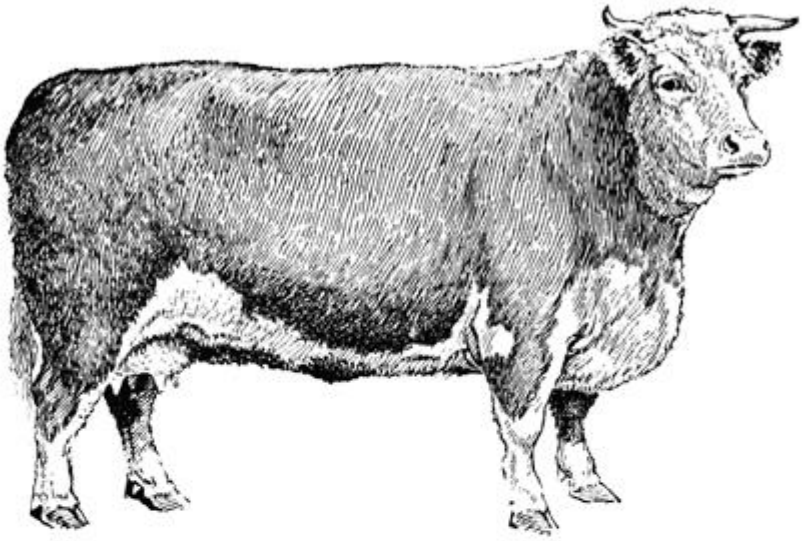
A Guide To The Awesome AMChecks



AMChecks

A Field Guide

O RLY?

Tony Webb

AM CHECKS

Oracle Morning Checks (AMChecks)

What is AMChecks? (aka 'the Story So far')

AMChecks is a (very) loose collection of scripts – a bunch of korn shell scripts (ksh) and SQL scripts (including anonymous PL/SQL blocks but no actual procedures) that are intended to run from a 'central' linux server, primarily via a series of crons on that server. For anything very useful you'll also need a database repository on this server as well – Standard Edition will do, although there are still some scripts that can run without a repository.

The basic idea behind AMChecks is to have something that can keep an eye on your Oracle infrastructure and give enough information to help you look after your Oracle estate while you're off doing more exciting things (say, perhaps creating more scripts for AMChecks or, more rewardingly, picking the fluff out of your belly button). It's as lightweight as I could make it – all you need to do for each target database is create a new login that has read access to the data dictionary*. If you want to monitor mountpoint space (linux only at present) then there's a little more effort required but you can add this in later once you're happy with how AMChecks is running 'out of the box'.

What isn't it?

It's not a banana called Graham. It's also not intended to replace anything you've paid actual money for. It will not give you a shiny interface and it won't allow you to drill down into performance metrics. It is not intended to replace Oracle Cloud Control or serious monitoring software like BMC Patrol or Quest's Foglight. You might find it useful if you have nothing else checking up on your databases. Feel free to give it a go and feel equally free to delete it when you find something more appropriate for your organisation.

Is it free?

Yes. Of course it's free. Do you think I could actually charge for this? ..although all donations, monetary or otherwise, will be gratefully received!

Please be aware that whereas I've tried to develop this for the lowest common denominator (so no diagnostic pack, no tuning pack, no enterprise edition even) there is no actual guarantee that running the scripts will not invalidate your Oracle license in any way. I mean they *look* OK but use them at your own risk. You have been warned! If you spot a red flag please let me know ASAP.

There is NO support offered for AMChecks either – you're pretty much on your own although there might be some other poor unfortunates who also use the scripts that may be able to help you. If you can help them and either add new scripts or improve existing scripts (I suspect there's A LOT of scope here..) please let me know – that would be 👍*awesome*👍.

Pre-Installation

Check that you are clear to proceed as there may be company change control procedures, management sign-off or other non-technical 'opportunities' to overcome before any actual installation. Consider a plan for how you will roll-out these checks (e.g. test each script against a 'sandbox' database first, then deploy to your UAT environment). Be cautious – some databases will behave differently to each script so a 'big bang' deployment might just end up being all too appropriately named.

Identify the (linux) server that you intend to run the scripts from and make sure that you can ping all of your intended Oracle database hosts from this machine.

At the very least install an Oracle client on this box but realistically you're going to want to install the database software too (check your Oracle licensing first of course before installing anything).

Make sure that you can run **`sqlplus`** to these boxes too.

You don't need a new OS user/login but you might want to consider it. I developed these using user 'oracle' but feel free to give another user a whirl.

Check that this user has access to **`cron`** (unless you plan to use an alternative scheduler for your checks) and that the cron daemon is running on your 'master' server. E.g. run: **`ps -elf | grep crond`**

Find your Linux administrator (anywhere that has a significant amount of caffeine is a good place to start your search) if it's not you (if it is you I'm not going to help you 'find yourself' – this isn't self-help therapy) and discuss this section with them.

Make sure that you have the korn shell installed on your nominated server (unless you want to have fun converting the ksh scripts into some other kind of script (please share the pain/joy if you do!). Once installed, note the path of your ksh on the server. If it's NOT **/bin/ksh** then either change the first line of **every** .ksh script to where your ksh is or create a symbolic link from where it is to /bin/ksh using the relevant user (probably 'root'). Go with whatever your Linux administrator advises. e.g.

```
ln -s /blah/ksh /bin/ksh
```

Find or create a `tnsnames.ora` file. This should only contain the databases you want to check, at least to begin with. Edit the file to have all connect information for a database on a single line e.g.

```
FRED = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = bedrock.com)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = FRED)))
WILMA = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = bedrock.com)(PORT = 1522)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = WILMA)))
```

This may not be your usual tnsnames.ora file – you'll want to keep it simple, so you might need to add new simple entries for your high availability databases and remove other entries. You also don't want to have any entries for your standby databases, with the possible exception of active standbys. Only instances that you are licensed to run queries against should be in your tnsnames.ora.

Test this most-likely-slimmed-down tnsnames.ora (using **`tnsping`** with **`TNS_ADMIN`** is probably the easiest way to test). Keep this file safe as you will need to copy this into your 'AMChecks' top level directory once you've downloaded the scripts.

AM CHECKS

Installation

- Grab a caffeine-infused beverage. Tea is a good example although green tea is to be avoided, not just for this step but for the entirety of your life if possible: it tastes foul.
- Unpack the files that you've downloaded onto the box you intend to run your scripts from. The suggested directory is: `~oracle/amchecks`
- Copy your `tnsnames.ora` that you were working on in the pre-installation steps into this directory but call it `tnsnames.ora.full`. Create a slimmed down version of this file, called `tnsnames.ora` which contains just one line – the connect details for your 'master' database.
- Create the following OS directories if they don't exist (it seems unlikely unless you were foolish enough to try AMChecks before):

```
mkdir -p ~oracle/amchecks/external_tables
mkdir -p /tmp/amchecks
```

- If you don't have a database to host your checks, create it and come back here when you are done. Unless you have a massive Oracle estate the schema itself should be well under 1 Gig.
- Create 2 new users – one to own the metadata and one to run the checks. Do this by editing (to change the passwords) and then running `users.sql` when connected to your host/master database (NOT the databases you will be monitoring) as a god-like user (e.g. 'system'). Note that if you want to tighten up access you can almost certainly replace the '`SELECT_CATALOG_ROLE`' role with a bunch of more specific, focused system or object permissions. Be my guest but I've not explored this option yet. Also resist using '`SELECT ANY DICTIONARY`' especially if you are on Oracle 9 databases (don't be embarrassed - you're in good company!) – you don't really want to be able to run : `select name, password from sys.user$`. Currently '`SELECT ANY DICTIONARY`' is only needed for 1 script: `password_expiry_check11.sql` The amu user shouldn't really be able to see any business sensitive data or change anything. Don't change this by adding any additional permissions.

- Know how to change all of your 'amu' passwords – this can be pretty easy as you can run something like: `ALTER USER amu IDENTIFIED BY t0pSecritPasswd;` on every database in a few seconds (but remember to change the `CONNECT_PWD` in the file mentioned in the next section). Currently there is no provision to have different passwords on your monitored databases (although it wouldn't be that hard to add in this functionality).

- If you are using *`prog_checks.ksh`* (extremely unlikely even if you do have Progress databases TBH) ensure that the hard-coded password for user *`amu`* in this file is corrected. This should be the only script with the hard-coded password.

AM CHECKS

- Edit `~oracle/amchecks/.amcheck` using your favourite editor and change the database, connection details and anything else that looks likely. If you're not comfortable with this then, no offence, you probably don't want to continue with the installation.

```
export ORACLE_SID=ORCL
export CONNECT_USERNAME=amu
export CONNECT_PWD= fabum0use
export CONNECT="${CONNECT_USERNAME}/${CONNECT_PWD}"
export OWNER_CONNECT=amo/ ferretH8er
export READER_CONNECT=amr/ orch1dfr0g
export AMCHECK_TNS="${CONNECT}@${ORACLE_SID}"
export MAIL_RECIPIENT='fred.flintstone@bedrock.com,wilma.rubble@bedrock.com'
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/11204/
export PATH=$PATH:${ORACLE_HOME}/bin
alias SQLPLUS='sqlplus $CONNECT@$ORACLE_SID'
export TERM=xterm
export FROM_ADDRESS="DBAs@bedrock.com"
```

I'm not sure why I have the CONNECT_USERNAME and CONNECT_PWD separate. Also I'm not convinced that I'm using all of these variables you know; I'll need to check at some point…

ORACLE_SID is the local database name of your master database; amu is the user that runs the checks; amo is the owner of the metadata objects. Amr is intended to be a purely read only user but I'm not currently using it as it's too similar to amu at this stage.

Check file permissions of this file. Be sure to keep it as private as possible as it contains the passwords for the AMChecks (do something like `chmod 500 ~oracle/amchecks/.amcheck` once you are finished).

- At this point you can run one of the scripts – `is_oracle_ok.ksh`. You will need to tell it not to connect to the database to determine the list of targets but to get the list from parsing the tnsnames.ora file. Do this by running: `~oracle/amchecks/is_oracle_ok.ksh -t -p`

  As an aside, once you 'go live' you'll probably stop using the '-t' option but it can be useful to run it periodically to check that your tnsnames.ora is not bloated with dead entries.

  If this all works OK (it probably won't as you might need to change environment variables or some other bits and bobs) then you can add other databases into your tnsnames.ora file until it is indistinguishable from `tnsnames.ora.full`

- Now setup `sendmail` on your 'master' server if it's not already enabled. Do this as follows (again you'll need to track down your linux administrator or whoever is responsible for linux mail) Note that some commands may vary depending on linux distro, version etc.:

  - `ps –elf | grep sendmail:` (look for a running process or two ..other than your `grep`!)
  - `systemctl list-unit-files | grep` (look for an enabled service)
    or better still, as root type `chkconfig |grep sendmail` (this will show the run levels too).
  - If its installed but not started, kick it off as root with `chkconfig sendmail` or `service sendmail start on` which will add it as an automatically started service.
  - If it's not installed, then install it (as root) e.g. `yum install sendmail`
  - Find and check your config file – e.g. `/etc/mail/sendmail.cf` checking that your relay host (if present) is added plus anything else your organisation normally sets. This will vary so you will have your own specific changes to make here.

- If sendmail refuses to start (as root running `service sendmail start`) you may have something else using your port (25). Check this via: `lsof -i :25` (as root) to see what is using it (assuming `lsof` is installed..).

Generally this section tends to be the most frustrating (it'd the least Oracley and least databasey) but stick with it – it's worth it!

- Next you can add a cron to periodically check your database connectivity by something like the following:

```
02,07,12,17,22,27,32,37,42,47,52,57 * * * 1-5  ksh -c '/home/oracle/amchecks/is_oracle_ok.ksh -t -m'
```

This will run Monday through Friday every 5 minutes. You can schedule however you want. Of course, if you have any other scheduler then feel free to use that in preference to cron.

If your sendmail is working and your environment variables are all OK then you should get an e-mail that looks something like this:

From: DBAs@Bedrock.com
Sent: 07 December 2016 14:57
To: Webb, Tony; DBA; Flintstone, Fred
Subject: WARNING - Oracle Connectivity
Attachments: tmp17284.html; ATT00001.txt

Wed 7 Dec 14:57

```
Processing AA01 (on pkaa01 (unix1p0))                Looks OK (10 msec)
Processing AB022 (on aopab022)                       Looks OK (40 msec)
Processing CATALOG (on unix5)                        Looks OK (10 msec)
Processing VMSID1 (on linux47 [VMCLUST-01])          Looks OK (0 msec)
Processing VMSID2 (on linux47 [VMCLUST-01])          Looks OK (0 msec)
Processing WMDB4 (on pkwmdb4 (unix2p0))              ** Unable to tnsping WMDB4! ((11520 msec)) **
```

Different command line parameters will alter if/when you get an e-mail. When testing keep the parameters simple.

This is a good point to stop and celebrate your success before carrying on!

- Next you want to create and populate your master database. This will take a while but again it's something that you should start small with. Add 1 or 2 non-critical, and non-production target databases first. Running some of the other check scripts on a database has the potential, like most new queries, to impact performance of your database so it's a good idea pick a suitable looking victim database to test with first.

  So, as user `amo`, on your 'master' database, run '`am_schema.sql`'

  You are encouraged to keep an additional file named '`add_to_schema.sql`' that contains any changes you have made to the `amo` schema, post-install. You might have your own tables (rare), your own columns (very common) so you should keep a note of these should you ever reinstall. If your download includes an '`add_to_schema.sql`' already then take a look. You may or may not want to run it.

  The schema will be mostly empty at this stage. Add in a couple of databases and their servers (add the server into `amo.am_server` before adding the databases into `amo.am_database`). You can start using other tables and bulking out these two tables later.

From the command line run the main *am_checks.ksh* script. Specify it to run on one database (the one containing Amchecks) and to run just one script. You might as well e-mail the output too:

```
./amchecks.ksh -d ORCL -s instance_summary.sql -m
```

Again, once this seems to be working (you might get a few issues again) you should stop. Maybe try a few more databases and a few more scripts. The next section describes how to add in your databases

How to Add In Your Databases

The idea of Amchecks is to keep the installation on monitored databases as small as possible. With that in mind this is the process to add in a 'new' database to AmChecks:

Step 1: Create a new entry for the database server in table `'am_server'` if an entry does not already exist.

Step 2: Add a new entry for the database in table `'am_database'`.

Step 3: Logon to the database to be monitored and create user amu plus it's associated profile (this is the first section in script 'users.sql'. There is no need to create the amo user so just cut-n-paste the first few SQL statements. You'll want something like this:

```
CREATE PROFILE am_profile LIMIT
     PASSWORD_LIFE_TIME  UNLIMITED
     PASSWORD_GRACE_TIME DEFAULT
     PASSWORD_REUSE_MAX  UNLIMITED
     PASSWORD_REUSE_TIME UNLIMITED
     PASSWORD_LOCK_TIME  DEFAULT
     FAILED_LOGIN_ATTEMPTS 10;

CREATE USER amu PROFILE am_profile IDENTIFIED BY change_this;
GRANT CONNECT, SELECT_CATALOG_ROLE, CREATE TABLE TO amu;
```

Step 4: Back on the master database, add entries to table `'am_scriptskip'` if you want to omit particular scripts from the usual checks that run against the database for any particular reason.

Step 5. Ensure that the 'new' password has an entry in the tnsnames.ora file for amchecks. The entry should be all on one line to make parsing consistent.

Back to The Testing..

Eventually let rip with all of the scripts on the one database, i.e. use the '`-d`' flag but NOT the '`-s`' flag and then one script on all databases ('`-s`' but not '`-d`').

At this point you may start exploring the SIDSKIP and SCRIPTSKIP tables…

Next, read the relevant READ_ME and any other likely looking text files in the installation directory. These SHOULD be more up-to-date than this document.

There will be some bugs and some things won't work for you whereas they do for me, probably because I've made some unintended assumption that works for my databases but not for yours. Sorry about that. If you do manage to fix any of these or you improve the code (which, quite frankly wouldn't be that difficult in some places..) please let me know as you may be helping others who use this script; more importantly you might be helping me ☺.

Important Tables and Columns

If you decide to keep using the scripts on anything other than an ad hoc basis you'll gain more familiarity with the AMChecks database. You are encouraged to add your own columns and tables over time. If you can share these with other DBAs then even better but please avoid using any features that you wouldn't find in a 'vanilla' Standard Edition (SE, SE1, SE2) database running on Linux.

Personally I find maintenance and enhancements of AMChecks a good way to keep my hand in should I have to work on other, non-Oracle, databases or, worse still, do non-DBA work!

If you don't have the luxury of this then here are the key tables and columns you should know about.

AM_SERVER

COLUMN: SERVER

The short name for the server (e.g. hostname –s). It must be unique.

COLUMN: DISABLED

This column appears on a lot of tables. As the name suggests, it's an easy way to disable that particular row, so setting this to 'Y' in this table will cause the server in question to be ignored (unless you are referencing it via a non-disabled row in another able e.g. am_database). One thing to watch out for – disabling a row in a 'skip' table disables the skip itself, not the thing you are skipping. I've not explained that very well but hopefully you know what I mean!

COLUMN: CLUSTER_NAME

This is optional. If you have some kind of container for your server, e.g. a VM Cluster then specify this here.

COLUMN: PHYSICAL SERVER

This is the full name including the domain name (e.g. hostname).

COLUMN: PING_DISABLED

Some servers may not be pingable for one reason or another. To skip just the ping tests (in *is_oracle_ok.ksh*) set this column to 'Y'.

VIRTUAL COLUMN: PHYSICAL_SERVER_ABBREV

This is derived from PHYSICAL SERVER and for many, or quite possibly all, of your entries it will be the same as column 'SERVER'.

AM CHECKS

AM_DATABASE

COLUMN: SERVER
The short name of the server where the database lives (join column to AM_SERVER).

COLUMN: XXX_IND
You can have as many 'IND' columns as you like. e.g. ***PRODUCTION_IND***. All IND columns should only have 'Y' and 'N' specified. Each column can be used to restrict/categorise which databases you run your scripts against. You can see a few examples of this in the sample cron. Again, you are encouraged to add your own IND columns as necessary.

COLUMN: OS_CHECKS_IND
This is different from the other IND columns (although it could be used in a similar fashion). It determines whether or not this database has an external table for that server's space reporting. Only one database on each server should have this set. See a later section for more details on this.

COLUMN: RUN_ORDER
This will determine when the database is processed when a list of databases is being processed.

AM_SCRIPTS

COLUMN: SCRIPT_ID
A unique identifier for each script

COLUMN: SCRIPT_TYPE
This is used by ***amchecks.ksh*** when running a series of related scripts. Scripts are normally categorised into either 'CHECK' which is a health check; 'SUMMARY' which is informative or 'SPECIAL' which describes scripts that don't fall into either of the previous two categories.

COLUMNS: DB_VERSION_FROM and DB_VERSION_TO
Some scripts don't work on older versions of the database. If you want to exclude scripts in this fashion set one or both of these columns. Number format should be three digits – e.g. ***101*** for version 10.1, ***110*** for version 11.0.

COLUMNS: PARAM1 and PARAM2
Columns to describe what parameters a script should take. Use the string '***DBNAME***' to get the actual database name passed as a parameter, else use a literal value.

COLUMN: RUN_ON_MASTER
Indicates that when processing through a list of scripts for a target this particular script needs to get it's information from the 'master' database. Normally the script will have a param (see PARAM1 above) of '***DBNAME***'

AM CHECKS

AM_SIDSKIP

When NOT to run scripts against a particular database. Useful for planned outages.

COLUMN: SIDSKIP_TYPE

DAILY – means skip the database for every day between the specified 'from' and 'to' dates/times.; SATURDAY, SUNDAY etc. – specifying an actual day will only exclude the scripts for that database on a particular day of the week

COLUMN: SIDSKIP_NOTES

The message to display when skipping a particular database.

COLUMN: DISABLED

If you have a database that periodically needs to be skipped but at irregular times it is worth having an open-ended entry here and just toggling the DISABLED column to 'Y' when you want it skipped rather than explicitly setting an outage window. However, be careful that youdon't forget to re-enable this when you want checks to start again (..I'm speaking from experience here!)

AM_SCRIPTSKIP

Which scripts should be excluded for specific databases (don't forget that you can also exclude scripts for specific versions of the database via the AM_SCRIPTS table itself).

AM_OS_SPACE

A brief history of filesystem space usage on database hosts. Used for alerting, nor capacity planning.

AM_TABLESPACE_SPACE

Tablespace growth information. Could be used for capacity planning.

AM CHECKS

Adding OS space checks to AMChecks

For each (target) server:

as oracle:

```
mkdir -p ~oracle/amchecks/external_tables
mkdir -p /tmp/amchecks
. oraenv
```
(pick your ONE database where the external table will 'live')

Copy the latest version of script *'os_space_check.ksh'* from the master AMChecks database server *~oracle/amchecks* directory to this server's *~oracle/amchecks* directory and make it executable.

as sys:

GRANT CREATE TABLE TO amu; (should already be set)

```
CREATE DIRECTORY amcheck_dir AS '/u01/app/oracle/amchecks/external_tables';
```
(or wherever your amchecks external tables will live)

```
GRANT ALL ON DIRECTORY amcheck_dir TO amu;
```

as amu:

```
CREATE TABLE AMU.AM_OS_SPACE_LOAD
(
  SERVER           VARCHAR2(30 CHAR),
  FILESYSTEM       VARCHAR2(200 CHAR),
  SIZEK            NUMBER(10),
  USEDK            NUMBER(10),
  AVAILK           NUMBER(10),
  PCTUSED          NUMBER(3),
  MOUNTPOINT       VARCHAR2(200 CHAR),
  DF_DOW           VARCHAR2(10 CHAR),
  DF_TIMESTAMP     VARCHAR2(40 CHAR)
)
ORGANIZATION EXTERNAL
(
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY AMCHECK_DIR
  ACCESS PARAMETERS
  (    RECORDS DELIMITED BY NEWLINE NOBADFILE NODISCARDFILE NOLOGFILE
       SKIP 0 FIELDS TERMINATED BY ', ' MISSING FIELD VALUES ARE NULL
  )
LOCATION (AMCHECK_DIR: 'am_os_space_load.dbf')
)
REJECT LIMIT UNLIMITED;
```

as oracle: `crontab -e`

```
# Amchecks crons:
##################
18   * * * *  ksh -c '/u01/app/oracle/amchecks/os_space_check.ksh 1>
/tmp/amchecks/os_space_check_last_output 2>&1'
```

(check/change full path to os_space_check.ksh)

Actually, change the cron so it runs while you wait (don't just run it from the command line) then change it to the time you really want it run..

as amu:

```
select * from AM_OS_SPACE_LOAD;
```
If all looks good, correct the cron and put your feet up.

as amo on master database: *UPDATE AM_DATABASE SET OS_CHECKS_IND = 'Y' WHERE…*
update the record for the database chosen in the first step)

AM CHECKS

Sample Cron:

```
#
# Amchecks below here
#
#########################################################################################################################################################################
## Database Connectivity checks should list everything once a day and check for errors only every 15 minutes
## Database check scripts - e.g. space, backups. Run these once in the morning, lunchtime and the evening with only error checks except the morning check
#########################################################################################################################################################################
18      * * * *  ksh -c '/home/oracle/amchecks/os_space_check.ksh 1> /tmp/amchecks/os_space_check_last_output 2>&1'
#########################################################################################################################################################################
########################
# Non-Prd Section
########################
04,34 06-18 * * 1-5  ksh -c '/home/oracle/amchecks/is_oracle_ok.ksh -v -e -c -s -m -l non-prdisook -h "Non-Prod Connectivity Test" -S90 PRODUCTION_IND+N 1> /tmp/amchecks/nonprod_is_oracle_ok_last_output
2>/tmp/amchecks/nonprod_is_oracle_ok_last_error'
24 08 * * 3  ksh -c '/home/oracle/amchecks/amchecks.ksh -c -m -x -t "CHECK" -h "Non-Prod Error Check" PRODUCTION_IND+N 1> /tmp/amchecks/nonprod_amchecks_nonprod_output 2>/tmp/amchecks/nonprod_amchecks_nonprod_error'
50 08 * * 3  ksh -c '/home/oracle/amchecks/amchecks.ksh -c -m -x -t "CHECK" -a "fred.flintstone.@bedrock.com" -h "Non-Prod Error Check" PRODUCTION_IND+N 1> /tmp/amchecks/nonprod_amchecks_nonprod_output
2>/tmp/amchecks/nonprod_amchecks_nonprod_error'
########################
# All Databases Section
########################
22      05 * * *  ksh -c '/home/oracle/amchecks/reconcile.ksh -m 1> /tmp/amchecks/reconcile_output 2>/tmp/amchecks/reconcile_last_error'
37      06 * * 1  ksh -c '/home/oracle/amchecks/amchecks.ksh -d ORCL -c -s cluster_summary.sql -H -h "VMWare Cluster Summary" -m 1> /tmp/amchecks/cluster_summary_last_output 2>/tmp/amchecks/cluster_summary_last_run'
31      06 * * *  ksh -c '/home/oracle/amchecks/amchecks.ksh -d ORCL1 -c -s rman_catalog_summary.sql -H -h "RMAN Backup Summary" -m 1> /tmp/amchecks/rman_backup_last_output 2>/tmp/amchecks/rman_backup_last_run'
35      06 * * 1  ksh -c '/home/oracle/amchecks/amchecks.ksh -d ORCL -c -s rman_speed.sql -H -h "RMAN Backup Speeds" -m 1> /tmp/amchecks/rman_speed_last_output 2>/tmp/amchecks/rman_speed_last_run'
29      08,12,16 * * *  ksh -c '/home/oracle/amchecks/is_oracle_ok.ksh -v -e -c -m 1> /tmp/amchecks/is_oracle_ok_last_output 2>/tmp/amchecks/is_oracle_ok_last_run'
32      07 * * *  ksh -c '/home/oracle/amchecks/is_sss_ok.ksh -M 1> /tmp/amchecks/is_sss_ok_last_output 2>/tmp/amchecks/is_sss_ok_last_run'
03,08,13,18,23,28,33,38,43,48,53,58 * * * *  ksh -c '/home/oracle/amchecks/is_sss_ok.ksh -S 1> /tmp/amchecks/is_sss_ok_last_output 2>/tmp/amchecks/is_sss_ok_last_error'
11 17 * * *  ksh -c '/home/oracle/amchecks/total_space_check.ksh -m 1> /tmp/amchecks/total_space_check_output 2>/tmp/amchecks/total_space_check_error'
31 17 * * *  ksh -c '/home/oracle/amchecks/tablespace_space_check.ksh -m 1> /tmp/amchecks/tablespace_space_check_output 2>/tmp/amchecks/tablespace_space_check_error'
00 05 * * *  ksh -c '/home/oracle/amchecks/amchecks.ksh -c -m 1> /tmp/amchecks/amchecks_last_output 2>/tmp/amchecks/amchecks_last_error'
53        06,09,12,15 * * 1-5  ksh -c '/home/oracle/amchecks/all_os_space_check.ksh -R -S -m 1> /tmp/amchecks/all_os_space_check_output
2>/tmp/amchecks/all_os_space_check_last_run'
58       09,12,15 * * 1-5 ksh -c '/home/oracle/amchecks/all_os_space_check.ksh -v -p -r -S -m -t "Prod OS Space Alerts" PRODUCTION_IND+Y 1>
/tmp/amchecks/all_os_space_check_output 2>/tmp/amchecks/all_os_space_check_last_run'
58       06       * * 1-5 ksh -c '/home/oracle/amchecks/all_os_space_check.ksh -v -p -S -m 1> /tmp/amchecks/all_os_space_check_output
2>/tmp/amchecks/all_os_space_check_last_run'
53        06 * * 0,6  ksh -c '/home/oracle/amchecks/all_os_space_check.ksh -v -p -S -m 1> /tmp/amchecks/all_os_space_check_output
2>/tmp/amchecks/all_os_space_check_last_run'

#####################
# Production Section
#####################
02,32 * * * 0,6  ksh -c '/home/oracle/amchecks/is_oracle_ok.ksh -v -e -c -s -m -l prdisook -h "Production Connectivity Test" -S90 PRODUCTION_IND+Y 1>
/tmp/amchecks/prod_is_oracle_ok_last_output 2>/tmp/amchecks/prod_is_oracle_ok_last_error'
02,07,12,17,22,27,32,37,42,47,52,57 * * * 1-5  ksh -c '/home/oracle/amchecks/is_oracle_ok.ksh -v -e -c -s -m -l prdisook -h "Production Connectivity Test" -S15
PRODUCTION_IND+Y 1> /tmp/amchecks/prod_is_oracle_ok_last_output 2>/tmp/amchecks/prod_is_oracle_ok_last_error'
04 6,16 * * *  ksh -c '/home/oracle/amchecks/amchecks.ksh -c -m -x -t "CHECK" -h "URGENT Production Error Check" PRODUCTION_IND+Y 1>
/tmp/amchecks/amchecks_last_output 2>/tmp/amchecks/amchecks_last_error'
00 07 * * 1  ksh -c '/home/oracle/amchecks/amchecks.ksh -c -m -h "Production Error Check" PRODUCTION_IND+Y 1> /tmp/amchecks/amchecks_last_output
2>/tmp/amchecks/amchecks_last_error'
```

Some Scripts Explained

is_oracle_ok.ksh

This is where it all began. A script originally designed to read through your *tnsnames.ora* file and *tnsping* all instances that it finds. Functionality was added to e-mail the results, then to connect using *sqlplus*, then to skip specific databases and then to have the database list held in a database.

Be careful about how you schedule multiple occurrences of this as they could trip over each other when reporting results.

```
################################## is_oracle_ok #######################################

Checking databases listed in DOUDBAM1...

        Processing AB01 (on pkamis (unxux1p0))                          Looks OK (0 msec)
        Processing ABDBZEA2 (on srvux123)                               Looks OK (80 msec)
        Processing AOPARCH1 (on srvux123)                               Looks OK (70 msec)
        Processing CAT92 (on unxux5)                                    Looks OK (10 msec)
        Processing CMDB (on pkcmdb (unxux2p2))                          Looks OK (0 msec)
        Processing CUSTSP (on custsp)                                   Looks OK (60 msec)
        Processing CUSTUAT1 (on PRDERM2DB01 [CL-VMT01])                 Looks OK (0 msec)
        Processing DOPRMAN1 (on PRDRMANDB01 [CL-VMT01])                 Looks OK (0 msec)
        Processing DOPUATA1 (on PRDTALDB01 [CL-VMT01])                  Looks OK (0 msec)
        Processing DOPTUAT2 (on PRDTALDB01 [CL-VMT01])                  Looks OK (0 msec)
        Processing DOUDBAM1 (on UATDDMDB01 [CL-VMT01])                  Looks OK (0 msec)
        Processing ELDBPRD (on unx0serv03 [CL:vCL-VMT Server])          Looks OK (10 msec)
        Processing FDB (on pkfdb (unxux1p2))                            Looks OK (10 msec)
        Processing GDB (on pkgdb (unxux1p2))                            Looks OK (0 msec)
        Processing GDB_NEW (on SITDB01 [CL-VMT01])                      Looks OK (0 msec)
        Processing LIVEDBRD (on unxux1p0)                               Looks OK (10 msec)
        Processing MDB (on pkmdb (unxux1p2))                            Looks OK (0 msec)
        Processing MDB_NEW (on UATNISDB01 [CL-VMT01])                   Looks OK (0 msec)
        Processing MART (on ukmart (unxux1p3))                          Looks OK (0 msec)
        Processing MARTN1 (on pkmartn1 (unxux1p3))                      Looks OK (0 msec)
        Processing MARTN2 (on unxux2p3)                                 Looks OK (10 msec)
        Processing PRDCRM1 (on SITDB01 [CL-VMT01])                      Looks OK (0 msec)
        Processing PRDDBX (on FINDB02 [CL-VMT01])                       Looks OK (10 msec)
        Processing PRDFINAL (on FINDB01 [CL-VMT01])                     Looks OK (10 msec)
        Processing PRDXL (on SITMAXWLDB01 [CL-VMT01])                   Looks OK (10 msec)
        Processing PRIN (on pkP6W8db (unxux1p0))                        Looks OK (0 msec)
        Processing RETAIL1 (on pkretdb (unxux1p0))                      Looks OK (0 msec)
        Processing TMPDB112 (on pktemp (unxux1p0))                      Looks OK (10 msec)
        Processing WNIS (on pkwnis (unxux1p0))                          Looks OK (0 msec)

Connectivity checks ran at Mon 27 Mar 2017 at 03:22:01 pm and completed in 14 seconds (Timeout=10 seconds)

################################## is_oracle_ok #######################################
```

AM CHECKS

amchecks.ksh

This is the main script. It's basically a harness to process a bunch of scripts listed in a database against a bunch of databases. It's pretty flexible but has mutated over the years to a point now where it would seriously benefit from a complete rewrite!

Be careful about file permissions. This script will run **any** sql script you tell it to (I thought about restricting this but this flexibility also makes the script really useful for ad hoc reports so no checks are currently in place). For this reason keep the file permissions on all sql and on the amchecks directory tight.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Running checks on db01 - Host: pkdb01 (unixabc1)                                    +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

db01 (Enterprise Edition - 9.2.0.8.0) running on unixabc1 in ARCHIVELOG mode

Started on Sun 11th Sep 2016 19:35:46 (87 Days 20 Hours 0 Minutes 52 Seconds)

Alert log       => /u01/app/oracle/admin/db01/bdump/alert_db01.log
log_archive_dest_1    => location=/u01/app/oracle/admin/db01/arch mandatory reopen


Account expiry for the AMCheck user(s)
--------------------------------------------------------------------------------
The password for AMU expires on ** No Expiry Date Set **

Tablespace Space Usage
======================

Tablespace                Size (Mb)  Used (Mb)  Free (Mb)   %Used                        Theory Max (Mb)  %Theory
------------------------- ---------- ---------- ---------- ------ ---------------------- ---------------  -------  ----------------------
ADMIN                           512         88        424  17.22 [XXX-----------------]             512    17.22  [XXX-----------------]
db01_IDXMSF                   4,096      3,385        711  82.65 [XXXXXXXXXXXXXXXX----]          32,768    10.33  [XX------------------]
db01_TABMSF                   9,216      8,033      1,183  87.16 [XXXXXXXXXXXXXXXXX---]           9,216    87.16  [XXXXXXXXXXXXXXXXX---]
FDBA_DATA                     3,072      2,269        803  73.87 [XXXXXXXXXXXXXX------]           3,072    73.87  [XXXXXXXXXXXXXX------]
FDBA_INDEX                    1,024        812        212  79.33 [XXXXXXXXXXXXXXX-----]           1,024    79.33  [XXXXXXXXXXXXXXX-----]
MLOG                            250          0        250   0.03 [--------------------]             250     0.03  [--------------------]
SYSTEM                          250        176         74  70.56 [XXXXXXXXXXXXXX------]             250    70.56  [XXXXXXXXXXXXXX------]
TEMP (t)                      1,024      1,024          0 100.00 [XXXXXXXXXXXXXXXXXXXX]           1,024   100.00  [XXXXXXXXXXXXXXXXXXXX]
TOOLS                           512        108        404  21.09 [XXXX----------------]             512    21.09  [XXXX----------------]
UNDO (u)                      1,024         19      1,005   1.84 [--------------------]           1,024     1.84  [--------------------]
USERS                            50         23         27  46.25 [XXXXXXXXX-----------]              50    46.25  [XXXXXXXXX-----------]
WZABC_IDXMSF                    128          2        126   1.27 [--------------------]             128     1.27  [--------------------]
WZABC_TABMSF                    128          3        125   2.59 [--------------------]             128     2.59  [--------------------]
------------------------- ---------- ---------- ----------
Total                        21,286     15,944      5,342


Table stats (age by schema)
===========================

Table Owner          Table Count     <1 Week   <1 Month  <6 Months  >6 Months   NO STATS
-------------------- ----------- ---------- ---------- ---------- ---------- ----------
ADMIN                        (1)          0          0          0          0          1
ABRIMA                     (187)          0          0          0        185          2
AMU                          (1)          0          0          0          0          1
ELPHNNT                  (1,249)          0          0          0       1218         31
FDBA                       (137)          0          0          0        134          3
OEMADM                       (4)          0          0          0          0          4
PATROL                      (15)          0          0          0          0         15
PERFSTAT                    (41)          0          0          0          0         41
WZABC                       (19)          0          0          0         19          0

Hourly breakdown of recent redo log generation (redo logs are 20 MB each)
==========================================================================

Day                 00  01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23
------------------- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
Thursday  01-12-16   0   0   2   0   0   1   0   0   0   1   0   1   0   1   0   1   1   1   0   0   0   0   0   0
Friday    02-12-16   0   0   2   0   0   1   0   0   0   1   0   1   0   1   0   1   0   1   0   0   0   0   0   0
Saturday  03-12-16   1   0   2   1   0   1   0   0   0   1   0   1   0   0   0   0   0   1   0   2   1   0   0
Sunday    04-12-16   1   0   0   1   0   1   0   0   0   1   0   1   0   1   0   0   0   2   0   0   0   1   0   0
Monday    05-12-16   0   0   0   0   0   1   0   0   1   1   0   1   0   1   0   2   0   1   0   0   0   0   0   0
Tuesday   06-12-16   0   0   2   0   0   1   0   0   0   1   0   1   0   1   0   1   0   1   0   0   0   0   0   0
Wednesday 07-12-16   0   0   2   0   0   1   0   0   0   1   0   1   0   1   0   1   0   1   0   0   0   0   0   0
Thursday  08-12-16   0   0   2   0   0   1   0   0   0   3   0   1   0   1   0   0   0   0   0   0   0   0   0   0


Backup Type      Latest Backup
---------------  -----------------
Archivelog       08-12-16 15:31:50
DB               08-12-16 09:27:27
```

AM CHECKS

all_os_space_check.ksh

This is an attempt to move out of the database and to do some host checks. I'm not too keen on this as it means you have to spend a little effort creating external tables on your monitored databases. Still, it's there if you don't have anything else looking after disk space; you might need it (as I did). It can be useful but can also be prone to spamming.

Space Summary Alerts **Sat 1 oct 06:53**

OS Space Alerts

Server	Mountpoint	Today	Yesterday	Last Wk	Last Mth	Day Growth	Wk Growth	Mth Growth	
UNIXDB01	/u05/oradata/HR1PRD	100%	33%	5%	4%	67%	95%	96%	*ERROR

Note: If no yesterday/weekly/monthly data is found, values show are rounded to the nearest date to make comparisons possible

Recent Daily High Water Marks

Server	Mountpoint	PCUsed	Date
UNIXDB01	/u05/oradata/HR1PRD	4%	Sat 17-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Sun 18-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Mon 19-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	2%	Tue 20-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Wed 21-09-16 04:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Thu 22-09-16 04:18:01
UNIXDB01	/u05/oradata/HR1PRD	5%	Fri 23-09-16 04:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Sat 24-09-16 04:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Sun 25-09-16 00:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Mon 26-09-16 04:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Tue 27-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Wed 28-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Thu 29-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	4%	Fri 30-09-16 03:18:01
UNIXDB01	/u05/oradata/HR1PRD	16%	Fri 30-09-16 09:18:01
UNIXDB01	/u05/oradata/HR1PRD	25%	Fri 30-09-16 12:18:01
UNIXDB01	/u05/oradata/HR1PRD	33%	Fri 30-09-16 15:18:01
UNIXDB01	/u05/oradata/HR1PRD	100%	Sat 01-10-16 03:18:01

AM CHECKS

tablespace_space_check.ksh and total_space_check.ksh

These are space reports showing how your databases are growing. They'll be a bit useless until you've been capturing details for a few months.

Space Summary Sun 2 Oct 17:33

Database Growth

Database	Date Now	Space (GB)	Last Mth(GB)	Month Growth		6 Mth(GB)	6 Months Growth		Last Yr(GB)	Year Growth	
ADB01	02-OCT-16	14.50	14.47	0.03(GB)	0.21%	14.29	0.21(GB)	1.47%	14.12	0.38(GB)	2.69%
ADB02	02-OCT-16	12.86	12.88	-0.02(GB)	-0.16%	13.06	-0.20(GB)	-1.53%	12.85	0.01(GB)	0.08%
ADB03	02-OCT-16	13.50	13.50	0.00(GB)	0.00%	13.50	0.00(GB)	0.00%	13.50	0.00(GB)	0.00%
ADB04	02-OCT-16	23.31	22.73	0.58(GB)	2.55%	21.89	1.42(GB)	6.49%	21.89	1.42(GB)	6.49%
BDB01	02-OCT-16	6.43	4.25	2.18(GB)	51.29%	3.71	2.72(GB)	73.32%	3.71	2.72(GB)	73.32%
BDB02	02-OCT-16	11.91	11.91	0.00(GB)	0.00%	11.91	0.00(GB)	0.00%	11.91	0.00(GB)	0.00%
BDB03	02-OCT-16	11.47	2.45	9.02(GB)	368.16%	2.46	9.01(GB)	366.26%	2.46	9.01(GB)	366.26%
BDB04	02-OCT-16	2.72	2.72	0.00(GB)	0.00%	2.72	0.00(GB)	0.00%	2.72	0.00(GB)	0.00%
CAT1	02-OCT-16	1.18	1.16	0.02(GB)	1.72%	1.17	0.01(GB)	0.85%	1.24	-0.06(GB)	-4.84%
CAT2	02-OCT-16	2.23	2.11	0.12(GB)	5.69%	2.19	0.04(GB)	1.83%	2.08	0.15(GB)	7.21%
CAT3	02-OCT-16	0.69	0.69	0.00(GB)	0.00%	0.66	0.03(GB)	4.55%	4.82	-4.13(GB)	-85.68%
CMDB1	02-OCT-16	19.11	18.79	0.32(GB)	1.70%	16.35	2.76(GB)	16.88%	16.35	2.76(GB)	16.88%
..											
ZOODB01	02-OCT-16	66.96	66.97	-0.01(GB)	-0.01%	66.87	0.09(GB)	0.13%	66.79	0.17(GB)	0.25%

Database	Date	Space (GB)
ADB01	30-NOV-15	14.12
	07-DEC-15	14.16
	11-JAN-16	14.22
	01-FEB-16	14.25
	14-MAR-16	14.28
	25-APR-16	14.32
	31-MAY-16	14.35
	01-JUN-16	14.35
	03-JUL-16	14.40
	05-AUG-16	14.47
	29-SEP-16	14.51
	02-OCT-16	14.50
ADB02	11-JAN-16	12.85
	15-FEB-16	12.85
	07-MAR-16	12.85
	04-APR-16	13.06
	19-MAY-16	12.85
	22-JUN-16	12.85
	01-JUL-16	12.86
	02-AUG-16	12.86
	05-SEP-16	12.88
	01-OCT-16	12.86
..		

Note that if historical space details aren't found the details used for the space calculations above will be based on the nearest date since the specified time.

Tablespace Space Summary Mon 5 Dec 17:49

Database Growth By Tablespace

Database	Tablespace	Date Now	Space (MB)	Last Mth(MB)			Last Yr(MB)		
DB01 on hatux1p0	ADMIN	05-DEC-2016	88	88			88		
	DB01_IDXMSF	05-DEC-2016	3,385	3,385			3,385		
	DB01_TABMSF	05-DEC-2016	8,033	8,033			8,033		
	FDCS_DATA	05-DEC-2016	2,267	2,267			2,249	18(MB)	0.81%
	FDCS_INDEX	05-DEC-2016	808	807	1(MB)	0.12%	796	12(MB)	1.52%
	MLOG	05-DEC-2016	0	0			0		
	SYSTEM	05-DEC-2016	176	176			176		
	TEMP(T)	05-DEC-2016	1,024	1,024			1,024		
	TOOLS	05-DEC-2016	108	108			108		
	UNDO(U)	05-DEC-2016	31	21	10(MB)	46.14%	50	-19(MB)	-38.78%
	USERS	05-DEC-2016	23	23			23		
	WPPRD_IDXMSF	05-DEC-2016	2	2			2		
	WPPRD_TABMSF	05-DEC-2016	3	3			3		

AM CHECKS

SQL Scripts

102_instance_summary.sql
amazon_disclaimer.sql
am_expire.sql
am_schema.sql
archivelog_check.sql
blocking.sql
broken_jobs.sql
cluster.sql
cluster_summary.sql
database_space.sql
dba_feature_usage.sql
dblink_summary.sql
dfspace_check.sql
growth_report.sql
instance_summary.sql
license_pack_usage_details.sql
login.sql
metric_check.sql
metric_report.sql
metrics.sql
options_packs_usage_statistics.sql
os_space_check.sql
os_space_summary.sql
pa_patches.sql
parameter_summary.sql
password_expiry_check11.sql
password_expiry_check.sql
preaudit.sql
prompt.sql
r12_checks.sql
R12PROD_disclaimer.sql
R12_stats_disclaimer.sql
recent_creations.sql
redo_activity.sql
redo_check.sql
rman_catalog_summary.sql
rman_check_10.sql
rman_check_r12.sql
rman_disclaimer.sql
rman_speed_db.sql
rman_speed.sql
rman_summary_10.sql
rman_summary_r12.sql
segment_growth.sql
space_check.sql
space_summary.sql
sptrends.sql
standard_checks_header.sql
standby_check.sql
stats_check_10.sql
stats_check_9.sql
stats_check.sql
tableinfo.sql
tablespace_growth_check.sql
tablespace_growth_report.sql
total_space.sql
totrace.sql
true_space_check.sql
true_space_gig.sql
true_space_meg.sql
users.sql
version.sql
when_analyzed.sql

Writing Your Own Scripts

These scripts, particularly the wrapper shell script *amchecks.ksh* can be a handy tool for running ad hoc queries against all of your databases (or logical groupings of databases using the 'IND' columns in table *amo.am_database*). However, this is clearly open to abuse as you can run any crappy code against your nice highly tuned production instance's data dictionaries!

It's common sense really but run any new code against non-production databases first, preferably starting with just one database and then running on larger numbers of databases.

If you want 1 line output for every database rather than a page full of output for each database target you may want to turn off the database title/name and suppress timings by using the *-D* and *-f* flags on *amchecks.ksh* although you'll probably want to select the database name some other way in order to make the output useful.

For example, if you want to produce a list of how long all of your databases have been up for, you could create something like:

adhoc99.sql:

```
SET LINES 140
SET PAGES 0

COL instance_started FORMAT A80 HEADING "Instance Started"
COL uptime           FORMAT A55 HEADING "Uptime"

SELECT v.instance_name || ' running on ' ||
       NVL(SUBSTR(v.host_name,1, INSTR(v.host_name, '.')-1),v.host_name) || ' started on ' ||
       TO_CHAR(v.startup_time, 'Dy ddth Mon YYYY "at" hh24:mi:ss') AS instance_started,
       '(up ' || TO_CHAR(FLOOR((SYSDATE-v.startup_time)) || ' Days ' ||
              MOD(FLOOR((SYSDATE-v.startup_time)*24),24) || ' Hours ' ||
              MOD(FLOOR((SYSDATE-v.startup_time)*24*60),60) || ' Minutes '||
              MOD(FLOOR((SYSDATE-v.startup_time)*24*60*60),60) || ' Seconds' ||')') AS uptime
FROM sys.v_$instance v;

exit;
```

Run this script via: *amchecks.ksh -Df -s adhoc99.sql -m*

html output will look something like:

```
################################################## amchecks ##################################################

APPLE running on fruitsv1 started on Sun 11th Sep 2016 at 19:35:46              (up 114 Days 19 Hours 7 Minutes 26 Seconds)
APRICOT running on fruitsv2 started on Sun 01st Jan 2017 at 16:34:14            (up 2 Days 22 Hours 8 Minutes 58 Seconds)
BANANA running on fruitsrv3 started on Sun 11th Sep 2016 at 19:35:46            (up 114 Days 19 Hours 7 Minutes 26 Seconds)
CHERRY running on fruitsrv1 started on Sun 01st Jan 2017 at 16:34:14            (up 2 Days 22 Hours 8 Minutes 58 Seconds)
…
################################################## amchecks ##################################################
```

Thanks

Thanks to the poor unfortunate DBAs who had to work with me and put up with odd behaviour (from me as well as the scripts) and for the Linux guys who helped me with setting up the mail.

Thanks to Charles Berlin for his O'Rly? Generator (https://dev.to/rlyy) for the cover page!

Thanks to any brave/reckless souls who actually download and use these scripts!