



VISCOSITY
NORTH AMERICA

www.viscosityna.com



AutoUpgrade Myths and Methods

Sean Scott



25 years working with Oracle technology

UTOUG Board ∴ RAC SIG Board

Oracle OpenWorld ∴ Collaborate/IOUG ∴ Regional UG

RAC/MAA ∴ DR/HA ∴ TFA/AHF ∴ Exadata/ODA

Automation ∴ DevOps ∴ Containers ∴ Virtualization

Ultramarathon Running ∴ Bouldering





Why Upgrade?

2009: Oracle 11gR2 Released. Also...



bing

“In **May 2009**, Facebook overtook MySpace in U.S.-based web traffic for the first time.”

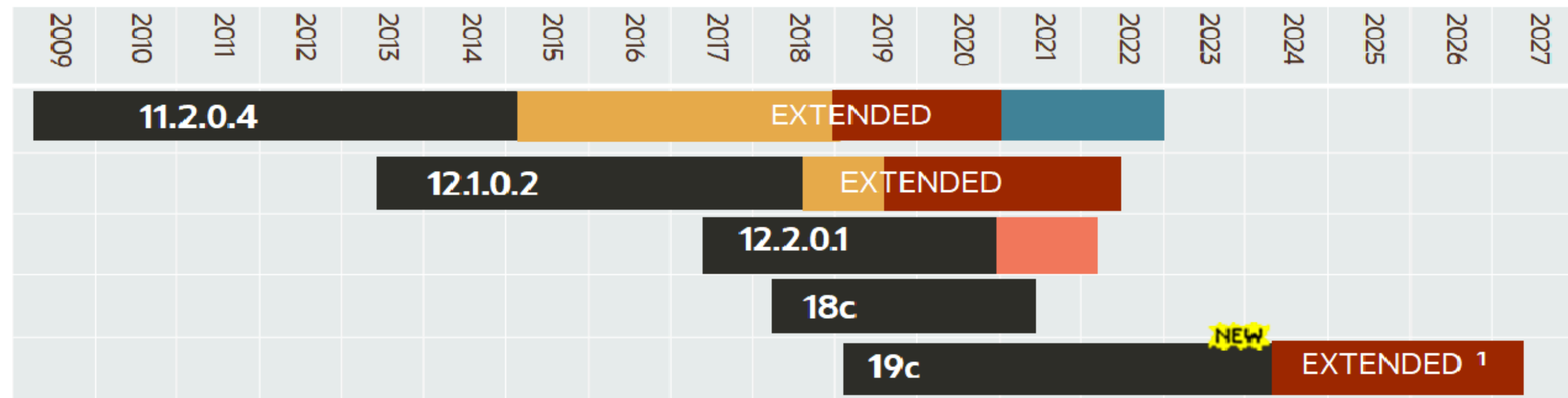
2013: Oracle 12cR1 Released. Also...

“Twerk” and “Selfie”
added to the dictionary



* Live action comedy released in the three-month window between Melissa McCarthy films “Identity Thief” and “The Heat”

Release and Support Timelines



Error Correction during Premier Support
 Error Correction with Waived Extended Support Fee
 Error Correction with Paid Extended Support
 Error Correction with Paid Market Driven Support
 Limited Error Correction

¹Oracle Database 19c is the most current Long Term Release.

Oracle 19c Support Roadmap

- Oracle **strongly recommends** customers upgrade to Oracle Database 19c
- Terminal release for Oracle Database 12c Release 2
- No Extended Support for 12.2.0.1 or 18c
- Extended and Long-Term Support will only be available for 19c
 - Premier Support through April 2024 *
 - Extended Support through April 2027 *

“Oracle 19c is the 11.2.0.4 of the 12c family”

* Per MOS Note 742060.1, updated August 3 2020



Why AutoUpgrade?

The Perfect Upgrade

- Repeatable: Reliable results every time
- Configurable: Central inventory for variables
- Automatic: Minimum operator interaction
- Visible: Logs diagnostics and status
- Scoped: Analyze, fix, deploy
- Inclusive: Runs all operations from pre- through post-
- Resumable: Simple restart on error
- Durable: Built in protection, retry, rollback
- Licensed; Supported; Performant

Available Tools

- Command line scripts
- PUU: Parallel Upgrade Utility
- Data Pump
- CTAS: Create Table as SQL
- ZDG: Zero Downtime Upgrade with GoldenGate
- FPP: Fleet Patching & Provisioning—formerly Rapid Home Provisioning
- DBUA: Database Upgrade Assistant—GUI, Silent
- AutoUpgrade

	Repeatable	Configurable	Automatic	Visible	Scoped	Inclusive	Resumable	Durable	Licensed	Supported	Performant
Scripts	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	✘	✘	⚠
PUU	⚠	✘	✘	✔	✘	✘	⚠	✘	✔	✔	✔
Data Pump	⚠	⚠	✘	✔	✘	✘	✘	✘	✔	✔	✘
CTAS	✘	✘	✘	✘	✘	✘	✘	✘	✔	⚠	✔
ZDT	✔	✔	✔	✔	⚠	✘	⚠	⚠	✔	✔	✔
FPP	✔	✔	✔	✔	⚠	✘	⚠	⚠	✔	✔	✔
DBUA - GUI	✘	⚠	✘	✔	✔	✔	✔	✔	✔	✔	✔
DBUA - Silent	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
AutoUpgrade	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

AutoUpgrade vs. DBUA

DBUA

- Parameters on command line
- Difficult to resume
- Good error correction
- Good pre-checks, fixes
- GRP optional
- Pre-, post-tasks optional
- Works w/NOARCHIVELOG

AutoUpgrade

- Fine-grained configuration
- Easy to resume
- Excellent error correction
- Better pre-checks, fixes
- GRP by default
- Pre-, post-tasks by default
- Superior feedback
- Better diagnostic files

AutoUpgrade: Myths & Misconceptions

- ❌ Only for large environments
- ❌ Difficult to configure for <10 databases
- ❌ Limited upgrades
- ❌ Hard to use
- ❌ Slow
- ❌ Forces Multi-tenant
- ✅ Not RAC friendly
(See note 2485457.1)



Get AutoUpgrade!

- Included with Oracle 19.3 and newer (\$OH/rdbms/admin)
- Download:
MOS Note 2485457.1
- No extra license
- Requires Java 8+
- ~4MB jar file





AutoUpgrade in Action

AutoUpgrade in Action

```
java -jar autoupgrade.jar -config config.txt -mode deploy
```



AutoUpgrade in Action — Modes

- Analyze
 - Read-only analysis of a database
 - Identifies issues
- Fixups
 - Performs automatic fixes
 - Identifies items for manual intervention

AutoUpgrade in Action — Modes

- Upgrade
 - When running on a different server without a source ORACLE_HOME
 - Runs upgrade, post-upgrade checks and fixes only
 - No GRP, no post-upgrade actions

AutoUpgrade in Action — Modes

- Deploy
 - Setup, pre-upgrade steps and checks
 - Creates GRP
 - Performs pre-upgrade fixes
 - Drains load
 - Upgrades database
 - Post-upgrade steps, fixes and action

AutoUpgrade in Action — Configuration

```
java -jar autoupgrade.jar -create_sample_file config.txt
```



AutoUpgrade in Action — Configuration

```
cat << EOF > /opt/oracle/autoupgrade/config.txt
# Global parameters
global.autoupg_log_dir=/opt/oracle/autoupgrade

# Database parameters
vna1.upgrade_node=$(hostname -s)
vna1.sid=VNA
vna1.start_time=now
#vna1.pdbs=*
vna1.run_utlrp=yes
vna1.timezone_upg=yes
vna1.target_version=19.8
vna1.target_home=/opt/oracle/product/19c/dbhome_1
vna1.source_home=/opt/oracle/product/12.1.0.2/dbhome_1
EOF
```



AutoUpgrade in Action — Configuration

- Two types of parameters:
 - Global - parameters used for all upgrades
 - AutoUpgrade log directory
 - Consistent values in all environments, e.g. target ORACLE_HOME
 - Local - parameters specific to a database
 - Values unique to an “upgrade”
 - Host, database name, SID

AutoUpgrade in Action — Configuration

```
# Global parameters
global.autoupg_log_dir=/opt/oracle/autoupgrade
global.target_home=/opt/oracle/product/19c/dbhome_1
global.target_version=19.8

# Database parameters
vna1.upgrade_node=autoup
vna1.sid=VNA
vna1.start_time=now
vna1.run_utlrp=no
vna1.timezone_upg=yes
vna1.log_dir=/opt/oracle/autoupgrade/VNA
vna1.source_home=/opt/oracle/product/12.1.0.2/dbhome_1
```

AutoUpgrade in Action — Configuration

```
# Global parameters
```

```
global.target_home=/opt/oracle/product/19c/dbhome_1
```

```
global.target_version=19.8
```

```
# Database 1
```

```
vna1.upgrade_node=node1
```

```
...
```

```
# Database 10
```

```
vna10.upgrade_node=node10
```

```
vna10.target_version=19.6
```

```
vna10.target_home=/opt/oracle/product/19c/dbhome_10
```


AutoUpgrade in Action — Configuration

```
# Global parameters
```

```
...  
global.add_during_upgrade_pfile=/opt/oracle/init.add.during.ora
```

```
# Database parameters
```

```
...  
vna1.add_during_upgrade_pfile=/opt/oracle/initVNA.add.during.ora  
vna1.del_during_upgrade_pfile=/opt/oracle/initVNA.del.during.ora  
...  
vna1.add_after_upgrade_pfile=/opt/oracle/initVNA.add.after.ora  
vna1.del_after_upgrade_pfile=/opt/oracle/initVNA.del.after.ora
```


AutoUpgrade in Action — Configuration

```
# Global parameters
```

```
...
```

```
global.remove_underscore_parameters=yes
```



AutoUpgrade in Action — Configuration

```
# Global parameters
```

```
...
```

```
global.before_action=/opt/oracle/pre_upgrade.ALL.sh
```

```
# Database parameters
```

```
...
```

```
vna1.before_action=/opt/oracle/pre_upgrade.VNA.sh
```

```
vna1.after_action=/opt/oracle/post_upgrade.VNA.sh
```

```
...
```

```
vna1.add_after_upgrade_pfile=/opt/oracle/initVNA.add.after.ora
```

```
vna1.del_after_upgrade_pfile=/opt/oracle/initVNA.del.after.ora
```



AutoUpgrade Console

AutoUpgrade Console

```
$ java -jar autoupgrade.jar -config config.txt -mode analyze
AutoUpgrade tool launched with default options
Processing config file ...
+-----+
| Starting AutoUpgrade execution |
+-----+
1 databases will be analyzed
Type 'help' to list console commands
upg>
```

AutoUpgrade Console

```
upg> Job 100 completed
```

```
----- Final Summary -----
```

```
Number of databases          [ 1 ]
```

```
Jobs finished successfully   [1]
```

```
Jobs failed                   [0]
```

```
Jobs pending                  [0]
```

```
----- JOBS FINISHED SUCCESSFULLY -----
```

```
Job 100 for VNA
```

```
$
```


AutoUpgrade Console

```
exit                // To close and exit
help                // Displays help
lsj [(-r|-f|-p|-e) | -n <number>] // list jobs by status up to n elements.
    -f Filter by finished jobs.
    -r Filter by running jobs.
    -e Filter by jobs with errors.
    -p Filter by jobs being prepared.
    -n <number> Display up to n jobs.

lsr                 // Displays the restoration queue
lsa                 // Displays the abort queue
tasks               // Displays the tasks running
clear               // Clears the terminal
```

AutoUpgrade Console

```
resume -job <number> // Restarts a previous job that was running
status [-job <number> [-long]] // Lists all the jobs or a specific job
restore -job <number> // Restores the database to its state prior to the upgrade
restore all_failed // Restores all failed jobs to their previous states prior
to the upgrade
logs // Displays all the log locations
abort -job <number> // Aborts the specified job
h[ist] // Displays the command line history
/[<number>] // Executes the command specified from the history. The
default is the last command
```


AutoUpgrade Console

```
resume -job <number> // Restarts a previous job that was running
status [-job <number> [-long]] // Lists all the jobs or a specific job
restore -job <number> // Restores the database to its state prior to the upgrade
restore all_failed // Restores all failed jobs to their previous states prior
to the upgrade
logs // Displays all the log locations
abort -job <number> // Aborts the specified job
h[ist] // Displays the command line history
/[<number>] // Executes the command specified from the history. The
default is the last command
```



Monitoring AutoUpgrade

AutoUpgrade Logs

```
`-- VNA
  |-- 100
    |-- autoupgrade_20200902.log
    |-- autoupgrade_20200902_user.log
    |-- autoupgrade_err.log
    `-- prechecks
      |-- prechecks_vna.log
      |-- vna_checklist.cfg
      |-- vna_checklist.json
      |-- vna_checklist.xml
      |-- vna_preupgrade.html
      `-- vna_preupgrade.log
  `-- temp
    |-- after_upgrade_pfile_VNA.ora
    |-- before_upgrade_pfile_VNA.ora
    `-- during_upgrade_pfile_VNA.ora
```

```
upg> Job 100 completed
----- Final Summary -----
Number of databases          [ 1 ]

Jobs finished successfully   [1]
Jobs failed                  [0]
Jobs pending                  [0]
----- JOBS FINISHED SUCCESSFULLY -----
Job 100 for VNA
```

AutoUpgrade Logs

```
|-- 101
| |-- autoupgrade_20200903.log
| |-- autoupgrade_err.log
| |-- dbupgrade
| | |-- autoupgrade20200903011356vna.log
| | |-- catupgrd20200903011356vna0.log
| | |-- catupgrd20200903011356vna_datapatch_upgrade.log
| | |-- catupgrd20200903011356vna_stderr.log
| | |-- during_upgrade_pfile_catctl.ora
| | |-- phase.log
| | |-- upg_summary.log
| | |-- vna_autocompile20200903011356vna0.log
| |-- drain
| | `-- drain_vna.log
| |-- postchecks
| | |-- postchecks_vna.log
| | |-- vna_checklist.cfg
| | |-- vna_checklist.json
| | |-- vna_checklist.xml
| | |-- vna_postupgrade.html
| | `-- vna_postupgrade.log
| |-- postfixups
| | |-- postchecks_vna.log
| | `-- postfixups_vna.log
| |-- postupgrade
| | `-- postupgrade.log
| |-- prechecks
| | |-- prechecks_vna.log
| | |-- vna_checklist.cfg
| | |-- vna_checklist.json
| | |-- vna_checklist.xml
| | |-- vna_preupgrade.html
| | `-- vna_preupgrade.log
| |-- prefixups
| | |-- prechecks_vna.log
| | `-- prefixups_vna.log
| `-- preupgrade
|     `-- preupgrade.log
|-- temp
```




Demo

Monitor AutoUpgrade Status

- Start a simple web server on the upgrade host:

```
cd $ORACLE_BASE/autoupgrade/cfgtools/upgrade/auto  
python -m SimpleHTTPServer 8080
```
- View the upgrade status page:
<http://upgradehost:8080/state.html>



Recommendations, Preparations & Opportunities

AutoUpgrade Recommendations

- Check DB_RECOVERY_FILE_DEST_SIZE and DB_RECOVERY_FILE_DEST
- Run in analyze and review
- Fix as much as possible ahead of time
 - Run ultrp.sql
 - Collect dictionary, fixed object status
 - Empty recycle bin
- Re-run analyze; repeat

AutoUpgrade Recommendations

- When migrating:
 - Run analyze and fixups on source
 - Run upgrade on target
- *Practice, practice, practice!*

Upgrade Preparations

- Increase AWR retention
- Remove SQLPATH, glogin.sql, login.sql
- Backup
 - TNS files
 - Parameter, password files, oratab
 - ORACLE_HOME
- Profile IO and CPU loads
- OraChk

Upgrade Preparations

- Check for obsolete features - APEX, Streams
- Remove unused features, demo schemas
- Review SCHEDULER_JOBS
- Backup wallets
 - Convert ACL to ACE
- Confirm backups by restoring them
- Review optimizer changes
- PRACTICE

Opportunities

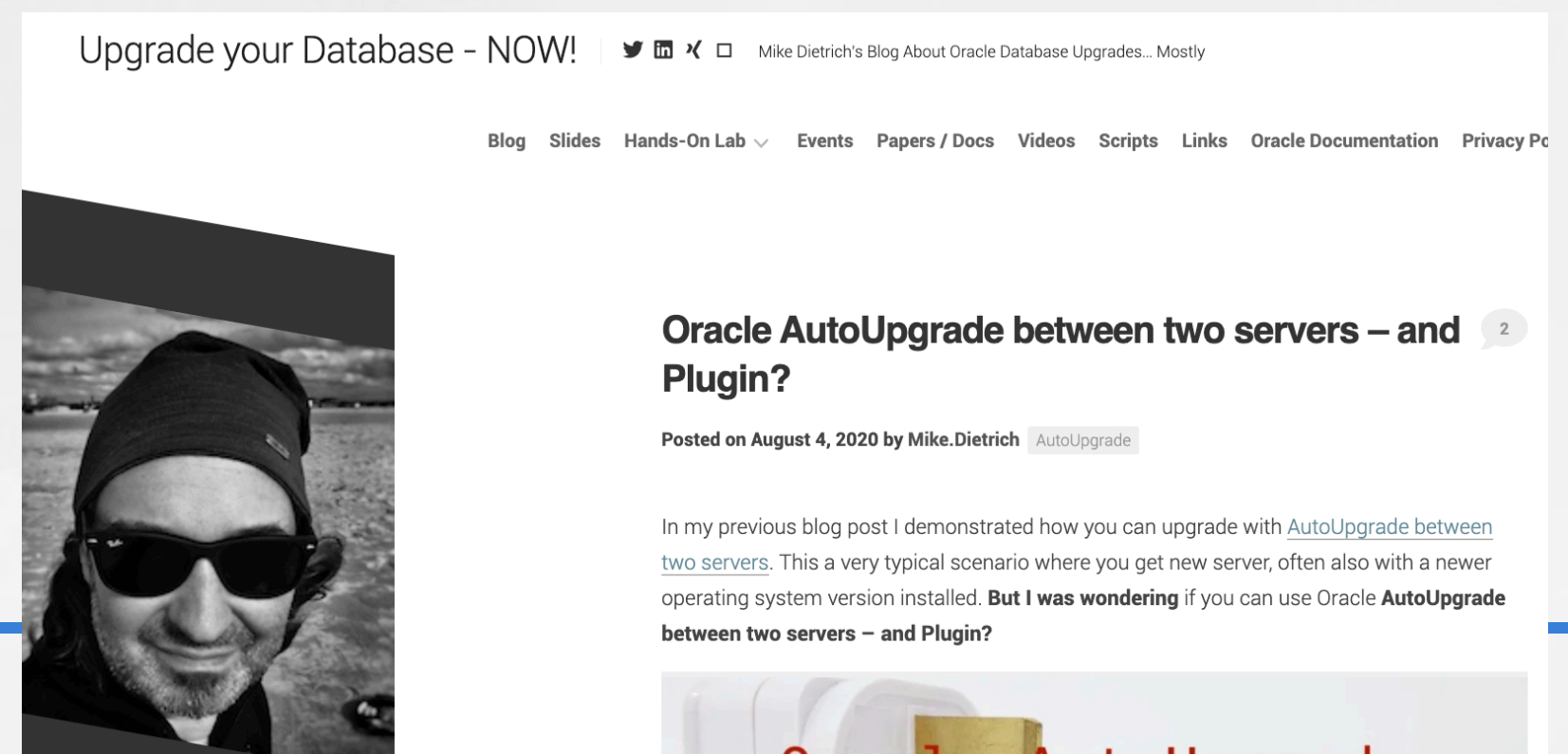
- Review documentation
- Validate backup/recovery procedures
 - *Backups don't guarantee recovery!*
- Change credentials for service accounts
- Upgrade wallets & certificates
- Upgrade listener security, client/host allowed versions
- Address technical debt



Questions

For More Information...

- [Database Upgrade Guide — Using Oracle Database AutoUpgrade](#)
- Upgrade Compatibility Matrix: [MOS Note 551141.1](#)
- AutoUpgrade Tool: [MOS Note 2485457.1](#)
- Pre-Upgrade JAR: [MOS Note 884522.1](#)
- <https://mikedietrichde.com/>



The screenshot shows a blog post from Mike Dietrich's website. The page title is "Upgrade your Database - NOW!". The main heading of the post is "Oracle AutoUpgrade between two servers – and Plugin?". The post is dated August 4, 2020, and is categorized under "AutoUpgrade". The text of the post begins with "In my previous blog post I demonstrated how you can upgrade with AutoUpgrade between two servers. This a very typical scenario where you get new server, often also with a newer operating system version installed. But I was wondering if you can use Oracle AutoUpgrade between two servers – and Plugin?". The author's name, Mike Dietrich, is visible in the top right corner of the post content. A small profile picture of the author is visible on the left side of the page.



oraclesean.com



<https://www.linkedin.com/in/soscott/>



@oraclesean



<https://github.com/oraclesean>



sean.scott@viscosityna.com



Search "OracleSean" on YouTube