Tiddalik & Nahri Moodle Installation Maintenance & Update Guide

The following guide supports the backup, update and upgrade processes required for Tiddalik and Nahri's LMS systems. To perform server processes you should be able to access the command line through putty or the cli using ssh. All credentials for servers and virtual machines are in the company Bitwarden app.

Table of Contents

Tiddalik & Nahri Moodle Installation Maintenance & Update Guide....................................1
- System Overview...2
  - Moodle Installations...2
  - Infrastructure...2
- Pre-Update Checklist...2
  - Performance Monitoring..2
- Step-by-Step Update Procedure..2
  - 1. Check Active Users..2
  - 2. Verify Plugin Status...2
  - 3. Test Plugin Compatibility (Major Version Updates)...................................3
  - 4. Access Server Environment..3
  - 5. Verify Backup Completion..3
  - 6. Switch to Domain User..3
  - 7. Navigate to Moodle Installation..3
  - 8. Perform Moodle Update...4
  - 9. Test Web Interface..4
  - 10. Update Additional Plugins..4
  - 11. Verify Cron Operations..4
  - 12. Post-Update Testing..5
- Troubleshooting..5
  - Common Issues...5
  - Recovery Procedures...5
- Important File Locations..5
  - Configuration Files..5
  - Backup Script Structure...5
  - Credentials Storage...6
- Regular Maintenance Schedule...6
  - Daily...6
  - Weekly...6
  - Monthly...6
  - Before Major Updates..6

System Overview

Moodle Installations

- **learn.tiddalik.com.au**: Production LMS (Moodle 4.4 branch) with Edwiser theme
- **staging.learn.tiddalik.com.au**: Testing platform (Moodle 4.5 branch) for trials and updates
- **learn.nahri.com.au**: Client LMS (production)
- **staging.learn.nahri.com.au**: Development site (may be retired)

Infrastructure

- **Host Server**: Davros (Azure VM)
- **Management**: Webmin/Virtualmin
- **Backup Storage**: Azure Blob Storage
- **Backup Script**: mdl_bu.sh (automated daily via cron)
  – see https://github.com/alextdel/mdl_bu_script

Pre-Update Checklist

Performance Monitoring

Regularly monitor server capacity:

- Page load times (even with minimal users this is currently slow - 12/6/2025)
- CPU utilisation
- Memory usage
- Disk capacity
- Consider dedicated Moodle hosting for future high-traffic scenarios

Step-by-Step Update Procedure

1. Check Active Users

Web Interface:

- Navigate to Site Administration dashboard
- Check the "Online Users" block in the admin sidebar
- Ensure no users are currently online before proceeding

2. Verify Plugin Status

Web Interface:

- Go to **Site Administration** → **Plugins** → **Plugins Overview**
- Check all plugins are current for the pre-update version
- Update any outdated plugins before proceeding with Moodle core update

3. Test Plugin Compatibility (Major Version Updates)

For full dot-point version updates:

- Verify all current plugins, scripts, and payment gateway plugins are compatible with target Moodle version
- Test on staging site (staging.learn.tiddalik.com.au) using identical plugin versions
- Pay particular attention to:
    - Edwiser theme compatibility
    - Payment gateway plugins
    - Custom functionality

4. Access Server Environment

Note: Server credentials stored in Bitwarden application

SSH Connection:

- Connect to Davros server via SSH/PuTTY (using ssh key preferably)

5. Switch to Domain User

Command Line:

```
su - [domain_admin_username]
# Enter password from Bitwarden
```

6. Verify Backup Completion

Check Backup Status:

- Confirm latest automated backup (mdl_bu.sh) has completed successfully
- Verify database and file backups are stored in Azure Blob Storage
- Check backup logs for any errors
- **Location**: Backup files stored as per mdl_bu.conf configuration

Manual Backup Verification:

```
# Check recent backups in backup directory
ls -la /path/to/backup/store
# Review latest backup log
tail -n 50 /path/to/backup/logs/latest.log
```

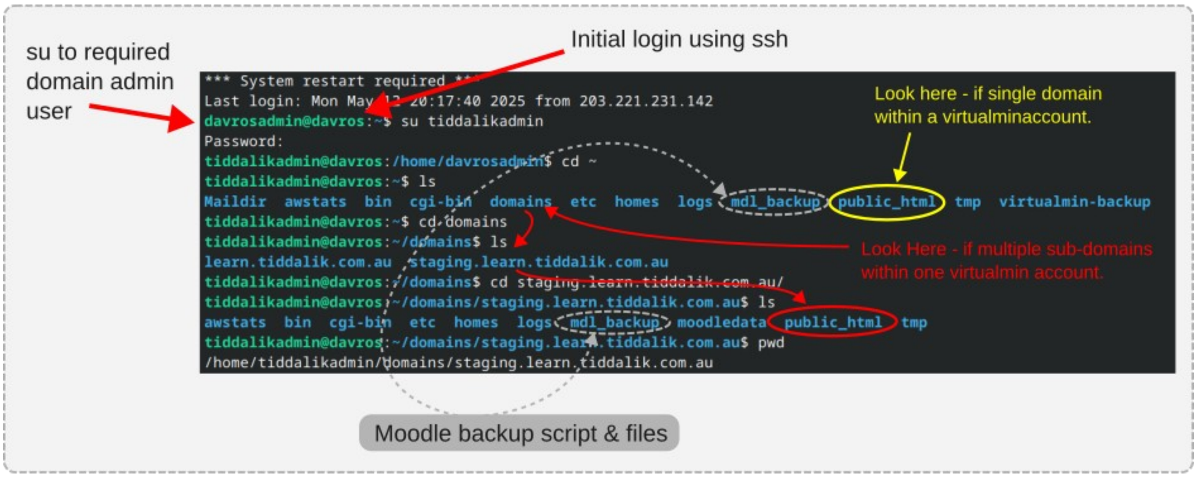
7. Navigate to Moodle Installation

Directory Structure:

- **Standard installations**: public_html/ directory
- **Staging sub-domains**: domains/[subdomain]/public_html/ directory

```
# Example navigation
cd /home/[domain_admin_username]/public_html
# or for staging sites
```

cd /home/[domain_admin_username]/domains/public_html



```
*** System restart required ***
Last login: Mon May 12 20:17:40 2025 from 203.221.231.142
davrosadmin@davros:~$ su tiddalikadmin
Password:
tiddalikadmin@davros:/home/davrosadmin$ cd ~
tiddalikadmin@davros:~$ ls
Maildir  awstats  bin  cgi-bin  domains  etc  homes  logs  mdl_backup  public_html  tmp  virtualmin-backup
tiddalikadmin@davros:~$ cd domains
tiddalikadmin@davros:~/domains$ ls
learn.tiddalik.com.au  staging.learn.tiddalik.com.au
tiddalikadmin@davros:~/domains$ cd staging.learn.tiddalik.com.au/
tiddalikadmin@davros:~/domains/staging.learn.tiddalik.com.au$ ls
awstats  bin  cgi-bin  etc  homes  logs  mdl_backup  moodledata  public_html  tmp
tiddalikadmin@davros:~/domains/staging.learn.tiddalik.com.au$ pwd
/home/tiddalikadmin/domains/staging.learn.tiddalik.com.au
```

su to required domain admin user

Initial login using ssh

Look here - if single domain within a virtualminaccount.

Look Here - if multiple sub-domains within one virtualmin account.

Moodle backup script & files

8. Perform Moodle Update

Git Update Process:

# Check current branch (if required)
git branch

# Check status of the local moodle git repo
git status

Note: this will show some changed files that are untracked, they include some plugin files and cache files – these can be added to the git .ignore file if required – I haven't done this and just leave them.

# Pull latest updates
git pull

# Enable maintenance mode
php admin/cli/maintenance.php --enable

Note: at this stage the moodle web interface will not be accessible to users.

# Run upgrade process
php admin/cli/upgrade.php (normal use)

OR
php admin/cli/upgrade.php –non-interactive (if used in a script: I haven't tested this)

# If upgrade successful, disable maintenance mode
php admin/cli/maintenance.php –disable

Note: test (refresh) the moodle web interface – it should now be accessible to users.

Important Notes:

- Monitor upgrade output for errors
- Do not disable maintenance mode if upgrade fails
- Keep terminal session active during upgrade process

9. Test Web Interface

Post-Update Verification:

- Access Moodle via web browser
- Log in as administrator
- Navigate through key areas:
  - Dashboard
  - Course management
  - User management
  - Plugin settings

10. Update Additional Plugins

Web Interface Method (Recommended):

- Go to **Site Administration → Plugins → Plugins Overview**
- Install available updates via web interface
- This method is preferred unless site must remain offline

CLI Method (if required):

1. **`Download the Updated Plugin:`** Locate the plugin's download link on the Moodle plugins site or from the Moodle plugin directory and use `wget` or a similar tool to download the ZIP file to the Moodle instance.

2. **Extract the Plugin:** Use `unzip` or a similar tool to extract the downloaded ZIP file.

3. **Put the Moodle instance into maintenance mode:**

   # Enable maintenance mode
   php admin/cli/maintenance.php –enable

4. **Move and/or Rename the Plugin Folder:** If the extracted folder name matches the existing plugin folder, you might need to move and rename the existing folder (e.g., `mv x x-bk`) temporarily to avoid conflicts. Also a good practice to keep a backup of the previous version. Ensure the new plugin folder's name matches the expected format (e.g., if the plugin is a module, it should be under the `mod/` directory).

5. **Adjust Permissions (if needed):** If you're not using Git, you may need to adjust the ownership and permissions of the extracted plugin folder so that the web server process can access it.

6. **Run the Upgrade Script:**

   - Navigate to the root of your Moodle installation in the command line
     e.g. /home/tiddalikadmin/domains/learn.tiddalik.com.au/public_html/

   - Execute the following command, replacing */usr/bin/php* with the correct path to your PHP executable.

Note: try *php <command>* first – should work if correct path to the php cli version is already in the server's path record)

```
#code to run the upgrade once unzipped files are correctly located
php admin/cli/upgrade.php

# OR (if php cli is not set up in the server path record)
sudo -u apache /usr/bin/php admin/cli/upgrade.php
```

11. Verify Cron Operations

Check Cron Status:

- **Site Administration → General –** look at notifications, cron notices will be shown if the cron job has not run as expected.
- Confirm cron is running every minute
- Check for any failed tasks – Go to *Site administration>Server>Scheduled tasks*

Command Line Verification:

```
# Check recent cron execution
php admin/cli/cron.php --help
```

12. Post-Update Testing

Team Verification Process:

- Coordinate with Tiddalik team members
- Test range of course functions:
  - Course navigation
  - Content delivery
  - User enrolment
  - Assessment tools
  - Payment gateways (where applicable)
  - Theme functionality

Troubleshooting

Common Issues

- **Permission errors**: Verify file ownership and permissions
- **Database connection**: Check config.php credentials
- **Plugin conflicts**: Disable problematic plugins and test
- **Memory limits**: Monitor PHP memory usage during upgrades

Recovery Procedures

- **Failed upgrade**: Restore from backup using mdl_bu.sh backups
- **Plugin issues**: Access database directly to disable plugins
- **Maintenance mode stuck**: Use CLI to disable: php admin/cli/maintenance.php --disable

Important File Locations

Configuration Files

- **Moodle config**: [moodle_root]/config.php
- **Backup config**: mdl_bu_script/mdl_bu.conf

Backup Script Structure

```
public_html/
├── mdl_backup/
│   ├── mdl_bu_files/
│   └── mdl_bu_script/
│       ├── mdl_bu.conf
│       ├── mdl_bu.sh
│       ├── mdl_bu.conf.template
│       └── readme.md
│
```

Credentials Storage

- All system credentials stored in **Bitwarden application**
- Domain admin usernames and passwords
- Database connection details – found in each Moodle instances *config.php* within the public_html directory.

Regular Maintenance Schedule

Note: Ensure that you are receiving Moodle update emails – requires registering the site. See – *Site administration>General>Registration*.

Daily

- Automated backups via cron job
- Monitor system performance

Weekly

- Check plugin updates
- Review backup logs
- Monitor user activity

Monthly

- Performance assessment
- Security updates
- Staging site synchronisation

Before Major Updates

- Full system backup verification
- Staging environment testing
- Team coordination for testing schedule