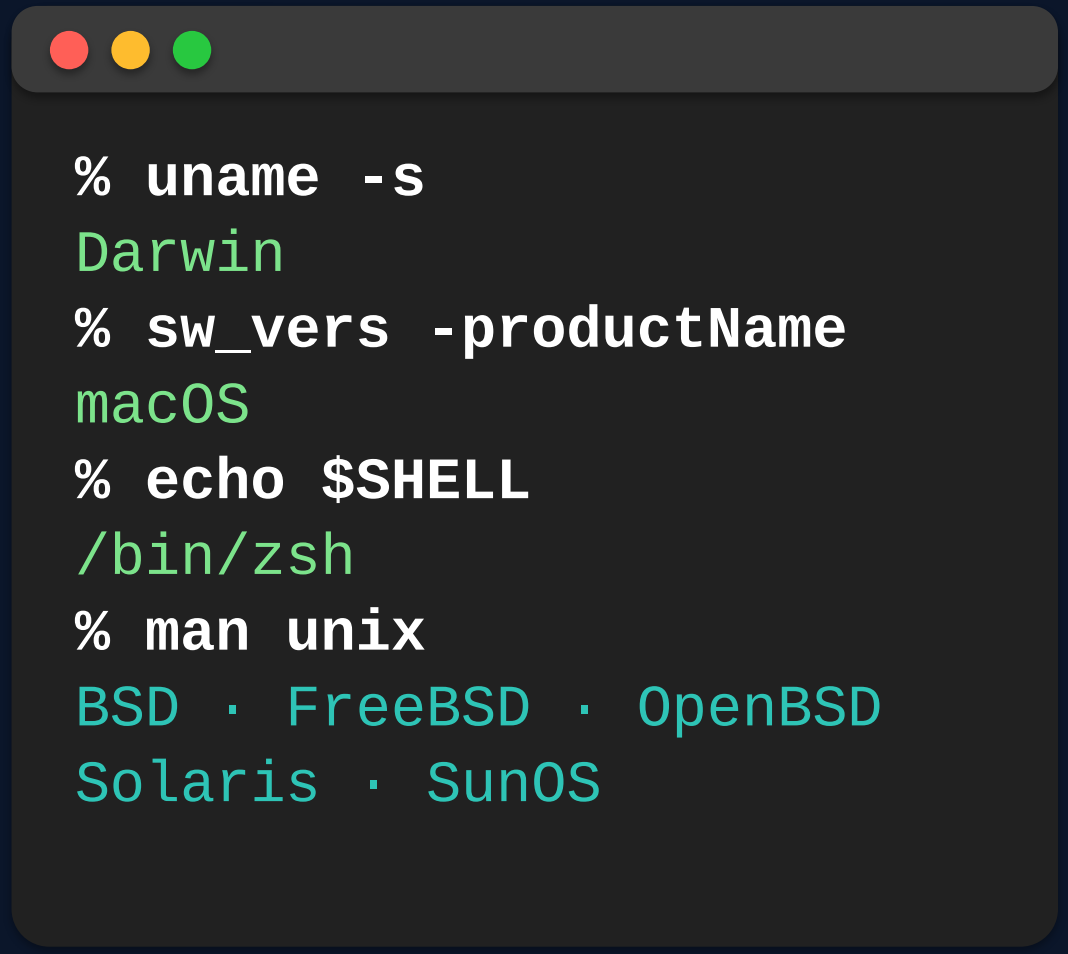


LESSON 12 OF 12

macOS UNIX

Darwin & BSD

Administering macOS: Desktop vs Server Roles

Full Course · macOS 27 Golden Gate

What You Will Learn in Lesson 12

- Desktop vs server: what changes for an administrator
- From macOS Server to services built into macOS
- Always-on settings: power, restart, FileVault authrestart
- File Sharing, Screen Sharing, Content Caching, Time Machine server
- Hosting a web service with launchd
- Managing many Macs: MDM and configuration profiles
- Monitoring, troubleshooting, and choosing BSD or Solaris for a server

Desktop vs Server Administration

	Desktop Mac	Server Mac
Users	One or a few people logged in	Often nobody logged in; services run as daemons
Uptime	Sleeps, restarts for updates any time	Never sleeps; planned maintenance windows
Services	LaunchAgents, apps	LaunchDaemons: file, web, database, caching
Access	Keyboard and screen	SSH, Screen Sharing, MDM
Network	DHCP, Wi-Fi is fine	Wired Ethernet, static IP or DHCP reservation
Priorities	Privacy, usability	Availability, monitoring, backups, change control

macOS Server Is Retired: What Replaced It

Old macOS Server role	Today on macOS
File server	General → Sharing → File Sharing (SMB)
Time Machine server	File Sharing → share options → 'Share as a Time Machine backup destination'
Caching server	General → Sharing → Content Caching
Remote management	Screen Sharing / Remote Management (Apple Remote Desktop)
Profile Manager (MDM)	Third-party or open source MDM + Apple Business / School Manager
Web, mail, DNS, wiki	Homebrew services (nginx, Apache, Postfix…) or Linux/BSD servers

Always-On Settings

```
% pmset -g                                  # current power settings
% sudo pmset -a sleep 0 disksleep 0 displaysleep 10   # never sleep
% sudo pmset -a womp 1                      # wake for network access
% sudo pmset -a autorestart 1               # power back on after an outage
% sudo systemsetup -setrestartfreeze on     # restart if the system freezes

% sudo fdesetup authrestart                 # reboot once without the FileVault prompt
% sudo shutdown -r +5 "Maintenance reboot"  # warn users, restart in 5 min
```

File Sharing (SMB)

```
# GUI: General → Sharing → File Sharing → ⓘ → add folders, set users & rights
% sudo mkdir -p /Users/Shared/LabShare
% sudo chown :labstudents /Users/Shared/LabShare && sudo chmod 2770 /Users/Shared/LabShare
% sudo sharing -a /Users/Shared/LabShare -S LabShare   # add an SMB share point
% sharing -l                                          # list share points

# From another Mac:   Finder → Go → Connect to Server →  smb://mwu-lab.local
% open smb://mwu-lab.local/LabShare
% smbutil view //student1@mwu-lab.local               # shares on a server
```

Screen Sharing, Content Caching, Time Machine Server

```
# Screen Sharing (VNC-compatible):  General → Sharing → Screen Sharing
% open vnc://mwu-lab.local                          # from another Mac

# Content Caching: saves internet bandwidth for every Apple device on the LAN
% AssetCacheManagerUtil status
% sudo AssetCacheManagerUtil activate

# Time Machine destination for other Macs:
#   File Sharing → share folder → ⓘ → 'Share as a Time Machine backup destination'
% tmutil destinationinfo                            # on the client Mac
```

Hosting a Web Service With launchd

```
% brew install nginx
% sudo brew services start nginx      # system daemon: starts at boot
% curl -I http://localhost:8080        # Homebrew nginx listens on 8080
% echo '<h1>MWU Lab Server</h1>' > /opt/homebrew/var/www/index.html

% sudo launchctl list | grep nginx
% ls /Library/LaunchDaemons | grep nginx   # the daemon plist brew created
% sudo /usr/libexec/ApplicationFirewall/socketfilterfw --add /opt/homebrew/bin/nginx
```

Managing Many Macs: MDM

- Apple Business Manager / Apple School Manager: own and assign devices
- Automated Device Enrollment: new Macs enroll at first boot
- MDM server pushes configuration profiles, apps and updates
- Profiles: Wi-Fi, VPN, FileVault, passcode, restrictions (.mobileconfig)
- Products: Jamf, Kandji, Mosyle, Intune; open source: NanoMDM, MicroMDM
- Check a Mac: profiles status -type enrollment ; sudo profiles show

Monitoring and Maintenance Routine

- Daily: backups succeeded? disk space? (df -h, tmutil latestbackup)
- Daily: failed services and errors (launchctl list, log show --last 1d)
- Weekly: softwareupdate --list; brew outdated; review admin group
- Weekly: listening ports (lsof -iTCP -sTCP:LISTEN) — anything new?
- Monthly: test a restore; review users, sudoers.d, firewall rules
- Automate checks with launchd scripts that email or notify you

Troubleshooting Checklist

Symptom	First commands	Lesson
Won't boot / update failed	Recovery → Disk Utility First Aid, reinstall; tmutil	2, 11
App crashes or hangs	Console crash reports, sample, spindump, Safe Mode	3, 9
Mac is slow	top -o cpu, Memory Pressure, fs_usage, df -h	3, 9, 11
Can't log in / no admin	dscl . -read /Groups/admin, Secure Token, Recovery	8
Service not running	launchctl print, log show --predicate, plutil -lint	9
Network problem	ifconfig → ping router → dig → nc -vz → curl	10
Disk full / can't eject	du -sh | sort -h, tmutil listlocalsnapshots, lsof	11

Choosing a Server OS

Choose	When you need	Strengths
macOS	Xcode build machines, small offices, caching, Apple-centric labs	Easy to manage, MDM, great hardware
FreeBSD	Storage, network appliances, high-traffic web/CDN	ZFS, jails, performance, permissive license
OpenBSD	Firewalls, routers, VPN gateways, security-critical hosts	pf, secure defaults, simplicity, code audits
Solaris 11.4	Existing Oracle/SPARC enterprise systems	ZFS, zones, SMF, DTrace, long support
Linux	Most cloud and container workloads	Huge ecosystem (see our RHEL, Rocky, Ubuntu courses)

Capstone Lab: Build a Lab Server

```
% sudo scutil --set LocalHostName mwu-server              # Lesson 10
% sudo pmset -a sleep 0 autorestart 1 womp 1              # Lesson 12
% sudo dseditgroup -o create labstudents; sudo sysadminctl -addUser student1 -password -   # 8
% sudo mkdir -p /Users/Shared/LabShare && sudo sharing -a /Users/Shared/LabShare -S LabShare
% sudo systemsetup -setremotelogin on                     # SSH, keys only (10)
% brew install nginx postgresql@17 && sudo brew services start nginx   # 5, 7, 9
% sudo /usr/libexec/ApplicationFirewall/socketfilterfw --setglobalstate on   # 10
% tmutil localsnapshot; pg_dump ... ; launchd backup job   # 7, 9, 11
% lsof -iTCP -sTCP:LISTEN -n -P        # document every open port
```

Write a one-page 'server document': name, IP, services, ports, users, backups.

Course Review: 12 Lessons

#	You can now	#	You can now
1	Explain UNIX, BSD, Solaris, Darwin, XNU	7	Run SQLite, PostgreSQL, MySQL; back up
2	Install, upgrade, recover macOS	8	Manage users, groups, sudo, Secure Tokens
3	Master the GUI, apps and utilities	9	Control processes; write launchd services
4	Work fluently in zsh and the file system	10	Configure networks, pf and SSH
5	Install software with brew, ports, pkg	11	Manage APFS, security layers, backups
6	Compile, script and debug	12	Run a Mac as a server; troubleshoot

Your Next Steps

- Rebuild the capstone server from memory, then script it
- Install FreeBSD or OpenBSD in a VM and repeat Lessons 4, 9 and 10
- Continue with our free RHEL, Rocky and Ubuntu Linux courses
- Read: The Direct Path to Linux Ubuntu (free chapters online)
- Certifications to consider: Apple Certified Support Professional, Linux+
- Teach someone else what you learned — that is how knowledge grows

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

Slides for this lesson: MyWebUniversity.com/presentations

YouTube: @MyWebUniversity · @MyWebUniversity-Dari

Wahidullah Lutfy — Founder and Director of IT of MyWebUniversity, and CS teacher

Book: The Direct Path to Linux Ubuntu (Amazon · Barnes & Noble)

Learn it, then pass it on.