100 Daily DevOps Shell Commands – Practical Scripts with Objectives

This document contains 100 commonly used shell scripting commands rewritten in a practical DevOps-friendly format. Each section includes a small script snippet and a clear objective explaining its real-world usage.

1. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

2. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

3. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

4. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

5. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

6. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

7. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

8. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

9. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

10. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

11. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

12. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

13. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

14. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

15. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

16. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

17. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

18. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

19. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

20. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

21. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

22. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')
```

```
if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

23. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

24. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

25. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

26. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

27. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

28. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

29. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

30. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

31. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

32. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

33. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

34. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

35. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

36. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

37. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

38. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

39. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

40. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

41. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

42. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

43. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

44. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

45. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

46. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

47. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

48. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

49. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

50. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

51. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

52. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

53. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

54. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

55. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

56. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

57. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

58. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

59. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

60. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

61. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

62. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

63. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

64. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

65. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

66. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

67. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

68. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

69. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

70. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

71. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

72. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')
```

```
if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

73. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

74. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

75. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

76. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

77. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

78. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

79. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

80. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

81. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

82. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

83. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

84. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

85. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

86. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

87. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

88. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

89. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

90. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.

91. Log Management Automation

```
#!/bin/bash
LOG_DIR="/var/log/myapp"
ARCHIVE_DIR="/var/log/myapp/archive"

mkdir -p $ARCHIVE_DIR
find $LOG_DIR/*.log -mtime +7 -exec mv {} $ARCHIVE_DIR \;
gzip $ARCHIVE_DIR/*.log
```

Objective:
Automate log archival and compression to support log retention and storage optimization.

92. Disk Usage Monitoring

```
#!/bin/bash
THRESHOLD=80
USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

if [ $USAGE -gt $THRESHOLD ]; then
  echo "Disk usage exceeded threshold"
fi
```

Objective:
Monitor disk usage and alert when it exceeds a defined threshold.

93. Memory Usage Check

```
#!/bin/bash
free -h
```

Objective:
Quickly check system memory usage for troubleshooting performance issues.

94. Service Health Check

```
#!/bin/bash
systemctl status nginx
```

Objective:
Verify whether a critical service is running correctly.

95. Restart Failed Service

```
#!/bin/bash
systemctl restart nginx
```

Objective:
Automatically restart a service after failure.

96. Process Monitoring

```
#!/bin/bash
ps aux | grep nginx
```

Objective:
Identify running processes related to a specific application.

97. Kill Hung Process

```
#!/bin/bash
PID=$1
kill -9 $PID
```

Objective:
Forcefully terminate an unresponsive process.

98. Directory Cleanup

```
#!/bin/bash
find /tmp -type f -mtime +3 -delete
```

Objective:
Remove temporary files older than 3 days to free up disk space.

99. Backup Important Files

```
#!/bin/bash
tar -czf backup_$(date +%F).tar.gz /etc
```

Objective:
Create compressed backups of critical configuration files.

100. Download Artifact

```
#!/bin/bash
wget https://example.com/app.tar.gz
```

Objective:
Download build artifacts or packages from remote repositories.