**ProLUG 101**

**Unit 8 Worksheet**

# Instructions

Fill out this sheet as you progress through the lab and discussions. Hold onto all of your work to send to me at the end of the course.

# Discussion Questions:

**Unit 8 Discussion Post 1**: It’s a 2 week holiday in your country and most of the engineers and architects who designed the system are out of town. You’ve noticed a pattern of logs filling up on a set of web servers from increased traffic. Your research shows, and then you verify, that the logs are being sent off real time to Splunk. Your team has just been deleting the logs every few days, but one of the 3rd shift engineers didn’t read the notes and your team suffered downtime. How might you implement a simple fix to stop gap the problem before all the engineering resources come back next week?

1. What resources helped you answer this?
2. Why can’t you just make a design fix and add space in /var/log on all these systems?
3. Why can’t you just make a design change and logrotate more often so this doesn’t happen?
4. For 2,3 if you are ok with that, explain your answer. (This isn’t a trick, maybe there is a valid reason.)

**Unit 8 Discussion Post 2**: You are the only Linux Administrator at a small healthcare company. The engineer/admin before you left you a lot of scripts to untangle. This is one of our many tasks as administrators, so you set out to accomplish it. You start to notice that he only ever uses nested if statements in bash. You also notice that every loop is a conditional `while true` and then he breaks the loop after a decision test each loop. You know his stuff works, but you think it could be more easily written for supportability, for you and future admins. You decide to write up some notes by reading some google, AI, and talking to your peers.

1. Compare the use of nested if versus case statement in bash.
2. Compare the use of conditional and counting loops. Under what circumstances would you use one or the other?

# Definitions/Terminology

Variables

Interpreted program

Compiled program

Truth table

And/or logic

Single/Dual/Multiple alternative logic

# Notes During Lecture/Class:

Links:

Terms:

Useful tools:

# Lab and Assignment

Unit 8 Scripting and system checks

Begin working on your project from the Project Guide

 Topics:

* + 1. System Stability
		2. System Performance
		3. System Security
		4. System monitoring
		5. Kubernetes
		6. Programming/Automation

You will research, design, deploy, and document a system that improves your administration of Linux systems in some way.

# Digging Deeper

1. Read <https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_07_01.html>

<https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_07_02.html>

<https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_07_03.html>

What did you learn about capabilities of bash that can help you in your scripting?

1. If you want to dig more into truth tables and logic, this is a good start: <https://en.wikipedia.org/wiki/Truth_table>

# Reflection Questions

1. What questions do you still have about this week?
2. Just knowing a lot about scripting doesn’t help much against actually doing it in a practical sense. What things are you doing currently at work or in a lab that you can apply some of this logic to?