Analysis/HLD Template

[SKU# Version x.x: Course Title]

Analysis and High Level Design for Lessons

## **<<STOP - make a copy of this doc before proceeding!!>>**

# Purpose

### Analysis

The Analysis is meant to house the vital information about a training deliverable and answer the main question; why is this training necessary? Specifically, it answers: What problems does this training solve? Who is/are the main audience(s)? What considerations or risks exist? If this is an update, what are the major changes? What performances are critical for our learners to be successful on the job? And finally, what products/technologies (both Red Hat and otherwise) will be included?

**Important:** Think of the analysis phase as the foundation for everything (HLD) to follow. Understanding the problem, goal, audience, and defining, vetting, and refining groups of tasks into performance objectives, MUST be complete before starting the HLD. Engage Dave and Steve the moment you begin developing the Analysis.

### High Level Design (HLD)

The High Level Design is a roadmap for a course, lesson, or specific learning deliverable. In this case we will focus on lessons. The main goal of the HLD is to come to a basic agreement on content layout, flow, and performances to be covered in the training. It should be treated as an *initial* design, to be reviewed, refined, and revised by the development team during detailed design. Changes must be agreed on by the tech lead and developers.

# How to use this document

[Click here for details](#_ftint3jnigwh)

# ANALYSIS

*Course Design Proposal and Responsibilities Signoff*

All reviewers as well as those needed for signoff of the entire Analysis section, will appear in the tables below.

Course design cannot be started until the first table is signed off by the appropriate players: ID(s), POs, etc.

##### **Document Contributors**

[Prod. Tech. Lead] please list those who approve, review, or are consulted on the Analysis.

| **Title** | **Name** | **Level of Involvement** | **Status** |
| --- | --- | --- | --- |
| Product Tech. Leader |  | Choose | Not started |
| Dev. Team Manager |  | Choose | Not started |
| Instructional Designer | Dave Sacco | Approver | Not started |
| Principal Content Architect | Steve Bonneville | Approver | Not started |
| Technical Editor |  | Choose | Not started |

### -Marketing and GTM-related Information–

## 

## Problem Statement

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

## [*Guidance and Example(s)*](#_i54k5fkbg3yb)*.*

#### 

## Goal Statement

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

## [*Guidance and Example(s)*](#_uazhlxujm67p)*.*

## Target Audience

* Lorem ipsum dolor sit amet, consectetur adipiscing elit…

## [*Guidance and Example(s)*](#_qjxywz7ey73p)*.*

## 

## Prerequisites

* Lorem ipsum dolor sit amet, consectetur adipiscing elit…

[*Guidance and Example(s)*](#_vc0wd3n3ecsf)*.*

### -End of Marketing and GTM-related Information–

## Job Task Analysis (JTA) & Performance Objectives (POs)

1. Objective…

[*Guidance and Example(s)*](#_hw420janbs5)*.*

## 

## Considerations and Risks

* Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet…

###### [Guidance and Example(s)](#_qd7qniw041h4).

## 

## Products/Technologies

* Lorem ipsum dolor sit amet…

[*Guidance and Example(s)*](#_r11r7di8loqe)*.*

# HIGH LEVEL DESIGN for Lessons

*Lesson Design Proposal and Responsibilities Signoff*

All reviewers as well as those needed for signoff of the entire High Level Design section will appear in the table below.

###### [Guidance and Example(s)](#_a2jc41rkfmss).

##### **Document Contributors**

[Prod. Tech. Lead] please list those who approve or review the HLD.

| **Title** | **Name** | **Level of Involvement** | **Status** |
| --- | --- | --- | --- |
| Product Tech. Leader |  | Choose | Not started |
| Dev. Team Manager |  | Choose | Not started |
| Instructional Designer | Dave Sacco | Choose | Not started |
| Principal Content Architect | Steve Bonneville | Choose | Not started |
| Technical Editor |  | Reviewer | Not started |

Note: Exams may not be aligned to lessons so we dropped the mapping table.

# Tips: Using Backward Design

In general terms, backward design refers to the practice of defining, even loosely, the performances (Labs, Quizzes and GE’s) for each section BEFORE writing the lecture content.

###### [Guidance and Example(s)](#_7wh2dqllmfjq).

Course Flow Begins Here

###### 

###### [Click here for detailed guidance and examples for each section below.](#_o5dxxg54iuow)

#### #: Lesson Title

Lesson Goal Statement:

#### Section #: Title

Section Type:

Objective Statement:

Scenario or Use Case:

Notes:

#### Summary (w/new rules for writing them)

#### References

#### Lesson & Section Timing Estimates

| **L1 Title + Keyword** | **Est Minutes** | **Comments** |
| --- | --- | --- |
| S1: keyword |  |  |
| S2: keyword |  |  |
| Lab |  |  |
| **Total** |  |  |
| **L2 Title + Keyword** | **Est Minutes** | **Comments** |
| S1: keyword |  | …etc. |

## **Note**: repeat the above structure for each Lesson and section.

Guidance/Examples

# How to use this document

Product Technical Lead: delete content in blue text, replacing it with your Analysis and HLD-specific content. The blue text is for guidance only.

The intent of this document is to present a unified template to facilitate the Analysis and Design of training deliverables in the form of lessons. Our ultimate goal is to make this process **consistent** and **repeatable**.

**For Major Course Updates PLEASE READ!**

There are a few different ways to approach the Analysis and HLD when making major updates to a course that will be broken into Lessons. Please contact your ID (Dave) or Principal (Technical) Content Architect (Steve) to discuss the current methods and to view some examples.

### 

### Problem Statement Guidance and Example

**Guidance**

Engage your Dev. Team Manager and Pr. Content Architect for assistance, if needed. Use the following questions as a guide:

* What is the customer need(s) we are trying to solve?
* What gaps do our customers have that are preventing them from solving their problems?
* What gaps do our services people have that are preventing them from effectively delivering solutions?
* What are features of the product that people are missing and should be showcased as part of a use-case based training solution?

Include pointers, links, etc. to supporting documentation.

**Example**

Companies that haven’t considered or are not far enough in their automation journey, often need to deliver a standardized, centralized automation that reduces routine, repeatable tasks, and complexity. This course is one way to move forward in that journey.

### Goal Statement Guidance and Example

**Guidance**

*For Course Goal Statements (Lessons Combined to create a course), follow these instructions:*  
A course goal is the highest level of “objective” minus the performances. It states the **benefit to the business and the learner in broad terms** without detailing how it will be achieved. It’s a taste of what the course provides. There is an art to writing a course goal and your friendly IDs can help!

* Keep this to 1-2 sentences.
* Link to other docs/resources as needed.

**Example**

Learn how to deploy and operate Ansible automation services for their organization that can extend to a larger scale.

*For Lessons that may be stand-alone or grouped but NOT into a course, follow these instructions:*

* If creating one (1) Lesson, this statement IS the PO created in the JTA.
* If this is a bundle of 2 or more but not necessarily a course, create a bundle goal statement that encompasses the content/intent of the Lesson bundle.

### Target Audience

Guidance and Example

**Guidance**

Target Audience should consist of the main audience/s for this specific course based on the skills and knowledge required to perform their role/s effectively.

* **When possible**, align to Red Hat defined personas. Refer to the deck (linked below) and specifically, Slide 10, as a baseline:  
  [Corporate Persona Deck](https://docs.google.com/presentation/d/1l5fwLo2LE6ZySA6ubqb1iWBgt3NHCFVdACeuLfaY_nk/edit#slide=id.p1)

**Example**

* **Network architects**, **network admin/ops**, and **automation architects** who are responsible for designing, managing, and automating an organization's network infrastructure.

### Prerequisites

Guidance and Example

**Guidance for prerequisite skills/knowledge:** Include any prerequisite skills and/or knowledge that the listed audience(s) must possess to be successful. Add other RHT course/s, if applicable.

* 1-3 bullets for each
* Link to additional courses/docs/resources as needed and **be specific**; listing lessons and sections within those resources, if known.

**Example:**

* Experience with network hardware such as Cisco systems that run IOS is helpful but not required.
* Previous experience with Ansible automation of Linux is helpful.
* Significant Linux experience is not needed for this course.
* Suggested prerequisite course: RH294 - Red Hat Enterprise Linux Automation with Ansible

### Job Task Analysis (JTA) and Performance Objectives

**Guidance:**

A Job Task Analysis (JTA) is a detailed list of the tasks your target audience must be able to perform to satisfy the course/lesson goal.

Performance Objectives (POs) are the broad “goal” satisfied by the tasks from the JTA. Think generally of Performance Objectives as a lesson goal and the individual tasks from the JTA as section objectives for that lesson goal.

Below are the steps to take to go from a JTA to a PO.

1. Start by listing all the tasks you believe are important for learners to know; don’t worry about grouping, organizing, or verifying (yet). Do this outside of this document (possibly Certiverse or a Sheet, TBD).
2. Enlist SMEs, Developers (if available), and others to provide feedback that the tasks are valid and belong; or if tasks are missing. Begin refining the list based on their feedback (re-writing, combining, grouping, deleting, etc.)
3. If groupings exist, put those tasks together and create a group that is essentially a draft performance objective that encompasses the group of tasks.
   1. **Example**
      1. Refined/verified Tasks:
         1. Configure an LDAP identity provider and automate group synchronization between OpenShift OAuth and an LDAP server.
         2. Configure an OIDC identity provider and automate group synchronization between OpenShift OAuth and an OIDC server.
         3. Generate a token and a client certificate and add them to a kubeconfig file.
      2. (Above satisfy this broader) Performance Objective:

Configure OpenShift clusters to authenticate by using LDAP and OIDC enterprise identity systems and to recognize groups that those systems define.

Some rules for writing POs:

**When Designing one (1) Lesson**

* JTA = tasks, grouped to satisfy the lesson with one Performance Objective that becomes the Lesson Goal.
  + All sub-tasks become sections for the lesson.
* The Lesson Goal in the Analysis section **IS** the Lesson Goal Statement in the HLD (verbatim). The Lesson goal statement should be the same as the PO (with no re-wording/updating) and consist of 1-3 sentences.

**When Designing a New Course Made up of Lessons**

* JTA = list all critical tasks, group them, then write a PO for each grouping.
* Each PO should align (exactly) to one lesson. All sub-tasks become sections for ea. lesson.
* These POs are the highest level performances that must be satisfied for the learner to perform their job.

**When Updating an Existing Course (major update)**

* JTA = analysis of existing chapters where one chapter *may* = one Lesson (this is general guidance and not to be taken as a rule). Consult Steve and Dave for assistance in this analysis, if needed.
* PO=Chapter goal statements (if used) that become Lesson goal statements. Follow guidance in the last section above if creating a brand new lesson to be inserted into the existing course).
* Each PO should align (exactly) to one lesson. All sub-tasks become sections for ea. Lesson.
  + In the HLD, we strongly recommend creating links to individual Analysis/HLD docs for each Lesson, similar to this **example**:

### Chapter 2: Configuring Link Aggregation

* **Lesson Analysis/HLD:** [AU0005L (RH358.2): Link Aggregation Lesson Analysis/HLD](https://docs.google.com/document/d/1_6l8LTfUgScnBJGsIAHQaB09QKhh4VWG-1Ix0Hcswnk/edit?usp=sharing)
* **Lesson Goal:** To configure and troubleshoot advanced network interface functionality, including bonding.
* **Lesson Status:** Existing content, rework to use bonding instead.
* **Complexity/Time to Develop:** High
* **Priority to Develop:** Low/Moderate (due to significant effort)

**Example POs:**

[PO 1] Install OpenShift on three servers, which are both an HA control plane and also run user workloads, without requiring access to the Internet nor server-management hardware.

[PO 2] Develop Ansible playbooks to automate routine tasks and system configurations, ensuring efficient and consistent operations.

[PO 3] Perform basic data manipulation (clean, filter, sort, split, merge/aggregate data) on imported data using Pandas.

[PO 4] Demonstrate proficiency in integrating Ansible automation services with other tools and technologies to support diverse organizational needs.

### 

### Considerations and Risks

Guidance and Example

**Guidance:**

Below are some typical considerations and possible risks:

* Will this course be delivered in a non-standard way?
* Major lab environment/classroom dependencies, changes, requirements.
* Specific delivery requirements (set-up, take-aways, etc.) per modality.
* Where risks affect part of the outline or specific content chunk, add comments directly into the outline so developers can easily find them.
* Can this **not** be run in the disconnected China data center?
* Link to other docs/resources as needed.

**Example:**

* ​​This course covers Red Hat Insights, which requires Internet access and Red Hat Insights accounts with appropriate access.
* Ensure alignment with the messaging and offerings for CentOS to RHEL Migration and modernization that are used by Services and other functions at Red Hat.
* Utilize Jupyter Notebook templates for examples and possible takeaway/job aids, as needed.
* ID’s have proposed a different structure than what we are used to (Lecture-Small GE-Small GE-Small GE instead of Lecture-GE-Lecture-GE etc). That is, instead of Lecture-GE pairs, use Lecture followed by 2 or 3 smaller chunked GEs).

### Products and Technologies

Guidance and Example

**Guidance:**

List the products and associate versions that will be used/touched in the training. Be sure to include 3rd party products/tools. If any strong risks or considerations occur as a result of their use, list them here as well as in the previous section.

* Primarily specify Red Hat products and versions
* Also partner products and technologies
* Specific lab environment technology may go here too.

**Example:**

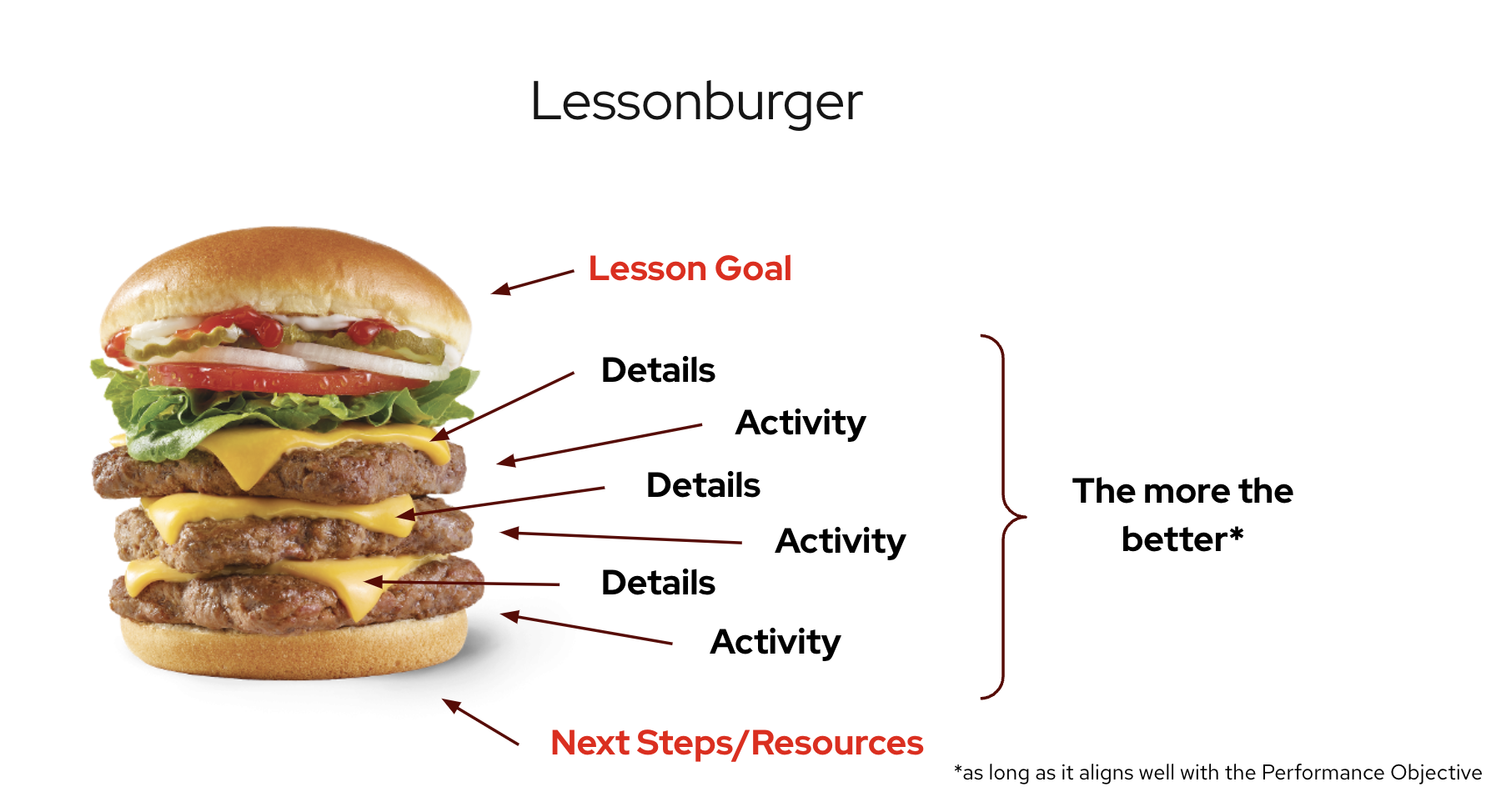
* Red Hat OpenShift Container Platform (4.12)

### 

### What is a Lesson?

A lesson is a structured, focused unit of a training program designed to teach a specific skill or concept. It is often, but not necessarily, part of a larger course or training series.

Think of a lesson as a self-contained entity, with the following elements:



### Using Backward Design

**Guidance:**

* Backward design may be thought of as purposeful task analysis: Given a task to be accomplished, how do we get there? Or one might call it planned coaching: What kinds of lessons and practices are needed to master key performances?
* Rather than creating assessments near the conclusion of a unit of study [Lesson] …, [backward design] reminds us to begin with the question, ***what would we accept as evidence that learners have attained the desired understandings and proficiencies?***… Many who have adopted this design approach report that the process of "thinking like an assessor" about evidence of learning not only helps them to clarify their goals but also results in a more sharply defined teaching and learning target, so that learners perform better knowing their goal. Greater coherence among desired results, key performances, and teaching and learning experiences leads to better student performance—the purpose of design.

[Adapted from Understanding by Design by Grant Wiggins and Jay McTighe]

**Backward Design for Labs**

List the objectives for each lab and align them with the performance objective for that lesson.

1. Determine which performances warrant a lab activity to validate students can perform what is needed to meet that goal.
2. List the skills and knowledge needed, including GEs, quizzes, and lecture to support the lab activity

**Additional Knowledge Checks (Quizzes and Guided Exercises [GEs])**

The purpose of knowledge (self) checks is to give the learner a chance to recall terms, processes, concepts (quizzes), and practice tasks with complete steps (GEs). Content in the course outline is typically (but not always) broken down in the following way:

There are no strict rules about when to add a GE or a quiz, how many per section, or whether they need to be separated by content/context/lecture. Don’t worry about having too many GEs or quizzes; if they add educational value and make the learner experience better, err on the side of too many rather than too few.

The more complex the task, the more need there may be for practice and reinforcement and further attention to breaking up larger passages. Remember the following when considering quizzes and GE’s:

* They must always lead back to/support the section and lesson objectives/goal.
* For quizzes, consult guidance in Confluence [here](https://spaces.redhat.com/display/PTL/Quizzes) and with your ID.
* For GEs, approach them similarly to labs but as practice of the steps and actions needed to meet the performance objective. There should be no surprises in a GE - it is guiding a student.

### Design Resources

Below are a list of workshops and documentation that define, in greater detail, our approach to backward design, objective/goal writing, and scaffolding concepts

* Miro w/links to the Workshops on [Objectives, Backward Design, & Content Categories](https://miro.com/app/board/uXjVOVcSyqQ=/)
* [Course Design Job Aids and checklists](https://drive.google.com/file/d/1DHjtOYS2YBBLxFszK_sM_D1xhkY8WNj8/view?usp=sharing)
* Refer to [Writing Content](https://github.com/RedHatTraining/scaffolding/blob/dd71d9d75baea888464b4e123bd97042569307a5/docs/WRITING.adoc) in the scaffolding git repo for details on how to boot-strap new content within your course git repo; in this case, a quiz. You have two choices:

1. From the command line, navigate to the section where the quiz is to be built and run sk sample [chapter] [topic] multichoice [or appropriate quiz type]to insert it into the course **OR**
2. In your course repo, locate the outline.yaml file and add the following line to the topic where a quiz will occur: - type: multichoice [or the appropriate quiz type].

Course Flow Start here

### High Level Design for Lessons Begins Here

Course Flow Begins Here

#### #: Lesson Title

Lesson Goal Statement:

Pulled from the JTA and is the Performance Objective created from that exercise. This is the PO, word-for-word.

[SAMPLE LESSON](https://docs.google.com/document/d/1YtXL5P6vUN5FEdAwmv8vkzetfxDMzNQL5y_79grpFd8/edit?tab=t.0#heading=h.6abjax2d9fhq) - View a self-contained lesson to see how it’s structured.   
[**Note**] this is a stand-alone lesson for Ansible Lightspeed and is still a WIP so you may see comments, etc. but the structure is valid. Note that the Lesson Goal Statement comes from the JTA in the Analysis section.

How would we test that a learner successfully performed the lesson goal?

#### Section #: Title

Section Type

* Enter the content type for the section here (Lecture, GE, Quiz, Lab, etc.)

Objective Statement

* (appears below ea. section)   
  For each section type, write an objective statement based on the tasks that make up this Lesson’s Goal Statement. This is the objective stating what the learner will do in the section. Be specific and keep these principles in mind:
  + What is the task?
  + Where or in which environment will it happen?
  + What defines learner success, is it % complete, timed, proficiency level (if known)?
  + If this is a new concept or performance, then point to resources for use in the developer’s first sprint.

Scenario or Use Case

* Think of a use case as a real world scenario or story that helps satisfy the section (Lecture, Quiz, GE, or Lab) objective. A use case can range from a simple sentence that leads into a quiz or GE, to a more detailed scenario set up with variables, code snippets, or diagrams.

Notes

* Enter helpful notes to developers.

#### Summary (w/new rules for writing them)

Write summaries as a reminder of what was learned and recommendation for what is next, including references/resources.

### **Guidance on creating the Summary section for Lessons**

We will use the standard “summary” section title for ea. Lesson but rather than a long bulleted list of ea. learning point from the lesson, focus on the 1-2 most impactful points, written in sentences.

Process: Summarize, briefly, the main points of the lesson, followed by possible next steps. Write this section in paragraph style when possible. If a Lesson is a structured, focused unit of training on one performance, not necessarily relying on other lessons around it, then the learner may need a little "call to action" after finishing one.

Some things to keep in mind when writing the next steps portion of the summary:

* Don’t include long bulleted lists restating objectives - while some bullets are acceptable, the entire next steps section should **not** consist of a bulleted list (see the example linked below).
* Avoid pointing or linking **directly** to another lesson or course as links break and become outdated and create a maintenance issue.

Consult with your ID and/or Principle Content Architect for guidance, as needed.

**Example**

In this course, you learned \_\_\_\_\_. As a next step in your learning journey, you might learn more about \_\_\_\_, or take the course, <course title/link>.

#### References

References refer to any resource that lives outside the current lesson. This list of references is a separate section, and is not included in the summary. Currently, references are listed at the end of each lecture.

#### Lesson & Section Timing Estimates

Keep in mind that lesson, section, and exercise timings are rough estimates, only a starting point that will be revised over time. For update courses, engage the Instructor Advocate to provide more realistic estimates when possible.

**Note**: an (L) lesson and (S) section keyword is usually a one-word descriptor that maps to the git repo directory where ea. asciidoc file is stored. Please refer to [this page (to be updated)](https://spaces.redhat.com/display/PTL/Keywords) in Confluence for specific keyword rules.

| **L1 Title + Keyword** | **Est Minutes** | **Comments** |
| --- | --- | --- |
| S1: keyword |  |  |
| S2: keyword |  |  |
| Lab |  |  |
| **Total** |  |  |
| **L2 Title + Keyword** | **Est Minutes** | **Comments** |
| S1: keyword |  | …etc. |

## 

### Summary