# Design and Content Guidelines for Hands-on Labs

## Lab manual template

Always copy the current markdown or gdoc template. Don't use an old copy of the template or re-purpose an existing lab manual because the current template might have changed.

## General self-paced lab guidelines

- Give learners a story to follow but keep it simple.
    - Combine learning objectives to define scenarios.
    - Incorporate decision points (e.g. alternatives) and pain points with solutions.
        - Include the *why*, in addition to the *how* and *what*.
        - Useful story arc:
            - Start with the problem.
            - Follow immediately by stating the solution.
            - Explain any concepts needed to implement the solution.
            - Provide tasks that implement the solution.
            - End with a clear accomplishment (highlight it!).
            - Review what they accomplished and how the skills are applicable to other scenarios.
    - Remember that simpler is always better!
- Design labs to help learners become problem-solvers and develop agency.
    - Clearly identify the problem to be solved and the skills needed to solve it.
    - Highlight which skills they already have that can be applied to the problem (e.g. learned from a previous task, or lab in quest).
- Learners should reach an *aha!* moment.
    - Design the project and tasks to help learners gain insight into the process and develop transferable skills that apply to a range of problems and scenarios.

# Construction of a lab

Below is how a lab is constructed.

# Lab title

The lab title should succinctly describe the objective of the lab.

- Begin with either a gerund (*Creating, Configuring,* etc.) or imperative (*Create*, *Configure*, etc.).
- Use title case.
- Include one or more service names that are a key aspect of the lab or that distinguish it from other labs.
- Include a programming language name if it's a key point of the lab.

| **Say this...** | **Not this...** |
|---|---|
| Working with Virtual Machines | How to Work with Virtual Machines |
| Setting up a PostgreSQL Data Disk | How to Set Up a PostgreSQL Data Disk on Google Cloud Platform |

# Overview

Provide a description (1–2 paragraphs) of the lab. The overview can include:

- The lab scenario
- A diagram of the architecture that the lab will create
    - Be aware of the issues with images, which we'll outline in a bit.
- Other conceptual information about the technologies used in the lab
    - Keep the conceptual information brief and relevant to the goals of this lab. You can link to concept topics in documentation for in-depth information.

| **Say this...** | **Not this...** |
|---|---|
| In this lab, you use Cloud Datastore and App Engine to develop a backend service with real-time synchronization and email notifications. | This tutorial will show you how to write a backend service. |

# What you'll learn

List 3–4 high-level tasks that the student will complete in the lab.

- Include an introductory phrase, such as "In this lab, you learn how to perform the following tasks," and end with a colon.
- Begin each objective with an imperative verb (*Create, Configure,* etc.)
- Use simple present tense.
- Do use ending punctuation.
- Don't include low-level actions.
- Don't make assurances about what the student will be able to do after finishing the lab.

| **Say this...** | **Not this...** |
|---|---|
| In this lab, you learn how to perform the following tasks:<br>● Create a network.<br>● Launch an instance in the network. | At the end of this lab you will be able to:<br>● Create subnets<br>● Create a network |

# Tasks and subtasks

Organize your lab manual into tasks and subtasks so that the learner can visualize and understand the overall process instead of being overwhelmed with a single block of steps.

## Tasks

Tasks represent the high-level organization of the lab procedure. Ensure that:

- All task steps are numbered.
- Each action has a task number.
- Tasks are single sentences.

  1. In the Cloud Console, on the **Navigation menu** (≡), click **Compute Engine** > **VM instances**.

  2. Click **Create**.

  3. Specify the following and leave the remaining settings as their defaults:

- If it's a one-step procedure, use a bullet instead of a "1."

### Task headings

The task heading should succinctly describe the task.

> **Example**
> Task 1. Set up your development environment

#### Task heading formatting

- Include the word “Task” and the task number followed by a period.
- Begin with an imperative (*Create, Configure,* etc.).
- Use sentence case.
- Do not use ending punctuation.

### Task content

For each task, write a short summary of what the learner will accomplish.

> **Example**
> In this task, you set up the software and source code needed to develop the app.

Include conceptual information if it helps learners understand the task and perform it more successfully. Do *not* include steps in the task content.

#### Task content formatting

- Use simple present tense.

## Subtasks

Create one level of subtasks (procedures) under each task to further organize the lab manual.

### Subtask headings

The subtask heading should succinctly describe the task.

> **Example**
> Clone the repository

#### Subtask heading formatting

- Begin with an imperative (*Create, Configure,* etc.).
- Use sentence case.
- Do not use ending punctuation.

### Subtask content

Subtasks contain the specific steps that the learner performs. It may take some effort to create the appropriate number of subtasks; the steps should be divided into logical groups, but no individual subtask should be unusually long.

See the Procedures (steps) section for detailed information about writing steps.

## Including notes

Use notes to call attention to information of special interest or importance that is outside of the flow of the lab. Notes can be in both conceptual sections and procedures.
Format the heading, followed by a colon, in bold, and start the note text after one space.

When in github, notes should be created with this format in markdown:
`<ql-infobox><strong>Note:</strong> the text of the note</ql-infobox>`

> **Note:** This is an example of note text in an infobox.

If you have multiple notes, use a line break after the heading and colon, and format the notes like a bulleted list.

> **Notes:**
> - First note.
> - Second note.
> - Third note.

# Images (screenshots, diagrams, etc.)

Use images only sparingly in your lab manual. Images increase the apparent length of the lab and block the flow of comprehension as learners try to follow instructions. In addition:

- Images cannot be easily localized - text in screenshots and diagrams would need to be localized, and that service isn't offered.
- Images present a barrier to learners who use screen reader technology.
- Never use an image for code output unless there is UI in the shot.
- Never use an image as a way to illustrate the steps used. Steps should be listed in the lab content.

## Alt text

- If you do use an image, *you must include alt text* that explains it. A screen reader will read the literal text you include in the alt text. Alt text:
    - concisely and clearly describes the function of the image.
    - is less than 125 characters (max for most screen readers).
    - does not begin with “photo of..” or “picture of..”.

**Example**

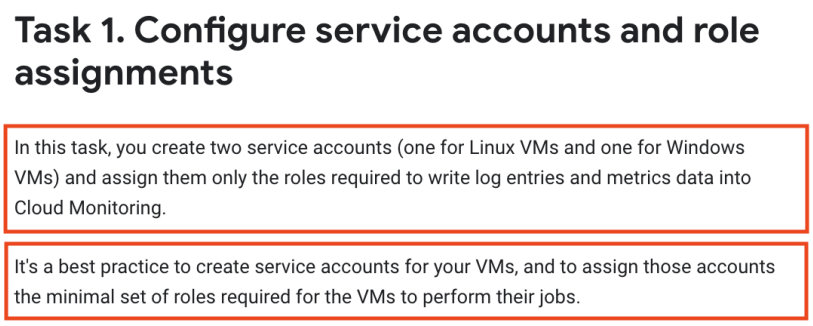
```
![List of images in Container Registry](img/Kubernetes_Container_reg.png)

Now you have a project-wide Docker image available which Kubernetes can access and orchestrate.
```

- You can also add mouse-over text - in markdown, it looks like this: [google cloud console](img/gcc.png "Google Cloud console").

## Paragraphs

- Use line breaks to add paragraphs and break up long pieces of text—this will make the text easier to read and digest:




- Compare this with the screenshot below, where the paragraph is one dense block. This paragraph is more difficult  to read:

## Overview

In this lab, you use the Web Security Scanner to scan an App Engine application for vulnerabilities. Web Security Scanner identifies security vulnerabilities in your App Engine, Google Kubernetes Engine (GKE), and Compute Engine web applications. It crawls your application, following all links within the scope of your starting URLs, and attempts to exercise as many user inputs and event handlers as possible. It can automatically scan and detect four common vulnerabilities, including cross-site-scripting (XSS), Flash injection, mixed content (HTTP in HTTPS), and outdated/insecure libraries. It enables early identification and delivers very low false positive rates. You can easily set up, run, schedule, and manage security scans.

# Procedures (steps)

## General guidelines

- Present each step in chronological order.
- Number each step.
- Use simple present tense.
- Include an imperative verb (*click, copy, press*, etc.) in each step.
- Include some user action (interaction with the user interface) in each step:
    - Mental actions, such as *deciding*, are not user actions.
    - Computer actions, such as results, are not user actions.
- For UI elements, mention the UI label before the action.

| **Say this...** | **Not this...** |
|---|---|
| For **Password**, type your password. | Type your password in the **Password** box. |
| For **Zone**, select the zone your lab is assigned to. | Select the zone your lab is assigned to in the **Zone** list. |

# Sentence structure

There are three basic parts to a step: purpose, location, and user action. An individual step must include an action, but the purpose and location are optional.

| Part | Example |
|---|---|
| **Purpose** | *To clone the repository,* execute the following command: |
| **Location** | ● In the navigation pane, click **Transactions**.<br>● For **Location**, click **Zonal**. |
| **User action** | Click **Next**. |

Many steps include all three sections.

## Combining actions

- In general, use a separate step for each user action.
- When a UI surface (screen or dialog box) contains multiple user actions, combine them if they are short and logically related.
- Do not combine actions that occur on different surfaces.

### Combining 2 actions

To combine 2 actions, separate them with a comma and the phrase “and then.”

| **Say this...** | **Not this...** |
|---|---|
| Type **test-instance-group-1**, and then click **Create**. | Click **Create**, and then, on the next page, select the instance group. |

### Combining 3 or more actions

- To combine 3 or more actions, create a table with the control labels in the first column and the values or options in the second column. Use this format:

| **Property** | **Value**<br>(type or select) |
|---|---|
| **[UI control label]** | **value** |

- Introduce the table with the phrase “Specify the following” followed by a colon.

    - Optional: Include the location, such as the name of the section or the dialog box.
    - If the UI surface includes controls that should not be changed, include the phrase “leave the remaining settings as their defaults.”
- Document a commit button, such as **Create**, as a separate step outside the table.

**Example**

1. On the **Create an instance** page, specify the following, and leave the remaining settings as their defaults:

| **Property** | **Value**<br>(type or select) |
|---|---|
| **Name** | **test-instance** |
| **Machine type** | **micro** |
| **Container** | **Deploy a container image to this VM instance** |
| **Access scopes** | **Allow default access** |

2. Click **Create**.

## Referring to UI elements (panes, menus, text boxes, etc.)

See Parts of the Google Cloud console.

### Text formatting

- In an action step, use bold formatting for the *exact* name of a UI element, including options on a menu or dropdown list.
- Do not use bold for any following punctuation, such as a comma or period.
- Do not use bold formatting when you refer to a UI element in regular text (not an action step).
- Do not use bold formatting when you refer to a UI element indirectly (not by exact name).

## Capitalization, icons, and ellipses

- For most UI labels, match the capitalization of the label. However, for labels in all caps, use sentence case.
- When referring to controls that include icons or ellipses in their labels, use only the text portion of the label.

| Button image | Reference |
|---|---|
|   | Click **Refresh**. |
|   | Click **Enable APIs and services**. |

Use these small icons inline with text. Refer to these links when in the repo.

Inline icon links:

| Name of icon | Link to icon |
|---|---|
| Add icon | https://storage.googleapis.com/cloud-training/images/add.png |
| Cloud Shell Editor icon | https://storage.googleapis.com/cloud-training/images/cloud-shell-editor.png |
| Activate Cloud Shell icon | https://storage.googleapis.com/cloud-training/images/devshell.png |
| Edit icon | https://storage.googleapis.com/cloud-training/images/edit.png |
| External link icon | https://storage.googleapis.com/cloud-training/images/external-link.png |
| Green checkmark icon | https://storage.googleapis.com/cloud-training/images/greencheck.png |
| Hide icon | https://storage.googleapis.com/cloud-training/images/hide.png |
| Menu icon | https://storage.googleapis.com/cloud-training/images/menu.png |

| | |
|---|---|
| More icon | https://storage.googleapis.com/cloud-training/images/more.png |
| Search icon | https://storage.googleapis.com/cloud-training/images/search.png |
| Settings icon | https://storage.googleapis.com/cloud-training/images/settings.png |
| Web preview icon | https://storage.googleapis.com/cloud-training/images/web-preview.png |
| Write icon | https://storage.googleapis.com/cloud-training/images/write.png |

## Unlabeled UI controls

For *unlabeled* UI elements, such as the controls in the Google Cloud console menu, include a proportional image of the control in parentheses after the name.

**Examples**

- In the Google Cloud console, in the **Navigation menu** (☰), click **Storage**.
- In the **Navigation menu** (☰), click **Storage**.
- On the Google Cloud console title bar, click **Activate Cloud Shell** ( ).

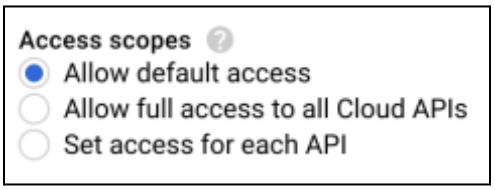
You don't have to include an image of the unlabeled control at every mention; use your judgment to ensure that learners will be familiar with the location of unlabeled controls.

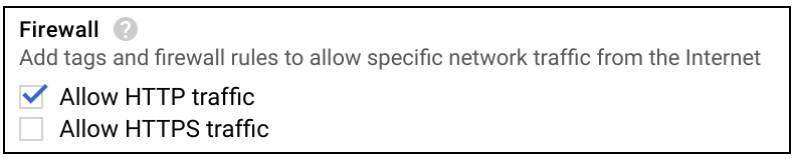
## UI element types

Follow these examples and rules for referring to UI controls and other onscreen elements.

- Pay special attention to the prepositions (*in, on*, etc.) and verbs (*click, select,* etc.).
- Note that most references to controls do not mention the control *type.*
- UI images in this table are only for illustration; try not to include screenshots in your tasks.

| Control Type | Example |
|---|---|
| Page | On the **Create an instance** page, specify the following: |
| Dialog box | In the **Compute Engine VM Instances** dialog, click **Create**. |

| | |
|---|---|
| Text box | • For **Name**, type a unique name for your instance.<br>• For **Name**, type **test-instance**<br><br>**Note:** See User-entered text (GUI only) for guidelines about formatting and punctuation of user-entered text. |
| Drop-down list | For **Machine type**, select **micro**. |
| Command button | Click **Start**.<br><br>**Note:** Use sentence case even if the UI label is all caps. |
| Option (radio) button | For **Access scopes**, select **Allow default access**.<br><br>**Note:** Refer to the label that is above or next to the More buttons.<br>  |
| Checkbox | • Select the **Deploy a container image to this VM instance** checkbox.<br>• Clear the **Deploy a container image to this VM instance** checkbox. |
| Menu item | • In the Cloud console menu, click **Activate Cloud Shell** ( ).<br>• In the **Navigation menu** ( ), click **Compute Engine > Images**.<br><br>**Note:** Use a bold “greater than” sign with a space before and after to separate elements of the menu path. |
| Navigation pane | In the navigation pane, click **Instance groups**. |
| Tab | • Click the **Disks** tab.<br>• On the **Disks** tab, select **Delete boot disk when instance is deleted**. |
| Link | Click **command line**.<br><br>**Note:** Use sentence case even if the link text is all caps. |
| Section | Under **Firewall**, select **Allow HTTP traffic**. |

| | |
|---|---|
| |  |
| Expandable section | Expand the **Management, disk, networking, SSH keys** section.<br>  |
| Overflow menu | In the row for **instance-1**, click the overflow ( ⋮ ) menu, and then click **Reset**. |

## Default (unchanged) settings

If some controls on a UI surface should remain unchanged, note this with a phrase like “leave the remaining settings as their defaults.”

**Example**

1. For **Name**, type **new instance**, and then leave the remaining settings as their defaults.
2. Click **Create**.

Also see Combining 3 or more actions.

## User-entered text (GUI only)

Use **bold** for text that the user should enter (type or paste) exactly, and use italics (not bold) and square brackets for text that the user creates (placeholder text).

**Examples**

- For **Name**, type **test-instance**
- For **Description**, type **Test instance for** [*your initials*]
  Replace the square brackets and italic text with your initials.

**Note:** If the user-entered text is at the end of the sentence, do *omit* ending punctuation. This ensures that the learner doesn’t mistakenly type the punctuation, and also makes copy/pasting easier for the learner.

# Referring to keyboard actions

- Use *press*, not *hit*, for keyboard actions.
- Use all capitals for key names, regardless of the actual labels on your keyboard.
- Use a plus sign (+) for key combinations (keys pressed at the same time).
- Use a comma for key sequences (keys pressed in order).
- Do not use bold.

**Examples**

- Press ENTER.
- Press CTRL+C.
- Press CTRL+O, ENTER, CTRL+X.

# Referring to code

Always use code formatting when referring to code, even in a sentence. However, text that the user enters in a GUI should be formatted as bold, not code.

## Results

If it's necessary to display the actual output, use the `Consolas` font in a Google Doc or the `<ql-code-block output>` tag in markdown.

Consider providing a *description* of the expected output instead of pasting the actual command output. This is more localization-friendly.

**Example**

| Actual code output | Description |
|---|---|
| `Error starting adb shell.`<br>`Activity not started.` | If an error occurs, a message states that the adb shell was not started. |

**Note:** Do *not* insert a screenshot of the code output; it is difficult to read, and the contents aren't accessible to screen readers.

## Placeholder text for code

To indicate placeholder text, use square brackets and format the text as italics.

Instruct the learner to paste the text into a text editor, replace the placeholder text with the correct text, and then paste the code block into Cloud Shell.

## Including results

Include the result of an action within a step only if it's unexpected or has some explanatory value. Remember that learners are familiar with standard UI; for example, they expect a new page to open when they click **Next**.

- List the result on a separate line from the instruction.
- Do not list a result as a separate step; results are not actions.
- Describe what happens, not what the user "sees."

| **Say this...** | **Not this...** |
| --- | --- |
| 1. Click **Activate Cloud Shell** ( ). A Cloud Shell session opens and displays a command-line prompt. | 1. Click **Activate Cloud Shell**.<br>2. A Cloud Shell session opens and displays a command-line prompt.<br><br>---<br><br>1. Click **Activate Cloud Shell**. You will see a Cloud Shell session open and a command-line prompt appear. |

## Arrows

To write a series of steps as a string, you can put an arrow ( > ) between the UI elements the learner will be clicking. Here are some guidelines:

- Arrows are formatted as ">", not "->".
- Arrows are bolded.
- Any button that is getting clicked is also bolded.

**Example**

**Note:** you may need to rerun this command and wait until the output returns three instances that are HEALTHY. You can also monitor it in the console (Navigation menu **>** Network services **>** Load balancing).

## Referring to UI display values

Use italics to refer to values displayed by the Cloud console, such as values in a table or status displays.

**Examples**

- The location of cluster-1 is *us-central1-a.*
- Continue when the status is *Running.*

# Fragments

Several parts of a typical lab manual have been rendered as reusable text that you can insert where appropriate. These fragments are available in the Fragments folder in the SPL repo.

To use a fragment, copy its link to your clipboard and embed it in your lab manual like this:

[[**import** file link]]

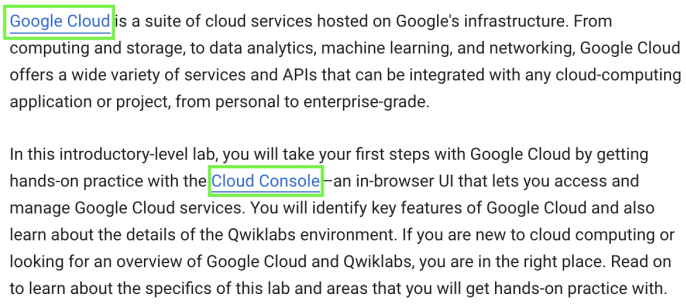
Fragments purposefully do not have headers, so embed them under a normal header in your lab manual.

## URLs

URLs are helpful to include in a lab when directing students to additional information, however, it should **not** be required reading for a lab. Since labs are timed, students should not be sent out of the lab. If you are using URLs in your instructions, use them the following way:

- URLs are formatted as hyperlinks.
- Hyperlink text describes the page the URL points to, the article title, or clearly explains where the user will be taken. Example: Learn more about how to [xxx] from [xxx] Guide.
- If URLs are inside an infobox, format them as HTML href tags.

**Good examples of URL use**

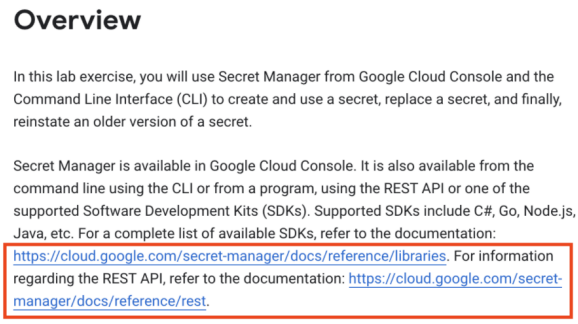


Google Cloud is a suite of cloud services hosted on Google's infrastructure. From computing and storage, to data analytics, machine learning, and networking, Google Cloud offers a wide variety of services and APIs that can be integrated with any cloud-computing application or project, from personal to enterprise-grade.

In this introductory-level lab, you will take your first steps with Google Cloud by getting hands-on practice with the Cloud Console—an in-browser UI that lets you access and manage Google Cloud services. You will identify key features of Google Cloud and also learn about the details of the Qwiklabs environment. If you are new to cloud computing or looking for an overview of Google Cloud and Qwiklabs, you are in the right place. Read on to learn about the specifics of this lab and areas that you will get hands-on practice with.

**Bad examples - do not use URLs like this**

- URLs are not formatted as hyperlinks
- Hyperlink text is not descriptive.
    - Example: Hyperlink text is the same as the URL it points to (see screenshot below)

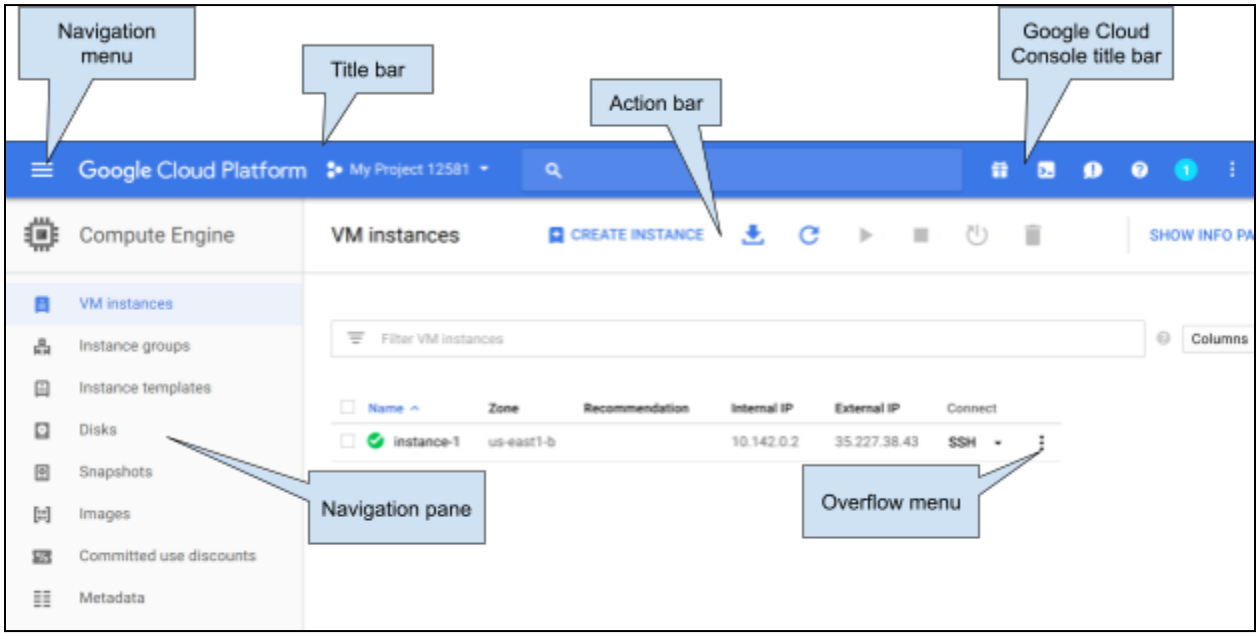


## Overview

In this lab exercise, you will use Secret Manager from Google Cloud Console and the Command Line Interface (CLI) to create and use a secret, replace a secret, and finally, reinstate an older version of a secret.

Secret Manager is available in Google Cloud Console. It is also available from the command line using the CLI or from a program, using the REST API or one of the supported Software Development Kits (SDKs). Supported SDKs include C#, Go, Node.js, Java, etc. For a complete list of available SDKs, refer to the documentation: https://cloud.google.com/secret-manager/docs/reference/libraries. For information regarding the REST API, refer to the documentation: https://cloud.google.com/secret-manager/docs/reference/rest.

# Appendix 1: Parts of the Google Cloud console




# Appendix 2: Additional lab writing resources

- Lab Design Standards resource