* [Console and Cloud Shell](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287" \l "console-and-cloud-shell)
* [Overview](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#overview)
  + [**What you will do:**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#what-you-will-do)
  + [You will learn:](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#you-will-learn)
* [Create a bucket using Console](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#create-a-bucket-using-console)
  + [**Step 1: Navigate to the Storage service**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-1-navigate-to-the-storage-service)
  + [**Step 2: Create the bucket**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-2-create-the-bucket)
  + [**Step 3: Explore Console features**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-3-explore-console-features)
* [Accessing Cloud Shell](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#accessing-cloud-shell)
  + [**Step 1: Open Cloud Shell**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-1-open-cloud-shell)
  + [**Step 2: Explore Cloud Shell features**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-2-explore-cloud-shell-features)
* [Create a bucket using Cloud Shell](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#create-a-bucket-using-cloud-shell)
  + [**Step 1: Open Cloud Shell**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-1-open-cloud-shell)
  + [**Step 2: Create a second bucket**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-2-create-a-second-bucket)
  + [**Step 3: Verify the bucket in Console**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-3-verify-the-bucket-in-console)
  + [**Step 4: Delete buckets**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-4-delete-buckets)
* [More Cloud Shell features](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#more-cloud-shell-features)
  + [**Step 1: Explore more Cloud Shell features**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-1-explore-more-cloud-shell-features)
  + [**Step 2: Tab feature**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-2-tab-feature)
  + [**Step 3: Theme color feature**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-3-theme-color-feature)
  + [**Step 4: Close Cloud Shell**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-4-close-cloud-shell)
* [Create persistent state in Cloud Shell](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#create-persistent-state-in-cloud-shell)
  + [**Step 1: Learn about Cloud Shell VM**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-1-learn-about-cloud-shell-vm)
  + [**Step 2: Identify available regions**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-2-identify-available-regions)
  + [**Step 3: Create and verify an environment variable**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-3-create-and-verify-an-environment-variable)
  + [**Step 4: Create a subdirectory**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-4-create-a-subdirectory)
  + [**Step 5: Create an empty file**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-5-create-an-empty-file)
  + [**Step 6: Append environment variable assignment to the file**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-6-append-environment-variable-assignment-to-the-file)
  + [Step 7: Create the environment variable with **source**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-7-create-the-environment-variable-with-source)
  + [**Step 8: Cycle Cloud Shell -- and the environment variable is gone**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-8-cycle-cloud-shell-and-the-environment-variable-is-gone)
  + [**Step 9: Modify the bash profile**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-9-modify-the-bash-profile)
  + [**Step 10: Cycle Cloud Shell and verify persistence**](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#step-10-cycle-cloud-shell-and-verify-persistence)
* [Review of the GCP interface](https://googlecloud.qwiklabs.com/labs/378/instructions?focus_id=3287#review-of-the-gcp-interface)

# CONSOLE AND CLOUD SHELL

# Overview

In this lab you will become familiar with the GCP web-based interface. There are two integrated environments, a GUI (graphical user interface) environment called Console, and a a CLI (command line interface) called Cloud Shell. In this class we will be using both environments.

Here are a few things you need to know about Console:

* Console is under continuous development. So occasionally the graphical layout changes. This is most often to accommodate new GCP features or changes in the technology resulting in a slightly different workflow.
* You can perform most common GCP actions in Console, but not all actions. In particular, very new technologies or sometimes detailed API or command options are not implemented (or not yet implemented) in Console. In these cases, the command line or the API are the best alternative.
* Console is extremely fast for some activities. Console can perform multiple actions on your behalf that might require many CLI commands. It can also perform repetitive actions. In a few clicks you can accomplish activities that would require a great deal of typing and would be prone to typing errors.
* Console is able to reduce errors by only offering up through its menus valid options. It is able to leverage access to the platform "behind the scenes" through the SDK to validate configuration before submitting them. A command line can't do this kind of dynamic validation.

## ****What you will do:****

* Get access to GCP
* Create a Cloud Storage bucket using Console
* Create a Cloud Storage bucket using Cloud Shell
* Become familiar with Cloud Shell features

## You will learn:

* How to navigate Console
* How to submit commands in Cloud Shell
* How to use , , and  to create persistent environment variables

# Create a bucket using Console

In this section you will be creating a bucket. However, the text also helps you become familiar with how actions are presented in the lab instructions in this class, and teaches you about the Console interface.

## ****Step 1: Navigate to the Storage service****

In this class we will use menu notation like this:

Console: **Products and services > Storage > Browser**

This means:

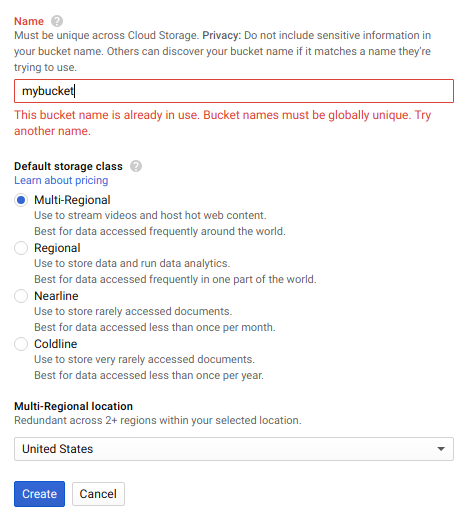
1. In a browser, go to: [https://console.cloud.google.com](https://console.cloud.google.com/)
2. Click on the **Products and Services** Icon in the upper left corner:fd7b4bb90ada52b0.png
3. Hover over **Storage** and then click on **Browser**:



## ****Step 2: Create the bucket****

Click on **[Create Bucket]**.

Result:



In the labs there are often dialogs in Console or options on the command line that need to be filled in with values. In this current case, you need to select a **Default storage class**. Values and properties will appear in lab instructions in a table like this:

|  |  |
| --- | --- |
| **Property** | **Value** |
| **Name:** | Your globally unique bucket name |
| **Default storage class**: | Multi-Regional |

Sometimes you will be instructed in text like this:

Select the \_Multi-Regional \_storage class.

Click **[Create]**

## ****Step 3: Explore Console features****

In the upper right corner of Console there is a **Notifications** icon. It looks like a bell.



Sometimes feedback from the underlying commands is provided there. If you are not sure what is happening, you should check **Notifications** for additional information and history.

# Accessing Cloud Shell

You can use the Cloud Shell to manage projects and resources via command-line, without having to install the Google Cloud SDK and other tools on your computer.

Cloud shell provides the following:

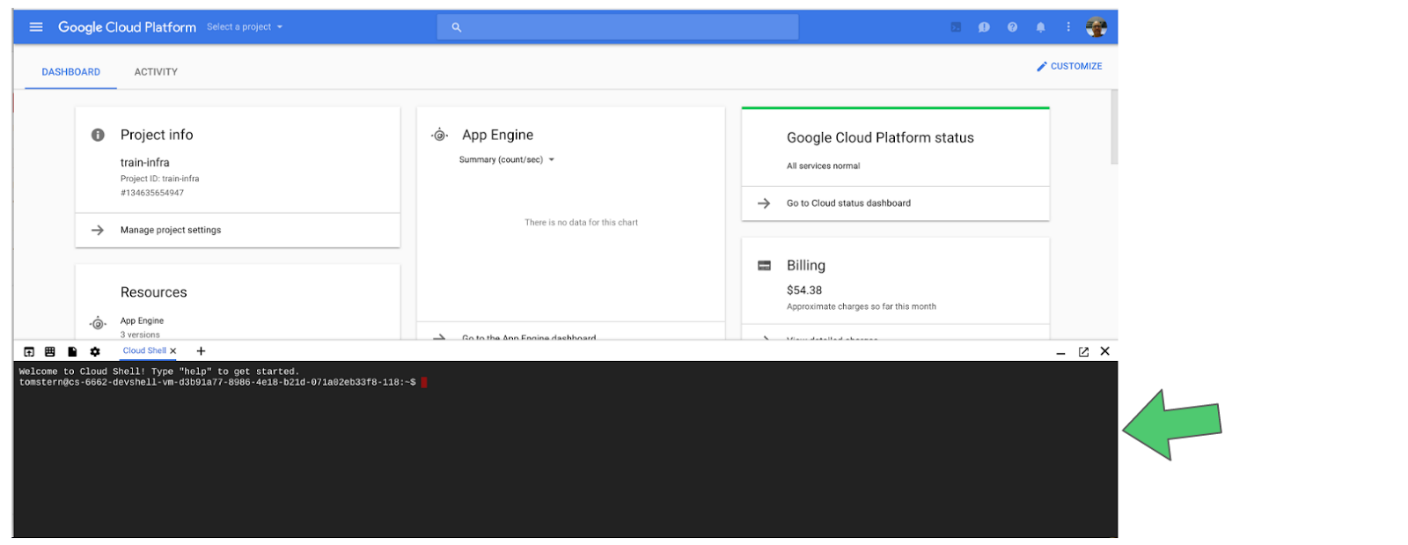
* Temporary Compute engine VM
* Command-line access to the instance via a browser
* 5 GB of persistent disk storage ($HOME dir)
* Pre-installed Google Cloud SDK and other tools
*  - for working with GCE and many GCP services
*  - for working with Cloud Storage
*  - for working with GKE and Kuberenetes
*  - for working with Big Query
* Language support for Java, Go, Python, Node.js, PHP, and Ruby
* Web preview functionality
* Built-in authorization for access to resources and instances

After 30 minutes of inactivity, the Cloud Shell instance is recycled. Only the /home directory persists. Any changes made to the system configuration, including environment variables, is lost between sessions.

## ****Step 1: Open Cloud Shell****

Open Cloud Shell by clicking on the icon in the upper right corner of Console.  


Cloud Shell will open beneath the Console window.

Result:  


## ****Step 2: Explore Cloud Shell features****

In the upper right corner of Cloud Shell are three icons.



1. The first one hides and restores the window, giving you full access to Console without closing Cloud Shell. Try it now.
2. Having Cloud Shell on the bottom of the Console window is useful when you are issuing individual commands. However, sometimes you will be editing files or want to see the full output of a command. For these uses, the second icon pops the Cloud Shell out into it's own full size terminal window. Try it now.
3. The **X** icon closes Cloud Shell. Every time you close Cloud Shell the Virtual Machine is recycled and all machine context is lost. Close Cloud Shell now.

# Create a bucket using Cloud Shell

## ****Step 1: Open Cloud Shell****

Open Cloud Shell by clicking the Cloud Shell icon.

## ****Step 2: Create a second bucket****

Use the  command to create a new bucket.

The bucket requires a globally unique name.

gsutil mb gs://[Bucket\_Name]

## ****Step 3: Verify the bucket in Console****

Verify that the bucket was created using Console.

Console: **Products and Services > Storage > Browser**

You have performed equivalent actions using Console and Cloud Shell.

You created a bucket using Console and another bucket using Cloud Shell.

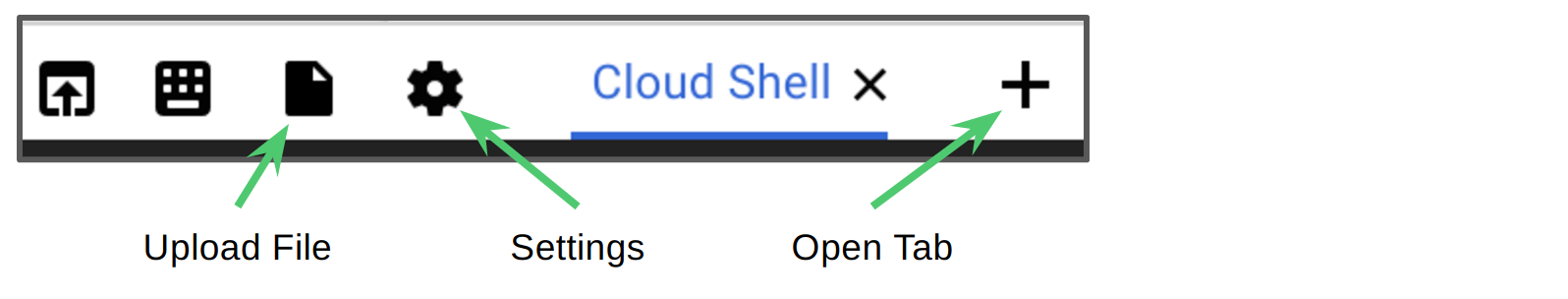
## ****Step 4: Delete buckets****

Select the buckets you created and delete them.

# More Cloud Shell features

## ****Step 1: Explore more Cloud Shell features****

Open Cloud Shell. On the upper left you will see several icons:



## ****Step 2: Tab feature****

Click on the Open Tab icon to open another Cloud Shell terminal in a tab. You can now switch back and forth between the terminal sessions by clicking on the **Cloud Shell** titles.

## ****Step 3: Theme color feature****

Click on Settings and select \_\_ Switch to dark theme\_\_. You can toggle back and forth between light and dark theme colors using these settings. Try it now.

Result:



## ****Step 4: Close Cloud Shell****

Close all the Cloud Shell sessions.

# Create persistent state in Cloud Shell

In this section you will learn a best practice for using Cloud Shell. The gcloud command often requires specifying values such as a **Region** or **Zone** or **Project ID**. Entering them repeatedly increases the chances of making typing errors. If you use Cloud Shell a lot you may want to set common values in environment variables and use them rather than typing the actual values.

## ****Step 1: Learn about Cloud Shell VM****

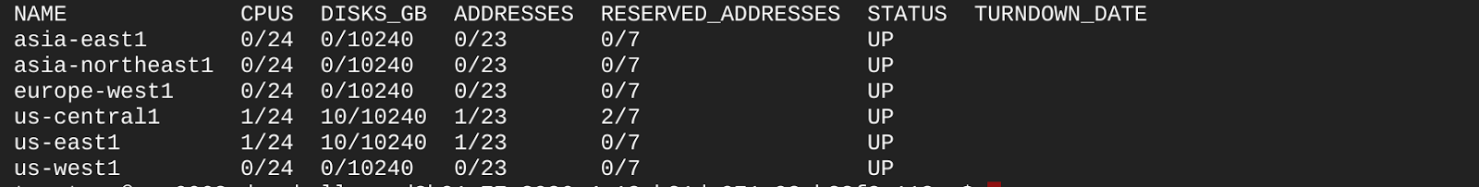
Open Cloud Shell from the Console. Note that this allocates a new VM for you.

## ****Step 2: Identify available regions****

List available regions. Select a region and create an environment variable to hold the default region value.

gcloud compute regions list

Result:



Select a region from the list.

## ****Step 3: Create and verify an environment variable****

Create an environment variable. Verify it with .

INFRACLASS\_REGION=[Your\_Region]

echo $INFRACLASS\_REGION

You can use environment variables like this in gcloud commands to reduce the opportunities for typos, and so that you won't have to remember a lot of detailed information.

When you close Cloud Shell and reopen it, a new VM will be allocated, and the environment variable you just set will disappear. In the next steps you will create a file to set the value so that you won't have to enter the command each time Cloud Shell is cycled.

## ****Step 4: Create a subdirectory****

Create a subdirectory for materials used in this class.

mkdir infraclass

## ****Step 5: Create an empty file****

Create a file in the infraclass directory called **config**.

touch infraclass/config

## ****Step 6: Append environment variable assignment to the file****

The following command will append the value to the **config** file.

echo INFRACLASS\_REGION=$INFRACLASS\_REGION >> ~/infraclass/config

Issue the command again with your Project\_ID. You can find the project ID by going to the Console Home page.

echo INFRACLASS\_PROJECT\_ID=[Project\_ID] >> ~/infraclass/config

## Step 7: Create the environment variable with

Use the  command to set the environment variables. Verify that the project variable was set using the  command.

source infraclass/config

echo $INFRACLASS\_PROJECT\_ID

This gives you a method to create environment variables and to easily recreate them if the Cloud Shell is cycled. However, you will still need to remember to issue the source command each time Cloud Shell is opened.

In the next step you will modify the .bash\_profile file so that the source command is issued automatically any time a terminal to Cloud Shell is opened.

## ****Step 8: Cycle Cloud Shell -- and the environment variable is gone****

Close and re-open Cloud Shell. Then issue the echo command again.

echo $INFRACLASS\_PROJECT\_ID

There will be no output because the environment variable no longer exists.

## ****Step 9: Modify the bash profile****

Change to the home directory using **cd** and edit the  file with either the  or  text editor. Add the following line to the end of the file.

Copy this line.

source infraclass/config

In  you would do the following.

 to edit the file  
 (little-oh) places you at the end of the file on a new line.

 to paste the above lines into the file.

 to write the file and quit the editor.

## ****Step 10: Cycle Cloud Shell and verify persistence****

Close Cloud Shell and open Cloud Shell to cycle the VM. This time when you use the echo command, the variable should still be set.

echo $INFRACLASS\_PROJECT\_ID

You should now see the expected value that you set in the config file.

If you ever find your Cloud Shell environment corrupted, you can find instructions on resetting it here:

<https://cloud.google.com/shell/docs/limitations#resetting_cloud_shell_to_default_state>

This will cause everything in your Cloud Shell environment to be set back to its original default state.

# Review of the GCP interface

Cloud Shell is an excellent interactive environment for exploring GCP using Google Cloud SDK commands like  and .

You can install the Google Cloud SDK on a computer or on a VM instance in GCP. The gcloud and gsutil commands can be automated using a scripting language like bash (Linux) or Powershell (Windows). You can also explore using the command line tools in Cloud Shell and then user the parameters as a guide for re-implementing in the SDK using one of the supported languages.

The GCP interface consists of two parts, Console and Cloud Shell

**Console, the GUI provides:**

* Fast way to get things done
* Presents options to you, instead of requiring you to know them
* Performs behind-the-scenes validation before submitting the commands

**Cloud Shell provides:**

* Best detailed control
* Complete range of options and features
* Path to automation through scripting

©Google, Inc. or its affiliates. All rights reserved. Do not distribute.