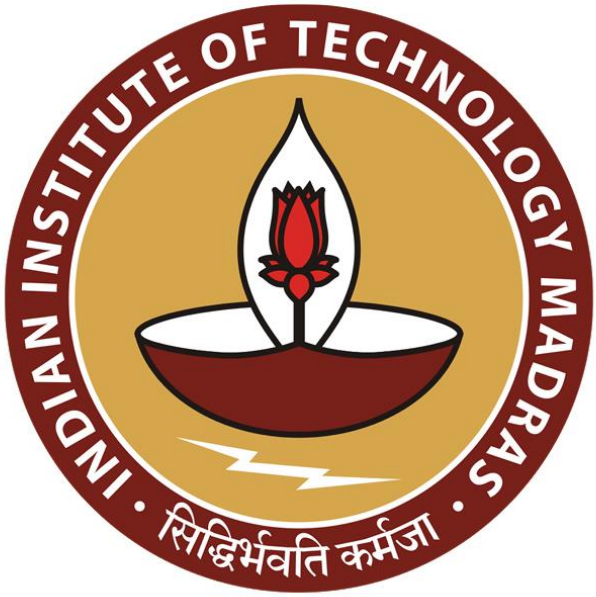


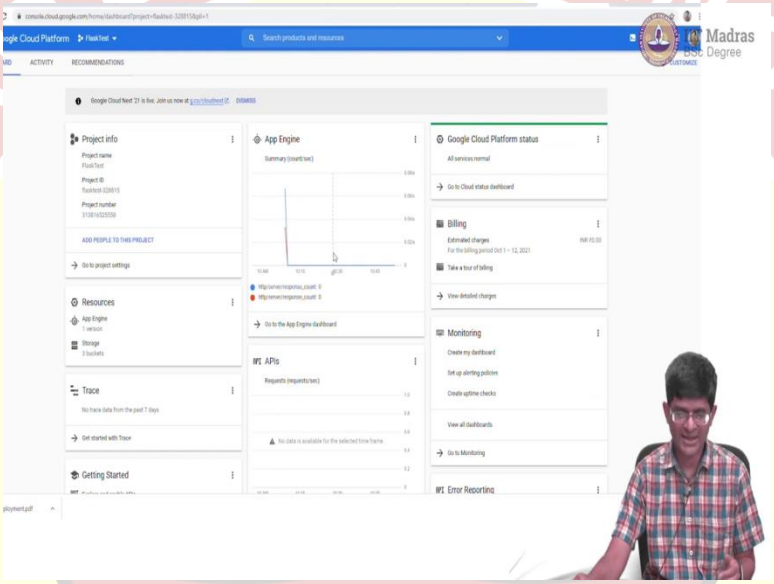
IIT Madras

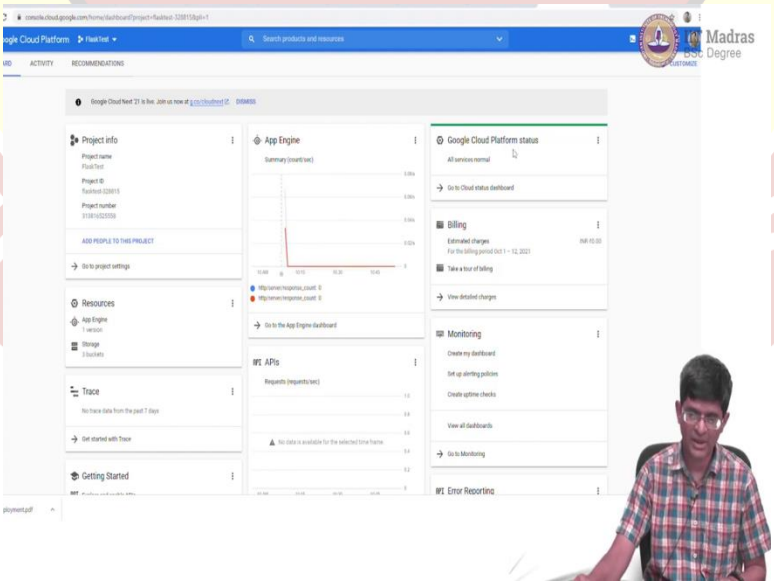
ONLINE DEGREE

Modern Application Development-I
Professor. Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Service Approach II

Hello, everyone, and welcome to this course on modern application development.

(Refer Slide Time: 00:16)





So, the third service provider that I want to just mention about over here, which is on a totally different scale than what Replit for example, does is Google App Engine. And Google App

Engine, as opposed to Google Compute Engine, is also a platform provider. So, unlike Google Compute Engine where you can pretty much just create a virtual machine and then you are in your own, you install everything thereafter that.

In App Engine, you can get started with, assuming that there is a certain version of python, the web server front end is taken care of for you, various other things are sort of automatically handled for you which means that you can now focus on your code. So, how exactly does App Engine work? You can sign up for an App Engine account, a Google Cloud account, so the URL that you might notice on the top in the URL bar is console.cloud.google.com.

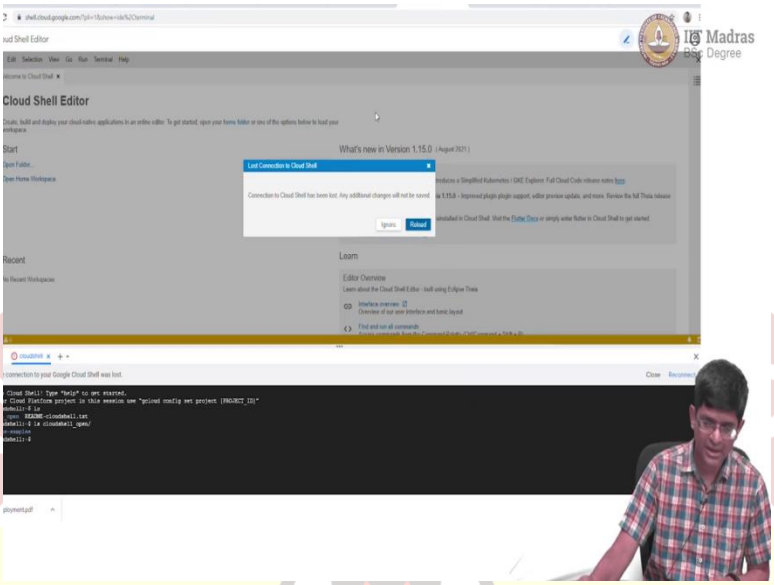
So, cloud.google.com is essentially how you get access to the Google Cloud. The console over there is essentially this thing, which shows you a sort of, a summary of everything that is available on Google Cloud for your particular project. So, you can see that up here on the top there is something which is flask test, which is I have just created a sort of dummy app, write a project to demonstrate how you can create an app out here.

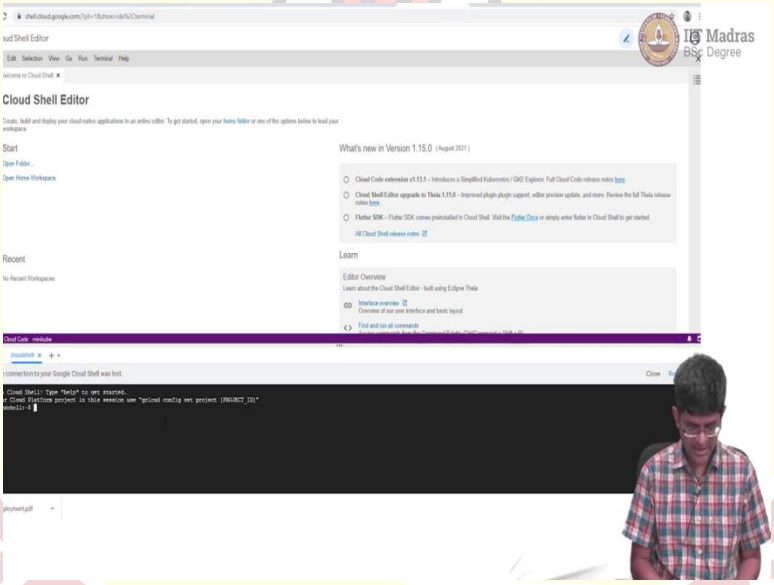
It tells me a little bit of information about the project, it says what kind of resources it is using. It also shows me what are the number of requests per second that have been received. In this case, it is zero because I had just tried accessing it once a little while back. And, that **is** about it. There are no more requests recently.

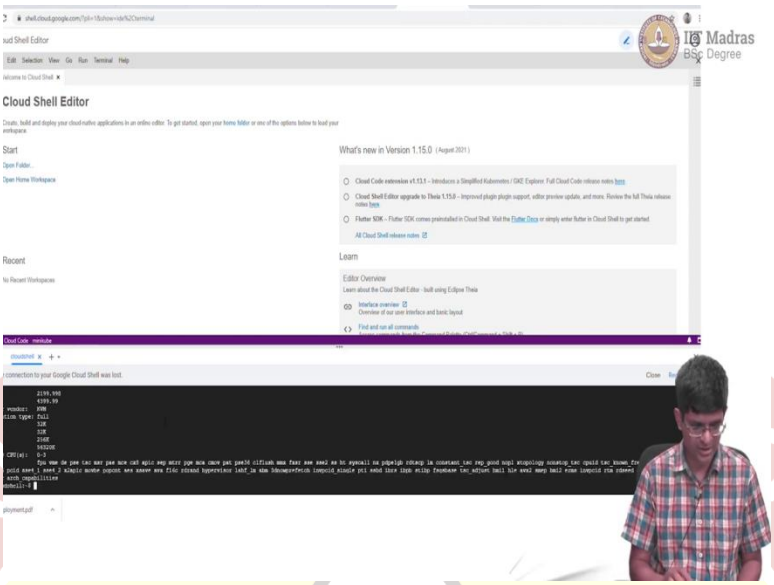
It shows you the overall platform status. So, that is the good thing. It is telling you that services are normal, what are the services? That includes the web front end, the HTTPS, the secure services, the monitoring, logging, any databases. I am not using a database, but if there was any database connectivity, what is the status of those all that is what is considered as part of the Cloud Platform. Of course, it also shows what are the estimated billing charges, because this is on a paid basis.

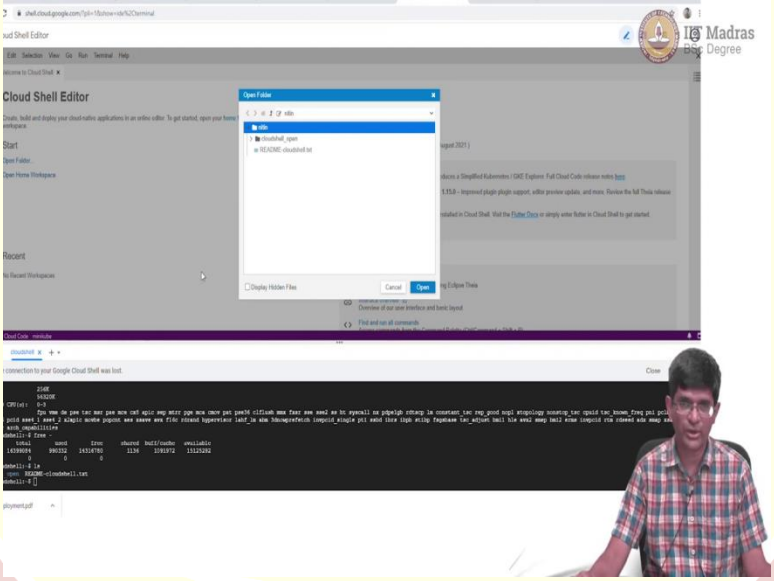
Having said that, I believe that the basic app engine at least like a very simple thing, if you try creating, it falls under the free tier, there are some things that you can use in order to experiment at least. And if you sign up newly, then you get some amount of credits for one year that you can use in order to learn more about the system. So, now this is the overall console, what you can do is you can also invoke something called the Cloud Shell.

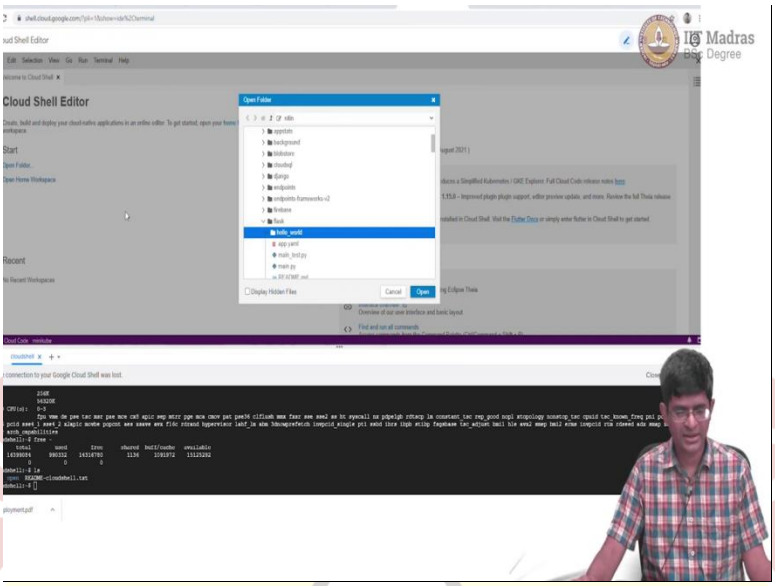
(Refer Slide Time: 02:54)

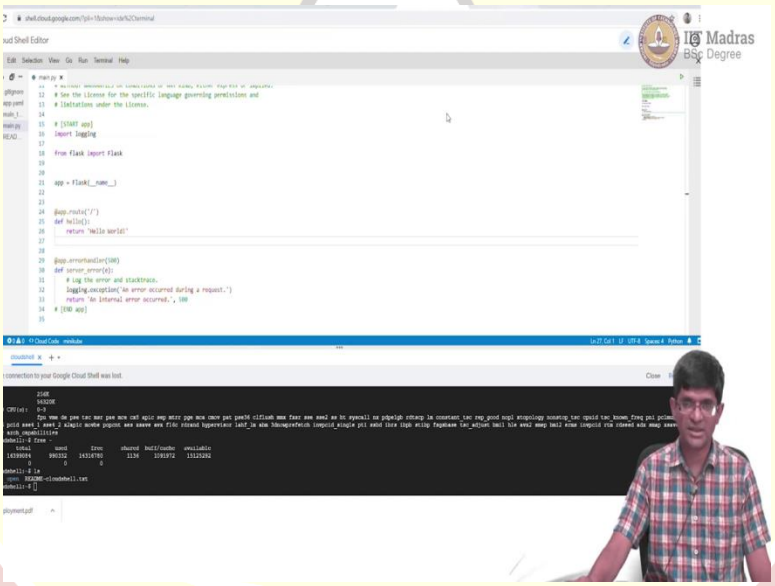





Cloud Shell Editor
Start
Open Folder...
Open Home Workspace
Recent
No Recent Workspaces
What's new in Version 1.15.0 (August 2021)
Learn
Editor Overview
Learn about the Cloud Shell Editor - built using Eclipse Theia
Interface overview
Overview of our user interface and basic layout
Find and run all commands


Cloud Shell Editor
Open Folder
Start
Recent
No Recent Workspaces
Display Hidden Files
Cancel
Open





And the Cloud Shell is, so the Cloud Shell, in the case of Google Cloud, is that it is essentially once again, providing you access to a Linux machine. Similar to what Replit or glitch had. But in this case, it also has a lot of the infrastructure or, other commands that have been custom installed over there that allow you to sort of get started using the Google Compute Engine, Google App Engine, and various things related to that.

So, you will see, for example, that, when you log in, it gives you some extra information, like to set your cloud platform project in the session, use G Cloud Config set, and so on. Now, what is all that I mean, it also has a slightly different interface, it is a little less sort of flashy than what

Replit or glitch look like. And it does not sort of present you with a nice browser interface, a nice file manager, all of those are there.

But the main focus is clearly on, getting things done. So, this is meant more for sort of professional work. Or at least that is the impression they want to give. And the fact of the matter is that yes, it is because the kind of scaling that it does. And the kind of applications that Google App Engine is typically used for are not really toy applications, or some small thing that you and a couple of friends might be developing.

Usually would want to go this when you are ready to sort of start scaling up, which means that you should probably have a slightly better understanding of how to use these things. So, you practice on small applications, but then start building up you will definitely need to know more about the shell and how to use that when you want to scale to large applications.

Once again, just like the others, this is a Linux machine which means that I can type in the same lscpu and the free commands. And I find that in fact, once again, over here, I have four CPUs. And in this case, it has like 16 GB of RAM that it provides to me. So, like I said that by itself does not really say anything, it is not that the 16 GB or the 26 GB or the 6 GB is going to make a difference to your application.

This is just for the temporary development environment that they are providing you. Once you have things working, you are expected to take that and figure out what kind of overall system you want to deploy it on. Now, in this case, what I did was I essentially had cloned one of the examples that they provide as part of the Google Cloud.

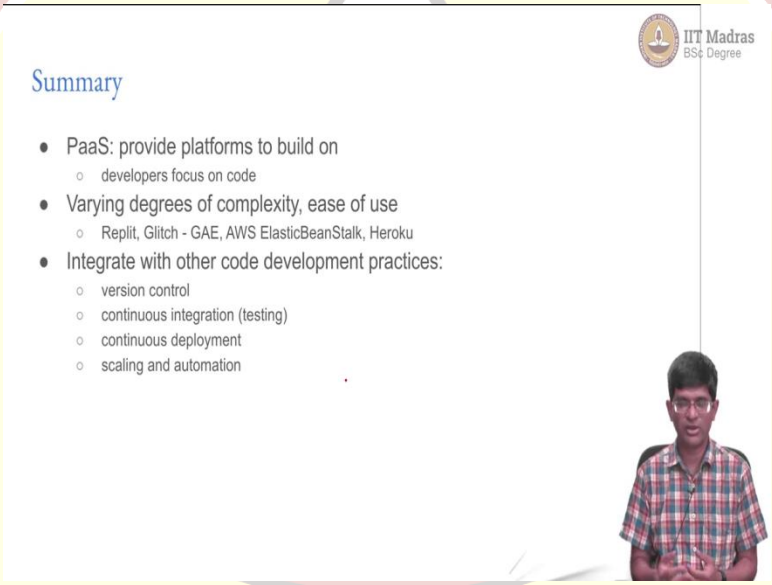
And, in fact, what you can do is you can pretty much just go to one of the folders. And those of you who are familiar with this might realize that, this is essentially showing the editor that they are using is pretty much like Visual Studio code. It is using the same interface. And the reason is, Visual Studio code itself is an application in JavaScript, and therefore can be run as part of a web based system as well.

I believe this is based on VS code, or at least something very similar to that which is being used. So, what we can do is, you can go in there and open any file that you want, in this case, I had already installed or downloaded some of the application examples, from App Engine. And what

we have is there is a standard one for flask. If we go down here, we find that there is a flask application that says hello world.

And if I open that folder, what I will find is that, it has, once again it gives me the same, kind of interface that I can use in order to edit files. And in my main.py, the main Python file, it shows me what it is going to run over there. And that is, once again, just start the flask. And in app dot route, slash, all that it does is return hello world. Now, if I go back to my console, I will find that there is a particular URL that I can go and visit that it shows me saying, this is where it is going to be deployed.

(Refer Slide Time: 07:24)



And the link that I have shown right at the top is the particular instance that I had created, something.appspot.com. Flash test was the name that I gave, it adds on some numbers over there to make it unique. And then it allows me to create something.appspot.com, and hello world, which is after all the output of the slash route that was created out there.

So, let us summarize what we looked at. In terms of the examples first, we saw replit, which you have all used as part of your course, the main focus, so there is more on learning how to code and, maybe even collaborative development, definitely collaborative teaching, it is helpful from that point of view. But it is also useful for collaborative development. They provide a setup whereby you can run your app.

And typically, the condition is that after a certain amount of time, the app will automatically shut down. If there are no requests, so then the app will shut down. And then if you access it again, then it restarts it. But it might take a certain amount of time. And they have some restrictions on how much load you are allowed to put on it, and so on. Glitch is the next option that we looked at very similar in some ways to what Replit can do does not really have the education aspect sort of built in from the beginning.

But they do have some kind of specialized features that try to sort of, make it easier for people who are developing an app and going through the initial stages of deployment, so that you can get it tested and multiple people can even work on developing and working on the app. And Google App Engine was one example I showed where, it is on a different scale. The complexity of the interface itself is significantly, it is harder to use.

But the reason for that is because it is meant for harder use cases, is meant for the usage in a scenario where you have a lot more scaling and other issues to be concerned about. And it is not just a very small example that you are trying to work with. So, the degree of complexity, ease of use can make a big difference, I am not saying that replit or Glitch cannot be used to develop a complex app they can. And I am not saying that Google App Engine cannot be used to develop a simple app. Both of those cases are possible. It is just which one the main focus is on.

And apart from that, of course, so the PaaS, in other words, just providing a complete platform, not just a machine, but also installing the operating system, installing a web server, taking care of HTTPS, taking care of, making sure that the file system has enough space, there is enough memory. What happens if a server crashes?

Do you get warnings when you are sort of, exceeding certain limits in terms of usage, and so on, all of that is taken care of as part of the platform. That is the service that they provide. And there are multiple different service providers, I showed you only Google App Engine, but AWS, Amazon has ElasticBeanStalk, there is something else called Heroku.

All of those have very similar kinds of behavior, I mean, they provide platforms, which would typically be the more popular ones Python plus flask is one of them phytonDjango, PHP Laravel, Node js, and react, or Node js, and various other kinds of platforms. All of those are common things that you are likely to find in the past providers.

Now, of course, one of the things that needs to be done is you do not want to just be developing code on a browser and, deploying it on a system somewhere. You need to be able to have an integration with a number of other best practices that are used in code development. This includes version control, what is called CI,CD; continuous integration, continuous delivery, continuous deployment, and various degrees of scaling and automation.

So, all of these aspects are also very important from the point of view of the final deployment of an app. And that is thing that will look at in the next video.