

Zero to JupyterHub

Scalable JupyterHub deployments for Education, Research,
Business Analytics & Unicorns

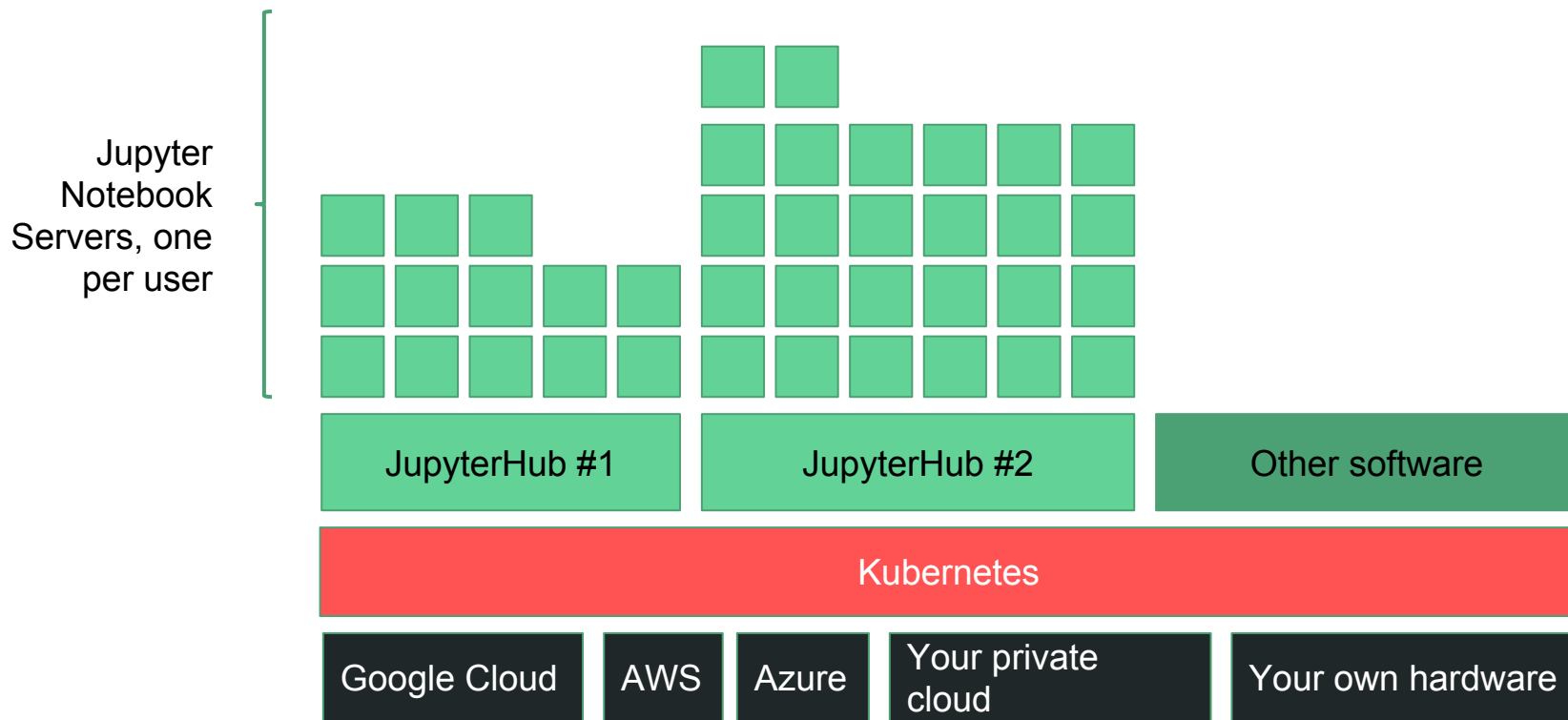
What does “Scaleable” mean?

Works well for large (several thousand active users) & small (10-50 users) installations

Doesn't need constant human operator intervention

Automatically scale cloud resources (up and down) on demand

Architecture



Why Kubernetes?

Why not {{ your-favorite-cluster-orchestrator-here }}

Appropriate level of abstraction

Abstracts away most detail of
underlying cloud providers /
hardware

Declarative high level primitives that
allow you to be as high level or low
level as needed

Utilize features of underlying
hardware when you want (GPUs,
SSDs, etc) easily

Strong,
welcoming &
diverse
community

Not controlled by one single
commercial entity

Fast paced releases that
miraculously keep backwards
compatibility

Has (relatively) done well on
fostering a warm, welcoming
environment for contributors & users

Enough talk, let's jump in!

Let's launch ourselves a JupyterHub with z2jh!

Tutorial Goals

1. Set up your own JupyterHub on Google Cloud (with free credits)
 2. Understand resources allocated for users (RAM, CPU, Disk)
 3. Basic debugging for when things go wrong
 4. Tear everything down
-

Sign up for Google Cloud Free Trial

300\$ of free credits!

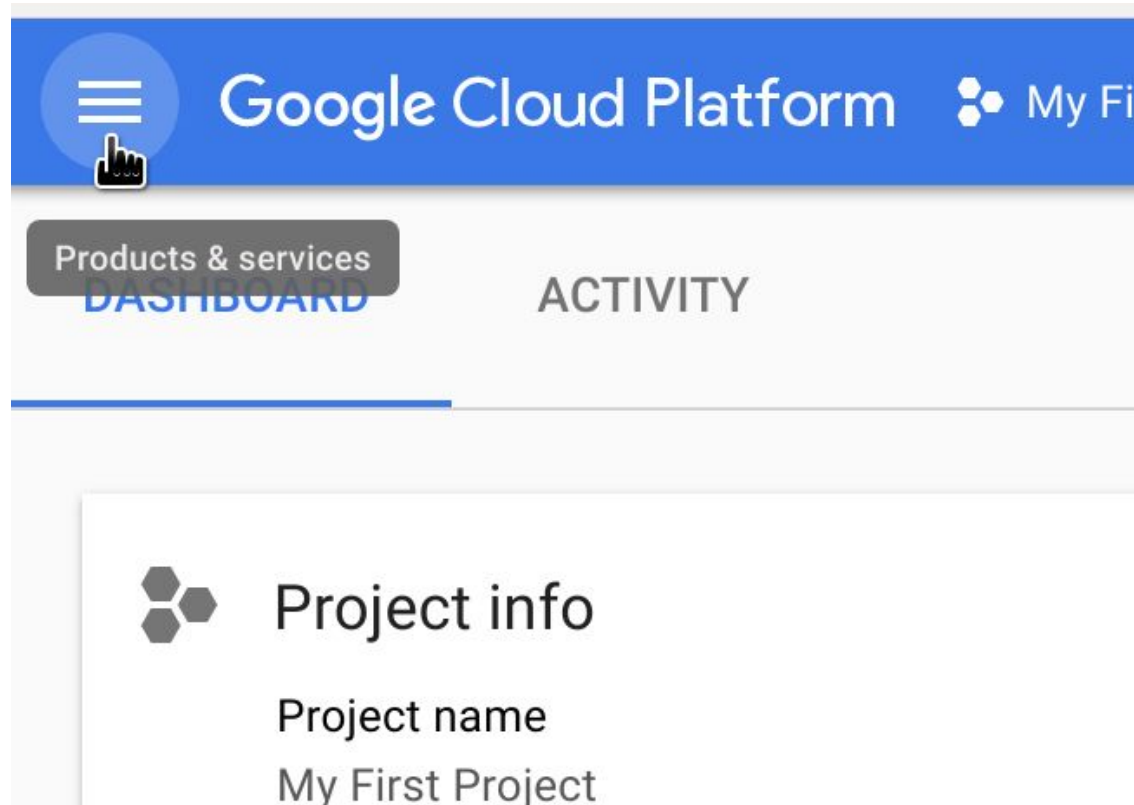
Go to console.cloud.google.com

Click 'Sign up for Free Trial' button

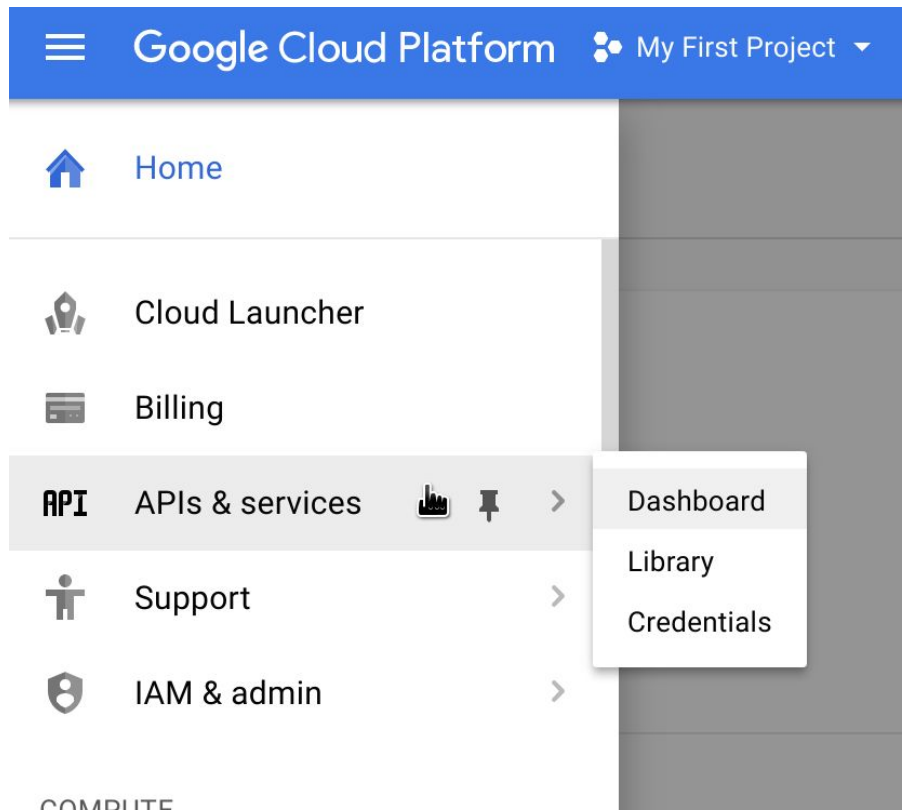
Select 'Individual' account type,
follow instructions

Unfortunately you **do** need a credit
card, but it won't be charged

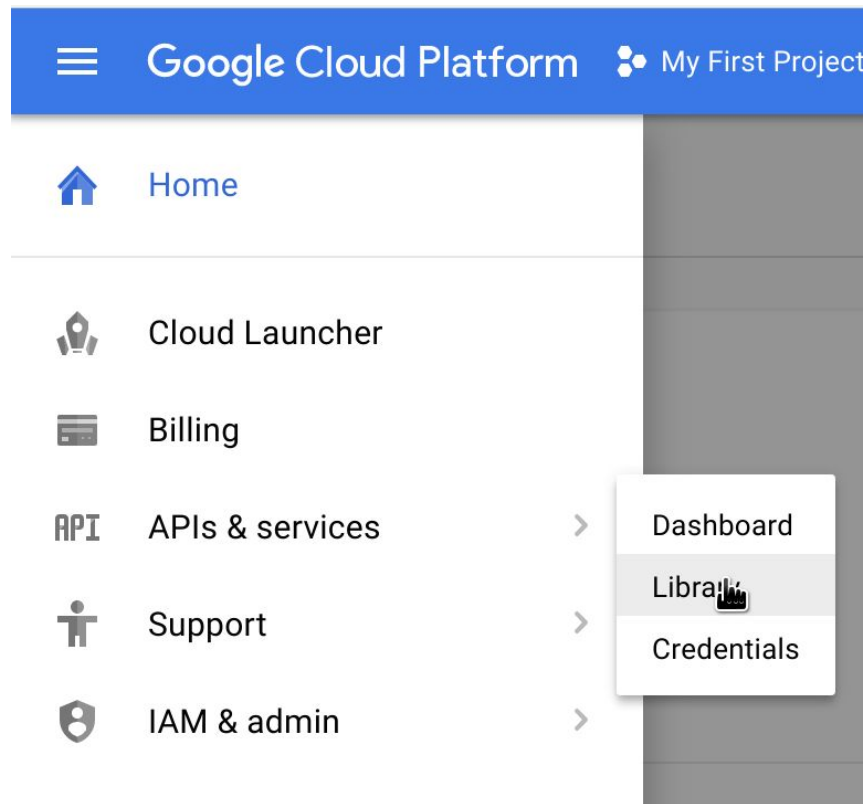
Activate the products we are gonna use



Activate the products we are gonna use




Activate the products we are gonna use



Activate the products we are gonna use

[Google APIs](#)

 compute engine api

[Back to popular APIs](#)

Name	Description
Google Compute Engine API	Creates and runs virtual machines on Google Cloud Platform.



Activate the products we are gonna use



Google Compute Engine API



ENABLE



About this API

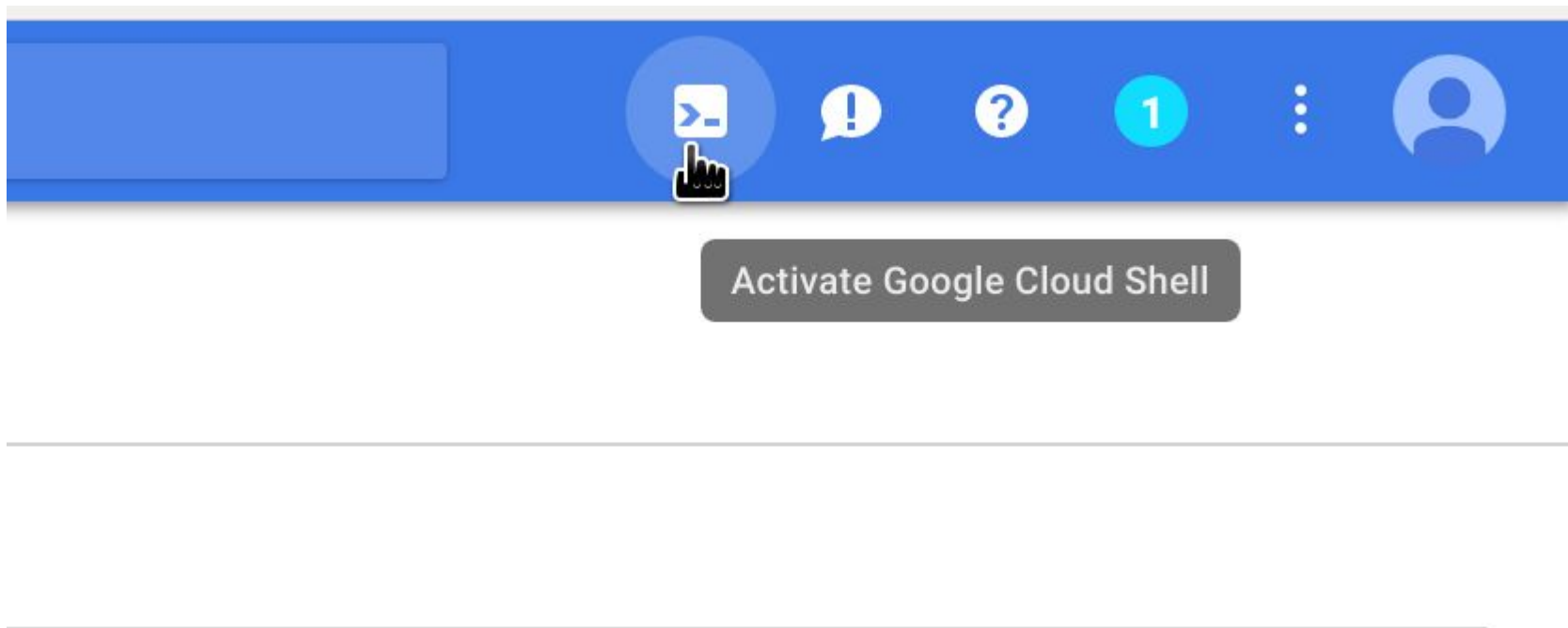
Creates and runs virtual machines on Google Cloud Platform.

Activate the products we are gonna use

[Back to popular APIs](#)

Name	Description
Google Container Engine API 	The Google Container Engine API is used to manage Kubernetes technology.
Google Container Registry API	Google Container Registry provides secure Docker Registry API specification, so we can use the Docker Registry API.

Start a terminal in your browser!



Create a Cluster

```
$ gcloud container cluster create jupytercon \  
    --zone=us-central1-b \  
    --cluster-version=1.7.3 \  
    --num-nodes=3 \  
    --machine-type=n1-standard-2
```

Introduce kubectl

\$ kubectl  get node

NAME	STATUS	AGE	VERSION
gke-test-default-pool-792c3248-4hhn	Ready	4m	v1.7.3
gke-test-default-pool-792c3248-5bhm	Ready	4m	v1.7.3
gke-test-default-pool-792c3248-gqpc	Ready	4m	v1.7.3



Package Manager / App store for
Kubernetes

Define, Install & Upgrade
Applications (“Charts”) that can run
on top of Kubernetes

**Capture entire state of any
installation with one (or more)
YAML files**

Not just JupyterHub: tons of
applications packaged this way

Install helm

```
$ curl -L https://bit.ly/install-helm | bash
```

Downloading

<https://kubernetes-helm.storage.googleapis.com/helm-v2.6.0-linux-amd64.tar.gz>

Preparing to install into /usr/local/bin

helm installed into /usr/local/bin/helm

Run 'helm init' to configure helm.

If this bothers you, go to github.com/kubernetes/helm and download latest release :)

Initialize helm

```
$ helm init
```

```
$HELM_HOME has been configured at /home/jupytercon17_tutorial/.helm.
```

```
Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.
```

```
Happy Helming!
```

Add JupyterHub helm repo

```
$ helm repo add jupyterhub \
  https://jupyterhub.github.io/helm-chart/
```

```
"jupyterhub" has been added to your repositories
```

```
$ helm repo update
```

```
Hang tight while we grab the latest from your chart repositories...
```

```
...Skip local chart repository
```

```
...Successfully got an update from the "jupyterhub" chart repository
```

```
...Successfully got an update from the "stable" chart repository
```

```
Update Complete. ✨ Happy Helming!✨
```

Make a config.yaml file

```
$ openssl rand -hex 32
```

```
e9a377353783f5a92de2275a2649c71939eb1b4d1ed753  
6d5bce1035c51a03b4
```

```
$ openssl rand -hex 32
```

```
e387facef31c7d5bda9c65368853d979e3e44ce754b647  
cb4c9c54f2ea557b2e
```

Make a config.yaml file

```
$ nano config.yaml
```

```
hub:
```

```
    cookieSecret: <first-secret>
```

```
proxy:
```

```
    secretToken: <second-token>
```


Make a hub

```
$ helm install jupyterhub/jupyterhub \  
  --version=v0.4 \  
  --name=jupytercon \  
  --namespace=jupytercon \  
  -f config.yaml
```

More Kubernetes Objects

```
$ kubectl --namespace=jupytercon verbget objectdeployment
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
hub-deployment	1	1	1	1	2m
proxy-deployment	1	1	1	1	2m

Live demos follow!

You can follow instructions from
<https://zero-to-jupyterhub.readthedocs.io/en/latest/setup-jupyterhub.html#install-jupyterhub>