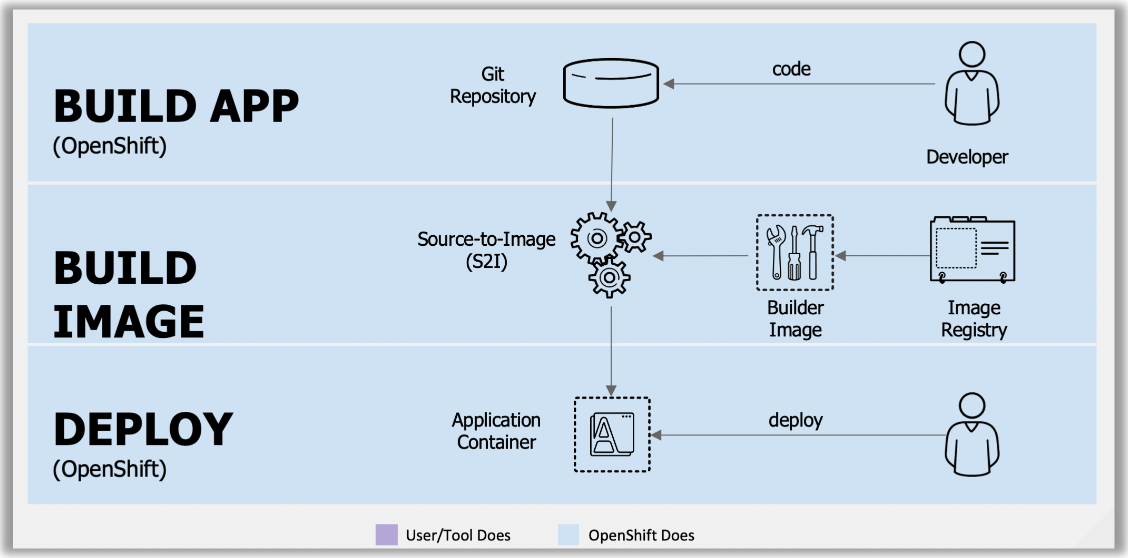
OPENSHIFT LABS

Simple Deployment Pipeline from GIT repo

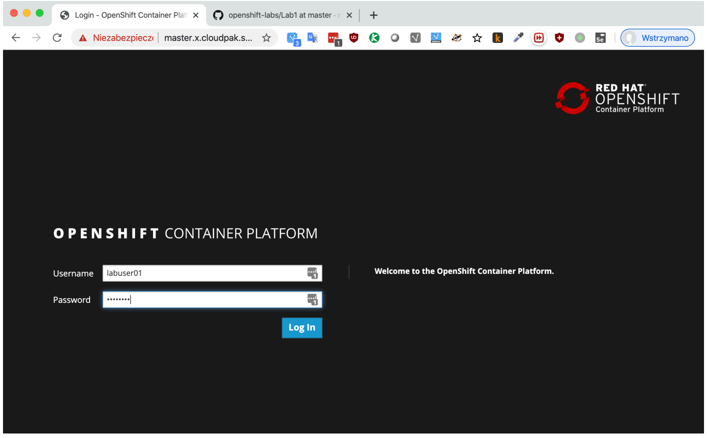
In this lab we will get familiar with the Source-To-Image mechanisms, that allows to easily deploy the application from Your code repository.



This lab has been created by Maciej Szulc (maciej.szulc@pl.ibm.com) for the Container, Kubernetes and Openshift workshop.

LOG IN TO THE CLUSTER AND ACCESS THE CATALOG

1. Open web browser and navigate to our cluser at https://master.x.cloudpak.site:8443
2. Login using credentials provided by IBM (labuserXX)



OPENSHIFT CONTAINER PLATFORM

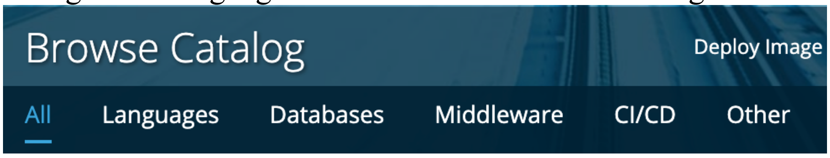
Username labuser01

Password ••••••••

Log In

Welcome to the OpenShift Container Platform.

3. Navigate to Languages tab shown on “Browse Catalog” screen

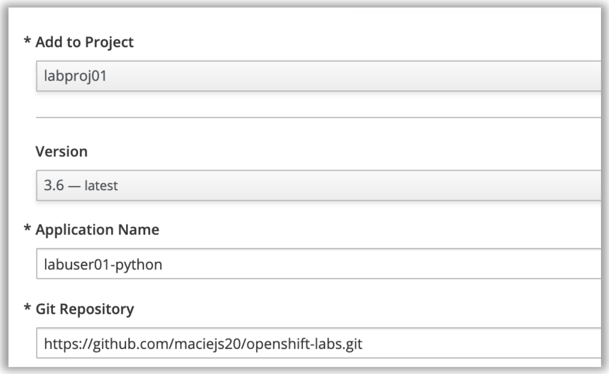


CREATE APPLICATION FROM PYTHON CODE

Now we will create a configuration that will get the code from github and create all required Kubernetes/Openshift elements.

4. Click on “**Python**” icon

5. Click on “**Python**” to start creation of runtime config. The system will present You with a creator.
6. Click “**Next**” on “Information” step
7. Fill the Application name as **<Your username>-python**, i.e.: labuser01-python.
8. Fill the Git repository as “**https://github.com/maciejs20/openshift-labs.git**”



* Add to Project

labproj01

Version

3.6 — latest

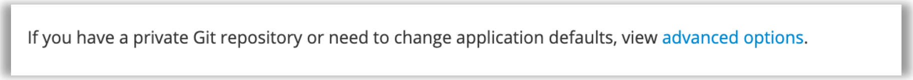
* Application Name

labuser01-python

* Git Repository

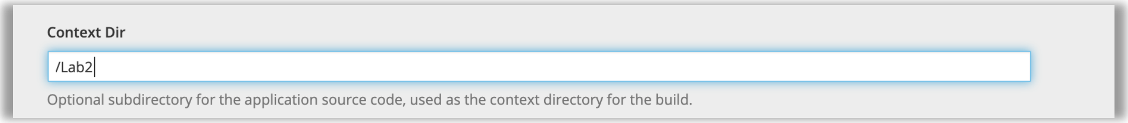
https://github.com/maciejs20/openshift-labs.git

9. Click on “Advanced options”



If you have a private Git repository or need to change application defaults, view advanced options.

10. Our code is not in the root directory – so we need to fill the context dir. Find “Context dir” and fill with “Lab2”

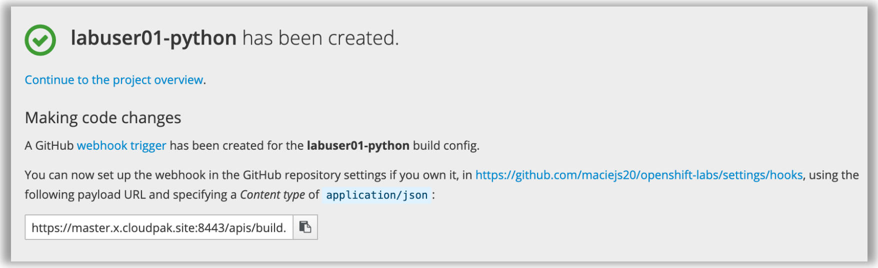


Context Dir

/Lab2

Optional subdirectory for the application source code, used as the context directory for the build.

11. Leave all remaining options intact and click “Create” in the bottom of the screen



labuser01-python has been created.

Continue to the project overview.

Making code changes

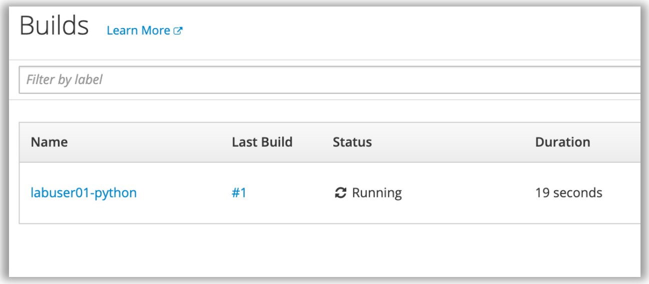
A GitHub webhook trigger has been created for the **labuser01-python** build config.

You can now set up the webhook in the GitHub repository settings if you own it, in https://github.com/maciejs20/openshift-labs/settings/hooks, using the following payload URL and specifying a *Content type* of `application/json` :

https://master.x.cloudpak.site:8443/apis/build.

VERIFY THE BUILD PROCESS

12. Navigate to “Builds” -> “Build” from the menu on the left side of the screen. Click on “labuser01-python to see the build details

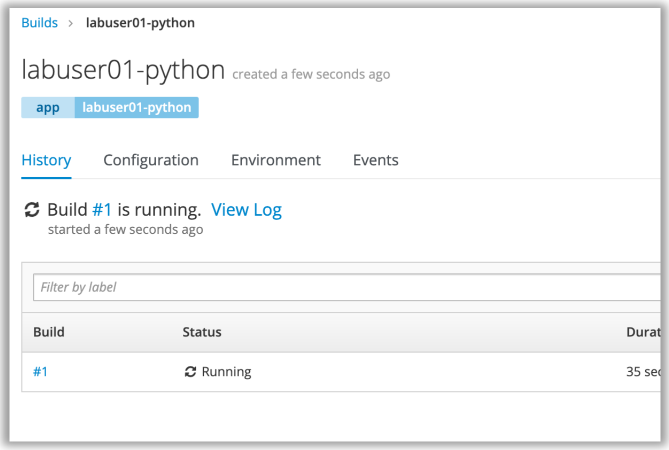


Builds Learn More

Filter by label

Name	Last Build	Status	Duration
labuser01-python	#1	Running	19 seconds

13. Click on “#1” to see current build details



Builds > labuser01-python

labuser01-python created a few seconds ago

app labuser01-python

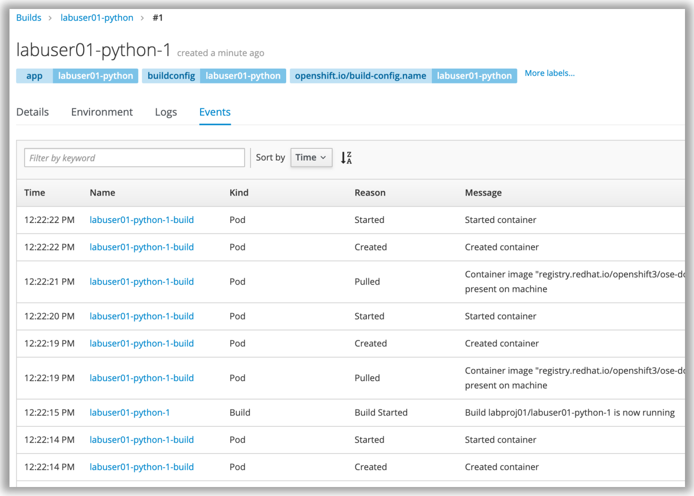
History Configuration Environment Events

Build #1 is running. View Log
started a few seconds ago

Filter by label

Build	Status	Durat
#1	Running	35 sec

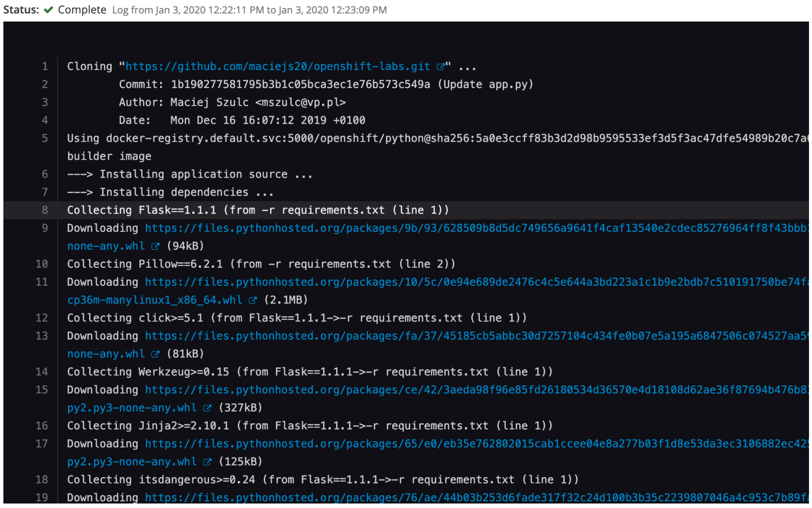
14. Navigate to “Logs”and “Events” to see the build process



Builds > labuser01-python > #1

labuser01-python-1 created a minute ago

Time	Name	Kind	Reason	Message
12:22:22 PM	labuser01-python-1-build	Pod	Started	Started container
12:22:22 PM	labuser01-python-1-build	Pod	Created	Created container
12:22:21 PM	labuser01-python-1-build	Pod	Pulled	Container image "registry.redhat.io/openshift3/ose-d present on machine
12:22:20 PM	labuser01-python-1-build	Pod	Started	Started container
12:22:19 PM	labuser01-python-1-build	Pod	Created	Created container
12:22:19 PM	labuser01-python-1-build	Pod	Pulled	Container image "registry.redhat.io/openshift3/ose-d present on machine
12:22:15 PM	labuser01-python-1	Build	Build Started	Build labproj01/labuser01-python-1 is now running
12:22:14 PM	labuser01-python-1-build	Pod	Started	Started container
12:22:14 PM	labuser01-python-1-build	Pod	Created	Created container



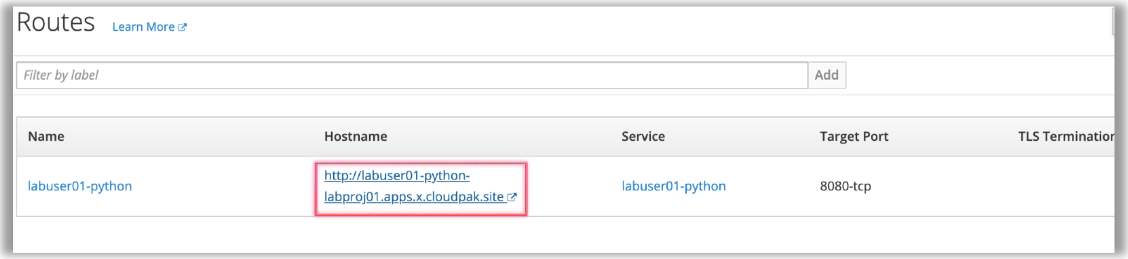
Status: ✓ Complete Log from Jan 3, 2020 12:22:11 PM to Jan 3, 2020 12:23:09 PM

```
Cloning "https://github.com/maciejs20/openshift-labs.git" ...
        Commit: 1b190277581795b3b1c05bca3ec1e76b573c549a (Update app.py)
        Author: Maciej Szulc <mszulc@vp.pl>
        Date:   Mon Dec 16 16:07:12 2019 +0100
Using docker-registry.default.svc:5000/openshift/python@sha256:5a0e3ccff83b3d2d98b9595533ef3d5f3ac47dfe54989b20c7a
builder image
---> Installing application source ...
---> Installing dependencies ...
Collecting Flask==1.1.1 (from -r requirements.txt (line 1))
Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb
none-any.whl (94kB)
Collecting Pillow==6.2.1 (from -r requirements.txt (line 2))
Downloading https://files.pythonhosted.org/packages/10/5c/0e94e689de2476c4c5e644a3bd223a1c1b9e2bdb7c510191750be74f
cp36m-manylinux1_x86_64.whl (2.1MB)
Collecting click>=5.1 (from Flask==1.1.1->-r requirements.txt (line 1))
Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a6847506c074527aa5
none-any.whl (81kB)
Collecting Werkzeug>=0.15 (from Flask==1.1.1->-r requirements.txt (line 1))
Downloading https://files.pythonhosted.org/packages/ce/42/3aeda98f96e85fd26180534d36570e4d18108d62ae36f87694b476b8
py2.py3-none-any.whl (327kB)
Collecting Jinja2>=2.10.1 (from Flask==1.1.1->-r requirements.txt (line 1))
Downloading https://files.pythonhosted.org/packages/65/e0/eb35e762802015cab1ccee04e8a277b03f1d8e53da3ec3106882ec42
py2.py3-none-any.whl (125kB)
Collecting itsdangerous>=0.24 (from Flask==1.1.1->-r requirements.txt (line 1))
Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89f
```

Wait for the app to build.

VERIFY IF APP WORKS

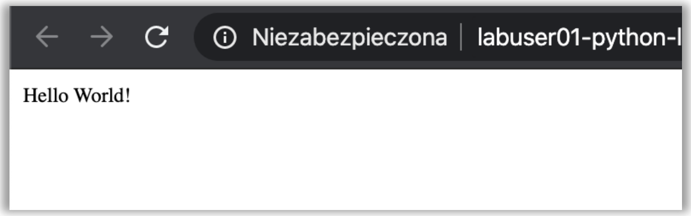
15. Use the ☰ icon to navigate to "**Applications**" -> **Routes**
16. Now we have to verify if Your app works.
    Click on Your route (will be named <Your_name>-python-<Your project>.apps.x.cloudpak.site)



Routes Learn More

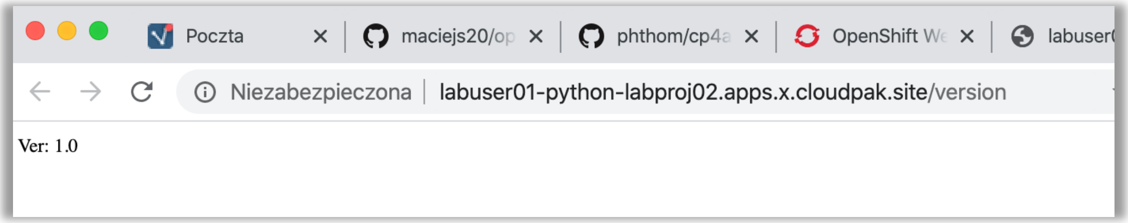
Name	Hostname	Service	Target Port	TLS Termination
labuser01-python	http://labuser01-python-labproj01.apps.x.cloudpak.site	labuser01-python	8080-tcp	

17. Verify if there is a "Hello world" message from app deployed.



```
Hello World!
```

18. Check the code version adding /version to the URL bar


Poczta
maciejs20/op
phthom/cp4a
OpenShift We
labuser
Niezabezpieczona | labuser01-python-labproj02.apps.x.cloudpak.site/version

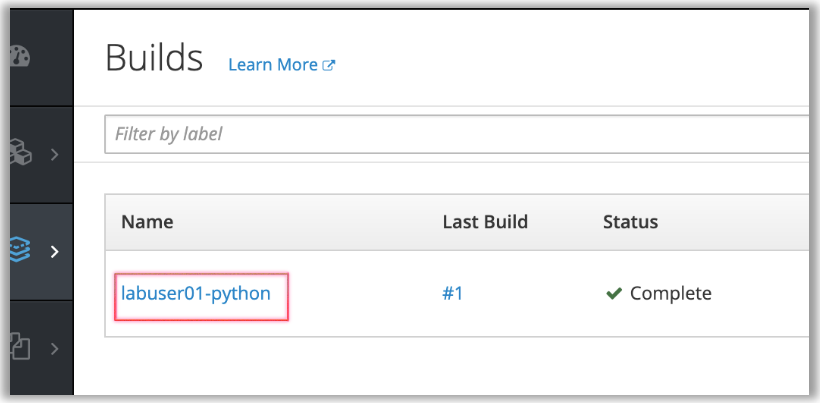
```
Ver: 1.0
```

CHANGE APPLICATION CODE

When we are aligning with CI/CD process, it is necessary to use automated build. Oopenshift S2I can re-build the application with any code change in git repo. For our lab we are using shared code repository that has no access to our cluster, that's why we can't use this mechanism directly.

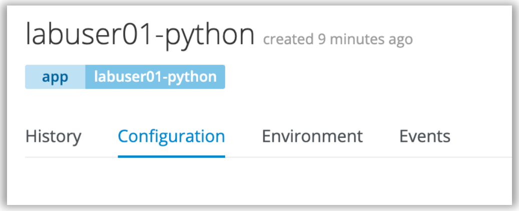
We will simply switch the context dir to simulate the change instead.

19. Use the ☰ icon to navigate to "**Builds**" -> **"Builds".** Click on Your build (labuserXX-python)

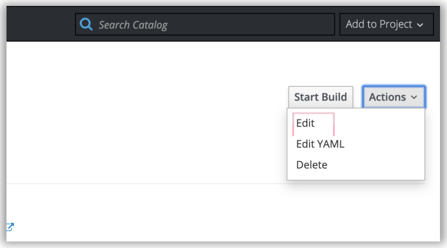


Name	Last Build	Status
labuser01-python	#1	✓ Complete

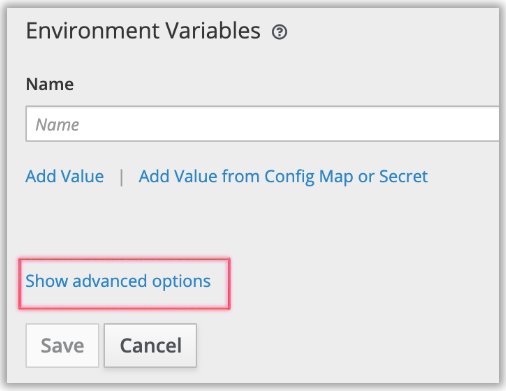
20. Click on **"Configuration"**



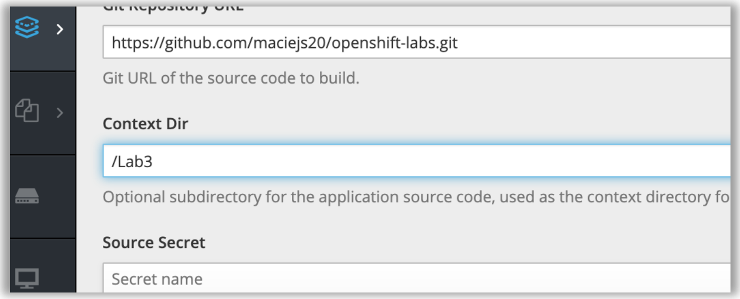
21. From the **"Actions"** dropdown select "**Edit**"



22. Click on “**Show advanced options”** at the bottom of the screen



23. Change the “**Context dir”** to **Lab3**



Git Repository URL: https://github.com/maciejs20/openshift-labs.git

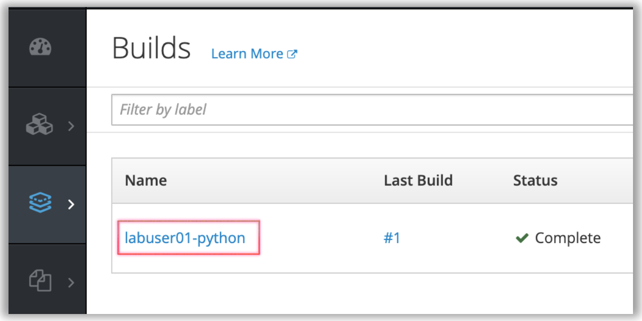
Git URL of the source code to build.

Context Dir: /Lab3

Optional subdirectory for the application source code, used as the context directory fo

Source Secret: Secret name

24. Click on **“Save”** on the bottom of the screen.
25. Use the ☰ icon to navigate to “**Builds**” -> **“Builds”.** Click on Your build.



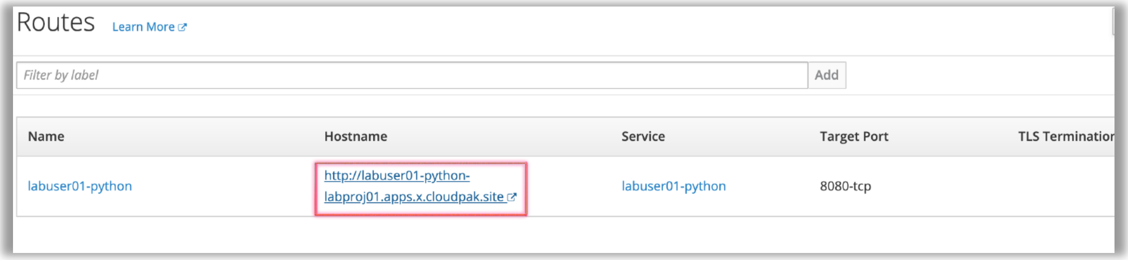
Name	Last Build	Status
labuser01-python	#1	✓ Complete

26. In a production environment the code repo calls the Openshift to automatically rebuild the application on code change. As we can’t use such mechanism in our lab environment, we have to re-build the app manually.

    Click on the Start Build Start Build button and wait for the build to complete.

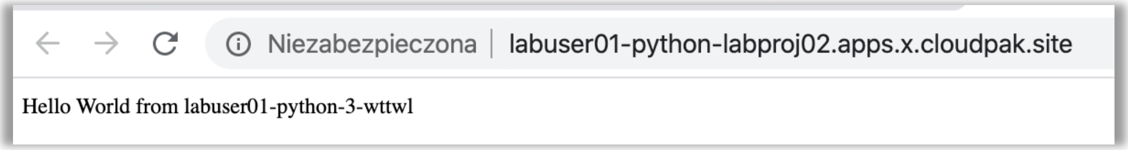
VERIFY IF NEW APP WORKS

27. Use the ☰ icon to navigate to “**Applications**” -> **Routes**
28. Now we have to verify if Your app works. Click on Your route (will be named <Your_name>-python-<Your project>.apps.x.cloudpak.site)



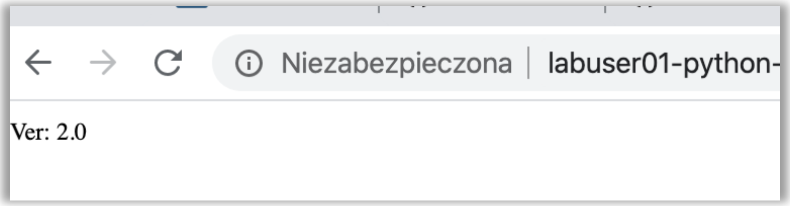
Name	Hostname	Service	Target Port	TLS Terminatior
labuser01-python	http://labuser01-python-labproj01.apps.x.cloudpak.site	labuser01-python	8080-tcp	

29. Verify if there is a “Hello world from <hostname>” message from app deployed. This version of the lab displays the hostname of the pod it in running on.



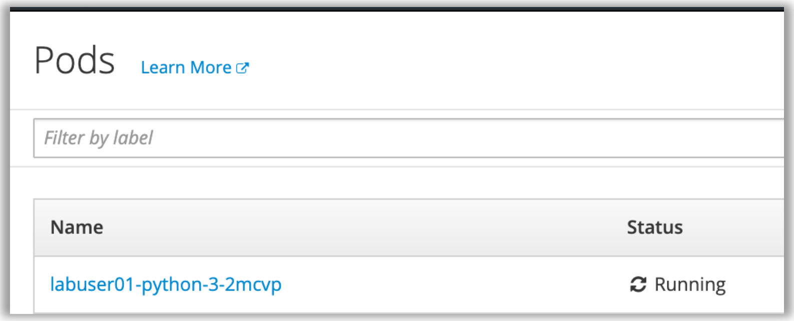
Hello World from labuser01-python-3-wttwl

30. Check the code version adding /version to the URL bar



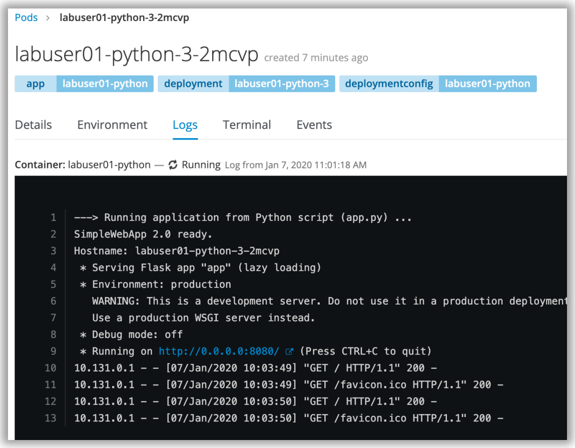
Ver: 2.0

31. Openshift allows to interact with Your pods easily. Use the ☰ icon to navigate to “**Applications**” -> **“Pods”.** Click on one ot the running pods for Your app.



Name	Status
labuser01-python-3-2mcvp	Running

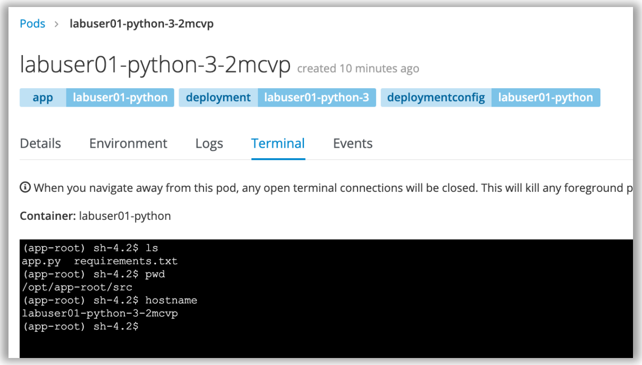
32. Go to “**Logs**” tab to see the logs from Your application.


Pods > labuser01-python-3-2mcvp
labuser01-python-3-2mcvp created 7 minutes ago
app labuser01-python
deployment labuser01-python-3
deploymentconfig labuser01-python
Details
Environment
Logs
Terminal
Events

Container: labuser01-python — Running Log from Jan 7, 2020 11:01:18 AM

```
---> Running application from Python script (app.py) ...
SimpleWebApp 2.0 ready.
Hostname: labuser01-python-3-2mcvp
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
10.131.0.1 - - [07/Jan/2020 10:03:49] "GET / HTTP/1.1" 200 -
10.131.0.1 - - [07/Jan/2020 10:03:49] "GET /favicon.ico HTTP/1.1" 200 -
10.131.0.1 - - [07/Jan/2020 10:03:50] "GET / HTTP/1.1" 200 -
10.131.0.1 - - [07/Jan/2020 10:03:50] "GET /favicon.ico HTTP/1.1" 200 -
```

33. Click on “**Terminal**” to connect to command line session inside the application pod.
Enter “ls -l” to see the directory content in the pod.
Enter “hostname” to see the pod hostname


Pods > labuser01-python-3-2mcvp
labuser01-python-3-2mcvp created 10 minutes ago
app labuser01-python
deployment labuser01-python-3
deploymentconfig labuser01-python
Details
Environment
Logs
Terminal
Events

ⓘ When you navigate away from this pod, any open terminal connections will be closed. This will kill any foreground p

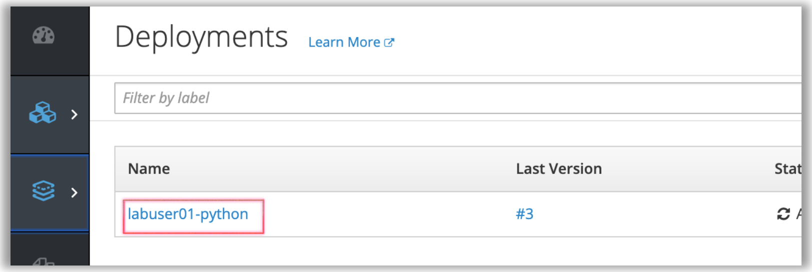
Container: labuser01-python

```
(app-root) sh-4.2$ ls
app.py  requirements.txt
(app-root) sh-4.2$ pwd
/opt/app-root/src
(app-root) sh-4.2$ hostname
labuser01-python-3-2mcvp
(app-root) sh-4.2$
```

USE AUTOSCALLER

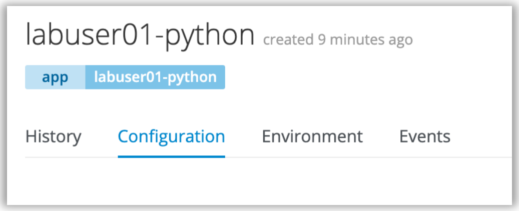
The build mechanisms in the Openshift allows to configure many things that should be configured manually in Kubernetes.
In the previous lab we have been using the GUI to manually increase and decrease the number of pods for the application. Now we will add an autoscaller to scale the number of pods to the resource consumption.

34. Use the ☰ icon to navigate to “**Applications**” -> **Deployments.** Click on Your deployment

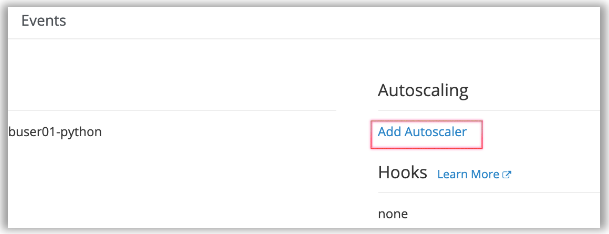


Name	Last Version	Stat
labuser01-python	#3	

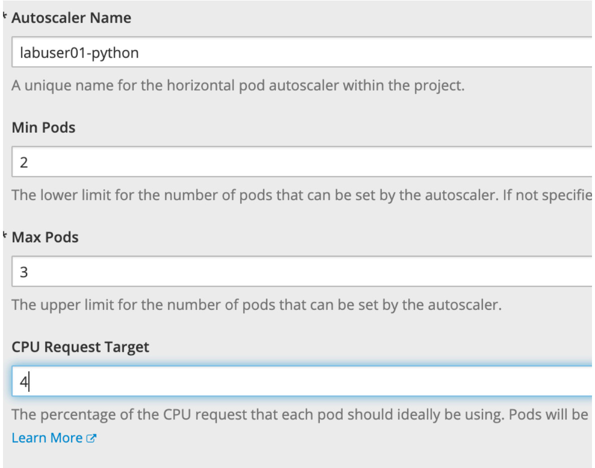
35. Click on **“Configuration” tab**



36. Click on **“Add autoscaller”**



37. Set the “Min Pods” to 2, “Max Pods” to 3 and “CPU Request Target” to 4. Leave other options intact.



* Autoscaler Name

labuser01-python

A unique name for the horizontal pod autoscaler within the project.

Min Pods

2

The lower limit for the number of pods that can be set by the autoscaler. If not specifie

* Max Pods

3

The upper limit for the number of pods that can be set by the autoscaler.

CPU Request Target

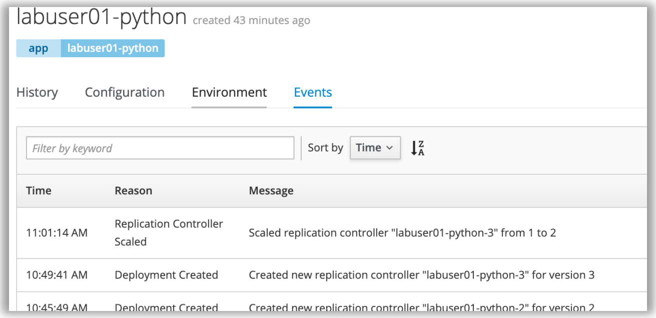
4

The percentage of the CPU request that each pod should ideally be using. Pods will be

Learn More

38. Click on “Save”

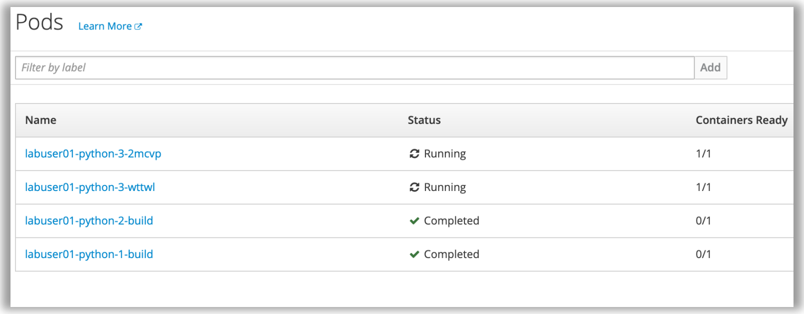
39. Click on “Events” to see autoscaller in work. It may take a minute for autoscaller to add the app instance.



Time	Reason	Message
11:01:14 AM	Replication Controller Scaled	Scaled replication controller "labuser01-python-3" from 1 to 2
10:49:41 AM	Deployment Created	Created new replication controller "labuser01-python-3" for version 3
10:45:49 AM	Deployment Created	Created new replication controller "labuser01-python-2" for version 2

40. Use the ☰ icon to navigate to “Applications” -> “Pods” to see the number of pods

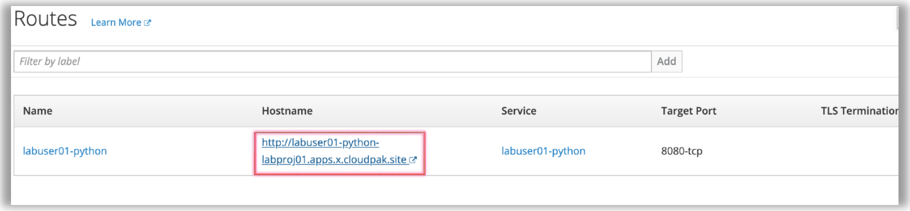
There should be 2 pods running.



Name	Status	Containers Ready
labuser01-python-3-2mcvp	Running	1/1
labuser01-python-3-wttwl	Running	1/1
labuser01-python-2-build	✓ Completed	0/1
labuser01-python-1-build	✓ Completed	0/1

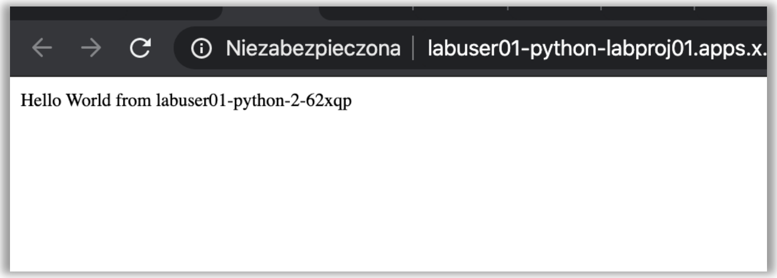
VERIFY IF SCALLED APP WORKS

41. Use the ☰ icon to navigate to “Applications” -> Routes
42. Now we have to verify if Your app works.
    Click on Your route (will be named <Your_name>-python-<Your project>.apps.x.cloudpak.site)



Name	Hostname	Service	Target Port	TLS Terminatio
labuser01-python	http://labuser01-python-labproj01.apps.x.cloudpak.site	labuser01-python	8080-tcp	

43. Verify if there is a “Hello world” message from app deployed.



```
Hello World from labuser01-python-2-62xqp
```

We have two pods running but usually You will see only single hostname in the response. Default route handling policy sticks the session (together with pod) to a cookie, so each time You are presented with data from the same pod.

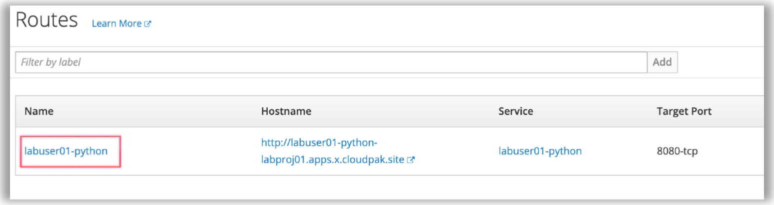
You may change this behavior with the lab below **or just clear the cookies and refresh the page.**

CHANGE ROUTE ANNOTATIONS (OPTIONAL)

Openshift routes are using Annotations to configure specific settings. All the details are described in the manual. We will use two annotations to change the route behavior:

- haproxy.router.openshift.io/balance – to set the policy
- haproxy.router.openshift.io/disable_cookies – to disable 'per-cookie' pod selection

44. Use the ☰ icon to navigate to "Applications" -> Routes
45. Click on the route config to open route config

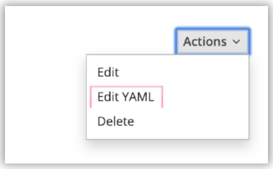


Routes Learn More

Filter by label Add

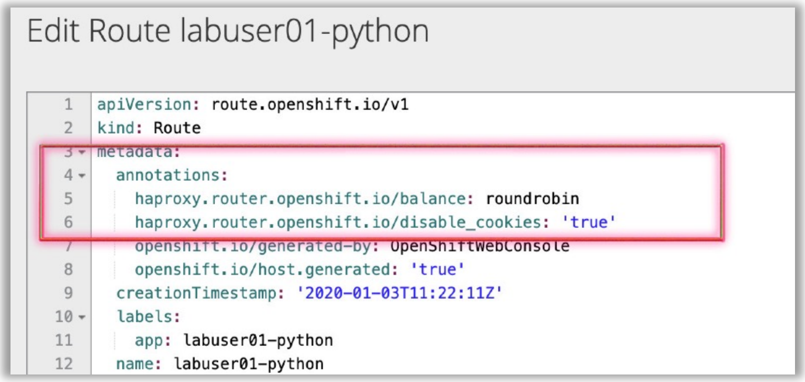
Name	Hostname	Service	Target Port
labuser01-python	http://labuser01-python-labproj01.apps.x.cloudpak.site	labuser01-python	8080-tcp

46. Click on "Actions" -> "Edit YAML"



47. Add following settings to "annotation" setting. Double check that - Yaml depends on the number of spaces – so it has to be like on the picture below!

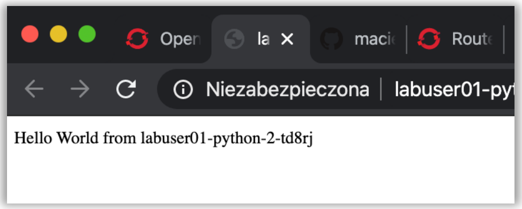
haproxy.router.openshift.io/balance: roundrobin
haproxy.router.openshift.io/disable_cookies: 'true'



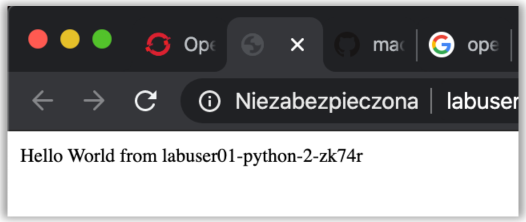
Edit Route labuser01-python

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  annotations:
    haproxy.router.openshift.io/balance: roundrobin
    haproxy.router.openshift.io/disable_cookies: 'true'
    openshift.io/generated-by: OpenShiftWebConsole
    openshift.io/host.generated: 'true'
  creationTimestamp: '2020-01-03T11:22:11Z'
  labels:
    app: labuser01-python
  name: labuser01-python
```

48. Wait for 1-2 minutes and verify if the host name changes.
It may require several page refreshes to see the host changed.



Hello World from labuser01-python-2-td8rj



Hello World from labuser01-python-2-zk74r