Class 1

- First of all DevOps is not a title/job/profession/set of tools or software, DevOps is a culture!
- A culture where Dev and Ops teammates cooperate with each other.
- DevOps culture is talking about:
  - Engineers empowerment - by giving engineers more responsibility over the whole application lifecycle process. (dev -> test -> deploy -> monitor)
  - Test Driven Development - write tests before you write code. Unit tests, integration tests, system tests. This will help increase the quality of your service and give you more confidence to release faster and more frequent.
  - Automation - automate everything that can be automated. Test automation, infrastructure automation, deployment automation, etc..
  - Monitoring - monitor your apps, build monitoring alerts well. It should save your time, don't flood with metrics and alerts.
  - Self service - provide a self service for any framework that you build or anything that you do. Don't be a bottleneck.

Benefits:

- Speed
  - Ensure faster time-to-market/delivery times that improves ROI.
  - DevOps is basically the application of Agile principles, and thus the end result is faster development of software, ensuring more frequent delivery.
- Reliability
  - Early detection and faster correction of defects that helps provide the best services and robust features that must be delivered to customers.
- Cultural
  - Happier, more productive teams
  - Higher employee engagement
  - Improved communication and collaboration

Methodologies

Waterfall:

- Gather and document requirements
- Design
- Code and unit test

- Perform system testing
- Perform user acceptance testing
- Fix any issues
- Deliver the finished product
- The biggest problem with this approach is that the customer's needs usually change during the development phase which causes delivering software that doesn't meet the revised needs, or you spend a lot of time and money in changing plans midway.

Agile:

- The idea in Agile is to develop software in small iterations (known as iterations) and be able to adapt to the changing customer needs better than in Waterfall.
- However, some of this model disadvantages are:
- Budget goals and deadlines are often missed.
- New features break old functions.

Agile & DevOps

- DevOps brings more flexibility on top of the Agile model.
- With continuous integration (CI) and continuous delivery (CD) pipelines, you can make sure that you can release often and the releases actually work and meet the customer needs.
- Cooperation between the development team and IT operations ensures that also the used tools are streamlined and do not form bottlenecks.
- With the effective tools, repetitive work can be automated and transparency is improved.

CI

- Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day.
- Each integration can then be verified by an automated build and automated tests.
- One of the key benefits of integrating regularly is that you can detect errors quickly and locate them more easily.
- As each change introduced is typically small, pinpointing the specific change that introduced a defect can be done quickly.
- In recent years CI has become a best practice for software development and is guided by a set of key principles.
- Among them are revision control, build automation and automated testing.
- Some popular tools used in CI which will be discussed later are:

- ○ Jenkins
- ○ Travis CI
- ○ Bamboo

CD (Continuous Delivery)

- Continuous delivery is an extension of continuous integration to make sure that you can release new changes to your customers quickly in a sustainable way.
- This means that on top of having automated your testing, you also have automated your release process and you can deploy your application at any point of time by clicking on a button.
- In theory, with continuous delivery, you can decide to release daily, weekly, or whatever suits your business requirements.
- However, if you truly want to get the benefits of continuous delivery, you should deploy to production as early as possible to make sure that you release small batches, that are easy to troubleshoot in case of a problem.

What do we deliver?

When we approach to the implementation of a CI/CD process we encounter the need to have one, through the entire pipeline, artifact that is being tested over and over again, attached with an indicating version to allow us to understand what code version used for that artifact. An artifact is usually built with a package manager and sent to an **artifact repository**. We use artifact repository in order to have one, centralized, accessible from every place we need service that allows us to send/get the artifact we need to build/test/deploy.

How do we deliver it?

When we want to deliver a new artifact we need to take two steps, deploy the artifact and roll out the new version. There are many cases and options to this and this can be varied between different platforms. Let's take a small example, If my artifact would be a java WAR (**W**eb application **AR**chive) and we would like to roll out a new WAR we just finished building and testing we will need to do the following:

1. Deploy the new WAR to the application servers
2. Restart in a rolling fashion our application servers

The CICD Pipeline

1. Plan - in agile methodology before development there is the planing stage, when each team plan the work of the next iteration.
2. Code - Write the code and it's test that in the next steps will be delivered.
3. Build - Building our artifact from the newer version of the code we wrote in the latest iteration.
4. Test - Run any test that is applicable for the application. Can be unit, integration, system, end-to-end, sanity, stress, load tests. **Everything** must be able to run automatically.
5. Release - In this stage we will finalize our artifact and send it to the repository.
6. Deploy - roll out the newer version of the artifact we created
7. Operate - Maintain and manage the day to day existence of our product (Scale, fix, response to incidents)
8. Monitor - Always have all the right metrics and alerts set in place in order to have confidence in the system that you are developing
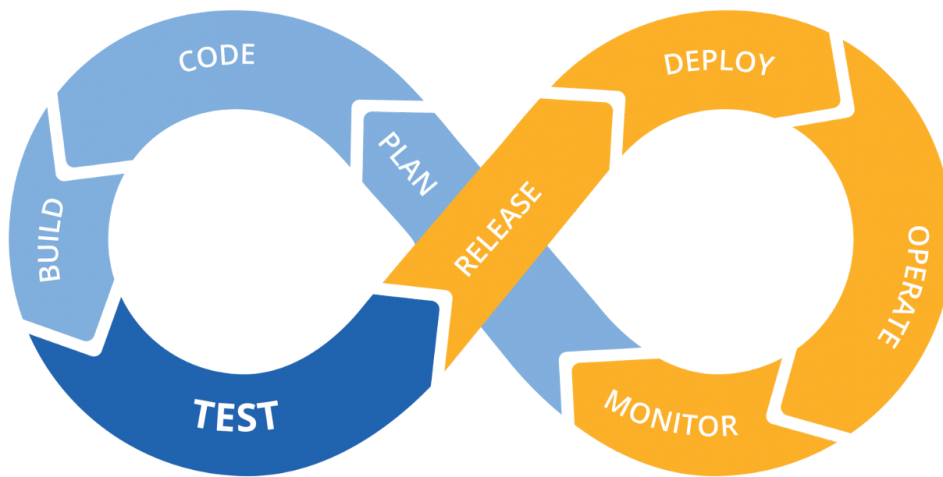
CD (Continuous Deployment)

● Continuous deployment goes one step further than continuous delivery.
● With this practice, every change that passes all stages of your production pipeline is released to your customers.
● There's no human intervention, and only a failed test will prevent a new change to be deployed to production.
● Continuous deployment is an excellent way to accelerate the feedback loop with your customers and take pressure off the team as there isn't a Release Day anymore.
● Developers can focus on building software, and they see their work go live minutes after they've finished working on it.

Python Introduction

Python is a clear and powerful object-oriented multi-purpose programming language. Some features of Python are:

A. Python supports object-oriented programming with classes and multiple inheritance.
B. Code can be grouped into modules and packages.
C. Python's automatic memory management frees you from having to manually allocate and free memory in your code.
D. Runs anywhere, including Mac OS X, Windows, Linux, and Unix, with unofficial builds also available for Android and iOS.
E. It is free open-source language

Python Interpreter

- Python is an interpreted language which means that our code works as such:
  - Instructions (code) are read
  - Code evaluated
  - Result is returned for each instruction in a sequence.
- And this is the interpreter job.
- Unlike compiled languages, in which whole context is passed through a compiler, which in return gives a translational unit.
- The Standard interpreter for Python is written in C.
- There is also a version of Python written in Python, which instead of Python bytecode it generates Python byte code (Jython).
- One of the advanteages is the increased productivity it provides, since there is no compilation step unlike many other languages.

Python Data Types

Type	Example
Numeric	a = 1, a =1.0
String	a = 'DevOps'
Boolean	isA = True, isA = False
List	arr = ['a','b',123]
Tuple	t = ('a','b')
Dictionary	Dict = {'name': 'Aviel','age': 28}

Numbers

int – whole number: **X=2**

float – a floating point number: **y = 2.**

long – are large whole numbers (happens automatically) when a number is big.

Boolean – Can hold one value out of the following two values: True/False **isHere = True**

String – A string is a sequence of characters, which is marked in a single/ double quotes: **Name = "John"**

 Data structures - List, Tuple, Dictionary will be discussed later.

https://en.wikibooks.org/wiki/Python_Programming/Variables_and_Strings

Simple Assignment Operator

= Simple assignment operator

Arithmetic Operators

+ Additive operator (also used for String concatenation)

- Subtraction operator

* Multiplication operator

/ Division operator

% Remainder operator

Equality and Relational Operators

== Equal to

!= Not equal to

> Greater than

>= Greater than or equal to

< Less than

<= Less than or equal to

Python installation

Enter Python download page:
https://www.python.org/downloads/

For other OS types click on

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Validate Python is installed by doing the following steps:
1. Open CMD / Terminal.
2. Type python --version
3. Python version will appear at command line.

PyCharm Setup

Go to https://www.jetbrains.com/pycharm/download and choose
"Community edition"

Download PyCharm

Professional

Full-featured IDE
for Python & Web
development

DOWNLOAD

Free trial

Community

Lightweight IDE
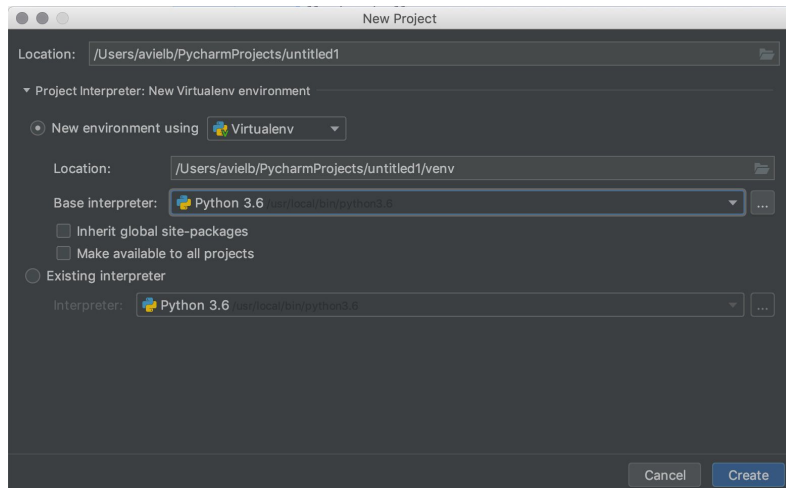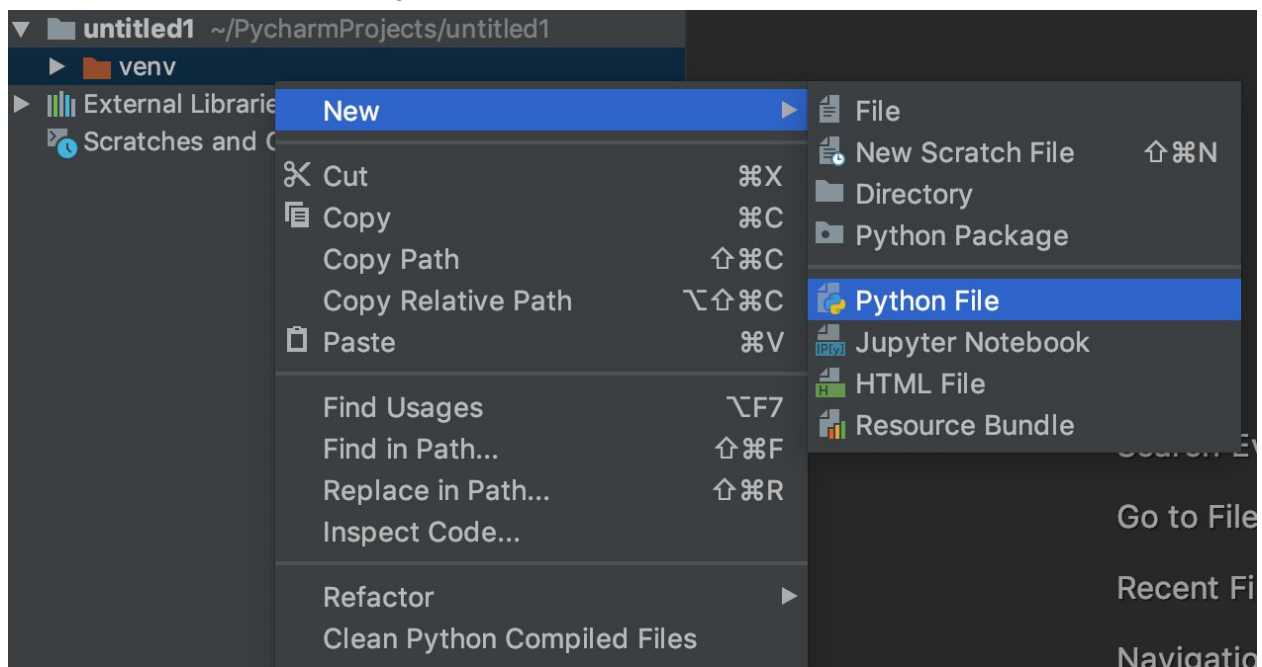for Python & Scientific
development

DOWNLOAD

Free, open-source

Keep following the installation with defaults settings
1. Next you will be asked to choose the Python interpreter
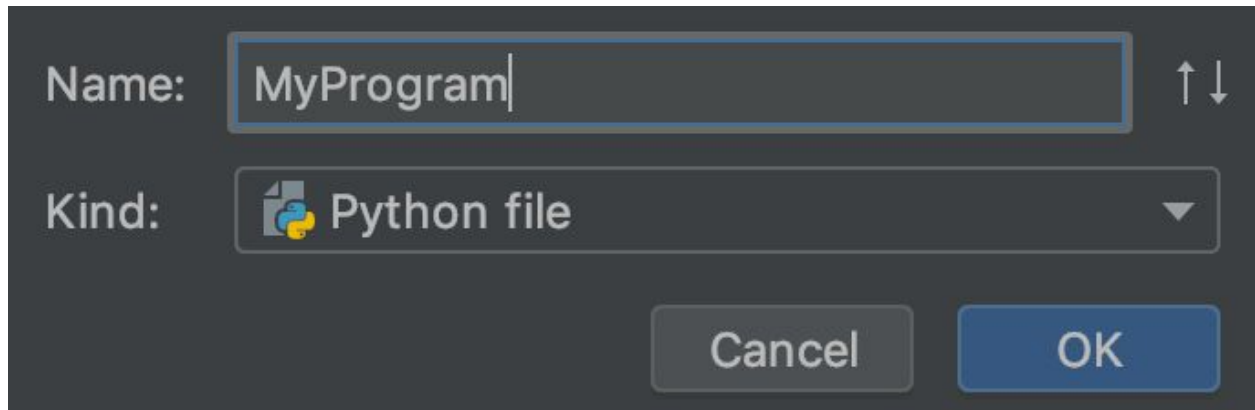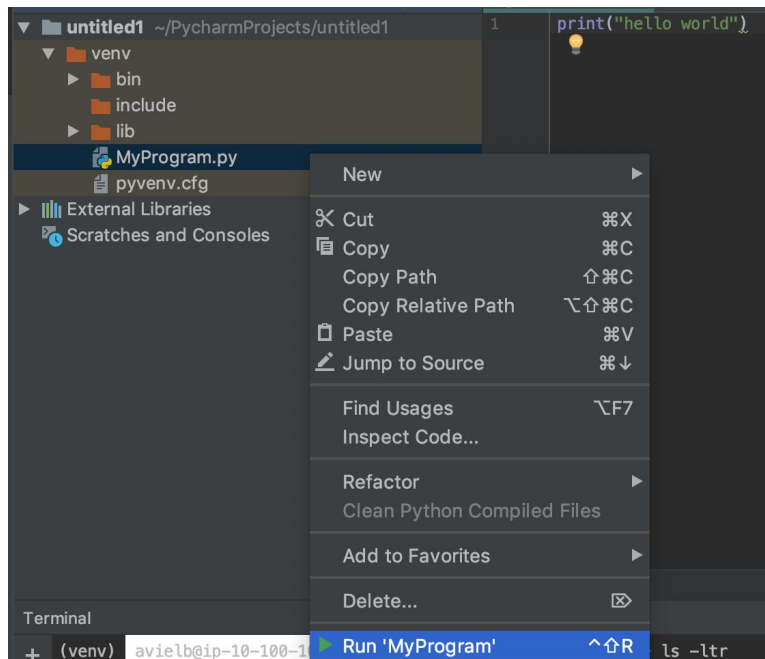2. Choose create new project:

Now that we have our environment set up and ready for use. Lets create a new python file and code our first "hello world" program.

And call the file: "MyProgram" and hit "OK"

In the file created, type the following: print("hello world") and run it as follows:

Check the output for the program in the screen bellow: