

Experiment 03: Working with Kind in an Air-Gapped Environment

Create Air Gapped Docker Registry

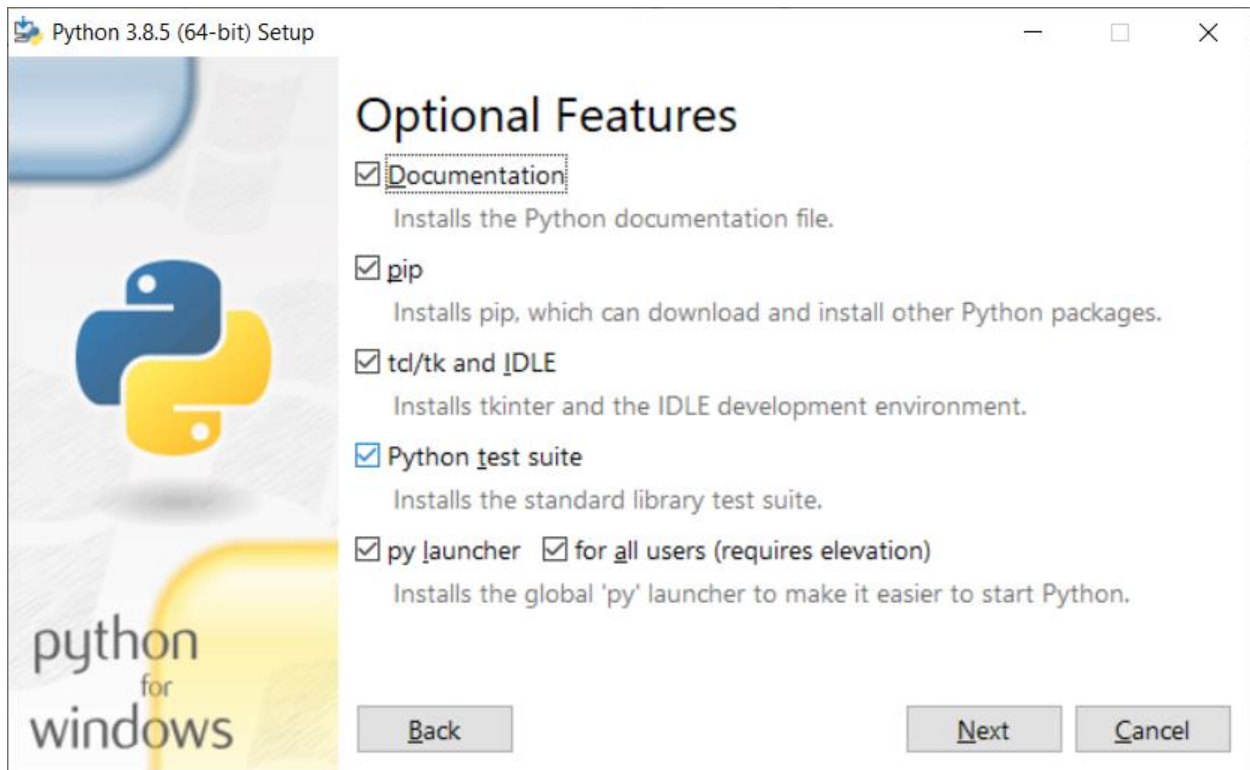
<https://www.python.org/downloads/>

<https://www.python.org/downloads/release/python-385/>

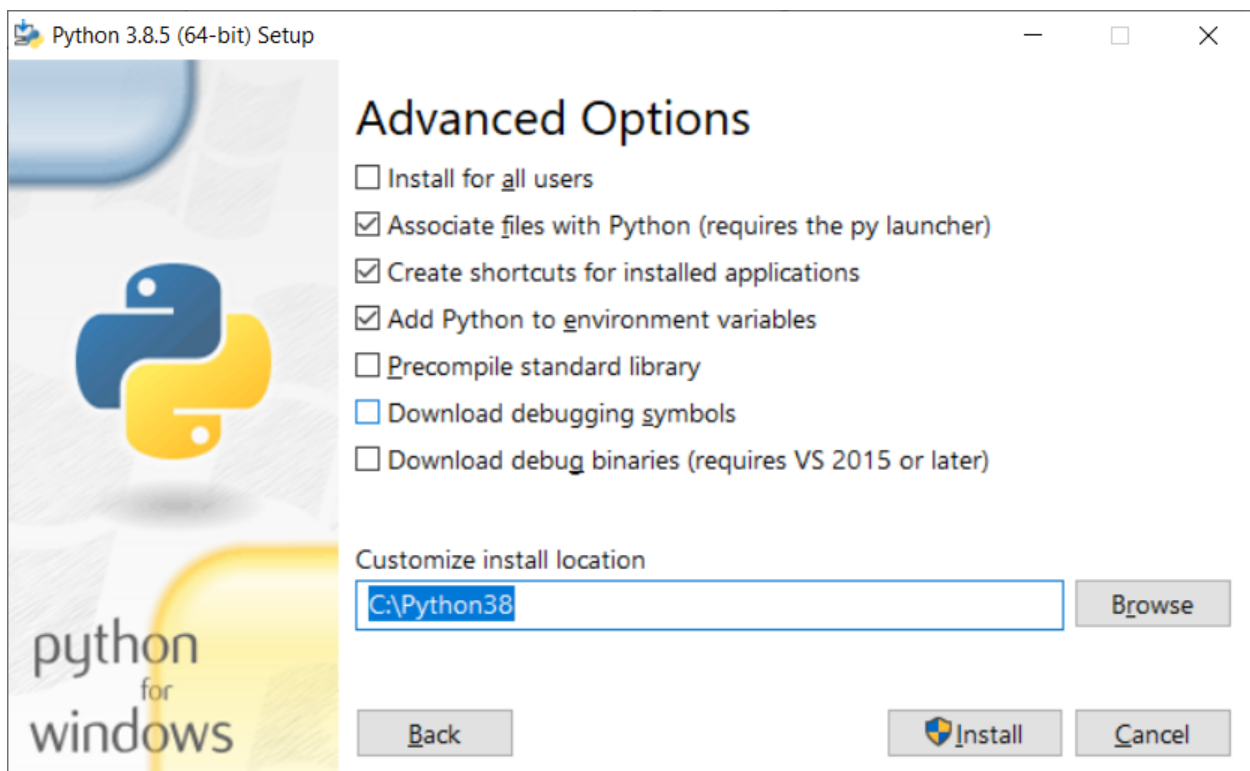
To use the Docker Standalone registry we install Python 3.X, in this case the current stable release is 3.8.5 and has required Security fixes that should make it the standard 3.X version for those reasons.



Check add Python to the PATH, and Select Customize Installation



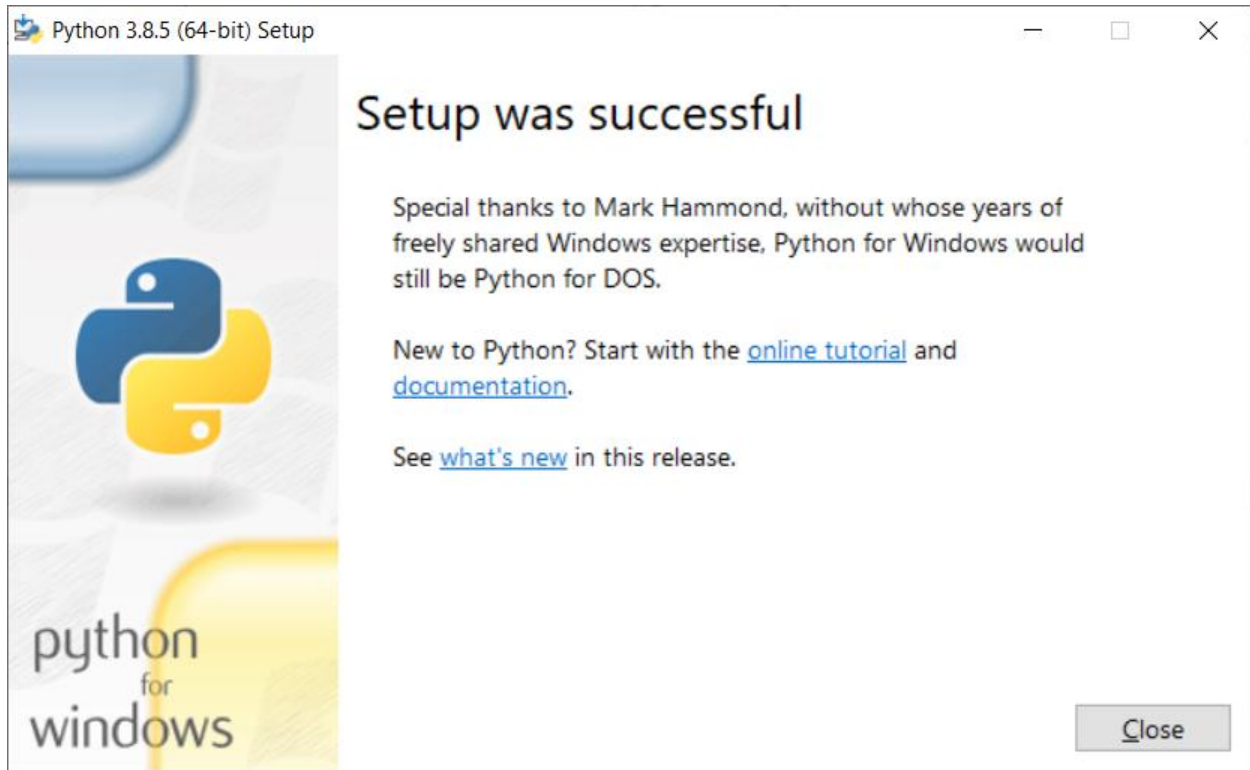
We can leave the defaults and select “Next”



Change the default location to something that isn't a mile long and select “Install”

Click the button to change the MAX_PATH length limitation of 260 characters

Think about how much work Mark Hammond must have done to turn Python into Windows from DOS



Select “Close”

From our project folder clone the standalone registry repository from github.com

```
C:\projects\kind> git clone https://github.com/dotcloud/docker-registry.git
```

Change to the folder

```
C:\projects\kind> cd docker-registry\config
```

Copy the sample YAML configuration file

```
C:\projects\kind\docker-registry\config> copy config_sample.yml config.yml
```

Change directory to the parent for the cloned repo and start the registry

```
C:\projects\kind\docker-registry\config> cd ..
```













```
C:\projects\kind\docker-registry> docker run -p 5000:5000 registry
```

```
C:\> cd c:\projects\kind
```

Create your cluster, which if this was truly airgapped would require you to have already tar'd or other pulled the images for kind, i.e. kindest/node:v1.18.2 or similar and the kind/server-kind/agent

```
C:\projects\kind> kind create cluster --name air-gap-kind
```

Creating cluster "air-gap-kind" ...

- Ensuring node image (kindest/node:v1.18.2)  ...
- ✓ Ensuring node image (kindest/node:v1.18.2) 
- Preparing nodes  ...
- ✓ Preparing nodes 
- Writing configuration  ...
- ✓ Writing configuration 
- Starting control-plane  ...
- ✓ Starting control-plane 
- Installing CNI  ...
- ✓ Installing CNI 
- Installing StorageClass  ...
- ✓ Installing StorageClass 

Set kubectl context to "kind-air-gap-kind"

You can now use your cluster with:

```
kubectl cluster-info --context kind-air-gap-kind
```

Thanks for using kind! 😊

Now we'll pull an image for Ubuntu as an example, in the real world for an Air Gapped environment you'd be using a jump box or some other type of internal connectivity to copy the images to use over to the environment since there is not Internet connectivity. For the grep command to work in the below stream we would be referencing the grep command from the Git installation.

```
C:\projects> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
54ee1f796a1e: Pull complete
f7bfea53ad12: Pull complete
46d371e02073: Pull complete
b66c17bbf772: Pull complete
Digest: sha256:31dfb10d52ce76c5ca0aa19d10b3e6424b830729e32a89a7c6eee2cda2be67a5
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

C:\projects> docker images | grep ubuntu | grep latest
ubuntu          latest          4e2eef94cd6b    2 weeks ago    73.9MB

C:\projects> docker tag 4e2eef94cd6b localhost:5000/ubuntu

C:\projects> docker push localhost:5000/ubuntu
The push refers to repository [localhost:5000/ubuntu]
a4399aeb9a0e: Pushed
35a91a75d24b: Pushed
ad44aa179b33: Pushed
2ce3c188c38d: Pushed
Digest: sha256:6f2fb2f9fb5582f8b587837afd6ea8f37d8d1d9e41168c90f410a6ef15fa8ce5 size: 1152
latest:

C:\projects> docker pull localhost:5000/ubuntu
Using default tag: latest
latest: Pulling from ubuntu
Digest: sha256:6f2fb2f9fb5582f8b587837afd6ea8f37d8d1d9e41168c90f410a6ef15fa8ce5
Status: Image is up to date for localhost:5000/ubuntu:latest
localhost:5000/ubuntu:latest
```

```
C:\projects\kind\k8s-wp> docker pull ubuntu:xenial-20200807
```

```
C:\projects\kind\k8s-wp> docker tag 4b22027ede29
localhost:5000/ubuntu:xenial-20200807
```

```
C:\projects\kind\kind-wp> kind load docker-image
localhost:5000/ubuntu:xenial-20200807
Image: "ubuntu:xenial-20200807" with ID
"sha256:4e2eef94cd6b93dd4d794c18b45c763f72edc22858e0da5b6e63a4566a54c03c"
not yet present on node "kind-control-plane", loading...
```

To view the images in KIND we need to use the cri-tools built in to our containerd node images.

```
C:\projects\kind\kind-wp> kind load docker-image localhost:5000/ubuntu
Image: "ubuntu:latest " with ID
"sha256:4b22027ede299794c18b45c763f72edc22858e0da5b6e63a4566a54def43423a"
not yet present on node "kind-control-plane", loading...
```

To view the images in KIND we need to use the cri-tools built in to our containerd node images.

```
C:\Users\wcsadmin\.crictl> kind get nodes
kind-control-plane
```

```
C:\Users\wcsadmin\.crictl> docker exec -it kind-control-plane crictl
images
```

IMAGE	TAG	IMAGE ID
SIZE		
docker.io/fabxc/instrumented_app	latest	da9b08dfc8399
254MB		
docker.io/kindest/kindnetd	0.1.0	f227066bdc5f9
35.6MB		
docker.io/library/ubuntu	latest	4e2eef94cd6b9
76.3MB		
docker.io/library/ubuntu	xenial-20200807	4b22027ede299
131MB		
k8s.gcr.io/coredns	1.3.1	eb516548c180f
40.5MB		
k8s.gcr.io/etcd	3.3.10	2c4adeb21b4ff
258MB		
k8s.gcr.io/ip-masq-agent	v2.4.1	19bb968f77bba
52MB		
k8s.gcr.io/kube-apiserver	v1.14.2	e455634c173b0
211MB		
k8s.gcr.io/kube-controller-manager	v1.14.2	58f6abb9fb1b3
159MB		
k8s.gcr.io/kube-proxy	v1.14.2	7387c4b88e2df
83.9MB		
k8s.gcr.io/kube-scheduler	v1.14.2	1c93cc1335f8d
83.1MB		

k8s.gcr.io/pause
746kB

3.1

da86e6ba6ca19

```
C:\projects> docker run -d --name ubuntu-air-gap ubuntu
6245ef44c5f648b7c1d25f700dcd6df3dde73fd83a409fe80e894a6bf7f9e00a
```

```
C:\projects> docker run -d --name ubuntu-air-gap2 localhost:5000/ubuntu
ac41296aa94471198f215d96f67953475170088a0d9c48b4b75b394d8ab5160a
```

```
C:\projects> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
ac41296aa944	localhost:5000/ubuntu	"/bin/bash"	11 seconds ago	Exited (0) 10
seconds ago		ubuntu-air-gap2		
6245ef44c5f6	ubuntu	"/bin/bash"	11 minutes ago	Exited (0) 11
minutes ago		ubuntu-air-gap		

```
C:\projects> docker rm ac41296aa944
ac41296aa944
```

```
C:\projects> docker rm 6245ef44c5f6
6245ef44c5f6
```

stop our registry containers after looking up the container IDs

```
C:\projects> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
f24b8d68bdf9	registry	"/entrypoint.sh /etc..."	40 minutes ago	Created
fervent_booth				
6647574b37db	registry	"/entrypoint.sh /etc..."	48 minutes ago	Exited (2) 40
seconds ago		vigorous_cori		
4899db42c740	kindest/node:v1.18.2	"/usr/local/bin/entr..."	4 hours ago	Up 4 hours
127.0.0.1:51089->6443/tcp	kind-control-plane			

```
C:\projects> docker stop 6647574b37db
6647574b37db
```

```
C:\projects> docker stop f24b8d68bdf9
f24b8d68bdf9
```

Remove the stopped registry containers

```
C:\projects> docker rm 6647574b37db
6647574b37db
```

```
C:\projects> docker rm f24b8d68bdf9
```

f24b8d68bdf9

Make sure they're gone for good

C:\projects>docker ps -a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
4899db42c740	kindest/node:v1.18.2	"/usr/local/bin/entr..."	4 hours ago	Up 4 hours
127.0.0.1:51089->6443/tcp	kind-control-plane			

C:\projects>

From this experiment we've setup and air-gapped registry, pushed an image into it and then run from that environment into our air-gapped KIND Kubernetes cluster. This allows us to simulate and environment that would be similar to our server air-gapped environment from a developer perspective.

We noted that the Ubuntu environment we ran was setup like a FaaS which just stands up, runs shell and exits.

Finally we cleaned up the environment to remove the ubuntu containers we created.

Lastly we'll pulverize our cluster

C:\Users\wcsadmin\.crictl>kind delete cluster --name air-gap-kind

Deleting cluster "air-gap-kind" ...