# Service Build & Deployment

1. Please install Kubernetes & Docker on system you plan to run this service

Please refer to installation documents as below

Kubernetes - [installation Setup](https://kubernetes.io/docs/tasks/access-application-cluster/access-cluster-services/)

Docker - [installation Setup](https://docs.docker.com/engine/install/)

1. Build Docker Image

Docker build <file path/UserInfoService> -t <repository on docker hub>/userinfoservice

Check created image

1. Deploy with K8S - creating deployment on host system

Kubectl apply -f <path to deployment.yaml>

Check Deployment running

Kubectl get deployment

1. Deploy Kubernetes Service – This is to expose service over PORT

Kubectl apply -f <path to service.yaml>

# Check Running Service

Kubectl get svc

# Note - For now I have created Dockerfile, deployment.yaml, service.yaml file under Django project folder created for this service

1. Run service with

Minikube service userinfoservice

I am using minikube to run it on my local system but if you are building on production system with kubernetes cluster you can run using “kubectl service userinfoservice”

# API Details

1. User create

Api details

HTTP METHOD: POST

URL : http://<hostname>:<port>/api/user\_create/

Description : User Create Method creates new users in the system on the basis of contact details.

This function can be used for creating bulk users or Single user.

Input format : Json Request

[

  {

    "first\_name": "Jai",

    "last\_name" : "Singh",

    "full\_name" : "Jai Singh",

"emails"  :  [

                   "1x@agma12H1ia1datrsl.com",

                   "x1a1@g2mad1f1H1sfaitrl.com"

                  ],

    "phones" : [

"989859f0",

                       "989810f0"

]

  },

  {

    "first\_name": "Jai",

    "last\_name" : "Singh",

    "full\_name" : "Jai Singh",

"emails"  :  [

                   "1x@agma12H1ia1datrsl.com",

                   "x1a1@g2mad1f1H1sfaitrl.com"

                  ],

    "phones" : [

"989859f0",

                       "989810f0"

]

  },

]

Output format :

{

"Success": "Users Created in System"

}

1. Get User By Id

HTTP Method : GET

URL : http://<hostname>:<port>/api/get\_user\_by\_id/

Description : Get User By Id gets list of all users as shared in request. Output is paginated to help requesting client to enable it to use on front end

Input format

{

"ids" : [

    "15",

    "65"

    ]

}

Output format

{

    "count": 1,

    "next": **null**,

    "previous": **null**,

    "results": [

        {

            "user\_id": 65,

            "first\_name": "Amitsl",

            "last\_name": "shlshl",

            "full\_name": "aisl",

            "email": [

                {

                    "email": "x@agmaitrl.com"

                },

                {

                    "email": "xa1@gmaitrl.com"

                }

            ],

            "phone": [

                {

                    "phone": "9898590"

                },

                {

                    "phone": "9898100"

                }

            ]

        }

    ]}

1. Get User By Name

HTTP Method : GET

URL : http://<hostname>:<port>/api/get\_user\_by\_name/

Description : Get User By Name function retrieves all users based on names in request data.

Input format

{

"names" : [

    "A",

    "j"

    ]

}

Output format

{

    "count": 3,

    "next": "http://192.168.99.100:8000/api/get\_user\_by\_name/?page=2",

    "previous": **null**,

    "results": [

        {

            "user\_id": 86,

            "first\_name": "Amitsl",

            "last\_name": "shlshl",

            "full\_name": "aisl",

            "email": [

                {

                    "email": "1x@agma12H1ia1datrl.com"

                },

                {

                    "email": "x1a1@g2mad1f1H1saitrl.com"

                }

            ],

            "phone": [

                {

                    "phone": "9898590"

                },

                {

                    "phone": "9898100"

                }

            ]

        },

        {

            "user\_id": 84,

            "first\_name": "Amitsl",

            "last\_name": "shlshl",

            "full\_name": "aisl",

            "email": [

                {

                    "email": "1x@agma12Hia1datrl.com"

                },

                {

                    "email": "x1a1@g2madf1H1saitrl.com"

                }

            ],

            "phone": [

                {

                    "phone": "9898590"

                },

                {

                    "phone": "9898100"

                }

            ]

        },

        {

            "user\_id": 79,

            "first\_name": "Amitsl",

            "last\_name": "shlshl",

            "full\_name": "aisl",

            "email": [

                {

                    "email": "x@agma12Hiadatrl.com"

                },

                {

                    "email": "xa1@g2madfH1saitrl.com"

                }

            ],

            "phone": [

                {

                    "phone": "89899"

                },

                {

                    "phone": "9898100"

                }

            ]

        }

    ]

}

1. Update contact

HTTP Method : PUT

URL : http://<hostname>:<port>/api/update\_contact

Description : Update Contact Method updates contacts of user. It can update single user.

Input format

[{

"contact\_type":"Email",

"flag":"0",

"user\_id":1,

"new\_contact":"h@gmail.com",

"old\_contact":"g@gmail.com"

}]

Output format

{

    "Success": "Email h@gmail.com updated for user id 1"

}