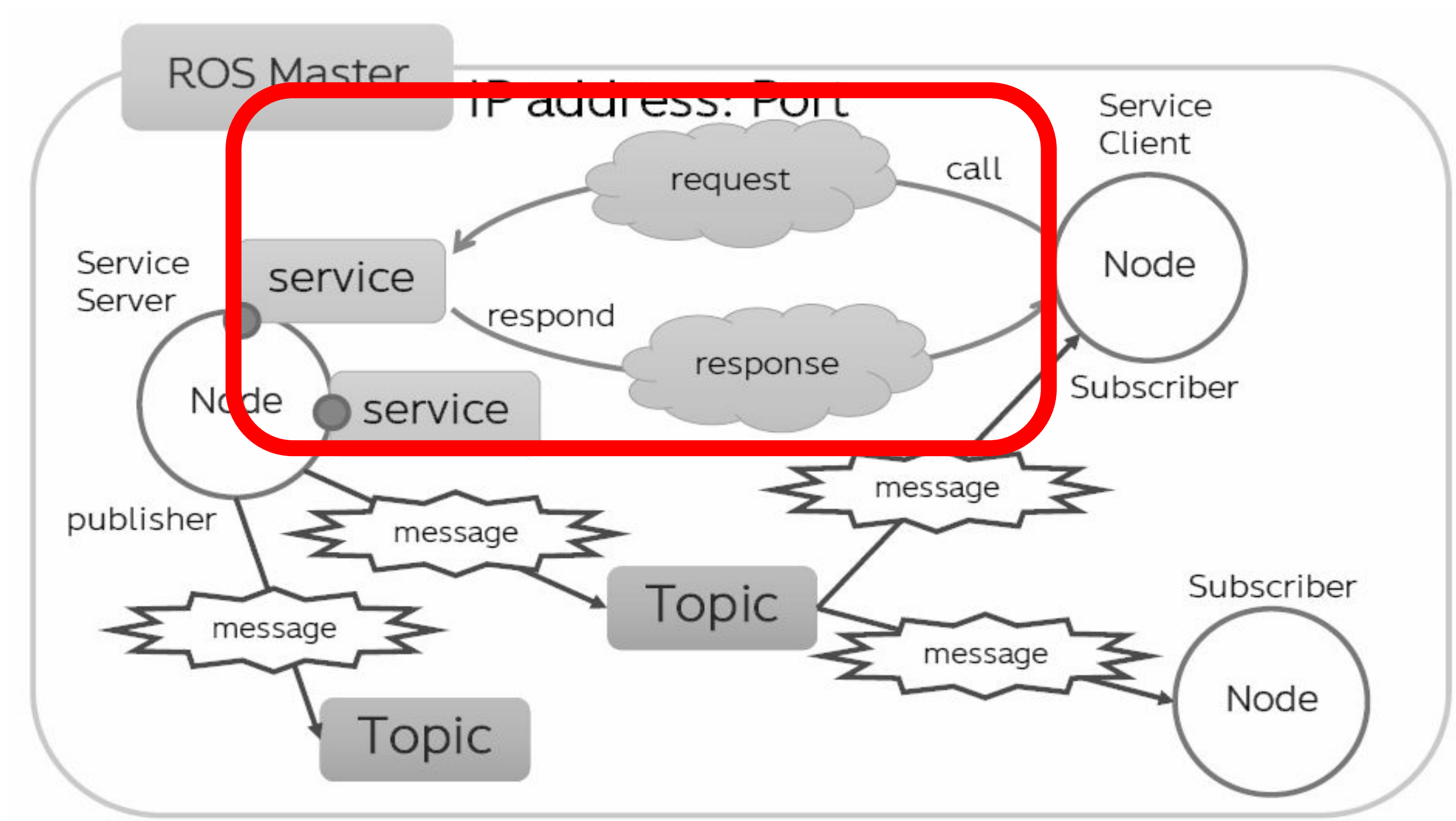
ROS

ROS

Services


ROS Master
IP address: Port
Service Client
call
request
Node
service
Service Server
respond
response
Subscriber
Node
service
publisher
message
message
message
Topic
message
Subscriber
Topic
Node

ROS services command-line

คำสั่งบน command-line ของ ROS ที่เกี่ยวกับ ROS services

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

ทำการเรียก service ชื่อ */service_name*

```
rosservice call /service_name service-args
```

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

ทำการแสดง service ทั้งหมดที่เป็น type *service-type* นี้

```
rosservice find service-type
```

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

ทำการแสดงข้อมูลของ service ชื่อ */service_name*

```
rosservice info /service_name
```

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

ทำการแสดง service ทั้งหมดที่ available ในขณะนั้น

```
rosservice list
```

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

ทำการแสดงชนิดของ service ชื่อ */service_name*

```
rosservice type /service_name
```

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

ทำการแสดง URI ของ service ชื่อ */service_name*

```
rosservice uri /service_name
```

ROS services command-line

- rosservice call
- rosservice find
- rosservice info
- rosservice list
- rosservice type
- rosservice uri
- rosservice node

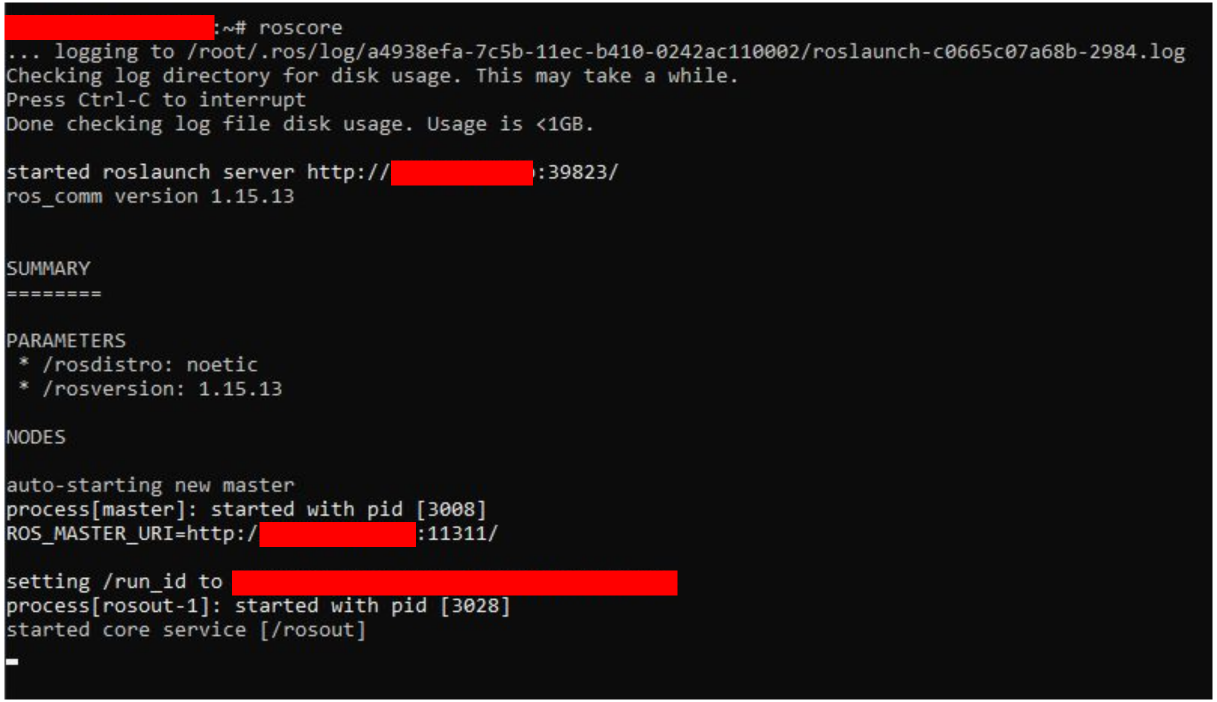
ทำการแสดง node ของ service ชื่อ */service_name*

```
rosservice node /service_name
```

Command-line Tutorial

เปิด terminal

```
$ roscore
```

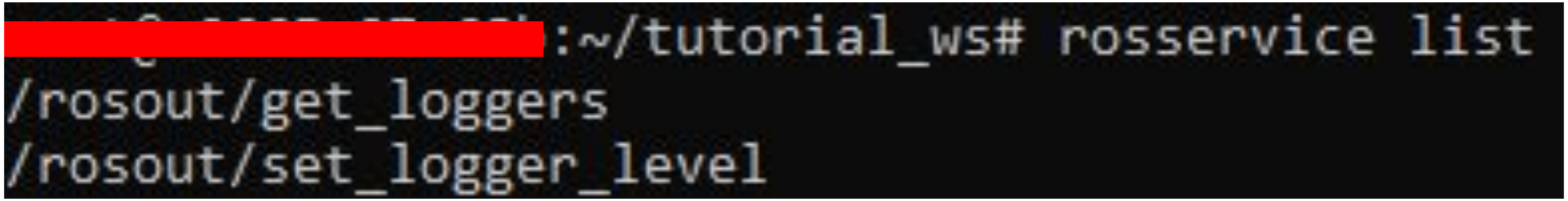


```
[redacted]:~# roscore
... logging to /root/.ros/log/a4938efa-7c5b-11ec-b410-0242ac110002/roslaunch-c0665c07a68b-2984.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://[redacted]:39823/
ros_comm version 1.15.13


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.13

NODES

auto-starting new master
process[master]: started with pid [3008]
ROS_MASTER_URI=http:/[redacted]:11311/

setting /run_id to [redacted]
process[rosout-1]: started with pid [3028]
started core service [/rosout]
```

เปิดหน้าต่างใหม่

```
rosservice list
```

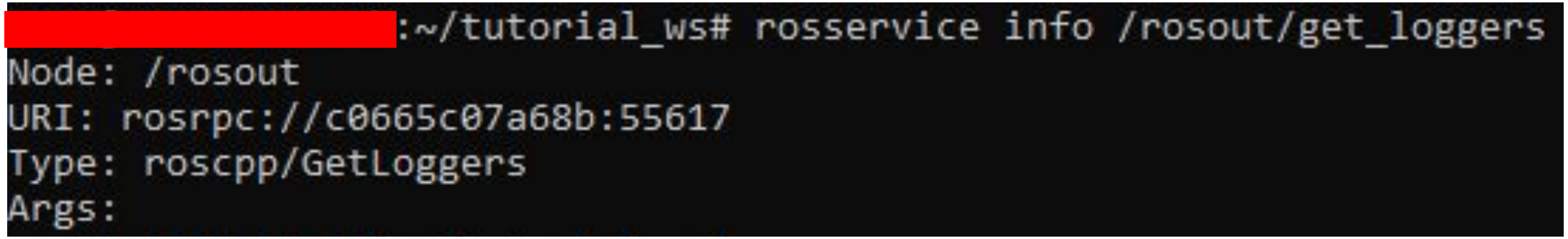
```
$ rosservice list
```



```
[redacted]:~/tutorial_ws# rosservice list
/rosout/get_loggers
/rosout/set_logger_level
```

```
rosservice info /service_name
```

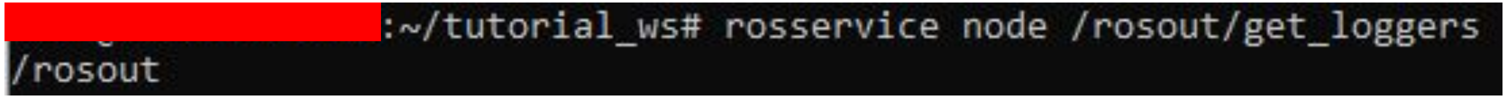
```
$ rosservice info /rosout/get_loggers
```



```
[redacted]:~/tutorial_ws# rosservice info /rosout/get_loggers
Node: /rosout
URI: rosrpc://c0665c07a68b:55617
Type: roscpp/GetLoggers
Args:
```

```
rosservice node /service_name
```

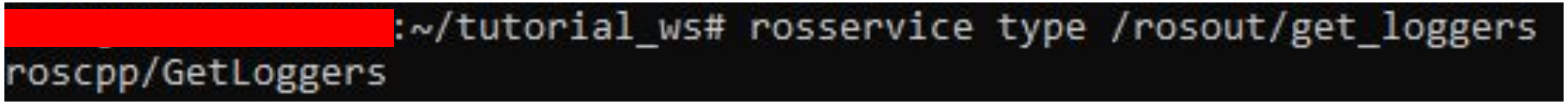
```
$ rosservice node /rosout/get_loggers
```



```
[redacted]:~/tutorial_ws# rosservice node /rosout/get_loggers
/rosout
```

```
rosservice type /service_name
```

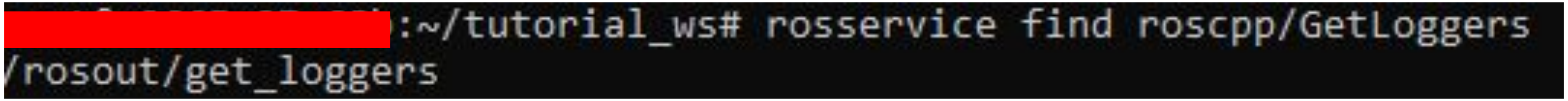
```
$ rosservice type /rosout/get_loggers
```



```
[redacted]:~/tutorial_ws# rosservice type /rosout/get_loggers
roscpp/GetLoggers
```

```
rosservice find service_type
```

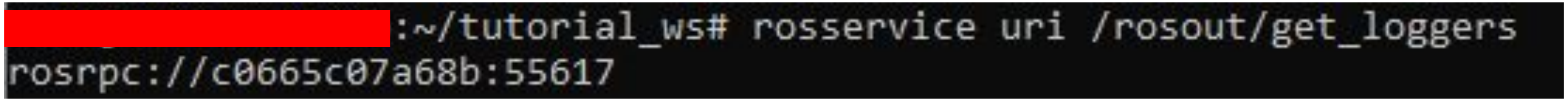
```
$ rosservice find roscpp/GetLoggers
```



```
[redacted]:~/tutorial_ws# rosservice find roscpp/GetLoggers
/rosout/get_loggers
```

```
rosservice uri /service_name
```

```
$ rosservice uri /rosout/get_loggers
```



```
[redacted]:~/tutorial_ws# rosservice uri /rosout/get_loggers
rosrpc://c0665c07a68b:55617
```

ROSPy

Tutorial

```
$ roscd your_package/src
```

```
$ gedit server.py
```

Server (server.py)

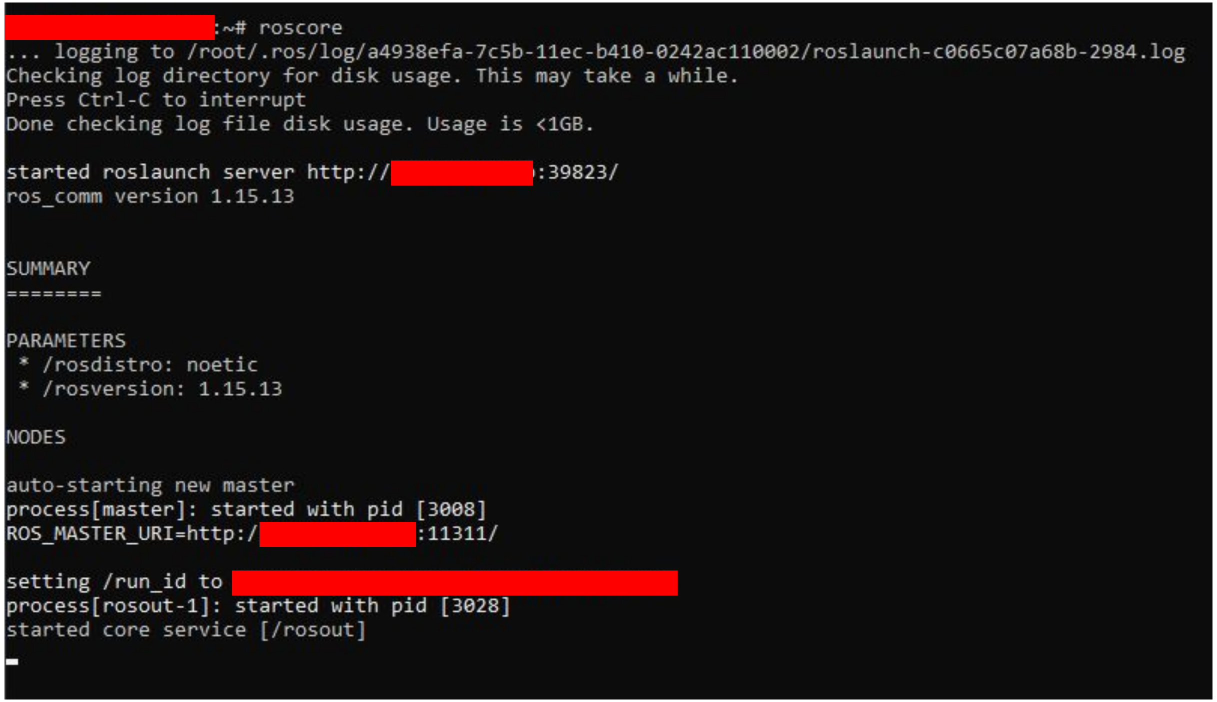
```python
#!/usr/bin/env python3
from std_srvs.srv import Empty, EmptyResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def trigger_server():
    rospy.init_node('trigger_server')
    s = rospy.Service('trigger', Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    trigger_server()
```

```
$ chmod +x server.py
```

TEST

server

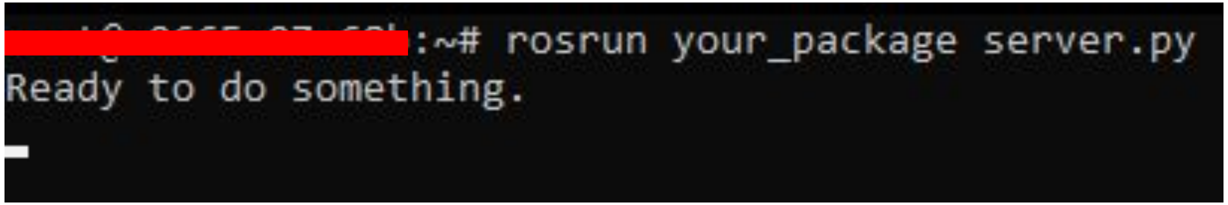
```
$ roscore
```



```
[redacted]:~# roscore
... logging to /root/.ros/log/a4938efa-7c5b-11ec-b410-0242ac110002/roslaunch-c0665c07a68b-2984.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://[redacted]:39823/
ros_comm version 1.15.13


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.13

NODES

auto-starting new master
process[master]: started with pid [3008]
ROS_MASTER_URI=http:/[redacted]:11311/

setting /run_id to [redacted]
process[rosout-1]: started with pid [3028]
started core service [/rosout]
```

เปิดหน้าต่างใหม่

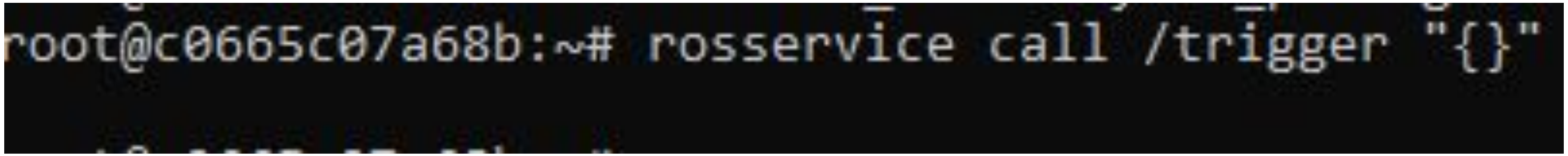
```
$ rosrun your_package server.py
```



```
[redacted]:~# rosrun your_package server.py
Ready to do something.
_
```

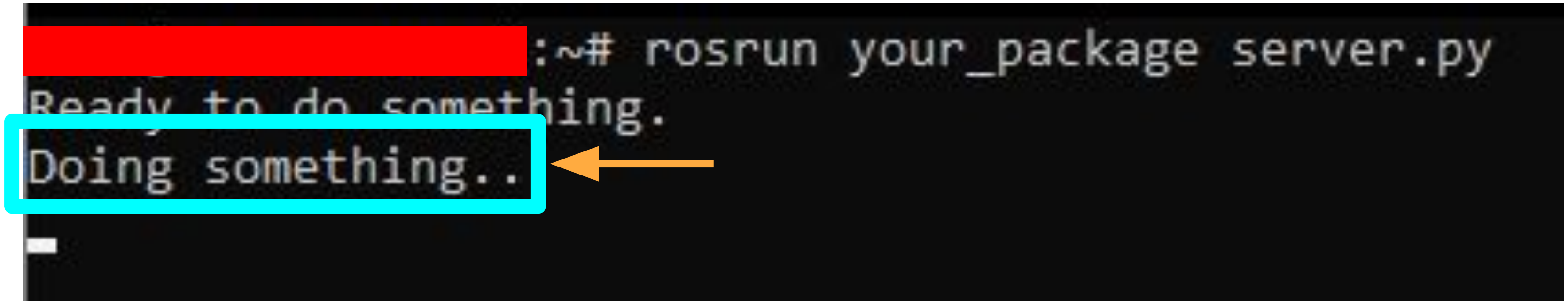
เปิดหน้าต่างใหม่

```
$ rosservice call /trigger "{}"
```



```
root@c0665c07a68b:~# rosservice call /trigger "{}"
```

กลับไปหน้า server



```
[redacted]:~# rosrun your_package server.py
Ready to do something.
Doing something..
```

กด ctrl+c เพื่อหยุดการทำงาน

```
$ roscd your_package/src
```

```
$ gedit client.py
```

Client (client.py)

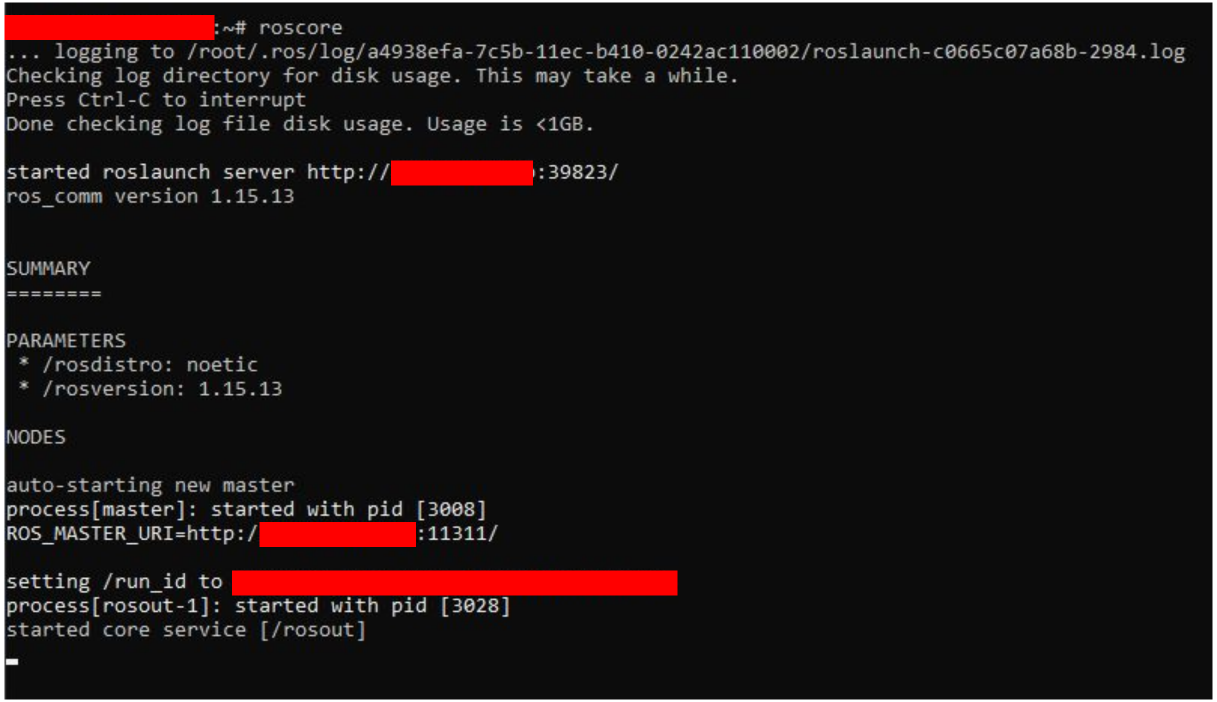
```
#!/usr/bin/env python3
from std_srvs.srv import Empty, EmptyResponse
import rospy

def user_trigger():
    rospy.wait_for_service('trigger')
    try:
        trigger = rospy.ServiceProxy('trigger', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_trigger()
```

```
$ chmod +x client.py
```

TEST

client

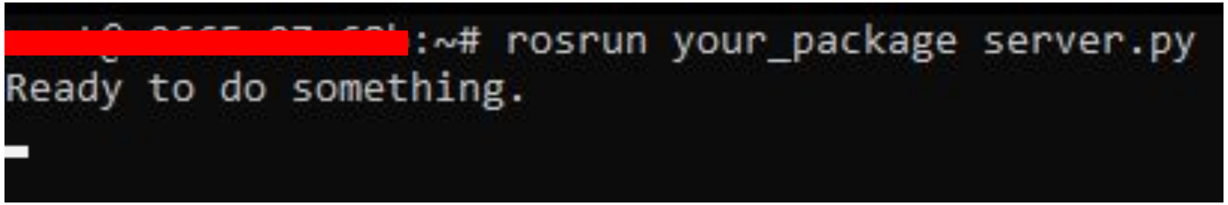
```
$ roscore
```



```
[redacted]:~# roscore
... logging to /root/.ros/log/a4938efa-7c5b-11ec-b410-0242ac110002/roslaunch-c0665c07a68b-2984.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://[redacted]:39823/
ros_comm version 1.15.13


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.13

NODES

auto-starting new master
process[master]: started with pid [3008]
ROS_MASTER_URI=http:/[redacted]:11311/

setting /run_id to [redacted]
process[rosout-1]: started with pid [3028]
started core service [/rosout]
```

เปิดหน้าต่างใหม่

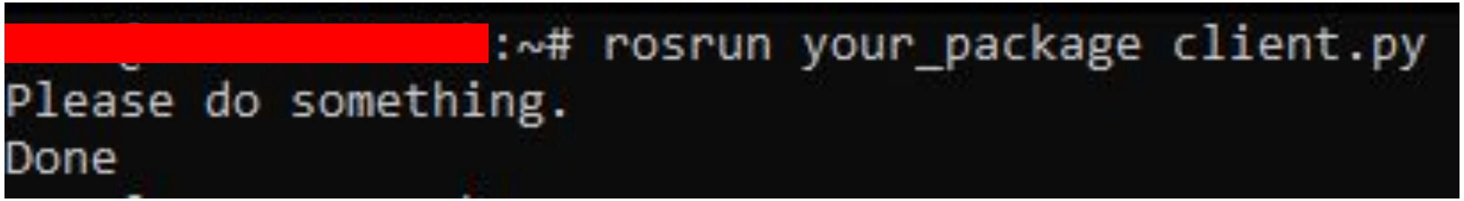
```
$ rosrun your_package server.py
```



```
[redacted]:~# rosrun your_package server.py
Ready to do something.
_
```

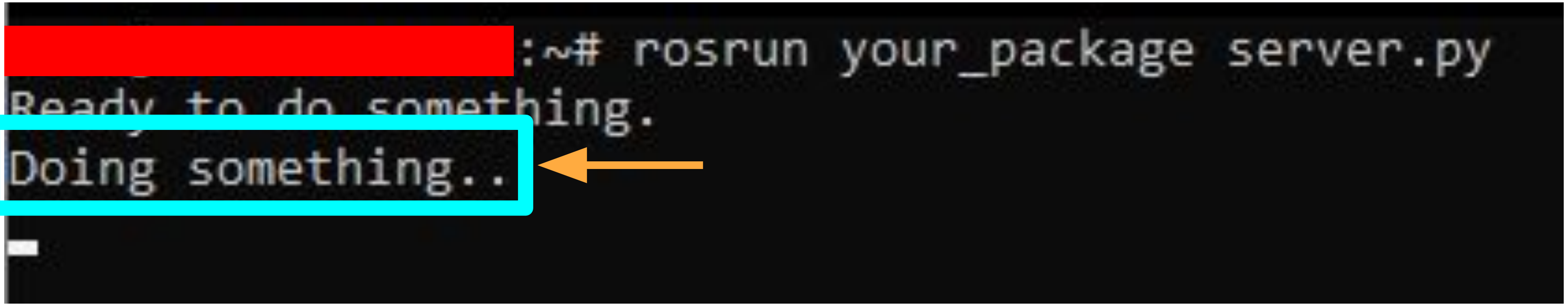
เปิดหน้าต่างใหม่

```
$ rosrun your_package client.py
```



```
[redacted]:~# rosrun your_package client.py
Please do something.
Done
```

กลับไปหน้า server



```
[redacted]:~# rosrun your_package server.py
Ready to do something.
Doing something..
```

กด ctrl+c เพื่อหยุดการทำงาน

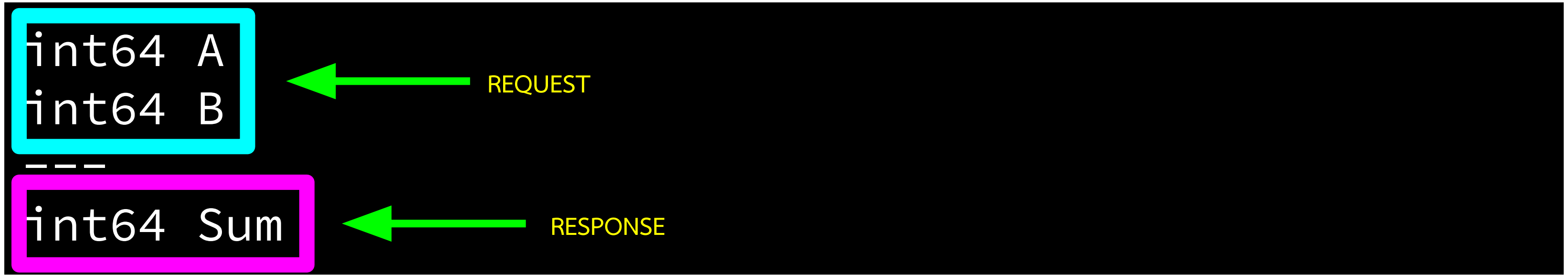
http://wiki.ros.org/std_srvs

Custom service

```
$ roscd your_package/
```

```
$ mkdir srv/
```

```
$ cd srv/
$ gedit Sum.srv
```



```
int64 A
int64 B
---
int64 Sum
```

```
$ cd ..
```

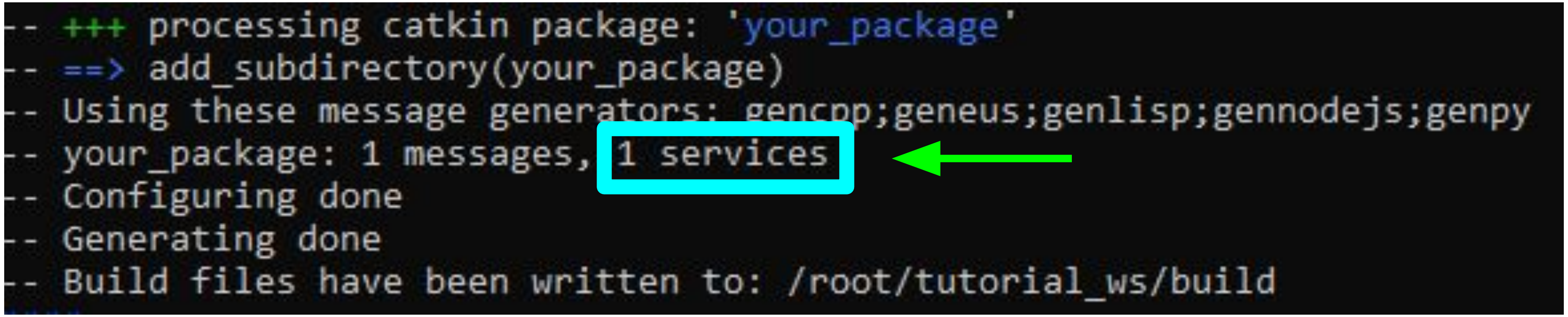
```
$ gedit CMakeLists.txt
```

```
# add_service_files(
#   FILES
~~#   Service1.srv~~
~~#   Service2.srv~~
# )
```

```
add_service_files(
  FILES
  Sum.srv
)
```

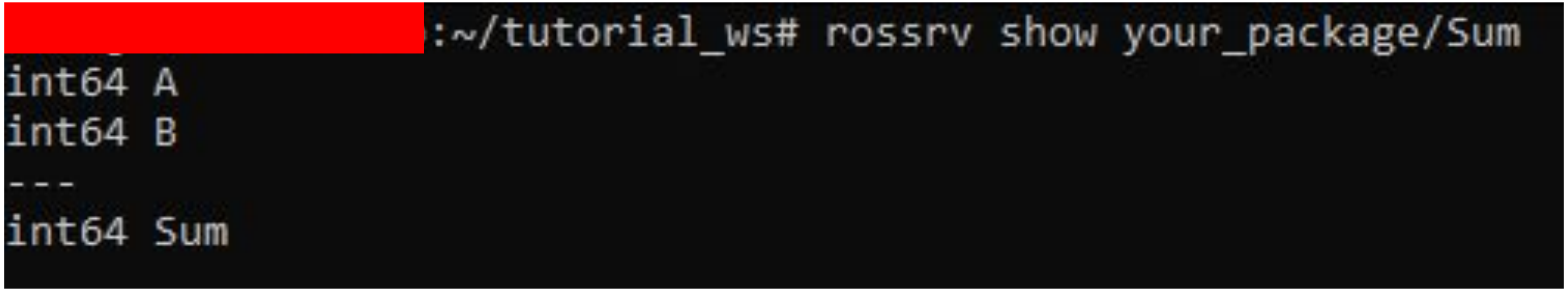
```
$ cd ../..
```

```
$ catkin_make
```



```
-- +++ processing catkin package: 'your_package'
-- ==> add_subdirectory(your_package)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- your_package: 1 messages, 1 services
-- Configuring done
-- Generating done
-- Build files have been written to: /root/tutorial_ws/build
```

```
$ rossrv show your_package/Sum
```



```
[redacted]:~/tutorial_ws# rossrv show your_package/Sum
int64 A
int64 B
---
int64 Sum
```

Use

custom service

on ROSpy

```
$ roscd your_package/src
```

```
$ cp server.py your_server.py
```

```
$ gedit your_server.py
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from std_srvs.srv import Empty, EmptyResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def trigger_server():
    rospy.init_node('trigger_server')
    s = rospy.Service('trigger', Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    trigger_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def trigger_server():
    rospy.init_node('trigger_server')
    s = rospy.Service('trigger', Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    trigger_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def trigger_server():
    rospy.init_node('trigger_server')
    s = rospy.Service('trigger', Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    trigger_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('trigger', Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('trigger', Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service($\boxed{\text{'sum'}}$, Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Empty, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Sum, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Doing something..")
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Sum, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Returning [%s + %s = %s]"%(req.A, req.B, (req.A + req.B)))
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Sum, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Returning [%s + %s = %s]"%(req.A, req.B, (req.A + req.B)))
    return EmptyResponse()

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Sum, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Returning [%s + %s = %s]"%(req.A, req.B, (req.A + req.B)))
    return SumResponse(req.A + req.B)

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Sum, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Returning [%s + %s = %s]"%(req.A, req.B, (req.A + req.B)))
    return SumResponse(req.A + req.B)

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Sum, server_callback)
    print("Ready to do something.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

Server (your_server.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def server_callback(req):
    print("Returning [%s + %s = %s]"%(req.A, req.B, (req.A + req.B)))
    return SumResponse(req.A + req.B)

def sum_server():
    rospy.init_node('sum_server')
    s = rospy.Service('sum', Sum, server_callback)
    print("Ready to sum.")
    rospy.spin()

if __name__ == "__main__":
    sum_server()
```

```
$ cp client.py your_client.py
```

```
$ gedit your_client.py
```

Client (your_client.py)

```python
#!/usr/bin/env python3
from std_srvs.srv import Empty, EmptyResponse
import rospy

def user_trigger():
    rospy.wait_for_service('trigger')
    try:
        trigger = rospy.ServiceProxy('trigger', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_trigger()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_trigger():
    rospy.wait_for_service('trigger')
    try:
        trigger = rospy.ServiceProxy('trigger', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_trigger()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_trigger():
    rospy.wait_for_service('trigger')
    try:
        trigger = rospy.ServiceProxy('trigger', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_trigger()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('trigger')
    try:
        trigger = rospy.ServiceProxy('trigger', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('trigger')
    try:
        trigger = rospy.ServiceProxy('trigger', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        trigger = rospy.ServiceProxy('sum', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        trigger = rospy.ServiceProxy('sum', Empty)
        print("Please do something.")
        resp1 = trigger()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Empty)
        print("Please do something.")
        resp1 = sum()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Empty)
        print("Please do something.")
        resp1 = sum()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Sum)
        print("Please do something.")
        resp1 = sum()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Sum)
        print("Please do something.")
        resp1 = sum()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Sum)
        print("Please sum for me.")
        resp1 = sum()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Sum)
        print("Please sum for me.")
        resp1 = sum()
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```python
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Sum)
        print("Please sum for me.")
        resp1 = sum(1,2)
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

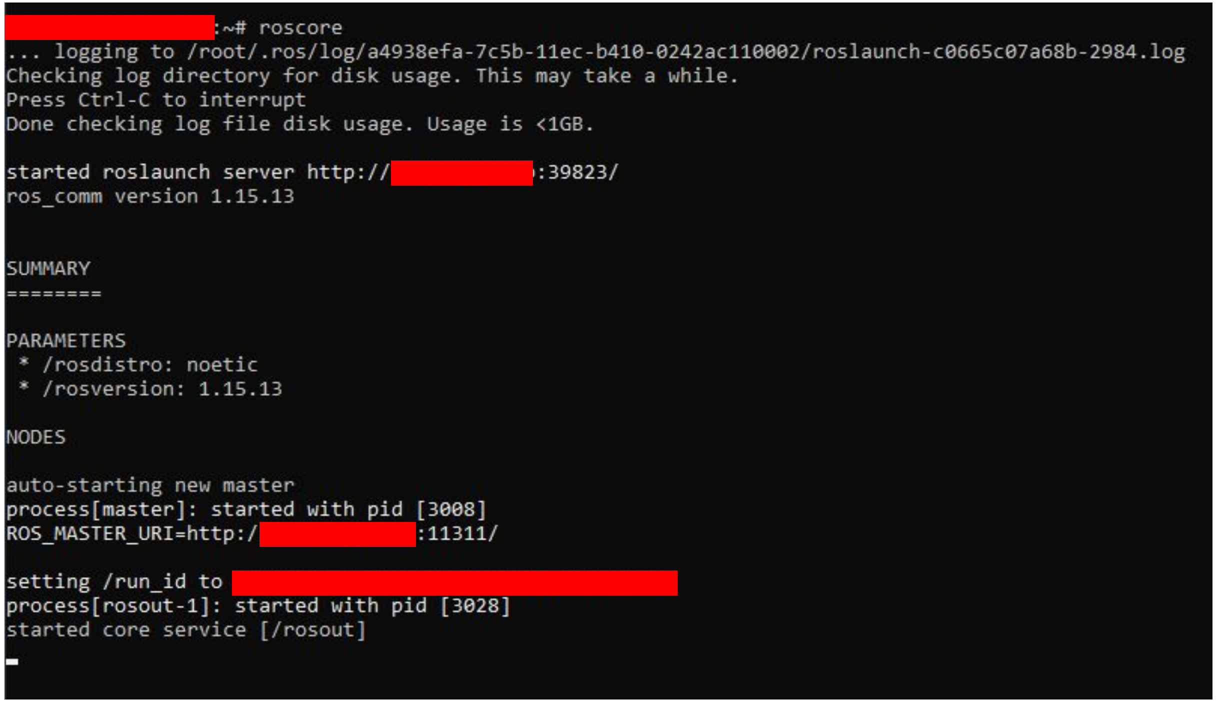
```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Sum)
        print("Please sum for me.")
        resp1 = sum(1,2)
        print("Done")
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

Client (your_client.py)

```
#!/usr/bin/env python3
from your_package.srv import Sum, SumResponse
import rospy

def user_sum():
    rospy.wait_for_service('sum')
    try:
        sum = rospy.ServiceProxy('sum', Sum)
        print("Please sum for me.")
        resp1 = sum(1,2)
        print(resp1)
    except rospy.ServiceException as e:
        print("Service call failed: %s"%e)

if __name__ == "__main__":
    user_sum()
```

TEST

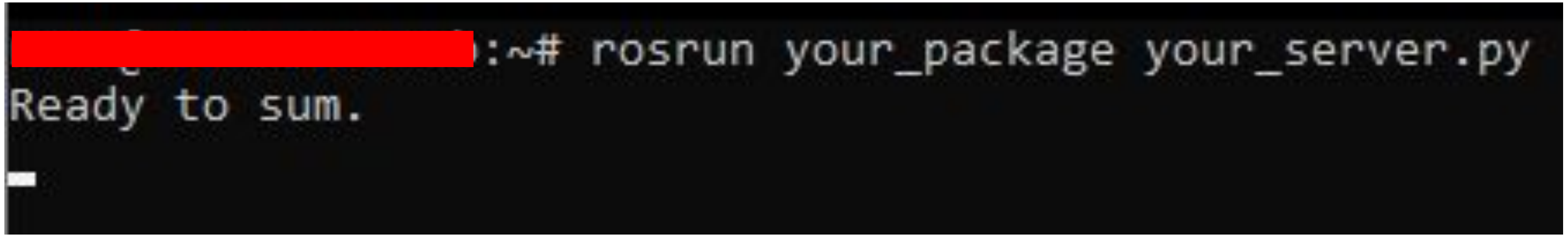
```
$ roscore
```



```
[redacted]:~# roscore
... logging to /root/.ros/log/a4938efa-7c5b-11ec-b410-0242ac110002/roslaunch-c0665c07a68b-2984.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://[redacted]:39823/
ros_comm version 1.15.13


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.13

NODES

auto-starting new master
process[master]: started with pid [3008]
ROS_MASTER_URI=http:/[redacted]:11311/

setting /run_id to [redacted]
process[rosout-1]: started with pid [3028]
started core service [/rosout]
```

เปิดหน้าต่างใหม่

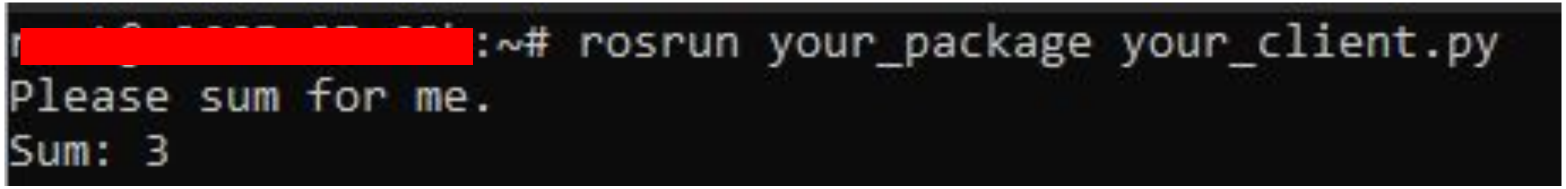
```
$ rosrun your_package your_server.py
```



```
[redacted]:~# rosrun your_package your_server.py
Ready to sum.
```

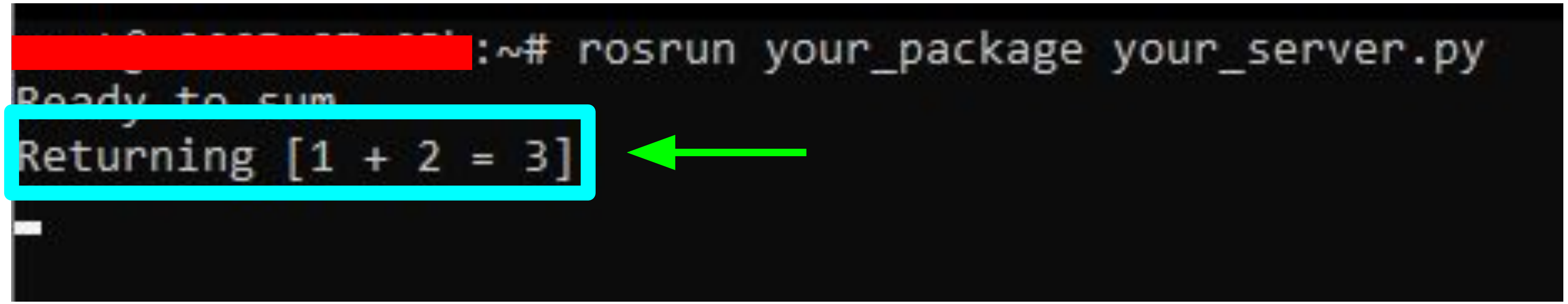
เปิดหน้าต่างใหม่

```
$ rosrun your_package your_client.py
```



```
r[redacted]:~# rosrun your_package your_client.py
Please sum for me.
Sum: 3
```

กลับไปหน้า your_server



```
[redacted]:~# rosrun your_package your_server.py
Ready to sum
Returning [1 + 2 = 3]
```

กด ctrl+c ใน<u>ทุกๆหน้า</u>เพื่อหยุดการทำงาน