

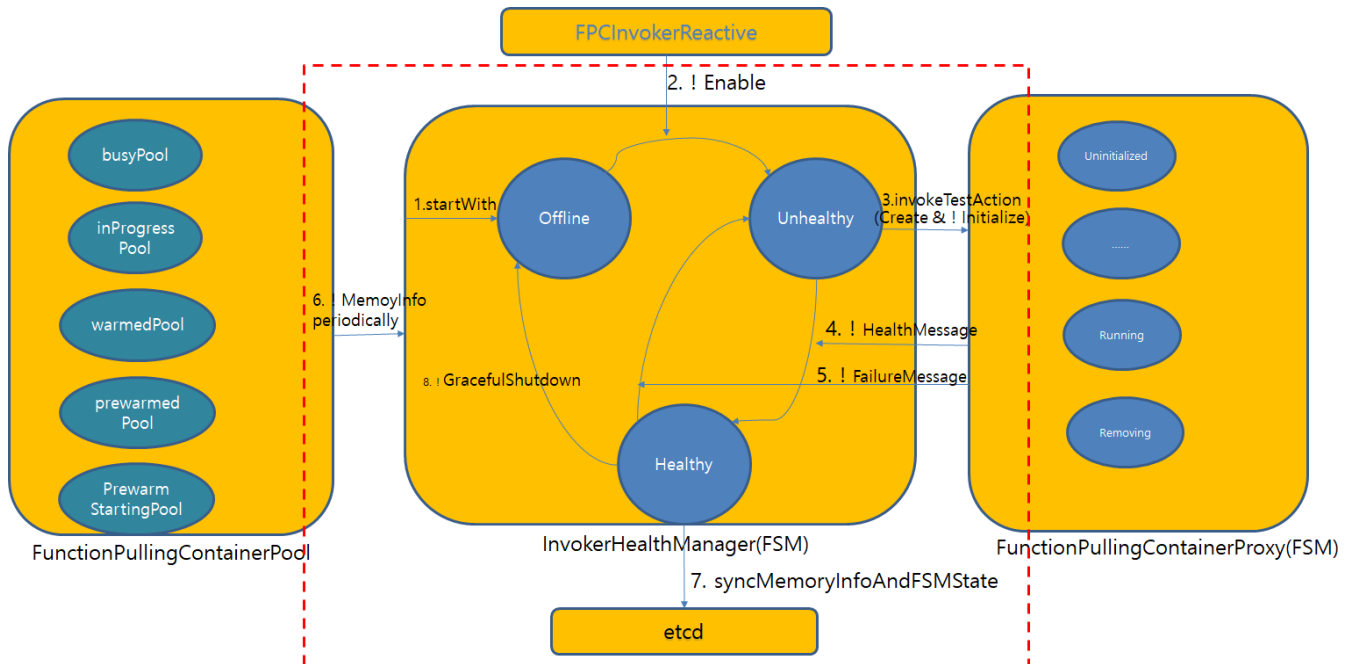
# InvokerHealthyManager

## 1.Introduction

Manage New scheduler's invoker healthy management, invokerHealthyManager actor will send the memory info(e.g. freeMemory, inprogressMemory, busyMemory) and invoker healthy fsm state(e.g. down, up, unhealthy) together to etcd.

Other components can use the healthy data which stored into etcd as well, e.g. when send creationContainerMessage to invoker instance, relative component can get the healthy invokers from etcd.

## 2.Architecture Diagram



1. When invoker starts, it creates a InvokerHealthManger actor with init state: Offline
2. During FPCInvokerReactive initializing, it sends Enable message to InvokerHealthManger actor and InvokerHealthManger will goto(Unhealthy)
3. During Unhealthy, InvokerHealthManger invokes test action: create healthyContainerProxy and send Initialize message to healthyContainerProxy.
4. After healthyContainerProxy initialized, it will executes the healthy activation and sends HealthMessage result to invokerHealthyManager, after finished, it will pull healthy activation and execute it and sends healthyMessage again.  
in invokerHealthyManager side, if failed result invocation number < bufferErrorTolerance, goto(Healthy), at the same time, it will stop to invoke the test action.
5. if FailureMessage comes from FunctionPullingContainerProxy, invokerHealthyManager will goto(Offline), and starts to invoke test action again immediately
6. FunctionPullingContainerPool sends MemoryInfo message to invokerHealthyManager periodically
7. InvokerHealthyManager sends the memoryInfo and its fsm state together to etcd
8. If invokerHealthyMananager receives GracefulShutdown, it will goto(Offline).

## 3.Design consideration

1. invoker healthy data in etcd, including memory info and invokerHealthyManager's state, e.g

```
whisk/invokers/0/0
{"busyMemory":0,"dedicatedNamespaces":[],"freeMemory":10240,"inProgressMemory":0,"status":"up","tags":{}}
whisk/invokers/1/1
{"busyMemory":0,"dedicatedNamespaces":[],"freeMemory":10240,"inProgressMemory":0,"status":"up","tags":{}}
whisk/invokers/2/2
{"busyMemory":0,"dedicatedNamespaces":[],"freeMemory":10240,"inProgressMemory":0,"status":"up","tags":{}}
whisk/invokers/3/3
{"busyMemory":0,"dedicatedNamespaces":[],"freeMemory":10240,"inProgressMemory":0,"status":"up","tags":{}}
whisk/invokers/4/4
{"busyMemory":0,"dedicatedNamespaces":[],"freeMemory":10240,"inProgressMemory":0,"status":"up","tags":{}}
whisk/invokers/5/5
{"busyMemory":0,"dedicatedNamespaces":[],"freeMemory":10240,"inProgressMemory":0,"status":"up","tags":{}}
```

**tags:** This is used for some special actions. e.g. need high memory, need power gpu. we can make the action's activations run on corresponding invokers

**dedicatedNamespaces:** dedicatedNamespaces's all actions run on corresponding invoker

2. when invokerHealthyManager's fsm state changes or received the memoryInfo from FunctionPullingContainerProxy, these healthy data will be stored into etcd immediately