**Course Objective:**

Microservice Architecture ເປັນຮູບແບບສະຖາປັດຕະຍະກຳການພັດທະນາລະບົບຊອບແວທີ່ຈະແບ່ງສ່ວນຂອງຊອບແວ ອອກຈາກກັນໃຫ້ເຮັດວຽກໜ້າທີ່ໃດໜຶ່ງເທົ່ານັ້ນ ຊ່ວຍໃຫ້ລະບົບແຕ່ລະຢ່າງເປັນອິດສະລະ ສາມາດ deploy, update, scale ສະເພາະສ່ວນໂດຍທີ່ບໍ່ກະທົບກັບສ່ວນອື່ນຫຼາຍ ຄືກັນກັບການພັດທະນາຮູບແບບ Monolith ທີ່ການເຮັດວຽກທຸກຢ່າງແມ່ນ ລວມສູນຢູ່ໃນລະບົບດຽວ ການ deploy, update ຫຼື ແກ້ໄຂບັນຫາແມ່ນສົ່ງຜົນກະທົບກັບທັງໝົດຂອງລະບົບ ແລະ ການ scale ໃຫ້ຮອງຮັບການໃຊ້ງານຈຳນວນຫຼາຍເຮັດໄດ້ຍາກ.

ການຝຶກອົບຮົມຈະນຳພາຮຽນຮູ້ ແລະ ເຈົ້າໃຈກ່ຽວກັບ Microservice ຍິ່ງຂຶ້ນ ແລະ ສາເຫດທີ່ Microservice ພວມໄດ້ຮັບ ຄວາມນິຍົມ, ບັນຫາທີ່ Microservice ຈະຊ່ວຍແກ້ໄຂ ຮວມທັງຂໍ້ເສຍ ແລະ ຂັ້ນຕອນການຈັດການກັບຄວາມຊັບຊ້ອນທີ່ມານຳ Microservice. ຜູ້ຝຶກອົບຮົມຈະໄດ້$ຮຽນຮູ້ການນຳໃຊ້ Docker Container ແລະ Kubernetes Container Orchestration ເພື່ອນຳໃຊ້ເປັນ Infrastructure ສຳລັບ Microservice ໃຫ້ໄດ້ຮັບປະສິດທິພາບສູງສຸດ.

**ໄລຍະເວລາ:**

* 6 ມື້ (ມື້ລະ 6 ຊົ່ວໂມງ)
* 70% ປະຕິບັດ
* 30% ທິສະດີ

ຄ່າຝຶກອົບຮົມ:

* ຄ່າຮຽນລວມທັງໝົດ **130,000 ບາດ**

**ຊັບພະຍາກອນທີ່ຕ້ອງໃຊ້:**

* ຄອມພິວເຕີສ່ວນຕົວ

**ລາຍລະອຽດການຝຶກອົບຮົມ:**

* ການອອກແບແບລະບົບ
  + Monolith vs Microservice
  + Virtualization vs Containerization
* Microservice
  + ເບື້ອງຕົ້ນກ່ຽວກັບ Microservice
  + ປະໂຫຍດຂອງ Microservice
  + ຂໍ້ເສຍຂອງ Microservice
* Docker Container
  + ເບື້ອງຕົ້ນກ່ຽວກັບ Docker
  + ຕິດຕັ້ງ Docker (ຢູ່ Local Environment)
  + ການນຳໃຊ້ Docker ພື້ນຖານ
  + ການສ້າງ Docker Image ເອງສຳລັບ Deployment
  + Docker Volume
  + Docker Network
* Docker Compose
  + ເບື້ອງຕົ້ນກ່ຽວກັບ Docker Compose
  + ສ້າງຊຸດລະບົບສຳລັບ Deployment
  + Load Balancing ດ້ວຍ Docker Compose
* Kubernetes
  + ເບື້ອງຕົ້ນກ່ຽວກັບ Kubernetes
    - ປະໂຫຍດຂອງ Kubernetes (Container Orchestration Tool)
    - ການເຮັດວຽກຮ່ວມກັນຂອງ Cluster ລະຫວ່າງ Master Node(s) ແລະ Worker Node(s)
    - ອົງປະກອບຍ່ອຍຂອງ Kubernetes, ໜ້າທີ່ ແລະ ການເຮັດວຽກຂອງແຕ່ອົງປະກອບ
    - ຕົວຢ່າງການນຳໃຊ້
  + ຕິດຕັ້ງ Kubernetes ແບບ Single Node ດ້ວຍ Minikube
  + ທຳຄວາມເຂົ້າໃຈກ່ຽວກັບ Pod, Deployment, ConfigMap, Secrets, Service, Volume
  + ສ້າງ Deployment ຈາກລະບົບຕົວຈິງໃຫ້ມີຄວາມສາມາດ:
    - ຮອງຮັບ Load Balancing ແລ່ນລະບົບຢູ່ຫຼາຍ Server ແລະ ກະຈາຍການເຮັດວຽກໃຫ້ແຕ່ລະ Node (Server) ໄດ້
    - ຮອງຮັບ Auto-Healing ເມື່ອ Microservice ໃດໜຶ່ງຂັດຂ້ອງ ຫຼື ດັບລົງ ຈະສ້າງຕົວໃໝ່ຂຶ້ນມາເຮັດວຽກແທນຢູ່ Server ທີ່ພ້ອມໃຫ້ບໍລິການທັນທີ
    - ຮອງຮັບ Auto-Scaling ເມື່ອມີການໃຊ້ງານຫຼາຍເກີນຄວາມສາມາດຮອງຮັບໄດ້ ໃຫ້ສ້າງ Service ຂຶ້ນໃໝ່ເພີ່ມພ້ອມຮັບຕາມຄວາມຕ້ອງການ ແລະ ຫຼຸດລົງຕາມການໃຊ້ງານຕົວຈິງ
  + ການສ້າງ Persistent Volume ເພື່ອເກັບຂໍ້ມູນຮ່ວມກັນລະຫວ່າງແຕ່ລະ Node ແລະ ການເຊື່ອມຕໍ່ເຂົ້າກັບ Cluster ແລະ Deployment
  + ການສ້າງ Ingress ເປີດຮອງຮັບການເຂົ້າໃຊ້ງານຈາກພາຍນອກ
* ສາທິດການນຳໃຊ້ Kubernetes ຕົວຈິງຢູ່ເທິງ Digital Ocean Managed Kubernetes Cluster
* Kubernetes GUI with Rancher
  + ການນຳໃຊ້ Rancher ລະບົບຈັດການ Kubernetes Cluster ດ້ວຍໜ້າ GUI ຜ່ານເວັບໄຊ
* ແນະນຳແນວທາງການສຶກສາເພີ່ມຕື່ມ ແລະ ເຕັກໂນໂລຊີທີ່ກ່ຽວຂ້ອງ

**RETECH Training Center and Dev**