# ARTS

## 什么是ARTS

ARTS主要分为四个模块，一个ARTS以周为单位进行

### Algorithm：

每周一个LeetCode的算法练习题，主要是为了编程训练和练习；如果不训练，看再多的算法书，也不会算法题目。

### Review

阅读并点评至少一篇英文技术文章，主要是为了学习英文，如果你的英文不行，基本上就无缘技术高手。

### Tip

学习至少一个技术技巧，主要是为了总结和归纳你在日常工作中所遇到的知识点

### Share

分享一篇有观点和思考的技术文章，主要是为了建立你的影响力，能够输出价值观。

## 第一周：

### Algorithm

LeetCode001：

|  |
| --- |
| public static int[] twoSum(int[] nums, int target) {  HashMap<Integer, Integer> indexNum = new HashMap<>();  for (int i = 0; i < nums.length; i++) {  indexNum.put(nums[i],i);  }  for (int i = 0; i < nums.length; i++) {  int complete = target - nums[i] ;  if (indexNum.containsKey(complete) && indexNum.get(complete) != i){  return new int[] {i,indexNum.get(complete)};  }  }  throw new IllegalArgumentException("No such solution..."); } |

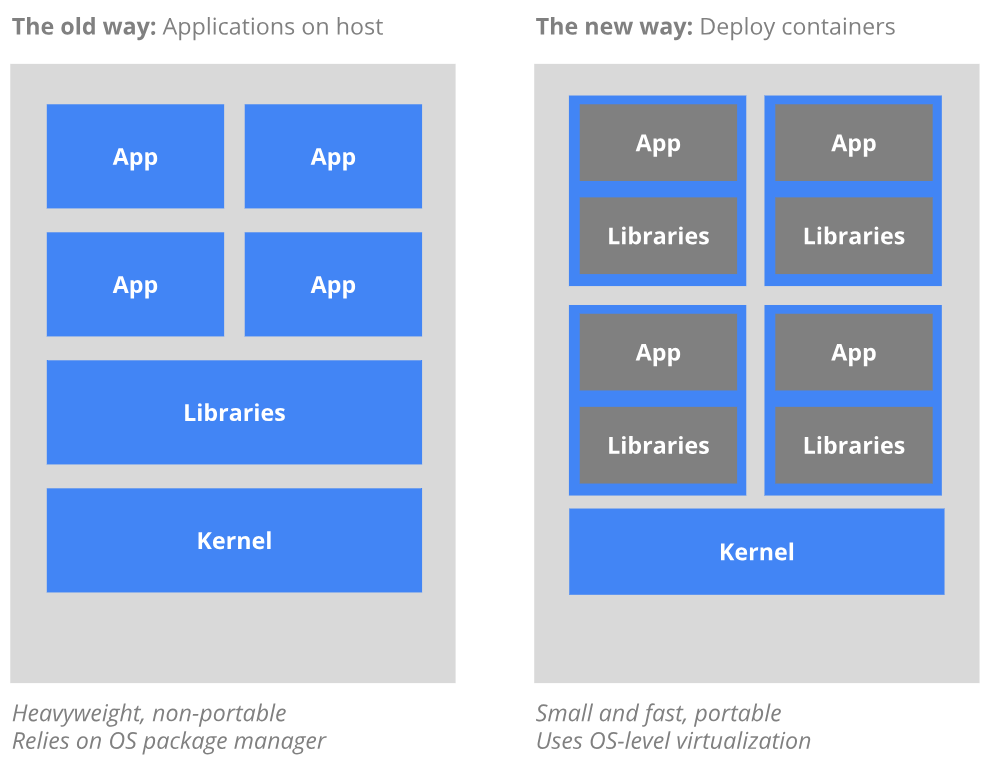
### Review

了解Kubernetes，了解渠道是Kubernetes官网：

**Kubernetes is** a portable, extensible **open-source platform** for **managing containerized workloads and services**, that **facilitates both declarative configuration and automation.**

|  |  |
| --- | --- |
| workloads | 工作负载 |
| that facilitates both  declarative configuration and automation | 这有利于声明配置和自动化 |
| [decade and a half of experience](https://research.google.com/pubs/pub43438.html) | 十五年的经验 |
| scenarios | 场景 |
| velocity | 速度 |
| hoc orchestration | 特别编排 |
| [Annotations](https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations/) | 注释 |
| build atop Kubernetes | 建立在Kubernetes上 |
| monolithic | 单片 |
| **Continuous Integration, Delivery,**  **and Deployment (CI/CD)** | **持续集成，交付和部署** |
| message buses | 消息总线 |
| Open Service Broker | 开放式服务代理 |
| portable mechanisms | 便携机制 |
| rollouts and rollbacks | 推出和回滚 |
|  |  |

。



知道了K8s是用来做什么的了

### Share



## 第二周

### Algorithm

### Review

Elastic Compute Service 弹性计算服务

### Tip

### Share