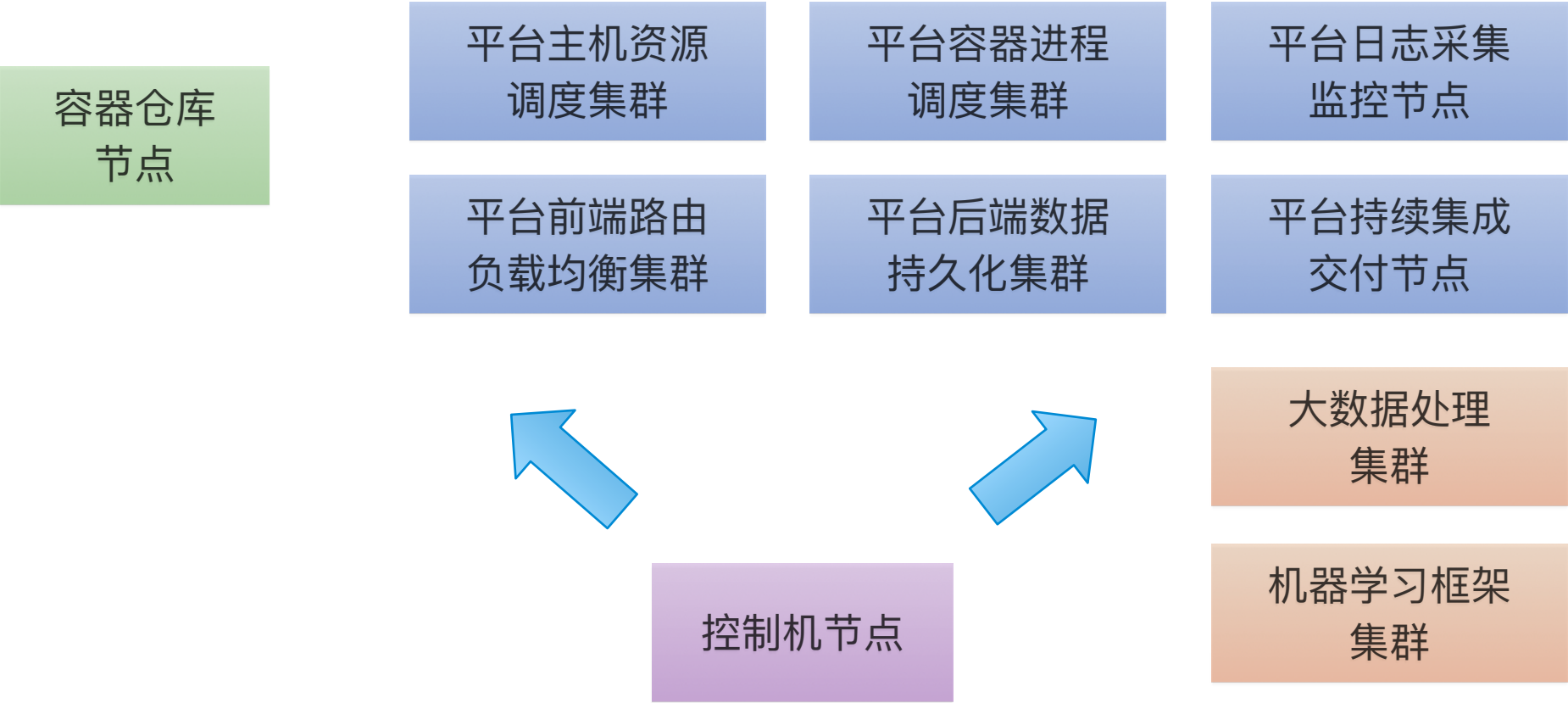
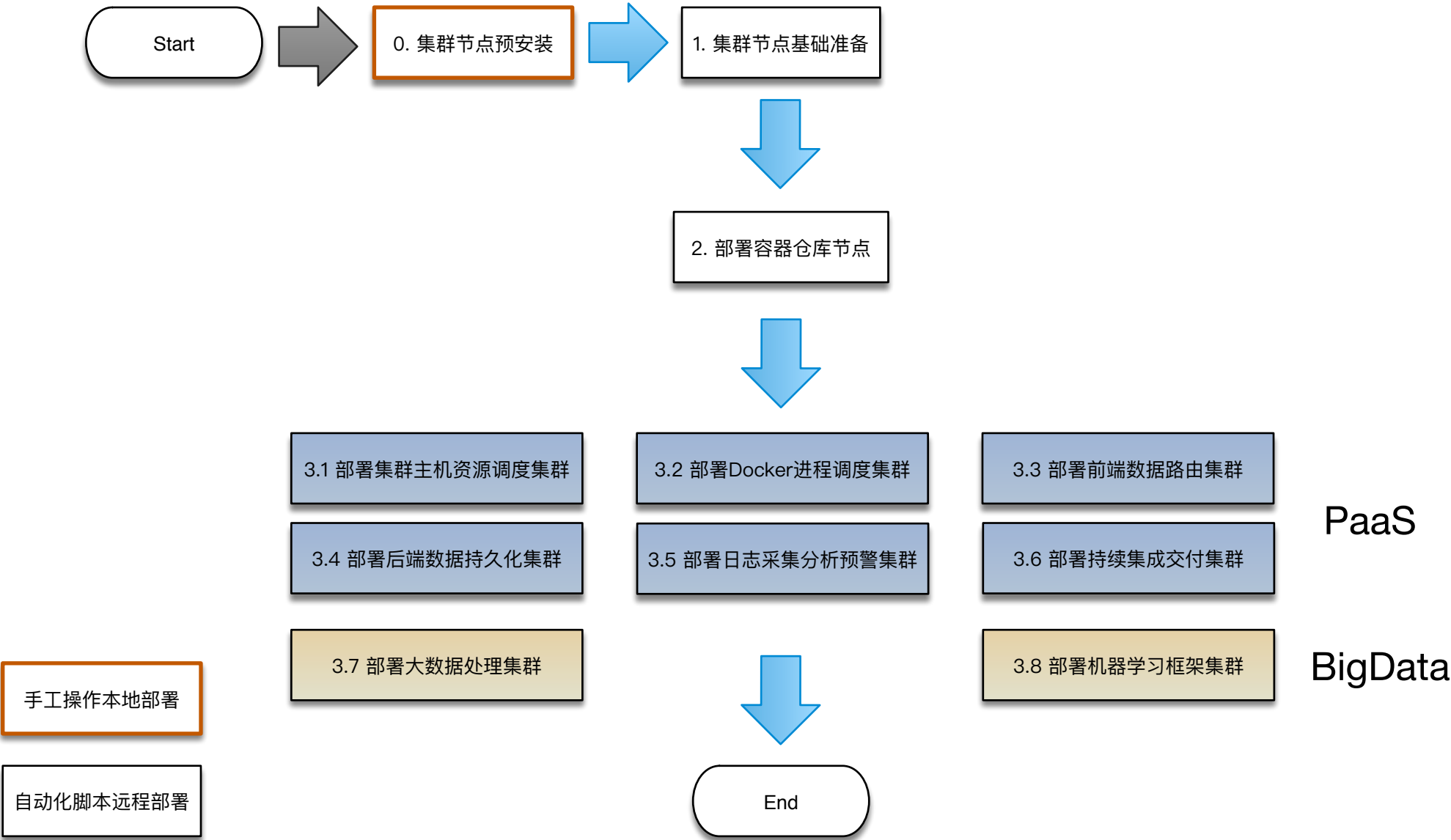


BDPaaS easy_deploy

面向企业数据分析挖掘应用的轻量级PaaS平台。使用单台控制机通过运维脚本自动化远程部署平台集群。
执行：easy_deploy (*) help



平台部署流程



0. 集群节点预安装

集群中每台主机基础环境安装，包括操作系统、网络工具、脚本环境等。此步骤是唯一需要手工操作本地部署的部分。

控制机预安装：

0、ubuntu 14.04(内核升级?)

1、ipython, pip, jq, wget, curl

2、OpenSSH-server

集群工作节点预安装：

0、ubuntu 14.04(内核升级?)

1、pip, wget, curl

2、OpenSSH-server

1. 集群节点基础准备

集群每个节点上都需要部署的基础软件，包括支持容器、网络、远程运维等。

执行： `easy_deploy (*) base install`

目标：

0、建立缺省用户(bdpaas)、组(bdpaas)，与管控机SSH互信

1、二进制安装Docker Engine 1.12.1，并设置启动脚本start-dockerd.sh的启动选项包括

允许非安全访问内网私有仓库；

镜像下载地址依次访问内网私有仓库、国内镜像、原始镜像；

docker虚拟网桥IP地址自定义。

2、远程运维客户端代理（salt-minion）。

2. 部署容器仓库节点

通过docker load加载registry镜像，构建容器私有仓库。

执行：`easy_deploy ($PRIVATE_REGISTRY) registry install`

`$ cat docker-registry-2.5.1.tag.gz | docker load`

`$ registry-bootstrap`

`$ docker load`加载其它基础镜像，并用registry_push_private_image导入仓库中。

设置环境变量PRIVATE_REGISTRY=192.168.11.110:5000

or

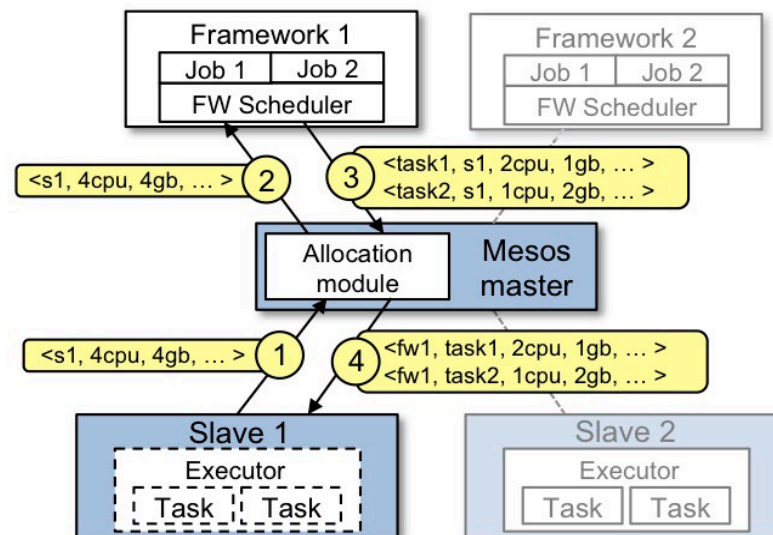
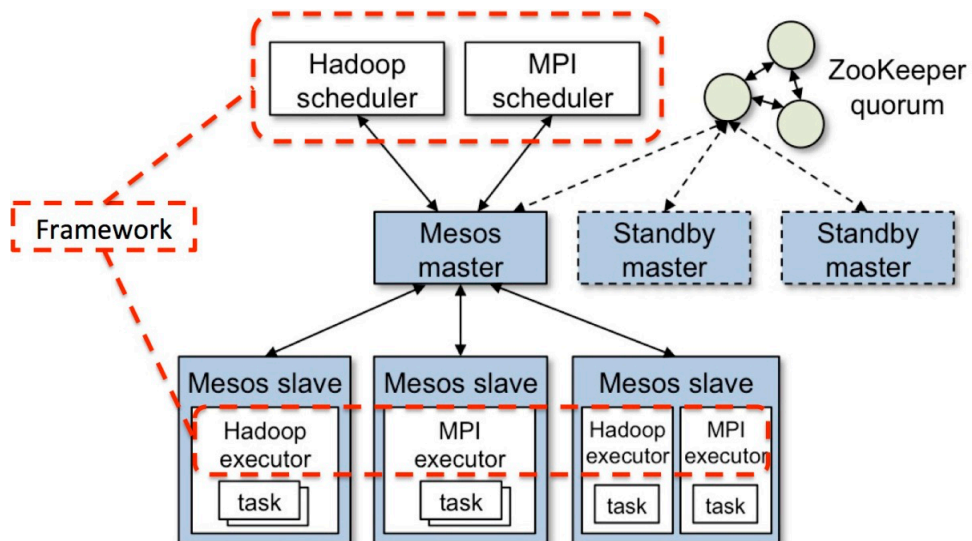
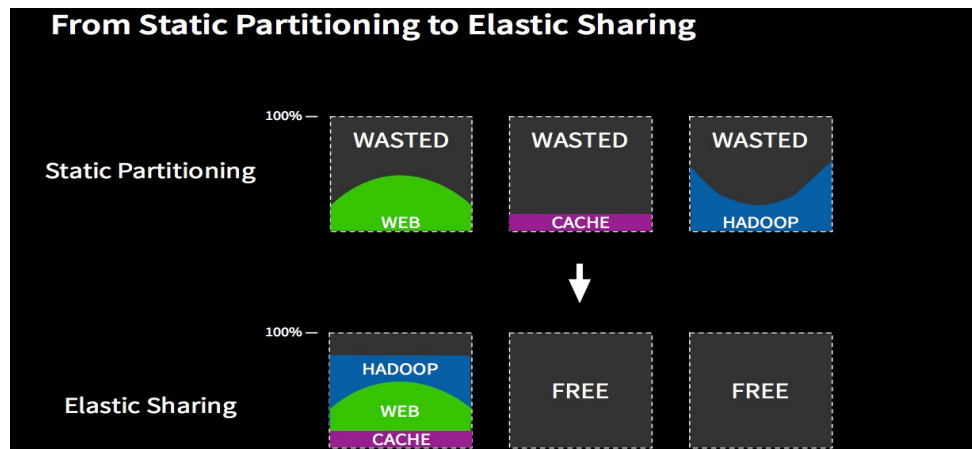
执行：`easy_deploy ($HARBOR_HOSTS) service install harbor`

3.1 部署集群主机资源调度集群

使用Mesos统一管理CPU、内存、存储等物理资源，提高硬件资源利用率。

执行：`easy_deploy ($MESOS_MASTERS) mesos-master install`

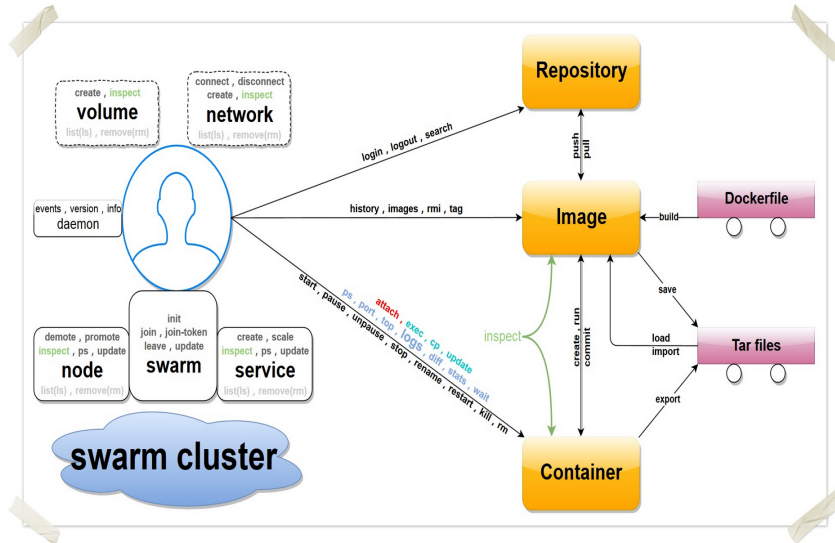
`easy_deploy ($MESOS_SLAVES) mesos-slave install`



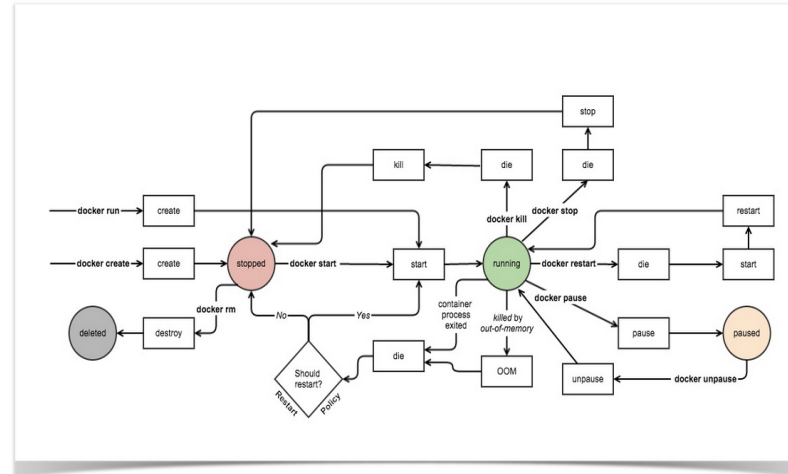
3.2 部署Docker进程调度节点

部署Marathon, Chronos支持长时服务及定时批任务。

执行: `easy_deploy ($MARATHON_HOSTS) marathon install`
`easy_deploy ($CHRONOS_HOSTS) chronos install`



container 事件状态图



3.3 部署前端数据路由节点

部署容器网络动态路由

执行: `easy_deploy (*) service install quagga`

部署分布式协调zookeeper、etcd

执行: `easy_deploy ($ZOOKEEPER_HOSTS) service install zookeeper`

执行: `easy_deploy ($ETCD_HOSTS) service install etcd`

部署服务注册发现Consul

执行: `easy_deploy ($CONSUL_HOSTS) service install consul`

部署反射代理前端Tengine

执行: `easy_deploy ($TENGINE_HOSTS) service install tengine`

or

执行: `easy_deploy ($TRAEFIK_HOSTS) service install traefik`

3.4 部署后端数据持久化节点

部署关系型数据库MariaDB

执行: `easy_deploy ($MARIADB_HOSTS) service install mariadb`

部署基于Mesos的MYSQL集群

执行: `easy_deploy ($MYSOS_HOSTS) service install mysos`

部署分布式对象存储Swift

执行: `easy_deploy ($SWIFT_HOSTS) service install swift`

部署分布式对象存储Ceph

执行: `easy_deploy ($CEPH_MON_HOSTS) service install ceph-mon`

执行: `easy_deploy ($CEPH_OSD_HOSTS) service install ceph-osd`

执行: `easy_deploy ($CEPH_RADOSGW_HOSTS) service install ceph-radosgw`

部署海量流式处理数据库Cassandra (for Spark)

执行: `easy_deploy ($CASSANDRA_HOSTS) service install cassandra`

部署文档数据库MongoDB

执行: `easy_deploy ($MONGODB_HOSTS) service install mongod`

部署文档实时数据库RethinkDB

执行: `easy_deploy ($RETHINKDB_HOSTS) service install rethinkdb`

部署时间序列数据库InfluxDB (for container monitor)

执行: `easy_deploy ($INFLUXDB_HOSTS) service install influxdb`

部署消息队列服务RabbitMQ

执行: `easy_deploy ($RQBBITMQ_HOSTS) service install rabbitmq`

部署缓存服务Redis

执行: `easy_deploy ($REDIS_HOSTS) service install redis`

3.5 部署日志采集分析预警集群

部署搜索数据库ElasticSearch

执行: `easy_deploy ($ELASTICSEARCH_HOSTS) service install elasticsearch`

部署日志抓取服务Logstash

执行: `easy_deploy ($LOGSTASH_HOSTS) service install logstash`

部署数据分析服务Kibana

执行: `easy_deploy ($KIBANA_HOSTS) service install kibana`

3.6 部署持续集成交付集群

部署代码版本管理服务GitLab

执行： `easy_deploy ($GITLAB_HOSTS) service install gitlab`

部署持续集成框架Jenkins

执行： `easy_deploy ($JENKINS_HOSTS) service install jenkins`

部署项目管理平台Redmine

执行： `easy_deploy ($REDMINE_HOSTS) service install redmine`

3.7 部署大数据处理集群

部署Spark

执行: `easy_deploy ($SPARK_HOSTS) service install spark`

部署Hadoop发行版CDH4 (for MR, HDFS, HBASE)

执行: `easy_deploy ($CDH4_HOSTS) service install cdh4`

执行: `easy_deploy ($HDFS_HOSTS) service install hdfs`

执行: `easy_deploy ($HBASE_HOSTS) service install hbase`

部署基于Actor模型的高并发高吞吐量的数据采集前端, 结合数据后端Cassandra和RabbitMQ(Kafka?)

Mesos + Spark + Canssandra + Actor Model + RabbitMQ

3.8 部署机器学习框架

以下按深度学习联盟(DMLC)的分布式深度学习开源平台架构搭建。

部署Parameter Server

执行: `easy_deploy ($PARAMETER_SERVER_HOSTS) service install paramter_server`

部署深度学习库MXNET

执行: `easy_deploy ($MXNET_HOSTS) service install mxnet`

部署Boosting模型XGBoost

执行: `easy_deploy ($XGBOOST_HOSTS) service install xgboost`

部署并行深度学习引擎Minerva

执行: `easy_deploy ($MINERVA_HOSTS) service install minerva`