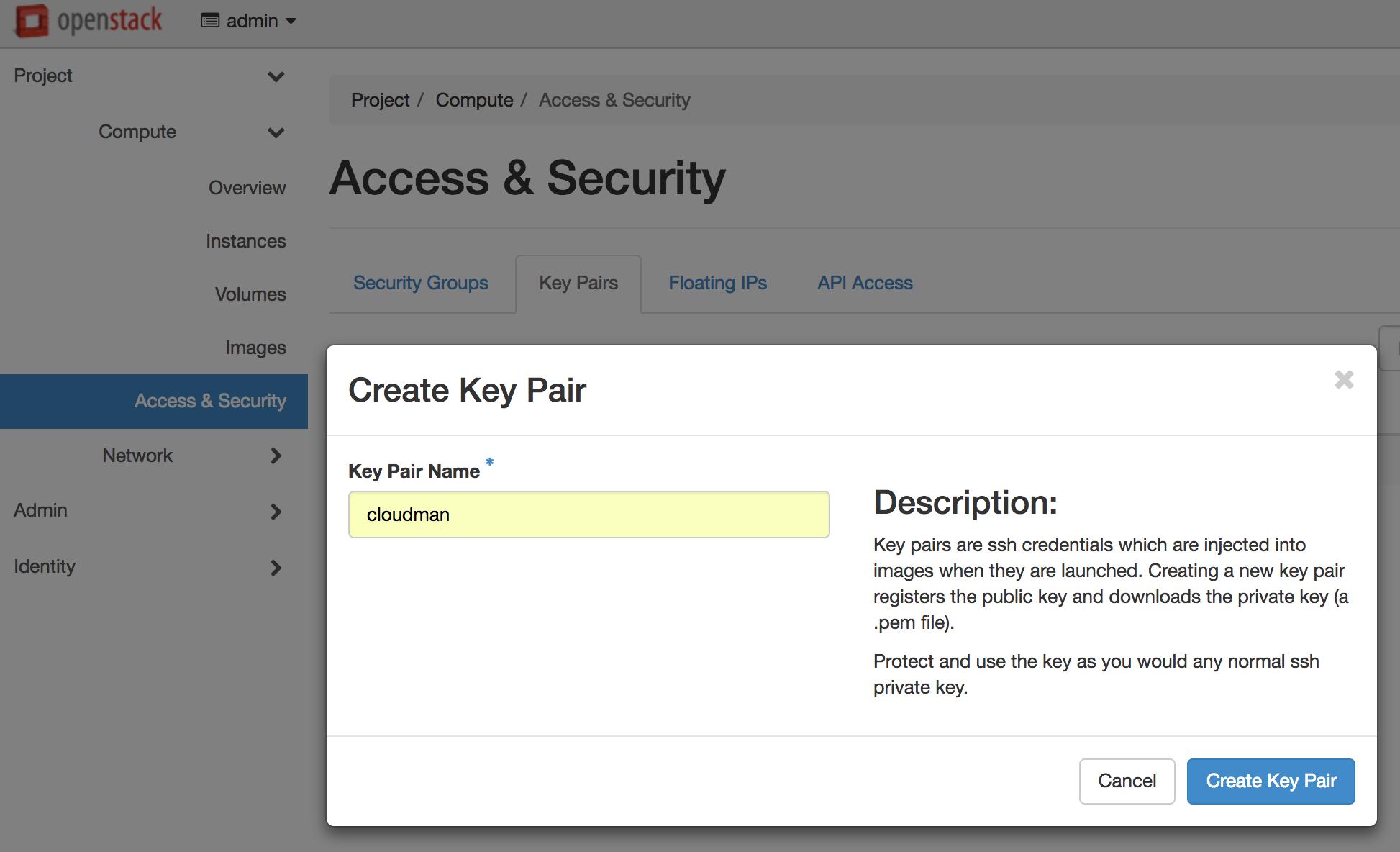
实现 instance 定制化，cloud-init（或 cloudbase-init）只是故事的一半，metadata service 则是故事的的另一半。两者的分工是：metadata service 为 cloud-init 提供自定义配置数据，cloud-init 完成配置工作。

**Metadata Service**

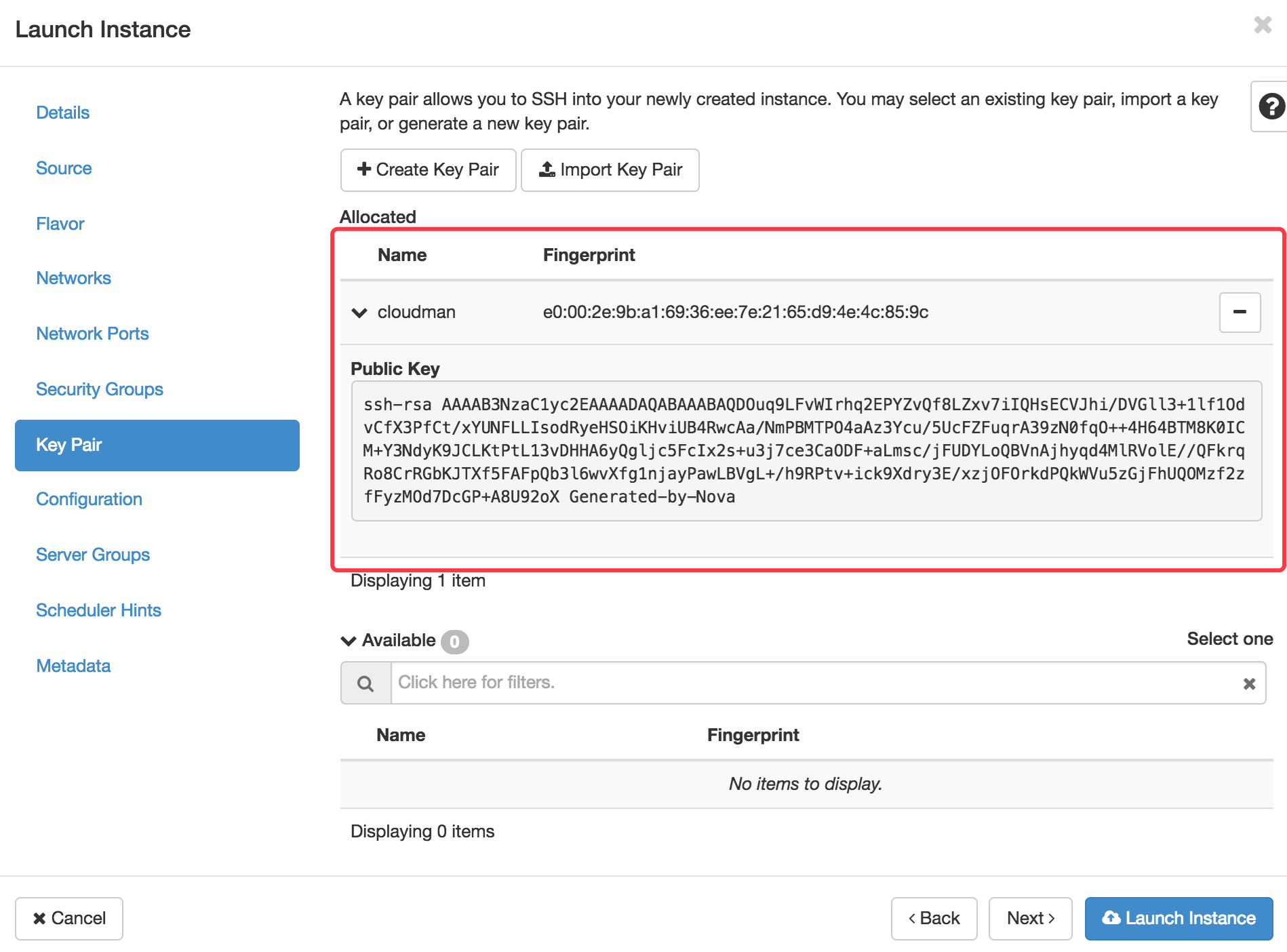
前面讨论了一些 cloud-init 和 cloudbase-init 相关的经验，收到了很多反馈，大家对 instance 启动时是如何完成自定义配置这个过程非常感兴趣，希望能够系统讲一下。这个主题确实很重要，实际应用场景很多，确实很有必要系统讨论一番，作为对现有教程的补充。  
  
instance 是通过 image 部署出来的，image 中包含了操作系统（例如 Ubuntu 16.04），最常用的软件（例如 SSH）以及最通用的配置（例如 eth0 dhcp）。然而在创建 instance 的时候，我们往往希望对 instance 进行一些额外的配置，比如：安装某些包、开启一些服务、添加 SSH 秘钥、配置 hostname 等等。  
  
有几个方法可以完成这项工作：  
  
1. 将这些东西统统做到 image 中。  
  
这种方案可以实现，但不现实。image 应该被看着是一个模板，存放的是通用的内容。在 image 中加入个性化配置的做法要么使 image 变得非常庞杂，要么导致数量众多的 image，不易管理。  
  
2. instance 部署出来之后手工完成个性化配置。  
  
由于需要手工操作，instance 数量多了之后工作量会激增，而且容易出错。  
  
3. 推荐方案：由 OpenStack Metadata Service 提供 instance 的配置信息（这些信息被统称为 metadata）。instance 启动时向 Metadata Service 请求并获得自己的 metadata，instance 的 cloud-init（或 cloudbase-init）根据 metadata 完成个性化配置工作。  
  
这个方案的优点是不需要修改基础 image，保证了 image 的稳定性，同时实现了 instance 自动化地个性配置。

**最高频的应用**

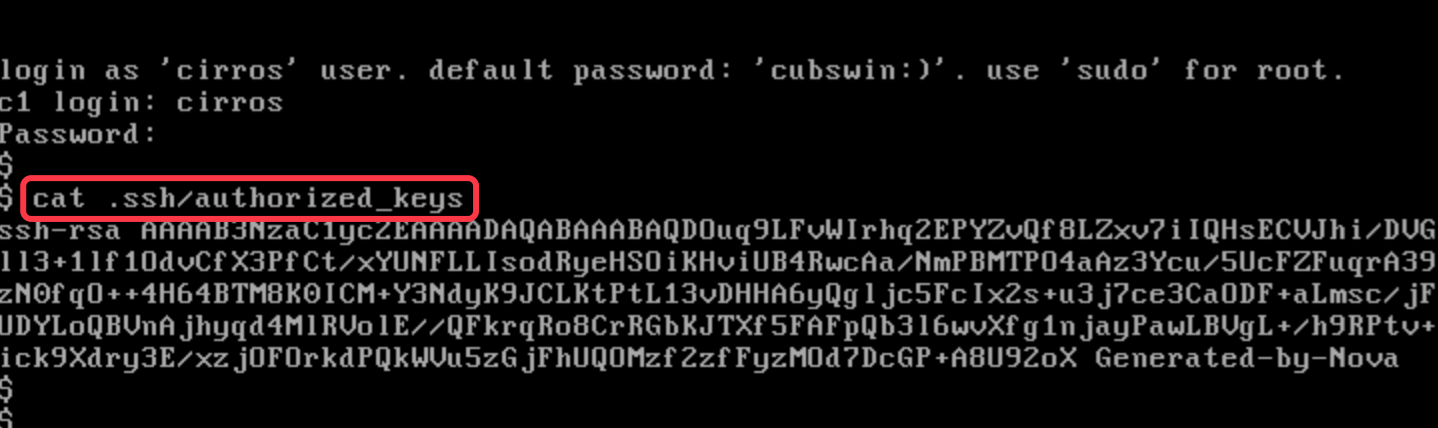
将 ssh public key 添加到 instance。  
  
首先在 “Project -> Compute -> Access & Security” 中创建 Key Pair。



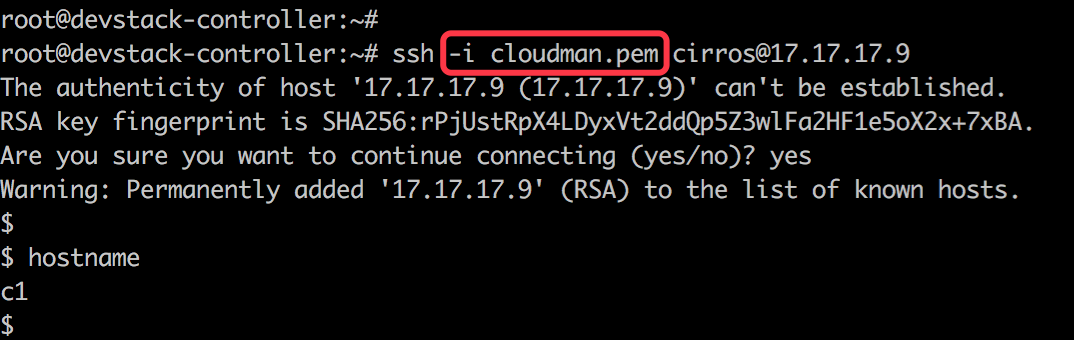
OpenStack 会创建一对 ssh pulbic key 和 private key，public key 存放在 OpenStack 数据库中，private key 会在我们点击 “Create Key Pair” 按钮时自动下载。  
  
现在 "cloudman" 这个 key pair 就是我们要用的 metadata 了。部署 instance 时，选择 "cloudman"。



instance 启动后，可以看到这个 cloudman 的 public key 已经保存到 .ssh/authorized\_keys 中了。



这样我们就可以用 cloudman 的 private key 直接登录 instance。

  
  
本节我们了解了 Metadata Service 的概念及其作用，并通过一个例子获得了些感性认识。下一节就要深入学习了，我们将从 Metadata Service 的架构开始。