Google Cloud Deployment Manager - Introduction 317 Automate deployment and modification of Google Cloud resources in a controlled, predictable way Deploy in multiple environments easily! Avoid configuration drift Avoid mistakes with manual configuration Think of it as version control for your environments Important Note - Always modify the resources created by Deployment Manager using Deployment Manager Google Cloud Deployment Manager - Advantages 318 All configuration is defined in a simple text file - YAML I want a VPC, a subnet, a database and ... Deployment Manager understands dependencies Creates VPCs first, then subnets and then the database (Default) Automatic rollbacks on errors (Easier to retry) If creation of database fails, it would automatic delete the subnet and VPC Version control your configuration file and make changes to it over time Free to use - Pay only for the resources provisioned Get an automated estimate for your configuration Google Cloud Deployment Manager 319 Cloud Deployment Manager - Example - type: compute.v1.instance name: my-first-vm properties: zone: us-central1-a machineType: <> disks: - deviceName: boot type: PERSISTENT boot: true autoDelete: true initializeParams: sourceImage: <> networkInterfaces: - network: <> # Give instance a public IP Address accessConfigs: - name: External NAT type: ONE\_TO\_ONE\_NAT 320 Cloud Deployment Manager - Terminology Configuration file: YAML file with resource definitions for a single deployment Templates: Reusable resource definitions that can be used in multiple configuration files Can be defined using: Python (preferred) OR JinJa2 (recommended only for very simple scripts) Deployment: Collection of resources that are deployed and managed together Manifests: Read-only object containing original deployment configuration (including imported templates) Generated by Deployment Manager Includes fully-expanded resource list Helpful for troubleshootin