Managed Services 82 Use only infrastructure from cloud provider Example: Using VM to deploy your applications or databases You are responsible for: Application Code and Runtime Configuring load balancing Auto scaling OS upgrades and patches Availability etc.. ( and a lot of things!) IAAS (Infrastructure as a Service) 83 Use a platform provided by cloud Cloud provider is responsible for: OS (incl. upgrades and patches) Application Runtime Auto scaling, Availability & Load balancing etc.. You are responsible for: Configuration (of Application and Services) Application code (if needed) Varieties: CAAS (Container as a Service): Containers instead of Apps FAAS (Function as a Service): Functions instead of Apps Databases - Relational & NoSQL (Amazon RDS, Google Cloud SQL, Azure SQL Database etc), Queues, AI, ML, Operations etc! PAAS (Platform as a Service) 84 Microservices Enterprises are heading towards microservices architectures Build small focused microservices Flexibility to innovate and build applications in different programming languages (Go, Java, Python, JavaScript, etc) BUT deployments become complex! How can we have one way of deploying Go, Java, Python or JavaScript .. microservices? Enter containers! 85 Create Docker images for each microservice Docker image has all needs of a microservice: Application Runtime (JDK or Python or NodeJS) Application code and Dependencies Runs the same way on any infrastructure: Your local machine Corporate data center Cloud Advantages Docker containers are light weight Compared to Virtual Machines as they do not have a Guest OS Docker provides isolation for containers Docker is cloud neutral Containers - Docker 86 Requirement : I want 10 instances of Microservice A container, 15 instances of Microservice B container and .... Typical Features: Auto Scaling - Scale containers based on demand Service Discovery - Help microservices find one anot her L o ad B ala n c e r - Dis tri b u te l o a d a mong m u l tip le ins t ances of a mic roser vic e S elf H e ali n g - Do h e a l t h c hec k s and rep l a c e f ailing ins t ances Z e r o Do w n t i m e Deplo y m e n t s - Rel e a se ne w ver sions wi t hou t d owntim e C o n t a i n e r Or c h e s t r a t i o n 87 Serverless What do we think about when we develop an application? Where to deploy? What kind of server? What OS? How do we take care of scaling and availability of the application? What if you don't need to worry about servers and focus on your code? Enter Serverless Remember: Serverless does NOT mean "No Servers" Serverless for me: You don't worry about infrastructure (ZERO visibility into infrastructure) Flexible scaling and automated high availability Most Important: Pay for use Ideally ZERO REQUESTS => ZERO COST You focus on code and the cloud managed service takes care of all that is needed to scale your code to serve millions of requests! And you pay for requests and NOT servers! 88 Serverless - My Perspective! Serverless - Important Features: 1: Zero worry about infrastructure, scaling and availability 2: Zero invocations => Zero Cost (Can you scale down to ZERO instances?) 3: Pay for invocations and NOT for instances (or nodes or servers) Serverless Level 1: Features (1 + 2) Serverless Level 2: Features (1 + 2 + 3) When I refer to Serverless, I'm referring to Level 2 HOWEVER cloud providers include managed services at Level 1 and Level 2: Level 1: Google App Engine (Google Calls it "App Engine is a fully managed, serverless platform"), AWS Fargate (AWS calls it "serverless compute engine for containers") Scale down to ZERO instances when there is no load, BUT you pay for number (and type) of instances running! Level 2: Google Functions, AWS Lambda, Azure Functions etc You pay for invocations 89 GCP Managed Services for Compute Service Details Category Compute Engine High-performance and general purpose VMs that scale globally IaaS Google Kubernetes Engine Orchestrate containerized microservices on Kubernetes Needs advanced cluster configuration and monitoring CaaS App Engine Build highly scalable applications on a fully managed platform using open and familiar languages and tools PaaS (CaaS, Serverless) Cloud Functions Build event driven applications using simple, single-purpose functions FaaS, Serverless Cloud Run Develop and deploy highly scalable containerized applications. Does NOT need a cluster! CaaS (Serverless