

# Part I - Introduction into Cloud Computing 2

## 1. What is Cloud Computing (CC)? 3

### 1.1. First Utility Computing 3

### 1.2. The New Features of Cloud Computing (CC) 4

### 1.3. Cloud Service Provider (CSP) 6

#### 1.3.1 Thin Provisioning 7

### 1.4. CC via a CSP (not via own cloud OS) 8

#### 1.4.1 Public Cloud 9

#### 1.4.2 Private Cloud 9

#### 1.4.3 Hybrid Cloud 9

#### 1.4.4 Infrastructure-as-a Service (IaaS) 10

##### 1.4.4.1 Resource Administration by the User 11

##### 1.4.4.2 The Role of the CSP 11

##### 1.4.4.3 Nested Virtualization 12

#### 1.4.5 Platform-as-a-Service (PaaS) 12

#### 1.4.6 Software-as-a-Service (SaaS) 13

#### 1.4.7 Special Access-Offers 14

##### 1.4.7.1 Community Cloud 14

##### 1.4.7.2 Virtual Private Cloud 15

##### 1.4.7.3 Multi Cloud 15

### 1.5. CC with own Cloud OS (not via CSP) 16

## 2. What are the Pros and Cons of CC? 17

### 2.1. CC via CSP 19

#### 2.1.1 Pros 19

##### 2.1.1.1 Server Consolidation via VM Migration 20

#### 2.1.2 Cons 21

##### 2.1.2.1 Vendor-Lock-in 22

### 2.2. CC with own Cloud OS 22

#### 2.2.1 Pros for own Cloud OS 23

#### 2.2.2 Cons for own Cloud OS 25

## 3. Technologies of CC 27

### 3.1. Common Technologies for Both, CSP Clouds and Own Cloud 27

### 3.2. Tenant and Multi-Tenancy 27

#### 3.2.1 Tenant 27

#### 3.2.2 Multi-Tenancy 28

### 3.3. Linux Containers (LXCs) 29

#### 3.3.1 Cgroups 30

#### 3.3.2 Namespace Isolation 30

#### 3.3.3 LXC Functionalities 31

3.4.	Virtualization by means of VMs	32
3.5.	Auto-Scaling	33
3.6.	Disjoint Software Technologies	34
3.7.	Specific Technologies for CC via CSPs	34
3.7.1	Language and Platform-specific APIs	35
3.8.	Specific Technologies for CC via own Cloud OS	35
3.8.1	Location of Own Cloud OS	36
3.8.2	Basic Services in own Cloud OS	38
4.	Virtual Machines (VMs)	39
4.1.	What is a VM?	40
4.2.	Advantages of Virtualization	42
4.2.1	Setup of a VM	45
4.3.	Disadvantages of Virtualization	46
4.4.	Host OS and Guest OS	47
4.5.	Hypervisor	48
4.5.1	ESXi, Hyper-V and KVM/QEMU	51
4.6.	Paravirtualization	53
4.6.1	Virtio for Paravirtualization	54
5.	Server Virtualization	55
5.1.	Comparison between Classical Computer and Virtualized Server	56
5.2.	How the Host OS sees Guest Oses, QEMUs and User Applications	58
5.3.	Privileged CPU instructions in Guest OS	59
5.4.	Server Virtualization in a Cloud	60
5.5.	Virtualization of Memory, CPU and IO	62
5.5.1	Memory Virtualization	62
5.5.2	CPU Virtualization	64
5.5.3	IO Virtualization	65
5.5.3.1	Example of IO Virtualization by Sending an Email from a VM	66
5.6.	Hardware Accelerators for Server Virtualization	66
5.6.1	Accelerators for Memory Virtualization	67
5.6.2	Accelerators for CPU Virtualization	68
5.6.3	Accelerators for IO Virtualization	68
5.6.3.1	SR-IOV	69
5.6.4	Status of Hardware Accelerators for Server Virtualization	70
5.6.5	Summary of Hardware Accelerators	70
5.7.	Inter-vCPU and Inter-VM Communication on the Same Server	71
5.7.1	Inter-VM Communication Without Ipvshmem on the Same Server	73
5.7.1.1	Linux Tap Device	73
5.7.1.2	Macvlan Driver	74
5.7.1.3	Macvtap	74

- 5.7.1.4 Inefficient Inter-VM Communication as a Result 75
- 5.7.2 Inter-VM Communication With Ipvshmem on the Same Server 76
- 5.8. Inter-VM Communication between Servers without Cloud OS 80
- 5.9. Inter-VM Communication between Servers with Cloud OS 82
  - 5.9.1 Communication Efficiency Problem in CSP Clouds 82
  - 5.9.2 Communication Efficiency Problem in OpenStack 82
  - 5.9.3 Alternative Inter-VM Communication between Servers in Open-Stack 86

## **Part II - OpenStack as Cloud OS 87**

### **6. Overview on OpenStack and its Services 88**

- 6.1. REST Protocol 89
- 6.2. Securing REST Requests and VMs 90
- 6.3. Calling and Controlling an OpenStack Service 91
  - 6.3.1 Extended UUID Usage for Information Items 92
  - 6.3.2 Calling via REST API 93
  - 6.3.3 Calling via Host OS Shell 94
  - 6.3.4 Calling via GUI 95
    - 6.3.4.1 Caveats 96
  - 6.3.5 Calling via CLI 96
  - 6.3.6 Calling via UUID 96
- 6.4. Overview on OpenStack Services 97
- 6.5. The 10 Most-Important OpenStack Services 101
- 6.6. The Smallest Possible OpenStack System 107
- 6.7. Controller Node and Other Nodes 109
- 6.8. Minimum Software Stack in a Compute Node 110
- 6.9. Intra-Service and Inter-Service Communication 110
  - 6.9.1 Oslo.Messaging 111
    - 6.9.1.1 Advanced Message Queuing Protocol (AMQP) 112
    - 6.9.1.2 ZeroMQ Message Transfer Protocol (ZMTP) 113
    - 6.9.1.3 Notifications 114
- 6.10. The Horizon Service 115
  - 6.10.1 Horizon GUI 116
  - 6.10.2 List of Tabs and Sub-tabs for the User GUI 119
    - 6.10.2.1 Compute Tab 120
    - 6.10.2.2 Volume Tab 120
    - 6.10.2.3 Network Tab 121
    - 6.10.2.4 Object Store Tab 122
  - 6.10.3 List of Tabs and Sub-tabs for the Admin GUI 122
    - 6.10.3.1 Overview Tab 122
    - 6.10.3.2 Compute Tab 122
    - 6.10.3.3 Volume Tab 124

- 6.10.3.4 Network Tab 125
- 6.10.3.5 System Tab 125
- 6.10.4 Horizon Projects (Tenants) And User Authorization 127**
- 6.10.5 Role-Based Access-Control 127**
- 6.10.6 Extensions to Horizon 128**
- 6.10.7 Abstract Service API of Horizon 129**
- 6.10.8 Horizon GUI Terminology 129**
- 6.11.General OpenStack Terminology 131**
- 6.12.The Nova Service 133**
  - 6.12.1 Nova Compute Cells 134**
  - 6.12.2 Nova Internal Setup 135**
  - 6.12.3 Parallel Processing Inside of Nova 137**
  - 6.12.4 Communication Between Nova Components 138**
- 6.13.The Neutron Service 139**
  - 6.13.1 Neutron Task List 139**
  - 6.13.2 Neutron Software-Defined Networks 141**
  - 6.13.3 Arbitrary Network Topologies by means of Neutron 142**
  - 6.13.4 Neutron Components for ISO Layer 2 Operation 142**
  - 6.13.5 Neutron Components for ISO layer 3 Operation 143**
  - 6.13.6 Further Neutron Components 143**
  - 6.13.7 Types of Data to be Exchanged in OpenStack 145**
    - 6.13.7.1 Example of Data-Type Exchanges 147
    - 6.13.7.2 Example of a Neutron Network Topology 149
- 6.14.The Keystone Service 150**
  - 6.14.1 Keystone Terminology 151**
  - 6.14.2 Securing API Requests 153**
  - 6.14.3 Fernet Tokens 154**
    - 6.14.3.1 Symmetric Data Encryption in Keystone 154
    - 6.14.3.2 The Fernet Algorithm for a Fernet Token 156
    - 6.14.3.3 Creating a Fernet Token 157
    - 6.14.3.4 Using a Fernet Token 158
    - 6.14.3.5 Checking the Fernet Token 158
    - 6.14.3.6 Example Interplay Between Caller, Callee and Horizon 160
    - 6.14.3.7 Advantages of Fernet Tokens 162
    - 6.14.3.8 Disadvantages of Fernet Tokens 163
  - 6.14.4 Token Revocation 163**
  - 6.14.5 Keystone REST Request Examples 163**
  - 6.14.6 Internal Keystone Setup 166**
- 6.15.The Cinder Service 167**
  - 6.15.1 Cinder Terminology 168**
  - 6.15.2 Cinder's External Setup 169**
  - 6.15.3 Cinder's Internal Setup 171**
  - 6.15.4 Cinder API 172**
    - 6.15.4.1 Volume Groups 173
    - 6.15.4.2 Storage QoS 174
    - 6.15.4.3 Volume Access 174
    - 6.15.4.4 Volume Moves 175
    - 6.15.4.5 Volume Scheduling 175
    - 6.15.4.6 Attachment of a Virtual Hard Drive to a VM 175

6.15.4.7	Accessing Records in Cinder Volumes	176
<b>6.15.5</b>	<b>Cinder Snapshots</b>	<b>177</b>
<b>6.15.6</b>	<b>Cinder Backups</b>	<b>177</b>
<b>6.15.7</b>	<b>Cinder Volume Encryption</b>	<b>178</b>
<b>6.16</b>	<b>The Swift Service</b>	<b>178</b>
<b>6.16.1</b>	<b>Manifest Objects</b>	<b>180</b>
<b>6.16.2</b>	<b>Swift Terminology</b>	<b>181</b>
<b>6.16.3</b>	<b>Swift Object Replication</b>	<b>182</b>
<b>6.16.4</b>	<b>Asynchronous Eventual Consistency for Replicas</b>	<b>182</b>
<b>6.16.5</b>	<b>Erasur Codes</b>	<b>183</b>
6.16.5.1	Data Storage as Use Case for Erasure Codes	184
<b>6.16.6</b>	<b>Swift Data Hierarchy</b>	<b>185</b>
6.16.6.1	Level of Swift Accounts	185
6.16.6.2	Level of Swift Containers	186
6.16.6.3	Level of Swift Objects	186
<b>6.16.7</b>	<b>Internal Swift Setup</b>	<b>187</b>
6.16.7.1	Account Layer Components	187
6.16.7.2	Container Layer Components	188
6.16.7.3	Object Layer Components	189
6.16.7.4	Proxy Server	189
<b>6.16.8</b>	<b>Reliability Concepts of Swift</b>	<b>190</b>
6.16.8.1	Manifest Objects and Reliability	190
6.16.8.2	Swift Proxies and Reliability	191
<b>6.16.9</b>	<b>Replica Policy and Reliability</b>	<b>192</b>
<b>6.16.10</b>	<b>Object Upload with Erasure Code Policy</b>	<b>192</b>
<b>6.16.11</b>	<b>Object Download with Erasure Code Policy</b>	<b>193</b>
<b>6.16.12</b>	<b>Load Balancing</b>	<b>194</b>
<b>6.16.13</b>	<b>Hash Rings</b>	<b>194</b>
6.16.13.1	Ring Partition Number	195
6.16.13.2	Disk Number	196
6.16.13.3	Data-to-Partition-to-Device Table	197
6.16.13.4	2nd Table In the Hash Ring	199
6.16.13.5	The Swift-Ring-Builder	199
<b>6.17</b>	<b>The Glance Service</b>	<b>200</b>
<b>6.17.1</b>	<b>Glance Internal Setup</b>	<b>200</b>
<b>6.17.2</b>	<b>Glance Tasks</b>	<b>201</b>
<b>6.18</b>	<b>The Trove Service</b>	<b>202</b>
<b>6.18.1</b>	<b>Trove Terminology</b>	<b>202</b>
<b>6.18.2</b>	<b>Supported DBMSes</b>	<b>203</b>
<b>6.18.3</b>	<b>Classification of Trove Datastores</b>	<b>203</b>
<b>6.18.4</b>	<b>Storage Locations</b>	<b>204</b>
<b>6.18.5</b>	<b>Trove Configuration Groups</b>	<b>204</b>
<b>6.18.6</b>	<b>Trove Internal Setup</b>	<b>204</b>
<b>6.18.7</b>	<b>Minimum-Requirements for a Database-VM</b>	<b>207</b>
<b>6.18.8</b>	<b>Trove Guest-Agent Configuration</b>	<b>208</b>
<b>6.18.9</b>	<b>Save Inter-Component Communication</b>	<b>208</b>
<b>6.18.10</b>	<b>Creation of Datastores</b>	<b>209</b>
<b>6.18.11</b>	<b>The Trove API</b>	<b>210</b>
6.18.11.1	Create and Manage Database VMs	210
6.18.11.2	Create and Manage Replicas of Database VMs	211

- 6.18.11.3 Create and Manage Backups of Database VMs 211
- 6.18.11.4 Create and Manage Databases 211
- 6.18.11.5 Manage Datastores 211
- 6.18.11.6 Create and Manage Configuration Groups 212
- 6.19. The Ceilometer Service 212**
  - 6.19.1 Ceilometer Terminology 213**
  - 6.19.2 Types of Meters 214**
  - 6.19.3 Ceilometer Components 215**
  - 6.19.4 Ceilometer Operations 217**
    - 6.19.4.1 Collecting Usage Data 217
    - 6.19.4.2 Collecting Events 217
    - 6.19.4.3 Collecting Alarms 217
  - 6.19.5 Supported Hypervisors 218**
  - 6.19.6 Other Supported Tools 218**
- 6.20. The Heat Service 218**
  - 6.20.1 Orchestrating 219**
  - 6.20.2 The native Heat API 221**
  - 6.20.3 AWS CloudFormation Templates 221**
  - 6.20.4 Heat Orchestration Templates (HOTs) 221**
    - 6.20.4.1 The HOT language 222
    - 6.20.4.2 HOT Features 223
    - 6.20.4.3 Differences between a Heat Stack and HOT 224
  - 6.20.5 Heat Setup 226**
  - 6.20.6 Heat Terminology 227**

## **Part III - Amazon Web Service AWS 229**

### **7. Overview 230**

- 7.1. Using AWS Services 231**
- 7.2. Calling AWS Services 232**
  - 7.2.1 Calling via an SDK 233**
  - 7.2.2 Regional Endpoints For Calls 233**
- 7.3. AWS Accounts and Users 234**
- 7.4. AWS Account IDs 235**
- 7.5. AWS Resource Names (ARNs) 236**
- 7.6. AWS Identity and Access Management (IAM) 236**
  - 7.6.1 Account Access Keys 237**
  - 7.6.2 IAM User Access Key 238**
  - 7.6.3 Temporary IAM Credentials 238**
- 7.7. Data Protection and AA in REST Requests 240**

### **8. Description of Selected AWS Services 241**

- 8.1. Simple Storage Service“ (S3) 241**

- 8.2. Buckets 242**
  - 8.2.1 Creation of a Bucket 242**
  - 8.2.2 Folders in a Bucket 243**
  - 8.2.3 Retrieving Objects in a Bucket 244**
  - 8.2.4 Automatic Versioning of Objects 244**
  - 8.2.5 Website Hosting 244**
  - 8.2.6 Additional AA for Buckets 245**
    - 8.2.6.1 Bucket-Access Control-Lists (Bucket ACLs) 245
    - 8.2.6.2 Bucket Access Policies 246
  - 8.2.7 Other Access Policies and ACLs 247**
    - 8.2.7.1 Object-Access Control-Lists 247
    - 8.2.7.2 User Access Policies 248
    - 8.2.7.3 Personal Opinion 249
    - 8.2.7.4 Entity Tag for Data Protection 250
- 8.3. Eventual Consistency and Simultaneous Writing 250**
  - 8.3.1 Storage Classes 251**
- 8.4. Calling S3 251**
- 8.5. S3 REST API 252**
  - 8.5.1 Operations On Buckets 252**
  - 8.5.2 Operations On Objects 252**
- 8.6. Elastic Compute Cloud (EC2) 252**
- 8.7. DynamoDB 255**
  - 8.7.1 Data Storage and Retrieval 257**
  - 8.7.2 Primary Key 260**
  - 8.7.3 Secondary Key 261**
  - 8.7.4 Global Secondary Key 262**
  - 8.7.5 Local Secondary Key 262**
  - 8.7.6 Multiple Secondary Indices 262**
  - 8.7.7 DynamoDB Streams 262**