Amazon AWS Training

Cloud computing

Cloud

- Provides computing resources on-demand
    - Resources are rapidly provisioned
    - Resources are created with minimal management effort
    - „Unlimited“ capacity

- Several cloud types
    - Public cloud
    - Private cloud
    - Hybrid cloud

Cloud types

Public cloud	**Private cloud**	**Hybrid cloud**
+ Elasticity and unlimited capacity	**+** Better cost control	**+** Unlimited capacity and elasticity
+ No up-front investment needed	**+** In-house knowledge	**+** Balance between up-front investment and on-demand cost
+ Focus on core business	**+** Independent on third party	**+** Lower overall costs
+ Cloud provider‘s experience		
+ Economy of scale	**-** Off the shelf hardware	
+ Custom hardware	**-** Limited capacity	
	- Worse economy of scale	
- Higher costs	**-** Up-front investment	

Infrastructure? Platform? Service?

- IaaS
  - Infrastructure as a service
  - Basic computing resources
    - Virtual machines
    - Virtual networks
    - Virtual disk
  - In short, whatever you can do with physical hardware, you can usually do with IaaS as well

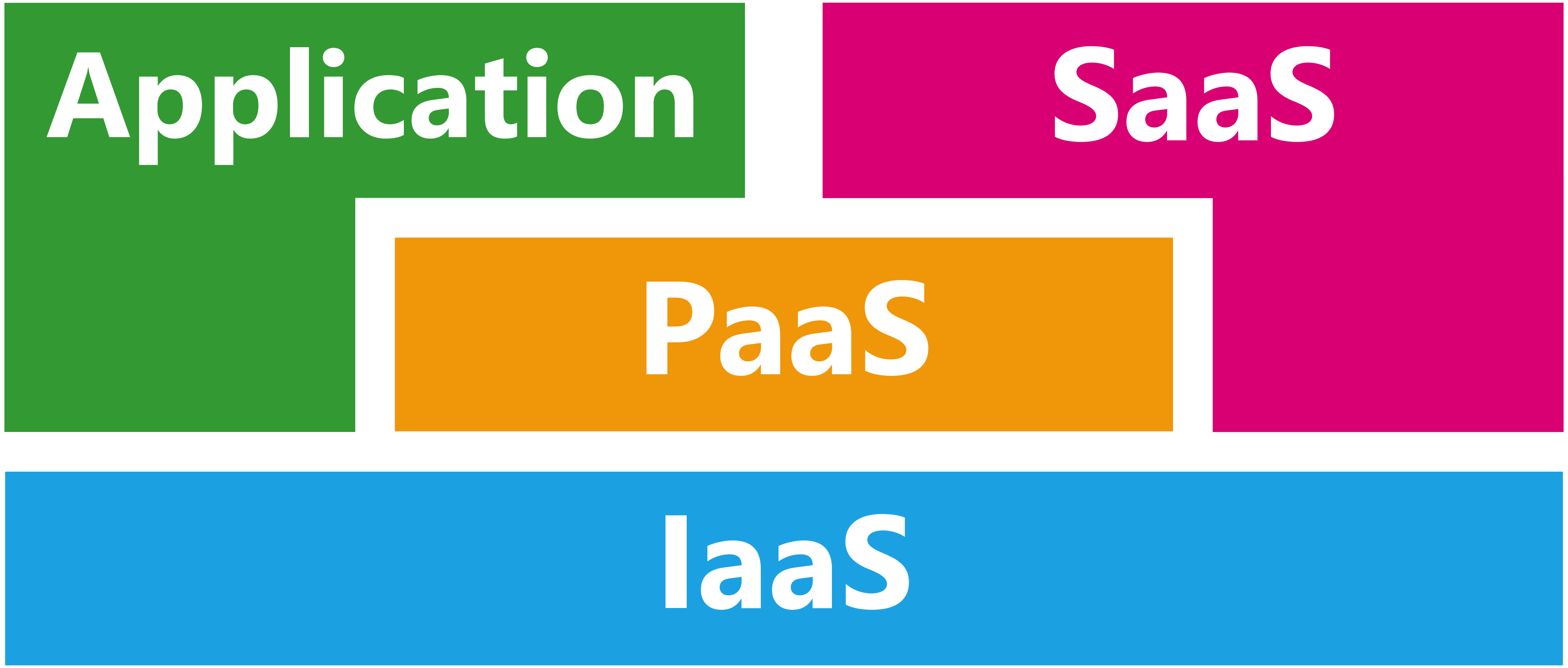
Infrastructure? Platform? Service?

- PaaS
  - Platform as a Service
  - Environment for deploying, orchestrating, managing and monitoring applications
  - Abstracts from used infrastructure
    - Developer doesn‘t care about the used virtual machines, the platform takes care
  - For example: Kubernetes, Cloud Foundry

Infrastructure? Platform? Service?

- SaaS
  - Software as a Service
  - User doesn‘t care about how and where the software runs
  - Provider is responsible for managing the infrastructure, installing updates etc.
  - Example:
    - User requests MySQL database with 100GB capacity and performance of 1000 ops/s
    - User gets username / password and hostname / port
    - Connects to the database and uses it
    - User doesn‘t care where the database is running, it simply runs for him

Infrastructure? Platform? Service?



Application

SaaS

PaaS

IaaS

Cloud native applications

- Exploit the advantages of running in cloud
    - Scalability
    - Density
    - Agility

- Microservices
    - Many small services with limited functionality
    - Loosely coupled through APIs (HTTP, AMQP)
    - Independent on each other (technically but also personally)

Cloud native applications

- DevOps
  - Operators developing and developers operating
  - Tasks being done manually in the past are more automated
- Containers
  - Build once, run everywhere
  - Help to increase the compute density
- Continuous Integration / Continuous Deployment
  - Many releases per day, each automatically tested and deployed

Cloud native applications

- Horizontal vs. vertical scalability
  - Run in more instances instead of run single bigger instance

- Stateless applications
  - State is causing problems
  - State is concentrated in single place (e.g. in database server)
  - Leads to rapid recovery

Cloud providers

- AWS is not the only cloud provider
  - Microsoft Azure and Google Cloud are the main competitors
  - Many smaller providers
- Major providers provide very similar services
  - There are no standards and no cross provider compatibility
  - Moving from one provider to another is possible, but is not effort less
  - Some services are proprietary and can create a lock-in!
  - Choosing an independent PaaS can prevent future problems, because the PaaS makes the applications independent on specific IaaS

Pricing

- Cloud is not cheap
  - To leverage the cost advantages, the applications have to be written for cloud

- Different price components
  - Per time unit (e.g. per hour of running virtual machine)
  - Per data transfer (e.g. per GB of transferred data)
  - Per operation (e.g. per million of DNS queries)
  - Careful: Amazon AWS bills per hour!

Amazon AWS

Amazon AWS in DBG

- *„Culture eats cloud for breakfast“* (Jakub Scholz ☺)
  - The fact that cloud allows something, doesn‘t mean that we will be able to use it
- Amazon AWS is organized into accounts
  - (not user accounts!)
  - Accounts can be interconnected
  - Sandbox account
  - Development account
  - Test account
  - Production account
  - Some additional special purpose accounts
  - Billing is summed up across all accounts and routed through I&O

Amazon AWS in DBG

- Sandbox account
  - Isolated from DBG networks
  - Access over Wi-Fi
  - Wide range of privileges / access rights

- Development accounts
  - Linked with DBG network
  - Access from our office computers
  - Unclear access rights

Tagging

- Most AWS resources can be tagged with one or more tags
- Some tags are considered mandatory by DBG to keep some order within the account
    - Name
    - Creator
    - Owner
    - CostCenter
- Used to split the cost bill
- Resources not matching these tags should be deleted

How to access AWS?

- Web console (login using SAML with DBG credentials and token)
  - Simple to use access to browser
  - Useful to try things for the first time or to setup a single machine
  - Not suitable for any serious deployments and CI/CD integration
- APIs
  - Access using access key, secret key and session token
  - Many different tools for scripting the deployments
    - AWS CLI tool
    - Ansible
    - Terraform
- The login credentials from SAML expire after 60 minutes
- Access to APIs and console is possible from DBG network. SSH access is not always possible.

Amazon AWS

Regions and Availability Zones

Regions and Availability zones

- AWS cloud is split into regions
  - Some regions are special (GovCloud, China)
  - Most regions are available to everyone
  - Prices differ per region
  - Each region has a name (us-east-1, eu-central-1)
- Every region is split into several Availability Zones (AZ)
  - Different regions have different number of Azs
  - Frankfurt has currently only 2
  - Most regions have 3 AZs, some have even more
  - One AZ is usually one or more datacenters
  - Each AZ has name based on its region (eu-west-1a, eu-west-1b, eu-west-1c)

Regions and Availability zones

- Not every service is available in every region
- Only few services are truly global (e.g. S3, Route 53, IAM)
- Most services are enclosed within a single region or even within one AZ
  - E.g. disk volumes or virtual machines are locked into their AZ
- Single AZ has no SLA
  - To build reliable applications, you have to span them across multiple AZs
  - Costs more effort than just starting one machine and having it run
- DBG preferred region is Frankfurt (eu-central-1)
  - Alternative region with 3 AZs is Ireland (eu-west-1)

Lab 1: AWS Web Console

- Login to the AWS Web Console
- Select the correct region
- Get familiar with the basic controls
- http://jsch.cz/awslab1

Amazon AWS

Identity management

Identity management

- Identity Management (IAM)
  - Manages the identity of users and resources
  - Users = People
  - Resources = Roles
  - Different resources can have different roles and make use of them
    - E.g. IAM role can be assigned to EC2 host and used by the software running on this host
  - As a part of software deployment, roles with required permissions should be created and assigned to resources
- IAM supports federated identities using SAML
  - Allows you to login with your DBG credentials

Amazon AWS

Infrastructure as a Service

VPC

- Virtual Private Cloud
- Your private virtual datacenter in AWS
    - Most IaaS resources which you create will be within VPC
- Every VPC has a CIDR
    - CIDR defines the range of IP addresses which you can use in your VPC
    - When using multiple VPCs, be careful so that the IP addresses don‘t collide with each other as that might cause problems
    - a.b.c.d/N
    - a.b.c.0/24 gives you 256 IP addresses (~254 instances)
    - a.b.0.0/16 gives you 65536 IP addresses
- VPC is region wide resource

Subnets

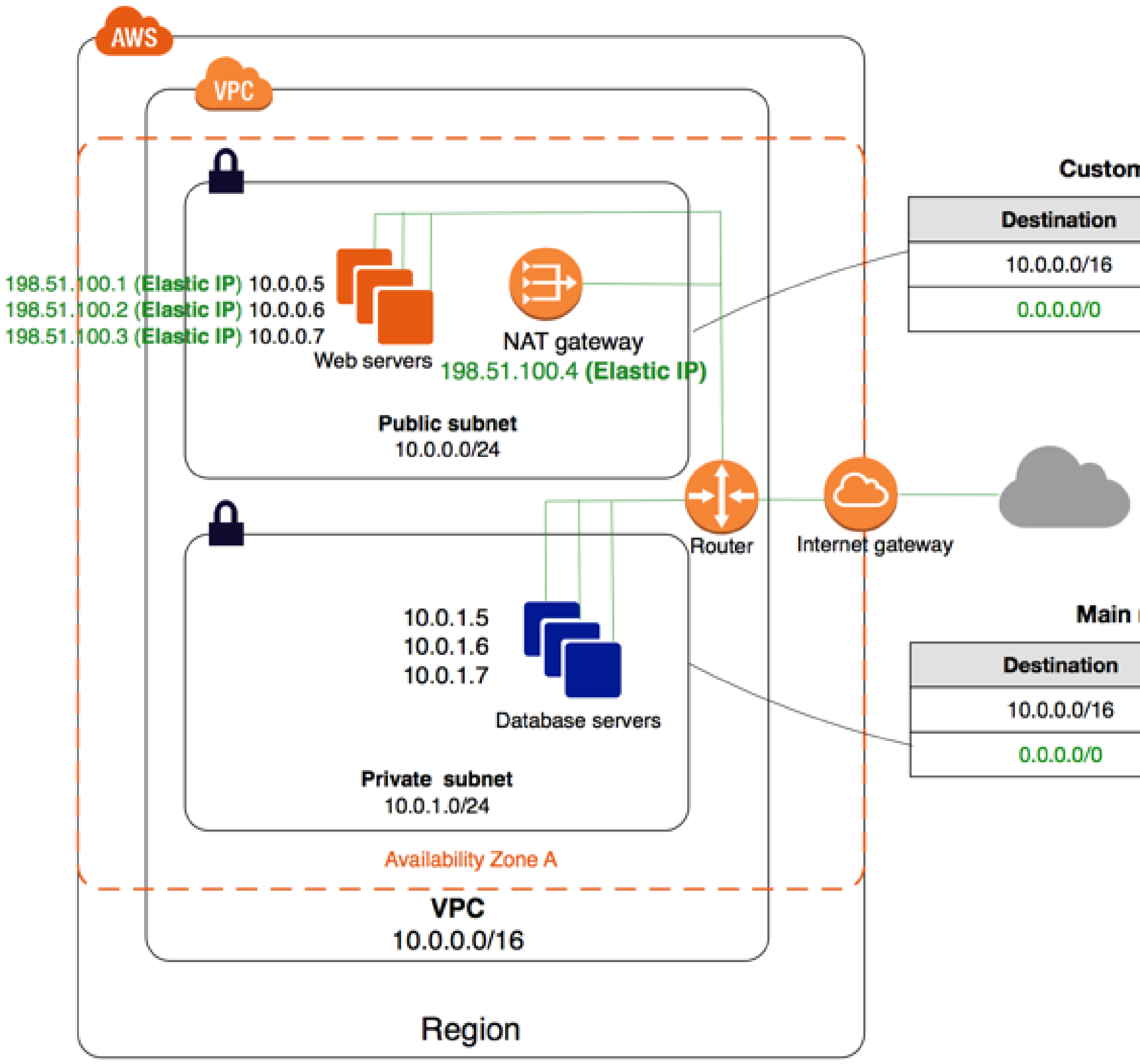
- Subnets are virtual networks in your datacenter
- Most resources you create later are in one or more subnets
- Subnets are linked with a Route Table which defines the routing in such subnet
  - Route Tables can be used to connect subnets to the outside (Internet) as well as to connect to other subnets
- Subnets always belong to single AZ

Internet Gateway / NAT Gateway

- Internet Gateway (IGW) gives your VPC access to internet
  - Without IGW, your VPC can communicate internally, but not externally (well, pretty much …)
  - Subnet with direct route to IGW is „Public“ subnet
  - Subnet without route to IGW is „Private“ subnet

- NAT Gateway protects your instances from direct internet access
  - Instances in private networks can get to internet through NAT
  - NAT gateway has to be placed in public subnet to route the trafic to internet
  - Private subnets can have a route to NAT gateway to be able to connect online (e.g. to download updates)

Internet Gateway / NAT Gateway



Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-gateway-id*

Other networking resources

- Network ACL
    - Can define access control on network level
- Elastic IPs
    - Reserved public IP addresses
    - For free when they are assigned to a resource
    - Paid when unassigned (the avoid misuse)
- VPC Peering
- VPN Connections and Direct connections

Lab 2: VPC and Networking

- Create a VPC
- Create Internet Gateway and NAT
- Create public and private subnets in at least two AZs and connect them to internet
- http://jsch.cz/awslab2

Storage

- Elastic Block Storage (EBS)
  - Different types of volumes
  - SSD disks
  - IOPS disks
  - Disks are limited to single AZ
  - Disks can be encrypted
  - Some instances are optimized for EBS performance
- Snapshots
  - Snapshot copies of EBS volumes
  - Can be used as backups
  - Can be created from volumes / volumes can be created from snapshots

Storage

- Elastic File System (EFS)
  - Basically NFS based file system
  - Mirrored across AZs within one region
  - Performance is not as good as with SSDs

Compute

- EC2 instances are the main compute resource
- Virtual machines
  - Different types
  - Memory optimizes, CPU optimized, Disk optimized, GPU, FPGA, …
  - T2 instances
    - Have burstable CPU
    - Each instance has some credits per time unit which it can consume
    - T2.micro is usually good starting point for some playtime
- EC2 instances can be stopped or restarted from the console
- Pricing
  - Per hour
  - On-demand vs. Reserved vs. Spot vs. Dedicated
  - Price is independent on utilization!

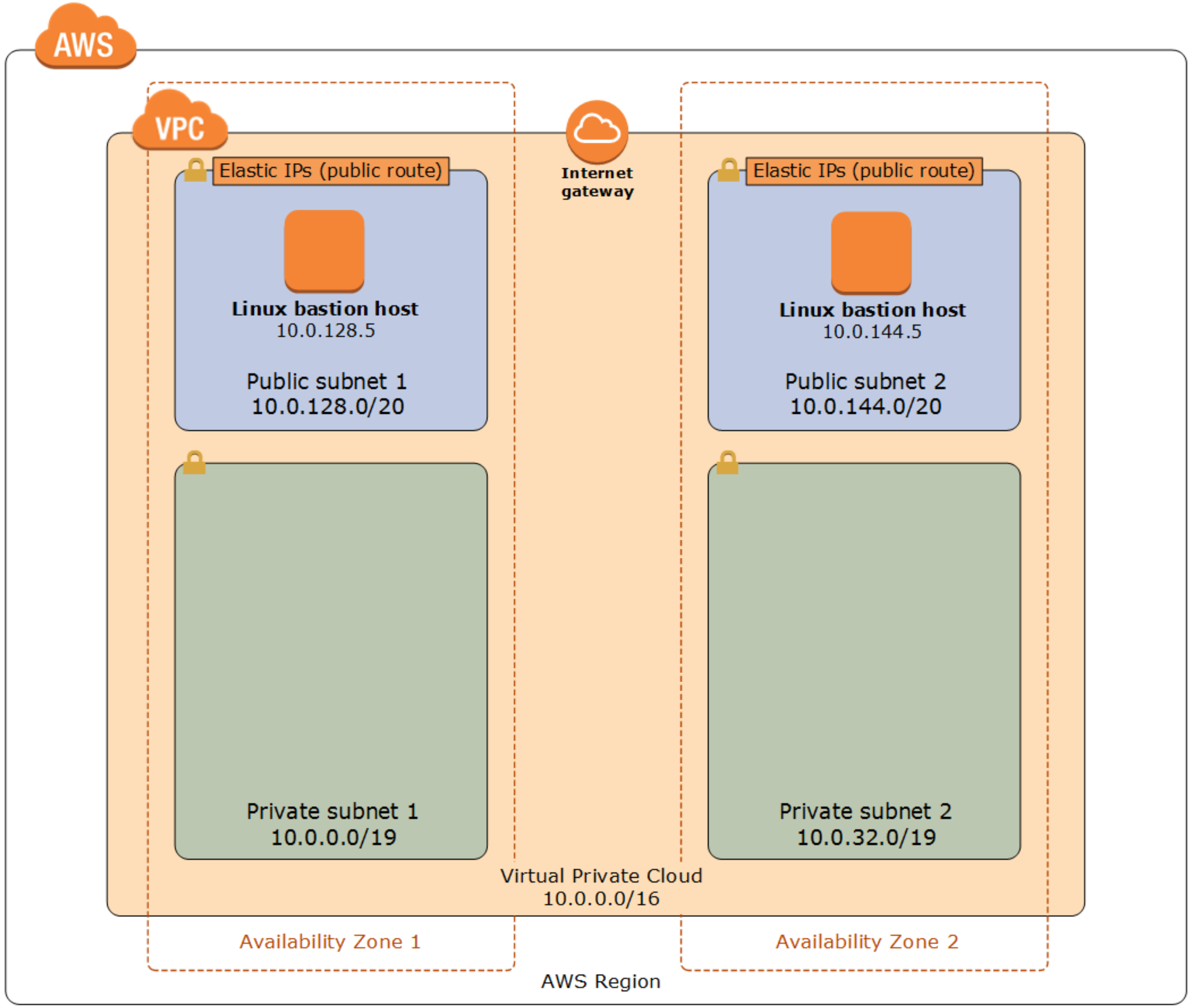
Compute

- Amazon Machine Images (AMI)
  - Volume snapshots with operating system
  - Are used to create new machine with specific operating system
  - Images available from Amazon or from Market place
  - Custom images can be created from existing volumes / hosts
  - Some images create more expensive instances (e.g. Red Hat Enterprise Linux)

Security Groups

- Security Groups (SG) are software firewall provided by Amazon
  - Instances are assigned into a security group
  - Security group defines which ports on the machine are opened to access from outside
    - SSH on node 22
    - HTTP(S) on 80 and 443

Security Groups



Lab 3: Create a Webserver

- Create a new EC2 instance in your public subnet
- Install a Nginx webserver on it
- Check that you can access the webserver
- http://jsch.cz/awslab3

Auto-Scaling

- Auto scaling is controlled using Auto Scaling Groups (ASG)
  - Specify desired, maximum and minimum number of running instances
  - Can be used to automatically start new machine when the old one crashed
    - Desired=1, Min=1, Max=1
  - Can be used to scale the resources
    - CPU utilization is the basic factor
    - Other factors can be build in (Memory, custom metrics, etc.)
  - Can scale machines across AZs
    - When one AZ goes down, can be used to automatically start a new machine in other AZ
- Bootstrapping
  - ASGs use Launch Configurations to start new instances
  - Custom AMI images
  - User data code

Load balancing

- Elastic Load Balancing (ELB)
  - Allows to route and balance traffic between multiple instances
  - Can terminate SSL connections
  - Classic load balancers
    - Support TCP/TLS or HTTP(S) routing
    - For HTTP, they support sticky sessions, but cannot route according to URL
  - Application load balancers
    - Support only HTTP(S)
    - Support more advanced routing
      - E.g. myshop.com/orders/* goes to instance A, myshop.com/items/* goes to instance B
  - Always use the DNS name, the IP address of ELB can change

Lab 4: Create an ASG with webservers

- Create an AMI image from your webserver from previous Lab
- Use the AMI image with an ASG which starts your webservers in your private subnet
- Use load balancer to balance the traffic between different webservers
- Check that when you terminate one of the servers, a new one will be spawned by ASG
- http://jsch.cz/awslab4

Amazon AWS

Managed services

Managed services

- Amazon AWS has many different managed services
  - Some of them are just packaged open source tools
  - Other are proprietary solutions
  - Careful with lock in
- Areas
  - Storage
  - Databases
  - PaaS
  - Messaging
  - Connectivity
  - Development

Storage

- S3
  - Virtually unlimited space for object storage
  - Objects (files) are stored in S3 buckets
  - Within the buckets, folder structure can be created and files can be uploaded into it
  - Objects are accessible through HTTP
  - Policies can be associated with buckets
    - Is it free to access by everyone? Or only by authenticated users? Or only internally?
  - Can be used to host complete websites
  - Considered usually as the cheapest way how to store and distribute files
  - Supports versioning of files

Storage

- Glacier
    - Cold data storage
    - Cheaper than S3
    - Slower access / bigger latency
    - Files from S3 which are not being accessed that often can be automatically offloaded into Glacier

Databases

- AWS has several DB offerings
  - RDS
    - Allows to easily start different databases (MySQL, PostgreSQL, Oracle, MSSQL, MariaDB)
  - Aurora
    - Amazon‘s own SQL engine with MySQL and PostgreSQL interface compatibility
    - Fully managed service
  - Dynamo
    - NoSQL database
    - Document store, Graph database, Key-Value store
  - Redshift
    - Proprietary SQL database based on very old PostgreSQL database
  - ElasticCache
    - Caching engine based on Memcached and/or Redis

PaaS

- Elastic Beanstalk
  - PaaS which allows to run applications in different languages
  - Java, .NET, Python, Node.js, Ruby, PHP, Go, Docker

- EC2 Container Service
  - Allows to run Docker containers on a fleet of EC2 instances
  - Different containers can have different roles etc.

PaaS

- Lambda
  - Serverless programming
  - Small functions which are triggered from outside (cron job, HTTP request, Messaging)
  - Paid per runtime
  - Can be used for automation tasks but also to construct bigger applications
  - Carefull about latency issues before the Lambda function starts

Messaging

- SNS / SQS
  - Amazon‘s proprietary messaging service
  - SQS = Simple Queue Service
  - SNS = Simple Notification service
  - Compared to something like AMQP very primitive
  - But very well integrated with other Amazon services (e.g. to trigger Lambda function per message etc.)

Connectivity

- Route53
  - DNS service
  - Can host public or private zones and register domains
  - Can be used independent on other Amazon services (a DNS hosting for services running elsewhere)
  - Good integration with other AWS services (e.g. ELB etc.)

- CloudFront CDN
- API Gateway

And lot more

- Development
- Analytics
- Go to „Services“ in the web console to see all services

Lab 5: S3 & Lambda

- Create new S3 bucket
- Create new Lambda function which is triggered by changes to files in the S3 bucket
- http://jsch.cz/awslab5

Amazon AWS

Monitoring

Cloudwatch

- Monitoring tool for Cloud resources
  - Create dashboards with charts
  - Create alerts (can be linked with Auto Scaling Groups)
  - By default the metrics are received every 5 minutes (every minute as paid service)
  - The default offering of metrics is limited for EC2 hosts
    - By default, Amazon doesn‘t have any information about your VM apart from CPU consumption
    - Additional metrics like memory consumption have to be provided by agent running inside of the VM

Other tools

- CloudTrail
    - Audit log for the activates happening in the Amazon account
- CostAdvisor
    - Can help to optimize costs (e.g. recommend better instances to be used etc.)

Lab 6: Delete all resources which you created

- Delete all resources which you created during the labs
- http://jsch.cz/awslab6

https://github.com/scholzj/dbg-aws-training