Cloud native technologies and cloud based services development and management

Dr. Yuri Demchenko
University of Amsterdam

SLICES-SC Summer School
Open-RAN/Core/Edge Solutions for Cloud-Native Telco Experimental Platforms
July 2022, Volos, Greece

Outline

- Cloud Computing and impact om telecom market
- Cloud native vs Cloud based
- NIST Cloud Computing definition and Reference Model
- General cloud use cases and usage scenarios
- Cloud enabling technologies: Virtualisation and Containers
  - Kubernetes architecture and components
- Summary and take away

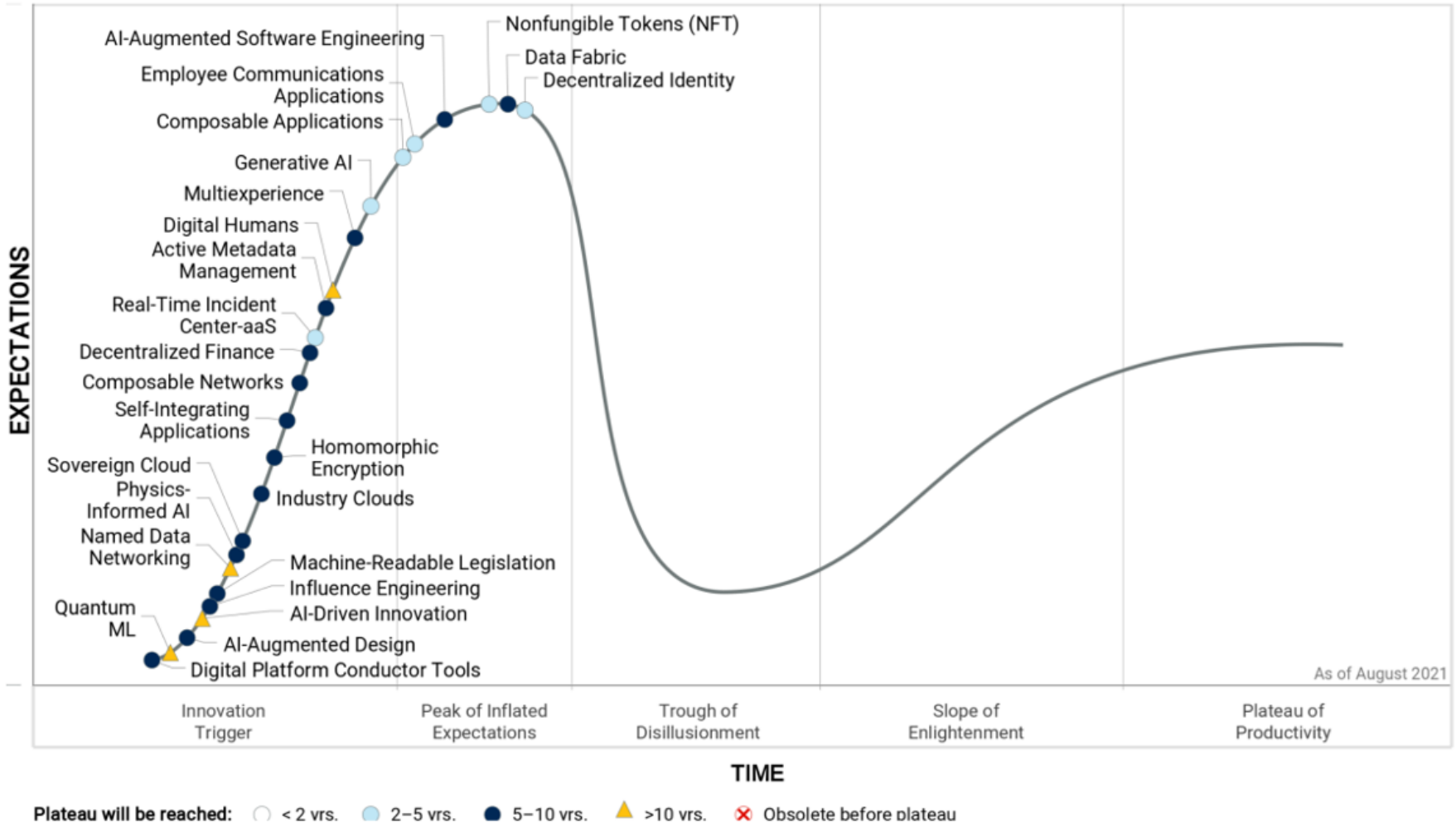
- Additional information: Major cloud providers: AWS, Microsoft Azure, Google Cloud Platform (GCP)

- Practice: Getting starting with AWS cloud (optional and self-study)

- Tutorial and practice materials
  https://drive.google.com/drive/folders/1nCqA0n51bS_98ACUudrAeqM468aZl7hq?usp=sharing

Hype Cycle for Emerging Technologies, 2021

https://www.gartner.com/en/newsroom/press-releases/2021-08-23-gartner-identifies-key-emerging-technologies-spurring-innovation-through-trust-growth-and-change

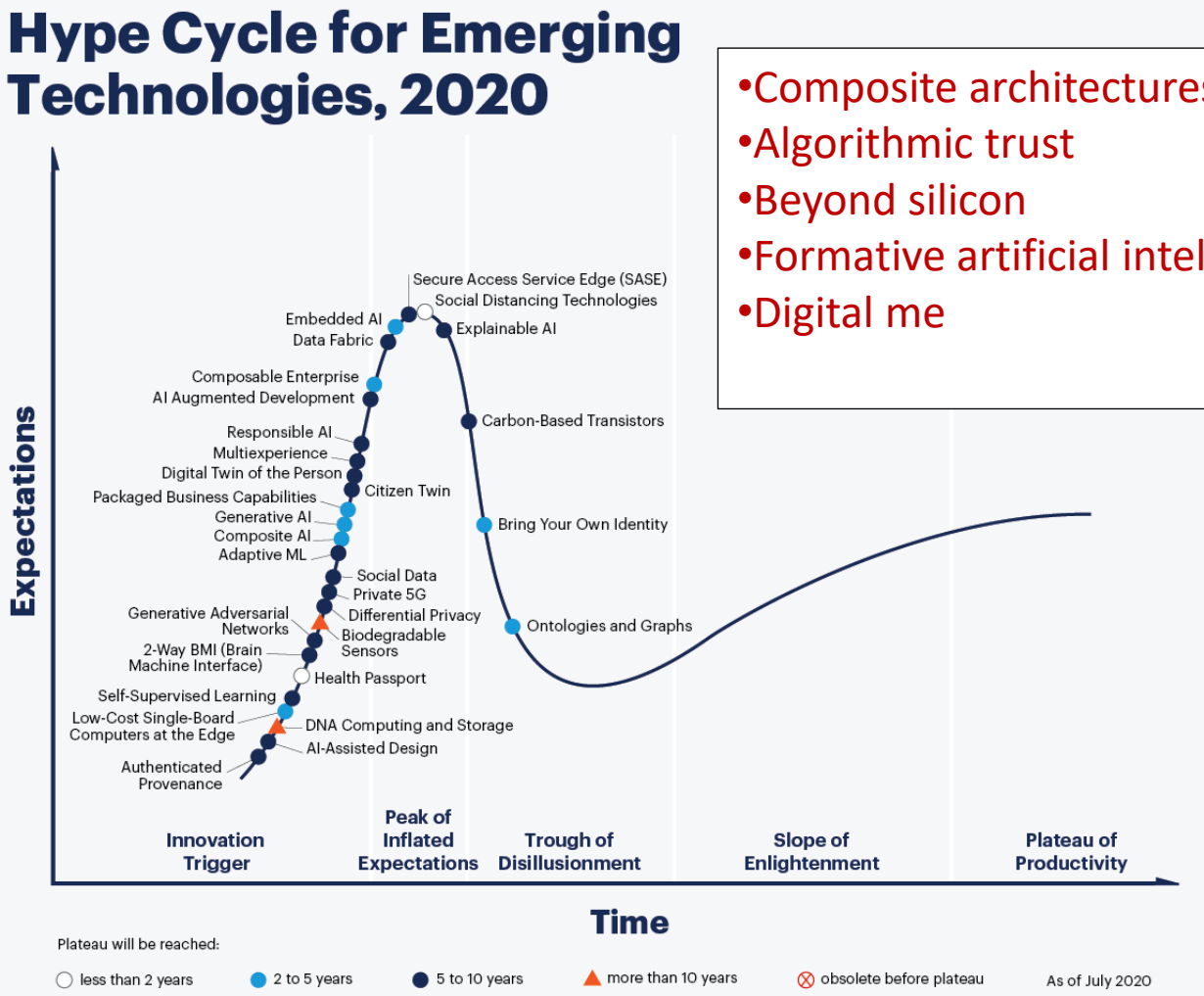


Source: Gartner (August 2021)

747576

- Composable Networks
- Composable Applications
- Self integrating applications
- Active metadata management
- Named Data Networks
- Sovereign Cloud & Industry Cloud

Gartner Technology Hypercycle (2020)



- Composite architectures
- Algorithmic trust
- Beyond silicon
- Formative artificial intelligence (AI)
- Digital me

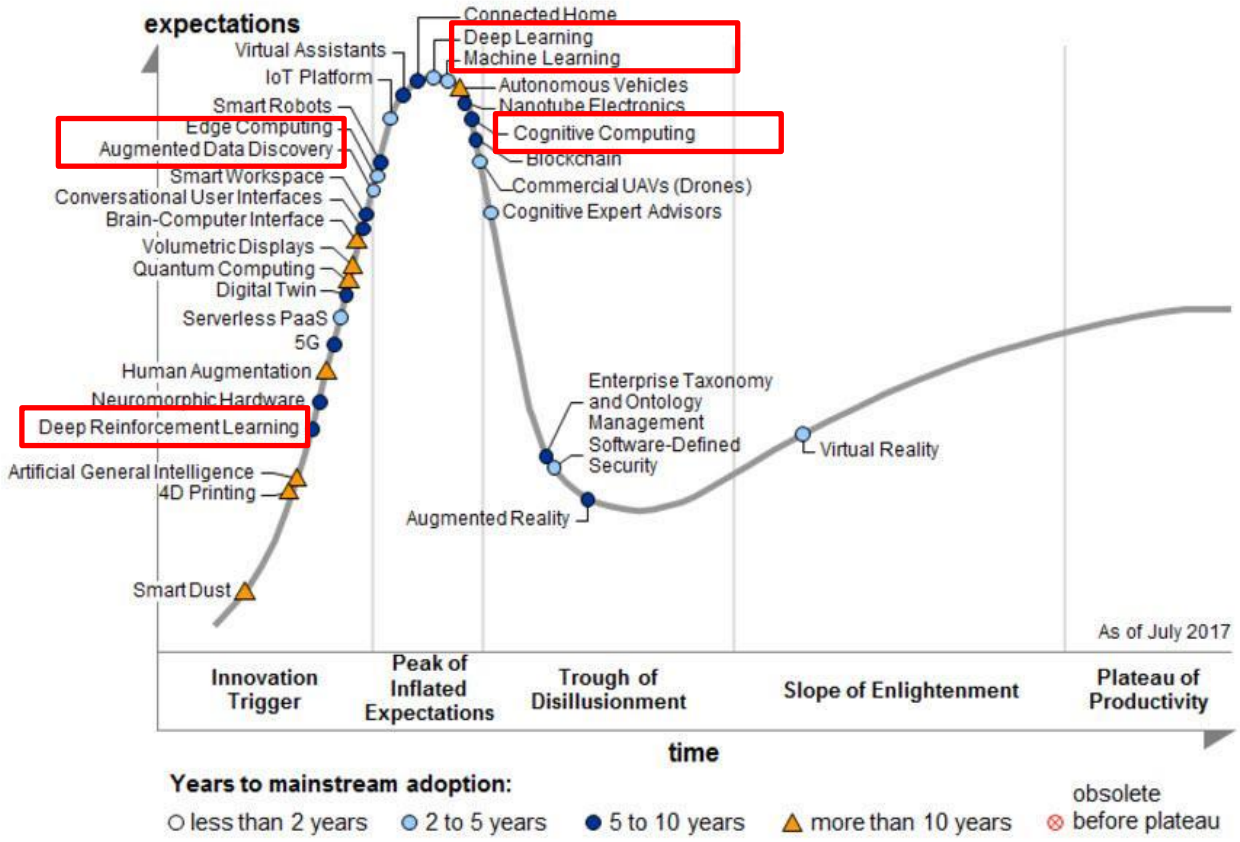
Value of the Gartner hypercycle: Must know technologies

- Compare trends with previous years 2013 and 2017
- Commonly used infrastructure technologies for modern services and applications: Cloud Computing (including Edge Computing) and Big Data are
- Growing use of data driven technologies and powered by AI/ML
- Infrastructure technologies are incorporating IoT and 5G networks

Gartner Technology Hypercycle (August 2017)

Hype Cycle for Emerging Technologies, 2017

We are in post Big Data and post Cloud Computing stage



Note: PaaS = platform as a service; UAVs = unmanned aerial vehicles

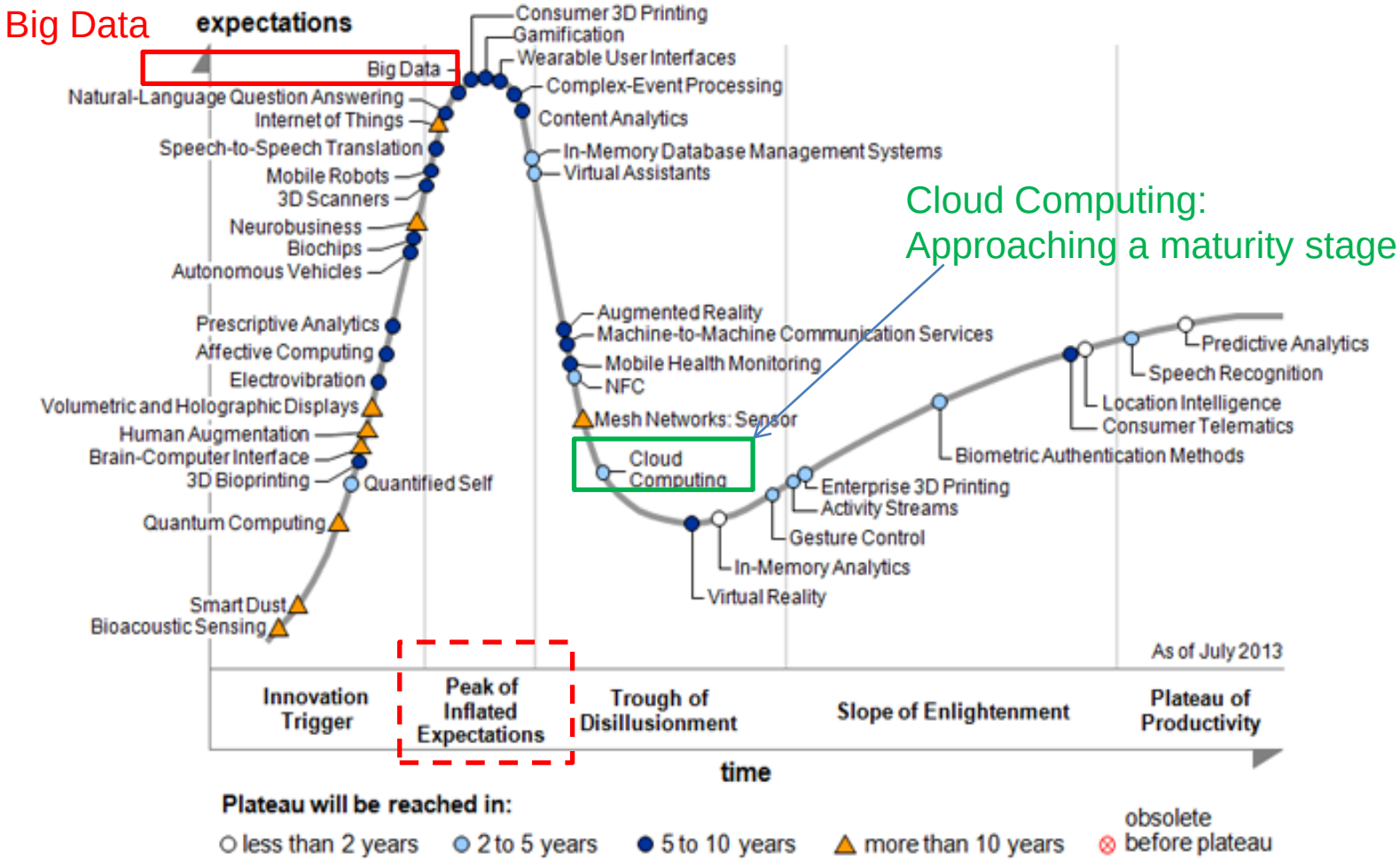
Source: Gartner (July 2017)

Big Data and Cloud Computing:
In the maturity stage –
already commodity services

- New trends are extending reach of clouds and their use for data driven applications
- Role of IoT and expectations are increasing
- 5G is entering into experimentation stage promising to link sensor network and cloud

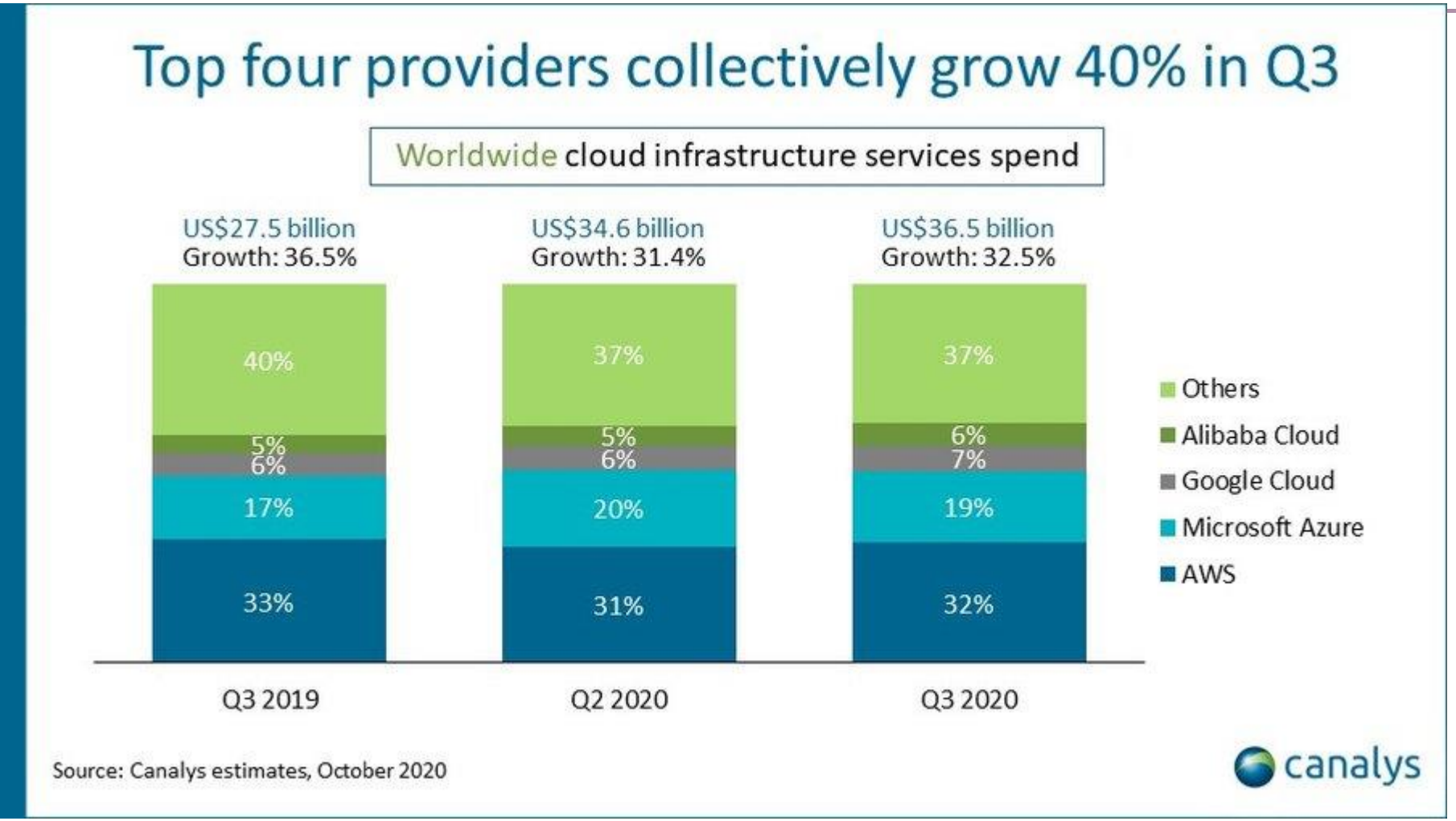
ıer-hype-cycle-for-emerging-technologies-2017/

Gartner Technology Hypercycle (August 2013)



Source http://www.gartner.com/newsroom/id/2575515
http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp

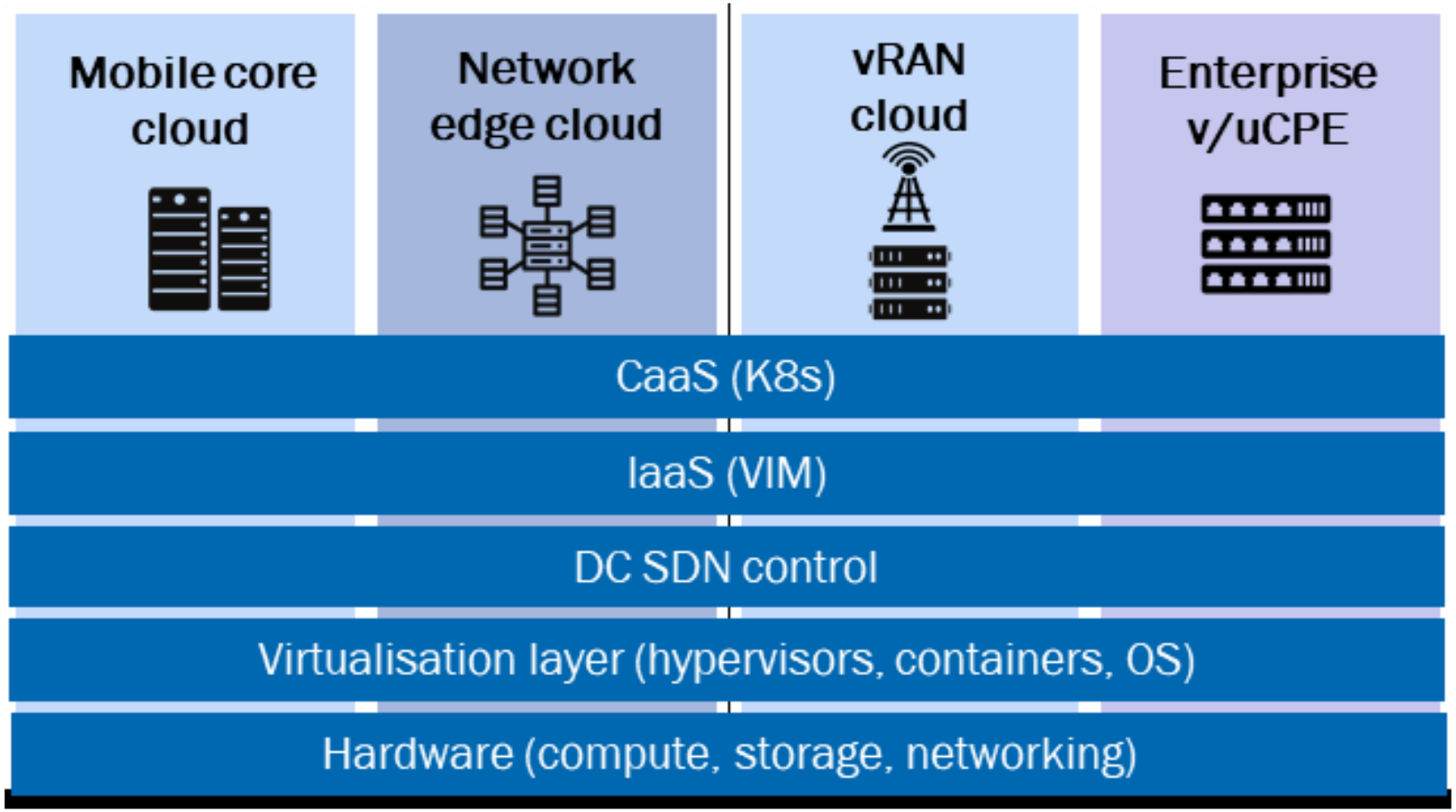
Cloud Services Market Trends
https://www.canalys.com/newsroom/worldwide-cloud-market-q320



- As of October 2020, [Canalys reports that the worldwide cloud market grew 33% this quarter to $36.5 billion.]
- AWS has 32% of the market and generated more revenue than the next three largest **combined**,
- Azure is at 19% of the market,
- Google Cloud at 7%,
- Alibaba Cloud close behind at 6%,
- other clouds with 37%

Cloud technologies power telecom market development

https://www.grandviewresearch.com/industry-analysis/global-telecom-services-market



- The global telecom services market size was valued at USD 1.74 trillion in 2019 and is expected to grow at a compound annual growth rate (CAGR) of 5.0% from 2020 to 2027.
- CSP spending on multi-cloud network infrastructure software, hardware and professional services will grow from USD4.3 billion in 2019 to **USD32 billion by 2025**, at a CAGR of 40%.
- Rising spending on wireless communication infrastructures due to the shift in customer inclination towards cloud-based technology and mobile devices is one of the key factors driving this industry.
- Cloudification of RAN
- vRAN will be the fastest-growing cloud domain in terms of CSP investments. Spending in this domain will grow at a CAGR of 132% to USD11 billion by 2025

Telecommunications operators are expected to cumulatively spend **$114 billion on network cloud** between 2019 and 2025, according to Analysys Mason.

Cloud Computing as a key IT technology development factor

- Cloud Computing has entered a maturing stage and currently a commodity services
- Cloud Computing is powering modern business and powering new technologies development that require elastic computing resources on-demand
  - Mobile applications
  - Big Data applications
  - Internet of Things (IoT)
  - Changes telecom market landscape
- In turn, other technologies demand accelerates Cloud Computing development
- Cloud Computing increases **business agility** and speeds up new services/products development to market
  - However still restrained by security concerns on business data protection

Cloud-native vs Cloud-based vs Cloud-enabled

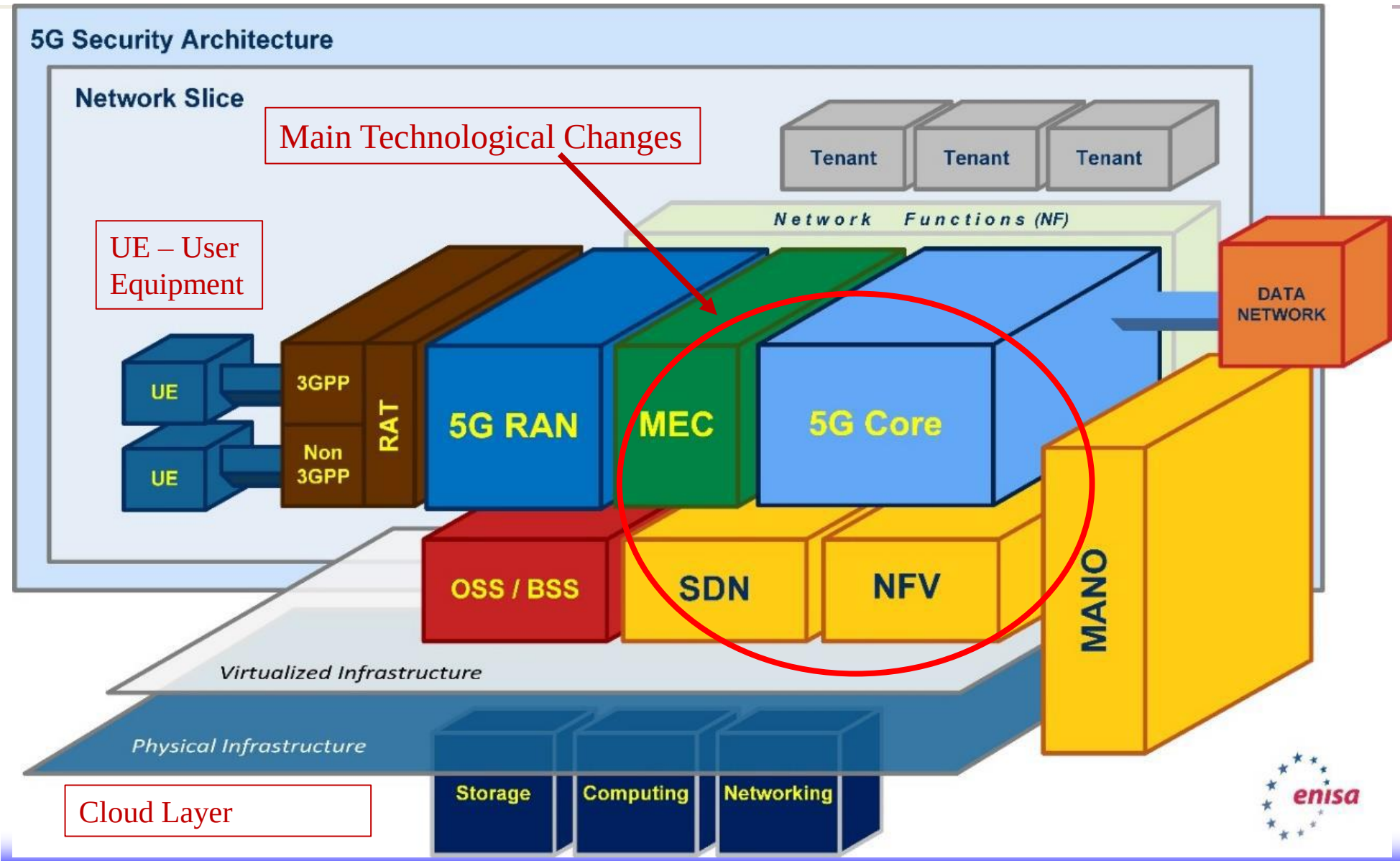
https://www.papertrail.com/solution/tips/cloud-based-cloud-native-and-cloud-enabled-applications-whats-the-difference/

- **Cloud-Native**
  - Cloud-native is born in the cloud. Cloud-native applications are architected from the ground up to run in a public cloud like in AWS, Azure, or GCP using cloud-based technologies. These cloud technologies allow for **accessibility and scalability** and allow developers to continue to **deliver new services more quickly and easily**. Cloud-native is comprised of **continuous integration, orchestrators, and container engines**.
  - Cloud-native is a new way of architecting our applications and infrastructure: **Services are broken into smaller and smaller pieces and reusing services wherever possible.** However, deployment in cloud provider data center relies on Services Level Agreement, so failure to be anticipated and resilient to be ensured.
- **Cloud-Based**
  - Cloud-based is the middle ground between cloud-native and cloud-enabled. **Applications are moved to cloud and re-designed to benefit from higher availability and scalability.** For example, if you move your in-house web application to AWS or Azure servers, you now have a “cloud-based” application. Benefit is that infrastructure and hosting environment are outsourced to cloud provider.
- **Cloud-Enabled**
  - Cloud-enabled usually refers to **applications built traditionally and then migrated to the cloud.** These applications were originally designed in a monolithic fashion and refactored to use virtual resources, but the underlying architecture remains the same. The application does not take advantage of shared services or resource pools.
  - Cloud enabled can be an approach for legacy applications or as the first step towards cloud adoption.

Designing Cloud Native Applications with Microservices and Containers on OpenStack cloud platform

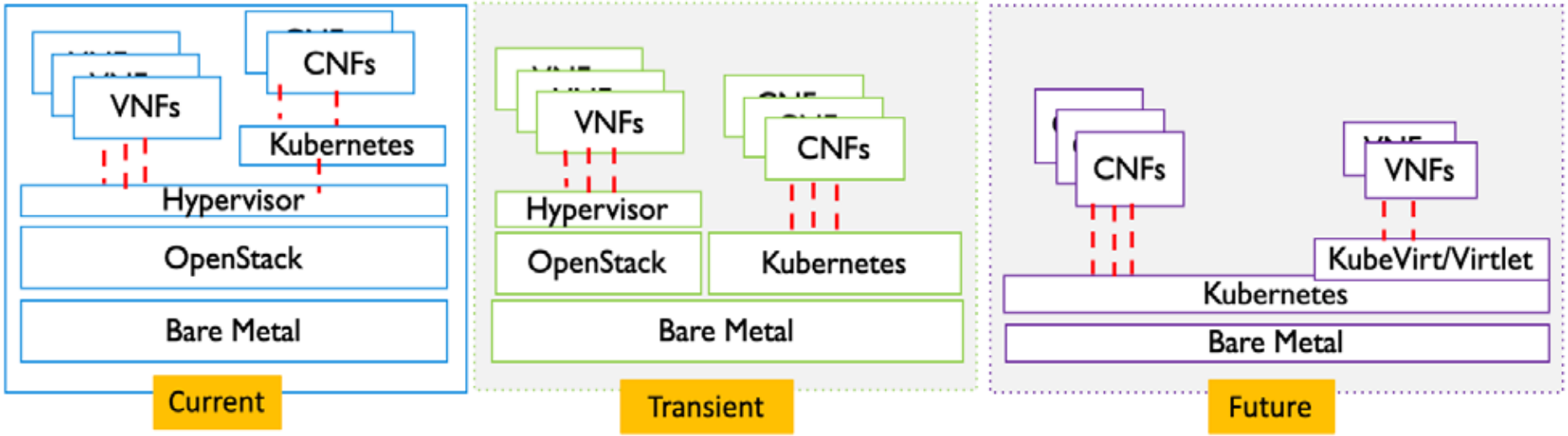
- Cloud native refers to an application designed, implemented and operated using cloud computing principles, technologies and architecture frameworks such as
  - Cloud Service Models and Cloud Computing Patterns,
  - Microservices, containers and dynamic scheduling.
- Cloud native with Openstack as the underlay Cloud Infrastructure, opens a new way possibilities when it comes to developing Cloud Native Apps.
  - Cloud Patterns as the framework for the application
  - Microservices used to design, build and test the application and how containers, container management technologies and Openstack projects for Containers, are used as the underlay infrastructure.

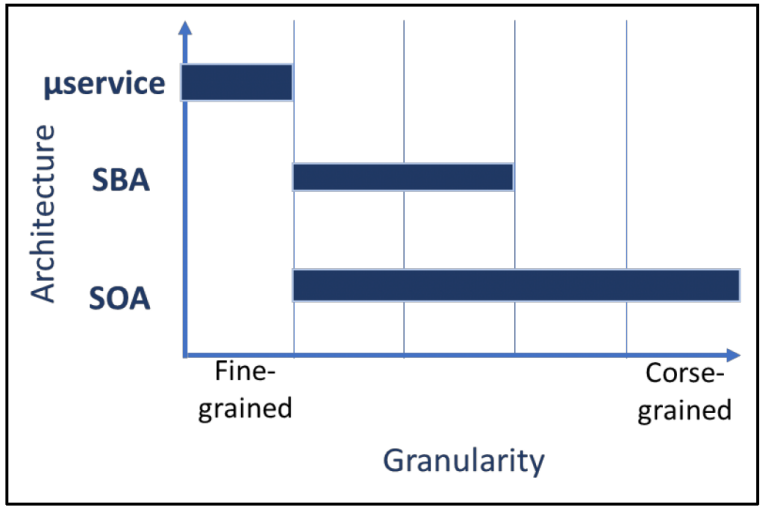
5G High-level Technical Architecture (ref. ENISA)



- Cloud Native in 5G Architecture

Evolution to Cloud Native Ecosystem

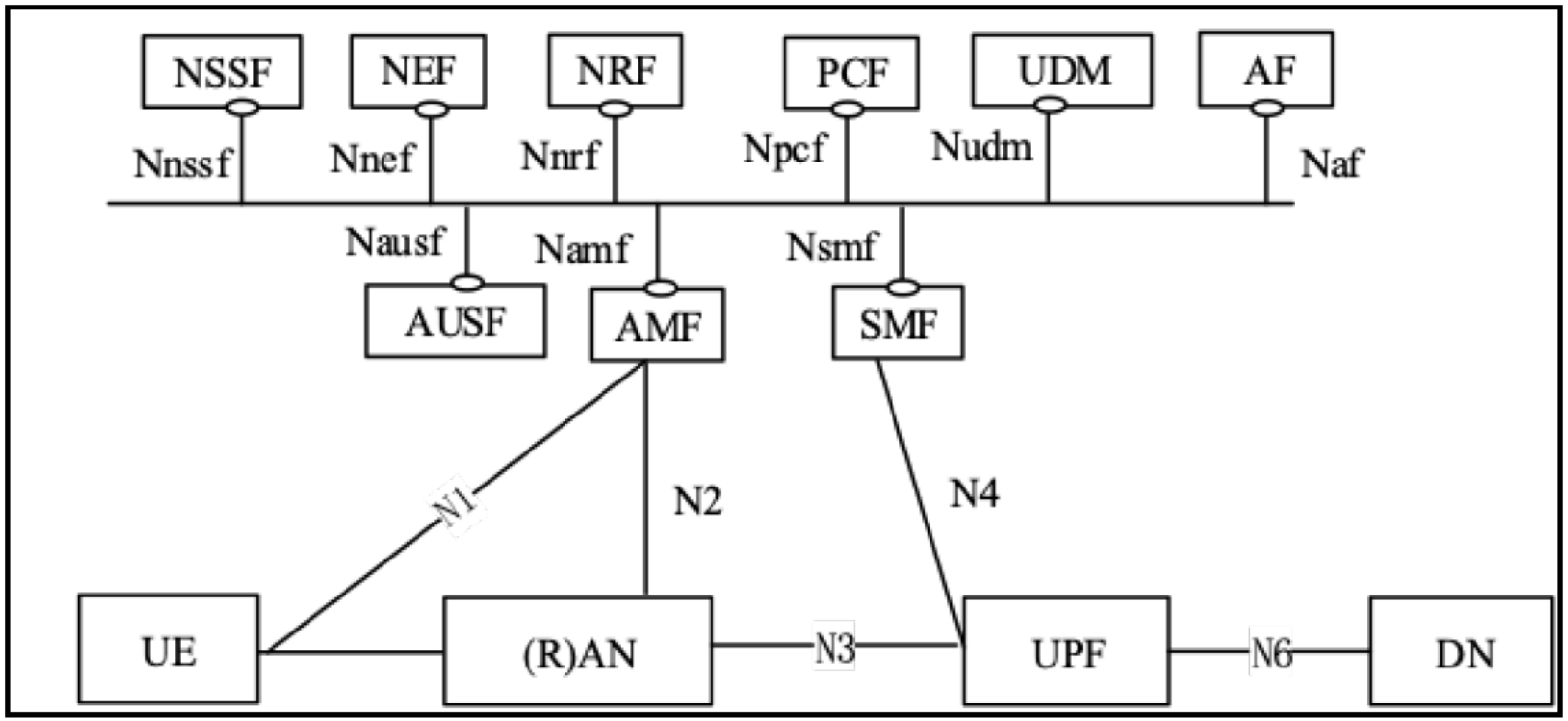




- Cloud Native: SBA vs SOA
  - Focus on microservices
  - RESTful interface
- SBA - Service Based Architecture
- SOA – Service Oriented Architecture
- Kubernetes platform still to advance on network virtualization

[ref] A 5G Americas White Paper. 5G and the Cloud, Dec 2019

5G-3GPP Architecture based on SBA Functions



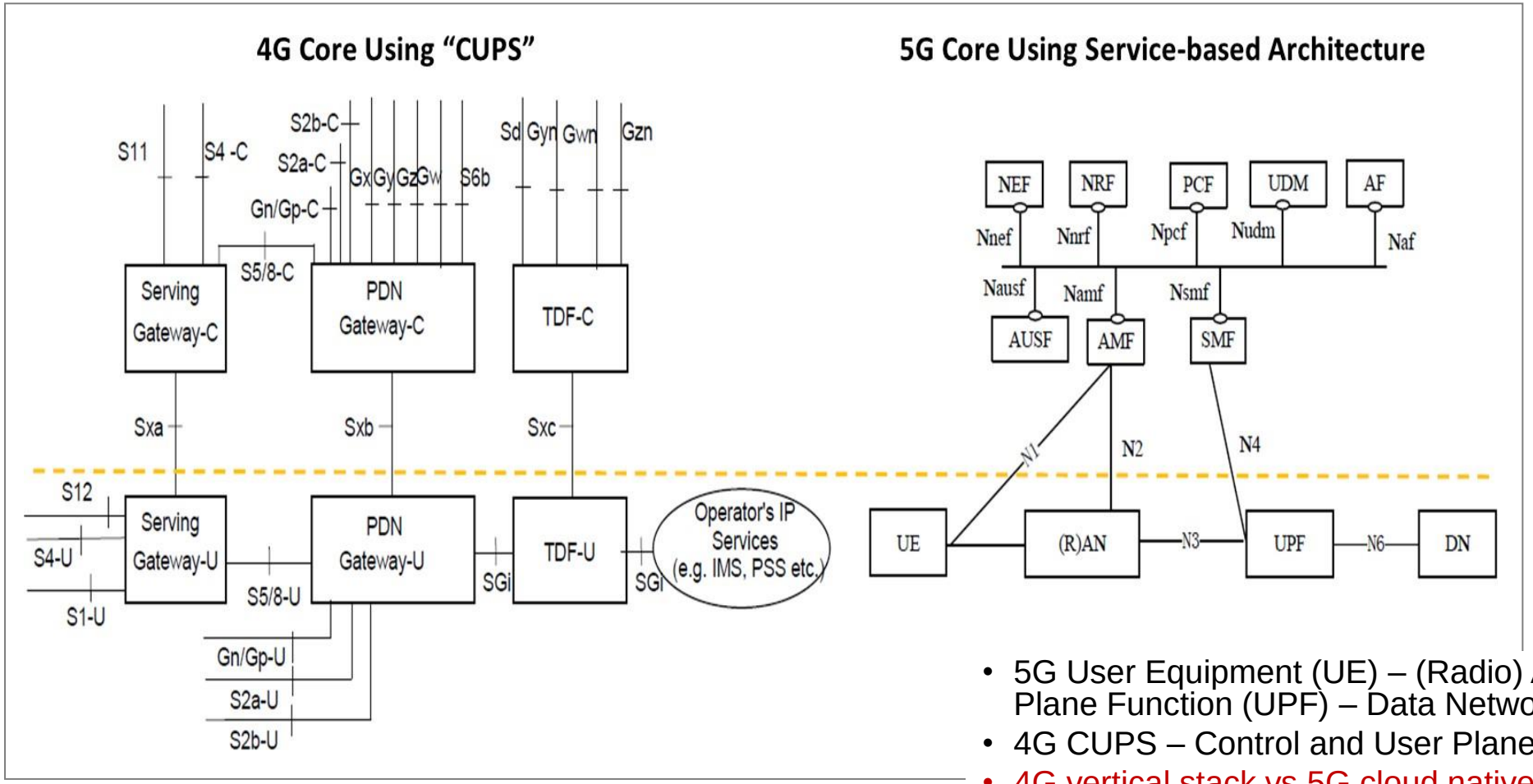
- User Equipment (UE) – (Radio) Access Network (RAN) – User Plane Function (UPF) – Data Network (DN)
- AUSF – Authentication Server Function
- AMF – Access and Mobility Management
- SMF – Session Management Function

- NSSF – Network Slice Selection Function
  - This module is responsible for selecting the 5G network slice instance serving User Equipment (UE) and which AMF, or list of AMFs, can be used by a device (UE)

- NEF – Network Exposure Function
- NRF – Network Resource Function
- PCF – Policy Control Function
- UDM – Unified Data Manager
- AF – Application Function(s)

[ref] A 5G Americas White Paper. 5G and the Cloud, Dec 2019

Comparison 4G and 5G SBA Model



- 5G User Equipment (UE) – (Radio) Access Network (RAN) – User Plane Function (UPF) – Data Network (DN)
- 4G CUPS – Control and User Plane Separation
- 4G vertical stack vs 5G cloud native layering model

[ref] Service Based Architecture for 5G Core Network, Heavy Reading White Paper, Sponsored by Huawei, 2017

VMs vs Containers: Test by Cloud Native Computing Foundation (CNCF, 2020)

	OpenStack	Kubernetes
Infra deploy time	~65 minutes	16 minutes*
NF deploy time	3 minutes, 39 seconds	< 30 seconds
Idle state RAM	17.8%	5.7%
Idle state CPU	7.2%	0.1%
Runtime NF RAM	17.9%	10.7%
Runtime NF CPU	28.8%	39.1%
Snake case PPS	3.97 million PPS	4.93 million PPS
Snake case latency	~2.1 milliseconds	~2.1 milliseconds
Pipeline case PPS	N/A	7.04 million PPS

https://docs.google.com/presentation/d/1nsPINvxQwZZR_7E4mAzr-50eFCBhbCHsmik6DI_yFA0/edit#slide=id.g5036f143e9_3_672

Benefits Cloud Native Approach: Re-usable/Composable Network Components

- Service Based Architecture
  - Vs Service Oriented Architecture
- Cloud Native approach and tools
  - Including DevOps
- VMs vs Containers
  - Noisy neighbor factor is known in Kubernetes
  - But network infrastructure has known type of workload

Cloud Computing - Main

- Cloud Computing definition by NIST
- NIST Cloud Computing Reference Architecture
- Use cases

What is Cloud Computing

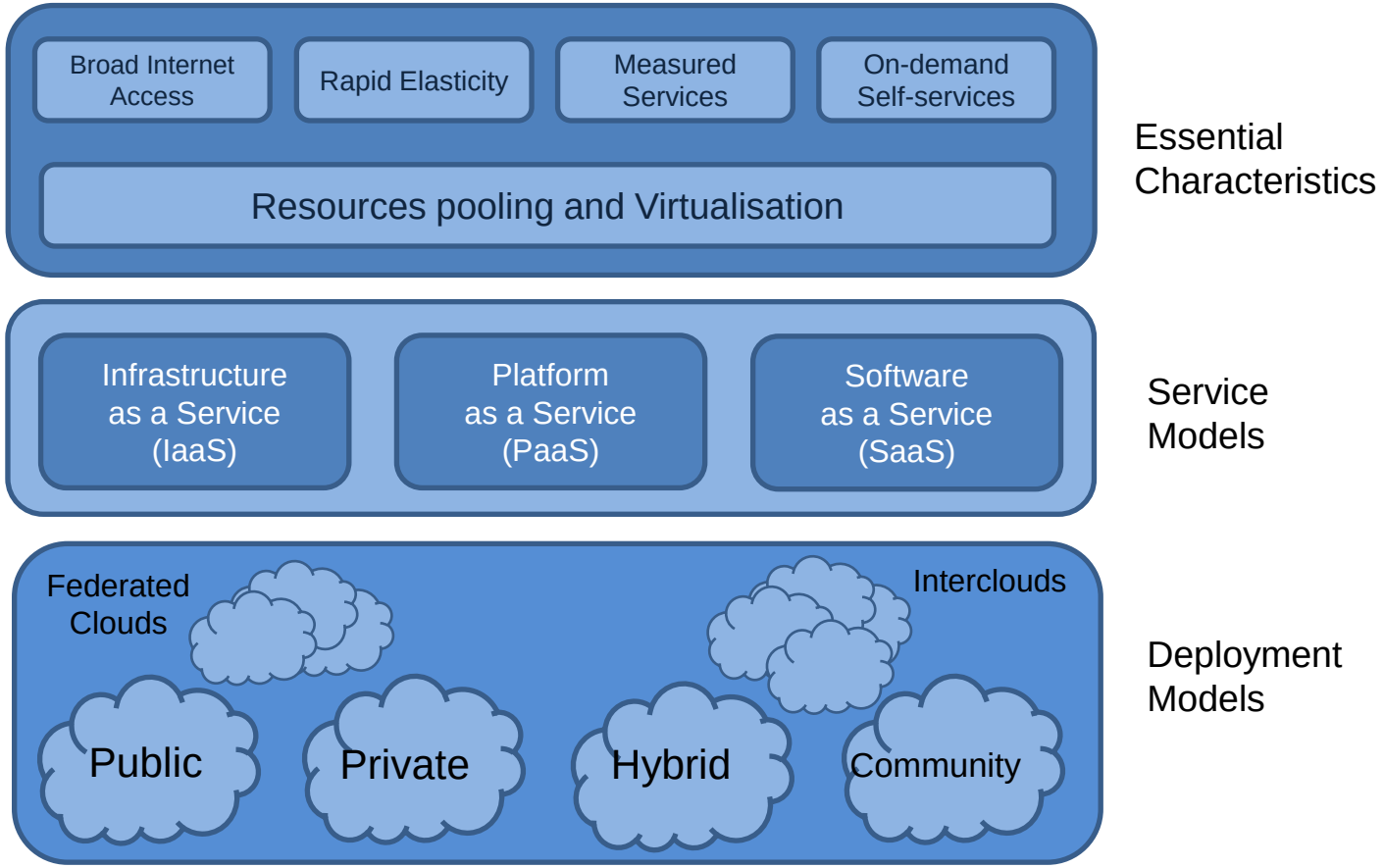
- Definition according to NIST SP 800-145 The NIST Definition of Cloud Computing http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf

  Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of ***five essential characteristics, three service models, and four deployment models***.

- Cloud computing is a fusion of the two basic technologies powered by ubiquitous Internet connectivity:
  - Utility Computing
  - Service Oriented Architecture (SOA)

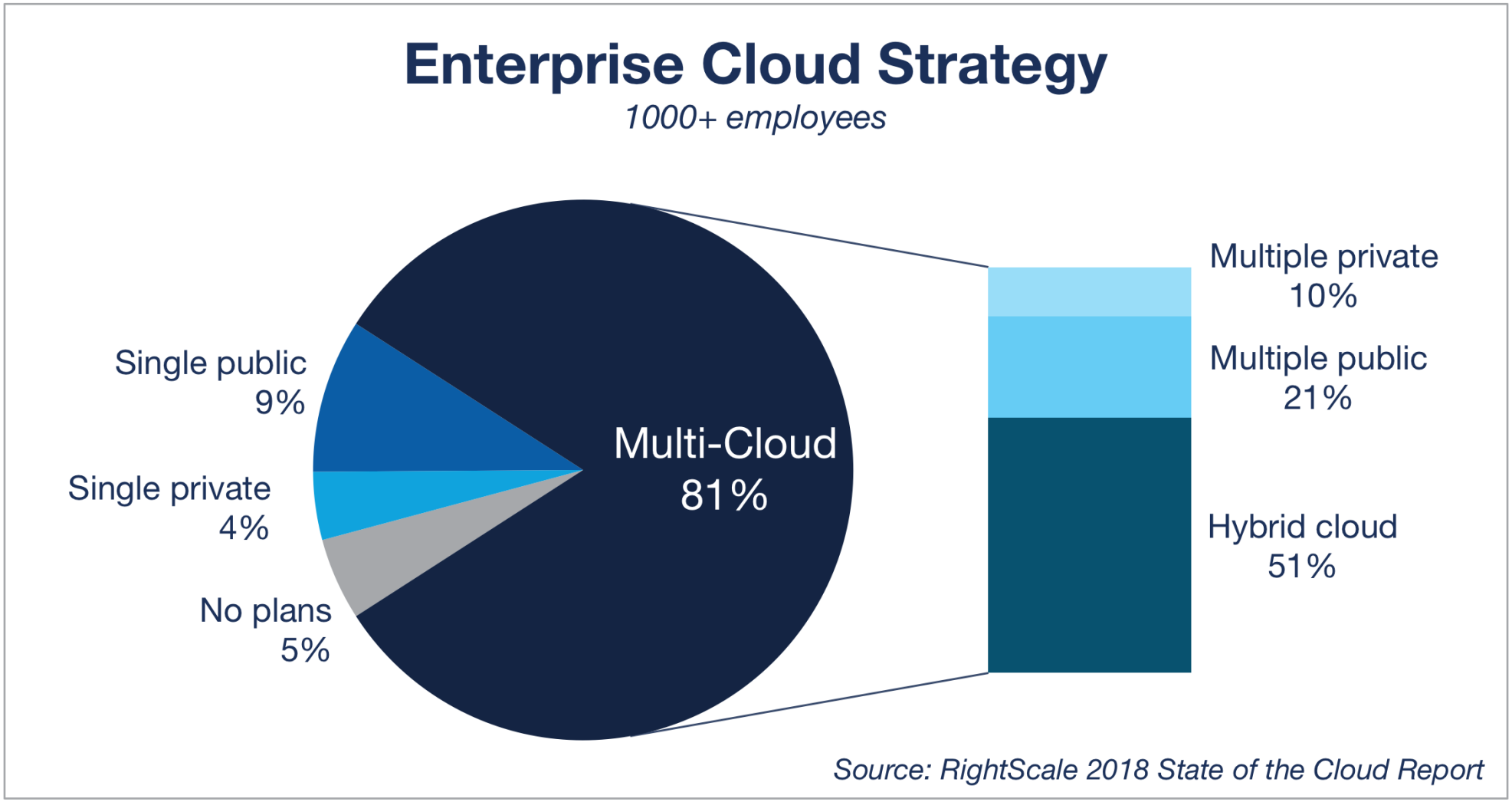
NIST Cloud Definition - Visualisation

Visual presentation of NIST Cloud Definition



- Five basic Cloud characteristics
  - On-demand self-service
  - Broad network access
  - Resource pooling
  - Rapid elasticity
  - Measured Service
- 3 basic service models
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
- Deployment models
  - Private clouds
  - Public clouds
  - Hybrid clouds
  - Community clouds

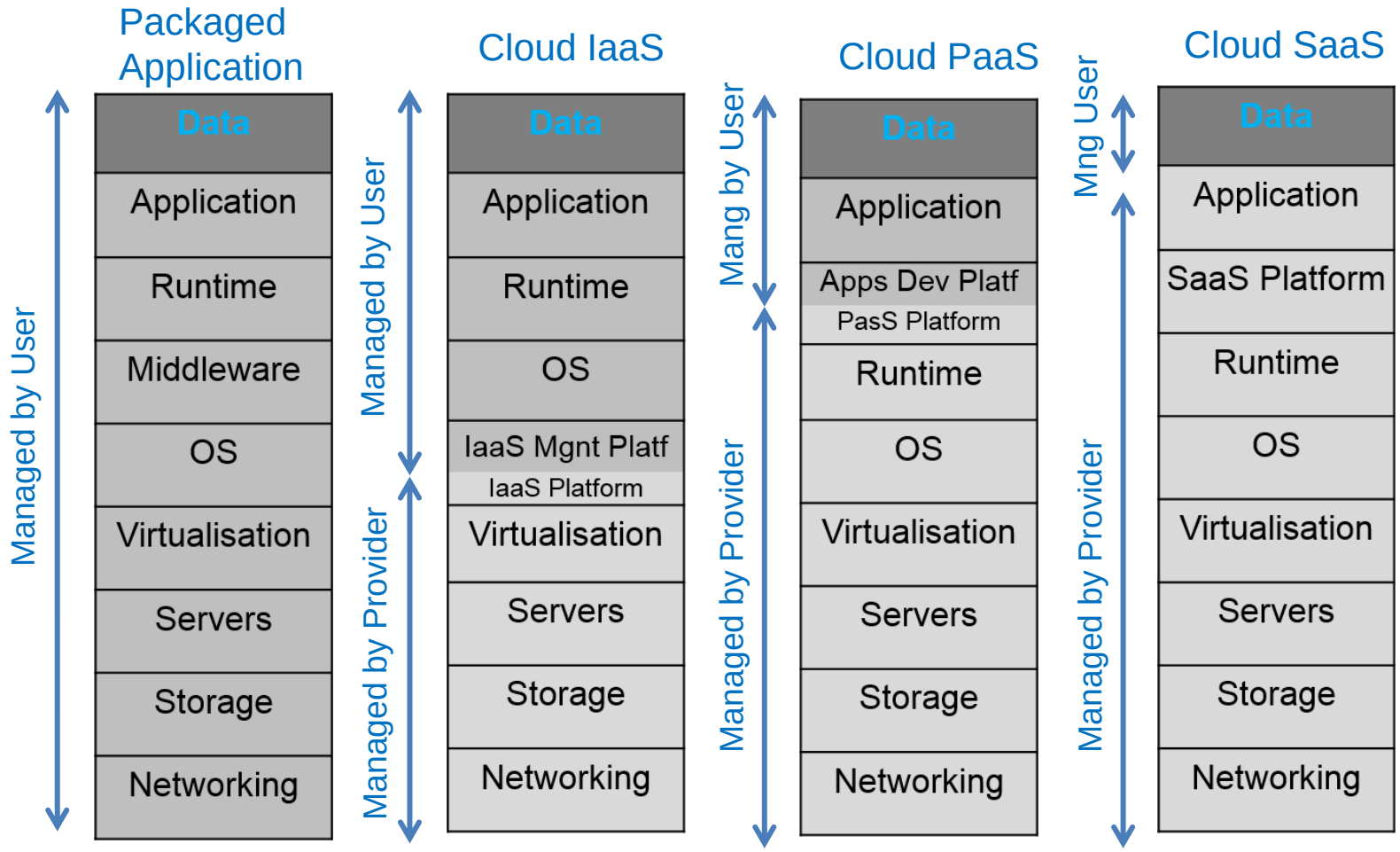
Enterprise Cloud Strategy



Technologies and Practices used for Multiclouds

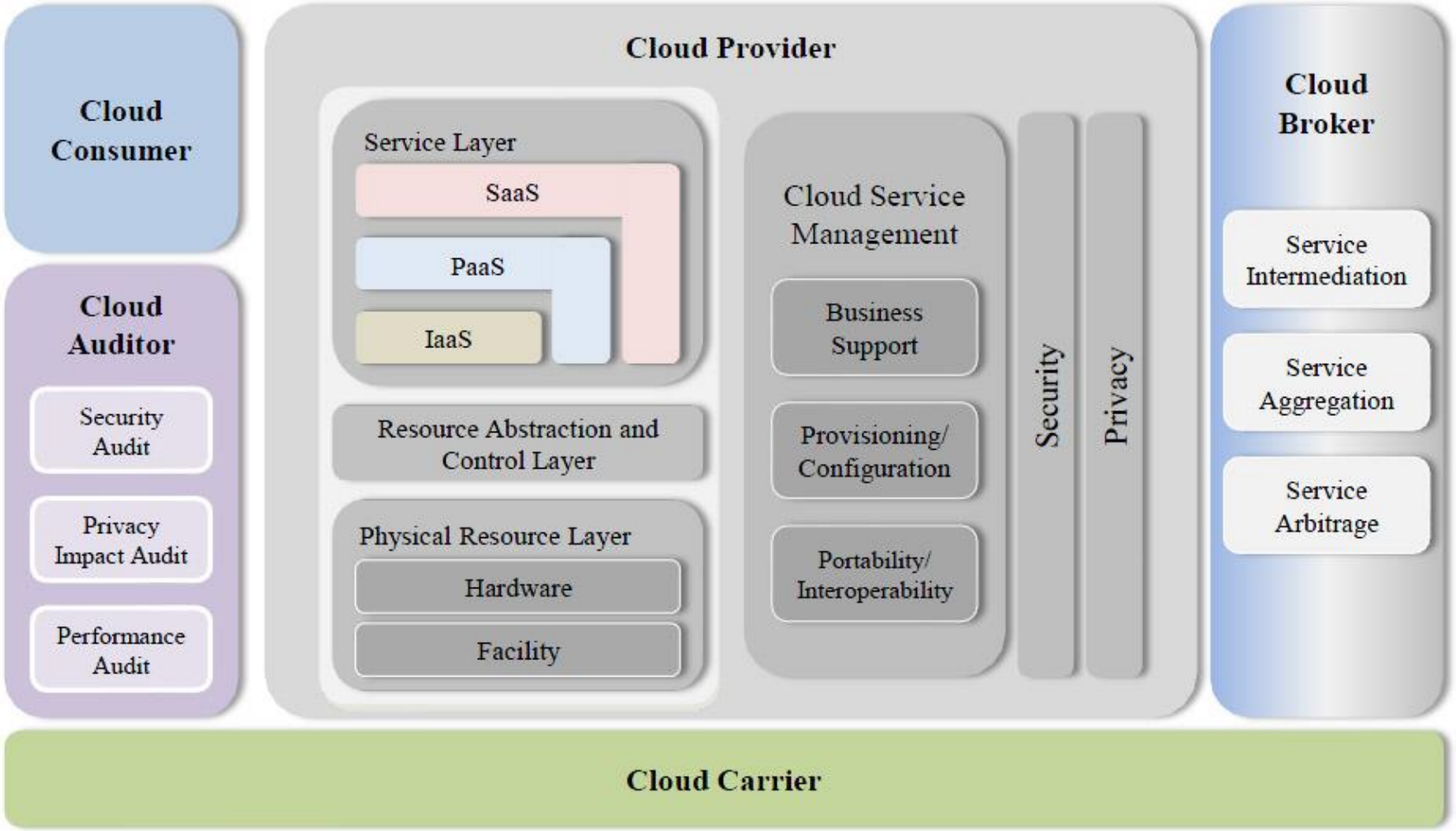
- Federate Access Control and Identity Management
  - Commonly used platform AAD – Azure Active Directory
  - Corresponding services by cloud providers
    - Amazon IAM (Identity and Access Management), Amazon Cognito
    - Cloud Identity and Access Management (Cloud IAM)
- Multi-cloud automation platforms (as part of DevOps CI/CD tools)
  - Ansible, Chef, Puppet, Terraform
- Software Defined Networks (SDN) for inter/multi-cloud network provisioning
  - Promising 5G cloud native network provisioning and automation
- Compliance tools and maturity assessment

Relation between IaaS, PaaS, SaaS



Note: Data always remain under user responsibility, however it may be processed on clouds. Physically, they may be processed at each of S/P/IaaS level

NIST Cloud Computing Reference Architecture (CCRA) 2.0 – Consolidated View (1)



Cloud Services Delivery Ecosystem: Actors and Roles

- Basic/Main actors – Define main business relation in cloud services delivery
  - Cloud Service Provider
  - Cloud Customer
  - Cloud User
  - Cloud Broker
- Other actors – Define other relations in cloud business
  - Cloud Carrier
  - Cloud Auditor
  - Cloud Developer, Cloud Integrator
  - Cloud/Intercloud Service Operator
  - Cloud Resource Provider
  - Physical Resource Provider
    - Can also be a “fixed” resources provider

General Cloud Use Cases (Classical, from 2010)

Use case 1:

- Moving part of workload to cloud in case of abrupt demand increase: cloudburst

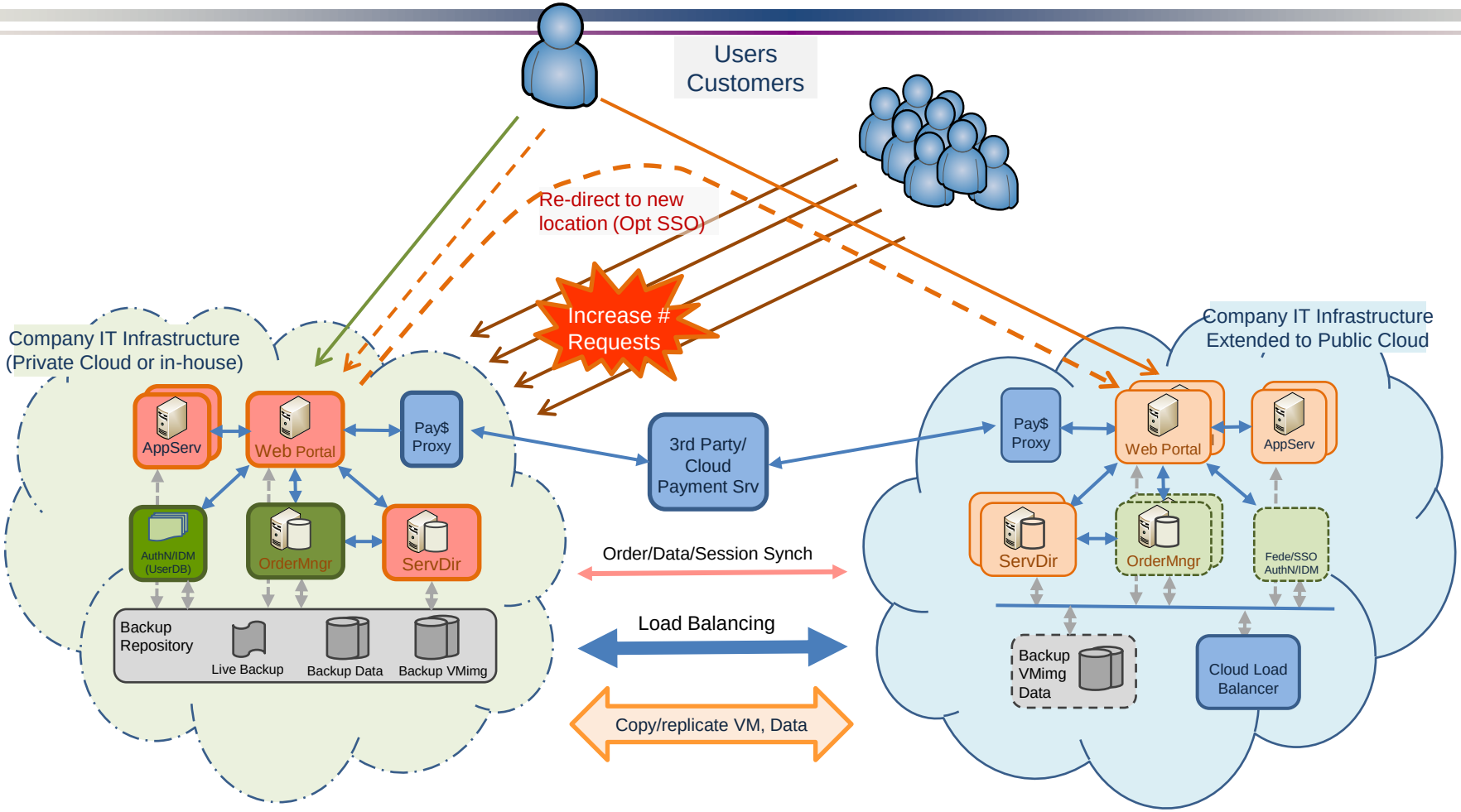
Use case 2:

- Disaster recovery
  - Moving/restoring emergency load in a partner cloud
  - Restoring own cloud based IT infrastructure

Use case 3:

- Service continuity when changing cloud provider

Cloudburst: Rapid load increase: seasonal, cyclical



Scenario: Abrupt demand increase for company's services: holiday shopping, seasonal, gaming, mobile Apps, ads campaign.

Cloudburst: Rapid load increase: seasonal, cyclical - Details

Scenario

- Webshops/eMarkets, entertainment sites have seasonal/holidays increase of load and users
- Surviving “disaster of success” when popular application or website attracts abrupt amount of users

Preconditions

- Company’s IT infrastructure is cloud based: private cloud or hosted on cloud
- Services and applications grouped to simplify services extension to cloud
  - Some 3rd party services (like payment systems) are already hosted on cloud
- The whole or part of IT infrastructure is backed up, including VM, Data, UserDB, topology, state/session

Sequence:

- Cloudburst scenario is triggered when increased number of requests causes services delay or interruption
- VM images and up-to-date order data (optionally UserDB) are backed up/replicated and transferred to suitable cloud provider (location, compatibility, cost)
- VMs and all necessary components are deployed in new cloud/location, data and states are synchronized
- Requests (all or part) are started to be re-directed to new location benefiting from elasticity of cloud resources
  - Additional capacity are automatically added to keep required Quality of Service (QoS), e.g. request processing time, downlowd sped, streaming quality
- Some services are typically not replicated to burst cloud, e.g. UserDB and order or payment processing
  - Initial client authentication can be done at the main site/portal and redirected using Singe Sign On (SSO) to new/cloud location
  - Data and processes synchronization must be in action
- External cloud resources and infrastructure stopped and de-commissioned, VM destroyed, after demand decrease (scale-down), all business related data are transferred back to company

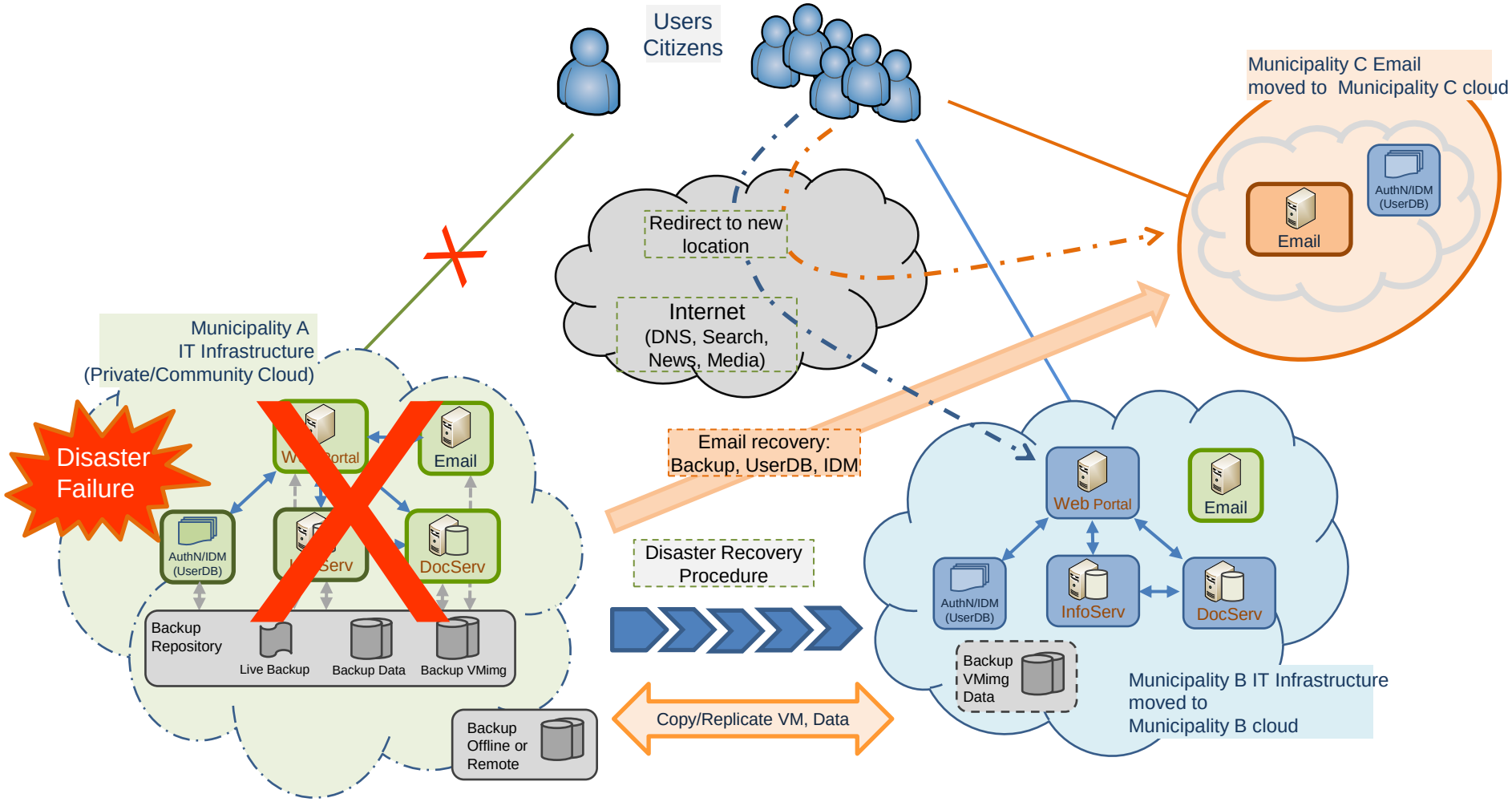
Challenges: Real time migration

Cloudburst: Design suggestions

- How will you launch your workloads? From a template, from a clone or from a dormant VM/instance?
- How will you connect, and how much data do you intend to push over this network connection?
  - Is it a point-to-point network, MPLS, EVPL or VPN, and is it production data, metadata, sensitive data or management traffic?
- How automated should this solution be? Is the cloud portal you have chosen easy enough to operate to take advantage of cloud bursting?
  - An API can provide full automation, but will require coding and additional business logic in your applications.
- How will the cloud handle your security policies? Does the cloud you have chosen have the governance and maturity you would expect for you data?
  - Can you even bring your own policies into the cloud? After all, the cloud holds your data, shouldn't it be able to support your existing IT policies?
- How will you handle load balancing? Will you need local and possibly global load balancing that can be dynamically updated to include the new workloads you have bursted into the cloud?
- How will you charge back in case of services outsourcing? Does your cloud bursting solution make it easy to charge back internal and external customers and set spend limits, controlling cloud sprawl and avoiding the auto-ballooning of cloud costs?

[ref] Cloudburst: Three steps for enterprise scaling into the cloud
http://blog.savvis.com/2012/04/cloud-burst-three-steps-for-enterprise-scaling.html /

Disaster Recovery (massive infrastructure failure)



Scenario: Due to natural disaster IT infrastructure of Municipality A destroyed; offline backup stored remotely is available; information service restored in the municipality B, email – in municipality C

Disaster Recovery: Services restored in a new location

Scenario

- Due to natural disaster IT infrastructure of Municipality A destroyed
- Offline backup stored remotely is available but cannot be used from Municipality A
- There is vital need for information both for citizens and for rescue team
- amount of users

Preconditions

- Municipalities' IT infrastructures are cloud based: using community cloud deployment model
- The whole IT infrastructure is backed up regularly, including VMs of all applications and services, Data, UserDB, topology
- Data and backups are replicated to/stored remotely

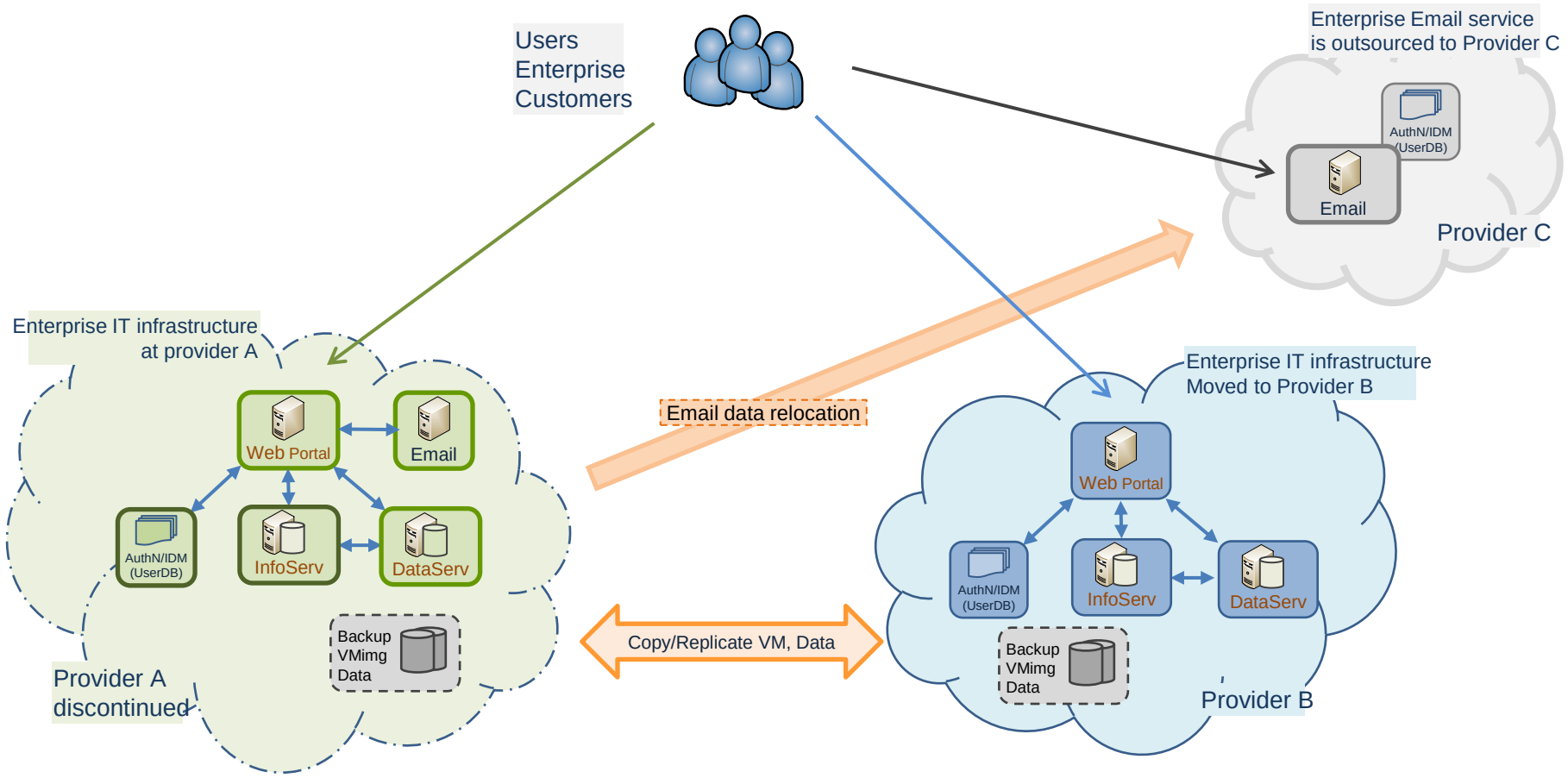
Sequence:

- Emergency Team (ET) starts working and following emergency response procedure
- ET accesses backup and transfers all files and images to previously defined location(s):
  - Information service is restored in the municipality B
  - Email service is restored in municipality C
  - Some services are provided by other municipalities
- New services location is registered in DNS and information is populated on Internet and on the web, by phone, newspapers
- Municipality A information services and email starting working on emergency mode; when original facility and datacenter are restored, services will be migrated to original location

Challenges:

- Full services backup and restoration must also include infrastructure and services topology
- Compatibility and standards for VM images, Data, service description and topology
- Compatible cloud platforms in Municipality A, B, C

Service continuity when changing Cloud provider



Scenario: Provider A discontinues its service; there is transition period; enterprise moves services to a new provider, assures services continuity, makes services optimisation

Service continuity when changing Cloud provider

Scenario

- Provider A discontinues its service; there is transition period; enterprise moves services to a new provider, assures services continuity

Preconditions

- Enterprise IT infrastructure is cloud based: private cloud or hosted on cloud
- The whole IT infrastructure is backed up, including VM, Data, UserDB, topology
- There is a transition period and a transition plan that also includes service/infrastructure optimization, some applications re-design

Sequence:

- Enterprise transfers/replicates either individual VM images or the whole infrastructure to new provider(s)
  - Main IT infrastructure is moved to provider B
  - Email service is moved to provider C
- Data are replicated to new location(s) and synchronised
- New services location is registered in DNS for correct Internet traffic forwarding; no other changes required
- Enterprise starts operating from a new location a new cloud provider as usual

Challenges:

- Full services backup and migration must also include infrastructure and services topology
- Compatibility and standards for VM images, Data, service description and topology
- Compatibility of cloud platforms at providers A, B, C

Self-check Questions

- Dropbox: What type of cloud service or application?
- You run/setup accountant company. What type of cloud service you would use?
- You are a researcher using unique (old) software for your data processing.
  - What type of cloud service you will use?
- You need to setup a summer school on (agile) programming techniques.
  - Will you use cloud? What type?
- You need to support a project group that besides writing reports needs also regular project calls (voice, video)

Cloud enabling technologies

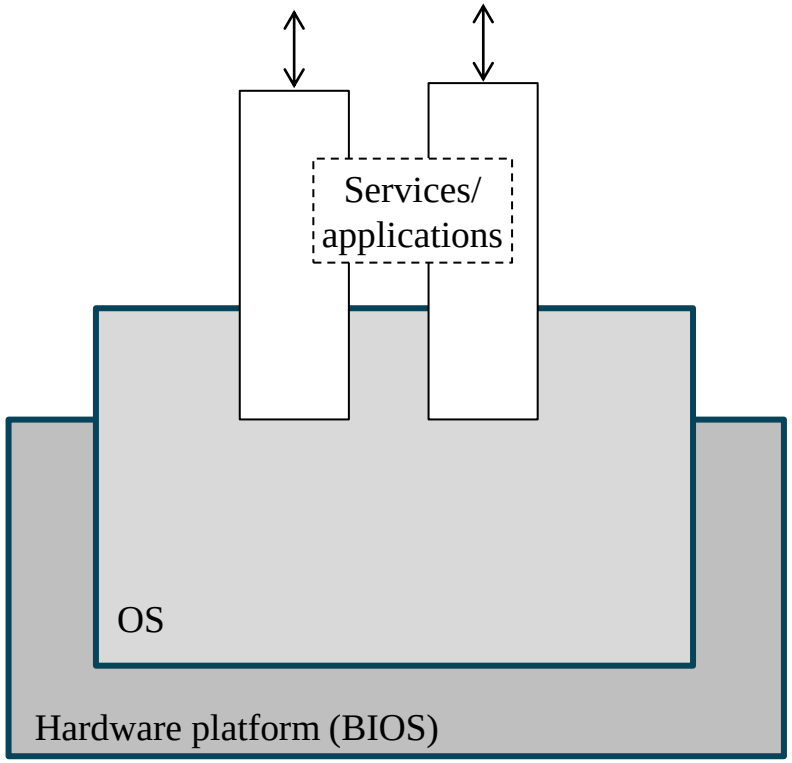
- Virtualisation
- Containers and Kubernetes

Virtualisation environment components (1)

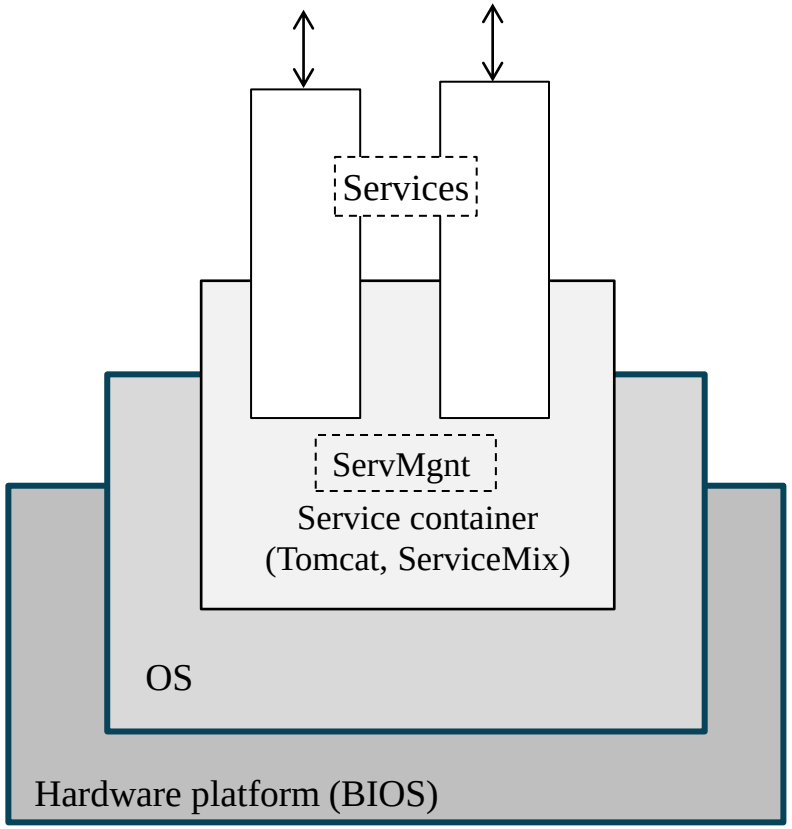
Service container allows for running multiple services on one computer/OS

- Web applications
- Services isolation

Simple Application/Services model



Web based Application and Web Services

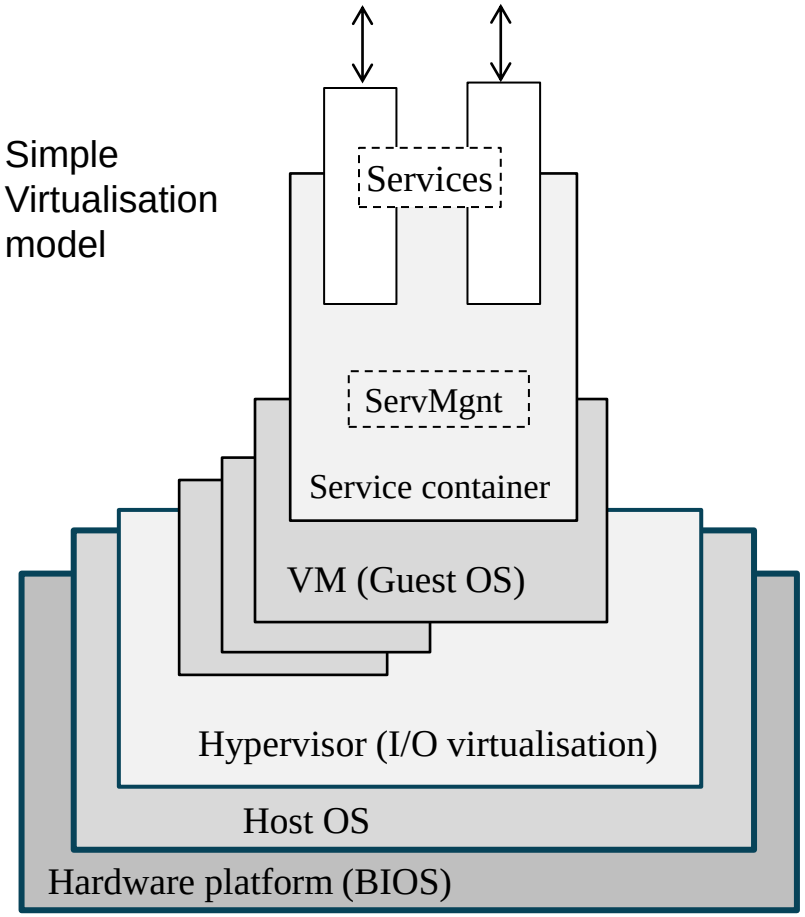


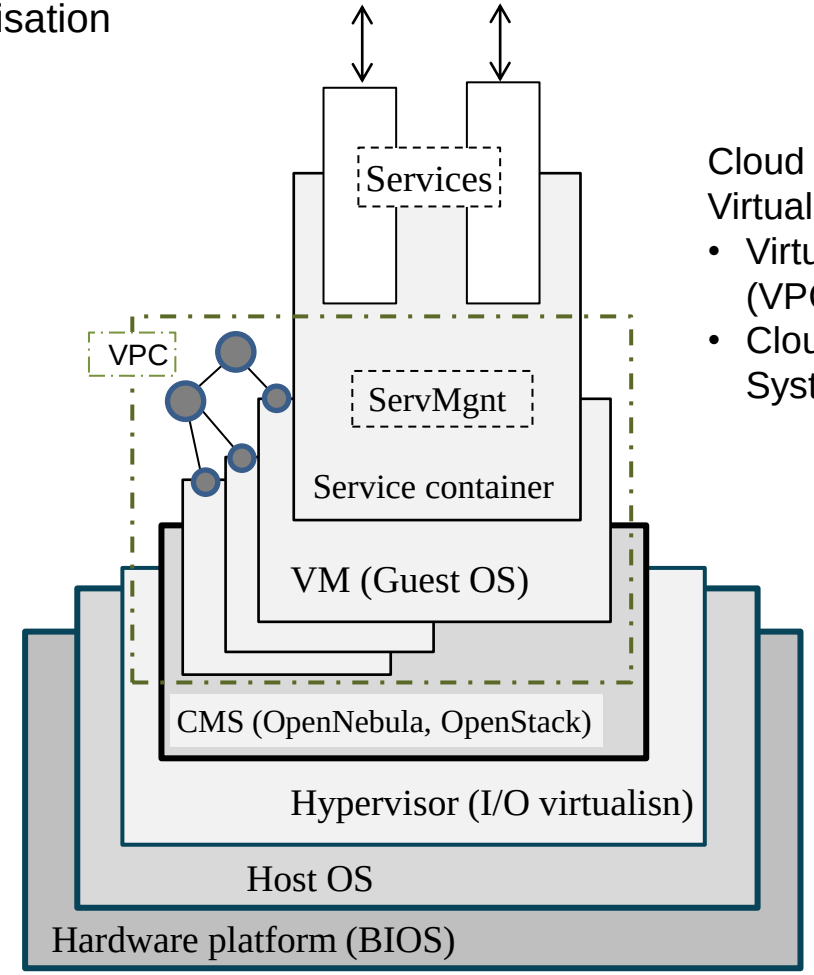
Virtualisation environment components (2)

Virtualisation and hypervisor allows for running multiple OS on one computer/OS

- Cloud Management Software provides flexible VM management
- Hypervisor provides VM isolation and CPU, Memory, I/O virtualisation

Simple Virtualisation model



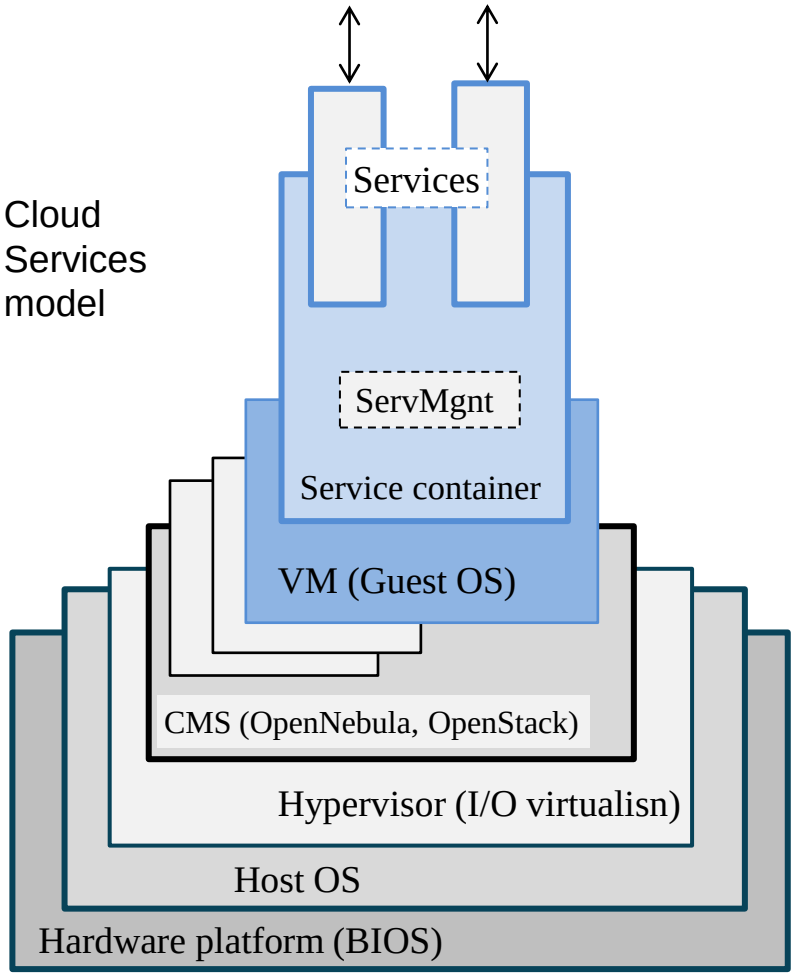


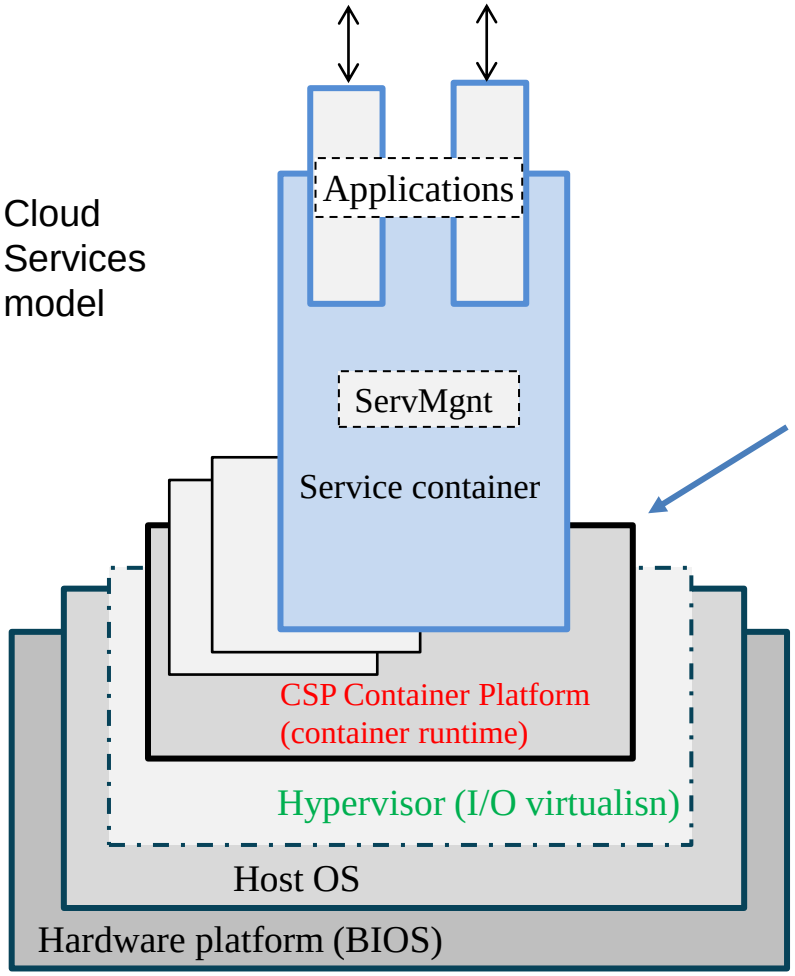
Cloud based Services Virtualisation

- Virtual Private Cloud (VPC)
- Cloud Management System (CMS)

Cloud: VM vs Container virtualisation

- Containers provide less overhead on for CMS and User applications
- But have a noisy neighbor effect





E.g. Docker Engine

Cloud Native Computing Foundation (CNCF)
https://www.cncf.io/

- CNCF is the open source, vendor-neutral hub of **cloud native computing**, hosting projects like Kubernetes and Prometheus to make cloud native universal and sustainable.
-

Docker and Kubernetes

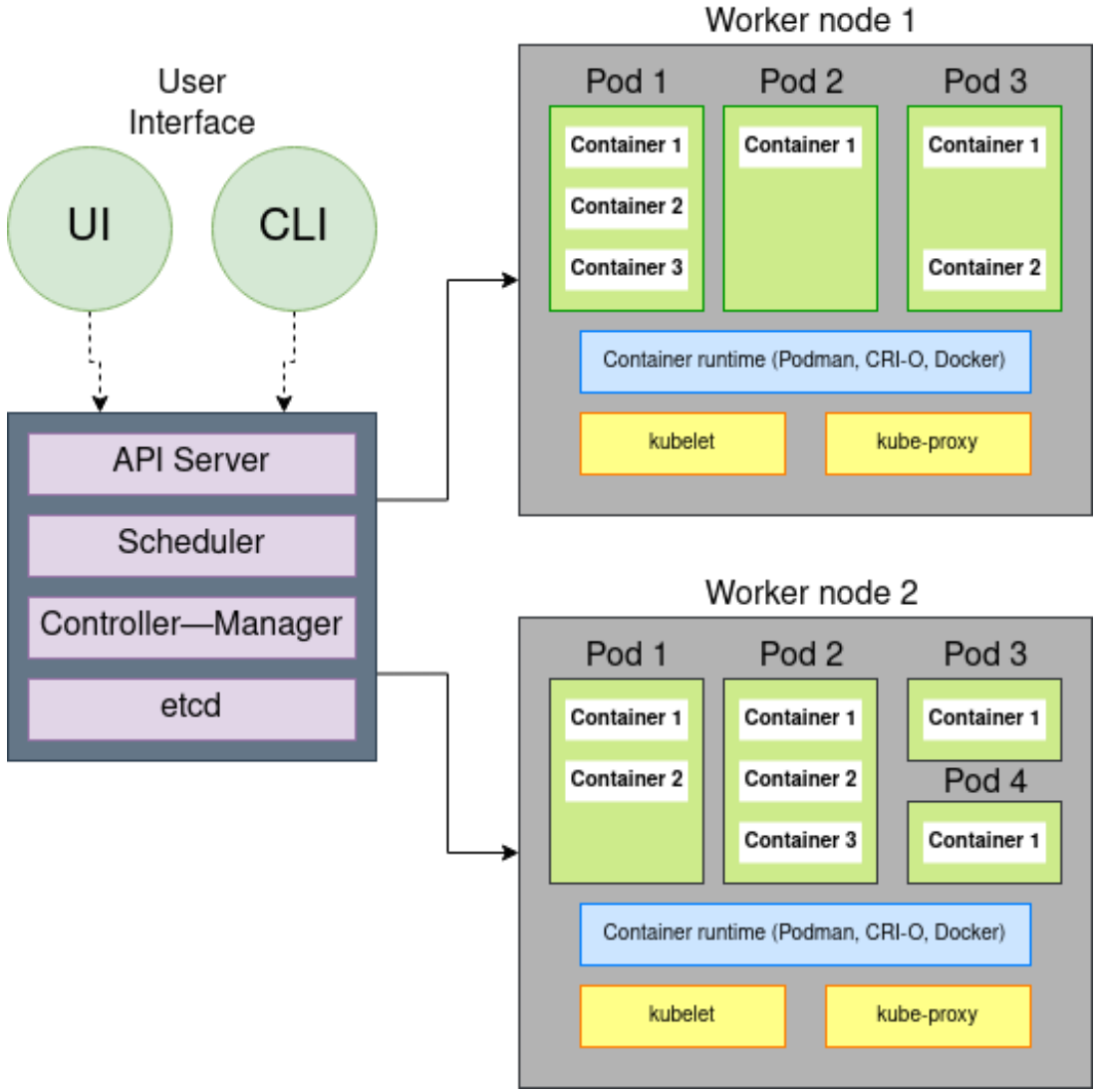
- Kubernetes (https://kubernetes.io/) is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.
  - The name Kubernetes originates from Greek, meaning helmsman or pilot.

Kubernetes provides benefits:

- **Service discovery and load balancing** Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
- **Storage orchestration** Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.
- **Automated rollouts and rollbacks** You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.
- **Automatic bin packing** You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.
- **Self-healing** Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
- **Secret and configuration management** Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

Kubernetes Architecture

https://www.cncf.io/blog/2019/08/19/how-kubernetes-works/



- A Kubernetes pod is a group of containers, and is the smallest unit that Kubernetes administers.
  - Pods have a single IP address that is applied to every container within the pod.
  - Containers in a pod share the same resources such as memory and storage. This allows the individual Linux containers inside a pod to be treated collectively as a single application, as if all the containerized processes were running together on the same host in more traditional workloads.
- A Kubernetes node manages and runs pods; it's the machine (whether virtualized or physical) that performs the given work.
  - Just as pods collect individual containers that operate together, a node collects entire pods that function together. When you're operating at scale, you want to be able to hand work over to a node whose pods are free to take it.
- The Kubernetes control plane is the main entry point for administrators and users to manage the various nodes.
  - Operations are issued to it either through HTTP calls or connecting to the machine and running command-line scripts. As the name implies, it controls how Kubernetes interacts with your applications.
- A cluster is all of the above components put together as a single unit.

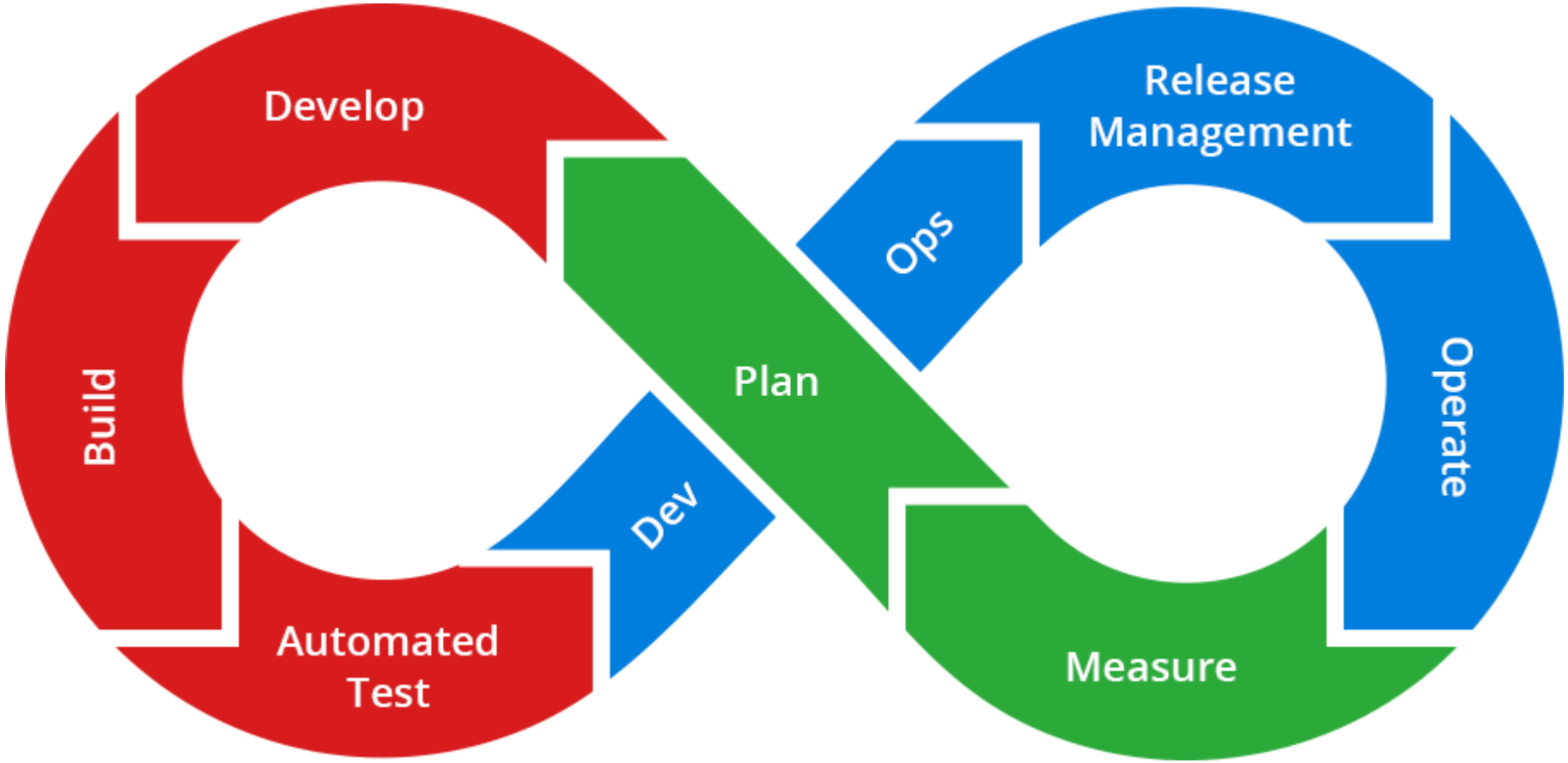
DevOps and Cloud based Software Development

- DevOps: main concepts and practices
- Example: AWS Tools for DevOps, Metrics and Logging

DevOps and CI/CD Pipeline

- DevOps definition and DevOps principles
- From DevOps to Site Reliability Engineering: Operationalising large sites and projects
  - Moving right to provide feedback left

Continuous Evolution in Product Development



DevOps – Formal Definition

DevOps - is the practice of operations and development engineers participating together through the entire service lifecycle; from the design and development process all the way to production support. DevOps is also characterized by operations staff making use of many of the same techniques as developers for their systems work.

- Extended “live” definition https://theagileadmin.com/what-is-devops/
- **DevOps has strong affinities with Agile and Lean approaches**.
- DevOps can be interpreted as an outgrowth of Agile – agile software development
  - prescribes close collaboration of customers, product management, developers, and (sometimes) QA to fill in the gaps and rapidly iterate towards a better product
- ***DevOps is simply extending Agile principles beyond the boundaries of “the code” to the entire delivered service***

DevOps Essentials

- Better Software, Faster time to market
- Infrastructure as Code and cloud based virtualisation
- Synergy of Development and Operations
- Covers the *entire* Application Life Cycle

5 Key DevOps Methodologies

- People over Process over Tools
- Continuous Delivery
- Lean Management
- Visible Ops style Change Control
- Infrastructure as Code

Lean thinking

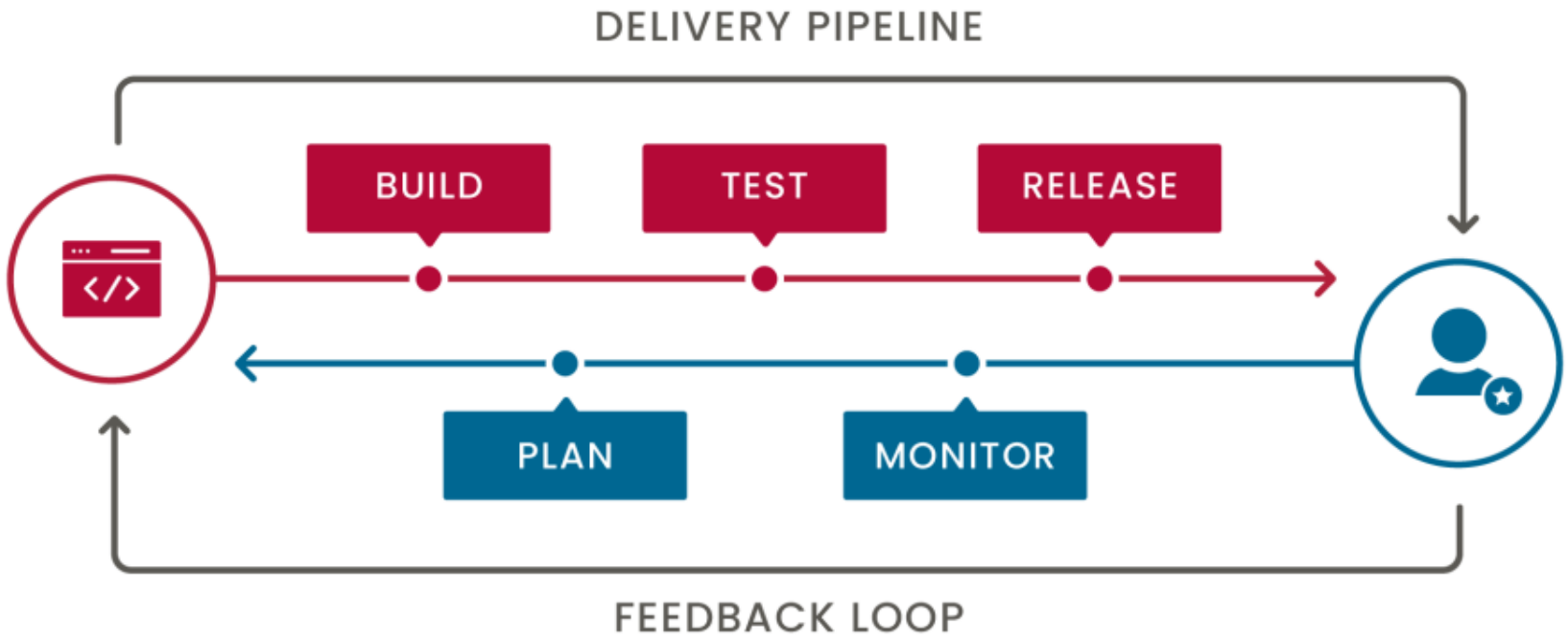
The Optimist

The Pessimist

The Lean Thinker

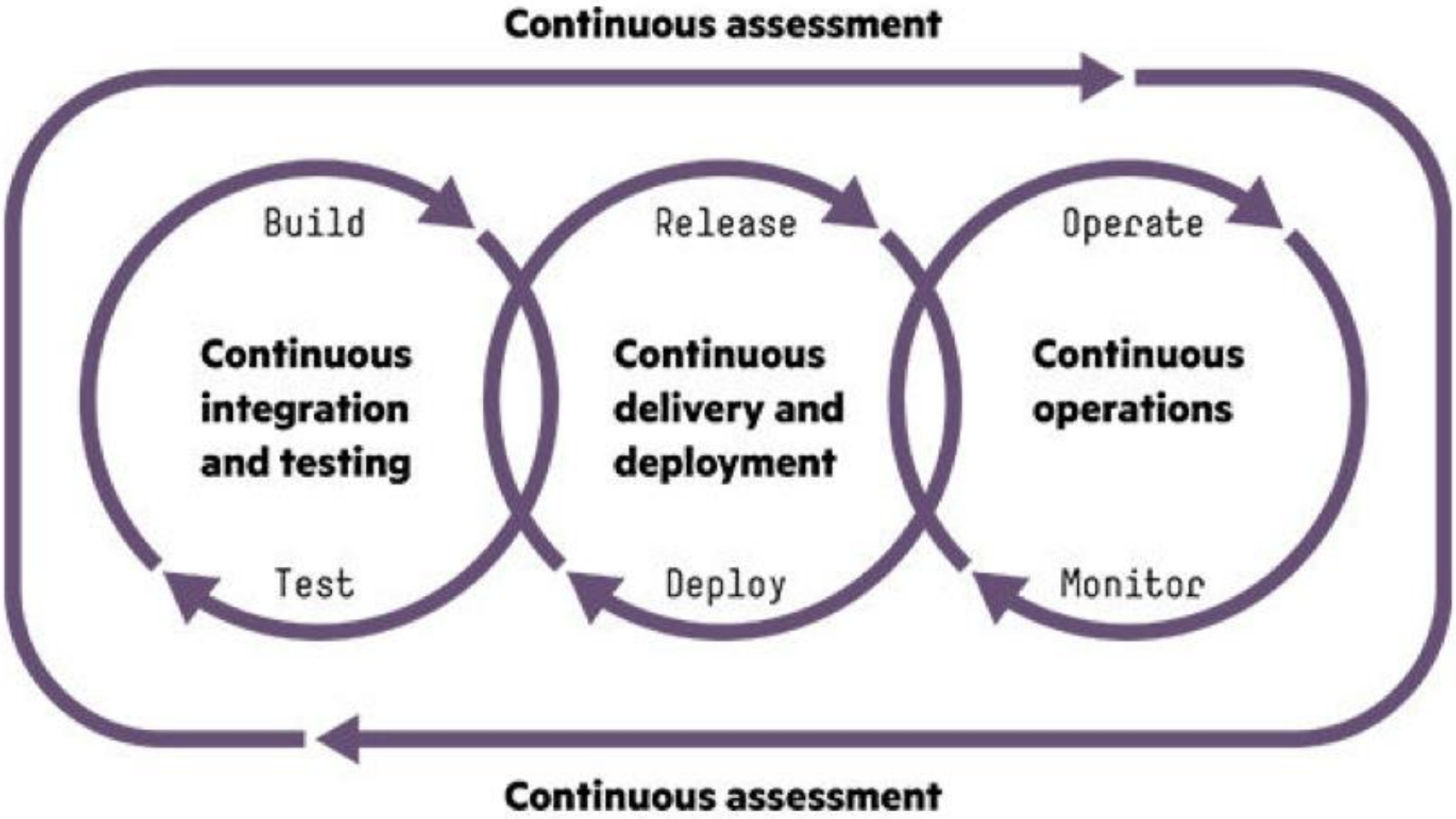
[ref] DataOps: 9 steps to transform your data science impact, Strata Conference 2018

Delivery Pipeline – Feedback Loop



- Feedback is essential at each step of the process

3 Cycles in the DevOps process



From DevOps to SRE: Operationalising SE/Apps, Data Analytics/DataOps and MLOps

- Benefit from current trends to operationalize engineering/development process
- SRE (Site Reliability Engineering) introduces practice of monitoring and fulfilling Service Level Objectives (SLO, subset of SLA) by defining Service Level Indicators (SLI)
  - To be leveraged to Data Analytics model monitoring and improvement
- SRE may provide an approach to facilitate early DA model deployment and testing with real data

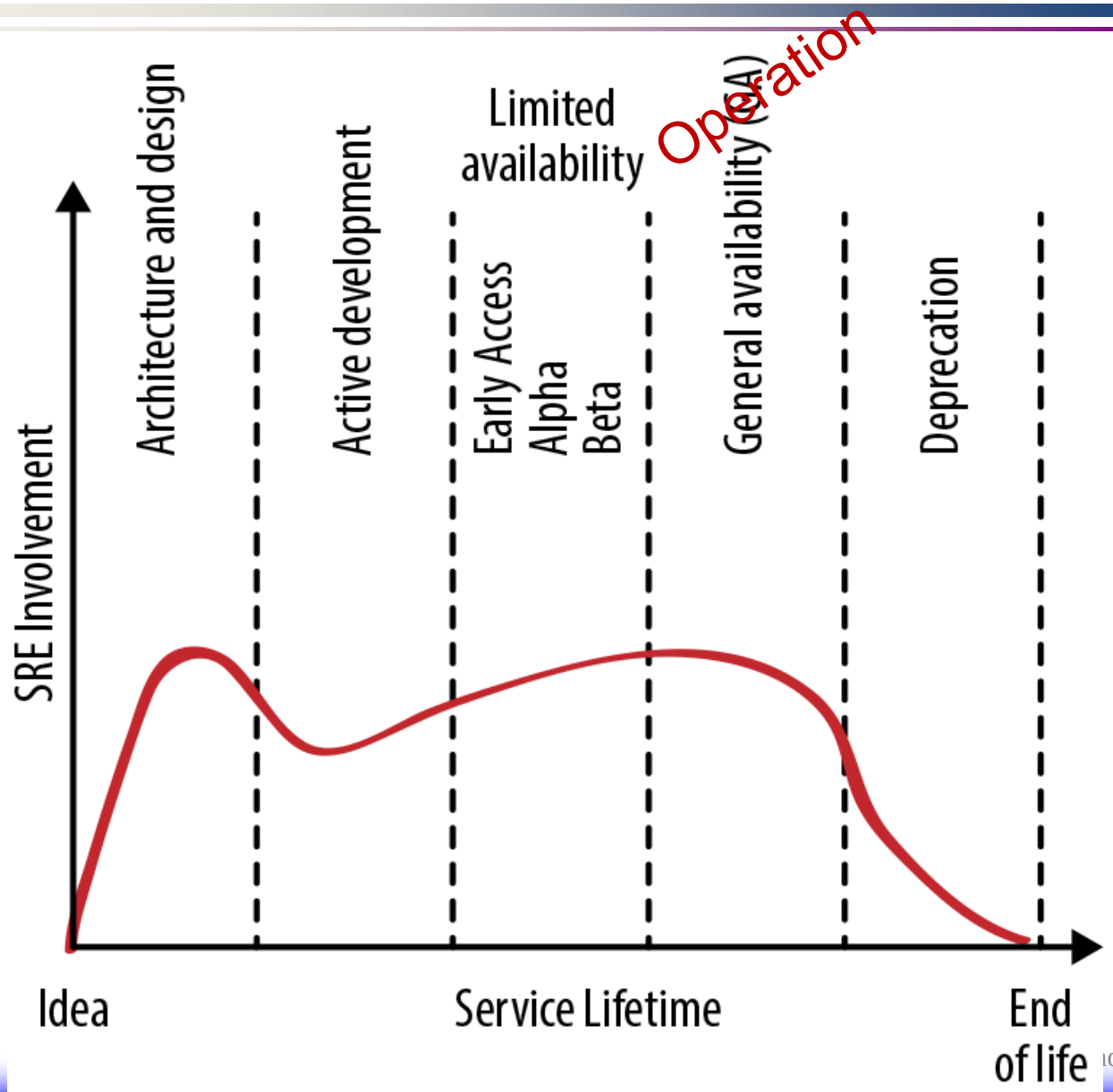
SRE and DevOps
https://www.ibm.com/cloud/learn/site-reliability-engineering

- **DevOps principles**: Reduce organizational silos, leverage tooling and automation
- **SRE practice**: Use the same tooling to automate and improve operations as developers use to develop and improve software

- **DevOps principles**: Accept failure as normal, implement gradual changes
- **SRE practice**: Use error budgets to continually deploy new features and functionality within acceptable levels of availability

- **DevOps principle**: Measure everything
- **SRE practice**: Base decisions to release new software on SLA metrics

SRE and Data Analytics Engagement



What are benefits of such fusion/integration

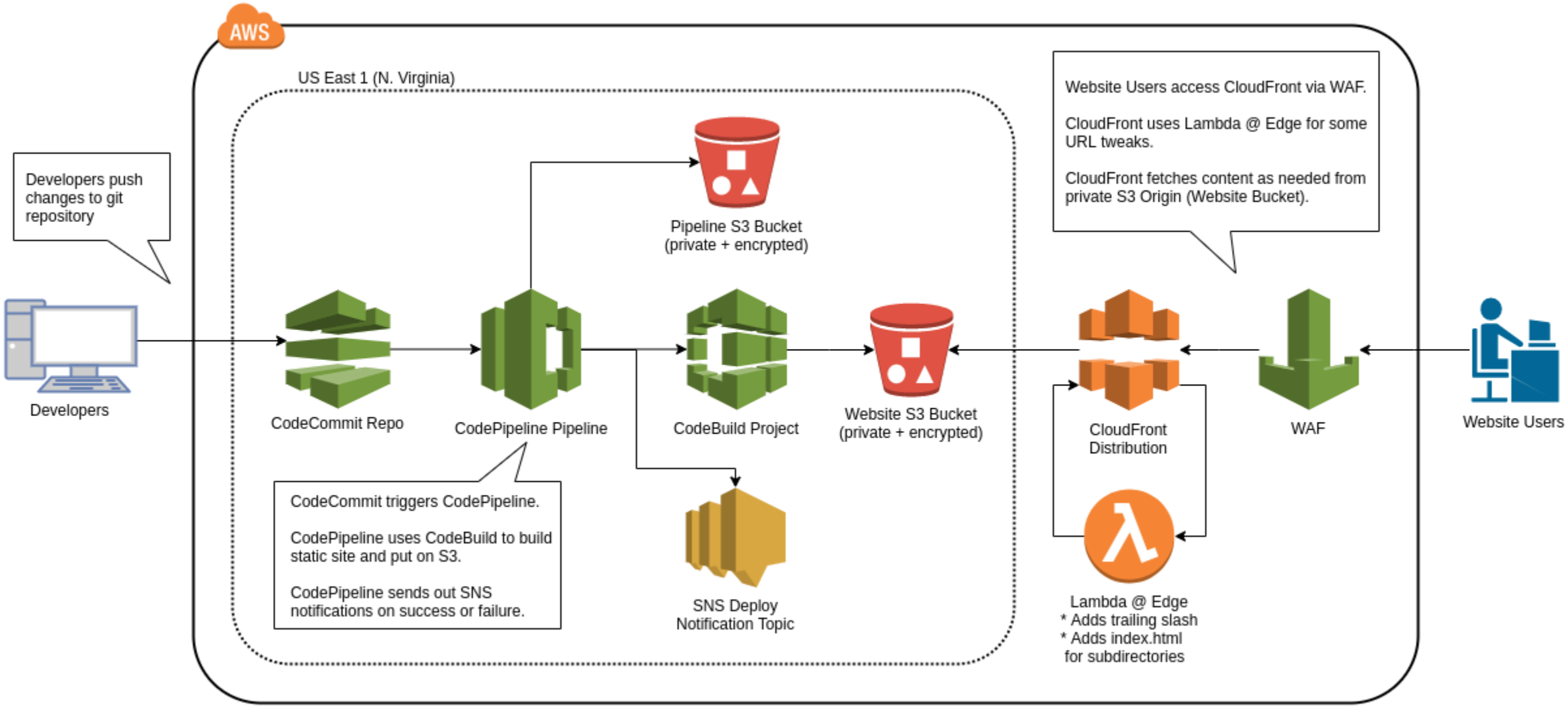
- SRE tends to create and maintain the whole operational model to enable continuous improvement
- Continuous model improvement
  - Essential for operational AI and Digital Twins
- Monitoring and optimization on both measurable and assessable/ranked KPI

[ref] https://sre.google/workbook/engagement-model/

AWS cloud based tools and services for DevOps

- AWS: CodeCommit, CodePipeLine, CodeBuild
- Monitoring and metrics
- Code inspection

Cloud based services development and CI/CD pipeline

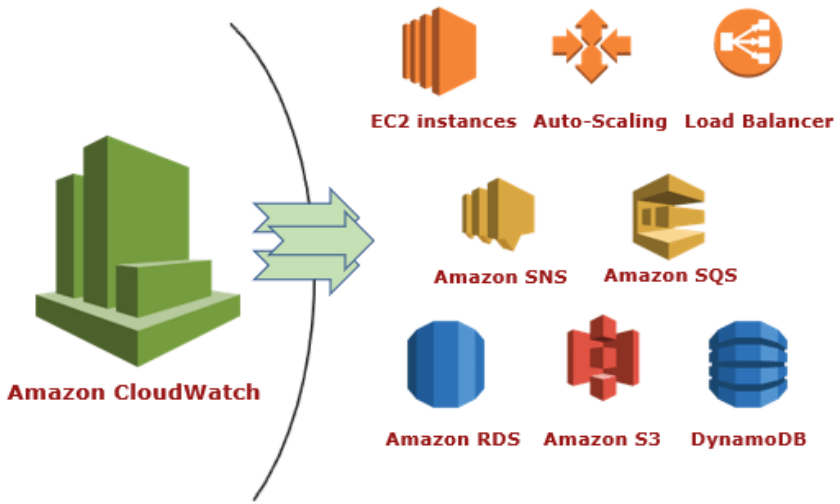


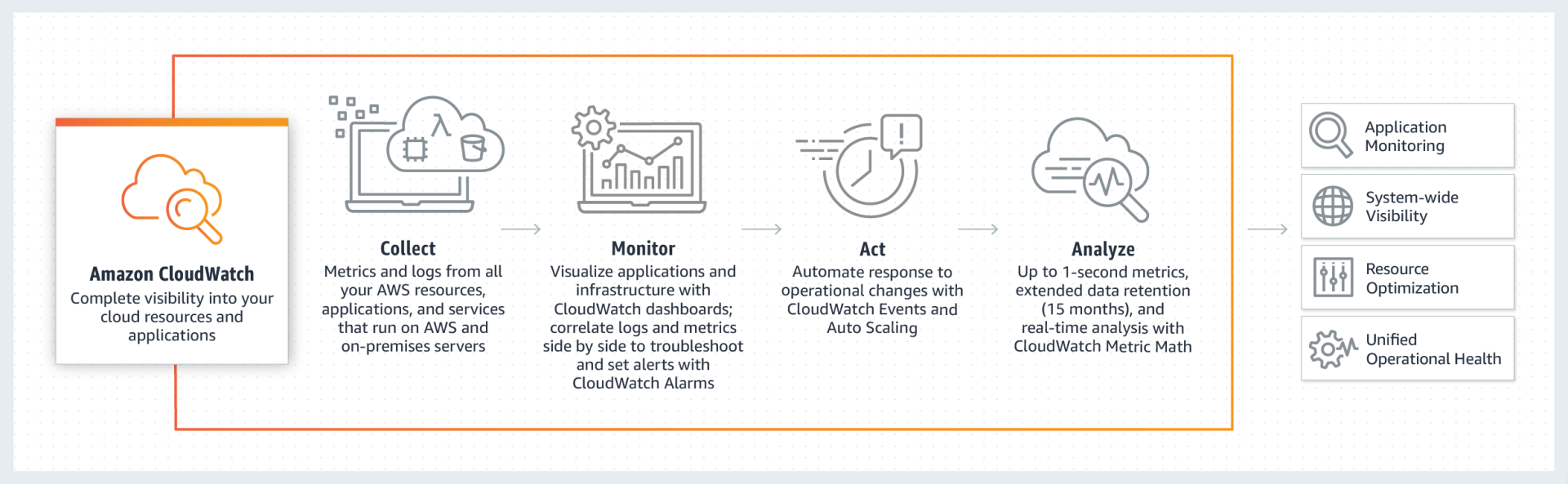
AWS Console Tools for Metrics and Logging

- CloudWatch
  - Metrics from service logs
  - Basic metrics included
  - Advanced metrics optional
- Other console tools
  - CloudFormation – Deployment template catch all monitoring setup
  - Trusted Advisor – AWS review of your services
  - Service Catalog – provides permission deployment templates and Amazon Machine Images (AMIs)
  - CloudTrail – Allows logging/monitoring of specific services API calls
  - Config – records/monitors the live environment for configuration
- 3rd party tools
  - OpsWorks – configuration management service, monitors the state of live environment as part of DevOps, working with Chef
  - Inspector (Security area) – produce list of potential security issues
    - Install agent on instances and use configurable logging
    - Templates for best security practices
  - X-Rays (Development area) –
    - Debug – Tracing – Service mapping

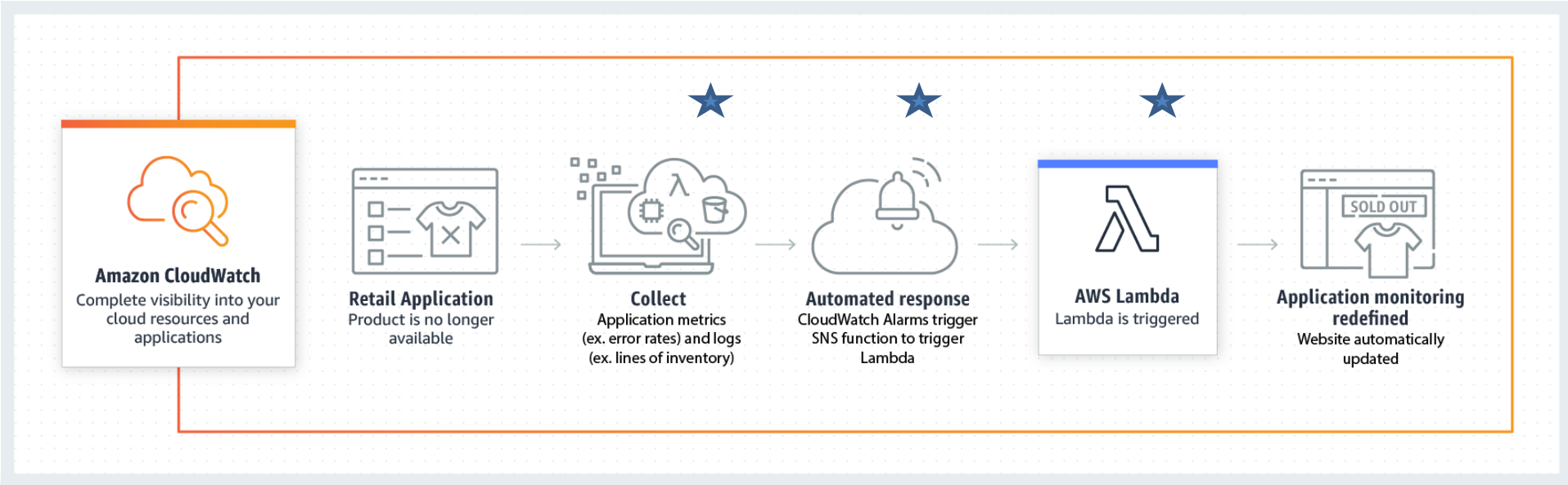
CloudWatch
https://aws.amazon.com/cloudwatch/

- Benefits
- Limitations
  - Limited Data Retention
  - The first limitation of CloudWatch is the restriction to **two weeks** of metrics data.
  - Limited Metrics at free tier: cost rise quickly with added metrics
  - Limited Dashboards



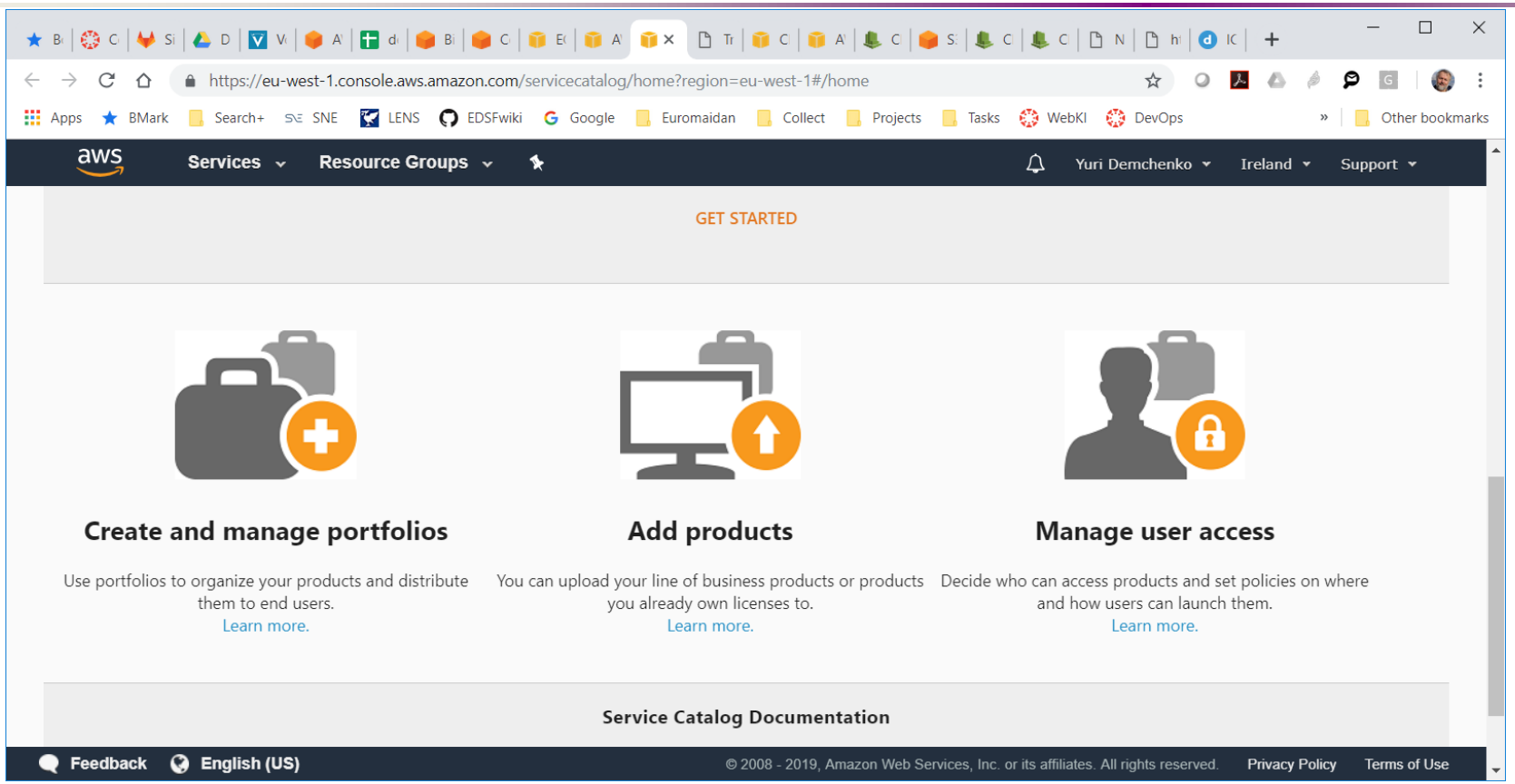


Infrastructure monitoring and troubleshooting (a) and Applications monitoring (b)



- Infra monitoring and troubleshooting
  - Collect, Alert, Unified view, Identify cause
- Applications monitoring
  - Collect, Analyse, Response

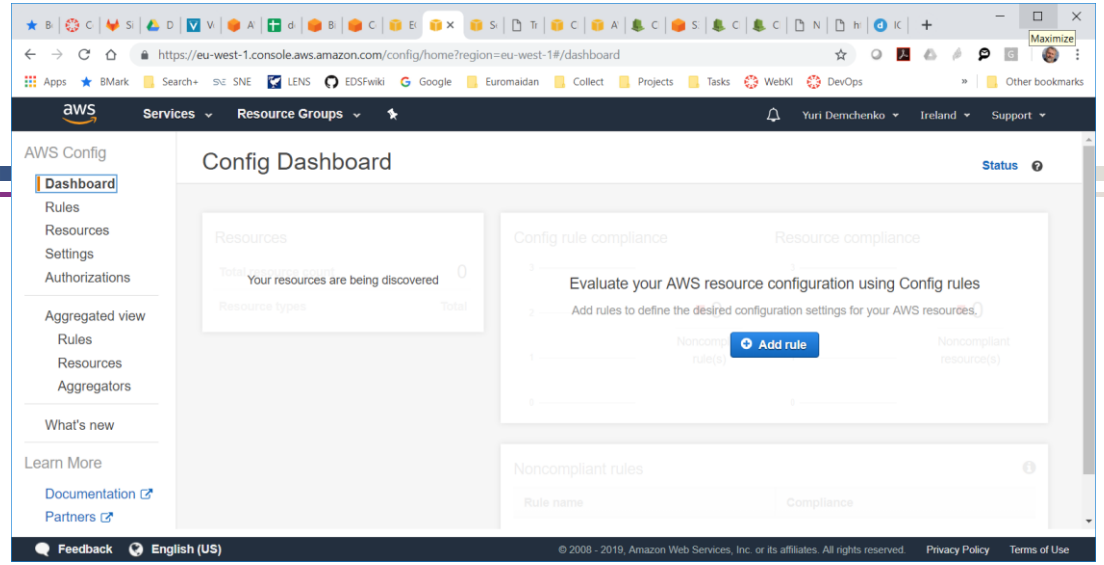
Service Catalog

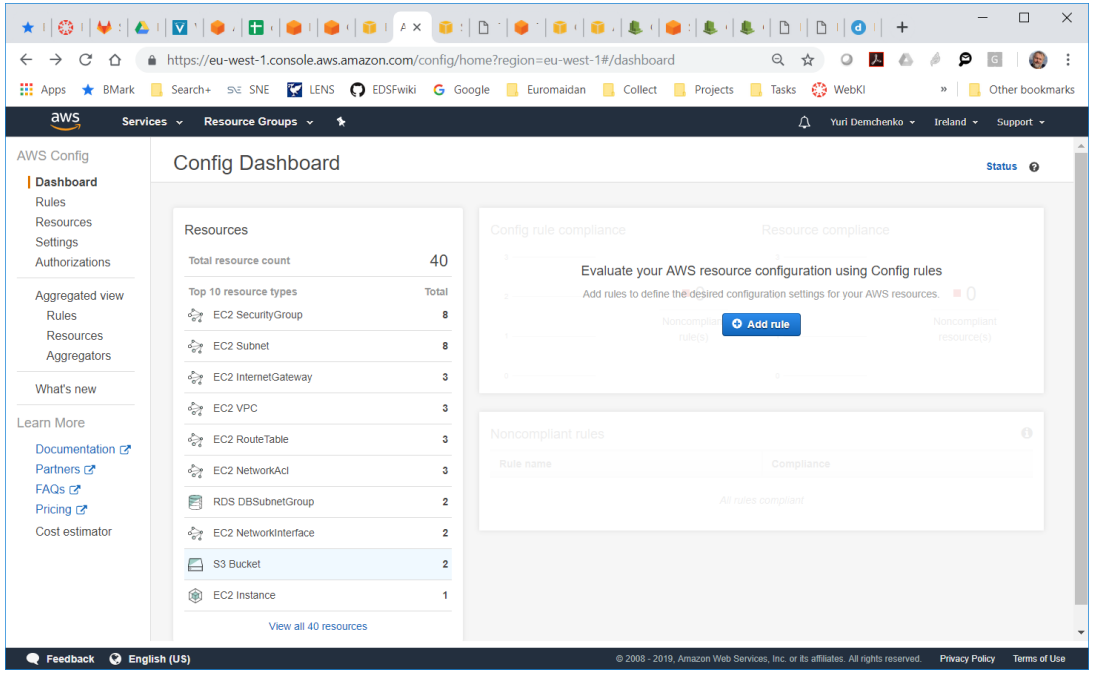


- Create catalog of enterprise services portfolio
- Add products: CF templates or Marketplace
- Set permissions
  - Launch control, e.g. version control

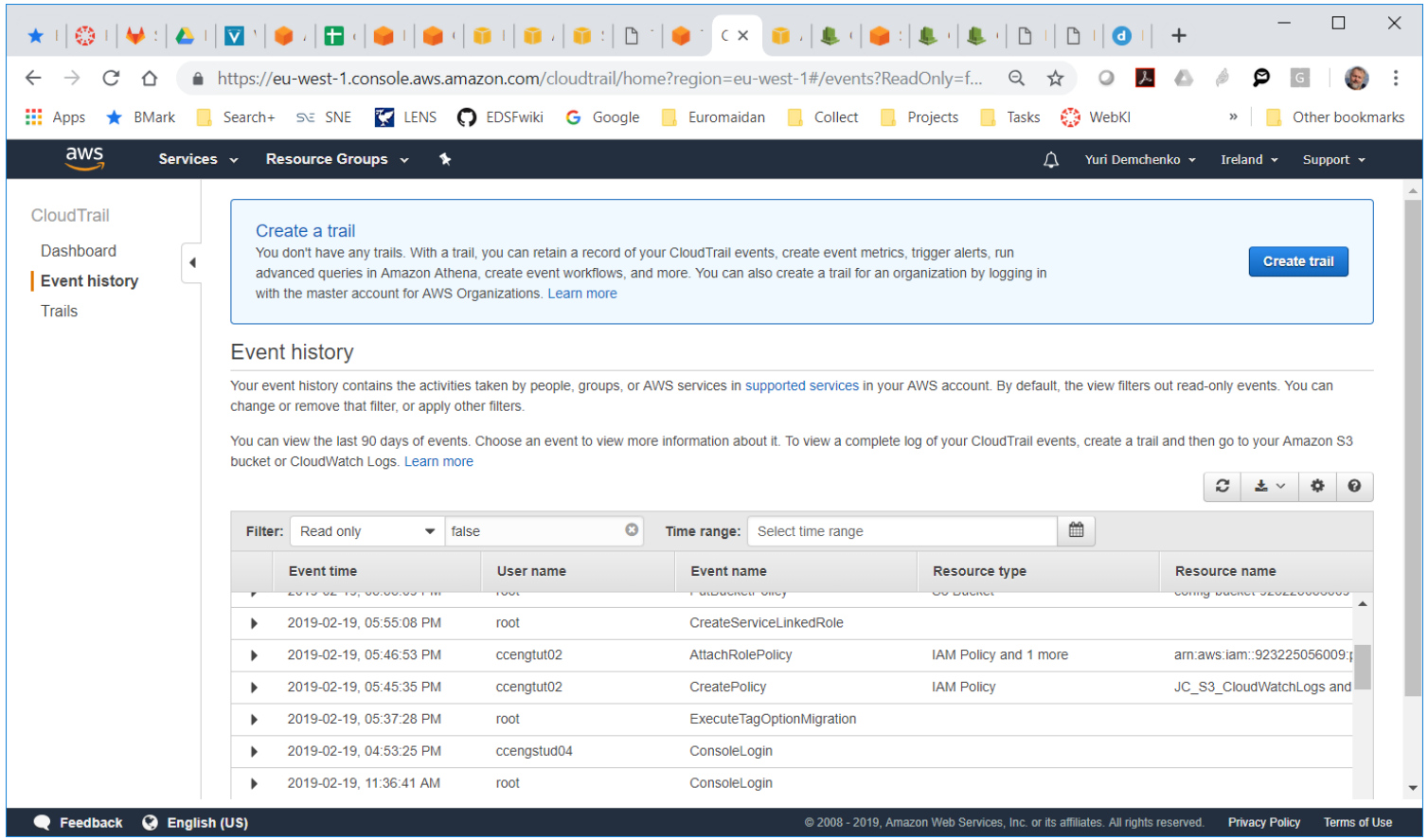
AWS Config

- Setup configuration monitoring rules
  - Records all configuration changes
  - Stores in dedicate S3 bucket
  - Creates Configuration dashboard
  - Search for all your resources in the region
- Rule samples – Select rules, e.g.
  - S3 buckets versioning
  - Run over all S3 buckets – produce report on compliance
  - Link to Manage resources



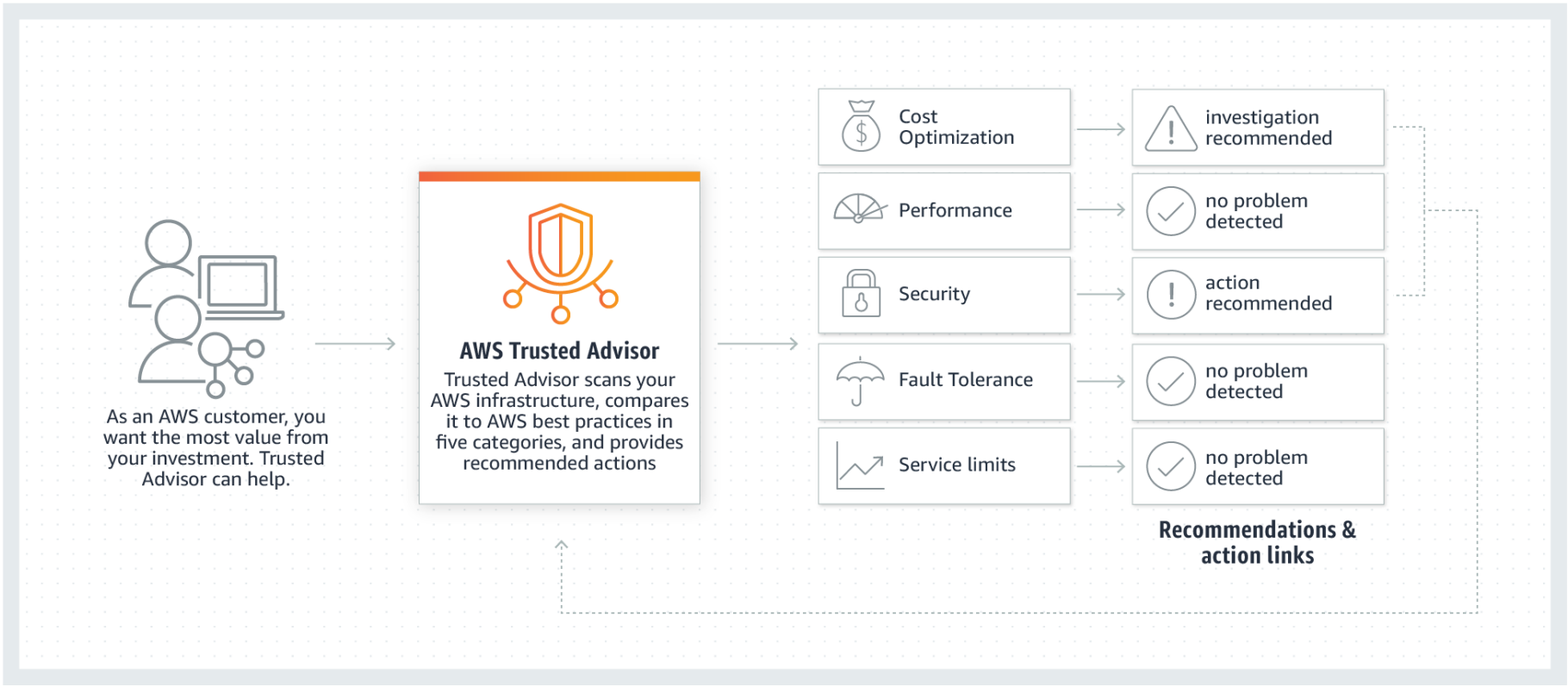


AWS CloudTrail



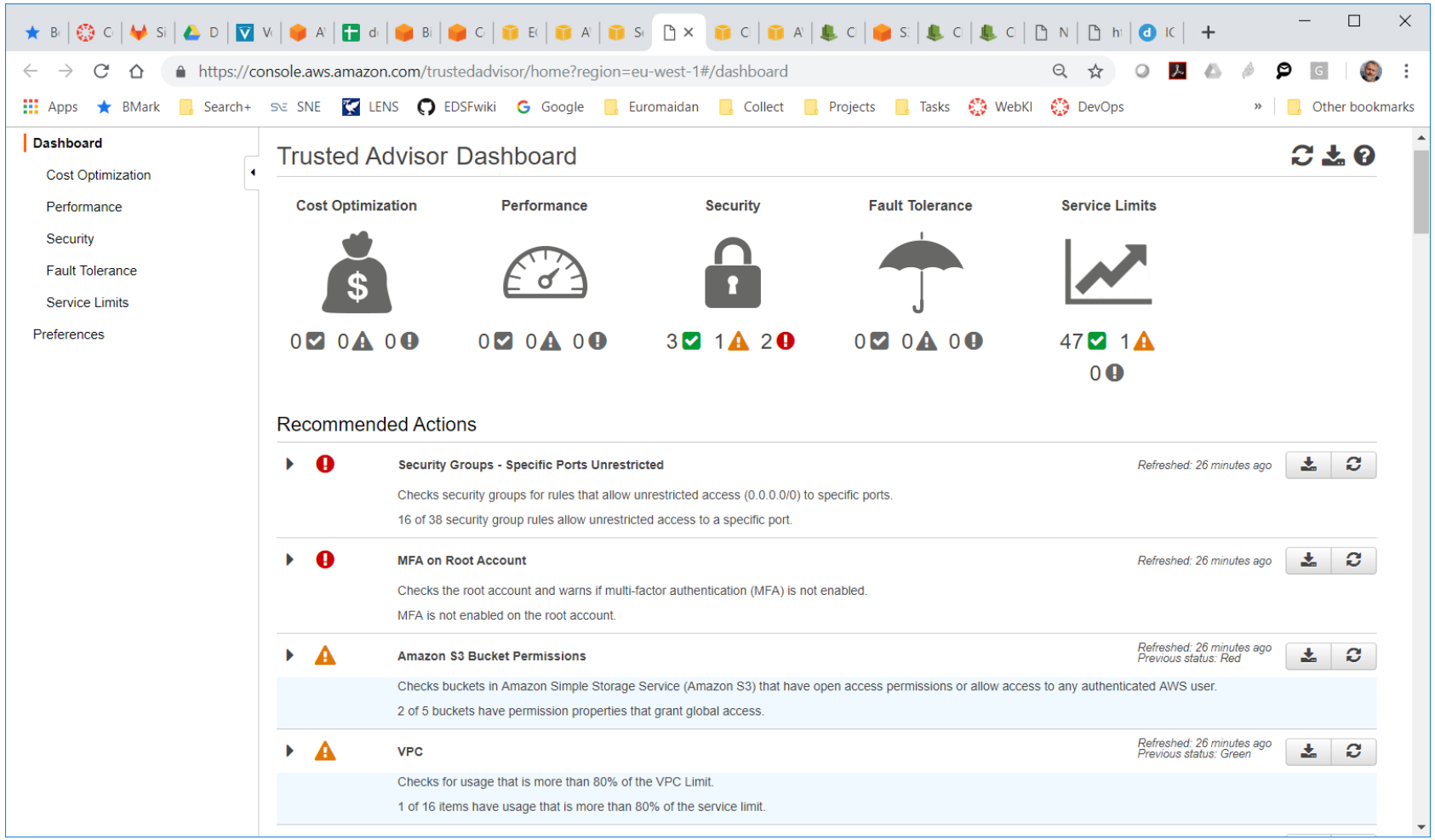
- Create and store logfile in S3
- Look up API history
- Example of S3 bucket
  - Log all/selected operations as events

Trusted Advisor



- Like combining CloudWatch and CloudFormation + number of preset rules
  - Cost optimisation
  - Performance
  - Security
  - Fault tolerance
- Especially recommended for checking security issues

Trusted Advisor



- Example Trusted Advisor

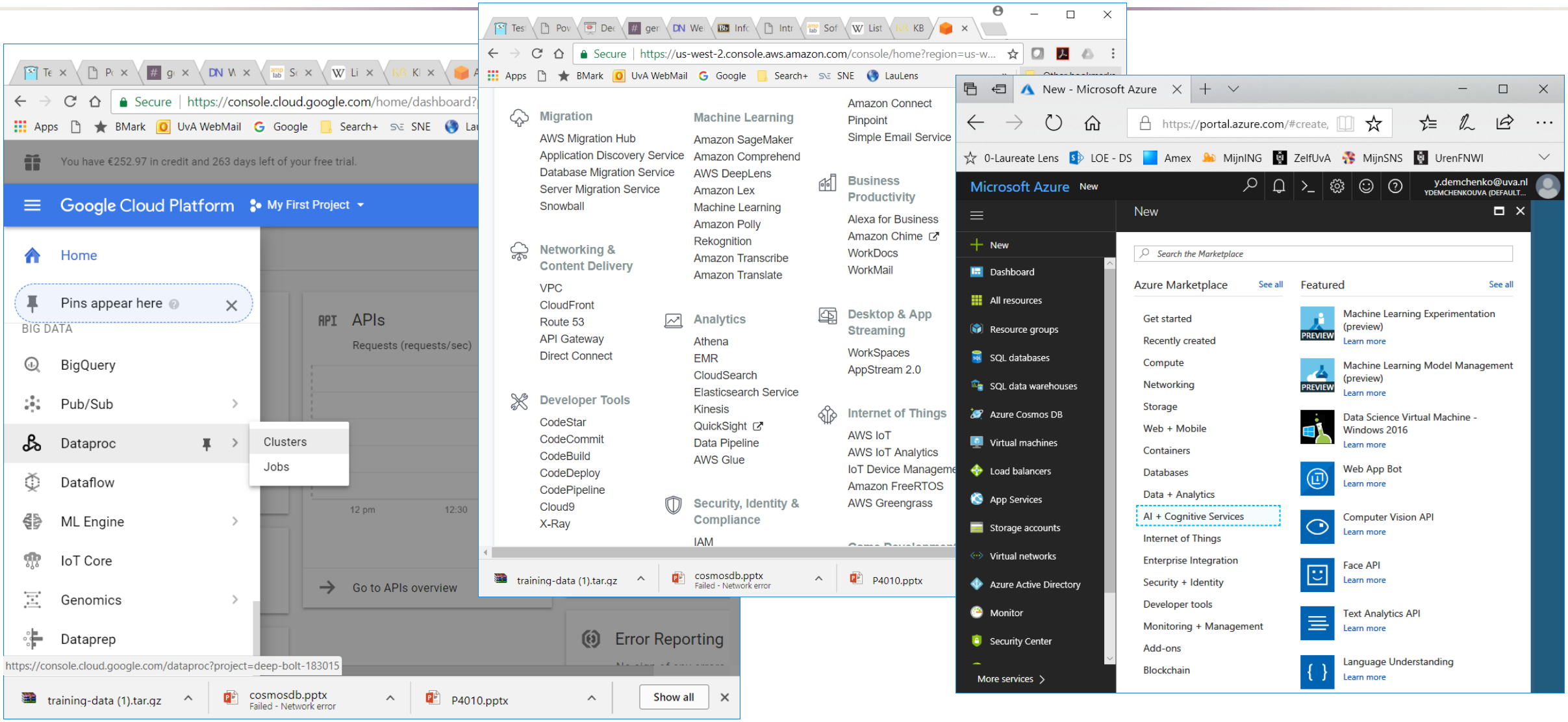
Summary and Take away

- Cloud computing is presently a mainstream technology widely used by business and industry
- Cloud Computing technology is well defined and has sufficient standardization base and best practices
- Cloud Computing technology/ecosystem defines a number of new actors and stakeholders
- Presented basic use cases illustrate the main cloud features and opportunities
  - Use them, refer to them when you need to decide on an appropriate scenario for cloud implementation at your company

Additional Information: Overview Cloud Service Providers

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)

AWS, Azure, Google Architectures and Services



Examples Cloud IaaS - Amazon AWS

- AWS Service Groups
- AWS Cloud Architecture
- Amazon EC2 Component Services
- AWS Machine Instances
- Network services at AWS
- Security groups and Virtual Private Clouds (VPC)

Amazon Web Services (AWS) Cloud

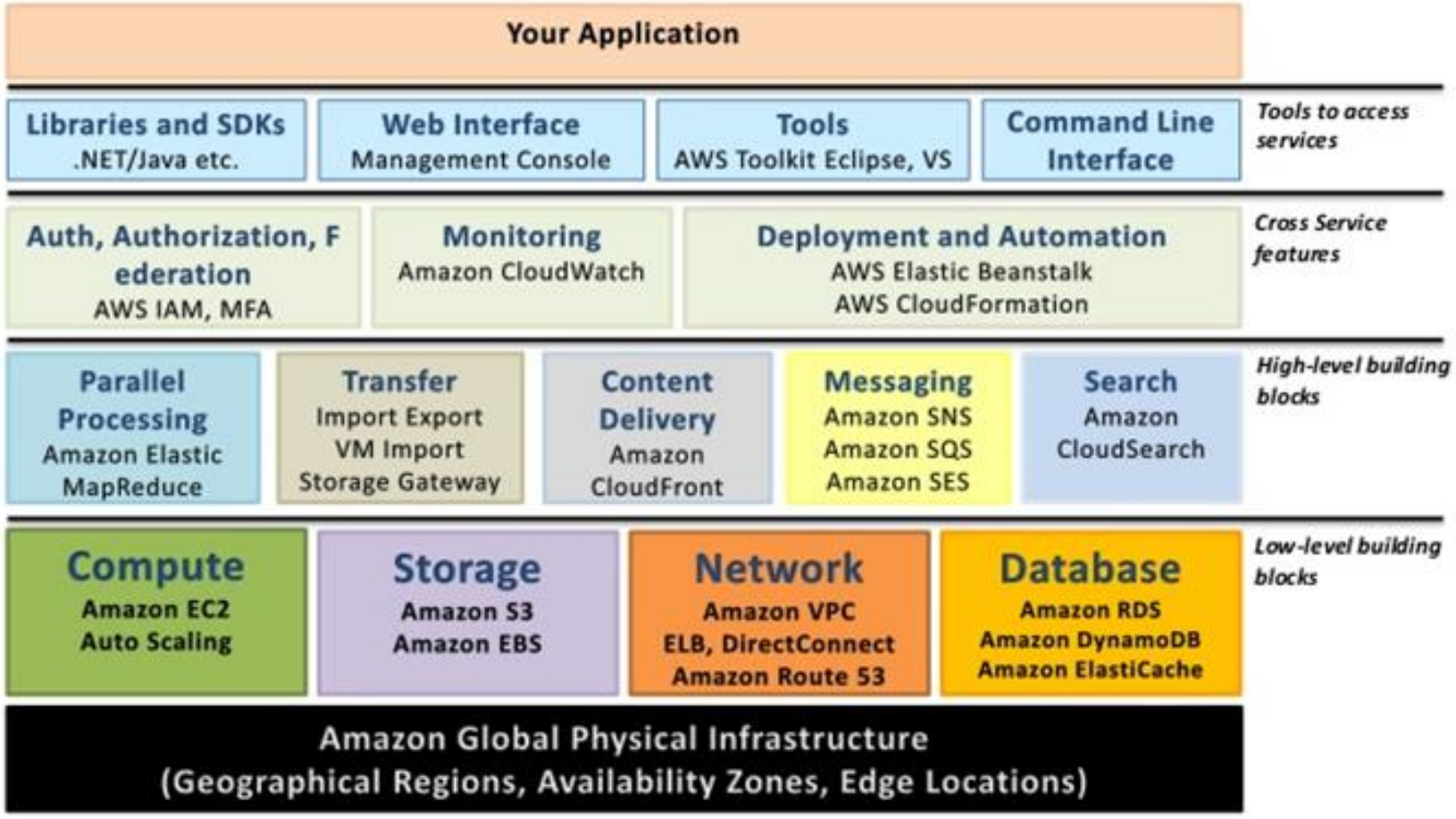
- AWS service groups
  - Compute: Elastic Compute Cloud (EC2) and EC2 Container service
  - Storage and Content Delivery: Simple Storage Service (S3), CloudFront, Glacier, etc.
  - Database: RDS, Dynamo, ElastiCache, Redshift
  - Networking: VPC, Direct Connect, Rout 53
  - Security and Identity Management: IAM (Identity and Access management), Directory, etc.
  - Developer tools
  - Management and Governance
  - Analytics and Machine Learning
  - Mobile Services
  - Applications Integration
  - Enterprise Applications
  - Internet of Things

Amazon EC2 and S3 API became a standard-de-facto interfaces for accessing and managing cloud services

- Majority of existing cloud management platforms offer EC2 and S3 interfaces

- AWS has 14+ availability zones to be choose from regions:
  - US Standard (default), US West (Oregon), US West (Northern California)
  - Europe: Ireland, Frankfurt, London, Paris
  - Asia Pacific (Singapore), Asia Pacific (Tokyo), Asia Pacific (Sydney), Asia Pacific (Mumbai)
  - South America (Sao Paulo)
  - GovCloud (US) Regions. The US Standard Region automatically routes requests to facilities in Northern Virginia or the Pacific Northwest using network maps

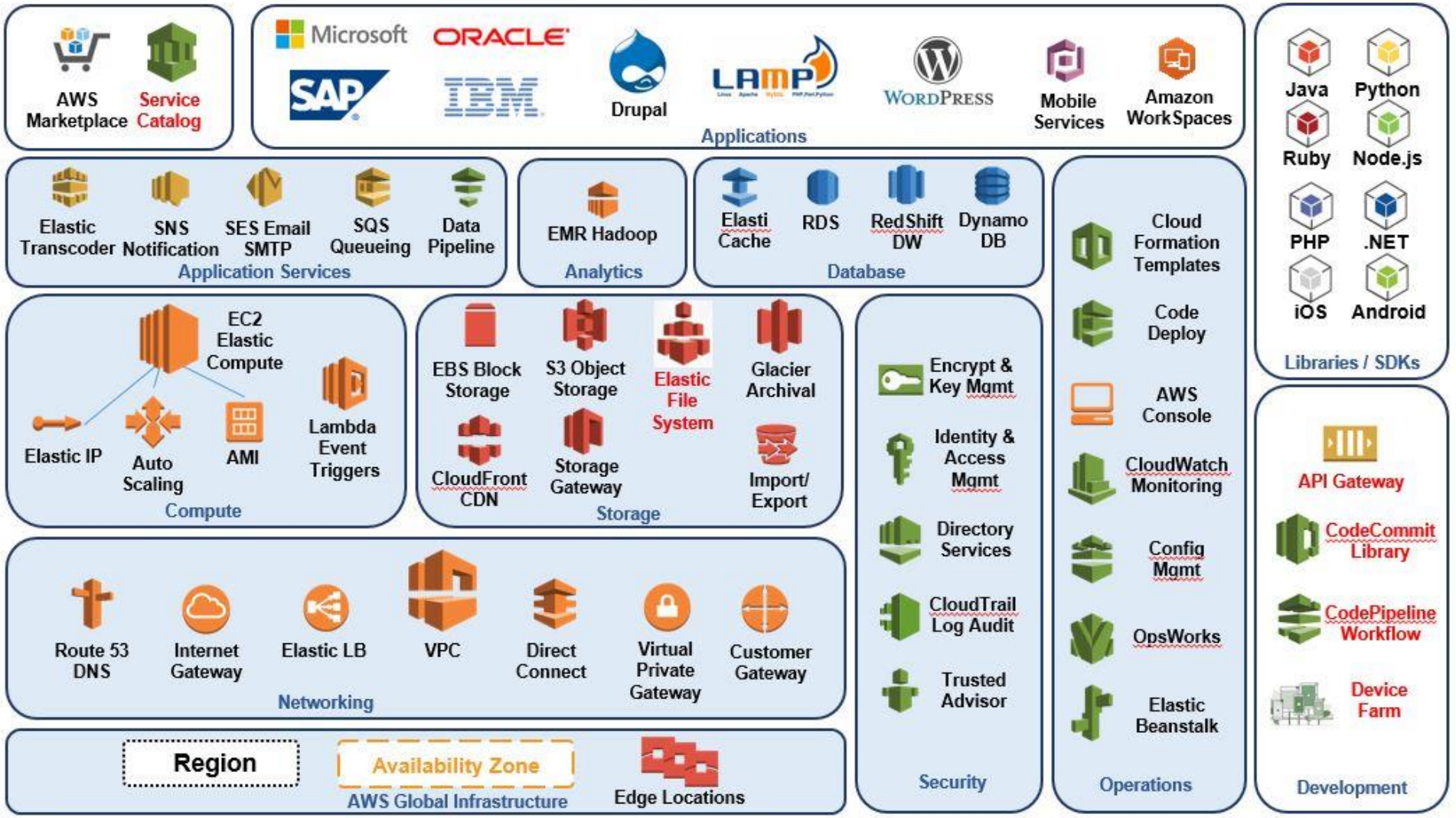
Amazon AWS Cloud Architecture (historical)



Credits “Building Powerful Web Applications in the AWS Cloud” by Louis Columbus
http://softwarestrategiesblog.com/2011/03/10/building-powerful-web-applications-in-the-aws-cloud/

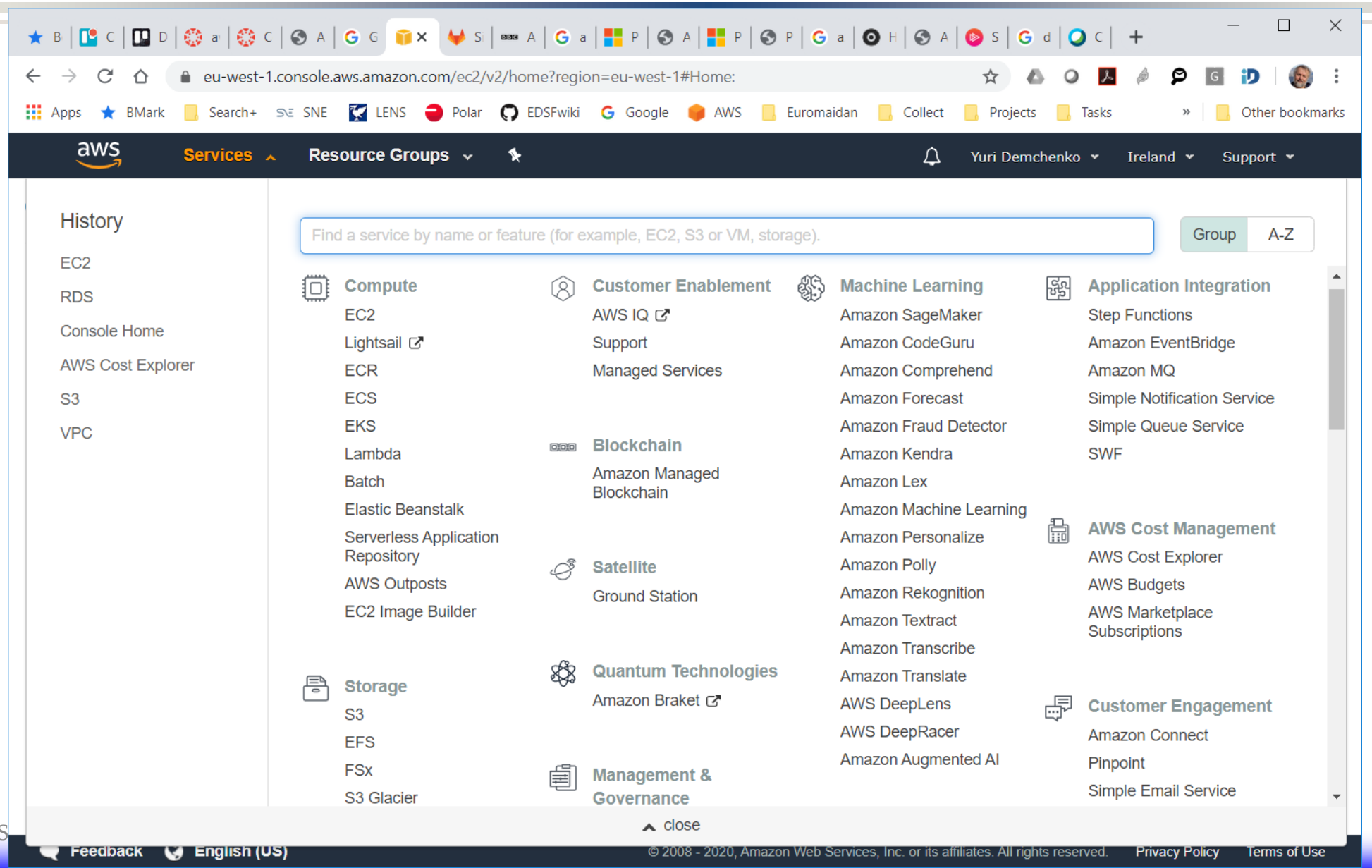
- AWS high level architecture of 2011
- Representing classic IaaS service model
- Currently AWS services spans from IaaS to PaaS to SaaS
- Cover new services type: Edge and microservices

AWS Products (est. 2015)



AWS icons - https://aws.amazon.com/architecture/icons/

Amazon AWS Console – Products (2020)



Amazon EC2 Component Services

- *EBS Elastic Block Store* provides highly available and reliable storage volumes that are independent and persistent across the life of an VM/AMI
- *VPC (Virtual Private Cloud)* allows organizations to use AWS resources along with their existing infrastructure in a VPN (Virtual Private Network) to increase compute capabilities beyond the local resources.
- *Elastic IP Address* is a persistent IP address which can be allocated to a particular user account and allows the user to dynamically assign it to any user VM.
- *CloudWatch* is a service to monitor AWS resource utilization, operational performance, and overall demand patterns
- *Auto Scaling* allows users to dynamically provision and relinquish resources used by their applications depending upon demand in run-time
- *Elastic Load Balancing* can balance load between multiple VMs located within a single availability zone or multiple zones
- *VM Import/Export* is a service that provides functionality to upload custom VM images and store a live instance as an image

Amazon EC2: Amazon Machine Instances (AMI)

EC2 AMIs (Amazon Machine Instances) forms the basic infrastructure on which applications can be deployed just as any other server

- ***Reserved* instances**, which can be purchased with a one time payment for a long term reservation and are available at considerably lower prices
- ***On-Demand* instances**, which are for immediate demands but with comparatively higher prices
- ***Spot* instances**, which is unused capacity for which users can bid for

AMI allows the following controls over the provisioned infrastructure:

- Location of the AMIs and management of static IP addresses
  - Using one of eight availability zones for a user to choose from:
- Network management and access control to the AMI configuration
- Use of a Web based management console
- For professional use, developers have an option to use **AWS command line tools** which allows them to write scripts to automate this management (e.g., Ruby, bash, python)

Amazon EC2 Machine Instances Types

VM instances are optimised for different types of applications and use cases (Amazon EC2 Instances http://aws.amazon.com/ec2/instance-types/ price https://aws.amazon.com/ec2/pricing/on-demand/):

- M3 - General Purpose
- C3 - Compute Optimized
- R3 - Memory Optimized
- G2 - GPU
- I2 - Storage Optimized
- HS1 - High storage density
- T1 - Low cost micro instances

Name	vCPUs	Baseline Perf	RAM (GiB)	CPU Cred/Hrr	Price/Hr (Linux)	Price/ Month
t2.micro	1	10%	1.0	6	$0.013	$9.50
t2.small	1	20%	2.0	12	$0.026	$19.00
t2.medium	2	40%	4.0	24	$0.052	$38.00

Example 1: M3 instance provides a balance between compute, memory, and network resources:

- High Freq. Intel Xeon E5-2670 (Sandy Bridge) Processors
- SSD-based instance storage for fast I/O performance
- Balance of compute, memory, and network resources

M3 instances come in configurations:

- m3.medium (1 core, 3.75 GB, SSD 1 x 4 GB)
- m3.large (2 core, 7.5 GB, SSD 1 x 32 GB)
- m3.xlarge (4 core, 15 GB, SSD 2 x 40 GB)
- m3.2xlarge (8 core, 30 GB, SSD 8 x 30 GB)

C3 instances are compute-optimized, using highest performing processors

- High Frequency Intel Xeon E5-2680 v2 (Ivy Bridge) Processors
- Support for Enhanced Networking
- Support for clustering
- SSD based instances storage

C3 instances come in configurations:

- c3.large (2 core, 3.75 GB, SSD 2 x 16 GB)
- c3.xlarge (4 core, 7.5 GB, 2 x 40 GB)
- c3.2xlarge (8 core, 15 GB, 2 x 80 GB)
- c3.4xlarge (16 core, 30 GB, 2 x 160 GB)
- c3.8xlarge (32 core, 60 GB, 2 x 320 GB)

Network services in Amazon Cloud

- Each EC2 instance is associated with one private and one public IP address
  - The private IP address is used within the EC2 LAN (Local Area Network)
  - The public IP Address is advertised to the Internet and has a 1:1 NAT (Network Address Translation) mapping to the private IP address of the EC2 instance
- The Elastic IP address is a public IP address accessible on the Internet and is associated with user's EC2 account
  - A user can associate this IP address to any particular EC2 instance rented with that account
- Amazon Route 53 is a DNS (Domain Name Server) service provided by Amazon to map EC2 instance IP addresses to a domain names
  - Recently introduced Latency Based Routing (LBR), to route application users to AWS end-points (EC2 instance) which have the best performance (least latency) at the time of request
- Elastic Load Balancer (ELB) is realised by internal network fabric of the Amazon Data Center
- Direct Connect allows connecting with dedicated line to one of Amazon data centers
  - Require user management router placing at Amazon location

Security Groups in Amazon Cloud

Each EC2 instance is set up with a firewall restricting connectivity

Communications are controlled by SECURITY GROUPS

- A security group acts as a virtual firewall
- Controls the traffic for one or more instances
- Add rules to each security group that allow traffic to or from its associated instances.

Rules include

- TCP and UDP, or a custom protocol
- The range of ports to allow ICMP. The ICMP type and code
- One or the following options for the source (inbound rules) or destination (outbound rules):
  - An individual IP address, in CIDR notation.
  - An IP address range, in CIDR notation (for example, 203.0.113.0/24).
  - The name (EC2-Classic) or ID (EC2-Classic or EC2-VPC) of a security group.

Amazon Virtual Private Cloud (Amazon VPC)

VPC is a logically isolated section of the AWS Cloud where customer provisioned resources are interconnected by a virtual network defined by customer

- Complete control over created virtual networking environment, including selection of own IP address range, creation of subnets, and configuration of route tables and network gateways
  - Create a public-facing subnet for your webservers with access to the Internet
  - Place your backend systems such as databases or application servers in a private-facing subnet with no Internet access
  - Use Network Address Translation (NAT) to connect VM instance from a private subnet to Internet
- Connect enterprise network via Virtual Private Network (VPN) and extend enterprise datacenter into AWS cloud
- Leverage multiple layers of security, including security groups and network access control lists in each subnet

Amazon Big Data Services

- SageMaker
- NoSQL databases
- EMR/Hadoop + Spark
- Messaging and stream processing: Kafka, Flume, Storm
- Data Analytics and Machine Learning

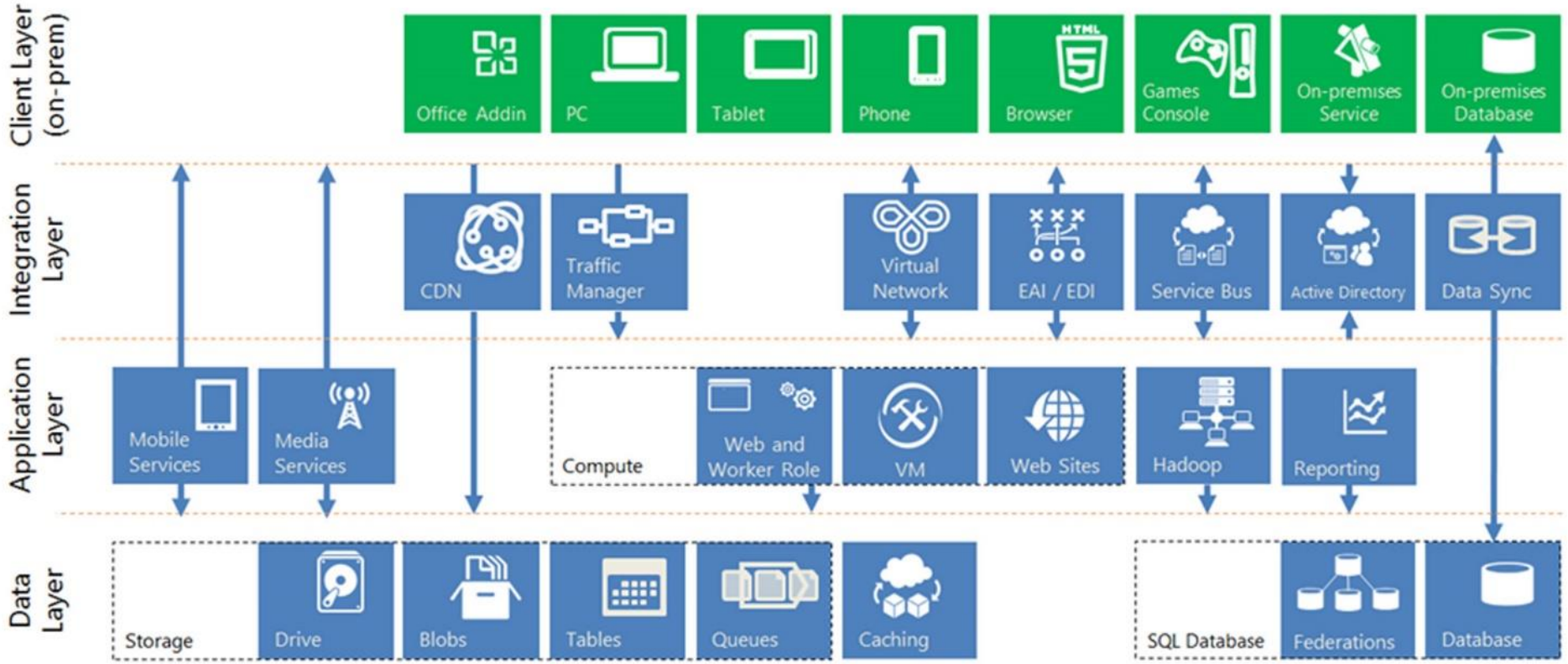
Microsoft Azure Cloud

- Microsoft Azure Cloud Architecture
- Microsoft Azure Application building blocks

Microsoft Azure Cloud

- Highly Big Data and business oriented general cloud oriented platform
  - Since April 2014 renamed from Windows Azure to Microsoft Azure
  - Strong growth in the last two years
- Microsoft Azure provides a comprehensive set of services that can be selectively compose to build user cloud apps
  - Global Data Center Footprint
    - 99.95% Monthly SLA. Pay only for what you use.
  - Flexible & Open Compute Options
    - Virtual Machines, Linux OS, Windows Server (only), Big Data, Data Analytics and Business Intelligence Services and platform
  - Managed Building Block Services
    - SQL Database, Cache, Service Bus,
  - Multiple Languages and reach development and DevOps

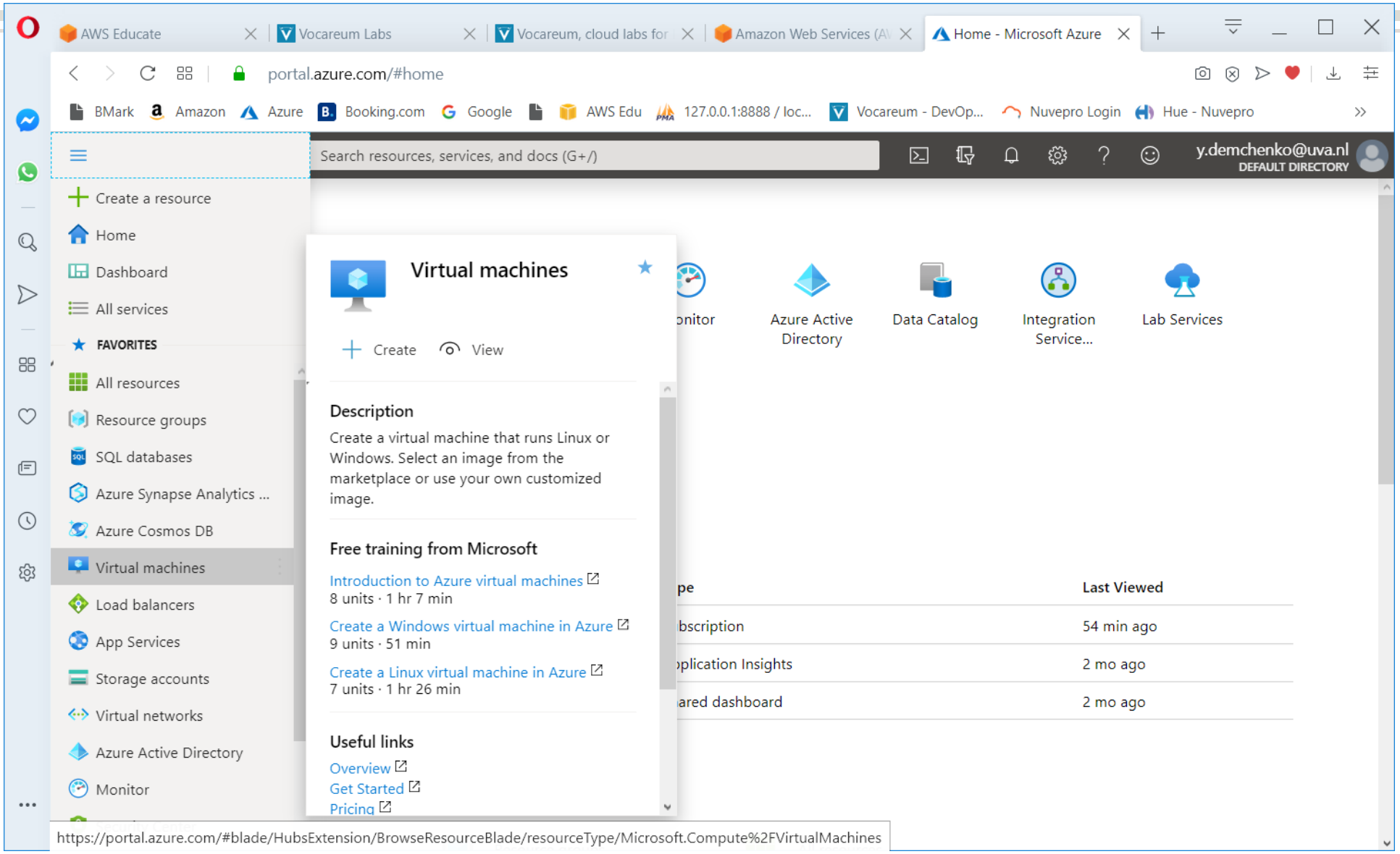
Microsoft Azure Architecture (details)



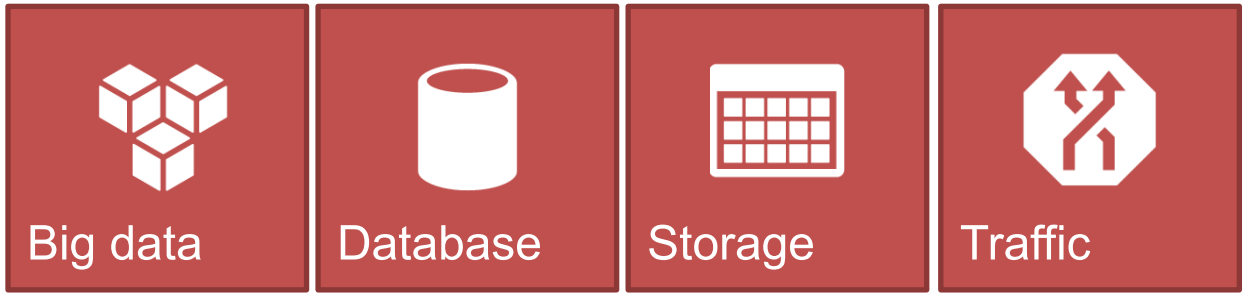
Functional Layers

- Data Layer
- Application Layer
- Integration Layer
- Client Layer (on premises)

Azure Console: Resource Menu from Homepage



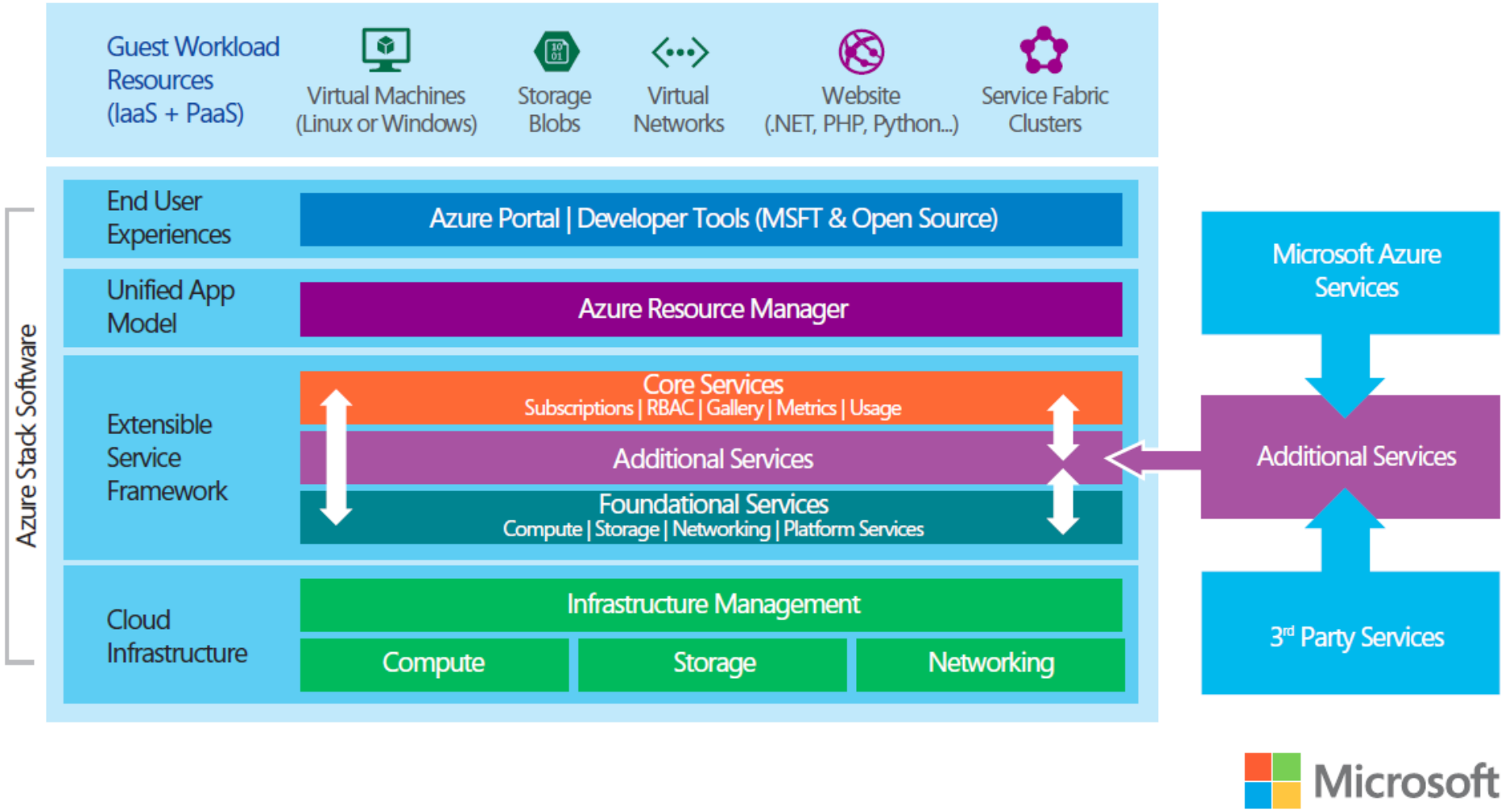
Microsoft Azure Application building blocks



Microsoft Azure application building blocks can be used by customer applications

Microsoft Azure Stack for Hybrid Cloud Model

Azure Stack architecture summary



Azure Stack is a hybrid cloud platform that enables you to use **Azure** services from your company or service provider datacenter.

- Released 2017

Microsoft

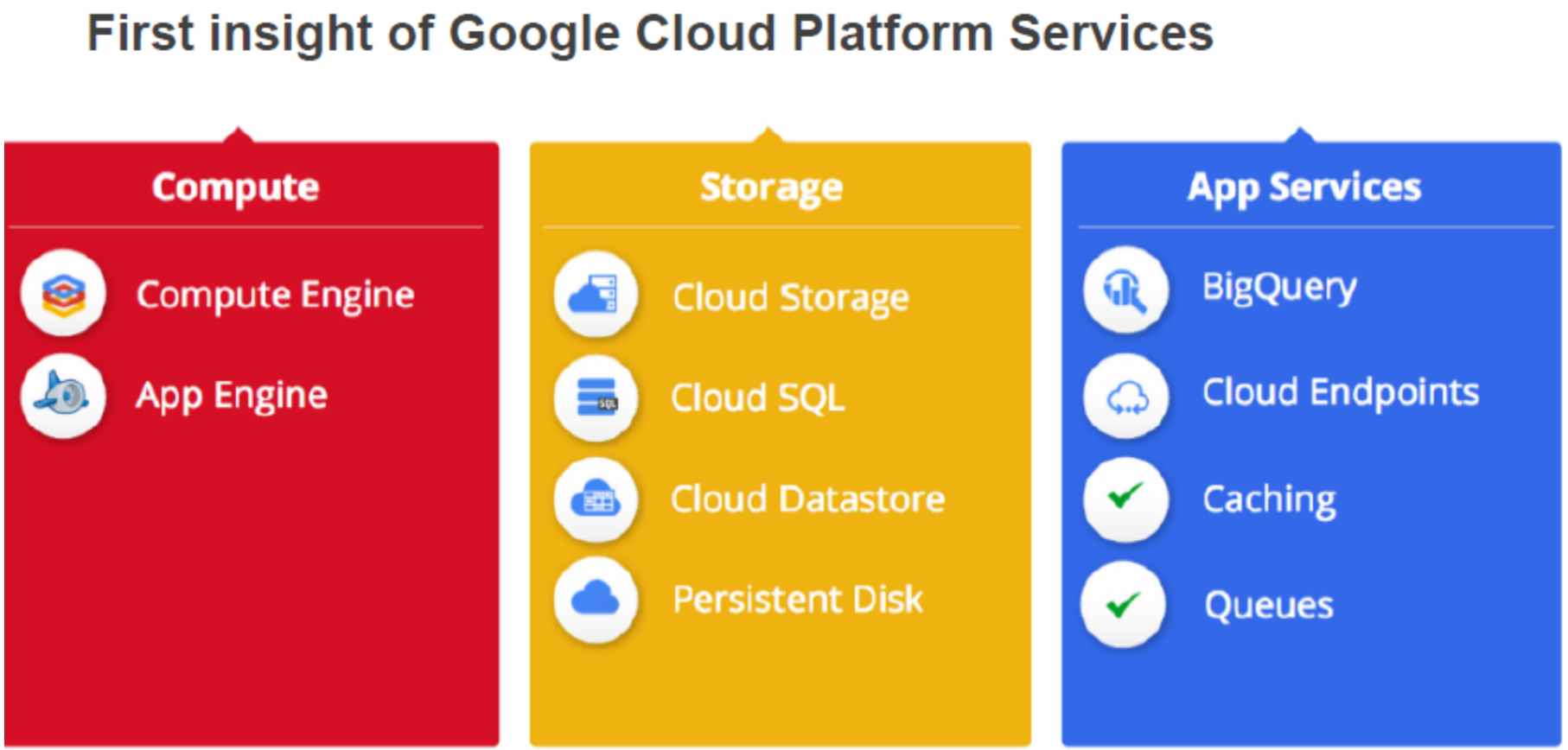
Subscription and Licensing of Azure Stack

- Azure Stack has an interesting business model, uses subscription price much like Azure. you pay only for what you use, just like Azure.
- You can pay per hour or per month, with a Base VM charge of $0.008/vCPU/hour or $6/vCPU/month).
- You can also use your existing Windows Server or Linux licenses.
- If you don’t have any licenses, then a Windows Server VM charge at $0.046/vCPU/hour or $34/vCPU/month.
- Similar pricing is for storage and for the Azure App Services, based on vCPU usage.
- As expected, it’s cheaper than Azure’s pricing for similar features, but here you’re paying for the hardware yourself, and that’s not going to come cheap.
- The biggest advantages of cloud services has been a shift from capital to operational expenditure. And Azure Stack is introducing that same model in house.

Google Cloud Platform (GCP)

- Compute
  - AppEngine
  - Google Functions (serverless with Node.js)
- Storage: Static, sharing, backup, for applications and computation
  - Cloud Spanner SQL database
- Big Data
  - BigQuery – Hadoop Data Warehouse
- Machine Learning services
  - Translate
  - Prediction
- Cloud endpoints

Google Cloud Platform (GCP) structure



Google App Engine (**GAE**) is a Platform as a Service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers.

Java
Python
PHP
GO

Google's Go:

- Go is an Google's open source programming environment.
- Tightly coupled with Google App Engine.
- Applications can be written using App Engine's Go SDK.
- C-type but optimised for concurrency in clouds

The sandbox:

- All hosted applications run in a **secure environment** that provides limited access to the underlying operating system.
- **Sandbox isolates the application** in its own secure, reliable environment.
- Limitations imposed by sandbox (for security):
  - An application can only access other computers over internet using the provided URL fetch and email services.

Free ???

Yes, free for up to 1 GB of storage and enough CPU and bandwidth to support 5 million page views a month. 10 Applications per Google account.

Machine Learning Focus

- Machine Learning embedded across most products
- Multiple Tensorflow ML models in use
  - Portable TensorFlow models
- Key models exposed via APIs (Democratizing Machine Learning)
  - Cloud Video Intelligence API
  - Cloud Vision API
  - Cloud Natural Language API
  - Cloud Translation API
  - Cloud Speech API

- Acquired Kaggle in 2017 - Data Science Enthusiasts

Google Machine Learning

- Support all stages of ML workflow
- Dataprep: Serverless platform for all stages of the analytics data lifecycle

Google Collaboratory

https://colab.research.google.com/#scrollTo=Nma_JWh-W-IF

- Colab, or ‘Colaboratory’, allows you to write and execute Python in your browser
  - Zero configuration required
  - Free access to GPUs
  - Easy sharing
- Application domains:
  - Data Science and Analytics
  - Machine Learning
- Colab subscriptions
  - Free
  - Pro – 9.99 $/mo
  - Pro+ - 49.99 $/mo