

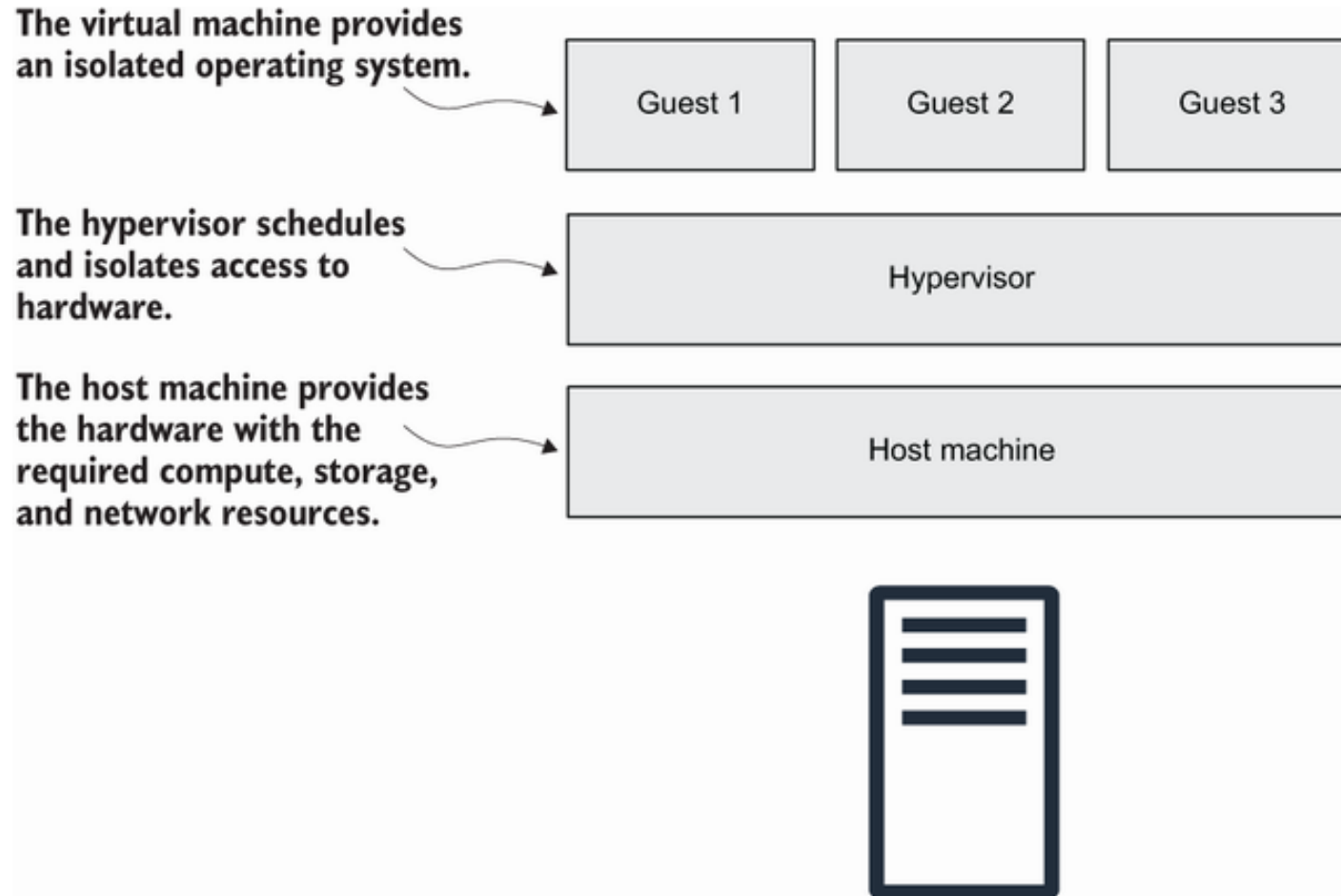
## Public Cloud Services IaaS auf Hyperscaler



# Costs...



# What is IaaS?



# Product Class

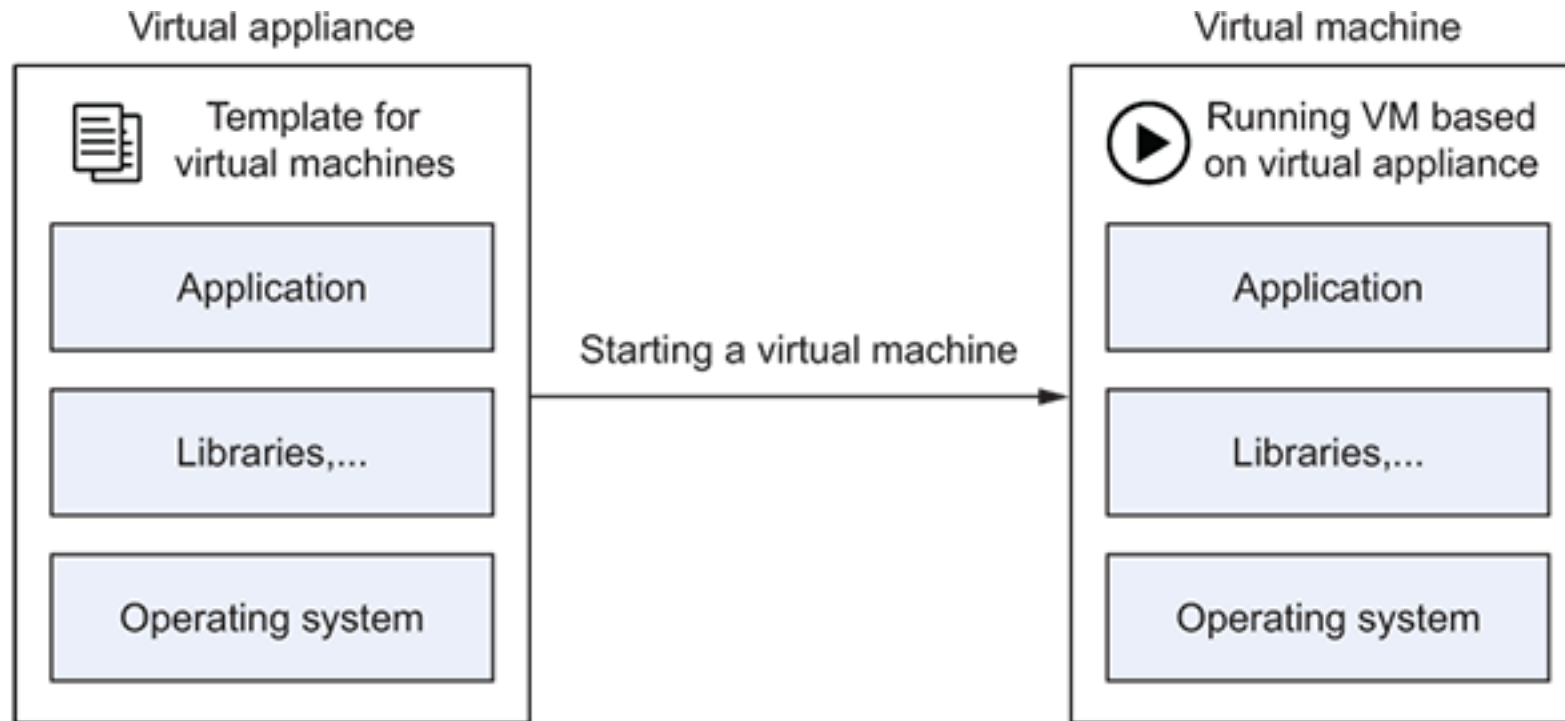
- Virtual Computing
- Basiert auf einem AMI
- CPU/RAM: Fixed Instance Types with defined Setups
- Disks: Extra Product, flexible to be added
- Netzwerk: Network Interface, connects to subnet, to be managed externally
- Possibility to scale
- Tags, Tags, Tags
- Access only via SSH Public/Private Key



Amazon  
**EC2**



# Templates

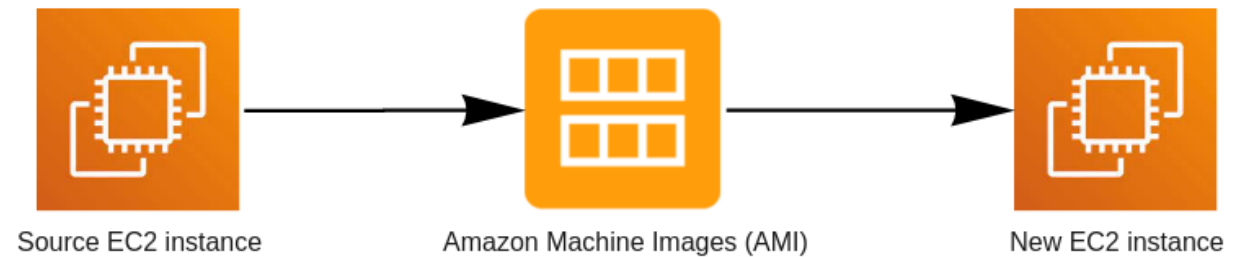




# Kinds of Templates

## AMI:

- Amazon Machine Image / Azure Machine Image
- Image format for EC2-Instances / Compute instances
- Can be derived from running instance of build via IaC.



## Common Use Case: Golden Image

## Why to work with Golden Images?

# Generating Images without Instance

## Packer:

- Hashicorp Tool utilizing HCL
- Automating generation of AMI (and all kinds of other images)
- Widely used to generate Image directly in any kind of pipeline

```
source "amazon-ebs" "ubuntu" {  
    ami_name      = "my-first-packer-image"  
    instance_type = "t2.micro"  
    region        = "us-east-1"  
    source_ami     = "ami-0557a15b87f6559cf"  
    ssh_username  = "ubuntu"  
}
```

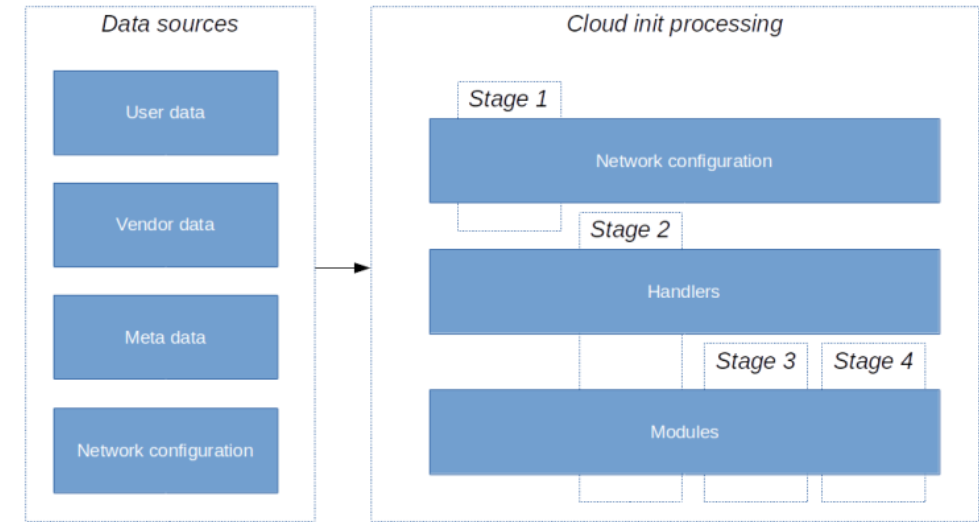
```
build {  
    name      = "ubuntu-nginx-image"  
    sources = ["source.amazon-ebs.ubuntu"]
```

```
    provisioner "shell" {  
        inline = [  
            "sudo apt update -y",  
            "sudo apt install nginx -y",  
            "git clone https://github.com/example/my-app.git  
/home/ubuntu/app",  
            "sudo ufw allow 'Nginx HTTP'",  
            "sudo systemctl enable nginx",  
            "sudo systemctl start nginx"  
        ]  
    }  
}
```

# Generating Images without Instance

## Cloud-Init:

- Data Sources: Input provided by provider / User / etc.
  - Can be jinja2 / yaml-structures representing folders
- Different Stages within the boot process consume the sources
  - Network setting
  - Handlers for taking specific data caring about it specifically
  - Modules for specific further settings



users:

- default
- name: eval
- gecos: eval
- primary\_group: eval
- sudo: ALL=(ALL) NOPASSWD:ALL
- groups: users, admin
- shell: /bin/bash
- lock\_passwd: false
- ssh\_authorized\_keys:
  - ecdsa-sha2-nistp521

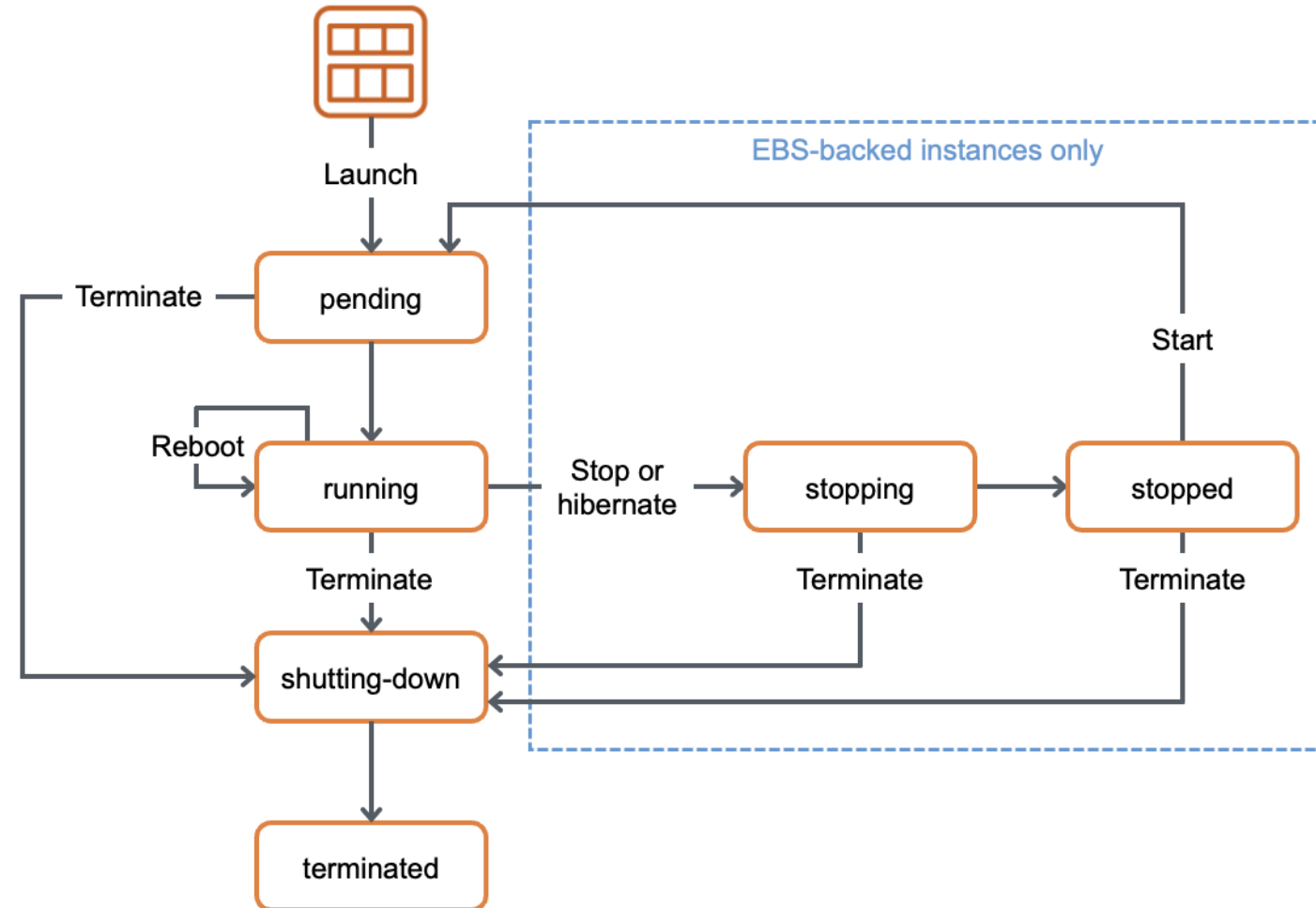
```
AAAAE2VjZHNhLXNoYTItbmlzdHA1MjEAAAABmlzdHA1MjEAAACFBAGd1sEHWR+J1kz4TokLXzp
TFGFO8dX3T1zWjQ0rJqsnrx1m8nTotpWTuqgQCgtlzQ8Usvy4wK3/pRV1raFtYThvEgHleB85YOaSi
FEYs1rUz6KkQ8lhKuSXYLp8YnJtV0MCJNfm8jY816RvOqa+v7mS/+67ly4PXwf1jfibzw1bSZHc4w==
snorwin@nano-x1
```



# Differences and Similarities to Switch Engines? Templating...

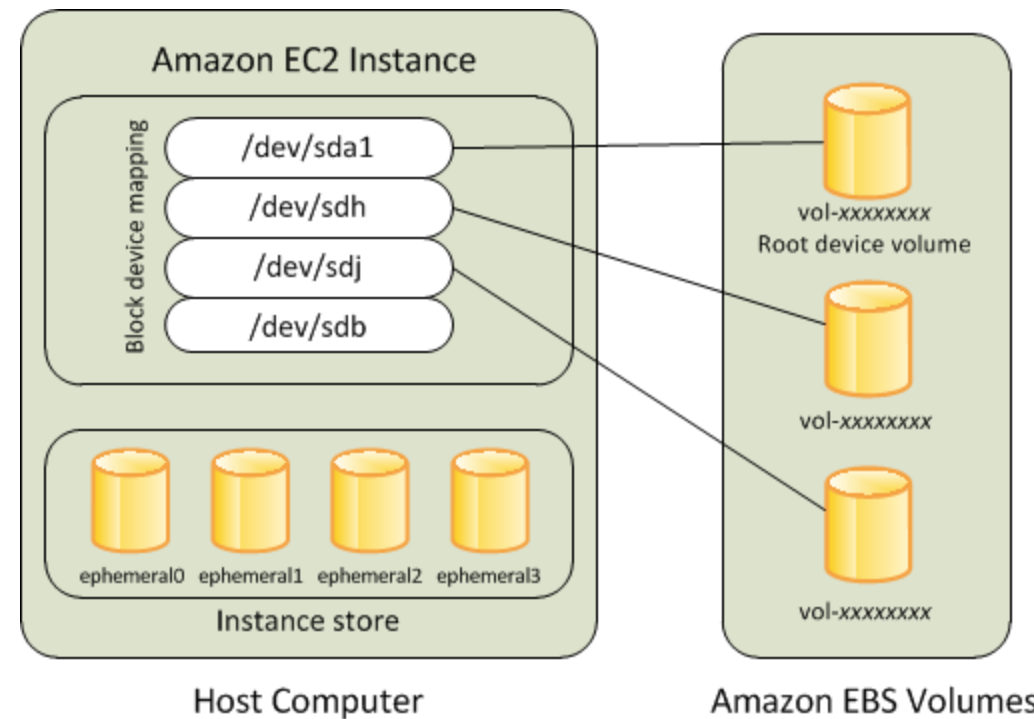
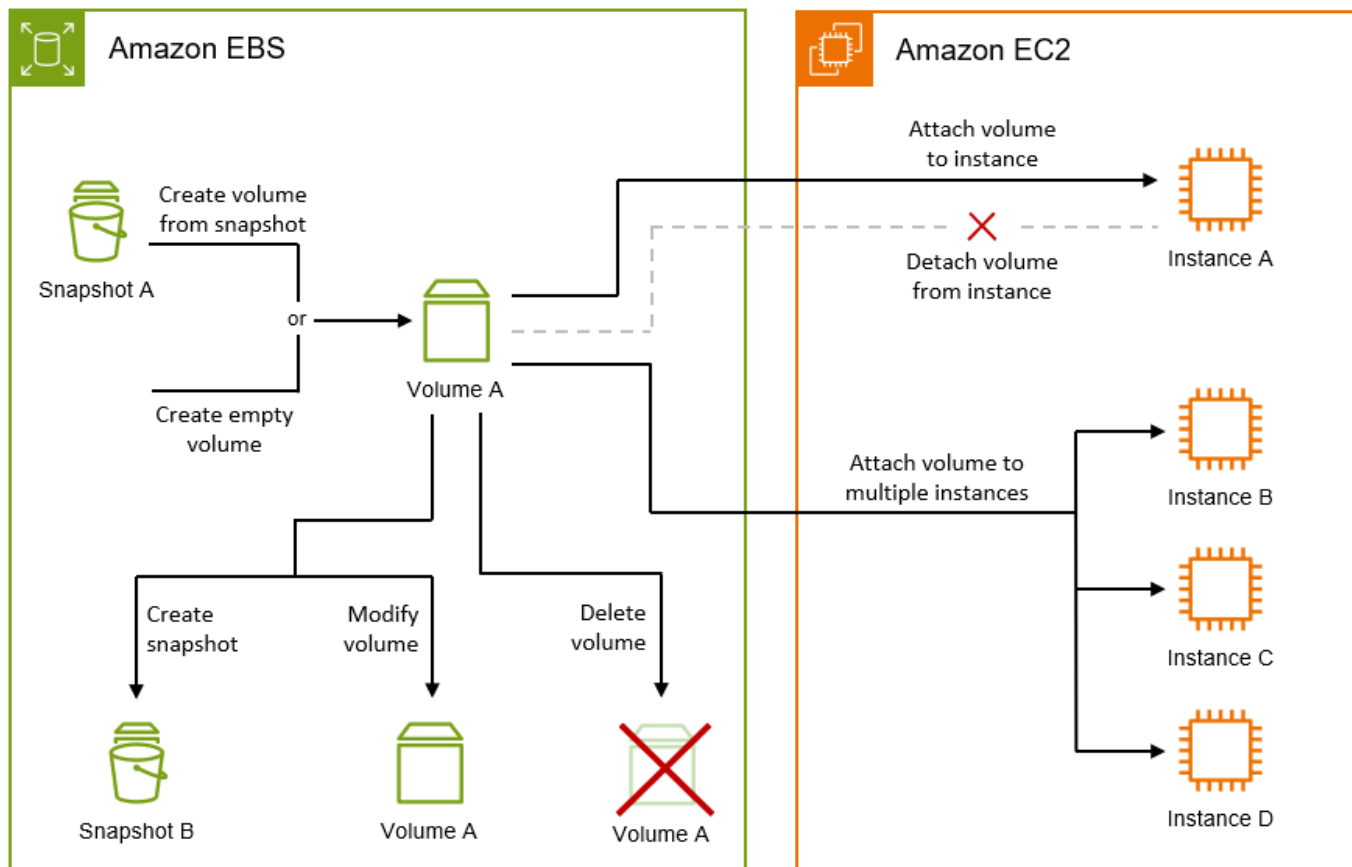
# Instance lifecycle

**DEMO**



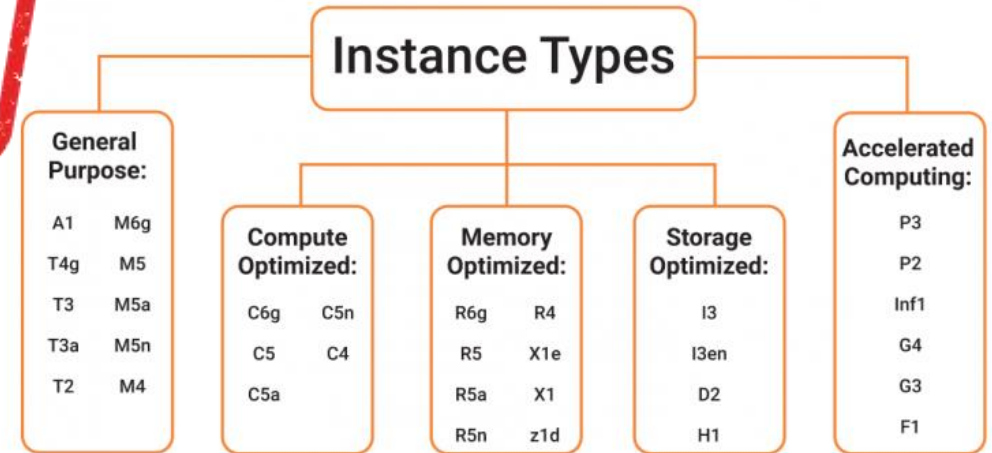
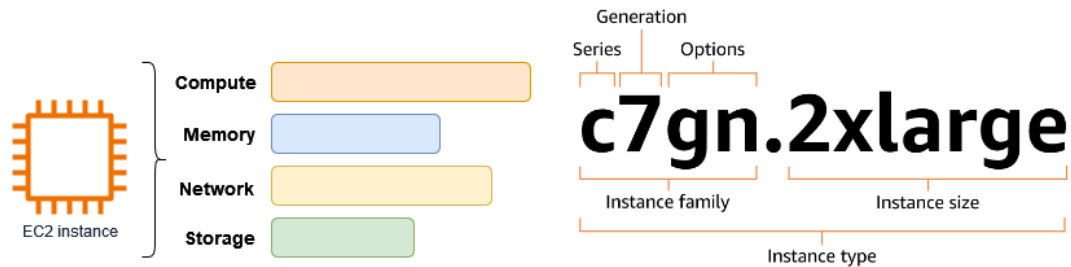


# EBS-Volumes



# Instance Classes

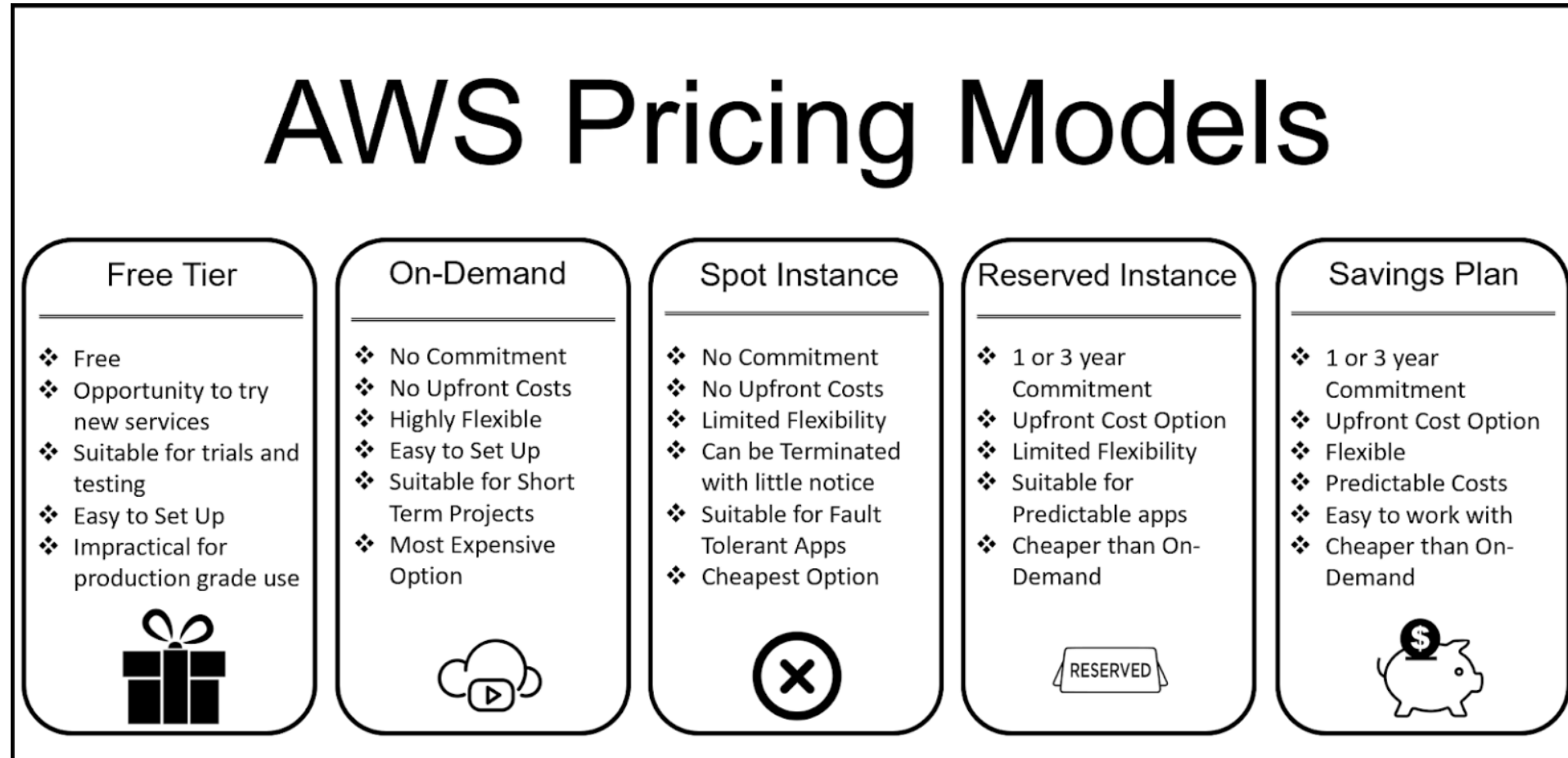
**DEMO**



Azure VM Types						
	General Purpose	Compute Optimized	Memory Optimized	Storage Optimized	GPU	High Performance Compute
Type	Av2, B, DCsv2, Dv2, Dsv2, Dv3, Dav4, Dasv4, Ddv4, Ddsv4, Dv4, Dsv4	Fsv2	M, Mv2, Dv2, Dsv2, Ev3, Esv3, Eav4, Easv4, Ev4, Esv4, Edv4, Edsv4	Lsv2	NC, NCv2, NCv3, ND, NDv2, NV, NVv3, NVv4	H, HBv2, HC, HB
Description	Balanced CPU and Memory	High Ratio of Compute to Memory	High Ratio of Memory to Compute	High disk Throughput & IO	Specialized with Single or Multiple NVIDIA GPUs	High Memory and Compute Power-fastest & most Powerful
Uses	Testing & Development, Small-medium databases, Low-medium traffic Webservers	Medium Traffic Web Servers, Network Appliances, Batch Processing, App Servers	Relational Databases, Services, Analytics, Larger Caches	Big Data, SQL, NoSQL Databases	Compute Intensive, Graphics-intensive, Visualizations Workload	Batch Processing, Analytics, Molecular, Modelling, Fluid Dynamics, Low Latency RDMA Networking

# Differences and Similarities to Switch Engines? Lifecycle, Blockstorage, Instances...

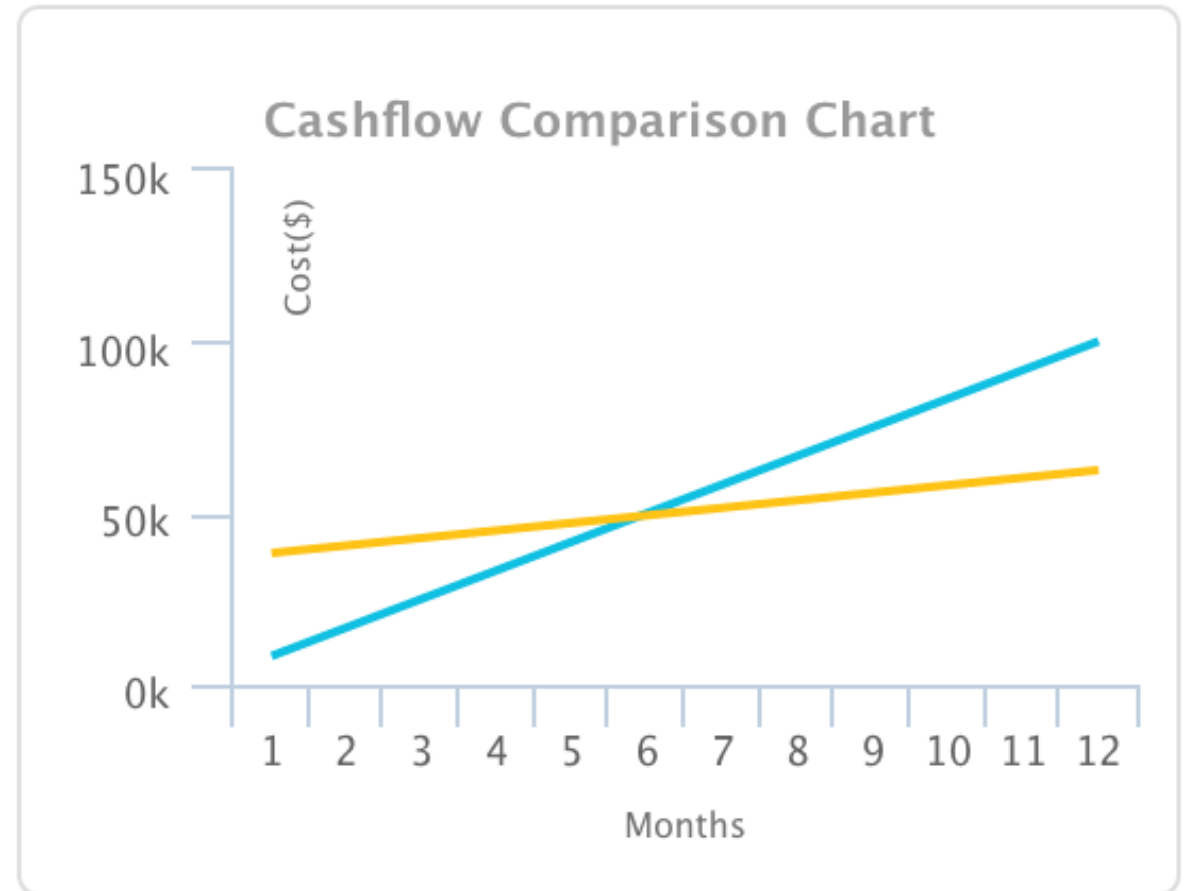
## Optimizing Costs, IaaS



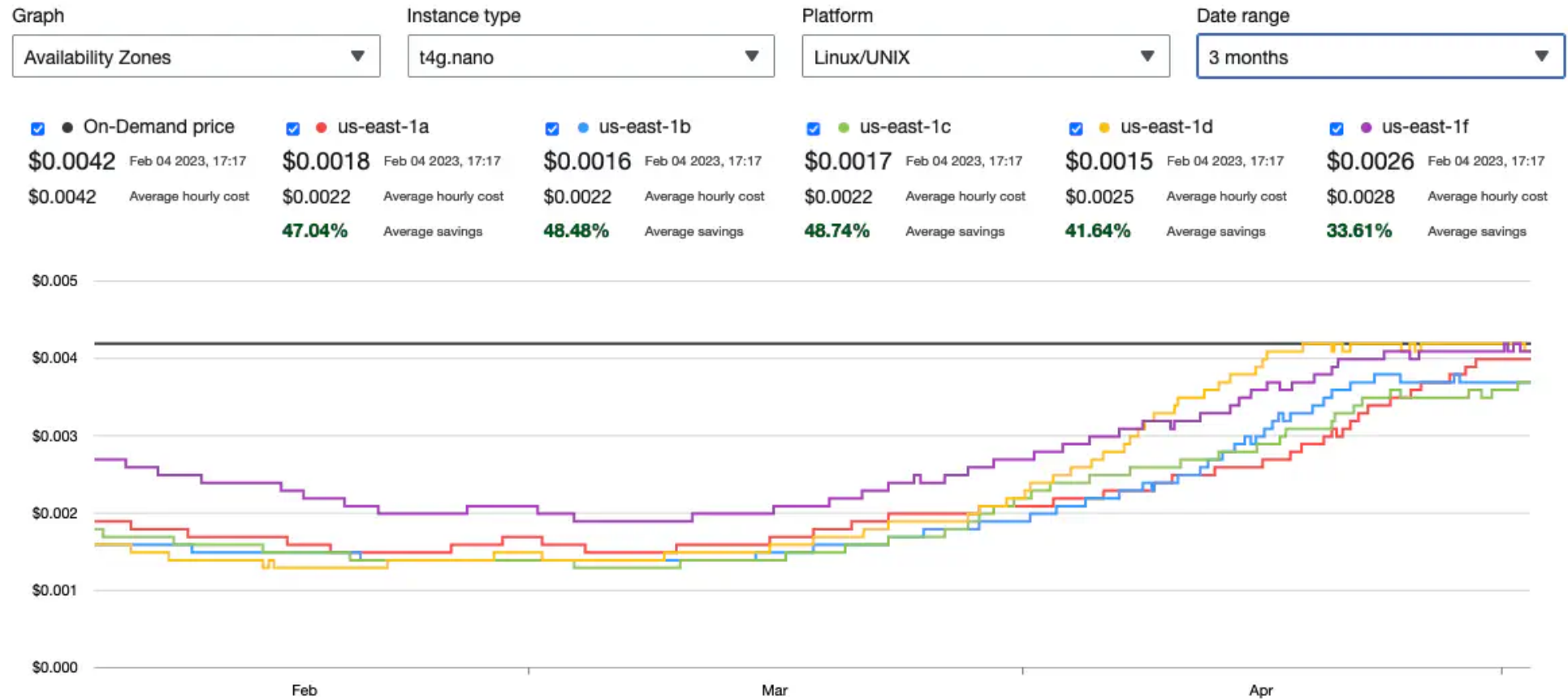


# Optimizing Costs, IaaS, Reserved Instances

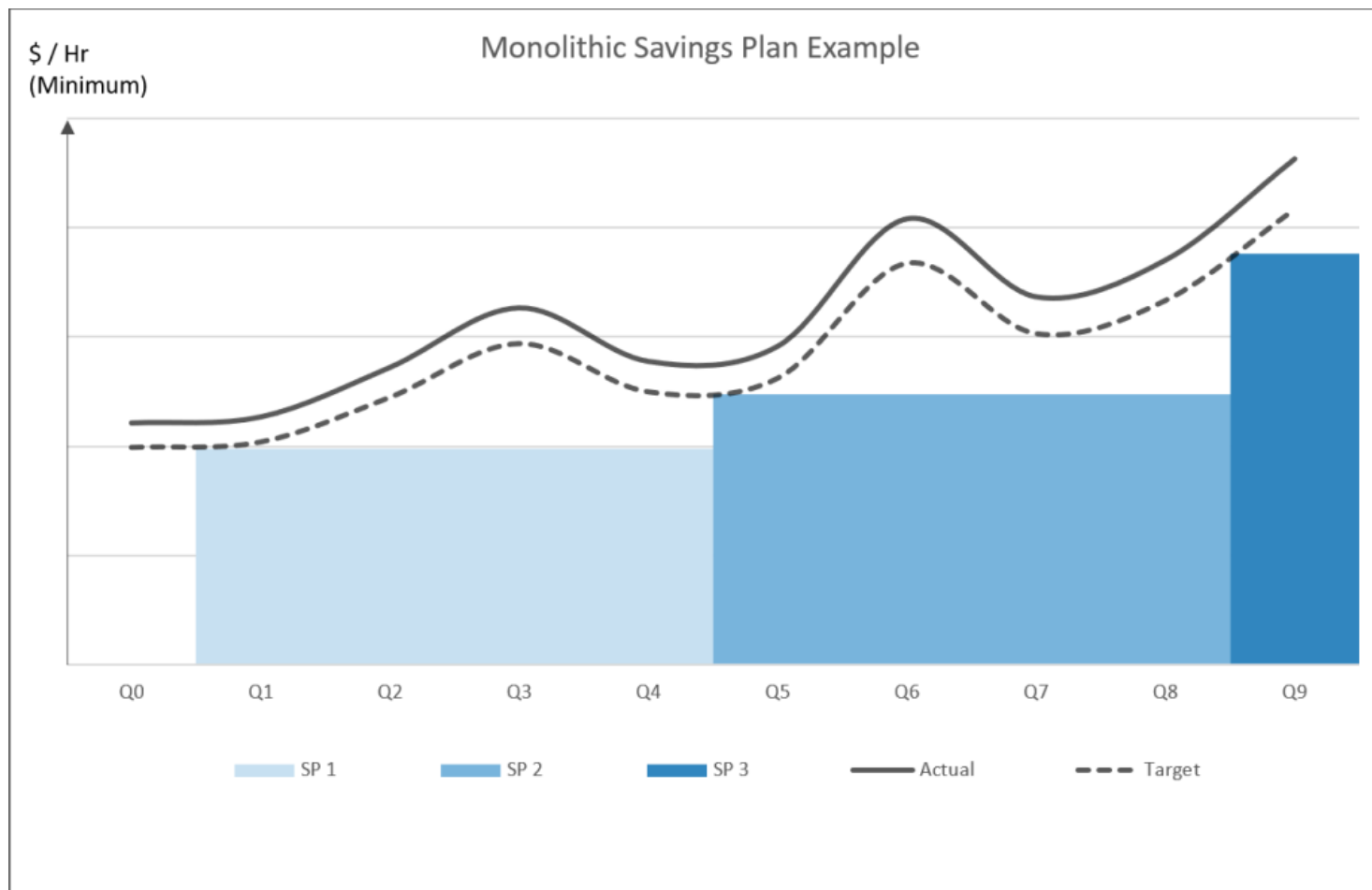
- Blue: On Demand
- Yellow: Upfront reserved Instance



# Optimizing Costs, IaaS, Spot Instances

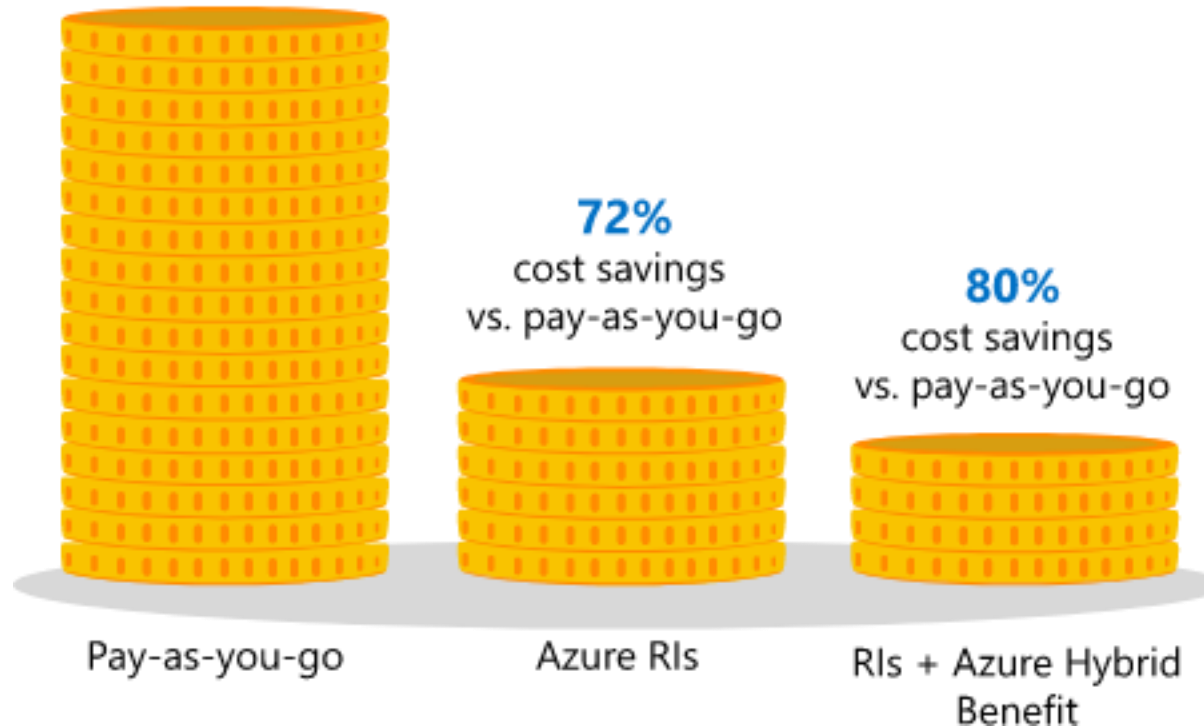


# Optimizing Costs, Computing in common, Saving Plans

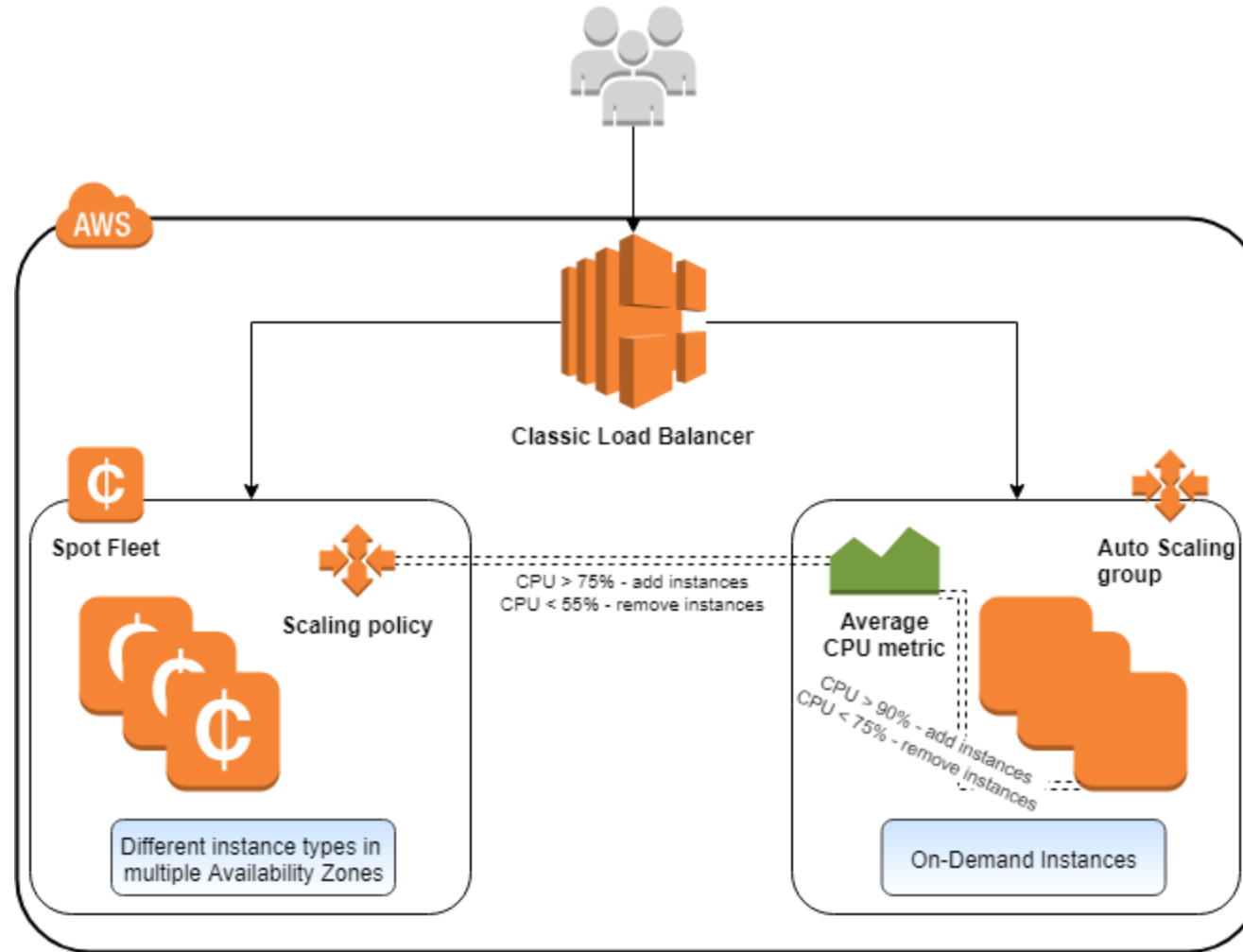


# Azure goes even further...

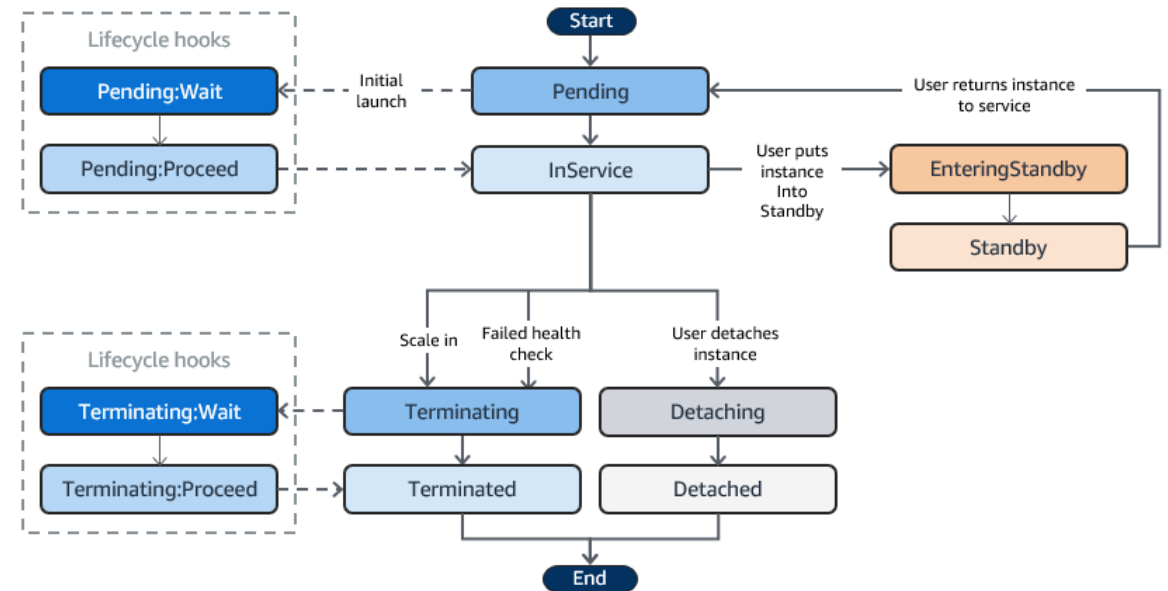
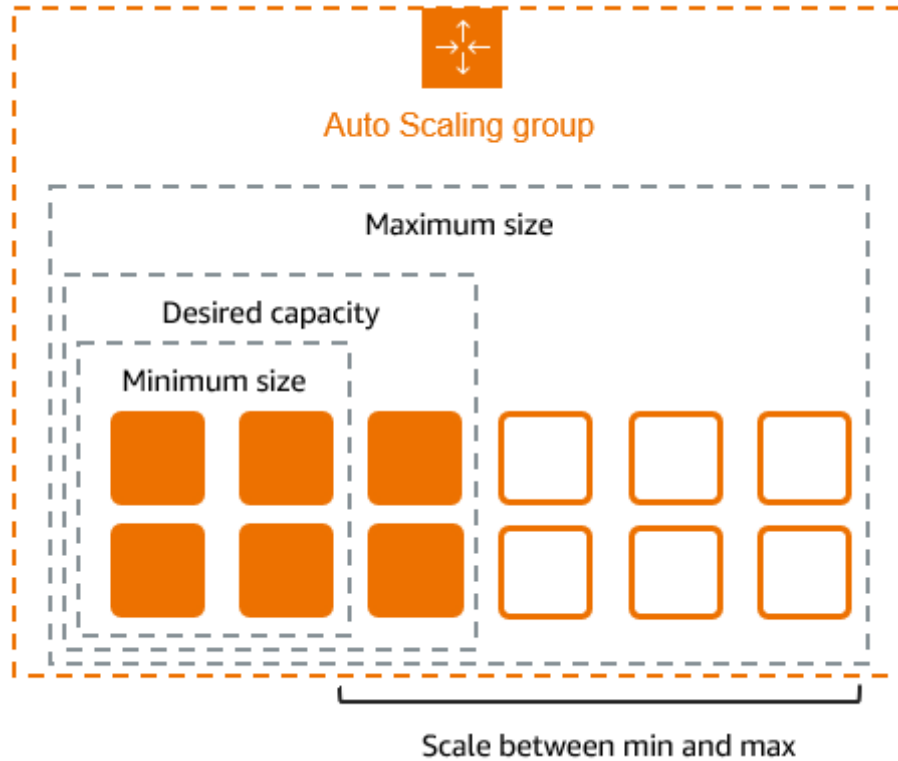
Save up to **80%** with RIs and Azure Hybrid Benefit



# Architectures for optimizing costs – IaaS



# Architecting for scale

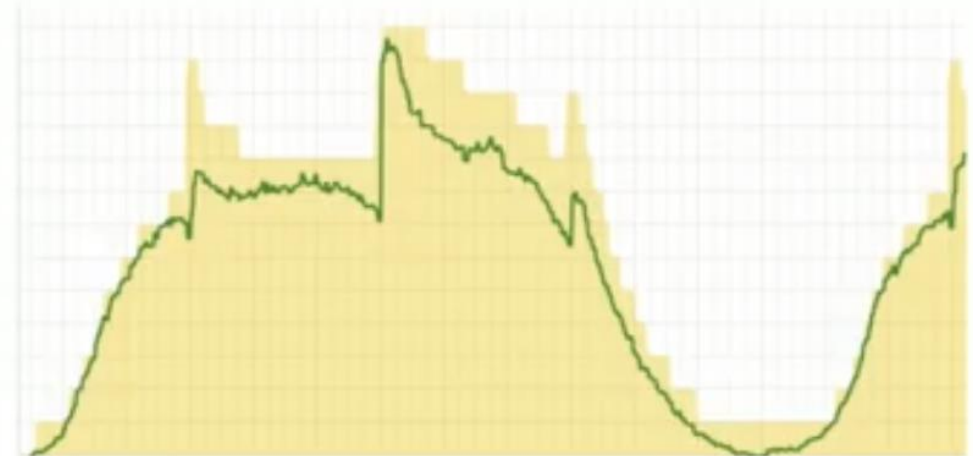




# Implementation on Switch Engines

## What is missing to implement that?

# Case Study - Supercell



<https://aws.amazon.com/solutions/case-studies/innovators/supercell/>

# Summary

- IaaS still main citizen in the cloud
  - Simple product
  - Well known to customer
  - Cheap / reliable / scalable (when used correctly)
- Billing drives Architecture
- IaC for setup / maintenance
  - Stay tuned for next week



# Optional Tutorials

- EC2:  
<https://awsacademy.instructure.com/courses/137586/modules> → Modul 6
- Azure:  
<https://learn.microsoft.com/de-de/training/modules/describe-azure-compute-networking-services/>

