Azure Compute
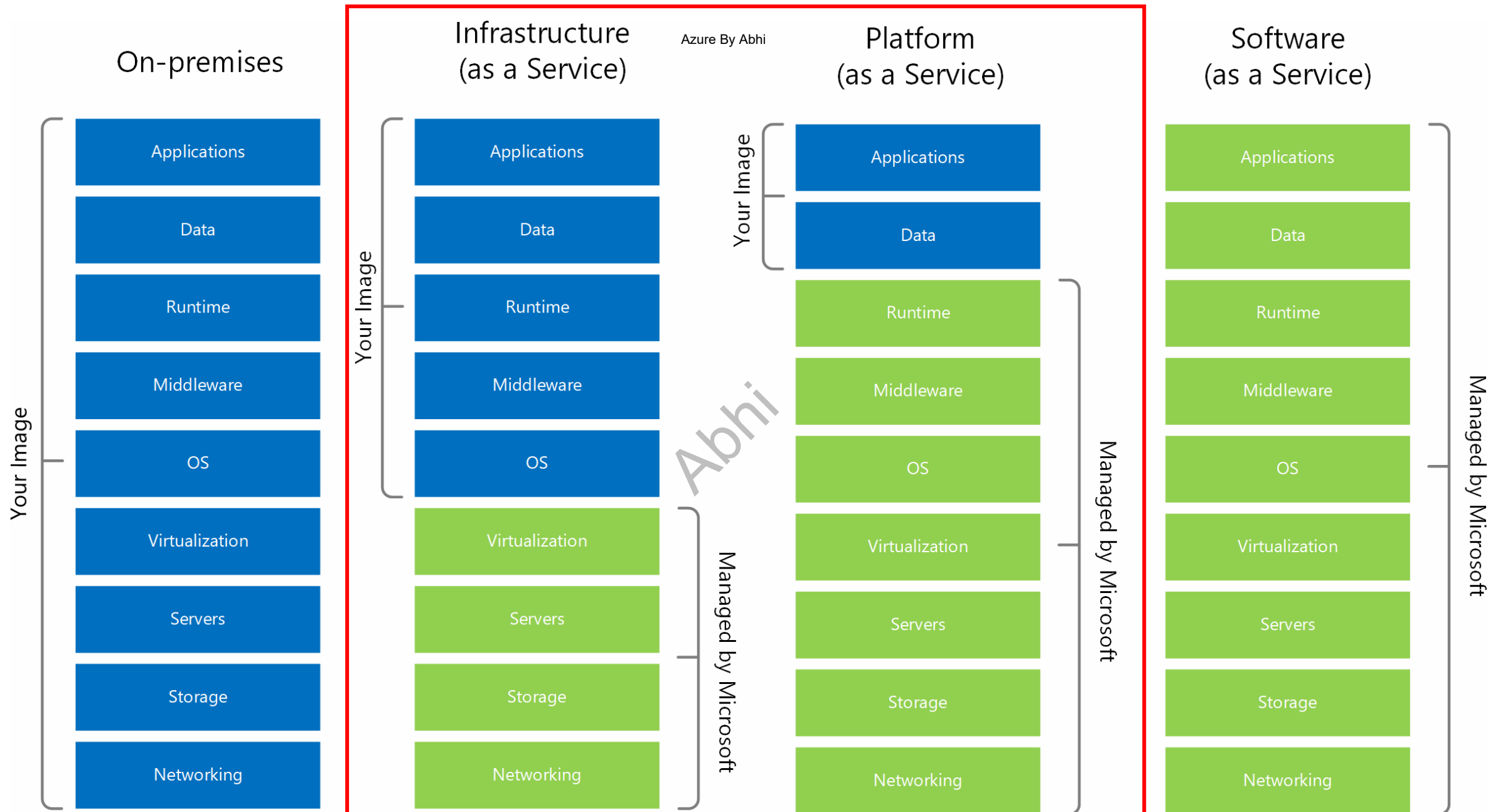
Abhijeet Kumar

Compute

- Set of cloud services for hosting and running applications

- Allows uploading your code and then running it

- Offers various levels of control and flexibility

On-premises

Applications
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking

Your Image

Infrastructure (as a Service)

Azure By Abhi

Applications
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking

Your Image

Managed by Microsoft

Platform (as a Service)

Applications
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking

Your Image

Managed by Microsoft

Software (as a Service)

Applications
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking

Managed by Microsoft

Abhi

Source: https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/strategy/monitoring-strategy

Azure By Abhi

Compute

- We'll talk about 4 types of Compute services:

Virtual Machines

App Services

AKS

Azure Functions

Virtual Machines

- A virtual (=not real) server running on a physical (=real) server

- Allows creating new servers extremely quick

- Based on existing resources of the physical server

- From the user's point of view – a regular server, nothing new
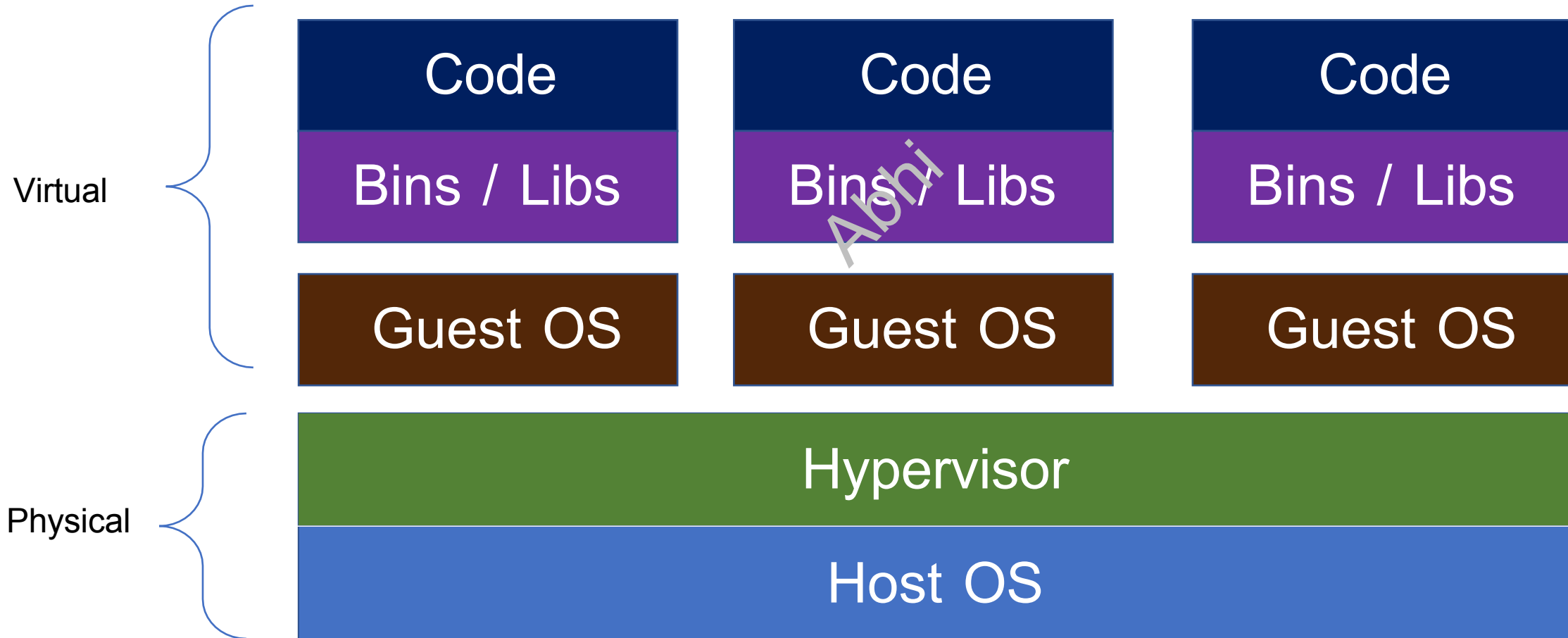
- Called an **Unmanaged Service**

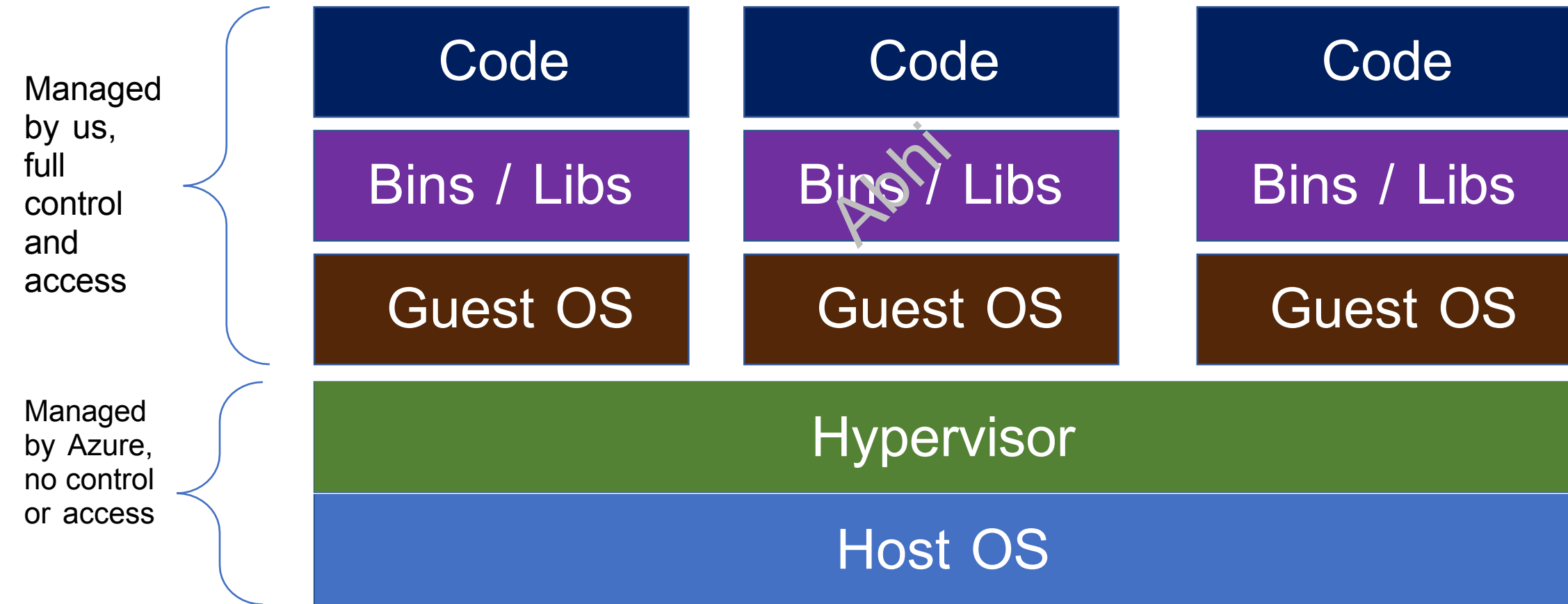# Virtual Machines Architecture

VM Density = No. of VMs per Host

In this case - 3

Virtual

Code	Code	Code
Bins / Libs	Bins / Libs	Bins / Libs
Guest OS	Guest OS	Guest OS

Physical

Hypervisor

Host OS

Virtual Machines In Azure

Managed by us, full control and access

Code	Code	Code
Bins / Libs	Bins / Libs	Bins / Libs
Guest OS	Guest OS	Guest OS

Managed by Azure, no control or access

Hypervisor

Host OS

Virtual Machines in Azure

- Steps for creating VM in Azure:

  - Selec
          t the location
  - Selec
          t the image (OS + Pre-Installed
  - Select the size

  - That's it, basically….

The Real Cost of VM

- Cost of VM includes:

  - VM

  - Disk

  - IP

  - Storage

The Real Cost of VM

Resource	Type	Monthly Cost ($)
VM	D2v3	154.76
Disk	P10	21.68
Public IP	Dynamic	2.92
Storage	LRS	<1$

Total: ~180$

Reducing the Cost of VM

- Most effective techniques to reduce costs of VM:

Auto Shutdown

Reserved Instances

Spot Instances

Disk Optimization

Auto Shutdown

- As simple as it sounds…

- Automatically shuts down the machine when not needed

  - Relevant mainly for test / dev machines

- Storage and IP (if static) costs still incurred

- Can save >50% of VM cost

Auto Shutdown

Create a virtual machine

Enable with custom storage account

○ Disable

OS guest diagnostics ⓘ   ○ On ● Off

Identity

System assigned managed identity ⓘ   ○ On ● Off

Auto-shutdown

Enable auto-shutdown ⓘ   ○ On ● Off

Canonical support plan

Add Ubuntu Advantage support plan

Backup

Enable backup ⓘ   ○ On ● Off

Review + create   < Previous   Next : Advanced >

Reserved Instances

- Allow upfront payment with substantial discount

- Usually offered for 1 or 3 years

  - Great for production machine which run continuously

- Offers great discounts (up to 62%)

- Can be divided to monthly payments

- Cannot be stopped / refunded

  - Unless…

Reserved Instances

Virtual Machines

REGION:
West Europe

OPERATING SYSTEM:
Linux

TYPE:
Ubuntu

TIER:
Standard

INSTANCE:
D4s v3: 4 vCPUs, 16 GB RAM, 32 GB Temporary storage, $0.24/hour

VIRTUAL MACHINES
✕ 730 Hours

Savings Options

Save up to 72% on pay-as-you-go prices with 1-year or 3-year Reserved Virtual Machine Instances. Reserved Instances are great for applications with steady-state usage and applications that require reserved capacity. Learn more about Reserved VM Instances pricing.

Compute (D4s v3)

⦿ Pay as you go
○ 1 year reserved (~36% discount)
○ 3 year reserved (~56% discount)

$175.20
Average per month
($0.00 charged upfront)

= $175.20
Average per month
($0.00 charged upfront)

Spot Instances

- Machines that run on unused capacity in Azure

- Can be evicted any moment when needed by Azure

- Offers up to 90% discount, price fluctuates according to demand

- Great for non-critical, non-continuous tasks

    - ie. Batch processes, long running calculations

Spot Instances

Azure By Abhi

Create a virtual machine

⚠️ Changing Basic options may reset selections you have made. Review all options prior to creating the virtual machine.

Availability options ⓘ	No infrastructure redundancy required ⌄
Image * ⓘ	Ubuntu Server 18.04 LTS - Gen1 ⌄

Browse all public and private images

Azure Spot instance ⓘ ◉ Yes ○ No

Eviction type ⓘ ◉ Capacity only: evict virtual machine when Azure needs the capacity for pay as you go workloads. Your max price is set to the pay as you go rate.

○ Price or capacity: choose a max price and Azure will evict your virtual machine when the cost of the instance is greater than your max price or when Azure needs the capacity for pay as you go workloads.

Eviction policy ⓘ ◉ Stop / Deallocate ○ Delete

Size * ⓘ	Standard_D2s_v3 - 2 vcpus, 8 GiB memory ($0.02005/hour) ⌄

Select size
View pricing history and compare prices in nearby regions

0.006

| Review + create | < Previous | Next : Disks > | Azure By Abhi |

Disk Optimization

- Make sure to select the right disk for the machine

- Default is Premium SSD – the most expensive option

- Non IO-intensive machines can do with Standard SSD

  - ie. App servers, in-memory cache

- Note: Disk type affects the SLA

Disk Optimization

Create a virtual machine

Basics   **Disks**   Networking   Management   Advanced   Tags   Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. Learn more

Disk options

OS disk type * (i)   Premium SSD   ∨

More Cost Saving Techniques

- Select the right size for your machine

  - CPU shouldn't rest, you pay for it☺

- Select Linux over Windows when possible

- Check price in nearby regions

Availability of a VM

SLA (%)	Yearly Downtime Allowed
95	18d 6h 17m 27s
99.5	1d 19h 49m 44s
99.9	8h 45m 56s
99.95	4h 22m 44s
99.99	52m 35s

SLA for Virtual Machines

Last updated: July 2020

- For all Virtual Machines that have two or more instances deployed across two or more Availability Zones in the same Azure region, we guarantee you will have Virtual Machine Connectivity to at least one instance at least 99.99% of the time.

- For all Virtual Machines that have two or more instances deployed in the same Availability Set or in the same Dedicated Host Group, we guarantee you will have Virtual Machine Connectivity to at least one instance at least 99.95% of the time.

- For any Single Instance Virtual Machine using Premium SSD or Ultra Disk for all Operating System Disks and Data Disks, we guarantee you will have Virtual Machine Connectivity of at least 99.9%.

- For any Single Instance Virtual Machine using Standard SSD Managed Disks for Operating System Disk and Data Disks, we guarantee you will have Virtual Machine Connectivity of at least 99.5%.

- For any Single Instance Virtual Machine using Standard HDD Managed Disks for Operating System Disks and Data Disks, we guarantee you will have Virtual Machine Connectivity of at least 95%.

Source: https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/v1_9/

Availability Concepts in Azure

Fault Domain

Update Domain

Availability Set

Availability Zone

Fault Domain

- Logical group of physical hardware that share a common

  power source and network switch

- Similar to rack in a traditional data center

Fault Domain

If there's a problem with the power or networking

in the domain (=rack) – all servers in it shut down

You want to make sure your servers are spread

across more than one fault domain (=rack)

Update Domain

- Logical group of physical hardware that can undergo

  maintenance and be rebooted at the same time

- Maintenance is done by Azure at its own discretion

Update Domain

If all your servers are in the same update domain –

they'll reboot at the same time during maintenance

You want to make sure your servers are spread

across more than one update domain

Availability Set

- A collection of Fault Domains and Update Domains your VMs will be spread across

- Can contain up to 3 Fault Domains and up to 20 Update Domains

- All domains (Fault & Update) are in the same Zone (=datacenter)

Availability Set Example

Availability Set Example
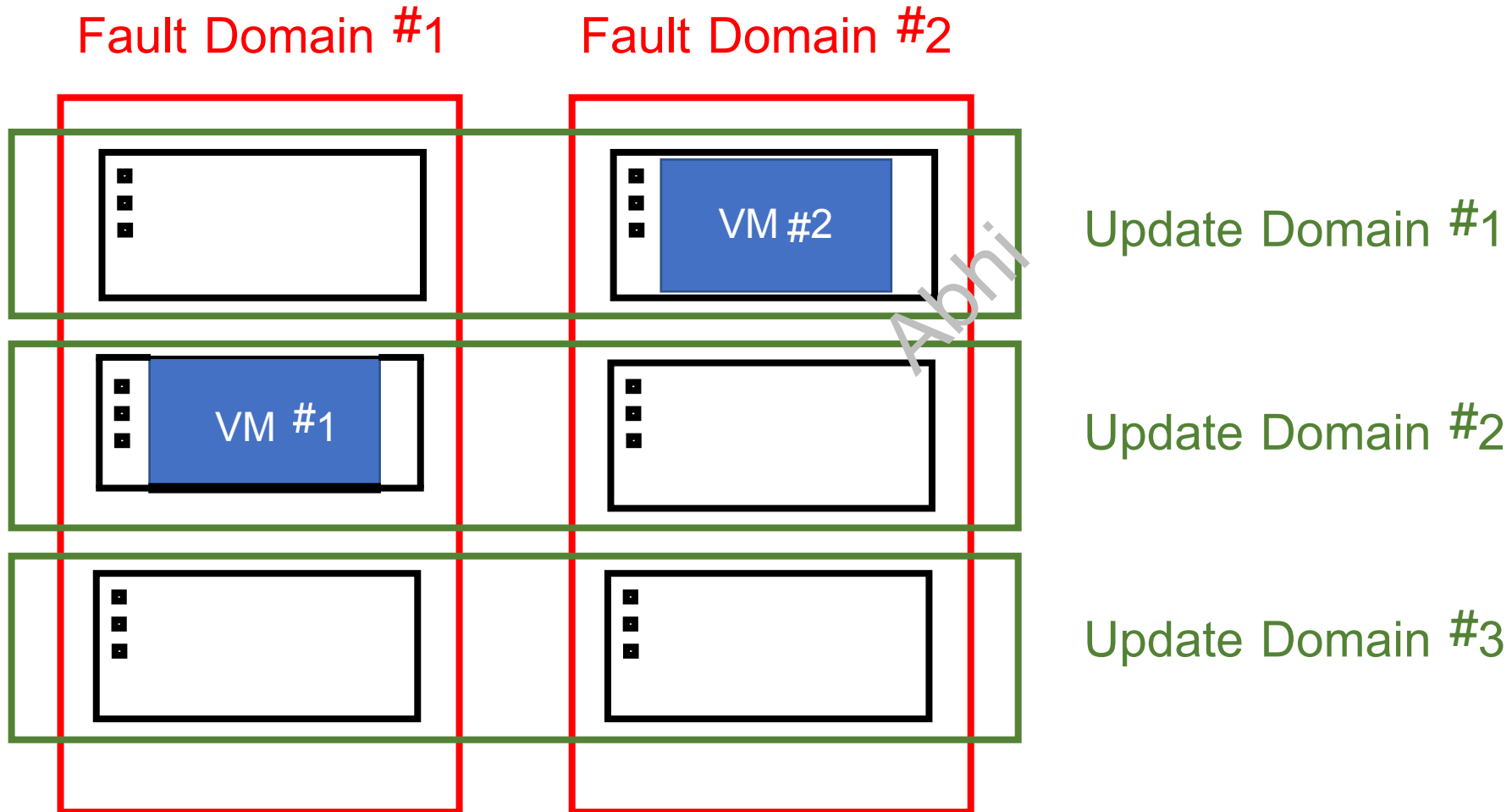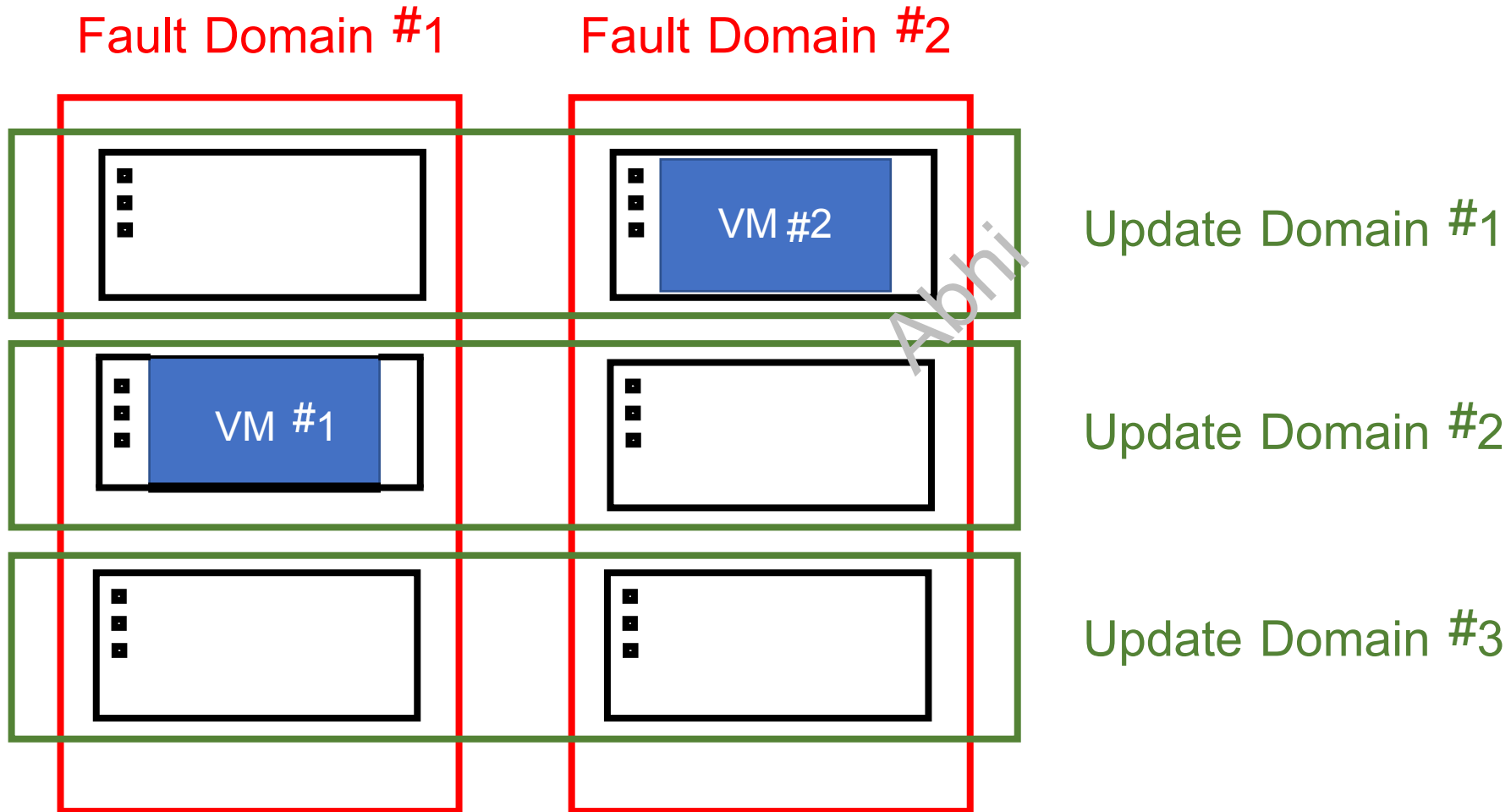
Azure By Abhi

Fault Domain #1   Fault Domain #2

VM #2

Update Domain #1

VM #1

Update Domain #2

Update Domain #3
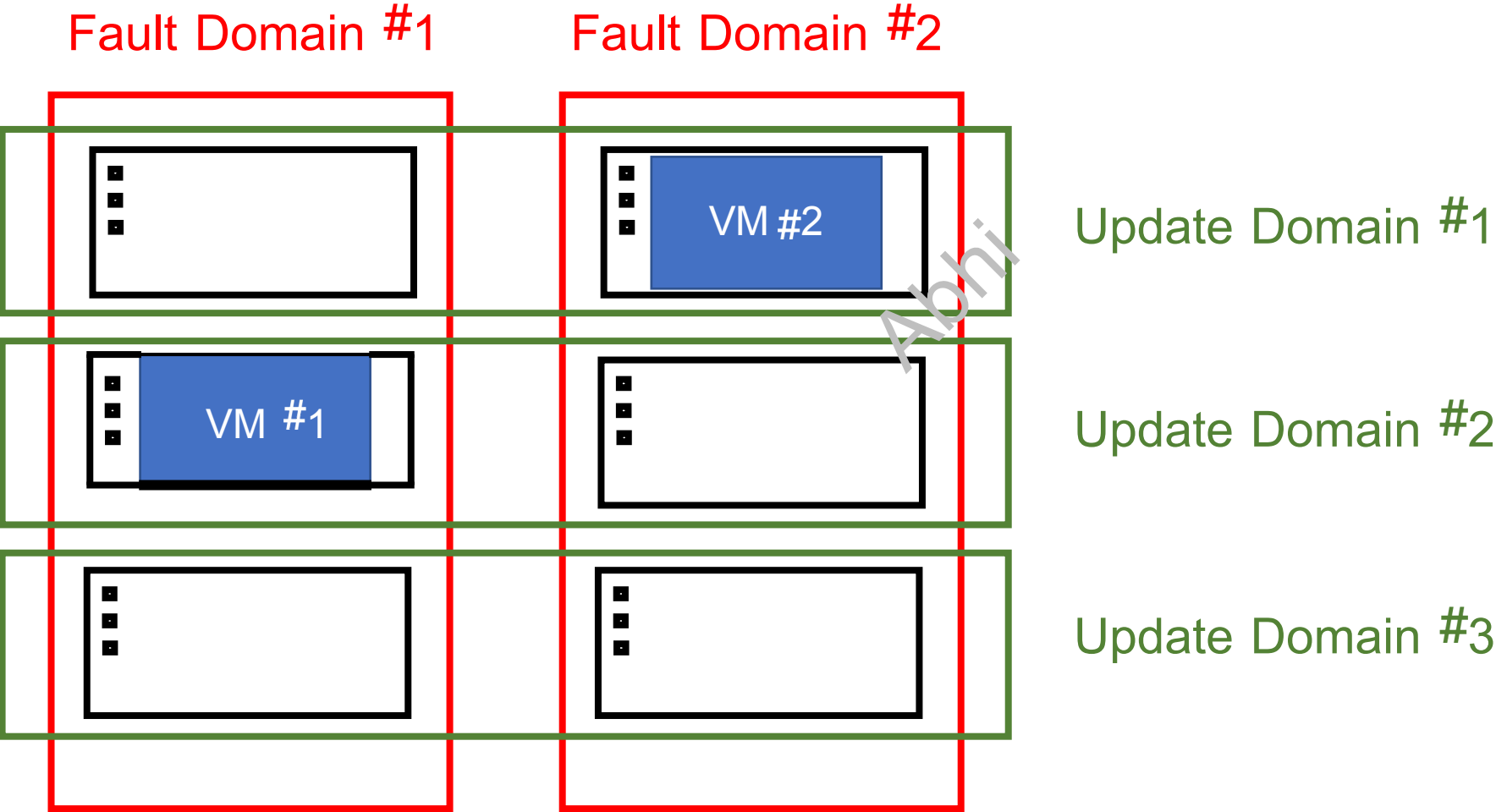
Azure By Abhi

Without Availability Set...
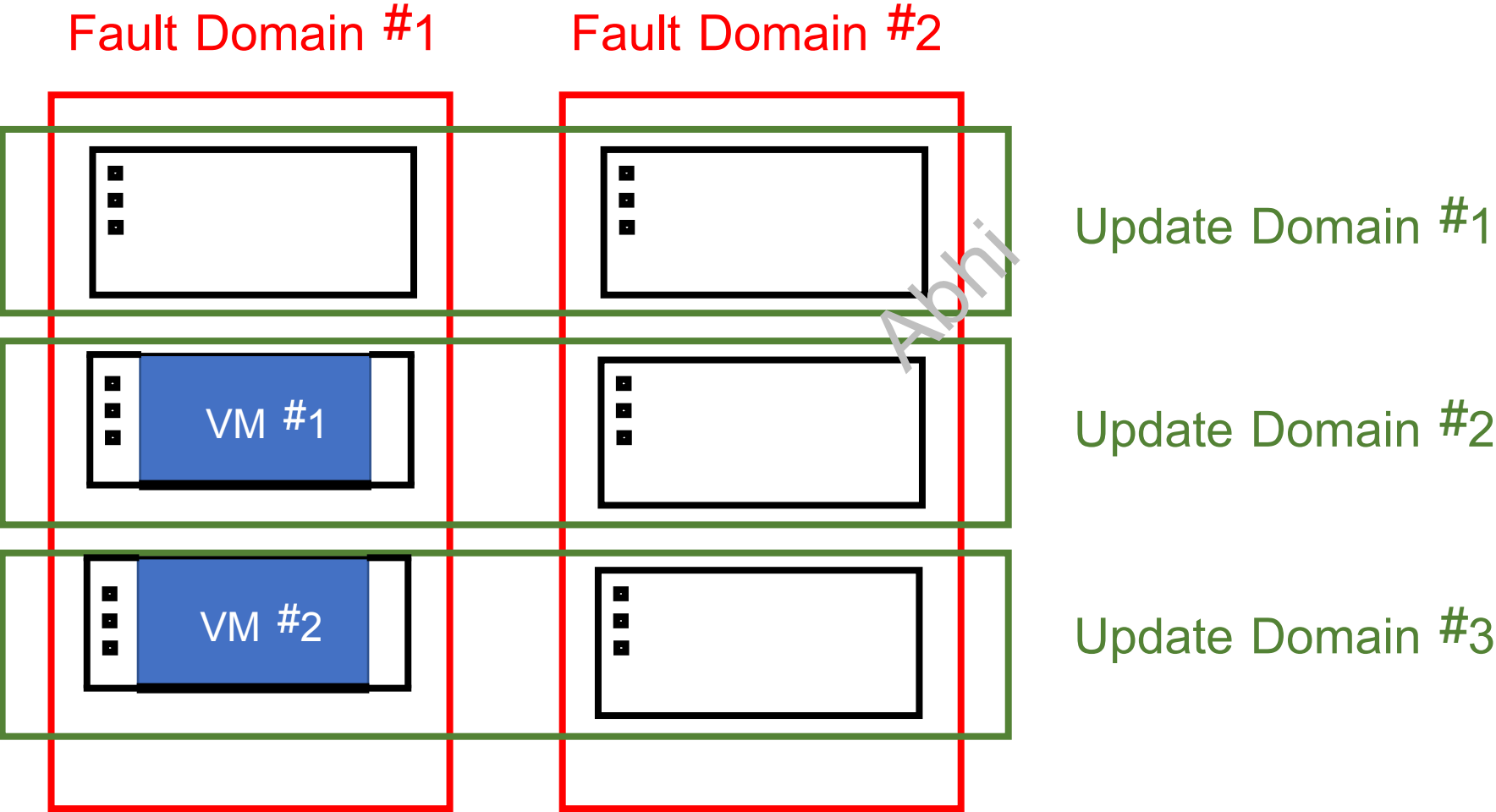
Taking Advantage of Availability Set

- Deploy identical VMs into the same Availability Set

- Ensures they won't be shut down simultaneously when a single fault domain shuts down or an update domain reboots

- If needed – deploy load balancer to route between the VMs

- Availability Set is free, you pay only for the additional VMs

Availability Zone

- A physically separate zone within an Azure region

- Technically – a building containing an autonomous data center

- Each zone functions as a fault & update domain

- Provides protection against a complete zone shutdown

    - Hence the better SLA

Taking Advantage of Availability Zone

- Deploy identical VMs into separate Availability Zones in the same Region

- Ensures they won't be shut down simultaneously when the zone shuts down

- If needed – deploy load balancer to route between the VMs

- Availability Zone is free, you pay only for the additional VMs

ARM Template

- Azure Resource Manager Template

- A JSON file describing the resource(s) to be created

- Used by Azure in (almost) all deployments

- Can be exported, modified, uploaded, deployed

- Can also be created from scratch

ARM Template

- ARM Template is a declarative way of deploying resources

Declarative	Imperative

Declarative
- Describes the end result
- Allows "What-If" operation
- Can deploy multiple resources at once
- Can be integrated in CI/CD processes
- Can be source controlled
-

Imperative
- Sends instructions to run
- Error prone
- Can't be verified
- Can't be source controlled
- Suited for quick and dirty operations

Used by: ARM Template

Used by: Azure CLI, PowerShell

(Although they can run ARM Template too)

Virtual Machine Scale Set

- A group of separate VMs sharing the same image

- Managed as a group

- Can be scaled out or in manually or according to predefined conditions

- Great for handling unpredictable load

Virtual Machine Scale Set

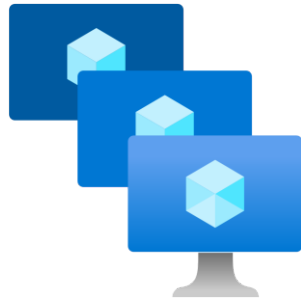- Once set up, the machines should NOT be modified

    - Change files, install apps etc.

- New machines created by the scale set will be based on the original image

- For web apps, a load balancer should be put in front of the scale set

Scale Set Architecture

Load Balancer

Virtual Machine Scale Set
(VMSS)

Scale Set Pricing

- Scale Set is free

- You pay for the VMs deployed in it

Azure Instance Metadata Services

- A little known feature of Azure VMs

- A REST API accessible from the VM

- Providing a lot of info about the machine

- Info includes:

    - SKU, storage, networking, scheduled events

- Accessible ONLY from the VM

Azure Instance Metadata Services

- With Scaleset –

    - Get notification about upcoming eviction

- Can be polled every ~1 min to get enough time to close things up

Azure Architecture Diagram

- When designing architecture for Azure apps it's a good idea to use Azure symbols in the diagram

- There are hundreds of them...

Download Azure Icons

- https://docs.microsoft.com/en-us/azure/architecture/icons/

Cloud Architecture

VM
Weather
API

VM
Catalog App

Cloud Architecture

A Word of Caution:

NEVER
leave a VM open to the internet this way

VM
Weather API

VM
Catalog App

Directly accessible from the internet

We will learn later on what should be done

-

- **Can be RDPed from anywhere**

App Services

- A fully managed web hosting for websites

- Publish your code – and it just runs

- No access to the underlying servers

- Secured and compliant

- Integrates with many source controls and DevOps engines:

  - GitHub, BitBucket, Azure DevOps, DockerHub and more

App Services

- Supported platforms:

  - .NET

  - .NET Core

  - Node.JS

  - Java

  - Python

  - PHP

- Supports containers

App Services

- App Types:

  - Web Apps

  - Web API

  - Web Jobs (batch processes)

App Services

- Extremely easy to deploy:

    1. Develop your app

    2. Create Web App (can be done from the IDE)

    3. Publish your code

    4. Viola!

App Services Tiers

OS/Software	Region:	Currency:	Display pricing by:
Linux	West Europe	US Dollar ($)	Month

	FREE Try for free	BASIC Dedicated environment for dev/test	STANDARD Run production workloads	PREMIUM Enhanced performance and scale	ISOLATED High-Performance, Security and Isolation
Web, mobile, or API apps	10	Unlimited	Unlimited	Unlimited	Unlimited
Disk space	1 GB	10 GB	50 GB	250 GB	1 TB
Maximum instances	–	Up to 3	Up to 10	Up to 30*	Up to 100
Custom domain	–	Supported	Supported	Supported	Supported
Auto Scale	–	–	Supported	Supported	Supported
Hybrid Connectivity	–	Supported	Supported	Supported	Supported
Virtual Network Connectivity	–	–	Supported	Supported	Supported
Private Endpoints	–	–	–	Supported	Supported
Compute Type	Shared	Dedicated	Dedicated	Dedicated	Isolated
Pay as you go price	Free	Promotional Price ~$13.140/month	~$69.35/month	~$83.95/month	~$208.050/month

Lowest price, goes up with CPU & RAM

Source: https://azure.microsoft.com/en-us/pricing/details/app-service/linux/

App Service Auto Scaling

- App Service can be autoscaled to support spikes in load

- Auto scale is based on various metrics

- Extremely flexible

App Service Auto Scaling

Deployment Slots

- When uploading code to App Service the new version is accessible immediately

- Sometimes we want to test the version before publishing it

- Deployment Slots allow us to upload code and test it separately from the main site

- After validation we swap the slots and promote it to production

Deployment Slots

- New slots are created from the portal

- Number of allowed slots depends on plan

  - ie. Standard plans allow for up to 5 slots

- Slot is a fully functional app service with a dedicated URL

- Slots can be accessed directly through their URL

- Free

Deployment Slots Traffic Splitting

- Traffic can be split between slots

- Some users will be routed to the production and some to the new slot

- Great for some deployment scenarios

Deployment Types

- Traffic Splitting enables various types of deployment

Basic

Rolling

Blue-Green

Basic Deployment

- All instances update to the new version at once

■ v1

■ v
  2

Basic Deployment

- All instances update to the new version at once

■ v1

■ v

2

Basic Deployment

Pros

- Simple
- Fast
-

Cons

Risky

- System might get unusable

Basic Deployment

- App Service implementation:

  - Simple deployment of new version

  - No use of deployment slots

Rolling Deployment

- Instances are updated gradually in batches

- Only if no errors are found the deployment resumes

v1

v

2

Rolling Deployment

- Instances are updated gradually in batches

- Only if no errors are found the deployment resumes

v1

v

2

Rolling Deployment

- Instances are updated gradually in batches

- Only if no errors are found the deployment resumes

v1

v

2

Abhi

Rolling Deployment

- Instances are updated gradually in batches

- Only if no errors are found the deployment resumes

■ v1

■ v

  2

Rolling Deployment

Pros
• Allows rollback

Cons
• Need to support two versions simultaneously
• Not easy to manage

Rolling Deployment

- App Service implementation:

  - Deploy to slot

  - Set traffic percentage so that small % will be routed to the new version

  - Gradually increase % of traffic to new version until 100%

  - Swap slots

Blue-Green Deployment

- New version uploaded and accessible only to testers

- After verification complete, traffic is routed to new version

v1

v
2

Blue-Green Deployment

- New version uploaded and accessible only to testers

- After verification complete, traffic is routed to new version

v1

v
2

Blue-Green Deployment

- New version uploaded and accessible only to testers

- After verification complete, traffic is routed to new version

v1

v

2

Blue-Green Deployment

Pros

- Simple
- New environment is always tested
-

Cons

Cost

- More instances

Blue-Green Deployment

- App Service implementation:

  - Deploy to a new slot

  - Testers work on the new slot using its dedicated URL

  - After testing is complete – swap slots

Cloud Architecture
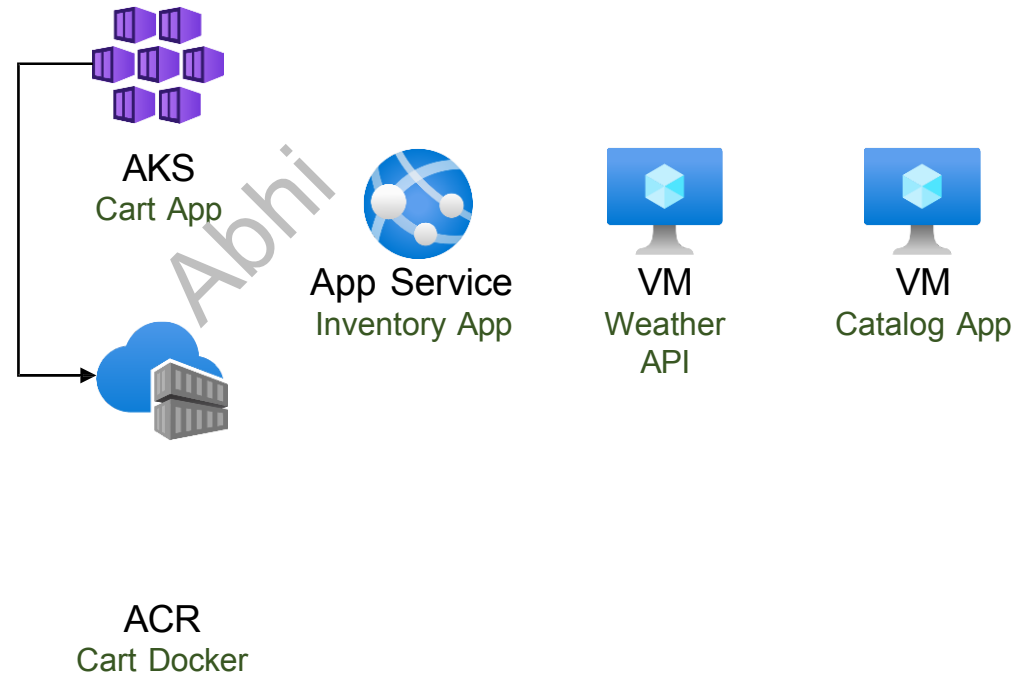
Abhi

App Service

Inventory App

VM

Weather API

VM

Catalog App

AKS

- Azure Kubernetes Services

- Managed Kubernetes on Azure

- Allows deploying containers and managing them using Kubernetes on Azure

- Paying only on the instances (=VMs) used

Cloud Architecture

AKS
Cart App

App Service
Inventory App

VM
Weather API

VM
Catalog App

ACR
Cart Docker

Azure Functions

- Small, focused functions running as a result of an event

- Great for Event Driven systems

- Automatically managed by Azure

  - Start, stop, autoscale

- Flexible pricing plans

- Serverless

Serverless

- Cloud resource that is completely managed by the cloud

- Users do not need to think about:

  - VMs

  - CPU

  - Memory

  - etc.

- It just works

Serverless

Remember:

Azure Function Example

```
namespace AzureCourse.Function
{
    0 references
    public static class EventGridFunction
    {
        [FunctionName("EventGridFunction")]
        0 references
        public static async Task Run(
            [HttpTrigger(AuthorizationLevel.Anonymous, "get", "post", Route = null)] HttpRequest req,
            [EventGrid(TopicEndpointUri = "MyEventGridTopicUriSetting", TopicKeySetting = "MyEventGridTopicKeySetting")]
                IAsyncCollector<EventGridEvent> outputEvents,
            ILogger log)
        {
            string name = req.Query["name"];

            var myEvent = new EventGridEvent("message-id-3", "user-added", $"{{name: {name} }}", "event-type", DateTime.UtcNow, "1.0");
            await outputEvents.AddAsync(myEvent);

        }
    }
}
```

Triggers and Bindings

Triggers

- The event that made the function run

- Quite a few

- Deeply integrated into other Azure services

- Technically not mandatory, but…

-

Bindings

Declarative connection to other resource(s)

- Input, output, or both

- Provided as parameter to the function

- Makes connecting to other resources extremely easy

- Not mandatory

Azure Function Example


```
namespace AzureCourse.Function
{
    0 references
    public static class EventGridFunction
    {
                             ventGridFunction")]

        public static async Task Run(
            [HttpTrigger(AuthorizationLevel.Anonymous, "get", "post", Route = null)] HttpRequest req,
            [EventGrid(TopicEndpointUri = "MyEventGridTopicUriSetting", TopicKeySetting = "MyEventGridTopicKeySetting")]
                IAsyncCollector<EventGridEvent> outputEvents,
            ILogger log)

                                  Query["name"];

            var myEvent = new EventGridEvent("message-id-3", "user-added", $"{{name: {name} }}", "event-type", DateTime.UtcNow, "1.0");
            await outputEvents.AddAsync(myEvent);

        }
    }
}
```

Trigger (HTTP)

Binding (EventGrid)

Trigger Types

- Blob Storage

- Cosmos DB

- Dapr

- Event Grid

- Event Hubs

- HTTP Requests

- IOT Hub

- Kafka

-

Queue
Storage

- RabbitMQ

- Service Bus

- Timer

Binding Types (Input or Output)

- Blob Storage

- Cosmos DB

- Dapr

- Event Grid

- Event Hubs

- HTTP Requests

- IOT Hub

- Kafka
-

Mobile Apps

- Notification Hub

- Queue Storage

- RabbitMQ

- SendGrid

- Service Bus

- SignalR

- Table Storage

Triggers and Bindings

- Example scenarios:

  - Run every 5 minutes (Timer Trigger) and calculate the sum of a column in a DB. If it's above 115, send an event in EventGrid (Binding)

Triggers and Bindings

- Example scenarios:

  - When a message arrives in the Orders Queue (Queue Trigger)

    save it in Cosmos DB (Binding) for future handling

Triggers and Bindings

- Example scenarios:

  - Receive HTTP Request (HTTP Trigger) with 4 numbers, and return the smallest one of them (no binding)

Supported Languages

- C

  #
- JavaScript (nodeJS)

- Java

- Python

- PowerShell

- F
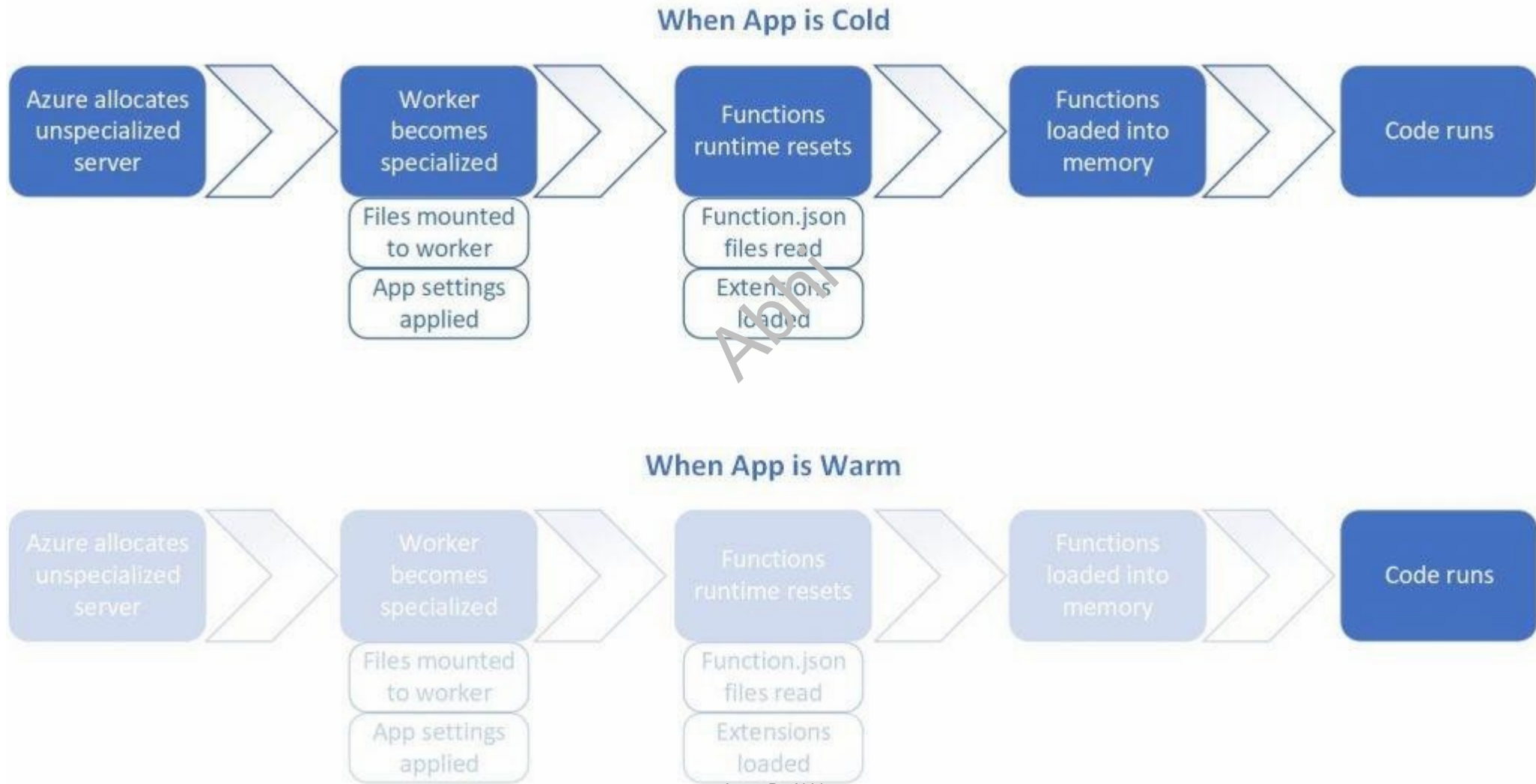
  #

Cold Start

- Azure Functions are completely managed by Azure

- After some time of inactivity Azure might take down the Function's host

- The next activation of the Function will take time

  - 2-3 seconds before the code runs

- A problem mainly for HTTP-Triggered functions

Cold Start

Source: https://azure.microsoft.com/en-us/blog/understanding-serverless-cold-start/

Cold Start

- How to avoid cold start?

  - Select the right hosting plan

Azure Functions Hosting Plans

Consumption

Premium

Dedicated

Consumption Plan

- Pay only for what you actually use

METER	PRICE	FREE GRANT (PER MONTH)
Execution Time*	$0.000016/GB-s	400,000 GB-s
Total Executions*	$0.20 per million executions	1 million executions

*Free grants apply to paid, consumption subscriptions only.

Note—A storage account is created by default with each Functions app. The storage account is not included in the free grant. Standard storage rates and networking rates charged separately as applicable.

- Note: In consumption plan there's a limit of 1.5GB RAM

Consumption Plan

METER	PRICE	FREE GRANT (PER MONTH)
Execution Time*	$0.000016/GB-s	400,000 GB-s
Total Executions*	$0.20 per million executions	1 million executions

- Calculation example:

- Executions / month: 9m

- Avg. memory consumed / execution: 800MB

- Avg. execution duration: 1.5s

- Total seconds: 9m * 1.5s = 13.5m secs

- Total GB / sec = 13.5m * 0.8 = 10.8m – 400K free grant = 10.4m GB/sec

- Payment for execution time: 10.4m * 0.000016$ = 166.4$

- Payment for executions: 9m-1m free grant = 8m * 0.2$ / m = 1.6$

Total Payment: 168$

Consumption Plan

- Downsides:

  - 1.5GB RAM limit

  - Cold Start

Premium Plan

- Pay for pre-warmed instances (hosts)

METER	PRICE
vCPU duration	vCPU: ~$123.37 vCPU/month
Memory duration	Memory: ~$8.833 GB/month

INSTANCE:

EP1: 1 Cores(s), 3.5 GB RAM, 250 GB Storage	⌄

EP1: 1 Cores(s), 3.5 GB RAM, 250 GB Storage

EP2: 2 Cores(s), 7 GB RAM, 250 GB Storage

EP3: 4 Cores(s), 14 GB RAM, 250 GB Storage

Instances

- Pay for scale-out instances

Premium Plan

- What you get:

  - No cold starts

  - No memory limit (up to host RAM)

  - Better performance

  - VNet integration

  - Predictable price

Premium Plan

METER	PRICE
vCPU duration	vCPU: ~$123.37 vCPU/month
Memory duration	Memory: ~$8.833 GB/month

- Calculation example:

  - 1 pre-warmed instance

  - 2 vCpus, 7GB RAM

  - No scale out

- vCPU cost: 123.37 X 2 = 246.74$

- Memory cost: 8.833 X 7 = 61.83$

Total Payment: 308.57$

Premium Plan

- Downsides:

  - More expensive

Dedicated Plan

- The Functions run on an existing App Service

- Great if server is under-utilized

- No additional costs

Dedicated Plan

- Make sure Always On setting is activated to avoid

  disabling functions:
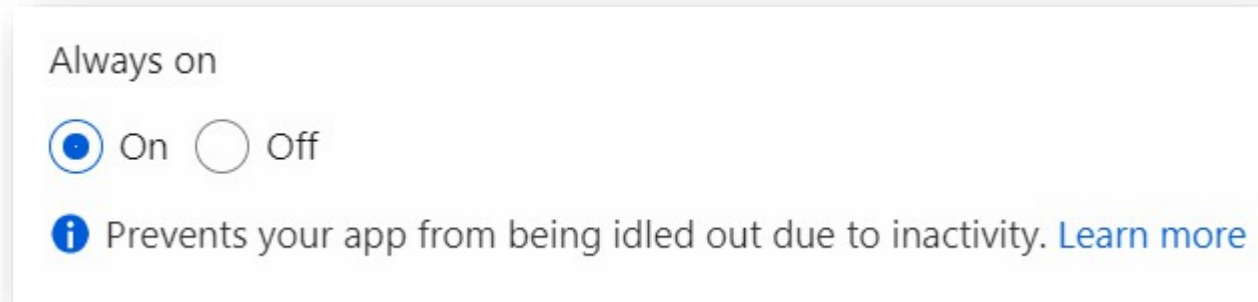
Dedicated Plan

- Downsides:

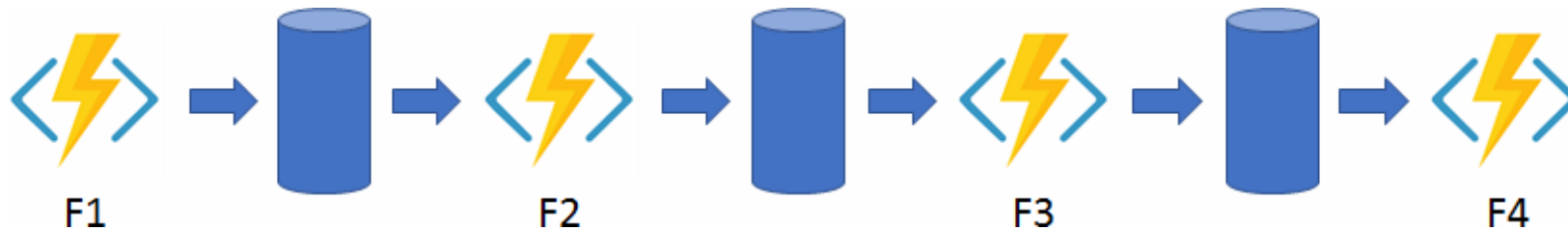 - No Auto-Scale

Durable Functions

- Stateful Functions that interact with external resources and keep track of flow

- Offer very simple syntax, hide complexities of managing state, retries, etc.

Durable Functions

- For example:

  - Function Chaining – call various Functions sequentially, and apply the output of each function to the next one:
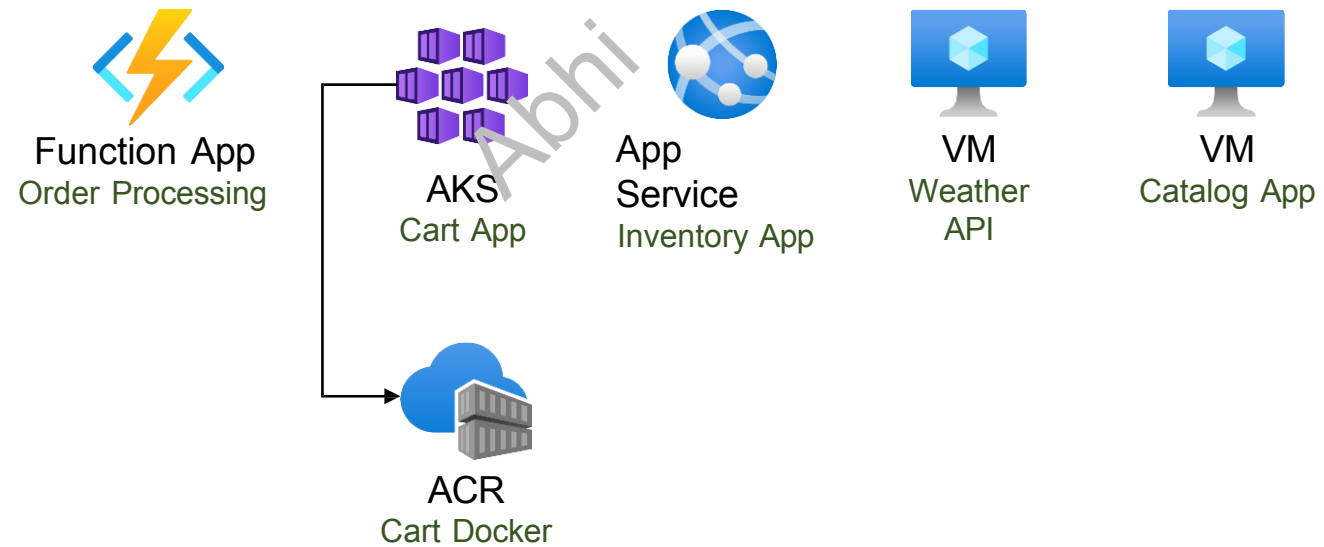
Durable Functions

```csharp
[FunctionName("Chaining")]
public static async Task<object> Run(
    [OrchestrationTrigger] IDurableOrchestrationContext context)
{
    try
    {
        var x = await context.CallActivityAsync<object>("F1", null);
        var y = await context.CallActivityAsync<object>("F2", x);
        var z = await context.CallActivityAsync<object>("F3", y);
        return  await context.CallActivityAsync<object>("F4", z);
    }
    catch (Exception)
    {
        // Error handling or compensation goes here.
    }
}
```
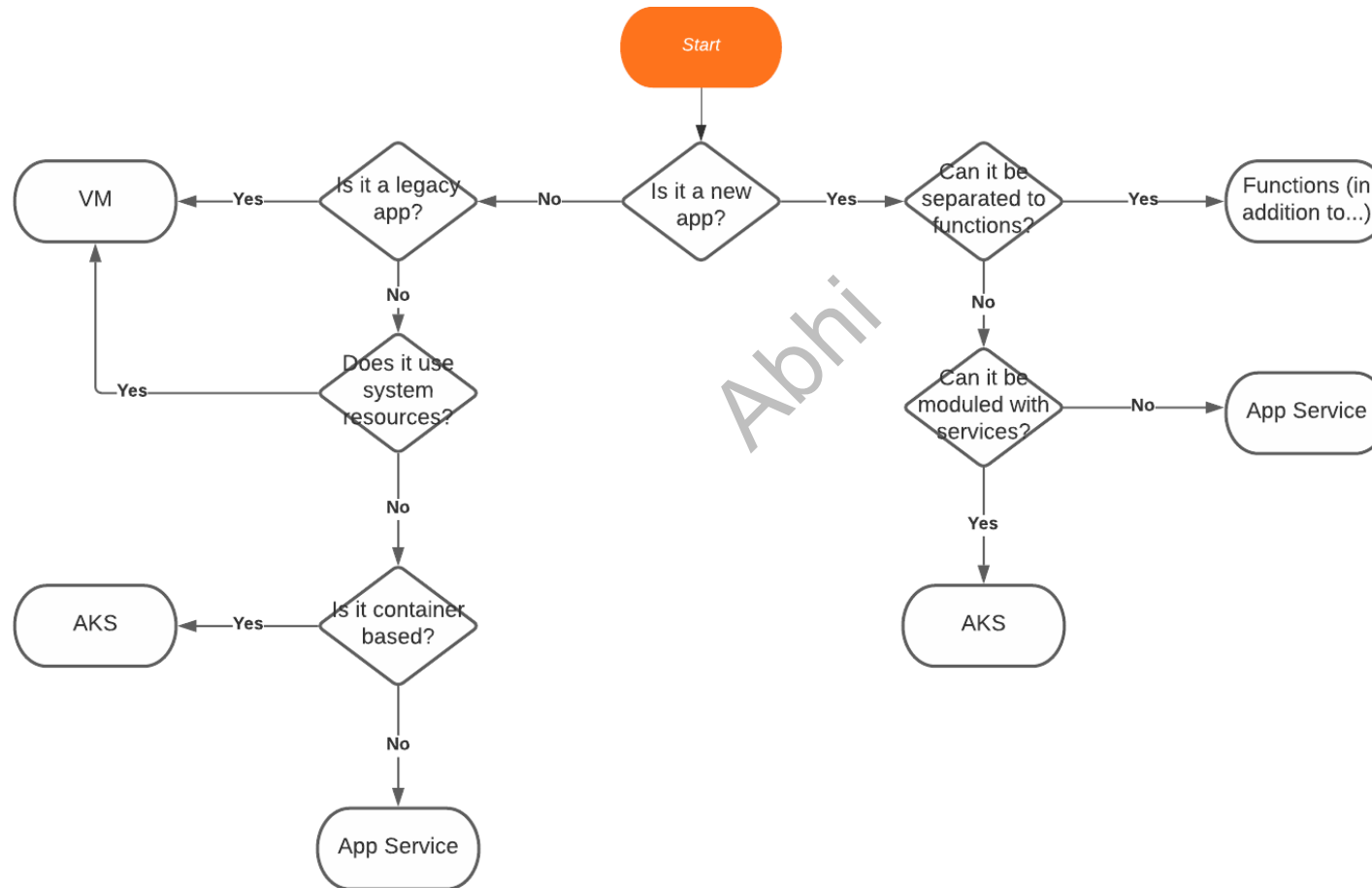
Cloud Architecture

Function App
Order Processing

AKS
Cart App

App Service
Inventory App

VM
Weather API

VM
Catalog App

ACR
Cart Docker

How to Choose Compute Type?

More Compute Options

- Logic Apps

- ACI – Azure Container Instance

- App Service Container – Deploy docker to App Service