Symmetric Multi-Cloud Architecture with HashiCorp Cloud Platform
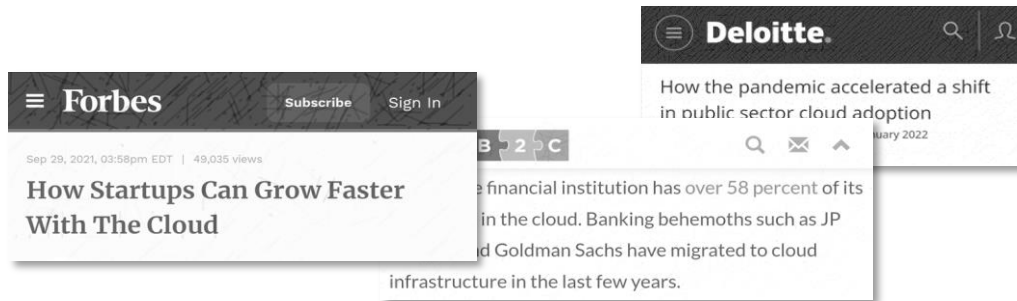
How Sunrise Banks Built a Symmetric Multi-Cloud Architecture with Terraform & Vault

HashiCorp   SUNRISE BANKS
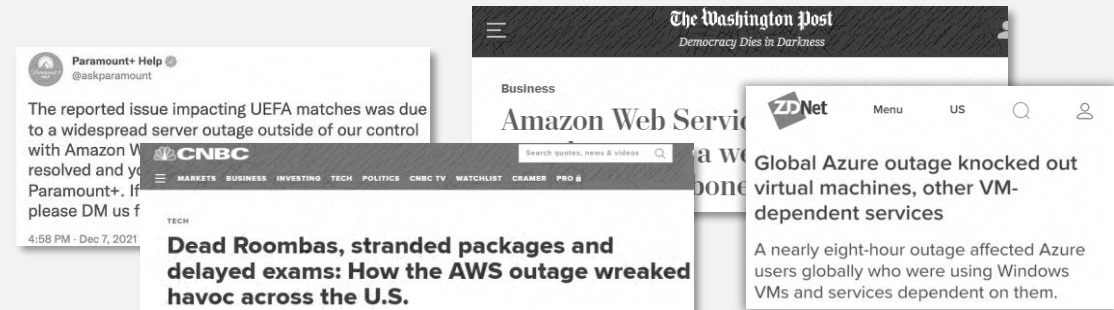
▷ The Promise of Cloud

- ▪ Scalable computing power for all
- ▪ Many business large and Small rely on the cloud today

▷ The Challenge of Multi Cloud

Multi-Cloud, Multi-Region is a Complex Implementation

▷ Failure is Inevitable and impacts Orgs differently

Social Media App	Bank / Airlines	Critical health care

- ▪ 1 Cloud   = Failure with the cloud
- ▪ 1 Cloud   = No Access to great services from other clouds

For many organizations,

- ▪ Multi-Cloud / Multi-region has become a Necessity
- ▪ Multi-Cloud = Resiliency, Freedom of Choice

HashiCorp   SUNRISE BANKS

▷ Me

First Name	Middle Name	Last Name
Sanjay	-	Narendran

- Born in Tamil Nadu, India, lives in St Paul, MN with wife and a 3-year-old
- Solutions Architect @ Sunrise Banks

▷ Multi-Cloud Challenge

Multi-Cloud, Multi-Region is a Significantly Complex Implementation

- Sizable challenge even for large enterprises
- Medium and Small enterprises, Complexity deters adoption

▷ Sunrise Banks

- A Technology first $2B bank Head Quartered in St Paul, MN. focused on communities' financial wellness.

- A proud member of GABV with the inseparable triple bottom line of people, planet and profit.

- A certified B Corp, a recognition given to organizations whose Leaders build Communities – not just profit.

Global Alliance for Banking on Values

Certified **B** Corporation

▷ The How

If the implementation is complex, then the design is wrong.
Is there a better design ?

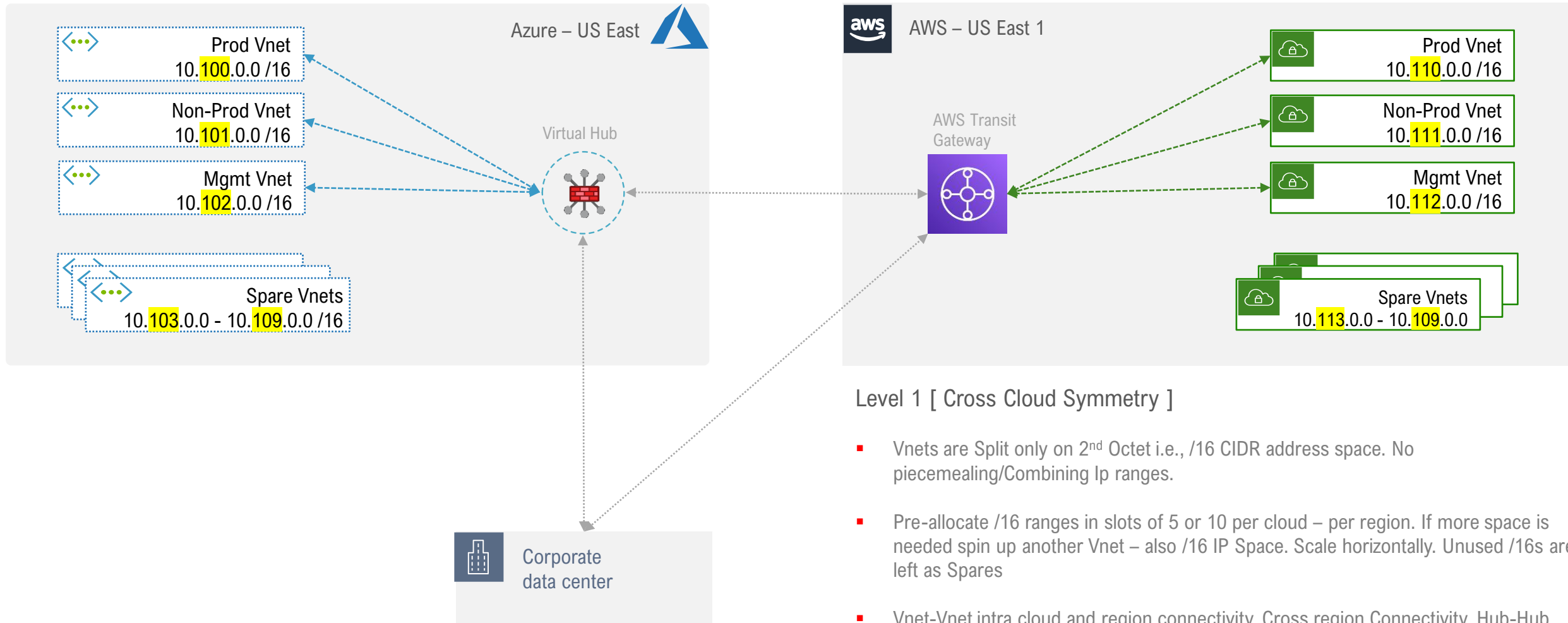| New Design	Implementation Platform
Symmetric Architecture	HashiCorp Cloud Platform

HashiCorp   SUNRISE BANKS

Symmetric Multi-Cloud Architecture

Level 1	Level 2	Level 3
Cross Cloud Symmetry	Inter-Vnet Symmetry	Standard Subnets

HashiCorp

SUNRISE BANKS

Symmetric Architecture : Level 1 [Cross Cloud Symmetry] ➡ Level 2 [Inter Vnet Symmetry] ➡ Level 3 [Standard Subnets]

Azure – US East

Prod Vnet
10.100.0.0 /16

Non-Prod Vnet
10.101.0.0 /16

Virtual Hub

Mgmt Vnet
10.102.0.0 /16

Spare Vnets
10.103.0.0 - 10.109.0.0 /16

AWS – US East 1

AWS Transit
Gateway

Prod Vnet
10.110.0.0 /16

Non-Prod Vnet
10.111.0.0 /16

Mgmt Vnet
10.112.0.0 /16

Spare Vnets
10.113.0.0 - 10.109.0.0

Corporate
data center

Level 1 [Cross Cloud Symmetry]

▪ Vnets are Split only on 2nd Octet i.e., /16 CIDR address space. No piecemealing/Combining Ip ranges.

▪ Pre-allocate /16 ranges in slots of 5 or 10 per cloud – per region. If more space is needed spin up another Vnet – also /16 IP Space. Scale horizontally. Unused /16s are left as Spares

▪ Vnet-Vnet intra cloud and region connectivity, Cross region Connectivity, Hub-Hub, Hubs-HQ connections all happen only via vendor regional hubs (i.e. Azure Virtual Hub, AWS Transit Gateway etc).

HashiCorp   SUNRISE BANKS

Symmetric Architecture : Level 1 [Cross Cloud Symmetry] ➔ Level 2 [Inter Vnet Symmetry] ➔ Level 3 [Standard Subnets]
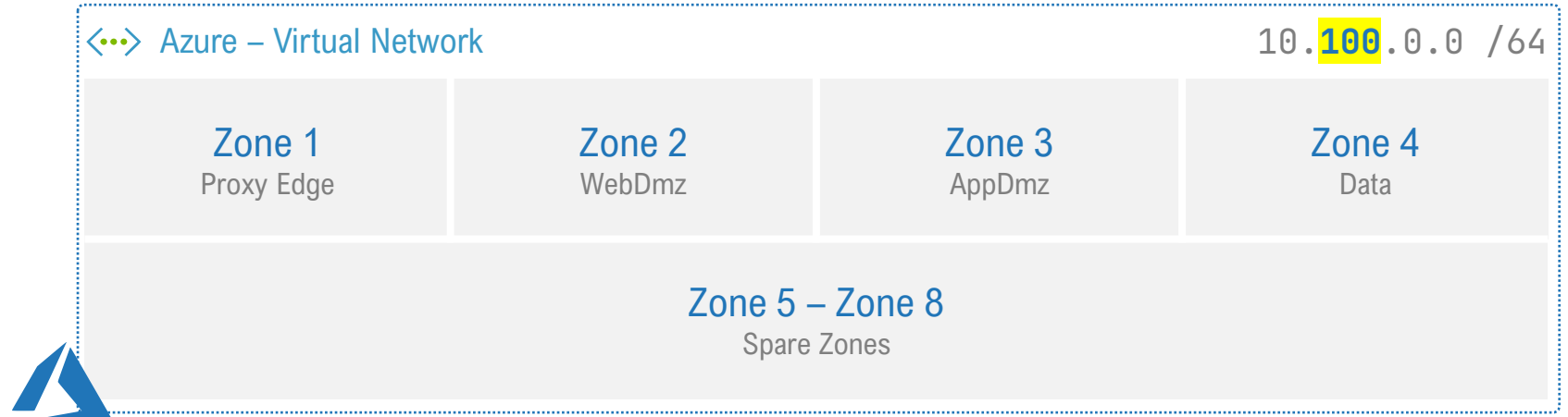
Symmetric Architecture : Level 1 [Cross Cloud Symmetry] ➡ Level 2 [Inter Vnet Symmetry] ➡ Level 3 [Standard Subnets]

Level 2 [Inter Vnet Symmetry]

- Vnet/VPC split into logical security zones

- Every Vnet/Vpc in any region any cloud follows the same pattern

- Leave Spare Ip ranges (i.e., Spare zones for future Zones or existing zone expansion)

- IP Space for each zone can vary, however it must be the same for all Vnets. Each Zone in every Vnet, across regions or across clouds will always start at the same 3rd Octet IP address

Azure – Virtual Network
10.100.0.0 /64

Zone 1	Zone 2	Zone 3	Zone 4
Proxy Edge	WebDmz	AppDmz	Data

Zone 5 – Zone 8
Spare Zones

AWS VPC
10.110.0.0 /64

Zone 1	Zone 2	Zone 3	Zone 4
Proxy Edge	WebDmz	AppDmz	Data

Zone 5 – Zone 8
Spare Zones

HashiCorp   SUNRISE BANKS

Symmetric Architecture : Level 1 [Cross Cloud Symmetry] ➡ Level 2 [Inter Vnet Symmetry] ➡ Level 3 [Standard Subnets]

Level 2 [Inter Vnet Symmetry]

- Vnet/VPC split into logical security zones

- Every Vnet/Vpc in any region any cloud follows the same pattern

- Leave Spare Ip ranges (ie Spare zones for future Zones or existing zone expansion)

- IP Space for each zone can vary, however it must be the same for all Vnets. Each Zone in every Vnet, across regions or across clouds will always start at the same 3rd Octet IP address

<···> Azure – Virtual Network   10.**100**.0.0 /64

1 - Proxy Edge	2 – WebDmz	3 – AppDmz	4 – Data
10.100.0.0/19	10.100.32.0/19	10.100.64.0/19	10.100.96.0/19

5 – 8 Spare Zones

AWS VPC   10.**110**.0.0 /64

1 - Proxy Edge	2 – WebDmz	3 – AppDmz	4 – Data
10.110.0.0/19	10.110.32.0/19	10.110.64.0/19	10.110.96.0/19

5 – 8 Spare Zones

HashiCorp   SUNRISE BANKS

Symmetric Architecture : Level 1 [Cross Cloud Symmetry] ➜ Level 2 [Inter Vnet Symmetry] ➜ Level 3 [Standard Subnets]

Level 3 [Standard Subnets]

- Identify and Pre-create Standard Subnets.

- Subnets are designed to group workload types – not applications (eg. Subnets for NVAs, VMs, K8S clusters, Databases etc)

- Resources within the subnets can/will vary across Vnets, but each Zone in Every Vnet across regions and clouds will have the same standard set of subnets.
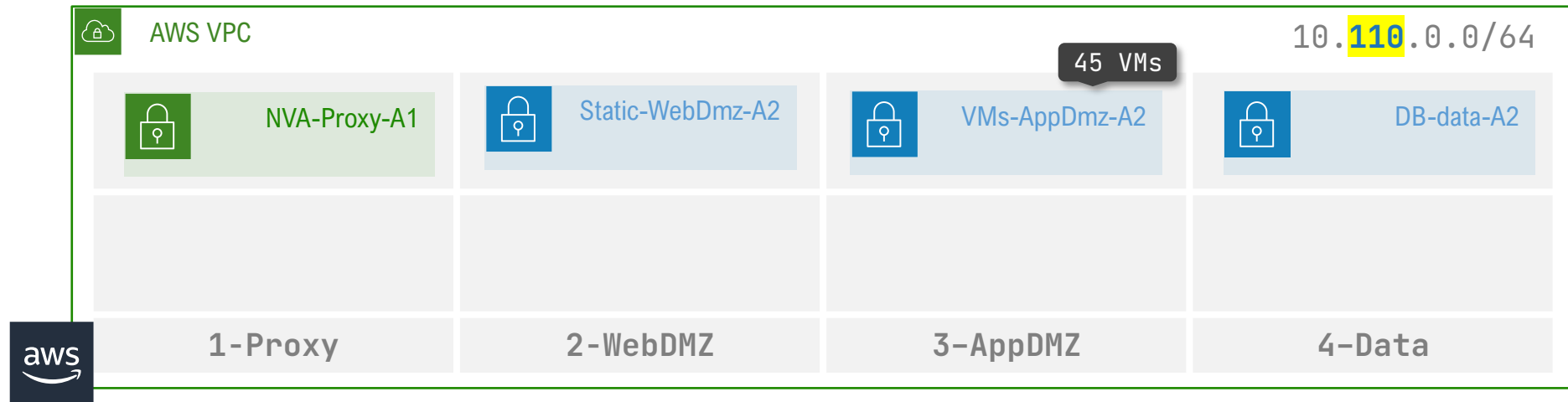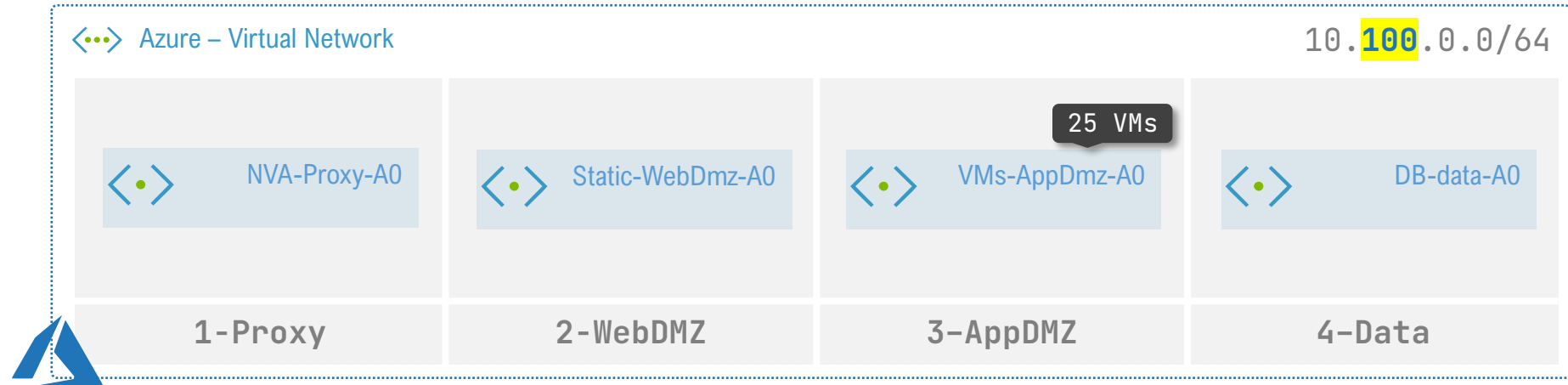
Azure – Virtual Network 10.`100`.0.0/64

NVA-Proxy-A0	Static-WebDmz-A0	VMs-AppDmz-A0	DB-data-A0
1-Proxy	**2-WebDMZ**	**3–AppDMZ**	**4-Data**

AWS VPC 10.`110`.0.0/64

NVA-Proxy-A1	Static-WebDmz-A2	VMs-AppDmz-A2	DB-data-A2
1-Proxy	**2-WebDMZ**	**3–AppDMZ**	**4-Data**

HashiCorp   SUNRISE BANKS

Symmetric Architecture : Level 1 [Cross Cloud Symmetry] ➜ Level 2 [Inter Vnet Symmetry] ➜ Level 3 [Standard Subnets]

Level 3 [Standard Subnets]

- There are cloud specific nuances which needs abstraction

- Regardless of Cloud specific abstractions, Standard subnets of each zone will be present in every vnet across regions or and clouds
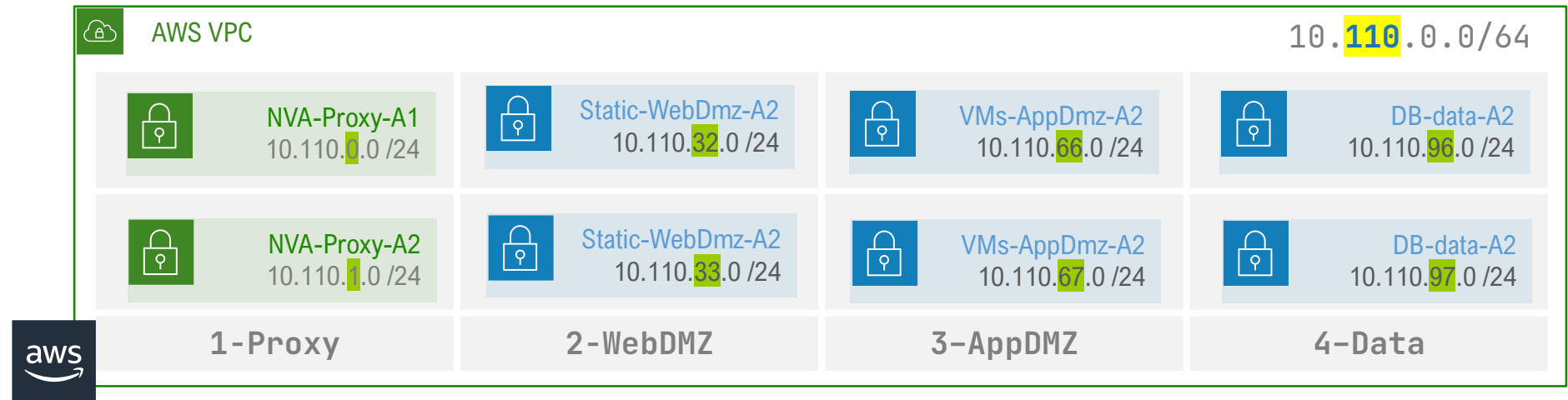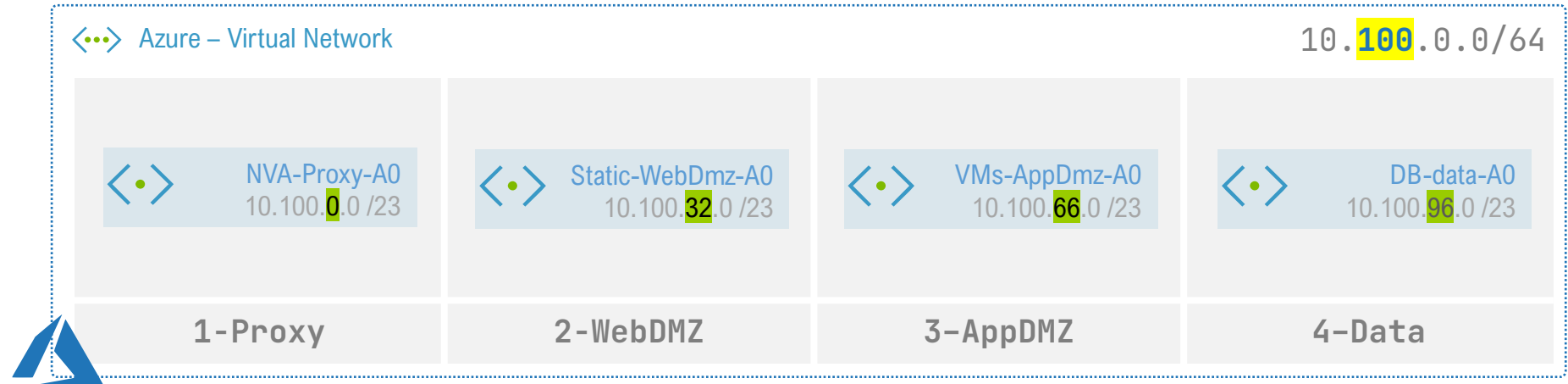
<·••> Azure – Virtual Network   10.100.0.0/64

<·•> NVA-Proxy-A0 10.100.0.0 /23	<·•> Static-WebDmz-A0 10.100.32.0 /23	<·•> VMs-AppDmz-A0 10.100.66.0 /23	<·•> DB-data-A0 10.100.96.0 /23
1-Proxy	**2-WebDMZ**	**3–AppDMZ**	**4-Data**

AWS VPC   10.110.0.0/64

🔒 NVA-Proxy-A1 10.110.0.0 /24	🔒 Static-WebDmz-A2 10.110.32.0 /24	🔒 VMs-AppDmz-A2 10.110.66.0 /24	🔒 DB-data-A2 10.110.96.0 /24
🔒 NVA-Proxy-A2 10.110.1.0 /24	🔒 Static-WebDmz-A2 10.110.33.0 /24	🔒 VMs-AppDmz-A2 10.110.67.0 /24	🔒 DB-data-A2 10.110.97.0 /24
1-Proxy	**2-WebDMZ**	**3–AppDMZ**	**4-Data**

HashiCorp   SUNRISE BANKS

Symmetric Architecture : Level 1 [Cross Cloud Symmetry] ➡ Level 2 [Inter Vnet Symmetry] ➡ **Level 3 [Standard Subnets]**

Level 3 [Standard Subnets]

- Each zone can contain any number of standard subnets of varying sizes

- They are expected to grow and change over time

- Leave Spare Ip ranges per zone for future Subnets and expansion of existing subnets

<·••> Azure – Virtual Network   10.**100**.0.0/64

SASE-Proxy-A0 10.100.**0**.0 /23	Static-WebDmz-A0 10.100.**32**.0 /23	VMs-AppDmz-A0 10.100.**64**.0 /23	DB-data-A0 10.100.**96**.0 /23
NVA-Proxy-A0 10.100.**2**.0 /22	API-WebDmz-A0 10.100.**34**.0 /23	AKS-AppDmz-A0 10.100.**66**.0 /22	NoSql-data-A0 10.100.**98**.0 /23
		AIML-AppDmz-A0 10.100.**70**.0 /22	Private-endpoint-A0 10.100.**100**.0 /22

Spare Subnets   Spare Subnets   Spare Subnets   Spare Subnets

1-Proxy	2-WebDMZ	3-AppDMZ	4-Data

HashiCorp   SUNRISE BANKS

Advantages

- **Cloud agnostic Blueprint** : Region level Blueprint of Hubs and Vnets : Lets us create entire datacenters at will on any cloud – any region.

- Symmetric Vnets/Vpc Design = Symmetric IAC Configs = our Tfvars all look the same

- Highly consistent cloud and region agnostic Network design

- Standard Security zones ensure consistent security posture

- Scalable and re-creatable multi-cloud pattern across the globe.

HashiCorp

SUNRISE BANKS

Implementation

HashiCorp Cloud Platform	Terraform Cloud	Vault

HashiCorp   SUNRISE BANKS

Implementation : Build one Vnet ➔ Build Any Vnet

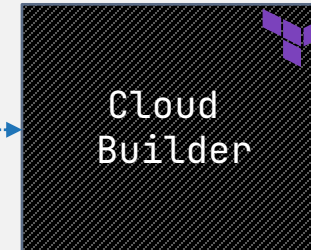## Multi-Cloud in 3 Steps

Hey Terraform,

1. Here is my TFVars

2. Create my datacenter in Azure US Central

3. Oh Btw, create one in AWS and GCP too

aws **AWS East 1 – Non Prod VPC**
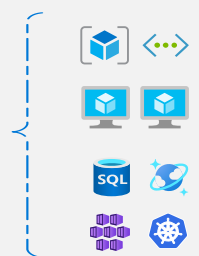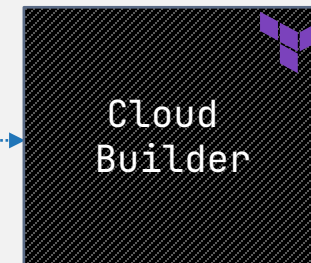
Terraform.TFVars

{} ➔ Cloud Builder ➔ VPC with Subnets, EC2, DBs etc . . .

Azure East 2 – Non Prod Vnet

Terraform.TFVars

{} ➔ Cloud Builder ➔ Vnet with Subnets, Vms, NICs, DBs etc . . .

HashiCorp   **SUNRISE BANKS**
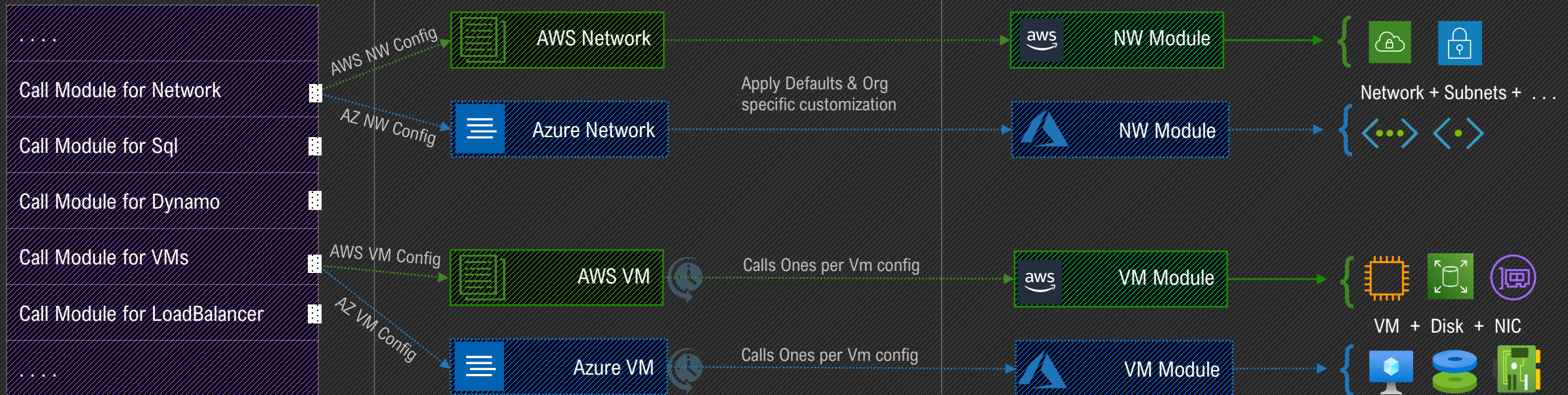
Implementation : Build one Vnet ➡ Build Any Vnet

Main Module (Main.tf)   Aggregator Modules   Primitives (Public / Private Modules)

... .

Call Module for Network   AWS Network   aws NW Module   { Network + Subnets + ...

Call Module for Sql   Azure Network   NW Module   { <···> <·>

Call Module for Dynamo   Apply Defaults & Org
                                 specific customization

Call Module for VMs   AWS VM Config   AWS VM   Calls Ones per Vm config   aws VM Module   { VM + Disk + NIC

Call Module for LoadBalancer   AZ VM Config   Azure VM   Calls Ones per Vm config   VM Module   {

... .

Terraform.TFVars

AWS VPC   10.110.0.0/64

NVA-Proxy-A1   Static-WebDmz-A2   VMs-AppDmz-A2   DB-data-A2
10.110.1.0 /24   10.110.32.0 /24   10.110.66.0 /24   10.110.96.0 /24

NVA-Proxy-A2   Static-WebDmz-A2   VMs-AppDmz-A2   DB-data-A2
10.110.1.0 /24   10.110.33.0 /24   10.110.67.0 /24   10.110.97.0 /24

aws   1-Proxy   2-WebDMZ   3-AppDMZ   4-Data

HashiCorp   SUNRISE BANKS

Implementation : Build one Vnet ➡ Build Any Vnet

Main Module Example

```
. . .
. . .

module "AZWinServers" {

    source  = "apps.terraform.io/urOrg/AzVMAgg/azure"
    version = "1.0.0"
    input_win_vmConfig_default      = var.az_win_vmConfig_default
    input_win_vmConfig              = var.az_win_vmConfig
    . . .
    . . .
}
```

Aggregator example

```
module "AZWinServersAggregate" {

    for_each = { for i, v in var.input_win_vmConfig : i =>
                            merge(var.input_win_vmConfig_default, v) }

    source  = "Azure/compute/azurerm"
    version = "3.14.0"
    . . .
    . . .
}
```

Terraform.tfvars example

```
az_win_vmConfig = {
  vm1 = {
    vm_hostname  = "AzVm1"
    subnet_name  = "ur-subnet-name"
    vm_size      = "Standard_DS_v2"
  },
  vm2 = {
    vm_hostname  = "AzVm2"
    subnet_name  = "ur-other-subnet-name"
    vm_size      = "Standard_BS"
  }
}
```

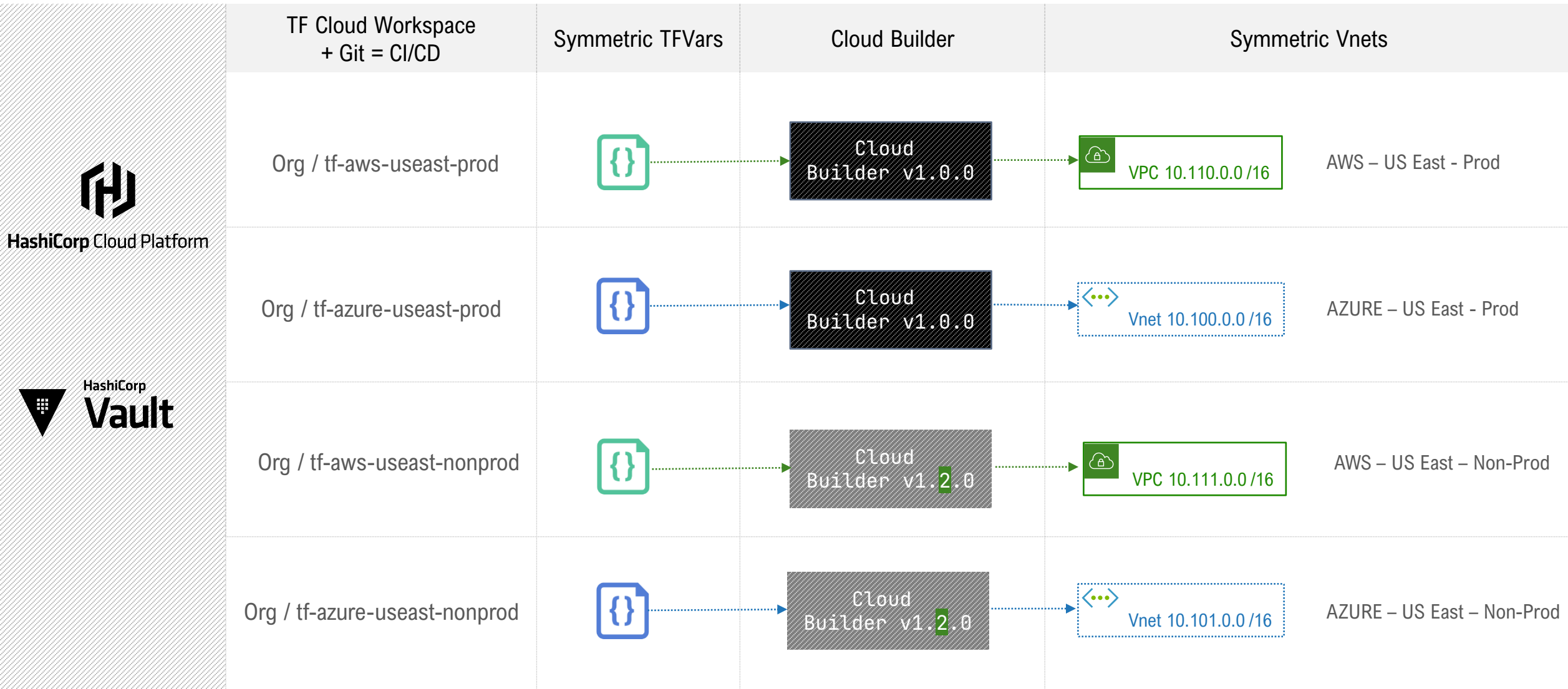## Primitives

```
module "windowsservers" {
    source              = "Azure/compute/azurerm"
    resource_group_name = azurerm_resource_group.example.name
    is_windows_image    = true
    vm_hostname         = "mywinvm"
    vm_os_simple        = "WindowsServer"
    . . .
    . . .
}
```

github.com/Azure/terraform-azurerm-compute

HashiCorp   SUNRISE BANKS

Implementation : Build one Vnet ➜ **Build Any Vnet**

	TF Cloud Workspace + Git = CI/CD	Symmetric TFVars	Cloud Builder		Symmetric Vnets
	Org / tf-aws-useast-prod	{}	Cloud Builder v1.0.0	VPC 10.110.0.0 /16	AWS – US East - Prod
	Org / tf-azure-useast-prod	{}	Cloud Builder v1.0.0	Vnet 10.100.0.0 /16	AZURE – US East - Prod
	Org / tf-aws-useast-nonprod	{}	Cloud Builder v1.2.0	VPC 10.111.0.0 /16	AWS – US East – Non-Prod
	Org / tf-azure-useast-nonprod	{}	Cloud Builder v1.2.0	Vnet 10.101.0.0 /16	AZURE – US East – Non-Prod

HashiCorp Cloud Platform

HashiCorp **Vault**

HashiCorp

SUNRISE BANKS

Symmetric patterns and blueprints simplify and solve complex real-world problems

Symmetric Cloud Architecture solves the multi-cloud problem the same way

https://www.linkedin.com/company/sunrise-banks

https://www.linkedin.com/in/sanjay-narendran

First Name	Middle Name	Last Name
Sanjay	-	Narendran
Veda	-	Sanjay

. . . Many years later . . . .

First Name	Middle Name	Last Name
<< >>	-	Veda
<< >>	-	Veda

. . . ☺ what is your name ? . . . .

HashiCorp   SUNRISE BANKS

Thank you !