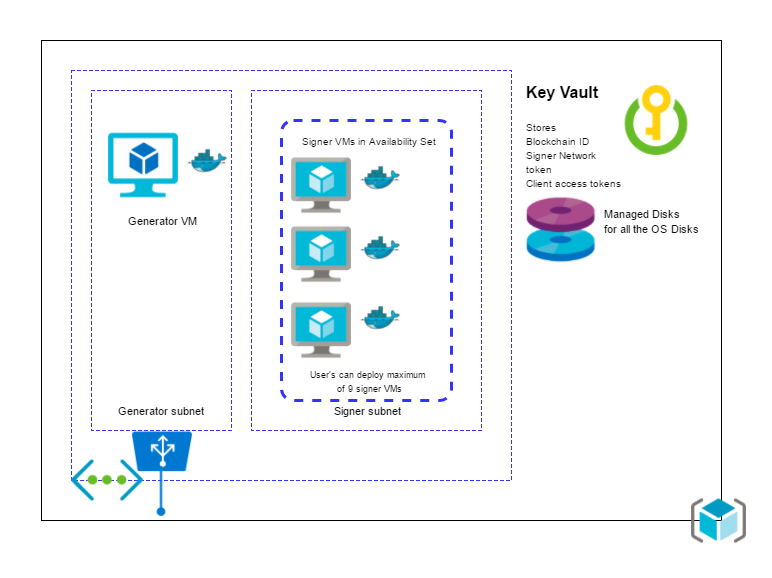
Chaincore Developer Edition

# Single Generator, Multi Signer, Single Region

## Architecture Diagram



## Description

1. A Generator VM (Ubuntu Canonical 16.04) which launches chaincore in port 1999 through chaincore Docker container.
   1. This VM is configured as a Block generator using a shell script as an extension to this VM.
   2. This VM is provisioned in a separate subnet ie generator subnet.
   3. It has an external load balancer from where internet traffic reaches the VM.
2. Next is the Signer VM (Ubuntu Canonical 16.04) which launches chaincore in port 1999 through chaincore Docker container.
   1. User has the option to provision up to 9 block signer VM’s
   2. These VM’s are configured as a Block signers using a shell script as an extension to the VM.
   3. These VM’s are provisioned in a separate subnet ie signer subnet.
3. A key vault to store the blockchain ID, network token for signers and client tokens for all the containers.
4. We use managed disks for provisioning the OS disks of the VM.

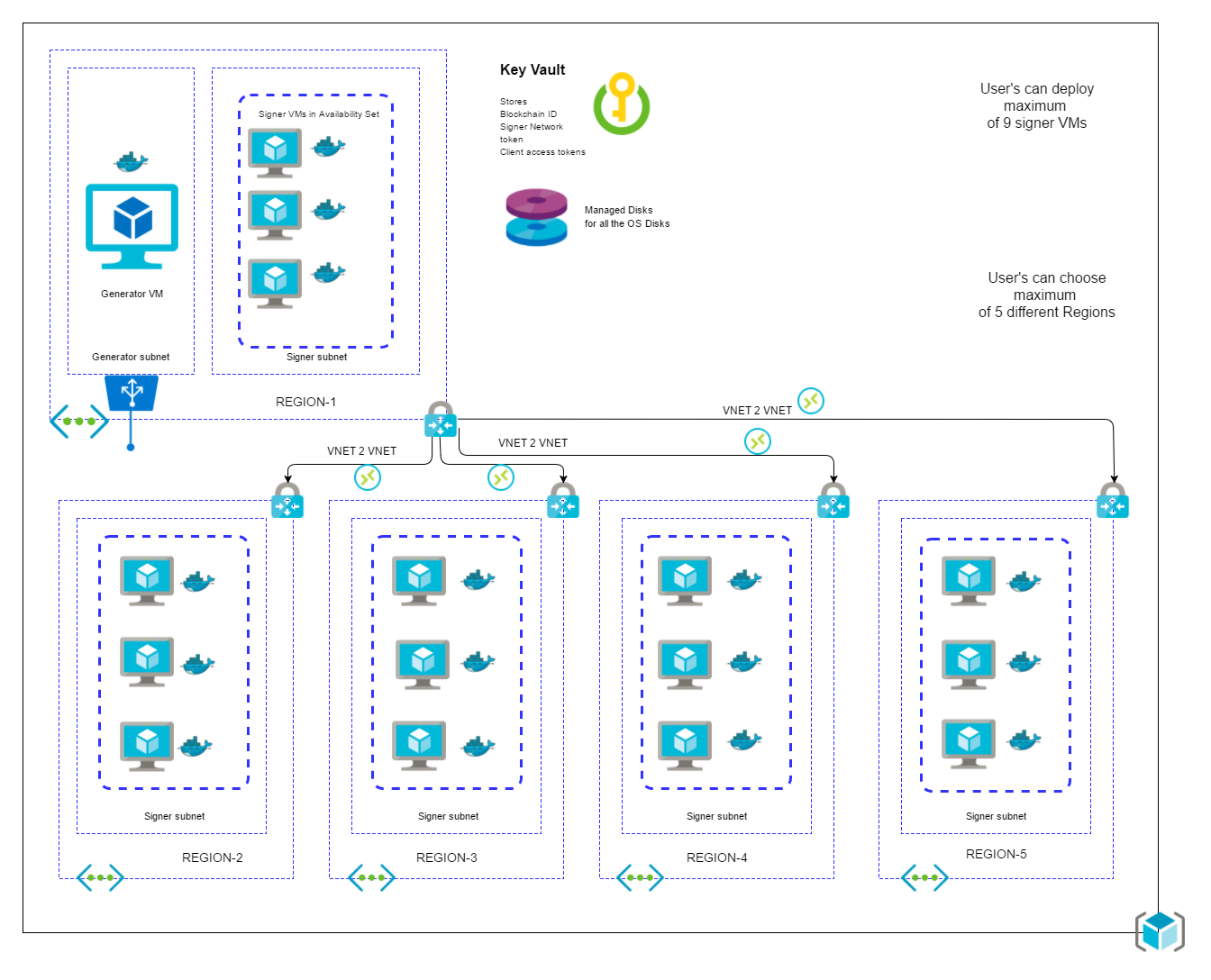
## Dataflow

The blockchain ID, network token for signers are stored in key vault from generator VM.

The Signer VMs retrieve blockchain ID and network token from the key vault and configure block signer and store signer client token in the key vault

# Single Generator, Multi Signer, Multi Region

## Architecture



## Description

1. A Generator VM (Ubuntu Canonical 16.04) which launches chaincore in port 1999 through chaincore Docker container.
   1. This VM is configured as a Block generator using a shell script as an extension to this VM.
   2. This VM is provisioned in a separate subnet ie generator subnet.
   3. It has an external load balancer from where internet traffic reaches the VM.
   4. Its deployed only in first region.
2. Next is the Signer VM (Ubuntu Canonical 16.04) which launches chaincore in port 1999 through chaincore Docker container.
   1. User has the option to provision up to 9 block signer VM’s
   2. These VM’s are configured as a Block signers using a shell script as an extension to the VM.
   3. These VM’s are provisioned in a separate subnet ie signer subnet.
   4. Its deployed in 5 regions based on number of regions selected by User.
3. A key vault to store the blockchain ID, network token for signers and client tokens for all the containers.
4. We use managed disks for provisioning the OS disks of the VM.
5. It deploys VNET gateways for each VNET in each region.
6. These VNETs communicate with each other using VNET2VNET connections

## Dataflow

The blockchain ID, network token for signers are stored in key vault from generator VM.

The Signer VMs retrieve blockchain ID and network token from the key vault and configure block signer and store signer client token in the key vault