

Cloud Solution Architecture

*for*

****

**Foodida MVP**

**Customer and Driver Application**

Document revisions:

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Author | Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Overview 5](#_Toc421984083)

[Architecture Design 5](#_Toc421984084)

[foodida Cloud 5](#_Toc421984085)

[Communication Flow 5](#_Toc421984086)

[Artefacts in Subnets 5](#_Toc421984087)

[External connectivity 6](#_Toc421984088)

[Infrastructure Components 6](#_Toc421984089)

[Virtual Private Cloud (VPC) 6](#_Toc421984090)

[Subnets 7](#_Toc421984091)

[Internet Gateway 7](#_Toc421984092)

[Route Tables 7](#_Toc421984093)

[Network ACL 8](#_Toc421984094)

[EC2 Security Groups 8](#_Toc421984095)

[Inf security groups 9](#_Toc421984096)

[NAT 9](#_Toc421984097)

[DNS 10](#_Toc421984098)

[IAM Roles 10](#_Toc421984099)

[Roles 10](#_Toc421984100)

[EC2 Instance Specifications 10](#_Toc421984101)

[NAT 11](#_Toc421984102)

[DNS 11](#_Toc421984103)

[Reverse Proxy 12](#_Toc421984104)

[API Server 12](#_Toc421984105)

[SAP Server 12](#_Toc421984106)

[RDS Specifications 13](#_Toc421984107)

[Database Subnet Groups 13](#_Toc421984108)

[S3 13](#_Toc421984109)

[Endpoints & Default Access Credentials 13](#_Toc421984110)

[Governance 14](#_Toc421984111)

[Scope 14](#_Toc421984112)

[Stakeholders 14](#_Toc421984113)

[Strategy & Roadmap 14](#_Toc421984114)

[Architecture & Technology Policies 14](#_Toc421984115)

[Security & Privacy 15](#_Toc421984116)

[Identity & Access Management 15](#_Toc421984117)

[Servers in EC2 15](#_Toc421984118)

[Source Code Repository 15](#_Toc421984119)

[Data Storage 15](#_Toc421984120)

[Operational Policies 15](#_Toc421984121)

[Consumption 16](#_Toc421984122)

[Bursting 16](#_Toc421984123)

[Management 16](#_Toc421984124)

[Monitoring 16](#_Toc421984125)

[Processes 16](#_Toc421984126)

[Strategy & Planning 16](#_Toc421984127)

[Deployment Processes 16](#_Toc421984128)

[Resource Management & Provisioning 16](#_Toc421984129)

[Monitoring, Alarms & Alerts 16](#_Toc421984130)

[Network Security 17](#_Toc421984131)

[Environments 17](#_Toc421984132)

[Security Groups 18](#_Toc421984133)

[EC2 Instance Specifications 19](#_Toc421984134)

[NAT 19](#_Toc421984135)

[DNS 19](#_Toc421984136)

[Reverse Proxy 20](#_Toc421984137)

[API Server 20](#_Toc421984138)

[SAP Server 20](#_Toc421984139)

[Relational Database Service (RDS) Specifications 21](#_Toc421984140)

[Database Subnet Groups 21](#_Toc421984141)

[S3 Bucket 21](#_Toc421984142)

[Endpoints & Default Access Credentials 21](#_Toc421984143)

[Backup & Restore 23](#_Toc421984144)

[Backup 23](#_Toc421984145)

[Restore 23](#_Toc421984146)

[Data Retention 24](#_Toc421984147)

[Capacity Scaling 24](#_Toc421984148)

[Disaster Recovery 24](#_Toc421984149)

[Monitoring 25](#_Toc421984150)

[RDS MySQL 25](#_Toc421984151)

# Overview

This document provides a comprehensive guide to the infrastructure of foodida solution, designed on Amazon Web Services (AWS).

The infrastructure is deployed on AWS’s Singapore South East region. The architecture makes use of multiple zones within this region. A single Virtual Private Cloud (VPC) is used to host the foodida architecture.

All the subnets within the foodida VPC are private, although one public subnet is used for access to the internet for in and out connections. NAT server is used for managing the traffic to and from the internet.

All the server instances have a security group assigned for first level of security. The public and private subnets also have route tables and ACL as a second level security.

The instances are in a dedicated private subnet hosted with a single zone. Nightly backup is taken on backup server and EBS snapshots taken and stored in the S3.

Extensive monitoring is set for all resources for pro-active decision making on performance scaling and capacity scaling.

# Architecture Design

This section illustrates in detail the architecture of the foodida. The diagrams are divided into different focuses so that specific details can be shown easily.

## foodida Cloud

Following diagram shows the foodida environments. Foodida consists of three different environments, ranging from development to production. Each environment is setup in a separate VPC in different regions.

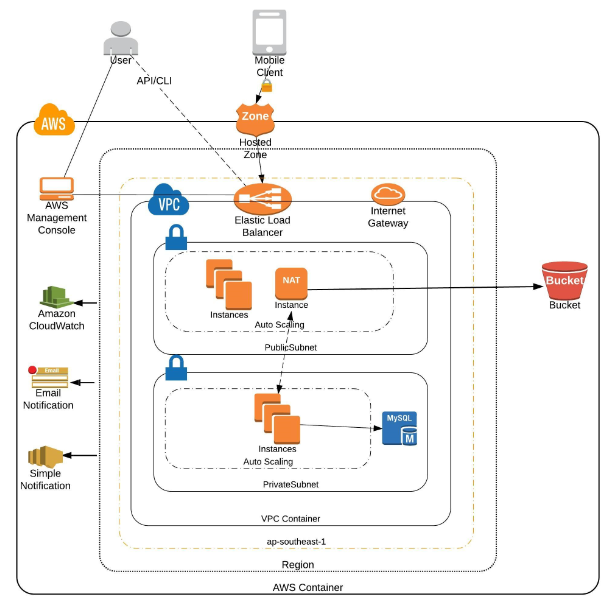


Fig. 1

### Communication Flow

### Artefacts in Subnets

Below are the artefacts setup in each of the subnets:

#### PublicA

* NAT
* DNS
* Reverse Proxy
* API Server
* SAP Server

#### PrivateA

* mysql

### External connectivity

The public subnet in ap-southeast-1a is used for outgoing and incoming traffic to and from the internet for the VPC. This public subnet hosts the NAT instance which provides routing to the internet via the internet gateway

In the zones ap-southeast-1a which are private subnets, will host all the server resources, e.g. mysql,SAP server etc..

# Infrastructure Components

This section details out all the infrastructure components used in this solution. Every component is listed detailing their purpose in the architecture and configuration around them.

The infrastructure is launched in the Singapore region and all three zones of this region are used by the architecture. Below are the region and zone codes in use:

|  |  |
| --- | --- |
| Region | Zones |
| ap-southeast-1 | ap-southeast-1a  ap-southeast-1b |

## Virtual Private Cloud (VPC)

Amazon Web Service’s VPC enables us to launch AWS resources within a virtual private network. This VPC closely resembles a traditional network that is operated in a typical data centre, but with a benefit of scalable infrastructure of AWS.

The foodida solution is hosted within a single VPC, which is logically isolated from other virtual networks in AWS. Below are the details of particulars of this VPC and their configuration around each service of the VPC.

The VPC is set to the following configuration:

|  |  |  |  |
| --- | --- | --- | --- |
| VPC Name | CIDR | Gateway | Tenancy |
| foodida-vpc | 10.0.0.0/16 | igw-foodida | Default |

Note: The name and the CIDR range can be changed to whatever is suitable. For the purpose of this documentation we will use the above values. The VPC name used here is also representative to this document.

#### Tenancy

We recommend for the tenancy to be set to “default”. If a “dedicated” option is set, then all the instances launched within the VPC will automatically be set to dedicated, and in a result cost will be increased considerably. If for any reason, a single-tenant (dedicated) instance is required to be launched, then this can be done at the instance level during the launch sequence.

### Subnets

Subnet is a range of IP addresses within the VPC. Both public and private subnets are available in this VPC. The resources that require access to the internet have an IP in public subnet. The resources that require no internet access and for added security, have an IP within the private subnet. But, these resource inside the private subnets can still access the internet via NAT instance within the public subnet as an interim solution.

Below are the launched subnets and attached to the need to update VPC:

|  |  |  |  |
| --- | --- | --- | --- |
| Subnet Name | CIDR | Type | Zone |
| publicAsubnet | 10.0.0.0/24 | public | ap-southeast-1a |
| privateAsubnet | 10.0.1.0/24 | private | ap-southeast-1b |
|  |  |  |  |
|  |  |  |  |

**Note:**

### Internet Gateway

Internet gateway lets the VPC have a connection to the internet. This is needed for our Public subnet to have connection to the outside world in order to accept incoming connections and outgoing responses too.

The internet gateway is created and attached to the :

|  |  |
| --- | --- |
| Name | VPC |
| igw-foodida | foodida-vpc |

### Route Tables

Route table provides routing for the subnet for the traffic originating from with the subnet. Below are the route tables for the subnets within the foodida-vpc:

#### public subnet route table

|  |  |
| --- | --- |
| Destination | Target |
| 0.0.0.0/0 | igw-foodida |
| 10.0.0.0/16 | Local |

The above route table for private subnets shows the non-local IPs (0.0.0.0/0) to be routed to “eni-xxxxxx” which is the Elastic Network Interface ID for the NAT instance in the public subnet.

#### foodida-vpc\_private subnet-Inf subnet route table

|  |  |
| --- | --- |
| Destination | Target |
| 0.0.0.0/0 | eni-xxxxxx |
| 10.0.0.0/16 | Local |
| <CIDR of FOODIDA> | vgw-xxxxxx |
| <CIDR of VIRTUSA> | vgw-xxxxxx |
|  |  |

### Network ACL

These are the access control list for subnets within the VPC. This is a security setting which allows and disallows the traffic to and from the subnets that are attached to the ACL rules. Network ACL is a stateless, meaning if a traffic from port 80 is allowed in, it does not have right to send back the response. For that, another outbound rule to allowing the response must also be set.

By default we leave the ACL rules to basic, which is allowing all traffic inbound and outbound. The ACLs are subnet level security, hence it is recommended to only be used for denying certain unwanted traffic to your public subnets.

The security groups will be used to fine tune the access control to the instances.

#### Public subnet ACL rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inbound | | | | |
| Rule | Source | Protocol | Port | Allow/Deny |
| 100 | 0.0.0.0/0 | ALL | ALL | Allow |
| \* | 0.0.0.0/0 | ALL | ALL | Deny |
| Outbound | | | | |
| 100 | 0.0.0.0/0 | ALL | ALL | Allow |
| \* | 0.0.0.0/0 | ALL | ALL | Deny |

#### Private subnet ACL rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inbound | | | | |
| Rule | Source | Protocol | Port | Allow/Deny |
| 100 | 0.0.0.0/0 | ALL | ALL | Allow |
| \* | 0.0.0.0/0 | ALL | ALL | Deny |
| Outbound | | | | |
| 100 | 0.0.0.0/0 | ALL | ALL | Allow |
| \* | 0.0.0.0/0 | ALL | ALL | Deny |

## EC2 Security Groups

The security groups are instance level settings providing security around the traffic flow to and from the instance. Security groups are stateful hence no return communication rule for inbound traffic is needed under the outbound rules.

Following are the settings for the security groups:

### Inf security groups

foodida-vpc\_inf-core-local

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Port | Protocol | Source/Destination |
| Outbound | | | |
| HTTP | 80 | TCP | 0.0.0.0/0 |
| HTTPS | 443 | TCP | 0.0.0.0/0 |
| DNS | 53 | TCP | foodida-vpc\_DNS |
| DNS | 53 | UDP | foodida-vpc\_DNS |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* Outbound port 53 tcp & udp for DNS
* Outbound port 80 & 443 for general internet (goes via NAT)

foodida-vpc\_inf-monitoring-local

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Port | Protocol | Source/Destination |
| Outbound | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### NAT

foodida-vpc\_NAT

|  |  |  |  |
| --- | --- | --- | --- |
| Inbound | | | |
| Type | Port | Protocol | Source/Destination |
| HTTP | 80 | TCP | 10.0.0.0/16 |
| HTTPS | 443 | TCP | 10.0.0.0/16 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Outbound | | | |
| HTTP | 80 | TCP | 0.0.0.0/0 |
| HTTPS | 443 | TCP | 0.0.0.0/0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* Inbound ports 80 & 443 for all instances within the VPC
* Outbound ports 80 & 443 for accessing the internet

### DNS

foodida-vpc\_DNS

|  |  |  |  |
| --- | --- | --- | --- |
| Inbound | | | |
| Type | Port | Protocol | Source/Destination |
| DNS | 53 | TCP | 10.0.0.0/16 |
| DNS | 53 | UDP | 10.0.0.0/16 |
| Outbound | | | |
| DNS | 53 | TCP |  |
| DNS | 53 | TCP |  |
| DNS | 53 | UDP |  |
| DNS | 53 | UDP |  |
| TCP | 4001 | TCP |  |
| TCP | 4001 | TCP |  |

* Inbound ports 53 for all instances within the VPC for DNS

## IAM Roles

Identity Access Management (IAM) is a web service that provides management of user permissions. IAM Role is a service that delegates access to AWS services on temporary basis to any of the AWS resources at a run-time programmatically assuming a role. These roles are then assigned series of user rights.

The resources within this infrastructure are using IAM Roles for authorisation to services.

### Roles

|  |  |
| --- | --- |
| Name | Access to |
| foodida-backup | EBS Snapshots |
|  |  |
|  |  |
|  |  |

## EC2 Instance Specifications

Elastic Cloud Compute (EC2) are the virtual servers on AWS, which can be provisioned within few minutes. EC2’s come in many different types ranging from small to very powerful servers.

In this EDW infrastructure we are using few different types of server depending on the task the server is assigned. Below are the instances highlighted with their details and configurations?

### NAT

Network Address Translator (NAT) server is required to sit in the public subnet to provide internet access to the servers within the private subnets in the VPC.

|  |  |
| --- | --- |
| Info | Value |
| Name | foodida-vpc.inf.nat |
| Size | m3.medium |
| AMI | Amazon Linux AMI VPC NAT x86\_64 PV |
| Stack | Amazon Linux, NAT |
| Services | * iptables on * ip6tables off * sendmail off * crond off * NAT on |
| IAM Role | None |
| Zone | ap-southeast-1a |
| Quantity | 1 |
| Load balanced | No |
| Storage (Ephemeral) | Root, 10GB |
|  | |
| Subnet | foodida-vpc\_publicAsubnet |
| Security Group | foodida-vpc\_nat  foodida-vpc\_inf-core-local  foodida-vpc\_inf-monitoring-local |
| IP |  |

### DNS

Local DNS server managing DNS names / endpoints to be used in the VPC.

|  |  |
| --- | --- |
| Info | Value |
| Name | foodida-vpc.dev1.inf.dns |
| Size | m3.medium |
| AMI |  |
| Stack | Linux |
| Services | * iptables on * ip6tables off * sendmail off * crond off |
| IAM Role | None |
| Zone | ap-southeast-1a  ap-southeast-1b |
| Quantity | 2 |
| Load balanced | No |
| Storage (Ephemeral) | Root, 10GB |
| eth0 | |
| Subnet | foodida-vpc.dev1\_privateAsubnet  foodida-vpc.dev1\_privateBsubnet |
| Security Group | foodida-vpc.dev1\_DNS  foodida-vpc.dev1\_inf-core-local  foodida-vpc.dev1\_inf-monitoring-local |

### Reverse Proxy

|  |  |
| --- | --- |
| Info | Value |
| Name |  |
| Size |  |
| AMI |  |
| Stack |  |
| Services |  |
| IAM Role |  |
| Zone |  |
| Quantity |  |
| Load balanced |  |
| Storage (Ephemeral) |  |
| Storage (Instance) |  |
|  | |
| Subnet |  |
| Security Group |  |
| IP |  |

### API Server

|  |  |
| --- | --- |
| Info | Value |
| Name |  |
| Size |  |
| AMI |  |
| Stack |  |
| Services |  |
| IAM Role |  |
| Zone |  |
| Quantity |  |
| Load balanced |  |
| Storage (Ephemeral) |  |
| Storage (Instance) |  |
| eth0 | |
| Subnet |  |
| Security Group |  |
| IP |  |

### SAP Server

|  |  |
| --- | --- |
| Info | Value |
| Name |  |
| Size |  |
| AMI |  |
| Stack |  |
| Services |  |
| IAM Role |  |
| Zone |  |
| Quantity |  |
| Load balanced |  |
| Storage (Ephemeral) |  |
| Storage (Instance) |  |
| eth0 | |
| Subnet |  |
| Security Group |  |
| IP |  |

## RDS Specifications

Relational Database Service (RDS) is used to launch MySQL database servers

### Database Subnet Groups

DB subnet groups are required before database servers are created. We need one DB Subnet Group for our “services” subnets.

#### DB Subnet Group

|  |  |
| --- | --- |
| Info | Value |
| Name | Services-Subnet |
| VPC | foodida-vpc |
| Availability Zone | ap-southeast-1b |
| Subnet ID | foodida-vpc\_privateAsubnet |

## S3

Simple Storage System (S3) is highly available and durable storage system within the AWS. It provides unlimited storage size, which removes any maintenance around disk usage monitoring.

The backup snapshots of the mysql databases are stored into S3.

## Endpoints & Default Access Credentials

Below are the endpoints for each environment and the default credentials which are used during the initial installation. It is highly recommended that these default credentials are changed by the foodida team.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# Governance

Cloud governance is an important part during the implementation phase and then the lifecycle planning for the cloud solution. There are five primary reasons for cloud governance:

1. Enables cloud centric IT model based on speed, agility and cost
2. Enables appropriate decision making
3. Enables integration with existing IT processes, policies and tools
4. Enables balanced coverage for key decisions, investments and risks
5. Enables anticipation and prevention of unauthorised cloud activities that expose organisational risks

## Scope

The scope is not just limited to project wide, but is an integral part of the foodida business.

## Stakeholders

|  |  |
| --- | --- |
| Owner | Accountable for |
| Head of Data | 1. Data ownership 2. Data access 3. Decisions around data architecture 4. Functional requirements 5. Decisions around usage of management and operational tools 6. Software licencing |
| Head of Service Delivery, IT | 1. Infrastructure ownership 2. Infrastructure architecture design & performance 3. Deployment operations 4. System uptime 5. Infrastructure cost |

## Strategy & Roadmap

The cloud implementation of foodida has a principle of being “cloud agnostic” to provide greater freedom in terms of changing the cloud infrastructure provider in future.

Primary concern is not to use any Amazon specific service which is not possible to be replicated in another cloud provider or even migrating to in-house data centre.

## Architecture & Technology Policies

The architecture growth and change should be done with the following checklist:

* Artefact must be launched within the foodida-vpc VPC
* Artefact must be in private subnets of foodida-vpc VPC
* Artefact must not have publically accessible Elastic IP assigned to it
* The sizing of the EC2 instance must be in-line with the vendor specifications and foodida business requirements

## Security & Privacy

All security layers available by the AWS is to be applied to the entire architecture.

### Identity & Access Management

The AWS Management Console must only be accessed by authorised AWS’s IAM User.

1. IAM accounts must be created for different level of accessibility
2. Users & Groups should be utilised to manage the access to AWS resources
3. Appropriate level of access should be granted to different types of users & groups
4. The AWS account must be switched on for Multi Factor Authentication (MFA) login
5. IAM Roles must be used if any of the EC2 server requires access to any other AWS service e.g. S3
6. Separate IAM users & groups should be created for each environment i.e. development, test, production

### Servers in EC2

Following checklist to be followed for provisioning EC2 servers:

1. Server must be launched within the foodida-vpc VPC
2. Server must be assigned a Security Group
3. The Security Group must only allow required protocols and ports for all in/out traffic
4. Iptables must be switched on and mimic the exact rules as security groups
5. Servers must only reside in Private subnets
6. The Private subnets must have an ACL & route table
7. Any server must only be accessed via SSH Key pair for Linux servers and strong passwords for Windows servers
8. Server must be assigned an IAM Role if it requires an access to S3, and the S3 rights given to that IAM Role

### Source Code Repository

### Data Storage

The data in the foodida has the following checklist:

1. The incoming data from external data sources must use SSL connection to S3 while uploading
2. The backup files are saved in backup server
3. The backup server also takes nightly EBS snapshots
4. Any data in Glacier has 7 year life cycle before its deleted

## Operational Policies

Some key operational policies are to be observed once the architecture is running with the foodida.

### Consumption

All the EC2 servers are provisioned as “on demand” instances. Once the architecture has been in production for three months and reaches relatively stable state, then appropriate amount and type of “reserved” instances should be purchased.

All the core instances should be purchased as “reserved”. This will achieve around 30% cost savings over the year. All the instances must be set for “100% utilisation” when purchasing.

### Bursting

Appropriate amount of scaling, either vertically or horizontally should be achieved in the event of spike in the traffic.

### Management

The entire foodida’s cloud architecture is to be managed via AWS Console.

### Monitoring

Every artefact in the foodida architecture should have relevant monitoring setup, which is to be used for making decisions on specification fine tuning, finding the threshold for scaling up or down, and getting alerted for any fault or loss of service threat, that can be pro-actively predicted.

## Processes

Some governance processes that needs to be met for ongoing management and future changes.

### Strategy & Planning

Every aspect of the foodida infrastructure and architecture has to be aligned with wider standards and must comply with “Architecture & Technology Policies” chapter above. Planning must be carried out first in the form of illustrative diagram showing the new or changed artefact in high level state positioned within the foodida and then more focused version clearly showing the connectivity and communication flows of the architecture. The architecture must comply to all of the security standards as mentioned in “Security & Privacy” chapter above.

### Deployment Processes

Deployment the artefacts in the foodida architecture has to be done via AWS management console. Every part of the architecture is to be automated and automation script created for any new or changed aspect of the architecture.The aim is to have the “Infrastructure as a Code”, where the entire foodida infrastructure can be provisioned via deployment scripts. This gives greater control and ease of provisioning further environments.

### Resource Management & Provisioning

The architecture is managed via the AWS management console only. All resource provisioning is done from it. The provisioning of the resources must satisfy the business and technical requirements, but not vastly exceed it, preventing unnecessary cost and unused compute power.

### Monitoring, Alarms & Alerts

Alarms are set on key metrics to identify stress level of a server, e.g. monitoring CPU, RAM and disk usage. The alarms are raised at a set threshold depending on functionality and responsibility of each server. These alarms are then sent to some alert system which sends an email to all subscribers of the alarms via the AWS’s Simple Email Service (SES).

# Network Security

Network security is achieved adhering to the following principles and practices and ensuring 5-levels of security is accomplished:

Level 1:

1. The foodida infrastructure is setup within a VPC
2. The VPC is divided into public and private subnet
3. All the artefacts are hosted within the private subnet
4. Public subnet is only used for NAT server
5. The connectivity to the servers within the private subnet is via VPN or AWS Direct Connect

Level 2:

1. VPC has an ACL setup for the private subnets to regulate the ingress & egress of the traffic
2. Public subnets have a separate ACL
3. Any unwanted traffic is to be blocked at this ACL level

Level 3:

1. Security Group is assigned to every server
2. Security Group has detailed rules around ingress and egress of the traffic
3. Security Group controls the rules at protocol and port level with source and destination of the traffic

Level 4:

1. Iptables are used on every server
2. Iptables provide OS level firewall
3. Iptables copies the rules from Security Groups

Level 5:

1. The servers are not accessed via SSH frequently
2. In case direct access is required all the servers are enabled SSH-Key login
3. The SSH-Keys are created via AWS Key-Pair utility

# Environments

The EDW infrastructure consists of 5 environments:

1. Dev
2. Test
3. Production

As a simple rule, the non-production environments are at 25% capacity of the Production environment.

In this document, under the “Infrastructure Components” chapter, the artefacts are sized for the Production environment. In this chapter, the size and capacity of the artefacts are shown for the use in the remaining four environments.

Every environment will be placed in its own VPC. The details of the VPC are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Environment | VPC Name | CIDR | Gateway |
| Dev | foodida-vpc.dev1 | 10.0.0.0/16 | vgw-xxxxxxxx |
| Test | foodida-vpc.test1 |  |  |
| Production | foodida-vpc.prod1 |  |  |
|  |  |  |  |
|  |  |  |  |

Note: The name and the CIDR range can be changed to whatever is suitable. For the purpose of this documentation we will use the above values. The VPC name used here is also representative to this document. The Tenancy is set to “Default” for all the environments.

The route tables, Network ACL rules, Security Groups and IAM Roles are created within each VPC same as outlined in above chapter “Infrastructure Components” for production environment.

## Security Groups

Following are the security group rules in addition to the “Production” environment security groups outlined above in this document. The below groups are for non-production environments only and are added to the instances in addition to the production rules.

### 

|  |  |  |  |
| --- | --- | --- | --- |
| Inbound | | | |
| Type | Port | Protocol | Source/Destination |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Inbound | | | |
| Type | Port | Protocol | Source/Destination |
| None |  |  |  |

## EC2 Instance Specifications

The instance settings required:

### NAT

|  |  |
| --- | --- |
| Info | Value |
| Name | foodida-vpc-<env>.network.nat |
| Size | m3.medium |
| AMI | Amazon Linux AMI VPC NAT x86\_64 PV |
| Stack | Amazon Linux, NAT |
| Services | * iptables on * ip6tables off * sendmail off * crond off * NAT on |
| IAM Role | None |
| Zone | ap-southeast-1a |
| Quantity | 1 |
| Load balanced | No |
| Storage (Ephemeral) | Root, 10GB |
| eth0 | |
| Subnet | foodida-vpc.<env>\_publicsubnet |
| Security Group | foodida-vpc.<env>\_nat |

### DNS

|  |  |
| --- | --- |
| Info | Value |
| Name | Foodida-vpc.<env>.inf.dns |
| Size | m3.medium |
| AMI |  |
| Stack | Linux |
| Services | * iptables on * ip6tables off * sendmail off * crond off |
| IAM Role | None |
| Zone | ap-southeast-1a  ap-southeast-1b |
| Quantity | 2 |
| Load balanced | No |
| Storage (Ephemeral) | Root, 10GB |
|  | |
| Subnet | foodida-vpc.dev1\_PrivateAsubnet  foodida-vpc.dev1\_PrivateBsubnet |
| Security Group | foodida-vpc.dev1\_DNS  foodida-vpc.dev1\_inf-core-local  foodida-vpc.dev1\_inf-monitoring-local |

### Reverse Proxy

|  |  |
| --- | --- |
| Info | Value |
| Name |  |
| Size |  |
| AMI |  |
| Stack |  |
| Services |  |
| IAM Role |  |
| Zone |  |
| Quantity |  |
| Load balanced |  |
| Storage (Ephemeral) |  |
| eth0 | |
| Subnet |  |
| Security Group |  |

### API Server

|  |  |
| --- | --- |
| Info | Value |
| Name |  |
| Size |  |
| AMI |  |
| Stack |  |
| Services |  |
| IAM Role |  |
| Zone |  |
| Quantity |  |
| Load balanced |  |
| Storage (Ephemeral) |  |
|  | |
| Subnet |  |
| Security Group |  |

### SAP Server

|  |  |
| --- | --- |
| Info | Value |
| Name |  |
| Size |  |
| AMI |  |
| Stack |  |
| Services |  |
| IAM Role |  |
| Zone |  |
| Quantity |  |
| Load balanced |  |
| Storage (Ephemeral) |  |
|  | |
| Subnet |  |
| Security Group |  |

## Relational Database Service (RDS) Specifications

Databases required in each environment.

### Database Subnet Groups

DB subnet groups are required before database servers are created. We need one DB Subnet Group for our “services” subnets. The below database subnet group is created for each environment.

|  |  |
| --- | --- |
| Info | Value |
| Name | Services-Subnet |
| VPC | foodida-vpc.<env> |
| Availability Zone | ap-southeast-1b |
| Subnet ID | foodida-vpc<env>\_privatesubnet |

## S3 Bucket

A single S3 bucket is used for all non-production environments. The separation for each environment is by folder level.

Bucket name:

|  |  |  |
| --- | --- | --- |
| Folder Path | Application | Environment |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Endpoints & Default Access Credentials

Below are the endpoints for each environment and the default credentials which are used during the initial installation. It is highly recommended that these default credentials are changed by the foodida team.

|  |
| --- |
| Endpoints |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# Backup & Restore

In the foodida architecture only the mysql instances are required to have data backup & restore procedure in place. Other systems are deployed on high availability architecture and their backup and restore is managed by AWS services.

### Backup

Below are the attributes for the backup process.

#### Backup utility

#### Storage

Once the backup script is running, it will save the file to the backup server. Every Vertica node will transfer its own backup to this backup server.

Storage utilisation example:

|  |  |
| --- | --- |
| Name | Value |
| Total Nodes |  |
| Total Data |  |
| Unique data of the node |  |
| Buddy data on the node |  |
| Total data on the node |  |
| **Total backup file size** |  |

#### EBS Snapshots

The backups on the backup server are available for 35 days. Any further backups will simply be added and the oldest will be removed. This backup server has multiple EBS attached, and the snapshot of these EBS drives are automatically taken every night. These snapshots are saved into S3. This operation is needed just for data resilience, as the EBS is single zone service, and to achieve higher data resilience the snapshots are saved into S3 which is regional wide service.

#### Cross Zone Backup

Another fully operational backup server in different availability zone of Dublin region is setup. The primary backup server after receiving nightly backup then rsyncs the data with this secondary backup server in different zone. This is for DR process to have backup data ready in different zone, in case of entire zone failure.

### Restore

The restore of the foodida database can be carried out from the backup files stored on the backup server.

For a single node out of the three node cluster to be failed, a restore is not required.

The restore is required on any of the following scenarios:

1. 49% of the nodes fail in the cluster with data loss
2. New cluster of foodida database is launched

#### Restore from backup

In the event of carrying out the restore, all backup files for each node are available in the backup server. The restore utility is then run on any one of the nodes.

Once the restore is complete, the database is then switched on.

#### Restoring Backup server from EBS Snapshot

In the event that the backup server had a major incident or crashed beyond recovery, then a new server can be setup. The next step is to create new EBS volumes from latest EBS snapshot. Once the new EBS volumes are available, they are then attached to the new server.

Please note, recovery from EBS snapshots will only be required if region wide failure occurs on AWS. The DR for region failover will only be done for Vertica and it will be in read-only mode until the original region is back up and running.

The volumes will need to be attached in the original order and striped with RAID0 using “*adadm”* tool for creating logical volume. The amount of time required for creating new volumes from the snapshots is dependent upon the total data stored in the snapshots.

### Data Retention

The EBS attached to the backup server has persistent data and is not deleted unless it is manually detached and deleted.

The EBS snapshots are also taken from the EBS volumes on nightly basis. These snapshots are kept with no time limit until the original EBS volumes are deleted. These snapshots are saved in S3 for additional data resilience.

# Capacity Scaling

Within the foodida resources may require capacity scaling, either to increase or decrease the compute power and/or storage allocated. The scaling can happen in either vertical (size of the individual server i.e. compute & storage) or horizontal (adding more nodes to the cluster).

Below the possible scaling options and methods are shown for each element of the foodida architecture.

# Disaster Recovery

The disaster recovery plan is part of the business continuity making sure all the services of the foodida are recovered within stated time. This plan exclusively deals with recovering the IT infrastructure from any disaster, and does not deal with what processes the staff has to follow after the recovery of the IT infrastructure, in order to deal with potentially lost data.

However, in the iOS Release phase of this engagement, DR is not being considered.

# Monitoring

All the EC2 instances setup within the foodida have the system monitoring setup.

## RDS MySQL

|  |  |
| --- | --- |
| Metric | Alarm Threshold |
| CPU Utilisation | If >= 100% for 5 mins |
| RAM Utilisation | If >= 100% for 5 mins |
| Disk Utilisation | If >= 75% used |