

Table of Contents

| | |
|--|------------|
| 1 Relevant Confluence Links | 6 |
| 2 CodeCommit Repositories | 7 |
| 3 VPCs MAIN CIDR | 8 |
| 4 EKS Deployment Account..... | 10 |
| 4.1 VPCs MAIN CIDR..... | 10 |
| 5 TRANSIT FRANKFURT - VPCs MAIN CIDR..... | 12 |
| 6 00 VF IE SHARED SERVICES..... | 13 |
| 6.1 00 VF IE TRANSIT GATEWAY | 13 |
| 6.1.1 1.Resource Attachments | 13 |
| 6.1.2 2.Routing..... | 14 |
| 6.1.3 3.Prerequisites | 15 |
| 6.1.4 4.To create a transit gateway | 15 |
| 6.1.5 CONNECTIVITY TEST | 16 |
| 6.1.6 CURRENT DESIGN AND ROUTE TABLES | 28 |
| 6.1.7 TGW IMPLEMENTATION: HOW TO | 36 |
| 6.2 01 VF IE SHARED SERVICES VPC..... | 46 |
| 6.2.1 | 46 |
| 6.2.2 1. INTRODUCTION..... | 46 |
| 6.2.3 2. DESIGN | 46 |
| 6.2.4 3. CONFIGURATION AND IMPLEMENTATION..... | 48 |
| 6.2.5 4. DNS | 48 |
| 6.2.6 5. NLB | 48 |
| 6.2.7 HAProxy AMI | 49 |
| 6.2.8 HAProxy CONFIG AUTOMATION | 56 |
| 6.2.9 HAProxy-EC2 IMPLEMENTATION..... | 69 |
| 6.2.10 HAProxy-ECS IMPLEMENTATION* | 135 |
| 6.2.11 HAProxy TROUBLESHOOTING..... | 149 |
| 6.3 02 VF IE SHARED SERVICES VPC (MYST) | 151 |
| 6.4 03 VF IE CENTRALIZED INTERNET ACCESS VPC | 151 |
| 6.4.1 1. INTRODUCTION..... | 151 |
| 6.4.2 2. CONFIGURATION | 151 |
| 6.4.3 3. VPC CIDR | 152 |
| 6.4.4 4. Intercepting traffic..... | 152 |
| 6.4.5 5. Certificate..... | 153 |
| 6.4.6 6. Squid configuration | 154 |
| 6.4.7 7. White List | 154 |
| 6.4.8 8. Iptables | 154 |
| 6.4.9 RELEASE 1 | 155 |
| 6.4.10 1.CHANGES | 162 |
| 6.4.11 Transparant Proxy Whitelist | 163 |
| 6.5 04 VF IE PORT RANGES | 164 |
| 6.6 05 VF IE NLB DESIGN | 166 |
| 6.6.1 Overview | 167 |
| 6.6.2 Solution Design | 167 |
| 6.6.3 DNS Configuration | 167 |
| 6.6.4 NLB Configuration | 168 |
| 6.6.5 Open Queries..... | 212 |
| 7 01 VF IE SANDBOX VPC..... | 213 |
| 8 02 VF IE TEST VPCs..... | 214 |
| 8.1 1. Introduction | 214 |
| 8.2 2. Design of TEST VPC's | 214 |
| 8.3 00 VF IE SIT4 - HOW TO | 216 |
| 8.3.1 PHASE 1..... | 217 |
| 8.3.2 PHASE 2..... | 222 |

| | | |
|-----------|---|------------|
| 8.3.3 | 01 SIT4 SERVERS | 225 |
| 8.3.4 | 02 SIT4 SECURITY GROUPS | 230 |
| 8.3.5 | 03 EFS Volumes | 261 |
| 8.3.6 | 04 ALB | 264 |
| 8.3.7 | 05 RDS | 271 |
| 9 | 03 VF IE PRE-PROD VPCs..... | 279 |
| 9.1 | 1. Introduction | 279 |
| 9.2 | 2. Design of PRE-PROD VPC's | 279 |
| 9.3 | 3. Subnet CIDR range | 280 |
| 9.3.1 | PRD1 | 280 |
| 9.3.2 | PRD2 | 281 |
| 10 | 04 VF IE DNS | 283 |
| 10.1 | | 283 |
| 10.2 | 1.Route53 Hosted Zones | 283 |
| 10.3 | 2.AWS TO GDC | 286 |
| 10.4 | 3.GDC TO AWS | 287 |
| 10.5 | 4.TGW | 288 |
| 10.6 | 5.SS VPC | 288 |
| 10.7 | 6. AWS Address Resolution | 288 |
| 10.8 | DNS TROUBLESHOOTING | 292 |
| 10.8.1 | POINT 1: GDC domain name resolution | 292 |
| 11 | 04 VF IE PROD VPC | 296 |
| 11.1 | 1. Introduction | 296 |
| 11.2 | 2. Design of PROD VPC | 296 |
| 11.3 | 3. Subnet CIDR range | 297 |
| 12 | 05 VFIE INFRA MONITORING | 299 |
| 12.1 | 1. Logging and Monitoring created by default when an AWS account is created via PCS: | 299 |
| 12.2 | 2. Logging and Monitoring created when PCS receives the demand and manages the account: | 300 |
| 12.3 | 3. Centralized logging / monitoring (https://confluence.sp.vodafone.com/x/FmNhC) | 301 |
| 12.4 | AWS LOGGING | 302 |
| 12.4.1 | VPC Flow Logs | 302 |
| 12.4.2 | Instance Logs - CloudWatch Agent..... | 303 |
| 12.5 | AWS Monitoring: CW alarms | 306 |
| 12.5.1 | EC2 Monitoring: alarms and notifications | 306 |
| 12.5.2 | RDS monitoring: alarms and notifications | 306 |
| 13 | 06 VF IE FIREWALL RULES..... | 307 |
| 13.1 | Production | 307 |
| 13.2 | PRE-PRODUCTION | 318 |
| 14 | 07 VFIE EC2 Image Builder..... | 327 |
| 14.1 | INTRODUCTION | 327 |
| 14.2 | PREREQUISITES | 328 |
| 14.3 | COMPONENTS | 328 |
| 14.3.1 | Image pipeline..... | 328 |
| 14.3.2 | Image recipe | 329 |
| 14.3.3 | Source image..... | 329 |
| 14.3.4 | Build components..... | 329 |
| 14.3.5 | Test components..... | 330 |
| 14.3.6 | Document | 330 |
| 14.4 | IMPLEMENTATION | 330 |
| 14.4.1 | CLOUDFORMATION STACK | 331 |
| 14.4.2 | SSM DOCUMENTS | 334 |
| 14.4.3 | AWS CONSOLE RESOURCES | 335 |
| 14.5 | HOW TO: NEW AMI | 336 |
| 14.5.1 | PERMISSIONS TO NEW AMI | 337 |
| 15 | 08 VFIE OVERALL NETWORK DESIGN..... | 340 |
| 15.1 | Network Diagram | 340 |

| | | |
|-----------|---|------------|
| 15.2 | Advertised Subnets | 340 |
| 15.3 | GDC & VFIE Integration Points | 341 |
| 15.3.1 | TNSNAMES.ORA (PROD)..... | 341 |
| 15.3.2 | GUI URL List..... | 341 |
| 16 | 09 VFIE SQUID PROXY | 343 |
| 17 | 10 VFIE Network Integration..... | 344 |
| 18 | 11 VFIE - Frankfurt Transit Gateway Solution | 349 |
| 18.1 | Overview | 349 |
| 18.2 | Network Design..... | 350 |
| 18.2.1 | SIT3 Design (encompasses PROD)..... | 350 |
| 18.2.2 | Overall Design | 350 |
| 18.2.3 | Network Design..... | 351 |
| 18.3 | AWS Design..... | 351 |
| 18.3.1 | Network Load Balancer..... | 351 |
| 18.3.2 | Target Groups..... | 352 |
| 18.3.3 | Transit Gateway | 352 |
| 18.3.4 | Endpoints | 352 |
| 18.4 | Supporting Documentation | 353 |
| 18.4.1 | DXL / TAAS Roadmap | 353 |
| 18.4.2 | TAAS Documentation..... | 353 |
| 18.4.3 | CAAS NEL mServices..... | 353 |
| 18.5 | 01 - VFIE TAAS Integration | 353 |
| 18.5.1 | Overview | 353 |
| 18.5.2 | Design..... | 353 |
| 18.5.3 | Connectivity | 354 |
| 18.5.4 | Supporting Documentation..... | 354 |
| 18.6 | 02 - VFIE mTAS - CaaS NEL Integration | 354 |
| 18.6.1 | Overview | 355 |
| 18.6.2 | Design..... | 355 |
| 18.6.3 | Connectivity | 357 |
| 18.6.4 | Supporting Documentation..... | 358 |
| 18.7 | 03 - VFIE OSB - CAAS / TAAS Integration..... | 358 |
| 18.7.1 | Overview | 358 |
| 18.7.2 | Design..... | 358 |
| 18.7.3 | TAAS Route 53 | 360 |
| 18.7.4 | Supporting Documentation..... | 360 |
| 18.7.5 | PROJECT DELIVERY RACI/TASK LIST - Flow1 | 361 |
| 18.8 | 04 - VFIE CIAM Webgate - CAAS Integration | 362 |
| 18.8.1 | Overview | 362 |
| 18.8.2 | Design..... | 362 |
| 18.8.3 | AWS Design..... | 363 |
| 18.8.4 | Firewall Connectivity | 365 |
| 18.8.5 | Connectivity Set-up Example | 366 |
| 18.8.6 | CaaS Ticket | 366 |
| 18.8.7 | Supporting Documentation..... | 366 |
| 18.9 | 05 - VFIE WEF SSR - TaaS Integration | 366 |
| 18.9.1 | Overview | 366 |
| 18.9.2 | Design..... | 366 |
| 18.9.3 | AWS Design..... | 367 |
| 19 | 12 VFIE NAT Gateway | 370 |
| 19.1 | Overview | 370 |
| 19.2 | High Level Design | 371 |
| 19.3 | Subnets | 371 |
| 19.3.1 | AWS Shared Services Subnets | 371 |
| 19.3.2 | Destination Subnets | 372 |
| 19.4 | NAT Gateway | 373 |
| 19.5 | Routing Tables | 373 |
| 19.6 | High-Level Implementation Steps | 374 |

- [00 VF IE SHARED SERVICES](#)
 - [00 VF IE TRANSIT GATEWAY](#)
 - [CONNECTIVITY TEST](#)
 - [CURRENT DESIGN AND ROUTE TABLES](#)
 - [TGW IMPLEMENTATION: HOW TO](#)
 - [01 VF IE SHARED SERVICES VPC](#)
 - [HAPROXY AMI](#)
 - [HAPROXY CONFIG AUTOMATION](#)
 - [HAPROXY-EC2 IMPLEMENTATION](#)
 - [HAPROXY-ECS IMPLEMENTATION*](#)
 - [HAPROXY TROUBLESHOOTING](#)
 - [02 VF IE SHARED SERVICES VPC \(MYST\)](#)
 - [03 VF IE CENTRALIZED INTERNET ACCESS VPC](#)
 - [RELEASE 1](#)
 - [RELEASE 2](#)
 - [Transparant Proxy Whitelist](#)
 - [04 VF IE PORT RANGES](#)
 - [05 VF IE NLB DESIGN](#)
- [01 VF IE SANDBOX VPC](#)
- [02 VF IE TEST VPCs](#)
 - [00 VF IE SIT4 - HOW TO](#)
 - [01 SIT4 SERVERS](#)
 - [02 SIT4 SECURITY GROUPS](#)
 - [03 EFS Volumes](#)
 - [04 ALB](#)
 - [05 RDS](#)
- [03 VF IE PRE-PROD VPCs](#)
- [04 VF IE DNS](#)
 - [DNS TROUBLESHOOTING](#)
- [04 VF IE PROD VPC](#)
- [05 VFIE INFRA MONITORING](#)
 - [AWS LOGGING](#)
 - [AWS Monitoring: CW alarms](#)
 - [EC2 Monitoring: alarms and notifications](#)
 - [RDS monitoring: alarms and notifications](#)
- [06 VF IE FIREWALL RULES](#)
- [07 VFIE EC2 Image Builder](#)
- [08 VFIE OVERALL NETWORK DESIGN](#)
- [09 VFIE SQUID PROXY](#)

- [10 VFIE Network Integration](#)
- [11 VFIE - Frankfurt Transit Gateway Solution](#)
 - [01 - VFIE TAAS Integration](#)
 - [02 - VFIE mTAS - CaaS NEL Integration](#)
 - [03 - VFIE OSB - CAAS / TAAS Integration](#)
 - [04 - VFIE CIAM Webgate - CAAS Integration](#)
 - [05 - VFIE WEF SSR - TaaS Integration](#)
- [12 VFIE NAT Gateway](#)

1 Relevant Confluence Links

- Haproxy - <https://confluence.sp.vodafone.com/x/qNxOCg>
- DNS strategy - <https://confluence.sp.vodafone.com/x/1HUICg>
- Routing in Transit Gateway - <https://confluence.sp.vodafone.com/x/idIOCg>
- How to update configuration of Transit Gateway - <https://confluence.sp.vodafone.com/x/6RsCCg>
- Shared Service VPC design - <https://confluence.sp.vodafone.com/x/73MCCg>
- AWS Image Builder - <https://confluence.sp.vodafone.com/x/79NoCg>

2 CodeCommit Repositories

- Centralized internet - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-centralized-internet/browse?region=eu-west-1>
- Shared Services VPC - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-sharedservices-vpc/browse?region=eu-west-1>
- AWS Infrastructure (a branch for each environment) - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-infrastructure/browse?region=eu-west-1>
- Cloud9 Environment - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-cloud9/browse?region=eu-west-1>
- Myst VPC - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-myst/browse?region=eu-west-1>
- CF Stacksets (monitoring, iam roles, etc.) - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-stacksets/browse?region=eu-west-1>
- DNS - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-dns/browse?region=eu-west-1>
- Transit GW - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-transit-gateway/browse?region=eu-west-1>
- Pipelines for AWS - <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-pipelines/browse?region=eu-west-1>

3 VPCs MAIN CIDR

Context:

- Range ./19 or ./23 per VPC
- Myst VPC : ./24
- Test to talk with each other, nothing to talk with production
- Stx -> separate oracle database

Number of VPC's:

- Sandbox: TRN1, TRN2
- Test: ST1, ST2, ST3, ST4
- Pre-prod: PRD1, PRD2
- Prop: PROD
- Shared Service: Myst, SS

If we split 10.181.0.0/16 in ./19 -> 8 different ranges:

- 10.181.0.0/19
- 10.181.32.0/19
- 10.181.64.0/19
- 10.181.96.0/19
- 10.181.128.0/19
- 10.181.160.0/19
- 10.181.192.0/19
- 10.181.224.0/19

| VPC | Environment | Account Name | Account ID | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|----------|-------------|----------------------------|--------------|------------------------------------|---------------|----------------------------|----------------------------|-------|
| TRN2-PoC | Sandbox | vf-iedelivery-sandbox-01 | 299879056526 | 10.181.0.0/20 | 255.255.240.0 | 10.181.0.0 - 10.181.15.255 | 10.181.0.1 - 10.181.15.254 | 4094 |
| Myst | Prod-SS | vf-iedelivery-produc-ss-01 | 267040142128 | 10.181.16.0/23 | 255.255.254.0 | 10.181.16.0 - 10.181.17.55 | 10.181.16.1 - 10.181.17.54 | 510 |
| SS | Prod-SS | vf-iedelivery-produc-ss-01 | 267040142128 | 198.19.220.0/23 46.108.156.0/27 | | | | |

| | | | | | | | | |
|------------------------------------|------------|--------------------------|--------------|------------------------|-----------------|-------------------------------|-------------------------------|------|
| TRN1 | Sandbox | vf-iedelivery-sandbox-01 | 299879056526 | 10.181.20.0/23 | 255.255.254.0 | 10.181.20.0 - 10.181.21.255 | 10.181.20.1 - 10.181.21.254 | 510 |
| Centralized Internet Access | Prod-SS | vf-iedelivery-prod-ss-01 | 267040142128 | 10.181.22.0/23 | 255.255.254.0 | 10.181.22.0 - 10.181.23.255 | 10.181.22.1 - 10.181.23.254 | 510 |
| Network Firewall VPC | Prod-SS | vf-iedelivery-prod-ss-01 | 267040142128 | 10.181.25.128/25 | 255.255.255.128 | 10.181.25.128 - 10.181.25.255 | 10.181.25.129 - 10.181.25.254 | 126 |
| Dev | Dev | vf-iedelivery-dev-01 | 17361906891 | 10.181.30.0/23 | 255.255.254.0 | 10.181.30.0 - 10.181.31.255 | 10.181.30.1 - 10.181.31.254 | 510 |
| PROD | Production | vf-iedelivery-prod-01 | 323874256692 | 10.181.32.0/19 | 255.255.224.0 | 10.181.32.0 - 10.181.63.255 | 10.181.32.1 - 10.181.63.254 | 8190 |
| PRD1 | Pre-Prod | vf-iedelivery-preprod-01 | 046978237480 | 10.181.64.0/19 | 255.255.224.0 | 10.181.64.0 - 10.181.95.255 | 10.181.64.1 - 10.181.95.254 | 8190 |
| PRD2 | Pre-Prod | vf-iedelivery-preprod-01 | 046978237480 | 10.181.96.0/19 | 255.255.224.0 | 10.181.96.0 - 10.181.127.255 | 10.181.96.1 - 10.181.127.254 | 8190 |
| SIT1 | Test | vf-iedelivery-test-01 | 046978237480 | 10.181.128.0/19 | 255.255.224.0 | 10.181.128.0 - 10.181.159.255 | 10.181.128.1 - 10.181.159.254 | 8190 |
| SIT2 | Test | vf-iedelivery-test-01 | 046978237480 | 10.181.160.0/19 | 255.255.224.0 | 10.181.160.0 - 10.181.191.255 | 10.181.160.1 - 10.181.191.254 | 8190 |
| SIT3 | Test | vf-iedelivery-test-01 | 046978237480 | 10.181.192.0/19 | 255.255.224.0 | 10.181.192.0 - 10.181.223.255 | 10.181.192.1 - 10.181.223.254 | 8190 |
| SIT4 | Test | vf-iedelivery-test-01 | 046978237480 | 10.181.224.0/19 | 255.255.224.0 | 10.181.224.0 - 10.181.255.255 | 10.181.224.1 - 10.181.255.254 | 8190 |

4 EKS Deployment Account

4.1 VPCs MAIN CIDR

Context:

- Range ./19 or ./23 per VPC
- Test to talk with each other, nothing to talk with production

Number of VPC's:

- Test: ST1, ST2, ST3, ST4
- Pre-prod: PRD1, PRD2
- Prop: PROD

If we split 10.182.0.0/16 in ./19 -> 8 different ranges:

- 10.182.0.0/19
- 10.182.32.0/19
- 10.182.64.0/19
- 10.182.96.0/19
- 10.182.128.0/19
- 10.182.160.0/19
- 10.182.192.0/19
- 10.182.224.0/19

| VPC | Environment | Account Name | Account ID | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|------|-------------|----------------------------|--------------|-----------------------|---------------|------------------------------|------------------------------|-------|
| PROD | Production | vf-iedelivery-eks-prod | 466441119255 | 10.182.0.0/19 | 255.255.224.0 | 10.182.0.0 - 10.182.31.255 | 10.182.0.1 - 10.182.31.254 | 8190 |
| PRD1 | Pre-Prod | vf-iedelivery-eks-preprod | 387095844067 | 10.182.32.0/19 | 255.255.224.0 | 10.182.32.0 - 10.182.63.255 | 10.182.32.1 - 10.182.63.254 | 8190 |
| PRD2 | Pre-Prod | vf-iedelivery-eks-preprod2 | 762937892420 | 10.182.64.0/19 | 255.255.224.0 | 10.182.64.0 - 10.182.95.255 | 10.182.64.1 - 10.182.95.254 | 8190 |
| SIT1 | Test | vf-iedelivery-eks-test-01 | 687158677620 | 10.182.96.0/19 | 255.255.224.0 | 10.182.96.0 - 10.182.127.255 | 10.182.96.1 - 10.182.127.254 | 8190 |

| | | | | | | | | |
|-------------|------|---------------------------------------|------------------|-----------------------------|-------------------|--|--|----------|
| SIT2 | Test | vf- iedelive ry-eks- test-02 | 617458290 184 | 10.182.128. 0/19 | 255.255.22 4.0 | 10.182.128. 0 - 10.182.159. 255 | 10.182.128. 1 - 10.182.159. 254 | 819 0 |
| SIT3 | Test | vf- iedelive ry-eks- test-03 | 483246254 322 | 10.182.160. 0/19 | 255.255.22 4.0 | 10.182.160. 0 - 10.182.191. 255 | 10.182.160. 1 - 10.182.191. 254 | 819 0 |
| SIT4 | Test | vf- iedelive ry-eks- test-04 | 701287184 434 | 10.182.192. 0/19 | 255.255.22 4.0 | 10.182.192. 0 - 10.182.223. 255 | 10.182.192. 1 - 10.182.223. 254 | 819 0 |
| DEV 1 | DEV | vf- iedelive ry-eks- dev-01 | 607188850 947 | 10.182.224. 0/19 | 255.255.22 4.0 | 10.182.224. 0 - 10.182.255. 255 | 10.182.224. 1 - 10.182.255. 254 | 819 0 |

5 TRANSIT FRANKFURT - VPCs MAIN CIDR

10.190.0.0/22

| VPC | Environment | Account Name | Account ID | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|------|---------------------|---------------------------|--------------|----------------|---------|---------------------------|---------------------------|-------|
| PROD | Prod-SS (frankfurt) | vf-iedeliver-y-prod-ss-01 | 267040142128 | 10.190.0/22 | | 10.190.0.0 - 10.190.3.255 | 10.190.0.1 - 10.190.3.254 | 1022 |

6 00 VF IE SHARED SERVICES

- [00 VF IE TRANSIT GATEWAY](#)
- [01 VF IE SHARED SERVICES VPC](#)
- [02 VF IE SHARED SERVICES VPC \(MYST\)](#)
- [03 VF IE CENTRALIZED INTERNET ACCESS VPC](#)
- [04 VF IE PORT RANGES](#)
- [05 VF IE NLB DESIGN](#)

6.1 00 VF IE TRANSIT GATEWAY

INTRODUCTION

A *transit gateway* is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks. It acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPC) and VPN connections. A transit gateway scales elastically based on the volume of network traffic. Routing through a transit gateway operates at layer 3, where the packets are sent to a specific next-hop attachment, based on their destination IP addresses.

The following are the key concepts for transit gateways:

- **attachment** — You can attach a VPC, an AWS Direct Connect gateway, a peering connection with another transit gateway, or a VPN connection to a transit gateway.
- **transit gateway route table** — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be a VPC or a VPN connection. By default, transit gateway attachments are associated with the default transit gateway route table.
- **associations** — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.
- **route propagation** — A VPC or VPN connection can dynamically propagate routes to a transit gateway route table. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP). With a peering attachment, you must create a static route in the transit gateway route table to point to the peering attachment.

6.1.1 1.Resource Attachments

A transit gateway attachment is both a source and a destination of packets. You can attach the following resources to your transit gateway:

- One or more VPCs
- One or more VPN connections
- One or more AWS Direct Connect gateways
- One or more transit gateway peering connections

If you attach a transit gateway peering connection, the transit gateway must be in a different Region.

When you attach a VPC to a transit gateway, you must enable one or more Availability Zones to be used by the transit gateway to route traffic to resources in the VPC subnets. To enable each Availability Zone, you specify exactly one subnet. The transit gateway places a network interface in that subnet using one IP address from the subnet. After you enable an Availability Zone, traffic can be routed to all subnets in that Availability Zone, not just the specified subnet. Resources that reside in Availability Zones where there is no transit gateway attachment will not be able to reach the transit gateway.

We recommend that you enable multiple Availability Zones to ensure availability.

6.1.2 2.Routing

Your transit gateway routes IPv4 and IPv6 packets between attachments using transit gateway route tables. You can configure these route tables to propagate routes from the route tables for the attached VPCs and VPN connections. You can also add static routes to the transit gateway route tables. When a packet comes from one attachment, it is routed to another attachment using the route table that matches the destination IP address.

For transit gateway peering attachments, only static routes are supported.

6.1.2.1 Route Tables

Your transit gateway automatically comes with a default route table. By default, this route table is the default association route table and the default propagation route table. Alternatively, if you disable route propagation and route table association, we do not create a default route table for the transit gateway.

You can create additional route tables for your transit gateway. This enables you to isolate subnets of attachments. Each attachment can be associated with one route table. An attachment can propagate their routes to one or more route tables.

You can create a blackhole route in your transit gateway route table that drops traffic that matches the route.

When you attach a VPC to a transit gateway, you must add a route to your subnet route table for traffic to route through the transit gateway.

6.1.2.2 Route Table Association

You can associate a transit gateway attachment with a single route table. Each route table can be associated with zero to many attachments and forward packets to other attachments.

6.1.2.3 Route Propagation

Each attachment comes with routes that can be installed to one or more transit gateway route tables. When an attachment is propagated to a transit gateway route table, these routes are installed in the route table.

For a VPC attachment, the CIDR blocks of the VPC are propagated to the transit gateway route table.

For a VPN connection attachment, routes in the transit gateway route table propagate to and from the transit gateway and your on-premises router using Border Gateway Protocol (BGP). The prefixes that are advertised over the BGP session are propagated to the transit gateway route table.

6.1.2.4 Routes for Peering Attachments

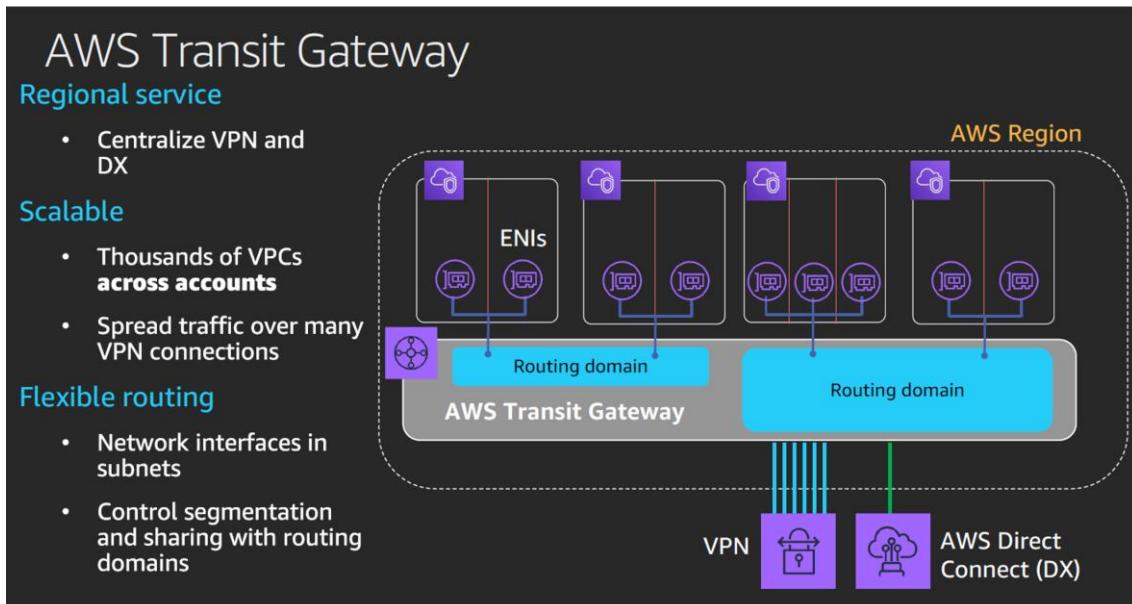
You can peer two transit gateways and route traffic between them. To do this, you create a peering attachment on your transit gateway, and specify the peer transit gateway with which to create the peering connection. You then create a static route in your transit gateway route table to route traffic to the transit gateway peering attachment. Traffic that's routed to the peer transit gateway can then be routed to the VPC and VPN attachments for the peer transit gateway.

6.1.3 3.Prerequisites

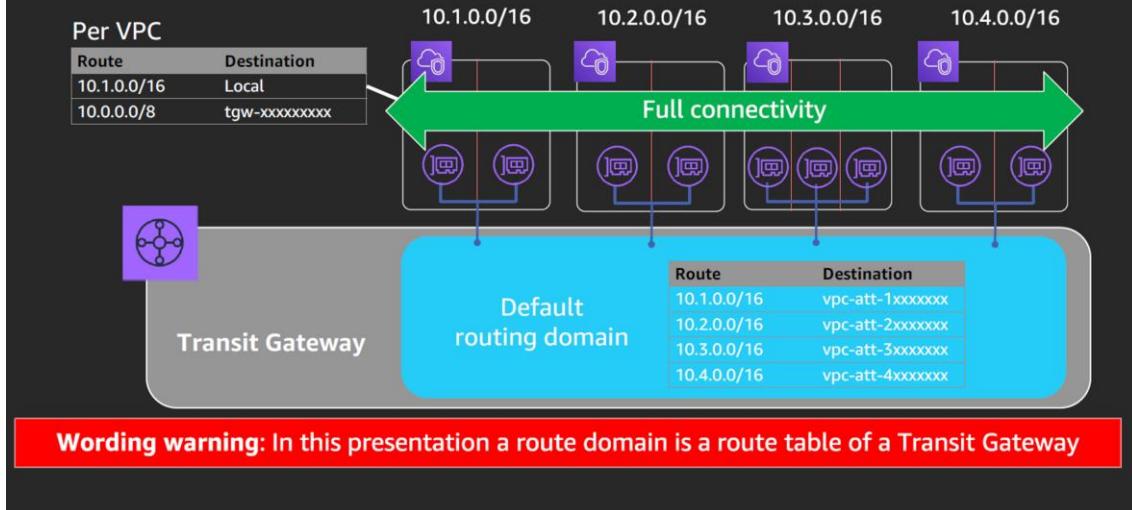
- The VPCs cannot have overlapping CIDRs. Launch one EC2 instance in each VPC.
- You must enable resource sharing from the master account for your organization.
- You cannot have identical routes pointing to two different VPCs. A transit gateway does not propagate the CIDRs of a newly attached VPC if an identical route exists in the transit gateway route tables.
- Verify that you have the permissions required to work with transit gateways.

6.1.4 4.To create a transit gateway

- For Amazon side ASN, type the private Autonomous System Number (ASN) for your transit gateway. This should be the ASN for the AWS side of a Border Gateway Protocol (BGP) session. The range is 64512 to 65534 for 16-bit ASNs. The range is 4200000000 to 4294967294 for 32-bit ASNs.
- For DNS support, choose enable if you need the VPC to resolve public IPv4 DNS host names to private IPv4 addresses when queried from instances in another VPC attached to the transit gateway.
- For VPN ECMP support, choose enable if you need Equal Cost Multipath (ECMP) routing support between VPN connections. If connections advertise the same CIDRs, the traffic is distributed equally between them. When you select this option, the advertised BGP ASN, the BGP attributes such as the AS-path and the communities for preference must be the same.
- For Default route table association, choose enable to automatically associate transit gateway attachments with the default route table for the transit gateway.
- For Default route table propagation, choose enable to automatically propagate transit gateway attachments to the default route table for the transit gateway.
- For Auto accept shared attachments, choose enable to automatically accept cross-account attachments.



Flat: Transit Gateway route domains (route tables)



6.1.5 CONNECTIVITY TEST

- [1.TRANSIT GATEWAY CONFIGURATION \(TEST-SS connectivity, release 1\)](#)
- [2. SHARED SERVICE VPC CONFIGURATION \(TEST-SS connectivity, release 1\)](#)
- [3. TEST VPC CONFIGURATION](#)
- [4. RESULTS TEST-SS CONNECTIVITY- PHASE 1](#)
 - [4.1 ST1 to SS](#)
 - [Solution: Add route to TGW in all subnets](#)
 - [4.2 SOLUTION: ROUTE TO TGW - PHASE 1](#)
- [5. RESULTS TEST-SS CONNECTIVITY - PHASE 2](#)
 - [5.1 ST1 to SS and to ST2](#)
- [6.TRANSIT GATEWAY CONFIGURATION \(release 2\)](#)
 - [6.1 TGW VPC attachments](#)
 - [6.2 TGW Route tables](#)
- [7. CONNECTIVITY TEST \(TEST-PREPROD-PROD-SS\)](#)
 - [7.1 Prod web ec2 \(10.181.39.200\)](#)
 - [7.2 Pre-Prod web ec2 \(10.181.71.39\)](#)
 - [7.3 SS private EC2 \(10.181.18.126\)](#)

This page will show the tests executed to check the connectivity between vpc's in different aws accounts through the transit gateway. For that purpose, the following resources will be created in the different vpc's:

- EC2 instances in different subnets (with / without TGW ENI) of the specific vpc's.
- Security groups to allow the communication between the vpc's (inbound/outbound rules to allow traffic from/to specific cidr).

- SSM endpoints and SSM security groups to connect to EC2 instances.

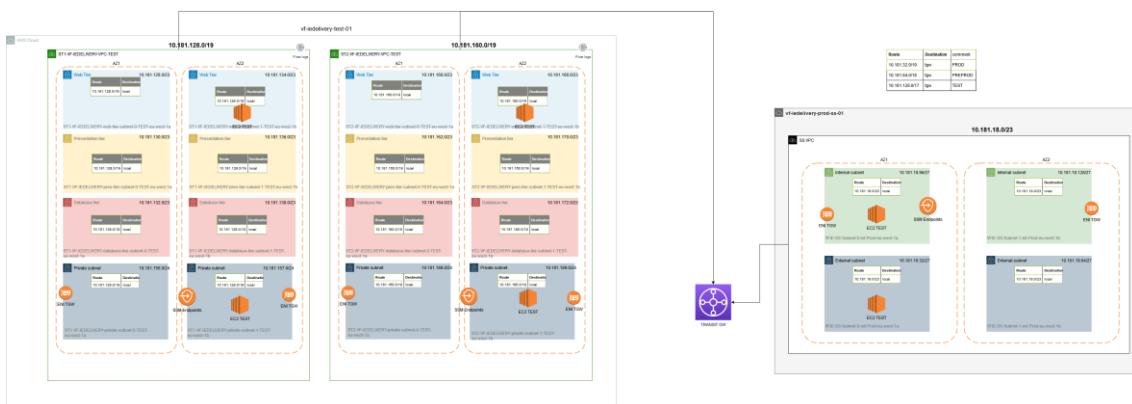
ICMP ping messages are used as the method used to confirm that EC2 instances in different VPCs can communicate between them.

These are the scenarios that have been analysed:

- Connectivity between TEST environment and Shared Service VPC. (Release 1)
- Connectivity between TEST environments. (Release 1)
- Connectivity between PRE-PROD and SS VPC. (Release 2)
- Connectivity between PROD and SS VPC. (Release 2)

Sections 1 to 5 were done before release 2, meaning that pre-prod and prod vpc's were not associated with the TGW.

The diagram below shows the EC2 instances that have been deployed for TEST and SS VPC. The same logic has been followed for PRE-PROD and PROD, but it is not shown for simplicity reasons.



6.1.5.1 1.TRANSIT GATEWAY CONFIGURATION (TEST-SS connectivity, release 1)

This is the transit gateway created in SS Account:

The screenshot shows the AWS CloudFormation console with the 'Create Transit Gateway' button and an 'Actions' dropdown. Below is a search bar and a table listing the transit gateway details:

| Name | Transit Gateway ID | Owner ID | State |
|------------------|-----------------------|--------------|-----------|
| VFIE-Transit-... | tgw-088852b5024504fd8 | 267040142128 | available |

Below the table, the 'Transit Gateway: tgw-088852b5024504fd8' section shows the 'Details' tab selected. It displays the following configuration parameters:

| | | | |
|--------------------------------|-----------------------|---------------------------------|--------------|
| Transit Gateway ID | tgw-088852b5024504fd8 | Owner account ID | 267040142128 |
| State | available | Amazon ASN | 64512 |
| DNS support | enable | VPN ECMP support | enable |
| Auto accept shared attachments | enable | Default association route table | disable |
| Association route table ID | - | Default propagation route table | disable |
| Propagation route table ID | - | | |

These are the attachments that have been deployed. Right now, the transit gateway allows connectivity between TEST VPC's and SS VPC:

VF IE Application Migration (AWS) – 03 VF-IE Network Design

Create Transit Gateway Attachment Actions ▾

Filter by tags and attributes or search by keyword

| Name | Transit Gateway attachment ID | Transit Gateway ID | Resource type | Resource ID |
|---|-------------------------------------|------------------------------|---------------|------------------------------|
| VFIE-Transit-GW-test-vpc-attachment-vpc-st2 | tgw-attach-0117c5343ff5b6e47 | tgw-088852b5024504fd8 | VPC | vpc-0270246dc82c513fc |
| VFIE-Transit-GW-test-vpc-attachment-vpc-st4 | tgw-attach-012868b212ffbcd4f | tgw-088852b5024504fd8 | VPC | vpc-070981e1316c4795f |
| VFIE-Transit-GW-test-vpc-attachment-vpc-st3 | tgw-attach-020480d4ce81d50db | tgw-088852b5024504fd8 | VPC | vpc-07bf6d07ae85e68ee |
| VFIE-Transit-GW-SS-vpc-attachment-vpc-0bc5e297e4fd20219 | tgw-attach-059eed3036acc6991 | tgw-088852b5024504fd8 | VPC | vpc-0bc5e297e4fd20219 |
| VFIE-Transit-GW-test-vpc-attachment-vpc-st1 | tgw-attach-0969c8e937cd6b7d8 | tgw-088852b5024504fd8 | VPC | vpc-01a98bfaa291c8f7d |

Transit Gateway Attachment: tgw-attach-0969c8e937cd6b7d8

Details Tags

| | | | |
|-------------------------------|--|---------------------------|---------------------------|
| Transit Gateway attachment ID | tgw-attach-0969c8e937cd6b7d8 | Transit Gateway owner ID | 267040142128 |
| Transit Gateway ID | tgw-088852b5024504fd8 | Resource owner account ID | 046978237480 (shared) |
| Resource type | VPC | State | available |
| Resource ID | vpc-01a98bfaa291c8f7d | Associated route table | tgw-rtb-0fd38027ab61b15a3 |
| Association state | associated | DNS support | enable |
| Subnet IDs | subnet-03951db5996dd1e023 subnet-00bd23fbcf0348c0 | IPv6 support | disable |

For the SS VPC, a TGW route table have been created with the following configuration:

Create Transit Gateway Route Table Actions ▾

Filter by tags and attributes or search by keyword

| Name | Transit Gateway route table ID | Transit Gateway ID | State | Default association route table | Default propagation route tab |
|---|----------------------------------|------------------------------|------------------|---------------------------------|-------------------------------|
| VFIE-Transit-GW-SharedServices-Route-Table-vpc | tgw-rtb-0a2e0ab02eced2f6f | tgw-088852b5024504fd8 | available | No | No |
| VFIE-Transit-GW-TEST-Route-Table-vpc | tgw-rtb-0fd38027ab61b15a3 | tgw-088852b5024504fd8 | available | No | No |

Transit Gateway Route Table: tgw-rtb-0a2e0ab02eced2f6f

Details Associations Propagations Routes Tags

The table below will return a maximum of 1000 routes. Narrow the filter or use export routes to view more routes.

Create route Replace route Delete route

Filter by attributes or search by keyword

| CIDR | Attachment | Resource type | Route type | Route state |
|-----------------|--|---------------|------------|-------------|
| 10.181.128.0/19 | tgw-attach-0969c8e937cd6b7d8 vpc-01a98bfaa291c8f7d | VPC | static | active |
| 10.181.160.0/19 | tgw-attach-0117c5343ff5b6e47 vpc-0270246dc82c513fc | VPC | static | active |
| 10.181.18.0/23 | tgw-attach-059eed3036acc6991 vpc-0bc5e297e4fd20219 | VPC | propagated | active |
| 10.181.192.0/19 | tgw-attach-020480d4ce81d50db vpc-07bf6d07ae85e68ee | VPC | static | active |
| 10.181.224.0/19 | tgw-attach-012868b212ffbcd4f vpc-070981e1316c4795f | VPC | static | active |

For all TEST VPC's, a TGW route table have been created with the following configuration:

Create Transit Gateway Route Table Actions ▾

Filter by tags and attributes or search by keyword

| Name | Transit Gateway route table ID | Transit Gateway ID | State | Default association route table | Default propagation route tab |
|--|----------------------------------|------------------------------|------------------|---------------------------------|-------------------------------|
| VFIE-Transit-GW-SharedServices-Route-Table-vpc | tgw-rtb-0a2e0ab02eced2f6f | tgw-088852b5024504fd8 | available | No | No |
| VFIE-Transit-GW-TEST-Route-Table-vpc | tgw-rtb-0fd38027ab61b15a3 | tgw-088852b5024504fd8 | available | No | No |

Transit Gateway Route Table: tgw-rtb-0fd38027ab61b15a3

Details Associations Propagations Routes Tags

The table below will return a maximum of 1000 routes. Narrow the filter or use export routes to view more routes.

Create route Replace route Delete route

Filter by attributes or search by keyword

| CIDR | Attachment | Resource type | Route type | Route state |
|-----------------|--|---------------|------------|-------------|
| 0.0.0.0/0 | tgw-attach-059eed3036acc6991 vpc-0bc5e297e4fd20219 | VPC | static | active |
| 10.181.128.0/19 | tgw-attach-0969c8e937cd6b7d8 vpc-01a98bfaa291c8f7d | VPC | propagated | active |
| 10.181.160.0/19 | tgw-attach-0117c5343ff5b6e47 vpc-0270246dc82c513fc | VPC | propagated | active |
| 10.181.192.0/19 | tgw-attach-020480d4ce81d50db vpc-07bf6d07ae85e68ee | VPC | propagated | active |
| 10.181.224.0/19 | tgw-attach-012868b212ffbcd4f vpc-070981e1316c4795f | VPC | propagated | active |
| 10.181.32.0/19 | - | - | static | blackhole |
| 10.181.64.0/19 | - | - | static | blackhole |
| 10.181.96.0/19 | - | - | static | blackhole |

6.1.5.2 2. SHARED SERVICE VPC CONFIGURATION (TEST-SS connectivity, release 1)

In the SS VPC, 2 EC2 instances have been created, in different subnets. According to AWS documentation, to provide connectivity to a AZ in a VPC just one subnet needs to be defined. Also, the route table of the subnet attached with the Transit GW (VFIE-SS-Subnet-0-int-Prod-eu-west-1a) were modified when the TGW deployment happened, adding a new route to the transit GW. This test will provide more clarity if all route tables on a VPC attached to a TGW need this modification, or not. In order to do that, one EC2 instance has been created in the subnet with the TGW ENI (VFIE-SS-Subnet-0-int-Prod-eu-west-1a) and another EC2 instance has been deployed in a different one (VFIE-SS-Subnet-0-ext-Prod-eu-west-1a).

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Publ. | IPv4 Public IP | IP |
|--|----------------------|---------------|-------------------|----------------|----------------|--------------|-------|----------------|----|
| TGW-connectivity-SS-int-vfie-delivery-EC2-Prod | i-0197b80a619584418 | t2.micro | eu-west-1a | running | 2/2 checks ... | None | - | - | - |
| TGW-connectivity-SS-ext-vfie-delivery-EC2-Prod | i-05e85ad10e29ad5... | t2.micro | eu-west-1a | running | 2/2 checks ... | None | - | - | - |

Instance: i-0197b80a619584418 (TGW-connectivity-SS-int-vfie-delivery-EC2-Prod) Private IP: 10.181.18.126

| Description | Status Checks | Monitoring | Tags |
|--|---|------------|------|
| Instance ID: i-0197b80a619584418 | Public DNS (IPv4): - | | |
| Instance state: running | IPv4 Public IP: - | | |
| Instance type: t2.micro | IPv6 IPs: - | | |
| Finding: You may not have permission to access AWS Compute Optimizer. | Elastic IPs: - | | |
| Private DNS: ip-10-181-18-126.eu-west-1.compute.internal | Availability zone: eu-west-1a | | |
| Private IPs: 10.181.18.126 | Security groups: SSM-endpoint-SS-vfie-delivery-sec-group-Prod, SS-TO-TEST-vfie-delivery-sec-group-PROD, view inbound rules, view outbound rules | | |
| Secondary private IPs | Scheduled events: No scheduled events | | |
| VPC ID: vpc-0bc5e297e4fd20219 (VFIE-SS-VPC) | AMI ID: vf-gdc-amzn2-hvm-2020-03-12T13-28-54Z-x86_64-gp2 (ami-03f3ecb7248bd2548) | | |
| Subnet ID: subnet-0c77421364cb58f2d (VFIE-SS-Subnet-0-int-Prod-eu-west-1a) | Platform details: - | | |

A security group called "SS-TO-TEST-vfie-delivery-sec-group-PROD" has been created to allow traffic to test environment (cidr = 10.181.128.0/17):

```

ingress {
  description = "All traffic from TEST"
  from_port = 0
  to_port = 0
  protocol = -1
  cidr_blocks = ["10.181.128.0/17"]
}

egress {
  description = "All traffic to TEST"
  from_port = 0
  to_port = 0
  protocol = -1
  cidr_blocks = ["10.181.128.0/17"]
}

```

6.1.5.3 3. TEST VPC CONFIGURATION

In the ST1 VPC, 2 EC2 instances have been created, in different subnets. According to AWS documentation, to provide connectivity to a AZ in a VPC just one subnet needs to be defined. Also, the route table of the subnet attached with the Transit GW (ST1-VF-IEDELIVERY-private-subnet-3-TEST-eu-west-1b) were modified when the TGW deployment happened, adding a new route to the transit GW. This test will provide more clarity if all route tables on a VPC attached to a TGW need this modification, or not. In order to do that, one EC2 instance has been created in the subnet with the TGW ENI (ST1-VF-IEDELIVERY-private-subnet-3-TEST-eu-west-1b) and another EC2 instance has been deployed in a different one (ST1-VF-IEDELIVERY-web-tier-subnet-1-TEST-eu-west-1b).

In the ST2 VPC, 2 EC2 instances have been created, one in the subnet with the TGW ENI (ST2-VF-IEDELIVERY-private-subnet-3-TEST-eu-west-1b) and another EC2 instance has been deployed in a different one (ST2-VF-IEDELIVERY-web-tier-subnet-1-TEST-eu-west-1b).

The screenshot shows the AWS CloudWatch Metrics console. At the top, there is a table listing four EC2 instances:

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Public DNS (IPv4) |
|--|----------------------------|-----------------|-------------------|----------------|-----------------------|--------------|-------------------|
| TGW-connectivity-ST1-2-vfie-delivery-EC2-TEST | i-00881f46d805f07b1 | t2.micro | eu-west-1b | running | 2/2 checks ... | None | - |
| TGW-connectivity-ST2-2-vfie-delivery-EC2-TEST | i-0614998985db1040c | t2.micro | eu-west-1b | running | 2/2 checks ... | None | - |
| TGW-connectivity-ST2-vfie-delivery-EC2-TEST | i-0964d9f608adbfaf18 | t2.micro | eu-west-1b | running | 2/2 checks ... | None | - |
| TGW-connectivity-ST1-vfie-delivery-EC2-TEST | i-0d9a572c40079c75f | t2.micro | eu-west-1b | running | 2/2 checks ... | None | - |

Below the table, a detailed view of the selected instance (i-0d9a572c40079c75f) is shown. The instance details include:

- Description:** TGW-connectivity-ST1-vfie-delivery-EC2-TEST
- Private IP:** 10.181.157.159
- Instance ID:** i-0d9a572c40079c75f
- Instance state:** running
- Instance type:** t2.micro
- Finding:** You may not have permission to access AWS Compute Optimizer.
- Private DNS:** ip-10-181-157-159.eu-west-1.compute.internal
- Private IPs:** 10.181.157.159
- Secondary private IPs:** -
- VPC ID:** vpc-01a98bfa291c8f7d (ST1-VF-IEDELIVERY-VPC-TEST)
- Subnet ID:** subnet-03951dh996dd1e623 (ST1-VF-IEDELIVERY-private-subnet-3-TEST-eu-west-1b)
- Public DNS (IPv4):** -
- IPv4 Public IP:** -
- IPv6 IPs:** -
- Elastic IPs:** -
- Availability zone:** eu-west-1b
- Security groups:** SSM-endpoint-ST1-VF-IEDELIVERY-sec-group-TEST, ST1-TO-SS-vfie-delivery-sec-group-TEST, ST1-TO-TEST-vfie-delivery-sec-group-TEST, view inbound rules, view outbound rules
- Scheduled events:** No scheduled events
- AMI ID:** vf-gdc-amzn2-hvm-2020-03-12T13-28-54Z.x86_64-gp2 (ami-033ecd7248bd2548)
- Platform details:** -

In both ST1 and ST2, the following security groups have been created: 1) security group to allow SSM traffic, 2) security group to allow traffic from/to SS account (10.181.18.0/23) and 3) security group to allow traffic to/from TEST environments (10.181.128.0/17).

Example for ST1:

- ST1-TO-TEST-vfie-delivery-sec-group-TEST:
 - ingress { description = "All traffic from TEST" from_port = 0 to_port = 0 protocol = -1 cidr_blocks = ["10.181.128.0/17"] }
 - egress { description = "All traffic to TEST" from_port = 0 to_port = 0 protocol = -1 cidr_blocks = ["10.181.128.0/17"] }
- ST1-TO-SS-vfie-delivery-sec-group-TEST:
 - ingress { description = "All traffic from SS VPC" from_port = 0 to_port = 0 protocol = -1 cidr_blocks = ["10.181.18.0/23"] }
 - egress { description = "All traffic to SS VPC" from_port = 0 to_port = 0 protocol = -1 cidr_blocks = ["10.181.18.0/23"] }

6.1.5.4 4. RESULTS TEST-SS CONNECTIVITY- PHASE 1

- **Test st1:**
- EC2 in Subnet with TGW attach. **Private IP: 10.181.157.159**
- EC2 in Subnet without TGW attach. **Private IP: 10.181.135.130**
- **Test st2:**
- EC2 in Subnet with TGW attach. **Private IP: 10.181.189.84**
- EC2 in Subnet without TGW attach. **Private IP: 10.181.169.179**
- **SS:**
- EC2 in Subnet with TGW attach. **Private IP: address: 10.181.18.126**
- EC2 in Subnet without TGW attach. **Private IP: 10.181.18.52**

| CONNECTIVITY | SS {with TGW ENI} | SS {without TGW ENI} | ST1 {with TGW ENI} | ST1 {without TGW ENI} |
|-----------------------|-------------------|----------------------|--------------------|-----------------------|
| ST1 {with TGW ENI} | YES | NO | | |
| ST1 {without TGW ENI} | NO | NO | | |
| ST2 {with TGW ENI} | YES | NO | YES | NO |
| ST2 {without TGW ENI} | NO | NO | NO | NO |

6.1.5.4.1 4.1 ST1 to SS

EC2 in Subnet with TGW attach trying to communicate with SS VPC EC2 instances:

```
Session ID: albamaria.diazfernandez@vodafone.com-0ed447256933b3e7c           Instance ID: i-0d9a572c40079c75f
sh-4.2$ ping 10.181.18.126
PING 10.181.18.126 (10.181.18.126) 56(84) bytes of data.
64 bytes from 10.181.18.126: icmp_seq=1 ttl=254 time=0.703 ms
64 bytes from 10.181.18.126: icmp_seq=2 ttl=254 time=0.512 ms
64 bytes from 10.181.18.126: icmp_seq=3 ttl=254 time=0.543 ms
64 bytes from 10.181.18.126: icmp_seq=4 ttl=254 time=0.619 ms
64 bytes from 10.181.18.126: icmp_seq=5 ttl=254 time=0.558 ms
64 bytes from 10.181.18.126: icmp_seq=6 ttl=254 time=0.583 ms
^C
--- 10.181.18.126 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5113ms
rtt min/avg/max/mdev = 0.512/0.586/0.703/0.064 ms
sh-4.2$ ping 10.181.18.52
PING 10.181.18.52 (10.181.18.52) 56(84) bytes of data.
^C
--- 10.181.18.52 ping statistics ---
13 packets transmitted, 0 received, 100% packet loss, time 12288ms
```

Problem: EC2 instance in SS VPC where there is not the TGW attachment does not receive any packet from TEST.

6.1.5.4.2 Solution: Add route to TGW in all subnets

6.1.5.4.3 4.2 SOLUTION: ROUTE TO TGW - PHASE 1

Because of the results in the previous section, it is clear now that all subnets need a route to send traffic to transit GW. A data source will be created in terraform for each VPC with a TGW attachment, to collect all route tables ids, and add a new route. At the moment, for TEST VPC's, this route is going to be the default route, "0.0.0.0/0". With this methodology, we avoid adding manually the route table ids. This is a sample of the terraform code added to do this:

```
#getting route table
ids for st1

data
"aws_route_tables" "rts_st1" {
    provider = "aws.test"

    vpc_id =
var.test_vpc_attachments["st1"].vpc_id
}

#converting set into
list
```

```

locals {
    aws_route_table_st1_list =
    tolist(data.aws_route_tables.rts_st1.ids)
    ...
}

resource
"aws_route" "test_vpc_route_st1" {
    provider = "aws.test"
    depends_on =
    ["aws_ram_principal_association.ram_principal",
     "aws_ram_resource_association.ram_assoc"]
    count =
    length(local.aws_route_table_st1_list)
    route_table_id          =
    element(local.aws_route_table_st1_list, count.index)
    destination_cidr_block   = "0.0.0.0/0"
    transit_gateway_id       = aws_ec2_transit_gateway.transit_gw.id
}

```

6.1.5.5 5. RESULTS TEST-SS CONNECTIVITY - PHASE 2

- **Test st1:**
 - EC2 in Subnet with TGW attach. **Private IP: 10.181.157.159**
 - EC2 in Subnet without TGW attach. **Private IP: 10.181.135.130**
- **Test st2:**
 - EC2 in Subnet with TGW attach. **Private IP: 10.181.189.84**
 - EC2 in Subnet without TGW attach. **Private IP: 10.181.169.179**
- **SS:**
 - EC2 in Subnet with TGW attach. **Private IP: address: 10.181.18.126**
 - EC2 in Subnet without TGW attach. **Private IP: 10.181.18.52**

| CONNECTIVITY | SS {with TGW ENI} | SS {without TGW ENI} | ST1 {with TGW ENI} | ST1 {without TGW ENI} | ST2 {with TGW ENI} | ST2 {without TGW ENI} |
|-----------------------|-------------------|----------------------|--------------------|-----------------------|--------------------|-----------------------|
| ST1 {with TGW ENI} | YES | YES | | | YES | YES |
| ST1 {without TGW ENI} | YES | YES | | | YES | YES |
| ST2 {with TGW ENI} | YES | YES | YES | YES | | |

| | | | | | | |
|-----------------------|-----|-----|-----|-----|--|--|
| ST2 {without TGW ENI} | YES | YES | YES | YES | | |
|-----------------------|-----|-----|-----|-----|--|--|

6.1.5.5.1 5.1 ST1 to SS and to ST2

EC2 in Subnet without TGW attach. Private IP: 10.181.135.130

```

sh-4.2$ #st2 with tgw attach
sh-4.2$ ping 10.181.189.84
PING 10.181.189.84 (10.181.189.84) 56(84) bytes of data.
64 bytes from 10.181.189.84: icmp_seq=1 ttl=254 time=0.789 ms
64 bytes from 10.181.189.84: icmp_seq=2 ttl=254 time=0.584 ms
64 bytes from 10.181.189.84: icmp_seq=3 ttl=254 time=0.628 ms
^Z
[3]+ Stopped(SIGTSTP)          ping 10.181.189.84
sh-4.2$ #st2 without tgw attach
sh-4.2$ ping 10.181.169.179
PING 10.181.169.179 (10.181.169.179) 56(84) bytes of data.
64 bytes from 10.181.169.179: icmp_seq=1 ttl=254 time=0.685 ms
64 bytes from 10.181.169.179: icmp_seq=2 ttl=254 time=0.549 ms
64 bytes from 10.181.169.179: icmp_seq=3 ttl=254 time=0.601 ms
64 bytes from 10.181.169.179: icmp_seq=4 ttl=254 time=0.552 ms
^Z
[4]+ Stopped(SIGTSTP)          ping 10.181.169.179
sh-4.2$ #ss with tgw attach
sh-4.2$ ping 10.181.18.126
PING 10.181.18.126 (10.181.18.126) 56(84) bytes of data.
64 bytes from 10.181.18.126: icmp_seq=1 ttl=254 time=0.594 ms
64 bytes from 10.181.18.126: icmp_seq=2 ttl=254 time=0.557 ms
64 bytes from 10.181.18.126: icmp_seq=3 ttl=254 time=0.588 ms
64 bytes from 10.181.18.126: icmp_seq=4 ttl=254 time=0.580 ms
^Z
[5]+ Stopped(SIGTSTP)          ping 10.181.18.126
sh-4.2$ #ss without tgw attach
sh-4.2$ ping 10.181.18.52
PING 10.181.18.52 (10.181.18.52) 56(84) bytes of data.
64 bytes from 10.181.18.52: icmp_seq=1 ttl=254 time=0.777 ms
64 bytes from 10.181.18.52: icmp_seq=2 ttl=254 time=0.584 ms
64 bytes from 10.181.18.52: icmp_seq=3 ttl=254 time=0.640 ms
64 bytes from 10.181.18.52: icmp_seq=4 ttl=254 time=0.639 ms
^Z
[6]+ Stopped(SIGTSTP)          ping 10.181.18.52
sh-4.2$ █

```

5.2 SS to TEST

```

sh-4.2$ ping 10.181.157.159
PING 10.181.157.159 (10.181.157.159) 56(84) bytes of data.
64 bytes from 10.181.157.159: icmp_seq=1 ttl=254 time=0.753 ms
64 bytes from 10.181.157.159: icmp_seq=2 ttl=254 time=0.664 ms
64 bytes from 10.181.157.159: icmp_seq=3 ttl=254 time=0.552 ms
^C
--- 10.181.157.159 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2041ms
rtt min/avg/max/mdev = 0.552/0.656/0.753/0.084 ms
sh-4.2$ ping 10.181.135.130
PING 10.181.135.130 (10.181.135.130) 56(84) bytes of data.
64 bytes from 10.181.135.130: icmp_seq=1 ttl=254 time=0.669 ms
64 bytes from 10.181.135.130: icmp_seq=2 ttl=254 time=0.654 ms
64 bytes from 10.181.135.130: icmp_seq=3 ttl=254 time=0.702 ms
^Z
[1]+ Stopped(SIGTSTP)           ping 10.181.135.130
sh-4.2$ ping 10.181.189.84
PING 10.181.189.84 (10.181.189.84) 56(84) bytes of data.
64 bytes from 10.181.189.84: icmp_seq=1 ttl=254 time=0.830 ms
64 bytes from 10.181.189.84: icmp_seq=2 ttl=254 time=0.597 ms
64 bytes from 10.181.189.84: icmp_seq=3 ttl=254 time=0.611 ms
^Z
[2]+ Stopped(SIGTSTP)           ping 10.181.189.84
sh-4.2$ ping 10.181.169.179
PING 10.181.169.179 (10.181.169.179) 56(84) bytes of data.
64 bytes from 10.181.169.179: icmp_seq=1 ttl=254 time=0.618 ms
64 bytes from 10.181.169.179: icmp_seq=2 ttl=254 time=0.626 ms
64 bytes from 10.181.169.179: icmp_seq=3 ttl=254 time=0.614 ms
^Z
[3]+ Stopped(SIGTSTP)           ping 10.181.169.179
sh-4.2$ 

```

6.1.5.6 6.TRANSIT GATEWAY CONFIGURATION (release 2)

This is the configuration of the TGW after adding pre-prod and prod vpc's to the configuration:

6.1.5.6.1 6.1 TGW VPC attachments

| Create Transit Gateway Attachment | | Actions | | | |
|---|-------------------------------|-----------------------|---------------|-----------------------|--|
| <input type="text"/> Filter by tags and attributes or search by keyword | | | | | |
| Name | Transit Gateway attachment ID | Transit Gateway ID | Resource type | Resource ID | |
| VFIE-Transit-GW-test-vpc-attachment-vpc-st2 | tgw-attach-0117c5343ff5b6e47 | tgw-088852b5024504fd8 | VPC | vpc-0270246dc82c513fc | |
| VFIE-Transit-GW-test-vpc-attachment-vpc-st4 | tgw-attach-012868b212f8bcd4f | tgw-088852b5024504fd8 | VPC | vpc-070981e1316c4795f | |
| VFIE-Transit-GW-test-vpc-attachment-vpc-st3 | tgw-attach-020480d4ce81d50db | tgw-088852b5024504fd8 | VPC | vpc-07bf6d07ae85e68ee | |
| VFIE-Transit-GW-preprod-vpc-attachment-vpc-prd1 | tgw-attach-0536362ff794df69f | tgw-088852b5024504fd8 | VPC | vpc-005d0af4e9a26cf3d | |
| VFIE-Transit-GW-prod-vpc-attachment-vpc-prod | tgw-attach-0552b3fc89b674bc8 | tgw-088852b5024504fd8 | VPC | vpc-0b2285ee1e4c38bae | |
| VFIE-Transit-GW-SS-vpc-attachment-vpc-0bc5e297e4fd20219 | tgw-attach-059eed3036acc6991 | tgw-088852b5024504fd8 | VPC | vpc-0bc5e297e4fd20219 | |
| VFIE-Transit-GW-test-vpc-attachment-vpc-st1 | tgw-attach-0969c8e937cd6b7d8 | tgw-088852b5024504fd8 | VPC | vpc-01a98bfaa291c87d | |
| VFIE-Transit-GW-preprod-vpc-attachment-vpc-prd2 | tgw-attach-0bad5fd7a40ce9209 | tgw-088852b5024504fd8 | VPC | vpc-0c64b9096fe8d7ef3 | |

6.1.5.6.2 6.2 TGW Route tables

Pre-prod TGW route table:

| CIDR | Attachment | Resource type | Route type | Route state |
|-----------------|--|---------------|------------|-------------|
| 0.0.0.0/0 | tgw-attach-059eed3036acc6991 vpc-0bc5e297e4fd20219 | VPC | static | active |
| 10.181.128.0/17 | - | - | static | blackhole |
| 10.181.32.0/19 | tgw-attach-0552b3fc89674bc8 vpc-0b2285ee1e4c38bae | VPC | propagated | active |
| 10.181.64.0/18 | - | - | static | blackhole |

Prod TGW route table:

| CIDR | Attachment | Resource type | Route type | Route state |
|-----------------|--|---------------|------------|-------------|
| 0.0.0.0/0 | tgw-attach-059eed3036acc6991 vpc-0bc5e297e4fd20219 | VPC | static | active |
| 10.181.128.0/17 | - | - | static | blackhole |
| 10.181.32.0/19 | - | - | static | blackhole |
| 10.181.64.0/19 | tgw-attach-0536362ff794dfe9f vpc-005d0af4e9a26cf3d | VPC | propagated | active |
| 10.181.96.0/19 | tgw-attach-0bad5fd7a40ce9209 vpc-0c64b9096fe8d7ef3 | VPC | propagated | active |

6.1.5.7 7. CONNECTIVITY TEST (TEST-PREPROD-PROD-SS)

In pre-prod and prod the same logic than the one deployed in TEST have been followed. SSM endpoints have been enabled in a private subnet of each vpc (prd1, prd2 and prod) that is involved in this test, as well as security groups to allow the required traffic.

- **Test st1:**
- EC2 in Subnet with TGW attach. Private IP: 10.181.157.159
- EC2 in Subnet without TGW attach. Private IP: 10.181.135.130
- **Test st2:**
- EC2 in Subnet with TGW attach. Private IP: 10.181.189.84
- EC2 in Subnet without TGW attach. Private IP: 10.181.169.179
- **SS:**
- EC2 in Subnet with TGW attach. Private IP: address: 10.181.18.126
- EC2 in Subnet without TGW attach. Private IP: 10.181.18.52

- Pre-prod
- PRD1
- EC2 in Subnet with TGW attach. Private IP: address: 10.181.93.37
- EC2 in Subnet without TGW attach. Private IP: 10.181.71.39
- PRD2
- EC2 in Subnet with TGW attach. Private IP: address: 10.181.125.144
- EC2 in Subnet without TGW attach. Private IP: 10.181.103.153
- Prod
- EC2 in Subnet with TGW attach. Private IP: address: 10.181.61.230
- EC2 in Subnet without TGW attach. Private IP: 10.181.39.200

6.1.5.7.1 7.1 Prod web ec2 (10.181.39.200)

Prod can talk with SS, but it cannot talk with any other environment.

```
sh-4.2$ #ss vpc
sh-4.2$ ping 10.181.18.126
PING 10.181.18.126 (10.181.18.126) 56(84) bytes of data.
64 bytes from 10.181.18.126: icmp_seq=1 ttl=254 time=1.36 ms
64 bytes from 10.181.18.126: icmp_seq=2 ttl=254 time=1.10 ms
64 bytes from 10.181.18.126: icmp_seq=3 ttl=254 time=1.11 ms
^C
--- 10.181.18.126 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.102/1.193/1.362/0.119 ms
sh-4.2$ ping 10.181.18.52
PING 10.181.18.52 (10.181.18.52) 56(84) bytes of data.
64 bytes from 10.181.18.52: icmp_seq=1 ttl=254 time=1.32 ms
64 bytes from 10.181.18.52: icmp_seq=2 ttl=254 time=1.32 ms
64 bytes from 10.181.18.52: icmp_seq=3 ttl=254 time=1.19 ms
64 bytes from 10.181.18.52: icmp_seq=4 ttl=254 time=1.14 ms
^Z
[1]+  Stopped(SIGTSTP)          ping 10.181.18.52
sh-4.2$ #pre-prod vpc
sh-4.2$ ping 10.181.93.37
PING 10.181.93.37 (10.181.93.37) 56(84) bytes of data.
^Z
[2]+  Stopped(SIGTSTP)          ping 10.181.93.37
```

6.1.5.7.2 7.2 Pre-Prod web ec2 (10.181.71.39)

Pre-Prod can talk with SS, but it cannot talk with any other environment (prod, test).

```
sh-4.2$ #ss
sh-4.2$ ping 10.181.18.126
PING 10.181.18.126 (10.181.18.126) 56(84) bytes of data.
64 bytes from 10.181.18.126: icmp_seq=1 ttl=254 time=0.832 ms
64 bytes from 10.181.18.126: icmp_seq=2 ttl=254 time=0.559 ms
64 bytes from 10.181.18.126: icmp_seq=3 ttl=254 time=0.573 ms
64 bytes from 10.181.18.126: icmp_seq=4 ttl=254 time=0.573 ms
^Z
[1]+  Stopped(SIGTSTP)          ping 10.181.18.126
sh-4.2$ ping 10.181.18.52
PING 10.181.18.52 (10.181.18.52) 56(84) bytes of data.
64 bytes from 10.181.18.52: icmp_seq=1 ttl=254 time=0.813 ms
64 bytes from 10.181.18.52: icmp_seq=2 ttl=254 time=0.586 ms
64 bytes from 10.181.18.52: icmp_seq=3 ttl=254 time=0.598 ms
^Z
[2]+  Stopped(SIGTSTP)          ping 10.181.18.52
sh-4.2$ #prod
sh-4.2$ ping 10.181.61.230
PING 10.181.61.230 (10.181.61.230) 56(84) bytes of data.
^Z
[3]+  Stopped(SIGTSTP)          ping 10.181.61.230
sh-4.2$ █
```

6.1.5.7.3 7.3 SS private EC2 (10.181.18.126)

```

sh-4.2$ #prod
sh-4.2$ ping 10.181.61.230
PING 10.181.61.230 (10.181.61.230) 56(84) bytes of data.
64 bytes from 10.181.61.230: icmp_seq=1 ttl=254 time=1.43 ms
64 bytes from 10.181.61.230: icmp_seq=2 ttl=254 time=1.13 ms
64 bytes from 10.181.61.230: icmp_seq=3 ttl=254 time=1.11 ms
64 bytes from 10.181.61.230: icmp_seq=4 ttl=254 time=1.14 ms
^Z
[1]+ Stopped(SIGTSTP)          ping 10.181.61.230
sh-4.2$ ping 10.181.39.200
PING 10.181.39.200 (10.181.39.200) 56(84) bytes of data.
64 bytes from 10.181.39.200: icmp_seq=1 ttl=254 time=1.16 ms
64 bytes from 10.181.39.200: icmp_seq=2 ttl=254 time=0.942 ms
64 bytes from 10.181.39.200: icmp_seq=3 ttl=254 time=0.885 ms
^Z
[2]+ Stopped(SIGTSTP)          ping 10.181.39.200
sh-4.2$ #PREPROD
sh-4.2$ ping 10.181.93.37
PING 10.181.93.37 (10.181.93.37) 56(84) bytes of data.
64 bytes from 10.181.93.37: icmp_seq=1 ttl=254 time=0.817 ms
64 bytes from 10.181.93.37: icmp_seq=2 ttl=254 time=0.585 ms
64 bytes from 10.181.93.37: icmp_seq=3 ttl=254 time=0.596 ms
64 bytes from 10.181.93.37: icmp_seq=4 ttl=254 time=0.745 ms
^Z
[3]+ Stopped(SIGTSTP)          ping 10.181.93.37
sh-4.2$ ping 10.181.71.39
PING 10.181.71.39 (10.181.71.39) 56(84) bytes of data.
64 bytes from 10.181.71.39: icmp_seq=1 ttl=254 time=0.829 ms
64 bytes from 10.181.71.39: icmp_seq=2 ttl=254 time=0.568 ms
64 bytes from 10.181.71.39: icmp_seq=3 ttl=254 time=0.758 ms
64 bytes from 10.181.71.39: icmp_seq=4 ttl=254 time=0.613 ms
^Z
[4]+ Stopped(SIGTSTP)          ping 10.181.71.39
sh-4.2$ ping 10.181.125.144
PING 10.181.125.144 (10.181.125.144) 56(84) bytes of data.
64 bytes from 10.181.125.144: icmp_seq=1 ttl=254 time=1.24 ms
64 bytes from 10.181.125.144: icmp_seq=2 ttl=254 time=0.682 ms
64 bytes from 10.181.125.144: icmp_seq=3 ttl=254 time=0.701 ms
^Z
[5]+ Stopped(SIGTSTP)          ping 10.181.125.144
sh-4.2$ ping 10.181.103.153
PING 10.181.103.153 (10.181.103.153) 56(84) bytes of data.
64 bytes from 10.181.103.153: icmp_seq=1 ttl=254 time=0.933 ms

```

6.1.6 CURRENT DESIGN AND ROUTE TABLES

6.1.6.1

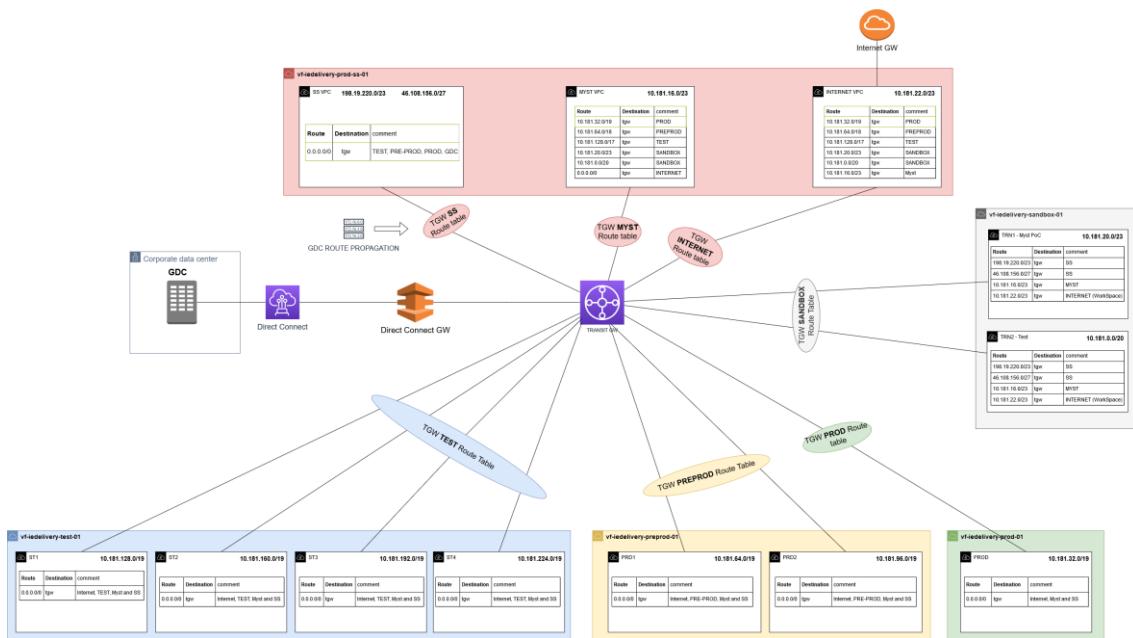
- [1.INTRODUCTION](#)
- [2.ARCHITECTURE](#)
- [3.TGW ROUTE TABLE](#)

- [3.1 sandbox TGW route table](#)
- [3.2 Test TGW route table](#)
- [3.3 Pre-Prod TGW route table](#)
- [3.4 Prod TGW route table](#)
- [3.5 SS TGW route table](#)
- [3.5 Internet TGW route table](#)
- [3.6 Myst TGW route table](#)
- [4. GDC CONNECTIVITY DETAILS](#)
 - [4.1 DIRECT CONNECT GW AND VIF](#)
 - [4.2 ROUTE PROPAGATION IN TGW](#)
 - [4.4 DX TGW ROUTE TABLE](#)
- [5. NETWORK MANAGER](#)

6.1.6.2 1.INTRODUCTION

This section shows the current design for VFIE regarding the TGW and the routing in place. See the table with the details about CIDR IP address per AWS VPC [here](#).

6.1.6.3 2.ARCHITECTURE



6.1.6.4 3.TGW ROUTE TABLE

6.1.6.4.1 3.1 sandbox TGW route table

| Route | Destination | Type |
|-----------------|---------------------|--------|
| 10.181.16.0/23 | myst vpc attach | STATIC |
| 198.19.220.0/23 | ss vpc attach | STATIC |
| 10.181.22.0/23 | Internet vpc attach | STATIC |

| Route | Destination | Type |
|----------------|--------------------|-------------|
| 10.181.32.0/19 | DROP PROD | STATIC |
| 10.181.64.0/19 | DROP PRE-PROD | STATIC |
| 10.181.96.0/19 | DROP PRE-PROD | STATIC |
| 10.181.0.0/20 | sandbox | PROPAGATED |
| 10.181.20.0/23 | sandbox | PROPAGATED |

6.1.6.4.2 3.2 Test TGW route table

| Route | Destination | Type |
|-----------------|---------------------|-------------|
| 10.181.16.0/23 | myst vpc attach | STATIC |
| 198.19.220.0/23 | ss vpc attach | STATIC |
| 10.181.22.0/23 | Internet vpc attach | STATIC |
| 10.181.32.0/19 | DROP PROD | STATIC |
| 10.181.64.0/19 | DROP PRE-PROD | STATIC |
| 10.181.96.0/19 | DROP PRE-PROD | STATIC |
| 10.181.128.0/19 | Test | PROPAGATED |
| 10.181.128.0/19 | Test | PROPAGATED |
| 10.181.192.0/19 | Test | PROPAGATED |
| 10.181.224.0/19 | Test | PROPAGATED |

6.1.6.4.3 3.3 Pre-Prod TGW route table

| Route | Destination | Type |
|-----------------|---------------------|-------------|
| 10.181.16.0/23 | myst vpc attach | STATIC |
| 198.19.220.0/23 | ss vpc attach | STATIC |
| 0.0.0.0/0 | Internet vpc attach | STATIC |
| 10.181.32.0/19 | DROP PROD | STATIC |
| 10.181.128.0/17 | DROP TEST | STATIC |
| 10.181.64.0/19 | pre-prod | PROPAGATED |
| 10.181.96.0/19 | pre-prod | PROPAGATED |

6.1.6.4.4 3.4 Prod TGW route table

| Route | Destination | Type |
|-----------------|---------------------|-------------|
| 10.181.16.0/23 | myst vpc attach | STATIC |
| 198.19.220.0/23 | ss vpc attach | STATIC |
| 0.0.0.0/0 | Internet vpc attach | STATIC |
| 10.181.64.0/18 | DROP PRE-PROD | STATIC |

| Route | Destination | Type |
|-----------------|-------------|--------|
| 10.181.128.0/17 | DROP TEST | STATIC |

6.1.6.4.5 3.5 SS TGW route table

| Route | Destination | Type |
|-----------------|----------------------|--------|
| 10.181.32.0/19 | prod vpc attach | STATIC |
| 10.181.64.0/19 | pre-prod1 vpc attach | STATIC |
| 10.181.96.0/19 | pre-prod2 vpc attach | STATIC |
| 10.181.128.0/19 | test1 vpc attach | STATIC |
| 10.181.160.0/19 | test2 vpc attach | STATIC |
| 10.181.192.0/19 | test3 vpc attach | STATIC |
| 10.181.224.0/19 | test4 vpc attach | STATIC |
| 10.181.0.0/20 | sandbox vpc attach | STATIC |
| 10.181.20.0/23 | sandbox vpc attach | STATIC |

6.1.6.4.6 3.5 Internet TGW route table

| Route | Destination | Type |
|-----------------|----------------------|--------|
| 10.181.32.0/19 | prod vpc attach | STATIC |
| 10.181.64.0/19 | pre-prod1 vpc attach | STATIC |
| 10.181.96.0/19 | pre-prod2 vpc attach | STATIC |
| 10.181.128.0/19 | test1 vpc attach | STATIC |
| 10.181.160.0/19 | test2 vpc attach | STATIC |
| 10.181.192.0/19 | test3 vpc attach | STATIC |
| 10.181.224.0/19 | test4 vpc attach | STATIC |
| 10.181.0.0/20 | sandbox vpc attach | STATIC |
| 10.181.20.0/23 | sandbox vpc attach | STATIC |
| 10.181.16.0/23 | myst vpc attach | STATIC |

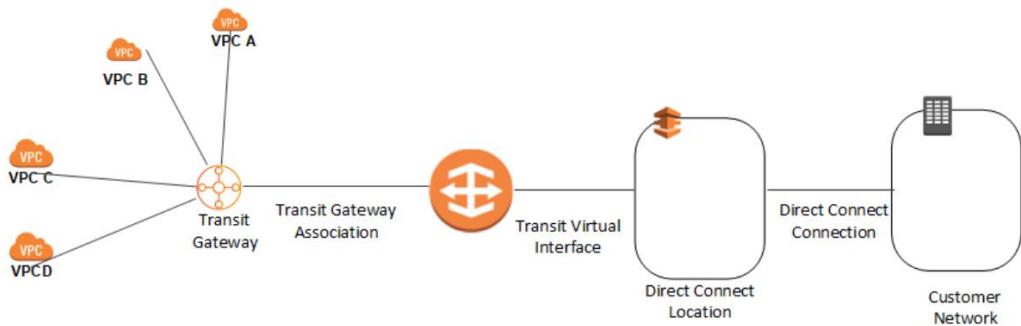
6.1.6.4.7 3.6 Myst TGW route table

| Route | Destination | Type |
|-----------------|----------------------|--------|
| 10.181.32.0/19 | prod vpc attach | STATIC |
| 10.181.64.0/19 | pre-prod1 vpc attach | STATIC |
| 10.181.96.0/19 | pre-prod2 vpc attach | STATIC |
| 10.181.128.0/19 | test1 vpc attach | STATIC |

| Route | Destination | Type |
|-----------------|---------------------|--------|
| 10.181.160.0/19 | test2 vpc attach | STATIC |
| 10.181.192.0/19 | test3 vpc attach | STATIC |
| 10.181.224.0/19 | test4 vpc attach | STATIC |
| 10.181.0.0/20 | sandbox vpc attach | STATIC |
| 10.181.20.0/23 | sandbox vpc attach | STATIC |
| 0.0.0.0/0 | internet vpc attach | STATIC |

6.1.6.5 4. GDC CONNECTIVITY DETAILS

The connectivity between GDC and AWS will be managed by PCS. Following that principle, the direct connect configuration will be done in a PCS's managed AWS account (VF-PC-SSNET-IRE-DUB-001). We have created a direct connect that enables you to create a single connection to your Direct Connect connection that all of your VPCs can use. The direct connect gateway will be associated with the transit gateway as shown in this picture:



6.1.6.5.1 4.1 DIRECT CONNECT GW AND VIF

This is our direct connect gateway configuration in VF-PC-SSNET-IRE-DUB-001 account:

Direct Connect > Direct Connect gateways > E9B17FE9-4691-447D-930E-6AB9E343CEB5

| General configuration | | | | | | | | | | | | | | |
|--|-----------------------------|--------------------------|----------|--------|-------------|-------|---------------|-----------|--------------|----------|----------------|-----------|--------------|----------|
| ID e9b17fe9-4691-447d-930e-6ab9e343ceb5 | AWS account 127683202867 | Amazon side ASN 65239 | | | | | | | | | | | | |
| Name VFIE-DC-GW-PROD | State Available | | | | | | | | | | | | | |
| Virtual interface attachments Gateway associations | | | | | | | | | | | | | | |
| Virtual interface attachments (2) <table border="1"> <thead> <tr> <th>ID</th> <th>Region</th> <th>AWS account</th> <th>State</th> </tr> </thead> <tbody> <tr> <td>dxvif-fgjmzpo</td> <td>eu-west-1</td> <td>127683202867</td> <td>Attached</td> </tr> <tr> <td>dxvif-fgppa880</td> <td>eu-west-1</td> <td>127683202867</td> <td>Attached</td> </tr> </tbody> </table> | | | ID | Region | AWS account | State | dxvif-fgjmzpo | eu-west-1 | 127683202867 | Attached | dxvif-fgppa880 | eu-west-1 | 127683202867 | Attached |
| ID | Region | AWS account | State | | | | | | | | | | | |
| dxvif-fgjmzpo | eu-west-1 | 127683202867 | Attached | | | | | | | | | | | |
| dxvif-fgppa880 | eu-west-1 | 127683202867 | Attached | | | | | | | | | | | |

The direct connect gateway has two virtual interfaces (transit type) attached:

DXVIF-FGJMZNPO

| General configuration | | | |
|--|--|---|----------------------------------|
| Virtual interface ID dxvif-fgjmznpo | State available | Amazon side ASN 65239 | AWS device ITXD2-251t75dn33mm |
| Virtual interface name 06-VFIE-IRE-IE1CGW01ATL1-1 | Direct Connect gateway e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Connection ID dxcon-futf5ng | MTU 1500 |
| AWS account 127683202867 | VLAN 114 | Location Interxion DUB2, Dublin, IRL | Jumbo frame capable true |
| Virtual interface type transit | Region eu-west-1 | | |

Peerings | Monitoring | Tags | Test history

| Peerings (1) | | | | | | | | |
|------------------------------------|------|---------|------------------------|---------------------|-----------------------|------------------|--|---------------------------------------|
| ID | Name | BGP ASN | BGP authentication key | Your router peer IP | Amazon router peer IP | AWS device | State | BGP status |
| <input type="radio"/> dxpeer-fg... | ipv4 | 64649 | GeE8pZxc7 | 10.216.197.250/30 | 10.216.197.249/30 | ITXD2-251t75d... | available | up |

DXVIF-FGPPA880

| General configuration | | | |
|--|--|--|---------------------------------|
| Virtual interface ID dxvif-fgppa880 | State available | Amazon side ASN 65239 | AWS device EirCL-2bi1jt9w5o9 |
| Virtual interface name 05-VFIE-IRE-IE1CGW01ATL1-1 | Direct Connect gateway e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Connection ID dxcon-fzifqil | MTU 1500 |
| AWS account 127683202867 | VLAN 113 | Location Eircom Clonshaugh, Dublin, IRL | Jumbo frame capable true |
| Virtual interface type transit | Region eu-west-1 | | |

Peerings | Monitoring | Tags | Test history

| Peerings (1) | | | | | | | | |
|------------------------------------|------|---------|------------------------|---------------------|-----------------------|-------------------|--|---------------------------------------|
| ID | Name | BGP ASN | BGP authentication key | Your router peer IP | Amazon router peer IP | AWS device | State | BGP status |
| <input type="radio"/> dxpeer-fg... | ipv4 | 64649 | GeE8pZxc7 | 10.216.197.246/30 | 10.216.197.245/30 | EirCL-2bi1jt9w... | available | up |

6.1.6.5.2 4.2 ROUTE PROPAGATION IN TGW

In VFIE SS account (vf-iedelivery-prod-ss-01) we need to associate for the direct connect gateway (this step was made after creating the direct connect gateway). The allowed prefixes are the ones we are propagating from AWS to on-premise:

TGW-0ADFFD31B174D61B4

| General configuration | | |
|-----------------------------|---|---------------------|
| ID tgw-0adffd31b174d61b4 | State available | Region eu-west-1 |
| Name VFIE-Transit-GW | Amazon side ASN 64512 | |

Direct Connect gateway associations

| Direct Connect gateway associations (1) | | | | | |
|---|--------------------------------------|--|----------------------------------|---|---|
| <input type="checkbox"/> Search direct connect gateway associations | | Edit Disassociate Associate Direct Connect gateway | | < 1 > ⌂ | |
| <input type="checkbox"/> | Direct Connect gateway | ▲ Direct Connect gateway owner | ▼ Allowed prefixes | ▼ State | ▼ |
| <input type="checkbox"/> | e9b17fe9-4691-447d-930e-6ab9e343ceb5 | 127683202867 | 198.19.220.0/24, 46.108.156.0/27 | associated | |

For our scenario, we just want SS VPC to have access to GDC. So, we will need to modify TGW SS Route table and allow propagation from GDC:

Transit Gateway Route Table: tgw-rb-04770da1934bab14b

| Name | Transit Gateway route table ID | Transit Gateway ID | State | Default association route table | Default propagation route table |
|---|---------------------------------|-----------------------------|------------------|---------------------------------|---------------------------------|
| VFIE-Transit-GW-Internet-Route-Table-vpc | tgw-rb-019a51506bdd4cc7 | tgw-0adff31b174d51b4 | available | No | No |
| VFIE-Transit-GW-PRE-PROD-Route-Table-vpc | tgw-rb-01db61ccf85f39a10 | tgw-0adff31b174d51b4 | available | No | No |
| VFIE-Transit-GW-PROD-Route-Table-vpc | tgw-rb-029b504a113b66127 | tgw-0adff31b174d51b4 | available | No | No |
| VFIE-Transit-GW-SharedServices-Route-Table-vpc | tgw-rb-04770da1934bab14b | tgw-0adff31b174d51b4 | available | No | No |
| VFIE-Transit-GW-TEST-Route-Table-vpc | tgw-rb-0395b0057ab76aaa | tgw-0adff31b174d51b4 | available | No | No |
| VFIE-Transit-GW-myst-Route-Table-vpc | tgw-rb-01a8487c905b2b109 | tgw-0adff31b174d51b4 | available | No | No |
| VFIE-Transit-GW-sandbox-Route-Table-vpc | tgw-rb-084ea4828fb299ebc | tgw-0adff31b174d51b4 | available | No | No |

Transit Gateway Route Table: tgw-rb-04770da1934bab14b

| Details | Associations | Propagations | Routes | Tags | | | | | | | | | | | | |
|---|---------------------------|--------------------------------------|---------|------|---------------|---------------|-------------|-------|------------------------------|------------------------|--------------------------------------|---------|-----------------------------|-----|----------------------|---------|
| Create propagation | Delete propagation | | | | | | | | | | | | | | | |
| <p>Filter by attributes or search by keyword</p> <table border="1"> <thead> <tr> <th>Attachment ID</th> <th>Resource type</th> <th>Resource ID</th> <th>State</th> </tr> </thead> <tbody> <tr><td>tgw-attach-0c1146abee499da4b</td><td>Direct Connect Gateway</td><td>e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>enabled</td></tr> <tr><td>tgw-attach-0f3560aa7fe99727</td><td>VPC</td><td>vpc-073c17fe9e74423c</td><td>enabled</td></tr> </tbody> </table> | | | | | Attachment ID | Resource type | Resource ID | State | tgw-attach-0c1146abee499da4b | Direct Connect Gateway | e9b17fe9-4691-447d-930e-6ab9e343ceb5 | enabled | tgw-attach-0f3560aa7fe99727 | VPC | vpc-073c17fe9e74423c | enabled |
| Attachment ID | Resource type | Resource ID | State | | | | | | | | | | | | | |
| tgw-attach-0c1146abee499da4b | Direct Connect Gateway | e9b17fe9-4691-447d-930e-6ab9e343ceb5 | enabled | | | | | | | | | | | | | |
| tgw-attach-0f3560aa7fe99727 | VPC | vpc-073c17fe9e74423c | enabled | | | | | | | | | | | | | |

This is the result:

Transit Gateway Route Table: tgw-rb-04770da1934bab14b

| Details | Associations | Propagations | Routes | Tags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---------------------|---------------|-------------|------|------------|---------------|------------|-------------|------------|---|---------------|------------|--------|---------------|--|-----|--------|--------|-----------------|--|-----|--------|--------|-----------------|---|-----|--------|--------|-----------------|--|-----|--------|--------|----------------|--|-----|--------|--------|-----------------|--|-----|--------|--------|----------------|--|-----|--------|--------|----------------|---|-----|--------|--------|----------------|--|-----|--------|--------|-----------------|---|---------------|------------|--------|----------------|---|---------------|------------|--------|---------------|---|---------------|------------|--------|------------------|---|---------------|------------|--------|----------------|---|---------------|------------|--------|----------------|---|---------------|------------|--------|---------------|---|---------------|------------|--------|-----------------|--|-----|------------|--------|----------------|---|---------------|------------|--------|-----------------|--|-----|------------|--------|
| <p>The table below will return a maximum of 1000 routes. Narrow the filter or use export routes to view more routes.</p> <p>Create route Replace routes Delete routes</p> <p>Filter by attributes or search by keyword</p> <table border="1"> <thead> <tr> <th>CIDR</th> <th>Attachment</th> <th>Resource type</th> <th>Route type</th> <th>Route state</th> </tr> </thead> <tbody> <tr><td>10.0.0.0/8</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>10.181.0.0/20</td><td>tgw-attach-048ff6dade99fe23 vpc-048ff6dade99fe23</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.128.0/19</td><td>tgw-attach-0e185fad3ffbcfd vpc-01a88ba291f87fd</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.160.0/19</td><td>tgw-attach-0c0338fe00d7afe vpc-0270246c82c513fc</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.192.0/19</td><td>tgw-attach-0809723aa241e4ed vpc-07bfad07ae5e58ee</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.20.0/23</td><td>tgw-attach-0e930a59fc022277 vpc-070df2886cda0c</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.224.0/19</td><td>tgw-attach-0e795f13a0347febb vpc-0709fe1e13164795f</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.32.0/19</td><td>tgw-attach-0f01364488ab0cd0 vpc-05228fe1e4c38bae</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.64.0/19</td><td>tgw-attach-020597498bbcc45 vpc-005d0d44e9a95d5d</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>10.181.96.0/19</td><td>tgw-attach-000d0812a778298 vpc-0549b9968d87e7d</td><td>VPC</td><td>static</td><td>active</td></tr> <tr><td>139.47.192.0/18</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>145.230.0.0/16</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>172.18.0.0/12</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>192.125.128.0/17</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>192.168.0.0/16</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>195.233.0.0/16</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>198.18.0.0/15</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>198.19.220.0/23</td><td>tgw-attach-0f3560aa7fe99727 vpc-073c17fe9e74423c</td><td>VPC</td><td>propagated</td><td>active</td></tr> <tr><td>37.25.136.0/21</td><td>tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5</td><td>Direct Connec</td><td>propagated</td><td>active</td></tr> <tr><td>46.108.156.0/27</td><td>tgw-attach-0f3560aa7fe99727 vpc-073c17fe9e74423c</td><td>VPC</td><td>propagated</td><td>active</td></tr> </tbody> </table> | | | | | CIDR | Attachment | Resource type | Route type | Route state | 10.0.0.0/8 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 10.181.0.0/20 | tgw-attach-048ff6dade99fe23 vpc-048ff6dade99fe23 | VPC | static | active | 10.181.128.0/19 | tgw-attach-0e185fad3ffbcfd vpc-01a88ba291f87fd | VPC | static | active | 10.181.160.0/19 | tgw-attach-0c0338fe00d7afe vpc-0270246c82c513fc | VPC | static | active | 10.181.192.0/19 | tgw-attach-0809723aa241e4ed vpc-07bfad07ae5e58ee | VPC | static | active | 10.181.20.0/23 | tgw-attach-0e930a59fc022277 vpc-070df2886cda0c | VPC | static | active | 10.181.224.0/19 | tgw-attach-0e795f13a0347febb vpc-0709fe1e13164795f | VPC | static | active | 10.181.32.0/19 | tgw-attach-0f01364488ab0cd0 vpc-05228fe1e4c38bae | VPC | static | active | 10.181.64.0/19 | tgw-attach-020597498bbcc45 vpc-005d0d44e9a95d5d | VPC | static | active | 10.181.96.0/19 | tgw-attach-000d0812a778298 vpc-0549b9968d87e7d | VPC | static | active | 139.47.192.0/18 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 145.230.0.0/16 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 172.18.0.0/12 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 192.125.128.0/17 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 192.168.0.0/16 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 195.233.0.0/16 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 198.18.0.0/15 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 198.19.220.0/23 | tgw-attach-0f3560aa7fe99727 vpc-073c17fe9e74423c | VPC | propagated | active | 37.25.136.0/21 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | 46.108.156.0/27 | tgw-attach-0f3560aa7fe99727 vpc-073c17fe9e74423c | VPC | propagated | active |
| CIDR | Attachment | Resource type | Route type | Route state | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.0.0.0/8 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.0.0/20 | tgw-attach-048ff6dade99fe23 vpc-048ff6dade99fe23 | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.128.0/19 | tgw-attach-0e185fad3ffbcfd vpc-01a88ba291f87fd | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.160.0/19 | tgw-attach-0c0338fe00d7afe vpc-0270246c82c513fc | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.192.0/19 | tgw-attach-0809723aa241e4ed vpc-07bfad07ae5e58ee | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.20.0/23 | tgw-attach-0e930a59fc022277 vpc-070df2886cda0c | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.224.0/19 | tgw-attach-0e795f13a0347febb vpc-0709fe1e13164795f | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.32.0/19 | tgw-attach-0f01364488ab0cd0 vpc-05228fe1e4c38bae | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.64.0/19 | tgw-attach-020597498bbcc45 vpc-005d0d44e9a95d5d | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10.181.96.0/19 | tgw-attach-000d0812a778298 vpc-0549b9968d87e7d | VPC | static | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 139.47.192.0/18 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 145.230.0.0/16 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 172.18.0.0/12 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 192.125.128.0/17 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 192.168.0.0/16 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 195.233.0.0/16 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 198.18.0.0/15 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 198.19.220.0/23 | tgw-attach-0f3560aa7fe99727 vpc-073c17fe9e74423c | VPC | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 37.25.136.0/21 | tgw-attach-0c1146abee499da4b e9b17fe9-4691-447d-930e-6ab9e343ceb5 | Direct Connec | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 46.108.156.0/27 | tgw-attach-0f3560aa7fe99727 vpc-073c17fe9e74423c | VPC | propagated | active | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

6.1.6.5.3 4.4 DX TGW ROUTE TABLE

We can see the direct connect association in the tab Transit Gateway Attachments:

New VPC Experience
Tell us what you think:

Managed Prefix
Lists [New](#)

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

▼ SECURITY

Network ACLs

Security Groups [New](#)

▼ VIRTUAL PRIVATE NETWORK (VPN)

Customer Gateways

Virtual Private Gateways

Site-to-Site VPN Connections

Client VPN Endpoints

▼ TRANSIT GATEWAYS

Transit Gateways

Transit Gateway Attachments

Transit Gateway Route Tables

Create Transit Gateway Attachment Actions ▾

| Name | Transit Gateway attachment ID | Transit Gateway ID | Resource type | Resource ID | State |
|---|-------------------------------------|-----------------------------|-------------------------------|---|------------------|
| VFIE-Transit-GW-pre-prod-vpc-attachment-prd1 | tgw-attach-02f597498bbc245 | tgw-0adff31b174d51b4 | VPC | vpc-0050a4e9a926c5fd | available |
| VFIE-Transit-GW-sandbox-vpc-attachment-lm2 | tgw-attach-048ff6dade99fe23 | tgw-0adff31b174d51b4 | VPC | vpc-048ff6dade99fe23 | available |
| VFIE-Transit-GW-test-vpc-attachment-s13 | tgw-attach-0809723aa241e4ed | tgw-0adff31b174d51b4 | VPC | vpc-07b60d7ae85e656ee | available |
| VFIE-Transit-GW-myst-vpc-attachment-vpc-07b601eb5f3bd9 | tgw-attach-08e08ab09a9f344e0e2 | tgw-0adff31b174d51b4 | VPC | vpc-07b60d1eb5f3bd9 | available |
| VFIE-Transit-GW-test-vpc-attachment-s12 | tgw-attach-0c4349660d7a7fe | tgw-0adff31b174d51b4 | VPC | vpc-0270246d82c213fc | available |
| VFIE-Transit-GW-internet-vpc-attachment-vpc-066cf10e6620a84cc | tgw-attach-06909e250999fe | tgw-0adff31b174d51b4 | VPC | vpc-066cf10e6620a84cc | available |
| VFIE-Transit-GW-sandbox-vpc-attachment-tm1 | tgw-attach-0e030a59f1cb2277 | tgw-0adff31b174d51b4 | VPC | vpc-070ff2886c6dc0a | available |
| VFIE-Transit-GW-test-vpc-attachment-s11 | tgw-attach-0e15ad3ff05bc45 | tgw-0adff31b174d51b4 | VPC | vpc-01a88ba291f87fd | available |
| VFIE-Transit-GW-test-vpc-attachment-s14 | tgw-attach-0e796f13a0347eeb | tgw-0adff31b174d51b4 | VPC | vpc-0709fe1e13164795f | available |
| VFIE-Transit-GW-pre-vpc-attachment-prd | tgw-attach-0f3560aa7fe99727 | tgw-0adff31b174d51b4 | VPC | vpc-06228fe1e4c38bae | available |
| VFIE-Transit-GW-GDC-direct-connect-Dx-attachment | tgw-attach-0c1146abee499da4b | tgw-0adff31b174d51b4 | Direct Connect Gateway | e9b17fe9-4691-447d-930e-6ab9e343ceb5 | available |

Transit Gateway Attachment: tgw-attach-0c1146abee499da4b

Add/Edit Tags

| Key | Value |
|------|--|
| Name | VFIE-Transit-GW-GDC-direct-connect-Dx-attachment |

We need to create a TGW Route Table for this attachment. DX can just communicate with SS, so the routes from SS VPC will be propagated to this route table.

The screenshot shows the AWS CloudFormation console with the 'Transit Gateway Route Table' page. The table lists various Transit Gateway route tables, including the selected one (tgw-rtb-03402c4f22cd27de5). The 'Associations' tab is selected, showing a single association with a Direct Connect Gateway (tgw-attach-0c1146abee499da4b).

For that case, in propagations, we will add the vpc attachment of SS VPC. These are the routes we received:

The screenshot shows the 'Routes' tab for the TGW route table tgw-rtb-03402c4f22cd27de5. It displays two propagated routes from the SS VPC:

| CIDR | Attachment | Resource type | Route type | Route state |
|-----------------|--|---------------|------------|-------------|
| 198.19.220.0/23 | tgw-attach-0f3560aa7fce99727 vpc-073c17ef0ee74423c | VPC | propagated | active |
| 46.108.156.0/27 | tgw-attach-0f3560aa7fce99727 vpc-073c17ef0ee74423c | VPC | propagated | active |

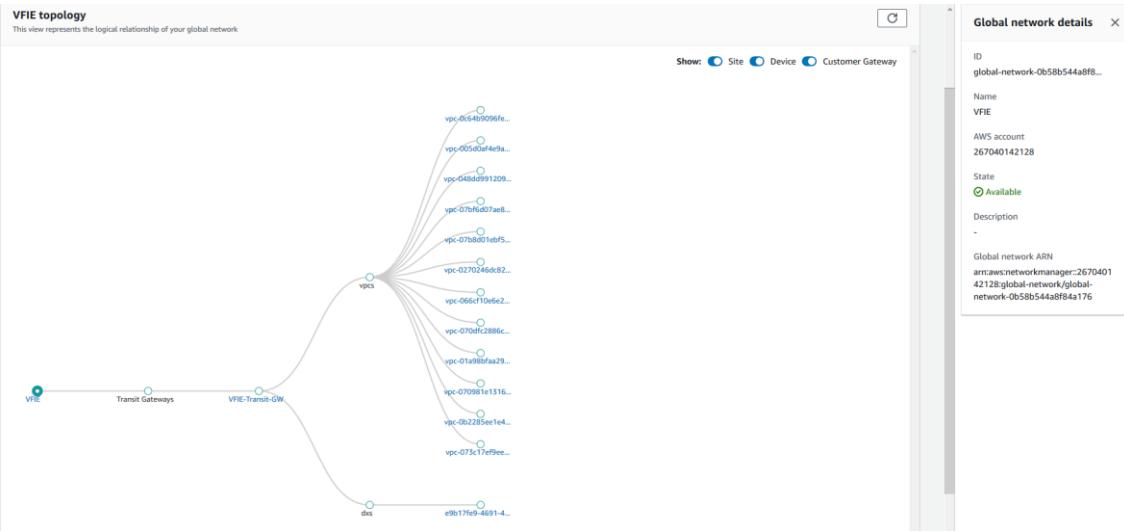
6.1.6.6 5. NETWORK MANAGER

I have created a Global Network in Network Manager to analyze the routes we have configured and monitoring the TGW behaviour. Check [here](#).

The screenshot shows the AWS Network Manager console with the 'Global networks > VFIE' page. It includes sections for:

- VFIE Inventory:** Shows 1 Transit Gateways, 0 Sites, and 0 Devices.
- Transit Gateways VPN status (1):** Shows the status of the TGW (tgw-0adfffd31b17...).
- Network events summary:** Displays recent events related to Network Manager Routing Update.

We can see in above picture the TGW that we are using, as well as last events. This is the topology to our network:



We can analyze routes [here](#). This is an example from SS VPC to GDC network:

Route Analyzer

VFIE Route Analyzer
The Route Analyzer analyzes the routing path between a specified source and destination. Note, Route Analyzer checks the routes on Transit Gateway route tables only. [Learn more](#)

| | |
|---|--|
| Source | Destination |
| Transit Gateway VFIE-Transit-GW | Transit Gateway VFIE-Transit-GW |
| Transit Gateway attachment VFIE-Transit-GW-SS-vpc-attachment-vpc-073c17ef9ee74423c | Transit Gateway attachment VFIE-Transit-GW-GDC-direct-connect-DX-attachment |
| IP address IPv4 or IPv6 address 198.19.220.42 | IP address IPv4 or IPv6 address 10.163.187.4 |

Include return path in results
 Middlebox appliance? [Info](#) If selected, state those that are known in the results

Results of route analysis
The result of source and destination route is displayed here. You can update either the source and/or destination and re-run the analysis for updated results.

| Forward path | | | Return path | | |
|------------------------------|------------------------------|-----------|------------------------------|------------------------------|-----------|
| Source | Destination | Status | Source | Destination | Status |
| tgw-attach-0f3560aa7fce99727 | tgw-attach-0c1146abee499da4b | Connected | tgw-attach-0c1146abee499da4b | tgw-attach-0f3560aa7fce99727 | Connected |

Forward path details:

- Source: 198.19.220.42 → VFIE-Transit-GW-SS-vpc-attachment-vpc-073c17ef9ee74423c (tgw-attach-0f3560aa7fce99727) VPC eu-west-1
- Destination: 10.163.187.4 → VFIE-Transit-GW-GDC-direct-connect-DX-attachment (tgw-attach-0c1146abee499da4b) Direct Connect Gateway eu-west-1

Return path details:

- Source: 198.19.220.42 → tgw-rtb-04770da1934bab14b (Transit Gateway route table) eu-west-1 tgw-0adffd31b174d61b4
- Destination: 10.163.187.4 → tgw-rtb-03402c4f22cd27de5 (Transit Gateway route table) eu-west-1 tgw-0adffd31b174d61b4

6.1.7 TGW IMPLEMENTATION: HOW TO

- [1. Introduction, Code and resources](#)
- [2. TGW Design](#)
 - [2.1 Design of the attachment and route table for a AWS Transit GW.](#)

- [2.3 Design of the attachments for VF IE VPCs](#)
- [3. TGW implementation](#)
 - [3.1 TGW resources](#)
 - [3.2 TGW configuration](#)
- [4. TGW: How to add a new VPC to the implementation](#)
 - [4.1 New AWS account](#)
 - [4.2 New VPC](#)
 - [4.3 New AWS provider](#)
 - [4.4 New VPC attachment\(s\) and VPC accepter](#)
 - [4.5 TGW Route table](#)
 - [4.6 TGW Route table association and propagation](#)
 - [4.7 VPC route tables](#)
 - [4.8 TGW INVITATION](#)

Please, follow the instructions here to modify/update entries in TGW.

6.1.7.1 1. Introduction. Code and resources

The transit gateway implementation and the corresponding CI/CD methodology (CodePipeline) have been done using Terraform. All the code is stored in the management account (vf-iedelivery-mgmt), in a codecommit repository:

- Code: Terraform
- Repository: <https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/vf-iedelivery-transit-gateway>
- AWS account: vf-iedelivery-mgmt - 831341508773
- CI/CD: CodePipeline <https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-transit-gateway--cross-account-prod-tf/view?region=eu-west-1>

These are the key resources for the implementation:

- **Codepipeline** for cross-account terraform code, where there are more than one provider. The code can be found in the repository called "vf-iedelivery-pipelines", where the file "[5-vf-iedelivery-transit-gateway.tf](#)" describes the pipeline associated with the transit gateway infrastructure. The resulting pipeline can be found here: <https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-transit-gateway--cross-account-prod-tf/view?region=eu-west-1>
- **Transit Gateway Infrastructure:** For each VPC that we want to connect to the transit GW, we will create one VPC attachment. Because of availability reasons, we will select 1 subnet per AZ available in the chosen VPC, providing connectivity through the transit GW to the whole AZ. We will follow a cross-account strategy because the vpc's we want to connect are in different accounts. We will create:
 - **In Shared Service Account** (vf-iedelivery-prod-ss-01): Transit GW, Transit GW route tables; AWS RAM resource share; VPC Attachment accepter for test, pre-prod and prod VPC's (until the attachment is available in the target account); VPC Attachment for Shared Service VPC; new route in Shared Service VPC route tables to send traffic to Transit GW.

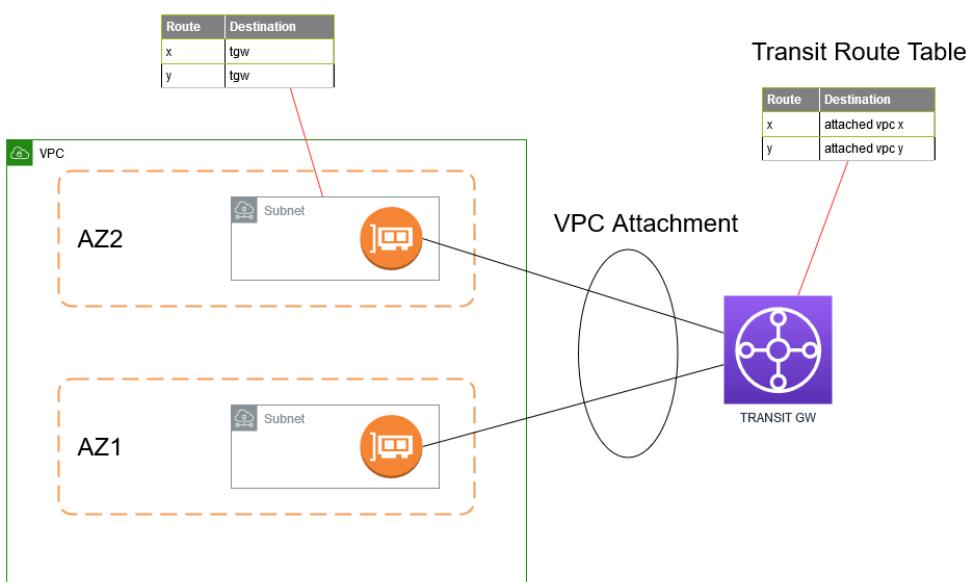
- **Test, Pre-prod and Prod Accounts:** TGW VPC Attachment for test, pre-prod and prod VPC's; shared resource transit GW; new route in each VPC route table to send traffic to Transit GW.

6.1.7.2 2. TGW Design

6.1.7.2.1 2.1 Design of the attachment and route table for a AWS Transit GW.

When you attach a VPC to a transit gateway, you must enable one or more Availability Zones to be used by the transit gateway to route traffic to resources in the VPC subnets. To enable each Availability Zone, you specify exactly one subnet. The transit gateway places a network interface in that subnet using one IP address from the subnet. After you enable an Availability Zone, traffic can be routed to all subnets in that Availability Zone, not just the specified subnet. Resources that reside in Availability Zones where there is no transit gateway attachment will not be able to reach the transit gateway.

Subnet Route Table



Best Practice:

Create a separate subnet for TGW attachment
ENI created in selected subnet

6.1.7.2.2 2.3 Design of the attachments for VF IE VPCs

As specified [here](#), the CIDR for the environments are:

- **10.181.32.0/19** for PROD
- **10.181.64.0/18** for PRE-PROD
- **10.181.128.0/17** for TEST

The requirements for the connection between different environments are:

- Allow communication between different TEST VPC's
- Allow communication between different PRE-PROD VPC's.
- Deny communication to PROD if it does not come from SS VPC.
- SS can communicate with PROD, PRE-PROD and TEST.

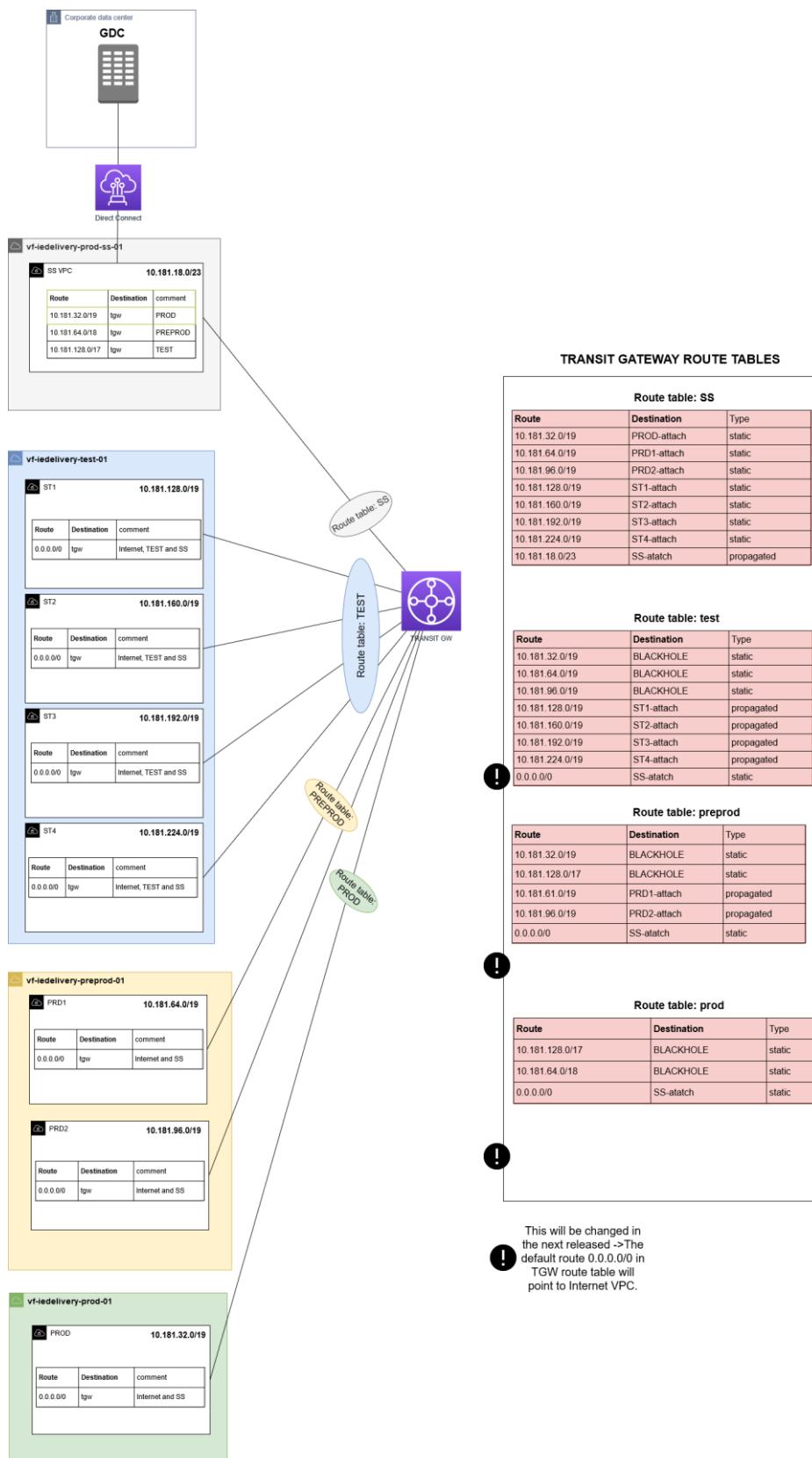
To achieve those requirements, a different **TGW route table** has been created for each environment:

- TGW Route table for **Shared Service VPC**: it has specific routes to send traffic to TEST, PRE-PROD and PROD VPC's.
- TGW Route table for **TEST VPC's**: it has a route to send traffic to SS VPC (static route) and to TEST VPC's (propagated route), but it blocks (blackhole) traffic to PRE-PROD and PROD.
- TGW Route table for **PRE-PROD VPC's**: it has a route to send traffic to SS VPC (static route) and to PRE-PROD VPC's (propagated route), but it blocks (blackhole) traffic to TEST and PROD.
- TGW Route table for **PROD VPC's**: it has a route to send traffic to SS VPC (static route), but it blocks (blackhole) traffic to TEST and PRE-PROD.

With the blackhole routes to drop the traffic that goes to a specific location, we isolate the environments by not allowing communication between test and pre-prod, test and prod, and pre-prod and prod.

Additionally to TGW route tables, to route traffic from VPC's to TGW, a route that points to the TGW needs to be added. For TEST, PRE-PROD and PROD, the destination will be the default route (0.0.0.0/0) for simplicity reasons, as we will implement a centralized internet access in the SS Account. So, traffic that goes either to the internet or to SS VPC, will be sent to the TGW. This route (0.0.0.0/0 → TGW) has to be added to each VPC route table inside a VPC. For SS VPC, the CIDR for the different environments will be set as the destination.

(THIS DESIGN IS NOT THE CURRENT ONE)



6.1.7.3 3. TGW implementation

6.1.7.3.1 3.1 TGW resources

The transit gw infrastructure is defined in the file called transitgw.tf, and these are the resources created:

- **aws_ec2_transit_gateway**: created in ss account, without a default route table and with auto_accept_shared_attachments, vpn_ecmp_support and dns_support enabled.
- **aws_ram_resource_share**, **aws_ram_resource_association** and **aws_ram_principal_association**: created in ss account to share transit gw with other accounts.
- **aws_ec2_transit_gateway_vpc_attachment**: created in each account depending on the vpc. A corresponding vpc accepter is creating while the invitation for the shared resource is pending.
- **aws_ec2_transit_gateway_route_table**: created in ss account.
- **aws_ec2_transit_gateway_route_table_association** and **aws_ec2_transit_gateway_route_table_propagation**: created in ss account.
- **aws_ec2_transit_gateway_route**: created in ss account.
- **aws_route**: created in each account, modifying the VPC route tables of the target vpc.

6.1.7.3.2 3.2 TGW configuration

The variables used to customized the TGW implementation for VFIE has been declared in a file called "variables.tf", in the root folder (repository). The values for the desired configuration are defined in the config file called "PROD.tfvars", stored under the folder "vars". Basically, we are specifying the "vpc-id" for each vpc that we want to associate with the transit gw, the subnets from that vpc where the tgw eni is going to be placed (1 subnet / AZ) and the cidr of the vpc. A different variable has been created for test, pre-prod, prod and ss. The principals to associate with the resource share (TGW) are defined in the variable called ram_principals (test account, pre-prod account and prod account).

```
test_vpc_attachments = {
    "st1" = {
        "vpc_id"      = "vpc-01a98bfaa291c8f7d"
        "subnet_ids"   = ["subnet-00dbd23fbcf0348c0","subnet-03951db996dd1e623"]
        "cidr"        = "10.181.128.0/19"
    },
    "st2" = {
        "vpc_id"      = "vpc-0270246dc82c513fc"
        "subnet_ids"   = ["subnet-0bc39ff0595463183","subnet-063adc8e44098bebe"]
        "cidr"        = "10.181.160.0/19"
    },
    "st3" = {
        "vpc_id"      = "vpc-07bf6d07ae85e68ee"
        "subnet_ids"   = ["subnet-053aab44cdffdbf78","subnet-0e716209201476ebb"]
        "cidr"        = "10.181.192.0/19"
    },
    "st4" = {
        "vpc_id"      = "vpc-070981e1316c4795f"
        "subnet_ids"   = ["subnet-0e89dcb4921397833","subnet-0be50912ce6807467"]
        "cidr"        = "10.181.224.0/19"
    }
}

preprod_vpc_attachments = {
    "prd1" = {
```

```

    "vpc_id"      = "vpc-005d0af4e9a26cf3d"
    "subnet_ids"   = ["subnet-073128e09985a2702","subnet-035edcb630e7407f0"]
    "cidr"        = "10.181.64.0/19"
},
"prd2" = {
    "vpc_id"      = "vpc-0c64b9096fe8d7ef3"
    "subnet_ids"   = ["subnet-04aa54593c7cf4f21","subnet-02d672f19551a8c59"]
    "cidr"        = "10.181.96.0/19"
}
}

prod_vpc_attachments = {
    "prod" = {
        "vpc_id"      = "vpc-0b2285ee1e4c38bae"
        "subnet_ids"   = ["subnet-04769a1c07c576288","subnet-03b0e53501550bd92"]
        "cidr"        = "10.181.32.0/19"
    }
}

ss_vpc_attachments = {
    "vpc_id"      = "vpc-0bc5e297e4fd20219"
    "subnet_ids"   = ["subnet-0f592101bf294734d","subnet-0c77421364cb58f2d"]
}
}

ram_principals = ["046978237480", "612841682649", "323874256692"]

```

6.1.7.4 4. TGW: How to add a new VPC to the implementation

The deployment of the TGW has been accomplished in 2 releases:

1. **RELEASE 1.** SS VPC and TEST VPC's configuration
2. **RELEASE 2.** PRE-PROD abd PROD VPC'S configuration

This section explains the process of the second release to repeat it if any new environment/vpc needs to be added.

6.1.7.4.1 4.1 New AWS account

If the new vpc(s) is in a different aws account than the ones that already are defined, the variable "**ram_principals**" needs to be modified.

For release 1. **ram_principals** = ["046978237480"]

For release 2. Adding pre-prod and prod aws accounts. **ram_principals** = ["046978237480", "612841682649", "323874256692"]

6.1.7.4.2 4.2 New VPC

If it is a new environment (one or more new vpc's to be associated with TGW), create a new variable for the information required to create the vpc attachments.

For release 2, **preprod_vpc_attachments** and **prod_vpc_attachments** were created, with the vpc's details (vpc id, vpc cidr, vpc subnets id).

If the vpc that is going to be added belongs to an environment with this variable (for example, a new prod vpc), then the corresponding variable needs to be updated (for example, **prod_vpc_attachments**).

6.1.7.4.3 4.3 New AWS provider

If the new vpc(s) is in a different aws account than the ones that already are defined, a new aws provider needs to be defined in the file "providers.tf". Different aws providers are needed because there will be resources (tgw vpc attachments, route in vpc route tables) created in

those accounts. The pipeline created to implement the TGW infrastructure will assume the role vf-iedelivery-ci-cd-deploy-role in each account when needed.

For release 1.

```
provider "aws" { alias = "SS" region = "eu-west-1" assume_role { role_arn = "arn:aws:iam::267040142128:role/vf-iedelivery-ci-cd-deploy-role" } }
provider "aws" { alias = "test" region = "eu-west-1" assume_role { role_arn = "arn:aws:iam::046978237480:role/vf-iedelivery-ci-cd-deploy-role" } }
```

For release 2.

```
provider "aws" { alias = "preprod" region = "eu-west-1" assume_role { role_arn = "arn:aws:iam::612841682649:role/vf-iedelivery-ci-cd-deploy-role" } }
provider "aws" { alias = "prod" region = "eu-west-1" assume_role { role_arn = "arn:aws:iam::323874256692:role/vf-iedelivery-ci-cd-deploy-role" } }
```

6.1.7.4.4 4.4 New VPC attachment(s) and VPC accepter

A new TGW VPC attachment (aws_ec2_transit_gateway_vpc_attachment) will be created in the account where the new vpc belongs. Until the VPC attachment status is available, an aws_ec2_transit_gateway_vpc_attachment_accepter is needed in the TGW account (SS account).

For Release 2. Example for pre-prod (same logic for prod):

```
resource "aws_ec2_transit_gateway_vpc_attachment" "preprod_vpc_attachment" {
  provider = "aws.preprod"
  depends_on = ["aws_ram_principal_association.ram_principal",
    "aws_ram_resource_association.ram_assoc"]
  for_each = var.preprod_vpc_attachments
  transit_gateway_id = aws_ec2_transit_gateway.transit_gw.id
  vpc_id      = each.value["vpc_id"]
  subnet_ids   = each.value["subnet_ids"]

  transit_gateway_default_route_table_association = false
  transit_gateway_default_route_table_propagation = false

  tags = merge(
    local.common_tags,
    {
      "Purpose" = "EC2 Transit Gateway attachment to preprod vf ie vpc's"
      "Name"    = join("-',[each.key,"VFIE-Transit-GW-pre-prod-vpc-attachment",
        each.value["vpc_id"]])
    },
  )
}

resource "aws_ec2_transit_gateway_vpc_attachment_accepter" "preprod_vpc_accepter" {
  provider = "aws.SS"
  for_each = var.preprod_vpc_attachments
  transit_gateway_attachment_id =
  aws_ec2_transit_gateway_vpc_attachment.preprod_vpc_attachment[each.key].id

  tags = merge(
    local.common_tags,
    {
      "Purpose" = "EC2 Transit Gateway attachment to preprod vf ie vpc's"
      "Name"    = join("-',[each.key,"VFIE-Transit-GW-pre-prod-vpc-attachment", each.key,
        each.value["vpc_id"]])
    },
  )
}
```

6.1.7.4.5 4.5 TGW Route table

In the design that we are following for VFIE we decided to create one TGW route table per environment (one for test, one for pre-prod and one for prod). If a new environment is added, then a new route table has to be created.

In release 2, prod TGW route table and pre-prod TGW route table were created. Example for pre-prod:

```
resource "aws_ec2_transit_gateway_route_table" "transit_gw_route_table_preprod_vpc" {
  provider = "aws.SS"
  transit_gateway_id = aws_ec2_transit_gateway.transit_gw.id

  tags = merge(
    local.common_tags,
    {
      "Purpose" = "EC2 Transit Gateway Route Table associated to VF IE VPC's preprod"
      "Name"    = "VFIE-Transit-GW-PRE-PROD-Route-Table-vpc"
    },
  )
}
```

6.1.7.4.6 4.6 TGW Route table association and propagation

In both cases where either a new environment is going to be associated to TGW, or a new vpc, a TGW route table association has to be created. This association will link the vpc and the TGW route table. In case we want the vpc to propagate its cidr to the TGW, a TGW route table propagation has to be created.

In release 2, association and propagation between pre-prod and prod route tables and pre-prod and prod vpc's were created. This is an example for pre-prod:

```
resource "aws_ec2_transit_gateway_route_table_association" "preprod_vpc_route_table_association" {
  provider = "aws.SS"
  for_each = var.preprod_vpc_attachments

  transit_gateway_attachment_id =
  aws_ec2_transit_gateway_vpc_attachment.preprod_vpc_attachment[each.key].id
  transit_gateway_route_table_id =
  aws_ec2_transit_gateway_route_table.transit_gw_route_table_preprod_vpc.id
}

resource "aws_ec2_transit_gateway_route_table_propagation" "preprod_vpc_route_table_propagation" {
  provider = "aws.SS"
  for_each = var.preprod_vpc_attachments
  transit_gateway_attachment_id =
  aws_ec2_transit_gateway_vpc_attachment.preprod_vpc_attachment[each.key].id
  transit_gateway_route_table_id =
  aws_ec2_transit_gateway_route_table.transit_gw_route_table_preprod_vpc.id
}
```

6.1.7.4.7 4.7 VPC route tables

All subnets inside the target vpc will need a route to the TGW. To do that, a new route that points to the TGW needs to be added. For VFIE, this route will be the default one in the environment vpc's. In addition, if a new environment is added, in SS a new route to the cidr for that environment will have to be added.

This is the logic for pre-prod:

```
data "aws_route_tables" "rts_prd1" {
```

```

provider = "aws.preprod"
vpc_id = var.preprod_vpc_attachments["prd1"].vpc_id
}

data "aws_route_tables" "rts_prd2" {
  provider = "aws.preprod"
  vpc_id = var.preprod_vpc_attachments["prd2"].vpc_id
}
locals {
  ...
  aws_route_table_prd2_list = tolist(data.aws_route_tables.rts_prd2.ids)
  aws_route_table_prod_list = tolist(data.aws_route_tables.rts_prod.ids)
...
}

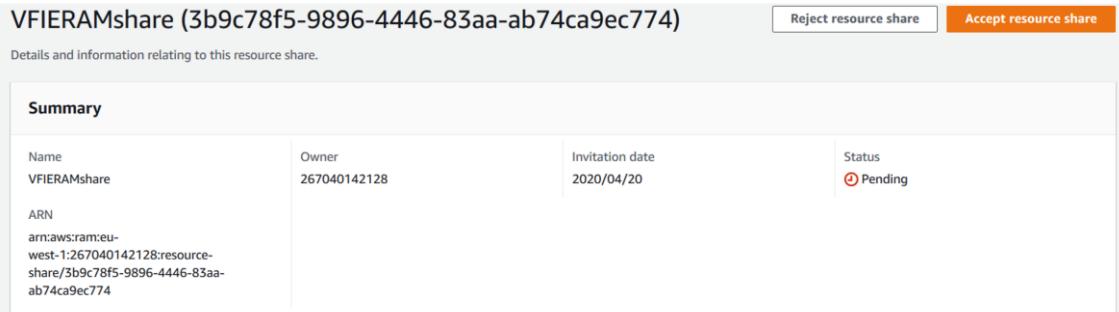
resource "aws_route" "preprod_vpc_route_prd1" {
  provider = "aws.preprod"
  depends_on = ["aws_ram_principal_association.ram_principal",
    "aws_ram_resource_association.ram_assoc"]
  count = length(local.aws_route_table_prd1_list)
  route_table_id      = element(local.aws_route_table_prd1_list, count.index)
  destination_cidr_block  = "0.0.0.0/0"
  transit_gateway_id   = aws_ec2_transit_gateway.transit_gw.id
}
resource "aws_route" "preprod_vpc_route_prd2" {
  provider = "aws.preprod"
  depends_on = ["aws_ram_principal_association.ram_principal",
    "aws_ram_resource_association.ram_assoc"]
  count = length(local.aws_route_table_prd2_list)
  route_table_id      = element(local.aws_route_table_prd2_list, count.index)
  destination_cidr_block  = "0.0.0.0/0"
  transit_gateway_id   = aws_ec2_transit_gateway.transit_gw.id
}

```

6.1.7.4.8 4.8 TGW INVITATION

There is one manual step for the TGW configuration, and it is accepting the TGW as a shared resource. When this is done, the TGW VPC attachments will be created in the account that has accepted the invitation, and once those attachments are available, the TGW VPC accepter can be deleted in SS account.

For release 2.



6.2 01 VF IE SHARED SERVICES VPC

6.2.1

- [1. INTRODUCTION](#)
- [2. DESIGN](#)
- [3. CONFIGURATION AND IMPLEMENTATION](#)
- [4. DNS](#)
- [5. NLB](#)

6.2.2 1. INTRODUCTION

The Shared Services VPC will be the connectivity point between GDC datacenter and VFIE AWS. This VPC is created in the SS account for vf ie (vf-iedelivery-prod-ss-01 - 267040142128). The code has been created in terraform, and it is stored in codecommit, in the management account for VFIE (vf-iedelivery-mgmt - 831341508773). This is the codecommit repository: <https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/vf-iedelivery-sharedservices-vpc>

6.2.3 2. DESIGN

The VPC design uses the CIDR IP Address range given by GDC and associated with this project:

- AZ1 - Dublin:Test Presentation (T-P):60:: 198.19.220.128/26
- AZ2 - Dublin:Test Presentation (T-P):60:: 198.19.220.192/26
- AZ1 - Dublin:Internal Presentation & Application combined (I-A):60:: 198.19.220.0/26
- AZ2 - Dublin:Internal Presentation & Application combined (I-A):60:: 198.19.220.64/26
- AZ1 - Dublin:Management Systems Zone (M2):16:: 46.108.156.0/28
- AZ2 - Dublin:Management Systems Zone (M2):16:: 46.108.156.16/28

This VPC will be composed of the following subnets:

- IA subnets for PROD and PRE-PROD traffic.
- TP subnets for TEST traffic.
- M2 subnets for MGMT traffic (tbc more details).
- Private subnets for other purposes, as ENI created by AWS TGW or SSM endpoints.

PROD:

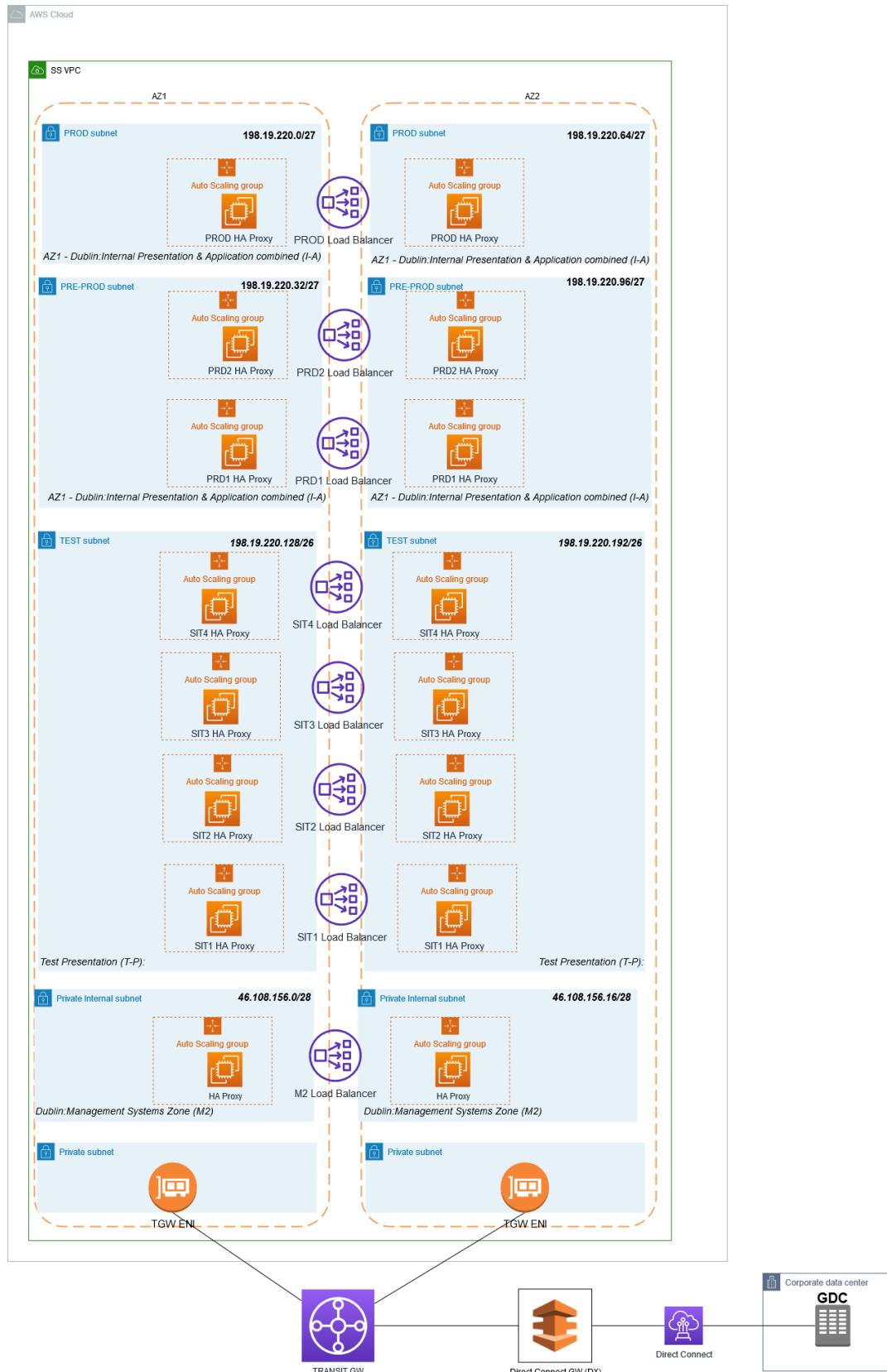
[\["198.19.220.0/27", "198.19.220.64/27"\]](#)

PRE-PROD:

[\["198.19.220.32/27", "198.19.220.96/27"\]](#)

The main idea is to use one NLB per tenant (customer requirement for achieving simplicity and port mapping), where the tenants are: SIT1, SIT2, SIT3 and SIT4 as test tenants; PRD1 and

PRD2 as pre-prod tenants; and PROD as prod tenant. The listener of each NLB will depends on the communications we want to allow between each aws tenant and gcd. A list of ports will be added as an input parameter to a terraform module, in order to create the corresponding NLB listeners and NLB target groups (per port). TPC will be the protocol used. The NLB target will be associated with an AutoScaling group, that will manage the HAProxy infrastructure. See below the diagram:



6.2.4 3. CONFIGURATION AND IMPLEMENTATION

See [here](#)

6.2.5 4. DNS

See [here](#)

6.2.6 5. NLB

There are at the moment 3 NLB created: one for prod, one for prd1 and one for prd2.

| Name | DNS name | State | VPC ID | Availability Zones | Type | Created At |
|-----------------------------------|-----------------------------------|---------------|------------------------------|-------------------------------|----------------|--------------------------------------|
| Fargate-ALB | internal-Fargate-ALB-58545... | active | vpc-07b8d01ebf5d3bcd9 | eu-west-1a, eu-west-1a | application | May 6, 2020 at 12:44:58 PM ... |
| SS-VF-IEDELIVERY-PRD1-NLB | SS-VF-IEDELIVERY-PRD1-N... | active | vpc-073c17ef9ee74423c | eu-west-1a, eu-west-1b | network | June 3, 2020 at 5:26:59 PM ... |
| SS-VF-IEDELIVERY-PRD2-NLB | SS-VF-IEDELIVERY-PRD2-N... | active | vpc-073c17ef9ee74423c | eu-west-1a, eu-west-1b | network | June 11, 2020 at 6:07:00 PM ... |
| SS-VF-IEDELIVERY-PROD-NLB | SS-VF-IEDELIVERY-PROD-... | active | vpc-073c17ef9ee74423c | eu-west-1b, eu-west-1a | network | May 28, 2020 at 11:43:45 A... |
| vfile-delivery-jenkinsfargate-nlb | vfile-delivery-jenkinsfargate-... | active | vpc-07b8d01ebf5d3bcd9 | eu-west-1a, eu-west-1b | network | May 6, 2020 at 11:49:47 AM ... |

Load balancer: SS-VF-IEDELIVERY-PROD-NLB

Description Listeners Monitoring Integrated services Tags

Basic Configuration

Name: SS-VF-IEDELIVERY-PROD-NLB
 ARN: arn:aws:elasticloadbalancing:eu-west-1:267040142128:loadbalancer/net/SS-VF-IEDELIVERY-PROD-NLB/07ed30b5f822b162
 DNS name: SS-VF-IEDELIVERY-PROD-NLB-07ed30b5f822b162.elb.eu-west-1.amazonaws.com (A Record)
 State: active
 Type: network
 Scheme: internal
 IP address type: ipv4
 VPC: vpc-073c17ef9ee74423c
 Availability Zones: subnet-0b2a2317095b5426d - eu-west-1b
 IPv4 address: Assigned from CIDR 198.19.220.32/27
 Private IPv4 address: Assigned from CIDR 198.19.220.32/27

These are the aws nlb domain names:

- [SS-VF-IEDELIVERY-PROD-NLB-07ed30b5f822b162.elb.eu-west-1.amazonaws.com](#)
- [SS-VF-IEDELIVERY-PRD2-NLB-0896a285b4124ae4.elb.eu-west-1.amazonaws.com](#)
- [SS-VF-IEDELIVERY-PRD1-NLB-fff2438b7fa415c1.elb.eu-west-1.amazonaws.com](#)

For each nlb, we will create an ALIAS record:

The screenshot shows the AWS Route 53 console. On the left, a list of 27 DNS records is displayed, including various NS, SOA, A, and CNAME types. On the right, a detailed 'Edit Record Set' dialog is open for a specific record named 'prod.haproxy.leaws.vodafone.com'. The dialog shows the record's name, type (A - IPv4 address), and alias settings. It also includes sections for 'Alias Target', 'Alias Hosted Zone ID', and 'Evaluate Target Health'. A 'Save Record Set' button is at the bottom right.

6.2.7 HAProxy AMI

6.2.7.1 We will need to install HAProxy software in the ec2 instances that we are going to use. Because the SS VPC does not have internet access, we will create a customized AMI for HAProxy using AWS IMAGE BUILDER.

The resources needed for AWS IMAGE BUILDER have been created using CloudFormation, as this is a new service in AWS, and terraform does not offer it yet. However, terraform will manage that CF stack. In the same CF Stack we will be creating also the resources for the OFMW domains used in this project, this is out of scope for this documentation.

6.2.7.2 1. AWS IMAGE BUILDER

Documentation:

- <https://docs.aws.amazon.com/imagebuilder/latest/userguide/image-builder-accessing-prereq.html>
- <https://docs.aws.amazon.com/imagebuilder/latest/userguide/image-builder-setting-up.html>
- <https://docs.aws.amazon.com/imagebuilder/latest/userguide/image-builder-image-deployment-console.html>
- <https://docs.aws.amazon.com/imagebuilder/latest/userguide/what-is-image-builder.html>

6.2.7.2.1 Image pipeline

An image pipeline is the automation configuration for building secure OS images on AWS. The Image Builder image pipeline is associated with an image recipe that defines the build, validation, and test phases for an image build lifecycle. An image pipeline can be associated with an infrastructure configuration that defines where your image is built. You can define attributes, such as instance type, subnets, security groups, logging, and other infrastructure-related configurations. You can also associate your image pipeline with a distribution configuration to define how you would like to deploy your image.

6.2.7.2.2 Image recipe

An Image Builder image recipe is a document that defines the source image and the components to be applied to the source image to produce the desired configuration for the output image. You can use an image recipe to duplicate builds. Image Builder image recipes can be shared, branched, and edited using the console wizard, the AWS CLI, or the API. You can use image recipes with your version control software to maintain shareable versioned image recipes.

6.2.7.2.3 Source image

The source image is the selected image and OS used in your image recipe document along with the components. The source image and the component definitions combined produce the desired configuration for the output image.

6.2.7.2.4 Build components

Build components are orchestration documents that define a sequence of steps for downloading, installing, and configuring software packages. They also define validation and security hardening steps. A component is defined using a YAML document format (as described in the following Document entry).

6.2.7.2.5 Test components

Test components are orchestration documents that define tests to run on software packages. A component is defined using a YAML document format (see the following definition for Document).

6.2.7.2.6 Document

A declarative document that uses the YAML format to list the execution steps for build, validation, and test of an AMI on an instance. The document is input to a configuration management application, which runs locally on an Amazon EC2 instance to execute the document steps.

These are the **steps** to follow:

1. Select source image. You select a source OS image, for example, an existing AMI.
2. Create image recipe. You add components to create an image recipe for your image pipeline. Components are the building blocks that are consumed by an image recipe, for example, packages for installation, security hardening steps, and tests. The selected OS and components make up an image recipe. Components are installed in the order in which they are specified and cannot be reordered after selection.
3. Output. Image Builder creates an OS image in the selected output format.
4. Distribute. You distribute your image to selected AWS Regions after it passes tests in the image pipeline.

The images that you build from the golden image are in your AWS account. You can configure your image pipeline to produce updated and patched versions of your AMI by entering a build schedule. When the build is complete, you can receive notification via [Amazon Simple Notification Service \(SNS\)](#). In addition to producing a final image, Image Builder generates an image recipe that can be used with existing version control systems and continuous integration/continuous deployment (CI/CD) pipelines for repeatable automation. You can share and create new versions of your image recipe.

6.2.7.2.7 PRE-REQUISITES

The following prerequisites must be verified in order to create an image pipeline with EC2 Image Builder.

6.2.7.2.7.1 EC2 Image Builder service-linked role

EC2 Image Builder uses a service-linked role to grant permissions to other AWS services on your behalf. You don't need to manually create a service-linked role. When you create your first Image Builder resource in the AWS Management Console, the AWS CLI, or the AWS API, Image Builder creates the service-linked role for you. For more information about the service-linked role that Image Builder creates in your account, see [Using service-linked roles for EC2 Image Builder](#).

6.2.7.2.7.2 Auto Scaling groups

EC2 Image Builder uses Auto Scaling groups to launch instances during the build and test phases of the image pipeline. When you use Amazon EC2 Auto Scaling, a required service-linked role is created in your account. If this role is not present in your account when you use Image Builder, the Image Builder service-linked role will create it for you.

6.2.7.2.7.3 Configuration requirements

- EC2 Image Builder does not support encrypted AMIs as the source for or output image of a pipeline.
- You must specify a VPC in the infrastructure configuration. Image Builder does not support EC2-Classic.
- Image Builder does not support Amazon VPC endpoints (PrivateLink).
- Instances used to build images and run tests using Image Builder must have access to the Systems Manager service. All build activity is orchestrated by SSM Automation. The SSM Agent will be installed on the source image if it is not already present, and it will be removed before the image is created.

6.2.7.2.7.4 AWS Identity and Access Management (IAM)

The IAM role that you associate with your instance profile must have permissions to run the build and test components included in your image. The following IAM role policies must be attached to the IAM role that is associated with the instance profile: EC2InstanceProfileForImageBuilder and AmazonSSMManagedInstanceCore.

If you configure logging, the instance profile specified in your infrastructure configuration must have s3:PutObject permissions for the target bucket (arn:aws:s3:::**BucketName**/*).

Select an IAM role to associate with the instance profile or Create a new role. If you create a new role, Image Builder will take you to the IAM console. As a starting point, use the following IAM role policies (you must attach both policies): EC2InstanceProfileForImageBuilder and AmazonSSMManagedInstanceCore.

6.2.7.3 2. CLOUDFORMATION STACK

The code is stored under the repository "vf-iedelivery-centralized-internet", as we need to have internet access. The stackset is defined in terraform in the file "5-imagebuilder.tf", and the cloudformation code is in the file "5-imagebuilder-stack-template.yml". TO-DO: automate the parameters that are given to the stackset.

First of all, we upload the ssm document that we will be using to create haproxy ami in a S3 bucket:

```
#uploading files to s3 bucket
resource "aws_s3_bucket_object" "HAPROXY-SSM-DOCUMENT" {
  bucket = "vf-iedelivery-centralized-internet-vpc-267040142128-123456"
  key    = "haproxy-ssm.yml"
  source = "${path.module}/ssm-documents/haproxy.yml"
  etag   = "${filemd5("${path.module}/ssm-documents/haproxy.yml")}"
#triggered changes
}
```

6.2.7.3.1 SSM DOCUMENT

- Under Build:
 - Install haproxy service - AWS package just provide the version 1.5

- Update haproxy version from 1.5 to 1.7, as we need to allow a dns feature. This feature will avoid the default behaviour of haproxy where the services fails if haproxy cannot resolve any of the defined backends. Documentation followed: <https://forums.aws.amazon.com/thread.jspa?threadID=225532> and <https://www.liulangmao.org/?p=5550>.
- Start and enable service
- Logs: HAProxy will emit log message for processing by a **syslog** server (documentation [here](#)).

```

name: haproxy-1.0.1
description: Image for haproxy-1.0.1
schemaVersion: 1.0
phases:
- name: build
  steps:
    - name: InstallPackages
      action: ExecuteBash
      inputs:
        commands:
          - yum install -y haproxy
    - name: UpdateHProxyVersion
      action: ExecuteBash
      inputs:
        commands:
          - yum info haproxy
          - yum install gcc pcre-static pcre-devel openssl-devel -y
          - wget https://www.haproxy.org/download/1.7/src/haproxy-
1.7.12.tar.gz
          - tar xzvf haproxy-1.7.12.tar.gz
          - cd haproxy-1.7.12
          - make TARGET=linux2628 USE_PCRE=1 USE_OPENSSL=1 USE_ZLIB=1
USE_CRYPT_H=1 USE_LIBCRYPT=1
          - make install
          - cp /usr/local/sbin/haproxy /usr/sbin/
          - cp examples/haproxy.init /etc/init.d/haproxy
          - chmod 755 /etc/init.d/haproxy
    - name: EnableAndStart
      action: ExecuteBash
      inputs:
        commands:
          - systemctl enable haproxy
          - systemctl start haproxy || service haproxy start
    - name: Logs
      action: ExecuteBash
      inputs:
        commands:
          - touch /etc/rsyslog.d/haproxy.conf
          - chmod 777 /etc/rsyslog.d/haproxy.conf
          - cat > /etc/rsyslog.d/haproxy.conf << 'EOF'
          - |
            # Collect log with UDP
            $ModLoad imudp
            $UDPServerAddress 127.0.0.1
            $UDPServerRun 514
            # Creating separate log files based on the severity
            local0.* /var/log/haproxy-traffic.log
            local0.notice /var/log/haproxy-admin.log
            EOF
          - systemctl restart rsyslog
    - name: update
      steps:
        - name: UpdateSoftware
          action: ExecuteBash
          inputs:
            commands:
              - yum update

```

6.2.7.3.2 CF STACK definition in terraform.

TODO: add parameters to stackset.

```

resource "aws_cloudformation_stack" "image_builder_stack" {
  depends_on = [aws_s3_bucket_object.OFMW-1213-SSM-DOCUMENT,
    aws_s3_bucket_object.OFMW-1221-SSM-DOCUMENT, aws_s3_bucket_object.HAPROXY-
    SSM-DOCUMENT, aws_s3_bucket_object.OFMW-BASE-SSM-DOCUMENT ]
  name = "${var.VPC_NAME}-${var.PROJECT}-Image-Builder-Stack-${var.ENV}"
  #iam_role_arn = "arn:aws:iam::267040142128:role/DevOps"
  #parameters = {

    #OFMW1213DOCUMENT = "https://vf-iedelivery-centralized-internet-vpc-
    267040142128-123456.s3-eu-west-1.amazonaws.com/OFMW_1213.yml"

  }
  template_body = "${file("${path.module}/5-imagebuilder-stack-
  template.yml")}"
  capabilities = ["CAPABILITY_IAM",]
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "Cloudformation stack for transparent proxies in
      Internet VPC"
    },
  )
}

```

6.2.7.3.3 CLOUDFORMATION CODE

Each time the ssm document changed, the version of ImageComponenthaproxy and ImageRecipehaproxy need to be updated. TODO: automate this.

```

AWSTemplateFormatVersion: '2010-09-09'
Description: 'VFIE delivery Image Builder'
Parameters:
  ...
  haproxyDOCUMENT:
    Type: String
    Description: Uri of SSM document for haproxy uploaded to S3 bucket
    Default: "s3://vf-iedelivery-centralized-internet-vpc-267040142128-
123456/haproxy-ssm.yml"
  SubnetId:
    Type: String
    Description: Public Subnet ID for the ec2 instance that is created by
image builder
    Default: "subnet-09a0b331473b23229"
  SecurityGroupId:
    Type: String
    Description: SSM Security group ID
    Default: "sg-064be9eac2ccd002c"
  InstanceProfileName:
    Type: String
    Description: Instance profile to associate with the ec2 instance that
is created by image builder
    Default: "SSM-EC2-CENTRALIZED-INTERNET-IAM-INSTANCE-PROFILE-vfie-
delivery-PROD"
  AMIid:
    Type: String
    Description: parent AMI Id for OFMW
    Default: "ami-05b44559c008e9b05"
  HAProxyAMIid:
    Type: String
    Description: parent AMI Id for HAProxy
    Default: "ami-01726b7eb435f48b5"
Resources:
  ...
  ######HAPROXY
  SNSTopichaproxy:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint: "albamaria.diazfernandez@vodafone.com"
          Protocol: "email"
      TopicName: "VFIEdelivery-AMI-haproxy"  ImageComponenthaproxy:
    Type: AWS::ImageBuilder::Component
    Properties:
      ChangeDescription: 'fifth version'
      Description: 'VFIE delivery haproxy SSM DOCUMENT'
      Name: 'VFIE-DELIVERY-haproxy-BUILD'
      Platform: 'Linux'
      Version: '1.0.5'
      Uri: !Ref haproxyDOCUMENT

  ImageRecipehaproxy:
    Type: AWS::ImageBuilder::ImageRecipe
    Properties:
      Components:
        - ComponentArn: !GetAtt ImageComponenthaproxy.Arn
      Description: 'VFIE delivery Image recipe'
      Name: 'VFIE-DELIVERY-haproxy'
      ParentImage: !Ref HAProxyAMIid
      Version: '1.0.5'

  ImagePipelinehaproxy:
    Type: AWS::ImageBuilder::ImagePipeline

```

```

Properties:
  Description: 'VFIE delivery haproxy Image pipeline'
  #DistributionConfigurationArn: String
  ImageRecipeArn: !GetAtt ImageRecipehaproxy.Arn
  #ImageTestsConfiguration:
  #  ImageTestsConfiguration
  InfrastructureConfigurationArn: !GetAtt ImageInfraConfighaproxy.Arn
  Name: 'VFIE-DELIVERY-haproxy'
  #Schedule:
  #  Schedule
  Status: 'ENABLED'
  #Tags:
  #  Key : Value

ImageInfraConfighaproxy:
  Type: AWS::ImageBuilder::InfrastructureConfiguration
  Properties:
    Description: 'VFIE delivery Image haproxy infrastructure config'
    InstanceProfileName: !Ref InstanceProfileName
    InstanceTypes:
      - 't2.medium'
    #KeyPair: String
    Logging:
      S3Logs:
        S3BucketName: 's3-access-logs-vf-iedelivery-267040142128-logs'
        S3KeyPrefix: 'haproxylogs'
    Name: 'VFIE-DELIVERY-haproxy-IMAGE-INFRA'
    SecurityGroupIds:
      - !Ref SecurityGroupId
      - sg-0b85489e5b93501d8
    SnsTopicArn: !Ref SNStopichaproxy
    SubnetId: !Ref SubnetId
    #Tags:
    #  Key : Value
    #TerminateInstanceOnFailure: Boolean

Imagehaproxy:
  Type: AWS::ImageBuilder::Image
  Properties:
    ImageRecipeArn: !GetAtt ImageRecipehaproxy.Arn
    InfrastructureConfigurationArn: !GetAtt ImageInfraConfighaproxy.Arn
    Tags:
      Name : "Image-haproxy"

Outputs:
...
Imagehaproxyarn:
  Description: The ARN of the created AMI for haproxy
  Value: !Ref Imagehaproxy

```

6.2.8 HAProxy Config Automation

- [CONFIGURATION](#)
 - [Config Variable](#)
 - [Config File](#)
 - [RESULT](#)
- [S3 SYNCHRONIZATION](#)
 - [LAMBDA CODE](#)
 - [CLOUDWATCH EVENT RULE](#)
 - [RESULTS](#)

This section describes the automation provided by PCS to ensure that the HAProxy will always have the latest configuration. The config file is created by terraform. Because of the complexity of this project and its network design, variables as a single source of truth are used. Each environment will have its own variable for defining the haproxy config. The variable type is a dictionary, with as many objects as ports the haproxy is listening to. Each port configuration has the following attributes: whitelist, port and server.

6.2.8.1 CONFIGURATION

The terraform code is stored in the repository "vf-iedelivery-sharedservices-vpc" in vfie mgmt. account:<https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-sharedservices-vpc/browse?region=eu-west-1>

6.2.8.1.1 Config Variable

In the root directory, we can see the variables declared:

```

169 #-----HAProxy CONFIG-----
170 variable "prod_haproxy_config" {
171   description = "map of ports and whitelist for PROD HAproxy config. "
172   type        = any
173   default     = {}
174 }
175
176 variable "prd1_haproxy_config" {
177   description = "map of ports and whitelist for PRD1 HAproxy config. "
178   type        = any
179   default     = {}
180 }
181
182 variable "prd2_haproxy_config" {
183   description = "map of ports and whitelist for PRD2 HAproxy config. "
184   type        = any
185   default     = {}
186 }
```

Under the [folder vars](#) we can see the file PROD.tfvar with the values of haproxy_config variables. Here it's the example for prd2:

```

prd2_haproxy_config = {
    "aws-1" = {
        "port"      = 15101,
        "whitelist" = "10.181.96.0/19",
        "server"    = "esb.app.prd2.equinor.vf-
ie.internal.vodafone.com:32007"
    },
    "aws-2" = {
        "port"      = 12101,
        "whitelist" = "10.181.96.0/19",
        "server"    = "meh.prd1.equinor.vf-
ie.internal.vodafone.com:22"
    },
    "aws-3" = {
        "port"      = 12103,
        "whitelist" = "10.181.96.0/19",
        "server"    = "meh.prd1.equinor.vf-
ie.internal.vodafone.com:16443"
    },
    "aws-4" = {
        "port"      = 22102,
        "whitelist" = "10.181.96.0/19",
        "server"    = "iebpoihr-vip.dc-dublin.de:33001"
    },
    "aws-5" = {
        "port"      = 22103,
        "whitelist" = "10.181.96.0/19",
        "server"    = "iebpojhr-vip.dc-dublin.de:33001"
    },
    "aws-6" = {
        "port"      = 22104,
        "whitelist" = "10.181.96.0/19",
        "server"    = "iebpokhr-vip.dc-dublin.de:33001"
    },
    "gdc-1" = {
        "port"      = 7001,
        "whitelist" = "37.25.160.19/32 10.74.120.112/28",
        "server"    = "cch-sal-alb.prd2.ieaws.vodafone.com:7001"
    },
    "gdc-2" = {
        "port"      = 8011,
        "whitelist" = "10.109.100.98/32 10.109.100.99/32
10.109.100.184/32 10.109.100.185/32 10.109.100.133/32 10.109.100.134/32
10.109.100.182/32 10.109.100.183/32 10.109.100.142/32 10.109.100.143/32
10.109.100.144/32 10.109.100.145/32 10.109.100.146/32 10.109.100.147/32
10.109.100.148/32 10.109.100.36/32 10.109.100.37/32 37.25.160.19/32",
        "server"    = "cch-sal-alb.prd2.ieaws.vodafone.com:8011"
    },
    "gdc-3" = {
        "port"      = 5500,
        "whitelist" = "37.25.160.19/32 10.74.120.112/28",
        "server"    = "cch-sal-alb.prd2.ieaws.vodafone.com:5500"
    },
    "gdc-4" = {
        "port"      = 5501,
        "whitelist" = "37.25.160.19/32 10.74.120.112/28",
        "server"    = "cch-sal-alb.prd2.ieaws.vodafone.com:5501"
    },
    "gdc-5" = {
        "port"      = 10120,
        "whitelist" = "198.18.65.3/32 198.18.65.4/32
198.18.65.5/32 198.18.65.6/32 198.18.74.200/32 198.18.74.219/32
10.109.100.184/32 10.109.100.185/32 10.109.100.133/32 10.109.100.134/32
10.109.100.182/32 10.109.100.183/32 10.109.100.142/32 10.109.100.143/32
10.109.100.144/32 10.109.100.145/32 10.109.100.146/32 10.109.100.147/32
10.109.100.148/32 10.109.100.36/32 10.109.100.37/32 37.25.160.19/32",
        "server"    = "cch-sal-alb.prd2.ieaws.vodafone.com:10120"
    }
}

```

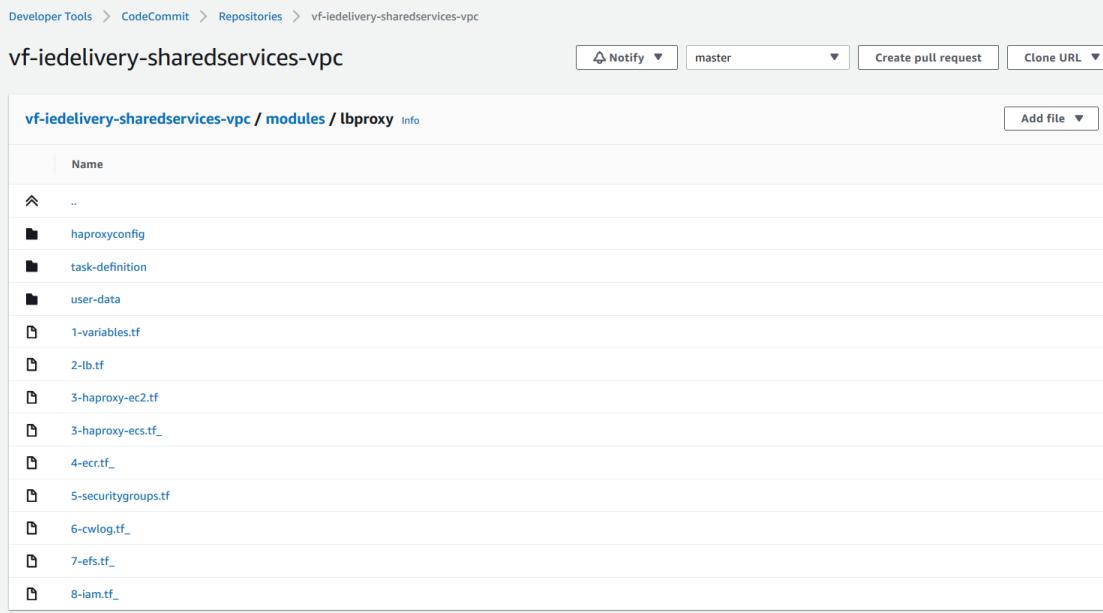
```

10.109.100.148/32 10.109.100.36/32 10.109.100.37/32 10.109.100.118/32
10.109.100.119/32 198.18.65.31/32 37.25.160.19/32",
    "server"      = "cch-sal-sftp.prd2.ieaws.vodafone.com:22"
}
}

```

6.2.8.1.2 Config File

The config file is defined inside the haproxy module: <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-sharedservices-vpc/browse/refs/heads/master/~/modules/lbproxy?region=eu-west-1>



Key files/folders:

- haproxyconfig folder
- user-data
- 3-haproxy-ec2.tf
- 5-securitygroups.tf

In order to create dynamically the haproxy file in terraform, we have used the terraform function "templatefile(path, vars)". Terraform documentation [here](#). It reads the file at the given path and renders its content as a template using a supplied set of template variables. The template syntax is the same as for string templates in the main terraform language, including interpolation sequences delimited with \${}. In the folder haproxyconfig we can see the templates used, called haproxy-global.tmpl and haproxy.tmpl. The first one is to set the global and default variables of the haproxy config. The second template is used as a template for each port configuration.

vf-iedelivery-sharedservices-vpc / modules / lbproxy / haproxyconfig Info

| Name |
|--------------------|
| .. |
| haproxy-global.tpl |
| haproxy-PRD1.cfg |
| haproxy-PRD2.cfg |
| haproxy-PROD.cfg |
| haproxy.tpl |

Note: At the moment, haproxy is ready the configuration from the hardcoded config files (*.cfg). This will be deleted once we migrate to the automated config file.

Global and default variables template file:

```
#HAPROXY CONFIG

global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0
defaults
    log           global
    option        tcplog
    mode          tcp
    retries       3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client ${timeout_client}
    timeout server ${timeout_server}
    default-server init-addr none #to avoid the config failing if it cannot
resolver one dns server name#END OF GLOBAL AND DEFAULT VARIABLES
```

Port template

```
#configuration for port: ${port}
frontend prod_${port}
    bind *:${port}
    acl white_list src ${whitelist}
    tcp-request connection reject if !white_list
    default_backend gcd_${port}

backend gcd_${port}
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server server_name ${server}
```

In the file where we defined our haproxy, 3-haproxy-ec2.tf, we created the following local variables:

```
locals{
  ..
  template_text_backend = [ for p in var.port_listeners:
    templatefile("${path.module}/haproxyconfig/haproxy tmpl", { port =
      p["port"], whitelist = p["whitelist"], server = p["server"] })
    template_text_global =
    templatefile("${path.module}/haproxyconfig/haproxy-global.tmpl",
    {timeout_client = "60s", timeout_server = "60s"})
    template_text_join = join("\n", local.template_text_backend)
    template_text = join("\n", [local.template_text_global,
    local.template_text_join])
}
```

Where we create a variable with the content (text) corresponding with the configuration of all the ports that the haproxy is going to be listened to. The variable var.port_listener has the value of our haproxy config (port_listeners = var.prod_haproxy_config). We will join all port configuration with the global configuration in the local variable template_text.

We will push that content to s3:

```
resource "aws_s3_bucket_object" "haproxy-tmpl" {
  bucket = "vf-iedelivery-centralized-internet-vpc-267040142128-123456"
  key    = "HAPROXY-${var.nlb_name}//haproxy-template.tmpl"
  content= local.template_text
}
```

Note: Right now the file is stored as ".tmpl" as haproxy are still reading the old-way hardcoded configuration.

6.2.8.1.3 RESULT

Amazon S3 > vf-iedelivery-centralized-internet-vpc-267040142128-123456 > HAPROXY-PRD1

vf-iedelivery-centralized-internet-vpc-267040142128-123456

Overview

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions Versions Hide Show

Name

haproxy-template.tpl

haproxy.cfg

For prd2:

```

#HAPROXY CONFIG

global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0
defaults
    log           global
    option        tcplog
    mode          tcp
    retries       3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr none #to avoid the config failing if it cannot
resolver one dns server name
#END OF GLOBAL AND DEFAULT VARIABLES

#configuration for port: 15101
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_15101

backend gcd_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server server_name esb.app.prd2.equinox.vf-
ie.internal.vodafone.com:32007

#configuration for port: 12101
frontend prod_12101
    bind *:12101
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_12101

backend gcd_12101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server server_name meh.prd1.equinox.vf-ie.internal.vodafone.com:22

#configuration for port: 12103
frontend prod_12103
    bind *:12103
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_12103

backend gcd_12103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server server_name meh.prd1.equinox.vf-ie.internal.vodafone.com:16443

#configuration for port: 22102
frontend prod_22102
    bind *:22102
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_22102

backend gcd_22102
    option tcp-check # perform a simple TCP check of healthiness against the
server

```

```

server server_name iebpoihr-vip.dc-dublin.de:33001

#configuration for port: 22103
frontend prod_22103
  bind *:22103
  acl white_list src 10.181.96.0/19
  tcp-request connection reject if !white_list
  default_backend gdc_22103

backend gdc_22103
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server server_name iebpojhr-vip.dc-dublin.de:33001

#configuration for port: 22104
frontend prod_22104
  bind *:22104
  acl white_list src 10.181.96.0/19
  tcp-request connection reject if !white_list
  default_backend gdc_22104

backend gdc_22104
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server server_name iebpokhr-vip.dc-dublin.de:33001

#configuration for port: 7001
frontend prod_7001
  bind *:7001
  acl white_list src 37.25.160.19/32 10.74.120.112/28
  tcp-request connection reject if !white_list
  default_backend gdc_7001

backend gdc_7001
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server server_name cch-sal-alb.prd2.ieaws.vodafone.com:7001

#configuration for port: 8011
frontend prod_8011
  bind *:8011
  acl white_list src 10.109.100.98/32 10.109.100.99/32 10.109.100.184/32
  10.109.100.185/32 10.109.100.133/32 10.109.100.134/32 10.109.100.182/32
  10.109.100.183/32 10.109.100.142/32 10.109.100.143/32 10.109.100.144/32
  10.109.100.145/32 10.109.100.146/32 10.109.100.147/32 10.109.100.148/32
  10.109.100.36/32 10.109.100.37/32 37.25.160.19/32
  tcp-request connection reject if !white_list
  default_backend gdc_8011

backend gdc_8011
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server server_name cch-sal-alb.prd2.ieaws.vodafone.com:8011

#configuration for port: 5500
frontend prod_5500
  bind *:5500
  acl white_list src 37.25.160.19/32 10.74.120.112/28
  tcp-request connection reject if !white_list
  default_backend gdc_5500

backend gdc_5500
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server server_name cch-sal-alb.prd2.ieaws.vodafone.com:5500

```

```

#configuration for port: 5501
frontend prod_5501
    bind *:5501
    acl white_list src 37.25.160.19/32 10.74.120.112/28
    tcp-request connection reject if !white_list
    default_backend gdc_5501

backend gdc_5501
    option tcp-check # perform a simple TCP check of healthiness against the
    server
    server server_name cch-sal-alb.prd2.ieaws.vodafone.com:5501

#configuration for port: 10120
frontend prod_10120
    bind *:10120
    acl white_list src 198.18.65.3/32 198.18.65.4/32 198.18.65.5/32
    198.18.65.6/32 198.18.74.200/32 198.18.74.219/32 10.109.100.184/32
    10.109.100.185/32 10.109.100.133/32 10.109.100.134/32 10.109.100.182/32
    10.109.100.183/32 10.109.100.142/32 10.109.100.143/32 10.109.100.144/32
    10.109.100.145/32 10.109.100.146/32 10.109.100.147/32 10.109.100.148/32
    10.109.100.36/32 10.109.100.37/32 10.109.100.118/32 10.109.100.119/32
    198.18.65.31/32 37.25.160.19/32
    tcp-request connection reject if !white_list
    default_backend gdc_10120

backend gdc_10120
    option tcp-check # perform a simple TCP check of healthiness against the
    server
    server server_name cch-sal-sftp.prd2.ieaws.vodafone.com:22

```

6.2.8.2 S3 SYNCHRONIZATION

Because our configuration is stored in a s3 bucket, it is difficult to automate the update into ec2. In ec2 user data, we create the following script (depending on the environment):

```

sudo cat > /etc/haproxy/haproxy-conf-refresh.sh << 'EOF'
sudo mv /etc/haproxy/haproxy.cfg{,.original}
sudo aws s3 sync s3://vf-iedelivery-centralized-internet-vpc-267040142128-123456/HAPROXY-
PRD1 /etc/haproxy/s3
sudo mv /etc/haproxy/s3/haproxy.cfg /etc/haproxy/haproxy.cfg
sudo systemctl restart haproxy
EOF

```

The goal is to automate the execution of this script in the ec2 instance when the s3 object with the configuration for the haproxy changes. Different ways of doing this:

- Using terraform null resource
- Using lambda function

The first trial was using terraform null resource, that will try to connect through ssh to haproxy and run a script that you define. It will be something like this:

```

resource "null_resource" "haproxy"{
  # Changes to any instance of the s3 object requires re-provisioning
  depends_on = [aws_autoscaling_group.haproxy_AG]
  count = length(data.aws_instances.haproxy.private_ips)
  triggers = {
    s3_object_etag = "${filemd5("${path.module}/haproxyconfig/${local.config_file_name}")}"
  }
  #triggered changes
}

```

```

connection {
  host = "${element(data.aws_instances.haproxy.public_ips, count.index)}"
}

provisioner "remote-exec" {
  # Bootstrap script called with private_ip of each node in the cluster
  inline = [
    "sudo /etc/haproxy/haproxy-conf-refresh.sh",
  ]
}
}

```

However, our haproxy are in a private subnet, and we will need a bastion host to connect to them using ssh. That's why this solution was not implemented.

The second one is based on ssm (in order to avoid ssh). A lambda function and a cloudwatch event rule was created. We can find the code in the root directory of the repository, under the folder "code". Lambda function and cw event has been defined in the file "4-lambda.tf"

6.2.8.2.1 LAMBDA CODE

It reads haproxy instance ids passing as a parameter the value for the Tag Name. Then, it uses ssm send command, passing that instance id and setting the document name to "AWS-RunShellScript".

```

import pprint
import boto3
import json
import os
session = boto3.Session(region_name="eu-west-1")

def get_instance_id(ec2_name):
    ec2_client = session.client('ec2')
    filters = [ {'Name': 'tag:Name',
                 'Values': [ ec2_name ]},
                ]
    response =
    ec2_client.describe_instances(Filters=filters) ["Reservations"]
    instanceid = []
    for r in response:
        for instance in r['Instances']:
            instanceid.append(instance['InstanceId'])
    print(instanceid)
    return instanceid

def run_ssm_command(Instanceids, command):
    ssm_client = session.client('ssm')

    response = ssm_client.send_command(
        InstanceIds= Instanceids ,
        DocumentName="AWS-RunShellScript",
        Parameters={
            'commands':[ command
                         ],
                     },
        )
    command_id = response['Command'][ 'CommandId']

    print(command_id)
def handler(event, context):

    #GET INSTANCE ID
    haproxy_prd1_instanceIds = get_instance_id("AG-PRD1-HAPROXY")
    haproxy_prd2_instanceIds = get_instance_id("AG-PRD2-HAPROXY")
    haproxy_prod_instanceIds = get_instance_id("AG-PROD-HAPROXY")

    #RUN SSM COMMAND
    run_ssm_command(haproxy_prd1_instanceIds, "/etc/haproxy/haproxy-conf-
refresh.sh")
    run_ssm_command(haproxy_prd2_instanceIds, "/etc/haproxy/haproxy-conf-
refresh.sh")
    run_ssm_command(haproxy_prod_instanceIds, "/etc/haproxy/haproxy-conf-
refresh.sh")

```

You can see lambda function [here](#)

6.2.8.2.2 CLOUDWATCH EVENT RULE

A cloudwatch event rule has been created to invoke lambda function when a change is done to a defined s3 bucket (where haproxy config is stored).

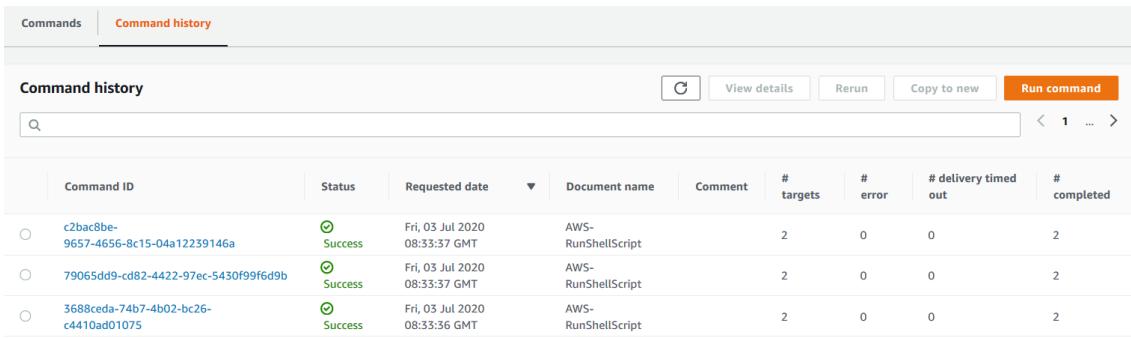
```

resource "aws_cloudwatch_event_rule" "s3_bucket_modification" {
  name          = "haproxy-config-change-rule"
  description   = "Capture changes to haproxy configuration"
  event_pattern = <>PATTERN
{
  "source": [
    "aws.s3"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3.amazonaws.com"
    ],
    "eventName": [
      "PutObject",
      "CompleteMultipartUpload",
      "DeleteObject",
      "DeleteObjects",
      "CreateMultipartUpload",
      "PutObjectAcl",
      "PutObjectLockLegalHold",
      "PutObjectLockRetention",
      "RestoreObject",
      "UploadPart",
      "UploadPartCopy",
      "SelectObjectContent"
    ],
    "requestParameters": {
      "bucketName": [
        "vf-iedelivery-centralized-internet-vpc-267040142128-123456"
      ]
    }
  }
}
PATTERN
}
resource "aws_cloudwatch_event_target" "lambda" {
  rule      = "${aws_cloudwatch_event_rule.s3_bucket_modification.name}"
  target_id = "SendToLambda"
  arn       = "${aws_lambda_function.gdc_pcs_lambda_sync_haproxy_s3.arn}"
}
resource "aws_lambda_permission" "allow_cloudwatch" {
  statement_id = "AllowExecutionFromCloudWatch"
  action       = "lambda:InvokeFunction"
  function_name =
"${aws_lambda_function.gdc_pcs_lambda_sync_haproxy_s3.function_name}"
  principal    = "events.amazonaws.com"
  source_arn   = aws_cloudwatch_event_rule.s3_bucket_modification.arn
}

```

6.2.8.2.3 RESULTS

In ssm console we can see the history for the run commands:



The screenshot shows the AWS CloudWatch Command history interface. At the top, there are tabs for 'Commands' and 'Command history', with 'Command history' being the active tab. Below the tabs is a search bar and a set of buttons: 'View details', 'Rerun', 'Copy to new', and 'Run command'. The main area displays a table of command history entries. The columns are: Command ID, Status, Requested date, Document name, Comment, # targets, # error, # delivery timed out, and # completed. There are three entries listed:

| Command ID | Status | Requested date | Document name | Comment | # targets | # error | # delivery timed out | # completed |
|--------------------------------------|---------|-------------------------------|--------------------|---------|-----------|---------|----------------------|-------------|
| c2bac8be-9657-4656-8c15-04a12239146a | Success | Fri, 03 Jul 2020 08:33:37 GMT | AWS-RunShellScript | | 2 | 0 | 0 | 2 |
| 79065dd9-cd82-4422-97ec-5430f99f6d9b | Success | Fri, 03 Jul 2020 08:33:37 GMT | AWS-RunShellScript | | 2 | 0 | 0 | 2 |
| 3688ceda-74b7-4b02-bc26-c4410ad01075 | Success | Fri, 03 Jul 2020 08:33:36 GMT | AWS-RunShellScript | | 2 | 0 | 0 | 2 |

6.2.9 HAProxy-EC2 IMPLEMENTATION

6.2.9.1

Error rendering macro 'toc'

**java.lang.RuntimeException: com.ctc.wstx.exc.WstxUnexpectedCharException:
Unexpected character ';' (code 59) expected '=' at [row,col {unknown-source}]: [51,297]**

6.2.9.2 1. Module Inputs

This section will explain what has been deployed in SS VPC. Following the design showed [here](#), we have a part of the architecture that follows the same pattern: NLB, AutoScaling group and HAproxy. In order to automate and make efficient the deployment and management, a terraform module has been created. This module is called '[lbproxy](#)', and will have the following inputs:

- Tags
- VPC name
- VPC cidr block
- **NLB name to differentiate AWS tenants. IMPORTANT: this name will be used in the module to look for the config name of the HAProxy and user data.**
- A boolean variable called 'is_prod' / 'is_test' / 'is_preprod' to true. - deprecated
- A list of subnets where the NLB will be available and the HAProxy will be deployed.
- VPC id
- A dictionary with the ports where we expect the traffic. The key will differentiate the origin of the traffic (aws or gcd) and the value will be the port number.
- SSM security group - needs to be updated

```

module "prod-nlHaproxy" {
  source          = "./modules/lbproxy"
  TAGS           = local.common_tags
  ENV            = var.ENV
  PROJECT        = var.PROJECT
  VPC_NAME       = var.VPC_NAME
  main_cidr_block = aws_vpc.vpc.cidr_block
  nlb_name       = "PROD"
  is_prod         = true
  subnets         = [aws_subnet.subnet_IA_tier[0].id,
                    aws_subnet.subnet_IA_tier[1].id ]
  vpc_id          = aws_vpc.vpc.id
  ssm_security_group = aws_security_group.ssm_endpoint_sg.id
  ssm_instance_profile = aws_iam_instance_profile.ec2_ssm.id
  port_listeners   = { "aws-1" : 15101,
                        "aws-2" : 20101,
                        "aws-3" : 20111,
                        "aws-4" : 20112,
                        "aws-5" : 20113,
                        "aws-6" : 20114,
                        "aws-7" : 12101,
                        "aws-8" : 12103,
                        "aws-9" : 22102,
                        "aws-10" : 22103,
                        "aws-11" : 22104,
                        "gdc-1" : 7001,
                        "gdc-2" : 8011,
                        "gdc-3" : 5500,
                        "gdc-4" : 5501,
                        "gdc-5" : 10120
                      }
}

```

6.2.9.3 2. NLB

The nlb architecture will be defined in the file '[2-lb.tf](#)'. An **internal network load balancer** will be deployed, with the input subnets associated with. The name convention is "\${var.VPC_NAME}-\${var.TAGS["Project"]}-\${var.nlb_name}-NLB"

One **target group** per port will be created, using a 'for_each' expression, in order to differentiate the target groups with a key. The name convention is \${var.VPC_NAME}-\${var.TAGS["Project"]}-\${var.nlb_name}-\${each.value}. The protocol will be 'TCP' and the target type will be **'instance'**, as we are using ec2 instances for HAProxy. For EC2 instance, instead of using a map variable for the ports, we could use just a list, and use "count" expression in the target group resource. However, the goal is to migrate to ECS Fargate when possible, and with ECS fargate it is not possible to use count in target group (as you cannot associate later on ecs service and target group in the dynamic block load balancer). One **lb listener** per port will be created, using a 'for_each' expression (same reasons as above explained). Each listener will have a forward action to the corresponding target group (same port).

For the health check, the configuration right now is using port 443, as we still don't have gcd connectivity, and we will find problems when trying to check traffic port. This will be modified when gcd connectivity is ready.

```

#-----LB-----
resource "aws_lb" "nlb" {
  name          = "${var.VPC_NAME}-${var.TAGS["Project"]}-${var.nlb_name}-NLB"
  load_balancer_type = "network"
  internal        = true
  subnets         = var.subnets
  enable_cross_zone_load_balancing = true
  enable_deletion_protection      = true

  tags = merge(
    var.TAGS,
    {
      "Purpose" = "NLB for SS VPC"
      "Name"     = "${var.VPC_NAME}-${var.TAGS["Project"]}-${var.nlb_name}-NLB"
    },
  )
}

```

For the target groups, I found dependency issues when updating ports. This will force me to delete ALL target groups and CREATE them again. This is the issue terraform experiences: Error deleting Target Group: ResourceInUse: Target group 'xxxx' is currently in use by a listener or a rule status code: 400

This could be solved using name_prefix instead of name, but the number of character for that attribute is just 6, what will cause confusions as we had prod, pre-prod and test nlb target groups. Following the [issue opened in terraform](#), the solution is to define a random string and add it to the name of the target group:

"As you're seeing, the AWS API doesn't allow a Target Group to be deleted when there's a Listener Rule attached to it. When a port number on the Target Group is changed, it forces the Target Group to be re-created. Unfortunately there is currently no way in Terraform to say "also delete this resource when another resource is deleted", but we have [some thoughts about how to address it](#). In addition to the name argument, aws_alb_target_group supports name_prefix, and will also generate a random name if neither of those is supplied. As suggested in [#636 \(comment\)](#), the name can be set in a Name tag instead. Because of how name_prefix works and the length limit on Target Group names, the prefix can only be six characters. Some of the other solutions using the random provider to add uniqueness to the names will work as well. Setting lifecycle { create_before_destroy = true } is also needed to break the dependency cycle between the resources."

```

resource "random_string" "target_group" {
  length = 4
  special = false
}

resource "aws_lb_target_group" "ecs_nlb_tg" {
  for_each = var.port_listeners
  name      = "${var.VPC_NAME}-${var.nlb_name}-${each.value}-
${random_string.target_group.result}"
  vpc_id    = var.vpc_id
  port      = each.value
  protocol  = "TCP"
  target_type = "instance"
  deregistration_delay = 60

  health_check {
    interval = 30
    protocol = "TCP"
    port      = 443
    healthy_threshold = 2
    unhealthy_threshold = 2
  }

  tags = merge(
    var.TAGS,
    {
      "Purpose" = "PROD NLB Target group for SS VPC"
      "Name"     = "${var.VPC_NAME}-${var.TAGS["Project"]}-${var.nlb_name}-
${each.value}"
    },
  )

  lifecycle {
    create_before_destroy = true
    ignore_changes = ["name"]
  }
}

resource "aws_lb_listener" "frontend" {
  depends_on = [aws_lb_target_group.ecs_nlb_tg]
  for_each = var.port_listeners
  load_balancer_arn = aws_lb.nlb.arn
  port      = each.value
  protocol  = "TCP"
  default_action {
    type          = "forward"
    target_group_arn = aws_lb_target_group.ecs_nlb_tg[each.key].arn
  }
}

```

6.2.9.4 3. LAUNCH CONFIGURATION AND AUTOSCALING GROUP

These are the locals we will be using:

- A list with the target group arn that it will be using by the Auto Scaling Group.
- The customized name for the haproxy config file. It will depend on the nlb name. Right now, available names: PROD, PRD1 and PRD2.
- The customized name for the haproxy user data file, depending also on the nlb name.

```

locals{
    target_group_arns = [for k,v in aws_lb_target_group.ecs_nlb_tg : aws_lb_target_group.ecs_nlb_tg[k].arn]
    config_file_name = "haproxy-${var.nlb_name}.cfg"
    user_data_file   = "haproxy-userdata-${var.nlb_name}.sh"
}

```

The config file will be uploaded to an s3 bucket. A folder with the name Haproxy-nlbName will be created, so each config file will be uploaded there.

```

#uploading files to s3 bucket
resource "aws_s3_bucket_object" "haproxy-config" {
    bucket = "vf-iedelivery-centralized-internet-vpc-267040142128-123456"
    key    = "Haproxy-${var.nlb_name}//haproxy.cfg"
    source = "${path.module}/haproxyconfig/${local.config_file_name}"
    etag =
    "#${filemd5("${path.module}/haproxyconfig/${local.config_file_name}")}"
    #triggered changes
}

```

Haproxy EC2 instances will be using a customized AMI creating using AWS IMAGE BUILDER. The details for that will be added in another section.

```

#----data specific ami
data "aws_ami" "haproxy" {
    most_recent      = true
    owners          = ["self"]
    filter {
        name    = "name"
        values  = ["VFIE-DELIVERY-haproxy*"]
    }
}
resource "aws_launch_configuration" "haproxy_conf" {
    name_prefix     = "Haproxy-${var.VPC_NAME}-${var.TAGS["Project"]}-${var.nlb_name}"
    image_id       = data.aws_ami.haproxy.id
    instance_type = "t2.medium"
    iam_instance_profile = var.ssm_instance_profile
    security_groups = [var.ssm_security_group,
    aws_security_group.haproxy_ec2_sg.id]
    user_data = "${file("${path.module}/user-data/${local.user_data_file}")}"
    lifecycle {
        create_before_destroy = true
    }
}

```

These are the details for the user data for PROD. The script function called haproxy-conf-refresh.sh will be used to update the config file in haproxy, synchronized with s3 bucket:

```
#!/bin/bash -xe
#haproxy config
sudo systemctl start haproxy || service haproxy start
sudo systemctl enable haproxy
sudo mkdir /etc/haproxy/old
sudo mkdir /etc/haproxy/s3
sudo touch /etc/haproxy/haproxy-conf-refresh.sh
sudo chmod 777 /etc/haproxy/haproxy-conf-refresh.sh
sudo cat > /etc/haproxy/haproxy-conf-refresh.sh << 'EOF'
sudo mv /etc/haproxy/haproxy.cfg{,.original}
sudo aws s3 sync s3://vf-iedelivery-centralized-internet-vpc-267040142128-
123456/HAPROXY-PROD /etc/haproxy/s3
sudo mv /etc/haproxy/s3/haproxy.cfg /etc/haproxy/haproxy.cfg
sudo systemctl restart haproxy
EOF
sudo chmod +x /etc/haproxy/haproxy-conf-refresh.sh
sudo /etc/haproxy/haproxy-conf-refresh.sh
```

Finally, here the details for the auto Scaling group:

```

resource "aws_autoscaling_group" "haproxy_AG" {
  depends_on          = [aws_lb_target_group.ecs_nlb_tg]
  name                = "${var.nlb_name}-HAPROXY-${var.VPC_NAME}-
${var.TAGS["Project"]}"
  launch_configuration = aws_launch_configuration.haproxy_conf.name
  min_size            = 1
  max_size            = 4
  desired_capacity    = 2
  health_check_grace_period = 300
  health_check_type   = "ELB"
  vpc_zone_identifier = [var.subnets[0], var.subnets[1]]
  target_group_arns   = local.target_group_arns

  tag {
    key      = "Environment"
    value    = "${var.TAGS["Environment"]}"
    propagate_at_launch = true
  }

  tag {
    key      = "Project"
    value    = "VF-IEDELIVERY"
    propagate_at_launch = true
  }

  tag {
    key      = "ManagedBy"
    value    = "DL-GDC-PLT-SVCS-Public-Cloud-Managed-
Services@vodafone.com"
    propagate_at_launch = true
  }

  tag {
    key      = "Confidentiality"
    value    = "C3"
    propagate_at_launch = true
  }

  tag {
    key      = "TaggingVersion"
    value    = "V2.3"
    propagate_at_launch = true
  }

  tag {
    key      = "SecurityZone"
    value    = "A"
    propagate_at_launch = true
  }

  tag {
    key      = "Name"
    value    = "AG-${var.nlb_name}-HAPROXY"
    propagate_at_launch = true
  }
}

```

6.2.9.5 4. HAProxy CONFIG

This section will describe the HAProxy configuration.

6.2.9.5.1 PRODUCTION

CodeCommit URL : <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-sharedservices->

<vpc/browse/refs/heads/master/--/modules/lbproxy/haproxyconfig/haproxy-PROD.cfg?region=eu-west-1>

Internal Base URL : prod.haproxy.ieaws.vodafone.com

External Base URL : *.prod.ieaws.vodafone.com

| HAPROXY Port | Direction | Source Whitelist | Destination IP | Destination Port | Description |
|--------------|-----------|--|--|------------------|---------------------|
| 443 | Inbound | N/A | internal-snd1-CchSalALB-1101482058.eu-west-1.elb.amazonaws.com | 443 | Health Check |
| 5500 | Inbound | 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24 10.163.78.0/23 195.232.228.93/32 | cch-sal-alb.prod.ieaws.vodafone.com | 5500 | |
| 5501 | Inbound | 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24 10.163.78.0/23 195.232.228.93/32 | cch-sal-alb.prod.ieaws.vodafone.com | 5501 | |
| 7001 | Inbound | 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24 10.163.78.0/23 47.73.21.74/32 47.73.21.75/32 176.125.13.67/32 176.125.13.68/32 176.125.13.69/32 176.125.13.70/32 195.232.228.93/32 | cch-sal-alb.prod.ieaws.vodafone.com | 7001 | Weblogic Admin |
| 8011 | Inbound | 37.25.160.19 10.109.100.239 10.109.100.240 10.109.100.241 10.109.100.11 10.109.100.12 10.109.100.13 10.109.100.14 10.109.100.15 10.109.100.16 10.109.100.17 10.109.100.38 10.109.100.39 10.109.100.24 | cch-sal-alb.prod.ieaws.vodafone.com | 8011 | CCHSAL Exposed APIs |

| HAProxy Port | Direction | Source Whitelist | Destination IP | Destination Port | Description |
|--------------|-----------|---|---|------------------|--------------------|
| | | 10.109.100.25 10.109.100.181 10.109.100.186 10.151.4.79 10.151.4.82 10.151.4.92 10.151.4.95 10.162.114.10 10.162.114.11 10.162.114.12 10.162.114.13 10.162.114.15 10.162.114.16 10.162.114.18 10.162.114.19 10.162.114.26 10.162.114.25 47.73.21.74 47.73.21.75 176.125.13.67 176.125.13.68 176.125.13.69 176.125.13.70 198.18.65.7 198.18.65.8 198.18.65.9 198.18.65.10 198.18.65.11 198.18.65.12 10.74.120.112/28 195.232.228.93/32 10.109.179.162 10.109.179.163 10.109.179.164 10.109.179.165 10.109.179.166 10.109.179.167 | | | |
| 8405 | Inbound | 127.0.0.1/32 | 127.0.0.1 | 8404 | Stats Health Check |
| 8443 | Inbound | 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32 10.78.177.156/32 10.109.100.245/32 10.109.100.246/32 10.109.100.247/32 10.109.100.250/32 | ieportal-dispatcher.prod.ieaws.vodafone.com | 8443 | IE Portal FE GUIs |

| HAPROXY Port | Direction | Source Whitelist | Destination IP | Destination Port | Description |
|--------------|-----------|---|--|------------------|------------------------------------|
| | | 10.109.100.251/32 10.109.100.252/32 10.109.100.9/32 10.109.100.10/32 37.25.161.156/32 37.25.161.157/32 | | | |
| 10120 | Inbound | 37.25.160.19 198.18.65.7 198.18.65.8 198.18.65.9 198.18.65.10 198.18.65.11 10.109.179.146 10.109.179.147 198.18.74.197 192.125.247.100 198.18.65.12 198.18.65.56 198.18.65.57 | <u>cch-sal-sftp.prod.ieaws.vodafone.com</u> | 22 | CCHSAL SFTP Endpoint |
| 10301 | Inbound | 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23 10.109.98.0/24 | <u>jenkins.prod.ieaws.vodafone.com</u> | 443 | Jenkins - DevOps Tooling |
| 10302 | Inbound | 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23 10.109.98.0/24 | <u>artifactory.prod.ieaws.vodafone.com</u> | 443 | JFrog Artifactory - DevOps Tooling |
| 10303 | Inbound | 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23 10.109.98.0/24 | <u>myst.prod.ieaws.vodafone.com</u> | 443 | MyST - DevOps Tooling |
| 10304 | Inbound | 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23 10.109.98.0/24 | <u>cch-sal-utilities.prod.ieaws.vodafone.com</u> | 1521 | OCM |
| 10304 | Inbound | 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23 10.109.98.0/24 | <u>cch-sal-utilities.prod.ieaws.vodafone.com</u> | 2484 | OCM |
| 13001 | Outbound | 10.181.32.0/19 | 10.78.48.37 | 8005 | UFE |
| 15101 | Outbound | 10.181.32.0/19 | 10.78.48.57 | 30050 | Amdocs OSB |
| 20101 | Outbound | 10.181.32.0/19 | 10.163.187.4 | 6543 | Jinny |

| HAProxy Port | Direction | Source Whitelist | Destination IP | Destination Port | Description |
|--------------|-----------|------------------|----------------|------------------|-------------|
| 20111 | Outbound | 10.181.32.0/19 | 10.163.184.4 | 1040 | SMSC |
| 20112 | Outbound | 10.181.32.0/19 | 10.163.184.4 | 1050 | SMSC |
| 20113 | Outbound | 10.181.32.0/19 | 10.163.184.4 | 1060 | SMSC |
| 20114 | Outbound | 10.181.32.0/19 | 10.163.184.4 | 1070 | SMSC |
| 12103 | Outbound | 10.181.32.0/19 | 198.18.67.252 | 16443 | MEH |
| 22102 | Outbound | 10.181.32.0/19 | 10.109.101.122 | 33001 | MML DB |
| 22103 | Outbound | 10.181.32.0/19 | 10.109.101.123 | 33001 | MML DB |
| 22104 | Outbound | 10.181.32.0/19 | 10.109.101.124 | 33001 | MML DB |
| 22105 | Outbound | 10.181.32.0/19 | 10.109.101.14 | 33001 | UFE DB |
| 27001 | Outbound | 10.181.32.0/19 | 10.109.100.244 | 22 | |
| 28001 | Outbound | 10.181.32.0/19 | 10.162.66.40 | 22 | Lotus Notes |
| 29001 | Outbound | 10.181.32.0/19 | 47.73.21.74 | 8090 | AppDynamics |
| 29003 | Outbound | 10.181.32.0/19 | 47.73.21.74 | 8181 | AppDynamics |

6.2.9.5.2 PRE-PRODUCTION 1 (PRD1)

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0

# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr last,libc,none #to avoid the config failing if
it cannot resolver one dns server name

##----- ENV: PROD-----
-----

#15101 - FROM PROD
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.64.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_15101

backend gcd_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server esb.app.prd1.equinix.vf-ie.internal.vodafone.com
10.78.48.83:32007 # send traffic on host/port; check its port; max
connections of the given value

#20101 - FROM PROD
frontend prod_20101
    bind *:20101
    acl white_list src 10.181.64.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20101

backend gcd_20101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM 10.163.187.4:6543 #
send traffic on host/port; check its port; max connections of the given
value

#20111 - FROM PROD
frontend prod_20111
    bind *:20111
    acl white_list src 10.181.64.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20111

backend gcd_20111

```

```

option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1040 #
send traffic on host/port; check its port; max connections of the given
value

#20112 - FROM PROD
frontend prod_20112
bind *:20112
acl white_list src 10.181.64.0/19
tcp-request connection reject if !white_list
default_backend gcd_20112

backend gcd_20112
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1050 #
send traffic on host/port; check its port; max connections of the given
value

#20113 - FROM PROD
frontend prod_20113
bind *:20113
acl white_list src 10.181.64.0/19
tcp-request connection reject if !white_list
default_backend gcd_20113

backend gcd_20113
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1060 #
send traffic on host/port; check its port; max connections of the given
value

#20114 - FROM PROD
frontend prod_20114
bind *:20114
acl white_list src 10.181.64.0/19
tcp-request connection reject if !white_list
default_backend gcd_20114

backend gcd_20114
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1070 #
send traffic on host/port; check its port; max connections of the given
value

#12103 - FROM PROD
frontend prod_12103
bind *:12103
acl white_list src 10.181.64.0/19
tcp-request connection reject if !white_list
default_backend gcd_12103

backend gcd_12103
option tcp-check # perform a simple TCP check of healthiness against the
server
server meh.prd1.equinox.vf-ie.internal.vodafone.com 198.18.67.253:16443
# send traffic on host/port; check its port; max connections of the given
value

#22102 - FROM PROD
frontend prod_22102
bind *:22102

```

```

acl white_list src 10.181.64.0/19
tcp-request connection reject if !white_list
default_backend gdc_22102

backend gdc_22102
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server iebeoahr-vip.dc-dublin.de 10.109.101.41:33001 # send traffic on
host/port; check its port; max connections of the given value

#22103 - FROM PROD
frontend prod_22103
    bind *:22103
    acl white_list src 10.181.64.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_22103

backend gdc_22103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server iebeobhr-vip.dc-dublin.de 10.109.101.42:33001 # send traffic on
host/port; check its port; max connections of the given value

#22104 - FROM PROD
#frontend prod_22104
#    bind *:22104
#    acl white_list src 10.181.64.0/19
#    tcp-request connection reject if !white_list
#    default_backend gdc_22104

#backend gdc_22104
#    option tcp-check # perform a simple TCP check of healthiness against
the server
#    server iebpokhr-vip.dc-dublin.de 10.109.101.124:33001 # send traffic on
host/port; check its port; max connections of the given value

#5500 - FROM GDC
frontend prod_5500
    bind *:5500
    acl white_list src 10.74.120.112/28 37.25.160.19
    tcp-request connection reject if !white_list
    default_backend gdc_5500

backend gdc_5500
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.prd1.ieaws.vodafone.com cch-sal-
alb.prd1.ieaws.vodafone.com:5500 # send traffic on host/port; check its
port; max connections of the given value

#5501 - FROM GDC
frontend prod_5501
    bind *:5501
    acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32
10.78.177.156/32
    tcp-request connection reject if !white_list
    default_backend gdc_5501

backend gdc_5501
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.prd1.ieaws.vodafone.com cch-sal-
alb.prd1.ieaws.vodafone.com:5501 # send traffic on host/port; check its
port; max connections of the given value

```

```

#7001 - FROM GDC
frontend prod_7001
  bind *:7001
  acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32
  10.78.177.156/32
  tcp-request connection reject if !white_list
  default_backend gcd_7001

backend gcd_7001
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-alb.prd1.ieaws.vodafone.com cch-sal-
  alb.prd1.ieaws.vodafone.com:7001 # send traffic on host/port; check its
  port; max connections of the given value

#10120 - FROM GDC
frontend prod_10120
  bind *:10120
  acl white_list src 37.25.160.19 198.18.65.3 198.18.65.4 198.18.65.5
  198.18.65.6 198.18.74.200 198.18.74.219 10.109.100.118 10.109.100.119
  198.18.65.31
  tcp-request connection reject if !white_list
  default_backend gcd_10120

backend gcd_10120
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-sftp.prd1.ieaws.vodafone.com cch-sal-
  sftp.prd1.ieaws.vodafone.com:22 # send traffic on host/port; check its
  port; max connections of the given value

#8011 - FROM GDC
frontend prod_8011
  bind *:8011
  acl white_list src 37.25.160.19 10.109.100.98 10.109.100.99
  10.109.100.96 10.109.100.97 10.109.100.179 10.109.100.180 10.109.100.131
  10.109.100.132 10.109.100.177 10.109.100.178 10.109.100.135 10.109.100.136
  10.109.100.137 10.109.100.138 10.109.100.139 10.109.100.140 10.109.100.141
  10.109.100.34 10.109.100.35 10.162.122.16 10.162.122.17 10.162.114.20
  10.162.114.21 10.162.111.89 10.162.111.92 10.162.111.81 10.162.111.82
  10.78.177.156 10.78.177.156/32 198.18.65.3 198.18.65.4 198.18.65.5
  198.18.65.6
  tcp-request connection reject if !white_list
  default_backend gcd_8011

backend gcd_8011
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-alb.prd1.ieaws.vodafone.com cch-sal-
  alb.prd1.ieaws.vodafone.com:8011 # send traffic on host/port; check its
  port; max connections of the given value

# added 22.07
#27001 - FROM PRD1
frontend prod_27001
  bind *:27001
  acl white_list src 10.181.64.0/19
  tcp-request connection reject if !white_list
  default_backend gcd_27001

backend gcd_27001
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server IEBSEMN-VIP.dc-dublin.de 10.109.100.105:22 # send traffic on
  host/port; check its port; max connections of the given value

```

```

#10304 - FROM GDC
frontend prod_10304
  bind *:10304
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19

  tcp-request connection reject if !white_list
  default_backend gdc_10304

backend gdc_10304
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server cch-sal-utilities.prd1.ieaws.vodafone.com cch-sal-
utilities.prd1.ieaws.vodafone.com:1521 # send traffic on host/port; check
its port; max connections of the given value

#10305 - FROM GDC
frontend prod_10305
  bind *:10305
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32
10.78.177.156/32
  tcp-request connection reject if !white_list
  default_backend gdc_10305
backend gdc_10305
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server cch-sal-utilities.prd1.ieaws.vodafone.com cch-sal-
utilities.prd1.ieaws.vodafone.com:2484 # send traffic on host/port; check
its port; max connections of the given value

#28001 - FROM PROD
frontend prod_28001
  bind *:28001
  acl white_list src 10.181.64.0/19
  tcp-request connection reject if !white_list
  default_backend gdc_28001

backend gdc_28001
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server LotusNotes.legacy 10.162.66.40:22 # send traffic on host/port;
check its port; max connections of the given value

# 443 -health check
frontend healthcheck
  bind *:443
  default_backend nodes

backend nodes
  balance roundrobin
  option ssl-hello-chk
  server web01 internal-sndl-CchSalALB-1101482058.eu-west-
1.elb.amazonaws.com:443 check

```

| Env | IF-ID | Direction | HAPROXY Config | Description | Status |
|----------|--|-----------|--|--|---------------|
| PRD 1 | IF354.0 1 IF354.0 8 IF354.0 6 IF354.0 9 | OUTBOUND | #15101 - FROM PROD frontend prod_15101 bind *:15101 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_15101 | CCH Sync -> OSB -> ANM CCH Sync -> OSB -> CRM CCH Sync -> OSB -> CM CCH Sync -> OSB -> CIAM | IN PROGRES |

| | | | | | |
|----------|--|--------------|--|--|--------------------|
| | IF354.0 5 IF354.0 3 IF354.1 3 | | backend gdc_15101 option tcp-check # perform a simple TCP check of healthiness against the server server esb.app.prd1.equinox.vf- ie.internal.vodafone.com 10.78.48.83:32007 | CCH Sync -> OSB -> OMS CCH Sync -> OSB -> RPL CCH Sync -> OSB -> CCS - SurePay | |
| PRD 1 | IF354.1 1 | OUTBOUN D | #20101 - FROM PROD frontend prod_20101 bind *:20101 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_20101 backend gdc_20101 option tcp-check # perform a simple TCP check of healthiness against the server server JINNY-APP-PROD-VF- IE.INTERNAL.VODAFONE.COM 10.163.187.4:6543 | CCH Sync-> Direct -> Jinny | IN PROGRES S |
| PRD 1 | IF355.0 1 | OUTBOUN D | #20111 - FROM PROD frontend prod_20111 bind *:20111 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_20111 backend gdc_20111 option tcp-check # perform a simple TCP check of healthiness against the server server SMSC-APP-SIT1-VF- IE.INTERNAL.VODAFONE.COM 10.162.229.54:1040 | CCH Batch-> Direct -> SMSC | IN PROGRES S |
| PRD 1 | IF355.0 1 | OUTBOUN D | #20112 - FROM PROD frontend prod_20112 bind *:20112 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_20112 backend gdc_20112 option tcp-check # perform a simple TCP check of healthiness against the server server SMSC-APP-SIT1-VF- IE.INTERNAL.VODAFONE.COM 10.162.229.54:1050 | CCH Batch-> Direct -> SMSC | IN PROGRES S |
| PRD 1 | IF355.0 1 | OUTBOUN D | #20113 - FROM PROD frontend prod_20113 bind *:20113 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list | CCH Batch-> Direct -> SMSC | IN PROGRES S |

| | | | | | |
|-------|-----------|----------|--|--|--------------------|
| | | | default_backend gdc_20113 backend gdc_20113 option tcp-check # perform a simple TCP check of healthiness against the server server SMSC-APP-SIT1-VF- IE.INTERNAL.VODAFONE.COM 10.162.229.54:1060 | | |
| PRD 1 | IF355.0 1 | OUTBOUND | #20114 - FROM PROD frontend prod_20114 bind *:20114 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_20114 backend gdc_20114 option tcp-check # perform a simple TCP check of healthiness against the server server SMSC-APP-SIT1-VF- IE.INTERNAL.VODAFONE.COM 10.162.229.54:1070 | CCH Batch-> Direct -> SMSC | IN PROGRES S |
| PRD 1 | IF354.1 4 | OUTBOUND | #12103 - FROM PROD frontend prod_12103 bind *:12103 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_12103 backend gdc_12103 option tcp-check # perform a simple TCP check of healthiness against the server server meh.prd1.equinox.vf- ie.internal.vodafone.com 198.18.67.253:16443 | CCH Sync-> Direct -> MEH | IN PROGRES S |
| PRD 1 | IF038.1 5 | OUTBOUND | #22102 - FROM PROD frontend prod_22102 bind *:22102 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_22102 backend gdc_22102 option tcp-check # perform a simple TCP check of healthiness against the server server iebeoahr-vip.dc-dublin.de 10.109.101.41:33001 | MML DB-<- Direct <- File Splitter DB | IN PROGRES S |
| PRD 1 | IF038.1 5 | OUTBOUND | #22103 - FROM PROD frontend prod_22103 bind *:22103 acl white_list src 10.181.64.0/19 | MML DB-<- Direct <- File Splitter DB | IN PROGRES S |

| | | | | | |
|-------|-----------|----------|--|------------------------------------|-------------|
| | | | <pre> tcp-request connection reject if !white_list default_backend gdc_22103 backend gdc_22103 option tcp-check # perform a simple TCP check of healthiness against the server server iebeobhr-vip.dc-dublin.de 10.109.101.42:33001 </pre> | | |
| PRD 1 | IF340.0 1 | OUTBOUND | <pre> #27001 - FROM PRD1 frontend prod_27001 bind *:27001 acl white_list src 10.181.64.0/19 tcp-request connection reject if !white_list default_backend gdc_27001 backend gdc_27001 option tcp-check # perform a simple TCP check of healthiness against the server server IEBSEMN-VIP.dc-dublin.de 10.109.100.105:22 </pre> | File Splitter -> Direct -> AR | IN PROGRESS |
| PRD 1 | ACCES S | INBOUND | <pre> #7001 - FROM GDC frontend prod_7001 bind *:7001 acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gdc_7001 backend gdc_7001 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd1.ieaws.vodafone.com cch- sal- alb.prd1.ieaws.vodafone.com:7001 </pre> | Proxy Access to Weblogic Admin GUI | IN PROGRESS |
| PRD 1 | ACCES S | INBOUND | <pre> #5500 - FROM GDC frontend prod_5500 bind *:5500 acl white_list src 10.74.120.112/28 37.25.160.19 tcp-request connection reject if !white_list default_backend gdc_5500 backend gdc_5500 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd1.ieaws.vodafone.com cch- </pre> | Proxy Access to Oracle OEM GUI | IN PROGRESS |

| | | | | | |
|-------|--|---------|---|--|--------------|
| | | | sal-alb.prd1.ieaws.vodafone.com:5500 | | |
| PRD 1 | ACCES S | INBOUND | <pre>#5501 - FROM GDC frontend prod_5501 bind *:5501 acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gcd_5501 backend gcd_5501 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd1.ieaws.vodafone.com cch- sal- alb.prd1.ieaws.vodafone.com:5501</pre> | Proxy Access to Oracle OEM GUI | IN PROGRES S |
| PRD 1 | IF138.6 4 IF356.0 2 IF173.0 8 IF220.0 4 IF175.0 9 | INBOUND | <pre>#8011 - FROM GDC frontend prod_8011 bind *:8011 acl white_list src 37.25.160.19 10.109.100.98 10.109.100.99 10.109.100.96 10.109.100.97 10.109.100.179 10.109.100.180 10.109.100.131 10.109.100.132 10.109.100.177 10.109.100.178 10.109.100.135 10.109.100.136 10.109.100.137 10.109.100.138 10.109.100.139 10.109.100.140 10.109.100.141 10.109.100.34 10.109.100.35 10.162.122.16 10.162.122.17 10.162.114.20 10.162.114.21 10.162.111.89 10.162.111.92 10.162.111.81 10.162.111.82 10.78.177.156 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gcd_8011 backend gcd_8011 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd1.ieaws.vodafone.com cch- sal- alb.prd1.ieaws.vodafone.com:8011</pre> | CCS - Surepay -> OSB -> CCH Sync UFE-> Direct - > CCH Sync Portal-> Direct -> CCH Sync GISMSH-> Direct -> CCH Sync GIG -> Tibco - > CCH Sync Tibco (old ESB)-> Direct - > CCH Sync | IN PROGRES S |
| PRD 1 | IF020.1 4 IF248.0 5 IF355.0 2 IF313.0 | INBOUND | <pre>#10120 - FROM GDC frontend prod_10120 bind *:10120 acl white_list src 37.25.160.19 198.18.65.3 198.18.65.4 198.18.65.5 198.18.65.6 198.18.74.200 198.18.74.219 10.109.100.118</pre> | MEH -> CCH Outbound/Batch MCCM<- MFT (Pull) MFT (Push) -> CCH | IN PROGRES S |

| | | | | | |
|-------|-----|---------|--|---|--------------|
| | | | 10.109.100.119 198.18.65.31 tcp-request connection reject if !white_list default_backend gdc_10120 backend gdc_10120 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- sftp.prd1.ieaws.vodafone.com cch- sal-sftp.prd1.ieaws.vodafone.com:22 IF162.0 3 IF283.0 2 IF222.0 2 IF115.0 4 IF340.0 4 | Batch ANM<- MFT (Pull) MFT (Push) -> CCH Batch CCH Batch<- MFT (Pull) MFT (Push) -> MCCM MCO<- MFT (Pull) MFT (Push) -> CCH Batch AIB -> Legacy Batch -> File Splitter Connect Direct Secure Plus -> Legacy Batch -> File Splitter File Splitter<- MFT (Pull) MFT (Push) -> AR File Splitter<- MFT (Pull) MFT (Push) -> Jupiter (ICCS) File Splitter<- MFT (Pull) MFT (Push) -> SingleServe SEPA Hub -> Group MFT -> Connect Direct -> File Splitter An Post -> Legacy Batch -> File Splitter Kiosk HAND -> Legacy Batch -> File Splitter Jupiter (ICCS)<- MFT (Pull) MFT (Push) -> File Splitter File Splitter<- MFT (Pull) MFT (Push) -> RPL | |
| PRD 1 | OCM | INBOUND | #10304 - FROM GDC frontend prod_10304 bind *:10304 acl white_list src 10.109.98.0/24 | Oracle Connection | IN PROGRES S |

| | | | | | |
|-------|-----------|---------|---|---|--------------------|
| | | | 10.163.78.0/23 37.25.160.19 tcp-request connection reject if !white_list default_backend gdc_10304 backend gdc_10304 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- utilities.prd1.ieaws.vodafone.com cch-sal- utilities.prd1.ieaws.vodafone.com:15 21 | Manager access to DBs | |
| PRD 1 | OCM (SSL) | INBOUND | #10305 - FROM GDC frontend prod_10305 bind *:10305 acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gdc_10305 backend gdc_10305 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- utilities.prd1.ieaws.vodafone.com cch-sal- utilities.prd1.ieaws.vodafone.com:24 84 | Oracle Connection Manager access to DBs (SSL) | IN PROGRES S |

6.2.9.5.3 PRE-PRODUCTION 2 (PRD2)

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0

# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr last,libc,none #to avoid the config failing if
it cannot resolver one dns server name

#----- ENV: PROD-----
-----

#15101 - FROM PROD
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_15101

backend gdc_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server esb.app.prd2.equinix.vf-ie.internal.vodafone.com
10.78.48.84:33007 # send traffic on host/port; check its port; max
connections of the given value

#12103 - FROM PROD
frontend prod_12103
    bind *:12103
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_12103

backend gdc_12103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server meh.prd1.equinix.vf-ie.internal.vodafone.com 198.18.67.253:16443
# send traffic on host/port; check its port; max connections of the given
value

#22102 - FROM PROD
frontend prod_22102
    bind *:22102
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_22102

```

```

backend gdc_22102
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server iebeomhr-vip.dc-dublin.de 10.109.101.83:33001 # send traffic on
host/port; check its port; max connections of the given value

#22103 - FROM PROD
frontend prod_22103
  bind *:22103
  acl white_list src 10.181.96.0/19
  tcp-request connection reject if !white_list
  default_backend gdc_22103

backend gdc_22103
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server iebeonhr-vip.dc-dublin.de 10.109.101.84:33001 # send traffic on
host/port; check its port; max connections of the given value

#22104 - FROM PROD
#frontend prod_22104
#  bind *:22104
#  acl white_list src 10.181.96.0/19
#  tcp-request connection reject if !white_list
#  default_backend gdc_22104

#backend gdc_22104
#  option tcp-check # perform a simple TCP check of healthiness against
the server
#  server iebpokhr-vip.dc-dublin.de iebpokhr-vip.dc-dublin.de:33001 # send
traffic on host/port; check its port; max connections of the given value

#5500 - FROM GDC
frontend prod_5500
  bind *:5500
  acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32
10.78.177.156/32
  tcp-request connection reject if !white_list
  default_backend gdc_5500

backend gdc_5500
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-alb.prd2.ieaws.vodafone.com cch-sal-
alb.prd2.ieaws.vodafone.com:5500 # send traffic on host/port; check its
port; max connections of the given value

#5501 - FROM GDC
frontend prod_5501
  bind *:5501
  acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32
10.78.177.156/32
  tcp-request connection reject if !white_list
  default_backend gdc_5501

backend gdc_5501
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-alb.prd2.ieaws.vodafone.com cch-sal-
alb.prd2.ieaws.vodafone.com:5501 # send traffic on host/port; check its
port; max connections of the given value

#7001 - FROM GDC
frontend prod_7001
  bind *:7001

```

```

    acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32
10.78.177.156/32
    tcp-request connection reject if !white_list
    default_backend gdc_7001

backend gdc_7001
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.prd2.ieaws.vodafone.com cch-sal-
alb.prd2.ieaws.vodafone.com:7001 # send traffic on host/port; check its
port; max connections of the given value

#10120 - FROM GDC
frontend prod_10120
    bind *:10120
    acl white_list src 37.25.160.19 198.18.65.3 198.18.65.4 198.18.65.5
198.18.65.6 198.18.74.200 198.18.74.219 10.109.100.184 10.109.100.185
10.109.100.133 10.109.100.134 10.109.100.182 10.109.100.183 10.109.100.142
10.109.100.143 10.109.100.144 10.109.100.145 10.109.100.146 10.109.100.147
10.109.100.148 10.109.100.36 10.109.100.37 10.109.100.118 10.109.100.119
198.18.65.31
    tcp-request connection reject if !white_list
    default_backend gdc_10120

backend gdc_10120
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-sftp.prd2.ieaws.vodafone.com cch-sal-
sftp.prd2.ieaws.vodafone.com:22 # send traffic on host/port; check its
port; max connections of the given value

#8011 - FROM GDC
frontend prod_8011
    bind *:8011
    acl white_list src 37.25.160.19 10.109.100.98 10.109.100.99
10.109.100.184 10.109.100.185 10.109.100.133 10.109.100.134 10.109.100.182
10.109.100.183 10.109.100.142 10.109.100.143 10.109.100.144 10.109.100.145
10.109.100.146 10.109.100.147 10.109.100.148 10.109.100.36 10.109.100.37
37.25.160.48 37.25.160.49 37.25.160.50 37.25.160.61 10.78.177.156
10.78.177.156
    tcp-request connection reject if !white_list
    default_backend gdc_8011

backend gdc_8011
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.prd2.ieaws.vodafone.com cch-sal-
alb.prd2.ieaws.vodafone.com:8011 # send traffic on host/port; check its
port; max connections of the given value

# 22.07 changes

#27001 - FROM PRD2
frontend prod_27001
    bind *:27001
    acl white_list src 10.181.96.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_27001

backend gdc_27001
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEBSEOP-VIP.dc-dublin.de 10.109.100.109:22 # send traffic on
host/port; check its port; max connections of the given value

```

```

#10304 - FROM GDC
frontend prod_10304
  bind *:10304
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19
  tcp-request connection reject if !white_list
  default_backend gdc_10304

backend gdc_10304
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server cch-sal-utilities.prd2.ieaws.vodafone.com cch-sal-
utilities.prd2.ieaws.vodafone.com:1521 # send traffic on host/port; check
its port; max connections of the given value

# added 23.07
#29002 - FROM PROD
frontend prod_29002
  bind *:29002
  acl white_list src 10.181.96.0/19
  tcp-request connection reject if !white_list
  default_backend gdc_29002

backend gdc_29002
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server smsc-sim.ieaws.vodafone.com smsc-sim.ieaws.vodafone.com:2775 #
send traffic on host/port; check its port; max connections of the given
value

#10305 - FROM GDC
frontend prod_10305
  bind *:10305
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32
10.78.177.156/32
  tcp-request connection reject if !white_list
  default_backend gdc_10305
backend gdc_10305
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server cch-sal-utilities.prd2.ieaws.vodafone.com cch-sal-
utilities.prd2.ieaws.vodafone.com:2484 # send traffic on host/port; check
its port; max connections of the given value

#HEALTH CHECK AND MONITORING
listen stats
  bind *:8404
  mode http
  stats enable
  stats hide-version
  stats uri /haproxy_stats
  stats refresh 30s
  #stats admin if LOCALHOST

```

| Env | IF-ID | Direction | HAPROXY Config | Description | Status |
|----------|--------------|-----------|--|---|---------------|
| PRD 2 | IF354.0 1 | OUTBOUND | #15101 - FROM PROD frontend prod_15101 bind *:15101 acl white_list src 10.181.96.0/19 tcp-request connection reject if !white_list default_backend gdc_15101 | CCH Sync -> OSB -> ANM CCH Sync -> OSB -> CRM CCH Sync -> OSB -> CM CCH Sync -> OSB -> CIAM CCH Sync -> | IN PROGRES |
| | IF354.0 8 | | | | |
| | IF354.0 6 | | | | |
| | IF354.0 9 | | | | |
| | IF354.0 | | backend gdc_15101 | | |

| | | | | | |
|-------|-----------------------------------|----------|---|---|--------------------|
| | 5 IF354.0 3 IF354.1 3 | | option tcp-check # perform a simple TCP check of healthiness against the server server esb.app.prd2.equinov.vf- ie.internal.vodafone.com esb.app.prd2.equinov.vf- ie.internal.vodafone.com:33007 | OSB -> OMS CCH Sync -> OSB -> RPL CCH Sync -> OSB -> CCS - SurePay | |
| PRD 2 | IF354.1 4 | OUTBOUND | #12103 - FROM PROD frontend prod_12103 bind *:12103 acl white_list src 10.181.96.0/19 tcp-request connection reject if !white_list default_backend gdc_12103 backend gdc_12103 option tcp-check # perform a simple TCP check of healthiness against the server server meh.prd1.equinov.vf- ie.internal.vodafone.com meh.prd1.equinov.vf- ie.internal.vodafone.com:16443 | CCH Sync-> Direct -> MEH | IN PROGRES S |
| PRD 2 | IF038.1 5 | OUTBOUND | #22102 - FROM PROD frontend prod_22102 bind *:22102 acl white_list src 10.181.96.0/19 tcp-request connection reject if !white_list default_backend gdc_22102 backend gdc_22102 option tcp-check # perform a simple TCP check of healthiness against the server server iebeomhr-vip.dc-dublin.de 10.109.101.83:33001 | MML DB-<- Direct <- File Splitter DB | IN PROGRES S |
| PRD 2 | IF038.1 5 | OUTBOUND | #22103 - FROM PROD frontend prod_22103 bind *:22103 acl white_list src 10.181.96.0/19 tcp-request connection reject if !white_list default_backend gdc_22103 backend gdc_22103 option tcp-check # perform a simple TCP check of healthiness against the server server iebeonhr-vip.dc-dublin.de 10.109.101.84:33001 | MML DB-<- Direct <- File Splitter DB | IN PROGRES S |
| PRD 2 | IF340.0 1 | OUTBOUND | #27001 - FROM PRD2 frontend prod_27001 bind *:27001 acl white_list src 10.181.96.0/19 tcp-request connection reject if !white_list | File Splitter -> Direct -> AR | IN PROGRES S |

| | | | | | |
|-------|----------|----------|--|------------------------------------|--------------|
| | | | <pre>default_backend gdc_27001 backend gdc_27001 option tcp-check # perform a simple TCP check of healthiness against the server server IEBSEOP-VIP.dc-dublin.de 10.109.100.109:22</pre> | | |
| PRD 2 | SMSC SIM | OUTBOUND | <pre>#29002 - FROM PROD frontend prod_29002 bind *:29002 acl white_list src 10.181.96.0/19 tcp-request connection reject if !white_list default_backend gdc_29002 backend gdc_29002 option tcp-check # perform a simple TCP check of healthiness against the server server smsc- sim.ieaws.vodafone.com smsc- sim.ieaws.vodafone.com:2775</pre> | UMS to SMSC Simulator | IN PROGRES S |
| PRD 2 | ACCES S | INBOUND | <pre>#5500 - FROM GDC frontend prod_5500 bind *:5500 acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gdc_5500 backend gdc_5500 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd2.ieaws.vodafone.com cch- sal- alb.prd2.ieaws.vodafone.com:5500</pre> | Proxy Access to Oracle OEM GUI DB1 | IN PROGRES S |
| PRD 2 | ACCES S | INBOUND | <pre>#5501 - FROM GDC frontend prod_5501 bind *:5501 acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gdc_5501 backend gdc_5501 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd2.ieaws.vodafone.com cch-</pre> | Proxy Access to Oracle OEM GUI DB1 | IN PROGRES S |

| | | | | | |
|-------|---|---------|--|--|--------------|
| | | | sal-alb.prd2.ieaws.vodafone.com:5501 | | |
| PRD 2 | ACCES S | INBOUND | #7001 - FROM GDC frontend prod_7001 bind *:7001 acl white_list src 10.74.120.112/28 37.25.160.19/32 195.232.228.93/32 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gcd_7001 backend gcd_7001 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd2.ieaws.vodafone.com cch- sal- alb.prd2.ieaws.vodafone.com:7001 | Proxy Access to Weblogic Admin GUI | IN PROGRES S |
| PRD 2 | TBC IF020.1 4 IF248.0 5 IF355.0 2 IF313.0 4 IF027.0 4 IF236.0 2 IF340.0 1 IF340.0 2 IF340.0 3 IF162.0 3 IF283.0 2 IF222.0 2 IF115.0 4 IF340.0 4 | INBOUND | #10120 - FROM GDC frontend prod_10120 bind *:10120 acl white_list src 37.25.160.19 198.18.65.3 198.18.65.4 198.18.65.5 198.18.65.6 198.18.74.200 198.18.74.219 10.109.100.184 10.109.100.185 10.109.100.133 10.109.100.134 10.109.100.182 10.109.100.183 10.109.100.142 10.109.100.143 10.109.100.144 10.109.100.145 10.109.100.146 10.109.100.147 10.109.100.148 10.109.100.36 10.109.100.37 10.109.100.118 10.109.100.119 198.18.65.31 tcp-request connection reject if !white_list default_backend gcd_10120 backend gcd_10120 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- sftp.prd2.ieaws.vodafone.com cch- sal-sftp.prd2.ieaws.vodafone.com:22 | MEH -> CCH Outbound/Batch MCCM<- MFT (Pull) MFT (Push) -> CCH Batch ANM<- MFT (Pull) MFT (Push) -> CCH Batch CCH Batch-<- MFT (Pull) MFT (Push) -> MCCM MCO<- MFT (Pull) MFT (Push) -> CCH Batch AIB -> Legacy Batch -> File Splitter Connect Direct Secure Plus -> Legacy Batch -> File Splitter File Splitter-<- MFT (Pull) MFT (Push) -> AR File Splitter-<- MFT (Pull) MFT (Push) -> Jupiter (ICCS) File Splitter-<- MFT | IN PROGRES S |

| | | | | | |
|-------|---|---------|--|--|--------------|
| | | | | (Pull) MFT (Push) -> SingleServe SEPA Hub -> Group MFT -> Connect Direct -> File Splitter An Post -> Legacy Batch -> File Splitter Kiosk HAND -> Legacy Batch -> File Splitter Jupiter (ICCS)<- MFT (Pull) MFT (Push) -> File Splitter File Splitter<- MFT (Pull) MFT (Push) -> RPL | |
| PRD 2 | IF138.6 4 IF356.0 2 IF173.0 8 IF220.0 4 IF175.0 9 | INBOUND | #8011 - FROM GDC frontend prod_8011 bind *:8011 acl white_list src 37.25.160.19 10.109.100.98 10.109.100.99 10.109.100.184 10.109.100.185 10.109.100.133 10.109.100.134 10.109.100.182 10.109.100.183 10.109.100.142 10.109.100.143 10.109.100.144 10.109.100.145 10.109.100.146 10.109.100.147 10.109.100.148 10.109.100.36 10.109.100.37 37.25.160.48 37.25.160.49 37.25.160.50 37.25.160.61 10.78.177.156 10.78.177.156 tcp-request connection reject if !white_list default_backend gcd_8011 backend gcd_8011 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- alb.prd2.ieaws.vodafone.com cch- sal- alb.prd2.ieaws.vodafone.com:8011 | Portal-> Direct -> OAL UFE-> Direct -> OAL CCS - Surepay -> OSB -> CCH Sync UFE-> Direct -> CCH Sync Portal-> Direct -> CCH Sync GISMSH-> Direct -> CCH Sync GIG -> Tibco -> CCH Sync Tibco (old ESB)-> Direct -> CCH Sync | IN PROGRES S |
| PRD 2 | OCM | INBOUND | #10304 - FROM GDC frontend prod_10304 bind *:10304 acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19 tcp-request connection reject if !white_list default_backend gcd_10304 | Oracle Connection Manager access to DBs | IN PROGRES S |

| | | | | | |
|-------|-----------|---------|---|---|--------------------|
| | | | backend gdc_10304 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- utilities.prd2.ieaws.vodafone.com cch-sal- utilities.prd2.ieaws.vodafone.com:15 21 | | |
| PRD 2 | OCM (SSL) | INBOUND | #10305 - FROM GDC frontend prod_10305 bind *:10305 acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32 10.78.177.156/32 tcp-request connection reject if !white_list default_backend gdc_10305 backend gdc_10305 option tcp-check # perform a simple TCP check of healthiness against the server server cch-sal- utilities.prd2.ieaws.vodafone.com cch-sal- utilities.prd2.ieaws.vodafone.com:24 84 | Oracle Connection Manager access to DBs (SSL) | IN PROGRES S |

6.2.9.5.4 SIT1

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0

# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr last,libc,none #to avoid the config failing if
it cannot resolver one dns server name

##----- ENV: SIT1-----
-----

#15101 - FROM SIT1 - AMDOCS OSB
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.128.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_15101

backend gcd_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server ieessbavr.dc-dublin.de 10.109.96.89:7007 # send traffic on
host/port; check its port; max connections of the given value

#20101 - FROM SIT1 - JINNY (PROD - NO TEST AVAILABLE)
frontend prod_20101
    bind *:20101
    acl white_list src 10.181.128.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20101

backend gcd_20101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM 10.163.187.4:6543 #
send traffic on host/port; check its port; max connections of the given
value

#20111 - FROM SIT1 - TEST SMSC
frontend prod_20111
    bind *:20111
    acl white_list src 10.181.128.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20111

backend gcd_20111

```

```

option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1040 #
send traffic on host/port; check its port; max connections of the given
value

#20112 - FROM SIT1 - TEST SMSC
frontend prod_20112
bind *:20112
acl white_list src 10.181.128.0/19
tcp-request connection reject if !white_list
default_backend gcd_20112

backend gcd_20112
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1050 #
send traffic on host/port; check its port; max connections of the given
value

#20113 - FROM SIT1 - TEST SMSC
frontend prod_20113
bind *:20113
acl white_list src 10.181.128.0/19
tcp-request connection reject if !white_list
default_backend gcd_20113

backend gcd_20113
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1060 #
send traffic on host/port; check its port; max connections of the given
value

#20114 - FROM SIT1 - TEST SMSC
frontend prod_20114
bind *:20114
acl white_list src 10.181.128.0/19
tcp-request connection reject if !white_list
default_backend gcd_20114

backend gcd_20114
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54::1070 #
send traffic on host/port; check its port; max connections of the given
value

#12103 - FROM SIT1 - MEH
frontend prod_12103
bind *:12103
acl white_list src 10.181.128.0/19
tcp-request connection reject if !white_list
default_backend gcd_12103

backend gcd_12103
option tcp-check # perform a simple TCP check of healthiness against the
server
server meh.sit1.equinov.vf-ie.internal.vodafone.com 198.18.67.221:16443
# send traffic on host/port; check its port; max connections of the given
value

#22102 - FROM SIT1 - MML DB
frontend prod_22102
bind *:22102

```

```

acl white_list src 10.181.128.0/19
tcp-request connection reject if !white_list
default_backend gdc_22102

backend gdc_22102
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN5HR-vip.dc-dublin.de 10.106.184.46:33001 # send traffic on
host/port; check its port; max connections of the given value

#22103 - FROM SIT1 - MML DB
frontend prod_22103
    bind *:22103
    acl white_list src 10.181.128.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_22103

backend gdc_22103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN6HR-vip.dc-dublin.de 10.106.184.47:33001 # send traffic on
host/port; check its port; max connections of the given value

#5500 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5500
    bind *:5500
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5500

backend gdc_5500
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit1.ieaws.vodafone.com cch-sal-
alb.sit1.ieaws.vodafone.com:5500 # send traffic on host/port; check its
port; max connections of the given value

#5501 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5501
    bind *:5501
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5501

backend gdc_5501
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit1.ieaws.vodafone.com cch-sal-
alb.sit1.ieaws.vodafone.com:5501 # send traffic on host/port; check its
port; max connections of the given value

#7001 - FROM GDC - Weblogic Admin
frontend prod_7001
    bind *:7001
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 47.73.21.74/32 47.73.21.75/32 176.125.13.67/32
176.125.13.68/32 176.125.13.69/32 176.125.13.70/32 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_7001

backend gdc_7001
    option tcp-check # perform a simple TCP check of healthiness against the
server

```

```

server cch-sal-alb.sit1.ieaws.vodafone.com cch-sal-
alb.sit1.ieaws.vodafone.com:7001 # send traffic on host/port; check its
port; max connections of the given value

#10120 - FROM GDC - CCHSAL SFTP
frontend prod_10120
  bind *:10120
  acl white_list src 37.25.160.19 10.109.96.88 10.109.96.125 198.18.64.204
  198.18.64.205 198.18.64.215 198.18.64.218 198.18.64.222
  tcp-request connection reject if !white_list
  default_backend gcdc_10120

backend gcdc_10120
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server cch-sal-sftp.sit1.ieaws.vodafone.com cch-sal-
    sftp.sit1.ieaws.vodafone.com:22 # send traffic on host/port; check its
    port; max connections of the given value

#8011 - FROM GDC - Weblogic API
frontend prod_8011
  bind *:8011
  acl white_list src 37.25.160.19 10.109.96.89 10.109.96.90 198.18.64.196
  198.18.64.200 198.18.64.201 198.18.64.203 198.18.64.207 198.18.64.208
  198.18.64.209 198.18.64.210 10.109.96.105 10.109.96.106 10.109.96.107
  10.109.96.108 10.109.96.109 10.109.96.110 10.109.96.49 10.109.96.47
  10.109.96.60 10.109.96.79
  tcp-request connection reject if !white_list
  default_backend gcdc_8011

backend gcdc_8011
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server cch-sal-alb.sit1.ieaws.vodafone.com cch-sal-
    alb.sit1.ieaws.vodafone.com:8011 # send traffic on host/port; check its
    port; max connections of the given value

#10304 - FROM GDC - OCM
frontend prod_10304
  bind *:10304
  acl white_list src 10.109.98.0/24 10.163.78.0/23
  tcp-request connection reject if !white_list
  default_backend gcdc_10304

backend gcdc_10304
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server cch-sal-utilities.sit1.ieaws.vodafone.com cch-sal-
    utilities.sit1.ieaws.vodafone.com:1521 # send traffic on host/port; check
    its port; max connections of the given value

#10305 - FROM GDC - OCM
frontend prod_10305
  bind *:10305
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32
  tcp-request connection reject if !white_list
  default_backend gcdc_10305
backend gcdc_10305
  option tcp-check # perform a simple TCP check of healthiness against the
  server
    server cch-sal-utilities.sit1.ieaws.vodafone.com cch-sal-
    utilities.sit1.ieaws.vodafone.com:2484 # send traffic on host/port; check
    its port; max connections of the given value

#HEALTH CHECK AND MONITORING

```

```
listen stats
  bind *:8404
  mode http
  stats enable
  stats hide-version
  stats uri /haproxy_stats
  stats refresh 30s
#stats admin if LOCALHOST
```

6.2.9.5.5 SIT2

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0

# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr last,libc,none #to avoid the config failing if
it cannot resolver one dns server name

##----- ENV: SIT2-----
-----

#15101 - FROM SIT2 - AMDOCS OSB
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.160.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_15101

backend gcd_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server ieessbbvr.dc-dublin.de 10.109.96.90:11007 # send traffic on
host/port; check its port; max connections of the given value

#20101 - FROM SIT2 - JINNY (PROD - NO TEST AVAILABLE)
frontend prod_20101
    bind *:20101
    acl white_list src 10.181.160.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20101

backend gcd_20101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM 10.163.187.4:6543 #
send traffic on host/port; check its port; max connections of the given
value

#20111 - FROM SIT2 - TEST SMSC
frontend prod_20111
    bind *:20111
    acl white_list src 10.181.160.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20111

backend gcd_20111

```

```

option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1040 #
send traffic on host/port; check its port; max connections of the given
value

#20112 - FROM SIT2 - TEST SMSC
frontend prod_20112
bind *:20112
acl white_list src 10.181.160.0/19
tcp-request connection reject if !white_list
default_backend gcd_20112

backend gcd_20112
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1050 #
send traffic on host/port; check its port; max connections of the given
value

#20113 - FROM SIT2 - TEST SMSC
frontend prod_20113
bind *:20113
acl white_list src 10.181.160.0/19
tcp-request connection reject if !white_list
default_backend gcd_20113

backend gcd_20113
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1060 #
send traffic on host/port; check its port; max connections of the given
value

#20114 - FROM SIT2 - TEST SMSC
frontend prod_20114
bind *:20114
acl white_list src 10.181.160.0/19
tcp-request connection reject if !white_list
default_backend gcd_20114

backend gcd_20114
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54::1070 #
send traffic on host/port; check its port; max connections of the given
value

#12103 - FROM SIT2 - MEH
frontend prod_12103
bind *:12103
acl white_list src 10.181.160.0/19
tcp-request connection reject if !white_list
default_backend gcd_12103

backend gcd_12103
option tcp-check # perform a simple TCP check of healthiness against the
server
server meh.sit2.equinox.vf-ie.internal.vodafone.com 198.18.67.220:16443
# send traffic on host/port; check its port; max connections of the given
value

#22102 - FROM SIT2 - MML DB
frontend prod_22102
bind *:22102

```

```

acl white_list src 10.181.160.0/19
tcp-request connection reject if !white_list
default_backend gdc_22102

backend gdc_22102
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN5HR-vip.dc-dublin.de 10.106.184.46:33001 # send traffic on
host/port; check its port; max connections of the given value

#22103 - FROM SIT2 - MML DB
frontend prod_22103
    bind *:22103
    acl white_list src 10.181.160.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_22103

backend gdc_22103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN6HR-vip.dc-dublin.de 10.106.184.47:33001 # send traffic on
host/port; check its port; max connections of the given value

#5500 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5500
    bind *:5500
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5500

backend gdc_5500
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit2.ieaws.vodafone.com cch-sal-
alb.sit2.ieaws.vodafone.com:5500 # send traffic on host/port; check its
port; max connections of the given value

#5501 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5501
    bind *:5501
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5501

backend gdc_5501
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit2.ieaws.vodafone.com cch-sal-
alb.sit2.ieaws.vodafone.com:5501 # send traffic on host/port; check its
port; max connections of the given value

#7001 - FROM GDC - Weblogic Admin
frontend prod_7001
    bind *:7001
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 47.73.21.74/32 47.73.21.75/32 176.125.13.67/32
176.125.13.68/32 176.125.13.69/32 176.125.13.70/32 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_7001

backend gdc_7001
    option tcp-check # perform a simple TCP check of healthiness against the
server

```

```

server cch-sal-alb.sit2.ieaws.vodafone.com cch-sal-
alb.sit2.ieaws.vodafone.com:7001 # send traffic on host/port; check its
port; max connections of the given value

#10120 - FROM GDC - CCHSAL SFTP
frontend prod_10120
  bind *:10120
  acl white_list src 37.25.160.19 10.109.96.88 10.109.96.125 198.18.64.204
  198.18.64.205 198.18.64.215 198.18.64.218 198.18.64.222
  tcp-request connection reject if !white_list
  default_backend gcdc_10120

backend gcdc_10120
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-sftp.sit2.ieaws.vodafone.com cch-sal-
  sftp.sit2.ieaws.vodafone.com:22 # send traffic on host/port; check its
  port; max connections of the given value

#8011 - FROM GDC - Weblogic API
frontend prod_8011
  bind *:8011
  acl white_list src 37.25.160.19 10.109.96.89 10.109.96.90 198.18.64.196
  198.18.64.200 198.18.64.201 198.18.64.203 198.18.64.207 198.18.64.208
  198.18.64.209 198.18.64.210 10.109.96.105 10.109.96.106 10.109.96.107
  10.109.96.108 10.109.96.109 10.109.96.110 10.109.96.49 10.109.96.47
  10.109.96.60 10.109.96.79
  tcp-request connection reject if !white_list
  default_backend gcdc_8011

backend gcdc_8011
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-alb.sit2.ieaws.vodafone.com cch-sal-
  alb.sit2.ieaws.vodafone.com:8011 # send traffic on host/port; check its
  port; max connections of the given value

#10304 - FROM GDC - OCM
frontend prod_10304
  bind *:10304
  acl white_list src 10.109.98.0/24 10.163.78.0/23
  tcp-request connection reject if !white_list
  default_backend gcdc_10304

backend gcdc_10304
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-utilities.sit2.ieaws.vodafone.com cch-sal-
  utilities.sit2.ieaws.vodafone.com:1521 # send traffic on host/port; check
  its port; max connections of the given value

#10305 - FROM GDC - OCM
frontend prod_10305
  bind *:10305
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32
  tcp-request connection reject if !white_list
  default_backend gcdc_10305
backend gcdc_10305
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-utilities.sit2.ieaws.vodafone.com cch-sal-
  utilities.sit2.ieaws.vodafone.com:2484 # send traffic on host/port; check
  its port; max connections of the given value

#HEALTH CHECK AND MONITORING

```

```
listen stats
  bind *:8404
  mode http
  stats enable
  stats hide-version
  stats uri /haproxy_stats
  stats refresh 30s
#stats admin if LOCALHOST
```

6.2.9.5.6 SIT3

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0

# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr last,libc,none #to avoid the config failing if
it cannot resolver one dns server name

##----- ENV: SIT3-----
-----

#15101 - FROM SIT3 - AMDOCS OSB
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.192.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_15101

backend gcd_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server ieessbbvr.dc-dublin.de 10.109.96.90:9007 # send traffic on
host/port; check its port; max connections of the given value

#20101 - FROM SIT3 - JINNY (PROD - NO TEST AVAILABLE)
frontend prod_20101
    bind *:20101
    acl white_list src 10.181.192.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20101

backend gcd_20101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM 10.163.187.4:6543 #
send traffic on host/port; check its port; max connections of the given
value

#20111 - FROM SIT3 - TEST SMSC
frontend prod_20111
    bind *:20111
    acl white_list src 10.181.192.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20111

backend gcd_20111

```

```

option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1040 #
send traffic on host/port; check its port; max connections of the given
value

#20112 - FROM SIT3 - TEST SMSC
frontend prod_20112
bind *:20112
acl white_list src 10.181.192.0/19
tcp-request connection reject if !white_list
default_backend gcd_20112

backend gcd_20112
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1050 #
send traffic on host/port; check its port; max connections of the given
value

#20113 - FROM SIT3 - TEST SMSC
frontend prod_20113
bind *:20113
acl white_list src 10.181.192.0/19
tcp-request connection reject if !white_list
default_backend gcd_20113

backend gcd_20113
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1060 #
send traffic on host/port; check its port; max connections of the given
value

#20114 - FROM SIT3 - TEST SMSC
frontend prod_20114
bind *:20114
acl white_list src 10.181.192.0/19
tcp-request connection reject if !white_list
default_backend gcd_20114

backend gcd_20114
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54::1070 #
send traffic on host/port; check its port; max connections of the given
value

#12103 - FROM SIT3 - MEH
frontend prod_12103
bind *:12103
acl white_list src 10.181.192.0/19
tcp-request connection reject if !white_list
default_backend gcd_12103

backend gcd_12103
option tcp-check # perform a simple TCP check of healthiness against the
server
server meh.sit3.equinox.vf-ie.internal.vodafone.com 198.18.67.250:16443
# send traffic on host/port; check its port; max connections of the given
value

#22102 - FROM SIT3 - MML DB
frontend prod_22102
bind *:22102

```

```

acl white_list src 10.181.192.0/19
tcp-request connection reject if !white_list
default_backend gdc_22102

backend gdc_22102
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN5HR-vip.dc-dublin.de 10.106.184.46:33001 # send traffic on
host/port; check its port; max connections of the given value

#22103 - FROM SIT3 - MML DB
frontend prod_22103
    bind *:22103
    acl white_list src 10.181.192.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_22103

backend gdc_22103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN6HR-vip.dc-dublin.de 10.106.184.47:33001 # send traffic on
host/port; check its port; max connections of the given value

#5500 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5500
    bind *:5500
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5500

backend gdc_5500
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit3.ieaws.vodafone.com cch-sal-
alb.sit3.ieaws.vodafone.com:5500 # send traffic on host/port; check its
port; max connections of the given value

#5501 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5501
    bind *:5501
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5501

backend gdc_5501
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit3.ieaws.vodafone.com cch-sal-
alb.sit3.ieaws.vodafone.com:5501 # send traffic on host/port; check its
port; max connections of the given value

#7001 - FROM GDC - Weblogic Admin
frontend prod_7001
    bind *:7001
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 47.73.21.74/32 47.73.21.75/32 176.125.13.67/32
176.125.13.68/32 176.125.13.69/32 176.125.13.70/32 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_7001

backend gdc_7001
    option tcp-check # perform a simple TCP check of healthiness against the
server

```

```

server cch-sal-alb.sit3.ieaws.vodafone.com cch-sal-
alb.sit3.ieaws.vodafone.com:7001 # send traffic on host/port; check its
port; max connections of the given value

#10120 - FROM GDC - CCHSAL SFTP
frontend prod_10120
  bind *:10120
  acl white_list src 37.25.160.19 10.109.96.88 10.109.96.125 198.18.64.204
  198.18.64.205 198.18.64.215 198.18.64.218 198.18.64.222
  tcp-request connection reject if !white_list
  default_backend gcdc_10120

backend gcdc_10120
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-sftp.sit3.ieaws.vodafone.com cch-sal-
  sftp.sit3.ieaws.vodafone.com:22 # send traffic on host/port; check its
  port; max connections of the given value

#8011 - FROM GDC - Weblogic API
frontend prod_8011
  bind *:8011
  acl white_list src 37.25.160.19 10.109.96.89 10.109.96.90 198.18.64.196
  198.18.64.200 198.18.64.201 198.18.64.203 198.18.64.207 198.18.64.208
  198.18.64.209 198.18.64.210 10.109.96.105 10.109.96.106 10.109.96.107
  10.109.96.108 10.109.96.109 10.109.96.110 10.109.96.49 10.109.96.47
  10.109.96.60 10.109.96.79
  tcp-request connection reject if !white_list
  default_backend gcdc_8011

backend gcdc_8011
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-alb.sit3.ieaws.vodafone.com cch-sal-
  alb.sit3.ieaws.vodafone.com:8011 # send traffic on host/port; check its
  port; max connections of the given value

#10304 - FROM GDC - OCM
frontend prod_10304
  bind *:10304
  acl white_list src 10.109.98.0/24 10.163.78.0/23
  tcp-request connection reject if !white_list
  default_backend gcdc_10304

backend gcdc_10304
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-utilities.sit3.ieaws.vodafone.com cch-sal-
  utilities.sit3.ieaws.vodafone.com:1521 # send traffic on host/port; check
  its port; max connections of the given value

#10305 - FROM GDC - OCM
frontend prod_10305
  bind *:10305
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32
  tcp-request connection reject if !white_list
  default_backend gcdc_10305
backend gcdc_10305
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-utilities.sit3.ieaws.vodafone.com cch-sal-
  utilities.sit3.ieaws.vodafone.com:2484 # send traffic on host/port; check
  its port; max connections of the given value

#HEALTH CHECK AND MONITORING

```

```
listen stats
  bind *:8404
  mode http
  stats enable
  stats hide-version
  stats uri /haproxy_stats
  stats refresh 30s
#stats admin if LOCALHOST
```

6.2.9.5.7 SIT4

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0

# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr last,libc,none #to avoid the config failing if
it cannot resolver one dns server name

##----- ENV: SIT4-----
-----

#15101 - FROM SIT4 - AMDOCS OSB
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.224.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_15101

backend gcd_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server ieessbbvr.dc-dublin.de 10.109.96.90:14010 # send traffic on
host/port; check its port; max connections of the given value

#20101 - FROM SIT4 - JINNY (PROD - NO TEST AVAILABLE)
frontend prod_20101
    bind *:20101
    acl white_list src 10.181.224.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20101

backend gcd_20101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM 10.163.187.4:6543 #
send traffic on host/port; check its port; max connections of the given
value

#20111 - FROM SIT4 - TEST SMSC
frontend prod_20111
    bind *:20111
    acl white_list src 10.181.224.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20111

backend gcd_20111

```

```

option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1040 #
send traffic on host/port; check its port; max connections of the given
value

#20112 - FROM SIT4 - TEST SMSC
frontend prod_20112
bind *:20112
acl white_list src 10.181.224.0/19
tcp-request connection reject if !white_list
default_backend gcd_20112

backend gcd_20112
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1050 #
send traffic on host/port; check its port; max connections of the given
value

#20113 - FROM SIT4 - TEST SMSC
frontend prod_20113
bind *:20113
acl white_list src 10.181.224.0/19
tcp-request connection reject if !white_list
default_backend gcd_20113

backend gcd_20113
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54:1060 #
send traffic on host/port; check its port; max connections of the given
value

#20114 - FROM SIT4 - TEST SMSC
frontend prod_20114
bind *:20114
acl white_list src 10.181.224.0/19
tcp-request connection reject if !white_list
default_backend gcd_20114

backend gcd_20114
option tcp-check # perform a simple TCP check of healthiness against the
server
server SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.162.229.54::1070 #
send traffic on host/port; check its port; max connections of the given
value

#12103 - FROM SIT4 - MEH
frontend prod_12103
bind *:12103
acl white_list src 10.181.224.0/19
tcp-request connection reject if !white_list
default_backend gcd_12103

backend gcd_12103
option tcp-check # perform a simple TCP check of healthiness against the
server
server meh.sit4.equinox.vf-ie.internal.vodafone.com 198.18.67.249:16443
# send traffic on host/port; check its port; max connections of the given
value

#22102 - FROM SIT4 - MML DB
frontend prod_22102
bind *:22102

```

```

acl white_list src 10.181.224.0/19
tcp-request connection reject if !white_list
default_backend gdc_22102

backend gdc_22102
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN5HR-vip.dc-dublin.de 10.106.184.46:33001 # send traffic on
host/port; check its port; max connections of the given value

#22103 - FROM SIT4 - MML DB
frontend prod_22103
    bind *:22103
    acl white_list src 10.181.224.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_22103

backend gdc_22103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server IEORN6HR-vip.dc-dublin.de 10.106.184.47:33001 # send traffic on
host/port; check its port; max connections of the given value

#5500 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5500
    bind *:5500
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5500

backend gdc_5500
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit4.ieaws.vodafone.com cch-sal-
alb.sit4.ieaws.vodafone.com:5500 # send traffic on host/port; check its
port; max connections of the given value

#5501 - FROM GDC - Weblogic Enterprise Manager
frontend prod_5501
    bind *:5501
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_5501

backend gdc_5501
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server cch-sal-alb.sit4.ieaws.vodafone.com cch-sal-
alb.sit4.ieaws.vodafone.com:5501 # send traffic on host/port; check its
port; max connections of the given value

#7001 - FROM GDC - Weblogic Admin
frontend prod_7001
    bind *:7001
    acl white_list src 10.74.120.112/28 37.25.160.19/32 10.109.98.0/24
10.163.78.0/23 47.73.21.74/32 47.73.21.75/32 176.125.13.67/32
176.125.13.68/32 176.125.13.69/32 176.125.13.70/32 195.232.228.93/32
    tcp-request connection reject if !white_list
    default_backend gdc_7001

backend gdc_7001
    option tcp-check # perform a simple TCP check of healthiness against the
server

```

```

server cch-sal-alb.sit4.ieaws.vodafone.com cch-sal-
alb.sit4.ieaws.vodafone.com:7001 # send traffic on host/port; check its
port; max connections of the given value

#10120 - FROM GDC - CCHSAL SFTP
frontend prod_10120
  bind *:10120
  acl white_list src 37.25.160.19 10.109.96.88 10.109.96.125 198.18.64.204
  198.18.64.205 198.18.64.215 198.18.64.218 198.18.64.222
  tcp-request connection reject if !white_list
  default_backend gcdc_10120

backend gcdc_10120
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-sftp.sit4.ieaws.vodafone.com cch-sal-
  sftp.sit4.ieaws.vodafone.com:22 # send traffic on host/port; check its
  port; max connections of the given value

#8011 - FROM GDC - Weblogic API
frontend prod_8011
  bind *:8011
  acl white_list src 37.25.160.19 10.109.96.89 10.109.96.90 198.18.64.196
  198.18.64.200 198.18.64.201 198.18.64.203 198.18.64.207 198.18.64.208
  198.18.64.209 198.18.64.210 10.109.96.105 10.109.96.106 10.109.96.107
  10.109.96.108 10.109.96.109 10.109.96.110 10.109.96.49 10.109.96.47
  10.109.96.60 10.109.96.79
  tcp-request connection reject if !white_list
  default_backend gcdc_8011

backend gcdc_8011
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-alb.sit4.ieaws.vodafone.com cch-sal-
  alb.sit4.ieaws.vodafone.com:8011 # send traffic on host/port; check its
  port; max connections of the given value

#10304 - FROM GDC - OCM
frontend prod_10304
  bind *:10304
  acl white_list src 10.109.98.0/24 10.163.78.0/23
  tcp-request connection reject if !white_list
  default_backend gcdc_10304

backend gcdc_10304
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-utilities.sit4.ieaws.vodafone.com cch-sal-
  utilities.sit4.ieaws.vodafone.com:1521 # send traffic on host/port; check
  its port; max connections of the given value

#10305 - FROM GDC - OCM
frontend prod_10305
  bind *:10305
  acl white_list src 10.109.98.0/24 10.163.78.0/23 37.25.160.19/32
  tcp-request connection reject if !white_list
  default_backend gcdc_10305
backend gcdc_10305
  option tcp-check # perform a simple TCP check of healthiness against the
  server
  server cch-sal-utilities.sit4.ieaws.vodafone.com cch-sal-
  utilities.sit4.ieaws.vodafone.com:2484 # send traffic on host/port; check
  its port; max connections of the given value

#HEALTH CHECK AND MONITORING

```

```
listen stats
  bind *:8404
  mode http
  stats enable
  stats hide-version
  stats uri /haproxy_stats
  stats refresh 30s
#stats admin if LOCALHOST
```

6.2.9.5.8 UTILITIES

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0

# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr last,libc,none #to avoid the config failing if
it cannot resolver one dns server name

#----- ENV: PROD-----
-----

#10301 - FROM GDC
frontend prod_10301
    bind *:10301
    acl white_list src 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23
10.109.98.0/24
    tcp-request connection reject if !white_list
    default_backend gcd_10301

backend gcd_10301
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server jenkins.utilities.ieaws.vodafone.com
jenkins.utilities.ieaws.vodafone.com:443 # send traffic on host/port;
check its port; max connections of the given value

#10302 - FROM GDC
frontend prod_10302
    bind *:10302
    acl white_list src 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23
10.109.98.0/24
    tcp-request connection reject if !white_list
    default_backend gcd_10302

backend gcd_10302
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server artifactory.utilities.ieaws.vodafone.com
artifactory.utilities.ieaws.vodafone.com:443 # send traffic on host/port;
check its port; max connections of the given value

#10303 - FROM GDC
frontend prod_10303
    bind *:10303
    acl white_list src 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23
10.109.98.0/24
    tcp-request connection reject if !white_list

```

```

default_backend gcd_10303

backend gcd_10303
    option tcp-check # perform a simple TCP check of healthiness against the
    server
        server myst.utilities.ieaws.vodafone.com
myst.utilities.ieaws.vodafone.com:443 # send traffic on host/port; check
its port; max connections of the given value

#10306 - FROM GDC
frontend prod_10306
    bind *:10306
    acl white_list src 37.25.160.19/32 10.74.120.112/28 10.163.78.0/23
10.109.98.0/24
    tcp-request connection reject if !white_list
    default_backend gcd_10306

backend gcd_10306
    option tcp-check # perform a simple TCP check of healthiness against the
    server
        server jmeter.utilities.ieaws.vodafone.com
jmeter.utilities.ieaws.vodafone.com:443 # send traffic on host/port; check
its port; max connections of the given value

#HEALTH CHECK AND MONITORING
listen stats
    bind *:8404
    mode http
    stats enable
    stats hide-version
    stats uri /haproxy_stats
    stats refresh 30s
    #stats admin if LOCALHOST

```

6.2.9.5.9 HAProxy Config file

References:

- <https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-configuration/>
- <https://www.haproxy.com/blog/introduction-to-haproxy-logging/>
- <https://cbonte.github.io/haproxy-dconv/1.7/configuration.html>

There are four essential sections to an HAProxy configuration file. They are `global`, `defaults`, `frontend`, and `backend`. These four sections define how the server as a whole performs, what your default settings are, and how client requests are received and routed to your backend servers

The configuration file can be found at `/etc/haproxy/haproxy.cfg` and its format can be tested using "`haproxy -c -f /etc/haproxy/haproxy.cfg`". The structure of this file is as follows:

| |
|--|
| <code>global</code> |
| <code># Settings under global define process-wide security and performance tunings that affect HAProxy at a low level</code> |
| <code>defaults</code> |
| <code># Its settings apply to all of the frontend and backend sections that come after it</code> |
| <code>frontend</code> |

| |
|---|
| # defines the IP addresses and ports that clients can connect to |
| backend |
| # defines a group of servers that will be load balanced and assigned to handle requests |

6.2.9.5.9.1 GLOBAL VARIABLES

For this project, we will use **maxconn**, that sets the maximum number of connections that HAProxy will accept, and **log**, that ensures that warnings emitted during startup and issues that arise during runtime get logged to syslog. It is possible to target the traditional UNIX socket where Syslog or journald, listen, /dev/log, or specify a **remote syslog** server so that log data is preserved externally to your load balancing server. This is the configuration used:

- maxconn 10000 # this would be a very high number of connections from a single microservice - can be modified
- log 127.0.0.1:514 local0

6.2.9.5.9.2 LOGGING CONFIGURATION

HAProxy can emit log message for processing by a **syslog** server. This is compatible with familiar syslog tools like **Rsyslog** (it is already installed in the AMI). The following needs to be added either to **/etc/rsyslog.conf** or to a new file within the **rsyslog.d** directory, like **/etc/rsyslog.d/haproxy.conf**:

| |
|---|
| # Collect log with UDP |
| \$ModLoad imudp |
| \$UDPServerAddress 127.0.0.1 |
| \$UDPServerRun 514 |
| |
| # Creating separate log files based on the severity |
| local0.* /var/log/haproxy-traffic.log |
| local0.notice /var/log/haproxy-admin.log |

6.2.9.5.9.3 DEFAULT VARIABLES

For this project, we will use:

- log global
- option tcplog
- mode tcp
- retries 3
- timeout connect 10s # the time that HAProxy will wait for a TCP connection to a backend server to be established
- timeout client 60s
- timeout server 60s

The `timeout connect` setting configures the time that HAProxy will wait for a TCP connection to a backend server to be established. The “s” suffix denotes seconds. Without any suffix, the time is assumed to be in milliseconds. The `timeout client` setting measures inactivity during periods that we would expect the client to be speaking, or in other words sending TCP segments. The `timeout server` setting measures inactivity when we’d expect the backend server to be speaking. When a timeout expires, the connection is closed. Having sensible timeouts reduces the risk of deadlocked processes tying up a connections that could otherwise be reused.

When operating HAProxy in TCP mode, which is set with `mode tcp`, `timeout server` should be the same as `timeout client`. That’s because HAProxy doesn’t know which side is supposed to be speaking and, since both apply all the time, having different values makes confusion more likely.

The `log global` setting is a way of telling each subsequent `frontend` to use the `log` setting that you defined in the `global` section. This isn’t required for logging, as new `log` lines can be added here or in each `frontend`. However, in most cases wherein only one syslog server is used, this is common.

The `mode` setting defines whether HAProxy operates as a simple TCP proxy or if it’s able to inspect incoming traffic’s higher-level HTTP messages. If most of your `frontend` and `backend` sections would use the same mode, it makes sense to specify it in the `defaults` section to avoid repetition.

6.2.9.5.9.4 FRONTEND AND BACKEND

For each port that the HAProxy will be listeded, defined in above table, a pair of frontend and backend will be created. The traffic could come from GDC or from AWS (production environment). This traffic will reach first the NLB for PROD, and it will be redirected to an HAProxy, that will determine the destination and next hop.

6.2.9.5.9.5 PROD HAProxy.CFG

```

## References
# https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-
# configuration/


# Globals are process wide configuration parameters
global
    maxconn 10000 # this would be a very high number of connections from a
single microservice - can be modified
    log 127.0.0.1:514 local0


# Defaults are configuration parameters which apply to each frontend and
backend to reduce/remove configuraiton duplication
# However settings can be overridden at the frontend / backend level
defaults
    log          global
    option        tcplog
    mode         tcp
    retries      3
    timeout connect 10s # the time that HAProxy will wait for a TCP
connection to a backend server to be established
    timeout client 60s
    timeout server 60s
    default-server init-addr none #to avoid the config failing if it cannot
resolver one dns server name
##----- ENV: PROD-----
-----
#15101 - FROM PROD
frontend prod_15101
    bind *:15101
    acl white_list src 10.181.32.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_15101

backend gcd_15101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server esb.app.prod.equinox.vf-ie.internal.vodafone.com
esb.app.prod.equinox.vf-ie.internal.vodafone.com:30050 # send traffic on
host/port; check its port; max connections of the given value
#20101 - FROM PROD
frontend prod_20101
    bind *:20101
    acl white_list src 10.181.32.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20101
backend gcd_20101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM JINNY-APP-PROD-VF-
IE.INTERNAL.VODAFONE.COM:6543 # send traffic on host/port; check its port;
max connections of the given value
#20111 - FROM PROD
frontend prod_20111
    bind *:20111
    acl white_list src 10.181.32.0/19
    tcp-request connection reject if !white_list
    default_backend gcd_20111
backend gcd_20111
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server SMSC-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM SMSC-APP-PROD-VF-
IE.INTERNAL.VODAFONE.COM:1040 # send traffic on host/port; check its port;
max connections of the given value
#20112 - FROM PROD
frontend prod_20112

```

```

bind *:20112
acl white_list src 10.181.32.0/19
tcp-request connection reject if !white_list
default_backend gdc_20112

backend gdc_20112
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server SMSC-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM SMSC-APP-PROD-VF-
IE.INTERNAL.VODAFONE.COM:1050 # send traffic on host/port; check its port;
max connections of the given value
#20113 - FROM PROD
frontend prod_20113
    bind *:20113
    acl white_list src 10.181.32.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_20113

backend gdc_20113
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server SMSC-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM SMSC-APP-PROD-VF-
IE.INTERNAL.VODAFONE.COM:1060 # send traffic on host/port; check its port;
max connections of the given value
#20114 - FROM PROD
frontend prod_20114
    bind *:20114
    acl white_list src 10.181.32.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_20114

backend gdc_20114
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server SMSC-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM SMSC-APP-PROD-VF-
IE.INTERNAL.VODAFONE.COM:1070 # send traffic on host/port; check its port;
max connections of the given value
#12101 - FROM PROD
frontend prod_12101
    bind *:12101
    acl white_list src 10.181.32.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_12101

backend gdc_12101
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server meh.prod.equinox.vf-ie.internal.vodafone.com meh.prod.equinox.vf-
ie.internal.vodafone.com:22 # send traffic on host/port; check its port;
max connections of the given value
#12103 - FROM PROD
frontend prod_12103
    bind *:12103
    acl white_list src 10.181.32.0/19
    tcp-request connection reject if !white_list
    default_backend gdc_12103

backend gdc_12103
    option tcp-check # perform a simple TCP check of healthiness against the
server
    server meh.prod.equinox.vf-ie.internal.vodafone.com meh.prod.equinox.vf-
ie.internal.vodafone.com:16443 # send traffic on host/port; check its
port; max connections of the given value

#22102 - FROM PROD
frontend prod_22102

```

```

bind *:22102
acl white_list src 10.181.32.0/19
tcp-request connection reject if !white_list
default_backend gcd_22102

backend gcd_22102
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server iebpoihr-vip.dc-dublin.de iebpoihr-vip.dc-dublin.de:33001 # send
traffic on host/port; check its port; max connections of the given value
#22103 - FROM PROD
frontend prod_22103
  bind *:22103
  acl white_list src 10.181.32.0/19
  tcp-request connection reject if !white_list
  default_backend gcd_22103

backend gcd_22103
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server iebpojhr-vip.dc-dublin.de iebpojhr-vip.dc-dublin.de:33001 # send
traffic on host/port; check its port; max connections of the given value
#22104 - FROM PROD
frontend prod_22104
  bind *:22104
  acl white_list src 10.181.32.0/19
  tcp-request connection reject if !white_list
  default_backend gcd_22104

backend gcd_22104
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server iebpokhr-vip.dc-dublin.de iebpokhr-vip.dc-dublin.de:33001 # send
traffic on host/port; check its port; max connections of the given value
#5500 - FROM GDC
frontend prod_5500
  bind *:5500
  acl white_list src 10.74.120.112/28
  tcp-request connection reject if !white_list
  default_backend gcd_5500

backend gcd_5500
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server cch-sal-alb.prod.ieaws.vodafone.com cch-sal-
alb.prod.ieaws.vodafone.com:5500 # send traffic on host/port; check its
port; max connections of the given value
#5501 - FROM GDC
frontend prod_5501
  bind *:5501
  acl white_list src 10.74.120.112/28
  tcp-request connection reject if !white_list
  default_backend gcd_5501

backend gcd_5501
  option tcp-check # perform a simple TCP check of healthiness against the
server
  server cch-sal-alb.prod.ieaws.vodafone.com cch-sal-
alb.prod.ieaws.vodafone.com:5501 # send traffic on host/port; check its
port; max connections of the given value
#7001 - FROM GDC
frontend prod_7001
  bind *:7001
  acl white_list src 10.74.120.112/28
  tcp-request connection reject if !white_list

```

```

default_backend gcd_7001

backend gcd_7001
    option tcp-check # perform a simple TCP check of healthiness against the
    server
        server cch-sal-alb.prod.ieaws.vodafone.com cch-sal-
alb.prod.ieaws.vodafone.com:7001 # send traffic on host/port; check its
port; max connections of the given value
#10120 - FROM GDC
frontend prod_10120
    bind *:10120
    acl white_list src 198.18.65.7 198.18.65.8 198.18.65.9 198.18.65.10
198.18.65.11 10.109.179.146 10.109.179.147 198.18.74.197 192.125.247.100
198.18.65.12 198.18.65.56 198.18.65.57
    tcp-request connection reject if !white_list
    default_backend gcd_10120

backend gcd_10120
    option tcp-check # perform a simple TCP check of healthiness against the
    server
        server cch-sal-sftp.prod.ieaws.vodafone.com cch-sal-
sftp.prod.ieaws.vodafone.com:22 # send traffic on host/port; check its
port; max connections of the given value

#8011 - FROM GDC
frontend prod_8011
    bind *:8011
    acl white_list src 10.109.100.239 10.109.100.240 10.109.100.241
10.109.100.11 10.109.100.12 10.109.100.13 10.109.100.14 10.109.100.15
10.109.100.16 10.109.100.17 10.109.100.38 10.109.100.39 10.109.100.24
10.109.100.25 10.109.100.181 10.109.100.186 10.151.4.79 10.151.4.82
10.151.4.92 10.151.4.95 10.162.114.10 10.162.114.11 10.162.114.12
10.162.114.13 10.162.114.15 10.162.114.16 10.162.114.18 10.162.114.19
10.162.114.26 10.162.114.25
    tcp-request connection reject if !white_list
    default_backend gcd_8011

backend gcd_8011
    option tcp-check # perform a simple TCP check of healthiness against the
    server
        server cch-sal-alb.prod.ieaws.vodafone.com cch-sal-
alb.prod.ieaws.vodafone.com:8011 # send traffic on host/port; check its
port; max connections of the given value
# 443 -health check
frontend healthcheck
    bind *:443
    default_backend nodes

backend nodes
    balance roundrobin
    option ssl-hello-chk
    server web01 internal-sndl-CchSalALB-1101482058.eu-west-
1.elb.amazonaws.com:443 check

```

Code block 1 haproxy.cfg**6.2.10 HAProxy-ECS IMPLEMENTATION*****6.2.10.1**

- [Introduction](#)
- [1. NLB](#)
- [2. TASK DEFINITION](#)

- [3. ECS SERVICE](#)
- [4. ECS Docker Image, ECR repository and endpoints](#)
 - [Dockerfile](#)
 - [ECR](#)
 - [Endpoints](#)

6.2.10.2 Introduction

*IT IS NOT IMPLEMENTED

Right now, AWS has a limitation in the number of target groups you can associate with an ECS service. This means an ECS service can just receive traffic from 5 different ports (5 different target groups). AWS documentation [here](#). For this project, we need haproxy to be receiving traffic from more than 10 different ports. As a remediation solution, we have migrated the haproxy service to EC2. This documentation will explain how to configure haproxy using ecs fargate.

6.2.10.3 1. NLB

One **target group** per port will be created, using a 'for_each' expression, in order to differentiate with a name the target groups in the fargate ecs block. The name convention is \${var.VPC_NAME}-\${var.TAGS["Project"]}-\${var.nlb_name}-\${each.value}. The protocol will be 'TCP' and the target type will be '**ip**', as we are using ecs fargate for HAProxy. This way, if the input listener map variable has, for example, this value: {"aws-1" :15101 }, the target group will be called \${var.VPC_NAME}-\${var.TAGS["Project"]}-\${var.nlb_name}-\${each.value}["aws-1"], that will make possible to automate the association between an ecs_service and each target group. One **lb listener** per port will be created, using a 'for_each' expression (same reasons as above explained). Each listener will have a forward action to the corresponding target group (same port).

Below it's part of the configuration for LB and target group when using ECS, and how target groups are associated with ECS services.

```

locals{
    list_port = values(var.port_listeners)
    list_keys = keys(var.port_listeners)
    port_per_service = chunklist(local.list_port,5)
    keys_per_service = chunklist(local.list_keys,5)
}
#-----LB-----
...
resource "aws_lb_target_group" "ecs_nlb_tg" {
    for_each = var.port_listeners
    name      = "${var.VPC_NAME}-${var.TAGS["Project"]}-${var.nlb_name}-
${each.value}"
    vpc_id    = var.vpc_id
    port      = each.value
    protocol  = "TCP"
    target_type = "ip"
    deregistration_delay = 60
}
resource "aws_lb_listener" "frontend" {
    for_each = var.port_listeners
    load_balancer_arn = aws_lb.nlb.arn
    port      = each.value
    protocol  = "TCP"
    default_action {
        type          = "forward"
        target_group_arn = aws_lb_target_group.ecs_nlb_tg[each.key].arn
    }
}
#-----FARGATE-----
...
resource "aws_ecs_service" "HAProxy_service" {
...
    dynamic "load_balancer" {
        for_each = local.keys_per_service[count.index]
        content {
            target_group_arn =
aws_lb_target_group.ecs_nlb_tg[load_balancer.value].arn
            container_name   = local.container_name
            container_port   = lookup(var.port_listeners, load_balancer.value,
0)
        }
    }
}

```

6.2.10.4 2. TASK DEFINITION

This is the **task definition** for PROD. We need to customize the following values: AWS log group (see options below), the port mapping (for each port haproxy we will be listening to), image (aws ecr) and efs volume if needed:

```
[
  {
    "name": "PROD-HAProxy-fargate",
    "essential": true,
    "logConfiguration": {
      "logDriver": "awslogs",
      "secretOptions": [],
      "options": {
        "awslogs-group": "SS-vpc/vfie-delivery-PROD-HAProxy-fargate",
        "awslogs-region": "eu-west-1",
        "awslogs-stream-prefix": "HAProxy-fargate"
      }
    },
    "portMappings": [
      {
        "hostPort": 15101,
        "protocol": "tcp",
        "containerPort": 15101
      },
      {
        "hostPort": 20101,
        "protocol": "tcp",
        "containerPort": 20101
      },
      {
        "hostPort": 20111,
        "protocol": "tcp",
        "containerPort": 20111
      },
      {
        "hostPort": 20112,
        "protocol": "tcp",
        "containerPort": 20112
      },
      {
        "hostPort": 20113,
        "protocol": "tcp",
        "containerPort": 20113
      },
      {
        "hostPort": 20114,
        "protocol": "tcp",
        "containerPort": 20114
      },
      {
        "hostPort": 12101,
        "protocol": "tcp",
        "containerPort": 12101
      },
      {
        "hostPort": 12102,
        "protocol": "tcp",
        "containerPort": 12102
      },
      {
        "hostPort": 12103,
        "protocol": "tcp",
        "containerPort": 12103
      },
      {
        "hostPort": 22101,
        "protocol": "tcp",
        "containerPort": 22101
      }
    ]
  }
]
```

```

        "hostPort": 22102,
        "protocol": "tcp",
        "containerPort": 22102
    },
    {
        "hostPort": 22103,
        "protocol": "tcp",
        "containerPort": 22103
    },
    {
        "hostPort": 22104,
        "protocol": "tcp",
        "containerPort": 22104
    }
],
"cpu": 1024,
"memory": 2048,
"ulimits": [
    {
        "name": "nofile",
        "softLimit": 32000,
        "hardLimit": 65536
    }
],
"memory": 2048,
"memoryReservation": 1024,
"image": "267040142128.dkr.ecr.eu-west-1.amazonaws.com/ss-
vfie-delivery-haproxy-fargate:latest",
"environment": [
    {
        "name": "AWS_REGION",
        "value": "eu-west-1"
    }
],
"mountPoints": [
    {
        "sourceVolume": "myEfsVolume",
        "containerPath": "/var/haproxy"
    }
]
}
]
```

6.2.10.5 3. ECS SERVICE

This is the part in terraform, in the file [3-haproxy-ecs](#). Please avoid using the locals number_services, port_per_service and keys_per_service. That implementation was created with test purposes.

```

locals{
    container_name = "${var.nlb_name}-HAProxy-fargate"
    task_definition = "${var.is_prod}" == true ? "haproxy-prod.json" :
"${var.is_test}" == true ? "haproxy-test.json" : "haproxy-preprod.json"
    list_port = values(var.port_listeners)
    list_keys = keys(var.port_listeners)
    number_services = ceil(length(local.list_port)/5)
    port_per_service = chunklist(local.list_port,5)
    keys_per_service = chunklist(local.list_keys,5)
}
#-----FARGATE-----
resource "aws_ecs_cluster" "HaProxycluster" {
    name = "${var.VPC_NAME}-${var.TAGS["Project"]}-${var.nlb_name}-HAProxy-
fargate"
    lifecycle {
        create_before_destroy = true
    }
}
resource "aws_ecs_service" "HAProxy_service" {
    count = local.number_services
    depends_on = [aws_lb_target_group.ecs_nlb_tg]
    name = "${var.VPC_NAME}-${var.TAGS["Project"]}-
${var.nlb_name}-HAProxy-service-${count.index}"
    cluster = "${aws_ecs_cluster.HaProxycluster.id}"
    task_definition = "${aws_ecs_task_definition.HAProxy-fargate.arn}"
    desired_count = 1
    platform_version = "1.4.0"
    launch_type = "FARGATE"
    deployment_maximum_percent = 100
    deployment_minimum_healthy_percent = 0

    network_configuration {
        subnets = [var.subnets[0], var.subnets[1]]
        security_groups = [aws_security_group.haproxy_ecs_sg.id]
        assign_public_ip = false
    }
}

dynamic "load_balancer" {
    for_each = local.keys_per_service[count.index]
    content {
        target_group_arn =
aws_lb_target_group.ecs_nlb_tg[load_balancer.value].arn
        container_name = local.container_name
        container_port = lookup(var.port_listeners, load_balancer.value,
0)
    }
}
resource "aws_ecs_task_definition" "HAProxy-fargate" {
    depends_on = [aws_cloudwatch_log_group.HAProxyFargateLogGroup,
aws_ecr_repository.ecs_repo]
    family = local.container_name
    container_definitions = "${file("${path.module}/task-
definition/${local.task_definition}")}"
    task_role_arn = aws_iam_role.HAProxyFargateTaskRole.arn
    execution_role_arn =
aws_iam_role.HAProxyFargateTaskExecutionRole.arn
    volume {
        name = "myEfsVolume"
        efs_volume_configuration {
            file_system_id = aws_efs_file_system.HAProxyFargateEFS.id
            root_directory = "/"
        }
    }
    cpu = 1024
    memory = 2048
}

```

```

    requires_compatibilitys = ["FARGATE"]
    network_mode           = "awsvpc"
}

```

6.2.10.6 4. ECS Docker Image, ECR repository and endpoints

If using ECS Fargate, a docker image, ECR repository and EFS volume will be used.

6.2.10.6.1 Dockerfile

The docker image MUST be in the root of the repository, in order to be able to automate the image push in ECR later.

Dockerfile with port 15101 opened (below). In this case, haproxy configuration file is in the root of the repository too. This will need to be updating with the corresponding port list.

```

FROM haproxy:2.0-alpine
COPY haproxy.cfg /usr/local/etc/haproxy/haproxy.cfg
EXPOSE 15101

```

Code block 2 Dockerfile

6.2.10.6.2 ECR

ECR will be used to store the docker images for haproxy. We will use the tag "latest" to define the latest configuration for haproxy. This repository is created in SS account, but mgmt account will need access. The idea is to automate the action of pushing a new docker image in the [pipeline](#) (vf-iedelivery-sharedservices-vpc-prod-tf). For that purpose, the IAM role that CodeBuild uses needs to be added to the permissions in ECR.

```

#-----ECR REPO-----
---
resource "aws_ecr_repository" "ecs_repo" {
  name = "ss-${var.PROJECT}-haproxy-fargate"
  lifecycle {
    create_before_destroy = true
  }
}
resource "aws_ecr_repository_policy" "JenkinsMasterPolicy" {
  repository = "${aws_ecr_repository.ecs_repo.name}"
  policy = <<EOF
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowAll",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::${var.account_id}:root",
          "arn:aws:iam::831341508773:root",
          "arn:aws:iam::831341508773:role/terraform-
20200330150018185100000001"
        ],
        "Action": "ecr:*"
      }
    ]
  }
}
EOF
}
resource "aws_ecr_lifecycle_policy" "repo-policy" {
  repository = "${aws_ecr_repository.ecs_repo.name}"
  policy = <<EOF
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Keep image deployed with tag '${var.tag}'",
      "selection": {
        "tagStatus": "tagged",
        "tagPrefixList": ["${var.tag}"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    },
    {
      "rulePriority": 2,
      "description": "Keep last 2 any images",
      "selection": {
        "tagStatus": "any",
        "countType": "imageCountMoreThan",
        "countNumber": 2
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
EOF
}

```

6.2.10.6.3 Endpoints

Shared Service VPC does not have Internet Access. We will need to add endpoints for: aws logs, aws ssm , aws ecr and s3.

```

#####
# ENDPOINTS
#####
#FOR ECR API
data "aws_vpc_endpoint_service" "ecr_api" {
  service = "ecr.api"
}
resource "aws_vpc_endpoint" "ecr_api" {
  vpc_id          = aws_vpc.vpc.id
  service_name    = data.aws_vpc_endpoint_service.ecr_api.service_name
  vpc_endpoint_type = "Interface"
  security_group_ids = [aws_security_group.ecr_endpoints_sec_group.id]
  subnet_ids      = [aws_subnet.subnet_priv[0].id]
  private_dns_enabled = true
  tags            = merge(
    local.common_tags,
    {
      "Name" = "${var.VPC_NAME}-${var.PROJECT}-ECR-
API-ENDPOINT-${local.common_tags["Environment"]}""
    },
  )
}
#FOR ECR DKR
data "aws_vpc_endpoint_service" "ecr_dkr" {
  service = "ecr.dkr"
}
resource "aws_vpc_endpoint" "ecr_dkr" {
  vpc_id          = aws_vpc.vpc.id
  service_name    = data.aws_vpc_endpoint_service.ecr_dkr.service_name
  vpc_endpoint_type = "Interface"
  security_group_ids = [aws_security_group.ecr_endpoints_sec_group.id]
  subnet_ids      = [aws_subnet.subnet_priv[0].id]
  private_dns_enabled = true
  tags            = merge(
    local.common_tags,
    {
      "Name" = "${var.VPC_NAME}-${var.PROJECT}-ECR-
DKR-ENDPOINT-${local.common_tags["Environment"]}""
    },
  )
}

#FOR S3
data "aws_vpc_endpoint_service" "s3" {
  service = "s3"
}
resource "aws_vpc_endpoint" "s3" {
  vpc_id          = aws_vpc.vpc.id
  service_name    = data.aws_vpc_endpoint_service.s3.service_name
  tags            = merge(
    local.common_tags,
    {
      "Name" = "${var.VPC_NAME}-${var.PROJECT}-S3-
ENDPOINT-${local.common_tags["Environment"]}""
    },
  )
}
resource "aws_vpc_endpoint_route_table_association" "public_s3" {
  vpc_endpoint_id = aws_vpc_endpoint.s3.id
  route_table_id  = aws_route_table.subnet_priv_route_table[0].id
}
resource "aws_vpc_endpoint_route_table_association" "IA_s3" {
  count           = length(aws_route_table.subnet_IA_tier_route_table)
  vpc_endpoint_id = aws_vpc_endpoint.s3.id
  route_table_id  =
  aws_route_table.subnet_IA_tier_route_table[count.index].id
}

```

```

}

#FOR SECRETS MANAGER
data "aws_vpc_endpoint_service" "secretsmanager" {
    service = "secretsmanager"
}
resource "aws_vpc_endpoint" "secretsmanager" {
    vpc_id          = aws_vpc.vpc.id
    service_name    =
    data.aws_vpc_endpoint_service.secretsmanager.service_name
    vpc_endpoint_type = "Interface"
    security_group_ids = [aws_security_group.ecr_endpoints_sec_group.id]
    subnet_ids       = [aws_subnet.subnet_priv[0].id]
    private_dns_enabled = true
    tags             = merge(
        local.common_tags,
        {
            "Name" = "${var.VPC_NAME}-${var.PROJECT}-
SECRETS-MANAGER-ENDPOINT-${local.common_tags["Environment"]}"
        },
    )
}
#FOR LOGS
data "aws_vpc_endpoint_service" "logs" {
    service = "logs"
}
resource "aws_vpc_endpoint" "logs" {
    vpc_id          = aws_vpc.vpc.id
    service_name    = data.aws_vpc_endpoint_service.logs.service_name
    vpc_endpoint_type = "Interface"
    security_group_ids = [aws_security_group.ecr_endpoints_sec_group.id]
    subnet_ids       = [aws_subnet.subnet_priv[0].id]
    private_dns_enabled = true
    tags             = merge(
        local.common_tags,
        {
            "Name" = "${var.VPC_NAME}-${var.PROJECT}-LOGS-
ENDPOINT-${local.common_tags["Environment"]}"
        },
    )
}

```

IAM

```

#-----IAM ROLES-----
-----
#TASK IAM EXECUTION ROLE
resource "aws_iam_role" "HAProxyFargateTaskExecutionRole" {
  name = "${var.nlb_name}-HAProxyFargateTaskExecutionRole-${var.VPC_NAME}"
  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}
resource "aws_iam_role_policy" "HAProxyFargateTaskExecution_policy" {
  name = "${var.nlb_name}-HAProxyFargateTaskExecution-policy-${var.VPC_NAME}-${var.PROJECT}"
  role = "${aws_iam_role.HAProxyFargateTaskExecutionRole.id}"
  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECRGetBatch",
      "Effect": "Allow",
      "Action": "ecr: *",
      "Resource": "*"
    },
    {
      "Sid": "CloudWatchLogsStream",
      "Effect": "Allow",
      "Action": "logs>CreateLogStream",
      "Resource": "*"
    },
    {
      "Sid": "CloudWatchLogEvents",
      "Effect": "Allow",
      "Action": "logs:PutLogEvents",
      "Resource": "*"
    }
  ]
}
EOF
}
#IAM ROLE TASK
resource "aws_iam_role" "HAProxyFargateTaskRole" {
  name = "${var.nlb_name}-HAProxyFargateTaskRole-${var.VPC_NAME}"
  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ecs-tasks.amazonaws.com",
          "ecs.amazonaws.com"
        ]
      }
    }
  ]
}
EOF
}

```

```

        ]
    },
    "Action": "sts:AssumeRole"
}
]
}
EOF
}
resource "aws_iam_role_policy" "HAProxyFargateTaskRole_policy" {
    name ="${var.nlb_name}-HAProxyFargateTask-policy-${var.VPC_NAME}-
${var.PROJECT}"
    role = "${aws_iam_role.HAProxyFargateTaskRole.id}"
    policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "taskrole1",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:GetDeployablePatchSnapshotForInstance",
                "ssm:GetDocument",
                "ssm:GetManifest",
                "ssm:GetParameters",
                "ssm>ListAssociations",
                "ssm>ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
                "ssm:PutConfigurePackageResult",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetParametersByPath"
            ],
            "Resource": "*"
        },
        {
            "Sid": "taskrole2",
            "Effect": "Allow",
            "Action": [
                "logs>CreateLogGroup",
                "logs>CreateLogStream",
                "logs>DescribeLogGroups",
                "logs>DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "taskrole3",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        }
    ]
}
EOF
}
resource "aws_iam_role_policy" "HAProxyFargateplugin_policy" {
    name ="${var.nlb_name}-HAProxyFargateplugin-policy-${var.VPC_NAME}-
${var.PROJECT}"
    role = "${aws_iam_role.HAProxyFargateTaskRole.id}"
    policy = <<EOF
{
    "Version": "2012-10-17",

```

```

"Statement": [
    {
        "Sid": "fargateplugin1",
        "Effect": "Allow",
        "Action": [
            "ecs:RegisterTaskDefinition",
            "ecs>ListClusters",
            "ecs:DescribeContainerInstances",
            "ecs>ListTaskDefinitions",
            "ecs:DescribeTaskDefinition"
        ],
        "Resource": "*"
    },
    {
        "Sid": "fargateplugin2",
        "Effect": "Allow",
        "Action": [
            "ecs:StopTask",
            "ecs>ListContainerInstances"
        ],
        "Resource": "arn:aws:ecs:eu-west-1:${var.account_id}:cluster/*"
    },
    {
        "Sid": "fargateplugin3",
        "Effect": "Allow",
        "Action": "ecs:RunTask",
        "Resource": "arn:aws:ecs:eu-west-1:${var.account_id}:task-
definition/*"
    },
    {
        "Sid": "fargateplugin4",
        "Effect": "Allow",
        "Action": [
            "ecs:StopTask",
            "ecs:DescribeTasks"
        ],
        "Resource": "arn:aws:ecs:eu-west-1:${var.account_id}:task/*"
    },
    {
        "Sid": "PassRoleToWorker",
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource":
"arn:aws:iam::${var.account_id}:role/ECSFargateTaskExecutionRole"
    }
]

}
EOF
}
resource "aws_iam_role_policy" "HAProxyFargateEcr_policy" {
    name ="${var.nlb_name}-HAProxyFargateEcr-policy-${var.VPC_NAME}-
${var.PROJECT}"
    role = "${aws_iam_role.HAProxyFargateTaskRole.id}"
    policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ecrpolicy1",
            "Effect": "Allow",
            "Action": "ecr:*",
            "Resource": "*"
        },
        {
            "Sid": "ecrpolicy2",
            "Effect": "Allow",
            "Action": "ecr:GetImage",
            "Resource": "arn:aws:ecr:eu-west-1:${var.account_id}:repository/*
        }
    ]
}
EOF
}

```

```

        "Action": [
            "ecs:StopTask",
            "ecs>ListContainerInstances"
        ],
        "Resource": "arn:aws:ecs:eu-west-1:${var.account_id}:cluster/*"
    }
}
EOF
}

```

6.2.11 HAProxy TROUBLESHOOTING

6.2.11.1 HAproxy servers are created in vf-iedelivery-prod-ss-01 account.

| Resource Groups | | | | | | | | | | DevOps @ vf-iedelivery-prod-ss-01 |
|---|---|----------------------|---------------|-------------------|--|---|--------------|--|-----|-----------------------------------|
| | Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Pul | | |
| <input type="checkbox"/> Filter by tags and attributes or search by keyword | | | | | | | | | | |
| <input type="checkbox"/> | AG-PROD-HAProxy | i-017a91da8a41c3270 | t2.medium | eu-west-1a | ● running | ● 2/2 checks ... | None | ⚠️ | | |
| <input type="checkbox"/> | AG-PROD-HAProxy | i-03bd66f7b9b2e6dc6 | t2.medium | eu-west-1b | ● running | ● 2/2 checks ... | None | ⚠️ | | |
| <input type="checkbox"/> | AG-PRD2-HAProxy | i-0af6b256f6f64c07 | t2.medium | eu-west-1a | ● running | ● 2/2 checks ... | None | ⚠️ | | |
| <input type="checkbox"/> | AG-PRD2-HAProxy | i-02fa57d42f4e64017 | t2.medium | eu-west-1b | ● running | ● 2/2 checks ... | None | ⚠️ | | |
| <input checked="" type="checkbox"/> | AG-PRD1-HAProxy | i-058d5afbba90a5ca7 | t2.medium | eu-west-1a | ● running | ● 2/2 checks ... | None | ⚠️ | | |
| <input type="checkbox"/> | AG-PRD1-HAProxy | i-0e209b7404d774911 | t2.medium | eu-west-1b | ● running | ● 2/2 checks ... | None | ⚠️ | | |
| <input type="checkbox"/> | B-TRANSPARENT-SQUID-PROXY-CENTRALIZED-INT... | i-03c7fe9dad68781fd | t2.micro | eu-west-1a | ● stopped | | None | ⚠️ | | |
| <input type="checkbox"/> | B-TEST-CENTRALIZED-INTERNET-ACCESS | i-05eb0799ded4cc3c | t3.micro | eu-west-1b | ● running | ● 2/2 checks ... | None | ⚠️ | ec2 | |
| <input type="checkbox"/> | A-TEST-CENTRALIZED-INTERNET-ACCESS | i-0f4a3e948f6e6312a | t3.micro | eu-west-1a | ● running | ● 2/2 checks ... | None | ⚠️ | | |
| <input type="checkbox"/> | TRANSPARENT-SQUID-PROXY-CENTRALIZED-UNITED... | i-010009eaff79a474eh | t3.medium | eu-west-1a | ● running | ● 2/2 checks ... | None | ⚠️ | ec2 | |

6.2.11.2 If we want to check the configuration:

- Go to vf-iedelivery-sharedservices-vpc. In the module lbproxy, you will find the folder haproxyconfig, with different config files, one per environment (automation in place but not used yet for this).
- If you updated it, after that, run the lambda function created for s3 synchronization between haproxy and s3: [gdc-pcs-lambda_sync_haproxy_s3-PROD](#)
- Execute manually that lambda. In future we will be triggering it, but now it is better to just run it when need it.
- There is an important variable that manage all the haproxy configuration, go to the file called variables.tf in root and in the folder vars, in PROD.tfvars, you will see the value for those variables. The variables define the haproxy configuration, and they are used to create nlb listeners, target groups and security group rules. They are also used to create dynamically the config for haproxy, but this is still in progress (waiting for gcd name resolution to use it). Any modification must be done to those variables too.

6.2.11.3 Checking backend server and logs:

sudo tail -f /var/log/haproxy-traffic.log | grep -v healthcheck

sudo tail -f /var/log/haproxy-traffic.log | grep -v prod_10120

sftp -vvv -oPort=22 msieqnx@cch-sal-sftp.prd1.ieaws.vodafone.com

sftp -vvv -oPort=10120 msieqnx@localhost

sftp -vvv -oPort=10120 -oHostName=localhost [msieqnx@cch-sal-sftp.prd1.ieaws.vodafone.com](#)

curl -v localhost:port → for this you will need to update the ACL of that port, adding 127.0.0.1 and the haproxy ip address.

curl -v backendserver:port

6.2.11.4 Info for checking Aws CW logs:

SS TGW network interfaces:

- eni-047a452678be00bf1 198.19.221.90
- **eni-09c180fd0b84780b1 198.19.221.36**
-

NLB haproxy-PRD1 network interfaces:

- eni-05f2f15057c4267b2 198.19.220.118
- eni-066e62d2e13ecb495 198.19.220.42

EC2 HAProxy PRD1 network interfaces:

- eni-0974063289181051a 198.19.220.37
- eni-04d97f93ed0343a9a 198.19.220.104

sftp ip address: 10.181.72.20 and 10.181.66.190

Sftp-prd1 nlb:

- eni-06cc41f8608f4177d - 10.181.70.36
- eni-05543cb3fefef8dfc2 - 10.181.64.228

6.2.11.5 STATUS NOW:

Backend servers not working:

- 20101 - JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM 10.163.187.4:6543 – NOT WORKING
- 20111 - SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.163.184.4:1040 – NOT WORKING
- 20112 - SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.163.184.4:1050 – NOT WORKING
- 20113 - SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.163.184.4:1060 – NOT WORKING
- 20114 - SMSC-APP-SIT1-VF-IE.INTERNAL.VODAFONE.COM 10.163.184.4:1070 – NOT WORKING

DNS GDC NAME RESOLUTION IS NOT WORKIN.

6.3 02 VF IE SHARED SERVICES VPC (MYST)

SP documentation here: <https://confluence.sp.vodafone.com/x/UHACCg>

6.4 03 VF IE CENTRALIZED INTERNET ACCESS VPC

Error rendering macro 'toc'

java.lang.RuntimeException: com.ctc.wstx.exc.WstxUnexpectedCharException: Unexpected character ';' (code 59) expected '=' at [row,col {unknown-source}]: [1,14970]

6.4.1 1. INTRODUCTION

VFIE wants to centralize the internet access for all vpc's across aws accounts. In order to do that, a vpc called 'centralized internet access' will be created in VFIE SS account. The agreed solution is to use transparent proxies. In a transparent proxy deployment, the user's client software is unaware that it is communicating with a proxy. Users request Internet content as usual, without any special client configuration, and the proxy serves their requests. The transparent proxy intercepts incoming packets, establishes a connection with the origin server and returns requested content to the client. It returns content as if it came directly from the origin server.

The software uses for in the transparent proxy is Squid. **Squid** is a caching and forwarding HTTP web proxy. A client program (e.g. browser) either has to specify explicitly the proxy server it wants to use (typical for ISP customers), or it could be using a proxy without any extra configuration: "transparent caching", in which case all outgoing HTTP requests are intercepted by Squid and all responses are cached.

The delivery of this solution will be done in different releases:

1. **Architecture release 1.** The internet vpc will be composed of:
 - o 1 Public subnet / AZ with a route to IGW.
 - o 1 Private subnet /AZ with a route to ENI of transparent proxy.
 - o 1 Transparent proxy in each Public Subnet.
 - o 1 EC2 instance in each private subnet to test transparent proxy.
 - o TGW attachment to internet VPC and TGW route table.
2. **Architecture release 2.** A network load balancer and an AutoScaling group, with the transparent proxy as the launch configuration, will be added to the public subnet provide high availability and resilience.
3. **Architecture release 3.** All resources (network load balancer, AutoScaling group and transparent proxy) will be moved to a private subnet, with a route to a NAT GW that will be created in the public subnet.

6.4.2 2. CONFIGURATION

The code will be stored in the management account (vf-iedelivery-mgmt). A code commit repository has been created (<https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/vf-iedelivery-centralized-internet>), as well as a codepipeline (<https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-centralized-internet-prod-tf/view?region=eu-west-1>).

6.4.3 3. VPC CIDR

The Centralized Internet VPC will have the cidr: **10.181.22.0/23**

| Subnet | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|---------------|-------------------------|------------------------|--------------------------------------|--------------------------------------|-----------|
| | 10.181.22.0/27 | 255.255.255.224 | 10.181.22.0 - 10.181.22.31 | 10.181.22.1 - 10.181.22.30 | 30 |
| Private (Nat) | 10.181.22.32/27 | 255.255.255.224 | 10.181.22.32 - 10.181.22.63 | 10.181.22.33 - 10.181.22.62 | 30 |
| Private (Nat) | 10.181.22.64/27 | 255.255.255.224 | 10.181.22.64 - 10.181.22.95 | 10.181.22.65 - 10.181.22.94 | 30 |
| Public | 10.181.22.96/27 | 255.255.255.224 | 10.181.22.96 - 10.181.22.127 | 10.181.22.97 - 10.181.22.126 | 30 |
| Public | 10.181.22.128/27 | 255.255.255.224 | 10.181.22.128 - 10.181.22.159 | 10.181.22.129 - 10.181.22.158 | 30 |
| Private | 10.181.22.160/27 | 255.255.255.224 | 10.181.22.160 - 10.181.22.191 | 10.181.22.161 - 10.181.22.190 | 30 |
| Private | 10.181.22.192/27 | 255.255.255.224 | 10.181.22.192 - 10.181.22.223 | 10.181.22.193 - 10.181.22.222 | 30 |
| Private (Nat) | 10.181.22.224/27 | 255.255.255.224 | 10.181.22.224 - 10.181.22.255 | 10.181.22.225 - 10.181.22.254 | 30 |
| Private (Nat) | 10.181.23.0/27 | 255.255.255.224 | 10.181.23.0 - 10.181.23.31 | 10.181.23.1 - 10.181.23.30 | 30 |
| | 10.181.23.32/27 | 255.255.255.224 | 10.181.23.32 - 10.181.23.63 | 10.181.23.33 - 10.181.23.62 | 30 |
| | 10.181.23.64/27 | 255.255.255.224 | 10.181.23.64 - 10.181.23.95 | 10.181.23.65 - 10.181.23.94 | 30 |
| | 10.181.23.96/27 | 255.255.255.224 | 10.181.23.96 - 10.181.23.127 | 10.181.23.97 - 10.181.23.126 | 30 |
| | 10.181.23.128/25 | 255.255.255.128 | 10.181.23.128 - 10.181.23.255 | 10.181.23.129 - 10.181.23.254 | 126 |

6.4.4 4. Intercepting traffic

In each availability zone, the route table associated to the private subnet sends the outbound traffic to the Squid instance. Squid intercepts the requested domain, then applies the following filtering policy:

- For HTTP requests, Squid retrieves the host header field included in all HTTP/1.1 request messages. This specifies the Internet host being requested.
- For HTTPS requests, the HTTP traffic is encapsulated in a TLS connection between the instance in the private subnet and the remote host. Squid cannot retrieve the host header field because the header is encrypted. A feature called [SslBump](#) would allow Squid to decrypt the traffic, but this would not be transparent for the client because the certificate would be considered invalid in most cases. The feature I use instead, called [SslPeekAndSplice](#), retrieves the Server Name Indication (SNI) from the TLS initiation. The SNI contains the requested Internet host. As a result, Squid can make filtering decisions without decrypting the HTTPS traffic.

- **Note 1:** Some older client-side software stacks do not support SNI. Here are the minimum versions of some important stacks and programming languages that support SNI: Python 2.7.9 and 3.2, Java 7 JSSE, wget 1.14, OpenSSL 0.9.8j, cURL 7.18.1
- **Note 2:** TLS 1.3 introduced an optional extension that allows the client to encrypt the SNI, which may prevent Squid from intercepting the requested domain.

The Peek and Splice feature looks at the TLS Client Hello message and SNI info (if any), sends an identical or similar (to the extent possible) Client Hello message to the server, and then looks at the TLS Server Hello message. The final decision to splice, bump, or terminate the connection can be made at any of those steps (but what Squid does at step N affects its ability to splice or bump at step N+1!).

| Action | Applicable processing steps | Description |
|------------------|-----------------------------------|--|
| peek | step1, step2 | When a peek rule matches during step1, Squid proceeds to step2 where it parses the TLS Client Hello and extracts SNI (if any). When a peek rule matches during step 2, Squid proceeds to step3 where it parses the TLS Server Hello and extracts server certificate while preserving the possibility of splicing the client and server connections; peeking at the server certificate usually precludes future bumping (see Limitations). |
| splice | step1, step2, and sometimes step3 | Become a TCP tunnel without decoding the connection. The client and the server exchange data as if there is no proxy in between. |
| stare | step1, step2 | When a stare rule matches during step1, Squid proceeds to step2 where it parses the TLS Client Hello and extracts SNI (if any). When a stare rule matches during step2, Squid proceeds to step3 where it parses the TLS Server Hello and extracts server certificate while preserving the possibility of bumping the client and server connections; staring at the server certificate usually precludes future splicing (see Limitations). |
| bump | step1, step2, and sometimes step3 | Establish a TLS connection with the server (using client SNI, if any) and establish a TLS connection with the client (using a mimicked server certificate). However , this is not what actually happens right now if a bump rule matches during step1. See bug 4327 |
| terminate | step1, step2, step3 | Close client and server connections. |

6.4.5 5. Certificate

To enable this module, Squid requires that you provide a certificate, though it will not be used to decode HTTPS traffic. The solution creates a certificate using OpenSSL.

```
mkdir /etc/squid/ssl
cd /etc/squid/ssl
openssl genrsa -out squid.key 4096
openssl req -new -key squid.key -out squid.csr -subj
"/C=XX/ST=XX/L=squid/O=squid/CN=squid"
openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
```

```
cat squid.key squid.crt >> squid.pem
```

6.4.6 6. Squid configuration

The following code shows the Squid configuration file. For HTTPS traffic, note the ssl_bump directives instructing Squid to “peek” (retrieve the SNI) and then “splice” (become a TCP tunnel without decoding) or “terminate” the connection depending on the requested host.

```
# Handling HTTP requests
http_port 3128
http_port 3129 intercept
acl allowed_http_sites dstdomain "/etc/squid/whitelist.txt"
http_access allow allowed_http_sites

# Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name "/etc/squid/whitelist.txt"
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
http_access deny all
```

6.4.7 7. White List

A S3 bucket (named [vf-iedelivery-centralized-internet-vpc-267040142128-123456](#)) has been created for storing the whitelist file and the squid config file. A lambda function has been implemented to create those files automatically. The text file located at /etc/squid/whitelist.txt contains the list of whitelisted domains, with one domain per line, that list is synchronized with S3. The allowed web domains are an input in terraform code.

The solution runs the following script every minute on the Squid instances to download and update the Squid configuration from S3. This makes it easy to maintain the Squid configuration from a central location. Note that it first validates the files with squid -k parse and then reload the configuration with squid -k reconfigure if no error was found.

```
cp /etc/squid/* /etc/squid/old/
aws s3 sync s3://<s3-bucket> /etc/squid
squid -k parse && squid -k reconfigure || (cp /etc/squid/old/* /etc/squid/; exit 1)
```

6.4.8 8. Iptables

Squid listens on port 3129 for HTTP traffic and 3130 for HTTPS. Because Squid cannot directly listen to 80 and 443, you have to redirect the incoming requests from instances in the private subnets to the Squid ports using iptables.

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
```

6.4.9 RELEASE 1

6.4.9.1

- [1.INTRODUCTION](#)
- [2.DESIGN](#)
- [3. CLOUDFORMATION PARAMETERS AND USER DATA](#)
- [4. CONNECTIVITY TEST](#)
 - [4.1 Connectivity from EC2 instance in private subnet in Internet VPC](#)
 - [4.2 Connectivity from EC2 instance in private subnet in a cross-account VPC](#)

6.4.9.2 1.INTRODUCTION

For the first release, just one transparent proxy will be deployed per public subnet. This design DOES NOT provide high availability and IS NOT resilience, as there are just one point of failure. However, it will be used to test the proxy configuration and that the internet traffic from a instance in another account is route to the internet VPC and redirect to the corresponding web server.

The code is stored in the management account (vf-iedelivery-mgmt). A code commit repository has been created (<https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/vf-iedelivery-centralized-internet>), as well as a codepipeline (<https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-centralized-internet-prod-tf/view?region=eu-west-1>).

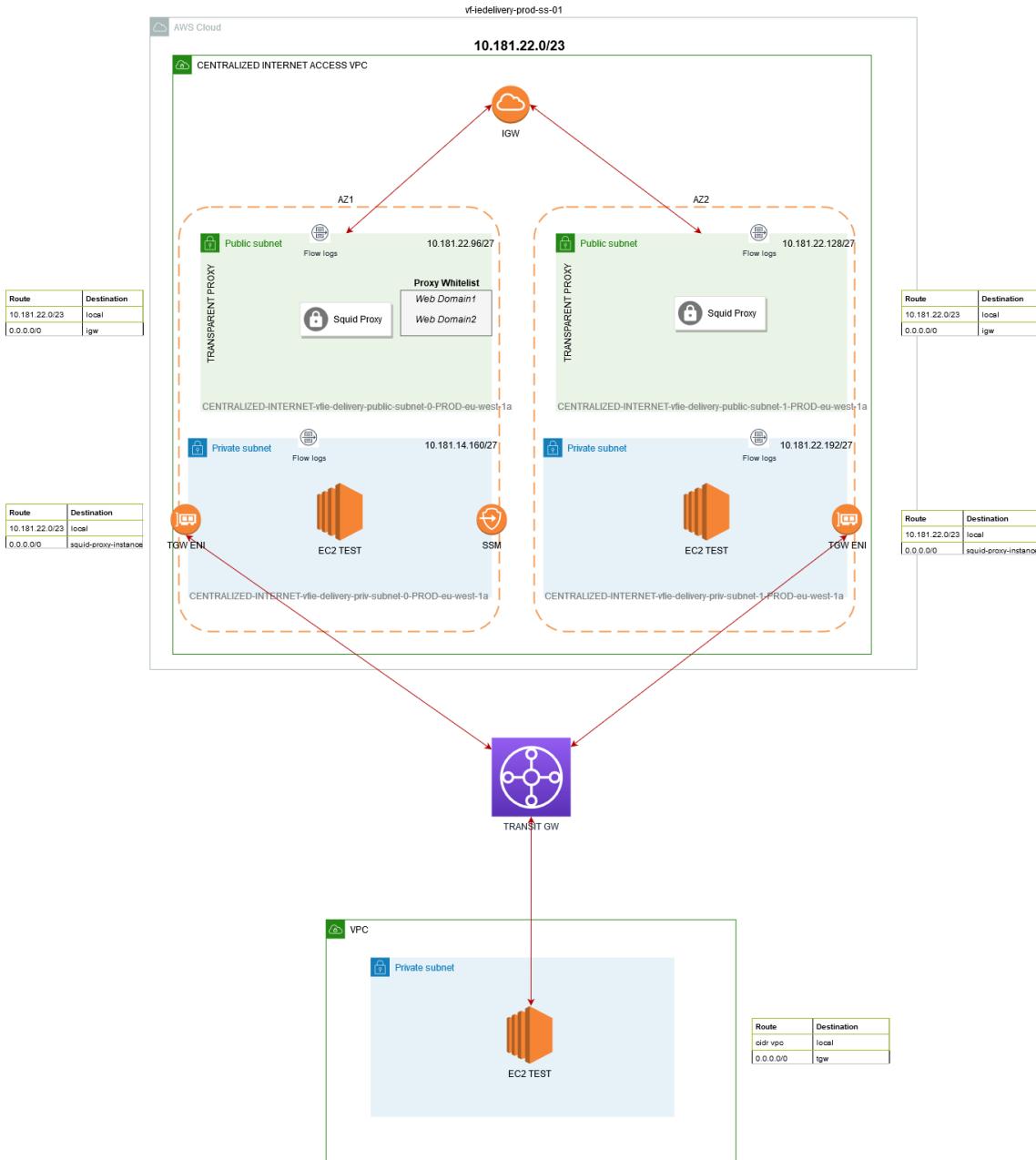
Resources created via terraform code:

- Internet VPC with cidr 10.181.22.0/23
- IGW and NAT GW
- 1 public subnet per AZ
- 1 private subnet per AZ with internet access through NAT GW (it is not shown in below design because it is not used in this release)
- 1 private subnet per AZ.
- All corresponding route tables per subnet and routes.
- SSM endpoints (ssm, ec2, ec2messages, ssessages, s3 endpoints) and corresponding security groups and IAM role for EC2.
- **CloudFormation stack with proxy configuration:**
- S3 bucket vf-iedelivery-centralized-internet-vpc-267040142128-123456
- Lambda function (and corresponding IAM role for lambda) to add automatically proxy config file and whitelist as S3 object in above S3 bucket.
- Whitelist and squid config as Custom::S3Object. Configuration [here](#).
- Security group to allow HTTP and HTTPS traffic from private instances (VFIE cidr = 10.181.0.0/16) to Proxy instances.
- Security group to allow outbound HTTP and HTTPS traffic in private instances.
- Proxy Instance with a network interface (public IP) and user data to install and start squid software.
- Test Instance in private subnets to test internet connectivity inside Internet VPC.
- New route added in private subnet route table: 0.0.0.0/0 → Proxy instance

Additionally, the transit **gateway repository** (<https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-transit-gateway/browse?region=eu-west-1>) has been also modified to add:

- New TGW VPC attachment to Centralized Internet VPC, where the vpc id is "vpc-066cf10e6e20a84cc" and subnets ids (private subnets) are "subnet-059861dbd8424d4dd" and "subnet-06adf1cd74fc855fc".
- Corresponding TGW Route table for Centralized Internet VPC, that has route to all environments but SS VPC. Meaning Centralized Internet VPC can send traffic to test, pre-prod and prod.
- Test, pre-prod and prod VPC default routes (0.0.0.0/0) point now to Internet VPC.
- Internet VPC route tables have been modified to add routes with destination prod, pre-prod and test cidr and target TGW.

6.4.9.3 2.DESIGN



6.4.9.4 3. CLOUDFORMATION PARAMETERS AND USER DATA

These are the parameters that terraform passes to the cloudformation stack

```
parameters = {
  Amild = var.hardened_linux_amazon_2_AMI
  InstanceType = "t3.medium"
  WhitelistDomains = ".amazon.com, .vodafone.com, .google.com, .amazonaws.com" #testing purposes
  VPC_ = aws_vpc.internet_vpc.id
  VFIECidrBlock = "10.181.0.0/16"
  PrivateSubnetA = element(aws_subnet.internet_subnet_priv.*.id, 0)
  PrivateSubnetB = element(aws_subnet.internet_subnet_priv.*.id, 1)
  PublicSubnetA = element(aws_subnet.internet_subnet_public.*.id, 0)
  PublicSubnetB = element(aws_subnet.internet_subnet_public.*.id, 1)
  LoggingBucket = local.logging_bucket["name"]
  NATInstanceProfile = aws_iam_instance_profile.ec2_ssm.id
  SSMSecurityGroupId = aws_security_group.ssm_endpoint_sg.id
  PrivateRouteTableId = element(aws_route_table.subnet_priv_route_table.*.id, 0)
```

```
PrivateRouteTableBld = element(aws_route_table.subnet_priv_route_table.*.id, 1)
}
```

And this is the user data of the transparent proxy:

```
#!/bin/bash -xe
sudo yum update -y --security

#disable source / destination check in proxy instance
instanceid=`curl -s http://169.254.169.254/latest/meta-data/instance-id`
aws ec2 modify-instance-attribute --no-source-dest-check --instance-id $instanceid --
region eu-west-1

#Install and Start proxy
sudo yum install -y squid
sudo systemctl start squid || service squid start
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130

#create a certificate for the sslBump squid module
sudo mkdir /etc/squid/ssl
sudo openssl genrsa -out /etc/squid/ssl/squid.key 4096
sudo openssl req -new -key /etc/squid/ssl/squid.key -out /etc/squid/ssl/squid.csr -subj
"/C=XX/ST=XX/L=squid/O=squid/CN=squid"
sudo openssl x509 -req -days 3650 -in /etc/squid/ssl/squid.csr -signkey
/etc/squid/ssl/squid.key -out /etc/squid/ssl/squid.crt
sudo touch /etc/squid/ssl/squid.pem
sudo chmod 777 /etc/squid/ssl/squid.pem
sudo cat /etc/squid/ssl/squid.key /etc/squid/ssl/squid.crt >> /etc/squid/ssl/squid.pem

#Refres the configuration from S3
sudo mkdir /etc/squid/old
sudo touch /etc/squid/squid-conf-refresh.sh
sudo chmod 777 /etc/squid/squid-conf-refresh.sh
sudo cat > /etc/squid/squid-conf-refresh.sh << 'EOF'
sudo cp /etc/squid/* /etc/squid/old/
sudo aws s3 sync s3://${S3Bucket} /etc/squid
sudo /usr/sbin/squid -k parse && /usr/sbin/squid -k reconfigure || (cp /etc/squid/old/*
/etc/squid/; exit 1)
EOF
sudo chmod +x /etc/squid/squid-conf-refresh.sh
sudo /etc/squid/squid-conf-refresh.sh

# Schedule tasks
sudo cat > ~/mycron << 'EOF'
* * * * * /etc/squid/squid-conf-refresh.sh
0 0 * * * sleep $($RANDOM % 3600); yum -y update --security
0 0 * * * /usr/sbin/squid -k rotate
EOF
sudo crontab ~/mycron
sudo rm ~/mycron
```

6.4.9.5 4. CONNECTIVITY TEST

6.4.9.5.1 4.1 Connectivity from EC2 instance in private subnet in Internet VPC

Accessing allowed domains:

Session ID: albamarria.diazfernandez@vodafone.com-0de5100d14e6a52ec

Instance ID: i-0c4034899e06da3b5

```
sh-4.2$ curl http://www.amazon.com
sh-4.2$ curl http://www.amazon.com
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>CloudFront</center>
</body>
</html>
sh-4.2$ curl http://www.amazon.com
sh-4.2$ curl http://www.amazon.com
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>CloudFront</center>
</body>
</html>
sh-4.2$ curl https://www.amazon.com
```

Session ID: albamarria.diazfernandez@vodafone.com-0de5100d14e6a52ec

Instance ID: i-0c4034899e06da3b5

```
sh-4.2$ curl https://www.amazon.com
<!--
      To discuss automated access to Amazon data please contact api-services-support@amazon.com.
      For information about migrating to our APIs refer to our Marketplace APIs at https://developer.amazon.amazon.com/gp/advertising/api/detail/main.html/ref=rm_5_ac for advertising use cases.
-->
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Sorry! Something went wrong!</title>
  <style>
    html, body {
      padding: 0;
      margin: 0
    }

    img {
      border: 0
    }

    #a {
      background: #232f3e;
      padding: 11px 11px 11px 192px
    }

    #b {
      position: absolute;
      left: 22px;
      top: 12px
    }

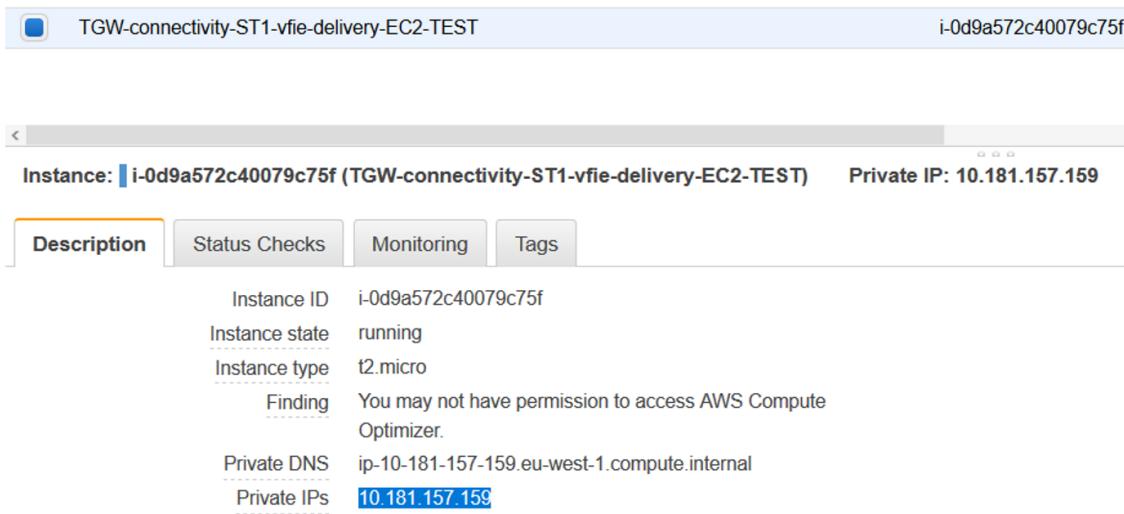
    #c {
      position: relative;
      max-width: 800px;
      padding: 0 40px 0 0
    }

    #e, #f {
      height: 35px;
      border: 0;
```

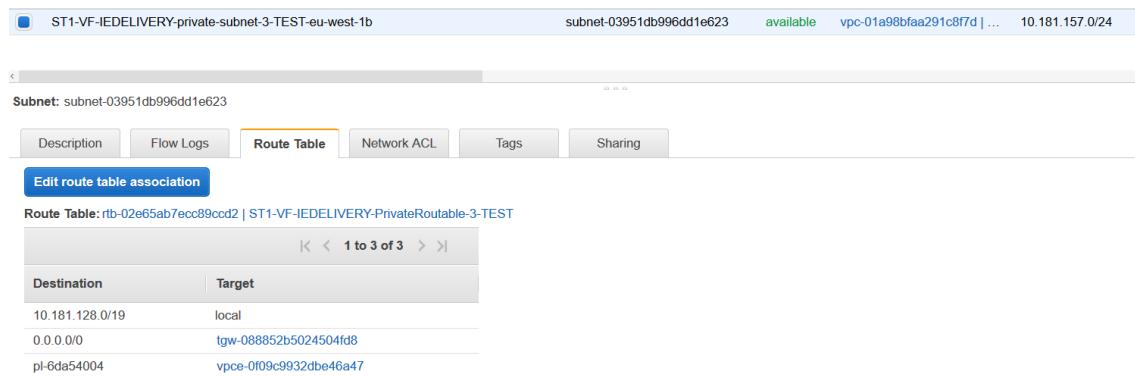
Trying to access not allowed domains:

The connectivity from an EC2 in another AZ is the same.

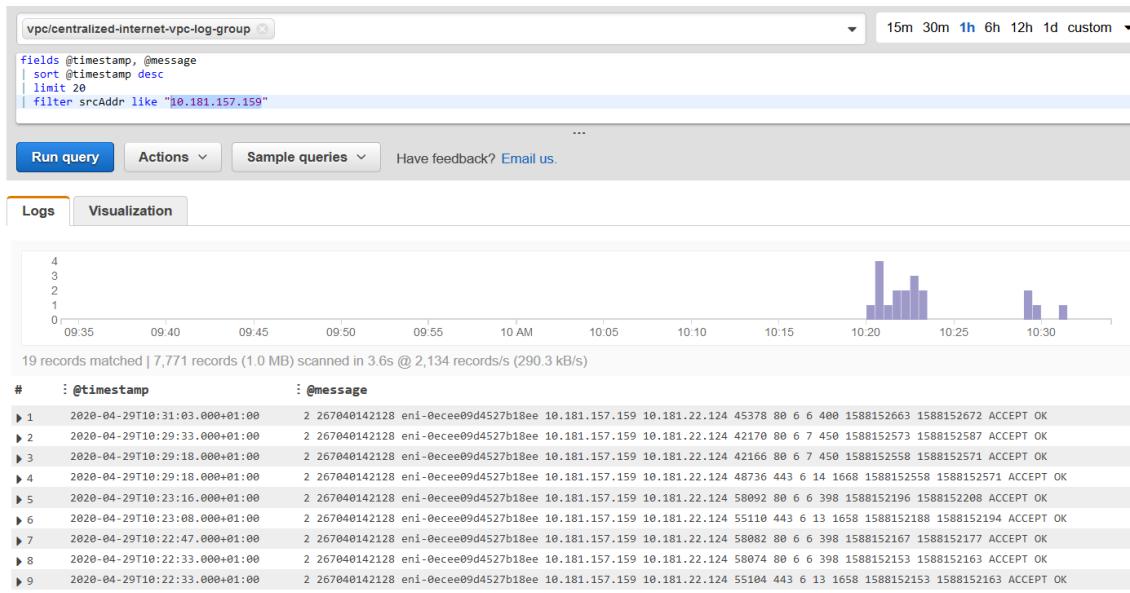
6.4.9.5.2 4.2 Connectivity from EC2 instance in private subnet in a cross-account VPC
A test EC2 instance has been used to check connectivity:



With route table:



If we create an SSM session with that instance, and then we type "curl http://www.amazon.com", "curl https://www.amazon.com" and "curl http://terraform.io" (the first two are allowed, but the third one is not allowed), we will see similar results than in 4.1. We can analyse the traffic if we access the log group created for the internet vpc (<https://eu-west-1.console.aws.amazon.com/cloudwatch/home?region=eu-west-1#logStream:group=vpc/centralized-internet-vpc-log-group>). If we access Insights in CloudWatch and we filter with the ip address of the test EC2 instance (10.181.157.159) we can see the traffic, and that it has been accepted. We can also see to which eni the traffic has been routed.



6.4.10 1.CHANGES

6.4.10.1 1.1.Security

In order to increment the security for this architecture, the transparent proxies have been moving into a **private subnet with NAT GW access**. The corresponding connectivity tests have been done to check that everything works as expected.

6.4.10.2 2.1.High Availability and Fault Tolerance - Solution 1

In order to provide High availability and fault tolerance, the following solution has been tested:

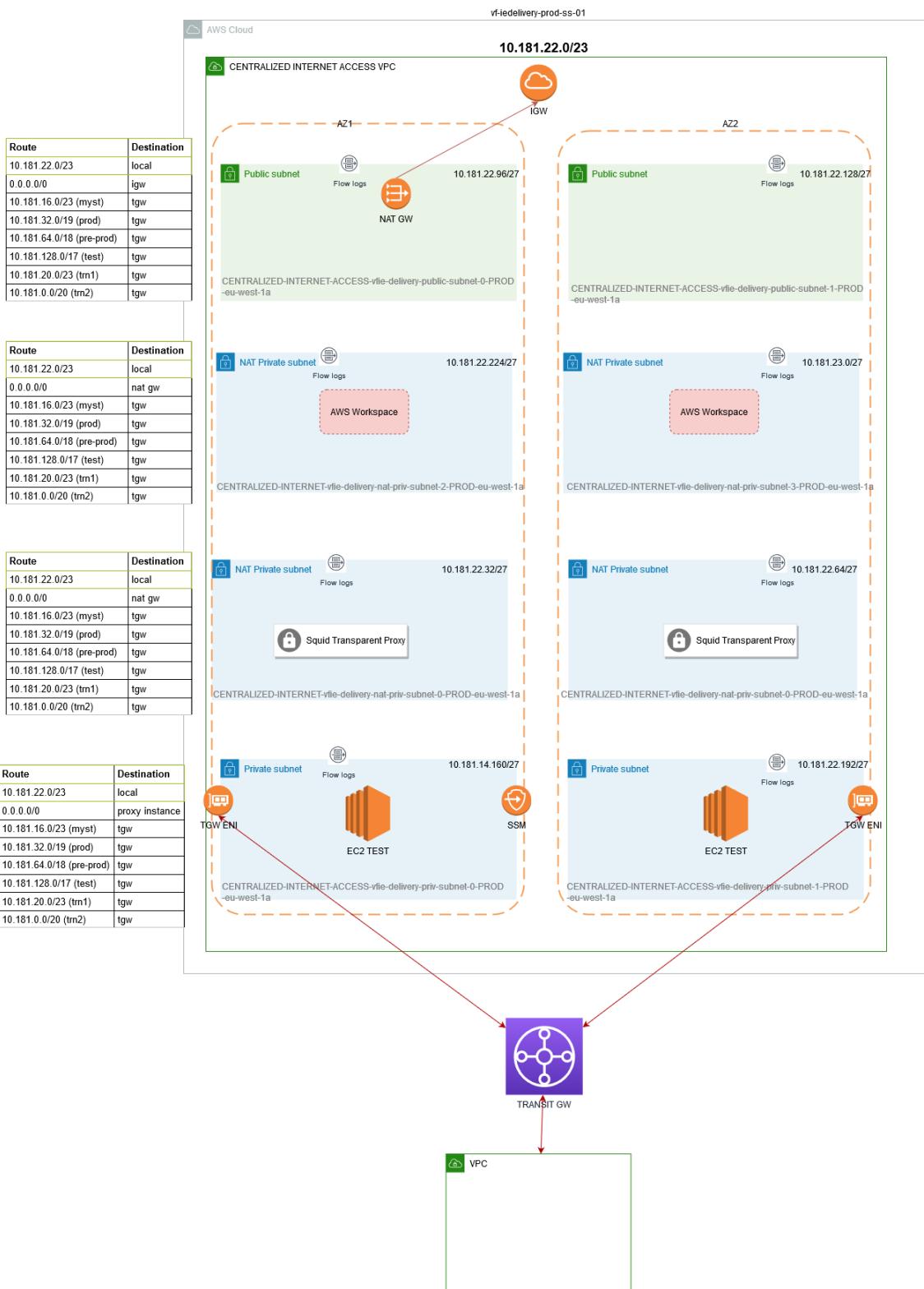
- Creation of a network load balancer with a listener on port 80 and protocol TCP.
- Target group for instance type (transparent proxy).
- Auto Scaling group with proxy configuration as launch configuration.

The main idea was to redirect the traffic from the private subnet where the traffic from the TGW came, to the transparent proxy instance, that will have a network load balancer in front. So, the route table for the private subnet with TGW ENI was modified, using as destination for the default route (0.0.0.0/0) the NLB ENI that is created by AWS when the NLB is created and deployed in a AZ.

Problem: the NLB ENI created by AWS is not meant to be used as the next hop in the route table, as it is used internally by the NLB when this resource is re-directing traffic between different AZ. AWS-managed ENI are not modifiable, so it is not possible to add any security group to this interface. The result is that, the traffic reaches the NLB ENI, but after that, it is dropped, so it never reached the transparent proxy.

6.4.10.3 3.1. AWS Workspace

As a temporary solution for testing purposes, the project needs to deploy AWS Workspaces. Two additional private subnets with Internet Access through the NAT GW have been deployed. All the routing logic to connect the Internet VPC with the rest of environment vpc's have been implemented.



6.4.11 Transparent Proxy Whitelist

| Whitelist | |
|---------------|--------------------|
| AWS Endpoints | .amazonaws.com |
| Yum | .aws.ce.redhat.com |

| Whitelist | |
|--------------------------|-------------|
| Artifactory Repositories | .bintray.io |

6.5 04 VF IE PORT RANGES

To standardise the ranges being used by each application, the below port ranges have been assigned on a per application basis. This applies to application flows in both directions with ranges being assigned to on-prem applications as well.

Please add additional ranges as required when new applications are required.

| PORTS | |
|----------------------------------|---------------|
| APP | Port Range |
| CCH | 10000 - 10999 |
| FSL VFIE Legacy | 11000 - 11099 |
| FSL AWS | 11100 - 11199 |
| MEH | 12000 - 12999 |
| UFE | 13000 - 13999 |
| CIAM | 14000 - 14999 |
| OSB | 15000 - 15999 |
| MFT | 16000 - 16999 |
| MEDIAROOM | 17000 - 17999 |
| GISMSH | 18000 - 18999 |
| TIBCO | 19000 - 19999 |
| I1 PROXY (JINNY / SMSC / RADIUS) | 20000 - 20999 |
| CDIRECT | 21000 - 21999 |
| MML | 22000 - 22999 |
| AUA | 23000 - 23999 |
| EPRS | 24000 - 24999 |
| MEDIATION ZONE | 25000 - 25999 |
| CIAM | 26000 - 26999 |
| BSS | 27000 - 27999 |
| LOTUS NOTES | 28000 - 28999 |
| TOOLING | 29000 - 29999 |
| VBOL | 30000 - 30999 |
| VTV | 31000 - 31999 |
| BOP | 32000 - 32999 |

| | |
|----------------------|---------------|
| WRM a.k.a ROBIGA | 34000 - 34099 |
| VFGP DMaaS NiFi | 34100 - 34199 |
| LDR | 34200 - 34299 |
| MIS | 34300 - 34399 |
| LEA | 34400 - 34499 |
| MEC | 34500 - 34599 |
| WEDO/RA | 34600 - 34699 |
| ER-IF | 34700 - 34799 |
| HGW | 34800 - 34899 |
| SVA | 34900 - 34999 |
| DPC | 35000 - 35099 |
| DRCC OCI | 35100 - 35199 |
| TURBO CHARGING | 35200 - 35299 |
| DRA | 35300 - 35399 |
| PCRF | 35400 - 35499 |
| Technotree | 35500 - 35599 |
| NGIN | 35600 - 35699 |
| E///PG | 35700 - 35799 |
| NPT | 35800 - 35899 |
| ESIGN | 36000 - 36999 |
| OneApp | 37000 - 37099 |
| Singleserve | 37100 - 37199 |
| Bill Archive Manager | 37200 - 37299 |
| GNP | 38000 - 38099 |
| GNM | 38100 - 38199 |
| EVO | 40200 - 40299 |
| POLLERS | 40100 - 40199 |
| PORTAL | 48000 - 48099 |
| Security / Pentest | 60000-60999 |
| FinHub | 48100 - 48199 |
| WEF | 48200 - 48299 |
| entitee | 48300 - 48399 |
| mTAS | 48400 - 48499 |
| IP Works | 48500 - 48599 |

| | |
|-----------|---------------|
| Radius | 48600 - 48699 |
| Optenet | 48700 - 48799 |
| SecureNet | 48800 - 48899 |
| CaaS DXL | 49000 - 49099 |
| TaaS | 49100 - 49199 |
| Grafana | 49200 - 49299 |
| SOM | 49300 - 49399 |
| OMS | 49400 - 49499 |

6.6 05 VF IE NLB DESIGN

- [Overview](#)
- [Solution Design](#)
- [DNS Configuration](#)
 - [Current Configuration](#)
 - [New Configuration](#)
- [NLB Configuration](#)
 - [Overview](#)
 - [Production](#)
 - [Summary](#)
 - [Target Groups](#)
 - [PRD1](#)
 - [Summary](#)
 - [Target Groups](#)
 - [PRD2](#)
 - [Summary](#)
 - [Target Groups](#)
 - [SIT1](#)
 - [Summary](#)
 - [Target Groups](#)
 - [SIT2](#)
 - [Summary](#)
 - [Target Groups](#)
 - [SIT3](#)
 - [Summary](#)
 - [Target Groups](#)
 - [SIT4](#)
 - [Summary](#)
 - [Target Groups](#)

- [Summary](#)
- [Target Groups](#)
- [DEV1](#)
 - [Summary](#)
 - [Target Groups](#)
- [Open Queries](#)

6.6.1 Overview

The Shared Services NLBs provide access into all the VFIE AWS Hosted applications, and directs the communication based on the ports used in the flow. The NLB that is hit is based on the FQDN of the request with an NLB created per account.

Under the current implementation, there is an AWS hard limit of 50 ports per NLB which we are quickly approaching. There is no option to increase this limit so it has been proposed to create additional NLB's and break down the inbound and outboundPr flows based on the traffic type. This will greatly increase the bandwidth in regards ports in use. In the future, AWS may increase this port limit but this solution should cover us until this happens.

6.6.2 Solution Design

Reference: [98c - Solution Design - VCI Public Cloud Services - Vodafone Global Confluence \(PCS Internal\)](#)

6.6.3 DNS Configuration

6.6.3.1 Current Configuration

| Traffic Type | Wildcard DNS | Destination NLB |
|--------------|----------------------------------|----------------------------|
| INBOUND | *.<ENV>.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-NLB |
| OUTBOUND | <ENV>.haproxy.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-NLB |

6.6.3.2 New Configuration

| Traffic Type | Wildcard DNS | Destination NLB | Status | Comments |
|--------------|---|--|----------|----------|
| INBOUND GUI | *.<ENV>.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-NLB | Existing | |
| INBOUND API | *.<ENV>-apii.ieaws.vodafone.com *.sit3-apii.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-API-NLB VF-IEDELIVERY-SS-<ENV>-APII-NLB | New | |
| INBOUND SFTP | *.<ENV>-sftp.ieaws.vodafone.com *.sit3-sfti.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-SFTP-NLB VF-IEDELIVERY-SS-<ENV>-SFTI-NLB | New | |
| INBOUND DB | *.<ENV>-DB.ieaws.vodafone.com *.sit3-dbi.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-DB-NLB | New | |

| Traffic Type | Wildcard DNS | Destination NLB | Status | Comments |
|---------------|---|--|--------|----------|
| | | VF-IEDELIVERY-SS-<ENV>-DBI-NLB | | |
| OUTBOUND API | <ENV>-api-out.ssnlb.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-API-OUT-NLB VF-IEDELIVERY-SS-<ENV>-APIO-NLB | New | |
| OUTBOUND SFTP | <ENV>-sftp-out.ssnlb.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-SFTP-OUT-NLB VF-IEDELIVERY-SS-<ENV>-SFTO-NLB | New | |
| OUTBOUND DB | <ENV>-db-out.ssnlb.ieaws.vodafone.com | VF-IEDELIVERY-SS-<ENV>-DB-OUT-NLB VF-IEDELIVERY-SS-<ENV>-DBO-NLB | New | |

6.6.4 NLB Configuration

6.6.4.1 Overview

Under the new design, the NLB's will be broken down into 7 distinct NLB's per environment. All inbound and outbound interfaces should go via these new NLB's.

6.6.4.2 Production

6.6.4.2.1 Summary

| Destination NLB | Count |
|------------------------------------|--------------------|
| VF-IEDELIVERY-SS-PROD-NLB | 8 |
| VF-IEDELIVERY-SS-PROD-API-NLB | 7 |
| VF-IEDELIVERY-SS-PROD-SFTP-NLB | 5 |
| VF-IEDELIVERY-SS-PROD-DB-NLB | 9 |
| VF-IEDELIVERY-SS-PROD-API-OUT-NLB | 15 |
| VF-IEDELIVERY-SS-PROD-SFTP-OUT-NLB | 9 |
| VF-IEDELIVERY-SS-PROD-DB-OUT-NLB | 5 |
| N/A | 3 (Needs removal?) |

6.6.4.2.2 Target Groups

| Name | Port | Protocol | Target type | Existing Load balancer | New Load balancer | Direction | Destination IP | Comment |
|---------------|------|----------|-------------|------------------------|-------------------|-----------|----------------|----------------|
| SS-PROD-from- | 443 | TCP | IP | VF-IEDELIVERY-SS- | N/A | INBOUND | N/A | Can be removed |

| | | | | | | | | |
|---|-----------|-----|----|----------------------------|------------------------------------|----------|-----|--|
| <u>1030 1-to- 443</u> | | | | PROD-NLB | | | | |
| <u>SS- PRO D- from- 1030 2-to- 443</u> | 443 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | N/A | INBOUND | N/A | Can be removed |
| <u>SS- PRO D- from- 1030 3-to- 443</u> | 443 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | N/A | INBOUND | N/A | Can be removed |
| <u>SS- PRO D- from- 3020 1-to- 9443</u> | 9443 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-API-NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 8011- to- 8011</u> | 8011 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-API-NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 1100 1-to- 7004</u> | 7004 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-API-OUT-NLB | OUTBOUND | | FSL being migrated to AWS so temporary |
| <u>SS- PRO D- from- 1210 3-to- 1644 3</u> | 1644 3 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-API-OUT-NLB | OUTBOUND | | |
| <u>SS- PRO D- from- 1300 1-to- 8005</u> | 8005 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-API-OUT-NLB | OUTBOUND | | |

| | | | | | | | | |
|---|------|-----|----|---|---|--------------|--|--|
| <u>SS- PRO D- from- 1510 1-to- 3005 0</u> | 3005 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2010 1-to- 6543</u> | 6543 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2011 1-to- 1040</u> | 1040 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2011 2-to- 1050</u> | 1050 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2011 3-to- 1060</u> | 1060 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2011 4-to- 1070</u> | 1070 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2900 1-to- 8090</u> | 8090 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | | |
| <u>SS- PRO D- from-</u> | 8090 | TCP | IP | VF- IEDELIVE RY-SS- | VF- IEDELIVE RY-SS- PROD- | OUTBOU ND | | |

| | | | | | | | | |
|---|------|-----|----|----------------------------|------------------------------------|----------|-----|--|
| <u>2900 2-to- 8090</u> | | | | PROD-NLB | API-OUT-NLB | | | |
| <u>SS- PRO D- from- 2900 3-to- 8181</u> | 8181 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-API-OUT-NLB | OUTBOUND | | |
| <u>SS- PRO D- from- 3005 3-to- 3005 3</u> | 3005 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-API-OUT-NLB | OUTBOUND | | OSB API - Should assign updtaed port to align with design - 15102? |
| <u>SS- PRO D- from- 1030 4-to- 1521</u> | 1521 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-DB-NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 1030 5-to- 2484</u> | 2484 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-DB-NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 2210 2-to- 3300 1</u> | 3300 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-DB-OUT-NLB | OUTBOUND | | |
| <u>SS- PRO D- from- 2210 3-to- 3300 1</u> | 3300 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-DB-OUT-NLB | OUTBOUND | | |
| <u>SS- PRO D- from- 2210 4-to-</u> | 3300 | TCP | IP | VF-IEDELIVE RY-SS-PROD-NLB | VF-IEDELIVE RY-SS-PROD-DB-OUT-NLB | OUTBOUND | | |

| | | | | | | | | |
|---|-----------|-----|----|---|--|--------------|-----|--|
| <u>3300 1</u> | | | | | | | | |
| <u>SS- PRO D- from- 2210 5-to- 3300 1</u> | 3300 1 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD-DB- OUT-NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2210 6-to- 3300 1</u> | 3300 1 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD-DB- OUT-NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 1001 1-to- 1001 1</u> | 1001 1 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 3000 1-to- 443</u> | 443 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 443- to- 443</u> | 443 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 7001- to- 7001</u> | 7001 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 8443- to- 8443</u> | 8443 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | N/A | |

| | | | | | | | | |
|--|------|-----|----|--|--|--------------|-----|--------------------------|
| <u>SS- PRO D- from- 9001- to- 9001</u> | 9001 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 1012- 0-to- 22</u> | 22 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- SFTP-NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 3010- 1-to- 22</u> | 22 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- SFTP-NLB | INBOUND | N/A | |
| <u>SS- PRO D- from- 2700- 1-to- 22</u> | 22 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- SFTP- OUT-NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 2800- 1-to- 22</u> | 22 | TCP | IP | VF- IEDELIVE RY-SS- PROD- NLB | VF- IEDELIVE RY-SS- PROD- SFTP- OUT-NLB | OUTBOU ND | | |
| <u>SS- PRO D- from- 3450- 1-to- 22- SFTP</u> | 22 | TCP | IP | VF- IEDELIVE RY-SS- PROD- ONPREM- NLB | VF- IEDELIVE RY-SS- PROD- SFTP- OUT-NLB | OUTBOU ND | | MEC |
| SS- PRO D- from- 3200- 1-to- 443 | 3200 | TCP | IP | N/A | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | N/A | New for BSP - GUI |
| SS- PRO D- from- | 3200 | TCP | IP | N/A | VF- IEDELIVE RY-SS- | INBOUND | N/A | New for BSP - SFTP |

| | | | | | | | | | |
|---|------|-----|----|-----|-------------------------------|---------|-----|--|---|
| 3200 2-to- 22 | | | | | PROD-SFTP-NLB | | | | |
| SS-PRO D-from- 3200 3-to- 443 | 3200 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-API-NLB | INBOUND | N/A | | New for BSP - API |
| SS-PRO D-from- 3200 4-to- 3200 4 | 3200 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | N/A | | New for BSP - Postgres Access |
| SS-PRO D-from- 3200 5-to- 3200 5 | 3200 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | N/A | | New for BSP - MSSQL Access |
| SS-PRO D-from- 3200 6-to- 3200 6 | 3200 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | N/A | | New for BSP - Document DB Access |
| SS-PRO D-from- 3470 1-to- 443 | 443 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | N/A | | New for BSP - EVO |
| SS-PRO D-from- 3480 1-to- 9499 | 9499 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | N/A | | New for BSP - Poller |
| SS-PRO D-from- 587- | 587 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-API-NLB | INBOUND | N/A | | SES - New for Pega |

| | | | | | | | | | |
|-------------------------------|--------|-----|----|-----|--------------------------------|---------|-----|--------------------|--|
| to-587 | | | | | | | | | |
| SS-PROD-from-588-to-443 | 588 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-API-NLB | INBOUND | N/A | SQS - New for Pega | |
| SS-PROD-from-3500 1-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-NLB | INBOUND | N/A | New for DPC | |
| SS-PROD-from-3500 2-to-3500 2 | 3500 2 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | N/A | New for DPC | |
| SS-PROD-from-3600 1-to-443 | 3600 1 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-API-NLB | INBOUND | N/A | New for eSign | |
| SS-PROD-from-3600 2-to-3600 2 | 3500 2 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | N/A | New for eSign | |
| SS-PROD-from-3400 1-to-443 | 443 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-NLB | INBOUND | N/A | New for Robiga | |
| SS-PROD-from-3400 2-to-443 | 443 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-API-NLB | INBOUND | N/A | New for Robiga | |

| | | | | | | | | |
|---------------------------|----|-----|----|-----|------------------------------------|----------|--|----------------------------|
| SS-PROD-from-3402 2-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-NLB | INBOUND | N/A | New for Robiga |
| SS-PROD-from-3410 1-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-OUT-NLB | OUTBOUND | 139.47.233.128/26 | New for Robiga - VFGP NiFi |
| SS-PROD-from-3420 1-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-OUT-NLB | OUTBOUND | 10.163.229.20 | New for Robiga - LDR |
| SS-PROD-from-3430 1-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-OUT-NLB | OUTBOUND | 10.151.4.84 | New for Robiga - MIS |
| SS-PROD-from-3440 1-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-OUT-NLB | OUTBOUND | 10.162.123.158 | New for Robiga - LEA |
| SS-PROD-from-3450 1-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-OUT-NLB | OUTBOUND | 198.18.74.210 | New for Robiga - RAID8 |
| SS-PROD-from-2500 1-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PROD-SFTP-OUT-NLB | OUTBOUND | 198.18.74.201 198.18.74.220 198.18.74.221 198.18.74.222 198.18.74.223 198.18.74.224 198.18.74.225 198.18.74.226 | New for Robiga - MZ |

| | | | | | | | | |
|---|------|-----|----|-----|---|--------------|--|---|
| SS-PRO D-from- 3470 3-to- 5658 | 3470 | TCP | IP | N/A | VF- IEDELIVE RY-SS- PROD- API-OUT- NLB | OUTBOU ND | 10.163.72.59(dra68 01) 10.163.72.60(dra68 02) 10.163.72.192(dra7 061) 10.163.72.193(dra7 062) | New for ER-IF - CCS Direct |
| SS-PRO D-from- 3470 1-to- 3470 1 | 3470 | TCP | IP | N/A | VF- IEDELIVE RY-SS- PROD-DB- NLB | INBOUND | N/A | New for ER-IF - Oracle RDS access |
| SS-PRO D-from- 3470 4-to- 443 | 443 | TCP | IP | N/A | VF- IEDELIVE RY-SS- PROD- API-NLB | INBOUND | N/A | New For ER-IF - ErCore- ER-IF |
| SS-PRO D-from- 3700 0-to- 443 | 443 | TCP | IP | N/A | VFIE- PROD- ONPREM- NLB | INBOUND | N/A | New For OneAPP (CIAM) To IDHUB |
| SS-PRO D-from 3800 1 to 443 | 443 | TCP | IP | | VF- IEDELIVE RY-SS- PROD- NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRO D-from 3800 2 to 443 | 443 | | | | VF- IEDELIVE RY-SS- PROD- APII-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRO D-from 3800 3 to 1433 | 1433 | | | | VF- IEDELIVE RY-SS- PROD-DB- NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRO D- | 22 | | | | VF- IEDELIVE RY-SS- | INBOUND | | ITR - 6946 GNP & GNM |

| | | | | | | | | |
|--------------------------------------|------|-----|----|--|--------------------------------|----------|--|-------------------------------------|
| from 3800 4 to 22 | | | | | PROD-SFTP-NLB | | | Number porting |
| SS-PROD-from 3800 5 to 1434 | 1434 | | | | VF-IEDELIVERY-SS-PROD-DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PROD-from 3810 1 to 443 | 443 | TCP | IP | | VF-IEDELIVERY-SS-PROD-NLB | OUTBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PROD-from 3810 2 to 443 | 443 | | | | VF-IEDELIVERY-SS-PROD-APII-NLB | OUTBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PROD-from 3810 3 to 1433 | 1433 | | | | VF-IEDELIVERY-SS-PROD-DB-NLB | OUTBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PROD-from 3810 4 to 22 | 22 | | | | VF-IEDELIVERY-SS-PROD-SFTP-NLB | OUTBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PROD-from 3810 5 to 1434 | 1434 | | | | VF-IEDELIVERY-SS-PROD-DB-NLB | OUTBOUND | | ITR - 6946 GNP & GNM Number porting |

6.6.4.3 PRD1

6.6.4.3.1 Summary

| Destination NLB | Count |
|---------------------------|-------|
| VF-IEDELIVERY-SS-PRD1-NLB | 5 |

| | |
|------------------------------------|----|
| VF-IEDELIVERY-SS-PRD1-API-NLB | 3 |
| VF-IEDELIVERY-SS-PRD1-SFTP-NLB | 3 |
| VF-IEDELIVERY-SS-PRD1-DB-NLB | 4 |
| VF-IEDELIVERY-SS-PRD1-API-OUT-NLB | 11 |
| VF-IEDELIVERY-SS-PRD1-SFTP-OUT-NLB | 2 |
| VF-IEDELIVERY-SS-PRD1-DB-OUT-NLB | 4 |
| N/A | 2 |

6.6.4.3.2 Target Groups

| Name | Port | Protocol | Target type | Load balancer | New Load balancer | Direction | Comment | |
|---|------|----------|-------------|---------------------------|-----------------------------------|-----------|-----------------------------------|--|
| <u>SS-PRD1-from-10101-to-7443</u> | 7443 | TCP | IP | VF-IEDELIVERY-SS-PRD1-NLB | N/A | OUTBOUND | Can be deleted - Old SAL instance | |
| <u>SS-PRD1-from-10102-to-9080</u> | 9080 | TCP | IP | VF-IEDELIVERY-SS-PRD1-NLB | N/A | OUTBOUND | Can be deleted - Old SAL instance | |
| <u>SS-PRD1-from-30201-to-9443</u> | 9443 | TCP | IP | VF-IEDELIVERY-SS-PRD1-NLB | VF-IEDELIVERY-SS-PRD1-API-NLB | INBOUND | | |
| <u>SS-PRD1-from-8011-to-8011</u> | 8011 | TCP | IP | VF-IEDELIVERY-SS-PRD1-NLB | VF-IEDELIVERY-SS-PRD1-API-NLB | INBOUND | | |
| <u>SS-PRD1-from-11001-to-7004</u> | 7004 | TCP | IP | VF-IEDELIVERY-SS-PRD1-NLB | VF-IEDELIVERY-SS-PRD1-API-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD1-from-11002-to-7004</u> | 7004 | TCP | IP | VF-IEDELIVERY-SS-PRD1-NLB | VF-IEDELIVERY-SS-PRD1-API-OUT-NLB | OUTBOUND | | |

| | | | | | | | | |
|---|-----------|-----|----|---------------------------------------|---|--------------|--|--|
| <u>SS- PRD1 -from- 11003 -to- 7004</u> | 7004 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 12103 -to- 16443</u> | 1644 3 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 13001 -to- 8005</u> | 8005 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 14001 -to- 443</u> | 443 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 15101 -to- 32007</u> | 3200 7 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 20101 -to- 6543</u> | 6543 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 20111 -to- 1040</u> | 1040 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 20112 -to- 1050</u> | 1050 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 10304</u> | 1521 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- DB-NLB | INBOUND | | |

| | | | | | | | | |
|---|-----------|-----|----|---------------------------------------|--|--------------|--|--|
| <u>-to- 1521</u> | | | | | | | | |
| <u>SS- PRD1 -from- 10305 -to- 2484</u> | 2484 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- DB-NLB | INBOUND | | |
| <u>SS- PRD1 -from- 22102 -to- 33001</u> | 3300 1 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- DB-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 22103 -to- 33001</u> | 3300 1 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- DB-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 22104 -to- 33001</u> | 3300 1 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- DB-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 22105 -to- 33001</u> | 3300 1 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- DB-OUT- NLB | OUTBOUN D | | |
| <u>SS- PRD1 -from- 10011 -to- 10011</u> | 1001 1 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- NLB | INBOUND | | |
| <u>SS- PRD1 -from- 30001 -to- 443</u> | 443 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- NLB | INBOUND | | |
| <u>SS- PRD1 -from- 7001- to- 7001</u> | 7001 | TCP | IP | VF- IEDELIVER Y-SS-PRD1- NLB | VF- IEDELIVER Y-SS-PRD1- NLB | INBOUND | | |
| <u>SS- PRD1 -from-</u> | 8443 | TCP | IP | VF- IEDELIVER | VF- IEDELIVER | INBOUND | | |

| | | | | | | | | |
|----------------------------------|-------|-----|----|----------------------------|-------------------------------------|----------|---------------|--|
| <u>8443-to-8443</u> | | | | Y-SS-PRD1-NLB | Y-SS-PRD1-NLB | | | |
| <u>SS-PRD1-from-9001-to-9001</u> | 9001 | TCP | IP | VF-IEDELIVER Y-SS-PRD1-NLB | VF-IEDELIVER Y-SS-PRD1-NLB | INBOUND | | |
| <u>SS-PRD1-from-10120-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-PRD1-NLB | VF-IEDELIVER Y-SS-PRD1-SFTP-NLB | INBOUND | | |
| <u>SS-PRD1-from-30101-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-PRD1-NLB | VF-IEDELIVER Y-SS-PRD1-SFTP-NLB | INBOUND | | |
| <u>SS-PRD1-from-27001-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-PRD1-NLB | VF-IEDELIVER Y-SS-PRD1-SFTP-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD1-from-28001-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-PRD1-NLB | VF-IEDELIVER Y-SS-PRD1-SFTP-OUT-NLB | OUTBOUND | | |
| SS-PRD1-from-35001-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVER Y-SS-PRD1-SFTP-NLB | INBOUND | New for DPC | |
| SS-PRD1-from-35002-to-35002 | 35002 | TCP | IP | N/A | VF-IEDELIVER Y-SS-PRD1-DB-NLB | INBOUND | New for DPC | |
| SS-PRD1-from-36001-to-443 | 36001 | TCP | IP | N/A | VF-IEDELIVER Y-SS-PRD1-API-NLB | INBOUND | New for eSign | |
| SS-PRD1-from-36002-to-36002 | 35002 | TCP | IP | N/A | VF-IEDELIVER Y-SS-PRD1-DB-NLB | INBOUND | New for eSign | |

| | | | | | | | | |
|---|------|-----|----|--|--------------------------------|---------|--|-------------------------------------|
| SS-PRD1 -from 38001 to 443 | 443 | TCP | IP | | VF-IEDELIVERY-SS-PRD1-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRD1 -from 38002 to 443 | 443 | | | | VF-IEDELIVERY-SS-PRD1-APII-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRD1 -from 38003 to 1433 | 1433 | | | | VF-IEDELIVERY-SS-PRD1-DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRD1 -from 38004 to 22 | 22 | | | | VF-IEDELIVERY-SS-PRD1-SFTP-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRD1 -from 38005 to 1434 | 1434 | | | | VF-IEDELIVERY-SS-PRD1-DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| | | | | | | | | |
| | | | | | | | | |

6.6.4.4 PRD2

6.6.4.4.1 Summary

| Destination NLB | Count |
|------------------------------------|-------|
| VF-IEDELIVERY-SS-PRD2-NLB | 6 |
| VF-IEDELIVERY-SS-PRD2-API-NLB | 2 |
| VF-IEDELIVERY-SS-PRD2-SFTP-NLB | 3 |
| VF-IEDELIVERY-SS-PRD2-DB-NLB | 3 |
| VF-IEDELIVERY-SS-PRD2-API-OUT-NLB | 3 |
| VF-IEDELIVERY-SS-PRD2-SFTP-OUT-NLB | 1 |
| VF-IEDELIVERY-SS-PRD2-DB-OUT-NLB | 4 |
| N/A | 0 |

6.6.4.4.2 Target Groups

| Name | Port | Protocol | Target type | Load balancer | New Load balancer | Direction | Comment | |
|--|-------|----------|-------------|----------------------------|------------------------------------|-----------|---------|--|
| <u>SS-PRD2-from-30201-to-9443</u> | 9443 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-API-NLB | INBOUND | | |
| <u>SS-PRD2-from-8011-to-8011</u> | 8011 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-API-NLB | INBOUND | | |
| <u>SS-PRD2-from-12103-to-16443</u> | 16443 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-API-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD2-from-13001-to-8005</u> | 8005 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-API-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD2-from-15101-to-33007</u> | 33007 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-API-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD2-from-10304-to-1521</u> | 1521 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-DB-NLB | INBOUND | | |
| <u>SS-PRD2-from-10305-to-2484</u> | 2484 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-DB-NLB | INBOUND | | |
| <u>SS-PRD2-from-22102-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-PRD2-NLB | VF-IEDELIVER Y-SS-PRD2-DB-OUT-NLB | OUTBOUND | | |

| | | | | | | | | |
|--|-------|-----|----|---------------------------|----------------------------------|----------|--|--|
| <u>SS-PRD2-from-22103-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD2-from-22104-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD2-from-22105-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-PRD2-from-10011-to-10011</u> | 10011 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-NLB | INBOUND | | |
| <u>SS-PRD2-from-14001-to-443</u> | 443 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-NLB | INBOUND | | |
| <u>SS-PRD2-from-30001-to-443</u> | 443 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-NLB | INBOUND | | |
| <u>SS-PRD2-from-7001-to-7001</u> | 7001 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-NLB | INBOUND | | |
| <u>SS-PRD2-from-8443-to-8443</u> | 8443 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-NLB | INBOUND | | |
| <u>SS-PRD2-from-9001-</u> | 9001 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-NLB | INBOUND | | |

| | | | | | | | | |
|---------------------------------|-------|-----|----|---------------------------|------------------------------------|----------|-------------|-------------------------------------|
| <u>to-9001</u> | | | | | | | | |
| <u>SS-PRD2-from-10120-to-22</u> | 22 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-SFTP-NLB | INBOUND | | |
| <u>SS-PRD2-from-30101-to-22</u> | 22 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-SFTP-NLB | INBOUND | | |
| <u>SS-PRD2-from-27001-to-22</u> | 22 | TCP | IP | VF-IEDELIVERY-SS-PRD2-NLB | VF-IEDELIVERY-SS-PRD2-SFTP-OUT-NLB | OUTBOUND | | |
| SS-PRD2-from-35001-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVERY-SS-PRD2-SFTP-NLB | INBOUND | New for DPC | |
| SS-PRD2-from-35002-to-35002 | 35002 | TCP | IP | N/A | VF-IEDELIVERY-SS-PRD2-DB-NLB | INBOUND | New for DPC | |
| SS-PRD2-from 38001 to 443 | 443 | TCP | IP | | VF-IEDELIVERY-SS-PRD2-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRD2-from 38002 to 443 | 443 | | | | VF-IEDELIVERY-SS-PRD2-APII-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRD2-from 38003 to 1433 | 1433 | | | | VF-IEDELIVERY-SS-PRD2-DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-PRD2-from 38004 to 22 | 22 | | | | VF-IEDELIVERY-SS-PRD2-SFTP-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |

| | | | | | | | | |
|----------------------------|------|--|--|--|------------------------------|---------|--|-------------------------------------|
| SS-PRD2-from 38005 to 1434 | 1434 | | | | VF-IEDELIVERY-SS-PRD2-DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
|----------------------------|------|--|--|--|------------------------------|---------|--|-------------------------------------|

6.6.4.5 SIT1

6.6.4.5.1 Summary

| Destination NLB | Count |
|------------------------------------|-------|
| VF-IEDELIVERY-SS-SIT1-NLB | 4 |
| VF-IEDELIVERY-SS-SIT1-API-NLB | 6 |
| VF-IEDELIVERY-SS-SIT1-SFTP-NLB | 3 |
| VF-IEDELIVERY-SS-SIT1-DB-NLB | 3 |
| VF-IEDELIVERY-SS-SIT1-API-OUT-NLB | 9 |
| VF-IEDELIVERY-SS-SIT1-SFTP-OUT-NLB | 0 |
| VF-IEDELIVERY-SS-SIT1-DB-OUT-NLB | 4 |
| N/A | 0 |

6.6.4.5.2 Target Groups

| Name | Port | Protocol | Target type | Load balancer | New Load balancer | Direction | Comment | |
|---|------|----------|-------------|---------------------------|-------------------------------|-----------|---------|--|
| <u>SS-SIT1-from-30201-to-9443</u> | 9443 | TCP | IP | VF-IEDELIVERY-SS-SIT1-NLB | VF-IEDELIVERY-SS-SIT1-API-NLB | INBOUND | | |
| <u>SS-SIT1-from-587-to-587</u> | 587 | TCP | IP | VF-IEDELIVERY-SS-SIT1-NLB | VF-IEDELIVERY-SS-SIT1-API-NLB | INBOUND | SES | |
| <u>SS-SIT1-from-588-to-443</u> | 588 | TCP | IP | VF-IEDELIVERY-SS-SIT1-NLB | VF-IEDELIVERY-SS-SIT1-API-NLB | INBOUND | SQS | |
| <u>SS-SIT1-from-8011-to-8011</u> | 8011 | TCP | IP | VF-IEDELIVERY-SS-SIT1-NLB | VF-IEDELIVERY-SS-SIT1-API-NLB | INBOUND | | |
| <u>SS-SIT1-from-</u> | 7443 | TCP | IP | VF-IEDELIVERY | VF-IEDELIVERY-SS-SIT1- | OUTBOUND | | |

| | | | | | | | | |
|--|------|-----|----|---------------------------------------|---|--------------|--|--|
| <u>10101 -to- 7443</u> | | | | Y-SS-SIT1- NLB | API-OUT- NLB | | | |
| <u>SS- SIT1- from- 10102 -to- 9080</u> | 9080 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1- from- 13001 -to- 8005</u> | 8005 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1- from- 14001 -to- 443</u> | 443 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1- from- 14002 -to- 443</u> | 443 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1- from- 15101 -to- 7007</u> | 7007 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1- from- 20101 -to- 6543</u> | 6543 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1- from- 20111 -to- 1040</u> | 1040 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1- from- 20112 -to- 1050</u> | 1050 | TCP | IP | VF- IEDELIVER Y-SS-SIT1- NLB | VF- IEDELIVER Y-SS-SIT1- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT1-</u> | 1521 | TCP | IP | VF- IEDELIVER | VF- IEDELIVER | INBOUND | | |

| | | | | | | | | |
|--|-------|-----|----|----------------------------|-----------------------------------|----------|--|--|
| <u>from-10304-to-1521</u> | | | | Y-SS-SIT1-NLB | Y-SS-SIT1-DB-NLB | | | |
| <u>SS-SIT1-from-10305-to-2484</u> | 2484 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-DB-NLB | INBOUND | | |
| <u>SS-SIT1-from-22102-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT1-from-22103-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT1-from-22104-to-33015</u> | 33015 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT1-from-22105-to-33015</u> | 33015 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT1-from-30001-to-443</u> | 443 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-NLB | INBOUND | | |
| <u>SS-SIT1-from-7001-to-7001</u> | 7001 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-NLB | INBOUND | | |
| <u>SS-SIT1-from-8443-to-8443</u> | 8443 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-NLB | INBOUND | | |

| | | | | | | | | |
|---|------|-----|----|----------------------------|---------------------------------|---------|--------------------|-------------------------------------|
| <u>SS-SIT1-from-10120-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-SFTP-NLB | INBOUND | | |
| <u>SS-SIT1-from-30101-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT1-NLB | VF-IEDELIVER Y-SS-SIT1-SFTP-NLB | INBOUND | | |
| SS-SIT1-from-35001-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT1-SFTP-NLB | INBOUND | New for DPC | |
| SS-SIT1-from-35002-to-35002 | 3500 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT1-DB-NLB | INBOUND | New for DPC | |
| SS-SIT1-from-587-to-587 | 587 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT1-API-NLB | INBOUND | SES - New for Pega | |
| SS-SIT1-from-588-to-443 | 588 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT1-API-NLB | INBOUND | SQS - New for Pega | |
| SS-SIT1-from 38001 to 443 | 443 | TCP | IP | | VF-IEDELIVER Y-SS-SIT1-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT1-from 38002 to 443 | 443 | | | | VF-IEDELIVER Y-SS-SIT1-APII-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT1-from 38003 to 1433 | 1433 | | | | VF-IEDELIVER Y-SS-SIT1-DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT1-from | 22 | | | | VF-IEDELIVER | INBOUND | | ITR - 6946 GNP & GNM |

| | | | | | | | | |
|---|------|--|--|--|--|---------|--|---|
| 38004 to 22 | | | | | Y-SS-SIT1- SFTP-NLB | | | Number porting |
| SS- SIT1- from 38005 to 1434 | 1434 | | | | VF- IEDELIVER Y-SS-SIT1- DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |

6.6.4.6 SIT2

6.6.4.6.1 Summary

| Destination NLB | Count |
|------------------------------------|-------|
| VF-IEDELIVERY-SS-SIT2-NLB | 3 |
| VF-IEDELIVERY-SS-SIT2-API-NLB | 3 |
| VF-IEDELIVERY-SS-SIT2-SFTP-NLB | 3 |
| VF-IEDELIVERY-SS-SIT2-DB-NLB | 4 |
| VF-IEDELIVERY-SS-SIT2-API-OUT-NLB | 12 |
| VF-IEDELIVERY-SS-SIT2-SFTP-OUT-NLB | 0 |
| VF-IEDELIVERY-SS-SIT2-DB-OUT-NLB | 4 |
| N/A | 0 |

6.6.4.6.2 Target Groups

| Name | Port | Protocol | Target type | Load balancer | New Load balancer | Direction | Comment | |
|--|------|----------|-------------|---------------------------------------|---|-----------|---------|--|
| <u>SS- SIT2- from- 30201 -to- 9443</u> | 9443 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-NLB | INBOUND | | |
| <u>SS- SIT2- from- 8011- to- 8011</u> | 8011 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-NLB | INBOUND | | |
| <u>SS- SIT2- from- 10101 -to- 7443</u> | 7443 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUND | | |
| <u>SS- SIT2- from- 10102</u> | 9443 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- NLB | OUTBOUND | | |

| | | | | | | | | |
|---|-----------|-----|----|---------------------------------------|---|--------------|--|--|
| <u>-to- 9443</u> | | | | | API-OUT- NLB | | | |
| <u>SS- SIT2- from- 11001 -to- 7004</u> | 7004 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from- 11002 -to- 7004</u> | 7004 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from- 11003 -to- 7004</u> | 7004 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from- 13001 -to- 8005</u> | 8005 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from- 14001 -to- 443</u> | 443 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from- 14002 -to- 443</u> | 443 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from- 15101 -to- 11007</u> | 1100 7 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from- 20101 -to- 6543</u> | 6543 | TCP | IP | VF- IEDELIVER Y-SS-SIT2- NLB | VF- IEDELIVER Y-SS-SIT2- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT2- from-</u> | 1040 | TCP | IP | VF- IEDELIVER | VF- IEDELIVER Y-SS-SIT2- | OUTBOUN D | | |

| | | | | | | | | |
|--|-------|-----|----|----------------------------|------------------------------------|----------|--|--|
| <u>20111-to-1040</u> | | | | Y-SS-SIT2-NLB | API-OUT-NLB | | | |
| <u>SS-SIT2-from-20112-to-1050</u> | 1050 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT2-from-10304-to-1521</u> | 1521 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-DB-NLB | INBOUND | | |
| <u>SS-SIT2-from-10305-to-2484</u> | 2484 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-DB-NLB | INBOUND | | |
| <u>SS-SIT2-from-22102-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT2-from-22103-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT2-from-22104-to-33016</u> | 33016 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT2-from-22105-to-33016</u> | 33016 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT2-from-30001-to-443</u> | 443 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-NLB | INBOUND | | |
| <u>SS-SIT2-</u> | 7001 | TCP | IP | VF-IEDELIVER | VF-IEDELIVER | INBOUND | | |

| | | | | | | | | |
|----------------------------------|-------|-----|----|----------------------------|---------------------------------|---------|---------------|-------------------------------------|
| <u>from-7001-to-7001</u> | | | | Y-SS-SIT2-NLB | Y-SS-SIT2-NLB | | | |
| <u>SS-SIT2-from-8443-to-8443</u> | 8443 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-NLB | INBOUND | | |
| <u>SS-SIT2-from-10120-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-SFTP-NLB | INBOUND | | |
| <u>SS-SIT2-from-30101-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT2-NLB | VF-IEDELIVER Y-SS-SIT2-SFTP-NLB | INBOUND | | |
| SS-SIT2-from-35001-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT2-SFTP-NLB | INBOUND | New for DPC | |
| SS-SIT2-from-35002-to-35002 | 35002 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT2-DB-NLB | INBOUND | New for DPC | |
| SS-SIT2-from-36001-to-443 | 36001 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT2-API-NLB | INBOUND | New for eSign | |
| SS-SIT2-from-36002-to-36002 | 35002 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT2-DB-NLB | INBOUND | New for eSign | |
| SS-SIT2-from 38001 to 443 | 443 | TCP | IP | | VF-IEDELIVER Y-SS-SIT2-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT2-from 38002 | 443 | | | | VF-IEDELIVER | INBOUND | | ITR - 6946 GNP & GNM |

| | | | | | | | |
|---|------|--|--|--|---------|--|---|
| to 443 | | | | Y-SS-SIT2- APII-NLB | | | Number porting |
| SS- SIT2- from 38003 to 1433 | 1433 | | | VF- IEDELIVER Y-SS-SIT2- DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS- SIT2- from 38004 to 22 | 22 | | | VF- IEDELIVER Y-SS-SIT2- SFTP-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS- SIT2- from 38005 to 1434 | 1434 | | | VF- IEDELIVER Y-SS-SIT2- DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |

6.6.4.7 SIT3

6.6.4.7.1 Summary

| Destination NLB | Count |
|------------------------------------|-------|
| VF-IEDELIVERY-SS-SIT3-NLB | 5 |
| VF-IEDELIVERY-SS-SIT3-API-NLB | 5 |
| VF-IEDELIVERY-SS-SIT3-SFTP-NLB | 5 |
| VF-IEDELIVERY-SS-SIT3-DB-NLB | 9 |
| VF-IEDELIVERY-SS-SIT3-API-OUT-NLB | 17 |
| VF-IEDELIVERY-SS-SIT3-SFTP-OUT-NLB | 7 |
| VF-IEDELIVERY-SS-SIT3-DB-OUT-NLB | 4 |
| N/A | 0 |

6.6.4.7.2 Target Groups

| Name | Port | Protocol | Target type | Load balancer | New Load balancer | Direction | Destination IP's | Comment |
|--|------|----------|-------------|---------------------------------------|---|-----------|------------------|---------|
| <u>SS- SIT3- from- 30201 -to- 9443</u> | 9443 | TCP | IP | VF- IEDELIVER Y-SS-SIT3- NLB | VF- IEDELIVER Y-SS-SIT3- API-NLB | INBOUND | N/A | |
| <u>SS- SIT3- from- 8011-</u> | 8011 | TCP | IP | VF- IEDELIVER Y-SS-SIT3- NLB | VF- IEDELIVER Y-SS-SIT3- API-NLB | INBOUND | N/A | |

| | | | | | | | | |
|---|------|-----|----|----------------------------|------------------------------------|----------|--|--|
| <u>to-8011</u> | | | | | | | | |
| <u>SS-SIT3-from-10101-to-7443</u> | 7443 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-10102-to-9443</u> | 9443 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-11001-to-7004</u> | 7004 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-11002-to-7004</u> | 7004 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-11003-to-7004</u> | 7004 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-13001-to-8005</u> | 8005 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-14001-to-443</u> | 443 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-14002-to-443</u> | 443 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-</u> | 9007 | TCP | IP | VF-IEDELIVER | VF-IEDELIVER Y-SS-SIT3- | OUTBOUND | | |

| | | | | | | | | |
|---|-------|-----|----|----------------------------|------------------------------------|----------|-----|--|
| <u>15101 -to- 9007</u> | | | | Y-SS-SIT3-NLB | API-OUT-NLB | | | |
| <u>SS-SIT3-from-20101 -to-6543</u> | 6543 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-20111 -to-1040</u> | 1040 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-20112 -to-1050</u> | 1050 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-443-to-443</u> | 443 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-10304 -to-1521</u> | 1521 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | |
| <u>SS-SIT3-from-10305 -to-2484</u> | 2484 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | |
| <u>SS-SIT3-from-22102 -to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-22103 -to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT3-from-</u> | 33017 | TCP | IP | VF-IEDELIVER | VF-IEDELIVER Y-SS-SIT3- | OUTBOUND | | |

| | | | | | | | | | |
|---|-------|-----|----|-------------------------------|-------------------------------------|----------|-----|-------------|--|
| <u>22104 -to- 33017</u> | | | | Y-SS-SIT3-NLB | DB-OUT-NLB | | | | |
| <u>SS-SIT3-from- 22105 -to- 33017</u> | 33017 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-DB-OUT-NLB | OUTBOUND | | | |
| <u>SS-SIT3-from- 30001 -to- 443</u> | 443 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-NLB | INBOUND | N/A | | |
| <u>SS-SIT3-from- 7001-to- 7001</u> | 7001 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-NLB | INBOUND | N/A | | |
| <u>SS-SIT3-from- 8443-to- 8443</u> | 8443 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-NLB | INBOUND | N/A | | |
| <u>SS-SIT3-from- 10120-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-SFTP-NLB | INBOUND | N/A | | |
| <u>SS-SIT3-from- 30101-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-NLB | VF-IEDELIVER Y-SS-SIT3-SFTP-NLB | INBOUND | N/A | | |
| <u>SS-SIT3-from- 34501-to- 22-SFTP</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT3-VL-NLB | VF-IEDELIVER Y-SS-SIT3-SFTP-OUT-NLB | OUTBOUND | | | |
| SS-SIT3-from- 35001-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-SFTP-NLB | INBOUND | N/A | New for DPC | |
| SS-SIT3-from- 35002 | 35002 | TCP | IP | N/A | VF-IEDELIVER | INBOUND | N/A | New for DPC | |

| | | | | | | | | |
|-----------------------------|-------|-----|----|-----|---------------------------------|---------|-----|---------------------------------|
| -to-35002 | | | | | Y-SS-SIT3-DB-NLB | | | |
| SS-SIT3-from-32001-to-443 | 32001 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-API-NLB | INBOUND | N/A | New for BSP - GUI |
| SS-SIT3-from-32002-to-22 | 32002 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-SFTP-NLB | INBOUND | N/A | New for BSP - SFTP |
| SS-SIT3-from-32003-to-443 | 32003 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-API-NLB | INBOUND | N/A | New for BSP - API |
| SS-SIT3-from-32004-to-32004 | 32004 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | New for BSP - Postgres Access |
| SS-SIT3-from-32005-to-32005 | 32005 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | New for BSP - MSSQL Access |
| SS-SIT3-from-32006-to-32006 | 32006 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | New for BSP - DocumentDB Access |
| SS-SIT3-from-34701-to-443 | 443 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | New for BSP - EVO |
| SS-SIT3-from-34801-to-9499 | 9499 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | New for BSP - Poller |
| SS-SIT3-from-36001 | 443 | TCP | IP | N/A | VF-IEDELIVER | INBOUND | N/A | New for eSign |

| | | | | | | | | |
|-----------------------------|-------|-----|----|----------------------|---------------------------------|---------|-----|-------------------------------------|
| -to-443 | | | | | Y-SS-SIT3-API-NLB | | | |
| SS-SIT3-from-36002-to-36002 | 36002 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-DB-NLB | INBOUND | N/A | New for eSign |
| SS-SIT3-from-34001-to-443 | 443 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-API-NLB | INBOUND | N/A | New for Robiga |
| SS-SIT3-from-34002-to-443 | 443 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-API-NLB | INBOUND | N/A | New for Robiga |
| SS-SIT3-from-34022-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT3-SFTP-NLB | INBOUND | N/A | New for Robiga |
| SS-SIT3-from-38000-to-443 | 443 | TCP | IP | VFIE-SIT3-ONPREM-NLB | VFIE-SIT3-ONPREM-NLB | INBOUND | N/A | New for OneApp (CIAM) to IDHUB |
| SS-SIT3-from 38001 to 443 | 443 | TCP | IP | | VF-IEDELIVER Y-SS-SIT3-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT3-from 38002 to 443 | 443 | | | | VF-IEDELIVER Y-SS-SIT3-APII-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT3-from 38003 to 1433 | 1433 | | | | VF-IEDELIVER Y-SS-SIT3-DBI-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT3-from | 22 | | | | VF-IEDELIVER | INBOUND | | ITR - 6946 GNP & GNM |

| | | | | | | | | |
|---|------|--|--|--|--|---------|--|---|
| 38004 to 22 | | | | | Y-SS-SIT3- SFTP-NLB | | | Number porting |
| SS- SIT3- from 38005 to 1434 | 1434 | | | | VF- IEDELIVER Y-SS-SIT3- DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |

6.6.4.8 SIT4

6.6.4.8.1 Summary

| Destination NLB | Count |
|------------------------------------|-------|
| VF-IEDELIVERY-SS-SIT4-NLB | 3 |
| VF-IEDELIVERY-SS-SIT4-API-NLB | 2 |
| VF-IEDELIVERY-SS-SIT4-SFTP-NLB | 3 |
| VF-IEDELIVERY-SS-SIT4-DB-NLB | 3 |
| VF-IEDELIVERY-SS-SIT4-API-OUT-NLB | 11 |
| VF-IEDELIVERY-SS-SIT4-SFTP-OUT-NLB | 0 |
| VF-IEDELIVERY-SS-SIT4-DB-OUT-NLB | 4 |
| N/A | 1 |

6.6.4.8.2 Target Groups

| Name | Port | Protocol | Target type | Load balancer | New Load balancer | Direction | Comment | |
|---|-------|----------|-------------|---------------------------------------|---|-----------|---------|--|
| SS- SIT4- 10101 - M3Xb - TEMP | 10101 | TCP | Instance | None associated | N/A | N/A | Delete | |
| SS- SIT4- from- 30201 -to- 9443 | 9443 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-NLB | INBOUND | | |
| SS- SIT4- from- 8011- to- 8011 | 8011 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-NLB | INBOUND | | |
| SS- SIT4- from- 10101 | 7443 | TCP | IP | VF- IEDELIVER | VF- IEDELIVER Y-SS-SIT4- | OUTBOUND | | |

| | | | | | | | | |
|---|-----------|-----|----|---------------------------------------|---|--------------|--|--|
| <u>-to- 7443</u> | | | | Y-SS-SIT4- NLB | API-OUT- NLB | | | |
| <u>SS- SIT4- from- 10102 -to- 9080</u> | 9080 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from- 11001 -to- 7004</u> | 7004 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from- 11002 -to- 7004</u> | 7004 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from- 11003 -to- 7004</u> | 7004 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from- 12103 -to- 16443</u> | 1644 3 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from- 13001 -to- 8005</u> | 8005 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from- 15101 -to- 14010</u> | 1401 0 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from- 20101 -to- 6543</u> | 6543 | TCP | IP | VF- IEDELIVER Y-SS-SIT4- NLB | VF- IEDELIVER Y-SS-SIT4- API-OUT- NLB | OUTBOUN D | | |
| <u>SS- SIT4- from-</u> | 1040 | TCP | IP | VF- IEDELIVER | VF- IEDELIVER Y-SS-SIT4- | OUTBOUN D | | |

| | | | | | | | | |
|--|-------|-----|----|----------------------------|------------------------------------|----------|--|--|
| <u>20111-to-1040</u> | | | | Y-SS-SIT4-NLB | API-OUT-NLB | | | |
| <u>SS-SIT4-from-20112-to-1050</u> | 1050 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-API-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT4-from-10304-to-1521</u> | 1521 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-DB-NLB | INBOUND | | |
| <u>SS-SIT4-from-10305-to-2484</u> | 2484 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-DB-NLB | INBOUND | | |
| <u>SS-SIT4-from-22102-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT4-from-22103-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT4-from-22104-to-33016</u> | 33016 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT4-from-22105-to-33016</u> | 33016 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-DB-OUT-NLB | OUTBOUND | | |
| <u>SS-SIT4-from-30001-to-443</u> | 443 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-NLB | INBOUND | | |
| <u>SS-SIT4-</u> | 7001 | TCP | IP | VF-IEDELIVER | VF-IEDELIVER | INBOUND | | |

| | | | | | | | | |
|--|------|-----|----|----------------------------|---------------------------------|---------|-------------|-------------------------------------|
| <u>from-7001-to-7001</u> | | | | Y-SS-SIT4-NLB | Y-SS-SIT4-NLB | | | |
| <u>SS-SIT4-from-8443-to-8443</u> | 8443 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-NLB | INBOUND | | |
| <u>SS-SIT4-from-10120-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-SFTP-NLB | INBOUND | | |
| <u>SS-SIT4-from-30101-to-22</u> | 22 | TCP | IP | VF-IEDELIVER Y-SS-SIT4-NLB | VF-IEDELIVER Y-SS-SIT4-SFTP-NLB | INBOUND | | |
| SS-SIT4-from-35001-to-22 | 22 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT4-SFTP-NLB | INBOUND | New for DPC | |
| SS-SIT4-2 from-35002-to-35002 | 3500 | TCP | IP | N/A | VF-IEDELIVER Y-SS-SIT4-DB-NLB | INBOUND | New for DPC | |
| SS-SIT4-from 38001 to 443 | 443 | TCP | IP | | VF-IEDELIVER Y-SS-SIT4-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT4-from 38002 to 443 | 443 | | | | VF-IEDELIVER Y-SS-SIT4-APII-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT4-from 38003 to 1433 | 1433 | | | | VF-IEDELIVER Y-SS-SIT4-DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |
| SS-SIT4-from | 22 | | | | VF-IEDELIVER | INBOUND | | ITR - 6946 GNP & GNM |

| | | | | | | | | |
|---|------|--|--|--|--|---------|--|---|
| 38004 to 22 | | | | | Y-SS-SIT4- SFTP-NLB | | | Number porting |
| SS- SIT4- from 38005 to 1434 | 1434 | | | | VF- IEDELIVER Y-SS-SIT4- DB-NLB | INBOUND | | ITR - 6946 GNP & GNM Number porting |

6.6.4.9 DEV1

6.6.4.9.1 Summary

| Destination NLB | Count |
|------------------------------------|-------|
| VF-IEDELIVERY-SS-DEV1-NLB | 2 |
| VF-IEDELIVERY-SS-DEV1-API-NLB | 1 |
| VF-IEDELIVERY-SS-DEV1-SFTP-NLB | 1 |
| VF-IEDELIVERY-SS-DEV1-DB-NLB | 3 |
| VF-IEDELIVERY-SS-DEV1-API-OUT-NLB | 9 |
| VF-IEDELIVERY-SS-DEV1-SFTP-OUT-NLB | 0 |
| VF-IEDELIVERY-SS-DEV1-DB-OUT-NLB | 2 |
| N/A | 0 |

6.6.4.9.2 Target Groups

| Name | Port | Protocol | Target type | Load balancer | New Load balancer | Direction | Comment |
|---|-------|----------|-------------|-----------------------------------|---|-----------|---------|
| <u>SS- DEV1- from- 48012- to- 48012</u> | 48012 | TCP | IP | VF- IEDELIVERY- SS-DEV1-NLB | VF- IEDELIVERY- SS-DEV1-API- NLB VF- IEDELIVERY- SS-DEV1-APII- NLB | INBOUND | |
| <u>SS- DEV1- from- 10101- to-7443</u> | 7443 | TCP | IP | VF- IEDELIVERY- SS-DEV1-NLB | VF- IEDELIVERY- SS-DEV1-API- OUT-NLB VF- IEDELIVERY- SS-DEV1- APIO-NLB | OUTBOUND | |
| <u>SS- DEV1- from- 10102- to-9443</u> | 9443 | TCP | IP | VF- IEDELIVERY- SS-DEV1-NLB | VF- IEDELIVERY- SS-DEV1-API- OUT-NLB VF- IEDELIVERY- | OUTBOUND | |

| | | | | | | | |
|--|-------|-----|----|---------------------------|---|----------|--|
| | | | | | SS-DEV1-APIO-NLB | | |
| <u>SS-DEV1-from-13001-to-8005</u> | 8005 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-API-OUT-NLB VF-IEDELIVERY-SS-DEV1-APIO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-15101-to-9007</u> | 9007 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-API-OUT-NLB VF-IEDELIVERY-SS-DEV1-APIO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-20101-to-6543</u> | 6543 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-API-OUT-NLB VF-IEDELIVERY-SS-DEV1-APIO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-20111-to-1040</u> | 1040 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-API-OUT-NLB VF-IEDELIVERY-SS-DEV1-APIO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-20112-to-1050</u> | 1050 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-API-OUT-NLB VF-IEDELIVERY-SS-DEV1-APIO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-22104-to-33017</u> | 33017 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-API-OUT-NLB VF-IEDELIVERY-SS-DEV1-APIO-NLB | OUTBOUND | |

| | | | | | | | |
|------------------------------------|-------|-----|----|---------------------------|---|----------|-------------|
| <u>SS-DEV1-from-22105-to-33017</u> | 33017 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-API-OUT-NLB VF-IEDELIVERY-SS-DEV1-APIO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-10304-to-1521</u> | 1521 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-DB-NLB VF-IEDELIVERY-SS-DEV1-DBI-NLB | INBOUND | |
| <u>SS-DEV1-from-10305-to-2484</u> | 2484 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-DB-NLB VF-IEDELIVERY-SS-DEV1-DBI-NLB | INBOUND | |
| <u>SS-DEV1-from-22102-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-DB-OUT-NLB VF-IEDELIVERY-SS-DEV1-DBO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-22103-to-33001</u> | 33001 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-DB-OUT-NLB VF-IEDELIVERY-SS-DEV1-DBO-NLB | OUTBOUND | |
| <u>SS-DEV1-from-7001-to-7001</u> | 7001 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-NLB | INBOUND | |
| <u>SS-DEV1-from-8443-to-8443</u> | 8443 | TCP | IP | VF-IEDELIVERY-SS-DEV1-NLB | VF-IEDELIVERY-SS-DEV1-NLB | INBOUND | |
| SS-DEV1- | 22 | TCP | IP | N/A | VF-IEDELIVERY- | INBOUND | New for DPC |

| | | | | | | | |
|-----------------------------|-------|-----|----|-----|---|----------|-------------------------------------|
| from-35001-to-22 | | | | | SS-DEV1-SFTP-NLB VF-IEDELIVERY-SS-DEV1-SFTI-NLB | | |
| SS-DEV1-from-35002-to-35002 | 35002 | TCP | IP | N/A | VF-IEDELIVERY-SS-DEV1-DB-NLB VF-IEDELIVERY-SS-DEV1-DBI-NLB | INBOUND | New for DPC |
| SS-DEV1-from 38001 to 443 | 443 | TCP | IP | | VF-IEDELIVERY-SS-PROD-NLB | INBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38002 to 443 | 443 | | | | VF-IEDELIVERY-SS-DEV1-APII-NLB | INBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38003 to 1433 | 1433 | | | | VF-IEDELIVERY-SS-DEV1-DB-NLB | INBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38004 to 22 | 22 | | | | VF-IEDELIVERY-SS-DEV1-SFTP-NLB | INBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38005 to 1434 | 1434 | | | | VF-IEDELIVERY-SS-DEV1-DB-NLB | INBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38101 to 443 | 443 | TCP | IP | | VF-IEDELIVERY-SS-PROD-NLB | OUTBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38002 to 443 | 443 | | | | VF-IEDELIVERY-SS-DEV1-APII-NLB | OUTBOUND | ITR - 6946 GNP & GNM Number porting |

| | | | | | | | |
|----------------------------|------|--|--|--|--------------------------------|----------|-------------------------------------|
| SS-DEV1-from 38003 to 1433 | 1433 | | | | VF-IEDELIVERY-SS-DEV1-DB-NLB | OUTBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38004 to 22 | 22 | | | | VF-IEDELIVERY-SS-DEV1-SFTP-NLB | OUTBOUND | ITR - 6946 GNP & GNM Number porting |
| SS-DEV1-from 38005 to 1434 | 1434 | | | | VF-IEDELIVERY-SS-DEV1-DB-NLB | OUTBOUND | ITR - 6946 GNP & GNM Number porting |
| | | | | | | | |

Route 53:

| Non EKS Enviro nment | Route 53 Private Hosted Zone | Record Name | Type | Existing Value | Existin g NLB | New Valu e | Comment |
|----------------------|---------------------------------------|---|-------|--|---------------------------|------------|--|
| SIT1 | NA | NA | NA | NA | | | EKS SIT1 not required |
| SIT2 | internal.vodafone.com | portal.publish.sit2.equinox.vf-ie.internal.vodafone.com | CNAME | ns-1536.awsdns-00.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400 | VF-IEDELIVERY-SS-SIT2-NLB | | Do we need this for EKS SIT2 ? - Noel m Gilch rist, Voda fone (Ext e rnal) |
| SIT3 | internal.vodafone.com | * vf-ie.internal.vodafone.com | CNAME | sit3.haproxy.ieaws.vodafone.com | VF-IEDELIVERY-SS-SIT3-NLB | | Do we need this for EKS SIT2 ? - Noel m Gilch |

| | | | | | | |
|------|--|--|---------------------|---|---|---|
| | | | | | | rist, Voda fone (Exte rnal) |
| | internal.vi odafone.co m | ieesbbvr.dc- dublin.de/servicebalance/servicebal anceservice.internal.vodafone.com | CN AM E | sit3.haproxy.ieaw s.vodafone.com | VF- IEDEL IVERY -SS- SIT3- NLB | Wha t is this for? - Noel m Gilch rist, Voda fone (Exte rnal) |
| SIT4 | NA | NA | No CN AM E | | | EKS SIT1 not requi red |
| PRD1 | internal.vi odafone.co m | *vf-ie.internal.vodafone.com | CN AM E | prd1.haproxy.iea ws.vodafone.co m | VF- IEDEL IVERY -SS- PRD1- NLB | Do we need this for EKS SIT2 ? - Noel m Gilch rist, Voda fone (Exte rnal) |
| PRD2 | internal.vi odafone.co m | *vf-ie.internal.vodafone.com | CN AM E | prd2.haproxy.iea ws.vodafone.co m | VF- IEDEL IVERY -SS- PRD2- NLB | Do we need this for EKS SIT2 ? - Noel m Gilch rist, Voda fone (Exte rnal) |

| | | | | | | |
|----------|--|--|---------------|---|---|---|
| PRO D | internal.vi odafone.co m | * vf-ie.internal.vodafone.com | CN AM E | prod.haproxy.iea ws.vodafone.co m | VF- IEDEL IVERY -SS- PROD -NLB | Do we need this for EKS SIT2 ? - Noel m Gilch rist, Voda fone (Exte rnal) |
| | internal.vi odafone.co m | esb.app.prod.equinov.vf ie.internal.vodafone.com | CN AM E | prod.haproxy.iea ws.vodafone.co m | VF- IEDEL IVERY -SS- PROD -NLB | Wha t is this for? - Noel m Gilch rist, Voda fone (Exte rnal) |

| | Task | Environment | Owner | Status | Comment |
|---|--|-------------|-------------------|--------|---|
| 1 | Review NLB Design | | PCS | OPEN | POC in progress in DEV1, then we will present in front of PCS TDF for review. |
| 2 | Review open queries around existing interfaces | | Noel Gilchrist | OPEN | |
| 3 | Build new NLBs | DEV1 | PCS | OPEN | New NLB's are configured for DEV1 |
| 4 | PCS TDF Review | All | PCS | OPEN | 13/03: Scheduled a TDF review on 15/03. |
| 5 | Build new NLBs | EKS SIT3 | PCS | OPEN | 13/03 : PCS will start working on after approval on TDF review |
| 6 | Submit DNS Request for updated domains | | Noel Gilchrist | OPEN | |
| 7 | Review ongoing project requirements | | Noel Gilchrist | OPEN | |
| 8 | Add new Project Requirements to NLB Design | | Noel Gilchrist | OPEN | |

| | Task | Environment | Owner | Status | Comment |
|----|---|--------------------|--------------|---------------|----------------|
| 9 | Deploy new project requirements as per new NLB design | | PCS | OPEN | |
| 10 | Add new Route53 entries for Outbound interfaces | | PCS | OPEN | |
| 11 | Migrate Outbound NLB interfaces to new NLBs | | PCS / VFIE | OPEN | |
| 12 | Migrate inbound NLB interfaces to new NLBs | | PCS / VFIE | OPEN | |

6.6.5 Open Queries

| # | Query | Comment | Owner | Status |
|---|-------|---------|-------|--------|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |

7 01 VF IE SANDBOX VPC

SANDBOX TRN1 VPC

| Subnet | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|------------|------------------|-----------------|-------------------------------|-------------------------------|-------|
| | 10.181.20.0/27 | 255.255.255.224 | 10.181.20.0 - 10.181.20.31 | 10.181.20.1 - 10.181.20.30 | 30 |
| | 10.181.20.32/27 | 255.255.255.224 | 10.181.20.32 - 10.181.20.63 | 10.181.20.33 - 10.181.20.62 | 30 |
| Priv (NAT) | 10.181.20.64/27 | 255.255.255.224 | 10.181.20.64 - 10.181.20.95 | 10.181.20.65 - 10.181.20.94 | 30 |
| Priv (NAT) | 10.181.20.96/27 | 255.255.255.224 | 10.181.20.96 - 10.181.20.127 | 10.181.20.97 - 10.181.20.126 | 30 |
| Public | 10.181.20.128/27 | 255.255.255.224 | 10.181.20.128 - 10.181.20.159 | 10.181.20.129 - 10.181.20.158 | 30 |
| Public | 10.181.20.160/27 | 255.255.255.224 | 10.181.20.160 - 10.181.20.191 | 10.181.20.161 - 10.181.20.190 | 30 |
| Data | 10.181.20.192/27 | 255.255.255.224 | 10.181.20.192 - 10.181.20.223 | 10.181.20.193 - 10.181.20.222 | 30 |
| Data | 10.181.20.224/27 | 255.255.255.224 | 10.181.20.224 - 10.181.20.255 | 10.181.20.225 - 10.181.20.254 | 30 |
| Data | 10.181.21.0/27 | 255.255.255.224 | 10.181.21.0 - 10.181.21.31 | 10.181.21.1 - 10.181.21.30 | 30 |
| | 10.181.21.32/27 | 255.255.255.224 | 10.181.21.32 - 10.181.21.63 | 10.181.21.33 - 10.181.21.62 | 30 |
| Web | 10.181.21.64/27 | 255.255.255.224 | 10.181.21.64 - 10.181.21.95 | 10.181.21.65 - 10.181.21.94 | 30 |
| Web | 10.181.21.96/27 | 255.255.255.224 | 10.181.21.96 - 10.181.21.127 | 10.181.21.97 - 10.181.21.126 | 30 |
| | 10.181.21.128/27 | 255.255.255.224 | 10.181.21.128 - 10.181.21.159 | 10.181.21.129 - 10.181.21.158 | 30 |
| | 10.181.21.160/27 | 255.255.255.224 | 10.181.21.160 - 10.181.21.191 | 10.181.21.161 - 10.181.21.190 | 30 |
| | 10.181.21.192/27 | 255.255.255.224 | 10.181.21.192 - 10.181.21.223 | 10.181.21.193 - 10.181.21.222 | 30 |
| | 10.181.21.224/27 | 255.255.255.224 | 10.181.21.224 - 10.181.21.255 | 10.181.21.225 - 10.181.21.254 | 30 |

8 02 VF IE TEST VPCs

8.1 1. Introduction

- Code: Terraform
- Repository: <https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/vf-iedelivery-infrastructure>
- Branch: test
- AWS account: vf-iedelivery-mgmt - 831341508773
- CI/CD: CodePipeline <https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-infrastructure-test-tf/view?region=eu-west-1>

Each VPC will be composed of:

- 1 Web tier subnet per AZ. Subnet IP range: '/23'
- 1 Presentation tier subnet per AZ. Subnet IP range: '/23'
- 1 Database tier subnet per AZ. Subnet IP range: '/23'
- 1 Private subnet for other purposes per AZ. Subnet IP range: '/27' (*to be modified to /24)
- VPC Flow logs enabled
- 1 different route table for each subnet.

8.2 2. Design of TEST VPC's

These are the subnet CIDR ranges that were chosen for test environment:

| VPC | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|-----|-----------------|---------------|-------------------------------|-------------------------------|-------|
| ST1 | 10.181.128.0/19 | 255.255.224.0 | 10.181.128.0 - 10.181.159.255 | 10.181.128.1 - 10.181.159.254 | 8190 |
| ST2 | 10.181.160.0/19 | 255.255.224.0 | 10.181.160.0 - 10.181.191.255 | 10.181.160.1 - 10.181.191.254 | 8190 |
| ST3 | 10.181.192.0/19 | 255.255.224.0 | 10.181.192.0 - 10.181.223.255 | 10.181.192.1 - 10.181.223.254 | 8190 |
| ST4 | 10.181.224.0/19 | 255.255.224.0 | 10.181.224.0 - 10.181.255.255 | 10.181.224.1 - 10.181.255.254 | 8190 |

The following diagram shows the state for the infrastructure deployed in AWS VF IE test account by PCS. In each VPC, 3 subnets per AZ were created: web tier, presentation tier and database tier. Additionally, although in the diagram is not shown for simplicity reasons, in each VPC there are also others 2 private subnets that can be used for other purposes (endpoints, ENI, etc.).



The network was split following this pattern. This table shows the subdivision for ST4, where the first six subnets with range /23 were used for web, presentation and database tiers.

"**10.181.254.64/27**" and "**10.181.254.96/27**" were used for private subnets that can be used for other purposes.

| Subnet address | Purpose | Netmask | Range of addresses | Useable IPs | Hosts |
|------------------------|--------------|---------------|-------------------------------|-------------------------------|-------|
| 10.181.224.0/23 | Web | 255.255.254.0 | 10.181.224.0 - 10.181.225.255 | 10.181.224.1 - 10.181.225.254 | 510 |
| 10.181.226.0/23 | Presentation | 255.255.254.0 | 10.181.226.0 - 10.181.227.255 | 10.181.226.1 - 10.181.227.254 | 510 |
| 10.181.228.0/23 | Database | 255.255.254.0 | 10.181.228.0 - 10.181.229.255 | 10.181.228.1 - 10.181.229.254 | 510 |
| 10.181.230.0/23 | Web | 255.255.254.0 | 10.181.230.0 - 10.181.231.255 | 10.181.230.1 - 10.181.231.254 | 510 |
| 10.181.232.0/23 | Presentation | 255.255.254.0 | 10.181.232.0 - 10.181.233.255 | 10.181.232.1 - 10.181.233.254 | 510 |
| 10.181.234.0/23 | Database | 255.255.254.0 | 10.181.234.0 - 10.181.235.255 | 10.181.234.1 - 10.181.235.254 | 510 |
| 10.181.236.0/23 | - | 255.255.254.0 | 10.181.236.0 - 10.181.237.255 | 10.181.236.1 - 10.181.237.254 | 510 |

| | | | | | |
|-------------------------|-------|-----------------|------------------------------------|------------------------------------|-----|
| 10.181.238.0/23 | - | 255.255.254.0 | 10.181.238.0 - 10.181.239.255 | 10.181.238.1 - 10.181.239.254 | 510 |
| 10.181.240.0/23 | - | 255.255.254.0 | 10.181.240.0 - 10.181.241.255 | 10.181.240.1 - 10.181.241.254 | 510 |
| 10.181.242.0/23 | - | 255.255.254.0 | 10.181.242.0 - 10.181.243.255 | 10.181.242.1 - 10.181.243.254 | 510 |
| 10.181.244.0/23 | - | 255.255.254.0 | 10.181.244.0 - 10.181.245.255 | 10.181.244.1 - 10.181.245.254 | 510 |
| 10.181.246.0/23 | - | 255.255.254.0 | 10.181.246.0 - 10.181.247.255 | 10.181.246.1 - 10.181.247.254 | 510 |
| 10.181.248.0/23 | - | 255.255.254.0 | 10.181.248.0 - 10.181.249.255 | 10.181.248.1 - 10.181.249.254 | 510 |
| 10.181.250.0/23 | - | 255.255.254.0 | 10.181.250.0 - 10.181.251.255 | 10.181.250.1 - 10.181.251.254 | 510 |
| 10.181.252.0/23 | - | 255.255.254.0 | 10.181.252.0 - 10.181.253.255 | 10.181.252.1 - 10.181.253.254 | 510 |
| 10.181.254.0/26 | - | 255.255.255.192 | 10.181.254.0 - 10.181.254.63 | 10.181.254.1 - 10.181.254.62 | 62 |
| 10.181.254.64/27 | Other | 255.255.255.224 | 10.181.254.64 - 10.181.254.95 | 10.181.254.65 - 10.181.254.94 | 30 |
| 10.181.254.96/27 | Other | 255.255.255.224 | 10.181.254.96 - 10.181.254.127 | 10.181.254.97 - 10.181.254.126 | 30 |
| 10.181.254.128/26 | - | 255.255.255.192 | 10.181.254.128 - 10.181.254.191 | 10.181.254.129 - 10.181.254.190 | 62 |
| 10.181.254.192/26 | - | 255.255.255.192 | 10.181.254.192 - 10.181.254.255 | 10.181.254.193 - 10.181.254.254 | 62 |
| 10.181.255.0/25 | - | 255.255.255.128 | 10.181.255.0 - 10.181.255.127 | 10.181.255.1 - 10.181.255.126 | 126 |
| 10.181.255.128/25 | - | 255.255.255.128 | 10.181.255.128 - 10.181.255.255 | 10.181.255.129 - 10.181.255.254 | 126 |

8.3 00 VF IE SIT4 - HOW TO

- [PHASE 1](#)
 - [1.DNS](#)
 - [2.OFMW AMI](#)
 - [3.EC2 SSH KEY PAIR](#)
 - [4.VARIABLES](#)
- [PHASE 2](#)
 - [1. VPC \(already created\)](#)
 - [2. EC2 SERVERS and route53 records](#)
 - [3. Security groups](#)

- [4. EFS volumes](#)
- [5. ALB](#)
- [6.RDS](#)
- [7. SFTP and OCM](#)

This is a guide about how to create a new environment for VFIE project, specifically for CCH-SAL, environment sit4.

8.3.1 PHASE 1

First of all, we are going to do any configuration that the environment needs (before starting to create any new resource). Please, take into account that VPC's and subnets were firstly created, and are not part of the scope of this guide.

All environments need:

- DNS hosted zones:
 - Public hosted zone: subdomain of ieaws.vodafone.com that is defined in SS account.
 - Private hosted zone to route the traffic with gcd as destination to haproxy.
- AMI. Customized AMI's are created in SS and shared with all tenant accounts. The tenant account will need to copy those AMI's to Ireland region.
 - OFMW 1213
 - OFMW 1221
- SSH Keys. Right now they are created manually and added to SSM parameter store. (TO BE REVIEWED THIS PROCESS)
- Variables.

8.3.1.1 1.DNS

- codecommit repository: vf-iedelivery-dns
- documentation for dns: <https://confluence.sp.vodafone.com/x/1HUICg>

First, create the public domain in test account vf-iedelivery-test-01. Example for sit4:

```
# PUBLIC HOSTED ZONE IN TEST FOR SUB-DOMAIN sit4.ieaws.vodafone.com
resource "aws_route53_zone" "vfie_subdomain_sit4" {
  provider = "aws.test"
  name = "sit4.ieaws.vodafone.com"
  comment = "vf ie subdomain delegation for sit4"
  tags = merge(
    local.common_tags,
    {
      "Name"      = "VFIE SUBDOMAIN sit4"
    },
  )
}
```

Then, create NS record in SS domain (ieaws.vodafone.com) in vf-iedelivery-prod-ss-01.Example for sit4:

```
#TEST NS RECORDS
resource "aws_route53_record" "vfie_subdomain_sit4_ss" {
  provider = "aws.SS"
  zone_id = aws_route53_zone.vfie_subdomain.zone_id
  name    = "sit4.ieaws.vodafone.com"
  type    = "NS"
  ttl     = "60"
  records = [
    aws_route53_zone.vfie_subdomain_sit4.name_servers.0,
    aws_route53_zone.vfie_subdomain_sit4.name_servers.1,
    aws_route53_zone.vfie_subdomain_sit4.name_servers.2,
    aws_route53_zone.vfie_subdomain_sit4.name_servers.3,
  ]
}
```

Once SIT4 Haproxy is ready (at the moment not yet created), we will add the following:

- Data resource for SIT4 VPC

```
data "aws_vpc" "sit4-vpc" {
  provider = "aws.test"
  filter {
    name    = "tag:Name"
    values = ["SIT4-VF-IEDELIVERY-VPC-TEST"]
  }
}
```

- Private hosted zone to route traffic with gcd destination to SS:

```
# PRIVATE HOSTED ZONE IN sit4 FOR RESOLVE GDC DOMAINS
resource "aws_route53_zone" "vfie_priv_sit4_gdc_nlb" {
  provider = "aws.test"
  name = "internal.vodafone.com"
  comment = "private hosted zone to send traffic to gcd - sit4 vpc"
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4 SS NLB"
    },
  )
  vpc {
    vpc_id = data.aws_vpc.sit4-vpc.id
  }
}
```

- DNS record to point to SIT4 HAProxy:

```
# ROUTING TRAFFIC TO sit4 SS NLB
resource "aws_route53_record" "vfie_priv_sit4_gdc_nlb_record" {
  provider = "aws.test"
  zone_id = aws_route53_zone.vfie_priv_sit4_gdc_nlb.zone_id
  name    = "*vf-ie.internal.vodafone.com"
  type    = "CNAME"
  ttl     = "60"
  records      = ["sit4.haproxy.ieaws.vodafone.com"]
}
```

8.3.1.2 2. OFMW AMI

We need to make sure ofmw 1213 and 1221 ami's have been shared with test account. Go to vf-iedelivery-prod-ss-01, in ec2 tab, check AMI:

| Name | AMI Name | AMI ID | Source | Owner | Visibility |
|------|--|------------------------|-------------------------------------|--------------|------------|
| | VFIE-DELIVERY-OFMW-1221 2020-07-28T15-49-23.609Z | ami-0d697b5f8eb72d72 | 267040142128/VFIE-DELIVERY-OFMW-... | 267040142128 | Private |
| | VFIE-DELIVERY-OFMW-1213 2020-07-28T15-49-28.127Z | ami-0ddcd901494b78aeec | 267040142128/VFIE-DELIVERY-OFMW-... | 267040142128 | Private |

8.3.1.3 3. EC2 SSH KEY PAIR

Create a ssh key pair (example key name sit4-cch-sal) and add the private content into a ssm parameter (secure string, example /test/sit4/cch-sal/ec2/ssh-key). This is done in test account vf-iedelivery-test-01.

8.3.1.4 4. VARIABLES

The following example is just for sit4 environment and with the assumption each environment will have the same number of nodes:

```

variable "DEPLOY_ROLE" {}
variable "ENV" {
  description = "Name of the environment"
}
variable "PROJECT" {
  description = "Name of the project, used in a lot of the resource naming"
}

variable "account_id" {
  default = ""
}
variable "sit4_main_cidr_block" {
  default = ""
}
variable "sit4_private_subnets" {
  default = []
}
variable "sit4_web_subnets" {
  default = []
}
variable "sit4_database_subnets" {
  default = []
}
variable "sit4_presentation_subnets" {
  default = []
}
variable "hosted_zone_sit4"{
  default = ""
}
variable "ec2_key_pair_name_sit4"{
  default = ""
}
variable "sit4_domain_names_acm_8011"{
  description = "List of domains to associate with a ACM certificate.
Maximum 50"
  type = "list"
  default = [""]
}
variable "sit4_domain_names_acm_7001"{
  description = "List of domains to associate with a ACM certificate.
Maximum 50"
  type = "list"
  default = [""]
}
#-----CIDR
variable "cidr_sit4"{
  description = "ip address range for sit4"
  default      = ["10.181.224.0/19"]
}
variable "cidr_workspaces"{
  description = "ip address range for workspaces"
  default      = ["10.181.22.0/23"]
}
variable "cidr_myst"{
  description = "ip address range for myst"
  default      = ["10.181.16.0/23"]
}
variable "cidr_ss_vpc_test"{
  description = "ip address range for ss ss test"
  default      = ["198.19.220.128/26", "198.19.220.192/26"]
}
#-----DOMAINS VARIABLES-----
variable "cch_ob_odi_nodes" {
  type      = number
}

```

```

    default = 0
}
variable "cch_ob_osb_nodes" {
  type    = number
  default = 0
}
variable "cch_ob_soa_nodes" {
  type    = number
  default = 0
}
variable "cch_ob_ums_osb_nodes" {
  type    = number
  default = 0
}
variable "cch_ob_ums_soa_nodes" {
  type    = number
  default = 0
}
variable "cch_ib_sal_osb_nodes" {
  type    = number
  default = 0
}
variable "cch_ib_sal_soa_nodes" {
  type    = number
  default = 0
}
variable "cch_portal_nodes" {
  type    = number
  default = 0
}
variable "pb_osb_nodes" {
  type    = number
  default = 0
}
variable "pb_odi_nodes" {
  type    = number
  default = 0
}
variable "oal_osb_nodes" {
  type    = number
  default = 0
}
variable "oal_soa_nodes" {
  type    = number
  default = 0
}
variable "cch_sal_web_nodes" {
  type    = number
  default = 0
}
}

```

For sit4, in vars folder, the file TEST.tfvars have the following values:

```

ENV = "TEST"
PROJECT = "vfie-delivery"
account_id = "046978237480"
ec2_key_pair_name_sit4 = "sit4-cch-sal"
hosted_zone_sit4 = "sit4.ieaws.vodafone.com."
sit4_main_cidr_block = "10.181.224.0/19"
sit4_private_subnets = ["10.181.254.64/27", "10.181.254.96/27",
"10.181.252.0/24", "10.181.253.0/24"]
sit4_web_subnets = ["10.181.224.0/23", "10.181.230.0/23"]
sit4_presentation_subnets = ["10.181.226.0/23", "10.181.232.0/23"]
sit4_database_subnets = ["10.181.228.0/23", "10.181.234.0/23"]
cch_ob_odi_nodes = 1
cch_ob_osb_nodes = 1
cch_ob_soa_nodes = 1
cch_ob_ums_osb_nodes = 1
cch_ob_ums_soa_nodes = 1
cch_ib_sal_osb_nodes = 1
cch_ib_sal_soa_nodes = 1
cch_portal_nodes = 1
pb_osb_nodes = 1
pb_odi_nodes = 1
oal_osb_nodes = 1
oal_soa_nodes = 1
cch_sal_web_nodes = 1

```

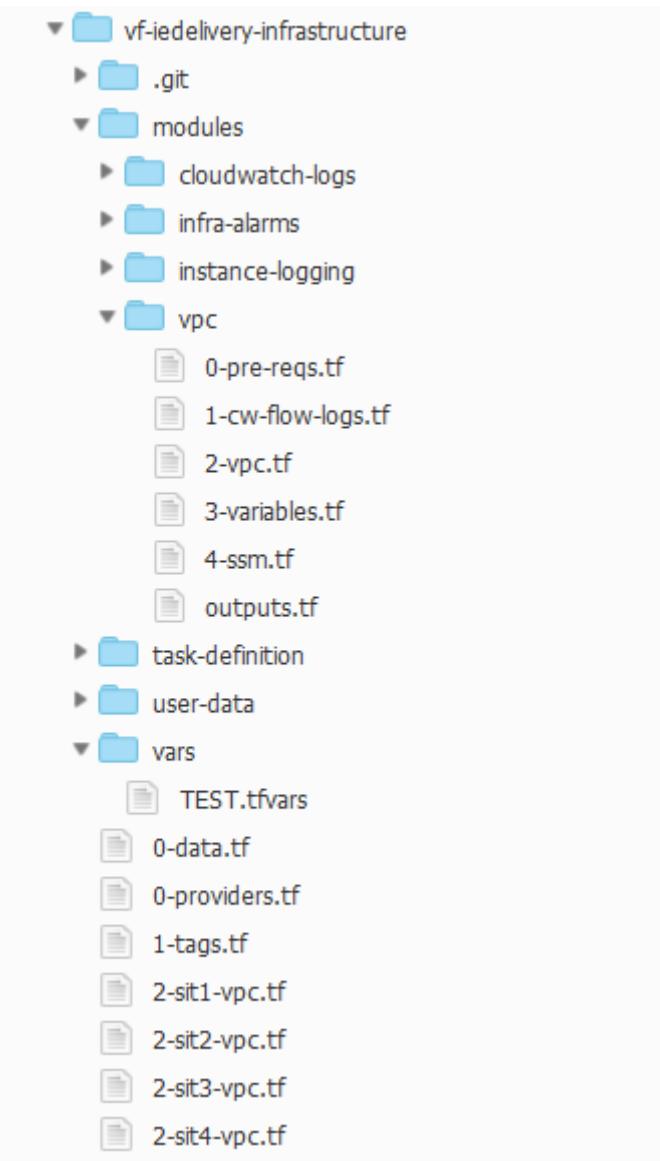
8.3.2 PHASE 2

This is the order we will follow:

1. servers, security groups, efs volume and logging.
2. ALB
3. RDS
4. SFTP and OCM log groups and iam. Later, SFTP and OCM fargate and nlb.
5. monitoring

8.3.2.1 1. VPC (already created)

The VPC and its components will be creating using a terraform module called 'vpc'.



For sit4, in the file called 2-sit4-vpc.tf (root path), we declare the vpc module for the environment sit4:

```

module "test-vpc-4" {
  source          = "./modules/vpc"
  TAGS           = local.common_tags
  ENV            = var.ENV
  PROJECT        = var.PROJECT
  VPC_NAME       = "SIT4"
  main_cidr_block = var.sit4_main_cidr_block
  enable_ipv6    = false
  enable_dns_hostnames = true
  enable_dns_support = true
  private_subnets = var.sit4_private_subnets
  web_tier_subnets = var.sit4_web_subnets
  pres_tier_subnets = var.sit4_presentation_subnets
  database_tier_subnets = var.sit4_database_subnets
  enable_internet_gateway = false
  enable_nat_gateway = false
  enable_ssm        = true
}

```

We define how the specific VPC is going to be like with the following variables:

- VPC_NAME : this will define the environment (sit1, sit2, sit3, sit4, prd1, prd2, prod). It will be part of the naming convention for all resources.
- main_cidr_block
- enable_dns_hostnames and enable_dns_support (always true)
- private_subnets, web_tier_subnets, pres_tier_subnets, database_tier_subnets. Public_subnets are empty.
- enable_internet_gateway = false as there will not be locally internet connectivity, but a centralized one through SS account.
- enable_nat_gateway = false
- enable_ssm = true; in case we need to connect to the servers through SSM

Please, see the code for the vpc module here :<https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-infrastructure/browse/refs/heads/test/./modules/vpc?region=eu-west-1> (vf-iedelivery-mgmt account).

The module will create:

- VPC and subnets
- Route tables
- SSM endpoints and security groups
- VPC flow logs to s3 and CW

8.3.2.2 2. EC2 SERVERS and route53 records
[Here](#)

8.3.2.3 3. Security groups
[Here](#)

Note: we will be commenting out the lines where the sftp-ocm nlb appears until this is created.

8.3.2.4 4. EFS volumes

[Here](#)

8.3.2.5 5. ALB

[Here](#)

8.3.2.6 6.RDS

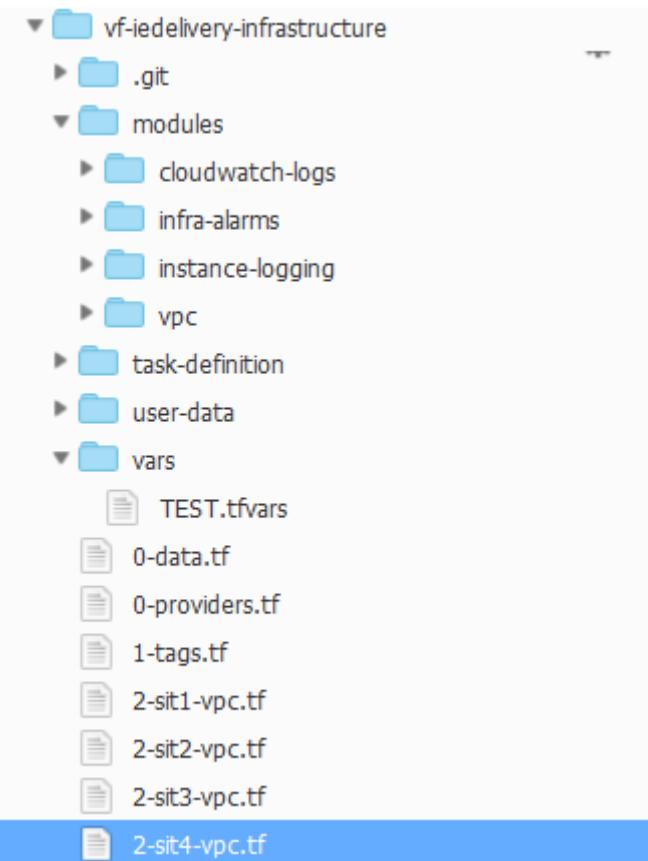
[Here](#)

8.3.2.7 7. SFTP and OCM

Note: ECR permissions once IAM roles are created (must be done before ecs services are created). Update account id in task definition for ocm.

8.3.3 01 SIT4 SERVERS

In the file called 2-sit4-vpc.tf we will define all ec2 instances, user data, route53 records for the instances and logging.



First of all, the AMI will be read in terraform as a data resource. It is defined in the root directory, in the file called 0-data.tf. This data will read the latest AMI in the target account (test in this case). WARNING: target account (test) must be the OWNER of the AMI. Meaning: if the AMI was shared with target account, a copy of it must be created in Ireland region.

```

data "aws_route53_zone" "sit4HostedZone" {
  name          = var.hosted_zone_sit4
}

-----data specific ami
data "aws_ami" "ofmw1213" {
  most_recent    = true
  owners         = ["self"]
  filter {
    name    = "name"
    values  = ["VFIE-DELIVERY-OFMW-1213*"]
  }
}
-----data specific ami
data "aws_ami" "ofmw1221" {
  most_recent    = true
  owners         = ["self"]
  filter {
    name    = "name"
    values  = ["VFIE-DELIVERY-OFMW-1221*"]
  }
}
-----data hardened ami rhel7.7
data "aws_ami" "rhel_hardened_ami" {
  most_recent    = true
  owners         = ["self"]
  filter {
    name    = "name"
    values  = ["vf-gdc-rhel-7.7-hvm-*"]
  }
}

```

Let's take the example for one OFMW domain, CCH_OB_ODI.

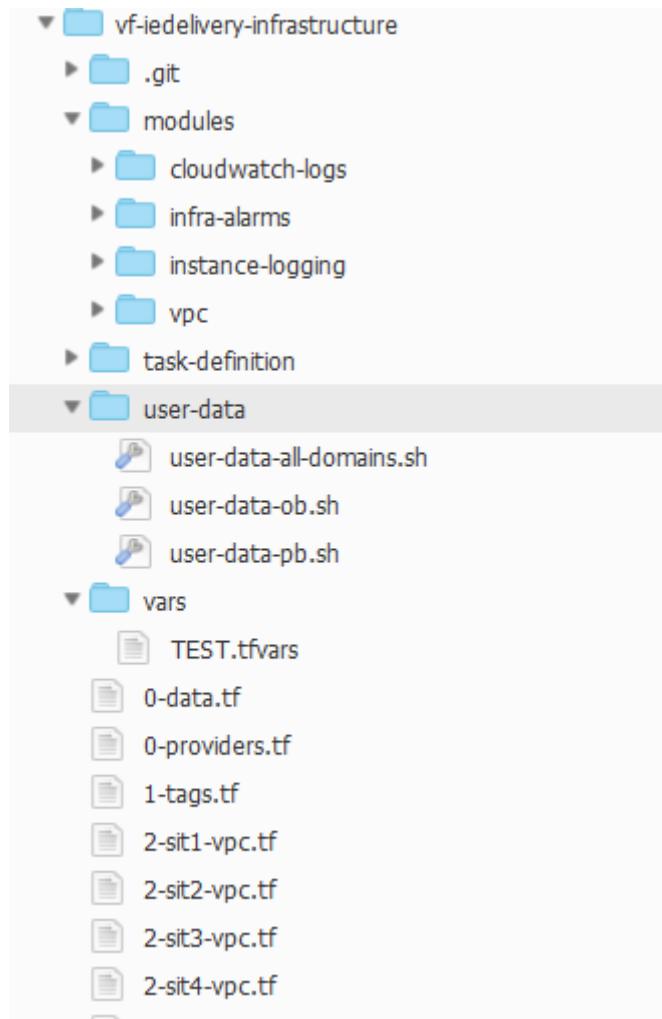
8.3.3.1 User data

```

data "template_file" "sit4-cch-ob-odi-admin" {
  depends_on = [aws_efs_file_system.sit4-cch-ob-odi,
aws_efs_file_system.sit4-cch-ob-multidomain-share,
aws_efs_file_system.sit4-software]
  template    = "${file("${path.module}/user-data/user-data-ob.sh")}"
  vars = {
    SoftwareEFS = aws_efs_file_system.sit4-software.id
    Region      = "eu-west-1"
    ssm_parameter = module.sit4-cch-ob-odi-
logging.MOD_CWAGENT_BASE_CONFIG_SSM_PARAM_ADMIN.name
    DomainEFS   = aws_efs_file_system.sit4-cch-ob-odi.id #EFS ID OF
INSTANCE
    CchObMultiDomainEFS = aws_efs_file_system.sit4-cch-ob-multidomain-
share.id
  }
}
# User data for Managed servers
data "template_file" "sit4-cch-ob-odi-managed" {
  depends_on = [aws_efs_file_system.sit4-cch-ob-odi,
aws_efs_file_system.sit4-cch-ob-multidomain-share,
aws_efs_file_system.sit4-software]
  template    = "${file("${path.module}/user-data/user-data-ob.sh")}"
  vars = {
    SoftwareEFS = aws_efs_file_system.sit4-software.id
    Region      = "eu-west-1"
    ssm_parameter = module.sit4-cch-ob-odi-
logging.MOD_CWAGENT_BASE_CONFIG_SSM_PARAM_MANAGED.name
    DomainEFS   = aws_efs_file_system.sit4-cch-ob-odi.id #EFS ID OF
INSTANCE
    CchObMultiDomainEFS = aws_efs_file_system.sit4-cch-ob-multidomain-
share.id
  }
}

```

We have two different type of user data per domain because of the application logging (done by simplepoint). The template file used is the same, you can find it in the folder user-data.



Once that's defined, we declare the **admin server** ec2 instance:

```

resource "aws_instance" "sit4-cch-ob-odi-as" {
  depends_on          = [aws_efs_file_system.sit4-cch-ob-
  odi, aws_security_group.sit4-cch-ob-odi-as, aws_security_group.sit4-cch-ob-
  odi ]
  ami                 = data.aws_ami.ofmw1213.id
  instance_type       = "m5.large"
  monitoring         = true
  subnet_id          = module.test-vpc-4.subnet_presentation_ids[0]
  iam_instance_profile = module.test-vpc-
4.iam_instance_profile_ec2_ssm_id
  # user_data          = data.template_file.sit4-cch-ob-odi.rendered
  user_data          = data.template_file.sit4-cch-ob-odi-
admin.rendered
  key_name            = var.ec2_key_pair_name_sit4
  vpc_security_group_ids = [aws_security_group.sit4-ec2-
monitoring.id,aws_security_group.sit4-cch-ob-odi-as.id,
aws_security_group.sit4-cch-ob-odi.id, aws_security_group.sit4-ssm-
jumpbox.id]
  tags                = merge(
    local.common_tags,
    {
      "Name" = "sit4-cch-ob-odi-as"
    },
  )
  root_block_device{
    volume_type = "gp2"
    volume_size = 60
    encrypted   = true
  }
  lifecycle {
    ignore_changes = [user_data,ami]
  }
}

```

And the route53 record:

```

resource "aws_route53_record" "cch-ob-odi-as" {
  zone_id = data.aws_route53_zone.sit4HostedZone.zone_id
  name    = "cch-ob-odi-as.sit4.ieaws.vodafone.com"
  type    = "A"
  ttl     = "300"
  records = [aws_instance.sit4-cch-ob-odi-as.private_ip]
}

```

And then the nodes:

```
#EC2 sit4-cch-ob-odiX cch-ob-odiX.sit4.ieaws.vodafone.com
resource "aws_instance" "sit4-cch-ob-odi" {
  depends_on = [aws_efs_file_system.sit4-cch-ob-odi, aws_security_group.sit4-cch-ob-odi-ms, aws_security_group.sit4-cch-ob-odi]
  count      = var.cch_ob_odi_nodes
  ami        = data.aws_ami.ofmw1213.id
  instance_type = "m5.large"
  monitoring = true
  subnet_id   = count.index%2==0 ? module.test-vpc-4.subnet_presentation_ids[0] : module.test-vpc-4.subnet_presentation_ids[1]
  iam_instance_profile = module.test-vpc-4.iam_instance_profile_ec2_ssm_id
  user_data    = data.template_file.sit4-cch-ob-odi-managed.rendered
  key_name     = var.ec2_key_pair_name_sit4
  vpc_security_group_ids = [aws_security_group.sit4-ec2-monitoring.id, aws_security_group.sit4-cch-ob-odi-ms.id, aws_security_group.sit4-cch-ob-odi.id, aws_security_group.sit4-ssm-jumpbox.id]
  tags         = merge(
    local.common_tags,
    {
      "Name" = "sit4-cch-ob-odi${count.index+1}"
    },
  )
  root_block_device{
    volume_type = "gp2"
    volume_size = 60
    encrypted = true
  }
  lifecycle {
    ignore_changes = [user_data, ami]
  }
}
resource "aws_route53_record" "cch-ob-odi-node" {
  zone_id = data.aws_route53_zone.sit4HostedZone.zone_id
  count   = var.cch_ob_odi_nodes
  name    = "cch-ob-odi${count.index+1}.sit4.ieaws.vodafone.com"
  type    = "A"
  ttl     = "300"
  records = [aws_instance.sit4-cch-ob-odi[count.index].private_ip]
}
```

Finally, the part of application logging (done and managed by singlepoint):

```
#logging
module "sit4-cch-ob-odi-logging" {
  source = "./modules/cloudwatch-logs"
  TAGS   = local.common_tags
  VPC_NAME ="SIT4"
  Environment ="sit4"
  OFMWDomain = "cch-ob-odi"
  ManagedServerTemplate = "odi" ## can be standard, odi or ohs
  enable_odi = "1"
  enable_ohs = "0"
}
```

8.3.4 02 SIT4 SECURITY GROUPS

- [1.OFMW domain ec2 security groups](#)
- [2.OFMW domain efs security groups](#)

- [3.OFMW OHS domain ec2 security groups](#)
- [4.RDS ORACLE SECURITY GROUPS](#)
- [5.OCM AND SFTP SECURITY GROUPS](#)
- [6. ssm jumpbox and monitoring](#)

This section will explain the terraform code for all the security groups. The requirements are specified here: <https://confluence.sp.vodafone.com/x/smnezCQ>

The file with all the security groups for sit4 can be found here: <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-infrastructure/browse/refs/heads/test/--/3-sit4-securitygroups.tf?region=eu-west-1>

8.3.4.1 1.OFMW domain ec2 security groups

Each OFMW domain will have associate these security groups:

- environment-domain-as (admin server). Example: sit4-cch-ob-odi-as
- environment-domain-ms (node server). Example: sit4-cch-ob-odi-ms
- environment-domain. Example: sit4-cch-ob-odi
- environment-ec2-monitoring*(new!). Example: sit4-ec2-monitoring

Security groups definition for CCH_OB_ODI domain:

```

#####
# CCH_OB_ODI SECURITY GROUPS
#####
#sit4-cch-ob-odi
resource "aws_security_group" "sit4-cch-ob-odi" {
    name          = "sit4-cch-ob-odi"
    description   = "Security group for sit4-cch-ob-odi"
    vpc_id        = module.test-vpc-4.vpc_id
    tags          = merge(
        local.common_tags,
        {
            "Name"      = "sit4-cch-ob-odi"
        },
    )
}

resource "aws_security_group_rule" "sit4-cch-ob-odi_i_1" {
    type          = "ingress"
    from_port     = 22
    to_port       = 22
    protocol      = "tcp"
    cidr_blocks   = var.cidr_myst
    description   = "allow inbound from Myst VPC."
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_i_2" {
    type          = "ingress"
    from_port     = 0
    to_port       = 0
    protocol      = "-1"
    self          = true
    description   = "allow inbound within SG."
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_1" {
    type          = "egress"
    from_port     = 0
    to_port       = 0
    protocol      = "-1"
    self          = true
    description   = "allow outbound within SG."
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_2" {
    type          = "egress"
    from_port     = 2049
    to_port       = 2049
    protocol      = "tcp"
    source_security_group_id = aws_security_group.sit4-cch-ob-odi-efs.id
    description   = "allow inbound from sit4-cch-ob-odi-efs"
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_3" {
    type          = "egress"
    from_port     = 1521
    to_port       = 1521
    protocol      = "tcp"
    source_security_group_id = aws_security_group.sit4-rds-oracle-db.id
    description   = "allow inbound from sit4-rds-oracle-db"
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_4" {
    type          = "egress"
    from_port     = 2049
    to_port       = 2049
    protocol      = "tcp"
}

```

```

    source_security_group_id = aws_security_group.sit4-cch-ob-multidomain-
share.id
    description      = "allow inbound from sit4-cch-ob-multidomain-share"
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_5" {
    type          = "egress"
    from_port     = 2049
    to_port       = 2049
    protocol      = "tcp"
    source_security_group_id = aws_security_group.sit4-software.id
    description   = "allow inbound from sit4-software "
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_6_SMSC" {
    type          = "egress"
    from_port     = 20111
    to_port       = 20111
    protocol      = "tcp"
    cidr_blocks   = var.cidr_ss_vpc_test
    description   = "allow outbound from SMSC-IF355.01"
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_7_SMSC" {
    type          = "egress"
    from_port     = 20112
    to_port       = 20112
    protocol      = "tcp"
    cidr_blocks   = var.cidr_ss_vpc_test
    description   = "allow outbound from SMSC-IF355.01"
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_8_SMSC" {
    type          = "egress"
    from_port     = 20113
    to_port       = 20113
    protocol      = "tcp"
    cidr_blocks   = var.cidr_ss_vpc_test
    description   = "allow outbound from SMSC-IF355.01"
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}
resource "aws_security_group_rule" "sit4-cch-ob-odi_eg_9_SMSC" {
    type          = "egress"
    from_port     = 20114
    to_port       = 20114
    protocol      = "tcp"
    cidr_blocks   = var.cidr_ss_vpc_test
    description   = "allow outbound from SMSC-IF355.01"
    security_group_id = aws_security_group.sit4-cch-ob-odi.id
}

#sit4-cch-ob-odi-as
resource "aws_security_group" "sit4-cch-ob-odi-as" {
    name        = "sit4-cch-ob-odi-as"
    description = "Security group for sit4-cch-ob-odi-as"
    vpc_id      = module.test-vpc-4.vpc_id
    tags = merge(
        local.common_tags,
        {
            "Name"      = "sit4-cch-ob-odi-as"
        },
    )
}
resource "aws_security_group_rule" "sit4-cch-ob-odi-as_i_1" {
    type          = "ingress"

```

```

from_port      = 7001
to_port       = 7001
protocol      = "tcp"
source_security_group_id = aws_security_group.sit4-cch-sal-web.id
description    = "allow inbound from sit4-cch-sal-web"
security_group_id = aws_security_group.sit4-cch-ob-odi-as.id
}

resource "aws_security_group_rule" "sit4-cch-ob-odi-as_eg_1" {
  type          = "egress"
  from_port     = 443
  to_port       = 443
  protocol      = "tcp"
  cidr_blocks   = var.cidr_myst
  description    = "allow outbound to Artifactory ALB."
  security_group_id = aws_security_group.sit4-cch-ob-odi-as.id
}
#sit4-cch-ob-odi-ms
resource "aws_security_group" "sit4-cch-ob-odi-ms" {
  name          = "sit4-cch-ob-odi-ms"
  description   = "Security group for sit4-cch-ob-odi-ms"
  vpc_id        = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-cch-ob-odi-ms"
    },
  )
}

resource "aws_security_group_rule" "sit4-cch-ob-odi-ms_i_1" {
  type          = "ingress"
  from_port     = 15101
  to_port       = 15101
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-web.id
  description    = "allow inbound from sit4-cch-sal-web."
  security_group_id = aws_security_group.sit4-cch-ob-odi-ms.id
}

```

8.3.4.2 2.OFMW domain efs security groups

```

#sit4-cch-ob-odi-efs
resource "aws_security_group" "sit4-cch-ob-odi-efs" {
  name          = "sit4-cch-ob-odi-efs"
  description   = "Security group for sit4-cch-ob-odi-efs"
  vpc_id        = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-cch-ob-odi-efs"
    },
  )
}

```

8.3.4.3 3.OFMW OHS domain ec2 security groups

```

#####
# CCH_SAL_WEB SECURITY GROUPS
#####

#sit4-cch-sal-alb
resource "aws_security_group" "sit4-cch-sal-alb" {
    name          = "sit4-cch-sal-alb"
    description   = "Security group for sit4-cch-sal-alb"
    vpc_id        = module.test-vpc-4.vpc_id
    tags          = merge(
        local.common_tags,
        {
            "Name"      = "sit4-cch-sal-alb"
        },
    )
}

#WORKSPACES
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_1" {
    type          = "ingress"
    from_port     = 7001
    to_port       = 7001
    protocol      = "tcp"
    cidr_blocks   = var.cidr_workspaces
    description   = "allow inbound from Internet VPC (workspaces)."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_2" {
    type          = "ingress"
    from_port     = 443
    to_port       = 443
    protocol      = "tcp"
    cidr_blocks   = var.cidr_workspaces
    description   = "allow inbound from Internet VPC (workspaces)."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_3" {
    type          = "ingress"
    from_port     = 8011
    to_port       = 8011
    protocol      = "tcp"
    cidr_blocks   = var.cidr_workspaces
    description   = "allow inbound from Internet VPC (workspaces)."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
#HAProxy
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_4" {
    type          = "ingress"
    from_port     = 443
    to_port       = 443
    protocol      = "tcp"
    cidr_blocks   = var.cidr_ss_vpc_test
    description   = "allow inbound from HAProxy."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_5" {
    type          = "ingress"
    from_port     = 7001
    to_port       = 7001
    protocol      = "tcp"
    cidr_blocks   = var.cidr_ss_vpc_test
    description   = "allow inbound from HAProxy."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_6" {
    type          = "ingress"

```

```

from_port      = 8011
to_port       = 8011
protocol      = "tcp"
cidr_blocks   = var.cidr_ss_vpc_test
description    = "allow inbound from HAProxy."
security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
#DOMAINS
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_7" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-odi.id
  description    = "allow inbound from sit4-cch-ob-odi."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_7b" {
  type          = "ingress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-odi.id
  description    = "allow inbound from sit4-cch-ob-odi."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}

resource "aws_security_group_rule" "sit4-cch-sal-alb_i_8" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-osb.id
  description    = "allow inbound from sit4-cch-ob-osb."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_9" {
  type          = "ingress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-osb.id
  description    = "allow inbound from sit4-cch-ob-osb."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}

resource "aws_security_group_rule" "sit4-cch-sal-alb_i_10" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-soa.id
  description    = "allow inbound from sit4-cch-ob-soa."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_11" {
  type          = "ingress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-soa.id
  description    = "allow inbound from sit4-cch-ob-soa."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_12" {
  type          = "ingress"

```

```

from_port      = 7001
to_port       = 7001
protocol      = "tcp"
source_security_group_id = aws_security_group.sit4-cch-ob-ums-osb.id
description    = "allow inbound from sit4-cch-ob-ums-osb."
security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_13" {
  type          = "ingress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-osb.id
  description    = "allow inbound from sit4-cch-ob-ums-osb."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_14" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-soa.id
  description    = "allow inbound from sit4-cch-ob-ums-soa."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_15" {
  type          = "ingress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-soa.id
  description    = "allow inbound from sit4-cch-ob-ums-soa."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_16" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-osb.id
  description    = "allow inbound from sit4-cch-ib-sal-osb."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_17" {
  type          = "ingress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-osb.id
  description    = "allow inbound from sit4-cch-ib-sal-osb."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_18" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-soa.id
  description    = "allow inbound from sit4-cch-ib-sal-soa."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_19" {
  type          = "ingress"
  from_port     = 8011

```

```

    to_port          = 8011
    protocol        = "tcp"
    source_security_group_id = aws_security_group.sit4-cch-ib-sal-soa.id
    description      = "allow inbound from sit4-cch-ib-sal-soa."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_20" {
    type          = "ingress"
    from_port     = 7001
    to_port       = 7001
    protocol     = "tcp"
    source_security_group_id = aws_security_group.sit4-cch-portal.id
    description      = "allow inbound from sit4-cch-portal."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_21" {
    type          = "ingress"
    from_port     = 443
    to_port       = 443
    protocol     = "tcp"
    source_security_group_id = aws_security_group.sit4-cch-portal.id
    description      = "allow inbound from sit4-cch-portal."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_22" {
    type          = "ingress"
    from_port     = 7001
    to_port       = 7001
    protocol     = "tcp"
    source_security_group_id = aws_security_group.sit4-pb-osb.id
    description      = "allow inbound from sit4-pb-osb."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_23" {
    type          = "ingress"
    from_port     = 8011
    to_port       = 8011
    protocol     = "tcp"
    source_security_group_id = aws_security_group.sit4-pb-osb.id
    description      = "allow inbound from sit4-pb-osb."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_24" {
    type          = "ingress"
    from_port     = 7001
    to_port       = 7001
    protocol     = "tcp"
    source_security_group_id = aws_security_group.sit4-pb-odi.id
    description      = "allow inbound from sit4-pb-odi."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_24b" {
    type          = "ingress"
    from_port     = 8011
    to_port       = 8011
    protocol     = "tcp"
    source_security_group_id = aws_security_group.sit4-pb-odi.id
    description      = "allow inbound from sit4-pb-odi."
    security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_25" {
    type          = "ingress"
    from_port     = 7001
    to_port       = 7001
    protocol     = "tcp"
}

```

```

source_security_group_id = aws_security_group.sit4-oal-osb.id
description      = "allow inbound from sit4-oal-osb."
security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_26" {
  type          = "ingress"
  from_port    = 8011
  to_port      = 8011
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-oal-osb.id
  description      = "allow inbound from sit4-oal-osb."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_27" {
  type          = "ingress"
  from_port    = 7001
  to_port      = 7001
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-oal-soa.id
  description      = "allow inbound from sit4-oal-soa."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_28" {
  type          = "ingress"
  from_port    = 8011
  to_port      = 8011
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-oal-soa.id
  description      = "allow inbound from sit4-oal-soa."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_29" {
  type          = "ingress"
  from_port    = 7001
  to_port      = 7001
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-web.id
  description      = "allow inbound from sit4-cch-sal-web."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_29b" {
  type          = "ingress"
  from_port    = 7777
  to_port      = 7777
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-web.id
  description      = "allow inbound from sit4-cch-sal-web."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_30" {
  type          = "ingress"
  from_port    = 7001
  to_port      = 7001
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
  description      = "allow inbound from sit4-cch-sal-client-monitoring."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_i_30b" {
  type          = "ingress"
  from_port    = 8011
  to_port      = 8011
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
}

```

```

description      = "allow inbound from sit4-cch-sal-client-monitoring."
security_group_id = aws_security_group.sit4-cch-sal-alb.id
}

resource "aws_security_group_rule" "sit4-cch-sal-alb_eg_1" {
  type          = "egress"
  from_port    = 7777
  to_port      = 7777
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
  description   = "allow outbound to sit4-cch-sal-web-ms."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
resource "aws_security_group_rule" "sit4-cch-sal-alb_eg_2" {
  type          = "egress"
  from_port    = 7001
  to_port      = 7001
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-web-as.id
  description   = "allow outbound to sit4-cch-sal-web-as."
  security_group_id = aws_security_group.sit4-cch-sal-alb.id
}
#sit4-cch-sal-web
resource "aws_security_group" "sit4-cch-sal-web" {
  name          = "sit4-cch-sal-web"
  description   = "Security group for sit4-cch-sal-web"
  vpc_id        = module.test-vpc-4.vpc_id
  tags          = merge(
    local.common_tags,
    {
      "Name"      = "sit4-cch-sal-web"
    },
  )
}

resource "aws_security_group_rule" "sit4-cch-sal-web_i_1" {
  type          = "ingress"
  from_port    = 22
  to_port      = 22
  protocol     = "tcp"
  cidr_blocks  = var.cidr_myst
  description   = "allow inbound from Myst VPC."
  security_group_id = aws_security_group.sit4-cch-sal-web.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web_i_2" {
  type          = "ingress"
  from_port    = 0
  to_port      = 0
  protocol     = "-1"
  self         = true
  security_group_id = aws_security_group.sit4-cch-sal-web.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web_eg_1" {
  type          = "egress"
  from_port    = 0
  to_port      = 0
  protocol     = "-1"
  self         = true
  description   = "allow outbound within SG."
  security_group_id = aws_security_group.sit4-cch-sal-web.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web_eg_2" {
  type          = "egress"
  from_port    = 2049
  to_port      = 2049
  protocol     = "tcp"
}

```

```

source_security_group_id = aws_security_group.sit4-cch-sal-web-efs.id
description      = "allow inbound from sit4-cch-sal-web-efs"
security_group_id = aws_security_group.sit4-cch-sal-web.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web_eg_3" {
  type          = "egress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-rds-oracle-db.id
  description    = "allow inbound from sit4-rds-oracle-db"
  security_group_id = aws_security_group.sit4-cch-sal-web.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web_eg_4" {
  type          = "egress"
  from_port     = 2049
  to_port       = 2049
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-software.id
  description    = "allow inbound from sit4-software"
  security_group_id = aws_security_group.sit4-cch-sal-web.id
}
#sit4-cch-sal-web-as
resource "aws_security_group" "sit4-cch-sal-web-as" {
  name          = "sit4-cch-sal-web-as"
  description   = "Security group for sit4-cch-sal-web-as"
  vpc_id        = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-cch-sal-web-as"
    },
  )
}

resource "aws_security_group_rule" "sit4-cch-sal-web-as_i_1" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-alb.id
  description    = "allow inbound from sit4-cch-sal-alb ."
  security_group_id = aws_security_group.sit4-cch-sal-web-as.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-as_i_2" {
  type          = "ingress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  cidr_blocks   = var.cidr_workspaces
  description    = "allow inbound from Internet VPC (workspaces)."
  security_group_id = aws_security_group.sit4-cch-sal-web-as.id
}
#sit4-cch-sal-web-ms
resource "aws_security_group" "sit4-cch-sal-web-ms" {
  name          = "sit4-cch-sal-web-ms"
  description   = "Security group for sit4-cch-sal-web-ms"
  vpc_id        = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-cch-sal-web-ms"
    },
  )
}

```

```

resource "aws_security_group_rule" "sit4-cch-sal-web-ms_i_1" {
  type          = "ingress"
  from_port     = 7777
  to_port       = 7777
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-alb.id
  description    = "allow inbound from sit4-cch-sal-alb ."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
#DOMAINS
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_7" {
  type          = "egress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-odi.id
  description    = "allow inbound from sit4-cch-ob-odi."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_7b" {
  type          = "egress"
  from_port     = 15101
  to_port       = 15101
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-odi.id
  description    = "allow inbound from sit4-cch-ob-odi."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}

resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_8" {
  type          = "egress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-osb.id
  description    = "allow inbound from sit4-cch-ob-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_9" {
  type          = "egress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-osb.id
  description    = "allow inbound from sit4-cch-ob-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}

resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_10" {
  type          = "egress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-soa.id
  description    = "allow inbound from sit4-cch-ob-soa."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_11" {
  type          = "egress"
  from_port     = 8001
  to_port       = 8001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-soa.id
  description    = "allow inbound from sit4-cch-ob-soa."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}

```

```

resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_12" {
  type          = "egress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-osb.id
  description    = "allow inbound from sit4-cch-ob-ums-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_13" {
  type          = "egress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-osb.id
  description    = "allow inbound from sit4-cch-ob-ums-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_14" {
  type          = "egress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-soa.id
  description    = "allow inbound from sit4-cch-ob-ums-soa."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_15" {
  type          = "egress"
  from_port     = 8001
  to_port       = 8001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-soa.id
  description    = "allow inbound from sit4-cch-ob-ums-soa."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_16" {
  type          = "egress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-osb.id
  description    = "allow inbound from sit4-cch-ib-sal-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_17" {
  type          = "egress"
  from_port     = 8011
  to_port       = 8011
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-osb.id
  description    = "allow inbound from sit4-cch-ib-sal-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_18" {
  type          = "egress"
  from_port     = 7001
  to_port       = 7001
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-soa.id
  description    = "allow inbound from sit4-cch-ib-sal-soa."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_19" {

```

```

type          = "egress"
from_port    = 8001
to_port      = 8001
protocol     = "tcp"
source_security_group_id = aws_security_group.sit4-cch-ib-sal-soa.id
description   = "allow inbound from sit4-cch-ib-sal-soa."
security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_20" {
  type          = "egress"
  from_port    = 7001
  to_port      = 7001
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-portal.id
  description   = "allow inbound from sit4-cch-portal."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_21" {
  type          = "egress"
  from_port    = 7003
  to_port      = 7003
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-portal.id
  description   = "allow inbound from sit4-cch-portal."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_22" {
  type          = "egress"
  from_port    = 7001
  to_port      = 7001
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-osb.id
  description   = "allow inbound from sit4-pb-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_23" {
  type          = "egress"
  from_port    = 8011
  to_port      = 8011
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-osb.id
  description   = "allow inbound from sit4-pb-osb."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_24" {
  type          = "egress"
  from_port    = 7001
  to_port      = 7001
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-odi.id
  description   = "allow inbound from sit4-pb-odi."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_24b" {
  type          = "egress"
  from_port    = 15101
  to_port      = 15101
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-odi.id
  description   = "allow inbound from sit4-pb-odi."
  security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_25" {
  type          = "egress"
  from_port    = 7001

```

```

    to_port          = 7001
    protocol        = "tcp"
    source_security_group_id = aws_security_group.sit4-oal-osb.id
    description      = "allow inbound from sit4-oal-osb."
    security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_26" {
    type            = "egress"
    from_port       = 8011
    to_port         = 8011
    protocol        = "tcp"
    source_security_group_id = aws_security_group.sit4-oal-osb.id
    description      = "allow inbound from sit4-oal-osb."
    security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_27" {
    type            = "egress"
    from_port       = 7001
    to_port         = 7001
    protocol        = "tcp"
    source_security_group_id = aws_security_group.sit4-oal-soa.id
    description      = "allow inbound from sit4-oal-soa."
    security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_28" {
    type            = "egress"
    from_port       = 8001
    to_port         = 8001
    protocol        = "tcp"
    source_security_group_id = aws_security_group.sit4-oal-soa.id
    description      = "allow inbound from sit4-oal-soa."
    security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_29" {
    type            = "egress"
    from_port       = 7001
    to_port         = 7001
    protocol        = "tcp"
    source_security_group_id = aws_security_group.sit4-cch-sal-web.id
    description      = "allow inbound from sit4-cch-sal-web."
    security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}
resource "aws_security_group_rule" "sit4-cch-sal-web-ms_e_29b" {
    type            = "egress"
    from_port       = 7777
    to_port         = 7777
    protocol        = "tcp"
    source_security_group_id = aws_security_group.sit4-cch-sal-web.id
    description      = "allow inbound from sit4-cch-sal-web."
    security_group_id = aws_security_group.sit4-cch-sal-web-ms.id
}

```

8.3.4.4 4.RDS ORACLE SECURITY GROUPS

```
#sit4-rds-oracle-db
resource "aws_security_group" "sit4-rds-oracle-db" {
  name        = "sit4-rds-oracle-db"
  description = "Security group for sit4-rds-oracle-db"
  vpc_id      = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-rds-oracle-db"
    },
  )
}

resource "aws_security_group_rule" "sit4-rds-oracle-db_i_1" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-odi.id
  description    = "allow inbound from sit4-cch-ob-odi."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}

resource "aws_security_group_rule" "sit4-rds-oracle-db_i_2" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-osb.id
  description    = "allow inbound from sit4-cch-ob-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}

resource "aws_security_group_rule" "sit4-rds-oracle-db_i_3" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-soa.id
  description    = "allow inbound from sit4-cch-ob-soa."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}

resource "aws_security_group_rule" "sit4-rds-oracle-db_i_4" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-osb.id
  description    = "allow inbound from sit4-cch-ob-ums-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}

resource "aws_security_group_rule" "sit4-rds-oracle-db_i_5" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-soa.id
  description    = "allow inbound from sit4-cch-ob-ums-soa."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}

resource "aws_security_group_rule" "sit4-rds-oracle-db_i_6" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-osb.id
  description    = "allow inbound from sit4-cch-ib-sal-osb."
}
```

```

    security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_7" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-soa.id
  description    = "allow inbound from sit4-cch-ib-sal-soa."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_8" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-portal.id
  description    = "allow inbound from sit4-cch-portal."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_9" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-osb.id
  description    = "allow inbound from sit4-pb-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_10" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-odi.id
  description    = "allow inbound from sit4-pb-odi."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_11" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-oal-osb.id
  description    = "allow inbound from sit4-oal-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_12" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-oal-soa.id
  description    = "allow inbound from sit4-oal-soa."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_13" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-web.id
  description    = "allow inbound from sit4-cch-sal-web."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_14" {

```

```

type          = "ingress"
from_port    = 1521
to_port      = 1521
protocol     = "tcp"
source_security_group_id = aws_security_group.sit4-cch-sal-ocm.id
description   = "allow inbound from sit4-cch-sal-ocm."
security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_15" {
  type          = "ingress"
  from_port    = 2484
  to_port      = 2484
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-ocm.id
  description   = "allow inbound from sit4-cch-sal-ocm."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_i_16" {
  type          = "ingress"
  from_port    = 1521
  to_port      = 1521
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
  description   = "allow inbound from sit4-cch-sal-client-monitoring."
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_e_1_MML" {
  type          = "egress"
  from_port    = 22102
  to_port      = 22102
  protocol     = "tcp"
  cidr_blocks  = var.cidr_ss_vpc_test
  description   = "allow outbound from haproxy MML"
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_e_2_MML" {
  type          = "egress"
  from_port    = 22103
  to_port      = 22103
  protocol     = "tcp"
  cidr_blocks  = var.cidr_ss_vpc_test
  description   = "allow outbound from haproxy MML"
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-db_e_3_MML" {
  type          = "egress"
  from_port    = 22104
  to_port      = 22104
  protocol     = "tcp"
  cidr_blocks  = var.cidr_ss_vpc_test
  description   = "allow outbound from haproxy MML"
  security_group_id = aws_security_group.sit4-rds-oracle-db.id
}

#rds-oracle-logger-db
resource "aws_security_group" "sit4-rds-oracle-logger-db" {
  name          = "sit4-rds-oracle-logger-db"
  description   = "Security group for sit4-rds-oracle-logger-db"
  vpc_id        = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"    = "sit4-rds-oracle-logger-db"
    },
  )
}

```

```

}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_1" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-osb.id
  description    = "allow inbound from sit4-cch-ob-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_2" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ob-ums-osb.id
  description    = "allow inbound from sit4-cch-ob-ums-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_3" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-ib-sal-osb.id
  description    = "allow inbound from sit4-cch-ib-sal-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_4" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-osb.id
  description    = "allow inbound from sit4-pb-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_5" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-oal-osb.id
  description    = "allow inbound from sit4-oal-osb."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_6" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-portal.id
  description    = "allow inbound from sit4-cch-portal."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_7" {
  type          = "ingress"
  from_port     = 1521
  to_port       = 1521
  protocol      = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-ocm.id
  description    = "allow inbound from sit4-cch-sal-ocm."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_8" {

```

```
type          = "ingress"
from_port    = 2484
to_port      = 2484
protocol     = "tcp"
source_security_group_id = aws_security_group.sit4-cch-sal-ocm.id
description   = "allow inbound from sit4-cch-sal-ocm."
security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
resource "aws_security_group_rule" "sit4-rds-oracle-logger-db_i_9" {
  type          = "ingress"
  from_port    = 1521
  to_port      = 1521
  protocol     = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
  description   = "allow inbound from sit4-cch-sal-client-monitoring."
  security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
}
```

8.3.4.5 5. OCM AND SFTP SECURITY GROUPS

```

#####
# OCM AND SFTP FARGATE SECURITY ROLES
#####
#cch-sal-sftp SecurityGroup
resource "aws_security_group" "sit4-cch-sal-sftp" {
  name        = "sit4-cch-sal-sftp"
  description = "Security group for CCH-SAL-SFTP."
  vpc_id      = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-cch-sal-sftp-tf"
    },
  )
}
resource "aws_security_group_rule" "sit4-cch-sal-sftp_ig" {
  depends_on = [aws_security_group.sit4-cch-sal-sftp]
  type       = "ingress"
  from_port  = 22
  to_port    = 22
  protocol   = "tcp"
  cidr_blocks = var.sit4_web_subnets
  description = "allow ingress traffic from Web Tier"
  security_group_id = aws_security_group.sit4-cch-sal-sftp.id
}
resource "aws_security_group_rule" "sit4-cch-sal-sftp_ig1" {
  depends_on = [aws_security_group.sit4-cch-sal-sftp]
  type       = "ingress"
  from_port  = 22
  to_port    = 22
  protocol   = "tcp"
  cidr_blocks      = var.cidr_workspaces
  description = "allow ingress traffic from Workspaces"
  security_group_id = aws_security_group.sit4-cch-sal-sftp.id
}
resource "aws_security_group_rule" "sit4-cch-sal-sftp_eg_1" {
  type       = "egress"
  from_port  = 2049
  to_port    = 2049
  protocol   = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-sftp-efs.id
  description = "allow egress traffic on 2049 to VPC"
  security_group_id = aws_security_group.sit4-cch-sal-sftp.id
}
resource "aws_security_group_rule" "sit4-cch-sal-sftp_eg_2" {
  type       = "egress"
  from_port  = 443
  to_port    = 443
  protocol   = "tcp"
  cidr_blocks      = ["0.0.0.0/0"]
  description = "allow egress traffic on 433"
  security_group_id = aws_security_group.sit4-cch-sal-sftp.id
}
resource "aws_security_group_rule" "sit4-cch-sal-sftp_eg_3" {
  type       = "egress"
  from_port  = 2049
  to_port    = 2049
  protocol   = "tcp"
  source_security_group_id = aws_security_group.sit4-pb-multidomain-share.id
  description = "allow outbound to sit4-pb-multidomain-share."
  security_group_id = aws_security_group.sit4-cch-sal-sftp.id
}
resource "aws_security_group_rule" "sit4-cch-sal-sftp_eg_4" {
  type       = "egress"
  from_port  = 2049

```

```

        to_port          = 2049
        protocol         = "tcp"
        source_security_group_id = aws_security_group.sit4-cch-ob-multidomain-
share.id
        description       = "allow outbound to sit4-cch-ob-multidomain-share."
        security_group_id = aws_security_group.sit4-cch-sal-sftp.id
    }
#####
#cch-sal-sftp-efs SecurityGroup
resource "aws_security_group" "sit4-cch-sal-sftp-efs" {
    name           = "sit4-cch-sal-sftp-efs"
    description    = "Security group for CCH-SAL-SFTP-EFS."
    vpc_id         = module.test-vpc-4.vpc_id
    tags = merge(
        local.common_tags,
    {
        "Name"      = "sit4-cch-sal-sftp-efs-tf"
    },
)
}
resource "aws_security_group_rule" "sit4-cch-sal-sftp-efs_ig" {
    type      = "ingress"
    from_port = 2049
    to_port   = 2049
    protocol  = "tcp"
    source_security_group_id = aws_security_group.sit4-cch-sal-sftp.id
    description = "allow ingress NFS traffic from cch-sal-sftp"
    security_group_id = aws_security_group.sit4-cch-sal-sftp-efs.id
}

resource "aws_security_group_rule" "sit4-cch-sal-sftp-efs_ig_2" {
    type      = "ingress"
    from_port = 2049
    to_port   = 2049
    protocol  = "tcp"
    source_security_group_id = aws_security_group.sit4-ssm-jumpbox.id
    description = "allow ingress NFS traffic from ssm-jumpbox"
    security_group_id = aws_security_group.sit4-cch-sal-sftp-efs.id
}

#####
# cch-sal-ocm SecurityGroup
resource "aws_security_group" "sit4-cch-sal-ocm" {
    name           = "sit4-cch-sal-ocm"
    description    = "Security group for CCH-SAL-OCM."
    vpc_id         = module.test-vpc-4.vpc_id
    tags = merge(
        local.common_tags,
    {
        "Name"      = "sit4-cch-sal-ocm-tf"
    },
)
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm_ig3" {
    depends_on = [local.sit4_nlb_cidr_blocks]
    type      = "ingress"
    from_port = 1521
    to_port   = 1521
    protocol  = "tcp"
    cidr_blocks     = local.sit4_nlb_cidr_blocks
    description = "allow ingress traffic from NLB"
    security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm_ig4" {
}

```

```

depends_on = [local.sit4_nlb_cidr_blocks]
type      = "ingress"
from_port = 2484
to_port   = 2484
protocol  = "tcp"
cidr_blocks      = local.sit4_nlb_cidr_blocks
description = "allow ingress traffic from NLB"
security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}

resource "aws_security_group_rule" "sit4-cch-sal-ocm_eg_1" {
  type      = "egress"
  from_port = 443
  to_port   = 443
  protocol  = "tcp"
  cidr_blocks      = ["0.0.0.0/0"]
  description = "allow egress traffic on 433"
  security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm_eg_2" {
  type      = "egress"
  from_port = 2049
  to_port   = 2049
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-ocm-efs.id
  description = "allow egress traffic on 2049 to VPC"
  security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm_eg_3" {
  type      = "egress"
  from_port = 1521
  to_port   = 1521
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-rds-oracle-db.id
  description = "allow egress traffic on 1521 to rds oracle"
  security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm_eg_4" {
  type      = "egress"
  from_port = 2484
  to_port   = 2484
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-rds-oracle-db.id
  description = "allow egress traffic on 2484 to rds oracle"
  security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm_eg_3b" {
  type      = "egress"
  from_port = 1521
  to_port   = 1521
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
  description = "allow egress traffic on 1521 to rds-oracle-logger-db"
  security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm_eg_4b" {
  type      = "egress"
  from_port = 2484
  to_port   = 2484
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-rds-oracle-logger-db.id
  description = "allow egress traffic on 2484 to rds-oracle-logger-db"
  security_group_id = aws_security_group.sit4-cch-sal-ocm.id
}

```

```

#####
#cch-sal-ocm-efs SecurityGroup
resource "aws_security_group" "sit4-cch-sal-ocm-efs" {
  name        = "sit4-cch-sal-ocm-efs"
  description = "Security group for CCH-SAL-OCM-EFS."
  vpc_id      = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-cch-sal-ocm-efs-tf"
    },
  )
}

resource "aws_security_group_rule" "sit4-cch-sal-ocm-efs_ig" {
  depends_on = [aws_security_group.sit4-cch-sal-ocm-efs]
  type      = "ingress"
  from_port = 2049
  to_port   = 2049
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-ocm.id
  description = "allow ingress NFS traffic from SIT4 cch-sal-ocm"
  security_group_id = aws_security_group.sit4-cch-sal-ocm-efs.id
}
resource "aws_security_group_rule" "sit4-cch-sal-ocm-efs_ig_2" {
  depends_on = [aws_security_group.sit4-cch-sal-ocm-efs]
  type      = "ingress"
  from_port = 2049
  to_port   = 2049
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-ssm-jumpbox.id
  description = "allow ingress NFS traffic from SIT4 ssm-jumpbox"
  security_group_id = aws_security_group.sit4-cch-sal-ocm-efs.id
}

```

8.3.4.6 6. ssm jumpbox and monitoring

```

#-----ssm-jumpbox-----
resource "aws_security_group" "sit4-ssm-jumpbox" {
  name      = "sit4-ssm-jumpbox"
  description = "Security group for ssm-jumpbox"
  vpc_id = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-ssm-jumpbox"
    },
  )
}

resource "aws_security_group_rule" "sit4-ssm-jumpbox_eg1" {
  type      = "egress"
  from_port = 443
  to_port   = 443
  protocol  = "tcp"
  cidr_blocks      = ["0.0.0.0/0"]
  description = "allow egress traffic on 433"
  security_group_id = aws_security_group.sit4-ssm-jumpbox.id
}
resource "aws_security_group_rule" "sit4-ssm-jumpbox_eg2" {
  type      = "egress"
  from_port = 2049
  to_port   = 2049
  protocol  = "tcp"
  cidr_blocks = var.cidr_sit4
  description = "allow egress traffic on 433"
  security_group_id = aws_security_group.sit4-ssm-jumpbox.id
}

#-----ec2-monitoring-----
resource "aws_security_group" "sit4-ec2-monitoring" {
  name      = "sit4-ec2-monitoring"
  description = "Security group for sit4-ec2-monitoring"
  vpc_id = module.test-vpc-4.vpc_id
  tags = merge(
    local.common_tags,
    {
      "Name"      = "sit4-ec2-monitoring"
    },
  )
}

resource "aws_security_group_rule" "sit4-ec2-monitoring_ig_1" {
  type      = "ingress"
  from_port = 7001
  to_port   = 7001
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
  description = "allow ingress sit4-cch-sal-client-monitoring"
  security_group_id = aws_security_group.sit4-ec2-monitoring.id
}
resource "aws_security_group_rule" "sit4-ec2-monitoring_ig_2" {
  type      = "ingress"
  from_port = 8001
  to_port   = 8001
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
  description = "allow ingress sit4-cch-sal-client-monitoring"
  security_group_id = aws_security_group.sit4-ec2-monitoring.id
}
resource "aws_security_group_rule" "sit4-ec2-monitoring_ig_3" {

```

```

type      = "ingress"
from_port = 8011
to_port   = 8011
protocol  = "tcp"
source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
description = "allow ingress sit4-cch-sal-client-monitoring"
security_group_id = aws_security_group.sit4-ec2-monitoring.id
}
resource "aws_security_group_rule" "sit4-ec2-monitoring_ig_4" {
type      = "ingress"
from_port = 15101
to_port   = 15101
protocol  = "tcp"
source_security_group_id = aws_security_group.sit4-cch-sal-client-
monitoring.id
description = "allow ingress sit4-cch-sal-client-monitoring"
security_group_id = aws_security_group.sit4-ec2-monitoring.id
}

resource "aws_security_group_rule" "sit4-ec2-monitoring_eg1" {
type      = "egress"
from_port = 29001
to_port   = 29001
protocol  = "tcp"
cidr_blocks      = var.cidr_ss_vpc_test
description = "allow egress traffic for ss vpc"
security_group_id = aws_security_group.sit4-ec2-monitoring.id
}
resource "aws_security_group_rule" "sit4-ec2-monitoring_eg2" {
type      = "egress"
from_port = 29003
to_port   = 29003
protocol  = "tcp"
cidr_blocks      = var.cidr_ss_vpc_test
description = "allow egress traffic for ss vpc"
security_group_id = aws_security_group.sit4-ec2-monitoring.id
}

-----
#-----ec2-monitoring-----
resource "aws_security_group" "sit4-cch-sal-client-monitoring" {
name      = "cch-sal-client-monitoring"
description = "Security group for cch-sal-client-monitoringg"
vpc_id = module.test-vpc-4.vpc_id
tags = merge(
  local.common_tags,
  {
    "Name"      = "sit4-cch-sal-client-monitoring"
  },
)
}

resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg1" {
type      = "egress"
from_port = 7001
to_port   = 7001
protocol  = "tcp"
source_security_group_id = aws_security_group.sit4-ec2-monitoring.id
description = "allow egress traffic for ec2-monitoring"
security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg2" {
type      = "egress"
from_port = 8001
to_port   = 8001
protocol  = "tcp"
}

```

```

source_security_group_id = aws_security_group.sit4-ec2-monitoring.id
description = "allow egress traffic for ec2-monitoring"
security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg3" {
  type      = "egress"
  from_port = 8011
  to_port   = 8011
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-ec2-monitoring.id
  description = "allow egress traffic for ec2-monitoring"
  security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg4" {
  type      = "egress"
  from_port = 15101
  to_port   = 15101
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-ec2-monitoring.id
  description = "allow egress traffic for ec2-monitoring"
  security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg5" {
  type      = "egress"
  from_port = 7001
  to_port   = 7001
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-alb.id
  description = "allow egress traffic for cch-sal-alb"
  security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg6" {
  type      = "egress"
  from_port = 8011
  to_port   = 8011
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-cch-sal-alb.id
  description = "allow egress traffic for cch-sal-alb"
  security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg7" {
  type      = "egress"
  from_port = 1521
  to_port   = 1521
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-rds-oracle-db.id
  description = "allow egress traffic for rds-oracle-db"
  security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg8" {
  type      = "egress"
  from_port = 1521
  to_port   = 1521
  protocol  = "tcp"
  source_security_group_id = aws_security_group.sit4-rds-oracle-logger-
db.id
  description = "allow egress traffic for rds-oracle-logger-db"
  security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id
}
resource "aws_security_group_rule" "sit4-cch-sal-client-monitoring_eg9" {
  type      = "egress"
  from_port = 443
  to_port   = 443
  protocol  = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
  description = "allow egress traffic to internet"
}

```

```
    security_group_id = aws_security_group.sit4-cch-sal-client-monitoring.id  
}
```

8.3.5 03 EFS Volumes

This section described the efs volumes created for sit4. They are defined in the file called "4-sit4-efs-volumes.tf".

8.3.5.1 1.Domain EFS volume

Example: CCH_OB_ODI

```
#-----CCH_OB_ODI-----
resource "aws_efs_file_system" "sit4-cch-ob-odi" {
  creation_token = "sit4-cch-ob-odi"
  encrypted      = true
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "EFS file system for sit4-cch-ob-odi"
      "Name"    = "sit4-cch-ob-odi"
    },
  )
}
resource "aws_efs_mount_target" "sit4-cch-ob-odiMountTarget1" {
  file_system_id = "${aws_efs_file_system.sit4-cch-ob-odi.id}"
  subnet_id     = module.test-vpc-4.subnet_database_ids[0]
  security_groups = [aws_security_group.sit4-cch-ob-odi-efs.id]
}
resource "aws_efs_mount_target" "sit4-cch-ob-odiMountTarget2" {
  file_system_id = "${aws_efs_file_system.sit4-cch-ob-odi.id}"
  subnet_id     = module.test-vpc-4.subnet_database_ids[1]
  security_groups = [aws_security_group.sit4-cch-ob-odi-efs.id]
}
```

8.3.5.2 2.multidomain and all domains EFS volumes

```

#-----multidomain PB ODI AND PB OSB-----
-----
resource "aws_efs_file_system" "sit4-pb-multidomain-share" {
  creation_token = "sit4-pb-multidomain-share"
  encrypted      = true
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "EFS file system for sit4-pb-multidomain-share"
      "Name"     = "sit4-pb-multidomain-share"
    },
  )
}
resource "aws_efs_mount_target" "sit4-pb-multidomain-shareMountTarget1" {
  file_system_id = "${aws_efs_file_system.sit4-pb-multidomain-share.id}"
  subnet_id      = module.test-vpc-4.subnet_database_ids[0]
  security_groups = [aws_security_group.sit4-pb-multidomain-share.id]
}
resource "aws_efs_mount_target" "sit4-pb-multidomain-shareMountTarget2" {
  file_system_id = "${aws_efs_file_system.sit4-pb-multidomain-share.id}"
  subnet_id      = module.test-vpc-4.subnet_database_ids[1]
  security_groups = [aws_security_group.sit4-pb-multidomain-share.id]
}
#-----multidomain CCH_OB_ODI AND CCH_OB_OSB-----
-----
resource "aws_efs_file_system" "sit4-cch-ob-multidomain-share" {
  creation_token = "sit4-cch-ob-multidomain-share"
  encrypted      = true
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "EFS file system for sit4-cch-ob-multidomain-share"
      "Name"     = "sit4-cch-ob-multidomain-share"
    },
  )
}
resource "aws_efs_mount_target" "sit4-cch-ob-multidomain-shareMountTarget1" {
  file_system_id = "${aws_efs_file_system.sit4-cch-ob-multidomain-share.id}"
  subnet_id      = module.test-vpc-4.subnet_database_ids[0]
  security_groups = [aws_security_group.sit4-cch-ob-multidomain-share.id]
}
resource "aws_efs_mount_target" "sit4-cch-ob-multidomain-shareMountTarget2" {
  file_system_id = "${aws_efs_file_system.sit4-cch-ob-multidomain-share.id}"
  subnet_id      = module.test-vpc-4.subnet_database_ids[1]
  security_groups = [aws_security_group.sit4-cch-ob-multidomain-share.id]
}

#-----ALL DOMAINS -----
-
resource "aws_efs_file_system" "sit4-software" {
  creation_token = "sit4-software"
  encrypted      = true
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "EFS file system for sit4-software"
      "Name"     = "sit4-software"
    },
  )
}
resource "aws_efs_mount_target" "sit4-softwareMountTarget1" {
  file_system_id = "${aws_efs_file_system.sit4-software.id}"

```

```

    subnet_id      = module.test-vpc-4.subnet_database_ids[0]
    security_groups = [aws_security_group.sit4-software.id]
}
resource "aws_efs_mount_target" "sit4-softwareMountTarget2" {
    file_system_id = "${aws_efs_file_system.sit4-software.id}"
    subnet_id      = module.test-vpc-4.subnet_database_ids[1]
    security_groups = [aws_security_group.sit4-software.id]
}

```

8.3.5.3 3.SFPT and OCM EFS volumes

```

resource "aws_efs_file_system" "sit4-cch-sal-sftp" {
    creation_token = "sit4-cch-sal-sftp"
    encrypted      = true
    tags = merge(
        local.common_tags,
        {
            "Purpose" = "EFS file system for Sandbox SIT4 CCH-SAL-SFTP"
            "Name"     = "sit4-cch-sal-sftp"
        },
    )
}
resource "aws_efs_mount_target" "sit4-cch-sal-sftpEFSMountTarget1" {
    file_system_id = "${aws_efs_file_system.sit4-cch-sal-sftp.id}"
    subnet_id      = module.test-vpc-4.subnet_database_ids[0]
    security_groups = [aws_security_group.sit4-cch-sal-sftp-efs.id]
}
resource "aws_efs_mount_target" "sit4-cch-sal-sftpEFSMountTarget2" {
    file_system_id = "${aws_efs_file_system.sit4-cch-sal-sftp.id}"
    subnet_id      = module.test-vpc-4.subnet_database_ids[1]
    security_groups = [aws_security_group.sit4-cch-sal-sftp-efs.id]
}
#-----CCH-SAL-OCM-----
resource "aws_efs_file_system" "sit4-cch-sal-ocm" {
    creation_token = "sit4-cch-sal-ocm"
    encrypted      = true
    tags = merge(
        local.common_tags,
        {
            "Purpose" = "EFS file system for SIT4 CCH-SAL-OCM"
            "Name"     = "sit4-cch-sal-ocm"
        },
    )
}
resource "aws_efs_mount_target" "sit4-cch-sal-ocmEFSMountTarget1" {
    file_system_id = "${aws_efs_file_system.sit4-cch-sal-ocm.id}"
    subnet_id      = module.test-vpc-4.subnet_database_ids[0]
    security_groups = [aws_security_group.sit4-cch-sal-ocm-efs.id]
}
resource "aws_efs_mount_target" "sit4-cch-sal-ocmEFSMountTarget2" {
    file_system_id = "${aws_efs_file_system.sit4-cch-sal-ocm.id}"
    subnet_id      = module.test-vpc-4.subnet_database_ids[1]
    security_groups = [aws_security_group.sit4-cch-sal-ocm-efs.id]
}

```

8.3.6 04 ALB

1.Application LB and its certificate

```

#####
# APPLICATION LOAD BALANCER
#####
resource "aws_lb" "sit4-cch-alb" {
  name          = "sit4-cch-alb"
  internal      = true
  load_balancer_type = "application"
  security_groups = [aws_security_group.sit4-cch-sal-alb.id]
  subnets        = [module.test-vpc-4.subnet_web_ids[0], module.test-
vpc-4.subnet_web_ids[1]]
  enable_deletion_protection = true
  #access_logs {
    # bucket  = local.logging_bucket["name"]
    # prefix   = "alb/serviceapp"
    # enabled   = true
  }
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "SIT4 ALB CCH"
      "Name"     = "sit4-cch-alb"
    },
  )
}
resource "aws_route53_record" "sit4-cch-alb_record" {
  depends_on = [aws_lb.sit4-cch-alb]
  zone_id   = data.aws_route53_zone.sit4HostedZone.zone_id
  name      = "cch-sal-alb.${var.hosted_zone_sit4}"
  type      = "A"

  alias {
    name          = aws_lb.sit4-cch-alb.dns_name
    zone_id       = aws_lb.sit4-cch-alb.zone_id
    evaluate_target_health = true
  }
}
#####
# CERTIFICATE
#####
#ACM certificate
resource "aws_acm_certificate" "sit4-cch-alb-cert" {
  depends_on = [aws_route53_record.sit4-cch-alb_record]
  domain_name = "sit4-cch-alb.sit4.ieaws.vodafone.com"
  validation_method = "DNS"
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "ACM for SIT4 CCH ALB"
      "Name"     = "SIT4-CCH-ALB"
    },
  )
  lifecycle {
    create_before_destroy = true
  }
}

resource "aws_route53_record" "sit4-cch-alb-cert-validation" {
  depends_on = [aws_acm_certificate.sit4-cch-alb-cert]
  #name      = aws_acm_certificate.sit4-cch-alb-
  cert.domain_validation_options.0.resource_record_name
  #type      = aws_acm_certificate.sit4-cch-alb-
  cert.domain_validation_options.0.resource_record_type
  #zone_id  = data.aws_route53_zone.sit4HostedZone.zone_id
  #records   = [aws_acm_certificate.sit4-cch-alb-
  cert.domain_validation_options.0.resource_record_value]
  #ttl       = 60
}

```

```
for_each = {
    for dvo in aws_acm_certificate.sit4-cch-alb-
cert.domain_validation_options : dvo.domain_name => {
        name    = dvo.resource_record_name
        record  = dvo.resource_record_value
        type    = dvo.resource_record_type
    }
}
allow_overwrite = true
name           = each.value.name
records        = [each.value.record]
ttl            = 60
type           = each.value.type
zone_id        = data.aws_route53_zone.sit4HostedZone.zone_id
}

resource "aws_acm_certificate_validation" "sit4-cch-alb-cert" {
    depends_on = [aws_acm_certificate.sit4-cch-alb-cert]
    certificate_arn      = aws_acm_certificate.sit4-cch-alb-cert.arn
    #validation_record_fqdns = [aws_route53_record.sit4-cch-alb-cert-
validation.fqdn]
    validation_record_fqdns = [for record in aws_route53_record.sit4-cch-
alb-cert-validation : record.fqdn]
}
```

2.Listeners

```

#####
# LISTENER
#####
#listener 8011
resource "aws_lb_listener" "listener_8011" {
  depends_on = [aws_acm_certificate.sit4-cch-alb-cert]
  load_balancer_arn = "${aws_lb.sit4-cch-alb.arn}"
  port          = "8011"
  protocol      = "HTTPS"
  ssl_policy    = "ELBSecurityPolicy-2016-08"
  certificate_arn = aws_acm_certificate_validation.sit4-cch-alb-
cert.certificate_arn
  default_action {
    type          = "fixed-response"

    fixed_response {
      content_type = "text/plain"
      message_body = "Invalid host"
      status_code  = "500"
    }
  }
}

#listener 7001
resource "aws_lb_listener" "listener_7001" {
  depends_on = [aws_acm_certificate.sit4-cch-alb-cert]
  load_balancer_arn = "${aws_lb.sit4-cch-alb.arn}"
  port          = "7001"
  protocol      = "HTTPS"
  ssl_policy    = "ELBSecurityPolicy-2016-08"
  certificate_arn = aws_acm_certificate_validation.sit4-cch-alb-
cert.certificate_arn
  default_action {
    type          = "fixed-response"

    fixed_response {
      content_type = "text/plain"
      message_body = "Invalid host"
      status_code  = "500"
    }
  }
}

#listener 443
resource "aws_lb_listener" "listener_443" {
  depends_on = [aws_acm_certificate.sit4-cch-alb-cert]
  load_balancer_arn = "${aws_lb.sit4-cch-alb.arn}"
  port          = "443"
  protocol      = "HTTPS"
  ssl_policy    = "ELBSecurityPolicy-2016-08"
  certificate_arn = aws_acm_certificate_validation.sit4-cch-alb-
cert.certificate_arn
  default_action {
    type          = "fixed-response"

    fixed_response {
      content_type = "text/plain"
      message_body = "Invalid host"
      status_code  = "500"
    }
  }
}

```

3.Target groups

```

#####
# TARGET GROUPS
#####
#sit4-cch-sal-web-as
resource "aws_lb_target_group" "sit4-cch-sal-web-as" {
    name          = "sit4-cch-sal-web-as"
    port          = 7777
    protocol      = "HTTP"
    target_type   = "instance"
    vpc_id        = module.test-vpc-4.vpc_id

    health_check {
        interval = 30
        path     = "/healthcheck"
        protocol = "HTTP"
        timeout  = 10
        healthy_threshold = 2
        unhealthy_threshold = 10
        matcher   = "200"
    }

    tags = merge(
        local.common_tags,
        {
            "Purpose" = "Target group for sit4-cch-sal-web-as"
        },
    )
}

resource "aws_lb_target_group_attachment" "sit4-cch-sal-web-as-target" {
    count      = var.cch_sal_web_nodes
    target_group_arn = "${aws_lb_target_group.sit4-cch-sal-web-as.arn}"
    target_id    = "${aws_instance.sit4-cch-sal-web[count.index].id}"
}

#sit4-cch-sal-web-ms
resource "aws_lb_target_group" "sit4-cch-sal-web-ms" {
    name          = "sit4-cch-sal-web-ms"
    port          = 7777
    protocol      = "HTTP"
    target_type   = "instance"
    vpc_id        = module.test-vpc-4.vpc_id

    health_check {
        interval = 30
        path     = "/healthcheck"
        protocol = "HTTP"
        timeout  = 10
        healthy_threshold = 2
        unhealthy_threshold = 10
        matcher   = "200"
    }

    tags = merge(
        local.common_tags,
        {
            "Purpose" = "Target group for sit4-cch-sal-web-ms"
        },
    )
}

resource "aws_lb_target_group_attachment" "sit4-cch-sal-web-ms-target-1" {
    count      = var.cch_sal_web_nodes
    target_group_arn = "${aws_lb_target_group.sit4-cch-sal-web-ms.arn}"
    target_id    = "${aws_instance.sit4-cch-sal-web[count.index].id}"
}

```

4.ACM certificate multi domain

```

#-----ACM CERTIFICATE MULTI DOMAIN-----
-----
locals{
    validation_sit4_fqdns_8011 = [for s in aws_route53_record.sit4-
multidomain-8011-cert-cert-record : s.fqdn]
    validation_sit4_fqdns_7001 = [for s in aws_route53_record.sit4-
multidomain-7001-cert-cert-record : s.fqdn]
}

#-----8011-----
resource "aws_acm_certificate" "sit4-multidomain-8011-cert" {
    depends_on = [aws_route53_record.cch-ob-osb, aws_route53_record.cch-ob-
soa, aws_route53_record.cch-ob-ums-osb, aws_route53_record.cch-ob-ums-soa,
aws_route53_record.cch-ib-sal-osb, aws_route53_record.cch-ib-sal-soa,
aws_route53_record.pb-osb, aws_route53_record.oal-osb,
aws_route53_record.oal-soa, aws_route53_record.pb-odi,
aws_route53_record.cch-ob-odi]
    domain_name      = var.sit4_domain_names_acm_8011[0]
    subject_alternative_names = slice(var.sit4_domain_names_acm_8011, 1,
length(var.sit4_domain_names_acm_8011))

    validation_method = "DNS"
    tags = merge(
        local.common_tags,
        {
            "Purpose" = "ACM for multidomains ALB 8011"
            "Name"   = "sit4-multidomain-8011-cert"
        },
    )
    lifecycle {
        ignore_changes = [subject_alternative_names]
    }
}
resource "aws_route53_record" "sit4-multidomain-8011-cert-cert-record" {
    #count    = length(var.sit4_domain_names_acm_8011)
    #name    = lookup(aws_acm_certificate.sit4-multidomain-8011-
cert.domain_validation_options[count.index], "resource_record_name")
    #type    = lookup(aws_acm_certificate.sit4-multidomain-8011-
cert.domain_validation_options[count.index], "resource_record_type")
    #zone_id = data.aws_route53_zone.sit4HostedZone.zone_id
    #records = [lookup(aws_acm_certificate.sit4-multidomain-8011-
cert.domain_validation_options[count.index], "resource_record_value")]
    #ttl     = 60
    for_each = {
        for dvo in aws_acm_certificate.sit4-multidomain-8011-
cert.domain_validation_options : dvo.domain_name => {
            name    = dvo.resource_record_name
            record = dvo.resource_record_value
            type   = dvo.resource_record_type
        }
    }
    allow_overwrite = true
    name          = each.value.name
    records       = [each.value.record]
    ttl           = 60
    type          = each.value.type
    zone_id       = data.aws_route53_zone.sit4HostedZone.zone_id
}

resource "aws_acm_certificate_validation" "sit4-multidomain-8011-cert-
validation" {
    certificate_arn          = aws_acm_certificate.sit4-multidomain-8011-
cert.arn
    validation_record_fqdns = local.validation_sit4_fqdns_8011
}

```

```

}
resource "aws_lb_listener_certificate" "sit4-multidomain-8011-
cert_listenerCert" {
  listener_arn      = aws_lb_listener.listener_8011.arn
  certificate_arn  = aws_acm_certificate_validation.sit4-multidomain-8011-
cert-validation.certificate_arn
}

#-----7001-----
resource "aws_acm_certificate" "sit4-multidomain-7001-cert" {
  depends_on = [aws_route53_record.cch-ob-osb-lb-as,
    aws_route53_record.cch-ob-odi-lb-as, aws_route53_record.cch-ob-soa-lb-as,
    aws_route53_record.cch-ob-ums-osb-lb-as, aws_route53_record.cch-ob-ums-
    soa-lb-as, aws_route53_record.cch-ib-sal-lb-as,
    aws_route53_record.cch-ib-sal-soa-lb-as, aws_route53_record.pb-osb-lb-as,
    aws_route53_record.pb-odi-lb-as, aws_route53_record.oal-osb-lb-as,
    aws_route53_record.oal-soa-lb-as, aws_route53_record.cch-portal-lb-as,
    aws_route53_record.cch-sal-web-lb-as]
  domain_name        = var.sit4_domain_names_acm_7001[0]
  subject_alternative_names = slice(var.sit4_domain_names_acm_7001, 1,
    length(var.sit4_domain_names_acm_7001))

  validation_method = "DNS"
  tags = merge(
    local.common_tags,
    {
      "Purpose" = "ACM for multidomains ALB 7001"
      "Name"     = "sit4-multidomain-7001-cert"
    },
  )
  lifecycle {
    ignore_changes = [subject_alternative_names]
  }
}

resource "aws_route53_record" "sit4-multidomain-7001-cert-cert-record" {
  #count    = length(var.sit4_domain_names_acm_7001)
  #name    = lookup(aws_acm_certificate.sit4-multidomain-7001-
cert.domain_validation_options[count.index], "resource_record_name")
  #type    = lookup(aws_acm_certificate.sit4-multidomain-7001-
cert.domain_validation_options[count.index], "resource_record_type")
  #zone_id = data.aws_route53_zone.sit4HostedZone.zone_id
  #records = [lookup(aws_acm_certificate.sit4-multidomain-7001-
cert.domain_validation_options[count.index], "resource_record_value")]
  #ttl     = 60
  for_each = {
    for dvo in aws_acm_certificate.sit4-multidomain-7001-
cert.domain_validation_options : dvo.domain_name => {
      name    = dvo.resource_record_name
      record  = dvo.resource_record_value
      type    = dvo.resource_record_type
    }
  }
  allow_overwrite = true
  name          = each.value.name
  records       = [each.value.record]
  ttl           = 60
  type          = each.value.type
  zone_id       = data.aws_route53_zone.sit4HostedZone.zone_id
}

resource "aws_acm_certificate_validation" "sit4-multidomain-7001-cert-
validation" {
  certificate_arn      = aws_acm_certificate.sit4-multidomain-7001-
cert.arn
  validation_record_fqdns = local.validation_sit4_fqdns_7001
}

```

```

resource "aws_lb_listener_certificate" "sit4-multidomain-7001-
cert_listenerCert" {
  listener_arn      = aws_lb_listener.listener_7001.arn
  certificate_arn = aws_acm_certificate_validation.sit4-multidomain-7001-
cert-validation.certificate_arn
}

```

5.Listener rule example

```

#####
# LISTENER RULE FOR LISTENER 8011-----
-----
#####

#CCH-OB-OSB
resource "aws_lb_listener_rule" "cch-ob-osb_listener8011" {
  listener_arn = aws_lb_listener.listener_8011.arn
  priority     = 7
  action {
    type          = "forward"
    target_group_arn = aws_lb_target_group.sit4-cch-sal-web-ms.arn
  }
  condition {
    host_header {
      values = ["cch-ob-osb.sit4.ieaws.vodafone.com"]
    }
  }
}
resource "aws_route53_record" "cch-ob-osb" {
  zone_id = data.aws_route53_zone.sit4HostedZone.zone_id
  name    = "cch-ob-osb.${var.hosted_zone_sit4}"
  type    = "A"

  alias {
    name           = aws_lb.sit4-cch-alb.dns_name
    zone_id       = aws_lb.sit4-cch-alb.zone_id
    evaluate_target_health = true
  }
}

```

8.3.7 05 RDS

We will create two rds databases: cch-sal and cch-sal-logger. For the admin password, we will be using a random string.

8.3.7.1 1.CCH-SAL db instance and route53 record

```

resource "random_password" "sit4_master_password_cch_sal" {
  length  = 10
  special = false
}

resource "aws_db_instance" "sit4-cch-sal-rds" {
  identifier          = "sit4-cchsal"
  name                = "cchsal"
  engine               = "oracle-ee"
  #https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_CreateDBInstance.html
  engine_version       = "12.1.0.2.v20"
  instance_class       = "db.m5.large"
  license_model        = "bring-your-own-license"
  character_set_name   = "AL32UTF8"
  multi_az             = true

  username              = "admin"
  password              =
  random_password.sit4_master_password_cch_sal.result
  port                  = "1521"
  allocated_storage     = "700"
  max_allocated_storage = "7168"
  storage_encrypted     = true
  storage_type           = "gp2"

  maintenance_window      = "mon:04:00-mon:05:00"
  backup_window            = "01:00-02:00"
  backup_retention_period = 30
  copy_tags_to_snapshot    = true
  auto_minor_version_upgrade = false
  allow_major_version_upgrade = false
  deletion_protection      = true
  skip_final_snapshot       = true
  delete_automated_backups = false

  enabled_cloudwatch_logs_exports = ["trace", "audit", "alert",
"listener"]
  performance_insights_enabled    = true
  performance_insights_retention_period = 731
  monitoring_interval            = 60
  monitoring_role_arn            = aws_iam_role.sit4-
RDSEnhancedMonitoringRole.arn

  vpc_security_group_ids      = [aws_security_group.sit4-rds-oracle-db.id]
  db_subnet_group_name         = aws_db_subnet_group.sit4-cch-sal-rds.name
  parameter_group_name         = aws_db_parameter_group.sit4-cch-sal-
rds.name
  option_group_name            = aws_db_option_group.sit4-cch-sal-rds.name

  tags                         = merge(
    local.common_tags,
    {
      "Name" = "sit4-cch-sal-rds"
    },
  )

  lifecycle {
    ignore_changes = [identifier, name, enabled_cloudwatch_logs_exports]
  }
}

#ROUTE 53 RECORD
resource "aws_route53_record" "sit4-cch-sal-rds" {
  zone_id = data.aws_route53_zone.sit4HostedZone.zone_id
  name    = "cch-sal-db.${var.hosted_zone_sit4}"
  type    = "CNAME"
}

```

```
    ttl      = "300"
    records = [aws_db_instance.sit4-cch-sal-rds.address]
}
```

8.3.7.2 2.CCH-SAL db parameter group, option group and subnet group

```

resource "aws_db_parameter_group" "sit4-cch-sal-rds" {
  name      = "sit4-cch-sal-parametergroup-oracle-ee"
  description = "sit4-cch-sal-parametergroup-oracle-ee"
  family    = "oracle-ee-12.1"
  tags      = merge(
    local.common_tags,
    {
      "Name" = "sit4-cch-sal-parameter-group"
    },
    )
}
resource "aws_db_option_group" "sit4-cch-sal-rds" {
  name                  = "sit4-cch-sal-optiongroup-oracle-ee"
  engine_name           = "oracle-ee"
  major_engine_version = "12.1"
  tags                  = merge(
    local.common_tags,
    {
      "Name" = "sit4-cch-sal-optiongroup-oracle-
ee"
    },
    )
  option{
    option_name = "Timezone"
    option_settings {
      name   = "TIME_ZONE"
      value  = "Europe/Dublin"
    }
  }
  option{
    option_name = "OEM"
    vpc_security_group_memberships = [aws_security_group.sit4-rds-
oracle-db.id]
    port     = 5500
  }
  option{
    option_name = "SSL"
    vpc_security_group_memberships = [aws_security_group.sit4-rds-
oracle-db.id]
    port     = 2484
    option_settings {
      name   = "FIPS.SSLFIPS_140"
      value  = "FALSE"
    }
    option_settings {
      name   = "SQLNET.CIPHER_SUITE"
      value  = "SSL_RSA_WITH_AES_256_CBC_SHA"
    }
    option_settings {
      name   = "SQLNET.SSL_VERSION"
      value  = "1.2 or 1.0"
    }
  }
}
resource "aws_db_subnet_group" "sit4-cch-sal-rds" {
  name      = "sit4-cch-sal"
  subnet_ids = module.test-vpc-4.subnet_database_ids
  description = "sit4-cch-sal"
  tags      = merge(
    local.common_tags,
    {
      "Name" = "sit4-cch-sal"
    },
    )
}

```

3.CCH-SAL-LOGGER db instance and route53 record

```

#-----LOGGER ORACLE DB
resource "random_password" "sit4-master_password_logger" {
  length  = 10
  special = false
}

resource "aws_db_instance" "sit4-logger-rds" {
  identifier          = "sit4-logger"
  name                = "logger"
  engine              = "oracle-ee"
  #https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_CreateDBInstance.html
  engine_version      = "12.1.0.2.v20"
  instance_class      = "db.m5.large"
  license_model       = "bring-your-own-license"
  character_set_name  = "AL32UTF8"
  multi_az            = true

  username            = "admin"
  password            = random_password.sit4-
  master_password_logger.result
  port                = "1521"
  allocated_storage   = "700"
  max_allocated_storage = "7168"
  storage_encrypted   = true
  storage_type         = "gp2"
  maintenance_window   = "mon:04:00-mon:05:00"
  backup_window        = "01:00-02:00"
  backup_retention_period = 30
  copy_tags_to_snapshot = true
  auto_minor_version_upgrade = false
  allow_major_version_upgrade = false
  deletion_protection    = true
  skip_final_snapshot     = true
  delete_automated_backups = false

  enabled_cloudwatch_logs_exports = ["trace", "audit", "alert",
"listener"]
  performance_insights_enabled    = true
  performance_insights_retention_period = 731
  monitoring_interval           = 60
  monitoring_role_arn           = aws_iam_role.sit4-
RDSEnhancedMonitoringRole.arn

  vpc_security_group_ids      = [aws_security_group.sit4-rds-oracle-
logger-db.id]
  db_subnet_group_name        = aws_db_subnet_group.sit4-logger-rds.name
  parameter_group_name        = aws_db_parameter_group.sit4-logger-
rds.name
  option_group_name           = aws_db_option_group.sit4-logger-rds.name

  tags                         = merge(
    local.common_tags,
    {
      "Name" = "sit4-cch-sal-rds"
    },
  )
}

lifecycle {
  ignore_changes = [identifier, name, enabled_cloudwatch_logs_exports]
}
}

#ROUTE 53 RECORD
resource "aws_route53_record" "sit4-logger-rds" {
  zone_id = data.aws_route53_zone.sit4HostedZone.zone_id
  name    = "logger-db.${var.hosted_zone_sit4}"
}

```

```
type      = "CNAME"
ttl       = "300"
records  = [aws_db_instance.sit4-logger-rds.address]
}
```

4.CCH-SAL-LOGGER db parameter group, option group and subnet group

```

resource "aws_db_parameter_group" "sit4-logger-rds" {
  name      = "sit4-logger-parametergroup-oracle-ee"
  description = "sit4-logger-parametergroup-oracle-ee"
  family    = "oracle-ee-12.1"
  tags      = merge(
    local.common_tags,
    {
      "Name" = "sit4-logger-parameter-group"
    },
  )
}
resource "aws_db_option_group" "sit4-logger-rds" {
  name                  = "sit4-logger-optiongroup-oracle-ee"
  engine_name           = "oracle-ee"
  major_engine_version = "12.1"
  tags      = merge(
    local.common_tags,
    {
      "Name" = "sit4-logger-optiongroup-oracle-ee"
    },
  )
option{
  option_name = "Timezone"
  option_settings {
    name = "TIME_ZONE"
    value = "Europe/Dublin"
  }
}
option{
  option_name = "OEM"
  vpc_security_group_memberships = [aws_security_group.sit4-rds-oracle-db.id]
  port = 5500
}
option{
  option_name = "SSL"
  vpc_security_group_memberships = [aws_security_group.sit4-rds-oracle-db.id]
  port = 2484
  option_settings {
    name = "FIPS.SSLFIPS_140"
    value = "FALSE"
  }
  option_settings {
    name = "SQLNET.CIPHER_SUITE"
    value = "SSL_RSA_WITH_AES_256_CBC_SHA"
  }
  option_settings {
    name = "SQLNET.SSL_VERSION"
    value = "1.2 or 1.0"
  }
}
}
resource "aws_db_subnet_group" "sit4-logger-rds" {
  name      = "sit4-logger"
  description = "sit4-logger"
  subnet_ids = module.test-vpc-4.subnet_database_ids
  tags      = merge(
    local.common_tags,
    {
      "Name" = "sit4-logger"
    },
  )
}

```

9 03 VF IE PRE-PROD VPCs

9.1 1. Introduction

- Code: Terraform
- Repository: <https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/vf-iedelivery-infrastructure>
- Branch: pre-prod
- AWS account: vf-iedelivery-mgmt - 831341508773
- CI/CD: CodePipeline <https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-infrastructure-preprod-tf/view?region=eu-west-1>

Each VPC will be composed of:

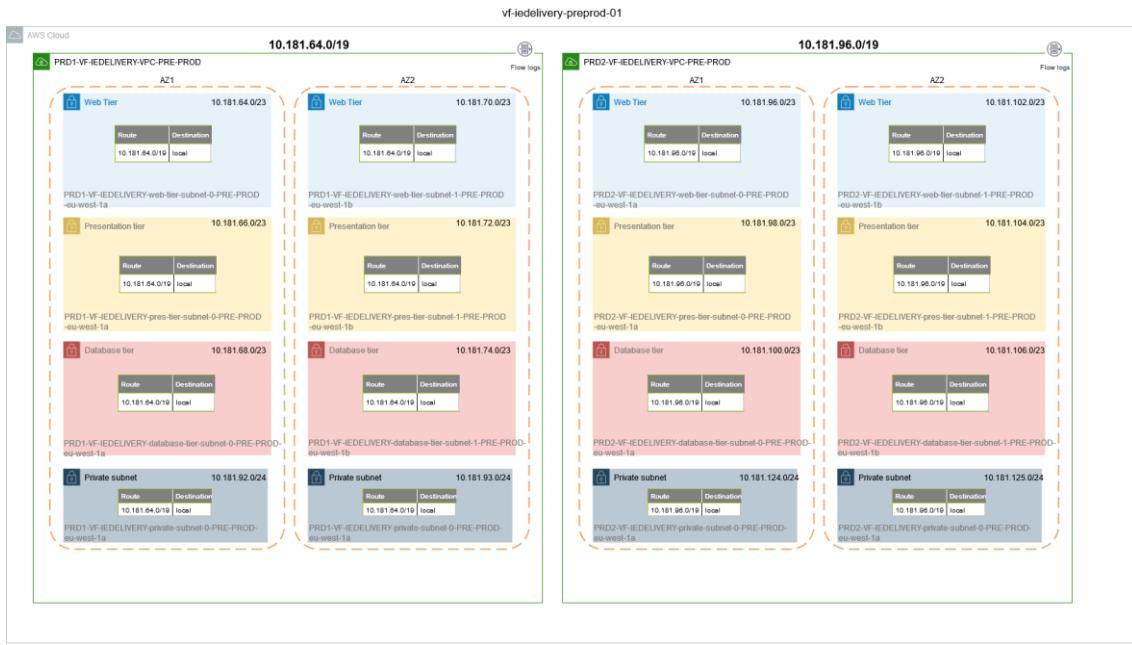
- 1 Web tier subnet per AZ. Subnet IP range: '/23'
- 1 Presentation tier subnet per AZ. Subnet IP range: '/23'
- 1 Database tier subnet per AZ. Subnet IP range: '/23'
- 1 Private subnet for other purposes per AZ. Subnet IP range: '/24'
- VPC Flow logs enabled
- 1 different route table for each subnet.

9.2 2. Design of PRE-PROD VPC's

These are the subnet CIDR ranges that were chosen for pre-prod environment:

| VPC | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|------|----------------|---------------|------------------------------|------------------------------|-------|
| PRD1 | 10.181.64.0/19 | 255.255.224.0 | 10.181.64.0 - 10.181.95.255 | 10.181.64.1 - 10.181.95.254 | 8190 |
| PRD2 | 10.181.96.0/19 | 255.255.224.0 | 10.181.96.0 - 10.181.127.255 | 10.181.96.1 - 10.181.127.254 | 8190 |

The following diagram shows the state for the infrastructure deployed in AWS VF IE pre-prod account by PCS. In each VPC, 3 subnets per AZ were created: web tier, presentation tier and database tier. Additionally, there are also others 2 private subnets that can be used for other purposes (endpoints, ENI, etc.).



9.3 3. Subnet CIDR range

9.3.1 PRD1

| Subnet | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|----------|----------------|---------------|-----------------------------|-----------------------------|-------|
| Web | 10.181.64.0/23 | 255.255.254.0 | 10.181.64.0 - 10.181.65.255 | 10.181.64.1 - 10.181.65.254 | 510 |
| Pres | 10.181.66.0/23 | 255.255.254.0 | 10.181.66.0 - 10.181.67.255 | 10.181.66.1 - 10.181.67.254 | 510 |
| Database | 10.181.68.0/23 | 255.255.254.0 | 10.181.68.0 - 10.181.69.255 | 10.181.68.1 - 10.181.69.254 | 510 |
| Web | 10.181.70.0/23 | 255.255.254.0 | 10.181.70.0 - 10.181.71.255 | 10.181.70.1 - 10.181.71.254 | 510 |
| Pres | 10.181.72.0/23 | 255.255.254.0 | 10.181.72.0 - 10.181.73.255 | 10.181.72.1 - 10.181.73.254 | 510 |
| Database | 10.181.74.0/23 | 255.255.254.0 | 10.181.74.0 - 10.181.75.255 | 10.181.74.1 - 10.181.75.254 | 510 |
| | 10.181.76.0/23 | 255.255.254.0 | 10.181.76.0 - 10.181.77.255 | 10.181.76.1 - 10.181.77.254 | 510 |
| | 10.181.78.0/23 | 255.255.254.0 | 10.181.78.0 - 10.181.79.255 | 10.181.78.1 - 10.181.79.254 | 510 |
| | 10.181.80.0/23 | 255.255.254.0 | 10.181.80.0 - 10.181.81.255 | 10.181.80.1 - 10.181.81.254 | 510 |

| | | | | | |
|------|----------------|---------------|-----------------------------|-----------------------------|-----|
| | 10.181.82.0/23 | 255.255.254.0 | 10.181.82.0 - 10.181.83.255 | 10.181.82.1 - 10.181.83.254 | 510 |
| | 10.181.84.0/23 | 255.255.254.0 | 10.181.84.0 - 10.181.85.255 | 10.181.84.1 - 10.181.85.254 | 510 |
| | 10.181.86.0/23 | 255.255.254.0 | 10.181.86.0 - 10.181.87.255 | 10.181.86.1 - 10.181.87.254 | 510 |
| | 10.181.88.0/23 | 255.255.254.0 | 10.181.88.0 - 10.181.89.255 | 10.181.88.1 - 10.181.89.254 | 510 |
| | 10.181.90.0/23 | 255.255.254.0 | 10.181.90.0 - 10.181.91.255 | 10.181.90.1 - 10.181.91.254 | 510 |
| Priv | 10.181.92.0/24 | 255.255.255.0 | 10.181.92.0 - 10.181.92.255 | 10.181.92.1 - 10.181.92.254 | 254 |
| Priv | 10.181.93.0/24 | 255.255.255.0 | 10.181.93.0 - 10.181.93.255 | 10.181.93.1 - 10.181.93.254 | 254 |
| | 10.181.94.0/24 | 255.255.255.0 | 10.181.94.0 - 10.181.94.255 | 10.181.94.1 - 10.181.94.254 | 254 |
| | 10.181.95.0/24 | 255.255.255.0 | 10.181.95.0 - 10.181.95.255 | 10.181.95.1 - 10.181.95.254 | 254 |

9.3.2 PRD2

| Subnet | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|--------|-----------------|---------------|-------------------------------|-------------------------------|-------|
| Web | 10.181.96.0/23 | 255.255.254.0 | 10.181.96.0 - 10.181.97.255 | 10.181.96.1 - 10.181.97.254 | 510 |
| Pres | 10.181.98.0/23 | 255.255.254.0 | 10.181.98.0 - 10.181.99.255 | 10.181.98.1 - 10.181.99.254 | 510 |
| Datab | 10.181.100.0/23 | 255.255.254.0 | 10.181.100.0 - 10.181.101.255 | 10.181.100.1 - 10.181.101.254 | 510 |
| Web | 10.181.102.0/23 | 255.255.254.0 | 10.181.102.0 - 10.181.103.255 | 10.181.102.1 - 10.181.103.254 | 510 |
| Pres | 10.181.104.0/23 | 255.255.254.0 | 10.181.104.0 - 10.181.105.255 | 10.181.104.1 - 10.181.105.254 | 510 |
| Datab | 10.181.106.0/23 | 255.255.254.0 | 10.181.106.0 - 10.181.107.255 | 10.181.106.1 - 10.181.107.254 | 510 |
| | 10.181.108.0/23 | 255.255.254.0 | 10.181.108.0 - 10.181.109.255 | 10.181.108.1 - 10.181.109.254 | 510 |
| | 10.181.110.0/23 | 255.255.254.0 | 10.181.110.0 - 10.181.111.255 | 10.181.110.1 - 10.181.111.254 | 510 |
| | 10.181.112.0/23 | 255.255.254.0 | 10.181.112.0 - 10.181.113.255 | 10.181.112.1 - 10.181.113.254 | 510 |

| | | | | | |
|------|-----------------|---------------|----------------------------------|----------------------------------|-----|
| | 10.181.114.0/23 | 255.255.254.0 | 10.181.114.0 - 10.181.115.255 | 10.181.114.1 - 10.181.115.254 | 510 |
| | 10.181.116.0/23 | 255.255.254.0 | 10.181.116.0 - 10.181.117.255 | 10.181.116.1 - 10.181.117.254 | 510 |
| | 10.181.118.0/23 | 255.255.254.0 | 10.181.118.0 - 10.181.119.255 | 10.181.118.1 - 10.181.119.254 | 510 |
| | 10.181.120.0/23 | 255.255.254.0 | 10.181.120.0 - 10.181.121.255 | 10.181.120.1 - 10.181.121.254 | 510 |
| | 10.181.122.0/23 | 255.255.254.0 | 10.181.122.0 - 10.181.123.255 | 10.181.122.1 - 10.181.123.254 | 510 |
| Priv | 10.181.124.0/24 | 255.255.255.0 | 10.181.124.0 - 10.181.124.255 | 10.181.124.1 - 10.181.124.254 | 254 |
| Priv | 10.181.125.0/24 | 255.255.255.0 | 10.181.125.0 - 10.181.125.255 | 10.181.125.1 - 10.181.125.254 | 254 |
| | 10.181.126.0/24 | 255.255.255.0 | 10.181.126.0 - 10.181.126.255 | 10.181.126.1 - 10.181.126.254 | 254 |
| | 10.181.127.0/24 | 255.255.255.0 | 10.181.127.0 - 10.181.127.255 | 10.181.127.1 - 10.181.127.254 | 254 |

10 04 VF IE DNS

10.1

- [1.Route53 Hosted Zones](#)
- [2.AWS TO GDC](#)
- [3.GDC TO AWS](#)
- [4.TGW](#)
- [5.SS VPC](#)
- [6. AWS Address Resolution](#)

10.21.Route53 Hosted Zones

This section describes the public and private hosted zones that are going to be created for VF IE. In order to maintain the IaC centralized, a codecommit repository has been created: <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-dns/browse?region=eu-west-1>. To automate the changes, a cross-account codepipeline has been created: <https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-dns--cross-account-prod-tf/view?region=eu-west-1>

| Environment | AWS Tenant | Domain | Type | Account | VPC | Purpose | Created ? |
|-----------------|------------|------------------------------------|---------|--------------------------|--------|---|------------------------------|
| Shared Services | SS | ieaws.vodafone.com | Public | vf-iedelivery-prod-ss-01 | | Public hosted zone for VFIE project. Infoblox public zone has been delegated to this zone. | Yes |
| Shared Services | SS | ieaws.vodafone.com | Private | vf-iedelivery-prod-ss-01 | SS VPC | Private hosted zone for VFIE project. Infoblox private zone has been delegated to this zone. | Yes, delegation in progress. |
| Shared Services | SS | ? | Public | vf-iedelivery-prod-ss-01 | | Shared Services CI/CD | No |
| Sandbox | sdn1 | sdn1.ieaws.vodafone.co m | Public | vf-iedelivery-sandbox-01 | | Sub-domain of VFIE for sandbox. Workload Sandbox | Yes |

| Environment | AWS Tenant | Domain | Type | Account | VPC | Purpose | Created? |
|-------------|------------|--|---------|---------------------------|---------|--|----------|
| Sandbox | sdn1-ss | sdn1-ss.ieaws.vodafone.com | Public | vf-iedeliver-y-sandbox-01 | | Sub-domain of VFIE for sandbox. Shared Services Sandbox | Yes |
| TEST | sit1 | sit1.ieaws.vodafone.com | Public | vf-iedeliver-y-test-01 | | Sub-domain of VFIE for test. | No |
| TEST | sit1 | internal.vodafone.com | Private | vf-iedeliver-y-test-01 | ST1 VPC | Routing for AWS TEST to GDC connections. We will add records with a wildcard for prod gcd services, pointing to SS TEST NLB. | No |
| TEST | sit2 | sit2.ieaws.vodafone.com | Public | vf-iedeliver-y-test-01 | | Sub-domain of VFIE for test. | No |
| TEST | sit2 | internal.vodafone.com | Private | vf-iedeliver-y-test-01 | ST2 VPC | Routing for AWS TEST to GDC connections. We will add records with a wildcard for prod gcd services, pointing to SS TEST NLB. | No |
| TEST | sit3 | sit3.ieaws.vodafone.com | Public | vf-iedeliver-y-test-01 | | Sub-domain of VFIE for test. | No |
| TEST | sit3 | internal.vodafone.com | Private | vf-iedeliver-y-test-01 | ST3 VPC | Routing for AWS TEST to GDC connections. We will add records with | No |

| Environment | AWS Tenant | Domain | Type | Account | VPC | Purpose | Created? |
|-------------|------------|--|---------|---------------------------|----------|--|----------|
| | | | | | | a wildcard for prod gcd services, pointing to SS TEST NLB. | |
| TEST | sit4 | sit4.ieaws.vodafone.com | Public | vf-iedeliver-y-test-01 | | Sub-domain of VFIE for test. | No |
| TEST | sit4 | internal.vodafone.com | Private | vf-iedeliver-y-test-01 | ST4 VPC | Routing for AWS TEST to GDC connections. We will add records with a wildcard for prod gcd services, pointing to SS TEST NLB. | No |
| PRE-PROD | prd1 | prd1. ieaws.vodafone.com | Public | vf-iedeliver-y-preprod-01 | | Sub-domain of VFIE for pre-prod. | Yes |
| PRE-PROD | prd1 | internal.vodafone.com | Private | vf-iedeliver-y-preprod-01 | PRD1 VPC | Routing for AWS PREPROD to GDC connections. We will add records with a wildcard for prod gcd services, pointing to SS PREPROD NLB. | Yes |
| PRE-PROD | prd2 | prd2. ieaws.vodafone.com | Public | vf-iedeliver-y- | | Sub-domain of VFIE for pre-prod | Yes |

| Environment | AWS Tenant | Domain | Type | Account | VPC | Purpose | Created? |
|-------------|------------|--|---------|---------------------------|----------|--|----------|
| | | | | preprod-01 | | | |
| PRE-PROD | prd2 | internal.vodafone.com | Private | vf-iedeliver-y-preprod-01 | PRD2 VPC | Routing for AWS PREPROD to GDC connections. We will add records with a wildcard for prod gcd services, pointing to SS PREPROD NLB. | Yes |
| PROD | prod | prod.ieaws.vodafone.com | Public | vf-iedeliver-y-prod-01 | | Sub-domain of VFIE for prod | Yes |
| PROD | prod | internal.vodafone.com | Private | vf-iedeliver-y-prod-01 | Prod VPC | Routing for AWS PROD to GDC connections. We will add records with a wildcard for prod gcd services, pointing to SS PROD NLB. | Yes |

10.32.AWS TO GDC

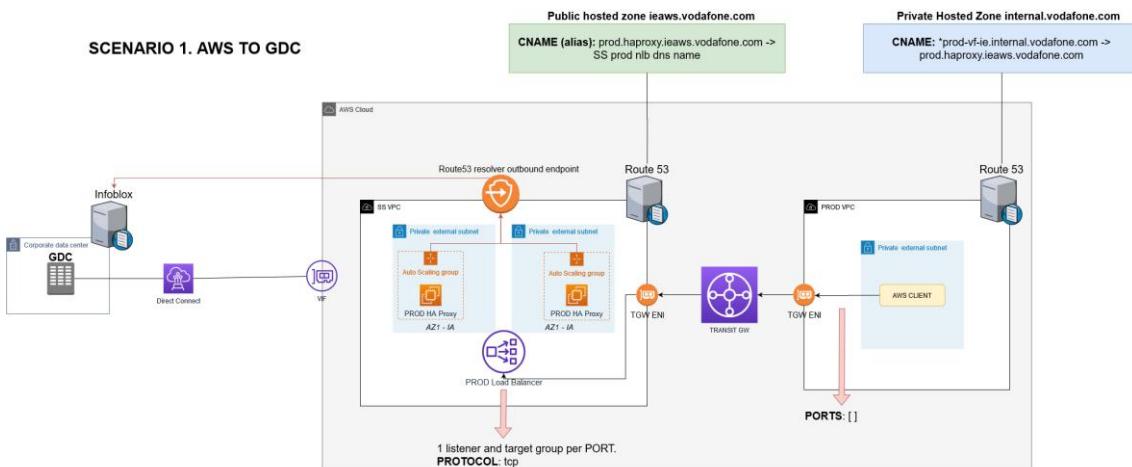
This section will provide the details needed to understand the communication between a client in AWS tenant and a server in GDC datacenter. The diagram just shows the case for AWS Prod tenant, but the same pattern will be followed for other AWS environments.

In this scenario, the client is on the right side, in AWS Prod VPC (prod account). There will be a list of ports where the HAProxy will be listening to (the list can be updated during the build phase). The client will try to connect with a gcd service, with a name convention like 'prod-vf-ie.internal.vodafone.com'. This is a list with some examples:

- FSL-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- FSL-GTW-PROD-VF-IE.INTERNAL.VODAFONE.COM
- MEH-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM

- PORTAL-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- OSB-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- MFT-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- MEDIAROOM-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- GISMSH-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- TIBCO-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- SMSC-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- RADIUS-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- MML-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- AUA-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
- NPT-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM

Prod VPC has a private hosted zone, internal.vodafone.com, where a cname record, pointing '*prod-vf-ie.internal.vodafone.com' (wildcard) to the SS PROD NLB. In this record an alias name of the SS PROD NLB is used, in order to avoid problems if the NLB changed in the SS VPC. This alias name for the SS NLB will be maintained. The AWS client will send the traffic to SS PROD NLB, that will redirect the request to an HAProxy in SS VPC. Then, HAProxy can have in its config file either a gcd ip address, or a gcd server domain name. If it is a gcd server domain name, the Route53 resolver outbound endpoint will be used. In the configuration for the outbound endpoint, a Forward Rule has been created, with the Infoblox name server ip address, to send the dns queries of internal.vodafone.com. For the case of an ip address, we don't need further configuration, as once VIF is established, BGP propagate the corresponding routes to SS VPC.

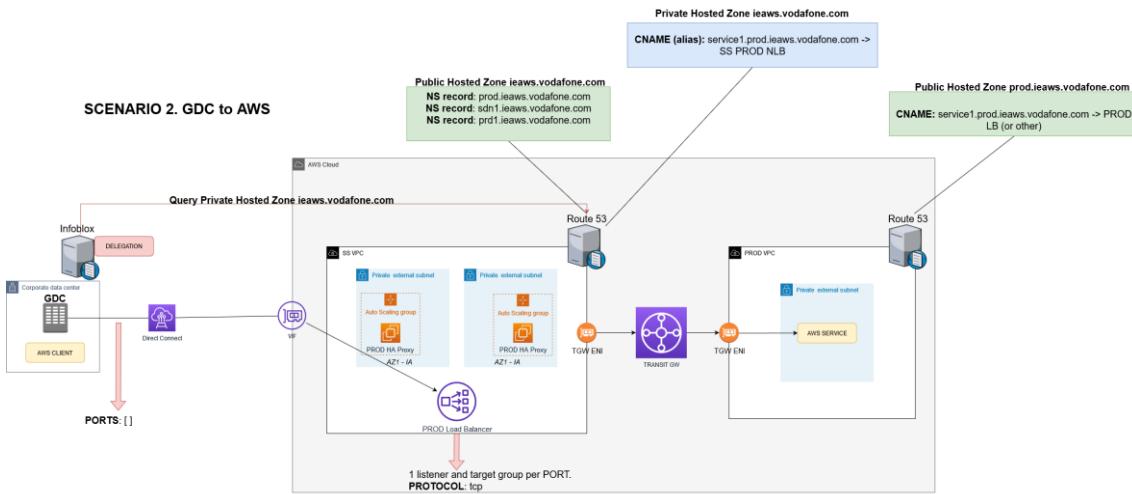


10.43.GDC TO AWS

This section will provide the details needed to understand the communication between a client in GDC datacenter and a server in AWS tenant. The diagram just shows the case for AWS Prod tenant, but the same pattern will be followed for other AWS environments.

First of all, the private zone in Infoblox has a delegation of 'ieaws.vodafone.com' that points to the Private Hosted Zone in SS VPC. In this Private hosted zone, we will be populating the services that we offer in AWS, as this will be the visibility that GDC will have about AWS. We

will add alias records, pointing the published service, for example, service1.prod.ieaws.vodafone.com, to the SS PROD NLB. So, when a client in GDC wants to connect to a server in AWS, it will query the private zone in Infoblox, and then it will get as a response the SS PROD NLB. Once the request reaches SS PROD NLB and it's redirected to an HAProxy, the HAProxy will use the Public Hosted zone for 'ieaws.vodafone.com' to resolve the service1 published in PROD. In PROD VPC, a subdomain called 'prod.ieaws.vodafone.com' has been created, and the corresponding NS records have been added to the public hosted zone 'ieaws.vodafone.com' in SS VPC. In this subdomain, we will add the CNAME for the services published, with the real resolution, for example, a LB in prod. This will be the response that the HAProxy will get.



10.54.TGW

All communications are going to depend on the routing of the TGW. Please, see the documentation and implementation [here](#).

10.65.SS VPC

Please, see the documentation about the SS VPC [here](#).

10.76. AWS Address Resolution

| FQDN | CNAME | LB Address |
|--|---|--|
| pb-osb-lb-as.prod.ieaws.vodafone.com | prod.ieaws.vodafone.com | SS-VF-IEDELIVERY-PROD-NLB-07ed30b5f822b162.elb.eu-west-1.amazonaws.com |
| pb-osb.prod.ieaws.vodafone.com | | |
| pb-odi-lb-as.prod.ieaws.vodafone.com | | |
| cch-ib-osb-lb-as.prod.ieaws.vodafone.com | | |

| | | |
|---|--|--|
| cch-ib- osb.prod.ieaws.vodafone.com | | |
| cch-ib-soa-lb- as.prod.ieaws.vodafone.com | | |
| cch-ib- soa.prod.ieaws.vodafone.com | | |
| cch-ob-ums-osb-lb- as.prod.ieaws.vodafone.com | | |
| cch-ob-ums- osb.prod.ieaws.vodafone.com | | |
| cch-ob-ums-soa-lb- as.prod.ieaws.vodafone.com | | |
| cch-ob-ums- soa.prod.ieaws.vodafone.com | | |
| cch-ob-osb-lb- as.prod.ieaws.vodafone.com | | |
| cch-ob- osb.prod.ieaws.vodafone.com | | |
| cch-ob-soa-lb- as.prod.ieaws.vodafone.com | | |
| cch-ob- soa.prod.ieaws.vodafone.com | | |
| cch-ob-odi-lb- as.prod.ieaws.vodafone.com | | |
| cch-portal-lb- as.prod.ieaws.vodafone.com | | |
| cch- portal.prod.ieaws.vodafone.com | | |
| oal-osb-lb- as.prod.ieaws.vodafone.com | | |
| oal-osb.prod.ieaws.vodafone.com | | |
| oal-soa-lb- as.prod.ieaws.vodafone.com | | |
| oal-soa.prod.ieaws.vodafone.com | | |
| cch-sal- sftp.prd1.ieaws.vodafone.com | | |
| jenkins.ieaws.vodafone.com | | |
| artifactory.ieaws.vodafone.com | | |
| myst.ieaws.vodafone.com | | |
| ocm.ieaws.vodafone.com | | |

| | | |
|--|---|--|
| pb-osb-lb-as.prd2.ieaws.vodafone.com | prd2.ieaws.vodafone.com | SS-VF-IEDELIVERY-PRD2-NLB-0896a285b4124ae4.elb.eu-west-1.amazonaws.com |
| pb-osb.prd2.ieaws.vodafone.com | | |
| pb-odi-lb-as.prd2.ieaws.vodafone.com | | |
| cch-ib-osb-lb-as.prd2.ieaws.vodafone.com | | |
| cch-ib-osb.prd2.ieaws.vodafone.com | | |
| cch-ib-soa-lb-as.prd2.ieaws.vodafone.com | | |
| cch-ib-soa.prd2.ieaws.vodafone.com | | |
| cch-ob-ums-osb-lb-as.prd2.ieaws.vodafone.com | | |
| cch-ob-ums-osb.prd2.ieaws.vodafone.com | | |
| cch-ob-ums-soa-lb-as.prd2.ieaws.vodafone.com | | |
| cch-ob-ums-soa.prd2.ieaws.vodafone.com | | |
| cch-ob-osb-lb-as.prd2.ieaws.vodafone.com | | |
| cch-ob-osb.prd2.ieaws.vodafone.com | | |
| cch-ob-soa-lb-as.prd2.ieaws.vodafone.com | | |
| cch-ob-soa.prd2.ieaws.vodafone.com | | |
| cch-ob-odi-lb-as.prd2.ieaws.vodafone.com | | |
| cch-portal-lb-as.prd2.ieaws.vodafone.com | | |
| cch-portal.prd2.ieaws.vodafone.com | | |
| oal-osb-lb-as.prd2.ieaws.vodafone.com | | |
| oal-osb.prd2.ieaws.vodafone.com | | |
| oal-soa-lb-as.prd2.ieaws.vodafone.com | | |
| oal-soa.prd2.ieaws.vodafone.com | | |

| | | |
|--|---|--|
| cch-sal-sftp.prd2.ieaws.vodafone.com | | |
| pb-osb-lb-as.prd1.ieaws.vodafone.com | prd1.ieaws.vodafone.com | SS-VF-IEDELIVERY-PRD1-NLB-fff2438b7fa415c1.elb.eu-west-1.amazonaws.com |
| pb-osb.prd1.ieaws.vodafone.com | | |
| pb-odi-lb-as.prd1.ieaws.vodafone.com | | |
| cch-ib-osb-lb-as.prd1.ieaws.vodafone.com | | |
| cch-ib-osb.prd1.ieaws.vodafone.com | | |
| cch-ib-soa-lb-as.prd1.ieaws.vodafone.com | | |
| cch-ib-soa.prd1.ieaws.vodafone.com | | |
| cch-ob-ums-osb-lb-as.prd1.ieaws.vodafone.com | | |
| cch-ob-ums-osb.prd1.ieaws.vodafone.com | | |
| cch-ob-ums-soa-lb-as.prd1.ieaws.vodafone.com | | |
| cch-ob-ums-soa.prd1.ieaws.vodafone.com | | |
| cch-ob-osb-lb-as.prd1.ieaws.vodafone.com | | |
| cch-ob-osb.prd1.ieaws.vodafone.com | | |
| cch-ob-soa-lb-as.prd1.ieaws.vodafone.com | | |
| cch-ob-soa.prd1.ieaws.vodafone.com | | |
| cch-ob-odi-lb-as.prd1.ieaws.vodafone.com | | |
| cch-portal-lb-as.prd1.ieaws.vodafone.com | | |
| cch-portal.prd1.ieaws.vodafone.com | | |
| oal-osb-lb-as.prd1.ieaws.vodafone.com | | |
| oal-osb.prd1.ieaws.vodafone.com | | |
| oal-soa-lb-as.prd1.ieaws.vodafone.com | | |

| | | |
|--|--|--|
| oal-soa.prd1.ieaws.vodafone.com | | |
| cch-sal-sftp.prd1.ieaws.vodafone.com | | |

10.8 DNS TROUBLESHOOTING

Steps to follow to see the traffic and the routing logic to troubleshoot dns.

10.8.1 POINT 1: GDC domain name resolution

We are using a Route53 resolver outbound endpoint to send gcd queries to Infoblox.

IP addresses used in AWS side to send queries:

| IP addresses (2) | | | | | Remove from endpoint | Add IP address | | |
|-------------------------------|-----------------------|---|-------------------------|-------------------|--------------------------------------|--------------------------------|----------------------|-------------------|
| | | | | | < | 1 | > | @ |
| IP address | IP address ID | Status | Subnet | Availability Zone | | | | |
| 198.19.220.70 | rni-e8a1a04988cf4d889 | Attached | subnet-07f6874db697c... | eu-west-1b | | | | |
| 198.19.220.30 | rni-ab8e2e9cf74f4f3eb | Attached | subnet-088022fa29ac6... | eu-west-1a | | | | |

Rules created:

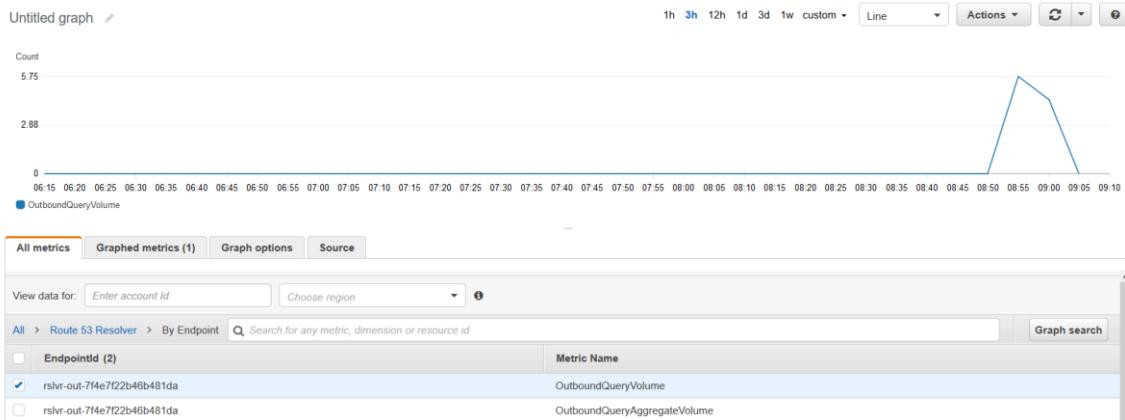
| Rules (2) | | | | | | | Create rule | | | |
|--|------------------|---|-----------------------------|---------|------------------------|---------------------|-----------------------------|-------------------|----------------------|-------------------|
| | | | | | | | < | 1 | > | @ |
| Name | ID | Status | Outbound endpoint | Type | Domain name | Target IP addresses | | | | |
| dc-dublin-de-rule | rslvr-rr-3556... | Complete | rslvr-out-7f4e7f22b46b481da | Forward | dc-dublin.de. | 1 | | | | |
| internal-vodafone-com-rule | rslvr-rr-9046... | Complete | rslvr-out-7f4e7f22b46b481da | Forward | internal.vodafone.com. | 1 | | | | |

GDC Target IP address (Infoblox dns server):

| Target IP addresses (1) | | Delete | Add target IP address | | |
|-------------------------------|------|------------------------|---------------------------------------|----------------------|-------------------|
| | | < | 1 | > | @ |
| IP address | Port | | | | |
| 47.73.122.145 | 53 | | | | |

10.8.1.1 RESOLVER METRICS

We can see the Route53 Resolver metrics here: [https://eu-west-1.console.aws.amazon.com/cloudwatch/home?region=eu-west-1#metricsV2:graph=~\(view~timeSeries~stacked~false~metrics~\(~\(~'AWS*2fRoute53Resolver~'OutboundQueryVolume~'EndpointId~'rslvr-out-7f4e7f22b46b481da\)\)~region~'eu-west-1\);query=~*7bAWS*2fRoute53Resolver*2cEndpointId*7d](https://eu-west-1.console.aws.amazon.com/cloudwatch/home?region=eu-west-1#metricsV2:graph=~(view~timeSeries~stacked~false~metrics~(~(~'AWS*2fRoute53Resolver~'OutboundQueryVolume~'EndpointId~'rslvr-out-7f4e7f22b46b481da))~region~'eu-west-1);query=~*7bAWS*2fRoute53Resolver*2cEndpointId*7d) (aws account: vf-iedelivery-prod-ss-01)



10.8.1.2 HAProxy DNS TEST

We can login one of our HAProxy ec2 instances. Right now, we are testing prd1.

| Filter by tags and attributes or search by keyword | | | | | | | | |
|--|---|---------------------|---------------|-------------------|----------------|------------------|--------------|--------|
| | Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Action |
| <input type="checkbox"/> | RHEL-SSM-Myst-vfie-delivery-EC2-PROD | i-015eb9693a2531790 | t2.micro | eu-west-1a | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | TRASPARENT-SQUID-PROXY-CENTRALIZED-IN... | i-019002edff73471cb | t3.medium | eu-west-1a | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | AG-PRD2-HAPROXY | i-02fa57d42f4e64017 | t2.medium | eu-west-1b | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | AG-PROD-HAPROXY | i-053cf32e3f8c5d443 | t2.medium | eu-west-1a | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | AG-PRD1-HAPROXY | i-058d5afba90a5ca7 | t2.medium | eu-west-1a | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | B-TEST-CENTRALIZED-INTERNET-ACCESS | i-05eb8798ded4cca3c | t3.micro | eu-west-1b | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | B-TRASPARANT-SQUID-PROXY-CENTRALIZED-I... | i-08cca6b4de6f7e7f8 | t3.medium | eu-west-1b | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | AG-PRQD-HAPROXY | i-0a0fa9b168fe2b60b | t2.medium | eu-west-1b | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | AG-PRD2-HAPROXY | i-0af6b256f68f64c07 | t2.medium | eu-west-1a | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | AG-PRD1-HAPROXY | i-0e209b7404d774911 | t2.medium | eu-west-1b | ● running | ✓ 2/2 checks ... | None | |
| <input type="checkbox"/> | A-TEST-CENTRALIZED-INTERNET-ACCESS | i-0f4a3e948f6e6312a | t3.micro | eu-west-1a | ● running | ✓ 2/2 checks ... | None | |

Results at the moment (it's failing):

Session ID: albamaría.diazfernandez@vodafone.com-04facb62daa97c0b3

Instance ID: i-058d5afbba90a5ca7

```
sh-4.2$ nslookup JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM
Server:      198.19.220.2
Address:     198.19.220.2#53

** server can't find JINNY-APP-PROD-VF-IE.INTERNAL.VODAFONE.COM: SERVFAIL

sh-4.2$ nslookup iebpokhr-vip.dc-dublin.de
Server:      198.19.220.2
Address:     198.19.220.2#53

** server can't find iebpokhr-vip.dc-dublin.de: SERVFAIL

sh-4.2$ nslookup iebpojhr-vip.dc-dublin.de
Server:      198.19.220.2
Address:     198.19.220.2#53

** server can't find iebpojhr-vip.dc-dublin.de: SERVFAIL

sh-4.2$ nslookup meh.prod.equinox.vf-ie.internal.vodafone.com
Server:      198.19.220.2
Address:     198.19.220.2#53

** server can't find meh.prod.equinox.vf-ie.internal.vodafone.com: SERVFAIL

sh-4.2$ nslookup meh.prod.equinox.vf-ie.internal.vodafone.com
Server:      198.19.220.2
Address:     198.19.220.2#53

** server can't find meh.prod.equinox.vf-ie.internal.vodafone.com: SERVFAIL

sh-4.2$ nslookup iebpojhr-vip.dc-dublin.de
Server:      198.19.220.2
Address:     198.19.220.2#53

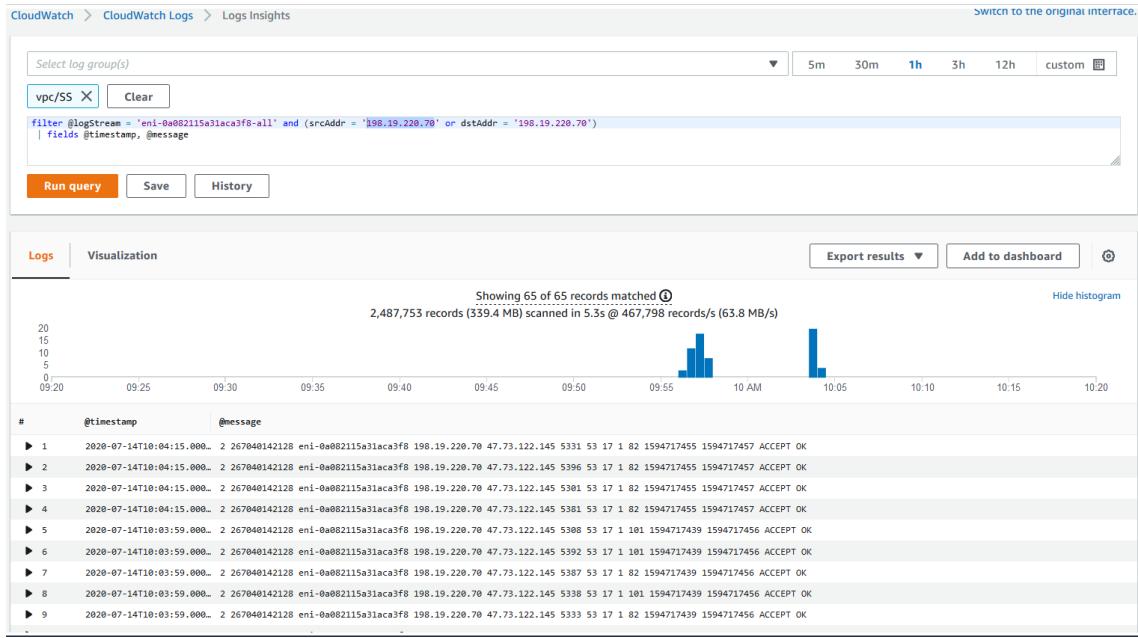
** server can't find iebpojhr-vip.dc-dublin.de: SERVFAIL
```

10.8.1.3 RESOLVER ENDPOINT IP ADDRESS LOGS

All logs from SS VPC are sent to the log group called VPC/SS. For the first IP Address 198.19.220.70, we can see the traffic going from route53 endpoint to gcd infoblox server. However, **we don't see any reply back**.

URL:[https://eu-west-1.console.aws.amazon.com/cloudwatch/home?region=eu-west-1#logsV2:logs-insights\\$3FqueryDetail\\$3D\\$257E\\$2528end\\$257E\\$0\\$257Estart\\$257E-3600\\$257EtimeType\\$257E\\$2527RELATIVE\\$257Eunit\\$257E\\$2527seconds\\$257EeditorString\\$257E\\$2527filter*20*40logStream*20*3d*20*27eni-0a082115a31aca3f8-all*27*20and*20*28srcAddr*20*3d*20*27198.19.220.70*27*20or*20dstAddr*20*3d*20*27198.19.220.70*27*29*0a*20*7c*20fields*20*40timestamp*2c*20*40message\\$257EisLiveTail\\$257Efalse\\$257EqueryId\\$257E\\$2527beffca0c-b9f1-4356-9da3-0093f32d5928\\$257Esource\\$257E\\$2528\\$257E\\$2527vpc*2fSS\\$2529\\$2529](https://eu-west-1.console.aws.amazon.com/cloudwatch/home?region=eu-west-1#logsV2:logs-insights$3FqueryDetail$3D$257E$2528end$257E$0$257Estart$257E-3600$257EtimeType$257E$2527RELATIVE$257Eunit$257E$2527seconds$257EeditorString$257E$2527filter*20*40logStream*20*3d*20*27eni-0a082115a31aca3f8-all*27*20and*20*28srcAddr*20*3d*20*27198.19.220.70*27*20or*20dstAddr*20*3d*20*27198.19.220.70*27*29*0a*20*7c*20fields*20*40timestamp*2c*20*40message$257EisLiveTail$257Efalse$257EqueryId$257E$2527beffca0c-b9f1-4356-9da3-0093f32d5928$257Esource$257E$2528$257E$2527vpc*2fSS$2529$2529)

VF IE Application Migration (AWS) – 03 VF-IE Network Design



TGW ROUTING

We have created in Network manager a Global network for VFIE. We can review the route analyzer for this case:

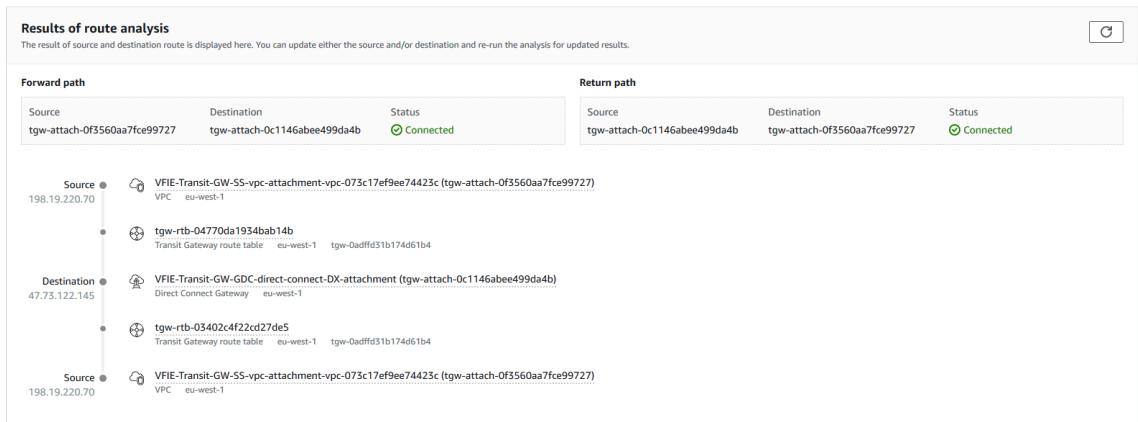
Query:

VFIE Route Analyzer

The Route Analyzer analyzes the routing path between a specified source and destination. Note, Route Analyzer checks the routes on Transit Gateway route tables only. [Learn more](#)

| | |
|--|--|
| Source | Destination |
| Transit Gateway | Transit Gateway |
| VFIE-Transit-GW | VFIE-Transit-GW |
| Transit Gateway attachment | Transit Gateway attachment |
| VFIE-Transit-GW-SS-vpc-attachment-vpc-073c17ef9ee74423c | VFIE-Transit-GW-GDC-direct-connect-DX-attachment |
| IP address | IP address |
| IPv4 or IPv6 address | IPv4 or IPv6 address |
| 198.19.220.70 | 47.73.122.145 |
| <input checked="" type="checkbox"/> Include return path in results | |
| <input type="checkbox"/> Middlebox appliance? Info | If selected, state those that are known in the results |
| Run route analysis | |

Result:



11 04 VF IE PROD VPC

11.11. Introduction

- Code: Terraform
- Repository: <https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/vf-iedelivery-infrastructure>
- Branch: master
- AWS account: vf-iedelivery-mgmt - 831341508773
- CI/CD: CodePipeline <https://eu-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/vf-iedelivery-infrastructure-prod-tf/view?region=eu-west-1>

Each VPC will be composed of:

- 1 Web tier subnet per AZ. Subnet IP range: '/23'
- 1 Presentation tier subnet per AZ. Subnet IP range: '/23'
- 1 Database tier subnet per AZ. Subnet IP range: '/23'
- 1 Private subnet for other purposes per AZ. Subnet IP range: '/24'
- VPC Flow logs enabled
- 1 different route table for each subnet.

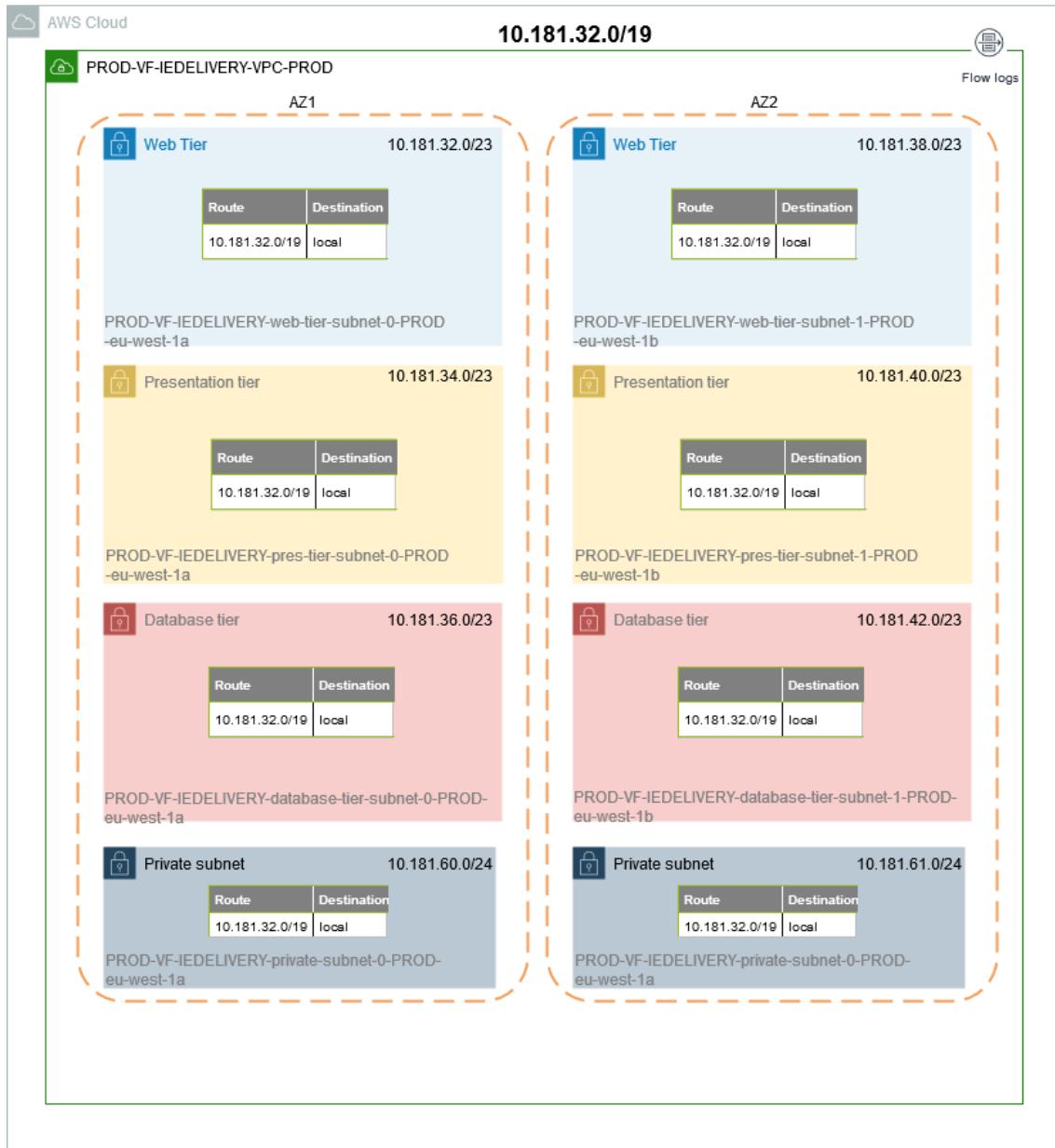
11.22. Design of PROD VPC

These are the subnet CIDR ranges that were chosen for prod environment:

| VPC | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|------|-----------------------|---------------|-----------------------------|-----------------------------|-------|
| PROD | 10.181.32.0/19 | 255.255.224.0 | 10.181.32.0 - 10.181.63.255 | 10.181.32.1 - 10.181.63.254 | 8190 |

The following diagram shows the state for the infrastructure deployed in AWS VF IE prod account by PCS. In each VPC, 3 subnets per AZ were created: web tier, presentation tier and database tier. Additionally, there are also others 2 private subnets that can be used for other purposes (endpoints, ENI, etc.).

vf-iedelivery-prod-01



11.33. Subnet CIDR range

| subnet | Subnet address | Netmask | Range of addresses | Useable IPs | Hosts |
|--------|-----------------------------|----------------------------|--|--|-------|
| Web | <code>10.181.32.0/23</code> | <code>255.255.254.0</code> | <code>10.181.32.0</code> - <code>10.181.33.255</code> | <code>10.181.32.1</code> - <code>10.181.33.254</code> | 510 |
| Pres | <code>10.181.34.0/23</code> | <code>255.255.254.0</code> | <code>10.181.34.0</code> - <code>10.181.35.255</code> | <code>10.181.34.1</code> - <code>10.181.35.254</code> | 510 |
| Datab | <code>10.181.36.0/23</code> | <code>255.255.254.0</code> | <code>10.181.36.0</code> - <code>10.181.37.255</code> | <code>10.181.36.1</code> - <code>10.181.37.254</code> | 510 |

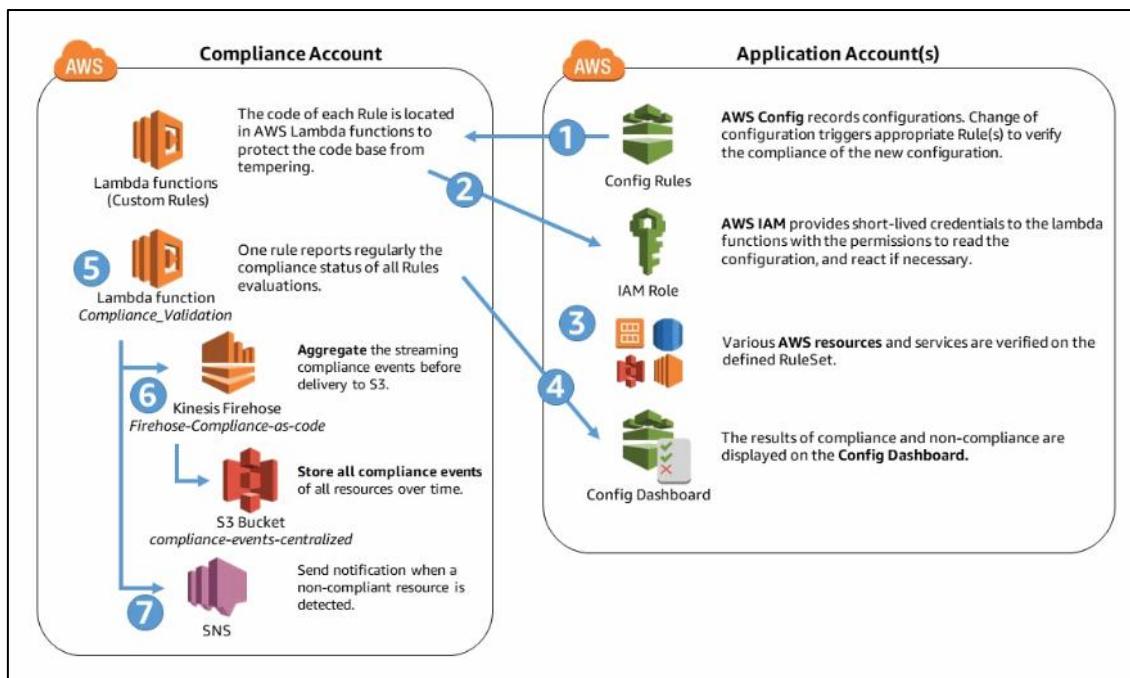
| | | | | | |
|-------|----------------|---------------|--------------------------------|--------------------------------|-----|
| Web | 10.181.38.0/23 | 255.255.254.0 | 10.181.38.0 - 10.181.39.255 | 10.181.38.1 - 10.181.39.254 | 510 |
| Pres | 10.181.40.0/23 | 255.255.254.0 | 10.181.40.0 - 10.181.41.255 | 10.181.40.1 - 10.181.41.254 | 510 |
| Datab | 10.181.42.0/23 | 255.255.254.0 | 10.181.42.0 - 10.181.43.255 | 10.181.42.1 - 10.181.43.254 | 510 |
| | 10.181.44.0/23 | 255.255.254.0 | 10.181.44.0 - 10.181.45.255 | 10.181.44.1 - 10.181.45.254 | 510 |
| | 10.181.46.0/23 | 255.255.254.0 | 10.181.46.0 - 10.181.47.255 | 10.181.46.1 - 10.181.47.254 | 510 |
| | 10.181.48.0/23 | 255.255.254.0 | 10.181.48.0 - 10.181.49.255 | 10.181.48.1 - 10.181.49.254 | 510 |
| | 10.181.50.0/23 | 255.255.254.0 | 10.181.50.0 - 10.181.51.255 | 10.181.50.1 - 10.181.51.254 | 510 |
| | 10.181.52.0/23 | 255.255.254.0 | 10.181.52.0 - 10.181.53.255 | 10.181.52.1 - 10.181.53.254 | 510 |
| | 10.181.54.0/23 | 255.255.254.0 | 10.181.54.0 - 10.181.55.255 | 10.181.54.1 - 10.181.55.254 | 510 |
| | 10.181.56.0/23 | 255.255.254.0 | 10.181.56.0 - 10.181.57.255 | 10.181.56.1 - 10.181.57.254 | 510 |
| | 10.181.58.0/23 | 255.255.254.0 | 10.181.58.0 - 10.181.59.255 | 10.181.58.1 - 10.181.59.254 | 510 |
| Priv | 10.181.60.0/24 | 255.255.255.0 | 10.181.60.0 - 10.181.60.255 | 10.181.60.1 - 10.181.60.254 | 254 |
| Priv | 10.181.61.0/24 | 255.255.255.0 | 10.181.61.0 - 10.181.61.255 | 10.181.61.1 - 10.181.61.254 | 254 |
| | 10.181.62.0/24 | 255.255.255.0 | 10.181.62.0 - 10.181.62.255 | 10.181.62.1 - 10.181.62.254 | 254 |
| | 10.181.63.0/24 | 255.255.255.0 | 10.181.63.0 - 10.181.63.255 | 10.181.63.1 - 10.181.63.254 | 254 |

12 05 VFIE INFRA MONITORING

12.11. Logging and Monitoring created by default when an AWS account is created via PCS:

- **CloudTrail** (AWS documentation: <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-concepts.html>)
 - What is clouptrail? Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Events include both API and non-API account activity taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs. You can use CloudTrail to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure. There are two types of events that can be logged in CloudTrail: management events and data events. By default, trails log management events, but not data events. Management events are also known as control plane operations, like configuring security, registering devices, configuring rules for routing data, setting up logging, etc. Data events are also known as data plane operations, example: amazon s3 object-level API activity and AWS Lambda function execution activity.
 - A dedicated CloudTrail is set up and sent directly to Cyber Defence. Details:
 - Log Group name: gdc-pcs-cloudtrail
 - IAM role: gdc-pcs-cloudtrail-role-DO-NOT-DELETE
 - Trail name: pcacsec1-ct1
- **GuardDuty** (AWS documentation: https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings.html)
 - What is GuardDuty? Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following data sources: VPC Flow Logs, AWS CloudTrail event logs, and DNS logs. It uses threat intelligence feeds, such as lists of malicious IP addresses and domains, and machine learning to identify unexpected and potentially unauthorized and malicious activity within your AWS environment.
 - GuardDuty is enabled using the IAM role **AWSServiceRoleForAmazonGuardDuty**
 - Actions with Findings :
 - Locating and Analyzing GuardDuty Findings
 - Archiving, Downloading, and Providing Feedback on GuardDuty Findings
 - Filtering Findings
 - Suppression Rules
- **AWS Config** (AWS documentation: Compliance Engine (<https://confluence.sp.vodafone.com/x/G2NhC>)
- **Compliance Engine** (<https://confluence.sp.vodafone.com/x/G2NhC>)
- AWS Config manages and schedules the evaluation of the rules within each Application Account across all enabled regions. Each rule is linked to a Lambda Function which implements the custom logic for checking. The Lambda is invoked by AWS Config and the function checks compliance by assuming a pre-defined role in the application account. The compliance state of each rule is recorded in AWS Config.
- Continuous monitoring, continuous assessment

- CloudWatch Alarms
- - VPC_ROUTE_TABLE_CHANGE_LOG_METRIC_ALARM
 - VPC_SECURITY_GROUP_CHANGE_LOG_METRIC_ALARM
 - IAM_POLICY_CHANGE_LOG_METRIC_ALARM
 - NETWORK_GATEWAY_CHANGE_LOG_METRIC_ALARM
 - VPC_CHANGE_LOG_METRIC_ALARM
 - S3_BUCKET_POLICY_CHANGE_LOG_METRIC_ALARM
 - CONSOLE_AUTHENTICATION_FAILURE_LOG_METRIC_ALARM
 - CONFIG_CHANGE_LOG_METRIC_ALARM
 - NETWORK_ACL_CHANGE_LOG_METRIC_ALARM
 - CMK_DELETION_CHANGE_LOG_METRIC_ALARM
- A report is created 4 times per week (Monday, Tuesday, Thursday and Friday) and send it to the security risk team. Owners of the AWS accounts will be in the distribution list to received weekly this report.



12.22. Logging and Monitoring created when PCS receives the demand and manages the account:

- **CloudWatch** (AWS documentation: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>)
 - It is not setup automatically. It will be configured once the infrastructure is deployed.
 - The infrastructure will be monitored via CloudWatch, with instances running the CloudWatch Agent to report more metrics back to CloudWatch. CloudWatch

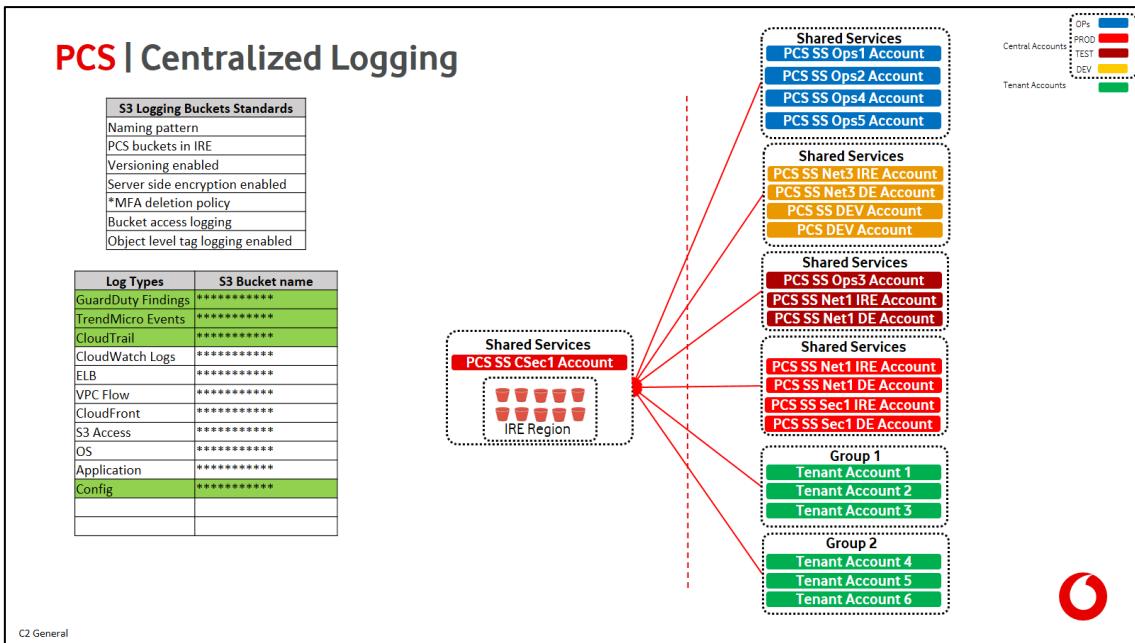
Alarms will be created in the specified environments (prod, pre-prod, etc.) and they will send an alert to SNS topics in the management account.

- Example of EC2 metrics: CPU Usage, (RAM) Memory Usage, Disk Space Usage, Instance Status Check....
- **S3 Access Logs**
 - Deployed by stacksets (already placed for VFIE). For each VFIE account, the following has been created:
 - S3 bucket: vf-iedelivery-`-${AWS::AccountId}`-logs
 - S3 logging bucket: s3-access-logs-vf-iedelivery-`-${AWS::AccountId}`-logs
- **IAM**
 - There is a dedicated IAM role created called TPCsec that Cyber Defence can use to assume role into any account created through PCS
- **VPC Flow Logs**
 - When you create a VPC you create flow logs that get sent to the S3 logging bucket for the account, and should ideally get sent to a CloudWatch log group also. There is a compliance engine rule checking that each created VPC has flow logs enabled.
- **Session Manager Logs**
 - Not set up automatically - Session Manager logging that sends Session Manager logs to the S3 logging bucket also.

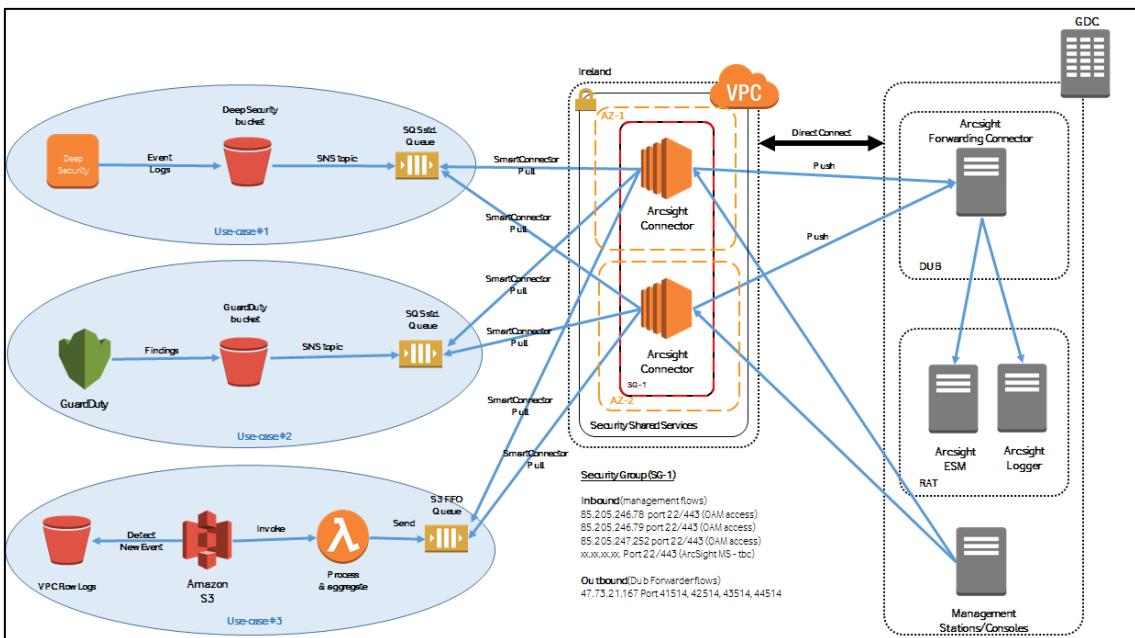
12.33. Centralized logging / monitoring (<https://confluence.sp.vodafone.com/x/FmNhC>)

PCS provides centralised logging of all AWS infrastructure resources into S3 buckets into Group Technology Security accounts. These logs are fed back to the GTS SIEM/SOC solution (ArcSight). This is a default for all PCS managed accounts

- This is a diagram of the centralized logging overview:



- This is a diagram of the ArcSight Integration with PCS centralised Logging



12.4 AWS LOGGING

12.4.1 VPC Flow Logs

2 VPC Flow Logs are created per VPC per environment: one goes to CloudWatch to allow for filters and easier investigation, encrypted via KMS, the other goes directly into the S3 logging buckets that are created via the StackSets i.e. vf-iedelivery-<ACCOUNT_ID>-logs

12.4.2 Instance Logs - CloudWatch Agent

12.4.2.1 Configuration: module instance-logging

The deployment of the CloudWatch Agent Config is fully managed by Terraform. A module has been created within the vf-iedelivery-infrastructure repo → modules/instance-logging that creates the CloudWatch Agent SSM Parameter, the JSON config of the CloudWatch agent, the CloudWatch Log Group for instance logs and the KMS Customer Master Key used to encrypt this log group.

```

Developer Tools > CodeCommit > Repositories > vf-iedelivery-infrastructure
vf-iedelivery-infrastructure
Notify dev Create pull request Clone URL
vf-iedelivery-infrastructure / modules / instance-logging Info Add file
Name
..
0-data.tf
1-cw-logs.tf
2-base-cloudwatch-agent-config.tf
2-base-cloudwatch-agent-config.tpl
outputs.tf
variables.tf

```

Resources created in 1-cw-logs.tf (see code [here](#)):

- **Cloudwatch group** encrypted with kms key and name "instancelogs-
\${var.TAGS["Environment"]}-\${var.TAGS["Project"]}"
- **kms key** with policy taken from
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/encrypt-log-data-kms.html>
- **kms alias** with name "alias/\${aws_cloudwatch_log_group.log_group.name}-key"
- **ssm parameter** to store the key arn with name
"/\${var.TAGS["Environment"]}/\${aws_cloudwatch_log_group.log_group.name}-key-cmk-arn"

Resources created in 2-base-cloudwatch-agent-config.tf (see [here](#)):

- ssm parameter with the cw agent configuration using template file for more flexibility

Template file for cw agent config (see aws documentation [here](#)):

```

${jsonencode(
{
  "agent": {
    "metrics_collection_interval": 60,
    "omit_hostname": true,
    "run_as_user": "root"
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/var/log/messages",
            "log_group_name": "${instance_log_group_name}",
            "log_stream_name": "{instance_id}-messages"
          },
          {
            "file_path": "/var/log/secure",
            "log_group_name": "${instance_log_group_name}",
            "log_stream_name": "{instance_id}-secure"
          },
          {
            "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-
cloudwatch-agent.log",
            "log_group_name": "${instance_log_group_name}",
            "log_stream_name": "{instance_id}-cw-agent"
          },
          {
            "file_path": "/var/log/amazon/ssm/amazon-ssm-agent.log",
            "log_group_name": "${instance_log_group_name}",
            "log_stream_name": "{instance_id}-ssm-agent"
          },
          {
            "file_path": "/var/log/amazon/ssm/errors.log",
            "log_group_name": "${instance_log_group_name}",
            "log_stream_name": "{instance_id}-ssm-agent-errors"
          }
        ]
      }
    }
  },
  "metrics": {
    "namespace": "CWAgentLinux",
    "append_dimensions": {
      "InstanceId": "$${aws:InstanceId}",
      "ImageID": "$${aws:ImageId}"
    },
    "metrics_collected": {
      "disk": {
        "measurement": [
          "disk_used_percent",
          "disk_inodes_free"
        ],
        "ignore_file_system_types": [
          "devtmpfs",
          "tmpfs",
          "overlay"
        ],
        "drop_device": true,
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      },
      "cpu": {
        "measurement": [

```

```

        "cpu_usage_idle",
        "cpu_usage_iowait",
        "cpu_usage_user",
        "cpu_usage_system"
    ],
    "metrics_collection_interval": 60,
    "resources": [
        "*"
    ],
},
"mem": {
    "measurement": [
        "mem_used_percent"
    ],
    "metrics_collection_interval": 60
},
"diskio": {
    "measurement": [
        "diskio_io_time"
    ],
    "metrics_collection_interval": 60
},
"swap": {
    "measurement": [
        "swap_used_percent"
    ],
    "metrics_collection_interval": 60
}
}
}
)
}
)
```

Configuration: user-data

When instances are created, they are bootstrapped with a user-data.sh script that contains an AWS CLI Run Command call to configure the CloudWatch Agent on the instances:

*Note: It has been tested in sandbox, to be tested in pre-prod before adding it to the AWS AMI Builder as part of the SSM document

```

#!/bin/bash -xe

cd /tmp
wget https://s3.amazonaws.com/amazoncloudwatch-
agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm
sudo rpm -U ./amazon-cloudwatch-agent.rpm
instance_id=$(wget -q -O - http://169.254.169.254/latest/meta-
data/instance-id)
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a
fetch-config -m ec2 -s -c ssm:${ssm_parameter}
sudo userdel -r cwagent
```

Documentation about installing aws cloudwatch agent:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/download-cloudwatch-agent-commandline.html> and
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-commandline-fleet.html>

1. The configuration file that we are going to use it is stored on a ssm parameter.

2. In this command, `-a fetch-config` causes the agent to load the latest version of the CloudWatch agent configuration file, and `-s` starts the agent. Enter one the following commands. Replace `configuration-file-path` with the path to the agent configuration file. This file is called `config.json` if you created it with the wizard, and might be called `amazon-cloudwatch-agent.json` if you created it manually.

12.5 AWS Monitoring: CW alarms

Alarm name convention in PCS: <Priority>_<Project-Name>_<Environment>_<Resource-Type>_<Alarm-Type>_<Resource-ID>

12.5.1 EC2 Monitoring: alarms and notifications

12.5.2 RDS monitoring: alarms and notifications

13 06 VF IE FIREWALL RULES

13.1 Production

| Firewall Communication Matrix | | | | | | | | | | | | | | |
|-------------------------------|--|-----------------|--|--|---|----------|------------|-------------|---------------|---------------|------------------------------|-------------------------|-------------------------|---|
| Status | Source | | Destination | | Network Address Translation | Protocol | Port | Environment | Service Group | Traffic Class | Remark | FW INC | ROUTING INC | CCP INC |
| | Source IP Address | Source Hostname | Destination IP Address | Destination Hostname | Protocol | Src Port | IP address | | | | | | | |
| PRODUCTION | | | | | | | | | | | | | | |
| VERIFIED | 10.10 9.100 .24 10.10 9.100 .25 10.10 9.100 .181 10.10 9.100 .186 | | 198.1 9.220 .0/27 198.1 9.220 .64/2 17 | AW S Pro d Sub nets 1- 1 1 2 0 | 1 0 1 0 1- 1 0 1 2 0 | | | TCP | PR OD | | Porta l to CCH Sync | INC00 00375 74948 | INC00 00375 74942 | https://ccp.vodafone.com/commsmatrix/aggr/142130 |
| VERIFIED | 10.10 9.100 .11 10.10 9.100 .12 10.10 9.100 .13 10.10 9.100 .14 10.10 9.100 .15 10.10 9.100 .16 10.10 9.100 .17 10.10 9.100 .38 | | 198.1 9.220 .0/27 198.1 9.220 .64/2 17 | AW S Pro d Sub nets 1- 1 1 2 0 | 1 0 1 0 1- 1 0 1 2 0 | | | TCP | PR OD | | UFE to CCH Sync | INC00 00375 74948 | INC00 00375 74942 | https://ccp.vodafone.com/commsmatrix/aggr/142130 |

| | | | | | | | | | | | | | | |
|-------------------------|--|-----|---|---|---------------------------------------|-------------|----------|---|-------------------------|-------------------------|-------------------------|---|--|--|
| | 10.10 9.100 .39 | | | | | | | | | | | | | |
| IMPL EME NTE D | 10.15 1.4.7 9 10.15 1.4.8 2 10.15 1.4.9 2 10.15 1.4.9 5 10.16 2.114 .10 10.16 2.114 .11 10.16 2.114 .12 10.16 2.114 .13 10.16 2.114 .15 10.16 2.114 .16 10.16 2.114 .18 10.16 2.114 .19 10.16 2.114 .26 10.16 2.114 .25 | | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AW S Pro d Sub nets 0 | 1 0 1 0 1- 1 1 0 | T C P | PR OD | GIS MSH/ Tibco to CCH Sync | INC00 00375 74948 | INC00 00375 74942 | INC00 00375 74942 | https://ccp.vodafone.com/commsmatrix/aggr/142130 | | |
| IMPL EME NTE D | 10.10 9.100 .239 10.10 9.100 .240 10.10 9.100 .241 | OSB | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AW S Pro d Sub nets 0 | 1 0 1 0 1- 1 1 0 | T C P | PR OD | OSB to CCH Sync | INC00 00375 74948 | INC00 00375 74942 | INC00 00375 74942 | https://ccp.vodafone.com/commsmatrix/aggr/142130 | | |

| | | | | | | | | | | | | | | | | |
|-------------------------|---|----------------------------|---|--|--|--|-------------|----------|--|--------------------------------|-------------------------|-------------------------|--|---|--|--|
| | | | | | | | | | | | | | | | | |
| IMPL EME NTE D | 10.10 9.179 .146 10.10 9.179 .147 | MFT | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AW S Pro d Sub nets 1- .64/2 7 | 1 0 1 1 1 1 0 1 1 3 9 | | T C P | PR OD | | OSB to CCH Batch | INC00 00375 74948 | INC00 00375 74942 | | https://ccp.vodafone.com/commsmatrix/aggr/142130 | | |
| IMPL EME NTE D | 198.1 8.74. 197 192.1 25.24 7.100 | Conn ectDi rect | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AW S Pro d Sub nets 1- .64/2 7 | 1 0 1 1 1 1 0 1 1 3 9 | | T C P | PR OD | | OSB to CCH FS | INC00 00375 74948 | INC00 00375 74942 | | https://ccp.vodafone.com/commsmatrix/aggr/142130 | | |
| VERI FIED | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AWS Prod Subn ets | 10.16 3.184 .4 | 2 0 1 1 1 1 2 0 1 1 2 2 0 1 1 3 2 0 1 1 4 2 0 1 0 1 | 2 0 1 1 1 1 2 0 1 1 2 2 0 1 1 3 2 0 1 1 4 2 0 1 0 1 | | T C P | PR OD | | CCH Sync to SMS C | INC00 00375 74948 | INC00 00375 74942 | | https://ccp.vodafone.com/commsmatrix/aggr/142130 | | |
| VERI FIED | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AWS Prod Subn ets | 10.1 63.18 7.4 | 2 0 1 1 1 1 2 0 1 1 2 2 0 1 1 2 2 0 1 | 2 0 1 1 1 1 2 0 1 1 2 2 0 1 1 2 2 0 1 | | T C P | PR OD | | CCH Sync to JINN Y | INC00 00375 74948 | INC00 00375 74942 | | https://ccp.vodafone.com/commsmatrix/aggr/142130 | | |

| | | | e Range | | | 201020110220010301-1032010401-10420 | | | n GUI's | | |
|----------------|---------------------------------------|---|-------------------|---------------------------|---------------|-------------------------------------|-------|---|-------------------|-------------------|---|
| IMPL EME NTE D | 10.74 .120. 112/2 8 | VFIE Web sens e Prox y Clea nzon e Rang e | 46.10 8.156 .0/27 | AW S Mg mt-Sub nets | 9 0 0-9 0 3 0 | T C P | PR OD | Web sens e Prox y to Dev Ops Tool GUI's | INC00 00375 74948 | INC00 00375 74942 | https://ccp.vodafone.com/commsmatrix/aggr/142130 |
| IMPL EME NTE D | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AWS Prod Subn ets | 198.1 8.67. 252 | ME H VIP | 2 2 1 6 4 4 3 | T C P | PR OD | CCH Sync to MEH | INC00 00376 00491 | INC00 00376 00476 | https://ccp.vodafone.com/commsmatrix/aggr/143431 |
| IMPL EME NTE D | 198.1 9.220 .0/25 | AWS Prod Subn ets | 47.73 .21.7 4 | vgl m12 vr.d c-dubl in.de | 8 0 9 0 | T C P | PR OD | CCH to AppD ynam ics Contr olle | INC00 00376 00491 | INC00 00376 00476 | https://ccp.vodafone.com/commsmatrix/aggr/143431 |

| | | | | | | | | | | | |
|-------------------------|------------------------------|---|-------------------------|------------------------------------|---|-------------|----------|--|-------------------------|-------------------------|---|
| IMPL EME NTE D | 10.74 .120. 112/2 8 | VFIE Web sens e Prox y Clea nzon e Rang e | 198.1 9.220 .0/25 | AW S Pro d Sub nets | 7 0 0 1 4 4 3 8 8 0 0- 8 8 1 0 | T C P | PR OD | Web sens e Prox y to Webl ogic Admi n GUI's | INC00 00376 00491 | INC00 00376 00476 | https://ccp.vodafone.com/commsmatrix/aggr/143431 |
| IMPL EME NTE D | 10.10 9.98. 0/24 | Dubli n VDI Subn et | 198.1 9.220 .0/25 | AW S Pro d Sub nets | 3 3 0 0 3 3 0 0 1 4 4 3 7 0 0 1 8 8 0 0- 8 8 1 0 | T C P | PR OD | VDI Subn et to OCM & Webl ogic Admi n GUI | INC00 00376 00491 | INC00 00376 00476 | https://ccp.vodafone.com/commsmatrix/aggr/143431 |
| IN PRO GRE SS | 10.16 3.78. 0/23 | Dubli n VDI Subn et | 198.1 9.220 .0/25 | AW S Pro d Sub nets | 3 3 0 0 3 3 0 0 1 4 4 3 7 0 0 1 8 | T C P | PR OD | VDI Subn et to OCM & Webl ogic Admi n GUI | INC00 00376 00491 | INC00 00376 00476 | https://ccp.vodafone.com/commsmatrix/aggr/143431 |

| | | | | | | | | | | | | | | | |
|-------------------------|--|--------------------------------------|---|------------------------------------|---|-------------|------------------|--|--|-------------------------|-------------------------|--|---|--|--|
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| IMPL EME NTE D | 10.78 .177. 156 | IEJE NKV R | 198.1 9.220 .0/25 | AW S Pro d Sub nets | 3 3 0 0 0 1 4 4 3 7 0 0 1 1 1 8 0 1 1 1 8 8 0 0 -8 8 1 0 | T C P | P R O D | | Nagi os monit oring to OCM & Webl ogic Admi n GUI | INC00 00376 00491 | INC00 00376 00476 | | https://ccp.vodafone.com/commsmatrix/aggr/143431 | | |
| VERI FIED | 10.10 9.100 .24 10.10 9.100 .25 10.10 9.100 .181 10.10 9.100 .186 10.10 9.100 .11 10.10 9.100 .12 10.10 9.100 .13 10.10 | UFE/ Porta I/O S /0/27 B | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AW S Pro d Sub nets | 8 0 1 1 1 1 8 8 0 0 -8 8 1 0 | T C P | P R O D | | UFE/ Porta I/O S B to CCH Sync | INC00 00376 00491 | INC00 00376 00476 | | https://ccp.vodafone.com/commsmatrix/aggr/143431 | | |

| | | | | | | | | | | | | | | |
|-------------------------|---|---|---------------------|------------------|-------------|----------|---|-------------------------|-------------------------|-------------------------|---|--|--|--|
| | 9.100 .14 10.10 9.100 .15 10.10 9.100 .16 10.10 9.100 .17 10.10 9.100 .38 10.10 9.100 .39 10.10 9.100 .239 10.10 9.100 .240 10.10 9.100 .241 | | | | | | | | | | | | | |
| IMPL EME NTE D | 10.15 1.4.7 9 10.15 1.4.8 2 10.15 1.4.9 2 10.15 1.4.9 5 10.16 2.114 .10 10.16 2.114 .11 10.16 2.114 .12 10.16 2.114 .13 10.16 2.114 .15 10.16 2.114 .16 10.16 2.114 .18 | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AW S Pro d | 8 0 1 1 | T C P | PR OD | Tibco /GIS MSH to CCH Sync | INC00 00376 00491 | INC00 00376 00476 | INC00 00376 00476 | https://ccp.vodafone.com/commsmatrix/aggr/143431 | | | |

| | | | | | | | | | | | | | | | |
|-------------------------|---|----------------------------|--|------------------------------------|---|--|-------------|----------|--|------------------------------------|-------------------------|-------------------------|--|---|--|
| | 10.16 2.114 .19 10.16 2.114 .26 10.16 2.114 .25 | | | | | | | | | | | | | | |
| VERI FIED | 198.1 8.65. 56 198.1 8.65. 57 | | 198.1 9.220 .0/27 198.1 9.220 .64/2 | AW S Pro d Sub nets | 1 0 1 1 1- 1 7 | | T C P | PR OD | | NIFI to CCH Batch | INC00 00376 00491 | INC00 00376 00476 | | https://ccp.vodafone.com/commsmatrix/aggr/143431 | |
| VERI FIED | 37.25 .160. 19 | R2 HAPr oxy | 198.1 9.220 .0/27 198.1 9.220 .64/2 | AW S Pro d Sub nets | 7 0 0 1 1 7 | | T C P | PR OD | | R2 Prox y to CCH | INC00 00376 32263 | INC00 00376 32426 | | https://ccp.vodafone.com/commsmatrix/aggr/144029 | |
| VERI FIED | 198.1 9.220 .0/27 98.19 .220. 64/27 | AWS Prod Subn ets | 10.1 63.18 7.4 | JIN NY | 6 5 4 3 | | T C P | PR OD | | CCH Sync to JINN Y | INC00 00376 32263 | INC00 00376 32426 | | https://ccp.vodafone.com/commsmatrix/aggr/144029 | |
| VERI FIED | 198.1 9.220 .0/27 98.19 .220. 64/27 | AWS Prod Subn ets | 10.78 .48.5 7 | OS B VIP | 3 5 0 0 0 3 5 0 0 1 | | T C P | PR OD | | CCH Sync to OSB | INC00 00376 32263 | INC00 00376 32426 | | https://ccp.vodafone.com/commsmatrix/aggr/144029 | |
| IMPL EME NTE D | 47.73 .21.7 4 47.73 .21.7 5 | Sites cope | 198.1 9.220 .0/27 98.19 .220. 64/27 | AW S Pro d Sub nets | 8 0 1 1 7 0 0 1 1 0 1 2 0 | | T C P | PR OD | | Sites cope to CCH Prod | | | | https://ccp.vodafone.com/commsmatrix/aggr/146163 | |

| | | | | | | | | | | | |
|-------------------------|--|--|--|---|--|-------------|----------|--|------------------------------------|-------------------------|---|
| IMPL EME NTE D | 176.1 25.13 .67 176.1 25.13 .68 176.1 25.13 .69 176.1 25.13 .70 | Sites cope | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AW S Pro d Sub nets 0 | 8 0 1 1 7 0 0 1 1 0 1 2 0 | T C P | PR OD | | Sites cope to CCH Prod | | https://ccp.vodafone.com/commsmatrix/aggr/146163 |
| IMPL EME NTE D | 198.1 9.220 .0/24 46.10 8.156 .0/27 | AWS SS VPC | 47.73 .122. 145 47.73 .122. 171 47.73 .81.1 08 | Info blox DN S serv er | 5 3 3 0 | T C P | PR OD | | AWS to Inflob lox | | https://ccp.vodafone.com/commsmatrix/aggr/146163 |
| IMPL EME NTE D | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AWS Prod Subn ets | 10.16 3.184 .4 | SM SC | 1 0 4 0 1 0 5 0 1 0 6 0 1 0 7 0 | T C P | PR OD | | CCH Sync to SMS C | INC00 00375 76265 | https://ccp.vodafone.com/commsmatrix/aggr/144029 |
| IN PRO GRE SS | 198.1 9.220 .0/27 198.1 9.220 .64/2 7 | AWS Prod Subn ets | 10.10 9.100 .242 10.10 9.100 .243 10.10 9.100 .244 | Bill- AM DD Svr | 2 2 | T C P | PR OD | | CCH Sync to AMD D | INC00 00377 64880 | https://ccp.vodafone.com/user/commsmatrix/145944 |
| IMPL EME NTE D | 47.73 .43.2 05 47.73 .43.2 35 145.2 30.15 .70 47.73 | DNS Reso lvers On-premi se | 198.1 9.220 .0/24 | AW S DN S serv er | 5 3 | T C P | PR OD | | GDC DNS to AWS DNS | INC00 00376 81800 | https://ccp.vodafone.com/commsmatrix/aggr/144961 |

| | | | | | | | | | | | | | | | |
|-------------------------|---|------------------------------|-------------------------|--|--------------------------|--|-------------|----------|--|--|-------------------------|--|--|---|--|
| | .122. 145 47.73 .122. 171 195.2 33.11 3.167 47.73 .81.8 1 47.73 .81.1 08 | | | | | | | | | | | | | | |
| IMPL EME NTE D | 198.1 9.220 .0/25 | AWS Prod Subn ets | 10.16 2.66. 40 | Lotu s Not es | 2 2 | | T C P | PR OD | | CCH to Lotus Note s (SFT P) | INC00 00375 76265 | | | https://ccp.vodafone.com/commsmatrix/aggr/144029 | |
| IMPL EME NTE D | 10.10 9.98. 0/24 | Dubli n VDI Subn et | 198.1 8.75. 195 | I1P RO XY- VIP | 3 0 2 2 | | T C P | PR OD | | VDI Subn et to File Tran sfer serve r | INC00 00377 64880 | | | https://ccp.vodafone.com/user/commsmatrix/145944 | |
| IN PRO GRE SS | 10.16 3.78. 0/23 | Dubli n VDI Subn et | 198.1 8.75. 195 | I1P RO XY- VIP | 3 0 2 2 | | T C P | PR OD | | VDI Subn et to File Tran sfer serve r | INC00 00377 64880 | | | https://ccp.vodafone.com/user/commsmatrix/145944 | |
| IMPL EME NTE D | 198.1 8.65. 7 198.1 8.65. 8 198.1 8.65. 9 198.1 8.65. 10 198.1 8.65. 11 198.1 8.65. 12 | MEH PRO D | 198.1 9.220 .0/27 | CC H PR 1 198.1 9.220 .64/2 7 | 8 0 1 1 nets | | T C P | PR OD | | MEH to AWS | | | | | |

| | | | | | | | | | | | | | |
|------|-------|------|-------|------|---|--|---|----|------|--|--|--|--|
| IMPL | 198.1 | CCH | 47.73 | vgl | 8 | | T | PR | AWS | | | | |
| EME | 9.220 | PRO | .21.7 | m12 | 1 | | C | OD | to | | | | |
| NTE | .0/27 | D | 4 | vr.d | 8 | | P | | AppD | | | | |
| D | 198.1 | Subn | | c- | 1 | | | | | | | | |
| | 9.220 | ets | | dubl | | | | | | | | | |
| | .64/2 | | | in.d | | | | | | | | | |
| | 7 | | | e | | | | | | | | | |

13.2 PRE-PRODUCTION

| Firewall Communication Matrix | | | | | | | | | | | | | | |
|-------------------------------|--|-----------------|---|---|--|---------|-------------|-------------|-------------|---------------|--------------------|--------------------|-------------|---|
| Status | Source | | Destination | | Network Address Translation | | Protocol | Environment | Service | Transport | Remark | FW INC | ROUTING INC | CCP |
| | Source IP Address | Source Hostname | Destination IP Address | Destination Hostname | Port | Src/Dst | IP address | Protocol | Environment | Service Group | Transport | Remark | FW INC | ROUTING INC |
| PRE-PRODUCTION | | | | | | | | | | | | | | |
| VERIFIED | 10.10 9.100 .179 10.10 9.100 .180 10.10 9.100 .184 10.10 9.100 .185 | | 198.1 9.220 .32/2 10.10 9.100 .180 10.10 9.100 .184 10.10 9.100 .185 | AW S Pro d 1- Sub net s 1 2 0 | 1 0 1 0 1 0 0 1 2 0 | | T C P | PR OD | | | Portal to CCH Sync | INC000375 75317 | | https://ccp.vodafone.com/commsmatrix/aggr/142149 |
| VERIFIED | 10.10 9.100 .135 10.10 9.100 .136 10.10 9.100 .137 10.10 9.100 .138 10.10 9.100 .139 | | 198.1 9.220 .32/2 10.10 9.100 .136 10.10 9.100 .137 10.10 9.100 .138 10.10 9.100 .139 | AW S Pro d 1- Sub net s 1 2 0 | 1 0 1 0 1 0 0 1 2 0 | | T C P | PR OD | | | UFE to CCH Sync | INC000375 75317 | | https://ccp.vodafone.com/commsmatrix/aggr/142149 |

| | | | | | | | | | | | | | | |
|--------------|--|--|--|---|-------------|----------|---------------------------------|-------------------------|--|--|--|--|--|---|
| | 10.10 9.100 .140 10.10 9.100 .141 10.10 9.100 .34 10.10 9.100 .35 10.10 9.100 .142 10.10 9.100 .143 10.10 9.100 .144 10.10 9.100 .145 10.10 9.100 .146 10.10 9.100 .147 10.10 9.100 .148 10.10 9.100 .36 10.10 9.100 .37 | | | | | | | | | | | | | |
| VERI FIED | 10.16 2.122 .16 10.16 2.122 .17 10.16 2.114 .20 10.16 2.114 .21 | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pro d Sub net s 1- | 1 0 1 0 1 1 0 2 0 | T C P | PR OD | TIB CO to CCH Sync | INC00 00375 75317 | | | | | | https://ccp.vodafone.com/commsmatrix/aggr/142149 |
| VERI FIED | 10.16 2.111 .89 10.16 2.111 .92 10.16 | 198.1 9.220 .32/2 7 198.1 9.220 | AW S Pro d Sub net s 0 | 1 0 1 0 1 1 0 | T C P | PR OD | GIS MSH to CCH Sync | INC00 00375 75317 | | | | | | https://ccp.vodafone.com/commsmatrix/aggr/142149 |

| | | | | | | | | | | | | | | | |
|----------|--|--------------------------------|--|--|--|-------------|----------|--|-------------------------------|-------------------------|--|--|--|---|--|
| | 2.111 .81 10.16 2.111 .82 | | .96/2 7 | | 1 2 0 | | | | | | | | | | |
| VERIFIED | 10.10 9.100 .96 10.10 9.100 .97 10.10 9.100 .98 10.10 9.100 .99 | OSB | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pro d Sub net s | 1 0 1 0 1- 1 0 1 2 0 | T C P | PR OD | | OSB to CCH Sync | INC00 00375 75317 | | | | https://ccp.vodafone.com/commsmatrix/aggr/142149 | |
| VERIFIED | 10.10 9.100 .118 10.10 9.100 .119 | MFT | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pro d Sub net s | 1 0 1 0 1- 1 0 1 2 0 | T C P | PR OD | | MFT to CCH Batch | INC00 00375 75317 | | | | https://ccp.vodafone.com/commsmatrix/aggr/142149 | |
| VERIFIED | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Prod Sub nets | 10.1 63.18 7.4 | | 6 5 4 3 | T C P | PR OD | | CCH Sync to JINNY | INC00 00375 75317 | | | | https://ccp.vodafone.com/commsmatrix/aggr/142149 | |
| VERIFIED | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Prod Sub nets | 10.16 3.184 .4 | | 1 0 4 0 1 0 5 0 1 0 6 0 1 0 7 0 | T C P | PR OD | | CCH Sync to SMS C | INC00 00375 75317 | | | | https://ccp.vodafone.com/commsmatrix/aggr/142149 | |
| VERIFIED | 198.1 9.220 .32/2 7 198.1 | AW S Prod | 10.78 .48.8 3 10.78 | OS B VIP | 3 2 0 0 7 | T C P | PR OD | | CCH Sync to OSB | INC00 00375 75317 | | | | https://ccp.vodafone.com/commsmatrix/aggr/142149 | |

| | | | | | | | | | |
|--------------|--|--------------------------------|---|---|-------------|----------|-----------------------------|-------------------------|---|
| | 9.220 .96/2 7 | Sub nets | .48.8 4 | 3 3 0 0 1 3 3 0 0 2 3 3 0 0 4 7 0 0 2 5 5 5 6 | T C P | PR OD | CCH Sync to MEH | INC00 00375 75317 | https://ccp.vodafone.com/commsmatrix/aggr/142149 |
| VERI FIED | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Prod Sub nets | 198.1 8.67. 253 | ME H VIP 2 2 1 6 0 8 0 1 6 4 4 3 | T C P | PR OD | CCH Sync to MML DB | INC00 00375 75317 | https://ccp.vodafone.com/commsmatrix/aggr/142149 |
| VERI FIED | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Prod Sub nets | 10.10 9.101 .41 10.10 9.101 .42 10.10 9.101 .43 10.10 9.101 .44 10.10 9.101 .45 10.10 9.100 .118 10.10 9.100 .119 | MM L SC 0 AN 0 0 0 0 3 3 0 0 1 | T C P | PR OD | CCH Sync to MML DB | INC00 00375 75317 | https://ccp.vodafone.com/commsmatrix/aggr/142149 |

| | | | | | | | | | | |
|----------|--|--|---------------------------------|------------------|-------------|----------|------------------------------|-------------------------|--|---|
| VERIFIED | 10.10 9.100 .179 10.10 9.100 .180 10.10 9.100 .184 10.10 9.100 .185 | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S d Sub net s | 8 0 1 1 | T C P | PR OD | Port al to CCH Sync | INC00 00376 07345 | | https://ccp.vodafone.com/commsmatrix/aggr/143432 |
| VERIFIED | 10.10 9.100 .135 10.10 9.100 .136 10.10 9.100 .137 10.10 9.100 .138 10.10 9.100 .139 10.10 9.100 .140 10.10 9.100 .141 10.10 9.100 .34 10.10 9.100 .35 10.10 9.100 .142 10.10 9.100 .143 10.10 9.100 .144 10.10 9.100 .145 10.10 9.100 .146 10.10 9.100 .147 10.10 9.100 | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S d Sub net s | 8 0 1 1 | T C P | PR OD | UFE to CCH Sync | INC00 00376 07345 | | https://ccp.vodafone.com/commsmatrix/aggr/143432 |

| | | | | | | | | | | | | | | |
|--------------|--|---|---|--|-------------|---------------------|---------------------------------|-------------------------|-------------------------|--|--|--|--|---|
| | .148 10.10 9.100 .36 10.10 9.100 .37 | | | | | | | | | | | | | |
| VERI FIED | 10.16 2.122 .16 10.16 2.122 .17 10.16 2.114 .20 10.16 2.114 .21 | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pro d Sub net s | 8 0 1 1 Sync | T C P | PR OD | TIB CO to CCH Sync | INC00 00376 07345 | | | | | | https://ccp.vodafone.com/commsmatrix/aggr/143432 |
| VERI FIED | 10.16 2.111 .89 10.16 2.111 .92 10.16 2.111 .81 10.16 2.111 .82 | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pro d Sub net s | 8 0 1 1 Sync | T C P | PR OD | GIS MSH to CCH Sync | INC00 00376 07345 | | | | | | https://ccp.vodafone.com/commsmatrix/aggr/143432 |
| VERI FIED | 10.10 9.100 .96 10.10 9.100 .97 10.10 9.100 .98 10.10 9.100 .99 | OSB 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pro d Sub net s | 8 0 1 1 Sync | T C P | PR OD | OSB to CCH Sync | INC00 00376 07345 | | | | | | https://ccp.vodafone.com/commsmatrix/aggr/143432 |
| VERI FIED | 198.1 8.65. 31 | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pre pro d Sub net s | 1 0 1 1- 1 0 1 3 9 | T C P | PR EP RO D | NiFi to CCH Sync | INC00 00376 07070 | INC00 00376 12452 | | | | | https://ccp.vodafone.com/commsmatrix/aggr/144028 |

| | | | | | | | | | | | | |
|-------------------------|--|------------------------------|---|---|--|-------------|---------------------|--|---|-------------------------|-------------------------|---|
| VERI FIED | 198.1 9.220 .32/2 7 | AW S Prep rod | 10.16 2.229 .54 | SM SC | 1 0 4 0 1 0 5 0 1 0 6 0 1 0 7 0 | T C P | PR EP RO D | | CCH Sync to SMS C | INC00 00376 07070 | INC00 00376 12452 | https://ccp.vodafone.com/commsmatrix/aggr/144028 |
| | 198.1 9.220 .96/2 7 | Sub nets | | | | | | | R2 Prox y to CCH | INC00 00376 07070 | INC00 00376 12452 | https://ccp.vodafone.com/commsmatrix/aggr/144028 |
| | 37.25 .160. 19 | R2 HAP roxy | 198.1 9.220 .32/2 7 | AW S Pre pro d Sub net s | 8 0 1 1 7 0 0 1 1 1 1 1 2 0 | T C P | PR EP RO D | | | | | |
| | 198.1 9.220 .32/2 7 | AW S Prep rod | 10.78 .48.8 3 10.78 | OS B VIP | 3 5 0 0 | T C P | PR EP RO D | | CCH Sync to OSB | INC00 00376 07070 | INC00 00376 12452 | https://ccp.vodafone.com/commsmatrix/aggr/144028 |
| VERI FIED | 198.1 9.220 .32/2 7 | 198.1 9.220 .96/2 7 | Sub nets | .48.8 4 | 1 3 5 0 0 0 | | | | | | | |
| IMPL EME NTE D | 198.1 8.74. 197 192.1 25.24 7.100 | Con nect Dire ct | 198.1 9.220 .32/2 7 | AW S Pre pro d Sub net s | 1 0 1 2 0 | T C P | PR EP RO D | | Con nect Dire ct to CCH Sync | INC00 00377 64880 | | https://ccp.vodafone.com/user/commsmatrix/145944 |
| IN PRO GRE SS | 198.1 9.220 .32/2 7 | AW S Prep rod | 10.10 9.100 .104 10.10 | Bill- AM DD Svr | 2 2 | T C P | PR EP RO D | | CCH Sync to AMD D | INC00 00377 64880 | | https://ccp.vodafone.com/user/commsmatrix/145944 |
| | 198.1 9.220 .96/2 7 | Sub nets | .105 10.10 9.100 .106 10.10 | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | |
|--------------|--|--------------------------------------|---|--|--|-------------|---------------------|--|-------------------------------------|--|--|--|--|--|--|---|---|--|
| | | | 9.100 .107 10.10 9.100 .108 10.10 9.100 .109 | | | | | | | | | | | | | | | |
| VERI FIED | 37.25 .160. 48 37.25 .160. 49 37.25 .160. 50 37.25 .160. 61 | Load runner | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pre pro d Sub net s | 8 0 1 1 7 0 0 1 2 | T C P | PR EP RO D | | Load runner to CCH Sync | | | | | | | | https://ccp.vodafone.com/commsmatrix/aggr/146163 | |
| VERI FIED | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | AW S Pre rod Sub nets | 37.25 .160. 48 37.25 198.1 9.220 37.25 7 | Loa dru nne r .160. 0 49 37.25 .160. 50 37.25 .160. 61 | 3 0 0 5 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 | T C P | PR EP RO D | | CCH Sync to Load runner | | | | | | | https://ccp.vodafone.com/commsmatrix/aggr/146163 | | |
| VERI FIED | 198.1 8.65. 3 198.1 8.65. 4 198.1 8.65. 5 198.1 8.65. 6 | MEH PRE PRO D | 198.1 9.220 .32/2 7 198.1 9.220 .96/2 7 | CC H PR 1 EP 1 RO D Sub net s | 8 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 | T C P | PR EP RO D | | MEH to AW S | | | | | | | | | |
| IN PRO | 198.1 9.220 .32/2 7 | CCH PRE PRO D | 10.10 9.100 .98 10.10 | PR D2 3 0 | 3 3 | T C P | PR EP | | AW S to PRD | | | | | | | | | |

| | | | | | | | | | | | | | |
|------------------------|------------------------------|------------------------|--|---------------------------|--------------------------------------|-------------|---------------------|--|-------------------------|----------|--|---|--|
| GRE SS | 198.1 9.220 .96/2 7 | Sub nets | 9.100 .99 10.78 .48.8 4 | OS B | 0 7 | | | RO D | | 2 OSB | | | |
| IN PRO GRE SS | 198.1 9.220 .32/2 7 | AW S Prep rod | 10.10 9.101 .81 10.10 9.101 9.101 9.101 10.10 9.101 .84 10.10 9.101 .39 10.10 9.101 .40 10.10 9.101 .41 10.10 9.101 .42 | MM L DB Lin k | 3 3 0 0 0 0 0 1 | T C P | PR EP RO D | CCH DB to MML DB (DB Link) | INC00 00378 08948 | | | https://ccp.vodafone.com/user/commsmatrix/146319 | |

14 07 VFIE EC2 Image Builder

-
-
- [INTRODUCTION](#)
- [PREREQUISITES](#)
- [COMPONENTS](#)
 - [Image pipeline](#)
 - [Image recipe](#)
 - [Source image](#)
 - [Build components](#)
 - [Test components](#)
 - [Document](#)
- [IMPLEMENTATION](#)
 - [CLOUDFORMATION STACK](#)
 - [SSM DOCUMENTS](#)
 - [AWS CONSOLE RESOURCES](#)
- [HOW TO: NEW AMI](#)
 - [PERMISSIONS TO NEW AMI](#)

14.1 INTRODUCTION

VFIE needs a customized image for the ec2 servers. For cch-sal application, we have 13 different domains (cch-ob-osb, cch-ob-odi, etc.), where each domain has one admin server and several nodes (documentation [here](#)). We are going to use AWS EC2 image builder for creating, managing and deploying customized, secure and up-to-date server images, that are pre-installed and pre-configured with software and settings to meet the IT requirements. These requirements are needed for the target OFMW servers to support Myst provisioning, such as Myst Agent, Oracle Java, SQL Plus, oracle users, etc (documentation [here](#)).

With AWS EC2 image builder, you follow these steps.

1. **Select source image.** You select a source OS image, for example, an existing AMI.
2. **Create image recipe.** You add components to create an image recipe for your image pipeline. Components are the building blocks that are consumed by an image recipe, for example, packages for installation, security hardening steps, and tests. The selected OS and components make up an image recipe. Components are installed in the order in which they are specified and cannot be reordered after selection.
3. **Output.** Image Builder creates an OS image in the selected output format.
4. **Distribute.** You distribute your image to selected AWS Regions after it passes tests in the image pipeline.

The images that you build from the golden image are in your AWS account. You can configure your image pipeline to produce updated and patched versions of your AMI by entering a build schedule. When the build is complete, you can receive notification via [Amazon Simple Notification Service \(SNS\)](#). In addition to producing a final image, Image Builder generates an image recipe that can be used with existing version control systems and continuous integration/continuous deployment (CI/CD) pipelines for repeatable automation. You can share and create new versions of your image recipe.

14.2 PREREQUISITES

- EC2 Image Builder service linked role
 - Documentation: <https://docs.aws.amazon.com/imagebuilder/latest/userguide/image-builder-service-linked-role.html>
 - Image Builder uses the service-linked role named **AWSServiceRoleForImageBuilder** to allow EC2 Image Builder to access AWS resources on your behalf.
- Configuration requirements
 - Instances used to build images and run tests using Image Builder must have access to the **Systems Manager service**. All build activity is orchestrated by SSM Automation. The SSM Agent will be installed on the source image if it is not already present, and it will be removed before the image is created.
 - Select a public subnet with internet access in advance setting for the image pipeline. The security groups associated with the launched instance (via AWS builder image) must have opened ports 443 and 80.
 - Launch
- AWS IAM
 - The IAM role that you associate with your instance profile must have permissions to run the build and test components included in your image. The following IAM role policies must be attached to the IAM role that is associated with the instance profile: **EC2InstanceProfileForImageBuilder** and **AmazonSSMManagedInstanceCore**.
 - If you configure logging, the instance profile specified in your infrastructure configuration must have `s3:PutObject` permissions for the target bucket (`arn:aws:s3:::BucketName/*`)

14.3 COMPONENTS

14.3.1 Image pipeline

An image pipeline is the automation configuration for building secure OS images on AWS. The Image Builder image pipeline is associated with an image recipe that defines the build, validation, and test phases for an image build lifecycle. An image pipeline can be associated with an infrastructure configuration that defines where your image is built. You can define attributes, such as instance type, subnets, security groups, logging, and other infrastructure-

related configurations. You can also associate your image pipeline with a [distribution configuration](#) to define how you would like to deploy your image.

[VFIE CCH-SAL](#): We will have different image pipelines depending on how many customized images the application needs. At the moment, we create the following images: OFMW_1213 and OFMW_1221. We also use an image pipeline for HAProxy image, see documentation [here](#). Right now, we don't have any schedule for the pipelines, we run them manually. We will be adding automation later on.

14.3.2 Image recipe

An Image Builder image recipe is a document that defines the source image and the components to be applied to the source image to produce the desired configuration for the output image. You can use an image recipe to duplicate builds. Image Builder image recipes can be shared, branched, and edited using the console wizard, the AWS CLI, or the API. You can use image recipes with your version control software to maintain shareable versioned image recipes.

[VFIE CCH-SAL](#): Our source/parent image is PCS hardened AMI for rhel 7 and SSM documents are defined by SinglePoint and PCS. PCS added SSM documents for monitoring (CW Agent) and activating TrendMicro. Singlepoint added SSM documents for the application. Each time we have a new AMI or an update on any SSM document, a new version of the recipe and of the pipeline needs to be created.

14.3.3 Source image

The source image is the selected image and OS used in your image recipe document along with the components. The source image and the component definitions combined produce the desired configuration for the output image.

[VFIE CCH-SAL](#): Our source/parent image is **PCS hardened AMI for rhel 7** (documentation [here](#)). This AMI has already SSM Agent installed. We will need to add automation when new hardened AMI are published.

14.3.4 Build components

Build components are orchestration documents that define a sequence of steps for downloading, installing, and configuring software packages. They also define validation and security hardening steps. A component is defined using a YAML document format (as described in the following Document entry).

VFIE CCH-SAL: We have different SSM documents. For each image pipeline, we will use a SSM document for OFMW base, and a specific SSM document, like OFMW 1213 or OFMW 1221. In addition, PCS included another SSM document to install CW Agent and activate Trend Micro.

14.3.5 Test components

Test components are orchestration documents that define tests to run on software packages. A component is defined using a YAML document format (see the following definition for Document).

VFIE CCH-SAL: Not used at the moment.

14.3.6 Document

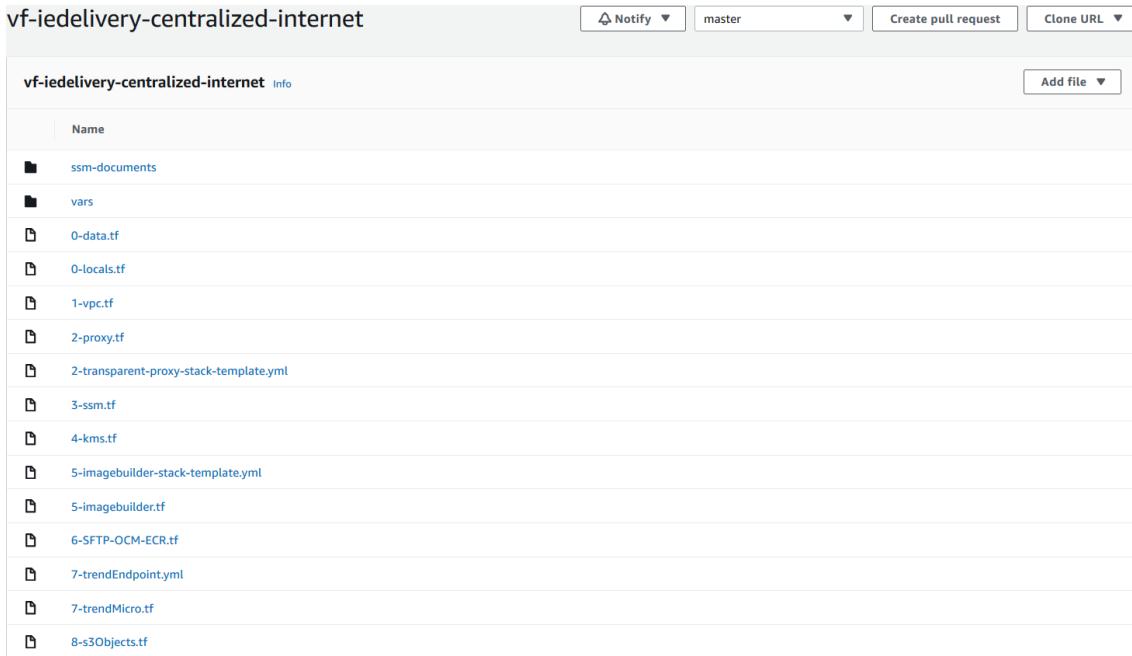
A declarative document that uses the YAML format to list the execution steps for build, validation, and test of an AMI on an instance. The document is input to a configuration management application, which runs locally on an Amazon EC2 instance to execute the document steps.

VFIE CCH-SAL: Each document is defined in our codecommit repository, in terraform. They are stored in a folder, and they are pushed to S3 bucket, via a pipeline (repository pipeline). Terraform will trigger a new object version to s3 if it detects a change in the ssm document. Right now, we will need MANUALLY to update the version of the ssm document, recipe and pipeline in the CF template managed by terraform.

14.4 IMPLEMENTATION

The terraform code for this is stored in the codecommit repository vf-iedelivery-centralized-internet: <https://eu-west-1.console.aws.amazon.com/codesuite/codecommit/repositories/vf-iedelivery-centralized-internet/browse?region=eu-west-1>. The reason of creating here the code is that AWS image builder will create images that will be shared, so it makes sense to create those customized images in our vfie shared services account. In addition, we need internet access, and to avoid problems with transparent proxy in other vpc, we will be using Internet VPC.

Right now, terraform does not support AWS Image builder, as it is a new service, so we will be managing a CloudFormation stack in terraform.



Key files here:

- ssm-documents folder: we will be storing recipe components here. Those components will be pushed to an s3 bucket.
- 5-imagebuilder-stack-template.yml : CF template
- 5-imagebuilder.tf : CF stack
- 8-s3Objects.tf: ssm files to be pushed to S3 bucket
- 0-data.tf: resources that have been already created and are going to be used. Example: subnet with internet access
- 0-locals.tf: variables used.

14.4.1 CLOUDFORMATION STACK

The cloudformation stack will have as **variables**:

- **OFMWBASEDOCUMENT** = SSM document shared for all OFMW domains (created by SinglePoint)
- **OFMW1213DOCUMENT** = SSM document just for OFMW 1213 (created by SinglePoint)
- **OFMW1221DOCUMENT** = SSM document just for OFMW 1221 (created by SinglePoint)
- **PCSSMDOCUMENT** = SSM document created by PCS with Trend Micro activation and AWS CloudWatch Agent
- **haproxyDOCUMENT** = SSM document created by PCS with HAProxy configuration (sfw installation and version update)
- **SubnetId** = Subnets where the image created by AWS EC2 image builder will be launch. **IMPORTANT: it needs internet access!**
- **SecurityGroupId** = Security group that will be attached to the image launched by AWS EC2 image builder. **IMPORTANT: it needs 443 and 80 outbound rules**
- **InstanceProfileName** = IAM instance profile that EC2 image builder will associated with the launched instance when creating new image (see requisites above). **IMPORTANT: SSM, S3 and Image builder access**
- **AMId** = Parent AMI for OFMW 1213 and OFMW 1221. It is a PCS hardened RHEL7 AMI (latest one)

- **HAProxyAMId** = Parent AMI for HAProxy. It is a PCS hardened Amazon Linux 2 AMI (latest one)

For each customized image that we want to create, we will be defining the following resources:

- SNS topic
- Image Builder Component (as many as we need = SSM documents)
- Image Recipe with the components for the image and the source AMI
- Image pipeline with the image recipe and the image infrastructure configuration
- Image infrastructure configuration, where we define: instance profile, instance type, s3 bucket to send the logs, security groups to associate with the image launched by AWS when creating the image, SNS topic and subnet Id (with internet access).
- Image: This resource is not mandatory, but it will remove the need to manually run the pipeline when we deploy a new change in the CF template

Example of the code for OFMW 1213:

```

SNSTopic1213:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: "albamaria.diazfernandez@vodafone.com"
        Protocol: "email"
    TopicName: "VFIEdelivery-AMI-1213"

ImageComponentBase:
  Type: AWS::ImageBuilder::Component
  Properties:
    ChangeDescription: 'initial version'
    Description: 'VFIE delivery OFMW Base OS Nitro SSM DOCUMENT'
    Name: 'VFIE-DELIVERY-OFMW-BASE-OS-NITRO-SSM-DOCUMENT'
    Platform: 'Linux'
    Version: '1.0.1'
    Uri: !Ref OFMWBASEDOCUMENT

ImageComponentPCS:
  Type: AWS::ImageBuilder::Component
  Properties:
    ChangeDescription: 'initial version'
    Description: 'VFIE delivery OFMW PCS trend Micro and CW Agent'
    Name: 'VFIE-DELIVERY-PCS-TRENDMICRO-CW-AGENT-SSM-DOCUMENT'
    Platform: 'Linux'
    Version: '1.0.1'
    Uri: !Ref PCSSSMDOCUMENT

#OFMW1213 COMPONENT
ImageComponent1213:
  Type: AWS::ImageBuilder::Component
  Properties:
    ChangeDescription: 'forth version'
    Description: 'VFIE delivery OFMW 1213 SSM DOCUMENT'
    Name: 'VFIE-DELIVERY-OFMW-1213-BUILD'
    Platform: 'Linux'
    Version: '1.0.5'
    Uri: !Ref OFMW1213DOCUMENT

ImageRecipe1213:
  Type: AWS::ImageBuilder::ImageRecipe
  Properties:
    Components:
      - ComponentArn: !GetAtt ImageComponentBase.Arn
      - ComponentArn: !GetAtt ImageComponent1213.Arn
      - ComponentArn: !GetAtt ImageComponentPCS.Arn
    Description: 'VFIE delivery OFMW Image recipe'
    Name: 'VFIE-DELIVERY-OFMW-1213'
    ParentImage: !Ref AMIid
    Version: '1.0.6'

ImagePipeline1213:
  Type: AWS::ImageBuilder::ImagePipeline
  Properties:
    Description: 'VFIE delivery OFMW 1213 Image pipeline'
    #DistributionConfigurationArn: String
    ImageRecipeArn: !GetAtt ImageRecipe1213.Arn
    #ImageTestsConfiguration:
    #  ImageTestsConfiguration
    InfrastructureConfigurationArn: !GetAtt ImageInfraConfig1213.Arn
    Name: 'VFIE-DELIVERY-OFMW-1213'
    #Schedule:
    #  Schedule

```

```

Status: 'ENABLED'
#Tags:
# Key : Value

ImageInfraConfig1213:
Type: AWS::ImageBuilder::InfrastructureConfiguration
Properties:
Description: 'VFIE delivery OFMW Image 1213 infrastructure config'
InstanceProfileName: !Ref InstanceProfileName
InstanceTypes:
- 't3a.medium'
#KeyPair: String
Logging:
S3Logs:
S3BucketName: 's3-access-logs-vf-iedelivery-267040142128-logs'
S3KeyPrefix: 'OFMW1213logs'
Name: 'VFIE-DELIVERY-OFMW-1213-IMAGE-INFRA'
SecurityGroupIds:
- !Ref SecurityGroupId
- sg-0b85489e5b93501d8
SnsTopicArn: !Ref SNSTopic1213
SubnetId: !Ref SubnetId
#Tags:
# Key : Value
#TerminateInstanceOnFailure: Boolean

Image1213:
Type: AWS::ImageBuilder::Image
Properties:
ImageRecipeArn: !GetAtt ImageRecipe1213.Arn
InfrastructureConfigurationArn: !GetAtt ImageInfraConfig1213.Arn
Tags:
Name : "Image-OMFW-1213"

```

Note: IF ANY CHANGE, UPDATE MANUALLY THE VERSION OF THE COMPONENTS AND IMAGE RECIPE!!

14.4.2 SSM DOCUMENTS

These are the ssm documents used:

- haproxy.yml
- OFMW_1213.yml
- OFMW_1221.yml
- OFMW_base.yml
- PCS_SSM.yml

These files will be pushed to an s3 bucket in ss account. The s3 bucket is defined in the local variables (0-locals.tf):

- s3_bucket = "vf-iedelivery-centralized-internet-vpc-267040142128-123456"
- s3_bucket_URI = "s3://\${local.s3_bucket}"

We can see the terraform code for s3 objects in 8-s3Objects.tf. For each resource, we will be defining the s3 bucket, key for the object, source of the content and the etag, to trigger any changes (update version).

Example:

```
#uploading files to s3 bucket
resource "aws_s3_bucket_object" "PCS-SSM-DOCUMENT" {
  bucket = local.s3_bucket
  key    = "PCS_SSM.yml"
  source = "${path.module}/ssm-documents/PCS_SSM.yml"
  etag   = "${filemd5("${path.module}/ssm-documents/PCS_SSM.yml")}"
#triggered changes
}

#uploading files to s3 bucket
resource "aws_s3_bucket_object" "OFMW-BASE-SSM-DOCUMENT" {
  bucket = local.s3_bucket
  key    = "OFMW_base.yml"
  source = "${path.module}/ssm-documents/OFMW_base.yml"
  etag   = "${filemd5("${path.module}/ssm-documents/OFMW_base.yml")}"
#triggered changes
}
```

14.4.3 AWS CONSOLE RESOURCES

IMAGE COMPONENTS

| Component name | Version | Platform | Type | Description | Date created | Owner | ARN |
|---|---------|----------|-------|---|----------------------|------------------|---|
| VFIE-DELIVERY-haproxy-BUILD | 1.0.5 | Linux | Build | VFIE delivery haproxy SSM DOCUMENT | Jun 23, 2020 4:33 PM | 267040142128 128 | arn:aws:imagebuilder:eu-west-1:267040142128:component/vfie-delivery-haproxy-build/1.0.5/1 |
| VFIE-DELIVERY-OFMW-1213-BUILD | 1.0.4 | Linux | Build | VFIE delivery OFMW 1213 SSM DOCUMENT | Jul 14, 2020 5:57 PM | 267040142128 128 | arn:aws:imagebuilder:eu-west-1:267040142128:component/vfie-delivery-ofmw-1213-build/1.0.4/1 |
| VFIE-DELIVERY-OFMW-1221-BUILD | 1.0.3 | Linux | Build | VFIE delivery OFMW 1221 SSM DOCUMENT | Jul 14, 2020 5:57 PM | 267040142128 128 | arn:aws:imagebuilder:eu-west-1:267040142128:component/vfie-delivery-ofmw-1221-build/1.0.3/1 |
| VFIE-DELIVERY-OFMW-BASE-OS-NITRO-SSM-DOCUMENT | 1.0.1 | Linux | Build | VFIE delivery OFMW Base OS Nitro SSM DOCUMENT | Jul 14, 2020 5:57 PM | 267040142128 128 | arn:aws:imagebuilder:eu-west-1:267040142128:component/vfie-delivery-ofmw-base-os-nitro-ssm-document/1.0.1/1 |

IMAGE RECIPES

Recipes

A recipe specifies the activities needed to make changes to the source image. A recipe cannot be modified once it is created.

| Created by me | Any OS | | | | | |
|-------------------------|--------|-------|-----------------------|----------------------|--------------|--|
| VFIE-DELIVERY-haproxy | 1.0.5 | Linux | ami-01726b7eb435f48b5 | Jun 23, 2020 4:34 PM | 267040142128 | arn:aws:imagebuilder:eu-west-1:267040142128:image-recipe/vfie-delivery-haproxy/1.0.5 |
| VFIE-DELIVERY-OFMW-1213 | 1.0.5 | Linux | ami-0100fb121d185f6b1 | Jul 14, 2020 5:58 PM | 267040142128 | arn:aws:imagebuilder:eu-west-1:267040142128:image-recipe/vfie-delivery-ofmw-1213/1.0.5 |
| VFIE-DELIVERY-OFMW-1221 | 1.0.4 | Linux | ami-0100fb121d185f6b1 | Jul 14, 2020 5:58 PM | 267040142128 | arn:aws:imagebuilder:eu-west-1:267040142128:image-recipe/vfie-delivery-ofmw-1221/1.0.4 |

IMAGE PIPELINES

Image pipelines

The image pipeline in Image Builder defines all aspects of the process to customize images. It consists of the image recipe, infrastructure configuration, distribution, and test settings.

| Date created | Version | Status | Last run | ARN |
|----------------------|---------|---------|----------|--|
| Jun 2, 2020 9:17 AM | 1.0.5 | Enabled | - | arn:aws:imagebuilder:eu-west-1:267040142128:image-pipeline/vfie-delivery-haproxy |
| May 26, 2020 3:28 PM | 1.0.5 | Enabled | - | arn:aws:imagebuilder:eu-west-1:267040142128:image-pipeline/vfie-delivery-ofmw-1213 |
| May 27, 2020 1:45 PM | 1.0.4 | Enabled | - | arn:aws:imagebuilder:eu-west-1:267040142128:image-pipeline/vfie-delivery-ofmw-1221 |

IMAGE CREATED

VFIE-DELIVERY-OFMW-1221

Summary

| | | | |
|---|--------------------------------------|----------------------------------|---------------|
| Description VFIE delivery OFMW Image recipe | Date created May 27, 2020 1:45 PM | Last run May 27, 2020 2:33 PM | Next run - |
| IAM role SSM-E2-CENTRALIZED-INTERNET-IAM-INSTANCE-PROFILE-vfie-delivery-PROD | Build schedule | Status Enabled | |

Output images

Output images of the pipeline

| Version | Date created | Status | Reason for failure | ARN |
|---------|----------------------|-----------|--------------------|---|
| 1.0.0/1 | May 27, 2020 2:33 PM | Available | - | arn:aws:imagebuilder:eu-west-1:267040142128:image/vfie-delivery-ofmw-1221/1.0.0/1 |

14.5 HOW TO: NEW AMI

- In shared services account:
 - When a new hardened AMI is released, it needs to be copied (and ebs volume encrypted) to eu-west-1 region. This now is done MANUALLY.

- Image pipeline needs to be run. In terraform, we have a data resource for hardened ami that will read the latest one. However, we will need to update the VERSION of the AWS EC2 image builder MANUALLY. For this case, if there is no modifications for SSM documents (image components, we just need to update version of image recipe. In case of modification for image component, update the file in terraform, and in the cloudformation template, update the version of that component and the one for the image recipe need to be updated.
- Run vf-iedelivery-centralized-internet pipeline. It will take around 30 min for the image pipeline to create an image.
- In tenant accounts:
 - Once we have in ss account the new image created, terraform will add launch permissions to the most recent AMI and ebs volume permissions for each ebs volume. If the pipeline reaches the timeout (or the image creation fails) before the new image are created, MANUALLY needs to be run again (vf-iedelivery-centralized-internet), so the part of adding permissions to ami and ebs are executed.
 - Once in tenant account we have access to new AMI, we will need to copy (encrypting ebs volume) to eu-west-region-1. This is done MANUALLY.
 - In our repository (vf-iedelivery-infrastructure), we have a terraform data resource to read the latest AMI. If we release a change in the corresponding codepipeline (for vf-iedelivery-mgmt-ENVIRONMENT), we will update that AMI. We should ignore AMI changes until we want to upgrade it.
- EBS VOLUMES:
 - <https://aws.amazon.com/blogs/security/how-to-share-encrypted-amis-across-accounts-to-launch-encrypted-ec2-instances/> IF ANY EBS VOLUME IS ENCRYPTED WITH DEFAULT EBS KEY, IT NEEDS TO BE COPIED WITH A CUSTOM CKM. THEN, FOLLOW THIS DOCUMENTATION.

14.5.1 PERMISSIONS TO NEW AMI

```

-----data specific ami
data "aws_ami" "ofmw1213" {
  most_recent      = true
  owners           = ["self"]
  filter {
    name   = "name"
    values = ["VFIE-DELIVERY-OFMW-1213 *"]
  }
}
data "aws_ami" "ofmw1221" {
  most_recent      = true
  owners           = ["self"]
  filter {
    name   = "name"
    values = ["VFIE-DELIVERY-OFMW-1221 *"]
  }
}

-----ami launch permissions
resource "aws_ami_launch_permission" "ofmw1213" {
  count = length(var.ram_principals)
  image_id  = data.aws_ami.ofmw1213.id #latest one
  account_id = var.ram_principals[count.index]
  lifecycle {
    ignore_changes = [image_id]
  }
}
resource "aws_ami_launch_permission" "ofmw1221" {
  count = length(var.ram_principals)
  image_id  = data.aws_ami.ofmw1221.id #latest one
  account_id = var.ram_principals[count.index]
  lifecycle {
    ignore_changes = [image_id]
  }
}

-----ebs volume snapshot permissions for root volume
resource "aws_snapshot_create_volume_permission" "ofmw1213" {
  count = length(var.ram_principals)
  snapshot_id = data.aws_ami.ofmw1213.root_snapshot_id
  account_id = var.ram_principals[count.index]
  lifecycle {
    ignore_changes = [snapshot_id]
  }
}
resource "aws_snapshot_create_volume_permission" "ofmw1221" {
  count = length(var.ram_principals)
  snapshot_id = data.aws_ami.ofmw1221.root_snapshot_id
  account_id = var.ram_principals[count.index]
  lifecycle {
    ignore_changes = [snapshot_id]
  }
}

-----ebs volume snapshot permissions forebs volume 1
resource "aws_snapshot_create_volume_permission" "ofmw1213_1" {
  count = length(var.ram_principals)
  snapshot_id =
  element(data.aws_ami.ofmw1213.block_device_mappings.*.ebs.snapshot_id,1)
  account_id = var.ram_principals[count.index]
  lifecycle {
    ignore_changes = [snapshot_id]
  }
}
resource "aws_snapshot_create_volume_permission" "ofmw1221_1" {
  count = length(var.ram_principals)
  snapshot_id =
  element(data.aws_ami.ofmw1221.block_device_mappings.*.ebs.snapshot_id,1)
}

```

```
account_id = var.ram_principals[count.index]
lifecycle {
  ignore_changes = [snapshot_id]
}
}
#----ebs volume snapshot permissions forebs volume 2
resource "aws_snapshot_create_volume_permission" "ofmw1213_2" {
  count = length(var.ram_principals)
  snapshot_id =
element(data.aws_ami.ofmw1213.block_device_mappings.*.ebs.snapshot_id,2)
  account_id = var.ram_principals[count.index]
  lifecycle {
    ignore_changes = [snapshot_id]
  }
}
resource "aws_snapshot_create_volume_permission" "ofmw1221_2" {
  count = length(var.ram_principals)
  snapshot_id =
element(data.aws_ami.ofmw1221.block_device_mappings.*.ebs.snapshot_id,2)
  account_id = var.ram_principals[count.index]
  lifecycle {
    ignore_changes = [snapshot_id]
  }
}
```

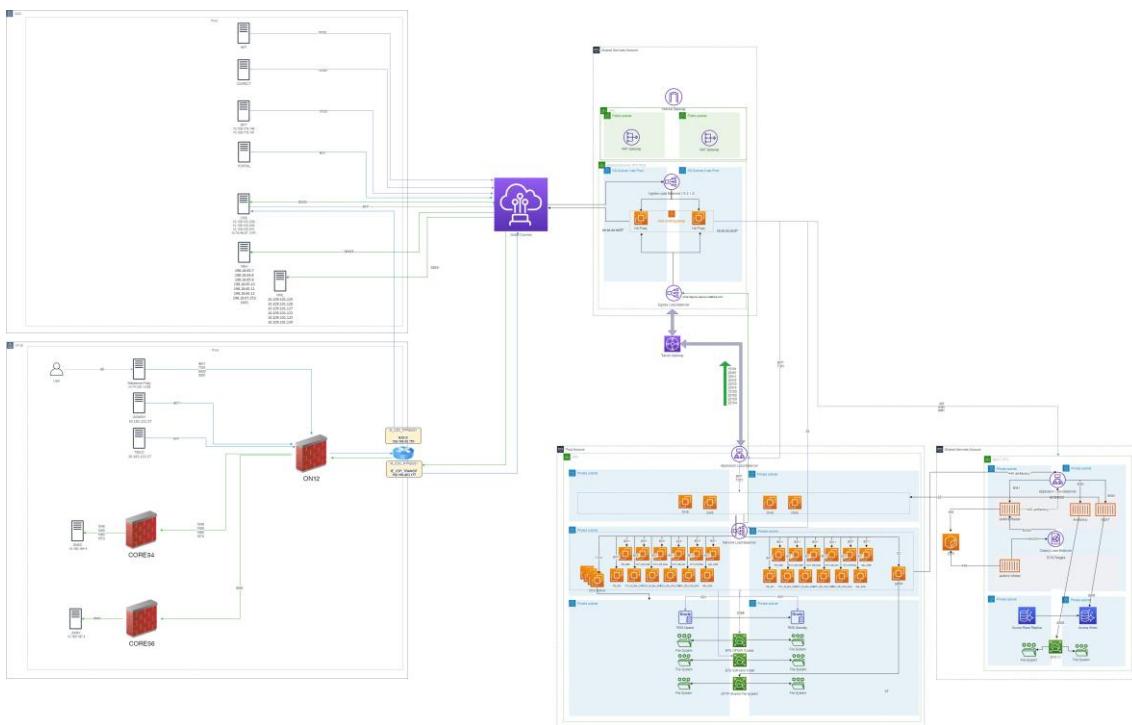
15 08 VFIE OVERALL NETWORK DESIGN

The Overall Network Design covers the integration with the VFIE and GDC networks from AWS.

- [Network Diagram](#)
- [Advertised Subnets](#)
- [GDC & VFIE Integration Points](#)
 - [TNSNAMES.ORA \(PROD\)](#)
 - [GUI URL List](#)

15.1 Network Diagram

PROD



15.2 Advertised Subnets

IE_CSP_TRANSIT

The IE_CSP_TRANSIT VRF is a direct connection between VFIE and AWS, bypassing the GDC Firewalls and Routers. Once a subnet is advertised on this VRF via the M Routers (m_cdc_rtr20 & 21), the traffic will be routed directly to AWS.

- 10.163.78.0/23
- 10.151.4.0/24
- 10.162.114.0/24
- 10.74.120.112/28
- 10.162.122.0/24

- 10.162.111.0/24
- 10.163.187.0/24
- 10.163.184.0/24
- 10.162.229.0/24
- 10.162.66.0/24
- 10.180.54.0/24
- 10.162.124.0/24

To ensure that VFIE Traffic is routed correctly to the right interface on the M routers for IE_CSP_TRANSIT, routing has been added on ON12 for the associated AWS subnets as below

| PURPOSE | Destination | Mask | Next Hop IP | Next Hop Desc | Next Hop Owner | Install on |
|---------------------------|------------------|------|----------------|----------------|----------------|------------|
| VFIE AWS Dedicated Subnet | 198.19.220.0 /24 | | 192.168.50.170 | m_cdc_rtr20/21 | NSU | ON12 |

15.3 GDC & VFIE Integration Points

15.3.1 TNSNAMES.ORA (PROD)

```

prod-cchsal =
(DESCRIPTION=
(CONNECT_DATA=(SERVICE_NAME=CCHSAL))
(SOURCE_ROUTE=yes)
(ADDRESS=(PROTOCOL=tcp)(HOST=cch-sal-
utilities.prod.ieaws.vodafone.com)(PORT=10305))
(ADDRESS=(PROTOCOL=tcp)(HOST=cch-sal-
db.prod.ieaws.vodafone.com)(PORT=2484))
)

prod-cchsal-logger =
(DESCRIPTION=
(CONNECT_DATA=(SERVICE_NAME=LOGGER))
(SOURCE_ROUTE=yes)
(ADDRESS=(PROTOCOL=tcp)(HOST=cch-sal-
utilities.prod.ieaws.vodafone.com)(PORT=10305))
(ADDRESS=(PROTOCOL=tcp)(HOST=logger-
db.prod.ieaws.vodafone.com)(PORT=2484))
)

```

15.3.2 GUI URL List

| URL | Description |
|---|-------------------|
| https://jenkins.prod.ieaws.vodafone.com:10301 | Jenkins |
| https://artifactory.prod.ieaws.vodafone.com:10302 | Jfrog Artifactory |
| https://myst.prod.ieaws.vodafone.com:10303/console | MyST |
| https://pb-osb-lb-as.prod.ieaws.vodafone.com:7001/console | PB_OSB |

| | |
|---|----------------|
| https://pb-odi-lb-as.prod.ieaws.vodafone.com:7001/console | PB_ODI |
| https://cch-ib-sal-osb-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_IB_SAL_OSB |
| https://cch-ib-sal-soa-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_IB_SAL_SOA |
| https://cch-ob-ums-osb-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_OB_UMS_OSB |
| https://cch-ob-ums-soa-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_OB_UMS_SOA |
| https://cch-ob-osb-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_OB_OSB |
| https://cch-ob-soa-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_OB_SOA |
| https://cch-ob-odi-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_OB_ODI |
| https://cch-portal-lb-as.prod.ieaws.vodafone.com:7001/console | CCH_PORTAL |
| https://oal-osb-lb-as.prod.ieaws.vodafone.com:7001/console | OAL_OSB |
| https://oal-soa-lb-as.prod.ieaws.vodafone.com:7001/console | OAL_SOA |

16 09 VFIE SQUID PROXY

The Squid Proxy sits in the Internet Gateway and controls which domains can be accessed on the internet via a whitelist. Below is a list of the current domains accessible.

| Domain | Status |
|---|-------------|
| .amazon.com | Deployed |
| .vodafone.com | Deployed |
| .amazonaws.com | Deployed |
| .bintray.io | Deployed |
| .redhat.com | Deployed |
| .cdn.amazonlinux.com | Deployed |
| .download.mono-project.com | Deployed |
| .mirrors.fedoraproject.org | Deployed |
| .dl.yarnpkg.com | Deployed |
| .qualys.eu | Deployed |
| .hana.ondemand.com | Deployed |
| .mft.aon.com | Deployed |
| .zellis.com | Deployed |
| .westeurope.cloudapp.azure.com | Deployed |
| .maven.org | Deployed |
| .adobe.com | Deployed |
| .apache.org | Deployed |
| .google.com | Deployed |
| .amazoncognito.com | Deployed |
| .omniture.com | Deployed |
| .atomz.com | Deployed |
| .npmjs.org | Deployed |
| .nodejs.org | Deployed |
| .github.com | Deployed |
| .googleapis.com | IN PROGRESS |

Any additional domains that need to be added to the whitelist should be requested either via Service Catalogue or via an ongoing project.

17 10 VFIE Network Integration

Below are the existing integration points between applications deployed in AWS and VFIE/VCI based systems

| S | D | RTPROD | PRD2 | PRD1 | SIT4 | SIT3 | SIT2 | SIT1 | DEV1 | DEV2 |
|--------------------------------|---|--|--|--|--|---|--|--|---|---|
| ou es rc ti e n | es oy up ate tive on | | | | | | | | | |
| A | VV CS d oc se O S B | esb.pr od.dox .equin ox.vod afone.i nal.vo prod. hapro xy.iea ws.vo dafon e.com (hapr oxy) | esb.ap prd2 .equin ox.vf- ie.inter nal.vo dafone prd2.h aprox y.ieaw s.vod afone. com (h aproxy) | esb.ap prd1 .equin ox.vf- ie.inter nal.vo dafone prd1.h aprox y.ieaw s.vod afone. com (h aproxy) | eesbb vr.dc- equin ox.vf- de sit4.h aprox y.ieaw afone. com (h aproxy) | eesbby r.dc- dublin. de sit3.ha prox.y.i eaws.v afon e.com (h aproxy) | eesbbv r.dc- dublin.d e sit2.ha proxy.i eaws.v odafon e.com (h aproxy) | eesbay r.dc- dublin.d e sit1.ha proxy.i eaws.v odafon e.com (h aproxy) | eesbb r.dc- dublin. de dev1. proxy.i eaws.v odafon e.com (h aproxy) | eesbb r.dc- dublin. de dev2. proxy.i eaws.v odafon e.com (h aproxy) |
| P | VV CS tal A E M Di sp o at ch er | portal - dispat cher.p rod.ie aws.v odafo ne.co m | portal - dispat cher.p rd2.ie aws.v odafo ne.co m | portal - dispat cher.p rd1.ie aws.v odafo ne.co m | portal - dispat cher.s it4.iea ws.vo dafon e.com | portal-dispatch her.sit 3.ieaw s.vodaf one.co m | portal-dispatch her.sit 2.ieaw s.vodaf one.co m | portal-dispatch her.sit 1.ieaw s.vodaf one.co m | 8 N/A - 4 Dev 4 CIAM 3 hosted in AWS | N/A - Dev CIAM hosted in AWS |

| | | | | | | | | | | | | |
|----|--------|----------|-------------------------|---------------------------|---------------------------|---|--|--|--|--|--|---|
| C | A | V | oal- osb.pr | 8 oal- 0 osb.pr | 8 oal- 0 osb.pr | 8 ie2135 0 yr.dc- 1 d2.iea | 7 iesalcvr 4 dc- 1 d1.iea | 7 iesalmv 4 dc- 1 dublin. | 7 iesalavr 4 dc- 1 dublin.d | 7 iesesbb 4 vr.dc- 1 dublin.d | 7 iesesbb 4 vr.dc- 1 dublin. | 7 |
| C | S | H | ws.vo dafon e.com | 1 ws.vo dafone .com | 1 ws.vo dafone .com | 1 de sit4.h aprox y.ieaw s.vod afone. com (| 3 e sit3.ha proxy.i eaws.v odafon e.com (| 3 e sit2.ha proxy.i eaws.v odafon e.com (| 3 e sit1.ha proxy.i eaws.v odafon e.com (| 1 dev1. aproxy ws.vo dafon (hapr oxy) | 1 dev2. aproxy ws.vo dafon (hapr oxy) | 1 |
| A | L | O | | | | | | | | | | |
| A | L | O | | | | | | | | | | |
| S | B | | | | | | | | | | | |
| F | V | 89.19. | 789.19. | 789.19. | 789.19. | 789.19.7 | 789.19.7 | 789.19.7 | 789.19. | 789.19. | 789.19. | 7 |
| S | F | 69.17 | 071.65 | 071.65 | 071.65 | 01.65 | 01.65 | 01.65 | 071.65 | 071.65 | 071.65 | 0 |
| L | I | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | E | 4 | prd2.h aprox | 4 prd1.h aprox | 4 sit4.h aprox | 4 sit3.ha proxy.i | 4 sit2.ha proxy.i | 4 sit1.ha proxy.i | 4 dev1. aproxy | 4 dev2. aproxy | 4 dev1. aproxy | 4 |
| a | t | N | prod. aproxy | 1 y.ieaw | 1 y.ieaw | 1 y.ieaw | 1 eaws.v | 1 eaws.v | 1 xy.iea | 1 xy.iea | 1 xy.iea | 1 |
| e | w | e | xy.iea | 1 s.vod | 1 s.vod | 1 s.vod | 1 odafon | 1 odafon | 1 ws.vo | 1 ws.vo | 1 ws.vo | 1 |
| w | t | w | ws.vo | 0 afone. | 0 afone. | 0 afone. | 0 e.com (| 0 e.com (| 0 dafon | 0 dafon | 0 dafon | 0 |
| ay | w | w | dafon | 0 com (| 0 com (| 0 com (| 0 haprox | 0 haprox | 0 e.com | 0 e.com | 0 e.com | 0 |
| o | r | r | e.com | 1 hapro xy) | 1 hapro xy) | 1 hapro xy) | 1 y) | 1 y) | 1 (hapr oxy) | 1 (hapr oxy) | 1 (hapr oxy) | 1 |
| r | k | | | | | | | | | | | |
| U | V | T | ieufoa | 3 ieufoa | 3 ieufoa | 3 IEOR | 3 ieorn2- | 3 ieorn2- | 3 n/a - | n/a - | n/a - | n |
| F | C | C | b- | 3 b- | 3 b- | 3 N1- | 3 scan.dc | 3 scan.dc | 3 basket | / basket | / basket | / |
| E | I | F | scan.d | 0 scan.d | 0 scan.d | 0 SCAN | 0 - | 0 - | 0 will be | a will be | a will be | a |
| D | N | c- | 0 c- | 0 c- | 0 010.106 | 0 010.106 | 0 dublin.d | 0 dublin.d | 0 stored | stored | stored | |
| B | e | dublin. | 0 dublin. | 0 dublin. | 0 0.184.[3 | 0 e | 0 e | 0 e | 0 locally | locally | locally | |
| (P | t | de | 3 de | 3 de | 3 4,35,3 | 3 IEORN | 3 IEORN | 3 IEORN | 3 Portal | Portal | Portal | . |
| ro | w | ce | IEUF | 3 IEUFO | 3 IEUFO | 3 [6] | 3 3HR- | 3 3HR- | 3 will | will | will | |
| ss | r | o | OAHR | 0 AHR- | 0 AHR- | 0 IEOR | 0 VIP - | 0 VIP - | 0 not be | not be | not be | |
| C | k | -VIP | -VIP | -VIP | -VIP | 0 N1HR- | 0 10.106. | 0 10.106. | 0 able to | able to | able to | |
| o | 9.101. | 10.10 | 10.109 | 10.109 | 10.109 | 1 VIP - | 1 184.39 | 1 184.39 | 1 recove | recove | recove | |
| nt | 13 | 2.101.1 | 2.101.1 | 2.101.1 | 2.101.1 | 2 10.106 | 3 IEORN | 3 IEORN | 3 r | r | r | |
| in | 13 | 2.3 | 2.3 | 2.3 | 2.3 | 2 184.3 | 3 4HR- | 3 4HR- | 3 basket | basket | basket | |
| ui | IEUF | 2 IEUFO | 2 IEUFO | 2 IEUFO | 2 [2 | 3 2 | 3 VIP - | 3 VIP - | 3 from | from | from | |
| ty | OBHR | 0 BHR- | 0 BHR- | 0 BHR- | 0 BHR- | 0 IEOR | 0 10.106. | 0 10.106. | 0 UFE | UFE | UFE | |
| D | -VIP | -VIP | -VIP | -VIP | -VIP | 0 N2HR- | 1 184.40 | 1 184.40 | 1 184.40 | 5 | 5 | |
| B) | 10.10 | 10.109 | 10.109 | 10.109 | 10.109 | 1 VIP - | 2 sit3.ha | 2 sit3.ha | 2 | | | |
| | 9.101. | 2.101.1 | 2.101.1 | 2.101.1 | 2.101.1 | 2 10.106 | 2 proxy.i | 2 proxy.i | 2 | | | |
| | 14 | 24 | 24 | 24 | 24 | 2 184.3 | 2 eaws.v | 2 eaws.v | 2 | | | |
| | prod. | 1 prd2.h | 1 prd1.h | 1 prd1.h | 1 prd1.h | 13 | 1 odafon | 1 odafon | 1 | | | |
| | hapro | 5 approx | 5 approx | 5 approx | 5 approx | 5 sit4.h | 4 haprox | 4 haprox | 0 | | | |
| | xy.iea | y.ieaw | y.ieaw | y.ieaw | y.ieaw | 2 y) | 2 y) | 2 y) | 4 | | | |
| | ws.vo | s.vod | s.vod | s.vod | s.vod | 2 y) | 2 y) | 2 y) | 2 | | | |
| | dafon | afone. | afone. | afone. | afone. | 2 afone. | 2 afone. | 2 afone. | 2 | | | |
| | e.com | com (| com (| com (| com (| 1 com (| 1 com (| 1 com (| 1 | | | |

| | | (haproxy) | haproxy) | haproxy) | haproxy) | 0 | 0 | 0 | 0 | | |
|----|----|-------------------------|----------------------------|---------------------------|---------------------------|--------------------------|-----------------------------|-----------------------------|--------------------------|---------------------------|---------------------------|
| U | VV | app.pr | app.pr | app.pr | app.uf | app.siti | app.siti | app.siti | app.sit | app.sit | app.sit |
| F | CS | od.ufe. | d2.ufe. | d1.ufe. | e.sit4. | nt3.ufe. | nt2.ufe. | nt1.ufe. | int3.uf | int3.uf | int3.uf |
| E | I | .equin | equino | equino | equino | equinox | equinox | equinox | e.equi | e.equi | e.equi |
| (O | N | ox.vod | x.voda | x.voda | x.vf- | vodafo | vodafo | vodafo | nox.vo | nox.vo | nox.vo |
| m | e | afone.i | fone.ie | fone.ie | ie.inter | ne.ie | ne.ie | ne.ie | dafon | dafon | dafon |
| n | i | t.e | 3prd2.h | 3prd1.h | 3nal.vo | 3sit3.ha | 3sit2.ha | 3sit1.ha | 1e.ie | 3e.ie | 3e.ie |
| c | h | w.an | prod. | aprox | aprox | dafone | proxy.i | proxy.i | dev1. | dev2. | dev2. |
| a | n | ne.r | aproxy | y.ieaw | y.ieaw | .com | 0eaws.v | 0eaws.v | 0eaws.v | 0aproxy | 0aproxy |
| I | C | k | xy.iae | 1s.vod | 1s.vod | 1sit4.h | 1odafon | 1odafon | 1xy.iae | 1xy.iae | 1xy.iae |
| C | at | al | ws.vo | afone.com | afone.com | aprox | haprox | haprox | dafon | dafon | dafon |
| a | g | log | dafone.com | (haproxy) | (haproxy) | y) | y) | y) | .com | (haproxy) | (haproxy) |
| A | I | T | *.ama | 4*.ama | 4*.ama | 4*.ama | 4*.amaz | 4*.amaz | 4*.ama | 4*.ama | 4*.ama |
| W | n | C | zonco | 4zonco | 4zonco | 4zonco | 4oncogni | 4oncogni | 4zonco | 4zonco | 4zonco |
| S | t | F | gnito.c | 3gnito.c | 3gnito.c | 3gnito.c | 3to.com | 3to.com | 3to.com | 3gnito.c | 3gnito.c |
| C | e | o | om | om | om | om | *.amaz | *.amaz | *.ama | *.ama | *.ama |
| o | r | g | *.ama | *.ama | *.ama | *.ama | *.onaws.com | *.onaws.com | *.ama | *.ama | *.ama |
| n | i | e | zonaw | zonaw | zonaw | zonaw | *.onaws.com | *.clearm | *.clearm | zonaw | zonaw |
| t | o | t | s.com | s.com | s.com | s.com | *.clearm | *.clearm | *.clearm | s.com | s.com |
| G | a | G | *clear | *clear | *clear | *clear | *.obile.ie | *.obile.ie | *.clear | *.clear | *.clear |
| a | t | e | mobile | mobile | mobile | mobile | .ie | .ie | .ie | .ie | .ie |
| w | e | w | | | | | | | | | |
| a | w | y | | | | | | | | | |
| G | I | V | www.g | 4www.g | 4www.g | 4www.g | 4www.g | 4www.g | 4www.g | 4www.g | 4www.g |
| o | n | S | oogle | 4oogle | 4oogle | 4oogle | 4oogle | 4oogle | 4oogle. | 4oogle. | 4oogle. |
| o | t | . | .com | 3.com | 3.com | 3.com | 3.com | 3.com | 3.com | 3.com | 3.com |

| | | | | | | | | | | | | | | | | | | | |
|---|---|-----|-------------------------|------------------------|-------------------------|------------------------|-------------------------|------------------------|-------------------------|------------------------|--------------------------|------------------------|--------------------------|------------------------|--------------------------|------------------------|-------------------------|------------------------|-------------------------|
| g | i | e | r | n | A | n | P | I | e | n | t | G | a | t | e | w | a | y | |
| A | I | V | my.om | 4 | my.om | 4 | my.om | 4 | my.om | 4 | my.om | 4 | my.om |
| d | n | S | niture. | 4 | niture.c | 4 | niture.c | 4 | niture.c | 4 | niture. | 4 | niture. |
| o | t | com | com | 3 | om | 3 | om | 3 | om | 3 | om | 3 | om |
| b | e | Cl | api3.o | | api3.o | | api3.o | | api3.o | | api3.o | | api3.o | | api3.o | | api3.o | | api3.o |
| o | n | r | mnitur | | mnitur | | mnitur | | mnitur | | mniture | | mniture | | mniture | | mniture | | mniture |
| u | e | e | e.com | | e.com | | e.com | | e.com | | .com | | .com | | .com | | .com | | .com |
| d | t | (A | G | | | | | | | | | | | | | | | | |
| b | e | S | w | | | | | | | | | | | | | | | | |
| e | a | a | rc | y | | | | | | | | | | | | | | | |
| h | & | h | & | Pr | | | | | | | | | | | | | | | |
| P | r | o | m | ot | e) | | | | | | | | | | | | | | |
| G | r | I | V | eu2.a | 4 | eu2- | 4 | eu2- | 4 | eu2- | 4 | eu2- | 4 | eu2- | 4 | eu2- | 4 | eu2- | |
| r | o | n | S | pi.vod | 4 | stagin | 4 | stagin | 4 | stagin | 4 | stagin | 4 | stagin | 4 | stagin | 4 | stagin | |
| u | t | u | afone. | 3 | afone. | 3 | afone. | 3 | afone. | 3 | afone. | 3 | afone. |
| p | e | e | com | | com | | com | | com | | com | | com | | com | | com | | com |
| D | r | xL | n | C | l | e | u | s | t | t | G | r | a | t | | | | | |

| | | | | | | | | | | | | | | |
|---|---|---|---|--|--|--|--|--|--|--|--|--|--|--|
| e w a y | | | | | | | | | | | | | | |
| V I V n G o u p A P X | api.de veloped tr.voda fone.c om ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | 4apist 4aging 3ref.d evelo per.v odafo ne.co m | |
| C C H S A L I B S A L O S B | AVcch- Wib- S sal- osb.p rod.i eaws. vodaf one.c om | 8cch- 0ib- 1sal- 1osb.p rd2.i eaws. vodaf one.c om | 8cch- 0ib- 1sal- 1osb.p rd1.i eaws. vodaf one.c om | 8le2135 0yr.dc- 1dublin. 1de sit4.h aprox y.ieaw s.vod afone. com (h aproxy) | 9iesal 0cvr.d 8c- 0dubli n.de 1sit3. 0apro 1xy.ie 0aws.v 2odafo ne.co m (ha proxy) | 9iesal 4mvr.d 4c- 3dubli n.de 1sit2. 0apro 1xy.ie 0aws.v 2odafo ne.co m (ha proxy) | 9iesal 4avr.d 4c- 3dubli n.de 1sit1. 0apro 1xy.ie 0aws.v 2odafo ne.co m (ha proxy) | 9iesalc 0vr.dc- 8dublin. 0de 1dev1. 1sit1. 0apro 1xy.ie 0aws.v 2odafo ne.co m (ha proxy) | 9iesalc 4vr.dc- 4dublin. 3de 1dev2. 1hapro 0xy.iea 1ws.vo 0dafon 2e.com (hap oxy) | 9iesalc 4vr.dc- 4dublin. 3de 1dev2. 1hapro 0xy.iea 1ws.vo 0dafon 2e.com (hap oxy) | 9iesalc 4vr.dc- 4dublin. 3de 1dev2. 1hapro 0xy.iea 1ws.vo 0dafon 2e.com (hap oxy) | 9iesalc 4vr.dc- 4dublin. 3de 1dev2. 1hapro 0xy.iea 1ws.vo 0dafon 2e.com (hap oxy) | 9iesalc 4vr.dc- 4dublin. 3de 1dev2. 1hapro 0xy.iea 1ws.vo 0dafon 2e.com (hap oxy) | 9iesalc 4vr.dc- 4dublin. 3de 1dev2. 1hapro 0xy.iea 1ws.vo 0dafon 2e.com (hap oxy) |
| V es ta A Pl n e G a t e w a y | Vestaa nSuth.vo tdafon e.ie vestat opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafone 4.ie 3uat- 3uat- 3opup.v odafon e.ie | 4uat- 4vestaa 3uth.vo 4dafone 4.ie 3uat- 3vestat 3opup.v odafon e.ie | 4uat- 4vestaa 3uth.vo 4fone.ie 4uat- 3vestato 3vestat 3opup.v odafon. ie | 4uat- 4vestaa 3th.voda 3fone.ie 4uat- 3vestato 3vestato 3pup.vo dafone. ie | 4uat- 4vestaa 3th.voda 4fone.ie 4uat- 3vestato 3vestato 3pup.vo dafone. ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie | 4uat- 4vestaa 3uth.vo 4dafon 4.e.ie 4uat- 3uat- 3vestat 3opup.v odafo ne.ie |

18 11 VFIE - Frankfurt Transit Gateway Solution

- [Overview](#)
- [Network Design](#)
 - [SIT3 Design \(encompasses PROD\)](#)
 - [Overall Design](#)
 - [Network Design](#)
- [AWS Design](#)
 - [Network Load Balancer](#)
 - [Dublin Shared Services NLB](#)
 - [Frankfurt Shared Services NLB](#)
 - [Target Groups](#)
 - [Dublin Target Groups](#)
 - [Frankfurt Target Groups](#)
 - [Transit Gateway](#)
 - [Endpoints](#)
- [Supporting Documentation](#)
 - [DXL / TAAS Roadmap](#)
 - [TAAS Documentation](#)
 - [Overview](#)
 - [Multi-region implementation](#)
 - [CAAS NEL mServices](#)

18.1 Overview

To allow for VFIE services to be consumed by applications hosted in the Frankfurt AWS region, a Transit Gateway solution has been built between AWS Frankfurt and AWS Dublin, utilising the VFIE Shared Services account. The Frankfurt Transit Gateway will reside in the VFIE Shared Services AWS account **vf-iedelivery-prod-ss (267040142128)** and will cover all environments.

This solution is being used for the following services:

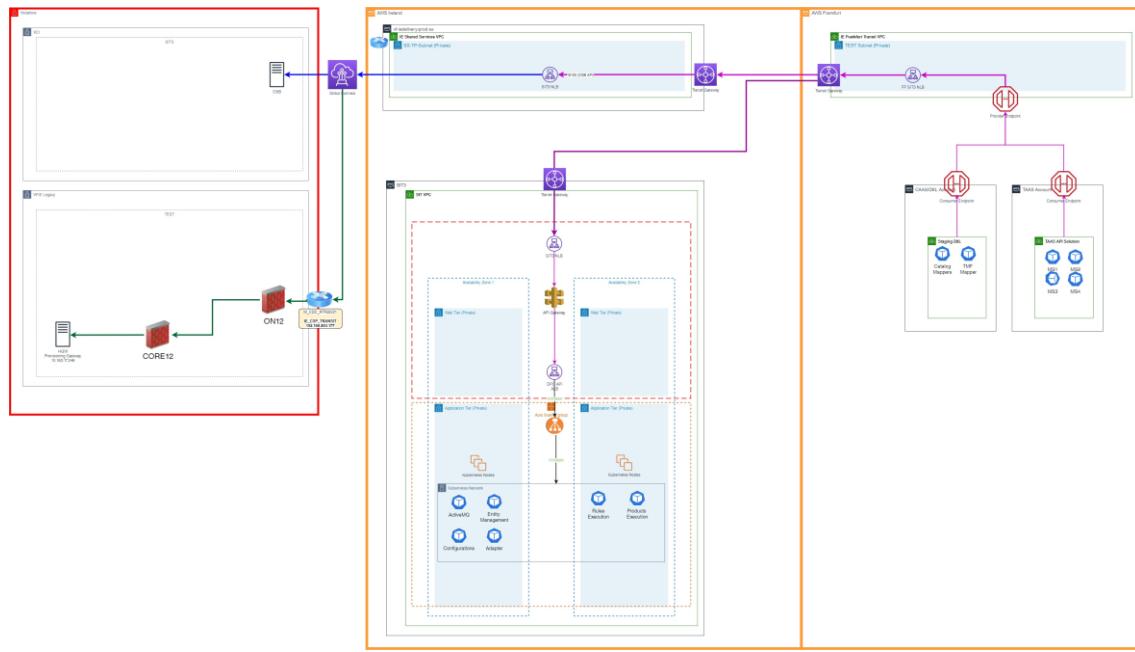
- DXL
 - DXL → Digital Product Catalogue (AWS)
- TAAS
 - TaaS → Amdocs OSB (VCI Network)
 - TaaS → Server-Side-Rendering (SSR) (AWS)
- CAAS NEL mServices
 - CAAS → HGW (VFIE Network)

18.2 Network Design

[DRAWIO Design File](#)

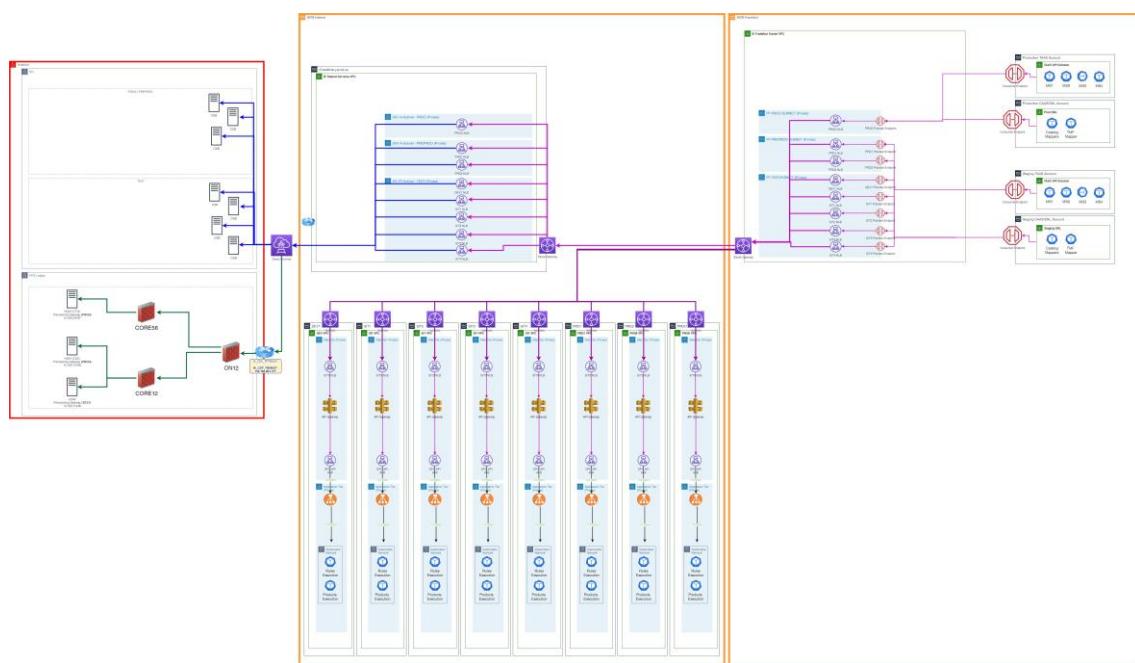
18.2.1 SIT3 Design (encompasses PROD)

This design covers a single environment so that the simplified architecture can be viewed and the flows traced



18.2.2 Overall Design

This design covers the overall solution implemented for every environment



18.2.3 Network Design

The following is the Subnet Design for the new **IE Frankfurt Transit VPC 10.190.0.0/22**

| Purpose | Name | Subnet |
|------------|-------------------|---------------|
| Production | FF-PROD-SUBNET | 10.190.0.0/24 |
| Preprod | FF-PREPROD-SUBNET | 10.190.1.0/24 |
| Test/Dev | FF-TEST-SUBNET | 10.190.2.0/24 |

18.3 AWS Design

18.3.1 Network Load Balancer

18.3.1.1 Dublin Shared Services NLB

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|--------------------------------|-------------|---|----------|
| VF-IEDELIVERY-SS-DEV1-APIO-NLB | DEV1 | 198.19.220.128/26, 198.19.220.192/26 | |
| VF-IEDELIVERY-SS-SIT1-APIO-NLB | SIT1 | 198.19.220.128/26, 198.19.220.192/26 | |
| VF-IEDELIVERY-SS-SIT2-APIO-NLB | SIT2 | 198.19.220.128/26, 198.19.220.192/26 | |
| VF-IEDELIVERY-SS-SIT3-APIO-NLB | SIT3 | 198.19.220.128/26, 198.19.220.192/26 | |
| VF-IEDELIVERY-SS-SIT4-APIO-NLB | SIT4 | 198.19.220.128/26, 198.19.220.192/26 | |
| VF-IEDELIVERY-SS-PRD1-APIO-NLB | PRD1 | 198.19.220.32/27, 198.19.220.96/27 | |
| VF-IEDELIVERY-SS-PRD2-APIO-NLB | PRD2 | 198.19.220.32/27, 198.19.220.96/27 | |
| VF-IEDELIVERY-SS-PROD-APIO-NLB | PROD | 198.19.220.0/27, 198.19.220.64/27 | |

18.3.1.2 Frankfurt Shared Services NLB

The NLB will be created in the new **IE Frankfurt Transit VPC** for each VFIE environment.

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|----------------------------|-------------|--|----------|
| VF-IEDELIVERY-FKT-DEV1-NLB | DEV1 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | |
| VF-IEDELIVERY-FKT-SIT1-NLB | SIT1 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | |
| VF-IEDELIVERY-FKT-SIT2-NLB | SIT2 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | |
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | |

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|----------------------------|-------------|--|----------|
| VF-IEDELIVERY-FKT-SIT4-NLB | SIT4 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | |
| VF-IEDELIVERY-FKT-PRD1-NLB | PRD1 | 10.190.1.0/26, 10.190.1.64/26, 10.190.1.128/26 | |
| VF-IEDELIVERY-FKT-PRD2-NLB | PRD2 | 10.190.1.0/26, 10.190.1.64/26, 10.190.1.128/26 | |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | 10.190.0.0/26, 10.190.0.64/26, 10.190.0.128/26 | |

18.3.2 Target Groups

18.3.2.1 Dublin Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|------------------------------|-------|----------|-------------|---------------------------------|-----------|------------------------------------|
| SS-<ENV>-from-15101-to-30050 | 30050 | TCP | IP | VF-IEDELIVERY-SS-<ENV>-APIO-NLB | OUTBOUND | Route to Amdocs OSB (VCI Network) |
| SS-<ENV>-from-34801-to-8181 | 8181 | TCP | IP | VF-IEDELIVERY-SS-<ENV>-APIO-NLB | OUTBOUND | Route to HGW (VFIE Legacy Network) |

18.3.2.2 Frankfurt Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|--------------------------------------|-------|----------|-------------|-----------------------------|-----------|---|
| FF-TRANSIT-<ENV>-from-15101-to-15101 | 15101 | TCP | IP | VF-IEDELIVERY-FKT-<ENV>-NLB | OUTBOUND | Route to Shared Services NLB on port 15101 for Amdocs OSB |
| FF-TRANSIT-<ENV>-from-35002-to-443 | 443 | TCP | IP | VF-IEDELIVERY-FKT-<ENV>-NLB | OUTBOUND | Route to Workload account NLB for DPC API |
| FF-TRANSIT-<ENV>-from-34801-to-34801 | 34801 | TCP | IP | VF-IEDELIVERY-FKT-<ENV>-NLB | OUTBOUND | Route to Shared Services NLB on port 34801 for HGW |

18.3.3 Transit Gateway

18.3.4 Endpoints

18.4 Supporting Documentation

18.4.1 DXL / TAAS Roadmap

The current DXL integration will support the solution for the next 6-12 months. As part of the VF Group roadmap, all VF OpCo's are to move their Digital Microservices over to the new TAAS solution once it has been made available. VFIE are one of the first OpCo's to trial TAAS and should start this integration inside the next 3 months. TAAS is currently deployed in AWS Frankfurt, and will use the above Transit Gateway solution in parallel to DXL to integrate with VFIE Web Services.

As part of the TAAS Roadmap, the TAAS solution will be deployed to multiple AWS regions to simplify the integration for all VF OpCo's. Once this happens, the VFIE Microservices will be migrated from TAAS Frankfurt to TAAS Dublin, and the Transit Gateway solution should not be required unless a new service requires this connectivity in the interim.

18.4.2 TAAS Documentation

18.4.2.1 Overview

[01. Introduction to TaaS - Cloud Engineering - Vodafone Global Confluence](#)

18.4.2.2 Multi-region implementation

[CE-DP-003 - Edge Networking - Cloud Engineering - Vodafone Global Confluence](#)

18.4.3 CAAS NEL mServices

The NEL mServices are currently deployed in CAAS but will be migrated to TAAS once available. This service requires the ability to query the VFIE HGW to retrieve the IMSI associated with an API. To achieve this, they will use the VFIE Transit Gateway implemented between the Dublin and Frankfurt Shared Services accounts for Vodafone Ireland, and then use the IE_CSP_TRANSIT VRF to connect back into the VFIE Legacy network to the HGW exposed API.

18.501 - VFIE TAAS Integration

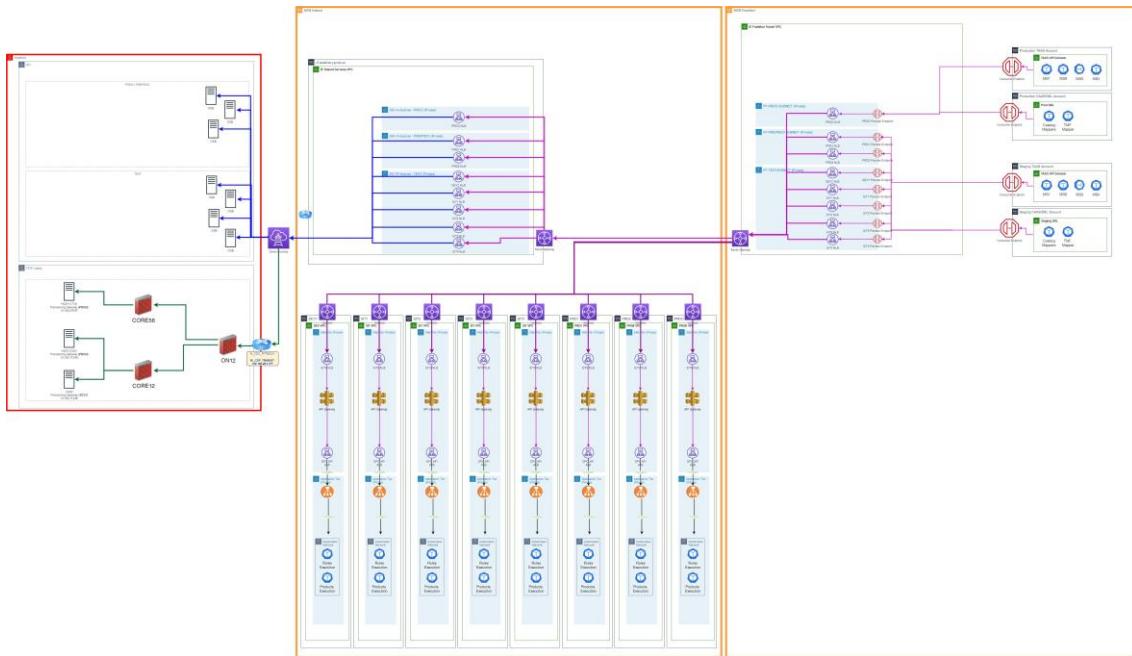
18.5.1 Overview

The VF Group managed TAAS solution integrates with the VFIE Shared Services Account to reach the VFIE on-prem exposed APIs. As the TAAS environment is based in AWS Frankfurt, it is necessary to route the traffic via the Frankfurt Transit Gateway already built in the VFIE Shared Services account.

18.5.2 Design

18.5.2.1 Network Design

The design for this connectivity was already taken into account when building the Frankfurt Transit Gateway solution.



18.5.2.2 AWS Design

18.5.2.2.1 TAAS Account Details

| Account Number | Names | Environment | OU Name | Purpose |
|----------------|-----------------------------|-------------|-------------------------|----------------------|
| 619368790323 | IE-Digital-Network-Non-Prod | Non-Prod | cep-ie-digital-non-prod | Ireland - On Premise |
| 868855661286 | IE-Digital-Network-Prod | Prod | cep-ie-digital-prod | Ireland - On Premise |

18.5.2.2.2 VFIE Shared Services NLB Endpoints (Frankfurt)

| NLB Name | Environment | Subnets (for the different AZs) | Port | Comments |
|----------------------------|-------------|--|-------|----------|
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | 15101 | |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | 10.190.0.0/26, 10.190.0.64/26, 10.190.0.128/26 | 15101 | |

18.5.3 Connectivity

18.5.4 Supporting Documentation

[11 VFIE - Frankfurt Transit Gateway Solution](#)

18.602 - VFIE mTAS - CaaS NEL Integration

- [Overview](#)
- [Design](#)
 - [Network Design](#)

- [AWS Design](#)
 - [CAAS Account Details](#)
 - [VFIE Shared Services NLB Endpoints \(Frankfurt\)](#)
- [Target Groups](#)
 - [Dublin Target Groups](#)
 - [Frankfurt Target Groups](#)
- [CAAS Provider Endpoints](#)
 - [SIT3](#)
 - [PROD](#)
- [Connectivity](#)
 - [Test](#)
 - [Production](#)
- [Supporting Documentation](#)

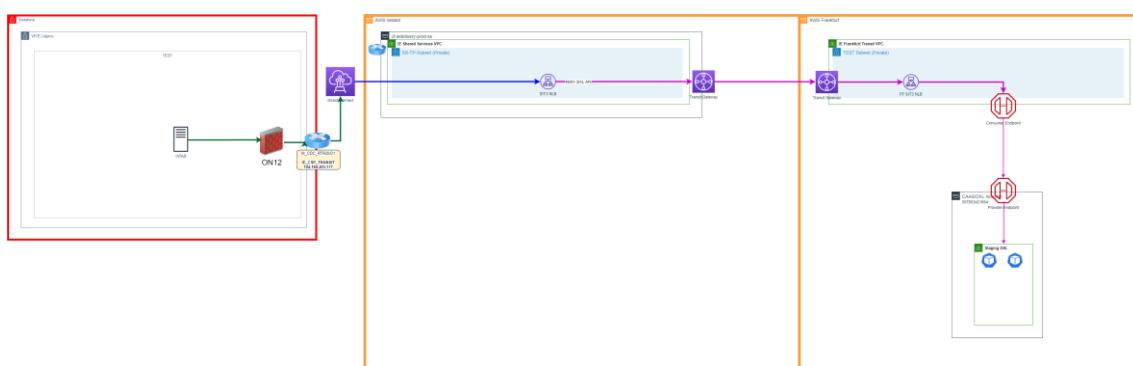
18.6.1 Overview

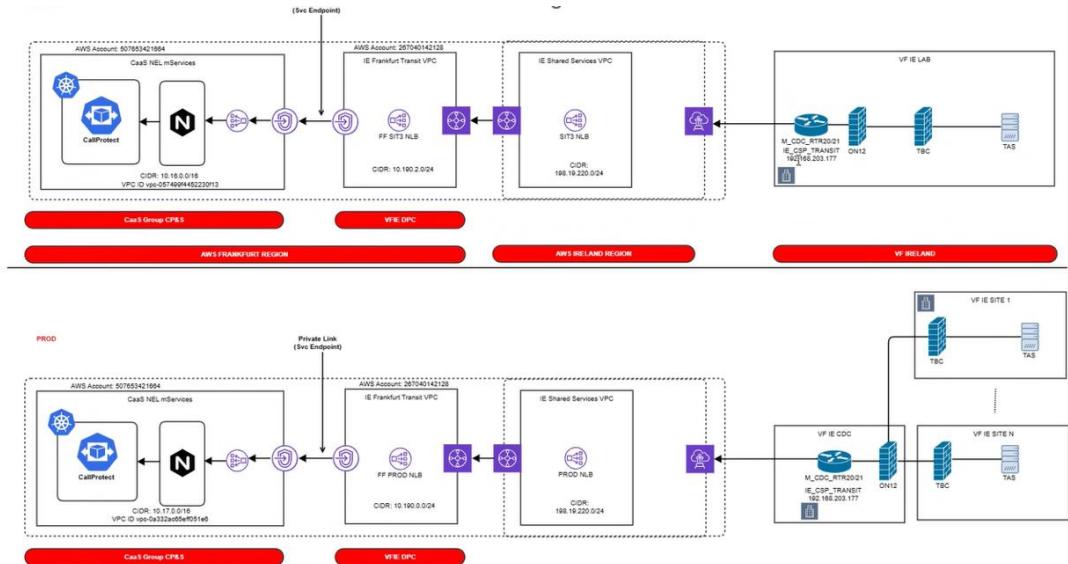
The VF-IE mTAS solution hosted in the Legacy VFIE Network needs to communicate with the VF Group Caas NEL (DXL) endpoints. As the CAAS environment is based in AWS Frankfurt, it is necessary to route the traffic via the Frankfurt Transit Gateway already built in the VFIE Shared Services account.

18.6.2 Design

18.6.2.1 Network Design

The connectivity will go via the IE_CSP_TRANSIT, over the Frankfurt Transit Gateway and on to the CAAS Endpoints.





18.6.2.2 AWS Design

18.6.2.2.1 CAAS Account Details

| Account Number | Names | Environment | OU Name | Purpose |
|----------------|--------------------|-------------|---------|---------|
| 507653421664 | CaaS NEL mServices | Non-Prod | | |
| 507653421664 | CaaS NEL mServices | Prod | | |

18.6.2.2.2 VFIE Shared Services NLB Endpoints (Frankfurt)

| NLB Name | Environment | Subnets (for the different AZs) | Port | Comments |
|----------------------------|-------------|--|-------|----------|
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | 49001 | |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | 10.190.0.0/26, 10.190.0.64/26, 10.190.0.128/26 | 49001 | |

18.6.2.3 Target Groups

18.6.2.3.1 Dublin Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|------------------------------|-------|----------|-------------|---------------------------------|-----------|------------------------|
| SS-<ENV>-from-49001-to-49001 | 49001 | TCP | IP | VF-IEDELIVERY-SS-<ENV>-APII-NLB | INBOUND | Route to Frankfurt NLB |

18.6.2.3.2 Frankfurt Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|------------------------------------|------|----------|-------------|-----------------------------|-----------|-------------------------|
| FF-TRANSIT-<ENV>-from-49001-to-443 | 443 | TCP | IP | VF-IEDELIVERY-FKT-<ENV>-NLB | INBOUND | Route to CaaS NEL (DXL) |

18.6.2.4 CAAS Provider Endpoints

18.6.2.4.1 SIT3

| | |
|---------------------------|--|
| Service Name | com.amazonaws.vpce.eu-central-1.vpce-svc-09d053703cdfae563 |
| Port | 443 |
| Allowed Principles | 267040142128 |

18.6.2.4.2 PROD

| | |
|---------------------------|--|
| Service Name | com.amazonaws.vpce.eu-central-1.vpce-svc-064fbba0a0f83a7e6 |
| Port | 443 |
| Allowed Principles | 267040142128 |

18.6.3 Connectivity

18.6.3.1 Test

| Firewall Rules | | | | | | | | | | | | |
|---------------------|-----------|---------------|-------------|---------------|-----------------|--------------------|------------|-------------|---------------|----------|------|------------|
| Rule Purpose | Source | | | Closest Fwall | Destination | | | | Closest Fwall | Service | | Install On |
| | Hostname | IP Address | Subnet Mask | | Firewall | Host Name | IP Address | Subnet Mask | | Protocol | Port | |
| mTAS to DXL S01-LAB | IE680vMTA | 10.151.1 52.6 | /32 | | AWS Test Subnet | 198.19.220.1 28/25 | /27 | ON12 | HTTP S | 490 01 | | |

18.6.3.2 Production

| Firewall Rules | | | | | | | | | | | | |
|-------------------------------|---------------|---------------|-------------|---------------|-------------------|----------------|------------|-------------|---------------|----------|------|------------|
| Rule Purpose | Source | | | Closest Fwall | Destination | | | | Closest Fwall | Service | | Install On |
| | Hostname | IP Address | Subnet Mask | | Firewall | Host Name | IP Address | Subnet Mask | | Protocol | Port | |
| VOLTE MTAS 680 Fixed MTAS 680 | IE680vMTA S01 | 10.151.16 6.5 | /32 | | AWS Prod Subnet 1 | 198.19.22 0.64 | /27 | ON12 | HTTP S | 490 01 | | |
| | IE680vMTA S02 | 10.151.16 6.6 | | | AWS Prod Subnet 2 | 198.19.22 0.0 | /27 | | | | | |

| | | | | | | | | | | | |
|----------------|---------------|--------------|-----|--|-------------------|----------------|---------|------|--------|--------|--|
| VOLTE MTAS 706 | IE706vMTA S01 | 10.151.47 .6 | /32 | | AWS Prod Subnet 1 | 198.19.22 0.64 | /27 /27 | ON12 | HTTP S | 490 01 | |
| Fixed MTAS 706 | IE706vMTA S02 | 10.151.47 .5 | | | AWS Prod Subnet 2 | 198.19.22 0.0 | | | | | |

18.6.4 Supporting Documentation

[11 VFIE - Frankfurt Transit Gateway Solution](#)

18.703 - VFIE OSB - CAAS / TAAS Integration

- [Overview](#)
- [Design](#)
 - [Network Design](#)
 - [AWS Design](#)
 - [CAAS Account Details](#)
 - [TAAS Account Details](#)
 - [VFIE Shared Services NLB Endpoints \(Frankfurt\)](#)
 - [Target Groups](#)
 - [Dublin Target Groups](#)
 - [Frankfurt Target Groups](#)
 - [VFIE Subnets](#)
- [TAAS Route 53](#)
 - [TEST](#)
 - [PREPROD](#)
 - [PROD](#)
- [Supporting Documentation](#)
- [PROJECT DELIVERY RACI/TASK LIST - Flow1](#)

18.7.1 Overview

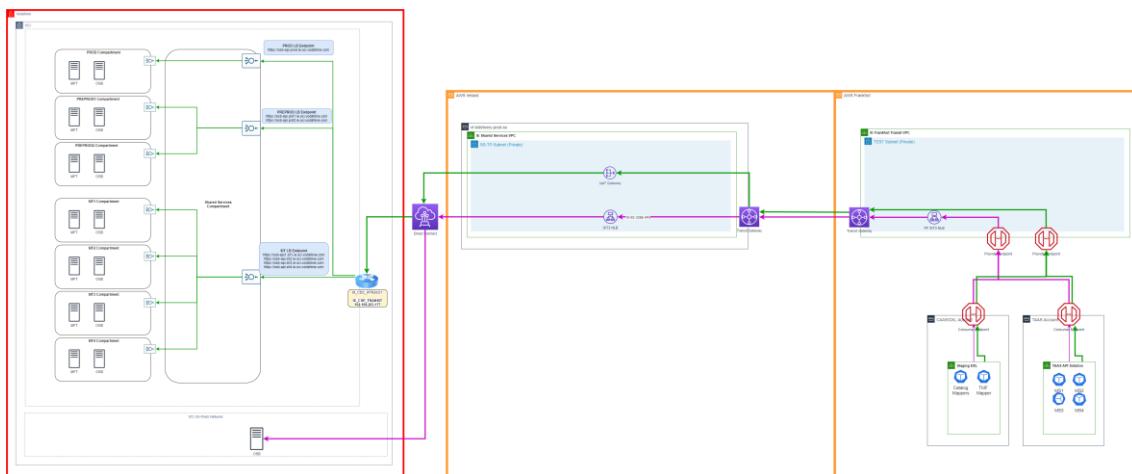
As part of the PMX programme, we're looking to optimise the network path from DXL/TAAS to the OSB via the AWS backbone instead of going via the internet as currently happens. Based on that, we're utilising the existing VFIE Frankfurt Transit Gateway solution to define the new design below, and implement a more robust solution.

18.7.2 Design

18.7.2.1 Network Design

The connectivity will go from the TAAS / CAAS endpoints, over the Frankfurt Transit Gateway, via the IE_CSP_TRANSIT, and on to either the DRCC OCI OSB or the VCI OSB. As DRCC

OCI OSB uses borrowed Oracle IP addresses, this needs to be routed via the Shared Services NAT Gateway as the OSB IP address cannot be placed behind an NLB.



18.7.2.2 AWS Design

18.7.2.2.1 CAAS Account Details

| Account Number | Names | Environment | OU Name | Purpose |
|----------------|--------------------|-------------|---------|---------|
| 507653421664 | CaaS NEL mServices | Non-Prod | | |
| 507653421664 | CaaS NEL mServices | Prod | | |

18.7.2.2.2 TAAS Account Details

| Account Number | Names | Environment | OU Name | Purpose |
|----------------|-----------------------------|-------------|---------|---------|
| 924279027542 | ie-digital-network-non-prod | Non-Prod | | |
| 118291558653 | ie-digital-app01-non-prod | Non-Prod | | |
| 606544231873 | ie-digital-app01-prod | Prod | | |
| 521557890114 | ie-digital-network-prod | Prod | | |

18.7.2.2.3 VFIE Shared Services NLB Endpoints (Frankfurt)

| NLB Name | Environment | Subnets (for the different AZs) | Port | Comments |
|----------------------------|-------------|--|-------|----------|
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | 15101 | |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | 10.190.0.0/26, 10.190.0.64/26, 10.190.0.128/26 | 15101 | |

18.7.2.3 Target Groups

18.7.2.3.1 Dublin Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|-----------------------------|-------|----------|-------------|--------------------------------|-----------|-----------------------------------|
| SS-PROD-from-15101-to-30050 | 30050 | TCP | IP | VF-IEDELIVERY-SS-PROD-APII-NLB | INBOUND | Route to Amdocs OSB (VCI Network) |

| | | | | | | |
|----------------------------|------|-----|----|--------------------------------|---------|-----------------------------------|
| SS-SIT3-from-15101-to-9007 | 9007 | TCP | IP | VF-IEDELIVERY-SS-PROD-APII-NLB | INBOUND | Route to Amdocs OSB (VCI Network) |
|----------------------------|------|-----|----|--------------------------------|---------|-----------------------------------|

18.7.2.3.2 Frankfurt Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|-------------------------------------|-------|----------|-------------|----------------------------|-----------|---|
| FF-TRANSIT-PROD-from-15101-to-15101 | 15101 | TCP | IP | VF-IEDELIVERY-FKT-PROD-NLB | INBOUND | Route to Shared Services NLB on port 15101 for Amdocs OSB |
| FF-TRANSIT-SIT3-from-15101-to-15101 | 15101 | TCP | IP | VF-IEDELIVERY-FKT-SIT3-NLB | INBOUND | Route to Shared Services NLB on port 15101 for Amdocs OSB |

18.7.2.3.3 VFIE Subnets

| Source | Env | Account No. | CIDR |
|----------|--------------------------|--------------|-------------------|
| Vodafone | Shared Services (Dublin) | 267040142128 | 198.19.220.0/24 |
| DRCC OCI | SIT | ON-PREM | |
| DRCC OCI | PROD | ON-PREM | 167.234.165.32/27 |

18.7.3 TAAS Route 53

18.7.3.1 TEST

| Record Name | CNAME /Consumer Endpoint/IP | Environment |
|----------------------------------|-----------------------------|-------------|
| ieesbbvr.dc-dublin.de | | On-prem |
| osb-api.sit1.ie.oci.vodafone.com | 104.242.241.213 | DRCC OCI |
| osb-api.sit2.ie.oci.vodafone.com | 104.242.241.213 | DRCC OCI |
| osb-api.sit3.ie.oci.vodafone.com | 104.242.241.213 | DRCC OCI |

18.7.3.2 PREPROD

| Record Name | CNAME /Consumer Endpoint/IP | Environment |
|----------------------------------|-----------------------------|-------------|
| osb-api.prd1.ie.oci.vodafone.com | | DRCC OCI |
| osb-api.prd2.ie.oci.vodafone.com | | DRCC OCI |

18.7.3.3 PROD

| Record Name | CNAME /Consumer Endpoint/IP | Environment |
|----------------------------------|-----------------------------|-------------|
| esb.prod.dox.equinox.vodafone.ie | | On-prem |
| osb-api.prod.ie.oci.vodafone.com | 167.234.165.112 | DRCC OCI |

18.7.4 Supporting Documentation

[11 VFIE - Frankfurt Transit Gateway Solution](#)

18.7.5 PROJECT DELIVERY RACI/TASK LIST - Flow1

| Components | Environments | Activity | Accountable | Responsible | Consulted | Approximate Start Date | Approximate Completion Date | Comments | Status |
|--------------------------------|--------------|---|-------------|------------------------|-----------------------------|------------------------|-----------------------------|---|-------------|
| Confluence Initial Review | All | VFIE Pre-requisite documentation to be provided for PCS TDF Internal Review | VFIE | VFIE, CaaS /TaaS , PCS | Security | 08 Sep 2025 | | | In Progress |
| Security Alignment | All | Security review | VFIE | VFIE | Security, CaaS /TaaS , PCS | | | | |
| Confluence Final Documentation | All | Document the set up and design as per each teams scope | VFIE | VFIE, CaaS /TaaS , PCS | Security | | Go Live | Ongoing until demand is handed over to PCMS . | In Progress |
| PCS Build | NON-PROD | Need to whitelist the TaaS accounts: 924279027542 and 118291558653 | PCS | PCS | CaaS /TaaS , Security, VFIE | 10 Sep 2025 | 11 Sep 2025 | | Open |
| TaaS / CaaS Build | NON-PROD | Create Consumer Endpoints and Route 53 entries: ieesbbvrdc-dublin.de port: 15101 Please use the below: Endpoint Services: com.amazonaws.vpce.eu-central-1.vpce-svc-0861763c45ff82100 | VFIE | TaaS / CaaS | | 11 Sep 2025 | 19 Sep 2025 | | Open |
| End to End Testing | NON-PROD | | VFIE | VFIE | PCS. TaaS | 22 Sep 2025 | 23 Sep 2025 | | Open |

| | | | Prod | | | | | | | |
|----------------------------------|------|---|-------|------------------------|----------------------------|-------------|-------------|--|--|------|
| PCS Build | Prod | Need to whitelist the Taas accounts:6 06544231873 and 521557890114 | PCS | PCS | Caas /Taas, Security, VFIE | 24 Sep 2025 | 25 Sep 2025 | | | Open |
| TaaS / CaaS Build | Prod | Create Consumer Endpoints and Route 53 entries: ieesbbvr.dc-dublin.de Port: 15101 Please use the below: Endpoint Services: com.amazonaws.vpce.eu-central-1.vpce-svc-080e2602b4bc0790a | VFIE | TaaS / CaaS | | 26 Sep 2025 | 30 Sep 2025 | | | Open |
| End to End Testing | Prod | | VFIE | VFIE | PCS, Taas | 01 Oct 2025 | 02 Oct 2025 | | | Open |
| Hand Over Relevant Support Teams | All | L1,L2,L3 | VFIE, | VFIE, CaaS / TaaS, PCS | Security | | | | | |
| | | | | | | | | | | |

18.804 - VFIE CIAM Webgate - CAAS Integration

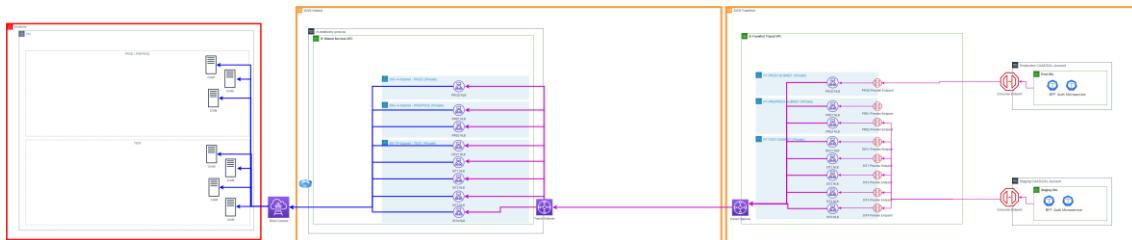
18.8.1 Overview

As part of the WEF programme, the WEF Auth Microsite needs to integrate with the CIAM Webgate endpoint via the WEF BFF for authentication purposes. In Production, it has the option to go via the internet to connect to n.vodafone.ie, but as the non-production environments are not exposed to the internet, we need to route these calls via the VFIE Frankfurt Transit Gateway solution. We need to implement this connectivity across all envs.

18.8.2 Design

18.8.2.1 Network Design

The connectivity will go from CAAS endpoints, over the Frankfurt Transit Gateway, via the IE_CSP_TRANSIT, and on to the VCI hosted CIAM Webgate.



18.8.2.1.1 Frankfurt VPC

The following is the Subnet Design for the new **IE Frankfurt Transit VPC 10.190.0.0/22**

| Purpose | Name | Subnet |
|------------|-------------------|---------------|
| Production | FF-PROD-SUBNET | 10.190.0.0/24 |
| Preprod | FF-PREPROD-SUBNET | 10.190.1.0/24 |
| Test/Dev | FF-TEST-SUBNET | 10.190.2.0/24 |

18.8.3 AWS Design

18.8.3.1 Network Load Balancer

18.8.3.1.1 Dublin Shared Services NLB

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|--------------------------------|-------------|---|-----------------------------|
| VF-IEDELIVERY-SS-DEV1-APIO-NLB | DEV1 | 198.19.220.128/26, 198.19.220.192/26 | Not Needed for this project |
| VF-IEDELIVERY-SS-SIT1-APIO-NLB | SIT1 | 198.19.220.128/26, 198.19.220.192/26 | Not Needed for this project |
| VF-IEDELIVERY-SS-SIT2-APIO-NLB | SIT2 | 198.19.220.128/26, 198.19.220.192/26 | REQUIRED |
| VF-IEDELIVERY-SS-SIT3-APIO-NLB | SIT3 | 198.19.220.128/26, 198.19.220.192/26 | REQUIRED |
| VF-IEDELIVERY-SS-SIT4-APIO-NLB | SIT4 | 198.19.220.128/26, 198.19.220.192/26 | REQUIRED |
| VF-IEDELIVERY-SS-PRD1-APIO-NLB | PRD1 | 198.19.220.32/27, 198.19.220.96/27 | Not Needed for this project |
| VF-IEDELIVERY-SS-PRD2-APIO-NLB | PRD2 | 198.19.220.32/27, 198.19.220.96/27 | REQUIRED |
| VF-IEDELIVERY-SS-PROD-APIO-NLB | PROD | 198.19.220.0/27, 198.19.220.64/27 | REQUIRED |

18.8.3.1.2 Frankfurt Shared Services NLB

The NLB will be created in the new **IE Frankfurt Transit VPC** for each VFIE environment.

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|----------------------------|-------------|--|-----------------------------|
| VF-IEDELIVERY-FKT-DEV1-NLB | DEV1 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | Not Needed for this project |

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|----------------------------|-------------|--|-----------------------------|
| VF-IEDELIVERY-FKT-SIT1-NLB | SIT1 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | Not Needed for this project |
| VF-IEDELIVERY-FKT-SIT2-NLB | SIT2 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | REQUIRED |
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | REQUIRED |
| VF-IEDELIVERY-FKT-SIT4-NLB | SIT4 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | REQUIRED |
| VF-IEDELIVERY-FKT-PRD1-NLB | PRD1 | 10.190.1.0/26, 10.190.1.64/26, 10.190.1.128/26 | Not Needed for this project |
| VF-IEDELIVERY-FKT-PRD2-NLB | PRD2 | 10.190.1.0/26, 10.190.1.64/26, 10.190.1.128/26 | REQUIRED |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | 10.190.0.0/26, 10.190.0.64/26, 10.190.0.128/26 | REQUIRED |

18.8.3.2 Target Groups

18.8.3.2.1 Dublin Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|----------------------------|------|----------|-------------|---------------------------------|-----------|-------------------------------------|
| SS-<ENV>-from-14001-to-443 | 443 | TCP | IP | VF-IEDELIVERY-SS-<ENV>-APIO-NLB | OUTBOUND | Route to CIAM Webgate (VCI Network) |

18.8.3.2.2 Destination Endpoints

| ENV | IP | PORT | FQDN | STATUS |
|------|--------------|------|--|----------|
| SIT1 | 37.25.163.15 | 443 | publish.sit1.portal.equinox.vodafone.ie | N/A |
| SIT2 | 37.25.163.18 | 443 | portal.publish.sit2.equinox.vf-ie.internal.vodafone.com | N/A |
| SIT3 | 37.25.163.19 | 443 | | COMPLETE |
| SIT4 | 37.25.163.17 | 443 | portal.publish.sit4.equinox.vf-ie.internal.vodafone.com | COMPLETE |
| PRD1 | 47.73.62.148 | 443 | | COMPLETE |
| PRD2 | 47.73.62.155 | 443 | | COMPLETE |
| PROD | 47.73.62.156 | 443 | | COMPLETE |

18.8.3.2.3 Frankfurt Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|--------------------------------------|-------|----------|-------------|-----------------------------|-----------|---|
| FF-TRANSIT-<ENV>-from-14001-to-14001 | 14001 | TCP | IP | VF-IEDELIVERY-FKT-<ENV>-NLB | OUTBOUND | Route to Shared Services NLB on port 14001 for CIAM Webgate |

18.8.3.2.4 CAAS Account Details

| Account Number | Names | Environment | Subnets | Purpose |
|----------------|--------------------|-------------|--|---------|
| 507653421664 | CaaS NEL mServices | Non-Prod | 10.1.78.0/24 10.25.0.0/16 10.1.84.0/24 | |
| 507653421664 | CaaS NEL mServices | Prod | 10.1.81.0/24 10.26.0.0/16 10.1.82.0/24 10.1.83.0/24 | |

18.8.3.2.5 Provider Endpoint

| NLB Name | Environment | Provider Endpoint for Frankfurt (by PCS) | New Provider Endpoints direct to Dublin (by PCS) |
|----------------------------|-------------|--|---|
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | com.amazonaws.vpce.eu-central-1.vpce-svc-0861763c45ff82100 | com.amazonaws.vpce.eu-west-1.vpce-svc-0a5267858cf4d81e8 |
| VF-IEDELIVERY-FKT-PRD2-NLB | PRD2 | com.amazonaws.vpce.eu-central-1.vpce-svc-07b8a20af7b145597 | com.amazonaws.vpce.eu-west-1.vpce-svc-04988aa29cf6d0fb7 |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | com.amazonaws.vpce.eu-central-1.vpce-svc-080e2602b4bc0790a | com.amazonaws.vpce.eu-west-1.vpce-svc-084880c4008c0722f |
| VF-IEDELIVERY-FKT-SIT4-NLB | SIT4 | | com.amazonaws.vpce.eu-west-1.vpce-svc-06fb9e11ddd71df90 |
| VF-IEDELIVERY-FKT-SIT2-NLB | SIT2 | com.amazonaws.vpce.eu-central-1.vpce-svc-06e3250521aa69033 | com.amazonaws.vpce.eu-west-1.vpce-svc-0895c2fb5af227ef4 |

18.8.4 Firewall Connectivity

[GIANT - WEF Authentication - AWS to the CIAM Webgate SIT4.xlsx](#) - done

[GIANT - ITR-8600 AWS to the CIAM Webgate SIT3, PRD2, Prod.xlsx](#) - done

[GIANT - WEF Authentication - AWS to the CIAM Webgate SIT2.xlsx](#) - CCP 244082, INC000097162749 for the firewall rules.

For expediting the firewall rules: Siddhesh Deoji, Vodafone <Siddhesh.Deoji@vodafone.com>; Prafulla Bachhav, Vodafone <prafulla.bachhav@vodafone.com>; DL-GDC-DCOPS-NTW-FW-Implementation <DL-GDC-DCOPS-NTW-FW-Implementation@vodafone.com>

18.8.5 Connectivity Set-up Example

SIT3:

728642754198 - CaaS account

com.amazonaws.vpce.eu-west-1.vpce-svc-0a5267858cf4d81e8 - servicename

vpce-0f29e328831c3a658-pij1giii.vpce-svc-0a5267858cf4d81e8.eu-west-1.vpce.amazonaws.com - dns for the endpoint

18.8.6 CaaS Ticket

SIT3: <https://cps.jira.agile.vodafone.com/browse/GAT-45473>

PRD2: <https://cps.jira.agile.vodafone.com/browse/GAT-45574>

Prod: <https://cps.jira.agile.vodafone.com/browse/GAT-45603>

SIT4: <https://cps.jira.agile.vodafone.com/browse/GAT-45874>

SIT2: <https://cps.jira.agile.vodafone.com/browse/GAT-45983>

18.8.7 Supporting Documentation

[11 VFIE - Frankfurt Transit Gateway Solution](#)

18.905 - VFIE WEF SSR - TaaS Integration

18.9.1 Overview

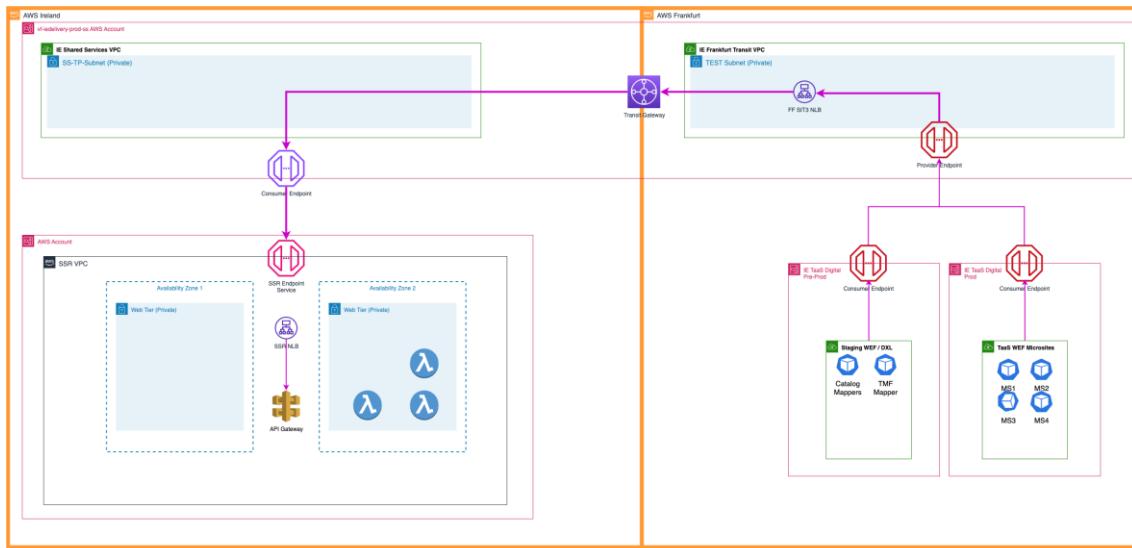
As part of the WEF migration to the TaaS platform, the WEF Microsites require connectivity to the Server Side Rendering Service (SSR) which is hosted in a VFIE managed AWS Account. The SSR service is built in the eu-west-1 (Ireland) AWS region. All traffic will be routed via the VFIE Frankfurt Transit Gateway solution. The solution implement will provide a route for traffic from ALL environments.

18.9.2 Design

18.9.2.1 Network Design

The connectivity is achieved via consumer endpoints in the DXL TaaS AWS accounts, over the Frankfurt Transit Gateway to a consumer endpoint configured in VF IE Shared Services AWS account in DUB which consumes from the SSR Endpoint Service in SSR AWS accounts (Test/dev, preprod & prod).

More detailed description of the traffic flow between IE TaaS WEF and SSR is available - [04. TaaS to Server-side Rendering Service Connectivity](#)



18.9.2.1.1 Frankfurt VPC

The following is the Subnet Design for the new **IE Frankfurt Transit VPC 10.190.0.0/22**

| Purpose | Name | Subnet |
|------------|-------------------|---------------|
| Production | FF-PROD-SUBNET | 10.190.0.0/24 |
| Preprod | FF-PREPROD-SUBNET | 10.190.1.0/24 |
| Test/Dev | FF-TEST-SUBNET | 10.190.2.0/24 |

18.9.3 AWS Design

18.9.3.1 Network Load Balancer

18.9.3.1.1 Dublin Shared Services NLB

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|---------------------------------|-------------|---|-----------------------------|
| VF-IEDELIVERY-SS-DEV1- APIO-NLB | DEV1 | 198.19.220.128/26, 198.19.220.192/26 | Not Needed for this project |
| VF-IEDELIVERY-SS-SIT1- APIO-NLB | SIT1 | 198.19.220.128/26, 198.19.220.192/26 | Not Needed for this project |
| VF-IEDELIVERY-SS-SIT2- APIO-NLB | SIT2 | 198.19.220.128/26, 198.19.220.192/26 | Not Needed for this project |
| VF-IEDELIVERY-SS-SIT3- APIO-NLB | SIT3 | 198.19.220.128/26, 198.19.220.192/26 | REQUIRED |
| VF-IEDELIVERY-SS-SIT4- APIO-NLB | SIT4 | 198.19.220.128/26, 198.19.220.192/26 | REQUIRED |

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|--------------------------------|-------------|---------------------------------------|-----------------------------|
| VF-IEDELIVERY-SS-PRD1-APIO-NLB | PRD1 | 198.19.220.32/27, 198.19.220.96/27 | Not Needed for this project |
| VF-IEDELIVERY-SS-PRD2-APIO-NLB | PRD2 | 198.19.220.32/27, 198.19.220.96/27 | Not Needed for this project |
| VF-IEDELIVERY-SS-PROD-APIO-NLB | PROD | 198.19.220.0/27, 198.19.220.64/27 | REQUIRED |

18.9.3.1.2 Frankfurt Shared Services NLB

The NLB will be created in the new **IE Frankfurt Transit VPC** for each VFIE environment.

| NLB Name | Environment | Subnets (for the different AZs) | Comments |
|----------------------------|-------------|--|-----------------------------|
| VF-IEDELIVERY-FKT-DEV1-NLB | DEV1 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | Not Needed for this project |
| VF-IEDELIVERY-FKT-SIT1-NLB | SIT1 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | Not Needed for this project |
| VF-IEDELIVERY-FKT-SIT2-NLB | SIT2 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | Not Needed for this project |
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | REQUIRED |
| VF-IEDELIVERY-FKT-SIT4-NLB | SIT4 | 10.190.2.0/26, 10.190.2.64/26, 10.190.2.128/26 | Not Needed for this project |
| VF-IEDELIVERY-FKT-PRD1-NLB | PRD1 | 10.190.1.0/26, 10.190.1.64/26, 10.190.1.128/26 | Not Needed for this project |
| VF-IEDELIVERY-FKT-PRD2-NLB | PRD2 | 10.190.1.0/26, 10.190.1.64/26, 10.190.1.128/26 | REQUIRED |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | 10.190.0.0/26, 10.190.0.64/26, 10.190.0.128/26 | REQUIRED |

18.9.3.2 Target Groups

18.9.3.2.1 Dublin Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|----------------------------|------|----------|-------------|----------------------------|-----------|---|
| SS-<ENV>-from-48002-to-443 | 443 | TCP | IP | VF-IEDELIVERY-SS-<ENV>-NLB | OUTBOUND | Route to Server Side Rendering (SSR) in AWS |

18.9.3.2.2 Destination Endpoints

| ENV | IP | PORT | FQDN | STATUS |
|------|----|------|------|--------|
| SIT1 | | | | |
| SIT2 | | | | |
| SIT3 | | | | |
| SIT4 | | | | |
| PRD1 | | | | |
| PRD2 | | | | |
| PROD | | | | |

18.9.3.2.3 Frankfurt Target Groups

| Name | Port | Protocol | Target type | Load balancer | Direction | Comment |
|-------------------------------------|-------|----------|-------------|-----------------------------|-----------|--|
| FF-TRANSIT-<ENV>-from-48200-to-xxxx | 14001 | TCP | IP | VF-IEDELIVERY-FKT-<ENV>-NLB | OUTBOUND | Route to Shared Services NLB on port 48002 for SSR |
| | | | | | | |
| | | | | | | |
| | | | | | | |

18.9.3.2.4 Provider Endpoint

| NLB Name | Environment | Provider Endpoint for Frankfurt (by PCS) | New Provider Endpoints direct to Dublin (by PCS) |
|----------------------------|-------------|--|--|
| VF-IEDELIVERY-FKT-SIT3-NLB | SIT3 | com.amazonaws.vpce.eu-central-1.vpce-svc-0861763c45ff82100 | |
| VF-IEDELIVERY-FKT-PRD2-NLB | PRD2 | com.amazonaws.vpce.eu-central-1.vpce-svc-07b8a20af7b145597 | |
| VF-IEDELIVERY-FKT-PROD-NLB | PROD | com.amazonaws.vpce.eu-central-1.vpce-svc-080e2602b4bc0790a | |

19 12 VFIE NAT Gateway

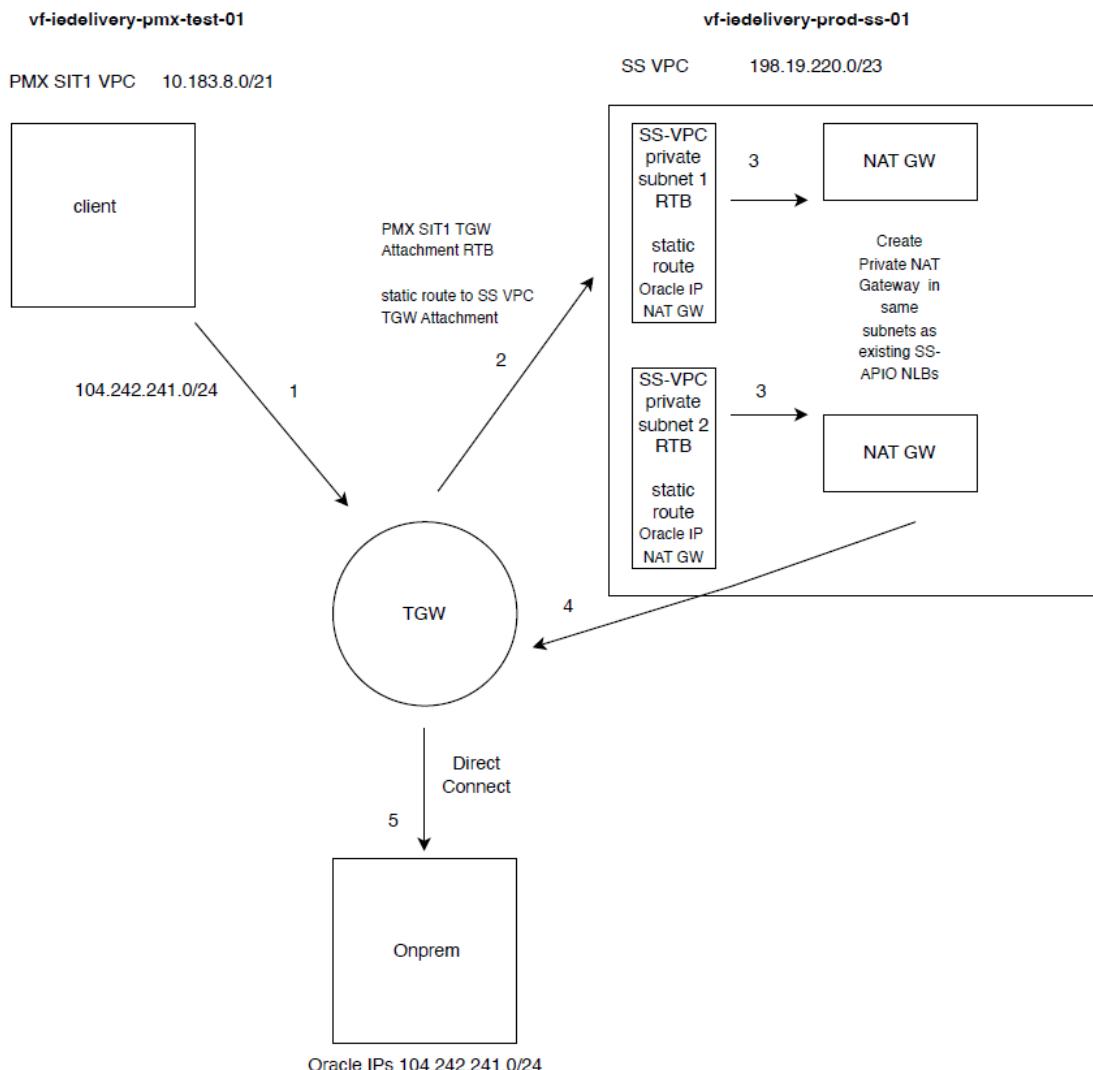
- [Overview](#)
- [High Level Design](#)
- [Subnets](#)
 - [AWS Shared Services Subnets](#)
 - [Destination Subnets](#)
- [NAT Gateway](#)
- [Routing Tables](#)
- [High-Level Implementation Steps](#)

19.1 Overview

Vodafone Ireland utilise an on-prem DRCC OCI implementation which uses IP Subnets provided by Oracle. As Oracle still own the associated subnets, it's not possible to add them as targets for an NLB as VF do not own them. To get around this issue, it has been agreed to use NAT Gateways in the Shared Services account to route the traffic for the OCI subnets to the VCI network.

Existing VPC's and Subnets will be used

19.2 High Level Design



Traffic Flow

Step 1: Client requests from PMX SIT1 VPC to on-premise Oracle IPs are routed to the TGW.
 Step 2: TGW routes traffic to SS VPC.

The PMX SIT1 TGW Attachment Route Table will have a static route to the SS VPC TGW Attachment ID.

Step 3: Traffic in the SS VPC subnets is routed to the NAT Gateways.

SS-VPC TGW private Subnet 1 and Subnet 2 Route Tables will have static routes for the destination

Oracle IP range and target as NAT Gateway.

Step 4: NAT Gateway routes traffic back to the TGW.

Step 5: TGW sends traffic through Direct Connect to on-premise Oracle IPs.

19.3 Subnets

19.3.1 AWS Shared Services Subnets

| ENV | Subnet | Security Zone | AZ | Existing |
|-----|--------|---------------|----|----------|
|-----|--------|---------------|----|----------|

| | | | | |
|---------|-------------------|-----|---|-----|
| TEST | 198.19.220.128/26 | T-P | 1 | Yes |
| | 198.19.220.192/26 | T-P | 2 | Yes |
| PROD | 198.19.220.0/27 | I-A | 1 | Yes |
| | 198.19.220.64/27 | I-A | 2 | Yes |
| PREPROD | 198.19.220.32/27 | I-A | 1 | Yes |
| | 198.19.220.96/27 | I-A | 2 | Yes |

19.3.2 Destination Subnets

| ENV | Subnet | Security Zone | DRCC OCI Region |
|------|--------------------|---------------|-----------------|
| TEST | 104.242.242.112/28 | T-A | Dublin 2 |
| | 104.242.241.208/28 | T-A | Dublin 2 |
| | 104.242.241.192/28 | T-P | Dublin 2 |
| | 104.242.241.108/28 | T-P | Dublin 2 |
| | 104.242.242.160/29 | T-P | Dublin 2 |
| | 104.242.242.176/28 | T-P | Dublin 2 |
| | 104.242.242.96/28 | T-A | Dublin 2 |
| | 104.242.242.168/29 | T-A | Dublin 2 |
| | 104.242.243.16/28 | T-A | Dublin 2 |
| | 104.242.243.80/28 | T-A | Dublin 2 |
| | 104.242.243.96/28 | T-A | Dublin 2 |
| | 104.242.242.144/28 | T-P | Dublin 2 |
| PROD | 104.242.241.224/28 | E1 (I) | Dublin 2 |
| | 104.242.241.240/28 | E2 (A) | Dublin 2 |
| | 167.234.175.224/28 | E1 (I) | Dublin 1 |
| | 104.242.242.128/28 | E1 (I) | Dublin 2 |
| | 167.234.167.96/28 | E1 (I) | Ratingen 2 |
| | 158.179.248.64/28 | E2 (A) | Ratingen 2 |
| | 167.234.167.112/30 | E2 (A) | Ratingen 2 |
| | 167.234.165.0/27 | E1 (I) | Dublin 1 |
| | 158.179.251.160/28 | E2 (A) | Dublin 1 |
| | 158.179.251.176/28 | E2 (A) | Dublin 1 |
| | 158.179.251.208/30 | E2 (A) | Dublin 1 |
| | 167.234.165.32/27 | E2 (A) | Dublin 1 |
| | 167.234.165.64/27 | E1 (I) | Dublin 2 |

| | | | |
|---------|--------------------|--------|------------|
| | 158.179.251.212/30 | E2 (A) | Dublin 2 |
| | 158.179.251.224/28 | E2 (A) | Dublin 2 |
| | 158.179.251.240/28 | E2 (A) | Dublin 2 |
| | 167.234.165.96/27 | E2 (A) | Dublin 2 |
| PREPROD | 167.234.167.192/28 | E1 (I) | Ratingen 2 |
| | 158.179.248.188/30 | E2 (A) | Ratingen 2 |
| | 167.234.167.208/28 | E2 (A) | Ratingen 2 |
| | 167.234.164.224/28 | E1 (I) | Dublin 1 |
| | 158.179.235.88/30 | E2 (A) | Dublin 1 |
| | 167.234.164.240/28 | E2 (A) | Dublin 1 |
| | 167.234.165.128/28 | E1 (I) | Dublin 1 |
| | 158.179.235.92/30 | E2 (A) | Dublin 1 |
| | 167.234.165.144/28 | E2 (A) | Dublin 1 |
| | 158.179.234.0/28 | E2 (A) | Dublin 2 |
| | 158.179.234.16/28 | E2 (A) | Dublin 2 |
| | 158.179.251.216/30 | E2 (A) | Dublin 2 |
| | 158.179.234.32/28 | E1 (I) | Dublin 2 |
| | 158.179.234.48/28 | E2 (A) | Dublin 2 |
| | 158.179.251.220/30 | E2 (A) | Dublin 2 |

19.4 NAT Gateway

| ENV | Subnet | Security Zone | AZ | NAT Gateway |
|---------|-------------------|---------------|----|-----------------|
| TEST | 198.19.220.128/26 | T-P | 1 | NAT-GW-TEST-TP1 |
| | 198.19.220.192/26 | T-P | 2 | NAT-GW-TEST-TP2 |
| PROD | 198.19.220.0/27 | I-A | 1 | NAT-GW-PROD-IA1 |
| | 198.19.220.64/27 | I-A | 2 | NAT-GW-PROD-IA2 |
| PREPROD | 198.19.220.32/27 | I-A | 1 | NAT-GW-PRE-IA1 |
| | 198.19.220.96/27 | I-A | 2 | NAT-GW-PRE-IA2 |

19.5 Routing Tables

19.6 High-Level Implementation Steps

Create Private NAT Gateways: Ensure NAT Gateways are created in Test, Pre-Prod and Production subnets as per **Section 4. NAT Gateway**

Update Route Tables:

- PMX <ENV> TGW Attachment Route Table: Add a route to the SS VPC TGW Attachment.
- SS-VPC TGW Subnet Route Table: Add a route for the Oracle IP range destination with the target set to the NAT Gateway.