# Technical Overview

## Description of Effort

The architecture presented attempts to align with a variety sources, mentioned below. A Kuali Coeus Application is deployed in a standard three (3) tier pattern: Web Proxy/Access Layer, Application Layer, and Database Layer. The architecture presented here aligns with best practice recommendations put forth by the industry to deploy such an architecture on the Amazon Web Services Platform.

A “legacy” (Version 5.2.1) deployment of this application is on-line and in production on premise at Boston University currently, however it is deployed at a smaller scale & with different system options than the planned deployment described in the Business Requirements Document (BRD). Notably, the new architecture is to support the following features/abilities that different from the current deployment:

* Higher uptime expectations
* Tighter RPO/RTO expectations
* Support for a larger user community
* Database Replication to support
* Infrastructure & Security as Code / Cloud Hosting
* Encryption in transit and at rest

The technical architecture described here attempts to incorporate these new features into the design while complying with the requirements within the BRD, as well as the operational concerns of the Boston University IT Department.

As such, presented here is a primary architecture, followed by design modifications that allow for flexibility in supporting application design issues, operational concerns, and possible testing outcomes.

### Internal Referenced Standards / Submittals

* Kauli Coeus Solution Architecture - DRAFT 1.7 – 29022016
* KCRM Proposal Development and Conflict of Interest – Version 1.0 – 06032016
* Security – **1.2.E Minimum Security Standards**
  + <http://www.bu.edu/policies/information-security-home/minimum-security-standards/>
* Security - **1.2.B Data Management Guide**
  + <http://www.bu.edu/policies/information-security-home/data-management-guide/>
* Security - **1.2.A Data Classification Guide**
  + <http://www.bu.edu/policies/information-security-home/data-classification-guide/>
* Various Kuali Foundation Documentation
  + <https://wiki.kuali.org/display/KRADOC/Home>

## Primary Architecture Plan



The application will be deployed in a 3-teir pattern, utilizing load balancers between tiers to decouple the application and infrastructure work loads.

Unlike the currently deployed legacy system, the AWS based workload will use an Application/Tomcat server that runs a merged copy of the “KC” and “KR” code base. This will result in an architecture that has six (6) subnets, fronted by a load balancer or an RDS service endpoint – effectively providing three tiers of logical architecture.

Four (4) system environments are needed:

* Production
* *Staging*
* *QA*
* *Development*

The Production environment will be deployed into it’s own Virtual Public Cloud (VPC) while the Staging, QA, and Development environments will be co-located into their own shared VPC.

A VPC is a logical container defined by IP space, or put into other terms, in it’s own network. VPC’s are separated from each other by not only the network segmentation, but by permission sets within the AWS Platform. This configuration will allow Production to be isolated from other

the non-production systems – thus isolating the “blast radius” of an issue in the non mission critical environments.



Within the Secondary VPC, Development, Staging, and Development will be deployed to utilize individual Availability Zones. (AZ’s) AZ’s represent physically separate datacenters, but present themselves to the architecture/network as difference subnets providing typically under 1ms of latency between AZ’s.

Development and QA will utilize Load Balancers to de-couple the tiers & to allow for a close alignment to Staging and Production.

Staging will be mimic the intended network architecture for Production, utilizing pairs of subnets at each tier separated by a load balancer as well as an RDS Endpoint that provides Oracle 12c services to the application.

As stated before, the Production environment will be deployed with in it’s own VPC, that will host only the Production Application and the required subnets to support it.

### Alignment with AWS Best Practices

AWS Architectures, when deployed against current best standard, align with the four (4) “Pillars of Design.”[[1]](#footnote-1) The proposed architecture and theses design standards align in this architecture in the following manner:

|  |  |  |
| --- | --- | --- |
| Pillar Name | Description | Technical Alignment |
| Security | The ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies. | Security Groups, NACL optional, Encryption at rest/transit, Logical Routing Segmentation, Ingress/Egress Logging (Flow Logs), AWS Platform Logging (CloudTrails/CloudWatch), Platform AuthN/AuthZ (IAM), VPC Deployment, Oracle SE ONN Optional |
| Reliability | The ability of a system to recover from infrastructure or service failures, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues. | Load Balancing, Geographic Redundancy (Multi-AZ Design) Automated Data Copy (Snapshots/AMIs), multi-path connectivity (Dual Connection VPN), RDS |
| Performance Efficiency | The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve. | Starting Resource Profiling, A/B Side Deployment, Provisioned IOPS, Object Storage utilization (S3), Infrastructure as Code (CloudFormation Templating) |
| Cost Optimization | The ability to avoid or eliminate unneeded cost or suboptimal resources. | RI Consideration, Platform Monitoring (CloudWatch), multiple instance types, AWS Linux deployment, RDS |

### System Components[[2]](#footnote-2)

**Web Browser**

Kuali Coeus users are required to use a major Internet browser such as Internet Explorer, Firefox, Safari or Chrome.

KCRM users access <http://kuali.bu.edu/kc>

Rice admins access <http://kuali.bu.edu/kr>

**Servers**

AWS resources are hosted in multiple locations world-wide. These locations are composed of regions and Availability Zones. Each region is a separate geographic area. Each region has multiple, isolated locations known as Availability Zones. Amazon EC2 provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across regions unless done specifically.[[3]](#footnote-3)

The system is planned to be deployed in the US-EAST-1 Region, and utilize 2 Availability Zones.

The servers considered for production deployment are:

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Proxy Server | KC App | Oracle Server |
| vCPU | 1 | 2 | 4 |
| RAM | 4GB | 6.5GB | 15GB |
| Instance Type | m3.medium | M4.large | Db.m3.large |
| 2nd Option | m4.large | C4.xlarge | Db.m4.2xlarge |

If the primary instance type does not meet the performance standards desired, the 2nd Option can be tested and evaluated.

**Server Operating Systems/Versions**

* Amazon Linux – 2015.09[[4]](#footnote-4)

**Application Software/Versions**

* Kuali Coeus – V5.2.1
* Kuali Rice – V3.2.6

**Supporting Software/Versions**

* Apache – v2.2
* Tomcat – v6.0
* Informatica – v9.0.1
* SAP
* SAP DW
* KC/SAP web service – Apache CXF
* Oracle 12c (12.1.0.2) Standard Edition (SE) – contains the KC and KR schemas as well as all attachments

**Environments – (Suggest multi VPC Deployment)**

* SB2 – Kuali major release
* DEV – development environment (distributed)
* SB – Kuali release (same as Prod with no BU customization)
* TEST (QA)
* Staging (UAT)
* Prod

### AWS Relational Database System (RDS) Deployment Description

#### Summary

The current plan of record is to utilize the AWS RDS system to support the Oracle database need to support the Kuali application.

Amazon Relational Database Service (Amazon RDS) is a managed service that provides cost-efficient and resizable capacity, while managing time-consuming database administration tasks.

Amazon RDS gives you access to the capabilities of a familiar RDBMS, Oracle 12c in this case. Code, applications, and tools already in use today with existing databases should work seamlessly with Amazon RDS.

Amazon RDS also automatically patches the database software and backs up your database, storing the backups for a user-defined retention period.

It should be noted that this choice is driven by a number of business requirements, as well as the possibility of changes that may take place within the Kuali development teams support of the Oracle RDBMS.

#### Design Concerns

RDS will provide the Oracle 12c Instances that the platform requires to function. These instances will be accessed via an RDS endpoint.

Oracle Instances in the Production & Staging will be deployed as “Multi-AZ”.

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for Database (DB) Instances. When a RDS DB is provisioned as a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable. In case of an infrastructure failure (for example, instance hardware failure, storage failure, or network disruption), Amazon RDS performs an automatic failover to the standby, so that the application can resume database operations as soon as the failover is complete. Since the endpoint for the DB Instance remains the same after a failover, applications can resume database operation without the need for manual administrative intervention.[[5]](#footnote-5)

RDS Database instances will be deployed into two (2) subnets. These subnets make up the DB Tier of the architecture, and each subnet is deployed to a separate availability zone. There is no direct access to these instances allowed by the system – they can only be accessed programmatically from the provided RDS Endpoint for the running instance.

#### Operational Concerns

Typically, any utility that can interact with Oracle using the standard FQDN and Instance port can be used with an Oracle instance hosted by RDS.

However, actions that require console / OS access are not supported. Also, the following features are not supported for Oracle 12c on Amazon RDS:

* Real Application Clusters (RAC)
* Data Guard / Active Data Guard
* Cloud Control (called Oracle Enterprise Manager Grid Control in previous Oracle versions)
* Automated Storage Management
* Database Vault
* Java Support
* Locator
* Oracle Label Security
* Spatial

In order to assist an Oracle DBA in understanding how to maintain and operate an Oracle instance within in the RDS system, AWS publishes a guide[[6]](#footnote-6) that explains how to handle the following tasks, using RDS tools and utilities:

• **System**Enabling and disabling Restricted Session  
Flushing the Shared Pool  
Flushing the Buffer Cache  
Disconnecting a Session (for version 11.2.0.3.v1 and later)   
Killing a Session   
Renaming the Global Name (for version 11.2.0.3.v1 and later)   
Granting Privileges to Non-Master Users   
Modifying DBMS\_SCHEDULER Jobs

• **Logs**Switching Online Log files  
Adding, Dropping and Resizing Online Redo Logs  
Setting Force Logging (for version 11.2.0.3.v1 and later)  
Retaining Archived Redo Logs (for version 11.2.0.2.v7 and later)   
Setting Supplemental Logging (for version 11.2.0.3.v1 and later)

• **Databases**Creating and Resizing Tablespaces and Data Files  
Setting Default Tablespace  
Setting Default Temporary Tablespace  
Checkpointing the Database  
Setting Distributed Recovery (for version 11.2.0.3.v1 and later)  
Granting SELECT or EXECUTE privileges to SYS Objects (for version 11.2.0.3.v1 and later)  
Setting the Database Time Zone  
Working with Automatic Workload Repository (AWR)  
Adjusting Database Links for Use with DB Instances in a VPC

The RDS instances will be configured to utilize service level encryption, based on the AWS Account keys. Data that is encrypted at rest includes the underlying storage for a DB instance, its automated backups, Read Replicas, and snapshots.

#### DB Data Recovery

RDS will be configured to hold 35 days of daily backups and transaction files for automated backup in production. Sizing for this will need to be established during pre-deployment testing.

Other environments (Dev, Test, etc) will be configured to hold 14 days of backups and transaction logs.

To accommodate manual backups, and long term point in time recovery images, a SnapShot can be created of the database. Amazon RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases.

**Note:** Creating this DB snapshot on a Single-AZ DB instance results in a brief I/O suspension that typically lasting no more than a few minutes. *Multi-AZ DB instances are not affected by this I/O suspension since the backup is taken on the standby.* This will allow for aggressive snapshotting of the DB in order to support the required RPO objectives.

Snapshots can be manually copied as well, using the rds copy-db-snaphot CLI tool.

#### RDS Alternative - EC2 Based Oracle Enterprise Edition Deployment Description

If it is determined that using RDS to support the system is no finically or architecturally preferable, the following information has been assembled to support the use of Oracle deployed on a set of EC2 Instances.

##### Summary



A pair of EC2 Instances will be deployed to support Oracle 12c. These systems will be configured to utilize the DataGuard feature of the Oracle DBMS – with one server running as a “Primary” while the other runs as a Secondary. The Application Servers will connect to the servers utilizing a JDBC connector configured in such a way as to be compatible with the DataGuard process.

In essence, if a failure on the Primary DB is detected, the JBDC Driver will switch DB targets from the Primary to Secondary DB Server Instance.

For added protection, the Redo Logs from the Primary and Secondary Oracle systems will be shipped to a NFS server located in a third availability zone to facilitate both a continuous incremental backup, as well as the ability to roll forward from a snapshot should the condition arise that supports such a recovery need.

This would also modify, slightly, the configuration of the non-production VPC and supporting environments.



The Oracle systems would be configured according to the Boston University IT Standard, with the following disk allocation



##### Design Concerns

The Oracle systems would be managed in their entirety by the Boston University DB IT Support team. It would require that they be fully integrated with the current Grid system to manage archive/backup operations, as well as monitoring and operational instrumentation.

All updates, patching and upgrades would also be the responsibility of the on premise team.

### Web/Proxy Tier Design Considerations

In order to balance the business requirement to allow access to the Kuali platform from the open internet, and the security requirements found in the Boston University Minimum Security Requirements, the FQDN for the application will need to be hosted at the edge of the Boston University network, and then routed to the AWS VPC if the Web/Proxy components are hosted in the AWS cloud.

If the routing of this ingress layer is problematic, then the Web/Proxy layer will need to be hosted in the on premise network, and then linked back to the AWS VPC. Both scenarios are explored below.

#### AWS Based Web/Proxy Layer

A DNS Alias for the load balancer fronting the Web/Proxy Layer will need to be associated with an IP hosted and controlled by the local on premise network. The alias will point to a FQDN, not an IP. Even though the load balancer address represents an internal non-routable IP, AWS advertises the DNS entry.



Once the request is routed to the Load Balancer in the AWS VPC, the user is connected to the application.

It also might be possible to avoid using the on premise network for pass through to the VPC for user connections by utilizing a **Cisco Cloud Services Router (CSR) 1000V virtual appliance.** The device would provide the same level of functionality / oversight that is found with the on-site equipment in the same logging format. NOTE: The exact feature set / requirement are not enumerated in the provided documentation from Security. This information would be required to be enumerated in order to properly evaluate the value of this solution.

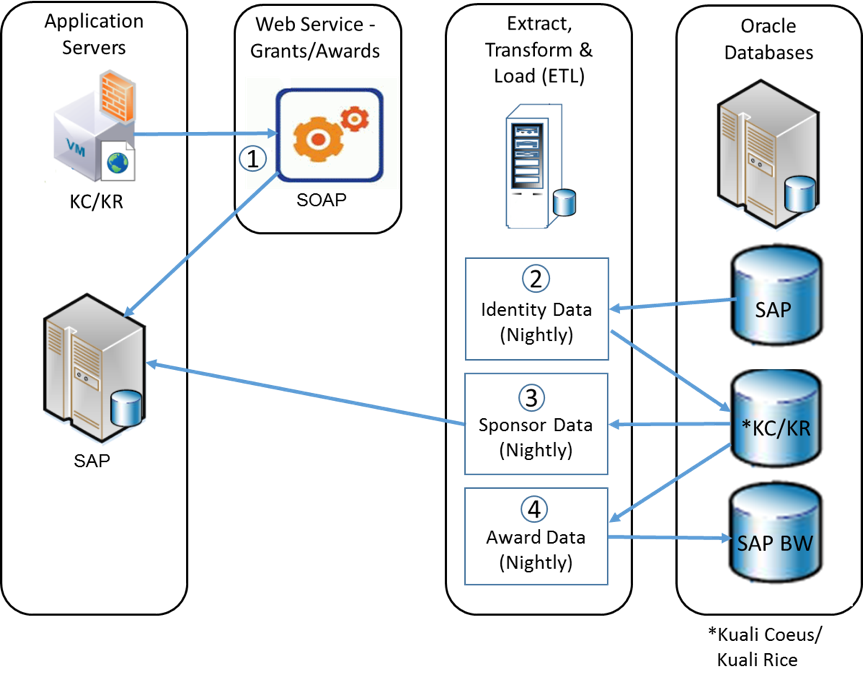
#### On-premise Web/Proxy Layer

If using an alias that resolves to a FQDN as opposed to an IP is problematic, or passing the traffic from the edge on premise network to the VPC is not supported by policies not yet discovered, the web layer could be moved to the on premise network, and then all traffic to the Tomcat/Application Server layer could be sent over the VPN to the application infrastructure.

One of the larger drawback to this pattern, is that the local infrastructure would need to provide load balancing for the Web Proxy Layer, or the layer would need to consist of a single server. Also, there would be no geographic dispersal of the Web ingress layer.

### Data integration Layer / Processes

As described in pages 11-12 of the Kuali Coeus Solution Architecture DRAFT, the application system requires access to both CAS, and Infomatica / SAP data transfers.



Functionally, these processes would not change. The VPC environment would need to route to the current on premise service interfaces in order to gain access to services, and data transfers. This will be provided for / secured through the use of routing tables, security groups, and the VPN connection between the Production VPC and the on premise environment.

### Security & Network Design

***NOTE:*** *This section describes a single environment. The Boston University Networking team will need to provide the IP address spacing for the VPC’s that host the various environments. In this example, we will assume that the production VPC environment has been provided the CIDR block 10.255.0.0/24, and the internal network is running off of the fictional address of 1.2.3.4/24*

### Network Structure & Routing



Application traffic will be routed from the Boston University edge network, to the AWS VPC over a VPN Tunnel.

Once the traffic has ingressed into the VPC, the Apache/Proxy Web Tier will receive the requests via an AWS load balancer, (Kauli-Proxylb-Production). Application traffic from the Web servers bound for the Application/Tomcat Layer will be fronted by another AWS load balancer, (Kauli-KRLB-Production). Traffic from the Tomcat systems to the DB will head to the RDS End point. Each teir will be separated in this manner, only allowing application traffic to pass from tier to tier via a load balancer.

Load balancers are configured to accept incoming traffic by specifying one or more listeners. A listener is a process that checks for connection requests. It is configured with a protocol and port number for connections from clients to the load balancer and a protocol and port number for connections from the load balancer to the instances.[[7]](#footnote-7)

(Backups or system configuration data that needs to be sent the the S3 Object Store will be routed to a S3 End point associated with the VPC when it is deployed.)

System administration traffic will be allowed to take a straight route between the Boston University Network, to the individual servers. (Access will be controlled by subnet host / IP Host range filtering)



### Security Groups

Security Groups in AWS are a melding of the concepts of Port Firewalls, and VLANs. Security Groups contain Instance (Virtual Server) network interfaces and define the allowed inbound and outbound traffic from a source by TCP/UDP/IMCP port. Security groups can define a source as an IP Domain, a specific resource ID, or other Security Groups. This concept of membership allows VLAN like logical grouping of resources.

Below is a high level visual of the Security Group hierarchy of this system architecture from the service and tier perspective.



Another two (2) overlapping Security Groups will be assigned to the DB hosts allowing access to port 1541 for SQL Operations and RDBMS functions. All other systems will be included in the other group, allowing SSH access.

### Load Balancers

Before a client sends a request to your load balancer, it resolves the load balancer's domain name using a Domain Name System (DNS) server. The DNS entry is controlled by Amazon, because your instances are in the amazonaws.com domain. The Amazon DNS servers return one or more IP addresses to the client. These are the IP addresses of the load balancer nodes for your load balancer. As traffic to your application changes over time, Elastic Load Balancing scales your load balancer and updates the DNS entry.

The client uses DNS round robin to determine which IP address to use to send the request to the load balancer. The load balancer node that receives the request uses a routing algorithm to select a healthy instance. It uses the round robin routing algorithm for TCP listeners, and the least outstanding requests routing algorithm (favors the instances with the fewest outstanding requests) for HTTP and HTTPS listeners.

Cross-zone load balancing will be utilized with the deployed load balancers. The cross-zone load balancing setting also determines how the load balancer selects an instance. If cross-zone load balancing is disabled, the load balancer node selects the instance from the same Availability Zone that it is in. If cross-zone load balancing is enabled, the load balancer node selects the instance regardless of Availability Zone. The load balancer node routes the client request to the selected instance.

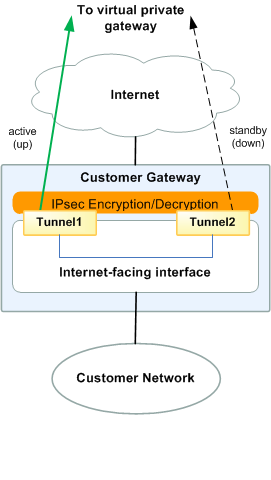
The Web Proxy Load Balancer will require SSL Server Certificates in order to provide Application communication encryption between the Client and the Application systems. An X.509 certificate (SSL server certificate) will be obtained by Boston University IT and deployed on the Web Proxy Load Balancer.[[8]](#footnote-8)

Back-End Server Authentication for SSL is supported, but operationally, Load Balancer hosting of the SSL Cert is recommended.[[9]](#footnote-9)

Proxy Protocol, as well as support for Sticky Sessions will also be leveraged.

LOAD BALANCER CONFIGS HERE

### VPN Connectivity



A site-to-site VPN provides connectivity between the Boston University Network and AWS VPCs in the US-EAST-1 Region. VPNs are not only region specific; they are also VPC specific in a given region.

AWS VPN connections are configured to be redundant by default on the AWS side. Each VPN connection provides a pair of end points for establishing a VPN tunnel. However, this will only provide fault tolerant connectivity to AWS if there is more than 1 router within the Boston University data center. At this time, there is only one Cisco ASA device, however a redundant ASR implementation is on the books for deployment soon.

For more information on specifically setting up a VPN connection utilizing Cisco Customer Gateway equipment, please see “**Amazon Virtual Private Cloud Network Administrator Guide.”[[10]](#footnote-10)**

**Direct Connect**

At some point in the future, if bandwidth usage, and/or latency issues to the AWS cloud arise, a Direct Connect cross connect from Boston University’s Internet2 backbone to AWS should be investigated.

AWS Direct Connect is a service that will allow Boston University to establish a dedicated network connection between the on premise network and one of the AWS Direct Connect locations. Using industry standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. This allows the same connection to be used to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within an [Amazon Virtual Private Cloud (VPC)](http://aws.amazon.com/vpc/) using private IP space, while maintaining network separation between the public and private environments.[[11]](#footnote-11)

Data that would have previously been transported over multi-tenant communication paths via VPN can now be delivered through a private network. Using AWS Direct Connect enables connection between AWS and the Boston University Enterprise network. In many circumstances, private network connections can reduce costs, increase bandwidth, reduce complexity and provide a more consistent network experience than Internet-based connections.[[12]](#footnote-12)

Consistent network experience, discounted data transfer rates out of AWS, and the technical advantage of leveraging/utilizing VLAN Tagging to VPCs could make shifting to Direct Connect connectivity beneficial to Boston University at some future state.

### Logging

**CloudTrail** is an AWS service and is AWS Region specific, and should be enabled from day one in all regions. With AWS CloudTrail, Boston University can get a history of AWS API calls and related events related to the account used to support this application. This includes calls made by using the AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services.

Boston University can identify which users and accounts called AWS for services that are supported by CloudTrail, the source IP address the calls were made from, and when the calls occurred.

In the simplest case, CloudTrail writes information to an Amazon S3 bucket that belongs to the AWS account where CloudTrail is turned on. These logs will be picked up by the University’s central logging service, Archsight.

**VPC Flow Logs** is a feature that enables you to capture information about the IP traffic going to and from network interfaces into the application VPCs. Flow log data is stored using Amazon CloudWatch Logs. After a flow log is create, it can be viewed and retrieved via the AWS CloudWatch monitoring console.

Flow logs can with a number of tasks; for example, to troubleshoot why specific traffic is not reaching an instance, which in turn can help diagnose overly restrictive security group rules.

Flow logs (and other CloudWatch logs) can be exported to an S3 bucket using the AWS Lambda service, so that they can be picked up by the University’s Arcsight system.

# Appendix XXXX – Subnet Descriptions

{

"Subnets": [

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-TomcatB-Private",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.64/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1b",

"SubnetId": "subnet-8a27b0fc",

"AvailableIpAddressCount": 11

},

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-ManagementB-Private",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.112/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1b",

"SubnetId": "subnet-a726b1d1",

"AvailableIpAddressCount": 11

},

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-ManagementA-Private",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.96/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1a",

"SubnetId": "subnet-e65ffbcc",

"AvailableIpAddressCount": 11

},

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-ProxyA-Public",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.0/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1a",

"SubnetId": "subnet-235efa09",

"AvailableIpAddressCount": 11

},

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-TomcatA-Private",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.32/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1a",

"SubnetId": "subnet-5d5efa77",

"AvailableIpAddressCount": 11

},

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-ProxyB-Public",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.16/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1b",

"SubnetId": "subnet-6127b017",

"AvailableIpAddressCount": 10

},

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-DatabaseB-Private",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.80/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1b",

"SubnetId": "subnet-f926b18f",

"AvailableIpAddressCount": 11

},

{

"VpcId": "vpc-5f27083b",

"Tags": [

{

"Value": "Kuali-DataBaseA-Private",

"Key": "Name"

}

],

"CidrBlock": "10.255.0.48/28",

"MapPublicIpOnLaunch": false,

"DefaultForAz": false,

"State": "available",

"AvailabilityZone": "us-east-1a",

"SubnetId": "subnet-445ffb6e",

"AvailableIpAddressCount": 11

}

]

}

# Appendix XXXX – Security Group Detail

**KualxProxyLB-Production**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Protocol | Port Range | Source | Destination |
| Inbound | | | | |
| All Traffic | TCP | ALL | 64.138.129.194/32 | -- |
| All Traffic | TCP | ALL | 68.188.198.130/32 |  |
| All Traffic | TCP | ALL | sg-e556bf80 | -- |
| Outbound | | | | |
| All Traffic | ALL | ALL | -- | Sg-e556bf80 |
| All Traffic | ALL | ALL | -- | 0.0.0.0/0 |

**KualxKRLB-Production**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Protocol | Port Range | Source | Destination |
| Inbound | | | | |
| All Traffic | TCP | ALL | 64.138.129.194/32 | -- |
| All Traffic | TCP | ALL | 68.188.198.130/32 |  |
| All Traffic | TCP | ALL | sg-e556bf80 | -- |
| Outbound | | | | |
| All Traffic | ALL | ALL | -- | Sg-e556bf80 |
| All Traffic | ALL | ALL | -- | 0.0.0.0/0 |

{{Continues}}

1. “AWS Well Architected Framework” – Page 5 - <https://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf> [↑](#footnote-ref-1)
2. Some reuse from the BU Kuali Coeus Solution Architecture DRAFT [↑](#footnote-ref-2)
3. AWS EC2 User Guide - <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html> [↑](#footnote-ref-3)
4. See the AWS Linux 2015.09 Release Notes for more details - <https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/> [↑](#footnote-ref-4)
5. https://aws.amazon.com/rds/details/multi-az/ [↑](#footnote-ref-5)
6. http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.CommonDBATasks.html [↑](#footnote-ref-6)
7. <http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/how-elb-works.html> [↑](#footnote-ref-7)
8. <http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elb-listener-config.html> [↑](#footnote-ref-8)
9. <http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elb-configure-load-balancer.html> [↑](#footnote-ref-9)
10. <http://docs.aws.amazon.com/AmazonVPC/latest/NetworkAdminGuide/Cisco_ASA.htm> [↑](#footnote-ref-10)
11. https://aws.amazon.com/directconnect/ [↑](#footnote-ref-11)
12. https://aws.amazon.com/directconnect/faqs/ [↑](#footnote-ref-12)