

<Project Name> Architecture

<Customer Name>

Ben Fellows

Published 2020

# Overview

The purpose of this document is to document the architecture, design and a high level understanding of the requirements for the <name of system>. This document will include detailed solution design for informing infrastructure and software teams.

## Versions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Notes** | **Author** | **Date** |
| 1.0 | First release | Ben Fellows | 1 January 2020 |

## Glossary of Terms

A glossary of AWS and other technical terms used in this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AWS | Amazon Web Services – the cloud provider |
| CloudFront | Amazon CloudFront is a Content Delivery Network (CDN). The purpose of a CDN is to cache web assets including images, stylesheets, scripts and static content such as HTML pages. |
| EC2 | The Compute/ Server platform that runs Linux, Windows and other servers for hosting applications.  An EC2 Instance is essentially a virtual server host. |
| Instance | Another term for an EC2 Server. |
| MFA | Multi factor authentication: This is used to provide a second level of protection on AWS user accounts. An MFA token is stored either on a virtual device (such as a mobile phone) or hardware device (e.g. Gemalto) |
| RDS | Relational Database Service: This provides a managed database environment for popular databases including MySQL, SQL Server, Oracle and Postgres |

# Background

The goal of this project is to create a seamless platform that....

## Requirements overview

The requirement is for… <state business objectives and requirements>.

The cloud specific requirements are to provide a scalable, secure environment that can meet the needs of the various applications and databases to be hosted within AWS. This would include ensuring stability of the system and management of data at rest.

# Architecture

## Principles & Best practice

Teem adheres to a set of key principles and best practices to ensure AWS accounts are secure and environments meet customer demands:

* Security
* Privacy
* Simplification
* Performance
* Availability

Please also read our Security Statement available on request.

### Security and privacy

Due to the nature of public cloud, the first priority for customers is privacy and security - it is important that we define and build solutions that serve both the end user’s privacy requirements and protect personal and company data. As a 3rd party provider we aim to provide an approach to security that gives end users and business customers complete control over their data. We will never own their data. In addition to this AWS will never own their data.

### Simplification

The approach to designing a robust architecture is ensuring that only the components, products and features that are required will be implemented. This ensures the AWS environment can be managed operationally.

Wherever possible we reduce complexity and ask customers for feedback constantly on how to improve our products.

### Performance

We can take advantage of AWS massive scale and ways of delivering technology to constantly improve performance of our application and APIs. We’re always looking for new ways of doing this whether it be to speed up our Front End Application or improve the speed of launching servers, docker containers and APIs as well as back end noSQL/SQL storage access.

### Availability

Availability is another critical component for us and we aim to make our app available 24/7 for customers around the globe. This backed by our desire to ensure customers are well supported.

### Automation

Automation is a key pillar of any cloud environment and being able to reduce wastage and excess time spent on repeatable tasks.

## AWS Account Structure

The following account structure will be done as per best practice recommendations from AWS around account security and provisioning.

A picture containing small, monitor, computer, sitting

Description automatically generated

*Figure: AWS Account Structure*

An AWS Account provides all the tools, software and resources you need to manage your own infrastructure. At a top level all the resources in an AWS account are able to fully controlled by administrators. Within an AWS account further controls can be put in place to limit the specific access staff and contractors require.

The purpose of the separation of AWS Accounts is as follows:

* Granular security controls to production and non-production assets
* Separate production applications and storage from rest of business
* Allow developers and staff to access development machines easily
* Allow managed process around change control for production and test environments
* Allow different shut down/deletion rules between development and test environments
* Remove the risk of same access between developers and systems administrators (production)
* Allow experimentation of new features in an isolated environment.

|  |  |  |
| --- | --- | --- |
| **Account name** | **Purpose** | **Identifier/alias** |
| Billing | Billing account | billing-main |
| Security | Stores all user data, groups, roles and security | security-iam |
| Prod | Existing prod account repurposed as a pure production account for production only data | prod |
| Dev | New development account for dev servers, apps and data.  Allow developers access. | dev |

### Account Controls

Account controls increase security, optimize costs and data protection/recovery.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Root MFA | Ensure Root MFA on Account |
| CloudTrail Enabled | Cloud Trail audits event history on your AWS account |
| Cloud Config Enabled | Cloud Config allows configuration changes across resources to be monitored |
| Guard Duty Enabled | Guard Duty provides threat detection across your AWS account based on auditing |
| Trusted Advisor Notifications | This is enabled once and email summary setup weekly |
| VPC | Create a default VPC with correct CIDR |
| Private and Public Subnets | Create default subnets with assigned CIDRs |
| Security Groups | Create default management security groups |
| IAM Policies, Groups and Roles | Create standard IAM Roles, Groups and Policies to be used by staff and vendors |
| Tagging policy set | Setup basic tagging rules |
| Data lifecycle set | Sets up rules for ensuring EBS volumes are backed up. |

## Billing

Billing will be done directly through the AWS Organizations feature (see <https://aws.amazon.com/organizations/> ). This feature allows an organization to manage multiple AWS accounts and security policies for access. Billing will occur at the top level of the organization account structure. AWS Cost Explorer is a tool that can be initially used to ensure billing reports and data can be reported on by account, regions and resource types.

## Regions

AWS regions are independent of each other and provide organizations the ability to run apps in different regions and provide DR/backup locations.

AWS resources will reside in the Oregon (us-west-2) region by default.

AWS region Sydney (ap-southeast-2) will be used for data recovery and backups. (more detail under Backups).

## Key Management

Amazon KMS provides key management as a service for handling encryption of services in AWS Accounts including EBS Volumes, S3 Buckets, Snapshots and other resources.

The following KMS keys will be created in the Ohio region to be used for EBS Volume encryption and S3 encryption:

|  |  |
| --- | --- |
| **KMS Key Alias** | **Account** |
| oregon-prod | prod |
| oregon -dev | dev |

SSH Keys that are used for EC2 Key Pairs will be stored in a centralized secure password manager.

SSH Keys will only be required in normal operations for root server access.

## IAM & Users Access

IAM (Identity Access Management) in AWS is the underpinning service required to grant access to users and services or 3rd parties to be able to manage resources in AWS accounts.

This section outlines the roles, users and groups as well as methods to access these resources within the AWS Account Structure.

The approach has been to simplify as much as possible. For IAM this means creating as many repeatable resources and templates as possible. For example, the logical naming of IAM resources in all AWS Accounts will match each other. The result is simpler automation and easier ongoing maintenance of changes to IAM policies and roles.

IAM Users will only get Read Only access to AWS accounts and as an exception will be provided additional access as required through a standard request process.

### IAM Roles

IAM Roles are used for several purposes including:

* Provide Access to from another AWS Account (Cross Account Role Access).
* Provide Access for particular services within AWS to access resources in a single AWS Account. For example: EC2 instances with an Instance Profile, attached to a role for backing up data to S3.

IAM Roles will be separated into 3 categories:

* User: Roles that can only be used by an authorized cross account user.
* Service: Roles that can only be used by an authorized service within the same or different AWS account typically denoted by an ARN.
* Third Party: Roles for 3rd party applications that need to access an AWS Account. E.g. CloudHealth, Teemops, Sumologic.

The following IAM Roles will exist in each of the following AWS accounts:

* prod
* dev

|  |  |  |  |
| --- | --- | --- | --- |
| **Role Name** | **Purpose** | **Policy** | **Category** |
| company-admin-readonly | Read only access to AWS resources | AWS Managed Policy: “ReadOnlyAccess”  Policy:  “CompanyAdminRODenyPolicy” | User |
| company-admin-full | Full access to AWS | Policies:  “Administrator”  “CompanyAdminFullDenyPolicy” | User |
| company-ops | Operations access to start/stop/snapshot | Policies Attached:  “CompanyOpsPolicy”  “ReadOnlyAccess” | User |

### Cross Account Role Access

Cross account role access works as described in the diagram below. To limit how a user from the security-iam account accesses the specific account (either dev or prod) a user must have following controls put in place:

* MFA must be enabled by principal user.
* Access only allowed from security-iam account or authorized 3rd party account

## AWS Network Design

The network architecture for new VPCs in AWS will be designed on the principles of simplification and pragmatic automation. A network architecture in AWS is concerned with separating services using Security Groups and Subnets as well as connectivity with external or customer gateways. Subnets are a construct that provides more readability and ease of CIDR allocation.

It is security groups that essentially provide the secure access required between services. In addition to this, increased availability is provided by utilising multiple availability zones.

Amazon Virtual Private Cloud (Amazon VPC) enables the launching of Amazon Web Services (AWS) resources into a virtual network that is defined. This virtual network closely resembles a traditional network that would operate in a data centre, with the benefits of using the scalable infrastructure of AWS.

Subnets will be partitioned to the logical, independent tiers for naming/ip addressing, routing, readability and assignment purposes.

The following subnets will be built, suffixed by the availability zone in which they reside. Each subnet will have the following predefined attributes:

• CIDR range

• Availability Zone

• Route Table: This will be either private or public.

Graphical user interface

Description automatically generated

## Recommendations

Here are list of recommendations from Teem ensuring implementation of applications that take advantage of public cloud services.

### Secure Development Practices

You should define a process for taking a feature into General Availability. Go through a rigorous security review process to ensure the correct controls and processes are in place and that any new code is scanned prior to deployment. Ensure all environments have full auditing in place which allows you to perform any necessary forensic activities in the future. This is critical to be able to ensure that our customers data and privacy is protected under a number of privacy standards including GDPR.

*Example: An example of this is conferencing metadata. Metadata is a critical key component of ensuring users and conference rooms can be connected seamlessly. However any unauthorised access to this data poses a threat to end users' privacy. The threat is that if a bad actor (in a security context someone with malicious intent) got hold of metadata they could use this information to sell to oppressive regimes, 3rd party advertisers or any malicious purpose. conferencing metadata needs to be fully encrypted and only accessed or viewed by authorised staff.*

*Example: In the event a customer cannot access any information due to a loss of password we need to perform a break the glass technique which is audited and only can be authorised by an Executive or Authorised member of the staff.*

### Continuous Integration

Applications should be stored in source code repositories allowing them to be automatically built and tested. This allows for a number of benefits including:

* Code Scanning
* Automated Testing of Code
* Build Code before deployment
* Alerts for any code issues
* End to end automation and deployment of code
* Reduce errors.

### Scalable applications

You should design, purchase and build applications that can take advantage of the scalability features of AWS. To take advantage of this an application must be developed with the following attributes:

* Stateless (stores data in database or in a cache separate from app server).
* Licensing allows for x number of instances to be used.
* Session management / authorisation is not “sticky”.
* Separate front end, back end and data tiers

### Simplify

Ensuring that application components are simplified at the start of a new project will reduce operational complexity further down. This means selecting and choosing frameworks that are

# Questions

CIDR: What CIDR Ranges do you want?

VPN: OpenVPN

Backups

SQL Backups



**Registered Offices**

C/ - Advanced Accounting Limited (CA)

76 Fergusson Street

Feilding 4702

New Zealand

**Teem Services Limited**

<https://www.teem.nz>

Email: [help@teem.nz](mailto:help@teem.nz)

Ph: +1 (917) 259 1054

Teem is a limited liability company based in New Zealand registered in 2019.

NZBN: 9429047593206

Tax #: 129-753-935