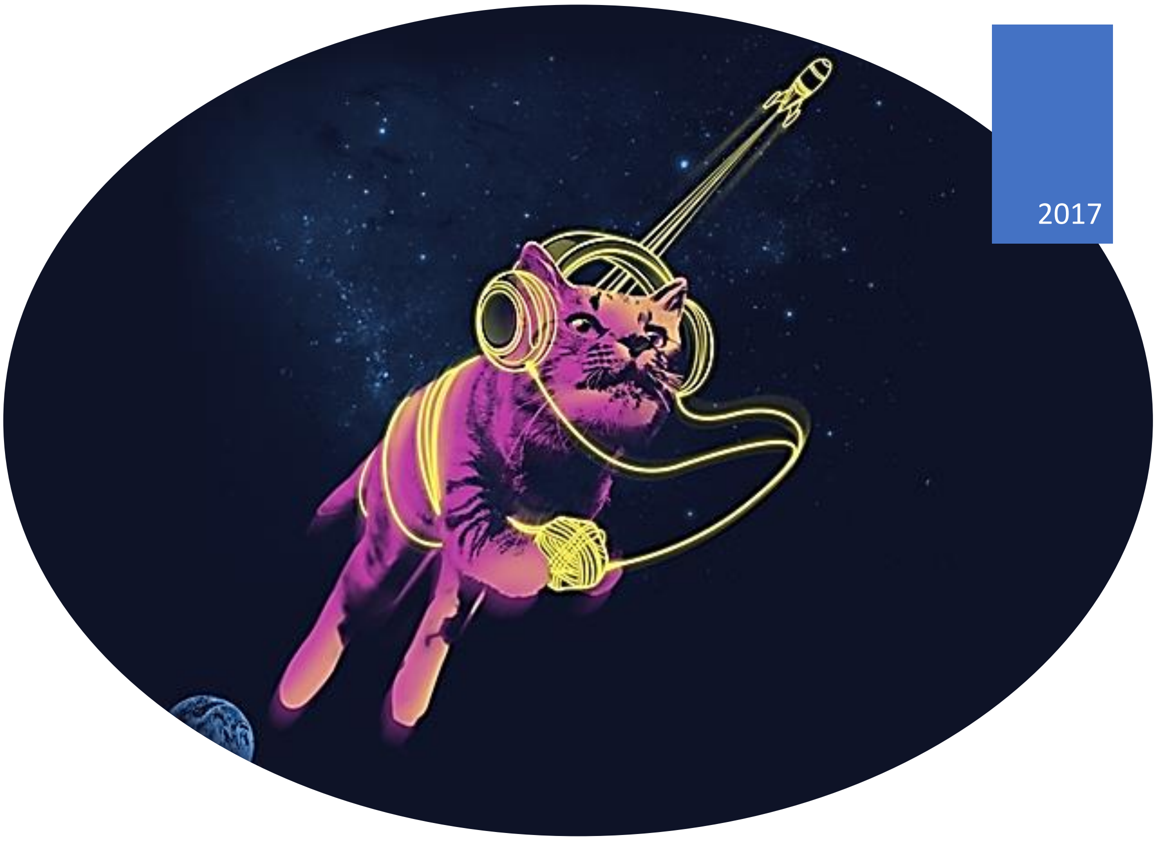

2017

Challenge 2

SECURITY DESIGN

KYLE JENNER

TERRAFORMING SPACE AGENCY

1 Contents

2 Contact Information & Document Control

The primary contacts for questions and discussions regarding this proposal are:

2.1 Document Information

Title	**Security Design**
Version	1.0
Author	Jenner, Kyle
Distribution Date	11/07/2017
Number of Pages (Excluding Cover)	19

3 Executive Summary

3.1 Document Purpose

The purpose of this document is to outline the key elements and design decisions which make up the proposed infrastructure design. The following document will include the conceptual and logical design but due to time constraints it will not include a low level physical design.

3.2 Project Overview

Terraforming Space Agency (TSA) recently designed and deployed the new infrastructure to support HumanLink version 2.0 application. Since the deployment one of the production sites was infected with a virus ransomware known as EatBrains.

EatBrains is an advanced malware that infects files and maps SMB shares in order to spread itself further. There is also a reported variant that has the potential to store itself in memory on network devices and to emulate routers within the environment. EatBrains is also capable to phone home for remote execution using the infected devices in the environment.

The virus was identified following an outbreak in one of the sites but the virus was brought under control. The following report will address the security issues identified in the infrastructure following an outbreak report and what changes will be implemented.

The outbreak report suggested a recovery plan of the infrastructure along with a documented walk through on how to recover the HumanityLink application should it be compromised.

3.3 Design Qualities

The following design qualities will be referenced.

Qualities	Ref	Example
Availability	DQA	System up time to deliver SLAs.
Manageability	DQM	Simplified management layer to reduce overall efficiency.
Performance	DQP	Ensure system performance to meet project requirements.
Recoverability	DQR	Ability to recover from a failure.
Security	DQS	Authorization and access to the system.

4 Conceptual Design

4.1 Requirements

Requirement	Ref
Identify security risks and implement changes to the design to mitigate the risks.	REQ001
Identify security risks in the application.	REQ002
The solution must be able to identify where a potential intruder would access the system and has remote execution capability.	REQ003
Produce a recovery plan for the infrastructure and application stack.	REQ004
Document a walkthrough to recover HumanLink application.	REQ005

4.2 Constraints

Constraint	Ref
Existing infrastructure must be used	CON001

4.3 Risks

Risk	Impact	Mitigation	Ref
The identified virus may evolve into a new variant.	High	The method used to protect against the known virus must have an ability to identify and prevent the spread of any new variant of the virus.	RIS001
Backup and replication media could become infected by the spread of ransomware.	High	Implement a backup solution that protects against such an event.	RIS002
No RPOs and RTOs have been specified in the project requirements for critical systems.	High	No specific RPO/RTO has been specified, TSA has assumed 4 hour RPO and 2 hour RTO is acceptable to production and will design components to meet this.	RIS003

4.4 Assumptions

Assumption	Ref	Additional Information
The first outbreak of the virus was detected and isolated manually.	ASU001	A virus breakout was detected manually and then identified as EatBrains.
No existing recovery plan exists.	ASU002	Currently there is no recovery plan.
Virus pattern for EatBrains exist.	ASU003	Now the virus has been detected a virus pattern has been released.
The virus outbreak didn’t affect the production HumanLink.	ASU004	The virus outbreak didn’t affect the production HumanLink only Windows VMs.
All of the Zombies are gone the threat was from another human.	ASU005	Security vulnerability does not need to include physical datacentre protection.
The robots were not affected by the virus.	ASU006	Only workloads in the datacentre were affected by the virus not the robots.
Assumed acceptable guaranteed level of availability is 99.9%	ASU007	No specific uptime has been stipulated.

4.5 Conceptual Design Overview

The following conceptual design will be a very high level view on what the solution will look like after completion. The conceptual design does not include any sizing figures, vendors or product specifications.

For the purpose of this document the design will be split into two elements, one will be referred to "Infrastructure" that is all workloads running on VMware on AWS and the other will be referred to "application" which is the production application running in native AWS.

4.5.1 Infrastructure Security Conceptual Design

The datacentre design stretches across 3 sites with each site including 3 availability zones. The outbreak was isolated to only one site the design proved resilient in this instance, however following the outbreak the security of the infrastructure will be tightened.

Each site already shares security constructs between them but this will be leveraged further by adding an anti-virus solution to be able to catch and quarantine any infected workload. Security will be enhanced by applying a "zero-trust" model following an assessment of the network.

End users will have their endpoints reviewed and changed to reflect a more secure method of connecting as well as their back end session. It's critical that the users can still connect from their remote location but can work from a secure workplace.

Additional monitoring tools will be added to centrally manage log files and collate them with the existing monitoring solution to provide faster and easier access to information.

The existing backup and replication solution will be tweaked to provide the required RPO / RTOs, the solution itself already proved critical in the recovery of infected machines.

Additional solution will be added to the infrastructure to be able to identify any intruder that has already breached security and has remote execution capabilities.

4.5.2 Application Security Conceptual Design

Application HumanLink is deployed into AWS but developed on the infrastructure referred to above, developers will access their own development environment remotely that will be secured further by running anti-virus at a hypervisor level as well as instant quarantine from the network. Developers will also fall in line with the "zero-trust" approach and only be allowed to communicate with the services they require.

Luckily this strain of the virus didn't affect the production application but it has instigated changes to make sure the deployment is more secure.

This will be done by having stricter control over user access, resource policies and by protecting data at rest and in transit. Further protection for malware will also be covered and added.

The robot army was not affected by the virus nor were the IOT devices at each terraform site, it will be proposed however to change remote devices the scientists, engineers and developers use to increase security.

5 Logical Design

The following logical design will take the conceptual design and put together a solution that will use technologies to meet the requirements. Each section will list a design decision and link it to the requirement for reference. Each design decision has considered the risks and constraints highlighted above.

5.1 Logical Design Overview

The premise of this design is to implement further security enhancements following a recent security breach. As with the conceptual design the design will be split into two elements, one will be referred to "Infrastructure" that is all workloads running on VMware on AWS and the other will be referred to "application" which is the production application running in native AWS.

The infrastructure was built for scalability and for reliance but was still vulnerable to a security breach. The bare metal servers running on VMware on AWS are controlled by VMware, no further changes can be made here but a security zone will added for internet facing services. vSAN is also be changed to use encryption at rest.

NSX security policies will be extended further by applying a "zero-trust" model, vRealize Network Insight (vRNI) will be configured to assess the infrastructure and only allow the required flows.

Trend Micro Deep Security will be introduced and integrated with the existing NSX to provide anti-virus and malware protection to VMs but also to provide further guest and network introspection services.

Further NSX integration will be implemented across the Horizon environment to secure the scientists, engineers and developers virtual desktop. The desktop pools will be configured in a way to make then disposable should one be infected. Applications are already virtualized and stored as read only and users persona will be stored on VMs protected by NSX and Trend Micro.

Splunk will be introduced to gather logs from the infrastructure and application environment with additional security add-ons for intrusion prevention and detection.

Rubrik is already in place to be able to recovery almost instantly should a machine be infected, each rubrik is read-only to the VM networking meaning a ransomware attack couldn't put the backup data at risk.

Moving onto the application, production is running in native AWS and due to the structure of the application the EatBrains virus but the infrastructure will be made more secure.

The production infrastructure will move to a move individual IAM accounts than it was previously enabling a least privilege environment furthermore each ECS task can be specified to a IAM to call the APIs and deploy the application. Further IAM integration can be made with DynamoDB to apply ACL permissions over resources.

Data at rest encryption for the DynamoDB will not be implemented but protection of data in transit will be configured to be accessed across the terraforming sites over the internet.

Finally, the remote scientists, engineers and developers will be given different endpoint devices to be able to connect to the Horizon environment and with it enabled with 2 form authentication devices.

5.2 Infrastructure Security Logical Design

An additional edge cluster will be configured within the VMware on AWS infrastructure to segregate internet facing services such as Access Points used for Horizon. Only the required ports will be allowed from external sources, further DFW will be enabled on the guest VM following a zero-trust model.

DDN001	Decision – Edge Cluster
	Justification – To provide extra layer of protection to minimize the risk of unauthorized access.
	Type – DQS
	Impact – Additional Edge Service Gateway (ESG) and logical switches to be configured for the edge cluster.
	Associated Risk – ESG appliance represents a SPOF
	Risk Mitigation – Deploy ESG in HA mode.
	Reference – REQ001 CON001 ASU007

vSAN across all 3 sites will be enabled for encryption at rest encrypting the vSAN datastore without the need for any special hardware such as encrypted disks. Both the flash tier and capacity tier is encrypted.

DDN002	Decision – Enable vSAN encryption.
	Justification – vSAN adds encryption at the datastore level protecting all objects in the vSAN datastore with no requirements on the hardware.
	Type – DQS
	Impact – vSAN encryption is enabled at the cluster level for each site, vSAN features such as Erasure Coding and usual vSphere features such vMotion and HA are all still supported.
	Associated Risk – A 3rd party Key Management Server (KMS) is required, these keys can be exposed and compromised.
	Risk Mitigation – Deploy HyTrust DataControl as the KMS system and regularly change the keys.
	Reference – REQ001 CON001

vRealize Network Insight (vRNI) will be introduced to be able to monitor the environment further, using vRNI information around the flows can be collected and used to deploy a zero-trust approach and only allow access required when configuring DFW policies.

DDN003	Decision – Deploy vRNI
	Justification – To assist with applying a zero-trust model along with audit and compliance features by looking back on changes within the network.
	Type – DQM, DQS
	Impact – Additional vRNI appliance is to be deployed.
	Associated Risk – Single vRNI appliance represents a SPOF.
	Risk Mitigation – Deploy vRNI in cluster mode.
	Reference – REQ001, REQ002 ASU007

Trend Micro Deep Security will be deployed to enable agent less anti-virus protection for all VMs within the infrastructure. NSX integration register Deep Security for advanced security services providing both NSX Guest Introspection and Network Introspection services.

DDN004	Decision – Deploy Trend Micro Deep Security.
	Justification – To enable agentless anti-malware, web reputation, additional firewall services and intrusion prevention.
	Type - DQS
	Impact – Deep Security virtual appliance must be deployed all hosts along with a centralised manager.
	Associated Risk – As the environment scales an appliance must be deployed on each host, failure to do so could leave workloads at risk.
	Risk Mitigation – Enable automatic provisioning of the virtual appliance to the protected clusters.
	Reference – REQ001, REQ003 ASU001, ASU003, ASU005, ASU006

Horizon pod security will be increased further by enabling agentless anti-malware protection. Already the desktops are stateless with the applications virtualised and user persona saved centrally. During the initial outbreak the infected machines were quickly quarantined and deleted allowing the users to simply log back onto another uninfected virtual desktop.

DDN005	Decision – Trend Micro Deep Security enabled for Horizon desktops.
	Justification – To enable a secure workplace for scientists, engineers and developers to work from.
	Type – DQA, DQR, DQS
	Impact – Each ESXi host within each pod will have the Deep Security virtual appliance deployed.
	Associated Risk – Outbreaks can spread across the network once a machine has been infected.
	Risk Mitigation – Enable automatic quarantine within Deep Security. Deploy a zero-trust model across all desktops to mitigate the risk of any infection spreading.
	Reference – REQ001 ASU001, ASU003

Splunk will be deployed to collect logs thought the infrastructure and within native AWS to quickly identify instruction attempts over a period of time by looking at things such as failed login attempts. Splunk can also pick up suspicious activity within the network and put it in context from what is normal from that endpoint.

Splunk Enterprise will be deployed with additional Splunk Enterprise Security solution extension to enhance security investigation and prevention. Splunk User Behaviour Analytics (UBA) will also be added to be able to detect internal threat through machine learning and know what sort of behaviour with the infrastructure is suspicious.

Some of these services will overlap with Trend Micro Deep Security but no one tool can catch all threats so it recommended to have a layered approach.

Splunk can also be integrated with vSphere, NSX, AWS and Trend Micro Deep Security through Splunkbase add-ons.

DDN006	Decision – Deploy Splunk
	Justification – To collate logs in the environment for easier tracking of events and to add to the protection but watching for unusual behaviour internally.
	Type – DQM, DQS
	Impact – Splunk Enterprise is to be configured along with Enterprise Security and UBA.
	Associated Risk – Splunk deployment represents a SPOF.
	Risk Mitigation – Implement Splunk components into clusters – Indexer Cluster and Search Head Cluster.
	Reference – REQ001, REQ002 ASU001, ASU003

During the initial breakout Rubrik was leveraged to restore services, this was critical in the restore process providing both instant recovery but also backup files that were isolated from the threat as they are stored in read-only.

It is recommended here to change the configured RPOs increase availability whilst leveraging instant recovery to achieve near to zero RTO.

DDN007	Decision – Configure Rubrik with smaller RPOs.
	Justification – Smaller RPO will provide greater availability options, these will be changes to 4 hours at each site.
	Type – DQA, DQR
	Impact – Backup policies will be configured to reflect the smaller RPO.
	Associated Risk – New VMs need to be added to the policy or could be at risk.
	Risk Mitigation – Leverage RESTful APIs to automate assignment when new VMs are created.
	Reference – REQ001 ASU002

5.3 Application Security Logical Design

The production HumanLink application was not affected by the virus directly but it did trigger the requirement to investigate what security measurements that could be added. The application is containerised and doesn't rely on SMB which prevented the EatBrains virus reaching the application.

The first change is to implement individual Identity and Access Management (IAM) accounts to implement a least privilege security model and only assign rights to users that they require.

DynamoDB is tightly integrated with IAM, using IAM tools ACL-type permissions can be applied to individual resources. Furthermore, Amazon EC2 Container Service (ECS) allows for specific IAM role for each ESC task allowing the instance role and task role to be managed separately.

DDN008	Decision – Implement individual IAM accounts.
	Justification – To implement a least privilege security model.
	Type – DQS
	Impact – Each user will have their own IAM account and set of access keys.
	Associated Risk – Access to a highly-privileged account is compromised.
	Risk Mitigation – Splunk UBA has been deployed, Splunk UBA will analyse user's normal behaviour and flag it when the user's activity suddenly changes and is seen as suspicious.
	Reference – REQ002 ASU004

AWS Security groups will be deployed to further add to the zero-trust model, security groups act as a firewall for associated container instances within ECS. Inbound and outbound traffic can be controlled at the container instance level.

For the multiple regions security groups will need to be configured in each region.

DDN009	Decision – AWS Security Groups will be configured where possible.
	Justification – To implement a zero-trust security model.
	Type – DQS
	Impact – AWS services will be configured to only talk to other services they need to.
	Associated Risk – Additional deployment steps and risk to human error.
	Risk Mitigation – Add security group configuration to the Terraform deployment code.
	Reference – REQ002 ASU004

As the production HumanLink application is accessed from the terraforming sites via the internet the data in transit will be further secured. This will mitigate the risk of accidental information disclosure and not compromise the data integrity.

DDN010	Decision – Protect data in transit.
	Justification – To protect against accidental information disclosure and to know that the data integrity has not been compromised.
	Type – DQS
	Impact – Data in transit will be configured to use SSL/TLS.
	Associated Risk – The application has not been developed to support SSL.
	Risk Mitigation – Developers will need to update the application.
	Reference – REQ002 ASU004

Docker container provides inherited security due to the nature of restricted capabilities. Processes such as web servers that just need to bind on a port do not have to run as root, they can just be granted net_bind_service capability instead. Where an average VM would require a lot of processers to run as root containers do not as a lot of the processes such as network and hardware management tools are handled by the infrastructure.

Containers do not need root access allowing further security measures to be added to restricting a malicious user's toolset and attack surface. Moving towards a whitelist model rather than blacklist and only allowing capabilities as required.

Docker by default drops all capabilities except those needed. For additional security AppArmour will be added, AppArmour is a Linux security model that protects an operation system and its application from security threats.

DDN010	Decision – Deploy AppArmor.
	Justification – To further secure Docker containers over and above the default.
	Type – DQS
	Impact – AppArmor security profile is associated with each program.
	Associated Risk – Developers need to add further integration for AppArmour which may affect the application.
	Risk Mitigation – Revert to the default Docker security measure.
	Reference – REQ002 ASU004

Remote scientists, engineers and developers are connecting to their Horizon pod remotely using their tablet devices, currently the devices used are a mismatch of manufactures. Standardised devices will be rolled out to enable biometric authentication.

DDN010	Decision – Standard remote devices enabled with biometric authentication.
	Justification – To secure remote endpoints from unauthorised access.
	Type – DQS
	Impact – The Horizon environment needs to be configured to support biometric authentication along with the remote client.
	Associated Risk – Not all tablet devices support biometric authentication.
	Risk Mitigation – Deploy only compatible devices such as iPad Air 2.
	Reference – REQ001, REQ002 ASU004

6 Estimated Duration and Timetable

As soon as humanely possible, there are hackers everywhere man!

7 Operational Guide

7.1 Infrastructure Recovery Plan

The following operation guide will cover the high-level steps to recover from a security compromise. Following the further NSX and Trend Micro Deep Security integration should a machine be compromised the VM will automatically go into quarantine to make sure no further infection can spread.

Following the detection of the malware the VM will be assigned a security tag, this tag will then be consumed by VMware NSX Service Composer which in turn will move the VM into a quarantined security group. This quarantined security group can then be configured to restrict all firewall communications to prevent any further communication on the network.

The administrator will receive an alert from vROps and potentially Splunk to which the quarantined VM can be cleaned before going back into production. In the case of EatBrains the recovery steps will vary depending on the infected system.

If the infected system is a production VM such as the file server used to store user's profiles on from VMware UEM. This VM once infected will encrypt the data and most likely won't be able to be cleaned. For this scenario restore the VM from Rubrik using instant restore, the data at most will be 4 hours old.

If the infected system is a virtual desktop however, the administrator can simply delete this VM from Horizon and ask the user to log back onto another virtual desktop from the pool. Once logged back in the user will have their applications attached again which are stored in read only and the users profile will be loaded back in.

By using VMware NSX in a zero-trust approach this should mitigate against the EatBrain malware spreading across the network but reducing the attack service and by using Trend Micro Deep Security the infected machines life on the network is limited.

Patching the environment is a critical task for the admin, VMware vSphere hosts will be regularly patched via their VMware on AWS service. The admin is responsible for updating server workloads, this manual task can be automated by using Microsoft WSUS and also by using update VMware templates when deploying servers.

Horizon desktops can be easily maintained by updating the gold images regularly and rolling out the updated image when users log off their sessions providing no downtime to users.

Splunk will provide mountains of information of correlated logs and by using UBA the admin will be alerts if suspicious activity is noticed, the admin must investigate watch of these alerts.

7.2 Application Recovery Plan

HumanLink application v2.0 has been developed to be a containerised scalable application, the following guide will assist the admin to recover the application should any security breach affect it.

The application is deployed via Terraform as code and must be deployed into each region and the Docker image is loaded to run on Amazon ECS which pulls all the data into a centralised DynamoDB.

In the event on an attack on the application the application can be redeployed onto the same infrastructure or any other infrastructure. By leveraging Terraform the code can simply redeploy the infrastructure as required.

DynamoDB is the scalable database for the application but it doesn't include any native backup or restore method this will need to be done by the admin. To do this the admin needs to use a created IAM user with a AWS Access Key with FullAccess permissions to the DynamoDB.

Disclaimer - the following has not been tested but is included for illustration purposes

The admin needs to use a batch tool to work with boto found at - https://github.com/bchew/dynamodump

To back up every table in a region run the following.

```
python dynamodump.py -m backup -r region -s "*" --accessKey AWS_ACCESS_KEY
--secretKey AWS_SECRET_KEY
```

To back up a single table run the following.

```
python dynamodump.py -m backup -r region -s TableName --accessKey
AWS_ACCESS_KEY --secretKey AWS_SECRET_KEY
```

To restore every table run the following.

```
python dynamodump.py -m restore -r region -s "*" --accessKey AWS_ACCESS_KEY
--secretKey AWS_SECRET_KEY --schemaOnly
python dynamodump.py -m restore -r region -s "*" --accessKey AWS_ACCESS_KEY
--secretKey AWS_SECRET_KEY --dataOnly
```

To insert a single table

```
python dynamodump.py -m restore -r eu-west-2 -s TableName --accessKey
AWS_ACCESS_KEY --secretKey AWS_SECRET_KEY --schemaOnly
python dynamodump.py -m restore -r eu-west-2 -s TableName --accessKey
AWS_ACCESS_KEY --secretKey AWS_SECRET_KEY --dataOnly
```

Below is an example of Terraform code that will provision AWS ECS cluster.

Disclaimer - the following has not been tested but is included for illustration purposes, additional configuration will be required by the admin.

```
provider "aws" {
   access_key = "${var.aws_access_key}"
   secret_key = "${var.aws_secret_key}"
   region = "${var.region}"
}

resource "aws_key_pair" "user" {
   key_name = "user-key"
   public_key = "${file(var.ssh_pubkey_file)}"
}

resource "aws_vpc" "main" {
   cidr_block = "10.0.0.0/16"
   enable_dns_hostnames = true
}

resource "aws_route_table" "external" {
   vpc_id = "${aws_vpc.main.id}"
   route {
      cidr_block = "0.0.0.0/0"
      gateway_id = "${aws_internet_gateway.main.id}"
   }
}

resource "aws_route_table_association" "external-main" {
   subnet_id = "${aws_subnet.main.id}"
   route_table_id = "${aws_route_table.external.id}"
}

# TODO: figure out how to support creating multiple subnets, one for each
# availability zone.
resource "aws_subnet" "main" {
   vpc_id = "${aws_vpc.main.id}"
   cidr_block = "10.0.1.0/24"
   availability_zone = "${var.availability_zone}"
```

```
}

resource "aws_internet_gateway" "main" {
  vpc_id = "${aws_vpc.main.id}"
}

resource "aws_security_group" "load_balancers" {
  name = "load_balancers"
  description = "Allows all traffic"
  vpc_id = "${aws_vpc.main.id}"

  # TODO: do we need to allow ingress besides TCP 80 and 443?
  ingress {
    from_port = 0
    to_port = 0
    protocol = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # TODO: this probably only needs egress to the ECS security group.
  egress {
    from_port = 0
    to_port = 0
    protocol = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_security_group" "ecs" {
  name = "ecs"
  description = "Allows all traffic"
  vpc_id = "${aws_vpc.main.id}"

  # TODO: remove this and replace with a bastion host for SSHing into
  # individual machines.
  ingress {
    from_port = 0
    to_port = 0
```

```
        protocol = "-1"
        cidr_blocks = ["0.0.0.0/0"]
    }

    ingress {
        from_port = 0
        to_port = 0
        protocol = "-1"
        security_groups = ["${aws_security_group.load_balancers.id}"]
    }

    egress {
        from_port = 0
        to_port = 0
        protocol = "-1"
        cidr_blocks = ["0.0.0.0/0"]
    }
}


resource "aws_ecs_cluster" "main" {
    name = "${var.ecs_cluster_name}"
}

resource "aws_autoscaling_group" "ecs-cluster" {
    availability_zones = ["${var.availability_zone}"]
    name = "ECS ${var.ecs_cluster_name}"
    min_size = "${var.autoscale_min}"
    max_size = "${var.autoscale_max}"
    desired_capacity = "${var.autoscale_desired}"
    health_check_type = "EC2"
    launch_configuration = "${aws_launch_configuration.ecs.name}"
    vpc_zone_identifier = ["${aws_subnet.main.id}"]
}

resource "aws_launch_configuration" "ecs" {
    name = "ECS ${var.ecs_cluster_name}"
    image_id = "${lookup(var.amis, var.region)}"
```

```
  instance_type = "${var.instance_type}"
  security_groups = ["${aws_security_group.ecs.id}"]
  iam_instance_profile = "${aws_iam_instance_profile.ecs.name}"
  # TODO: is there a good way to make the key configurable sanely?
  key_name = "${aws_key_pair.alex.key_name}"
  associate_public_ip_address = true
  user_data = "#!/bin/bash\necho ECS_CLUSTER='${var.ecs_cluster_name}' > /etc/ecs/ecs.config"
}



resource "aws_iam_role" "ecs_host_role" {
  name = "ecs_host_role"
  assume_role_policy = "${file("policies/ecs-role.json")}"
}

resource "aws_iam_role_policy" "ecs_instance_role_policy" {
  name = "ecs_instance_role_policy"
  policy = "${file("policies/ecs-instance-role-policy.json")}"
  role = "${aws_iam_role.ecs_host_role.id}"
}

resource "aws_iam_role" "ecs_service_role" {
  name = "ecs_service_role"
  assume_role_policy = "${file("policies/ecs-role.json")}"
}

resource "aws_iam_role_policy" "ecs_service_role_policy" {
  name = "ecs_service_role_policy"
  policy = "${file("policies/ecs-service-role-policy.json")}"
  role = "${aws_iam_role.ecs_service_role.id}"
}

resource "aws_iam_instance_profile" "ecs" {
  name = "ecs-instance-profile"
  path = "/"
  roles = ["${aws_iam_role.ecs_host_role.name}"]
}
```

In addition to Terrafom the following script can be used as an example for doing deployments to ECS,

```
$ python deploy/ecs-deploy.py deploy --cluster=<cluster> --service=<service> --image=<image>
```

8 References

vSAN Security Zone - https://www.vmware.com/files/pdf/products/vsan/vmware-vsan-security-zone-solution-overview.pdf

Trend Micro Deep Security - https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nsx/vmware-trendmicro-nsx-reference-architecture.pdf

Trend Micro Deep Security with NSX - https://www.trendmicro.tw/cloud-content/us/pdfs/sb02_ds_vmware_nsx.pdf

HyTrust KMS - https://www.hytrust.com/uploads/HyTrust-DataControl.pdf

vSAN data at rest encryption - https://blogs.vmware.com/virtualblocks/2017/04/11/vsan-6-6-native-data-at-rest-encryption/

vRNI on AWS - https://blogs.vmware.com/management/2017/06/gaining-insight-aws-workloads-vrni-3-4.html

Rubrik Air gap - https://www.rubrik.com/wp-content/uploads/2017/05/Air-Gap-Isolated-Recovery-and-Ransomware-Cost-vs.-Value.pdf

AWS Security Whitepaper - https://d0.awsstatic.com/whitepapers/Security/AWS_Security_Best_Practices.pdf

Configure Amazon ECS - http://docs.aws.amazon.com/AmazonECS/latest/developerguide/get-set-up-for-amazon-ecs.html#create-a-base-security-group

DynamoDB IAM - http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/authentication-and-access-control.html

Splunk Cluster - http://docs.splunk.com/Documentation/Splunk/6.6.2/Deploy/Useclusters

DynamoDB restore

https://medium.com/@MrCskncn/guide-to-backup-and-restore-big-tables-in-dynamodb-completely-44dd1bd73446

Terraform example - https://github.com/gruntwork-io/intro-to-terraform

Terraform, Docker and ECS - http://www.ybrikman.com/writing/2016/03/31/infrastructure-as-code-microservices-aws-docker-terraform-ecs/

Docker security - https://docs.docker.com/engine/security/security/#linux-kernel-capabilities

AppArmor with Docker - https://docs.docker.com/engine/security/apparmor/

Terraform ECS - https://github.com/alex/ecs-terraform