Architecture_for_kubernetes

Last edited by **czerniga-gft** 8 months ago

Architecture for kubernetes

<a href="https://git.gft.com/virtual-banking/banklitex/architecture"

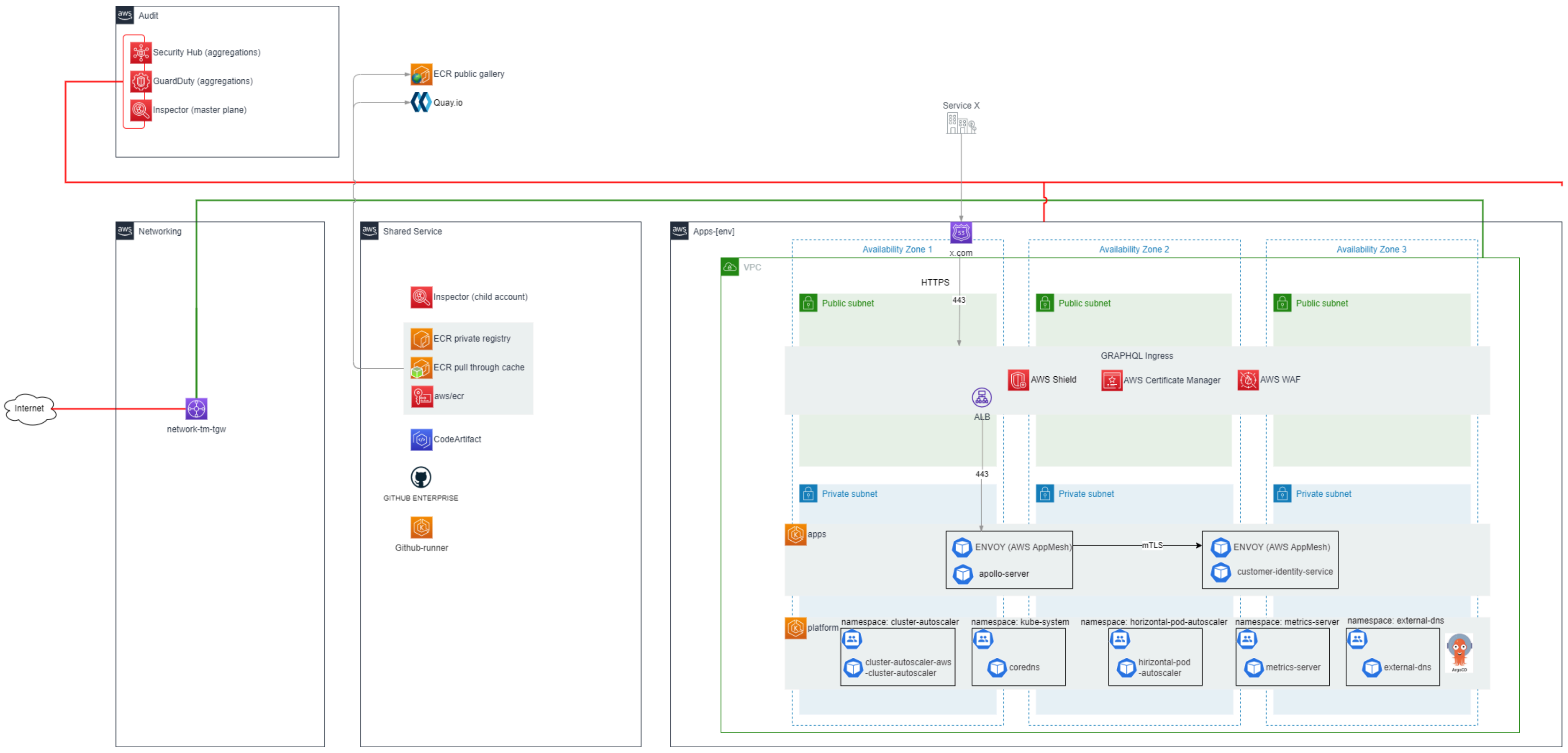
rel="nofollow">https://git.gft.com/virtual-banking/banklitex/architecture

        - Aside from the context application(s), what services are running on the cluster(s). Examples could include DataDog collection agent running as a daemonset, Kubernetes Dashboard, etc.
        - Are PodSecurityPolicy's defined? What policies?
        - Can privileged containers be run? Are there PodSecurityPolicies that controls when privileged containers can be run?
        - Are containers produced by the organization designed to run under a non-root user? Is there a PodSecurityPolicy that controls when containers can be run as root?
        - How are images built? Are there any CI build workers that run on the cluster? Do they use Docker-in-Docker or volume-mount the socket?
        - Do any pods use hostPath volume mounts?
        - Are requests and limits set for all pods?
        - Is sudo, passwd or any other binaries with the SUID or SGID bit set left in container images?
        - Are containers set to have a read-only root filesystem.
    - Multi Tenancy
        - Is there a requirement for tenant isolation? What level of isolation is required (soft vs hard multi-tenancy)?
        - Are there segregated clusters to handle hard multi-tenancy requirements? What are the clusters and what is running on each?
        - Are namespaces being used? How are they structured?
        - Is self-service provisioning of new namespaces required, or will they be created by an ops team?
        - Is there any requirement to enforce that certain workloads do not share worker nodes?
        - Are quotas defined on namespaces?
    - Detective Controls
        - Is cloud trail enabled?
        - Are control plane audit logs enabled?
        - How are audit logs for both infra and EKS API calls analyzed and monitored?
    - Networks Security
        - Are EKS control plane endpoints set to private?
        - What network policies if any have been defined, if any?
        - What security groups have been defined on the control plane and data plane?
        - Are security groups for pods in use? What rules have been defined?
        - What is the organization's policy for encryption? Is end-to-end encryption required?
        - Where is TLS used and where is not used?
        - o On load balancer services?
        - o Ingresses?
        - o At pods e.g. with AppMesh or another sidecar proxy?
        - How are TLS certificates managed and deployed?
    - Data Encryption and Secrets Management
        - What data volumes are needed by applications and are they encrypted?
        - Are worker node boot volumes encrypted?
        - Are there any secrets required by applications? E.g. connection strings, API keys, shared secrets. Are these encrypted using AWS KMS envelope encryption?
        - What back-end is used for storing secrets?
        - What is the policy for rotating application secrets and how is it implemented?
        - How are secrets provide to workloads? E.g. through environment variables or volume mounts. If environment variables are used is there any risk of leakage to log files? Has this been audited?
        - Are humans allowed to SSH to nodes?
    - Infrastructure Security
        - Are humans able to directly access worker nodes? How do they do this? What is the purpose of this access?
        - Have the CIS benchmarks for Kubernetes been run? How often are they run?
        - What are the subnets for worker nodes? Are they public or private?
        - Are there any automated security assessments run against the EC2 worker nodes? Has Amazon Inspector used to assess infrastructure?
        - Are there any requirements within the organization to bake intrusion detection products into the worker nodes? Has this been done and if so how?
    - Regulatory Compliance
        - Are there regulations that the infrastructure and applications need to comply with? Which regulations?
        - How is compliance handled?
        - How is compliance monitored regularly?
        - Are remediations automated?
    - Incident Response and Forensics
        - Is there an incident response plan for the case where a pod is compromised or exhibits suspicious behavior?
        - What penetration testing methodology is used, if any?
    - Image Security
        - What base images are used?
        - How are container images created?
        - Are images hardened / de-fanged?
        - Are container images scanned for vulnerabilities? How, where and when does this happen?
- Reliability
    - Atutomatically recovery from failure
        - How do you ensure that enough IP address space is available as pods are scheduled and rescheduled?
        - Are deployments spread across AZ's? Are there any deployments that reside in a single AZ? If so, why?

    - Test recovery procedure
        - Do you have resiliency testing to verify that worker nodes are replaced in the case of a node failure?
        - Do you have resiliency testing to verify that stateful workloads successfully reclaim persistent volumes when re-scheduled after a failure?
    - Scale horizontally to increase aggregate workload availability
        - Is the Horizontal Pod Autoscaler applied to stateless deployments?
        - Is the Cluster Autoscaler deployed?
- Performance Efficiency
    - Design Principles
    - Democratize advanced technologies
        - Can engineering teams deploy dev/test EKS clusters in a self-service manner, or do such activities need to be routed through a central team?
        - Can engineering teams manage their own application deployments, or do such activities need to be routed through a central team?
    - Experiment more often
        - What instance types are used? What is the basis for the choice of instance types?
        - What EBS volume types are used for persistent volumes? What is the basis for the choice of volume types?
- Cost Optimisation
    - Design Principles
    - Adopt a consumption model
        - How are requests and limits sized for workloads?
        - Could the Vertical Pod Autoscaler help tune pod sizing?
        - Would it be feasible to use spot instances for your cluster(s)? Note the workload(s) need to be fault-tolerant to run successfully on spot.
        - Would it be feasible to use Fargate for your cluster(s)? Note Fargate does not support: Classic LB or NLB; Daemonsets; privileged containers; HostPort or HostNetwor in pod specs; GPU;
    - Measure overall efficiency
        - What level of utilization are you currently achieving for worker nodes?
        - Are you taking any specific measures to improve node utilization?
        - Are there mechanisms in place to control costs for dev/test workloads? For example terminating clusters overnight or using Fargate launch type in dev/test.
    - Stop spending money on undifferentiated heavy lifting
        - Do you use self-managed or AWS-managed node groups?
        - Do you use eksctl for deploying your cluster(s), or CloudFormation or some other IaC, or another method?
    - Analyze and attribute expenditure
        - Is there any requirement for resource consumption by applications to be charged back to business units?
        - Are there shared clusters where applications from different business units run and which need to be separately charged back to those BU's?
        - Are there mechanisms in place to detect orphaned or unattributed infrastructure in dev/test environments to avoid cost overruns?

Diagrams

High-level architecture for integration layer (applications) kubernetes cluster



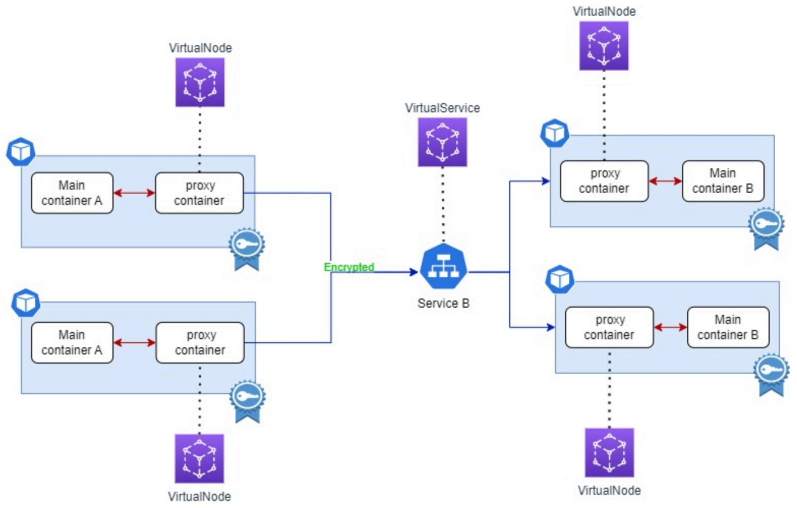
1. Apps
    1. Environment contains dedicated VPC with public and private subnets divided between three availability zones. Public subnets contains the public load balancers whereas the private subnets contains the internal resources like EKS cluster with worker nodes, Kafka cluster, Database and the rest of private resources.
    2. Application connects to the GraphQL endpoint over the HTTPS protected session. The traffic is then load balanced between the pods of graphql application deployment. We are using Route53 as a DNS service, ACM as a endpoint certificate provisioning service and AWS Shield & WAF to properly protect the endpoints.

    3. Communication within the cluster (for example between the graphql server and other microservices) is encrypted thanks to the usage of service mesh and internal certificates generated with Cert Manager platform application.
    4. Microservices and all the base platform maintenance applications (like cluster autoscaler or metrics server) are deployed as a Helm charts using the ArgoCD GitOps application. Whole infrastructure is deployed using Terraform. The helm charts contains the references to the microservice images which are placed on the Shared Service account
2. Shared Service
    1. Environment consist of three main parts:
        1. ECR repositories, which contain the images of developed microservices. Third party images are pulled from the public repositories like ECR public gallery or Quay.io and also stored in our private registries
        2. CodeArtifact service, which is used to store the artifacts of the custom plugins utilized for the development purposes
        3. EKS cluster used as an runtime environment for GitHub actions runners, which are mostly used to build and deploy new versions of microservices against the ECR repositories
3. Networking
    1. Networking account contains the transit gateway which allows all the accounts to communicate privately without exposing the traffic to the public internet. Traffic directed from inside of the clusters to the publicly available services is also routed via this account.
4. Audit:
    1. Audit accounts contains the audit related services like Security Hub, GuardDuty and Inspector

High-level architecture for shared services kubernetes cluseter

(insert image her)

Architecture of AppMesh based encryption in transit



1. After enabling AppMesh service mesh for chosen namespace, each of the pod is enriched with dedicated proxy Envoy sidecar container
2. Envoy proxies are responsible for intercepting the traffic incoming and outgoing from the pod, which giving us the additional capabilities in a context of communication security, for example:
    1. enabling encryption in transit for communication between the pods - traffic between pods is properly encrypted using the application dedicated certificates
    2. enabling mTLS authentication - not only client is checking the server certificate like in regular TLS, but server is also verifying the client certificate before establishing the encrypted connection
    3. network policies - AppMesh giving us a possibility to define a whitelist of applications to which our microservice is able to connect (to reduce the blast radius in case of microservice would be compromized)
    4. generating custom communication related metrics - to increase our visibility over the traffic between the pods
3. VirtualNodes and VirtualServices are the AppMesh dedicated kubernetes resources representing the configuration of the mentioned Envoy proxy sidecars, and could be used to properly implement the features listed above
4. Traffic encryption is not established between the applications directly, but between their dedicated proxy containers before traffic would leave the pod, so the application developers do not need to implement additional functionalities to make use of the benefits of the service mesh

Operational Excellence

Cluster Operations

How are eks cluster and nodes provisoined and maintained?

AWS managed instances through node groups. Using Amazon LInux 2 optimised for EKS

Is the deployment modeled in code and tracked in Git or similar?

Yes. Github enterprise using EKS & node groups modes. Modules & skeleton.

Are the node groups managed or self-managed?

managed

Are the standard EKS-optimized AMI’s used, or are custom AMI’s used. If custom, how are they maintained? What customizations are made to the AMI?

standard EKS optimised AMI. Experimented with bottlerocked but rolledback after figuring out it is not compliant with AWS Inspector used for vulns scanning.

Are there global (non-application-specific) Kubernetes resources? If so how are they provisioned and maintained? E.g. Daemonsets for cluster-level services like log shippers.

yes. Using argoCD and gitops approach.

How are application-specific Kubernetes resources provisioned and maintained?

argoCD

Do humans have kubectl access? If so what's the purpose of the access?

yes but only from jumphosts. Debugging and upgrading AMI version. Occasionally some quick recovery (i.e. deleting the replica set if stuck) to remediate in short term.

How and when are control plane updates applied? What is the current version of the control plane?

1.21. We control version through terraform. No updates yet upgraded since we picked up 1.21 for lowering the need in short term to upgrade.

How and when are data plane updates applied?

Updates to worke nodes are done through kubectl HOWTO upgrade AMI latest version for EKS managed node group

Which CNI is used? If not awsvpc then discuss the decision process behind this.

default eks (AWS CNI) is used

What is the subnet used for the worker nodes?

worker nodes are sitting in private subnet. LB can sit in private (majority) or public subnet and points ot worke nodes in private subnet.

Workload Operations

How are workloads deployed?

k8s resources are deployed using argocd. ArgoCD is deployed during bootstraping of the env (rarerly though our envs are staticcally crated) using manual kubectl interaction

Are Kubernetes manifests tracked in Git or similar?

Git

Is there a labeling standard used for workloads? What does it look like?

pod labels (app.kubernetes.io/name: <service name>) i.e. finexus-gateway
node labels (eks.amazonaws.com/nodegroup: <node group name>) i.e. preprod-apps-on-demand
istio-injection (allow values: disabled or enabled)
argocd labels (arocd.argoproj.io/instance=) allowed values infrastructure or apps
appmesh labels (i.e. appmesh.k8s.aws/sidecarInjectorWebhook, mesh name)

Are there any requirements to constraint pod placement? How have these requirements been handled?

We are currently introducing (PREPROD) pod anti affinity (required), same in prod, dev is left as preffered (pod anit affinity, but not required - cost optimisation). https://github.com/alrajhi-my/helm-template/commit/9872e8796b8efbc323558482da9ff4f2aa8b087f#diff-76d0ba85703cf1d5379944af329214ba061fcf64cf3d1f8b37946d35003f373e

How are workloads updated?

argocd

Is there ever a requirement to update a stateful resource such as a database as part of updating a workload? In such cases how do you handle the rolling deployment of the pods?

we use liquibase framework which is run as part of deployment and used to upgrade RDBMS (Aurora schema)

How are stateful data managed?

Liquibase currently is handling only forward changes this means if lowering the version of microservice liquibase will NOT do rollback. If any rollback would be required it would have to be handled manually. We aim to also minimize number of non-backward compatible changes if possible.

Cluster Observability

Best Practices
Run CloudWatch Agent as a Daemonset
Running the CloudWatch agent using the AWS-provided Kubernetes Daemonset eliminates the need to add it into a custom AMI image.

- References: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-setup-metrics.html
Monitor the ContainerInsights CloudWatch Metrics
- References: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-metrics-EKS.html

Is CloudWatch metrics enabled? Are you monitoring the ContainerInsights metrics?

Yes. Yes.

How are cluster logs collected?

data plane - fluentbit

Where are cluster logs stored?

cloudwatch logs. In the future cloudwatch logs + s3

Who has access to logs?

whoever has read access to k8s logs through kubectl + cloudwatch logs permissions in given account thorugh IAM access.

How are logs usually used?

debugging, monitoring, alerting

Are any analytics applied over cluster logs?

we are working currently on using metrics for analysing logs. Metrics on logs (ERRORS with treholds)

What is the retention window for logs?

mixed per env + log group. Applications log group have: DEV: 5 days, PREPROD: 7 days, PROD: 2 years. We have plans to split logs into active + archived (especially required for PROD). Analysis and planning in progress for log archival.

Workload Observability

Are any app metrics collected? How are they collected, forwarded, stored and viewed? What alerts have been defined and how are alerts routed? (AppMesh, X-ray, Istio,...)

Container insights contains fluentbit used for preprocessing and forwarding to CloudWatch Agent Daemon log which stores in CloudWatch logs. Alerts are configured on ERRORs level on logs + we are working now on process level metrics using Container Insights pod level metrics (i.e. number of restarts). X-Ray is not present (tracing) so no metrics collected from X-Ray. AppMesh related metrics (Envoy) are currently analysed to be propagated (most likely we will use latency + http errors code) but not yet defined. Alerts are currently routed thorugh SNS to emails through DL. In the future we might want to use also Slack as alternative channel and or PagerDuty IF there will be separate team for that.

Do applications use any distributed tracing framework(s)? How are traces collected, forwarded, stored and viewed?

no tracing used yet. In the future we might introduce Application ServiceLense as some future epansion

Do applications log to stdout, files, network socket, somewhere else?

STDOUT (https://github.com/alrajhi-my/microservice-libraries/blob/main/observability/src/main/resources/logback-spring.xml)

How are application logs collected, forwarded, stored and viewed?

container engine writes to local file. Fluentbit takes this and forwards to cloudwatch agent which forward later to cloudwatch for storing and viewing. Logs can be viewed through CloudWatch Logs insights or directly using kubectl logs from jumphost.

How do application owners access metrics, traces and logs?

Through AWS Web Console (Logs Insights, Alerts, Metrics), logs could also be tracked using kubectl logs from jumphost.

Security

§ Implement a strong identity foundation: Implement the principle of least privilege and enforce separation of duties with appropriate authorization for each interaction with your AWS resources. Centralize identity management, and aim to eliminate reliance on long-term static credentials.
§ Enable traceability: Monitor, alert, and audit actions and changes to your environment in real time. Integrate log and metric collection with systems to automatically investigate and take action.
§ Apply security at all layers: Apply a defense in depth approach with multiple security controls. Apply to all layers (for example, edge of network, VPC, load balancing, every instance and compute service, operating system, application, and code).
§ Automate security best practices: Automated software-based security mechanisms improve your ability to securely scale more rapidly and cost-effectively. Create secure architectures, including the implementation of controls that are defined and managed as code in version-controlled templates.
§ Protect data in transit and at rest: Classify your data into sensitivity levels and use mechanisms, such as encryption, tokenization, and access control where appropriate.
§ Keep people away from data: Use mechanisms and tools to reduce or eliminate the need for direct access or manual processing of data. This reduces the risk of mishandling or modification and human error when handling sensitive data.

§ Prepare for security events: Prepare for an incident by having incident management and investigation policy and processes that align to your organizational requirements. Run incident response simulations and use tools with automation to increase your speed for detection, investigation, and recovery.

Identity and Access Management

Are workloads given an IAM role using IAM Roles for Service Accounts or alternatively using kube2iam or kiam? If not then confirm that workloads are inheriting the node’s role.

We use IAM Roles for Service Accounts for system pods and majority of data flows (parts of data lake). Other business pods are using node level permissions. Example all tf files starting with iam in https://github.com/alrajhi-my/terraform-modules/tree/main/skeleton-vpc-apps

Is there ever a need for humans to run kubectl commands directly against clusters? How do they identify themselves (IAM user, SSO, OIDC, etc)?

There is still need to use kubectl to upgrade AMI version and do some debugging if something is not working as expected. In that case SSO users with required permissions (AdministratorAccess) are authenticating using AWS SSO to start SSM session on admin-jumphost from which they use IAM authentication to kubectl to do required operations.

Are roles defined to follow the principle of least-privilege?

Should be but further improvements possible by covering all business pods with IAM Roles for Service Accounts hence resigning from node level permissions and temporal priviledge enhancement for escalated permissions (there are plans in future to use Azure AD permissions for limited window access)

Is there any use of long-term credentials (tokens) by agents (e.g. CI workers) or users?

No long-term AWS IAM credentials used. there is long term github PAT used by github agents on shared-cluster to be able to communicate with Github Enterprise. But this is NOT using any AWS IAM credentials. For interaction between github enterprise actions and AWS infrastructure only Instance Profile is used with short living credentials (CodeArtifact, ECR uploads, AWS GLue)

Is there an RBAC policy to restrict kube-system access?

We mapped system:master to role to grant full permission access and cluster-viewer for view only

Are workloads able to access the EC2 instance metadata service?

- Using IMDSv2, hop = 2, except for apps cluster where it has problems with appmesh (envoy sidecars from appmesh are not compliant) and IMDSv2 is not enforced. Support ticket confirming this limitaion #10363768491 (account 037485385824)

Pod Security

Aside from the context application(s), what other applications are running on the cluster(s)

ThoughtMachine cluster in separate aws account - ThoughtMachine Vault, HashiCorp Vault, argocd, cert-manager, cluster-autoscaler, external-dns, ingress-nginx, metrics-server, tm-vault-installer
Applications cluster (integration layer) - MoneyThor, appmesh controller, argocd, aws-load-balancer-controller, cert-manager, cluster-autoscaler, external-dns, istio (but not used), kubernetes-external-secrets, metrics-server,reloader
Shared Cluster - actions-runner-controller (github), argocd, cert-manager, cluster-autoscaler, external-secrets,metrics-server
HC Vault - HC Vault

Aside from the context application(s), what services are running on the cluster(s). Examples could include DataDog collection agent running as a daemonset, Kubernetes Dashboard, etc.

- All cluster: cert-manager, argo-cd, cw container insight, cluster-autoscaler, external-dns, metrics-server

- App cluster: external-secret, reloader, load balancer controller, appmesh-controller
- TM: ingress nginx, istio
- HcVault: istio
- Shared-service: gh action runner controller

Are PodSecurityPolicy’s defined? What policies?

No, using default PSP created by AWS named: eks.privileged

Can privileged containers be run? Are there PodSecurityPolicies that controls when privileged containers can be run?

yes

Are containers produced by the organization designed to run under a non-root user? Is there a PodSecurityPolicy that controls when containers can be run as root?

Question 1: Yes, some microservices run under app user (UID: 1000, GID: 1000) (https://github.com/alrajhi-my/deposit-dormant-service/blob/main/Dockerfile#L5), some run unser root (apollo-server)
Question 2: No

How are images built? Are there any CI build workers that run on the cluster? Do they use Docker-in-Docker or volume-mount the socket?

- Using Docker in Docker for github action runner (Link: https://github.com/alrajhi-my/k8s-shared-service-platform/blob/main/actions-runner-controller/values.yaml#L15)

- Using Docker in Docker instead of Bind mount

Do any pods use hostPath volume mounts?

Yes, https://github.com/search?q=org%3Aalrajhi-my+hostPath&amp;type=code

Are requests and limits set for all pods?

All business pods have limits and requests defined. Most of system pods have limits and requests defined.

Is sudo, passwd or any other binaries with the SUID or SGID bit set left in container images?

- Sudo disabled (most microservices app), other like apollo is running under root already

- Passwd disabled
- SUID and GUID disabled, test by running commands: SUID: find . -perm /4000 GUID: find . -perm /2000

Are containers set to have a read-only root filesystem.

- Microservice apps: No

- ThoughMachine: Some pods defined readOnlyRootFilesystem: true for example: istio

Multi Tenancy

Is there a requirement for tenant isolation? What level of isolation is required (soft vs hard multi-tenancy)?

requirement to have tm workload separated from apps. Driven by separate aws accounts between apps (integration) and TM. Also istio enforced to exclude HC Vault cluster from TM cluster (by CIDR ranges).

Are there segregated clusters to handle hard multi-tenancy requirements? What are the clusters and what is running on each?

tm & apps & shared-cluster. All in separate aws accounts. Tm - hosting core banking system, apps - integration layer + moneythor, shared cluster - ci pipelines for github

Are namespaces being used? How are they structured?

yes. There is by default namespace per business pod.

Is self-service provisioning of new namespaces required, or will they be created by an ops team?

self-service

Is there any requirement to enforce that certain workloads do not share worker nodes?

tm and apps should not share same worker nodes and be in separat accounts. There are no requirements that we have to have split worker nodes though we were analysing separate worker nodes between business pods and system pods (i.e. coredns, metrics) for better isolation though we have not implemented that.

Are quotas defined on namespaces?

no

Detective Controls

Is cloud trail enabled?

yes

Are control plane audit logs enabled?

yes

How are audit logs for both infra and EKS API calls analyzed and monitored?

If there is any problem then they can be used in debugging.

Networks Security

Are EKS control plane endpoints set to private?

yes

What network policies if any have been defined, if any?

no

What security groups have been defined on the control plane and data plane?

There are two security groups have been defined for communication:

- Created by us, incoming traffic to API server port: https://github.com/alrajhi-my/terraform-modules/blob/main/eks-cluster/main.tf#L40 for example preprod app: sg-00456da91d40c90bf - preprod-eks-apps-cluster-sg
- EKS created security group applied to ENI that is attached to EKS Control Plane master nodes, as well as any managed workloads. for example: eks-cluster-sg-preprod-apps-908043291

Are security groups for pods in use? What rules have been defined?

no

What is the organization's policy for encryption? Is end-to-end encryption required?

yes in tm, apps, hc vault, not required in shared service cluster (CI) though all access to AWS is using TLS (kafka, other LB, aurora, s3, ecr etc.). Please note that at 2022-08-08 TM, HC all envs it is enabled, in apps dev, enabled, apps preprod, prod in-progress of introducing appmesh.

Where is TLS used and where is not used?

not required for intra-pod communication in shared-ci (no business / application data there)

o On load balancer services?

yes (all)

o Ingresses?

yes (all)

o At pods e.g. with AppMesh or another sidecar proxy?

- TM and HcVault are using istio service mesh

- Miroservices app is using appmesh (DEV, PREPROD + PROD in progress)

How are TLS certificates managed and deployed?

- LB - (AWS ALB - AWS CM, network LB - cert manager)

- Ingress - cert-manager
- pods - cert-manager (CA in secrets manager)
- pods of TM & HcVault which using istio using certificate managed by istio agent

Data Encryption and Secrets Management

What data volumes are needed by applications and are they encrypted?

All persistences (S3, Dynamodb, Aurora, kafka, EBS) are encrypted with KMS CMK

Are worker node boot volumes encrypted?

yes

Are there any secrets required by applications? E.g. connection strings, API keys, shared secrets. Are these encrypted using AWS KMS envelope encryption?

yes. all secrets are stored with AWS Secrets Manager with KMS CMK enabled.

What back-end is used for storing secrets?

aws secrets manager with KMS CMK enabled

What is the policy for rotating application secrets and how is it implemented?

no policy defined for rotation of secrets.

How are secrets provide to workloads? E.g. through environment variables or volume mounts. If environment variables are used is there any risk of leakage to log files? Has this been audited?

if application is not using direct pull using aws boto then it is being exposed using external-secrets which pulls from AWS Secrets Manager and injected into environment variables - this is for apps. TM is using HC Vault for storing secrets using volue mounts. Risk of leaking to logs is low. It was reviewed by developers in project.

Are humans allowed to SSH to nodes?

no, humans should use aws ssm sessions manager. Additionally we do not have port 22 opened in security group nor public jumphosts.

Infrastructure Security

Are humans able to directly access worker nodes? How do they do this? What is the purpose of this access?

Possible through SSM Sessions Manager if permissions enough to start session. Purpose is only for potential debugging. Not used in normal BAU.

Have the CIS benchmarks for Kubernetes been run? How often are they run?

Several CIS recommendations were applied as result of Security Hub recommendations. Security Hub is using AWS Inspector to track vulnerabilities. Real time scan through AWS Inspector & Security Hub.

What are the subnets for worker nodes? Are they public or private?

private, though LB might be created in private or public if ALB is marked as public.

Are there any automated security assessments run against the EC2 worker nodes? Has Amazon Inspector used to assess infrastructure?

AWS Inspector (v2) plus Security Hub (CIS Benchmark, AWS Foundational Security)

Are there any requirements within the organization to bake intrusion detection products into the worker nodes? Has this been done and if so how?

no requirements

Regulatory Compliance

Are there regulations that the infrastructure and applications need to comply with? Which regulations?

RMIT

How is compliance handled?

mostly AWS Security Hub

How is compliance monitored regularly?

yes

Are remediations automated?

no

Incident Response and Forensics

Is there an incident response plan for the case where a pod is compromised or exhibits suspicious behavior?

not that we are aware of (though there might be more general incident plan on bank level)

What penetration testing methodology is used, if any?

no penetration testing methodology used on k8s level. VATP (Vulnerability Assesment and Penetration Testing) used by human testing on mobile & public ingress level.

Image Security

What base images are used?

Amazon EKS optimized Amazon Linux AMI

How are container images created?

tm images are provided by 3rd party to us as imports to our ECR. Application (apps) images are being build by us and pushed into ECR by CI pipeline (github enterprise self-hosted worker nodes). Some selected images used by CI pipeline are hosted in ECR but were manually pushed there or were using ECR Pull through cache.

Are images hardened / de-fanged?

not (other than what AWS doing for them during creation of AMI)

Are container images scanned for vulnerabilities? How, where and when does this happen?

All images build and hosted on ECR have scans enabled by AWS enahnced Inspector scans + on push scans. Real time.

Reliability

Atutomatically recovery from failure

Are load balancers or ingresses used for all services? Are any services exposed to consumers through nodeports?

- Question 1: Only services need external users access to the services in EKS for example finexus-gateway. Apps microservice is using ingress load balancer, tm ingress nginx, hcvault using istio gateway service

- Question 2: No

How do you ensure that enough IP address space is available as pods are scheduled and rescheduled?

manual planning during design of the networking by AWS which built landing zone. We also have spare /21 ranges if there will be future expansion required.

Are deployments spread across AZ's? Are there any deployments that reside in a single AZ? If so, why?

yes, all node groups have 3 subnets each from differnt AZ defined.

Test recovery procedure

Do you have resiliency testing to verify that worker nodes are replaced in the case of a node failure?

BCP testing was conducted for that scenario
(https://alrajhibankmy.atlassian.net/wiki/spaces/HOME/pages/186220847/BCP+analysis+for+apps+cluster+against+lost+k8s+node)

Do you have resiliency testing to verify that stateful workloads successfully reclaim persistent volumes when re-scheduled after a failure?

- In TM account we have persistent volume and persistent volume claim, for example: storage class: general-encrypted

Scale horizontally to increase aggregate workload availability

Is the Horizontal Pod Autoscaler applied to stateless deployments?

yes

Is the Cluster Autoscaler deployed?

yes

Performance Efficiency

Design Principles

Democratize advanced technologies
Go global in minutes
Use serverless architectures
Experiment more often
Consider mechanical sympathy

Democratize advanced technologies

Can engineering teams deploy dev/test EKS clusters in a self-service manner, or do such activities need to be routed through a central team?

If there will be such need it will have to go through central team (cloud platform) and go under architectural review, planning & deployment.

Can engineering teams manage their own application deployments, or do such activities need to be routed through a central team?

app application squad can manager their app deployments by using gitops, argocd.

Experiment more often

What instance types are used? What is the basis for the choice of instance types?

TM Prod, Preprod: m5a.2xlarge (on-demand)
TM DEV: t3a.2xlarge (on-demand)

Apps Prod, Preprod: m5a.2xlarge (on-demand)
Apps DEV: t3a.2xlarge (on-demand)

Shared Services: t3a.xlarge (on-demand, spot)

HC Vault Prod, Preprod: t3a.medium (on-demand)
HC Vault DEV: t3a.medium (on-demand)

Basis of choice:

1. burstable types for either cost efficiency and variable cpu workloads
2. amd types as cheaper then intel
3. sizing of HC more or less adjusted to consumption by HC Vault
4. sizing of apps & tm instances as one of the most cost-efficient size for mixed microservices workloads and adjusted to pod sizes
5. on-demand for stability of the production workloads
6. preprod instance types same as prod to be able to have representative testing before production
7. mixed workload of shared services to guarantee always available some minimum number of resources to be able to run at least some github runners (build) even if problems with spot availabilities.

What EBS volume types are used for persistent volumes? What is the basis for the choice of volume types?

- All of the EBS volumes from PV are IO1 and GP2 for TM cluster (kubernetes storageclass)

Cost Optimisation

Design Priciples

Implement cloud financial management
Adopt a consumption model
Measure overall efficiency
Stop spending money on undifferentiated heavy lifting
Analyze and attribute expenditure

Adopt a consumption model

How are requests and limits sized for workloads?

For apps microservices we use defaults defined in helm-template which were adjusted to average java/kotlin based stateless microservice using Spring Boot. Requests on memory were also adjusted per pod so that with minimum traffic number of replicas (HPA) shold be equal to minimum. Pods in case they have higher needs can override them on microservice level.
our current defaults for apps integration parts are:

```
 limits:
    cpu: 3250m
    memory: 2Gi
  requests:
    cpu: 125m
    memory: 544Mi
```

TM is using it's own requests & limits since operating as black-box product with contracted with 3rd party SLA.
The rest of system pods are using default limits defined in helm templates coming from vendors / authors of products unless we observed we want to changes omething (I think it was mostly ArgoCD which we tuned the limits)."
our current and previous experience rather base around HPA. Our architecture is more towards stateless microservices and horizontal pod autoscaler.
we use spot on non-business-critial, non-realtime critical shared-services cluster. We experimented with SPOT and most likely will introduce that on DEV env where we do not need to satisfy SLA though it is not yet enabled.
We experimented with Fargate for our Datalake processing batch jobs driven from MWAA previously on test environment successfully. Though we migrated to normal eks managed because of doubts whether we will be able to satisfy KMS CMK requirements. We have plans to reevaluate that topic cause moving out just the batch processing for Fargate out of managed group might help us with metrics & transparency when using Container Insights. We have every night big spikes in number of nodes on eks cause we start large batch processing of S3 data and report generations. Though Fargate introduce delays in starting jobs (around 1 minute) though seems it will be possible to migrate to fargate.

Could the Vertical Pod Autoscaler help tune pod sizing?

our current and previous experience rather base around HPA. Our architecture is more towards stateless microservices and horizontal pod autoscaler.

Would it be feasible to use spot instances for your cluster(s)? Note the workload(s) need to be fault-tolerant to run successfully on spot.

we use spot on non-business-critial, non-realtime critical shared-services cluster. We experimented with SPOT and most likely will introduce that on DEV env where we do not need to satisfy SLA though it is not yet enabled.

Would it be feasible to use Fargate for your cluster(s)? Note Fargate does not support: Classic LB or NLB; Daemonsets; privileged containers; HostPort or HostNetwor in pod specs; GPU;

We experimented with Fargate for our Datalake processing batch jobs driven from MWAA previously on test environment successfully. Though we migrated to normal eks managed because of doubts whether we will be able to satisfy KMS CMK requirements. We have plans to reevaluate that topic cause moving out just the batch processing for Fargate out of managed group might help us with metrics & transparency when using Container Insights. We have every night big spikes in number of nodes on eks cause we start large batch processing of S3 data and report generations. Though Fargate introduce delays in starting jobs (around 1 minute) though seems it will be possible to migrate to fargate.

Measure overall efficiency

What level of utilization are you currently achieving for worker nodes?

All stats using 1 day period, avg from 1 day
shared-service: CPU: 17%, RAM: 14%
tm-DEV: CPU: 8%, RAM: 14%
tm-PREPROD: CPU: 7%, RAM: 14%
tm-PROD: CPU: 5%, RAM: 13%
hc-dev: CPU: 8%, RAM: 35%
apps-DEV: CPU: 29%, RAM: 64%
apps-PREPROD: CPU: 39%, RAM: 70%
apps-PROD: CPU: 7%, RAM: 55%

Are you taking any specific measures to improve node utilization?

cluster autoscaler + hpa.

Are there mechanisms in place to control costs for dev/test workloads? For example terminating clusters overnight or using Fargate launch type in dev/test.

not other then cluster autoscaler wiht HPA. We use everywhere cluster autoscaler with HPA. Introducing Fargate to majority of our workload is not possible because of Fargate limitations though it might be used in the future for batch processing. We do not terminate clusters cause we are still on development phase and can not coordinate easily testing / availability while working from 2 timezones and still have larget testing sessions so coordination of cluster unavailabiity will be complex process.

Stop spending money on undifferentiated heavy lifting

Do you use self-managed or AWS-managed node groups?

AWS-managed node groups

Do you use eksctl for deploying your cluster(s), or CloudFormation or some other IaC, or another method?

Terraform

Analyze and attribute expenditure

Is there any requirement for resource consumption by applications to be charged back to business units?

no

Are there shared clusters where applications from different business units run and which need to be separately charged back to those BU's?

no

Are there mechanisms in place to detect orphaned or unattributed infrastructure in dev/test environments to avoid cost overruns?

no, though when using shared custer wiht autoscaler the chances might be considered as relatively low.

Attachments:

architecture_platform_k8s_apps.svg (image/svg+xml) architecture_platform_k8s_apps.svg (image/svg+xml) architecture_platform_k8s_apps.png (image/png) architecture_platform_k8s_apps.png (image/png) architecture_platform_k8s_apps.png (image/png) diagrams_appmesh3.jpg (image/jpeg)