Take-Home Assignment for IO Net

Documentation Stage:
Solution document - This is an open-ended problem where there is a hypothetical situation which requires a technical solution.

Specific requirements:

Stage 1 (solution document) - Hypothetical Situation:

This is supposed to test your technical understanding and breadth of tools and system architecture designing skills in a constrained environment.

IO.net has decentralized compute suppliers. They can have following persona:

- Layman user - Gamer or Graphic designer who own GPU Laptops and PCs. They are not savvy in coding or any technical setups.
- Mining Data Center - web3 coin mining sites who own entire data centers of GPUs. They have technical capability to understand setups but need it to be scaled across their entire datacenter

In this condition we want the suppliers to run some software on their devices and that should initiate their devices to be used by io.net platform while they're being compensated for the jobs their devices were used for. IO.net is supposed to manage a Dynamic resource pool where computers join and leave the platform as they like.

Context: https://www.hashicorp.com/c2m
Use the above material as a reference where we would need to achieve scalability of 2 million devices (not workloads but 2M physical devices) that are decentralized all over the world and provided by the suppliers mentioned above. Answer the following questions with keeping the context in mind.

Design Contraints

1. Hashicorp BSL Changes
2. Hashicorp Consul not used for nomad server discovery.
3. Hashicorp Nomad is using token based authentication.
4. IO.net will maintain a management layer
    a. To maintain catalog of registered clients.
    b. To distribute configurations/certificates/tokens/etc.
    c. To allow clients/user to move from one region to another (Example: Region Failure)
5. One control plane of Nomad Server can handle 10k clients
    a. As of now, no public proof is mentioned. I am only able to find one blog post from Traefik which talks about number of workers.
6. Out-of-the-box features like multi-region / DAS in Nomad requires Enterprise license
7.

3rd Party Components

1. Docker Runtime is required in some cases.
2. Load Balancers like F5/Nginx+/Traefik Enterprise/etc.

Questions to be answered on the solution document:

- Nomad Integration: Detail the steps and components required to integrate a VM or computer with Nomad, enabling it to join the IO.net platform as a compute resource.

Answer:
Nomad Client Integration should be facilitated by Shell Script/PowerShell/etc.
Steps:

1. Download Script from io.net website
2. Run Script as Administrator
3. The Script should do the following
    a. Generate or Fetch Unquie Name from io.net sever
    b. Detect and Confirm, CPU/Memory and GPU details
    c. Download Required Drivers from Hashicorp Nomad or io.net Mirror Website
        i. Nivida GPU Driver
        ii. Runtime packages like java/python/etc.
    d. Detect or Ask Location (required to join right nomad server pool)
    e. Download Nomad Public Address, TLS Certificates, Tokens, Nomad Client Packages
        i. These values are region specific so io.net should maintain right structure
        ii. io.net server should also generate right token from nomad for this client upon request.
    f. Generate Nomad Client with right set of labels and resources
    g. Create/Start Nomad Client Service
    h. Generate Readme/HowTos for Stop/Restart/etc
    i. Clean up of temporary files if required

- What Nomad-specific configurations are necessary for different types of compute suppliers?

Answer:
Nomad Client can be configured with various options

1. CPU
    a. Nomad Client Auto Detects CPU capabilities
2. NUMA
    a. Nomad Client Auto Detects this capabilities (Enterprise License)
3. GPU
    a. Nivida is officially supported and requires docker as 3rd party component

    b. Ref Configurations: https://developer.hashicorp.com/nomad/plugins/devices/nvidia
4. Custom
    a. Incase of mismatch in resource definitions client.meta should be used
        i. Provide reasonable options like standard name of GPU instead of user input
        ii. etc.

- Describe the process of initiating a node into IO.net using Nomad. Would this be a self-registering agent, a script executed by the user, or another method? Please provide a detailed flow.

Answer:
As of now, I would prefer to have installation script for users where nomad will self register but all the AuthZ/AuthN details will be downloaded/generated by installation script. The flow is mentioned in the first question.

- Cross-OS Compatibility: Discuss how your Nomad-based solution accommodates various operating systems (Windows, Linux, macOS, including arm64 devices) and ensures compatibility across these diverse environments.

Answer:
As of now, nomad supports a lot of combination of Architectures like X86/arm86 with various Operating systems like windows/Linux/MacOS. Ref: https://developer.hashicorp.com/nomad/install

- Connectivity and Latency Management: Given the global distribution of devices, how does your Nomad solution address varying connectivity strengths and internet speeds?

Answer:

In my solution, there are two pieces
1. Nomad Control Plane
    a. This is collection of federated regional nomad servers
2. Nomad Data Planes
    a. This is collection of clients registered under a single regional nomad servers

Since,
1. Nomad Servers requires less 10ms latency between so they are regional.
2. Nomad Client doesn't require lower latency or bandwidth.
    a. Reduce the number of heartbeats (Ref: https://developer.hashicorp.com/nomad/docs/configuration/server#client-heartbeats)
    b. Client can also have custom meta keys describing there network speed so that it can be used later

- Remote Management & Dynamic Resource Pools:

  How would you leverage Nomad to remotely manage the state of machines and dynamically adjust resource pools as devices join and leave the platform? Discuss the advantages and challenges of your approach, particularly focusing on security and the absence of direct SSH access.

Answer:

In my solution, I have added a management layer which monitors resource usage across fleet/region and tweak nomad jobs to either increase or decrease resource usage based on the following factors

1. User preference for inclusion or exclusion for compute.
2. IO.net want to
    a. Utilize maximum compute available.
    b. Use resource from region where compute is cheaper

Since, nomad clients receive all there work from servers so it much easier, we just need to tweaker nomad jobs and clients will receive updates via nomad.

- Orchestration Automation with Nomad:

  Elaborate on automating the orchestration of Ray clusters in this decentralized environment, specifically using Nomad without relying on SSH for remote management. How would you configure Nomad jobs for head and worker nodes differently to optimize resource utilization and cluster performance?

Answer:

In my solution, I have mentioned about IO.net managed data plane. The main purpose of this data plane is to host

1. Ray Head Nodes
2. API Gateways
3. Load Balancers
4. Etc.

Since, Ray doesn't support Multi Head Nodes, it will be significant challenge to expand ray cluster beyond some limits which means, we have to develop another capability in IO.net management plane to dynamically create Ray Clusters using Nomad based on following

1. workload requires specific compute resources
2. Workload requires certain geo restrictions
3. Etc.

Ray Nodes (Head/Worker) needs to managed via docker container since it provides easy of dependency management and support GPUs. Nomad Jobs needs to be tweaked like number of worker nodes, and Head node resources to accommodate more workers. Additionally, Ray requires separate TLS certificates for each Ray Cluster managed via IO.net management layer for authentication.

Another Feature from Nomad can be used but it is locked behind enterprise license. https://developer.hashicorp.com/nomad/tutorials/autoscaler/dynamic-application-sizing-concepts

- Security Measures: What specific security measures and best practices would you implement within your Nomad setup to ensure secure management of remote machines and data protection?

Answer:

In my solution, all communication are encrypted via TLS

1. Nomad TLS (Ref: https://developer.hashicorp.com/nomad/tutorials/transport-security)
2. Ray TLS (Ref: https://docs.ray.io/en/latest/cluster/kubernetes/user-guides/tls.html)

Next, I will implement Nomad ACL + Token to limit, who can run what job and where.

- Scalability and Financial Feasibility: Evaluate the scalability of your Nomad-based solution in terms of handling up to 1 million devices. Discuss any financial implications if third-party solutions are involved in your design.

Answer:

In my solution, I have included several layers of infrastructure to allow scaling at various levels for Nomad Servers.

1. Number of Regions (Range 100-1000s)
2. Number of HA Load Balancer / Reverse Proxy (Serving Nomad APIs) (Range 100-1000s)
3. Number of Nomad Servers (Range 3-5)
4. Vertical Scaling of Nomad Servers like CPU / MEM / Network / Disk

One of the financial implications in this case, If I would have to use 3rd party LB like F5/Nginx+/Treafik Enterprise will increase the cost dramatically since my calculations are like this

100 (Regions) * 1 (LB) * 10000 (Clients) = 1 Milion Devices

Incase of Ray Cluster, I haven't found publicly available stats like maximum number of worker can be attached to a single ray head so if I consider conservative estimate like 1000 workers. That means, I would have to spread at least 1000 Ray head nodes across 1000 Regions to reach 1 million devices for ray compute capacity. Since, I have included IO.net managed data planes in each respective region, it can be easily spawned with nomad ondemand and teardown if no clients/work. This will require IO.net platform to monitor demand of ray work and spawn new head node and respective worker node in a given region with required set of constraints like GPU=A100 or CPU=8 Cores.

- API Gateways and External Endpoints: Explain how external services, such as API gateways or web interfaces, would interact with the Nomad server. Include considerations for authentication, authorization, and secure communication.

Answer:

In my solution, all above mentioned components will do traffic pass through only since nomad is using TLS + Token in authentication process so i can't terminate SSL/TLS on API Gateways/ Endpoints / etc.

- Scaling to a Million Devices:
  Nomad Cluster Configuration: Detail the configuration of Nomad servers and clients, including considerations for server quorum, federation, and scalability. How would you structure the Nomad clusters regionally to manage a global pool of devices effectively? How would you ensure that the Nomad cluster remains operational and efficient, even as the number of managed devices scales up significantly?

Answer:

Below shared architecture diagram explains most of stuff.

I want to have nomad servers spread across multiple datacenters/AZ within a region then multiple regions with federate with each other to reach million devices capacity. Nomad server in production recommends to have 3 to 5 servers so that incase of VM/Datacenter failure nomad server region will continue to operate normally.
To keep nomad cluster efficient,

- IO.net managed data plane components for a given region can scaled up/down on demand.
- Nomad Control Plane can also be downsize with resources given no workload.
- To be more network efficient, heartbeat settings from nomad worker to server will reduced so that less network bandwidth is used by default for management purposes.

- High Availability and Disaster Recovery: Discuss your approach to ensuring high availability of the Nomad cluster and strategies for disaster recovery. How would you prevent downtime and data loss, especially in a highly distributed environment?

Answer:

High Availability is achieved using

- Active-Active Load Balancer (Ref: https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0102823)
- multiple nomad servers across datacenters within a region.

Disaster Recovery is considered for following scenarios

- Handle Nomad Server Crash upto 2
- Handle Nomad Server Datacenter upto 1-2

- Handle Nomad Server Region is a complex thing
    - Create and store Nomad Server Offsite backup
    - A capability should be developed at some point in IO.net management layer

- Monitoring and Maintenance: Describe the monitoring and logging infrastructure needed to support this scale. How would you ensure the health of the Nomad servers and the vast array of client devices?

Answer:
I will split the nomad server and client monitoring/logging into separate buckets to explains.

Assumptions
- In house infrastructure to be used wherever possible.
- Open Source and Scaleable Solution.

Nomad Servers - Monitoring / Logging
Metrics should be sent to victoria metrics solution and Logs to be shipped to victorialogs that should be managed in separate DC in IO.net managed Data Plane under all regions.It can also be deployed via Nomad servers but I would recommend to decouple this component so that nomad server failure doesn't cause Metrics and logging solution to fail as well.

Nomad Clients managed in IO.net management data plane - Monitoring / Logging
Metrics should be sent to victoria metrics solution and Logs to be shipped to victorialogs that should be managed in separate DC in IO.net managed Data Plane under all regions.It can also be deployed via Nomad servers but I would recommend to decouple this component so that nomad server failure doesn't cause Metrics and logging solution to fail as well.

Nomad Clients installed on end user - Monitoring / Logging
In my opinion, IO.net shouldn't collect these metrics and logs since they are too massive and provide little to no benefit in my understanding. Despite not actively collecting metrics/logs from end user, still IO.net will have several options to have these ondemand if required.
- Nomad Clients reports some metrics to Nomad servers for scheduling purpose.
- Nomad Server UI also provides capability to see logs from nomad clients over network but it is not recommended since, it affects nomad internal and servers to process these for operator.

- Explain the process of node registration and job scheduling, emphasizing Nomad's role in ensuring efficient resource allocation and task distribution.

Answer:
Node Registration Process is already explained above.
Job Scheduling Process
1. Nomad User needs to create jobspec with contraints/type/groups/tasks.
2. Job Spec needs to be submitted to Nomad Server using CLI/API with right credentials.
    a. Job can be submitted in two modes aka Plan or Run.

3. Nomad Server validates the submitted jobspec
4. Nomad Server usages its configured scheduling algorithm like bin-packing or spread or custom to decide where this job should be run.
5. Nomad Clients start executing assigned group/task on it.

Ref: https://developer.hashicorp.com/nomad/docs/concepts/scheduling/scheduling

To Utilize resources efficiently

1. Nomad allows using custom scheduling algorithms.
2. Nomad Job spec with parameters like spread
3. Nomad also allows Dynamic Autoscaling (Enterperise License)

- Job Specifications (Jobspecs):

  Provide examples of job specifications (jobspecs) for both compute suppliers (e.g., a layman user with a GPU laptop and a mining data center). How do these specs ensure the correct deployment and operation of workloads on diverse hardware?

Answer:

In this case, I will be using two features from nomad jobspec called constraints and affinity rule.
With constraints, I will lock, how many total resources are available for this user and then with affinity, I will spread the task/load into several machines/datacenters if possible.
Note: sample job specs are uploaded to github for reference.

- Discuss how jobspecs are used to define task groups, services, and tasks within your solution, including any Nomad features (e.g., constraints, affinities) used to optimize placement and performance.

Answer:

Duplicate question to above one.

- Networking and Service Discovery:

  Detail how Nomad's networking and service discovery features would be leveraged to manage connectivity among the decentralized resources, especially considering the dynamic nature of devices joining and leaving the platform. Explain the integration of Consul for service discovery and how it supports the orchestration of Ray clusters within the IO.net platform.

Answer:

Nomad provides two types of service discovery

1. Nomad Native
2. Consul (hashicorp.com/nomad/docs/integrations/consul)

Ref:hashicorp.com/nomad/docs/networking/service-discovery

Integration of Consul for Ray Cluster

1. Ray workers can easily discover Ray Head nodes.
2. Most of the LB/R-Proxy/API-Gateways supports consul so it can easy to expose endpoint for end users.
3. Consul KV is easy storage for all configurations for Ray clusters.
    a. Dynamic updates to configurations can be propagated to all workloads.

Notes

1. Integration of Consul brings extra cost to project
    a. Enterprise license
    b. Consul Server Resources
        i. Additional network bandwidth requirements for the end users.
        ii. Additional CPU/Mem resources from end users.
    c. Installation of Consul Clients across fleet
    d. etc.

- Security and ACLs (Access Control Lists):

  Describe the security mechanisms, including ACLs, that you would implement within Nomad to secure access to the cluster and manage permissions across different types of compute suppliers.How do these security measures protect against unauthorized access and ensure the integrity and confidentiality of tasks executed on the platform?

Answer:

Nomad Server Access can be maintained by various methods

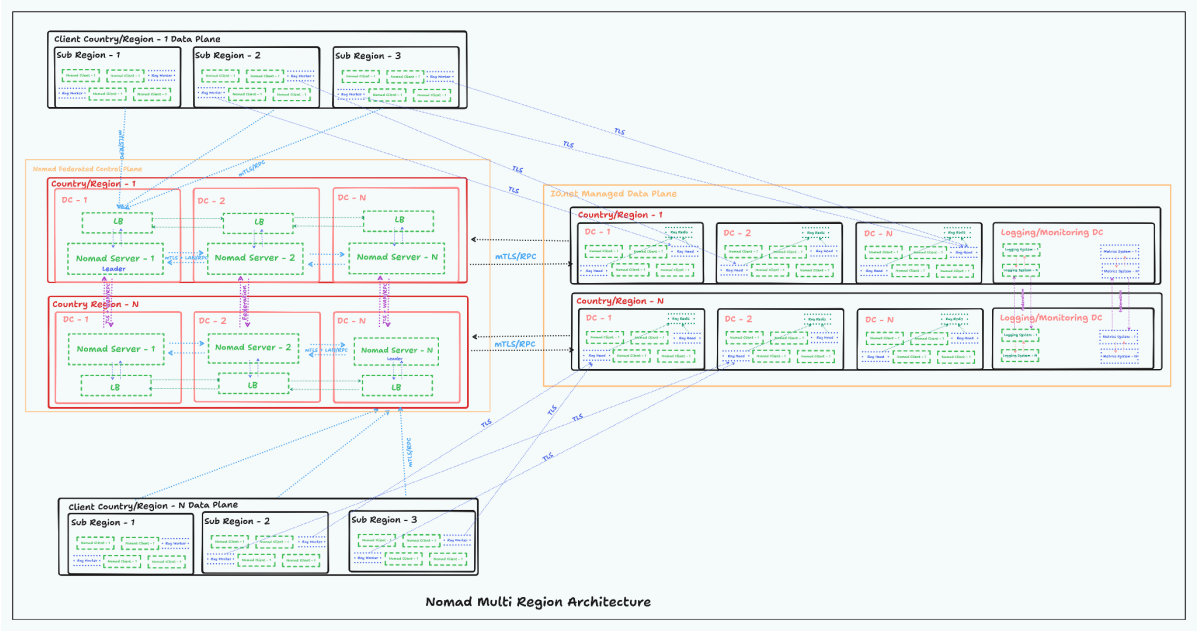
1. Nomad ACL + Tokens
2. Nomad ACL + OIDC (enterprise)

Nomad Jobs Access can also be maintained in similar way

1. ACL + Tokens
2. New approach: https://developer.hashicorp.com/nomad/docs/concepts/workload-identity

- **Proposed Architecture Diagram [important]:** Provide a comprehensive architecture diagram illustrating your proposed Nomad-based solution, including all components and how they interact within the ecosystem of IO.net.

Answer:



Github: https://github.com/116davinder/jubilant-carnival/tree/main

NOTE[important]: Please make sure to be as detailed as possible , use any illustrations or sources you need and explain each solution component comprehensively. This document will decide if we would move forward with your application. Feel free to email any questions to the interviewer incase the assignment is not clear.

Expectations

Provide your solution in a GitHub repository, ensuring it meets production standards and includes thorough documentation.

Submission Guidelines

Email your Solution Document link to smiral@io.net with the subject "[IONET] Nomad Engineering Take-Home Assignment Submission [SOLUTION DOCUMENT] [NAME]"

Important Note

This assignment is designed to assess your capabilities in system building, documentation, and problem-solving. We emphasize that there are no wrong solutions in this context, only better ones, reflecting our interest in your innovative approaches and your ability to navigate challenges. If you find yourself stuck at any point, we encourage you to reach out for assistance, underscoring the importance of collaboration and effective problem-solving strategies. To ensure a balanced approach between thorough research and time efficiency, we recommend spending no more than 2-3 hours on research and development for the solution document and 1 hour on python assignment.
Good luck, and we anticipate your innovative solutions and thought processes.