Sandia National Laboratories

Kubernetes for HPC Administration

Samuel Knight (sknigh@sandia.gov)
Sandia National Laboratories

Sunday, 11/14/2021
SIGHPC Systems Professionals Workshop (HPCSYSPROS21)
Supercomputing 2021
St. Louis, MO

U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2021-14435 C

Outline

- Introduction
- What is Kubernetes
- Provisioning
- Deployed Services
  - SSH Reverse Proxy
  - Slurm
  - Jupyterlab
  - Telemetry (Logs)
  - Telemetry (Metrics)
  - IP propagation and DNS
- Tools
- Conclusion

Introduction

- HPC administration is challenging
  - Scripts
  - Cron Jobs
  - Systemd services, Authentication, multiple nodes…
- Simple bare-metal servers introduce single points of failure
- Software stack compatibility varies by OS distribution and what other software is installed

What is Kubernetes

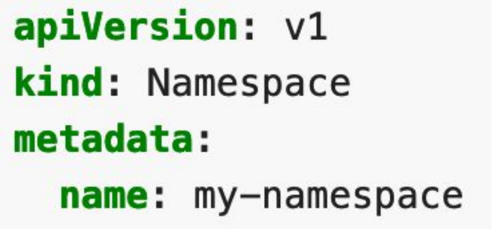
- Mature Google Project
- Container Orchestration Platform
- Deployment Lifecycle Mechanisms
  - Horizontal Scaling
  - Volume Provisioning/Mounting
  - Security Policies
  - Network Routing
  - DNS
  - Unified HTTP routing

```
$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
Node1 Ready master 1h v1.21
Node2 Ready <none> 1h v1.21
Node3 Ready <none> 1h v1.21
```

What is Kubernetes

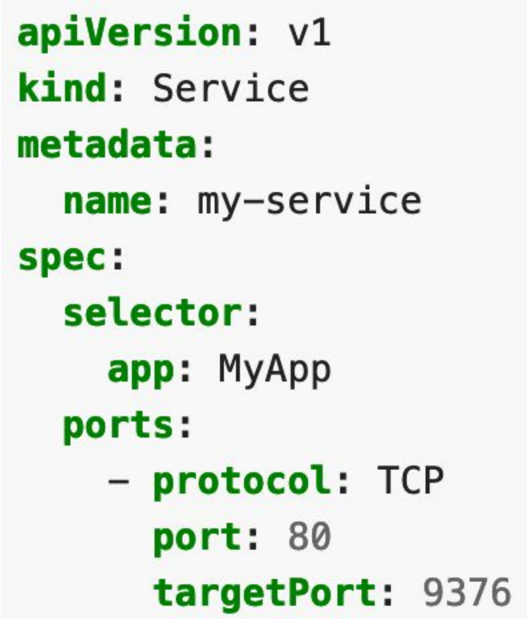
- Managed through standardized interfaces
  - Restful API server
  - Components defined with **YAML stubs**

namespace.yaml



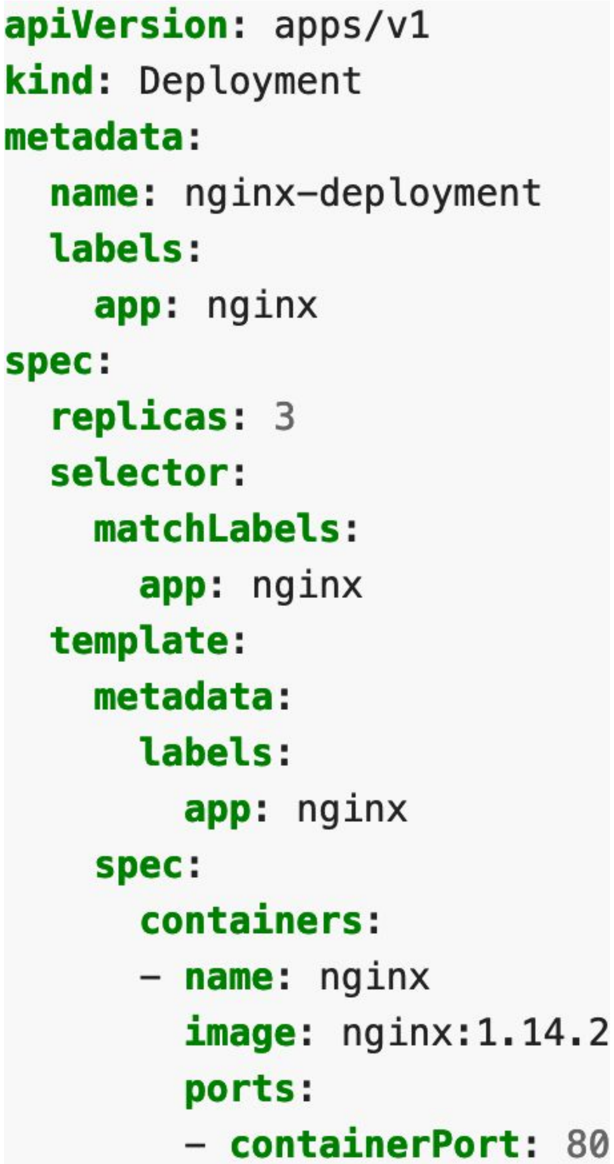
```
apiVersion: v1
kind: Namespace
metadata:
  name: my-namespace
```

service.yaml



```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

deployment.yaml



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

Provisioning

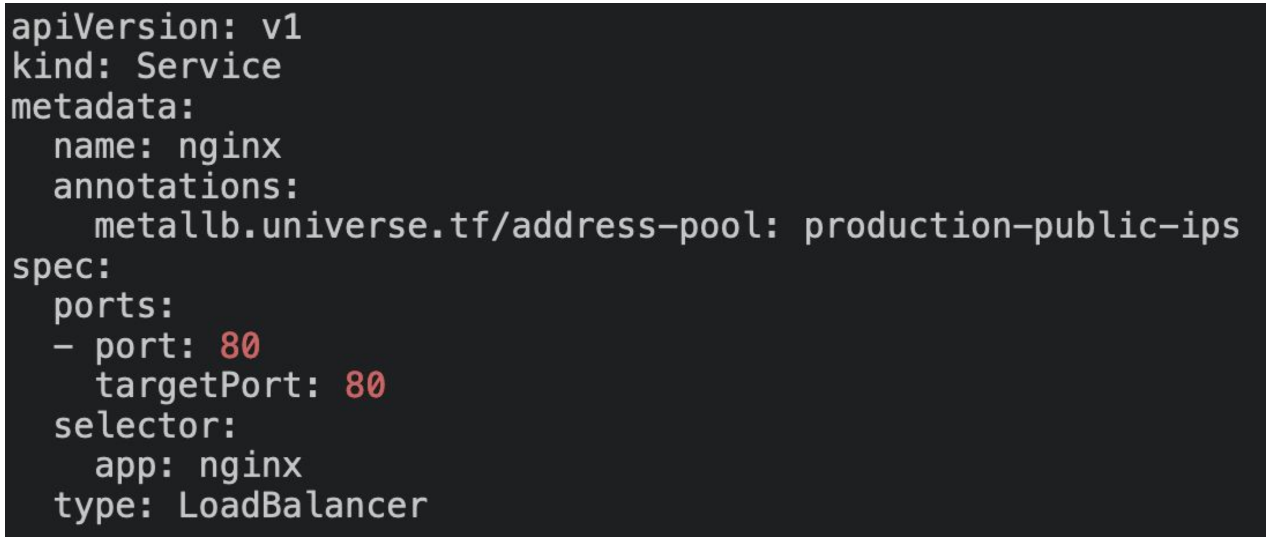
- Multitudes of Kubernetes Implementations and Provisioning Methods
  - OpenShift (RedHat)
  - MicroK8s (Ubuntu)
  - Docker Desktop (Shipped with Docker GUI on Mac and Windows)
  - K3s (Rancher Labs)
  - Kubeadm (First Party)
  - Minikube
  - **Kubespray**
    - Ansible-based
    - Provision multiple nodes
    - HA-capable without External Loadbalancer
- Filesystem
  - Backing Ceph RBD with RBD provisioner
  - *Ad hoc* NFS and Cephfs mounts

Deployed Services

MetalLB

- Uses Service annotations to map IP addresses with Services
- Promulgates IP routes to Kubernetes nodes with ARP
- Links
  https://metallb.universe.tf/
  https://github.com/metallb/metallb

service.yaml

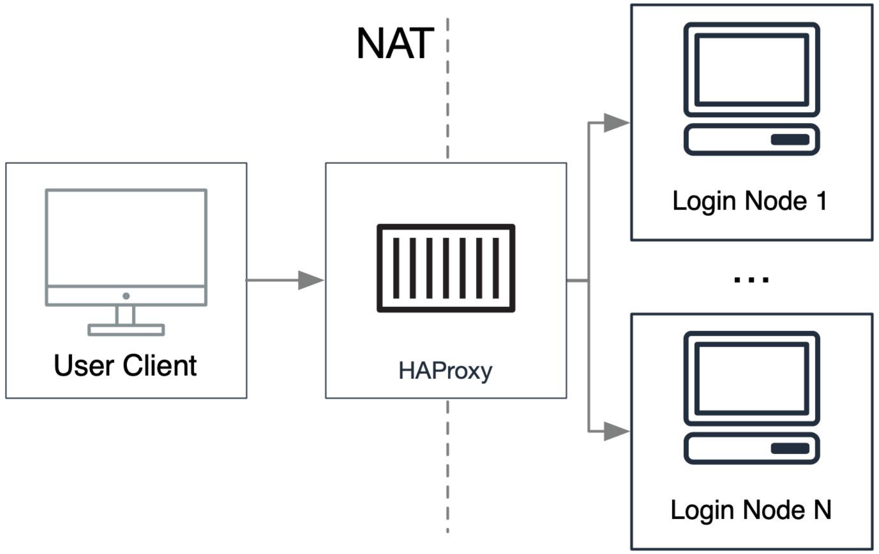


```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    metallb.universe.tf/address-pool: production-public-ips
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

Deployed Services

SSH Reverse Proxy – Load balance across multiple possible login nodes from a single host

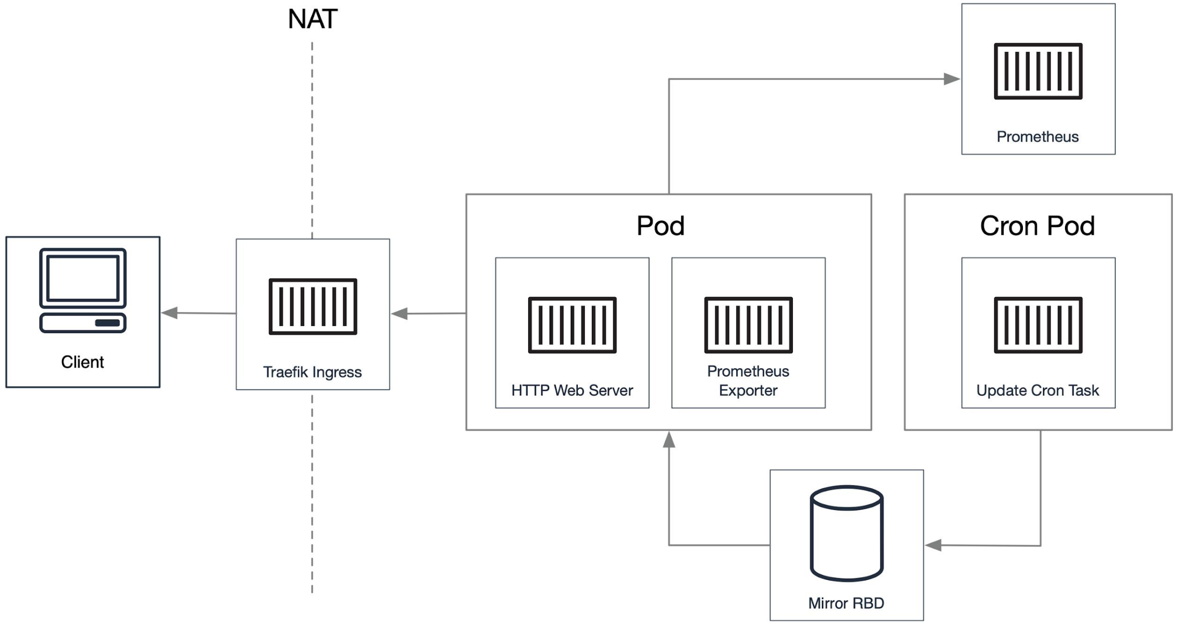
- Client initiates SSH connection on port 22
- Kubernetes routes to internal HAProxy container
- HAProxy forwards SSH to a single backend login node on port 22
  - Picks node in round-robin to balance load
  - Automatically removes unresponsive nodes from the pool



Deployed Services

Static Web Pages- Present a file over HTTP

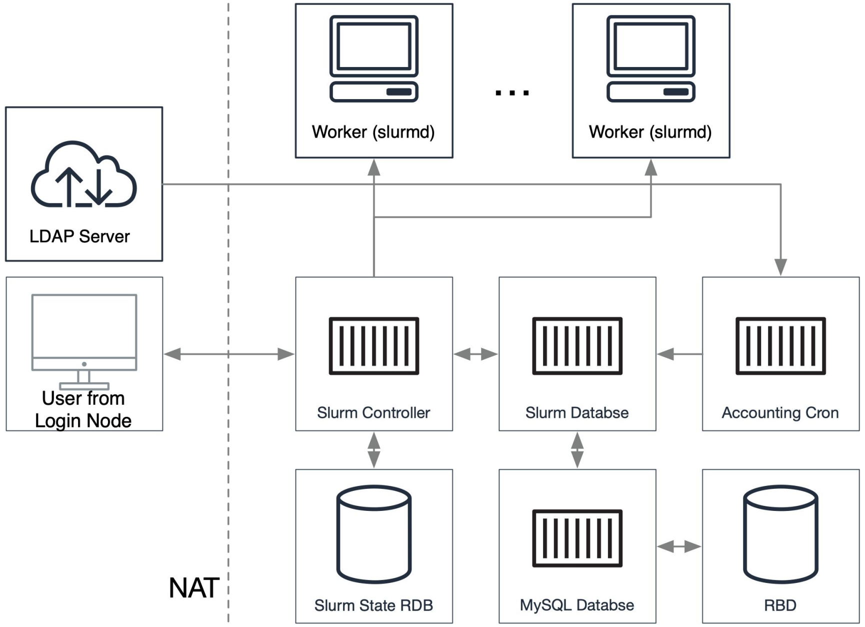
- Built on Nginx Container
- Binds to Backing volume, i.e. NFS mount or a dynamically provisioned RBD volume
- Optionally include a Prometheus exporter



Deployed Services

Slurm – Tool for batch scheduling workloads on HPC

- Slurmctld (controller) and Slurmdbd (database) reside in pods
- Slurmdbd uses SQL backend
- Slurmctld and SQL backend require volumes
- Accounting Cron script communicates with external LDAP service to update accounting information
- Specific Slurmctld and Slurmdb ports are exposed to worker nodes (slurmd services)

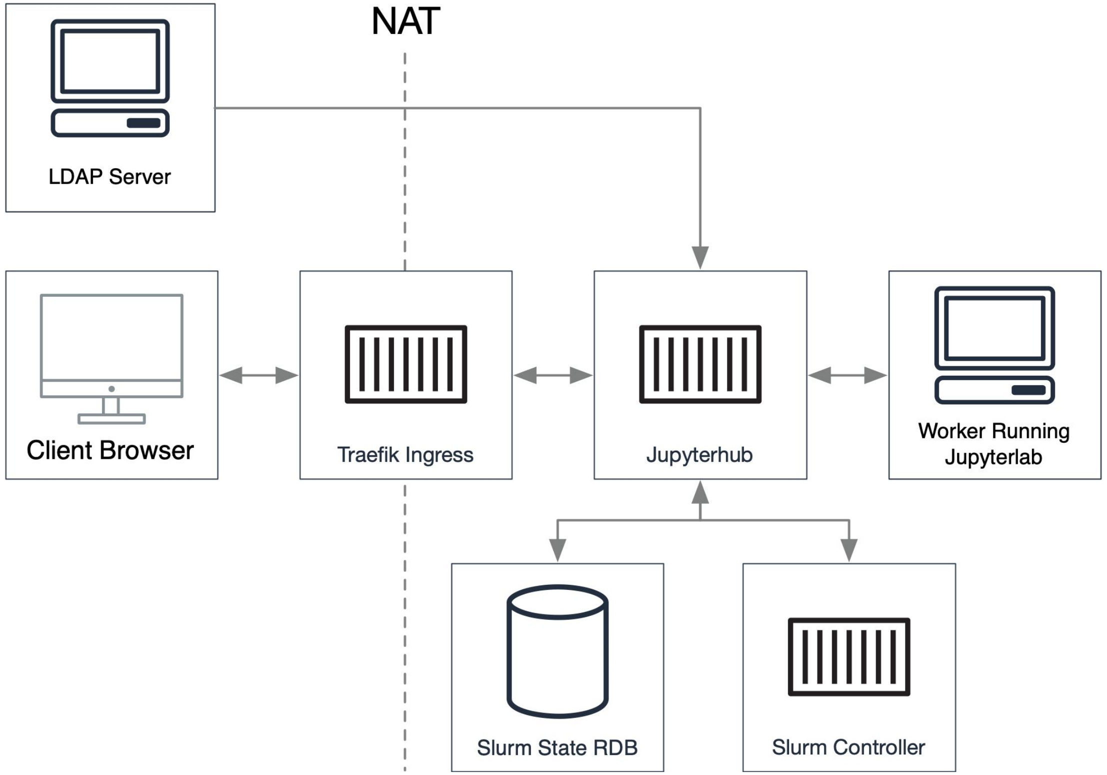


Deployed Services

Jupyterlab [1] – Web-based notebook

Jupyterhub [2] – Web-based multi-user frontend that spawns Jupyterlab instances

- HTTP page forwarded through reverse proxy
- Authentication page uses LDAP backend
- Communicates with Slurm controller to spawn Jupyterlab instances through Slurm using batchspawner [3]
- Jobs can be launched in different queues with wrapspawner [4]



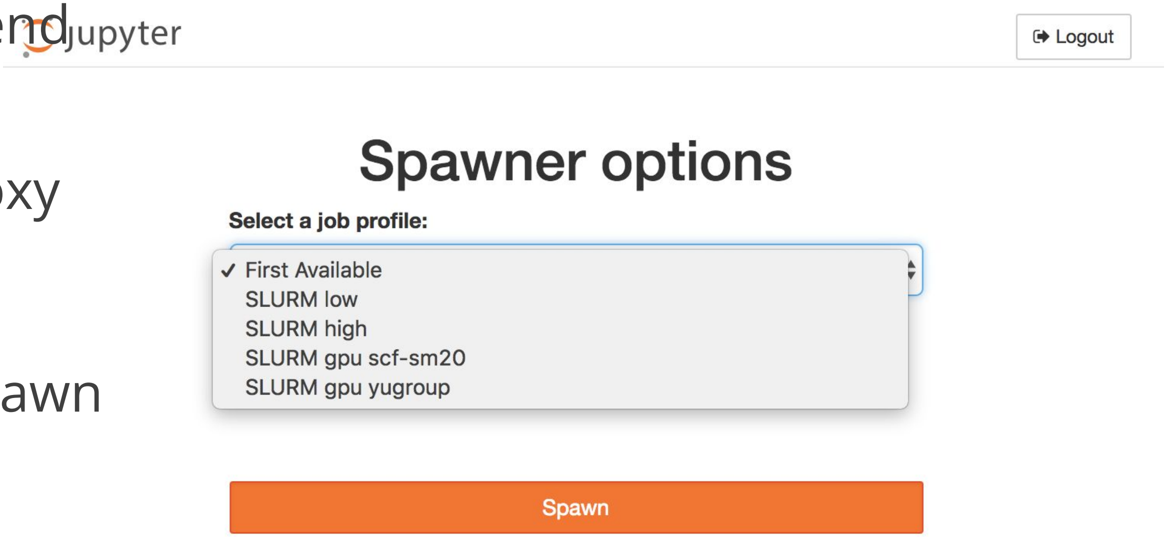
[1] https://jupyter.org/
[2] https://jupyter.org/hub
[3] https://github.com/jupyterhub/batchspawner
[4] https://github.com/jupyterhub/wrapspawner

Deployed Services

Jupyterlab [1] – Web-based notebook

Jupyterhub [2] – Web-based multi-user frontend that spawns Jupyterlab instances

- HTTP page forwarded through reverse proxy
- Authentication page uses LDAP backend
- Communicates with Slurm controller to spawn Jupyterlab instances through Slurm using batchspawner [3]
- Jobs can be launched in different queues with wrapspawner [4]



Spawner options

Select a job profile:

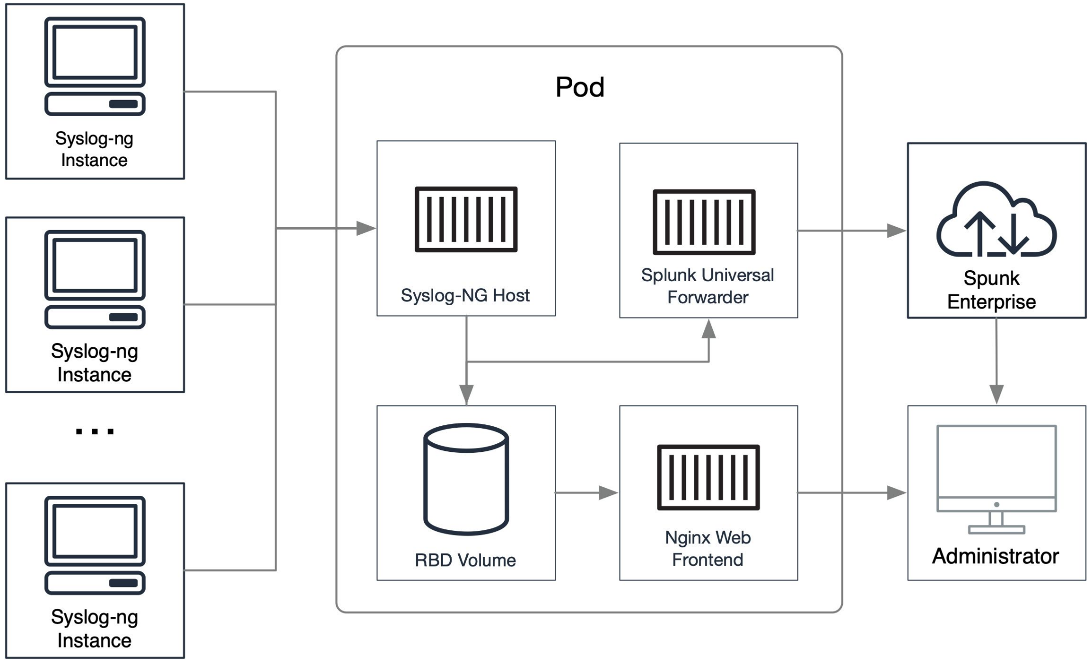
✓ First Available
SLURM low
SLURM high
SLURM gpu scf-sm20
SLURM gpu yugroup

[1] https://jupyter.org/
[2] https://jupyter.org/hub
[3] https://github.com/jupyterhub/batchspawner
[4] https://github.com/jupyterhub/wrapspawner

Deployed Services

Telemetry (Logs)

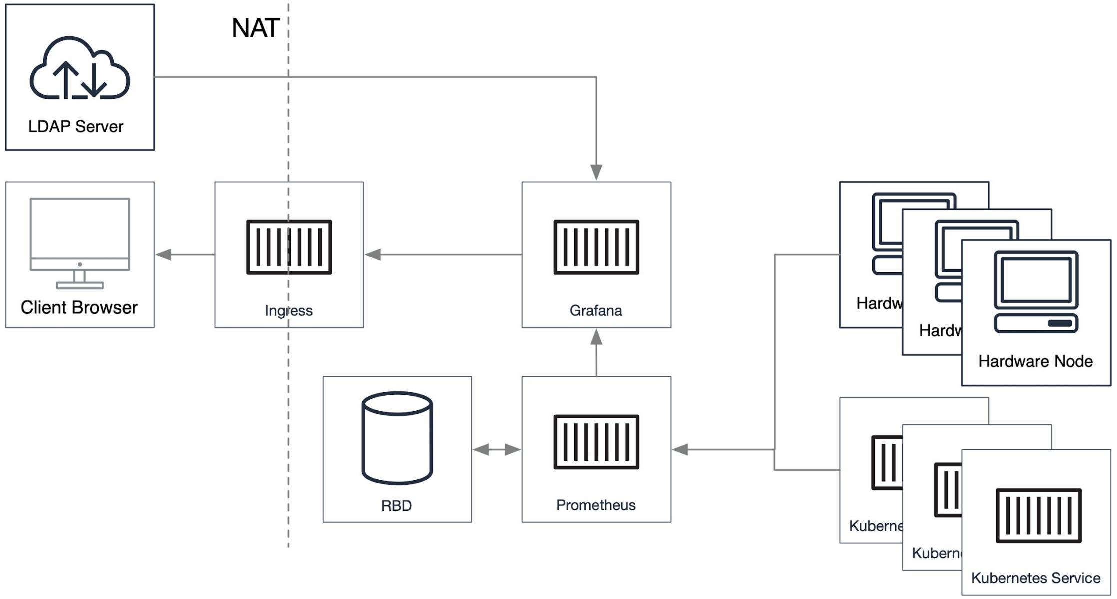
- Nodes push syslog output to aggregator pod
- Logs are written to an RBD volume
- Splunk Universal Forwarder watches RBD volume and pushes changes to a corporate Splunk instance



Deployed Services

Telemetry (Metrics)

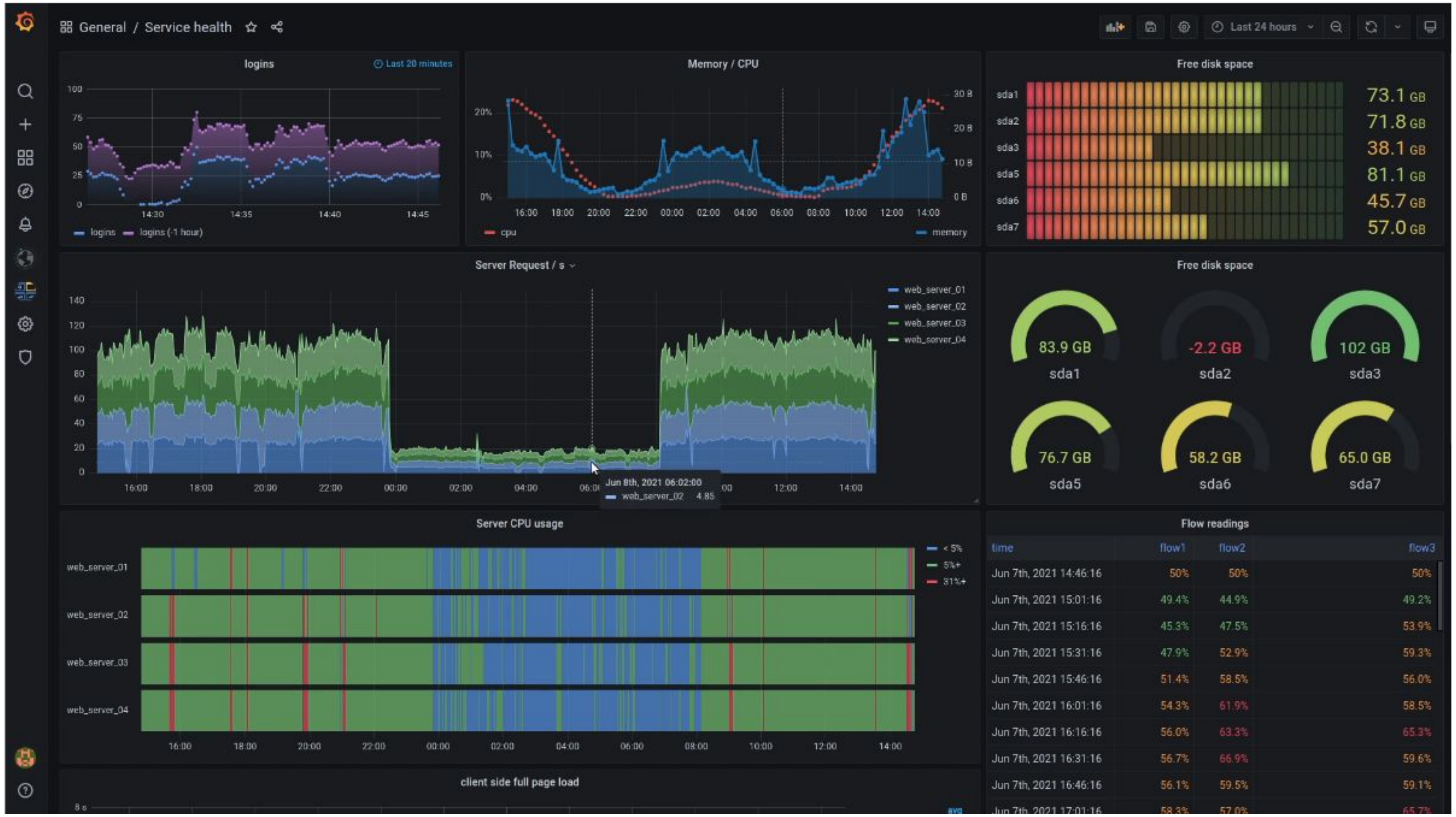
- Prometheus [1] is a time-series database
- Periodically scrapes targets
  - Automatically scrapes internal Kubernetes services
  - Can be configured to scrape nodes running node-exporter [2]
- Grafana is a *de facto* frontend for rendering dashboards



[1] https://github.com/prometheus/prometheus
[2] https://github.com/prometheus/node_exporter

Deployed Services

Telemetry (Metrics) – Grafana Dashboard



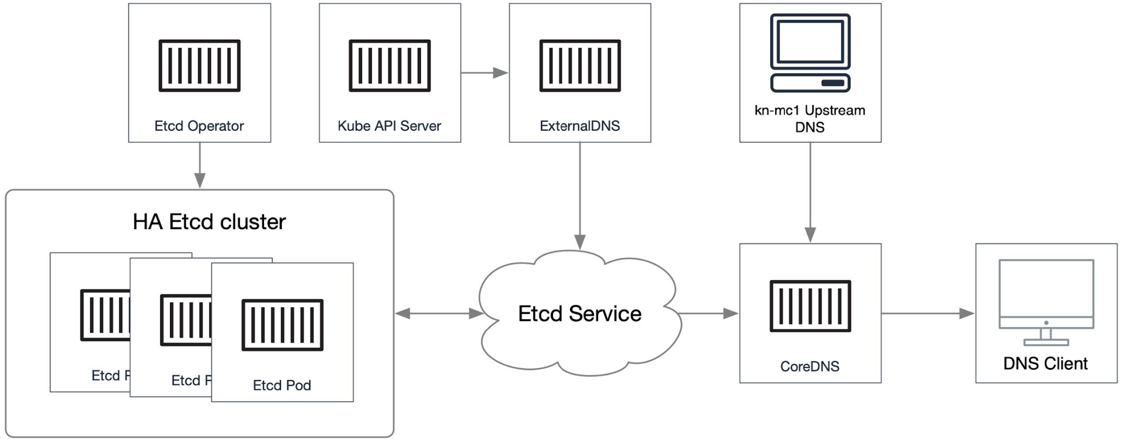
Flow readings

time	flow1	flow2	flow3
Jun 7th, 2021 14:46:16	50%	50%	50%
Jun 7th, 2021 15:01:16	49.4%	44.9%	49.2%
Jun 7th, 2021 15:16:16	45.3%	47.5%	53.9%
Jun 7th, 2021 15:31:16	47.9%	52.9%	59.3%
Jun 7th, 2021 15:46:16	51.4%	58.5%	56.0%
Jun 7th, 2021 16:01:16	54.3%	61.9%	58.5%
Jun 7th, 2021 16:16:16	56.0%	63.3%	65.3%
Jun 7th, 2021 16:31:16	56.7%	66.9%	59.6%
Jun 7th, 2021 16:46:16	56.1%	59.5%	59.1%
Jun 7th, 2021 17:01:16	58.3%	57.0%	65.7%

Deployed Services

ExternalDNS [1]

- Detects service annotations to map MetalLB IP addresses hosts and pushes it to a DNS
- Useful for providing hostnames to external nodes
- Improves high availability



[1] https://github.com/kubernetes-sigs/external-dns

Deployed Services

ExternalDNS [1]

- Detects service annotations to map MetalLB IP addresses hosts and pushes it to a DNS
- Useful for providing hostnames to external nodes
- Improves high availability

```
kind: Service
apiVersion: v1
metadata:
  name: jupyterhub-api-service
  annotations:
    external-dns.alpha.kubernetes.io/hostname: jupyterhub-host.k8s
    metallb.universe.tf/address-pool: internal
spec:
  type: LoadBalancer
  ports:
    - name: jupyterhub-api
      protocol: TCP
      port: 8081
```

[1] https://github.com/kubernetes-sigs/external-dns

Tools

- Kubectl
  - Primary method for interacting with Kubernetes API server
  - First party tool
- Kustomize [1]
  - Template-free tool that layers 'scoped' into kubectl
  - Includes syntactic sugar, e.g. assigning labels to a group of YAMLs globally setting namespace, etc.
- Helm
  - Template-based tool for installing "Packaged" deployments
  - Helmfile [2] - Secondary project for combining multiple helm packages into one YAML file
- SOPS [3] – Encypt YAML files with secrets using GPG

[1] https://kustomize.io/
[2] https://github.com/roboll/helmfile
[3] https://github.com/mozilla/sops

Discussion/Conclusion

Advantages

- Standardized interface for interacting with resources
- High availability
- Load Balancing
- Encapsulated software life-cycle
- Possible to version control most of the infrastructure
- Large and increasingly mature ecosystem

Disadvantages

- Kubernetes is complex and requires dedicated developers
- Slight application misalignment