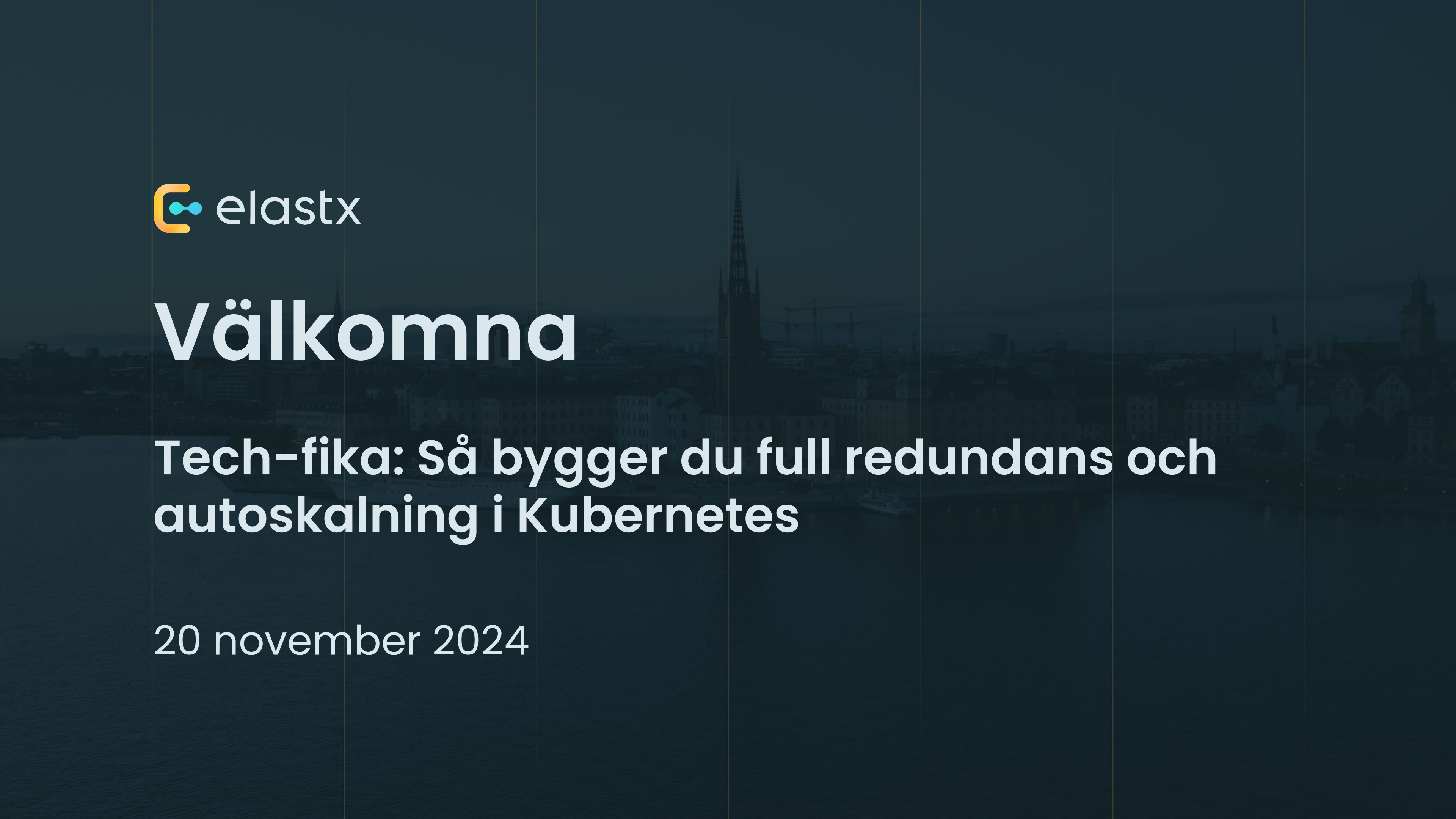
elastx

The Swedish Cloud Provider



elastx

Välkomna

Tech-fika: Så bygger du full redundans och autoskalning i Kubernetes

20 november 2024

Redundancy

- What is redundancy?
    - Something **no one** wants to pay for upfront ever....
    - Something **everyone** wants when the service goes down at 1 a.m. on a Sunday morning.
    - Insurance for **all**.
- Redundancy in, for example, databases means that the same data is stored in multiple locations to increase availability and security.
- Used to protect against data loss, system failures, and ensure continuous operation.

elastx

Why is Redundancy Good?

- Increased availability:
  - If one server crashes, the system can continue to function using redundant copies.
- Data recovery:
  - Data recovery after a failure or attack reduces the risk of data loss.
- Fault Tolerance:
  - Protects against hardware failures and human errors by having multiple backups.

elastx

What are the Disadvantages?

- Increased resource costs:
  - Redundancy means that more resources (storage, servers, networks) are required, likely increasing costs.
- System complexity:
  - Managing redundant copies can make the system more complex and harder to maintain.
- Consistency issues:
  - Risk of inconsistent data if synchronisation between redundant copies fails.

Common Redundancy Strategies

- Disk clustering
  - RAID et al: Methods for combining multiple disks for redundancy and better performance.
- Replication:
  - Automatic copying of data between servers or database systems to ensure availability and security.
- Failover Systems:
  - Automatic transitions to backup servers or systems in case of failure.

Examples of Leveraging Redundancy

- Banking Sector:
  - Critical systems that require high availability use redundancy to ensure services are always accessible.
- Cloud Services (Cloud Computing):
  - Data centres distribute data across multiple servers and geographical locations for high reliability.

elastx

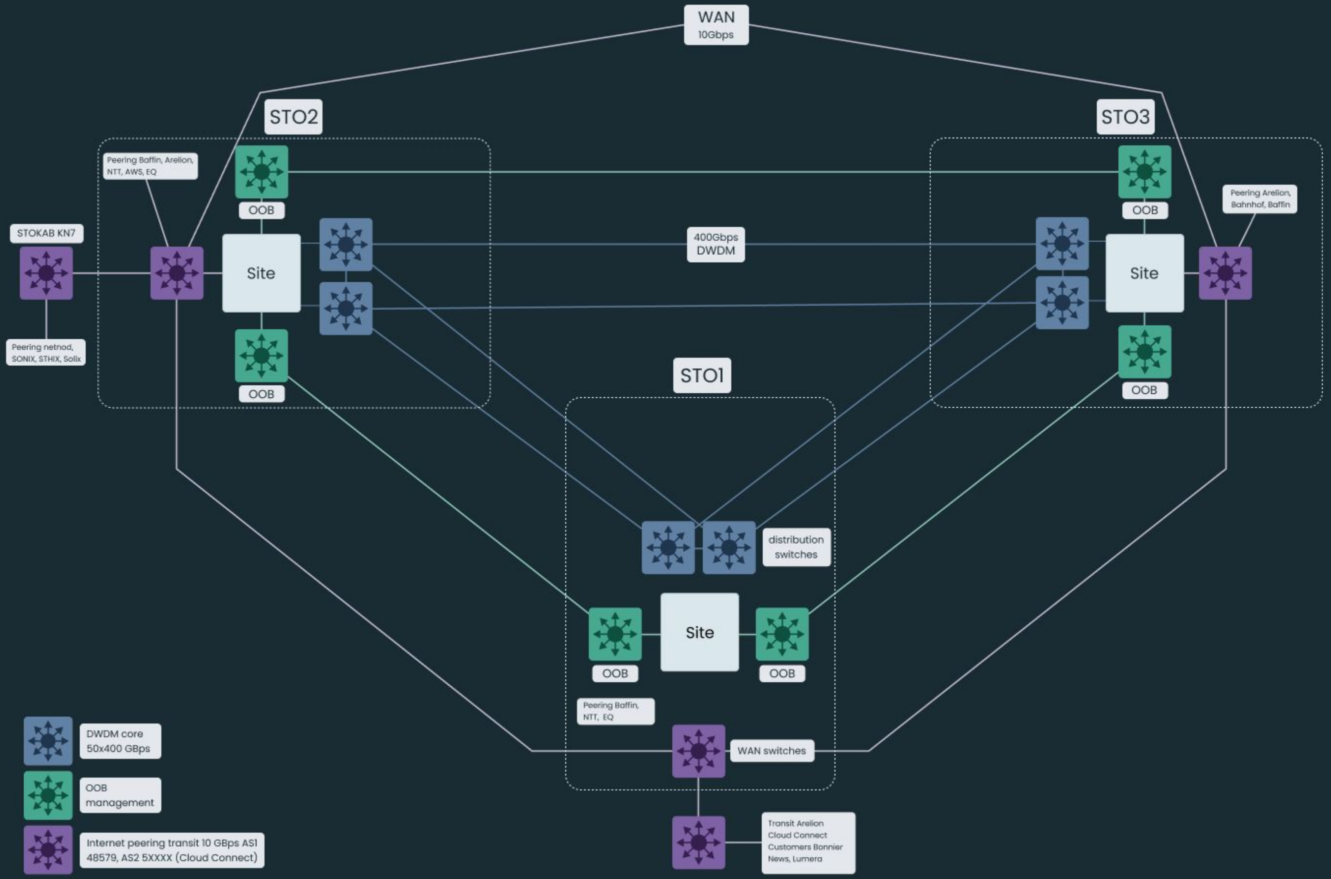
In Other Words

- Redundancy is an important strategy to protect data, enhance system availability, and handle failures.
- Despite the disadvantages of cost and complexity, redundancy is essential for mission-critical systems where data loss or downtime is not acceptable.

Redundancy in Databases

- Physical Redundancy:
    - Copying data across different physical storage devices or geographical locations.
- Logical Redundancy:
    - Copies of the same data at various locations in the database.
        - Purpose: Backup or faster access.
    - Data replicated across database servers or multiple systems to improve fault tolerance and load sharing.

Physical Redundancy


WAN
10Gbps
STO2
STO3
Peering Baffin, Arelion, NTT, AWS, EQ
STOKAB KN7
Peering netnod, SONIX, STHIX, Solix
OOB
Site
OOB
400Gbps DWDM
Peering Arelion, Bahnhof, Baffin
OOB
Site
OOB
STO1
distribution switches
OOB
Site
OOB
Peering Baffin, NTT, EQ
WAN switches
Transit Arelion Cloud Connect Customers Bonnier News, Lumera
DWDM core 50x400 GBps
OOB management
Internet peering transit 10 GBps AS1 48579, AS2 5XXXX (Cloud Connect)

elastx

Data and Logical Redundancy

- Master-Slave:
  - A primary server (master) handles write operations and replicates to one or more secondary servers (slaves) that handle read operations.
  - If the master goes down, a slave can take over after manual or automatic failover.
- Multi-master (e.g., Galera, Active-Active):
  - Multiple servers can both read and write data simultaneously, with each node having the same responsibility. The system synchronizes all nodes.
- Control plane clusters:
  - Distribute data and load over multiple nodes for improved availability and performance. If a node goes offline others continues to handle the work load.

elastx

Redundancy and Autoscaling with Kubernetes

Keep Your App Alive!

elastx

Why? Doesn’t Kubernetes do this already and automagically?

YES **and** **NO**

It does it automagically **if** it has enough information…

elastx

Resource Management

Requests are the minimum resources required to schedule the pod. It is also the amount that is reserved on the node when it is scheduled.

Limits are the maximum allowed usage of resources on the node.

Units
Memory - Bytes (K, M, G, Ki, Mi, Gi)
CPU - Number of Cores (1, 2, 5, fraction 0.1cpu or 100m (millicpu))

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
    image:
images.my-company.example/app:v4
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

Resource Management is Important

Requests and **Limits**

…are the foundation of workload management

…requests are a prerequisite for efficient autoscaling!

elastx

Resource Management - QoS

Guaranteed

Requests = Limits on every container in the pod

Burstable

Partial Requests and Limits on at least one container in the pod

Best effort

No Requests and Limits defined

Looking at Kubernetes Functions…

Pod Topology Spread Constraints	Pod Disruption Budget
Probes	PriorityClass
Horizontal Pod Autoscaler	Cluster Autoscaling

elastx

Pod Topology Spread Constraints

Purpose
Distribute pods evenly across available infrastructure (e.g., nodes, zones) to improve reliability and reduce risk of failure

Use Case
Critical for high-availability applications that need resilience against infrastructure failures

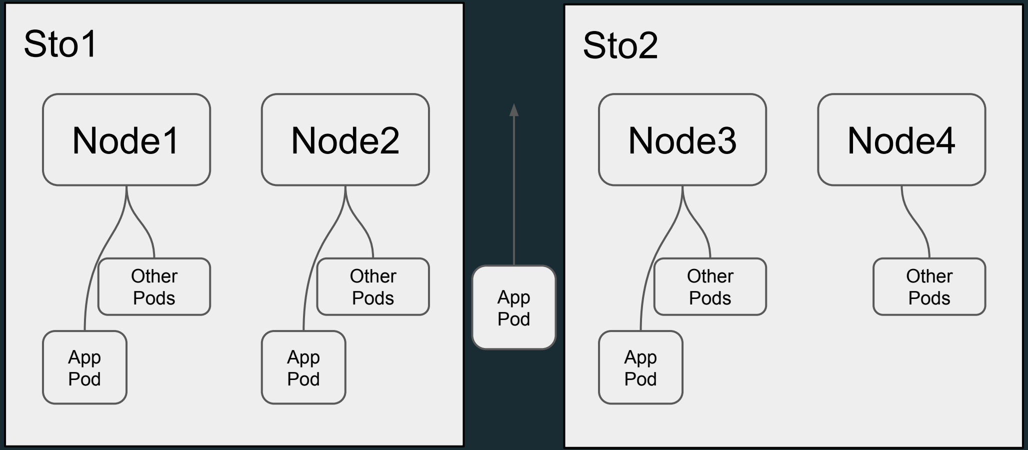
How it Works

- Allows you to define rules for spreading pods across different topological domains like nodes, zones, or regions
- Ensures even distribution to prevent overloading specific areas of the cluster

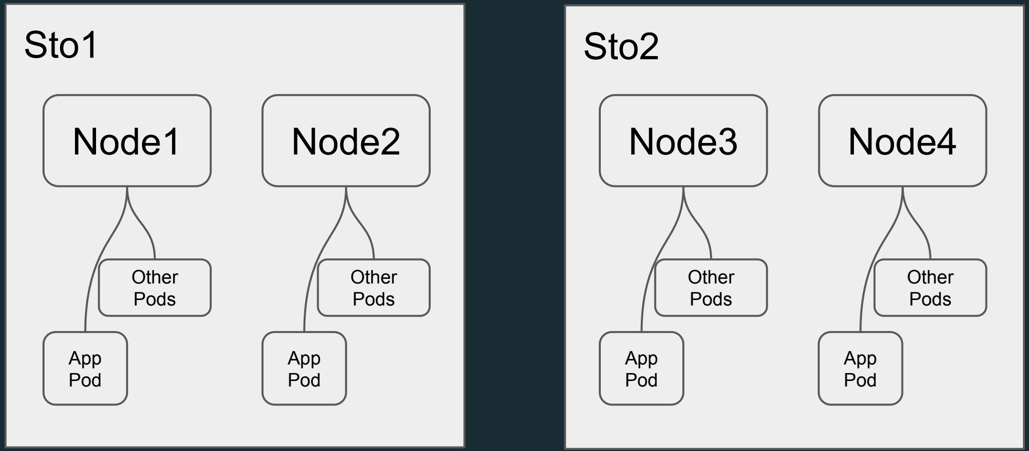
Key Benefits

- **Enhanced Resilience:** Reduces risk from single points of failure
- **Balanced Load:** Avoids resource hotspots by distributing pods
- **Customizable Rules:** Control over distribution to suit application needs

Pod Topology Spread Constraints



Pod Topology Spread Constraints



Pod Disruption Budget

Purpose
Limits the number of pods that can be unavailable during voluntary disruptions, ensuring application stability

Use Case
Ideal for stateful or critical applications that need strict availability guarantees

How it Works

- Defines rules to set the minimum available or maximum unavailable pods
- Applies during planned maintenance (e.g., node updates) or voluntary disruptions (e.g., scaling, manual restarts)

Key Benefits

- **Increased Reliability**: Maintains application availability during disruptions
- **Controlled Maintenance**: Allows safe, predictable updates to the cluster
- **Customizable Policies**: Tailored to application availability requirements

elastx

Probes

Purpose
Monitors pod health to ensure only healthy containers are accessible and running effectively

Use Case
Vital for production applications that require continuous availability and fault tolerance

How it Works

- **Liveness Probe**: Checks if a container is running. Restarts container if it fails
- **Readiness Probe**: Verifies if a container is ready to accept traffic. Removes it from the service endpoint if it fails
- **Startup Probe**: Confirms if a container has started successfully. Useful for slow-starting applications

Key Benefits

- **High Availability:** Automatically restarts or isolates unhealthy containers
- **Traffic Control:** Prevents traffic to pods that aren't ready, enhancing reliability
- **Customization:** Configurable probes to suit application needs

elastx

Priority Class

Purpose
Sets priorities for pods to control scheduling order and preemption, ensuring critical workloads run during resource shortages

Use Case
Essential for production environments with mixed critical and non-critical applications

How it Works

- Assigns priority levels to pods, with higher-priority pods scheduled before lower-priority ones
- **Enables preemption:** Higher-priority pods can evict lower-priority pods to free up resources when necessary

Key Benefits

- **Guaranteed Critical Workloads**: Ensures essential applications run even during peak demand
- **Resource Optimization**: Prevents less important workloads from blocking critical ones
- **Customizable**: Allows you to define priority levels based on application needs

elastx

QoS vs Priority Class

QoS

- **Focus**: Manages resource usage at the node level.
- **Categorization**: Pods are assigned a QoS class (Guaranteed, Burstable, BestEffort) based on their resource requests and limits.
- **Purpose**: Controls how pods behave during resource shortages on a single node, with Guaranteed pods having the highest priority.
- **Usage**: Ensures critical pods receive sufficient resources and are not affected by less important pods.

Priority Class

- **Focus**: Manages scheduling order and preemption at the cluster level.
- **Assignment**: Pods are given an explicit priority based on their defined Priority Class.
- **Purpose**: Determines which pods are scheduled or preempted first when resources are scarce across the cluster.
- **Usage**: Ensures business-critical applications are prioritized over less important workloads.

elastx

Horizontal Pod Autoscaler

Purpose
Automatically adjusts the number of pod replicas based on workload demands to ensure optimal performance

Use Case
Ideal for web applications and APIs with variable traffic patterns

How it Works

- Monitors CPU, memory, or custom metrics to determine pod utilization.
- Scales out by adding pods when demand increases; scales in by removing pods when demand drops

Key Benefits

- **Improved Performance:** Ensures applications have enough resources to handle spikes.
- **Cost Efficiency:** Scales down during low demand to save resources.
- **Customizable Metrics:** Can use custom metrics for more tailored scaling

Cluster Autoscaler

Purpose
Dynamically scales the resource pool i.e number of nodes in a cluster to match workload demands

Use Case
Ideal for unpredictable, bursty workloads like batch jobs and development environments

How it Works

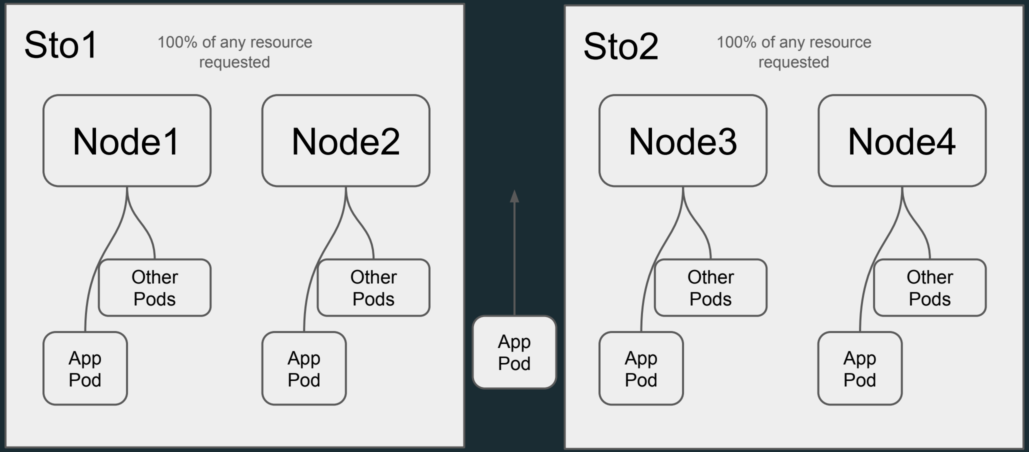
- **Scale-Up**: Adds nodes when pods can't be scheduled due to resource limits
- **Scale-Down**: Removes nodes with low utilization to save costs

Key Benefits

- **Cost Efficiency**: Only pays for the resources you need
- **Flexibility**: Handles variable workloads smoothly
- **High availability**: Ensures that the cluster always has sufficient capacity for the workload.

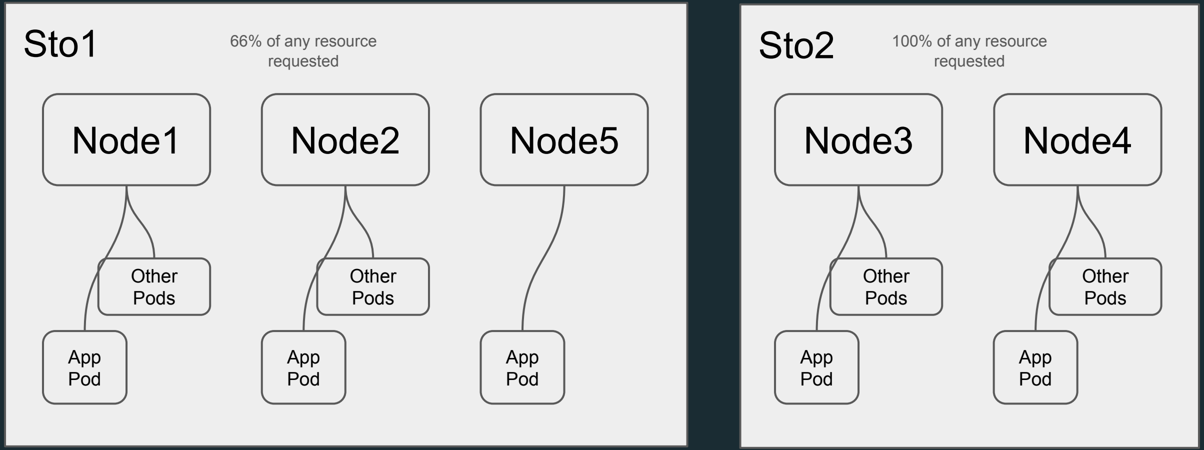
elastx

Cluster Autoscaler



elastx

Cluster Autoscaler



elastx

Thank you!

elastx