Kubernetes Cluster Analysis Report

Generated: 2025-10-27T10:59:12Z

1. Cluster Health Summary

🟢 **Overall Health**: HEALTHY

Key Metrics

Metric	Value
Total Pods	929
Total Nodes	29
Pods Missing Resources	596
OOM Events (Recent)	0
Node Issues	0
Namespaces at Risk	0

⚠ Potential Issues Identified

- **Missing Resource Requests and Limits**: 596 containers are missing resource requests or limits

2. Critical Issues (Top 5)

Issue #1: Missing Resource Requests and Limits

Priority: 1 (1=Highest)

Description: 596 containers are missing resource requests or limits

Impact: Prevents proper scheduling, impacts Velero backups, and can cause cluster instability

Recommendation:

Set resource requests and limits for all containers based on observed usage patterns

Examples:

- `asu/apveclient-7d56989558-nsh6m (container: apve-client)`
- `asu/apveweb-6c9cc5fc89-t8jb7 (container: apveweb)`
- `asu/arrdepapi-54fd8df557-rrk4x (container: arrdepapi)`

Action Items:

1. Audit all pods using: `kubectl get pods --all-namespaces -o json | jq '.items[] | select(.spec.containers[].resources.requests == null)'`
2. Implement LimitRange in each namespace
3. Update deployment manifests with appropriate resource values
4. Use Vertical Pod Autoscaler to recommend resource values

---

3. Resource Management Analysis

Missing Resource Requests and Limits

- **Missing Both**: 190 containers
- **Missing Requests Only**: 0 containers
- **Missing Limits Only**: 406 containers

Impact on Cluster Operations

Velero Backups:

- Pods without resource requests may not be properly backed up
- Restore operations may fail due to resource allocation issues
- Recommendation: Set resource requests to ensure Velero can calculate backup requirements

System Pods:

- System pods may be evicted when resource-constrained workloads consume all node resources
- Can lead to cluster instability and monitoring gaps
- Recommendation: Implement ResourceQuota and LimitRange policies

Cluster Stability:

- Without requests, scheduler cannot make informed placement decisions
- Without limits, pods can consume excessive resources and impact neighbors
- May trigger cascading failures during traffic spikes

Short-Lived Jobs Impact

- **Total Jobs**: 87
- **Short Jobs (<2 min)**: 85
- **Percentage**: 97.7%

Impact: High frequency of short-lived jobs can:

- Create scheduling churn and API server load
- Complicate resource capacity planning
- Impact cluster autoscaler effectiveness

Recommendations:

- Consider batching short jobs or using longer-running workers with queue patterns
- Set appropriate resource requests to prevent over-provisioning

- Implement job cleanup policies to prevent accumulation

Recommended Resource Allocation Strategy

```
# Example resource configuration
resources:
  requests:
    memory: "256Mi"  # Based on observed baseline usage
    cpu: "100m"      # 0.1 CPU cores
  limits:
    memory: "512Mi"  # 2x requests for burst capacity
    cpu: "500m"      # Allow bursting up to 0.5 cores
```

4. Node Analysis

✅ All nodes have healthy resource allocation.

5. RabbitMQ Stability Analysis

RabbitMQ Pods Found: 5

- `rabbitmq/rabbitmq-0`
- `rabbitmq/rabbitmq-1`
- `rabbitmq/rabbitmq-2`
- `rabbitmq/rabbitmq-3`
- `rabbitmq/rabbitmq-4`

Current Configuration

- ✓ Priority Class Configured: false
- ✓ Resource Limits Set: true

Recommendations for Maximum Stability

1. Create High-Priority PriorityClass

```
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: rabbitmq-critical
value: 1000000  # Higher than system-cluster-critical (2000000000 reserved for system)
globalDefault: false
description: "Priority class for RabbitMQ to prevent eviction"
```

2. Configure RabbitMQ Pod Resources

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: rabbitmq
spec:
  priorityClassName: rabbitmq-critical
  containers:
  - name: rabbitmq
    resources:
      requests:
        memory: "2Gi"    # Set based on your observed usage
        cpu: "1000m"     # 1 full CPU core
      limits:
        memory: "4Gi"    # Allow headroom for spikes
        cpu: "2000m"     # Allow burst capacity
```

3. Add PodDisruptionBudget

```yaml
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: rabbitmq-pdb
spec:
  minAvailable: 2  # For clustered RabbitMQ
  selector:
    matchLabels:
      app: rabbitmq
```

4. Node Affinity (Optional)

Consider dedicating specific nodes for RabbitMQ:

```yaml
affinity:
  nodeAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      preference:
        matchExpressions:
        - key: workload-type
          operator: In
          values:
          - messaging
```

How This Ensures RabbitMQ is Last to be Evicted

1. **PriorityClass**: Kubernetes evicts lower-priority pods first during resource pressure

2. **Resource Requests**: Guarantees RabbitMQ gets its requested resources
3. **Resource Limits**: Prevents RabbitMQ from being OOMKilled unnecessarily
4. **PodDisruptionBudget**: Prevents voluntary disruptions during maintenance

6. Namespace-by-Namespace Analysis

🟡 Medium Risk Namespaces

Namespace: `skb`

Metric	Value
Total Pods	27
Pods Missing Requests	8
Pods Missing Limits	25
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-bf49cdcf9-m4znq`
- `contentapi-55f4d4f459-f2lmd`
- `flifointerface-5c985cfc5d-2wnt2`
- `publicdailyapi-7564bdd4d8-d72lq`
- `schedulerservice-6887bb774d-zmc4t`

... and 3 more pods

Recommended Actions:

- Priority: MEDIUM (29.6% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `pvd`

Metric	Value
Total Pods	27
Pods Missing Requests	8
Pods Missing Limits	25
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-76c99cbcc5-5d9vx`
- `contentapi-76f46cf7dc-5vdcq`

- `flifointerface-fd5848574-z5v6h`
- `publicdailyapi-66f8674886-9p5nc`
- `schedulerservice-5f9fc69cd8-zhb9j`

... and 3 more pods

Recommended Actions:

- Priority: MEDIUM (29.6% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `bsr`

Metric	Value
Total Pods	26
Pods Missing Requests	7
Pods Missing Limits	24
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-65db6c5d8f-bngq8`
- `contentapi-64fdccfb49-2jvk6`
- `flifointerface-696dd9c474-r8f7v`
- `schedulerservice-57f86cb5c4-xkj8s`
- `sqldailyjob-29357295-87ms8`

... and 2 more pods

Recommended Actions:

- Priority: MEDIUM (26.9% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `ben`

Metric	Value
Total Pods	26
Pods Missing Requests	7
Pods Missing Limits	24
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-b4db7fcf7-sjjzk`
- `contentapi-64746747bd-w87mn`
- `flifointerface-5b7654874c-7cd6s`
- `publicdailyapi-5b7499db44-lqpmn`
- `schedulerservice-6f9d944d77-6knbh`

... and 2 more pods

Recommended Actions:

- Priority: MEDIUM (26.9% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `asu`

Metric	Value
Total Pods	26
Pods Missing Requests	7
Pods Missing Limits	24
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-6c9cc5fc89-t8jb7`
- `contentapi-6564b4ddf5-8tw55`
- `publicdailyapi-688755697-jkbwd`
- `schedulerservice-78df78b94d-nngxw`
- `sqldailyjob-29357295-shcmt`

... and 2 more pods

Recommended Actions:

- Priority: MEDIUM (26.9% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `lad`

Metric	Value
Total Pods	26

Metric	Value
Pods Missing Requests	7
Pods Missing Limits	24
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-765cb8b76f-r4q4g`
- `contentapi-8576bc59fb-bcb8z`
- `publicdailyapi-65658ccbfb-njx9r`
- `schedulerservice-f64d6c97-67sjl`
- `sqldailyjob-29357295-mnbd6`

... and 2 more pods

Recommended Actions:

- Priority: MEDIUM (26.9% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `jub`

Metric	Value
Total Pods	26
Pods Missing Requests	7
Pods Missing Limits	24
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-588468fbfc-n2zfp`
- `contentapi-65dfcf7f47-qbcwp`
- `publicdailyapi-64b5f8d79-c4fh5`
- `schedulerservice-bddd7d6d-mmq9f`
- `sqldailyjob-29357295-cxlnb`

... and 2 more pods

Recommended Actions:

- Priority: MEDIUM (26.9% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `fca`

Metric	Value
Total Pods	27
Pods Missing Requests	8
Pods Missing Limits	25
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-d9b79b8c-mkwlb`
- `contentapi-5bdf8c86b-zqxjf`
- `flifointerface-7c5b86868c-57zvt`
- `publicdailyapi-56f6fd749f-4p72d`
- `schedulerservice-6bb9c5458c-n4llr`

... and 3 more pods

Recommended Actions:

- Priority: MEDIUM (29.6% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `day`

Metric	Value
Total Pods	26
Pods Missing Requests	7
Pods Missing Limits	24
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-88f85b7b5-h6lth`
- `contentapi-787bc78b84-8x454`
- `publicdailyapi-5cd7b998f-dvq2l`
- `schedulerservice-7b785f999c-4dhl8`
- `sqldailyjob-29357295-6h7bg`

... and 2 more pods

Recommended Actions:

- Priority: MEDIUM (26.9% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `ord`

Metric	Value
Total Pods	26
Pods Missing Requests	7
Pods Missing Limits	24
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-5f5fdc6765-q2mpw`
- `contentapi-7548b8f8d8-j27tj`
- `flifointerface-6ff7f556c-wt9jd`
- `schedulerservice-7bf78fb6b9-q9fd7`
- `sqldailyjob-29357295-d5k8b`

... and 2 more pods

Recommended Actions:

- Priority: MEDIUM (26.9% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `guc`

Metric	Value
Total Pods	27
Pods Missing Requests	8
Pods Missing Limits	25
Risk Level	MEDIUM

Critical Pods Missing Resources:

- `apveweb-77698f676f-sjtf2`
- `contentapi-674c76bc5b-vpnt2`
- `flifointerface-74cd74ff8c-gdqtx`
- `publicdailyapi-569d7bc88d-bpdm5`
- `schedulerservice-d8787b4fc-xs6jz`

... and 3 more pods

Recommended Actions:

- Priority: MEDIUM (29.6% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

🟢 Low Risk Namespaces

Namespace: `pmg`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-c87f4946c-p8bwz`
- `contentapi-76df55d9c-5k2v8`
- `schedulerservice-5545977ccd-dvwh9`
- `sqldailyjob-29357295-7cc99`
- `sqldailyjob-29358735-xfzrh`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `cmg`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-77676d9db9-zvfj9`
- `contentapi-55c5958574-rpnhd`
- `schedulerservice-548f7497b7-kmmr6`
- `sqldailyjob-29357295-wd686`
- `sqldailyjob-29358735-v27nd`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `moc`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-6d65d6d8fc-rmltt`
- `contentapi-7669fd54c-wqwkb`
- `schedulerservice-7f5f4f5c88-kp878`
- `sqldailyjob-29357295-wh24s`
- `sqldailyjob-29358735-9lq4m`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `mab`

Metric	Value
Total Pods	25
Pods Missing Requests	6

Metric	Value
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-5b5959d687-h9dxh`
- `contentapi-678b99ddc5-pzlqd`
- `schedulerservice-85948576-ztj4q`
- `sqldailyjob-29357295-s9lxh`
- `sqldailyjob-29358735-f8q5x`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `stm`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-79d96b57f-8kphf`
- `contentapi-776b5f9b87-x7bch`
- `schedulerservice-d5d8466d4-6vbk7`
- `sqldailyjob-29357295-hlrwn`
- `sqldailyjob-29358735-fvnpr`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `cks`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-5bfd5c8d69-6zr25`
- `contentapi-7898c95f84-hppjk`
- `schedulerservice-5f54d88c69-vrqwk`
- `sqldailyjob-29357295-9wss9`
- `sqldailyjob-29358735-mqknv`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `atm`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-785bc568f9-fhskb`
- `contentapi-6f9c759885-s8kzc`
- `schedulerservice-cccb57894-247qb`
- `sqldailyjob-29357295-qwt4l`
- `sqldailyjob-29358735-qcrvh`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `udi`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-5995b7459f-2mqk9`
- `contentapi-668db64649-pxvwt`
- `schedulerservice-6f54959b54-hwhvq`
- `sqldailyjob-29357295-mhxkg`
- `sqldailyjob-29358735-zslhh`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `uba`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-577c8cb887-wx7dv`
- `contentapi-d47db87f4-hwxrq`
- `schedulerservice-796768d479-pmmhf`
- `sqldailyjob-29357295-ktfbk`
- `sqldailyjob-29358735-qnwgh`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `cgr`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-cf5946f8c-jmmtp`
- `contentapi-569796874b-7dt84`
- `schedulerservice-799cb87f75-5r8zk`
- `sqldailyjob-29357295-m2wkc`
- `sqldailyjob-29358735-rc2pl`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace: `cgh`

Metric	Value
Total Pods	25
Pods Missing Requests	6
Pods Missing Limits	23
Risk Level	LOW

Critical Pods Missing Resources:

- `apveweb-8687559c95-69nlg`

- `contentapi-85b5d48b5b-89nwk`
- `schedulerservice-6fb869cb9c-rbg9r`
- `sqldailyjob-29357295-vg2fr`
- `sqldailyjob-29358735-hrdtl`

... and 1 more pods

Recommended Actions:

- Priority: LOW (24.0% pods affected)
- Implement LimitRange to set defaults for new pods
- Update existing deployments with appropriate resource requests/limits
- Monitor resource usage patterns for 1-2 weeks before setting permanent values

Namespace-Level Recommendations

1. **Implement LimitRange defaults**:

```yaml
apiVersion: v1
kind: LimitRange
metadata:
  name: default-limits
  namespace: <namespace>
spec:
  limits:
  - default:
      memory: 512Mi
      cpu: 500m
    defaultRequest:
      memory: 256Mi
      cpu: 100m
    type: Container
```

2. **Set ResourceQuota per namespace**:

```yaml
apiVersion: v1
kind: ResourceQuota
metadata:
  name: namespace-quota
  namespace: <namespace>
spec:
  hard:
    requests.cpu: "10"
    requests.memory: 20Gi
    limits.cpu: "20"
    limits.memory: 40Gi
```

7. AI-Enhanced Insights

Kubernetes Cluster Analysis Report

Cluster Health Summary

The cluster is currently operating in a healthy state with no OOM (Out of Memory) events reported. However, a significant number of pods (596 out of 929) are missing resource requests and limits, which can lead to improper scheduling and potential instability during high load periods. This could also affect Velero's ability to perform reliable backups, making this a priority issue.

Critical Issues

1. Missing Resource Requests and Limits (Priority 1)

- **Impact**: The lack of defined resource requests and limits can lead to unpredictable pod behavior, poor scheduling decisions, and can affect the stability of system-critical services.
- **Recommendation**:
  - **Action**: Set appropriate resource requests and limits for all containers based on historical usage metrics.
  - **Example**:

    ```yaml
    resources:
      requests:
        cpu: "500m"
        memory: "512Mi"
      limits:
        cpu: "1"
        memory: "1Gi"
    ```

2. RabbitMQ Stability (Priority 2)

- **Impact**: Without a priority class, RabbitMQ may be evicted during OOM situations.
- **Recommendation**:
  - **Action**: Create a priority class for RabbitMQ to ensure it is the last workload to be evicted.
  - **Example**:

    ```yaml
    apiVersion: scheduling.k8s.io/v1
    kind: PriorityClass
    metadata:
      name: rabbitmq-priority
    value: 1000000
    globalDefault: false
    description: "Priority class for RabbitMQ pods"
    ```

  - **Resource Allocation**: Ensure RabbitMQ pods have resource limits set and consider increasing them based on peak usage.

3. Short-Lived Jobs Impact (Priority 3)

- **Impact**: The presence of 85 short-lived jobs (<2 min) can cause scheduling overhead and resource contention, particularly during burst workloads.
- **Recommendation**:
  - **Action**: Analyze the need for these jobs and consider consolidating or scheduling them during off-peak hours to reduce their impact on cluster stability.

Resource Management

Missing Requests/Limits

- **Impact**: Without proper resource definitions, pods can consume more resources than anticipated, leading to resource contention and instability, particularly for system components like Velero.
- **Benefit**: Establishing requests and limits will help the Kubernetes scheduler make better decisions, ultimately improving cluster stability and performance.

Short-Lived Jobs

- **Analysis**: Short-lived jobs consume resources during their execution and can lead to spikes in resource usage, affecting overall cluster performance.
- **Recommendation**:
  - Monitor the resource consumption of these jobs and adjust their scheduling or execution strategy to mitigate impacts.

Node Analysis

- **Poorly Balanced Node Pools**: Review node allocation to ensure optimal distribution of workloads.
- **OOMKilled Events**: Since there are no OOMKilled events currently, it's important to maintain vigilance and set resource limits to prevent future occurrences.
- **High Resource Requests**: Analyze nodes with high resource requests to identify potential bottlenecks in resource allocation.
- **Cluster Autoscaling**: Check current autoscaling settings to ensure they can accommodate peak loads and rapid scaling when needed.

Namespace Analysis

Risk Level Breakdown

- **Medium Risk Namespaces**:

  - **skb**: 8 pods missing resources
  - **pvd**: 8 pods missing resources
  - **bsr**: 7 pods missing resources
  - **Recommended Action**: Prioritize defining resource requests/limits for these namespaces.

- **Low Risk Namespaces**:

  - **pmg**: 6 pods missing resources
  - **cmg**: 6 pods missing resources
  - **Recommended Action**: While lower risk, it's still advisable to define resource limits to prevent any future issues.

Namespace-Specific Recommendations

- **Example for skb namespace**:
  - **Critical Pods**: Identify top resource consumers and set limits.
  - **Suggested Resource Values**:

    ```yaml
    resources:
      requests:
        cpu: "250m"
        memory: "256Mi"
      limits:
        cpu: "500m"
        memory: "512Mi"
    ```

Automation and Preventive Measures

1. **Automate Resource Limit Enforcement**:

   - Implement admission controllers (e.g., OPA/Gatekeeper) to enforce resource requests and limits on pod creation.

2. **Monitoring and Alerts**:

   - Set up Prometheus alerts for resource usage patterns to proactively identify when resources are nearing limits.

3. **Regular Audits**:

   - Schedule regular audits of all namespaces to ensure compliance with resource policies and to rectify any missing configurations.

4. **Documentation and Training**:

   - Ensure all teams are trained on the importance of resource management and the implications of missing configurations.

By following these recommendations, the cluster can maintain its health and performance, ensuring smooth operation and stability for all workloads.

Enhanced Recommendations

- AI analysis provided in summary section

Risk Assessment

See AI summary for detailed risk assessment

Automation Suggestions

- Implement ResourceQuota policies
- Set up LimitRange defaults for namespaces
- Configure PodDisruptionBudgets for critical workloads

Appendix

Data Collection Summary

- **Collection Time**: 2025-10-27T10:59:12Z
- **Total Pods Analyzed**: 929
- **Total Nodes Analyzed**: 29
- **Events Processed**: 991

Next Steps

1. Review critical issues and prioritize based on business impact
2. Implement resource requests/limits for high-risk namespaces first
3. Set up monitoring for OOM events and resource utilization
4. Establish policies (LimitRange, ResourceQuota) to prevent future issues
5. Schedule follow-up analysis after implementing changes

Resources

- Kubernetes Best Practices - Resource Management
- Pod Priority and Preemption
- Pod Disruption Budgets
- Vertical Pod Autoscaler