



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Help! Please Rescue Not-Ready Nodes Immediately

*Xiaoyu Zhang (Tencent Cloud)*

*Di Xu (Ant Group)*



# About us



Xiaoyu Zhang  
Kubernetes Member  
Tencent Cloud  
[@zhangxiaoyu-zidif](#)



Di Xu  
Kubernetes Member  
Top 50 Code  
Contributor  
Ant Group  
[@dixudx](#)



# Agenda



- What's the Problem
- What is NPD?
- Node Self Healing
- Lessons We've Learnt
- Some Discussions
- Summary



# Node Is Not Ready



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

```
1 [zhangxiaoyu-zidif@tencent ~]# kubectl get nodes
2 NAME                STATUS              AGE       VERSION
3 test-001            Ready,SchedulingDisabled,master 1d        v1.18.0
4 test-002            Ready               1d        v1.18.0
5 test-003            NotReady           1d        v1.18.0
6 test-004            Ready              1d        v1.18.0
7 test-005            Ready              1d        v1.18.0
```

## What Should We Do?

1. Find out what happened immediately ...
  - a. Check Node.Status.Conditions field.
  - b. Review Prometheus;
  - c. ELK Suit to analysis all kinds of logs;
2. Fix it ASAP!





# Node Problem Detector



KubeCon



CloudNativeCon

Europe 2020

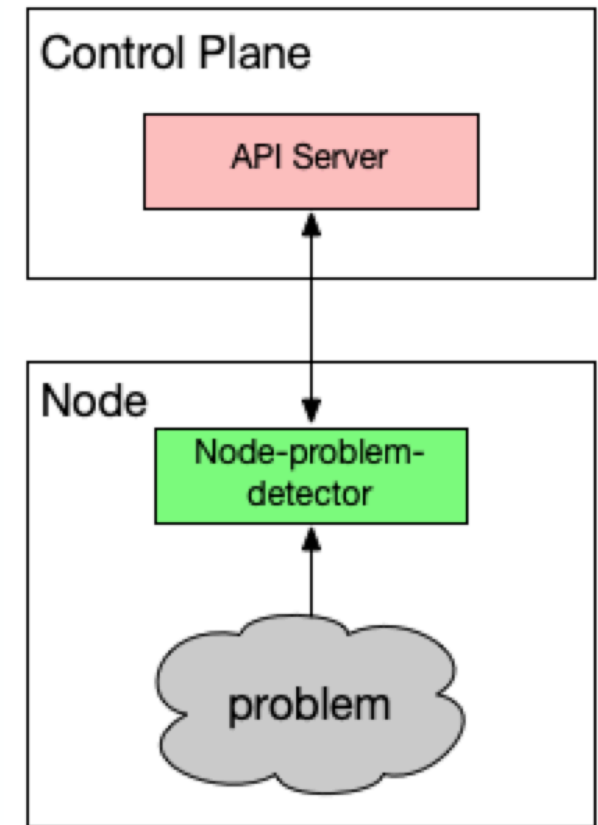
*Virtual*

**Node Problem Detector:** A Daemon Set detects node problems and reports them to API-Server.

User could run daemons or shell script to check specified problems periodically.

With Two Types:

- **NodeCondition**: A field in **NodeStatus** describes the condition of a node. Permanent problem making the node unavailable such as, KernelDeadlock, DockerHung, BadDisk etc;
- **Event**: A report of an event somewhere in the cluster. Temporary problem but informative, such as OOM Kill, etc.



# Node Self Healing

## What NPD Does Now?

Only detect problems and report them, that's not far enough.

## What We Really Need?

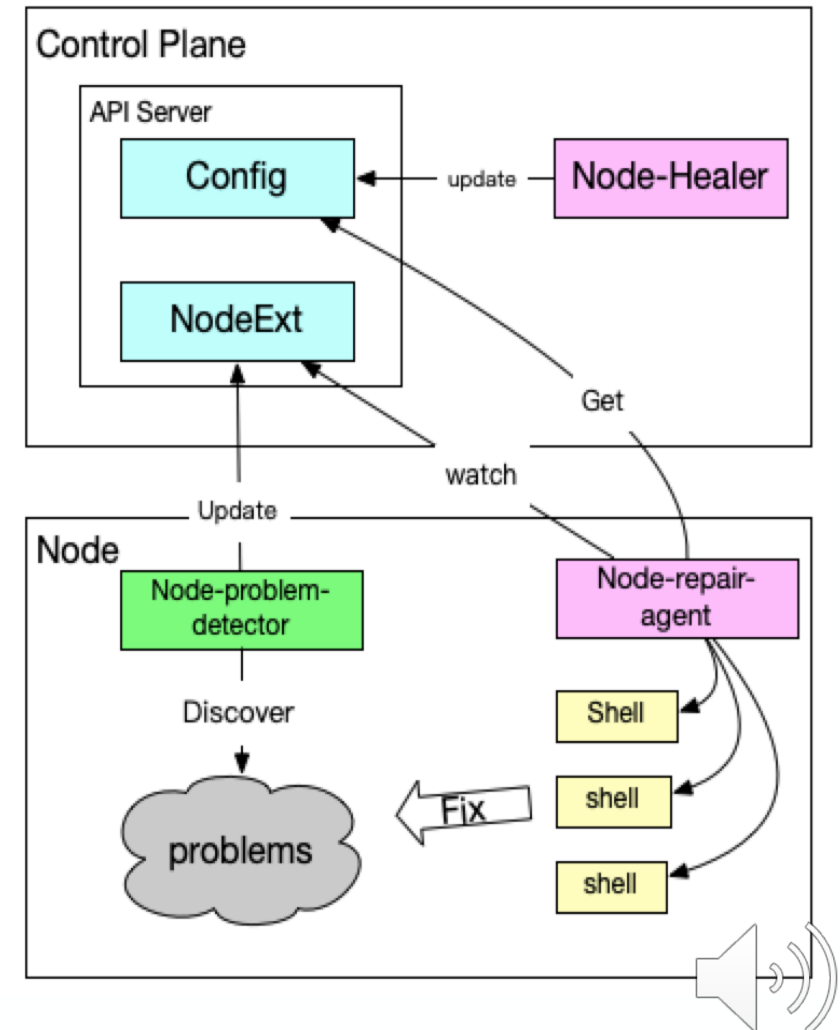
Observe events and/or node conditions and take actions to bring the Kubernetes cluster back to a healthy state.

## The Missing Parts

Fix them automatically using prepared repairing strategies.

So we should:

- Make node problem visible
- Update CRD, not the native resource
- Watch CRD, and fix problems immediately
- If problem can not be fixed, drain the pod and taint this Node



# Problem Category



## Hardware

- MCE (Machine Check Exceptions), such as Memory Errors, Processor Over-Heating, etc;
- Disk Problems, such as Bad Sectors, SSD Controller Damaged, etc;
- NICs Issues;
- Cloud Disk IO Hang;

## System Software

- Kernel Bug
- Systemd hang
- Crond Exceptions
- Logrotate Not Working: disk full

## Runtime

- Kubelet Bug, such PLEG;
- Container runtime, such docker, containerd, etc;
- CNI, CSI;

...



# How We Detected



KubeCon



CloudNativeCon

Europe 2020

Virtual

```
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: node-problem-detector-config
5    namespace: kube-system
6  data:
7    docker-monitor.json: |
8    {
9      "plugin": "journald",
10     "pluginConfig": {
11       "source": "dockerd"
12     },
13     "logPath": "/var/log/journal",
14     "lookback": "5m",
15     "bufferSize": 10,
16     "source": "docker-monitor",
17     "conditions": [],
18     "rules": [
19       {
20         "type": "temporary",
21         "reason": "CorruptDockerImage",
22         "pattern": "Error trying v2 registry: failed to register
23       }
24     ]
25   }
```

```
26  kernel-monitor.json: |
27  {
28     "plugin": "kmsg",
29     "logPath": "/dev/kmsg",
30     "lookback": "5m",
31     "bufferSize": 10,
32     "source": "kernel-monitor",
33     "conditions": [
34       {
35         "type": "KernelDeadlock",
36         "reason": "KernelHasNoDeadlock",
37         "message": "kernel has no deadlock"
38       },
39       {
40         "type": "ReadOnlyFilesystem",
41         "reason": "FilesystemIsReadOnly",
42         "message": "Filesystem is read-only"
43       }
44     ],
45     "rules": [
46       {
47         "type": "temporary",
48         "reason": "OOMKilling",
49         "pattern": "Kill process \\d+ (.+) scor
50     },
```



## Check fd Plugin

- Modify node condition when system fd amount exceeds 80%
- Use goroutine to concurrently calculate fd amount;

```
"rules": [  
  {  
    "type": "permanent",  
    "condition": "FDPressure",  
    "reason": "NodeHasFDPressure",  
    "message": "too many fds have been used",  
    "path": "/home/kubernetes/bin/check-fd",  
    "args": [  
      "-p=/host/proc"  
    ],  
    "timeout": "10s"  
  }  
]
```

```
FDPressure      False    Tue, 21 Jul 2020 20:56:15 +0800    Tue, 21 Jul 2020 20:16:10 +0800    NodeHasNoFDPressure  
node has no fd pressure
```

## Inode monitoring

- Modify node condition when the system disk inode amount exceeds 80%

```
InodesPressure  False    Tue, 21 Jul 2020 20:50:00 +0800    Tue, 21 Jul 2020 11:13:49 +0800    NodeHasNoInodesPressure  
node has no inodes pressure
```



# Lessons We've Learnt



- Remedy scripts are mostly written in shell
- Too many corner cases to be considered
- Hard to maintain
- Hard to extend
- Hard to keep idempotent
- Mostly unable to perform atomic operations
- Complex Interdependence



## How to Deploy

- rpm/deb packages + systemd
- DaemonSet

## How to Upgrade

- CI/CD Working Flow

## How to Config

- Remedy plugins & Strategy
- Static (built-in) vs dynamic (configmap)

## Resource Limitations

- NPD Resource Overheads





# Community Recommendations



## planetlabs/draino

- <https://github.com/planetlabs/draino>
- Automatically cordon and drain Kubernetes nodes based on node conditions.
- Cordoned immediately and drained after a configurable drain-buffer time

```
> kubectl describe node {node-name}
.....
Unschedulable:      true
Conditions:
  Type              Status  LastHeartbeatTime             LastTransitionTime             Reason                          Message
  ----              -
  OutOfDisk         False   Fri, 20 Mar 2020 15:52:41 +0100 Fri, 20 Mar 2020 14:01:59 +0100 KubeletHasSufficientDisk       kubelet has sufficient disk space available
  MemoryPressure    False   Fri, 20 Mar 2020 15:52:41 +0100 Fri, 20 Mar 2020 14:01:59 +0100 KubeletHasSufficientMemory     kubelet has sufficient memory available
  DiskPressure      False   Fri, 20 Mar 2020 15:52:41 +0100 Fri, 20 Mar 2020 14:01:59 +0100 KubeletHasNoDiskPressure       kubelet has no disk pressure
  PIDPressure       False   Fri, 20 Mar 2020 15:52:41 +0100 Fri, 20 Mar 2020 14:01:59 +0100 KubeletHasSufficientPID        kubelet has sufficient PID available
  Ready             True    Fri, 20 Mar 2020 15:52:41 +0100 Fri, 20 Mar 2020 14:02:09 +0100 KubeletReady                    kubelet is posting ready status. AppArmor enabled
  ec2-host-retirement True    Fri, 20 Mar 2020 15:23:26 +0100 Fri, 20 Mar 2020 15:23:26 +0100 NodeProblemDetector            Condition added with tooling
  DrainScheduled    True    Fri, 20 Mar 2020 15:50:50 +0100 Fri, 20 Mar 2020 15:23:26 +0100 Draino                          Drain activity scheduled 2020-03-20T15:50:34+01:00
```





# Summary

- Repairing policed are NOT universally valid in any infrastructure;
- No silver bullet;
- “KISS” ( **K**ep your **I**nfrastructure **S**imple and **S**tandard );





**KubeCon**



**CloudNativeCon**

Europe 2020

*Virtual*

