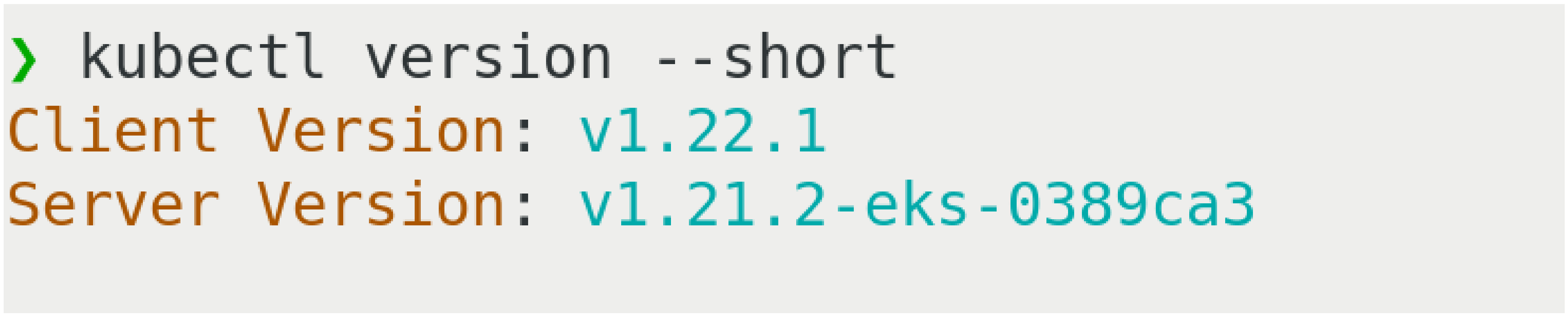
1. kubectl version –short

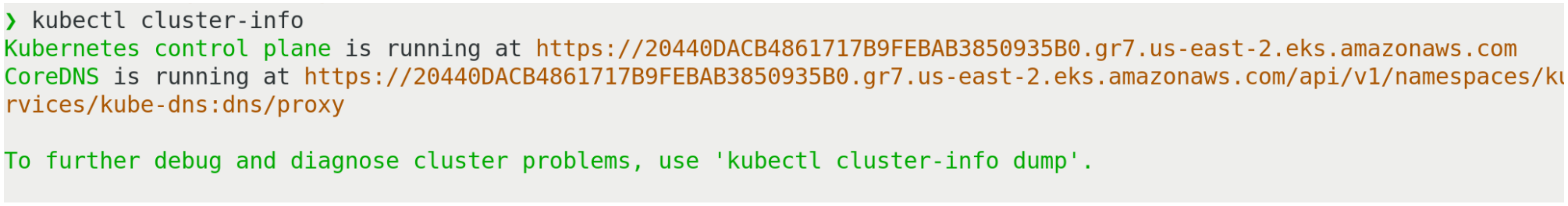


```
❯ kubectl version --short
Client Version: v1.22.1
Server Version: v1.21.2-eks-0389ca3
```

This command, helps us see **which version of the API server** is running.

This gives us important information when we’re troubleshooting specific errors, and it’s very useful to know if we’re on an older cluster like 1.16.

2. kubectl cluster-info



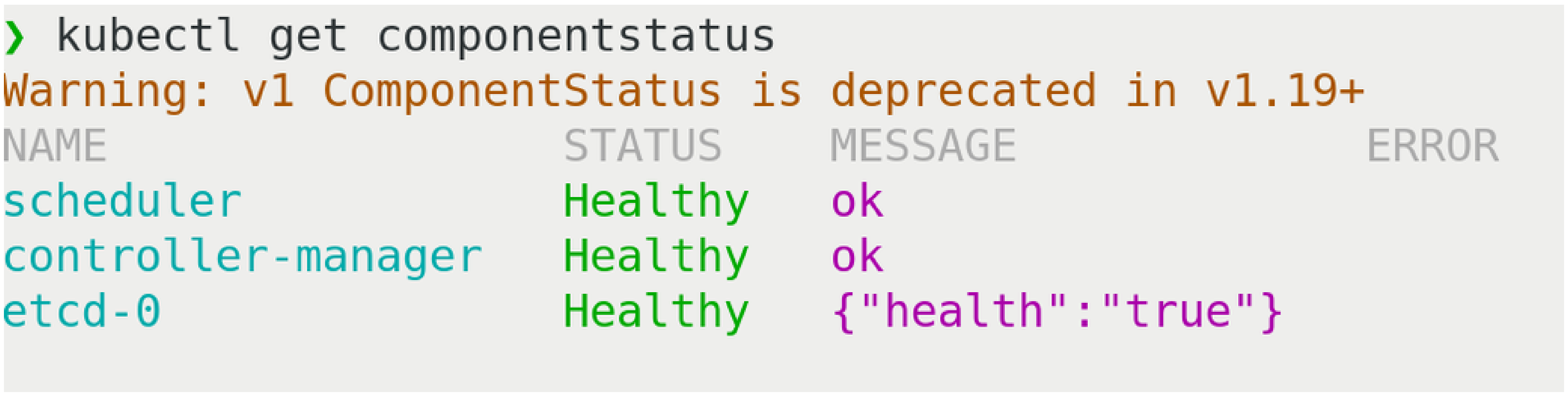
```
❯ kubectl cluster-info
Kubernetes control plane is running at https://20440DACB4861717B9FEBAB3850935B0.gr7.us-east-2.eks.amazonaws.com
CoreDNS is running at https://20440DACB4861717B9FEBAB3850935B0.gr7.us-east-2.eks.amazonaws.com/api/v1/namespaces/ku
rvices/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Next, we should understand **where the cluster is running** and **if CoreDNS is running.**

You can parse the control plane URL to know if you're dealing with a hosted cluster or something on-premises.

3. kubectl get componentstatus

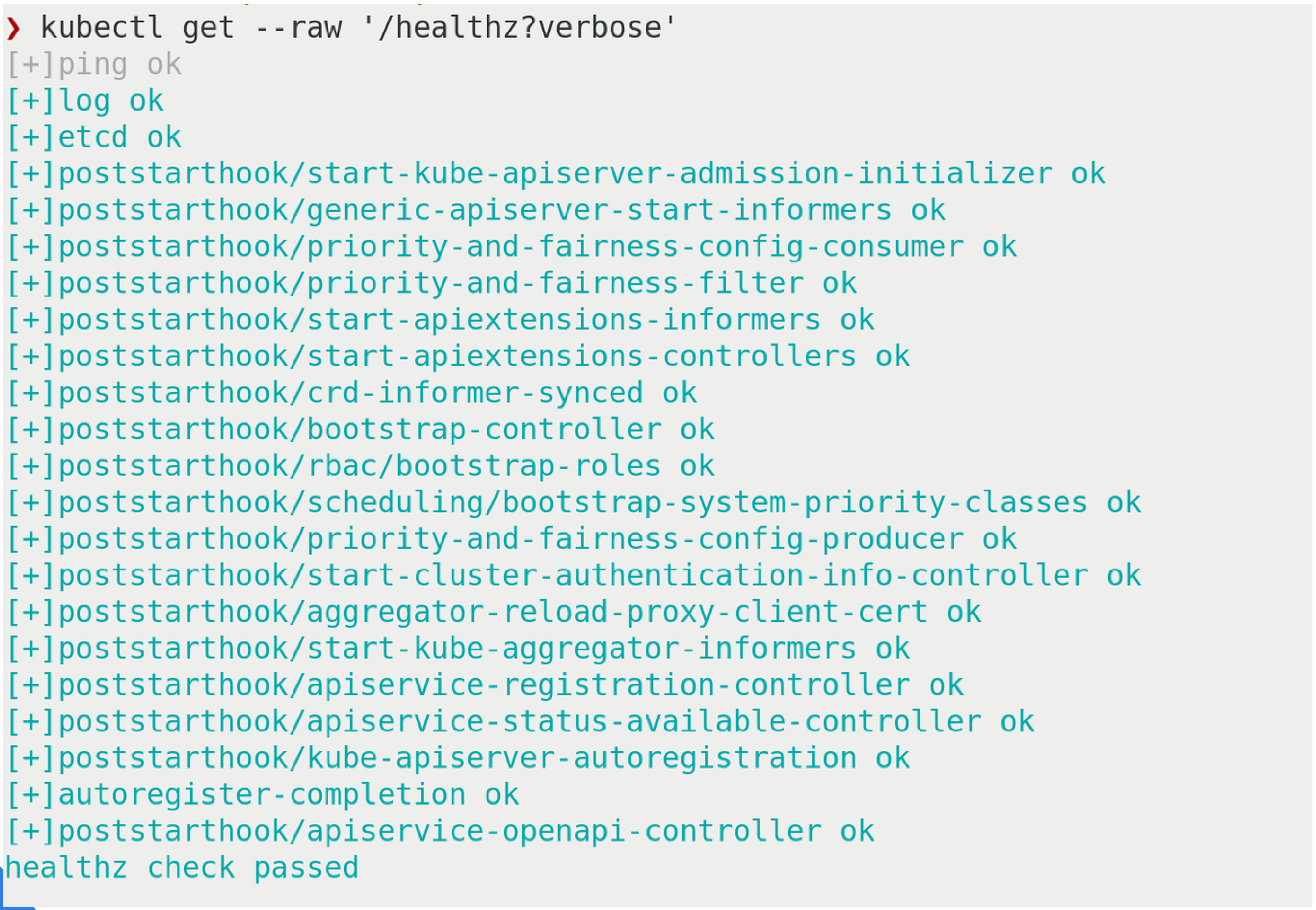


```
❯ kubectl get componentstatus
Warning: v1 ComponentStatus is deprecated in v1.19+
NAME                 STATUS    MESSAGE             ERROR
scheduler            Healthy   ok
controller-manager   Healthy   ok
etcd-0               Healthy   {"health":"true"}
```

This command will be the easiest way to discover if your **scheduler, controller-manager** and **etcd node(s) are healthy**. These are all critical control plane components to run your pods. Look for errors on the components that don't show an "ok" status.

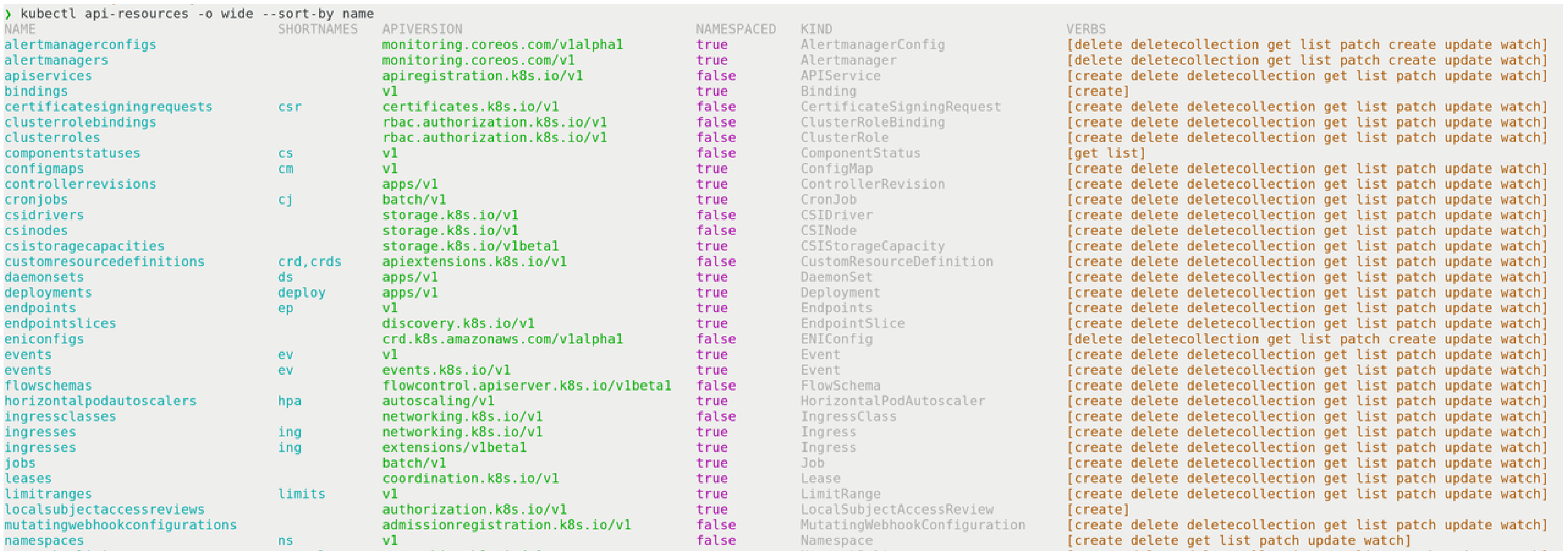
An alternative option to see other health endpoints, including etcd, is

kubectl get --raw '/healthz?verbose':



```
❯ kubectl get --raw '/healthz?verbose'
[+]ping ok
[+]log ok
[+]etcd ok
[+]poststarthook/start-kube-apiserver-admission-initializer ok
[+]poststarthook/generic-apiserver-start-informers ok
[+]poststarthook/priority-and-fairness-config-consumer ok
[+]poststarthook/priority-and-fairness-filter ok
[+]poststarthook/start-apiextensions-informers ok
[+]poststarthook/start-apiextensions-controllers ok
[+]poststarthook/crd-informer-synced ok
[+]poststarthook/bootstrap-controller ok
[+]poststarthook/rbac/bootstrap-roles ok
[+]poststarthook/scheduling/bootstrap-system-priority-classes ok
[+]poststarthook/priority-and-fairness-config-producer ok
[+]poststarthook/start-cluster-authentication-info-controller ok
[+]poststarthook/aggregator-reload-proxy-client-cert ok
[+]poststarthook/start-kube-aggregator-informers ok
[+]poststarthook/apiservice-registration-controller ok
[+]poststarthook/apiservice-status-available-controller ok
[+]poststarthook/kube-apiserver-autoregistration ok
[+]autoregister-completion ok
[+]poststarthook/apiservice-openapi-controller ok
healthz check passed
```

4. kubectl api-resources -o wide –sort-by name

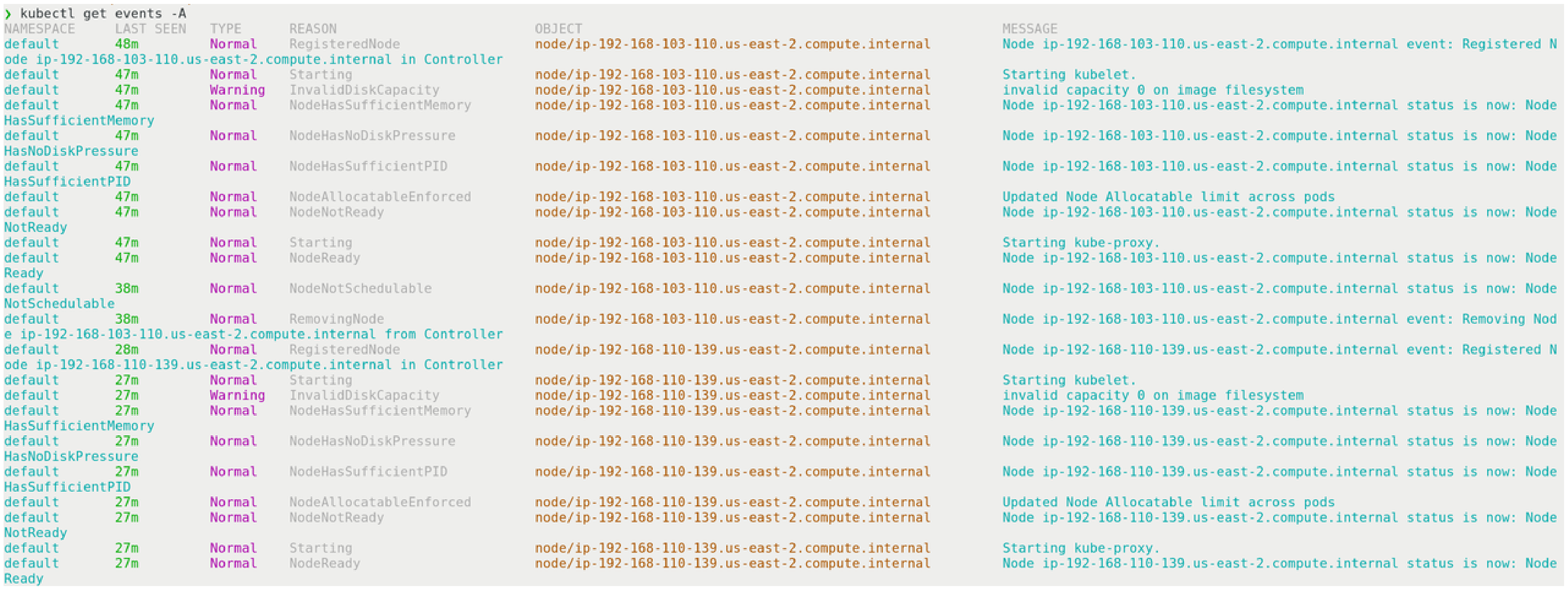


```
❯ kubectl api-resources -o wide --sort-by name
NAME                              SHORTNAMES   APIVERSION                             NAMESPACED   KIND                             VERBS
alertmanagerconfigs                            monitoring.coreos.com/v1alpha1         true         AlertmanagerConfig               [delete deletecollection get list patch create update watch]
alertmanagers                                  monitoring.coreos.com/v1               true         Alertmanager                     [delete deletecollection get list patch create update watch]
apiservices                                    apiregistration.k8s.io/v1              false        APIService                       [create delete deletecollection get list patch update watch]
bindings                                       v1                                     true         Binding                          [create]
certificatesigningrequests        csr          certificates.k8s.io/v1                 false        CertificateSigningRequest        [create delete deletecollection get list patch update watch]
clusterrolebindings                            rbac.authorization.k8s.io/v1           false        ClusterRoleBinding               [create delete deletecollection get list patch update watch]
clusterroles                                   rbac.authorization.k8s.io/v1           false        ClusterRole                      [create delete deletecollection get list patch update watch]
componentstatuses                 cs           v1                                     false        ComponentStatus                  [get list]
configmaps                        cm           v1                                     true         ConfigMap                        [create delete deletecollection get list patch update watch]
controllerrevisions                            apps/v1                                true         ControllerRevision               [create delete deletecollection get list patch update watch]
cronjobs                          cj           batch/v1                               true         CronJob                          [create delete deletecollection get list patch update watch]
csidrivers                                     storage.k8s.io/v1                      false        CSIDriver                        [create delete deletecollection get list patch update watch]
csinodes                                       storage.k8s.io/v1                      false        CSINode                          [create delete deletecollection get list patch update watch]
csistoragecapacities                           storage.k8s.io/v1beta1                 true         CSIStorageCapacity               [create delete deletecollection get list patch update watch]
customresourcedefinitions         crd,crds     apiextensions.k8s.io/v1                false        CustomResourceDefinition         [create delete deletecollection get list patch update watch]
daemonsets                        ds           apps/v1                                true         DaemonSet                        [create delete deletecollection get list patch update watch]
deployments                       deploy       apps/v1                                true         Deployment                       [create delete deletecollection get list patch update watch]
endpoints                         ep           v1                                     true         Endpoints                        [create delete deletecollection get list patch update watch]
endpointslices                                 discovery.k8s.io/v1                    true         EndpointSlice                    [create delete deletecollection get list patch update watch]
eniconfigs                                     crd.k8s.amazonaws.com/v1alpha1         false        ENIConfig                        [delete deletecollection get list patch create update watch]
events                            ev           v1                                     true         Event                            [create delete deletecollection get list patch update watch]
events                            ev           events.k8s.io/v1                       true         Event                            [create delete deletecollection get list patch update watch]
flowschemas                                    flowcontrol.apiserver.k8s.io/v1beta1   false        FlowSchema                       [create delete deletecollection get list patch update watch]
horizontalpodautoscalers          hpa          autoscaling/v1                         true         HorizontalPodAutoscaler          [create delete deletecollection get list patch update watch]
ingressclasses                                 networking.k8s.io/v1                   false        IngressClass                     [create delete deletecollection get list patch update watch]
ingresses                         ing          networking.k8s.io/v1                   true         Ingress                          [create delete deletecollection get list patch update watch]
ingresses                         ing          extensions/v1beta1                     true         Ingress                          [create delete deletecollection get list patch update watch]
jobs                                           batch/v1                               true         Job                              [create delete deletecollection get list patch update watch]
leases                                         coordination.k8s.io/v1                 true         Lease                            [create delete deletecollection get list patch update watch]
limitranges                       limits       v1                                     true         LimitRange                       [create delete deletecollection get list patch update watch]
localsubjectaccessreviews                      authorization.k8s.io/v1                true         LocalSubjectAccessReview         [create]
mutatingwebhookconfigurations                  admissionregistration.k8s.io/v1        false        MutatingWebhookConfiguration     [create delete deletecollection get list patch update watch]
namespaces                        ns           v1                                     false        Namespace                        [create delete get list patch update watch]
```

I like to list all the resources sorted by name for consistency. It’s easier for me to scan the resources in alphabetical order. Adding **-o wide** will show the verbs available on each resource.

Using this command will tell you what CRDs (custom resource definitions) have been installed in your cluster and what API version each resource is at.

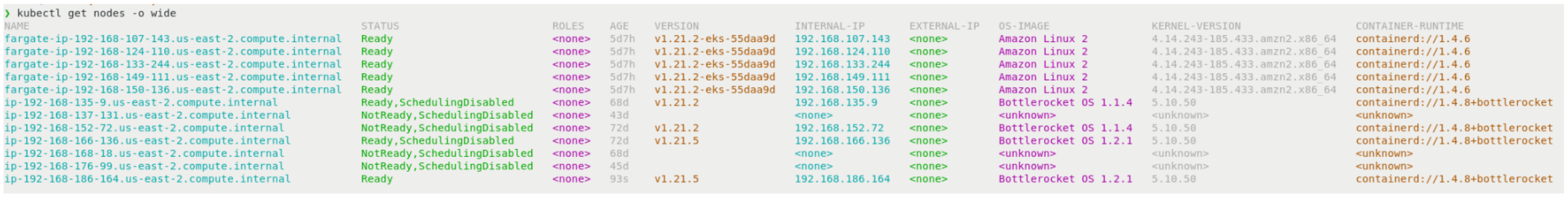
5. kubectl get events -A



```
❯ kubectl get events -A
NAMESPACE     LAST SEEN   TYPE      REASON                         OBJECT                                                     MESSAGE
default       48m         Normal    RegisteredNode                 node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal event: Registered N
ode ip-192-168-103-110.us-east-2.compute.internal in Controller
default       47m         Normal    Starting                       node/ip-192-168-103-110.us-east-2.compute.internal         Starting kubelet.
default       47m         Warning   InvalidDiskCapacity            node/ip-192-168-103-110.us-east-2.compute.internal         invalid capacity 0 on image filesystem
default       47m         Normal    NodeHasSufficientMemory        node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal status is now: Node
HasSufficientMemory
default       47m         Normal    NodeHasNoDiskPressure          node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal status is now: Node
HasNoDiskPressure
default       47m         Normal    NodeHasSufficientPID           node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal status is now: Node
HasSufficientPID
default       47m         Normal    NodeAllocatableEnforced        node/ip-192-168-103-110.us-east-2.compute.internal         Updated Node Allocatable limit across pods
default       47m         Normal    NodeNotReady                   node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal status is now: Node
NotReady
default       47m         Normal    Starting                       node/ip-192-168-103-110.us-east-2.compute.internal         Starting kube-proxy.
default       47m         Normal    NodeReady                      node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal status is now: Node
Ready
default       38m         Normal    NodeNotSchedulable             node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal status is now: Node
NotSchedulable
default       38m         Normal    RemovingNode                   node/ip-192-168-103-110.us-east-2.compute.internal         Node ip-192-168-103-110.us-east-2.compute.internal event: Removing Nod
e ip-192-168-103-110.us-east-2.compute.internal from Controller
default       28m         Normal    RegisteredNode                 node/ip-192-168-110-139.us-east-2.compute.internal         Node ip-192-168-110-139.us-east-2.compute.internal event: Registered N
ode ip-192-168-110-139.us-east-2.compute.internal in Controller
default       27m         Normal    Starting                       node/ip-192-168-110-139.us-east-2.compute.internal         Starting kubelet.
default       27m         Warning   InvalidDiskCapacity            node/ip-192-168-110-139.us-east-2.compute.internal         invalid capacity 0 on image filesystem
default       27m         Normal    NodeHasSufficientMemory        node/ip-192-168-110-139.us-east-2.compute.internal         Node ip-192-168-110-139.us-east-2.compute.internal status is now: Node
HasSufficientMemory
default       27m         Normal    NodeHasNoDiskPressure          node/ip-192-168-110-139.us-east-2.compute.internal         Node ip-192-168-110-139.us-east-2.compute.internal status is now: Node
HasNoDiskPressure
default       27m         Normal    NodeHasSufficientPID           node/ip-192-168-110-139.us-east-2.compute.internal         Node ip-192-168-110-139.us-east-2.compute.internal status is now: Node
HasSufficientPID
default       27m         Normal    NodeAllocatableEnforced        node/ip-192-168-110-139.us-east-2.compute.internal         Updated Node Allocatable limit across pods
default       27m         Normal    NodeNotReady                   node/ip-192-168-110-139.us-east-2.compute.internal         Node ip-192-168-110-139.us-east-2.compute.internal status is now: Node
NotReady
default       27m         Normal    Starting                       node/ip-192-168-110-139.us-east-2.compute.internal         Starting kube-proxy.
default       27m         Normal    NodeReady                      node/ip-192-168-110-139.us-east-2.compute.internal         Node ip-192-168-110-139.us-east-2.compute.internal status is now: Node
Ready
```

Now that we have an idea of what’s running in the cluster, we should look at what’s happening. If something broke recently, **you can look at the cluster events** to see what was happening before and after things broke.

6. kubectl get nodes -o wide

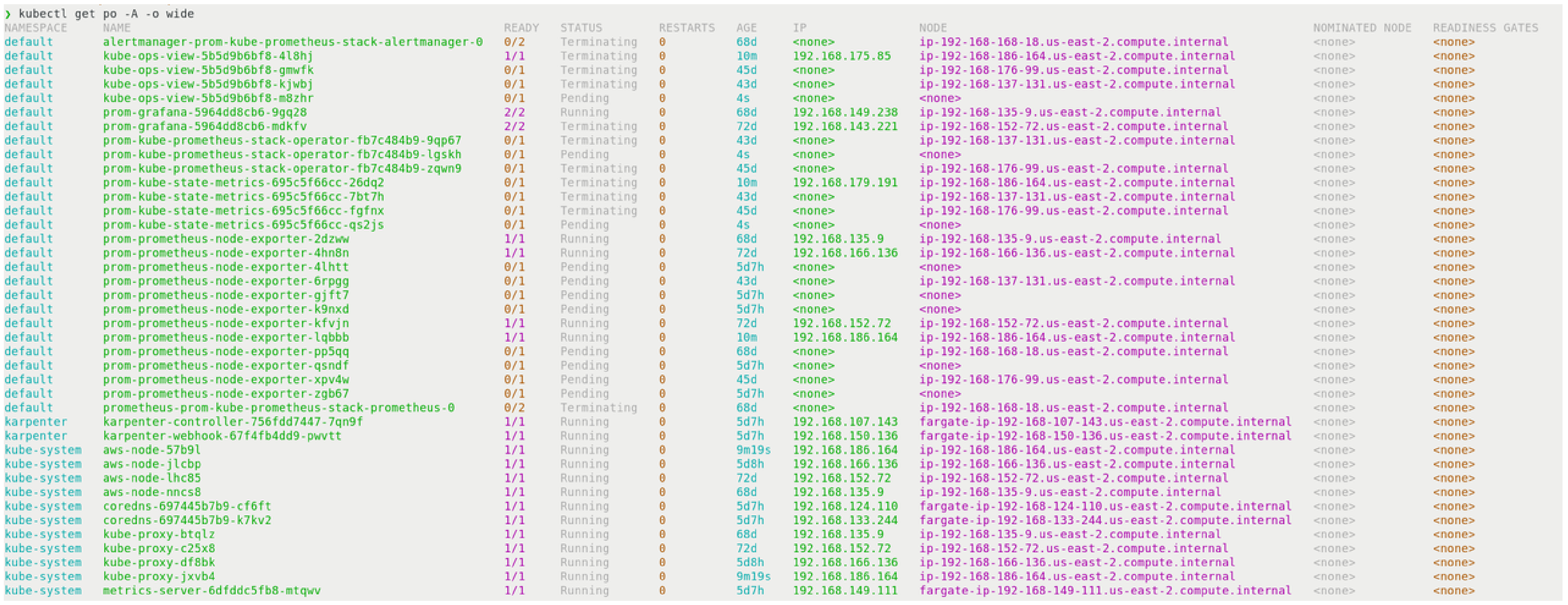


```
❯ kubectl get nodes -o wide
NAME                                                    STATUS                        ROLES    AGE    VERSION               INTERNAL-IP       EXTERNAL-IP   OS-IMAGE                KERNEL-VERSION                  CONTAINER-RUNTIME
fargate-ip-192-168-107-143.us-east-2.compute.internal   Ready                         <none>   5d7h   v1.21.2-eks-55daa9d   192.168.107.143   <none>        Amazon Linux 2          4.14.243-185.433.amzn2.x86_64   containerd://1.4.6
fargate-ip-192-168-124-110.us-east-2.compute.internal   Ready                         <none>   5d7h   v1.21.2-eks-55daa9d   192.168.124.110   <none>        Amazon Linux 2          4.14.243-185.433.amzn2.x86_64   containerd://1.4.6
fargate-ip-192-168-133-244.us-east-2.compute.internal   Ready                         <none>   5d7h   v1.21.2-eks-55daa9d   192.168.133.244   <none>        Amazon Linux 2          4.14.243-185.433.amzn2.x86_64   containerd://1.4.6
fargate-ip-192-168-149-111.us-east-2.compute.internal   Ready                         <none>   5d7h   v1.21.2-eks-55daa9d   192.168.149.111   <none>        Amazon Linux 2          4.14.243-185.433.amzn2.x86_64   containerd://1.4.6
fargate-ip-192-168-150-136.us-east-2.compute.internal   Ready                         <none>   5d7h   v1.21.2-eks-55daa9d   192.168.150.136   <none>        Amazon Linux 2          4.14.243-185.433.amzn2.x86_64   containerd://1.4.6
ip-192-168-135-9.us-east-2.compute.internal             Ready,SchedulingDisabled      <none>   68d    v1.21.2               192.168.135.9     <none>        Bottlerocket OS 1.1.4   5.10.50                         containerd://1.4.8+bottlerocket
ip-192-168-137-131.us-east-2.compute.internal           NotReady,SchedulingDisabled   <none>   43d                          <none>            <none>        <unknown>               <unknown>                       <unknown>
ip-192-168-152-72.us-east-2.compute.internal            NotReady,SchedulingDisabled   <none>   72d    v1.21.2               192.168.152.72    <none>        Bottlerocket OS 1.1.4   5.10.50                         containerd://1.4.8+bottlerocket
ip-192-168-166-136.us-east-2.compute.internal           Ready,SchedulingDisabled      <none>   72d    v1.21.5               192.168.166.136   <none>        Bottlerocket OS 1.2.1   5.10.50                         containerd://1.4.8+bottlerocket
ip-192-168-168-18.us-east-2.compute.internal            NotReady,SchedulingDisabled   <none>   68d                          <none>            <none>        <unknown>               <unknown>                       <unknown>
ip-192-168-176-99.us-east-2.compute.internal            NotReady,SchedulingDisabled   <none>   45d                          <none>            <none>        <unknown>               <unknown>                       <unknown>
ip-192-168-186-164.us-east-2.compute.internal           Ready                         <none>   93s    v1.21.5               192.168.186.164   <none>        Bottlerocket OS 1.2.1   5.10.50                         containerd://1.4.8+bottlerocket
```

Nodes are a first-class resource inside Kubernetes and are fundamental for pods to run.

Using the **-o wide** option will tell us additional details like operating system (OS), IP address, and container runtime.

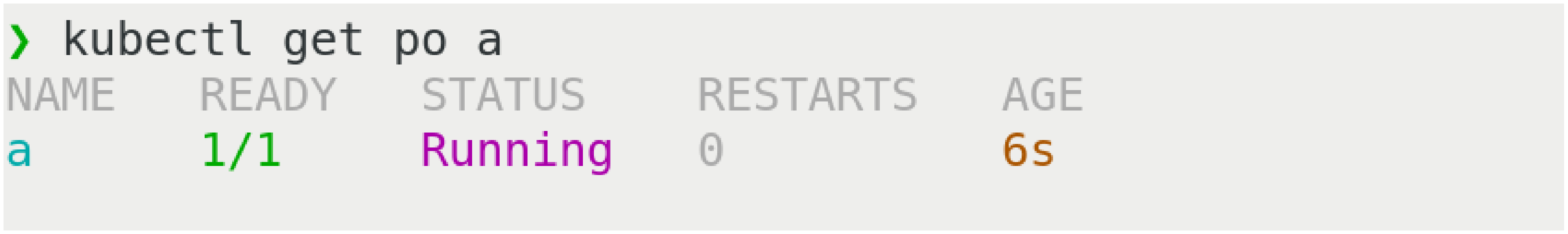
7. kubectl get pods -A -o wide



```
❯ kubectl get po -A -o wide
NAMESPACE     NAME                                                     READY   STATUS        RESTARTS   AGE     IP                NODE                                                    NOMINATED NODE   READINESS GATES
default       alertmanager-prom-kube-prometheus-stack-alertmanager-0   0/2     Terminating   0          68d     <none>            ip-192-168-168-18.us-east-2.compute.internal            <none>           <none>
default       kube-ops-view-5b5d9b6bf8-4l8hj                           1/1     Terminating   0          10m     192.168.175.85    ip-192-168-186-164.us-east-2.compute.internal           <none>           <none>
default       kube-ops-view-5b5d9b6bf8-gmwfk                           0/1     Terminating   0          45d     <none>            ip-192-168-176-99.us-east-2.compute.internal            <none>           <none>
default       kube-ops-view-5b5d9b6bf8-kjwbj                           0/1     Terminating   0          43d     <none>            ip-192-168-137-131.us-east-2.compute.internal           <none>           <none>
default       kube-ops-view-5b5d9b6bf8-m8zhr                           0/1     Pending       0          4s      <none>            <none>                                                  <none>           <none>
default       prom-grafana-5964dd8cb6-9gq28                            2/2     Running       0          68d     192.168.149.238   ip-192-168-135-9.us-east-2.compute.internal             <none>           <none>
default       prom-grafana-5964dd8cb6-mdkfv                            2/2     Terminating   0          72d     192.168.143.221   ip-192-168-152-72.us-east-2.compute.internal            <none>           <none>
default       prom-kube-prometheus-stack-operator-fb7c484b9-9qp67      0/1     Terminating   0          43d     <none>            ip-192-168-137-131.us-east-2.compute.internal           <none>           <none>
default       prom-kube-prometheus-stack-operator-fb7c484b9-lgskh      0/1     Pending       0          4s      <none>            <none>                                                  <none>           <none>
default       prom-kube-prometheus-stack-operator-fb7c484b9-zqwn9      0/1     Terminating   0          45d     <none>            ip-192-168-176-99.us-east-2.compute.internal            <none>           <none>
default       prom-kube-state-metrics-695c5f66cc-26dq2                 0/1     Terminating   0          10m     192.168.179.191   ip-192-168-186-164.us-east-2.compute.internal           <none>           <none>
default       prom-kube-state-metrics-695c5f66cc-7bt7h                 0/1     Terminating   0          43d     <none>            ip-192-168-137-131.us-east-2.compute.internal           <none>           <none>
default       prom-kube-state-metrics-695c5f66cc-fgfnx                 0/1     Terminating   0          45d     <none>            ip-192-168-176-99.us-east-2.compute.internal            <none>           <none>
default       prom-kube-state-metrics-695c5f66cc-qs2js                 0/1     Pending       0          4s      <none>            <none>                                                  <none>           <none>
default       prom-prometheus-node-exporter-2dzww                      1/1     Running       0          68d     192.168.135.9     ip-192-168-135-9.us-east-2.compute.internal             <none>           <none>
default       prom-prometheus-node-exporter-4hn8n                      1/1     Running       0          72d     192.168.166.136   ip-192-168-166-136.us-east-2.compute.internal           <none>           <none>
default       prom-prometheus-node-exporter-4lhtt                      0/1     Pending       0          5d7h    <none>            <none>                                                  <none>           <none>
default       prom-prometheus-node-exporter-6rpgg                      0/1     Pending       0          43d     <none>            ip-192-168-137-131.us-east-2.compute.internal           <none>           <none>
default       prom-prometheus-node-exporter-gjft7                      0/1     Pending       0          5d7h    <none>            <none>                                                  <none>           <none>
default       prom-prometheus-node-exporter-k9nxd                      0/1     Pending       0          5d7h    <none>            <none>                                                  <none>           <none>
default       prom-prometheus-node-exporter-kfvjn                      1/1     Running       0          72d     192.168.152.72    ip-192-168-152-72.us-east-2.compute.internal            <none>           <none>
default       prom-prometheus-node-exporter-lqbbb                      1/1     Running       0          10m     192.168.186.164   ip-192-168-186-164.us-east-2.compute.internal           <none>           <none>
default       prom-prometheus-node-exporter-pp5qq                      0/1     Pending       0          68d     <none>            ip-192-168-168-18.us-east-2.compute.internal            <none>           <none>
default       prom-prometheus-node-exporter-qsndf                      0/1     Pending       0          5d7h    <none>            <none>                                                  <none>           <none>
default       prom-prometheus-node-exporter-xpv4w                      0/1     Pending       0          45d     <none>            ip-192-168-176-99.us-east-2.compute.internal            <none>           <none>
default       prom-prometheus-node-exporter-zgb67                      0/1     Pending       0          5d7h    <none>            <none>                                                  <none>           <none>
default       prometheus-prom-kube-prometheus-stack-prometheus-0       0/2     Terminating   0          68d     <none>            ip-192-168-168-18.us-east-2.compute.internal            <none>           <none>
karpenter     karpenter-controller-756fdd7447-7qn9f                    1/1     Running       0          5d7h    192.168.107.143   fargate-ip-192-168-107-143.us-east-2.compute.internal   <none>           <none>
karpenter     karpenter-webhook-67f4fb4dd9-pwvtt                       1/1     Running       0          5d7h    192.168.150.136   fargate-ip-192-168-150-136.us-east-2.compute.internal   <none>           <none>
kube-system   aws-node-57b9l                                           1/1     Running       0          9m19s   192.168.186.164   ip-192-168-186-164.us-east-2.compute.internal           <none>           <none>
kube-system   aws-node-jlcbp                                           1/1     Running       0          5d8h    192.168.166.136   ip-192-168-166-136.us-east-2.compute.internal           <none>           <none>
kube-system   aws-node-lhc85                                           1/1     Running       0          72d     192.168.152.72    ip-192-168-152-72.us-east-2.compute.internal            <none>           <none>
kube-system   aws-node-nncs8                                           1/1     Running       0          68d     192.168.135.9     ip-192-168-135-9.us-east-2.compute.internal             <none>           <none>
kube-system   coredns-697445b7b9-cf6ft                                 1/1     Running       0          5d7h    192.168.124.110   fargate-ip-192-168-124-110.us-east-2.compute.internal   <none>           <none>
kube-system   coredns-697445b7b9-k7kv2                                 1/1     Running       0          5d7h    192.168.133.244   fargate-ip-192-168-133-244.us-east-2.compute.internal   <none>           <none>
kube-system   kube-proxy-btqlz                                         1/1     Running       0          68d     192.168.135.9     ip-192-168-135-9.us-east-2.compute.internal             <none>           <none>
kube-system   kube-proxy-c25x8                                         1/1     Running       0          72d     192.168.152.72    ip-192-168-152-72.us-east-2.compute.internal            <none>           <none>
kube-system   kube-proxy-df8bk                                         1/1     Running       0          5d8h    192.168.166.136   ip-192-168-166-136.us-east-2.compute.internal           <none>           <none>
kube-system   kube-proxy-jxvb4                                         1/1     Running       0          9m19s   192.168.186.164   ip-192-168-186-164.us-east-2.compute.internal           <none>           <none>
kube-system   metrics-server-6dfddc5fb8-mtqwv                          1/1     Running       0          5d7h    192.168.149.111   fargate-ip-192-168-149-111.us-east-2.compute.internal   <none>           <none>
```

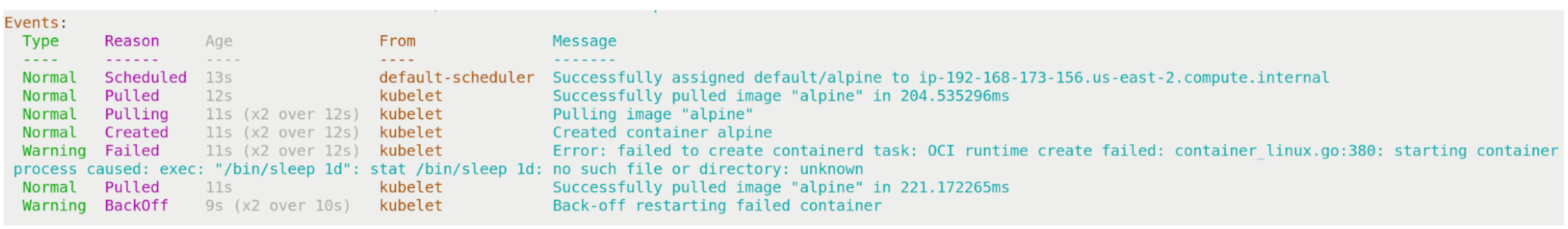
Using -A will list pods in all namespaces and -o wide will show us IP addresses, nodes, and where the pods are nominated. Using the information from listing nodes, you can look at which pods are failing on which nodes.

8. kubectl run a –image alpine –command – /bin/sleep 1d



```
❯ kubectl get po a
NAME    READY   STATUS    RESTARTS   AGE
a       1/1     Running   0          6s
```

Sometimes, the best way you can debug something is to start with the simplest example. This command doesn't have any direct output, **but you should see a running pod named “a” from it.**



```
Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  13s                default-scheduler  Successfully assigned default/alpine to ip-192-168-173-156.us-east-2.compute.internal
  Normal   Pulled     12s                kubelet            Successfully pulled image "alpine" in 204.535296ms
  Normal   Pulling    11s (x2 over 12s)  kubelet            Pulling image "alpine"
  Normal   Created    11s (x2 over 12s)  kubelet            Created container alpine
  Warning  Failed     11s (x2 over 12s)  kubelet            Error: failed to create containerd task: OCI runtime create failed: container_linux.go:380: starting container
 process caused: exec: "/bin/sleep 1d": stat /bin/sleep 1d: no such file or directory: unknown
  Normal   Pulled     11s                kubelet            Successfully pulled image "alpine" in 221.172265ms
  Warning  BackOff    9s (x2 over 10s)   kubelet            Back-off restarting failed container
```

If for some reason you don't see a running pod from this command, then using **kubectl describe po a** is your next-best option.

Look at the events to find errors for what might have gone wrong.

Credit: Justin Garrison, The new stack