Sequential Breakdown and Verification of the Update Process

In this lesson, we will go through the sequential breakdown of the cluster update process and verify the update to the cluster.

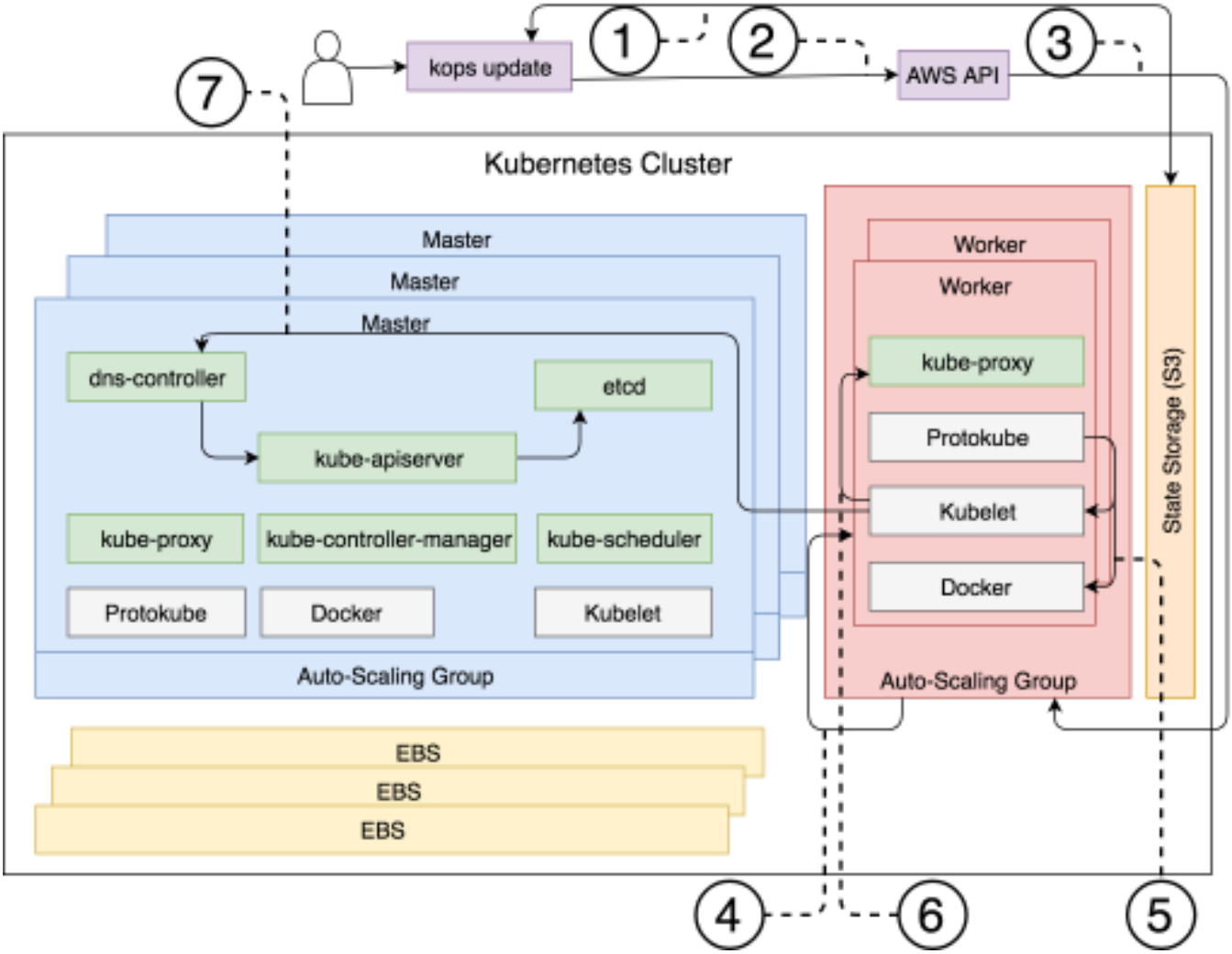
WE'LL COVER THE FOLLOWING

Sequential Breakdown of the Update Process

Let's see what happened when we executed the `kops update` command.

1. Kops retrieved the desired state from the S3 bucket.
2. Kops sent requests to AWS API to change the values of the workers ASG.
3. AWS modified the values of the workers ASG by increasing them by 1.
4. ASG created a new EC2 instance to comply with the new sizing.
5. Protokube installed Kubelet and Docker and created the manifest file with the list of Pods.
6. Kubelet read the manifest file and run the container that forms the `kube-proxy` Pod (the only Pod on the worker nodes).
7. Kubelet sent a request to the `kube-apiserver` (through the `dns-controller`) to register the new node and join it to the cluster. The information about the new node is stored in `etcd`.

This process is almost identical to the one used to create the nodes of the cluster.



The process behind the `kops update` command

The above illustration shows the sequence of the steps involved in the cluster update process.

Verification of the Update Process

Unless you are a very fast reader, ASG already created a new EC2 instance, and Kubelet joined it to the cluster. We can confirm that through the `kops validate` command.

```
kops validate cluster
```

The **output** is as follows.

```
Validating cluster devops23.k8s.local

INSTANCE GROUPS
NAME                ROLE    MACHINETYPE MIN MAX SUBNETS
master-us-east-2a Master t2.small     1   1   us-east-2a
```

```
master-us-east-2a Master t2.small    1   1   us-east-2a
master-us-east-2b Master t2.small    1   1   us-east-2b
master-us-east-2c Master t2.small    1   1   us-east-2c
nodes             Node   t2.small    2   2   us-east-2a,us-east-2b,us-east-2c

NODE STATUS
NAME                  ROLE   READY
ip-172-20-120-133... master True
ip-172-20-33-237...  node   True
ip-172-20-34-249...  master True
ip-172-20-65-28...   master True
ip-172-20-95-101...  node   True

Your cluster devops23.k8s.local is ready
```

We can see that now we have two nodes (there was one before) and that they are located somewhere inside the three `us-east-2` availability zones.

Similarly, we can use `kubectl` to confirm that Kubernetes indeed added the new worker node to the cluster.

```
kubectl get nodes
```

The **output** is as follows.

```
NAME                  STATUS ROLES  AGE VERSION
ip-172-20-120-133... Ready  master 13m v1.9.1
ip-172-20-33-237...  Ready  node   1m  v1.9.1
ip-172-20-34-249...  Ready  master 13m v1.9.1
ip-172-20-65-28...   Ready  master 13m v1.9.1
ip-172-20-95-101...  Ready  node   12m v1.9.1
```

That was easy, wasn't it? From now on, we can effortlessly add or remove nodes.

How about upgrading the cluster?

In the next lesson, we will go through the manual upgrading process of the cluster.