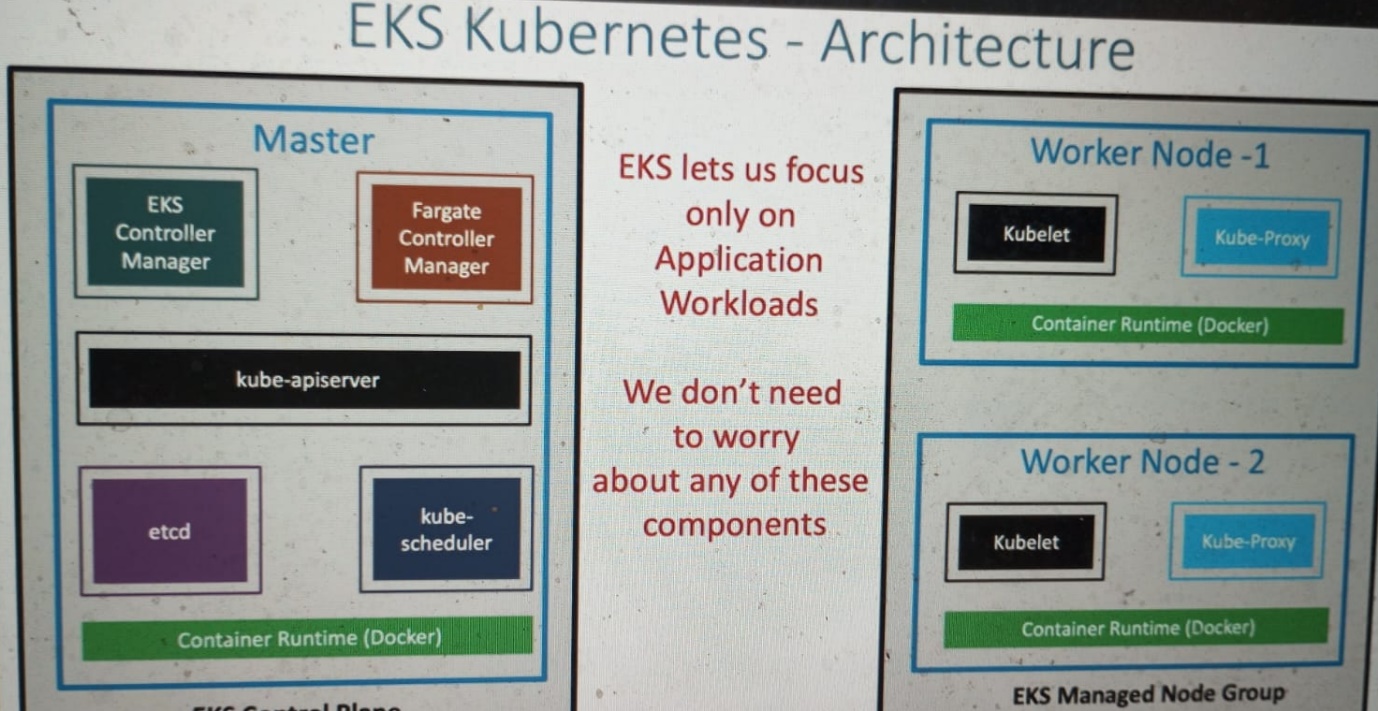
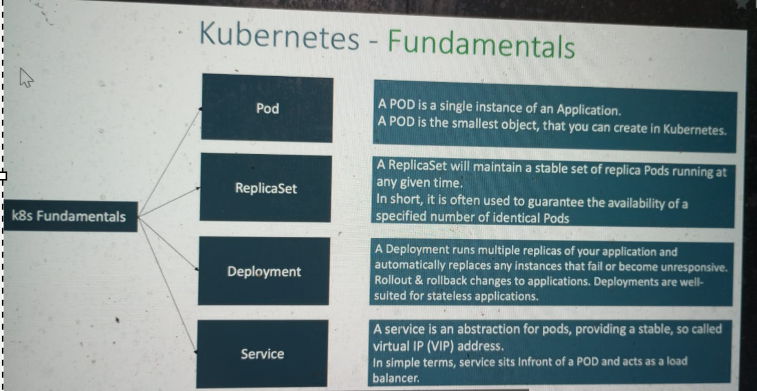
Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service provided by Amazon Web Services (AWS). It simplifies the process of deploying, managing, and scaling containerized applications using Kubernetes on AWS infrastructure. Below are key aspects and steps related to Amazon EKS:

**Key Concepts:**

1. **Cluster:** An EKS cluster is a managed Kubernetes cluster that consists of Amazon EC2 instances and runs Kubernetes control plane components.
2. **Node Group:** A node group is a set of Amazon EC2 instances that host your Kubernetes pods. EKS manages the deployment and scaling of node groups.
3. **Worker Node:** An EC2 instance that runs your Kubernetes pods. Worker nodes are part of a node group.
4. **Control Plane:** The Kubernetes control plane manages the cluster and is responsible for making global decisions about the cluster (e.g., scheduling), as well as detecting and responding to cluster events.





**Steps to Set Up Amazon EKS:**

**1. Create an EKS Cluster:**

* Use the AWS Management Console, AWS CLI, or an AWS SDK to create an EKS cluster.
* Specify details like the cluster name, VPC configuration, and the version of Kubernetes.

**2. Configure kubectl for EKS:**

* After creating the cluster, you need to configure kubectl to communicate with your EKS cluster.
* Run the following command to update your kubectl configuration:

bash

* aws eks --region <region> update-kubeconfig --name <cluster-name>

**3. Node Group:**

* Create one or more node groups to run your containerized applications.
* Configure the node group with details such as instance type, desired capacity, and key pair.

**4. Deploy Applications:**

* Use kubectl to deploy and manage your applications on the EKS cluster.
* Create Kubernetes deployment and service files or use Helm charts to deploy applications.

**5. Monitoring and Logging:**

* Set up monitoring and logging for your EKS cluster using AWS CloudWatch and AWS CloudTrail.
* Monitor metrics, logs, and events to ensure the health and performance of your applications.

**6. Scaling:**

* Configure Auto Scaling for your node groups to automatically adjust the capacity based on demand.
* Use Horizontal Pod Autoscaling (HPA) within Kubernetes to scale individual applications.

**7. Security:**

* Implement security best practices for your EKS cluster, including IAM roles, network policies, and pod security policies.
* Consider using AWS Identity and Access Management (IAM) roles for service accounts (IRSA) for fine-grained access control.

**8. Updates and Maintenance:**

* Regularly update your EKS cluster to the latest Kubernetes version and apply security patches.
* Monitor the AWS EKS release notes for updates and improvements.

**9. Integrate with Other AWS Services:**

* Integrate your EKS cluster with other AWS services such as Amazon RDS, Amazon S3, and AWS Identity and Access Management (IAM) roles.

**Additional Considerations:**

* **Networking:** Understand and configure VPC networking for your EKS cluster. EKS requires specific VPC configurations.
* **Storage:** Configure persistent storage for your applications using Amazon EBS volumes or other storage solutions.
* **IAM Roles:** Define IAM roles for your worker nodes and services running on the cluster.

By following these steps and considerations, you can effectively set up and manage your applications on Amazon EKS. Always refer to the official AWS documentation for the most up-to-date information and best practices.

EKS by Nana:

AWS EKS is a Managed Kubernetes Service from Amazon, which means AWS manages the Master Nodes for you. All the necessary applications/services are already pre-installed like the container runtime or master processes and in addition it also takes care of scaling and backups. 👍 You only create the Worker Nodes.

To create a K8s cluster in EKS you need to do following steps:

1) Setup or preparation steps - create AWS account - create a VPC - virtual private space - create an IAM role with Security Group (or in other words: create AWS user with list of permissions)

2) Create Cluster Control Plane - Master Nodes - choose basic information like cluster name and k8s version - choose region and VPC for your cluster - set security

3) Create Worker Nodes and connect to cluster The Worker Nodes are some EC2 instances with CPU and storage resources. - Create as a Node Group - Choose cluster it will attach to - Define Security Group, select instance type etc. With NodeGroup you have autoscaling, which means based on your needs depending on how much load the cluster has new Worker Nodes will automatically be added or removed in the cluster.

EKS control tool: Amazon EKS simplifies Kubernetes cluster management. Use AWS CLI to create, configure, and deploy containerized applications seamlessly.

Creating an Amazon EKS cluster involves several steps. Here's a simplified, step-by-step process:

1. **AWS CLI Setup:**
   * Install and configure the AWS CLI on your local machine.
2. **Create a VPC:**
   * Create a Virtual Private Cloud (VPC) to isolate your EKS resources.
3. **Create an IAM Role:**
   * Create an IAM role with the necessary permissions for EKS.
4. **Create the EKS Cluster:**
   * Use the AWS CLI to create the EKS cluster, specifying details like cluster name, version, and VPC configuration.

bash

* aws eks create-cluster --name <cluster-name> --role-arn <role-arn> --resources-vpc-config subnetIds=<subnet-ids>,securityGroupIds=<security-group-ids>

1. **Configure kubectl:**

* Update your kubectl configuration to connect to the newly created EKS cluster.

bash

* aws eks --region <region> update-kubeconfig --name <cluster-name>

1. **Create Worker Node IAM Role:**

* Create an IAM role for worker nodes with the Amazon EKS-optimized Amazon Machine Image (AMI) policies.

1. **Create Node Group:**

* Create a node group to deploy worker nodes into your EKS cluster.
* eksctl create nodegroup --cluster <cluster-name> --region <region> --name <node-group-name> --node-type <instance-type> --nodes <num-nodes>

1. **Verify Cluster Status:**

* Ensure that your EKS cluster and node group are active and running.
* kubectl get nodes

1. **Deploy Sample Application:**

* Deploy a sample application to verify the cluster setup.
* kubectl apply -f https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/deployment.yaml

1. **Clean Up (Optional):**

* If needed, delete the EKS cluster and associated resources.

Always refer to the official [Amazon EKS documentation](https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html) for detailed and up-to-date instructions. Additionally, be mindful of security considerations, IAM roles, and networking configurations during the setup process.