What is ECR?

Ans :

ECR (Elastic Container Registry) Amazon Elastic Container Registry (ECR) is a fully-managed [Docker](https://aws.amazon.com/docker/) container registry that makes it easy for developers to store, manage, and deploy Docker container images. Amazon ECR is integrated with [Amazon Elastic Container Service (ECS)](https://aws.amazon.com/ecs/), simplifying your development to production workflow. Amazon ECR eliminates the need to operate your own container repositories or worry about scaling the underlying infrastructure. Amazon ECR hosts your images in a highly available and scalable architecture, allowing you to reliably deploy containers for your applications. Integration with AWS Identity and Access Management (IAM) provides resource-level control of each repository. With Amazon ECR, there are no upfront fees or commitments. You pay only for the amount of data you store in your repositories and data transferred to the Internet.

2) List differences between AWS ECS fargate and ECS EC2

Ans :

**Fargate**

AWS Fargate manages the task execution. No EC2 instances to manage anymore. You pay for running tasks. That’s it. As easy as it sounds.

Each task that runs in Fargate comes with a dedicated Elastic Network Interface (ENI) with a private IP address. All containers of the same task can communicate with each other via localhost. Inbound and outbound task communication goes through the ENI. A public IP address can be enabled as well.

EC2 : An ECS container instance is nothing more than an EC2 instance that runs the [ECS Container Agent](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_agent.html). The EC2 instance is owned and managed by you. The instance appears in the list of EC2 instances like any other EC2 instance. The ECS Container Agent regularly polls the ECS API if new containers need to be started or stopped. Usually, you run a cluster of container instances in an auto-scaling group. ECS is free of charge. You only pay for the EC2 instances. The downside is that you have to scale, monitor, patch, and secure the EC2 instances yourself. Especially the scaling is not easy because

* There is no obvious metric to scale the cluster and no integration to scale when the task placement fails because of insufficient capacity.
* The auto-scaling group and ECS are not aware of each other which makes task deployments very hard during cluster scale in or rolling updates via CloudFormation.
* You have to [scale down without killing running tasks](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/container-instance-draining.html) which is an even more significant challenge for long lived tasks.

Even the [AWS reference architecture](https://github.com/aws-samples/ecs-refarch-cloudformation) does not include auto-scaling for the cluster. Check out our [reference architecture with auto-scaling](https://templates.cloudonaut.io/en/stable/ecs/) if you are interested.

### Comparison

|  | **ECS container instance** | **Fargate** |
| --- | --- | --- |
| Pricing | per running EC2 instance | per running task |
| Operational effort | high | low |
| EFS integration | hacky but possible [1](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/using_efs.html) [2](https://garbe.io/blog/2018/09/12/the-easiest-way-to-use-efs-volumes-with-ecs/) [3](https://aws.amazon.com/blogs/compute/amazon-ecs-and-docker-volume-drivers-amazon-efs/) | no |
| EBS integration | hacky but possible [1](https://aws.amazon.com/blogs/compute/amazon-ecs-and-docker-volume-drivers-amazon-ebs/) | no |
| Networking options | [multiple](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.html#network_mode) | ENI per task |

3) What is HashMap algorithm ?

Ans :

Grafana is open source visualization and analytics software. It allows you to query, visualize, alert on, and explore your metrics no matter where they are stored. In plain English, it provides you with tools to turn your time-series database (TSDB) data into beautiful graphs and visualizations.

After creating a dashboard like you do in [Getting started](https://grafana.com/docs/grafana/latest/guides/getting_started/), there are many possible things you might do next. It all depends on your needs and your use case.

For example, if you want to view weather data and statistics about your smart home, then you might create a playlist. If you are the administrator for a corporation and are managing Grafana for multiple teams, then you might need to set up provisioning and authentication.

The following sections provide an overview of things you might want to do with your Grafana database and links so you can learn more. For more guidance and ideas, check out the [Grafana Community forums](https://community.grafana.com/).

## **Explore metrics and logs**

Explore your data through ad-hoc queries and dynamic drilldown. Split view and compare different time ranges, queries and data sources side by side.

Refer to [Explore](https://grafana.com/docs/grafana/latest/features/explore/) for more information.

## **Alerts**

If you’re using Grafana alerting, then you can have alerts sent through a number of different [alert notifiers](https://grafana.com/docs/grafana/latest/alerting/notifications/), including PagerDuty, SMS, email, VictorOps, OpsGenie, or Slack.

Alert hooks allow you to create different notifiers with a bit of code if you prefer some other channels of communication. Visually define [alert rules](https://grafana.com/docs/grafana/latest/alerting/rules/) for your most important metrics.

## **Annotations**

Annotate graphs with rich events from different data sources. Hover over events to see the full event metadata and tags.

This feature, which shows up as a graph marker in Grafana, is useful for correlating data in case something goes wrong. You can create the annotations manually—just control-click on a graph and input some text—or you can fetch data from any data source.

Refer to [Annotations](https://grafana.com/docs/grafana/latest/reference/annotations/) for more information.

## **Dashboard variables**

[Template variables](https://grafana.com/docs/grafana/latest/reference/templating/) allow you to create dashboards that can be reused for lots of different use cases. Values aren’t hard-coded with these templates, so for instance, if you have a production server and a test server, you can use the same dashboard for both.

Templating allows you to drill down into your data, say, from all data to North America data, down to Texas data, and beyond. You can also share these dashboards across teams within your organization—or if you create a great dashboard template for a popular data source, you can contribute it to the whole community to customize and use.

## **Configure Grafana**

If you’re a Grafana administrator, then you’ll want to thoroughly familiarize yourself with [Grafana configuration options](https://grafana.com/docs/grafana/latest/installation/configuration/) and the [Grafana CLI](https://grafana.com/docs/grafana/latest/administration/cli/).

Configuration covers both config files and environment variables. You can set up default ports, logging levels, email IP addresses, security, and more.

## **Import dashboards and plugins**

Discover hundreds of [dashboards](https://grafana.com/grafana/dashboards) and [plugins](https://grafana.com/grafana/plugins) in the official library. Thanks to the passion and momentum of community members, new ones are added every week.

## **Authentication**

Grafana supports different authentication methods, such as LDAP and OAuth, and allows you to map users to organizations. Refer to the [User authentication overview](https://grafana.com/docs/grafana/latest/auth/overview/) for more information.

In Grafana Enterprise, you can also map users to teams: If your company has its own authentication system, Grafana allows you to map the teams in your internal systems to teams in Grafana. That way, you can automatically give people access to the dashboards designated for their teams.

Refer to [Grafana Enterprise](https://grafana.com/docs/grafana/latest/enterprise/) for more information.

## **Provisioning**

While it’s easy to click, drag, and drop to create a single dashboard, power users in need of many dashboards will want to automate the setup with a script. You can script anything in Grafana.

For example, if you’re spinning up a new Kubernetes cluster, you can also spin up a Grafana automatically with a script that would have the right server, IP address, and data sources preset and locked in so users cannot change them. It’s also a way of getting control over a lot of dashboards.

Refer to [Provisioning](https://grafana.com/docs/grafana/latest/administration/provisioning/) for more information.

## **Permissions**

When organizations have one Grafana and multiple teams, they often want the ability to both keep things separate and share dashboards. You can create a team of users and then set [permissions](https://grafana.com/docs/grafana/latest/permissions/overview/) on folders, dashboards, and down to the [data source level](https://grafana.com/docs/grafana/latest/enterprise/datasource_permissions/) if you’re using [Grafana Enterprise](https://grafana.com/docs/grafana/latest/enterprise/).

## **Grafana Cloud**

Grafana Cloud is a highly available, fast, fully managed OpenSaaS logging and metrics platform. Everything you love about Grafana, but Grafana Labs hosts it for you and handles all the headaches.

[Learn more about Grafana Cloud.](https://grafana.com/cloud/)

## **Grafana Enterprise**

[Grafana Enterprise](https://grafana.com/docs/grafana/latest/enterprise/) is a commercial edition of Grafana that includes additional features not found in the open source version.

Building on everything you already know and love about Grafana, Grafana Enterprise adds enterprise data sources, advanced authentication options, more permission controls, 24x7x365 support, and training from the core Grafana team.

[Learn more about Grafana Enterprise](https://grafana.com/enterprise). To purchase Enterprise or obtain a trial license, contact the Grafana Labs [Sales Team](https://grafana.com/contact?about=support&topic=Grafana%20Enterprise).

4) What is Kibana?

Ans :

Kibana is an open-source data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers powerful and easy-to-use features such as histograms, line graphs, pie charts, heat maps, and built-in geospatial support. Also, it provides tight integration with [Elasticsearch](https://aws.amazon.com/elasticsearch-service/the-elk-stack/what-is-elasticsearch/), a popular analytics and search engine, which makes Kibana the default choice for visualizing data stored in Elasticsearch.

## **Is Kibana free to use?**

Yes, Kibana is a free, open-source visualization tool. You can run Kibana on-premises, on Amazon EC2, or on [Amazon Elasticsearch Service](https://aws.amazon.com/elasticsearch-service/). With on-premises or Amazon EC2 deployments, you are responsible for provisioning the infrastructure, installing Kibana software, and managing the cluster. With Amazon Elasticsearch Service, Kibana is deployed automatically with your domain as a fully managed service, automatically taking care of all the heavy-lifting to manage the cluster.