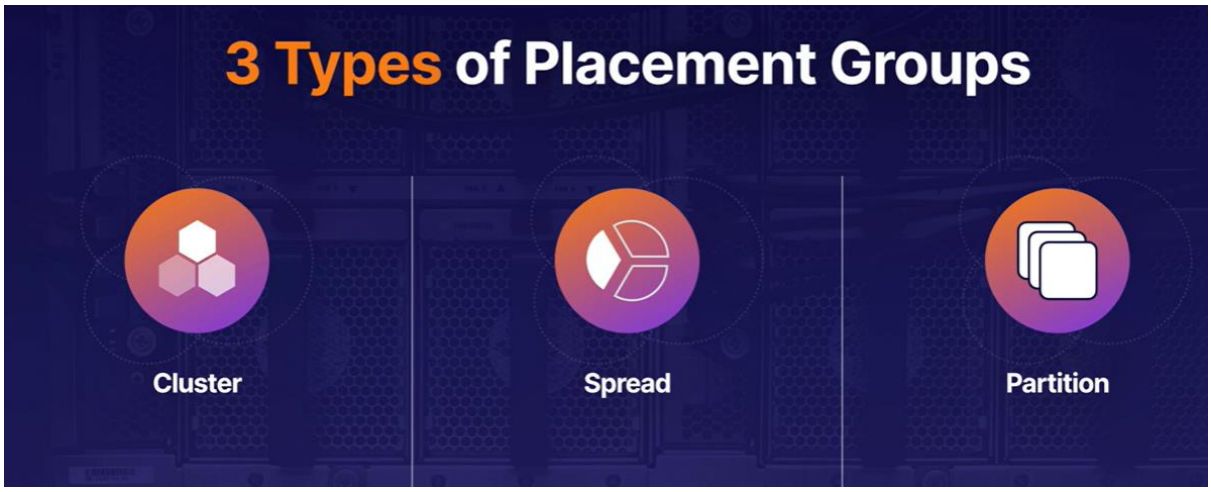
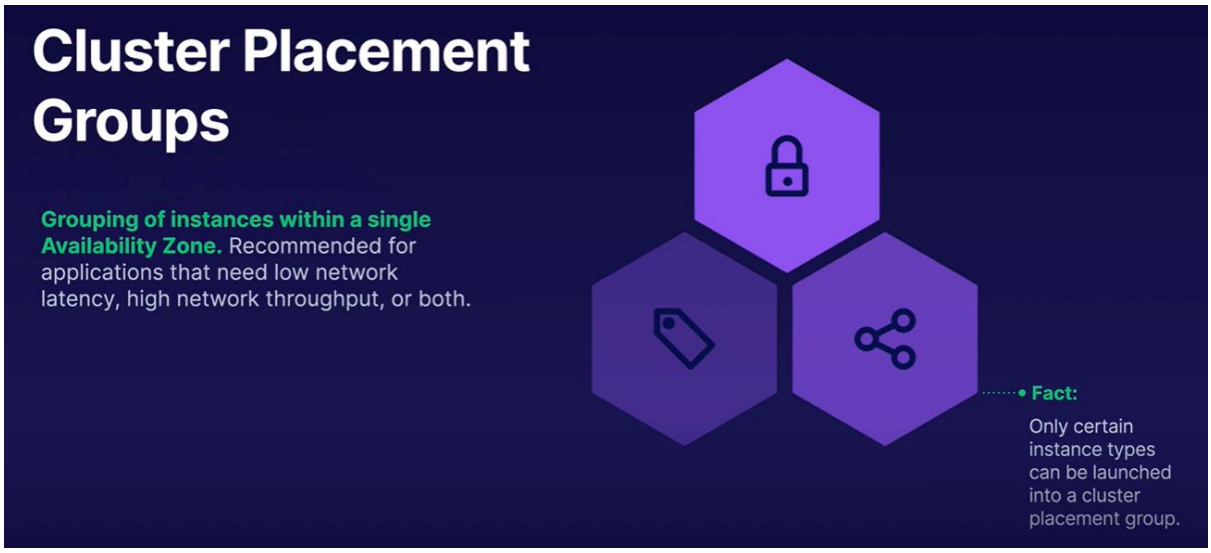
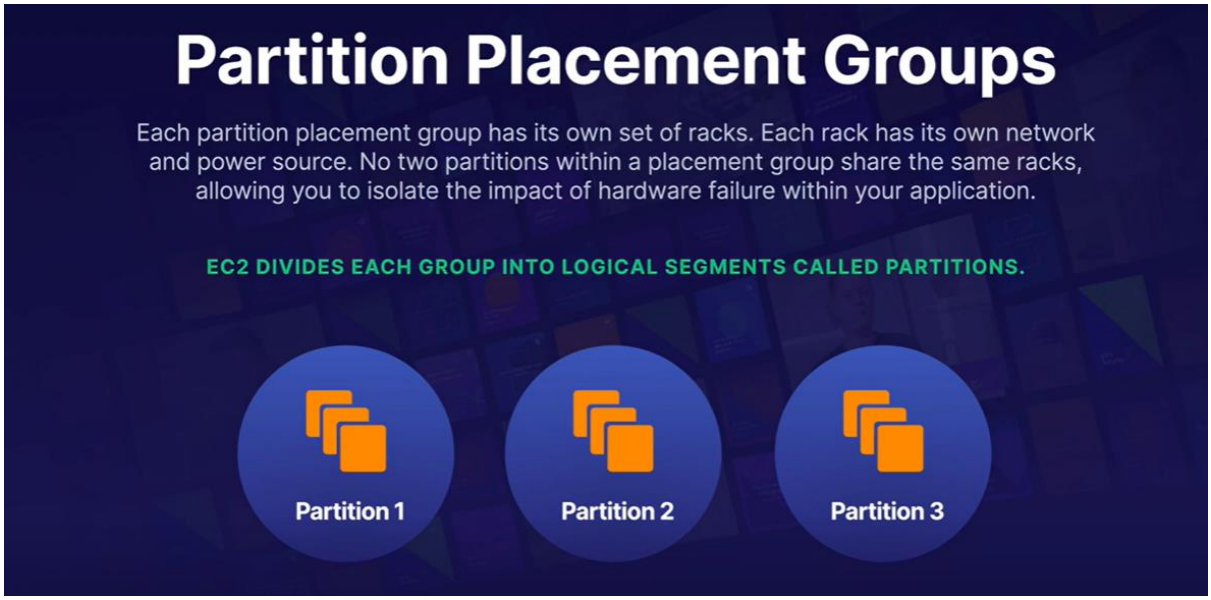
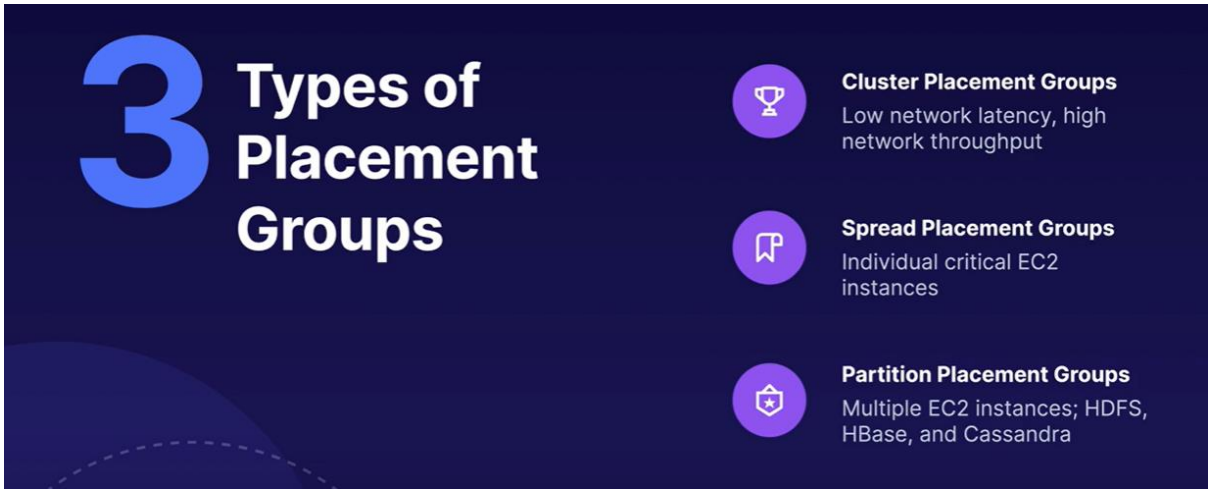
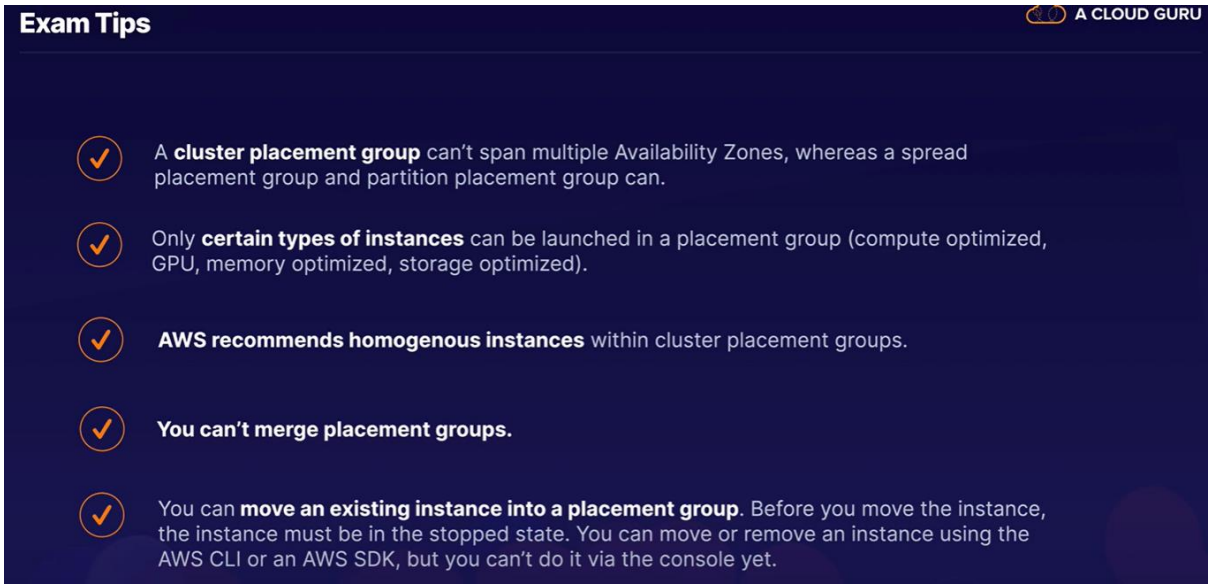
Optimising with EC2 Placement Groups:



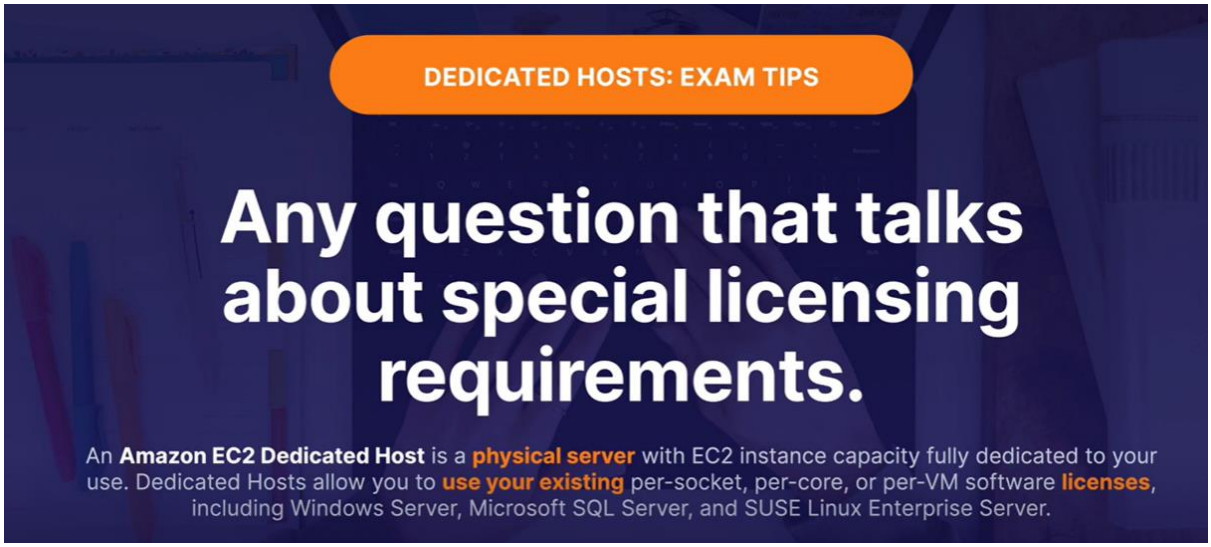




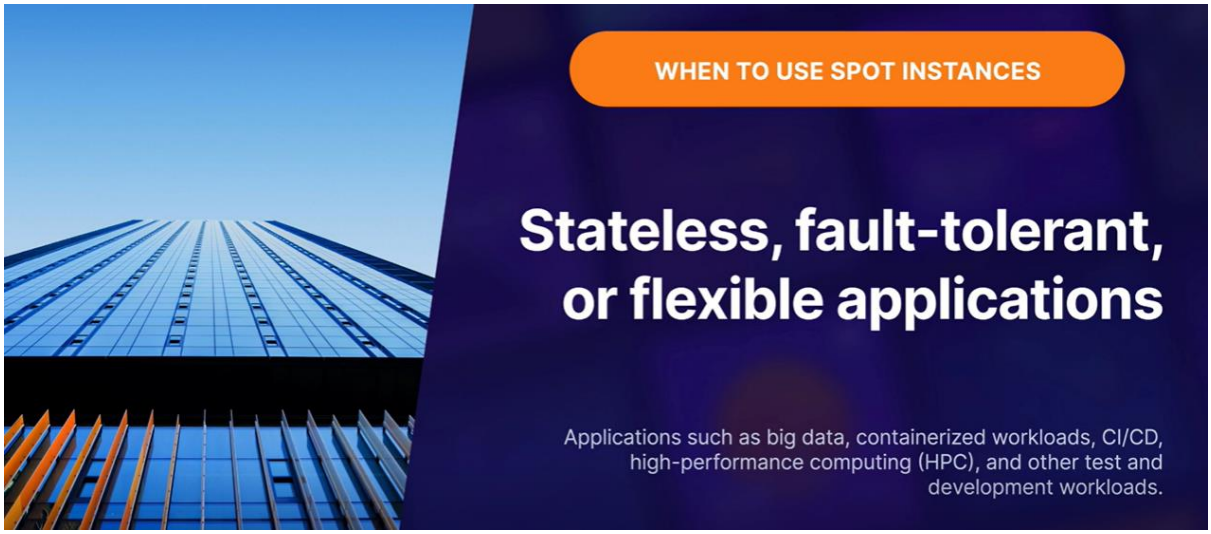


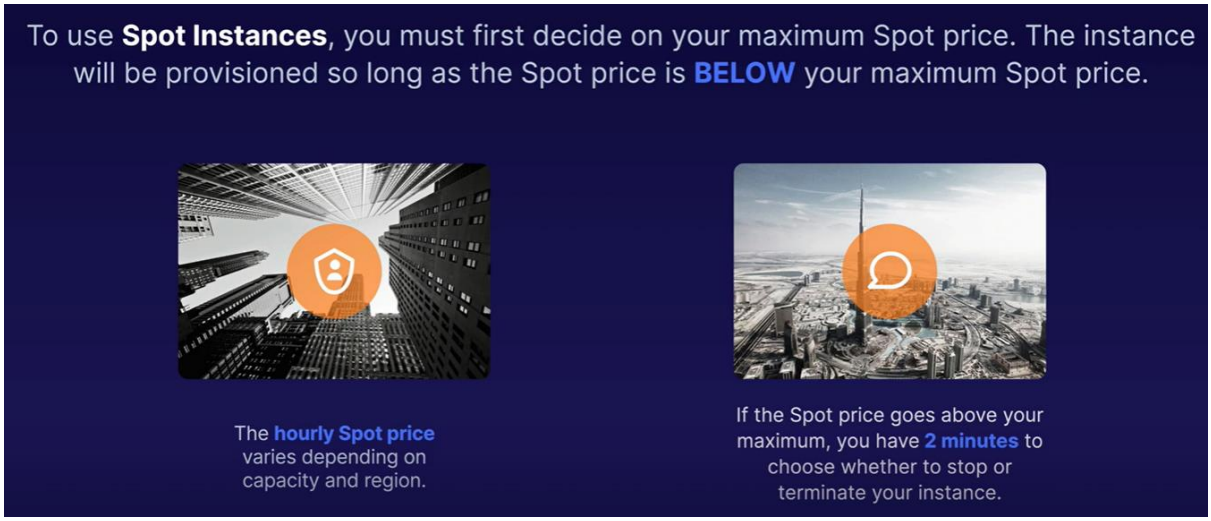


Solving Licensing Issues with Dedicated Hosts:



Timing Workloads with Spot Instances:





You may also use a **Spot block** to stop your Spot Instances from being terminated even if the Spot price goes over your max Spot price. You can set Spot blocks for between **1 to 6 hours** currently.

Spot Instances are useful for the following tasks:

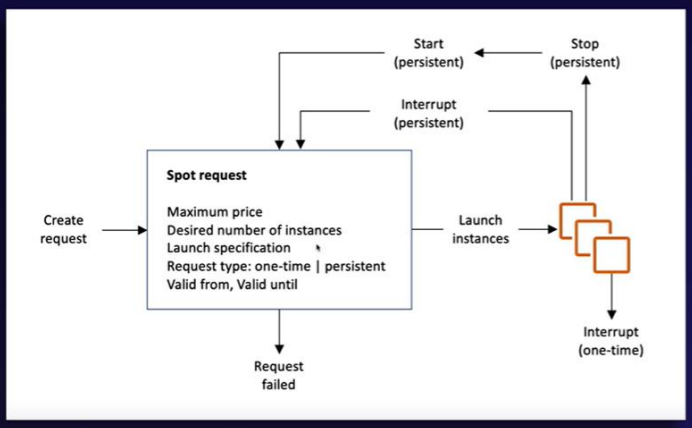
- Big data and analytics
- Containerized workloads
- CI/CD and testing
- Image and media rendering
- High-performance computing

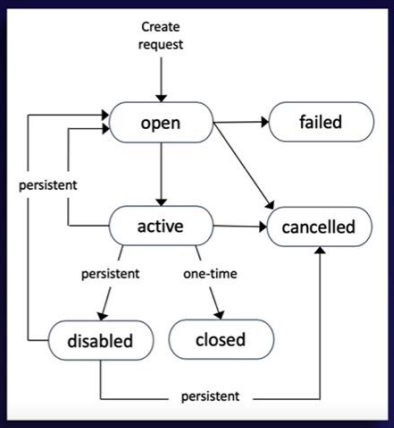
Spot Instances are not good for:

- ✓ Persistent workloads
- ✓ Critical jobs
- ✓ Databases

Terminating Spot Instances

A CLOUD GURU



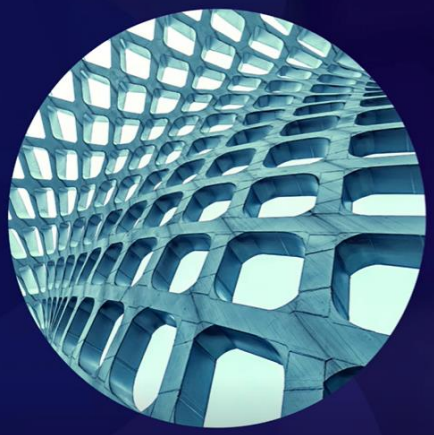


Spot Fleets

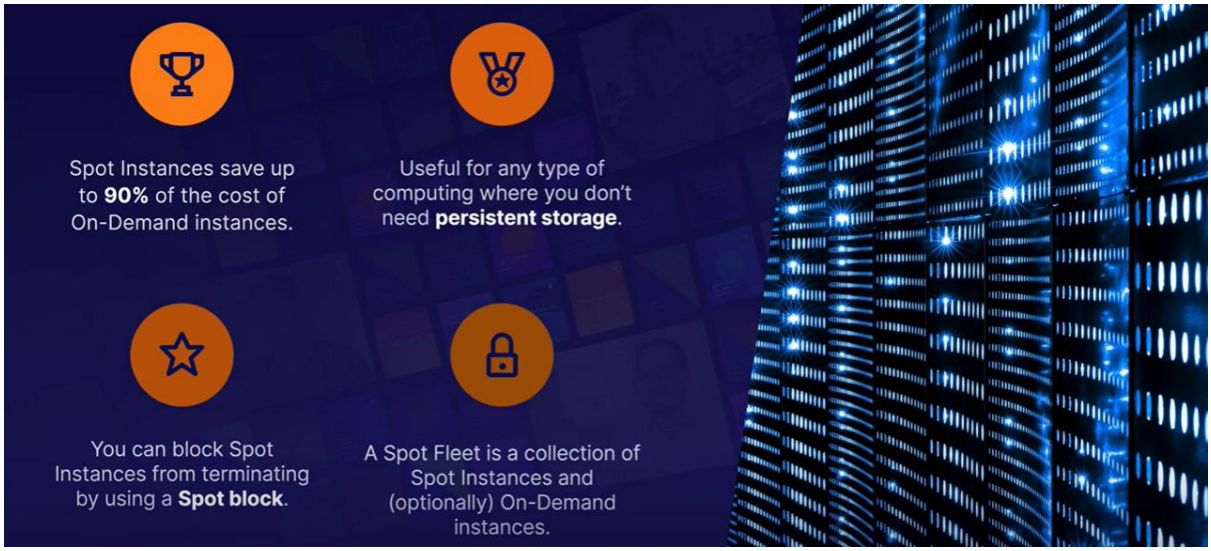
A Spot Fleet is a collection of Spot Instances and (optionally) On-Demand Instances.

The **Spot Fleet** attempts to launch the number of Spot instances and On-Demand instances to meet the target capacity you specified in the Spot Fleet request. The request for Spot Instances is fulfilled if there is available capacity and the **maximum price you specified in the request exceeds the current Spot price**. The Spot Fleet also attempts to maintain its target capacity fleet if your Spot Instances are interrupted.

Spot Fleets will try and match the target capacity with your price restraints.



1. Set up different launch pools. Define things like **EC2** instance type, operating system, and Availability Zone.
2. You can have **multiple** pools, and the fleet will choose the best way to implement depending on the strategy you define.
3. Spot fleets will **stop launching instances** once you reach your price threshold or capacity desire.


Spot Instances save up to **90%** of the cost of On-Demand instances.
Useful for any type of computing where you don't need **persistent storage**.
You can block Spot Instances from terminating by using a **Spot block**.
A Spot Fleet is a collection of Spot Instances and (optionally) On-Demand instances.