

# روز اول

## رشته: پردازش ابری

## مقدمه

**ماژول:** پیکربندی اولیه سرویس‌های زیرساخت ابری

**مدت زمان:** ۴ ساعت و ۳۰ دقیقه

هدف این ماژول راه‌اندازی و پیکربندی سرویس‌های اولیه بر بستر زیرساخت ابری می‌باشد. در این پروژه شما می‌بایست نیازمندی‌های لازم برای ایجاد و کارکرد دو سرور ابری و پیکربندی HTTP Server بر روی آنها جهت سرویس‌دهی را انجام دهید. همچنین پس از پیکربندی درست و اطمینان از کارکرد این وب‌سرویس‌ها، LoadBalancing را جهت مدیریت بار و ترافیک بر روی این دو سرور ابری تنظیم نمایید.

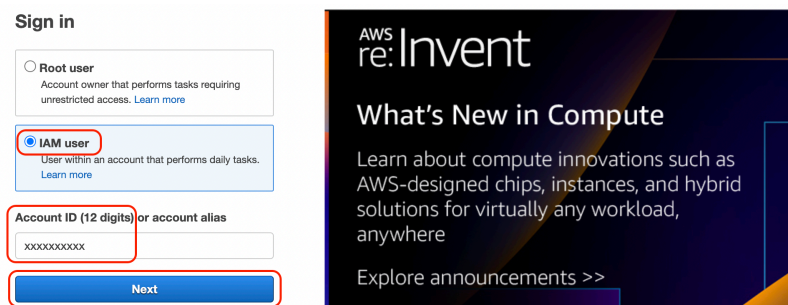
## قوانین و استانداردها

- اطلاعات اولیه

○ **اطلاعات ورود:** console.aws.amazon.com/console/home

○ **نوع ورود:** IAM User

○ **اکانت‌آیدی:** xxxxxxxx



○ **نام کاربری IAM:** ( )

○ **رمز عبور:** ( )

○ **نام ریجن:** ( )

○ **شناسه رقابت‌کننده:** ( )

- برای این مسابقه شما فقط و فقط باید از Region مخصوص خود استفاده کنید.

- اینترنتی که به شما اختصاص داده شده است فقط برای کارکرد و استفاده از پل amazon aws می‌باشد و هرگونه

استفاده دیگر از آن (جست‌جو، بررسی داکيومنتیشن و ...) غیر مجاز می‌باشد.

- برخی از آیتم‌های سیستم نمره‌دهی کاملاً بر اساس ساختار اسمی مشخص شده تصحیح خواهند شد، پس دقت لازم را

به خرج دهید تا تمامی منابع ابری که در طی انجام این ماژول ایجاد می‌کنید هیچ‌کدام بی‌نام نباشند و نام آنها طبق

استاندارد و ساختار اسمی زیر باشد:

○ [شماره]\_[نام]\_[نام سرویس‌آمازون]\_[شناسه رقابت‌کننده]

▪ مثال: comp\_20\_ec2\_myapp\_1 ، comp\_20\_ksk\_kafkasrv\_2

## سرویس Elastic Cloud Compute (EC2)

### پنل زیرساخت ابری

داخل Region خود ۲ سرور ابری با مشخصات زیر را ایجاد کنید:

- نام سرور: webserv (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- هاست نیم در سیستم عامل: webserv (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- ایمج سیستم عامل: Linux Ubuntu 20.04 (free tier eligible)
- پکیج منابع: t3.micro
- فضای دیسک سیستم عامل: 8 GB
- نوع دیسک: General Purpose SSD (gp2)
- آی پی پابلیک: داشته باشد
- نحوه دسترسی به سرور: کلید خصوصی

دسترسی عمومی برای تمامی سرویس‌های استفاده شده در این سناریو ایجاد شود.

- استفاده از Allow All Traffic و موارد مشابه مجاز نمی‌باشد.

شما باید طبق نیازمندی و سرویس‌های استفاده شده در پروژه، تنظیمات لازم را برای VPC و Security Group خود انجام دهید.

یک Block Storage (Volume) با فضای ۵ گیگابایت ایجاد کنید.

- این فضا را به سرور webserv\_1 متصل کنید.

### پی‌کر بندی سرور ابری

سرورهای webserv\_1 و webserv\_2

- بر روی سرورهای ابری ساخته شده، سرویس apache را نصب کنید:
  - محتوای وب‌سایت پیش فرض بر روی این سرورها باید متن "Welcome to National Skills 2021 - [Time&Date]" را نشان دهد.
  - باید هدر X-Served-By با مقدار hostname هنگام serve محتوا، ست شود.
- بر روی سرورهای ابری ساخته شده، docker را با کمک اطلاعات زیر نصب کنید:
  - **کلید GPG**
  - <https://download.docker.com/linux/ubuntu/gpg>
  - **پارامترهای Repository**
  - `deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \`  
`https://download.docker.com/linux/ubuntu focal stable`

## سرور 1 webserv\_1

- فضای اضافی متصل شده به سرور را به مسیر `/mnt/blk_gp2_5g` متصل کنید.
  - o این اتصال باید در `fstab` بصورت `persistant` پیگیربندی شود تا پس از ری‌استارت شدن سرور بصورت خودکار `mount` شود.
  - o فایل `index.html` را با محتوای `"This is additional_webservice"` در این مسیر قرار دهید.
- پکیج `docker-compose` را نصب کنید.
- یک فایل `docker-compose` با مشخصات زیر بنویسید و آن را در مسیر `/opt/container/docker-compose.yaml` قرار دهید:
  - o **نام کانتینر:** `additional_webservice`
  - o **ایمیج:** `nginx:alpine`
  - o **شماره پورت:** `8080`
  - o **دایرکتوری:** مسیر `/mnt/blk_gp2_5g` باید به مسیر `root` وب‌سرویس متصل گردد
  - o این `container` پس از هر اختلالی که منجر به `termination` اپلیکیشن شود باید به صورت خودکار مجدد راه‌اندازی شود.

این کانتینر باید هنگام نمره‌دهی `up/running` باشد و همچنین فایل `docker-compose` آن به درستی قابل اجرا باشد.

## سرور 2 webserv\_2

- یک `container` برای سرویس `Prometheus` با مشخصات زیر ایجاد کنید:
  - o **نام کانتینر:** `monitoring_prometheus`
  - o **ایمیج:** `prom/prometheus`
  - o **شماره پورت:** `9090`
  - o `Prometheus` باید طوری پیگیربندی شود تا متریک‌های `node_exporter` را `scrape` کند.
- یک `container` برای سرویس `Grafana` با مشخصات زیر ایجاد کنید:
  - o **نام کانتینر:** `monitoring_grafana`
  - o **ایمیج:** `grafana/grafana`
  - o **رمز عبور ادمین:** `Skills53`
  - o **شماره پورت:** `3000`
- یک `container` برای `node_exporter` با مشخصات زیر ایجاد کنید:
  - o **نام کانتینر:** `monitoring_node_exporter`
  - o **ایمیج:** `prom/node-exporter`

- یک داشبورد با نام "node exporter" در grafana ایجاد کنید.
- یک گراف با نام "system load" در داشبورد "node exporter" ایجاد کنید و در آن گراف system load را با تفکیک 1m و 5m و 15m به نمایش قرار دهید.

## سرویس Elastic Load Balancer

### پنل زیرساخت ابری

داخل Region خود یک Load Balancer با مشخصات زیر ایجاد کنید تا ترافیک ورودی را، بین دو وب سرور webserv\_1 و webserv\_2 پخش کند:

- پورت Load Balancer را روی 80 تنظیم کنید.
  - مدت زمانی که برای پاسخ هر probe می توان صبر کرد را ۴ ثانیه تنظیم کنید.
  - تعداد دفعاتی که لازم است یک target پاسخ صحیح دهد تا به عنوان نود سالم لحاظ شود را برابر ۱۰ تنظیم نمایید.
- پس از راه اندازی سرویس ELB، شما باید بتوانید از طریق اینترنت به endpoint لودبالانسر متصل شوید و محتوای وب webserv\_1 و webserv\_2 را ببینید. همچنین هدر مربوط به X-Served-By مربوط به هر سرور در response header برگردانده شود.