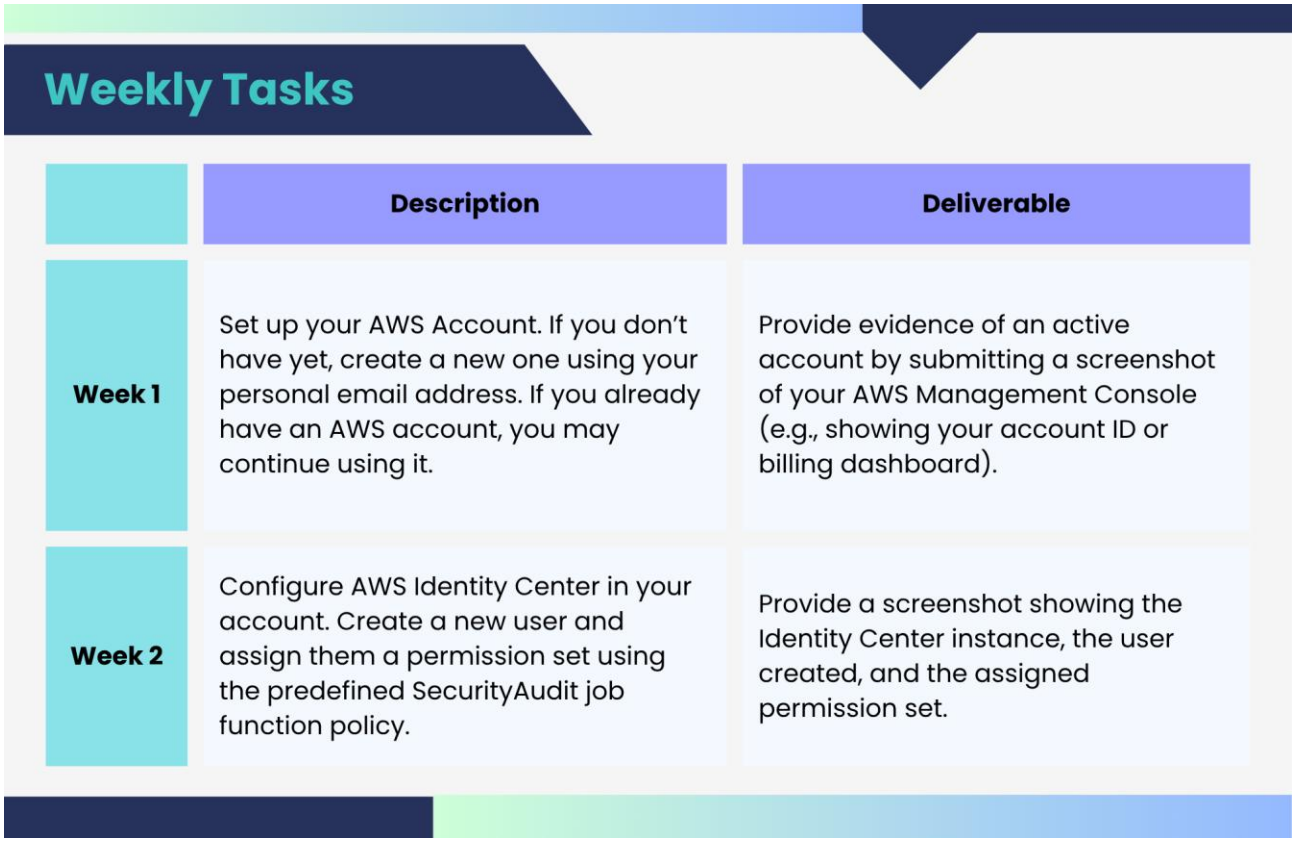
Weekly Tasks



Weekly Tasks

	Description	Deliverable
Week 3	Launch a new Windows Server EC2 instance using the Amazon Windows Server 2019 Base AMI or a later version. Ensure the instance is deployed in a public subnet and secured with a Security Group that permits RDP access (port 3389) only from your public IP address. Assign the instance a Name tag with the value: **CSN-Bootcamp-Week3**.	Provide screenshots of your running Windows EC2 instance in the AWS Management Console. The screenshots should clearly show the instance's Name tag set to CSN-Bootcamp-Week3, the Security Group inbound rules allowing RDP (port 3389) only from your public IP address, and a successful Remote Desktop (RDP) connection to the instance.

Weekly Tasks

	Description	Deliverable
Week 4	In this task, you'll create two separate VPCs in your AWS environment to simulate a multi-VPC architecture. Configure VPC-A with the CIDR block 10.10.0.0/16 and VPC-B with 10.20.0.0/16. In each VPC, create one public subnet and one private subnet (you don't need to attach Internet or NAT Gateways). Then, establish a VPC peering connection between the two VPCs and update the route tables in each VPC to allow traffic to flow between them via the peering connection.	Provide clear screenshots from the AWS Console showing: the two VPCs and their CIDR blocks, the public and private subnets in each VPC, the VPC peering connection in Active status, and the route tables with routes pointing to the other VPC's CIDR block through the peering connection.

Weekly Tasks

	Description	Deliverable
Week 5	For Week 5, your task is to deploy Grafana using Amazon ECS with Fargate. Grafana is a popular open-source dashboard tool that runs on port 3000. You'll use the official Docker image *grafana/grafana*, create a task definition that exposes port 3000, and run it in a public subnet. Ensure your security group allows inbound traffic on port 3000. Once the service/task is running, open **http://<PUBLIC-IP>:3000** in your browser.	Submit screenshots showing your ECS cluster and running service, the task definition with the grafana/grafana image, and the security group rule allowing port 3000 access. Finally, include a screenshot of the Grafana login page in your browser Grafana's default login credentials: • **Username:** admin • **Password:** admin

Weekly Tasks

	Description	Deliverable
Week 6	In this task, you will deploy Metabase on Amazon ECS using the Fargate launch type and connect it to a PostgreSQL database hosted on Amazon RDS. You'll use the official Metabase Docker image from Docker Hub and configure the environment variables needed for database connectivity. Ensure both ECS and RDS are in the same VPC to allow communication. The RDS security group should allow inbound traffic on port 5432 from the ECS task.	Provide clear screenshots showing the RDS PostgreSQL instance details, the ECS task definition and running service, the security group rules allowing traffic from ECS to RDS on port 5432, and the Metabase setup screen confirming a successful connection to the database.

Weekly Tasks

	Description	Deliverable
Week 7	This week, your task is to deploy a simple containerized application (e.g., Nginx or grafana/grafana) on Amazon ECS with Fargate, using appropriate CPU and memory settings. Once deployed in a public subnet, create a CloudWatch dashboard with widgets for CPUUtilization and MemoryUtilization to monitor the ECS task in real time. For deeper insight, simulate load by refreshing or stress testing the app and observe how the metrics change.	Submit screenshots showing your ECS cluster and running service, the task definition with CPU and memory settings, and your **CloudWatch dashboard with** CPU and memory widgets. Optionally, include a screenshot showing changes in metrics under simulated load.

Weekly Tasks

	Description	Deliverable
Week 8	This week's task is to deploy a personal static website such as a **resume**, **portfolio**, or "**About Me**" page using Amazon S3 and CloudFront. You will upload your HTML files to an S3 bucket, enable static website hosting, and configure the bucket for public access by adding the appropriate bucket policy. After that, you will set up a CloudFront distribution that uses the S3 bucket as its origin to securely deliver your content.	Provide screenshots of your S3 bucket configuration showing that static website hosting is enabled, along with the bucket policy used to allow public access. Include a screenshot of your CloudFront distribution settings and its deployment status. Lastly, provide a screenshot of your live website as accessed through the CloudFront domain.

Weekly Tasks

	Description	Deliverable
Week 9	Register a free domain on **freenom.com** and connect it to AWS Route 53 by updating the nameservers with those from your **hosted zone**. Create a CloudFront distribution that serves your S3 static website, then request and validate a **free SSL certificate** using ACM with DNS validation in Route 53. Attach the certificate to CloudFront and confirm your site loads over **HTTPS** using your custom domain. If you cannot use a domain, follow the same steps and explain what you would have done.	Provide screenshots of your Route 53 hosted zone, the validated SSL certificate in AWS Certificate Manager, CloudFront settings with your custom domain, and your website loading over HTTPS. If you simulated the steps, briefly describe your approach.

Weekly Tasks

	Description	Deliverable
Week 10	You are required to create an S3 bucket that will automatically trigger a Lambda function whenever a new file is uploaded to the bucket. The Lambda function should respond to the event by doing something simple. For example, it can print the name of the uploaded file, write a message to CloudWatch Logs, or send an email using Amazon SES to confirm the upload.	Provide screenshots of your S3 bucket showing the event trigger connected to your Lambda function. Also include a screenshot of your Lambda function code or settings. Finally, show proof that the function worked, like a log in CloudWatch or the email you received from SES.