Welcome to Your Team's Cloud Environment

This guide will help you access the cloud resources created for your class with **Katerina Doka**. By following the steps below, you will set up the necessary tools, download your private key securely, and connect to your team's virtual machines (VMs).

Understanding Your Team's Setup

Each team has been assigned a dedicated environment consisting of the following:

- **Bastion Host (Entry Point)**
  - This is your gateway into the cloud environment.
  - It is the only machine with a **public IP address**, which allows you to connect from your personal computer.
  - The **PublicIP** of the bastion host is listed in `teams_ips.csv`.
- **Private VMs (Team Machines)**
  - Each team has **four private VMs**, which are your working machines.
  - These machines **do not have public IPs** and can only be accessed after logging into the bastion host.
  - Their **private IPs** are listed in `teams_ips.csv` under the **PrivateIps** column (comma-separated).
  - The private VMs are **t3.large EC2 instances**, which means:
    - **2 vCPUs** (virtual processors)
    - **8GB RAM**
    - **100GB of storage**
- **Private Key Storage**
  - Your private SSH keys, which allow you to connect to your machines, are stored securely in an **AWS S3 bucket**.
  - Only your team has access to these keys, ensuring security and isolation.
- **Team Isolation**
  - Each team operates in its own private space, ensuring security and preventing interference between teams.
  - Access is managed through AWS Identity and Access Management (IAM) and SSH keys.

Step 1: Setting Up the AWS CLI

Why This Is Important

The AWS Command Line Interface (CLI) allows you to interact securely with AWS services. You will use it to download your private key from AWS S3, which is necessary to connect to your virtual machines.

Installation

macOS

```
brew install awscli
```

Windows

```
msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
aws --version  # Verify installation
```

Linux (Ubuntu/Debian)

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscl
unzip awscliv2.zip
sudo ./aws/install
aws --version  # Verify installation
```

Configure Your AWS Credentials

You will receive an **AWS Access Key ID** and **Secret Access Key** in `team_credentials.csv`.
Configure your AWS CLI using these credentials:

```
aws configure
```

When prompted, enter:

- **AWS Access Key ID**: `<your_access_key_id>`
- **AWS Secret Access Key**: `<your_secret_access_key>`
- **Default region name**: `eu-central-1`

- **Default output format**: `json`

What You Gain: Secure access to AWS services, including downloading your SSH key from S3.

Step 2: Downloading Your Private SSH Key

Why This Is Important

Your private SSH key allows you to log in to the bastion host and private VMs securely. This key is stored in an S3 bucket and must be downloaded to your local machine.

Download and Secure the Key

Replace `TEAM_NAME` with your actual team name (e.g., `team-01`):

```
aws s3 cp s3://PROJECT_NAME-HASH/TEAM_NAME/private_key.pem ~/.ssh/team_ke
chmod 600 ~/.ssh/team_key.pem
```

What You Gain: Your private SSH key is now securely stored and ready to be used for logging in.

Step 3: Connecting to Your VMs

Understanding the Connection Process

1. **Connect to the Bastion Host** (the only machine with a public IP).
2. **From the Bastion Host, connect to your team's private VMs** using their private IP addresses.

Connect to the Bastion Host

Find the **PublicIP** in `teams_ips.csv` and connect using SSH:

```
ssh -i ~/.ssh/team_key.pem ubuntu@<PublicIP>
```

What You Gain: You are now inside the cloud environment and can access your team's private VMs.

Access a Private VM from the Bastion Host

Once inside the bastion host, use the **PrivateIps** column in `teams_ips.csv` to connect to a private VM:

```
ssh -i ~/.ssh/team_key.pem ubuntu@<PrivateIP>
```

What You Gain: You now have full access to your team's VMs for running applications and development work.

Step 4: Configuring SSH for Easier Logins

Why This Helps

1. **SSH Agent**: Storing your SSH key in memory avoids having to specify the key file each time you connect.
2. **SSH Config File**: You can create shortcuts (aliases) for hosts so you don't have to remember IP addresses and usernames for each VM.

Using the SSH Agent

1. **Start the SSH Agent**

```
eval "$(ssh-agent -s)"
```

2. **Add Your SSH Key to the Agent**

```
ssh-add ~/.ssh/team_key.pem
```

3. **Check Your Loaded Keys**

```
ssh-add -l
```

What You Gain: The SSH agent keeps your key in memory, so you don't need to repeatedly type the path to your key.

Creating an SSH Config for Multiple Hosts

In addition to using the agent, you can simplify logins further by creating a configuration file in `~/.ssh/config`. This file stores information for each host, letting you use simple aliases instead of IP addresses and manual SSH options. Here's how:

1. **Create or Edit the Config File**

```
touch ~/.ssh/config
chmod 600 ~/.ssh/config
nano ~/.ssh/config
```

2. **Add Entries for the Bastion and Private VMs**

```
# Example SSH config entries

Host my-bastion
    HostName <PublicIP>
    User ubuntu
    IdentityFile ~/.ssh/team_key.pem

Host my-vm1
    HostName <PrivateIP_of_VM1>
    User ubuntu
    IdentityFile ~/.ssh/team_key.pem
    ProxyJump my-bastion

Host my-vm2
    HostName <PrivateIP_of_VM2>
    User ubuntu
    IdentityFile ~/.ssh/team_key.pem
    ProxyJump my-bastion

# Add more VMs as needed...
```

- **my-bastion**: An alias for the bastion host. Replace `<PublicIP>` with the bastion's public IP.
- **my-vm1**: An alias for a private VM. Replace `<PrivateIP_of_VM1>` with your VM's private IP.
- **ProxyJump my-bastion**: Tells SSH to connect to `my-vm1` by first going through `my-bastion`.

3. **Use Your Aliases to SSH**

```
# SSH into the bastion host
ssh my-bastion
```

```
# SSH directly into a private VM (it will automatically jump via the bast
ssh my-vm1
```

What You Gain:

- **Shorter Commands**: Just type `ssh my-bastion` or `ssh my-vm1`.
- **No Need to Specify Keys**: Your `IdentityFile` is already in the config.
- **Automatic Bastion Usage**: `ProxyJump` seamlessly routes you through the bastion.

Summary of Your Workflow

1. **Install and Configure AWS CLI**: Securely authenticate with AWS to access your team's resources.
2. **Download Your Private SSH Key**: Retrieve your key from S3 and secure it on your machine.
3. **Connect via the Bastion Host**: Use the **PublicIP** to enter the environment, then use **PrivateIps** to access your VMs.
4. **Set Up SSH Agent**: Simplify login by managing your SSH keys efficiently.

By following these steps, you will have secure and efficient access to your team's environment. If you have any questions or need assistance, please reach out to **Elektra**, the solution architect at **PCG**, by emailing **elektra.bilali@pcg.io**.

Happy coding!